**NuMicro® Family**

**1T 8051-based Microcontroller**

**MUG51TB9AE**

**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 and microprocessor 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](http://www.nuvoton.com)

TABLE OF CONTENTS

[1 GENERAL DESCRIPTION 9](#_Toc119664441)

[1.1 Key Features and Application 9](#_Toc119664442)

[2 FEATURES 10](#_Toc119664443)

[3 PART INFORMATION 14](#_Toc119664444)

[3.1 MUG51TBAE Package Type 14](#_Toc119664445)

[3.2 MUG51TBAE Naming Rule 15](#_Toc119664446)

[3.3 MUG51TBAE Selection Guide 16](#_Toc119664447)

[4 PIN CONFIGURATION 17](#_Toc119664448)

[4.1 Pin Configuration 17](#_Toc119664449)

[4.1.1 MUG51TBAE Pin Diagram 17](#_Toc119664450)

[4.1.2 MUG51TBAE Multi-function Pin Diagram 17](#_Toc119664451)

[4.2 Pin Description 20](#_Toc119664452)

[4.2.1 MUG51TBAE Pin Mapping 20](#_Toc119664453)

[4.2.2 MUG51TBAE Pin Functional Description 21](#_Toc119664454)

[5 BLOCK DIAGRAM 23](#_Toc119664455)

[5.1 MUG51TBAE Full Function Block 23](#_Toc119664456)

[6 FUNCTIONAL DESCRIPTION 24](#_Toc119664457)

[6.1 Memory Organization 24](#_Toc119664458)

[6.1.1 Program Memory 24](#_Toc119664459)

[6.1.2 Security Protection Memory (SPROM) 25](#_Toc119664460)

[6.1.3 96-Bit Unique Code (UID) 26](#_Toc119664461)

[6.1.4 Data Flash 26](#_Toc119664462)

[6.1.5 Data Memory 26](#_Toc119664463)

[6.1.6 Config Bytes 29](#_Toc119664464)

[6.1.7 Special Function Register (SFR) 35](#_Toc119664465)

[6.2 System Manager 203](#_Toc119664466)

[6.2.1 Clock System 203](#_Toc119664467)

[6.2.2 Power Management 217](#_Toc119664468)

[6.2.3 Reset 221](#_Toc119664469)

[6.2.4 Interrupt System 237](#_Toc119664470)

[6.3 Flash Memory Control 256](#_Toc119664471)

[6.3.1 In-application-programming (IAP) 256](#_Toc119664472)

[6.3.2 In-Circuit-Programming (ICP) 267](#_Toc119664473)

[6.3.3 On-Chip-Debugger (ICE) 267](#_Toc119664474)

[6.4 GPIO Port Structure and Operation 270](#_Toc119664475)

[6.4.1 GPIO Mode 270](#_Toc119664476)

[6.4.2 External Interrupt Pins 336](#_Toc119664477)

[6.4.3 Pin Interrupt (PIT) 337](#_Toc119664478)

[6.5 Timer 344](#_Toc119664479)

[6.5.1 Overview 344](#_Toc119664480)

[6.5.2 Features 344](#_Toc119664481)

[6.5.3 Timer/Counter 0 and 1 344](#_Toc119664482)

[6.5.4 Timer 2 and Input Capture 356](#_Toc119664483)

[6.5.5 Timer 3 370](#_Toc119664484)

[6.6 Watchdog Timer (WDT) 374](#_Toc119664485)

[6.6.1 Time-Out Reset Timer 374](#_Toc119664486)

[6.6.2 General Purpose Timer 375](#_Toc119664487)

[6.6.3 Register Description 376](#_Toc119664488)

[6.6.4 Typical Structure of WDT Service Routine 380](#_Toc119664489)

[6.7 Self Wake-up Timer (WKT) 381](#_Toc119664490)

[6.7.1 Overview 381](#_Toc119664491)

[6.7.2 Block Diagram 381](#_Toc119664492)

[6.7.3 Register Description 382](#_Toc119664493)

[6.8 Pulse Width Modulated (PWM) 388](#_Toc119664494)

[6.8.1 Overview 388](#_Toc119664495)

[6.8.2 Features 388](#_Toc119664496)

[6.8.3 Block Diagram 389](#_Toc119664497)

[6.8.4 Functional Description 390](#_Toc119664498)

[6.8.5 Register Description 396](#_Toc119664499)

[6.9 Serial Port (UART0 & UART1) 410](#_Toc119664500)

[6.9.1 Overview 410](#_Toc119664501)

[6.9.2 Features 410](#_Toc119664502)

[6.9.3 Functional Description 411](#_Toc119664503)

[6.9.4 Register Description 420](#_Toc119664504)

[6.10 Smart Card Interface (SC) 437](#_Toc119664505)

[6.10.1 Overview 437](#_Toc119664506)

[6.10.2 Features 437](#_Toc119664507)

[6.10.3 Block Diagram 437](#_Toc119664508)

[6.10.4 Operating Modes 438](#_Toc119664509)

[6.10.5 Smart Card Data Transfer 440](#_Toc119664510)

[6.10.6 Register Description 443](#_Toc119664511)

[6.11 Serial Peripheral Interface (SPI) 455](#_Toc119664512)

[6.11.1 Overview 455](#_Toc119664513)

[6.11.2 Features 455](#_Toc119664514)

[6.11.3 Block Diagram 456](#_Toc119664515)

[6.11.4 Functional Description 457](#_Toc119664516)

[6.11.5 Register Description 464](#_Toc119664517)

[6.12 Inter-Integrated Circuit (I2C) 469](#_Toc119664518)

[6.12.1 Overview 469](#_Toc119664519)

[6.12.2 Features 469](#_Toc119664520)

[6.12.3 Functional Description 469](#_Toc119664521)

[6.12.4 Register Description 481](#_Toc119664522)

[6.12.5 Typical Structure of I2C Interrupt Service Routine 489](#_Toc119664523)

[6.13 Analog Comparator Controller (ACMP) 493](#_Toc119664524)

[6.13.1 Overview 493](#_Toc119664525)

[6.13.2 Features 493](#_Toc119664526)

[6.13.3 Block Diagram 494](#_Toc119664527)

[6.13.4 Functional Description 495](#_Toc119664528)

[6.13.5 Register Description 497](#_Toc119664529)

[6.14 PDMA Controller (PDMA) 503](#_Toc119664530)

[6.14.1 Overview 503](#_Toc119664531)

[6.14.2 Features 503](#_Toc119664532)

[6.14.3 Block Diagram 503](#_Toc119664533)

[6.14.4 Functional Description 504](#_Toc119664534)

[6.14.5 Register Description 507](#_Toc119664535)

[6.15 Instruction Set 518](#_Toc119664536)

[6.15.1 Instruction Set And Addressing Modes 518](#_Toc119664537)

[6.15.2 Read-Modify-Write Instructions 520](#_Toc119664538)

[6.15.3 Instruction Set 520](#_Toc119664539)

[7 APPLICATION CIRCUIT 524](#_Toc119664540)

[7.1 Power Supply Scheme 524](#_Toc119664541)

[7.2 Peripheral Application Scheme 525](#_Toc119664542)

[8 ELECTRICAL CHARACTERISTICS 526](#_Toc119664543)

[9 PACKAGE DIMENSIONS 527](#_Toc119664544)

[9.1 QFN 33-pin (4.0 x 4.0 x 0.8 mm) 528](#_Toc119664545)

[10 ABBREVIATIONS 529](#_Toc119664546)

[10.1 Abbreviations 530](#_Toc119664547)

[11 REVISION HISTORY 531](#_Toc119664548)

List of Figures

[Figure 4.1‑1 MUG51TB9AE Pin Assignment 17](#_Toc119664358)

[Figure 4.1‑2 MUG51TB9AE Multi-function Pin Assignment 18](#_Toc119664359)

[Figure 5.1‑1 Functional Block Diagram 23](#_Toc119664360)

[Figure 6.1‑1 MUG51TBAE Program Memory Map 25](#_Toc119664361)

[Figure 6.1‑2 SPROM Memory Mapping And SPROM Security Mode 26](#_Toc119664362)

[Figure 6.1‑3 Data Memory Map 27](#_Toc119664363)

[Figure 6.1‑4 Internal 256 Bytes RAM Addressing 28](#_Toc119664364)

[Figure 6.1‑5 CONFIG0 Any Reset Reloading 30](#_Toc119664365)

[Figure 6.1‑6 CONFIG2 Power-On Reset Reloading 32](#_Toc119664366)

[Figure 6.2‑1 Clock System Block Diagram 203](#_Toc119664367)

[Figure 6.2‑2 Brown-out Detection Block Diagram 223](#_Toc119664368)

[Figure 6.2‑3 Boot Selecting Diagram 235](#_Toc119664369)

[Figure 6.3‑1. CRC-8 Block Diagram 266](#_Toc119664370)

[Figure 6.4‑1 Quasi-Bidirectional Mode Structure 271](#_Toc119664371)

[Figure 6.4‑2 Push-Pull Mode Structure 271](#_Toc119664372)

[Figure 6.4‑3 Input-Only Mode Structure 272](#_Toc119664373)

[Figure 6.4‑4 Open-Drain Mode Structure 272](#_Toc119664374)

[Figure 6.4‑5 GPIO Auto Return Timing And Setting 333](#_Toc119664375)

[Figure 6.4‑6 GPIO Internal Inverter And Buffer Structure 335](#_Toc119664376)

[Figure 6.4‑7 GPIO Internal Invert Active Application 335](#_Toc119664377)

[Figure 6.4‑8 GPIO Internal Buffer Application Example 336](#_Toc119664378)

[Figure 6.4‑9 Pin Interface Block Diagram 338](#_Toc119664379)

[Figure 6.5‑1 Timer/Counters 0 and 1 in Mode 0 345](#_Toc119664380)

[Figure 6.5‑2 Timer/Counters 0 and 1 in Mode 1 345](#_Toc119664381)

[Figure 6.5‑3 Timer/Counters 0 and 1 in Mode 2 346](#_Toc119664382)

[Figure 6.5‑4 Timer/Counter 0 in Mode 3 346](#_Toc119664383)

[Figure 6.5‑5 Timer 2 Compare Mode Functional Block Diagram 356](#_Toc119664384)

[Figure 6.5‑6 Timer 2 Input Capture Module Functional Block Diagram 357](#_Toc119664385)

[Figure 6.5‑7 Timer 3 Block Diagram 370](#_Toc119664386)

[Figure 6.6‑1 WDT as A Time-Out Reset Timer 375](#_Toc119664387)

[Figure 6.6‑2 Watchdog Timer Block Diagram 375](#_Toc119664388)

[Figure 6.7‑1 Self Wake-Up Timer (WKT) Block Diagram 381](#_Toc119664389)

[Figure 6.8‑1 PWM0 Block Diagram 389](#_Toc119664390)

[Figure 6.8‑2 PWM0 and Fault Brake Output Control Block Diagram 391](#_Toc119664391)

[Figure 6.8‑3 PWM Edge-aligned Type Waveform 392](#_Toc119664392)

[Figure 6.8‑4 PWM Center-aligned Type Waveform 393](#_Toc119664393)

[Figure 6.8‑5 PWM Complementary Mode with Dead-time Insertion 394](#_Toc119664394)

[Figure 6.8‑6 Fault Brake Function Block Diagram 394](#_Toc119664395)

[Figure 6.8‑7 PWM Interrupt Type 395](#_Toc119664396)

[Figure 6.9‑1 Serial Port Mode 0 Timing Diagram 411](#_Toc119664397)

[Figure 6.9‑2 Serial Port Mode 1 Timing Diagram 412](#_Toc119664398)

[Figure 6.9‑3 Serial Port Mode 2 and 3 Timing Diagram 413](#_Toc119664399)

[Figure 6.10‑1 Figure SC Controller Block Diagram 437](#_Toc119664400)

[Figure 6.10‑2 SC Interface Connection 438](#_Toc119664401)

[Figure 6.10‑3 SC Activation and Cold Reset Sequence 439](#_Toc119664402)

[Figure 6.10‑4 SC Warm Reset Sequence 439](#_Toc119664403)

[Figure 6.10‑5 SC Deactivation Sequence 440](#_Toc119664404)

[Figure 6.10‑6 SC Data Character 440](#_Toc119664405)

[Figure 6.10‑7 Initial Character TS 441](#_Toc119664406)

[Figure 6.10‑8 SC Error Signal 441](#_Toc119664407)

[Figure 6.10‑9 Transmit Direction Block Guard Time Operation 442](#_Toc119664408)

[Figure 6.10‑10 Receive Direction Block Guard Time Operation 442](#_Toc119664409)

[Figure 6.10‑11 Extra Guard Time Operation 442](#_Toc119664410)

[Figure 6.11‑1 SPI Block Diagram 456](#_Toc119664411)

[Figure 6.11‑2 SPI Multi-Master, Multi-Slave Interconnection 457](#_Toc119664412)

[Figure 6.11‑3 SPI Single-Master / Single-Slave Interconnection 458](#_Toc119664413)

[Figure 6.11‑4 SPI Clock Formats 460](#_Toc119664414)

[Figure 6.11‑5 SPI Clock and Data Format with CPHA = 0 461](#_Toc119664415)

[Figure 6.11‑6 SPI Clock and Data Format with CPHA = 1 461](#_Toc119664416)

[Figure 6.11‑7 SPI Overrun Waveform 463](#_Toc119664417)

[Figure 6.11‑8 SPI Interrupt Request 463](#_Toc119664418)

[Figure 6.12‑1 I2C Bus Interconnection 469](#_Toc119664419)

[Figure 6.12‑2 I2C Bus Protocol 470](#_Toc119664420)

[Figure 6.12‑3 START, Repeated START, and STOP Conditions 470](#_Toc119664421)

[Figure 6.12‑4 Master Transmits Data to Slave by 7-bit 471](#_Toc119664422)

[Figure 6.12‑5 Master Reads Data from Slave by 7-bit 471](#_Toc119664423)

[Figure 6.12‑6 Data Format of One I2C Transfer 471](#_Toc119664424)

[Figure 6.12‑7 Acknowledge Bit 472](#_Toc119664425)

[Figure 6.12‑8 Arbitration Procedure of Two Masters 473](#_Toc119664426)

[Figure 6.12‑9 Control I2C Bus According To The Current I2C Status 473](#_Toc119664427)

[Figure 6.12‑10 Flow and Status of Master Transmitter Mode 474](#_Toc119664428)

[Figure 6.12‑11 Flow and Status of Master Receiver Mode 475](#_Toc119664429)

[Figure 6.12‑12 Flow and Status of Slave Receiver Mode 477](#_Toc119664430)

[Figure 6.12‑13 Flow and Status of General Call Mode 478](#_Toc119664431)

[Figure 6.12‑14 I2C Time-Out Counter 480](#_Toc119664432)

[Figure 6.13‑1 Analog Comparator Block Diagram 494](#_Toc119664433)

[Figure 6.13‑2 Comparator Hysteresis Function 495](#_Toc119664434)

[Figure 6.13‑3 Comparator Reference Voltage Block Diagram 496](#_Toc119664435)

[Figure 6.13‑4 Analog Comparator Interrupt Sources 496](#_Toc119664436)

[Figure 6.14‑1 PDMA Interface Diagram 503](#_Toc119664437)

[Figure 6.14‑2 PDMA Controller Block Diagram 504](#_Toc119664438)

[Figure 6.14‑3 CRC-8 Block Diagram 506](#_Toc119664439)

[Figure 9.1‑1 QFN-33 Package Dimension 528](#_Toc119664440)

**List of Tables**

[Table 1.1‑1 MUG51TB9AE Key Features Support Table 9](#_Toc119664336)

[Table 6.1‑1 Special Function Register Map Table 36](#_Toc119664337)

[Table 6.2‑1 Power Mode and Clock Source 217](#_Toc119664338)

[Table 6.2‑2 Entry Setting of Power-down Mode 218](#_Toc119664339)

[Table 6.2‑3 BOF Reset Value 223](#_Toc119664340)

[Table 6.2‑4 Interrupt Vectors 238](#_Toc119664341)

[Table 6.2‑5 Interrupt Priority Level Setting 244](#_Toc119664342)

[Table 6.2‑6 Characteristics of Each Interrupt Source 245](#_Toc119664343)

[Table 6.3‑1 IAP Modes and Command Codes 265](#_Toc119664344)

[Table 6.4‑1 Configuration for Different I/O Modes 270](#_Toc119664345)

[Table 6.6‑1 Watchdog Timer-out Interval Under Different Pre-scalars 374](#_Toc119664346)

[Table 6.9‑1 Serial Port 0 Mode / Baud Rate Description 414](#_Toc119664347)

[Table 6.9‑2 Serial Port 1 Mode / Baud Rate Description 415](#_Toc119664348)

[Table 6.9‑3 Timer Define Value for Baud Rate 416](#_Toc119664349)

[Table 6.11‑1 SPI Master Clock Rate Define Table 459](#_Toc119664350)

[Table 6.11‑2 SPI Clock Suspend Interval Select 459](#_Toc119664351)

[Table 6.11‑3 Slave Select Pin Configurations 462](#_Toc119664352)

[Table 6.12‑1 Status Display In I2STAT Register 479](#_Toc119664353)

[Table 6.15‑1 Instruction Set And Addressing Modes 518](#_Toc119664354)

[Table 6.15‑2 Instructions Affect Flag Settings 519](#_Toc119664355)

[Table 6.15‑3 Instruction Set 523](#_Toc119664356)

[Table 10.1‑1 List of Abbreviations 530](#_Toc119664357)

# GENERAL DESCRIPTION

MUG51TB9AE is a Flash embedded 1T 8051-based low-power microcontroller. It runs up to 7.3728 MHz with 16 Kbytes embedded Flash memory, 1 Kbytes embedded SRAM, 4 Kbytes Flash loader memory (LDROM), 1.8V ~ 5.5V operating voltage, and -40°C ~105°C operating temperature. MUG51TB9AE supports enhanced low current consumption at 200 μA while CPU power-on before Flash memory is initialized. Its low-power feature makes it suitable for battery-free device which harvests power from the magnetic field of coil such as stylus pen powered by EMR (Electro-magnetic Resonance) technology and RFID card.

The MUG51TB9AE features low current consumption at 200 μA while CPU power-on before Flash memory is initialized. It is suitable for battery-free devices such as stylus pen powered by EMR (Electro-magnetic Resonance) technology and RFID card. The current consumption is less than 1.3 mA in normal run mode at 7.3728 MHz, and less than 1 μA in Power-down mode.

The MUG51TB9AE provides rich peripherals including 24 general purpose I/Os with internal inverter, four 16-bit Timers/Counters, 2 sets of UARTs with frame error detection and automatic address recognition, 1 set of ISO7816 Smartcard interface, 1 set of SPI, 2 sets of I2C, 6 enhanced PWM output channels with dead zone control, 2 sets of analog comparators, eight-channel shared pin interrupt for all I/O ports, low voltage reset (LVR) and brown-out detector (BOD) to enhance product performance, reduce external components and form factor simultaneously.

The MUG51TB9AE includes the QFN33 (4mm x 4mm) package.

## Key Features and Application

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Product Line** | **UART** | **ISO 7816-3** | **I2C** | **SPI** | **Timer** | **PWM** | **PDMA** | **ACMP** |
| MUG51TB9AE | 2 | 1 | 2 | 1 | 4 | 6 | 2 | 2 |

Table 1.1‑1 MUG51TB9AE Key Features Support Table

The MUG51TB9AE is suitable for a wide range of applications such as:

* + - Stylus pen
    - RFID card

# FEATURES

|  |  |  |  |
| --- | --- | --- | --- |
| ***Core and System*** | | | |
| **8051** | | * Fully static design 8-bit high performance 1T 8051-based CMOS microcontroller * Instruction set fully compatible with MCS-51 * 4-priority-level interrupts capability * Dual Data Pointers (DPTRs) | |
| **Power on Reset (POR)** | | * POR with 1.55V threshold voltage level | |
| **Brown-out Detector (BOD)** | | * 7-level selection, with brown-out interrupt and reset option. (4.4V / 3.7V / 3.0V / 2.7V / 2.4V / 2.0V / 1.8V) | |
| **Low Voltage Reset (LVR)** | | * LVR with 1.7V threshold voltage level | |
| **Security** | | * 96-bit Unique ID (UID) * 128-bit Unique Customer ID (UCID) * 128-bytes security protection memory SPROM | |
| ***Memories*** | | | |
| **Flash** | | * Up to 16 Kbytes of APROM for User Code * 4/3/2/1 Kbytes of Flash for loader (LDROM) configure from APROM for In-System-Programmable (ISP) * Flash Memory accumulated with pages of 128 Bytes from APROM by In-Application-Programmable (IAP) means whole APROM can be use as Data Flash * An additional 128 bytes security protection memory SPROM * Code lock for security by CONFIG | |
| **SRAM** | | * 256 Bytes on-chip RAM * Additional 1 KBytes on-chip auxiliary RAM (XRAM) accessed by MOVX instruction | |
| **PDMA** | | * Three modes: peripheral-to-memory, memory-to-peripheral, and memory-to-memory transfer * Source address and destination address must be word alignment in all modes. * Memory-to-memory mode: transfer length must be word alignment. | |
| ***Clocks*** | | | |
| **Internal Clock Source** | | * Default 7.3728 MHz median speed internal oscillator (MIRC) trimmed to ±1% (accuracy at 25 °C, 3.3 V), ±2% in 0~70°C * 38.4 kHz low-speed internal oscillator (LIRC) calibrating to ±2% by software from median speed internal oscillator | |
| ***Timers*** | | | |
| **16-bit Timer** | | * Two 16-bit Timers/Counters 0 and 1 compatible with standard 8051 * One 16-bit Timer 2 with three-channel input capture module with GPIO or ACMP output as input source select * One 16-bit auto-reload Timer 3, which can be the baud rate clock source of UARTs | |
| **Watchdog** | | * 6-bit free running up counter for WDT time-out interval * Selectable time-out interval is 1.66 ms ~ 3413.12 ms since WDT\_CLK = 38.4 kHz (LIRC) * Able to wake up from Power-down or Idle mode * Interrupt or reset selectable on watchdog time-out | |
| **Wake-up Timer** | | * 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 6 output pins can be selected * Supports maximum clock source frequency up to FSYS * Supports up to three PWM modules; each module provides 6 output channels * Supports independent mode for PWM output * Supports complementary mode for 3 complementary paired PWM output channels * Dead-time insertion with 8-bit resolution * Supports 16-bit resolution PWM counter * Supports mask function and tri-state enable for each PWM pin * Supports brake function | |
| ***Analog Interfaces*** | | | |
| **Analog Comparator (ACMP)** | | * Supports up to two comparators: ACMP0, ACMP1 * Supports hysteresis function * Supports wake-up function * Supports Comparator Reference Voltage (CRV) * Selectable input sources of negative input * 4 positive sources from pins * 4 negative sources include pin, bandgap voltage, and CRV | |
| ***Communication Interfaces*** | | | |
| **UART** | | * Supports up to two UARTs: UART0, UART1 * Supports 1 Smart Card configuration as UART function as UART2 * UART baud rate clock from MIRC * Full-duplex asynchronous communications * Programmable 9th bit * TXD and RXD pins of UART0 exchangeable via software | |
| **I2C** | | * Two sets of I2C devices * Master/Slave mode * Bidirectional data transfer between masters and slaves * Multi-master bus (no central master) * 7-bit addressing mode * Standard mode (100 kbps) and Fast mode (400 kbps) * Supports 8-bit time-out counter requesting the I2C interrupt if the I2C bus hangs up and timer-out counter overflows * Multiple address recognition (four slave addresses with mask option) * Supports hold time programmable | |
| **SPI** | | * One set of SPI device * Supports Master or Slave mode operation * Supports MSB first or LSB first transfer sequence * Slave mode up to 4 MHz | |
| **ISO 7816-3** | | * One set of ISO 7816-3 device * Supports ISO 7816-3 compliant T=0, T=1 * Supports full-duplex UART mode | |
| **GPIO** | | * Four I/O modes:   + - * Quasi-bidirectional mode       * Push-Pull Output mode       * Open-Drain Output mode       * Input only with high impendence mode * Schmitt trigger input / TTL mode selectable * Supports internal inverter and buffer * Each I/O pin configured as interrupt source with edge/level trigger setting * Standard interrupt pins INT0 and INT1 * Supports high drive and high sink current I/O * I/O pin internal pull-up or pull-down resistor enabled in input mode * Maximum I/O speed is FSYS * Enabling the pin interrupt function will also enable the wake-up function * Supports 5V-tolerance function | |
| ***ESD & EFT*** | | | |
| **ESD** | | * HBM ± 7 kV | |
| **EFT** | | * > ± 4.4 kV | |
| **Latch-up** | | * 150 mA | |

# PART INFORMATION

## MUG51TBAE Package Type

Package is Halogen-free, RoHS-compliant and TSCA-compliant.

|  |  |
| --- | --- |
| **Part No.** | **QFN33 (4x4mm)** |
| **MUG51TBAE** | MUG51TB9AE |

## MUG51TBAE Naming Rule

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

## MUG51TBAE Selection Guide

|  |  |  |
| --- | --- | --- |
| **Part Number** | | **MUG51TBAE** |
| **TB9AE** |
| **Flash (KB)** | | 16 |
| **SRAM (KB)** | | 1 |
| **LDROM (KB)** | | 4 |
| **SPROM (Bytes)** | | 128 |
| **System Frequency (MHz)** | | 7.3728 |
| **PLL (MHz)** | | - |
| **I/O** | | 24 |
| **16-bit Timer** | | 4 |
| **Connectivity** | **UART** | 2 |
| **SPI** | 1 |
| **I²C** | 2 |
| **ISO 7816-3** | 1 |
| **BPWM** | | 6 |
| **PDMA** | | 2 |
| **CRC** | | CRC - 8 |
| **CRC- Configurable** | | - |
| **Analog Comparator** | | 2 |
| **12-bit DAC** | | - |
| **Temperature Sensor** | | - |
| **Internal Voltage Reference** | | - |
| **Package** | | QFN33 |
| **Note:**  1. ISP ROM programmable 1/2/3/4 KB Flash for user program loader (LDROM) shared from ARPOM.  2. ISO 7816-3 configurable as standard UART function. | | |

# PIN CONFIGURATION

## Pin Configuration

Users can find pin configuaration informations in chapter 4 or by using [NuTool - PinConfigure](http://www.nuvoton.com/opencms/resource-download.jsp?tp_GUID=SW1020150724174251). The NuTool - PinConfigure contains all Nuvoton NuMicro® Family chip series with all part number, and helps users configure GPIO multi-function correctly and handily.

### MUG51TBAE Pin Diagram

#### QFN33 Package

Corresponding Part Number: MUG51TB9AE

MUG51TB9AE

|  |
| --- |
|  |

Figure 4.1‑1 MUG51TB9AE Pin Assignment

### MUG51TBAE Multi-function Pin Diagram

#### QFN33 Package

|  |
| --- |
|  |

Figure 4.1‑2 MUG51TB9AE Multi-function Pin Assignment

| **Pin** | **MUG51TC9AE Pin Function** |
| --- | --- |
| 1 | P2.5 / ACMP0\_P0 / ACMP1\_P0 / PINV / I2C0\_SCL / PBUF / PWM0\_CH0 / UART2\_TXD / T0 / INT0 |
| 2 | P2.4 / ACMP0\_N0 / PINV / I2C0\_SDA / PBUF / PWM0\_CH1 / UART2\_RXD / T1 / INT1 |
| 3 | P2.3 / ACMP0\_P1 / ACMP1\_P1 / PINV / I2C1\_SCL / UART1\_TXD / PBUF / PWM0\_CH2 / PWM0\_BRAKE / T0 |
| 4 | P2.2 / ACMP1\_N0 / PINV / I2C1\_SDA / UART1\_RXD / PBUF / PWM0\_CH3 |
| 5 | P2.1 / ACMP0\_P2 / ACMP1\_P2 / PINV / PBUF / UART2\_TXD / I2C1\_SCL / PWM0\_CH4 / PWM0\_BRAKE |
| 6 | P2.0 / ACMP0\_N1 / UART2\_RXD / I2C1\_SDA / PWM0\_CH5 / PWM0\_BRAKE |
| 7 | P5.5 / PINV / UART2\_RXD / PWM0\_CH0 / PBUF / STADC |
| 8 | P5.4 / PINV / UART2\_TXD / PWM0\_CH1 / PBUF |
| 9 | P5.3 / PINV / UART0\_TXD / I2C0\_SCL / PBUF |
| 10 | P5.2 / PINV / UART0\_RXD / I2C0\_SDA / PBUF |
| 11 | P0.3 / PINV / PBUF / SPI0\_SS / UART1\_TXD / I2C1\_SCL / STADC / PWM0\_CH2 / CLKO |
| 12 | P0.2 / PINV / SPI0\_CLK / UART1\_RXD / I2C1\_SDA / PWM0\_CH3 / PBUF |
| 13 | P0.1 / PINV / SPI0\_MISO / UART2\_RXD / UART0\_TXD / PWM0\_CH4 / PBUF |
| 14 | P0.0 / SPI0\_MOSI / UART2\_TXD / UART0\_RXD / PWM0\_CH5 |
| 15 | NC |
| 16 | nRESET |
| 17 | P5.0 / UART1\_TXD / I2C1\_SCL / UART0\_TXD / ICE\_DAT |
| 18 | P5.1 / PINV / UART1\_RXD / I2C1\_SDA / UART0\_RXD / ICE\_CLK |
| 19 | P4.1 / PINV / UART2\_TXD / I2C0\_SCL / PBUF / ACMP0\_O |
| 20 | P4.0 / UART2\_RXD / I2C0\_SDA / ACMP1\_O / INT1 |
| 21 | NC |
| 22 | NC |
| 23 | P1.6 / PINV / UART0\_TXD / PBUF |
| 24 | P1.7 / PINV / UART0\_RXD / PBUF |
| 25 | VSS |
| 26 | LDO |
| 27 | VDD |
| 28 | P3.3 / PINV / PBUF / IC0 / PWM0\_BRAKE |
| 29 | P3.2 / ACMP1\_N1 / PINV / PBUF / IC1 / CLKO |
| 30 | P3.1 / ACMP0\_P3 / ACMP1\_P3 / PINV / UART0\_TXD / PBUF / IC2 |
| 31 | P3.0 / UART0\_RXD / IC0 |
| 32 | NC |

## Pin Description

### MUG51TBAE **Pin Mapping**

|  |  |
| --- | --- |
|  | **Pin Number** |
| **Pin Name** | **33** |
| P2.5 | 1 |
| P2.4 | 2 |
| P2.3 | 3 |
| P2.2 | 4 |
| P2.1 | 5 |
| P2.0 | 6 |
| P5.5 | 7 |
| P5.4 | 8 |
| P5.3 | 9 |
| P5.2 | 10 |
| P0.3 | 11 |
| P0.2 | 12 |
| P0.1 | 13 |
| P0.0 | 14 |
| NC | 15 |
| nRESET | 16 |
| P5.0 | 17 |
| P5.1 | 18 |
| P4.1 | 19 |
| P4.0 | 20 |
| NC | 21 |
| NC | 22 |
| P1.6 | 23 |
| P1.7 | 24 |
| VSS | 25 |
| LDO | 26 |
| VDD | 27 |
| P3.3 | 28 |
| P3.2 | 29 |
| P3.1 | 30 |
| P3.0 | 31 |
| NC | 32 |

### MUG51TBAE Pin Functional Description

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

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

| **Group** | **Pin Name** | **Type** | **Description** |
| --- | --- | --- | --- |
| ACMP0 | ACMP0\_N0 | A | Analog comparator 0 negative input 0 pin. |
| ACMP0\_N1 | Analog comparator 0 negative input 1 pin. |
| ACMP0\_O | O | Analog comparator 0 output pin. |
| ACMP0\_P0 | A | Analog comparator 0 positive input 0 pin. |
| ACMP0\_P1 | Analog comparator 0 positive input 1 pin. |
| ACMP0\_P2 | Analog comparator 0 positive input 2 pin. |
| ACMP0\_P3 | Analog comparator 0 positive input 3 pin. |
| ACMP1 | ACMP1\_N0 | A | Analog comparator 1 negative input 0 pin. |
| ACMP1\_N1 | Analog comparator 1 negative input 1 pin. |
| ACMP1\_O | O | Analog comparator 1 output pin. |
| ACMP1\_P0 | A | Analog comparator 1 positive input 0 pin. |
| ACMP1\_P1 | Analog comparator 1 positive input 1 pin. |
| ACMP1\_P2 | Analog comparator 1 positive input 2 pin. |
| ACMP1\_P3 | Analog comparator 1 positive input 3 pin. |
| CLKO | CLKO | O | Clock Out |
| I2C0 | I2C0\_SCL | I/O | I2C0 clock pin. |
| I2C0\_SDA | I/O | I2C0 data input/output pin. |
| I2C1 | I2C1\_SCL | I/O | I2C1 clock pin. |
| I2C1\_SDA | I/O | I2C1 data input/output pin. |
| IC0 | IC0 | I/O | Input Capture channel 0 |
| IC1 | IC1 | I/O | Input Capture channel 1 |
| IC2 | IC2 | I/O | Input Capture channel 2 |
| ICE | ICE\_CLK | I | Serial wired debugger clock pin.  **Note:** It is recommended to use 100 kΩ pull-up resistor on ICE\_CLK pin |
| ICE\_DAT | O | Serial wired debugger data pin.  **Note:** It is recommended to use 100 kΩ pull-up resistor on ICE\_DAT pin |
| INT0 | INT0 | I | External interrupt 0 input pin. |
| INT1 | INT1 | I | External interrupt 1 input pin. |
| nRESET | nRESET | I | External reset input: active LOW, with an internal pull-up. Set this pin low reset to initial state.  **Note:** It is recommended to use 10 kΩ pull-up resistor and 10 uF capacitor on nRESET pin. |
| PWM0 | PWM0\_BRAKE | I | PWM0 Brake input pin. |
| PWM0\_CH0 | I/O | PWM0 channel 0 output/capture input. |
| PWM0\_CH1 | I/O | PWM0 channel 1 output/capture input. |
| PWM0\_CH2 | I/O | PWM0 channel 2 output/capture input. |
| PWM0\_CH3 | I/O | PWM0 channel 3 output/capture input. |
| PWM0\_CH4 | I/O | PWM0 channel 4 output/capture input. |
| PWM0\_CH5 | I/O | PWM0 channel 5 output/capture input. |
| SPI0 | SPI0\_CLK | I/O | SPI0 serial clock pin. |
| SPI0\_MISO | I/O | SPI0 MISO (Master In, Slave Out) pin. |
| SPI0\_MOSI | I/O | SPI0 MOSI (Master Out, Slave In) pin. |
| SPI0\_SS | I/O | SPI0 slave select pin. |
| T0 | T0 | I/O | External count input to Timer/Counter 0 or its toggle output. |
| T1 | T1 | I/O | External count input to Timer/Counter 1 or its toggle output. |
| UART0 | UART0\_RXD | I | UART0 data receiver input pin. |
| UART0\_TXD | O | UART0 data transmitter output pin. |
| UART1 | UART1\_RXD | I | UART1 data receiver input pin. |
| UART1\_TXD | O | UART1 data transmitter output pin. |
| UART2 | UART2\_RXD | I | UART2 data receiver input pin. |
| UART2\_TXD | O | UART2 data transmitter output pin. |

# BLOCK DIAGRAM

## MUG51TBAE Full Function Block

|  |
| --- |
|  |

Figure 5.1‑1 Functional Block Diagram

# FUNCTIONAL DESCRIPTION

## Memory Organization

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

The Data Memory occupies a separate address space from Program Memory. In MUG51TBAE, there are 256 bytes of internal scratch-pad RAM. For many applications those need more internal RAM, the MUG51TBAE provides another on-chip 1 Kbytes of RAM, which is called XRAM, accessed by MOVX instruction.

The whole embedded Flash, functioning as Program Memory, is divided into four blocks: Application ROM (APROM) normally for User Code, Loader ROM (LDROM) normally for Boot Code and defined special address from APROM, LIB memroy 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.

### Program Memory

The Program Memory stores the program codes to execute as shown in Figure 6.1‑1 MUG51TBAE 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 MUG51TBAE 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 MUG51TBAE can be up to 16 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)”. 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** |
| **-** | **-** | **IODEFAULT** | **-** | **-** | **LDSIZE[2:0]** | | |
| - | - | R/W | - | - | R/W | | |

Factory default value: 1111 1111b

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [5] | **IODEFAULT** | **GPIO Pin Default Select:**  1 = GPIO pin default in INPUT mode.  0 = GPIO pin default in Quasi mode. |
| [2:0] | **LDSIZE[2:0]** | **LDROM Size Select**  Flash size is 16KB:  111 = No LDROM. APROM is 16 Kbytes.  110 = LDROM is 1 Kbytes. APROM is 15 Kbytes.  101 = LDROM is 2 Kbytes. APROM is 14 Kbytes.  100 = LDROM is 3 Kbytes. APROM is 13 Kbytes.  0xx = LDROM is 4 Kbytes. APROM is 12 Kbytes. |

|  |
| --- |
|  |

Figure 6.1‑1 MUG51TBAE Program Memory Map

### 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.

### 96-Bit Unique Code (UID)

Before shipping out, each MUG51TBAE chip was factory pre-programmed with a 96-bit width serial number, which is guaranteed to be unique for each piece of MUG51TBAE. The serial number is called Unique Code or UID. The user can read the Unique Code only by IAP command. More details please see Chapter 6.3.1 In-application-programming (IAP).

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

### Data Flash

MUG51TBAE 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.

### Data Memory

#### Internal Data Memory

|  |
| --- |
|  |

Figure 6.1‑3 Data Memory Map

Figure 6.1‑3 Data Memory Map shows the internal Data Memory spaces available on MUG51TBAE. Internal Data Memory occupies a separate address space from Program Memory. The internal Data Memory can be divided into three blocks. They are the lower 128 bytes of RAM, the upper 128 bytes of RAM, and the 128 bytes of SFR space. Internal Data Memory addresses are always 8-bit wide, which implies an address space of only 256 bytes. Direct addressing higher than 7FH will access the special function registers (SFR) space and indirect addressing higher than 7FH will access the upper 128 bytes of RAM. Although the SFR space and the upper 128 bytes of RAM share the same logic address, 80H through FFH, actually they are physically separate entities. Direct addressing to distinguish with the higher 128 bytes of RAM can only access these SFR. Sixteen addresses in SFR space are either byte-addressable or bit-addressable. The bit-addressable SFR are those whose addresses end in 0H or 8H.

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

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

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

|  |
| --- |
|  |

Figure 6.1‑4 Internal 256 Bytes RAM Addressing

#### On-Chip XRAM

The MUG51TBAE provides additional on-chip 1 Kbytes auxiliary RAM called XRAM to enlarge the RAM space. It occupies the address space from 00H through 3FFH. The 1 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

### Config Bytes

The MUG51TBAE 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** | **-** | **-** | **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. |
| [6] | **-** | Reserved. |
| [5] | **OCDPWM** | **PWM Output State Under OCD Halt**  This bit decides the output state of PWM when OCD halts CPU.  1 = Tri-state pins those are used as PWM outputs.  0 = PWM continues. |
| [4] | **OCDEN** | **OCD Enable**  1 = OCD Disabled.  0 = OCD Enabled.  **Note:** If MCU run in OCD debug mode and OCDEN = 0, hard fault reset will disable. Only HFIF flag be asserted. |
| [3:2] | **-** | Reserved |
| [1] | **LOCK** | **Chip Lock Enable**  1 = Chip is unlocked. Flash Memory is not locked. Their contents can be read out through a parallel Writer/ICP programmer.  0 = Chip is locked. Whole Flash Memory is locked. Their contents read through a parallel Writer or ICP programmer will be all blank (FFH). Programming to Flash Memory is invalid.  Note that CONFIG bytes are always unlocked and can be read. Hence, once the chip is locked, the CONFIG bytes cannot be erased or programmed individually. The only way to disable chip lock is execute “whole chip erase”. However, all data within the Flash Memory and CONFIG bits will be erased when this procedure is executed.  If the chip is locked, it does not alter the IAP function. |

|  |
| --- |
|  |

Figure 6.1‑5 CONFIG0 Any Reset Reloading

##### CONFIG1

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

Factory default value: 1111 1111b

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **-** | Reserved |
| [5] | **IODEFAULT** | **GPIO Pin Default Select:**  1 = GPIO pin default in INPUT mode.  0 = GPIO pin default in Quasi mode. |
| [4:3] | **-** | Reserved |
| [2:0] | **LDSIZE[2:0]** | **LDROM Size Select**  Flash size is 16KB:  111 = No LDROM. APROM is 16 Kbytes.  110 = LDROM is 1 Kbytes. APROM is 15 Kbytes.  101 = LDROM is 2 Kbytes. APROM is 14 Kbytes.  100 = LDROM is 3 Kbytes. APROM is 13 Kbytes.  0xx = LDROM is 4 Kbytes. APROM is 12 Kbytes. |

##### CONFIG2

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

Factory default value: 1111 1111b

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

|  |
| --- |
|  |

Figure 6.1‑6 CONFIG2 Power-On Reset Reloading

##### CONFIG3

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

Factory default value: 1111 1111b

| Bit | Name | Description |
| --- | --- | --- |
| [5:] | **-** | Reserved |
| 1:0 | **CLKDIV[1:0]** | **System Clock Divider Sectection**  This field configures the system clock divider after MCU execution.  11 = System clock divided by 1.  10 = System clock divided by 2.  01 = System clock divided by 4.  00 = System clock divided by 6. |

##### CONFIG4

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

Factory default value: 1111 1111b

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

### Special Function Register (SFR)

The MUG51TBAE 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 MUG51TBAE 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.

#### Special Function Register

| **Page** | **Addr** | **0/8** | **1/9** | **2/A** | **3/B** | **4/C** | **5/D** | **6/E** | **7/F** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **F8** | **S1CON** | - | - | - | - | DMA1BAH | EIP1 | EIPH1 |
| **1** | PWM0DTEN | PWM0DTCNT | PWM0MEN | PWM0MD | LVRFLTEN | - | LVRDIS |
| **2** | P0MF10 | P0MF32 | - | - | - | - | - |
| **3** | - | - | - | - | - | - | - |
| **0** | **F0** | **B** | DMA1TSR | MTM1DA | SPI0CR0 | SPI0SR | SPI0DR | DMA0BAH | EIPH0 |
| **1** | - | - | SPI0CR1 | - | - | - | LVRTRIM |
| **2** | P1MF76 | P2MF10 | P2MF32 | P2MF54 | - | P3MF10 | P3MF32 |
| **3** | - | - | - | - | - | - | - |
| **0** | **E8** | **I2C1CON** | DMA0TSR | MTM0DA | DMA1CR0 | DMA1MAL | DMA1CNT | DMA1CCNT | EIP0 |
| **1** | PICON | PINEN | PIPEN | - | C2L | C2H | LDOTRIM |
| **2** | - | - | P4MF10 | - | - | - | P5MF10 |
| **3** | - | - | - | - | - | - | - |
| **0** | **E0** | **ACC** | - | - | - | - | - | - | - |
| **1** | CAPCON0 | CAPCON1 | CAPCON2 | C0L | C0H | C1L | C1H |
| **2** | P5MF32 | P5MF54 | - | BRCTRIM | - | - | - |
| **3** | - | - | - | - | - | - | - |
| **0** | **D8** | **P4** | SC0DR | SC0EGT | SC0ETURD0 | SC0ETURD1 | SC0IE | SC0IS | SC0TSR |
| **1** | PWM0PL | PWM0C0L | PWM0C1L | PWM0C2L | PWM0C3L | - | PWM0CON1 |
| **2** | - | - | - | - | - | - | - |
| **3** | - | - | - | - | - | - | - |
| **0** | **D0** | **PSW** | PWM0CON0 | ACMPCR0 | ACMPCR1 | ACMPSR | ACMPVREF | SC0CR0 | SC0CR1 |
| **1** | PWM0PH | PWM0C0H | PWM0C1H | PWM0C2H | PWM0C3H | PWM0NP | PWM0FBD |
| **2** | - | - | - | - | - | - | I2C1ADDRM |
| **3** | - | - | - | - | - | - | - |
| **0** | **C8** | **T2CON** | T2MOD | PIF | - | TL2 | TH2 | - | - |
| **1** | AUXR1 | RCMP2L | RCMP2H | PWM0C4L | PWM0C5L | - | - |
| **2** | - | - | - | - | - | - | I2C0ADDRM |
| **3** | PRTHCON0 | - | - | - | - | PWM0FBS | AUXR3 |
| **0** | **C0** | **I2C0CON** | I2C0ADDR0 | - | - | T3CON | RL3 | RH3 | TA |
| **1** | CKDIV | P3M1 | P3M2 | PWM0C4H | PWM0C5H | PORDIS |
| **2** | - | - | - | - | - | - |
| **3** | - | - | - | - | - | - |
| **0** | **B8** | **IP** | SADEN0 | SADEN1 | SADDR1 | I2C0DAT | I2C0STAT | I2C0CLK | I2C0TOC |
| **1** | P4M1 | P4M2 | P4S | P4SR | P5M1 | P5M2 | P5S |
| **2** | - | - | - | - | - | CWKH | RWKH |
| **3** | - | - | - | - | - | - | - |
| **0** | **B0** | **P3** | P5 | I2C1ADDR0 | I2C1DAT | I2C1STAT | I2C1CLK | I2C1TOC | IPH |
| **1** | P0M1 | P0M2 | P1M1 | P1M2 | P2M1 | P2M2 | PWM0INTC |
| **2** | - | - | - | - | - | - | - |
| **3** | - | - | - | - | - | - | - |
| **0** | **A8** | **IE** | SADDR0 | WDCON | BODCON1 | EIP2 | EIPH2 | IAPFD | IAPCN |
| **1** | - | - | ACMPCR2 | P3S | P3SR | P5SR | PIPS7 |
| **2** | - | - | - | - | - | - | - |
| **3** | - | - | - | - | - | - | - |
| **0** | **A0** | **P2** | ADCCON0 | AUXR0 | BODCON0 | IAPTRG | IAPUEN | IAPAL | IAPAH |
| **1** | PIPS0 | PIPS1 | PIPS2 | PIPS3 | PIPS4 | PIPS5 | PIPS6 |
| **2** | I2C0ADDR1 | I2C0ADDR2 | I2C0ADDR3 | I2C1ADDR1 | I2C1ADDR2 | I2C1ADDR3 | - |
| **3** | - | - | - | - | - | - | - |
| **0** | **98** | **SCON** | SBUF | SBUF1 | EIE0 | EIE1 | RSR | - | CHPCON |
| **1** | P0S | P0SR | P1S | P1SR | P2S | P2SR | - |
| **2** | - | - | - | - | - | - | - |
| **3** | WDCON1 | DMA0SEED | DMA1SEED | - | - | - | - |
| **0** | **90** | **P1** | SFRS | DMA0CR0 | DMA0MAL | DMA0CNT | DMA0CCNT | CKSWT | CKEN |
| **1** | P0UP | P1UP | P2UP | P3UP | P4UP | P5UP |
| **2** | - | - | - | - | - | - |
| **3** | DMA0CRC | DMA1CRC | - | - | - | - |
| **0** | **88** | **TCON** | TMOD | TL0 | TL1 | TH0 | TH1 | CKCON | WKCON |
| **1** | P0DW | P1DW | P2DW | P3DW | P4DW | P5DW |
| **2** | - | - | - | - |  |  |
| **3** | DMA0CR1 | DMA1CR1 |  |  | T2ACMP | PDL |
| **0** | **80** | **P0** | SP | DPL | DPH | - | - | RWKL | PCON |
| **1** | LIRCTRIM | - | CWKL |
| **2** | - | - | - |
| **3** | MRCTRIM0 | MRCTRIM1 | - |
| **Note:**  If(TA = 0xAA,TA = 0x55) 🡪 {rctrim0, rctrim1, ckswt, cken, iaptc, chpcon, bodcon0, iaptrg, iapuen, wdcon, bodcon1, vrftrim, ldotrim, lvrflten, rcnt, lvrdis, pordis, brctrim, mrctrim0, mrctrim1} Read/Write Enable.  Unoccupied addresses in the SFR space marked in “-“ are reserved for future use. Accessing these areas will have an in determinate effect and should be avoided. | | | | | | | | | |

Table 6.1‑1 Special Function Register Map Table

#### SFR Definitions And Reset Values

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

| **Register** | **Definition** | **Addr.** | **Page** | **MSB** |  |  |  |  |  |  | **LSB** | **Reset**  **Value** | **TA** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |  |  |
| **EIPH1** | Extensive Interrupt Priority High 1 | FFH | 0 | PSPI1H | PDMA1H | PDMA0H | PSMCH | PHFH | PWKTH | PT3H | PSH\_1 | 00000000b |  |
| **LVRDIS** | Lvr Disable | FFH | 1 | LVRDIS[7:0] | | | | | | | | 00000000b | **Y** |
| **-** | - | FFH | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | FFH | 3 | - | - | - | - | - | - | - | - | - |  |
| **EIP1** | Extensive Interrupt Priority 1 | FEH | 0 | PSPI1 | PDMA1 | PDMA0 | PSMC | PHF | PWKT | PT3 | PS\_1 | 00000000b |  |
| **-** | - | FEH | 1 | - | - | - | - | - | - | - | - |  |  |
| **-** | - | FEH | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | FEH | 3 | - | - | - | - | - | - | - | - | - |  |
| **DMA1BAH** | PDMA1 Base Address High Byte | FDH | 0 | MDAH[3:0] | | | | MAH[3:0] | | | | 00000000b |  |
| **LVRFLTEN** | LVR Filter Enable | FDH | 1 | LVRFLTEN[7:0] | | | | | | | | 00000000b | **Y** |
| **-** | - | FDH | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | FDH | 3 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | FCH | 0 | - | | | | | | | | - |  |
| **PWM0MD** | PWM0 Mask Data | FCH | 1 | - | - | PMD5 | PMD4 | PMD3 | PMD2 | PMD1 | PMD0 | 00000000b |  |
| **-** | - | FCH | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | FCH | 3 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | FBH | 0 | - | - | - | - | - | - | - | - | - |  |
| **PWM0MEN** | PWM0 Mask Enable | FBH | 1 | - | - | PMEN5 | PMEN4 | PMEN3 | PMEN2 | PMEN1 | PMEN0 | 00000000b |  |
| **-** | - | FBH | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | FBH | 3 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | FAH | 0 | - | - | - | - | - | - | - | - | - |  |
| **PWM0DTCNT** | PWM0 Deadtime Counter | FAH | 1 | PWM0DTCNT[7:0] | | | | | | | | 00000000b | **Y** |
| **P0MF32** | P0.3 And P0.2 Multi Function Select | FAH | 2 | P0MF3[3:0] | | | | P0MF2[3:0] | | | | 00000000b |  |
| **-** | - | FAH | 3 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | F9H | 0 | - | - | - | - | - | - | - | - | - |  |
| **PWM0DTEN** | PWM0 Deadtime Enable | F9H | 1 | - | - | - | PWMnDTCNT.8 | - | PDT45EN | PDT23EN | PDT01EN | 00000000b | **Y** |
| **P0MF10** | P0.1 And P0.0 Multi Function Select | F9H | 2 | P0MF1[3:0] | | | | P0MF0[3:0] | | | | 00000000b |  |
| **-** | - | F9H | 3 | - | - | - | - | - | - | - | - | - |  |
| **S1CON** | Serial Port1 Control | F8H | A | SM0\_1/FE\_1 | SM1\_1 | SM2\_1 | REN\_1 | TB8\_1 | RB8\_1 | TI\_1 | RI\_1 | 00000000b |  |
| **EIPH0** | Extensive Interrupt Priority High | F7H | 0 | PT2H | PSPIH | PFBH | PWDTH | PPWMH | PCAPH | PPIH | PI2CH | 00000000b |  |
| **-** | - | F7H | 1 | - | - | - | - | - | - | - | - |  |  |
| **P3MF32** | P3.3 And P3.2 Multi Function Select | F7H | 2 | P3MF3[3:0] | | | | P3MF2[3:0] | | | | 00000000b |  |
| **-** | - | F7H | 3 | - | - | - | - | - | - | - | - | - |  |
| **DMA0BAH** | PDMA0 Base Address High Byte | F6H | 0 | MDAH[3:0] | | | | MAH[3:0] | | | | 00000000b |  |
| **-** | - | F6H | 1 | - | - | - | - | - | - | - | - | - | **-** |
| **P3MF10** | P3.1 And PRC3.0 Multi Function Select | F6H | 2 | P3MF1[3:0] | | | | P3MF0[3:0] | | | | 00000000b |  |
| **-** | - | F6H | 3 | - | - | - | - | - | - | - | - | - |  |
| **SPI0DR** | SPI0 Data | F5H | 0 | SPDR[7:0] | | | | | | | | 00000000b |  |
| **-** | - | F5H | 1 | - | | | | | | | | - | **-** |
| **-** | - | F5H | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | F5H | 3 | - | - | - | - | - | - | - | - | - | **-** |
| **SPI0SR** | SPI0 Status | F4H | 0 | SPIF | WCOL | SPIOVF | MODF | DISMODF | DISSPIF | TXBFF | - | 00000000b |  |
| **-** | - | F4H | 1 | - | - | - | - | - | - | - | - | - | **-** |
| **P2MF54** | P2.5 And P2.4 Multi Function Select | F4H | 2 | P2MF5[3:0] | | | | P2MF4[3:0] | | | | 00000000b |  |
| **-** | **-** | F4H | 3 | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** | **-** |
| **SPI0CR0** | SPI0 Control 0 | F3H | 0 | SSOE | SPIEN | LSBFE | MSTR | CPOL | CPHA | SPR1 | SPR0 | 00000000b |  |
| **SPI0CR1** | SPI0 Control 1 | F3H | 1 | - | - | SPR3 | SPR2 | TXDMAEN | RXDMAEN | SPIS1 | SPIS0 | 00000000b |  |
| **P2MF32** | P2.3 And P2.2 Multifunction Select | F3H | 2 | P2MF3[3:0] | | | | P2MF2[3:0] | | | | 00000000b |  |
| **-** | - | F3H | 3 | - | - | - | - | - | - | - | - | - | **-** |
| **MTM1DA** | Memory To Memory Destination Address Low Byte | F2H | 0 | MDAL[7:0] | | | | | | | | 00000000b |  |
| **-** |  | F2H | 1 | - | - | - | - | - | - | - | - |  |  |
| **P2MF10** | P2.1 And P2.0 Multi Function Select | F2H | 2 | P2MF1[3:0] | | | | P2MF0[3:0] | | | | 00000000b |  |
| **-** | - | F2H | 3 | - | - | - | - | - | - | - | - | - | **-** |
| **DMA1TSR** | PDMA1 Transfer Status Register | F1H | 0 | - | - | - | - | - | ACT | HDONE | FDONE | 00000000b |  |
| **-** |  | F1H | 1 | - | - | - | - | - | - | - | - |  |  |
| **P1MF76** | P1.7 And P1\_6 Multifunction Select | F1H | 2 | P1MF7[3:0] | | | | P1MF6[3:0] | | | | 00000000b |  |
| **-** | - | F1H | 3 | - | - | - | - | - | - | - | - | - |  |
| **B** | B Register | F0H |  | B.7 | B.6 | B.5 | B.4 | B.3 | B.2 | B.1 | B.0 | 00000000b |  |
| **EIP0** | Extensive Interrupt Priority | EFH | 0 | PT2 | PSPI | PFB | PWDT | PPWM | PCAP | PPI | PI2C | 00000000b |  |
| **-** |  | - | 1 | - | - | - | - | - | - | - | - |  |  |
| **P5MF10** | P5.0 And P5.0 Multifunction Select | EFH | 2 | P5MF1[3:0] | | | | P5MF0[3:0] | | | | 00000000b |  |
| **-** |  | EFH | 3 |  |  |  | - | - | - | - | - | xxxxxxxxb |  |
| **DMA1CCNT** | PDMA1 Current Transfer Count | EEH | 0 | CCNT[7:0] | | | | | | | | 00000000b |  |
| **C2H** | Input Capture2 High Byte | EEH | 1 | C2H[7:0] | | | | | | | | 00000000b |  |
| **-** | **-** | EEH | 2 | **-** |  |  |  | **-** |  |  |  | **-** |  |
| **-** |  | EEH | 3 |  |  |  | - |  | - | - | - |  |  |
| **DMA1CNT** | PDMA1 Transfer Count | EDH | 0 | CNT[7:0] | | | | | | | | 00000000b |  |
| **C2L** | Input Capture 2 Low Byte | EDH | 1 | C2L[7:0] | | | | | | | | 00000000b |  |
| **-** | - | EDH | 2 | - | - | - | - | - | - | - | - | - |  |
|  |  | EDH | 3 | - | - | - | - | - | - | - | - | - |  |
| **DMA1MAL** | PDMA1 XRAM Base Address Low Byte | ECH | 0 | MAL[7:0] | | | | | | | | 00000000b |  |
| **-** | - | ECH | 1 | - | - | - |  | - |  |  |  | - | **-** |
| **-** | - | ECH | 2 | - | - | - | - | - | - | - | - | - |  |
|  |  | ECH | 3 | - | - | - | - | - | - | - | - | - |  |
| **DMA1CR0** | PDMA1 Control Register | EBH | 0 | PSSEL\_3 | PSSEL\_2 | PSSEL\_1 | PSSEL\_0 | HIE | FIE | RUN | EN | 00000000b |  |
| **PIPEN** | Pin Interrupt High Level/Rising Edge Enable | EBH | 1 | PIPEN7 | PIPEN6 | PIPEN5 | PIPEN4 | PIPEN3 | PIPEN2 | PIPEN1 | PIPEN0 | 00000000b |  |
| **P4MF10** | P4.1 And P4.0 Multi Function Select | EBH | 2 | P4MF1[3:0] | | | | P4MF0[3:0] | | | | 00000000b |  |
|  |  | EBH | 3 | - | - | - | - | - | - | - | - |  |  |
| **MTM0DA** | Memory To Memory Destination Address Low Byte | EAH | 0 | MDAL[7:0] | | | | | | | | 00000000b |  |
| **PINEN** | Pin Interrupt Low Level/Falling Edge Enable | EAH | 1 | PINEN7 | PINEN6 | PINEN5 | PINEN4 | PINEN3 | PINEN2 | PINEN1 | PINEN0 | 00000000b |  |
| **-** | - | EAH | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | EAH | 3 | - | - | - | - | - | - | - | - | - |  |
| **DMA0TSR** | PDMA0 Transfer Status Register | E9H | 0 | - | - | - | - | - | ACT | HDONE | FDONE | 00000000b |  |
| **PICON** | Pin Interrupt Control | E9H | 1 | PIT7 | PIT6 | PIT5 | PIT4 | PIT3 | PIT2 | PIT1 | PIT0 | 00000000b |  |
| **-** | - | E9H | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | E9H | 3 | - | - | - | - | - | - | - | - | - | **-** |
| **I2C1CON** | I2C1 Control Register | E8H |  | I | I2CEN | STA | STO | SI | AA | - | - | 00000000b |  |
| **-** | - | E7H | 0 | - | - | - | - | - | - | - | - | - |  |
| **C1H** | Input Capture Channel 1 High Byte | E7H | 1 | C1H[7:0] | | | | | | | | 00000000b |  |
| **-** | - | E7H | 2 | - | - | - | - | - | - | - | - | - |  |
|  |  | E7H | 3 | - | - | - | - | - | - | - | - |  |  |
| **-** | - | E6H | 0 | - | - | - | - | - | - | - | - | - |  |
| **C1L** | Input Capture1 Low Byte | E6H | 1 | C1L[7:0] | | | | | | | | 00000000b |  |
| **-** | - | E6H | 2 | - | - | - | - | - | - | - | - | - |  |
|  |  | E6H | 3 | - | - | - | - | - | - | - | - |  |  |
| **-** | - | E5H | 0 | - | - | - | - | - | - | - | - | - |  |
| **C0H** | Input Capture 0 High Byte | E5H | 1 | C0H[7:0] | | | | | | | | 00000000b |  |
| **-** | - | E5H | 2 | - | - | - | - | - | - | - | - | - |  |
|  |  | E5H | 3 | - | - | - | - | - | - | - | - |  |  |
| **-** | - | E4H | 0 | - | - | - | - | - | - | - | - | - |  |
| **C0L** | Input Capture 0 Low Byte | E4H | 1 | C0L[7:0] | | | | | | | | 00000000b |  |
| **-** | - | E4H | 2 | - | - | - | - | - | - | - | - | - |  |
|  |  | E4H | 3 | - | - | - | - | - | - | - | - |  |  |
| **-** | - | E3H | 0 | - | - | - | - | - | - | - | - | - |  |
| **CAPCON2** | Input Capture Control2 | E3H | 1 | - | ENF2 | ENF1 | ENF0 | - | - | - | CMOD | 00000000b |  |
| **-** | - | E3H | 2 | - | - | - | - | - | - | - | - | - |  |
|  |  | E3H | 3 | - | - | - | - | - | - | - | - |  |  |
| **-** | - | E2H | 0 | - | - | - | - | - | - | - | - | - |  |
| **CAPCON1** | Input Capture Control 1 | E2H | 1 | - | - | CAP2LS\_1 | CAP2LS\_0 | CAP1LS\_1 | CAP1LS\_0 | CAP0LS\_1 | CAP0LS\_0 | 00000000b |  |
| **P5MF54** | P5.5 And P5.4 Multi Function Select | E2H | 2 | P5MF5[3:0] | | | | P5MF4[3:0] | | | | 00000000b |  |
|  |  | E2H | 3 | - | - | - | - | - | - | - | - |  |  |
| **-** | - | E1H | 0 | - | - | - | - | - | - | - | - | - |  |
| **CAPCON0** | Input Capture Control 0 | E1H | 1 | - | CAPEN2 | CAPEN1 | CAPEN0 | - | CAPF2 | CAPF1 | CAPF0 | 00000000b |  |
| **P5MF32** | P5.3 And P5.2 Multi Function Select | E1H | 2 | P5MF3[3:0] | | | | P5MF2[3:0] | | | | 00000000b |  |
|  |  | E1H | 3 | - | - | - | - | - | - | - | - |  |  |
| **ACC** | Accumulator | E0H | A | ACC.7 | ACC.6 | ACC.5 | ACC.4 | ACC.3 | ACC.2 | ACC.1 | ACC.0 | 00000000b |  |
| **SC0TSR** | Sc0 Transfer Status Register | DFH | 0 | ACT | BEF | FEF | PEF | TXEMPTY | TXOV | RXEMPTY | RXOV | 00001010b |  |
| **PWM0CON1** | PWM Control 1 | DFH | 1 | PWMMOD\_1 | PWMMOD\_0 | GP | PWMTYP | FBINEN | PWMDIV\_2 | PWMDIV\_1 | PWMDIV\_0 | 00000000b |  |
| **-** | - | DFH | 2 | - | - | - | - | - | - | - | - | - |  |
|  |  | DFH | 3 | - | - | - | - | - | - | - | - |  |  |
| **SC0IS** | Sc0 Interrupt Status Register | DEH | 0 | - | - | Tx\_Er | ACERRIF | BGTIF | TERRIF | TBEIF | RDAIF | 00000010b |  |
| **-** | - | DEH | 1 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | DEH | 2 | - | - | - | - | - | - | - | - | - |  |
|  |  | DEH | 3 | - | - | - | - | - | - | - | - |  |  |
| **SC0IE** | SC0 Interrupt Enable Control Register | DDH | 0 | - | - | - | ACERRIEN | BGTIEN | TERRIEN | TBEIEN | RDAIEN | 00000000b |  |
| **PWM0C3L** | Pwm0 Channel 3 Duty Low Byte | DDH | 1 | PWM0C3[7:0] | | | | | | | | 00000000b |  |
| **-** | - | DDH | 2 | - | - | - | - | - | - | - | - | - |  |
|  |  | DDH | 3 | - | - | - | - | - | - | - | - |  |  |
| **SC0ETURD1** | SC0 etu Rate Divider Register1 | DCH | 0 | - | SCDIV\_2 | SCDIV\_1 | SCDIV\_0 | ETURDIV\_11 | ETURDIV\_10 | ETURDIV\_9 | ETURDIV\_8 | 00110001b |  |
| **PWM0C2L** | Pwm0 Channel 2 Duty Low Byte | DCH | 1 | PWM0C2[7:0] | | | | | | | | 00000000b |  |
| **-** | - | DCH | 2 | - | - | - | - | - | - | - | - | - |  |
|  |  | DDH | 3 | - | - | - | - | - | - | - | - | - |  |
| **SC0ETURD0** | SC0etu Rate Divider Register 0 | DBH | 0 | ETURDIV[7:0] | | | | | | | | 01110001b |  |
| **PWM0C1L** | Pwm0 Channe L1 Duty Low Byte | DBH | 1 | PWM0C1[7:0] | | | | | | | | 00000000b |  |
| **-** | - | DBH | 2 | - | - | - | - | - | - | - | - | - |  |
|  |  | DBH | 3 | - | - | - | - | - | - | - | - | - |  |
| **SC0EGT** | SC0 Extra Guard Time Register | DAH | 0 | SCnEGT[7:0] | | | | | | | | 00000000b |  |
| **PWM0C0L** | PWM0 Channel 0 Duty Low Byte | DAH | 1 | PWM0C0[7:0] | | | | | | | | 00000000b |  |
| **-** | - | DAH | 2 | - | - | - | - | - | - | - | - | - |  |
|  |  | DAH | 3 | - | - | - | - | - | - | - | - |  |  |
| **SC0DR** | SC0 Data Register | D9H | 0 | SC0DR[7:0] | | | | | | | | 00000000b |  |
| **PWM0PL** | PWM Period Low Byte | D9H | 1 | PWM0P[7:0] | | | | | | | | 00000000b |  |
| **-** | - | D9H | 2 | - | - | - | - | - | - | - | - | - |  |
| - | - | D9H | 3 | - | - | - | - | - | - | - | - |  |  |
| **P4** | Port4 | D8H | A | P4.7 | P4.6 | P4.5 | P4.4 | P4.3 | P4.2 | P4.1 | P4.0 | Outputlatch, 11111111b Input, XXXXXXXXb |  |
| **SC0CR1** | SC0 Control Register 1 | D7H | 0 | OPE | PBOFF | WLS1 | WLS0 | TXDMAEN | RXDMAEN | CLKKEEP | UARTEN | 00000000b |  |
| **PWM0FBD** | Brake Data | D7H | 1 | FBF | FBINLS | FBD5 | FBD4 | FBD3 | FBD2 | FBD1 | FBD0 | 00000000b |  |
| **I2C1ADDRM** | I2c1 Address Mask | D7H | 2 | I2C1ADDRM | | | | | | | | 00000000b |  |
| - | - | D7H | 3 | - | - | - | - | - | - | - | - |  |  |
| **SC0CR0** | SC0 Control Register 0 | D6H | 0 | NSB | T | RXBGTEN | CONSEL | AUTOCEN | TXOFF | RXOFF | SCEN | 00000000b |  |
| **PWM0NP** | PWM Negative Polarity | D6H | 1 | - | - | PNP5 | PNP4 | PNP3 | PNP2 | PNP1 | PNP0 | 00000000b |  |
| - | - | D6H | 2 | - | - | - | - | - | - | - | - | - |  |
| - | - | D6H | 3 | - | - | - | - | - | - | - | - |  |  |
| **ACMPVREF** | ACMP Reference Voltage Control | D5H | 0 | - | CRV1CTL\_2 | CRV1CTL\_\_1 | CRV1CTL\_0 | - | CRV0CTL\_2 | CRV0CTL\_1 | CRV0CTL\_0 | 00000000b |  |
| **PWM0C3H** | PWM0 Channel 3 Duty High Byte | D5H | 1 | PWM0C3[15:8] | | | | | | | | 00000000b |  |
| **PWM3CON1** | Pwm3 Control 1 | D5H | 2 | PWMMOD\_1 | PWMMOD\_0 | GP | PWMTYP | FBINEN | PWMDIV\_2 | PWMDIV\_1 | PWMDIV\_0 | 00000000b |  |
| - | - | D5H | 3 | - | - | - | - | - | - | - | - |  |  |
| **ACMPSR** | Analog Comparator Status Register | D4H | 0 | - | - | - | - | ACMP1O | ACMP1IF | ACMP0O | ACMP0IF | 00000000b |  |
| **PWM0C2H** | PWM0 Channel 2 Duty High Byte | D4H | 1 | PWM0C3[15:8] | | | | | | | | 00000000b |  |
| - | - | D4H | 2 | - | - | - | - | - | - | - | - | - |  |
| - | - | D4H | 3 | - | - | - | - | - | - | - | - |  |  |
| **ACMPCR1** | Analog Comparator Control Register1 | D3H | 0 | POSSEL\_1 | POSSEL\_0 | NEGSEL\_1 | NEGSEL\_0 | WKEN | HYSEN | ACMPIE | ACMPEN | 00000000b |  |
| **PWM0C1H** | PWM0 Channel 1 Duty High Byte | D3H | 1 | PWM0C1[15:8] | | | | | | | | 00000000b |  |
| - | - | D3H | 2 | - | - | - | - | - | - | - | - | - |  |
| - | - | D3H | 3 | - | - | - | - | - | - | - | - |  |  |
| **ACMPCR0** | Analog Comparator Control Register 0 | D2H | 0 | POSSEL\_1 | POSSEL\_0 | NEGSEL\_1 | NEGSEL\_0 | WKEN | HYSEN | ACMPIE | ACMPEN | 00000000b |  |
| **PWM0C0H** | PWM0 Channel 0 Duty High Byte | D2H | 1 | PWM0C1[15:8] | | | | | | | | 00000000b |  |
| - | - | D2H | 2 | - | - | - | - | - | - | - | - | - |  |
| - | - | D2H | 3 | - | - | - | - | - | - | - | - |  |  |
| **PWM0CON0** | PWM0 Control Register 0 | D1H | 0 | PWMRUN | LOAD | PWMF | CLRPWM | - | - | - | - | 00000000b |  |
| **PWM0PH** | PWM0 Period High Byte | D1H | 1 | PWM0P[15:8] | | | | | | | | 00000000b |  |
| - | - | D1H | 2 | - | - | - | - | - | - | - | - | - |  |
| - | - | D1H | 3 | - | - | - | - | - | - | - | - |  |  |
| **PSW** | Program Status Word | D0H | A | CY | AC | F0 | RS1 | RS0 | OV | - | P | 00000000b |  |
| - | - | CFH | 0 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | CFH | 1 |  | - | - | - | - | - | - |  | - | - |
| **I2C0ADDRM** | I2C0 Address Mask | CFH | 2 | I2C0ADDRM | | | | | | | | 00000000b |  |
| **AUXR3** | Auxiliary Register 3 | CFH | 3 | - | - | - | - | - | UART2DG | UART1DG | UART0DG | 00000000b |  |
| **-** | - | CEH | 0 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | CEH | 1 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | CEH | 2 | - | - | - | - | - | - |  |  | - |  |
| **PWM0FBS** | PWM0 Brake Source Select | CEH | 3 | - | - | - | - | - | - | PWM0FBS\_1 | PWM0FBS\_0 | 00000000b |  |
| **TH2** | Timer 2 High Byte | CDH | 0 | T2[15:8] | | | | | | | | 00000000b |  |
| **PWM0C5L** | PWM0 Channel5 Duty Low Byte | CDH | 1 | PWM0C5[7:0] | | | | | | | | 00000000b |  |
| **-** | - | - | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | **-** | CDH | 3 |  |  |  |  | - |  | - |  | - | - |
| **TL2** | Timer 2 Low Byte | CCH | 0 | T2[7:0] | | | | | | | | 00000000b |  |
| **PWM0C4L** | PWM0 Channel 4 Duty Low Byte | CCH | 1 | PWM0C4[7:0] | | | | | | | | 00000000b |  |
| - | - | CCH | 2 | - | - | - | - | - | - | - | - | - |  |
| - | - | CCH | 3 | - | - | - | - | - | - | - | - |  |  |
| - | - | CBH | 0 | - | - | - | - | - | - | - | - | - |  |
| **RCMP2H** | Timer2 Reload / Compare High Byte | CBH | 1 | RCMP2[15:8] | | | | | | | | 00000000b |  |
| - | - | CBH | 2 | - | - | - | - | - | - | - | - | - |  |
| - | - | CBH | 3 | - | - | - | - | - | - | - | - |  |  |
| **PIF** | Pin Interrupt Flags | CAH | 0 | PIF7 | PIF6 | PIF5 | PIF4 | PIF3 | PIF2 | PIF1 | PIF0 | 00000000b |  |
| **RCMP2L** | Timer2 Compare Low Byte | CAH | 1 | RCMP2[7:0] | | | | | | | | 00000000b |  |
| - | - | CAH | 2 | - | - | - | - | - | - | - | - | - |  |
| - | - | CAH | 3 | - | - | - | - | - | - | - | - |  |  |
| **T2MOD** | Timer2 Mode | C9H | 0 | LDEN | T2DIV\_2 | T2DIV\_1 | T2DIV\_0 | CAPCR | CMPCR | LDTS\_1 | LDTS\_0 | 00000000b |  |
| **AUXR1** | Auxiliary Register 1 | C9H | 1 |  |  |  |  | - | UART2PX | UART1PX | UART0PX | 00000000b |  |
| **-** | - | C9H | 2 | - | - | - | - | - | - | - | - | - |  |
| **PRTHCON0** | Port Low Return High Control 0 | C9H | 3 | P40RTH1 | P40RTH0 | P30RTH1 | P30RTH0 | P21RTH1 | P21RTH0 | P20RTH1 | P20RTH0 | 00000000b |  |
| **T2CON** | Timer2 Control | C8H | A | TF2 | - | - | T2HRF | T2LRF | TR2 | - | CM\_RL2 | 00000000b |  |
| **TA** | Time Access Protection | C7H | A | TA[7:0] | | | | | | | | 00000000b |  |
| **RH3** | Timer3 Reload Highbyte | C6H | 0 | RH3[15:8] | | | | | | | | 00000000b |  |
| **PORDIS** | Por Disable | C6H | 1 | PORDIS[7:0] | | | | | | | | 00000000b | **Y** |
| **-** | - | C6H | 2 | - | - | - | - | - | - | - | - | - |  |
|  |  | C6H | 3 | - | - | - | - | - | - | - | - |  |  |
| **RL3** | Timer3 Reload Low Byte | C5H | 0 | RL3[7:0] | | | | | | | | 00000000b |  |
| **PWM0C5H** | PWM0 Channel 5 Duty High Byte | C5H | 1 | PWM0C5[15:8] | | | | | | | | 00000000b |  |
| - | - | C5H | 2 | - | - | - | - | - | - | - | - | - |  |
| - | - | C5H | 3 | - | - | - | - | - | - | - | - |  |  |
| **T3CON** | Timer3 Control | C4H | 0 | SMOD\_1 | SMOD0\_1 | BRCK | TF3 | TR3 | T3PS\_2 | T3PS\_1 | T3PS\_0 | 00000000b |  |
| **PWM0C4H** | PWM0 Channel 4 Duty High Byte | C4H | 1 | PWM0C4[15:8] | | | | | | | | 00000000b |  |
| - | - | C4H | 2 | - | - | - | - | - | - | - | - | - |  |
| - | - | C4H | 3 | - | - | - | - | - | - | - | - |  |  |
| - | - | C3H | 0 | - |  |  |  |  |  |  |  | - |  |
| **P3M2** | Port3 Mode Select 2 | C3H | 1 | P3M2.7 | P3M2.6 | P3M2.5 | P3M2.4 | P3M2.3 | P3M2.2 | P3M2.1 | P3M2.0 | 00000000b |  |
| - | - | C3H | 2 | - | - | - | - | - | - | - | - | - |  |
| - | - | C3H | 3 | - | - | - | - | - | - | - | - |  |  |
| - | - | C2H | 0 | - | - | - | - | - | - | - | - | - |  |
| **P3M1** | Port3 Mode Select 1 | C2H | 1 | P3M1.7 | P3M1\_6 | P3M1\_5 | P3M1\_4 | P3M1\_3 | P3M1\_2 | P3M1\_1 | P3M1\_0 | 11111111B |  |
| - | - | C2H | 2 | - | - | - | - | - | - | - | - | - |  |
| - | - | C2H | 3 | - | - | - | - | - | - | - | - |  |  |
| **I2C0ADDR0** | I2C Own Slave Address | C1H | 0 | I2C0ADDR0\_7 | I2C0ADDR0\_6 | I2C0ADDR0\_5 | I2C0ADDR0\_4 | I2C0ADDR0\_3 | I2C0ADDR0\_2 | I2C0ADDR0\_1 | GC | 00000000b |  |
| **CKDIV** | Clock Divider | C1H | 1 | CKDIV[7:0] | | | | | | | | 00000000b |  |
| - | - | C1H | 2 | - | - | - | - | - | - | - | - | - |  |
| - | - | C1H | 3 | - | - | - | - | - | - | - | - |  |  |
| **I2C0CON** | I2C0 Control | C0H | A | I | I2CEN | STA | STO | SI | AA | - | - | 00000000b |  |
| **I2C0TOC** | I2C0 Time Out Counter | BFH | 0 | - | - | - | - | - | I2TOCEN | DIV | I2TOF | 00000000b |  |
| **P5S** | Port5 Schmitt Triggered Input | BFH | 1 | P5S.7 | P5S.6 | P5S.5 | P5S.4 | P5S.3 | P5S.2 | P5S.1 | P5S.0 | 00000000b |  |
| **RWKH** | Self Wakeup Timerre Load High Byte | BFH | 2 | RWK[15:8] | | | | | | | | 11111111b |  |
| **-** | - | BFH | 3 | - | - | - | - | - | - | - | - | - |  |
| **I2C0CLK** | I2C0 Clock | BEH | 0 | I2C0CLK[7:0] | | | | | | | | 00000000b |  |
| **P5M2** | Port5 Mode Select 2 | BEH | 1 | P5M2.7 | P5M2.6 | P5M2.5 | P5M2.4 | P5M2.3 | P5M2.2 | P5M2.1 | P5M2.0 | 00000000b |  |
| **CWKH** | Self Wakeup Timer Current Count High Byte | BEH | 2 | CWK[15:8] | | | | | | | | 00000000b |  |
| **-** | - | BEH | 3 | - | - | - | - | - | - | - | - | - |  |
| **I2C0STAT** | I2C0 Status | BDH | 0 | I2C0STAT\_7 | I2C0STAT\_6 | I2C0STAT\_5 | I2C0STAT\_4 | I2C0STAT\_3 | - | - | - | 11111000b |  |
| **P5M1** | Port5 Mode Select 1 | BDH | 1 | P5M1.7 | P5M1\_6 | P5M1\_5 | P5M1\_4 | P5M1\_3 | P5M1\_2 | P5M1\_1 | P5M1\_0 | 11111111b |  |
| **-** | - | BDH | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | BDH | 3 | - | - | - | - | - | - | - | - | - |  |
| **I2C0DAT** | I2C0 Data | BCH | 0 | I2C0DAT[7:0] | | | | | | | | 00000000b |  |
| **P4SR** | Port4 Slew Rate Control | BCH | 1 | P4SR.7 | P4SR.6 | P4SR.5 | P4SR.4 | P4SR.3 | P4SR.2 | P4SR.1 | P4SR.0 | 00000000b |  |
| **-** | - | BCH | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | BCH | 3 | - | - | - | - | - | - | - | - | - |  |
| **SADDR1** | Slave1 Address | BBH | 0 | SADDR1[7:0] | | | | | | | | 00000000b |  |
| **P4S** | Port4 Schmitt Triggered Input | BBH | 1 | P4S.7 | P4S.6 | P4S.5 | P4S.4 | P4S.3 | P4S.2 | P4S.1 | P4S.0 | 00000000b |  |
| **-** | - | BBH | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | BBH | 3 | - | - | - | - | - | - | - | - | - |  |
| **SADEN1** | UART0 Slave1 Address Mask | BAH | 0 | SADEN1[7:0] | | | | | | | | 00000000b |  |
| **P4M2** | Port4 Mode Select 2 | BAH | 1 | P4M2.7 | P4M2.6 | P4M2.5 | P4M2.4 | P4M2.3 | P4M2.2 | P4M2.1 | P4M2.0 | 00000000b |  |
| **-** | - | BAH | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | BAH | 3 | - | - | - | - | - | - | - | - | - | **-** |
| **SADEN0** | UART0 Slave 0 Address Mask | B9H | 0 | SADEN0[7:0] | | | | | | | | 00000000b |  |
| **P4M1** | Port 4 Mode Select 1 | B9H | 1 | P4M1.7 | P4M1\_6 | P4M1\_5 | P4M1\_4 | P4M1\_3 | P4M1\_2 | P4M1\_1 | P4M1\_0 | 11111111b |  |
| **-** | - | B9H | 2 | - | | | | | | | | - |  |
| **-** | - | B9H | 3 | - | - | - | - | - | - | - | - | - |  |
| **IP** | Interrupt Priority | B8H | A | - | PADC | PBOD | PS | PT1 | PX1 | PT0 | PX0 | 00000000b |  |
| **IPH** | Interrupt Priority High | B7H | 0 | - | PADCH | PBODH | PSH | PT1H | PX1H | PT0H | PX0H | 00000000b |  |
| **PWM0INTC** | PWM0 Interrupt Control | B7H | 1 | - | - | INTTYP1 | INTTYP0 | - | INTSEL2 | INTSEL1 | INTSEL0 | 00000000b |  |
| **-** | - | B7H | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | B7H | 3 | - | - | - | - | - | - | - | - | - |  |
| **I2C1TOC** | I2c1 Timeout Counter | B6H | 0 | - | - | - | - | - | I2TOCEN | DIV | I2TOF | 00000000b |  |
| **P2M2** | Port5 Mode Select 2 | B6H | 1 | P2M2.7 | P2M2.6 | P2M2.5 | P2M2.4 | P2M2.3 | P2M2.2 | P2M2.1 | P2M2.0 | 00000000b |  |
| **-** | - | B6H | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | B6H | 3 | - | - | - | - | - | - | - | - | - |  |
| **I2C1CLK** | I2c1 Clock | B5H | 0 | I2C1CLK[7:0] | | | | | | | | 00000000b |  |
| **P2M1** | Port2 Mode Select 1 | B5H | 1 | P2M1.7 | P2M1.6 | P2M1.5 | P2M1.4 | P2M1.3 | P2M1.2 | P2M1.1 | P2M1.0 | 11111111b |  |
| **-** | - | B5H | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | B5H | 3 | - | - | - | - | - | - | - | - | - |  |
| **I2C1STAT** | I2c1 Status | B4H | 0 | I2C1STAT\_7 | I2C1STAT\_6 | I2C1STAT\_5 | I2C1STAT\_4 | I2C1STAT\_3 | - | - | - | 11111000b |  |
| **P1M2** | P1 Mode Select 2 | B4H | 1 | P1M2.7 | P1M2.6 | P1M2.5 | P1M2.4 | P1M2.3 | P1M2.2 | P1M2.1 | P1M2.0 | 00000000b |  |
| **-** | - | B4H | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | B4H | 3 | - | - | - | - | - | - | - | - | - |  |
| **I2C1DAT** | I2c1 Data | B3H | 0 | I2C1DAT[7:0] | | | | | | | | 00000000b |  |
| **P1M1** | P1 Mode Select 1 | B3H | 1 | P1M1.7 | P1M1.6 | P1M1.5 | P1M1.4 | P1M1.3 | P1M1.2 | P1M1.1 | P1M1.0 | 11111111b |  |
| **-** | - | B3H | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | - | 3 | - | - | - | - | - | - | - | - | - |  |
| **I2C1ADDR0** | I2c1 Own Slave Address 0 | B2H | 0 | I2C1ADDR0\_7 | I2C1ADDR0\_6 | I2C1ADDR0\_5 | I2C1ADDR0\_4 | I2C1ADDR0\_3 | I2C1ADDR0\_2 | I2C1ADDR0\_1 | GC | 00000000b |  |
| **P0M2** | P0 Mode Select 2 | B2H | 1 | P0M2.7 | P0M2.6 | P0M2.5 | P0M2.4 | P0M2.3 | P0M2.2 | P02M2.1 | P2M2.0 | 00000000b |  |
| **-** | - | B2H | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | B2H | 3 | - | - | - | - | - | - | - | - | - |  |
| **P5** | Port 5 | B1H | 0 | P5.7 | P5.6 | P5.5 | P5.4 | P5.3 | P5.2 | P5.1 | P5.0 | Outputlatch, 11111111b Input, XXXXXXXXb |  |
| **P0M1** | P0 Mode Select 1 | B1H | 1 | P0M1.7 | P0M1.6 | P0M1.5 | P0M1.4 | P0M1.3 | P0M1.2 | P0M1.1 | P0M1.0 | 11111111b |  |
| **-** | - | B1H | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | B1H | 3 | - | - | - | - | - | - | - | - | - |  |
| **P3** | Port 3 | B0H | A | P3.7 | P3.6 | P3.5 | P3.4 | P3.3 | P3.2 | P3.1 | P3.0 | Outputlatch, 11111111b Input, XXXXXXXXb |  |
| **IAPCN** | IAP Control | AFH | 0 | IAPA\_17 | IAPA\_16 | FOEN | FCEN | FCTRL\_3 | FCTRL\_2 | FCTRL\_1 | FCTRL\_0 | 00110000b |  |
| **PIPS7** | Pin Interrupt Control Register 7 | AFH | 1 |  | PSEL\_2 | PSEL\_1 | PSEL\_0 |  | BSEL\_2 | BSEL\_1 | BSEL\_0 | 00000000b |  |
| **-** | - | AFH | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | AFH | 3 | - | - | - | - | - | - | - | - | - |  |
| **IAPFD** | IAP Flash Data | AEH | 0 | IAPFD[7:0] | | | | | | | | 00000000b |  |
| **P5SR** | P5 Slew Rate | AEH | 1 | P5SR.7 | P5SR.6 | P5SR.5 | P5SR.4 | P5SR.3 | P5SR.2 | P5SR.1 | P5SR.0 | 00000000b |  |
| **-** | - | AEH | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | AEH | 3 | - | - | - | - | - | - | - | - | - |  |
| **EIPH2** | Extensive Interrupt Priority High 2 | ADH | 0 | - | - | - | - | - | - | PI2C1H | PACMPH | 00000000b |  |
| **P3SR** | P3 Slew Rate | ADH | 1 | P3SR.7 | P3SR.6 | P3SR.5 | P3SR.4 | P3SR.3 | P3SR.2 | P3SR.1 | P3SR.0 | 00000000b |  |
| **-** | - | ADH | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | ADH | 3 | - | - | - | - | - | - | - | - | - |  |
| **EIP2** | Extensive Interrupt Priority 2 Rcnt | ACH | 0 | - | - | - | - | - | - | PI2C1 | PACMP | 00000000b |  |
| **P3S** | Port 3 Schmitt Triggered Input | ACH | 1 | P3S.7 | P3S.6 | P3S.5 | P3S.4 | P3S.3 | P3S.2 | P3S.1 | P3S.0 | 00000000b |  |
| **-** | - | ACH | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | ACH | 3 | - | - | - | - | - | - | - | - | - |  |
| **BODCON1** | Brown-out Detection Control 1 | ABH | 0 | - | - | - | - | - | LPBOD\_1 | LPBOD\_0 | BODFLT | POR: 00000001b Others: 00000UUUb | **Y** |
| **ACMPCR2** | Analog Comparator Control Register 2 | ABH | 1 | SPEED1\_1 | SPEED1\_0 | POE1 | POE0 | SPEED0\_1 | SPEED0\_0 | - | CRVEN | 00000000b |  |
| **-** | - | ABH | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | ABH | 3 | - | - | - | - | - | - | - | - | - |  |
| **WDCON** | Watchdog Timer Control | AAH | 0 | WDTR | WDCLR | WDTF | WIDPD | WDTRF | WDPS\_2 | WDPS\_1 | WDPS\_0 | POR: 00000111b WDT: 00001UUUbOthers: 0000UUUUb | **Y** |
| **-** | - | AAH | 1 | - | - | - | - | - | - | - | - | - | **-** |
| **-** | - | AAH | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | AAH | 3 | - | - | - | - | - | - | - | - | - |  |
| **SADDR0** | Slave 0 Address | A9H | 0 | SADDR0[7:0] | | | | | | | | 00000000b |  |
| **-** | - | A9H | 1 | - | - | - | - | - | - | - | - | - | **-** |
| **-** | - | A9H | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | A9H | 3 | - | - | - | - | - | - | - | - | - |  |
| **IE** | Interrupt Enable | A8H | A | EA | EADC | EBOD | ES | ET1 | EX1 | ET0 | EX0 | 00000000b |  |
| **IAPAH** | IAP Address High Byte | A7H | 0 | IAPA[15:8] | | | | | | | | 00000000b |  |
| **PIPS6** | Pin Interrupt Control 6 | A7H | 1 |  | PSEL\_2 | PSEL\_1 | PSEL\_0 |  | BSEL\_2 | BSEL\_1 | BSEL\_0 | 00000000b |  |
| **-** | - | A7H | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | A7H | 3 | - | - | - | - | - | - | - | - | - |  |
| **IAPAL** | IAP Address Low Byte | A6H | 0 | IAPA[7:0] | | | | | | | | 00000000b |  |
| **PIPS5** | Pin Interrupt Control 5 | A6H | 1 | - | PSEL\_2 | PSEL\_1 | PSEL\_0 | - | BSEL\_2 | BSEL\_1 | BSEL\_0 | 00000000b |  |
| **I2C1ADDR3** | I2cn Own Slave Address | A6H | 2 | I2C1ADDR3\_7 | I2C1ADDR3\_6 | I2C1ADDR3\_5 | I2C1ADDR3\_4 | I2C1ADDR3\_3 | I2C1ADDR3\_2 | I2C1ADDR3\_1 | GC | 00000000b |  |
| **-** | - | A6H | 3 | - | - | - | - | - | - | - | - | - |  |
| **IAPUEN** | IAP Update Enable | A5H | 0 | - | - | - | SPMEN | SPUEN | CFUEN | LDUEN | APUEN | 00000000b | **Y** |
| **PIPS4** | Pin Interrupt Control 4 | A5H | 1 | - | PSEL\_2 | PSEL\_1 | PSEL\_0 | - | BSEL\_2 | BSEL\_1 | BSEL\_0 | 00000000b |  |
| **I2C1ADDR2** | I2C1 Own Slave Address | A5H | 2 | I2C1ADDR2\_7 | I2C1ADDR2\_6 | I2C1ADDR2\_5 | I2C1ADDR2\_4 | I2C1ADDR2\_3 | I2C1ADDR2\_2 | I2C1ADDR2\_1 | GC | 00000000b |  |
| **-** | - | A5H | 3 | - | - | - | - | - | - | - | - | - |  |
| **IAPTRG** | IAP Trigger | A4H | 0 | - | - | - | - | - | - | - | IAPGO | 00000000b | **Y** |
| **PIPS3** | Pin Interrupt Control3 | A4H | 1 | - | PSEL\_2 | PSEL\_1 | PSEL\_0 | - | BSEL\_2 | BSEL\_1 | BSEL\_0 | 00000000b |  |
| **I2C1ADDR1** | I2cn Own Slave Address | A4H | 2 | I2C1ADDR1\_7 | I2C1ADDR1\_6 | I2C1ADDR1\_5 | I2C1ADDR1\_4 | I2C1ADDR1\_3 | I2C1ADDR1\_2 | I2C1ADDR1\_1 | GC | 00000000b |  |
| **-** | - | A4H | 3 | - | - | - | - | - | - | - | - | - |  |
| **BODCON0** | Brown-out Detection Control 0 | A3H | 0 | BODEN | BOV2 | BOV1 | BOV0 | BOF | BORST | BORF | BOS | POR: CCCCXC0XbBOD: UUUUXU1XbOthers: UUUUXUUXb | **Y** |
| **PIPS2** | Pin Interrupt Control 2 | A3H | 1 | - | PSEL\_2 | PSEL\_1 | PSEL\_0 | - | BSEL\_2 | BSEL\_1 | BSEL\_0 | 00000000b |  |
| **I2C0ADDR3** | I2cn Own Slave Address | A3H | 2 | I2C0ADDR3\_7 | I2C0ADDR3\_6 | I2C0ADDR3\_5 | I2C0ADDR3\_4 | I2C0ADDR3\_3 | I2C0ADDR3\_2 | I2C0ADDR3\_1 | GC | 00000000b |  |
| **-** | - | A3H | 3 | - | - | - | - | - | - | - | - | - |  |
| **AUXR0** | Auxiliary Register 0 | A2H | 0 | SWRF | RSTPINF | HardF | HardFInt | GF2 | - | 0 | DPS | 00000000b |  |
| **PIPS1** | Pin Interrupt Control 1 | A2H | 1 | - | PSEL\_2 | PSEL\_1 | PSEL\_0 | - | BSEL\_2 | BSEL\_1 | BSEL\_0 | 00000000b |  |
| **I2C0ADDR2** | I2cn Own Slave Address | A2H | 2 | I2C0ADDR2\_7 | I2C0ADDR2\_6 | I2C0ADDR2\_5 | I2C0ADDR2\_4 | I2C0ADDR2\_3 | I2C0ADDR2\_2 | I2C0ADDR2\_1 | GC | 00000000b |  |
| **-** | - | A2H | 3 | - | - | - | - | - | - | - | - | - |  |
| **ADCCON0** | Adc Control Register 0 | A1H | 0 | ADCF | ADCS | ETGSEL1 | ETGSEL0 | ADCHS3 | ADCHS2 | ADCHS1 | ADCHS0 | 00000000b |  |
| **PIPS0** | Pin Interrupt Control 1 | A1H | 1 | - | PSEL\_2 | PSEL\_1 | PSEL\_0 | - | BSEL\_2 | BSEL\_1 | BSEL\_0 | 00000000b |  |
| **I2C0ADDR1** | I2C0 Own Slave Address | A1H | 2 | I2C0ADDR1\_7 | I2C0ADDR1\_6 | I2C0ADDR1\_5 | I2C0ADDR1\_4 | I2C0ADDR1\_3 | I2C0ADDR1\_2 | I2C0ADDR1\_1 | - | 00000000b |  |
| **-** | - | A1H | 3 | - | - | - | - | - | - | - | - | - | **-** |
| **P2** | Port 2 | A0H | A | P2.7 | P2.6 | P2.5 | P2.4 | P2.3 | P2.2 | P2.1 | P2.0 | Outputlatch, 11111111b Input, XXXXXXXXb |  |
| **CHPCON** | Chip Control | 9FH | 0 | SWRST | IAPFF | - | - | - | - | BS | IAPEN | Software, 000000U0b Others, 000000C0b | **Y** |
| - | - | 9FH | 1 | - | - | - | - | - | - | - | - | - |  |
| - | - | 9FH | 2 | - | - | - | - | - | - | - | - | - |  |
| - | - | 9FH | 3 | - | - | - | - | - | - | - | - | - |  |
| - | - | 9EH | 0 | - | - | - | - | - | - | - | - | - | **-** |
| **P2SR** | P2 Slew Rate | 9EH | 1 | P2SR.7 | P2SR.6 | P2SR.5 | P2SR.4 | P2SR.3 | P2SR.2 | P2SR.1 | P2SR.0 | 00000000b |  |
| - | - | 9EH | 2 | - | - | - | - | - | - | - | - | - |  |
| - | - | 9EH | 3 | - | - | - | - | - | - | - | - | - |  |
| **RSR** | Reset Flag Register | 9DH | 0 | LVRF | PORF | HFRF | POF | RSTPINF | BORF | WDTRF | SWRF | 11010000b |  |
| **P2S** | Port2 Schmitt Triggered Input | 9DH | 1 | P2S.7 | P2S.6 | P2S.5 | P2S.4 | P2S.3 | P2S.2 | P2S.1 | P2S.0 | 00000000b |  |
| **-** | - | 9DH | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | 9DH | 3 | - | - | - | - | - | - | - | - | - |  |
| **EIE1** | Extensive Interrupt Enable 1 | 9CH | 0 | - | - | EI2C1 | ESPI1 | EHFI | EWKT | ET3 | ES1 | 00000000b |  |
| **P1SR** | P1 Slewrate | 9CH | 1 | P1SR.7 | P1SR.6 | P1SR.5 | P1SR.4 | P1SR.3 | P1SR.2 | P1SR.1 | P1SR.0 | 00000000b |  |
| **-** | - | 9CH | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | 9CH | 3 | - | - | - | - | - | - | - | - | - |  |
| **EIE0** | Extensive Interrupt Enable 1 | 9BH | 0 | ET2 | ESPI0 | EFB0 | EWDT | EPWM0 | ECAP | EPI | EI2C0 | 00000000b |  |
| **P1S** | Port1 Schmitt Triggered Input | 9BH | 1 | P1S.7 | P1S.6 | P1S.5 | P1S.4 | P1S.3 | P1S.2 | P1S.1 | P1S.0 | 00000000b |  |
| **-** | - | - | 2 | - | - | - | - | - | - | - | - | - |  |
| **DMA1SEED** | PDMA1 CRC Seed | 9BH | 3 | SEED[7:0] | | | | | | | | 00000000b |  |
| **SBUF1** | Serial Port1 Data Buffer | 9AH | 0 | SBUF1[7:0] | | | | | | | | 00000000b |  |
| **P0SR** | P0 Slew Rate | 9AH | 1 | P0SR.7 | P0SR.6 | P0SR.5 | P0SR.4 | P0SR.3 | P0SR.2 | P0SR.1 | P0SR.0 | 00000000b |  |
| **-** | - | 9AH | 2 | - | - | - | - | - | - | - | - | - |  |
| **DMA0SEED** | PDMA0 CRC Seed | 9AH | 3 | SEED[7:0] | | | | | | | | 00000000b |  |
| **SBUF** | Serial Port0 Data Buffer | 99H | 0 | SBUF[7:0] | | | | | | | | 00000000b |  |
| **P0S** | Port0 Schmitt Triggered Input | 99H | 1 | P0S.7 | P0S.6 | P0S.5 | P0S.4 | P0S.3 | P0S.2 | P0S.1 | P0S.0 | 00000000b |  |
| **-** | - | 99H | 2 | - | - | - | - | - | - | - | - | - |  |
| **WDCON1** | Watchdog Timer Control register 1 | 99H | 3 | - | - | - | - | - | - | - | WDPS | 00000000b | **Y** |
| **SCON** | Serial Port0 Control | 98H | A | SM0/FE | SM1 | SM2 | REN | TB8 | RB8 | TI | RI | 00000000b |  |
| **CKEN** | Clock Enable | 97H | 0 | - | - | - | LIRCEN | - | - | MIRCEN | CKSWTF | 00110100b | **Y** |
| **P5UP** | Port5 Pull Up Resister Control | 97H | 1 | P5UP.7 | P5UP.6 | P5UP.5 | P5UP.4 | P5UP.3 | P5UP.2 | P5UP.1 | P5UP.0 | 00000000b |  |
| - | - | 97H | 2 | - | - | - | - | - | - | - | - | - |  |
| - | - | 97H | 3 | - | - | - | - | - | - | - | - | - |  |
| **CKSWT** | Clock Switch | 96H | 0 | - | - | - | LIRCST | ECLKST | OSC\_2 | OSC\_1 | OSC\_0 | 00110000b | **Y** |
| **P4UP** | Port4 Pull Up Resister Control | 96H | 1 | P4UP.7 | P4UP.6 | P4UP.5 | P4UP.4 | P4UP.3 | P4UP.2 | P4UP.1 | P4UP.0 | 00000000b |  |
| - | - | 96H | 2 | - | - | - | - | - | - | - | - | - |  |
| - | - | 96H | 3 | - | - | - | - | - | - | - | - |  |  |
| **DMA0CCNT** | PDMA0 Current Transfer Count | 95H | 0 | CCNT[7:0] | | | | | | | | 00000000b |  |
| **P3UP** | Port3 Pull Up Resister Control | 95H | 1 | P3UP.7 | P3UP.6 | P3UP.5 | P3UP.4 | P3UP.3 | P3UP.2 | P3UP.1 | P3UP.0 | 00000000b |  |
| **-** | - | 95H | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | 95H | 3 | - | - | - | - | - | - | - | - | - |  |
| **DMA0CNT** | PDMA0 Transfer Count | 94H | 0 | CNT[7:0] | | | | | | | | 00000000b |  |
| **P2UP** | Port2 Pull Up Resister Control | 94H | 1 | P2UP.7 | P2UP.6 | P2UP.5 | P2UP.4 | P2UP.3 | P2UP.2 | P2UP.1 | P2UP.0 | 00000000b |  |
| **-** | - | 94H | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | 94H | 3 | - | - | - | - | - | - | - | - | - |  |
| **DMA0MAL** | PDMA0 XRAM Base Address Low Byte | 93H | 0 | MAL[7:0] | | | | | | | | 00000000b |  |
| **P1UP** | Port1 Pull Up Resister Control | 93H | 1 | P1UP.7 | P1UP.6 | P1UP.5 | P1UP.4 | P1UP.3 | P1UP.2 | P1UP.1 | P1UP.0 | 00000000b |  |
| **-** | - | 93H | 2 | - | - | - | - | - | - | - | - | - |  |
| **DMA1CRC** | PDMA1 CRC Checksum | 93H | 3 | CRC[7:0] | | | | | | | | 00000000b |  |
| **DMA0CR0** | PDMA0 Control Register | 92H | 0 | PSSEL\_3 | PSSEL\_2 | PSSEL\_1 | PSSEL\_0 | HIE | FIE | RUN | EN | 00000000b |  |
| **P0UP** | Port0 Pull Upresister Control | 92H | 1 | P0UP.7 | P0UP.6 | P0UP.5 | P0UP.4 | P0UP.3 | P0UP.2 | P0UP.1 | P0UP.0 | 00000000b |  |
| **-** | - | 92H | 2 | - | - | - | - | - | - | - | - | - |  |
| **DMA0CRC** | PDMA0 CRC Checksum | 92H | 3 | CRC[7:0] | | | | | | | | 00000000b |  |
| **SFRS** | Sfr Page Selection | 91H | A | - | - | - | - | - | - | SFRPAGE\_1 | SFRPAGE\_0 | 00000000b |  |
| **P1** | Port1 | 90H | A | P1\_7 | P1\_6 | P1\_5 | P1\_4 | P1\_3 | P1\_2 | P1\_1 | P1\_0 | Outputlatch, 11111111b Input, XXXXXXXXb |  |
| **WKCON** | Self Wakeup Timer Control | 8FH | 0 | - | - | - | WKTF | WKTR | WKPS\_2 | WKPS\_1 | WKPS\_0 | 00000000b |  |
| **P5DW** | Port 5 Pull Down Resister Control | 8FH | 1 | P5DW.7 | P5DW.6 | P5DW.5 | P5DW.4 | P5DW.3 | P5DW.2 | P5DW.1 | P5DW.0 | 00000000b |  |
| **-** | - | 8FH | 2 | - | - | - | - | - | - | - | - | - |  |
| **PDL** | Power Down Level | 8FH | 3 | - | - | - | - | - | - | PDLS1 | PDLS0 | 00000000b |  |
| **CKCON** | Clock Control | 8EH | 0 | FASTWK | PWMCKS | T1OE | T1M | T0M | T0OE | CLOEN | - | 10000010b |  |
| **P4DW** | Port4 Pull Down Resister Control | 8EH | 1 | P4DW.7 | P4DW.6 | P4DW.5 | P4DW.4 | P4DW.3 | P4DW.2 | P4DW.1 | P4DW.0 | 00000000b |  |
| **-** | - | 8EH | 2 | - | - | - | - | - | - | - | - | - |  |
| **T2ACMP** | TIMER2 and ACMP Output Connection | 8EH | 3 | T2CKS1 | T2CKS0 | IC2S1 | IC2S0 | IC1S1 | IC1S0 | IC0S1 | IC0S0 | 00000000b |  |
| **TH1** | Timer 1 High Byte | 8DH | 0 | TH1[7:0] | | | | | | | | 00000000b |  |
| **P3DW** | Port3 Pull Down Resister Control | 8DH | 1 | P3DW.7 | P3DW.6 | P3DW.5 | P3DW.4 | P3DW.3 | P3DW.2 | P3DW.1 | P3DW.0 | 00000000b |  |
| **-** | - | 8DH | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | 8DH | 3 | - | - | - | - | - | - | - | - | - |  |
| **TH0** | Timer 0 High Byte | 8CH | 0 | TH0[7:0] | | | | | | | | 00000000b |  |
| **P2DW** | Port2 Pull Down Resister Control | 8CH | 1 | P2DW.7 | P2DW.6 | P2DW.5 | P2DW.4 | P2DW.3 | P2DW.2 | P2DW.1 | P2DW.0 | 00000000b |  |
| **-** | - | 8CH | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | 8CH | 3 | - | - | - | - | - | - | - | - | - |  |
| **TL1** | Timer 1 Low Byte | 8BH | 0 | TL1[7:0] | | | | | | | | 00000000b |  |
| **P1DW** | Port1 Pull Down Resister Control | 8BH | 1 | P1DW.7 | P1DW.6 | P1DW.5 | P1DW.4 | P1DW.3 | P1DW.2 | P1DW.1 | P1DW.0 | 00000000b |  |
| **-** | - | 8BH | 2 | - | - | - | - | - | - | - | - | - |  |
| **DMA1CR1** | PDMA1 Control Register 1 | 8BH | 3 | - | - | - | - | XOROUT | REFOUT | REFIN | CRCEN | 00000000b |  |
| **TL0** | Timer 0 Low Byte | 8AH | 0 | TL0[7:0] | | | | | | | | 00000000b |  |
| **P0DW** | Port0 Pull Down Resister Control | 8AH | 1 | P0DW.7 | P0DW.6 | P0DW.5 | P0DW.4 | P0DW.3 | P0DW.2 | P0DW.1 | P0DW.0 | 00000000b |  |
| **-** | - | 8AH | 2 | - | - | - | - | - | - | - | - | - |  |
| **DMA0CR1** | PDMA0 Control Register 1 | 8AH | 3 | - | - | - | - | XOROUT | REFOUT | REFIN | CRCEN | 00000000b |  |
| **TMOD** | Timer 0 And 1 Mode | 89H | A | GATE | C/T | M1 | M0 | GATE | C/T | M1 | M0 | 00000000b |  |
| **TCON** | Timer 0 And 1 Control | 88H | A | TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 | 00000000b |  |
| **PCON** | Power Control | 87H | A | SMOD | SMOD0 | - | POF | GF1 | GF0 | PD | IDL | POR, 00010000b Others, 000U0000b |  |
| **RWKL** | Self Wakeup Timer Reload Lowbyte | 86H | 0 | RWK[7:0] | | | | | | | | 00000000b |  |
| **CWKL** | Self Wakeup Timer Current Count Low Byte | 86H | 1 | CWK[7:0] | | | | | | | | 00000000b |  |
| **-** | - | 86H | 2 | - | - | - | - | - | - | - | - | - |  |
| **-** | - | 86H | 3 | - | - | - | - | - | - | - | - |  |  |
| **-** | - | 85H | 0 | - | - | - | - | - | - | - | - | - | **-** |
| **-** | - | 85H | 1 | - | - | - | - | - | - | - | - |  | **-** |
| **-** | - | 85H | 2 | - | - | - | - | - | - | - | - | - |  |
| **MRCTRIM1** | MIRC TRIM Low byte | 85H | 3 | - | - | - | - | - | - | MRCTRIM\_1 | MRCTRIM\_0 | - |  |
| - | - | 84H | 0 | - | - | - | - | - | - | - | - | - | **-** |
| **LIRCTRIM** | LIRC Trim Value | 84H | 1 | LIRCTRIM[7:0] | | | | | | | | XXXXXXXXb |  |
| **-** | - | 84H | 2 | - | - | - | - | - | - | - | - | - |  |
| **MRCTRIM0** | MIRC TRIM High byte | 84H | 3 | MIRCTRIM[7:0] | | | | | | | | XXXXXXXXb |  |
| **DPH** | Data Pointer High Byte | 83H | A | DPTR[15:8] | | | | | | | | 00000000b |  |
| **DPL** | Data Pointer Low Byte | 82H | A | DPTR[7:0] | | | | | | | | 00000000b |  |
| **SP** | Stack Pointer | 81H | A | SP[7:0] | | | | | | | | 00000111b |  |
| **P0** | Port 0 | 80H | A | P0.7 | P0.6 | P0.5 | P0.4 | P0.3 | P0.2 | P0.1 | P0.0 | Output latch, 11111111b Input, XXXXXXXXb |  |

#### All SFR Description

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

##### Pn – Genral Port Input Output

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P0 | 80H, All Pages, Bit-addressable | 1111\_1111 b |
| P1 | 90H, All Pages, Bit-addressable | 1111\_1111 b |
| P2 | A0H, All Pages, Bit-addressable | 1111\_1111 b |
| P3 | B0H, All Pages, Bit-addressable | 1111\_1111 b |
| P4 | D8H, All Pages, Bit-addressable | 1111\_1111 b |
| P5 | B1H, Page 0 | 1111\_1111 b |

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

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

##### SP – Stack Pointer

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SP | 81H, All Pages | 0000\_0111b |

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

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

##### DPL – Data Pointer Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| DPL | 82H, All Pages | 0000\_0000b |

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

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

##### DPH – Data Pointer High Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| DPH | 83H, All Pages | 0000\_0000b |

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

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

##### LIRCTRIM – Low Speed Internal Oscillator Trim (TA Protected)

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| LIRCTRIM | 84H, Page 1 | XXXX\_XXXXb |

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

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

##### MIRCTRIM0 – Median Speed Internal Oscillator Trim 0

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| MIRCTRIM | 84H, Page 3, TA protected | XXXX\_XXXXb |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **MIRCTRIM[9:2]** | | | | | | | |
| R/W | | | | | | | |

|  |  |  |
| --- | --- | --- |
| **Bit** | **Name** | **Description** |
| [9:3] | **MIRCTRIM[9:2]** | **Median Speed Internal Oscillator Trim Value High Byte** |

##### MIRCTRIM1 – Median Speed Internal Oscillator Trim 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| MIRCTRIM1 | 85H, Page 3, TA protected | XXXX\_XXXXb |

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

|  |  |  |
| --- | --- | --- |
| **Bit** | **Name** | **Description** |
| [1:0] | **MIRCTRIM[1:0]** | **Median Speed Internal Oscillator Trim Value Low Byte** |

##### RWKL – Self Wake-up Timer Reload Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| RWKL | 86H, Page 0 | 0000\_0000b |

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

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

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

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| CWKL | 86H, Page 1 | 0000\_0000b |

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

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

##### PCON – Power 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 0 Mode / Baud Rate Description for details. |
| [6] | **SMOD0** | **Serial Port 0 Framing Error Flag Access Enable**  0 = SCON.7 accesses to SM0 bit.  1 = SCON.7 accesses to FE bit. |
| [5] | **-** | **Reserved** |
| [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 INT1 input signal’s logic level. Software cannot control it. |
| [2] | **IT1** | **External Interrupt 1 Type Select**  This bit selects by which type that INT1 is triggered.  0 = INT1 is low level triggered.  1 = INT1 is falling edge triggered. |
| [1] | **IE0** | **External Interrupt 0 Edge Flag**  If IT0 = 1 (falling edge trigger), this flag will be set by hardware when a falling edge is detected. It remain set until cleared via software or cleared by hardware in the beginning of its interrupt service routine.  If IT0 = 0 (low level trigger), this flag follows the inverse of the INT0 input signal’s logic level. Software cannot control it. |
| [0] | **IT0** | **External Interrupt 0 Type Select**  This bit selects by which type that INT0 is triggered.  0 = INT0 is low level triggered.  1 = INT0 is falling edge triggered. |

##### TMOD – Timer 0 and 1 Mode

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| TMOD | 89H, All Pages | 0000\_0000b |

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

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

##### TL0 – Timer 0 Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| TL0 | 8AH, Page 0 | 0000\_0000b |

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

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

##### PnDW – Port n Pull-down Resister Control

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

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

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

##### TL1 – Timer 1 Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| TL1 | 8BH, Page 0 | 0000\_0000b |

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

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

##### TH0 – Timer 0 High Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| TH0 | 8CH, Page 0 | 0000\_0000b |

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

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

##### TH1 – Timer 1 High Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| TH1 | 8DH, Page 0 | 0000\_0000b |

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

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

##### CKCON – Clock Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| CKCON | 8EH, Page 0 | 1000\_0010b |

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

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

##### WKCON – Self Wake-up Timer Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| WKCON | 8FH, Page 0 | 0000\_0000b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | **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. |

##### SFRS – SFR Page Selection

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SFRS | 91H, All Pages | 0000\_0000b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:2] | **-** | **Reserved** |
| [1:0] | **SFRPAGE[1:0]** | **SFR Page Select**  00 = Instructions access SFR Page 0.  01 = Instructions access SFR Page 1.  10 = Instructions access SFR page 2.  11 = Instructions access SFR page 3. |

##### DMAnCR – PDMAn Control Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| DMA0CR0 | 92H, Page 0 | 0000\_0000 b |
| DMA1CR0 | EBH, Page 0 | 0000\_0000 b |

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

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

##### PnUP – Port n Pull-up Resister Control

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

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

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

##### DMAnMA – PDMA XRAM Base Address Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| DMA0MAL | 93H, Page 0 | 0000\_0000 b |
| DMA1MAL | ECH, Page 0 | 0000\_0000 b |

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

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

##### DMAnCNT – PDMA Transfer Count

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| DMA0CNT | 94H, Page 0 | 0000\_0000 b |
| DMA1CNT | EDH, Page 0 | 0000\_0000 b |

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

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

##### DMAnCCNT – PDMA Current Transfer Count

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| DMA0CCNT | 95H, Page 0 | 0000\_0000 b |
| DMA1CCNT | EEH, Page 0 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **CCNT[7:0]** | | | | | | | |
| R | | | | | | | |

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

##### CKSWT – Clock Switch (TA Protected)

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| CKSWT | 96H, Page 0, TA protected | 0001 \_1000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | **LIRCST** | **MIRCST** | **OSC[2:0]** | | |
| R | R | R | R | R | W | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:5] | **-** | **Reserved** |
| [4] | **LIRCST** | **Low Speed Internal Oscillator Status**  0 = Low speed internal oscillator is not stable or is disabled.  1 = Low speed internal oscillator is enabled and stable. |
| [3] | **MIRCST** | **Median Speed Internal Oscillator Status**  0 = Median speed internal oscillator is not stable or disabled.  1 =Median speed internal oscillator is enabled and stable. |
| [2:0] | **OSC[2:0]** | **Oscillator Selection Bits**  This field selects the system clock source.  000 = Internal Median speed oscillator. Defaul value accoding to MIRCEN(CKEN.1) enabled  10x = Internal low speed oscillator according to LIRCEN(CKEN.4) enabled.  Others = Reserved.  Note that this field is write only. The read back value of this field may not correspond to the present system clock source. |

##### T2ACMP – TIMER2 and ACMP Output Connection

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| T2ACMP | 8EH, Page 3 | 0000 \_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **T2CKS[1:0]** | | **IC2S[1:0]** | | **IC1S[1:0]** | | **IC0S[1:0]** | |
| R/W | | R/W | | R/W | | R/W | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **T2CKS[1:0]** | **Timer2 Clock Source Select**  00 = Timer 2 Clock source from FSYS.  01 = Timer 2 Clock source from external P4.0.  10 = Timer 2 Clock source from external pin P5.4.  11 = Timer 2 Clock source from external pin P3.2. |
| [5:4] | **IC2S[1:0]** | **Input Capture 2 Source Select**  00 = IC2.  01 = ACMP0.  10 = ACMP1.  11 = IC2. |
| [3:2] | **IC1S[1:0]** | **Input Capture 1 Source Select**  00 = external IC1 source  connects to internal IC1.  01 = ACMP0 connects to IC1.  10 = ACMP1 connects to IC1.  11 = external IC1 source  connects to internal IC1. |
| [1:0] | **IC0S[1:0]** | **Input Capture 0 Source Select**  00 = IC0.  01 = ACMP0.  10 = ACMP1.  11 = IC0. |

##### CKEN – Clock Enable

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| CKEN | 97H, Page 0, TA protected | 0001\_0110 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | **LIRCEN** | **-** | | **MIRCEN** | **CKSWTF** |
| R/W | R/W | R/W | R/W | R/W | - | R/W | R |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:5] | **-** | **Reserved** |
| [4] | **LIRCEN** | **Low Speed Internal Oscillator 38.4 kHz Enable**  0 = The low speed internal oscillator Disabled.  1 = The low speed internal oscillator Enabled.  Note that when (1)WDT is enabled, (2)WKT is running by the clock source of the internal 38.4 kHz oscillator ,(3) BOD is enabled, or (4)LVR filter is enabled, a write 0 to LIRCEN will be ignored. LIRCEN is always 1 and the internal 38.4 kHz oscillator is always enabled. |
| [3:2] | **-** | **Reserved** |
| [1] | **MIRCEN** | **Median Speed Internal Oscillator Enable**  0 = The medianspeed internal oscillator Disabled.  1 = The median speed internal oscillator Enabled.  Note that once IAP is enabled by setting IAPEN (CHPCON.0), the median speed internal oscillator will be enabled automatically. The hardware will also set MIRCEN and MIRCST bits. After IAPEN is cleared, MIRCEN and MIRCST resume the original values. |
| [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. |

##### PDL – Power Down Level Select

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PDL | 8FH, Page 3 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | **-** | **-** | **-** | **-** | **-** | **PDLS\_1** | **PDLS\_0** |
| - | - | - | - | - | - | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:2] | **-** | **Reserved** |
| [1:0] | **PDLS[1:0]** | **Power Down Level Select**  00 = Power Down Level 0 (defaut).  01 = Power Down Level 1.  10 = Power Down Level 2.  Others = Reserved. |

##### SCON – Serial Port Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SCON | 98H, All Pages, Bit addressable | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **SM0/FE** | **Serial Port Mode Select**  SMOD0 (PCON.6) = 0:.  See Table 6.9‑1 Serial Port 0 Mode / Baud Rate Description for details.  SMOD0 (PCON.6) = 1:.  SM0/FE bit is used as frame error (FE) status flag. It is cleared by software.  0 = Frame error (FE) did not occur.  1 = Frame error (FE) occurred and detected. |
| [6] | **SM1** | Check with bit 7 description. |
| [5] | **SM2** | **Multiprocessor Communication Mode Enable**  The function of this bit is dependent on the serial port 0 mode.  Mode 0:  This bit select the baud rate between FSYS/12 and FSYS/2.  0 = The clock runs at FSYS/12 baud rate. It maintains standard 8051compatibility.  1 = The clock runs at FSYS/2 baud rate for faster serial communication.  Mode 1:  This bit checks valid stop bit.  0 = Reception is always valid no matter the logic level of stop bit.  1 = Reception is valid only when the received stop bit is logic 1 and the received data matches “Given” or “Broadcast” address.  Mode 2 or 3:  For multiprocessor communication.  0 = Reception is always valid no matter the logic level of the 9th bit.  1 = Reception is valid only when the received 9th bit is logic 1 and the received data matches “Given” or “Broadcast” address. |
| [4] | **REN** | **Receiving Enable**  0 = Serial port 0 reception Disabled.  1 = Serial port 0 reception Enabled in Mode 1,2, or 3. In Mode 0, reception is initiated by the condition REN = 1 and RI = 0. |
| [3] | **TB8** | **9th Transmitted Bit**  This bit defines the state of the 9th transmission bit in serial port 0 Mode 2 or 3. It is not used in Mode 0 or 1. |
| [2] | **RB8** | **9th Received Bit**  The bit identifies the logic level of the 9th received bit in serial port 0 Mode 2 or 3. In Mode 1, RB8 is the logic level of the received stop bit. SM2 bit as logic 1 has restriction for exception. RB8 is not used in Mode 0. |
| [1] | **TI** | **Transmission Interrupt Flag**  This flag is set by hardware when a data frame has been transmitted by the serial port 0 after the 8th bit in Mode 0 or the last data bit in other modes. When the serial port 0 interrupt is enabled, setting this bit causes the CPU to execute the serial port 0 interrupt service routine. This bit should be cleared manually via software. |
| [0] | **RI** | **Receiving Interrupt Flag**  This flag is set via hardware when a data frame has been received by the serial port 0 after the 8th bit in Mode 0 or after sampling the stop bit in Mode 1, 2, or 3. SM2 bit as logic 1 has restriction for exception. When the serial port 0 interrupt is enabled, setting this bit causes the CPU to execute to the serial port 0 interrupt service routine. This bit should be cleared manually via software. |

##### SBUF – Serial Port 0 Data Buffer

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SBUF | 99H, Page 0 | 0000\_0000 b |

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

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

##### PnS – Port n Schmitt Triggered Input

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P0S | 99H, Page 1 | 0000\_0000 b |
| P1S | 9BH, Page 1 | 0000\_0000 b |
| P2S | 9DH, Page 1 | 0000\_0000 b |
| P3S | ACH, Page 1 | 0000\_0000 b |
| P4S | BBH, Page 1 | 0000\_0000 b |
| P5S | BFH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **PnS[7:0]** | | | | | | | |
| 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. |

##### WDCON1 – Watchdog Timer Control Register 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| WDCON1 | AAH, Page 0, TA protected | POR 0000\_0001 b  WDT 0000\_000U b  Others 0000\_000U b |

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

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

##### SBUF1 – Serial Port 1 Data Buffer

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SBUF1 | 9AH, Page 0 | 0000 \_0000 b |

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

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

##### PnSR –Port n Slew Rate Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P0SR | 9AH, Page 1 | 0000\_0000 b |
| P1SR | 9CH, Page 1 | 0000\_0000 b |
| P2SR | 9EH, Page 1 | 0000\_0000 b |
| P3SR | ADH, Page 1 | 0000\_0000 b |
| P4SR | BCH, Page 1 | 0000\_0000 b |
| P5SR | AEH, Page 1 | 0000\_0000 b |

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

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

##### EIE0 – Extensive Interrupt Enable

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| EIE0 | 9BH, Page 0 | 0000 \_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **ET2** | **Enable Timer 2 Interrupt**  0 = Timer 2 interrupt Disabled.  1 = Timer 2 interrupt Enable. When interrupt generated, TF2 (T2CON.7) set 1 |
| [6] | **ESPI0** | **Enable SPI Interrupt**  0 = SPI interrupt Disabled.  1 = SPI interrupt Enable. When interrupt generated SPIF (SPInSR.7), SPIOVF (SPInSR.5), or MODF (SPInSR.4) set 1 . |
| [5] | **EFB0** | **Enable Fault Brake Interrupt**  0 = Fault Brake interrupt Disabled.  1 = Fault Brake interrupt Enable. When interrupt generated FBF (PWM0FBD.7) set 1. |
| [4] | **EWDT** | **Enable WDT Interrupt**  0 = WDT interrupt Disabled.  1 = WDT interrupt Enable. When interrupt generated WDTF (WDCON.5) set 1. |
| [3] | **EPWM0** | **Enable PWM0 Interrupt**  0 = PWM0 interrupt Disabled.  1 = PWM0 interrupt Enable. When interrupt generated PWMF (PWMnCON0.5) set 1. |
| [2] | **ECAP** | **Enable Input Capture Interrupt**  0 = Input capture interrupt Disabled.  1 = Input capture interrupt Enable. When interrupt generated CAPF[2:0] (CAPCON0[2:0]) set 1. |
| [1] | **EPI** | **Enable Pin Interrupt**  0 = Pin interrupt Disabled.  1 = Pin interrupt Enable. When interrupt generated PIF related bit set 1. |
| [0] | **EI2C0** | **Enable I2C0 Interrupt**  0 = I2C interrupt Disabled.  1 = I2C interrupt Enable. When interrupt generated SI (I2C0CON.3) or I2TOF (I2C0TOC.0) set 1. |

##### EIE1 – Extensive Interrupt Enable 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| EIE1 | 9CH, Page 0 | 0000 \_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **-** | **Reserved** |
| [5] | **EI2C1** | **Enable I2C1 Interrupt**  0 = I2C1 interrupt Disabled.  1 = I2C1 interrupt Enable. When interrupt generated SI (I2C1CON.3) or I2TOF (I2C1TOC.0) set 1. |
| [4] | **-** | **Reserved** |
| [3] | **EHFI** | **Enable Hard Fault Interrupt**  0 = hard fault interrupt Disabled and hard fault reset is Enabled.  1 = hard fault interrupt Enable. When interrupt generated HFIF (AUXR0.4) set 1. |
| [2] | **EWKT** | **Enable WKT Interrupt**  0 = WKT interrupt Disabled.  1 = WKT interrupt Enable. When interrupt generated WKTF (WKCON.4) set 1. |
| [1] | **ET3** | **Enable Timer 3 Interrupt**  0 = Timer 3 interrupt Disabled.  1 = Timer 3interrupt Enable. When interrupt generated TF3 (T3CON.4) set 1. |
| [0] | **ES1** | **Enable Serial Port 1 Interrupt**  0 = Serial port 1 interrupt Disabled.  1 = Serial port 1 interrupt Enable. When interrupt generated TI\_1 (S1CON.1) or RI\_1 (S1CON.0) set 1. |

##### RSR – Reset Flag Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| RSR | 9DH, Page 0 | 1101\_0000 b |

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

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

##### CHPCON – Chip Control (TA Protected)

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

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

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

##### PIPSn – Pin Interrupt Control

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

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

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

##### AUXR0 – Auxiliary Register 0

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

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

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

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

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

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **BODEN** | **Brown-Out Detection Enable**  0 = Brown-out detection circuit ON.  1 = Brown-out detection circuit OFF.  Note that BOD output is not available until 2~3 LIRC clocks after enabling. |
| [6:4] | **BOV[2:0]** | **CONFIG Brown-Out Voltage Select**  111 = VBOD is 1.8V.  110 = VBOD is 1.8V.  101 = VBOD is 2.0V.  100 = VBOD is 2.4V.  011 = VBOD is 2.7V.  010 = VBOD is 3.0V.  001 = VBOD is 3.7V.  000 = VBOD is 4.4V. |
| [3] | **BOF** | **Brown-Out Interrupt Flag**  This flag will be set as logic 1 via hardware after a VDD dropping below or rising above VBOD event occurs. If both EBOD (I.E.5) and EA (I.E.7) are set, a brown-out interrupt requirement will be generated. This bit should be cleared via software. |
| [2] | **BORST** | **Brown-Out Reset Enable**  This bit decides whether a brown-out reset is caused by a power drop below VBOD.  0 = Brown-out reset when VDD drops below VBOD Disabled.  1 = Brown-out reset when VDD drops below VBOD Enabled. |
| [1] | **BORF** | **Brown-Out Reset Flag**  When the MCU is reset by brown-out event, this bit will be set via hardware. This flag is recommended to be cleared via software. |
| [0] | **BOS** | **Brown-Out Status**  This bit indicates the VDD voltage level comparing with VBOD while BOD circuit is enabled. It keeps 0 if BOD is not enabled.  0 = VDD voltage level is higher than VBOD or BOD is disabled.  1 = VDD voltage level is lower than VBOD.  Note that this bit is read-only. |
| Note:  1. BODEN, BOV[2:0], and BORST are initialized by being directly loaded from CONFIG2 [6:4], and after all resets.  2. BOF reset value depends on different setting of CONFIG2 and VDD voltage level. | | |

##### IAPTRG – IAP Trigger (TA Protected)

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| IAPTRG | A4H, Page 0, TA protected | 0000 \_0000 b |

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

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

##### IAPUEN – IAP Updating Enable (TA Protected)

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| IAPUEN | A5H, Page 0, TA protected | 0000 \_0000 b |

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

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

##### IAPAL – IAP Address Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| IAPAL | A6H, Page 0 | 0000 \_0000 b |

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

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

##### IAPAH – IAP Address High Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| IAPAH | A7H, Page 0 | 0000 \_0000 b |

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

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

##### IE – Interrupt Enable

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| IE | A8H, All Pages, Bit addressable | 0000 \_0000 b |

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

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

##### SADDRn – UART Slave Address

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SADDR0 | A9H, Page 0 | 0000 \_0000 b |
| SADDR1 | BBH, Page 0 | 0000 \_0000 b |

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

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

##### WDCON – Watchdog Timer Control (TA Protected)

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

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

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

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

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

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

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

##### ACMPCR2 – Analog Comparator Control Register 2

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| ACMPCR2 | ABH, Page 1 | 0000 \_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **SPEED1** | **Analog Comparator 1 Speed Control**  00 = slow speed, propagation delay : 4.5us, 1.2uA (typ.).  01 = slow+ speed, propagation delay : 2.0us, 3uA (typ.).  10 = fast speed, propagation delay : 0.6us, 10uA (typ.).  11 = fast+ speed, propagation delay : 0.2us, 75uA (typ.). |
| [5] | **POE1** | **Analog Comparator 1 Polarity Output Enable**  0 = ACMP1 output directly.  1 = ACMP1 output inversely. |
| [4] | **POE0** | **Analog Comparator 0 Polarity Output Enable**  0 = ACMP0 outputs directly.  1 = ACMP0 outputs inversely. |
| [3:2] | **SPEED0** | **Analog Comparator 0 Speed Control**  00 = slow speed, propagation delay : 4.5us, 1.2uA (typ.).  01 = slow+ speed, propagation delay : 2.0us, 3uA (typ.).  10 = fast speed, propagation delay : 0.6us, 10uA (typ.).  11 = fast+ speed, propagation delay : 0.2us, 75uA (typ.). |
| [1] | **-** | **Reserved** |
| [0] | **CRVEN** | **CRV Enable Bit**  0 = CRV Disabled.  1 = CRV Enabled. |

##### EIP2 – Extensive Interrupt Priority 2

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| EIP2 | ACH, Page 0 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | | | **PI2C1** | **PACMP** |
| - | | | | | | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:5] | **-** | **Reserved** |
| [1] | **PI2C1** | I2C interrupt priority low bit |
| [0] | **PACMP** | ACMP interrupt priority low bit |
| **Note:** EIP2 is used in combination with the EIPH2 to determine the priority of each interrupt source. See Table 6.2‑5 Interrupt Priority Level Setting for correct interrupt priority configuration. | | |

##### EIPH2 – Extensive Interrupt Priority High 2

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| EIPH2 | ADH, Page 0 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | | | **PI2C1H** | **PACMPH** |
| - | | | | | | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:2] | **-** | Reserved. |
| [1] | **PI2C1H** | I2C interrupt priority high bit |
| [0] | **PACMPH** | ACMP interrupt priority high bit |
| **Note:** EIPH2 is used in combination with the EIP2 to determine the priority of each interrupt source. See Table 6.2‑5 Interrupt Priority Level Setting for correct interrupt priority configuration. | | |

##### IAPFD – IAP Flash Data

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| IAPFD | AEH, Page 0 | 0000 \_0000 b |

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

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

##### IAPCN – IAP Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| IAPCN | AFH, Page 0 | 0011\_0000 b |

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

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

##### PnM1 – Port n Mode Select 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P0M1 | B1H, Page 1 | 1111\_1111 b |
| P1M1 | B3H, Page 1 | 1111\_1111 b |
| P2M1 | B5H, Page 1 | 1111\_1111 b |
| P3M1 | C2H, Page 1 | 1111\_1111 b |
| P4M1 | B9H, Page 1 | 1111\_1111 b |
| P5M1 | BDH, Page 1 | 1111\_1111 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P0M1[7:0]** | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

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

##### PnM2 – Port n Mode Select 2

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P0M2 | B2H, Page 1 | 0000\_0000 b |
| P1M2 | B4H, Page 1 | 0000\_0000 b |
| P2M2 | B6H, Page 1 | 0000\_0000 b |
| P3M2 | C3H, Page 1 | 0000\_0000 b |
| P4M2 | BAH, Page 1 | 0000\_0000 b |
| P5M2 | BEH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **PnM2[7:0]** | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:0] | **PnM2[7:0]** | Port n mode select 2 |
| **Note:** PnM1 and PnM2 [n:0~5] are used in combination to determine the I/O mode of each pin of P0. See Table 6.4‑1 Configuration for Different I/O Modes | | |

##### I2C1DAT – I2C1 Data

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| I2C1DAT | B3H, Page 0 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:0] | **I2C1DAT[7:0]** | **I2C1 Data**  I2CnDAT contains a byte of the I2C data to be transmitted or a byte, which has just received. Data in I2CnDAT remains as long as SI is logic 1. The result of reading or writing I2CnDAT during I2C transceiver progress is unpredicted.  While data in I2CnDAT is shifted out, data on the bus is simultaneously being shifted in to update I2CnDAT. I2CnDAT always shows the last byte that presented on the I2C bus. Thus the event of lost arbitration, the original value of I2CnDAT changes after the transaction. |

##### PWMnINTC – PWM Interrupt Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0INTC | B7H, Page 1 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **-** | Reserved |
| [5:4] | **INTTYP[1:0]** | **PWM Interrupt Type Select**  These bit select PWM interrupt type.  00 = Falling edge on PWMn\_CH0/1/2/3/4/5 pin.  01 = Rising edge on PWMn\_CH0/1/2/3/4/5 pin.  10 = Central point of a PWM period.  11 = End point of a PWM period.  Note that the central point interrupt or the end point interrupt is only available while PWM operates in center-aligned type. |
| [3] | **-** | Reserved |
| [2:0] | **INTSEL[2:0]** | **PWM Interrupt Pair Select**  These bits select which PWM channel asserts PWM interrupt when PWM interrupt type is selected as falling or rising edge on PWM0/1/2/3/4/5 pin..  000 = PWMn\_CH0.  001 = PWMn\_CH1.  010 = PWMn\_CH2.  011 = PWMn\_CH3.  100 = PWMn\_CH4.  101 = PWMn\_CH5.  Others = PWMn\_CH0. |

##### IP – Interrupt Priority

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| IP | B8H, All Pages, Bit addressable | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **-** | Reserved |
| [5] | **PBOD** | Brown-out detection interrupt priority low bit |
| [4] | **PS** | Serial port 0 interrupt priority low bit |
| [3] | **PT1** | Timer 1 interrupt priority low bit |
| [2] | **PX1** | External interrupt 1 priority low bit |
| [1] | **PT0** | Timer 0 interrupt priority low bit |
| [0] | **PX0** | External interrupt 0 priority low bit |
| **Note:** used in combination with the IPH to determine the priority of each interrupt source. See Table 6.2‑5 Interrupt Priority Level Setting for correct interrupt priority configuration. | | |

##### SADENn – UART Slave n Address Mask

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SADEN0 | B9H, Page 0 | 0000\_0000 b |
| SADEN1 | BAH, Page 0 | 0000\_0000 b |

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

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

##### I2CnDAT – I2C Data

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| I2C0DAT | BCH, Page 0 | 0000\_0000 b |
| I2C1DAT | B3H, Page 0 | 0000\_0000 b |

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

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

##### I2CnSTAT – I2C Status

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| I2C0STAT | BDH, Page 0 | 1111\_1000 b |
| I2C1STAT | B4H, Page 0 | 1111\_1000 b |

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

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

##### I2CnCLK – I2C Clock

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| I2C0CLK | BEH, Page 0 | 0000\_1001 b |
| I2C1CLK | B5H, Page 0 | 0000\_1001 b |

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

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

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

##### AUXR3 – Auxiliary Register 3

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| AUXR3 | CFH, Page 3 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | **-** | **-** | **~~-~~** | **~~-~~** | **UART2DG** | **UART1DG** | **UART0DG** |
| - | - | - | - | **~~-~~** | R/W | R/W | R/W |

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

##### I2CnTOC – I2Cn Time-out Counter

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| I2C0TOC | BFH, Page 0 | 0000\_0000b |
| I2C1TOC | B6H, Page 0 | 0000\_0000b |

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

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

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

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| CWKH | BEH, Page 2 | 0000\_0000 b |

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

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

##### RWKH – Self Wake-up Timer Reload High Byte

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

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

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

##### I2CnCON – I2C Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| I2C0CON | C0H, All Pages, Bit-addressable | 0000\_0000 b |
| I2C1CON | E8H, All Pages | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **I** | **I2Cn Hold Time Extend Enable**  0 = I2C DATA to SCL hold time extend disabled.  1 = I2C DATA to SCL hold time extend enabled, extend 8 system clock. |
| [6] | **I2CEN** | **I2Cn Bus Enable**  0 = I2C bus Disabled.  1 = I2C bus Enabled.  Before enabling the I2C, SCL and SDA port latches should be set to logic 1. |
| [5] | **STA** | **START Flag**  When STA is set, the I2C generates a START condition if the bus is free. If the bus is busy, the I2C waits for a STOP condition and generates a START condition following.  If STA is set while the I2C is already in the master mode and one or more bytes have been transmitted or received, the I2C generates a repeated START condition.  Note that STA can be set anytime even in a slave mode, but STA is not hardware automatically cleared after START or repeated START condition has been detected. User should take care of it by clearing STA manually. |
| [4] | **STO** | **STOP Flag**  When STO is set if the I2C is in the master mode, a STOP condition is transmitted to the bus. STO is automatically cleared by hardware once the STOP condition has been detected on the bus.  The STO flag setting is also used to recover the I2C device from the bus error state (I2CnSTAT as 00H). In this case, no STOP condition is transmitted to the I2C bus.  If the STA and STO bits are both set and the device is original in the master mode, the I2C bus will generate a STOP condition and immediately follow a START condition. If the device is in slave mode, STA and STO simultaneous setting should be avoid from issuing illegal I2C frames. |
| [3] | **SI** | **I2Cn Interrupt Flag**  SI flag is set by hardware when one of 26 possible I2C status (besides F8H status) is entered. After SI is set, the software should read I2CnSTAT register to determine which step has been passed and take actions for next step.  SI is cleared by software. Before the SI is cleared, the low period of SCL line is stretched. The transaction is suspended. It is useful for the slave device to deal with previous data bytes until ready for receiving the next byte.  The serial transaction is suspended until SI is cleared by software. After SI is cleared, I2C bus will continue to generate START or repeated START condition, STOP condition, 8-bit data, or so on depending on the software configuration of controlling byte or bits. Therefore, user should take care of it by preparing suitable setting of registers before SI is software cleared. |
| [2] | **AA** | **Acknowledge Assert Flag**  If the AA flag is set, an ACK (low level on SDA) will be returned during the acknowledge clock pulse of the SCL line while the I2C device is a receiver or an own-address-matching slave.  If the AA flag is cleared, a NACK (high level on SDA) will be returned during the acknowledge clock pulse of the SCL line while the I2C device is a receiver or an own-address-matching slave. A device with its own AA flag cleared will ignore its own salve address and the General Call. Consequently, SI will note be asserted and no interrupt is requested.  Note that if an addressed slave does not return an ACK under slave receiver mode or not receive an ACK under slave transmitter mode, the slave device will become a not addressed slave. It cannot receive any data until its AA flag is set and a master addresses it again.  There is a special case of I2CnSTAT value C8H occurs under slave transmitter mode. Before the slave device transmit the last data byte to the master, AA flag can be cleared as 0. Then after the last data byte transmitted, the slave device will actively switch to not addressed slave mode of disconnecting with the master. The further reading by the master will be all FFH. |
| [1:0] | **-** | Reserved |

##### I2CnADDRx – I2Cn Own Slave Address

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

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | | **6** | **5** | | **4** | **3** | **2** | **1** | **0** | |
| **I2CnADDRx[7:1]** | | | | | | | | | **GC** | |
| R/W | | | | | | | | | R/W | |
| **Bit** | **Name** | | | **Description** | | | | | |
| [7:1] | **I2CnADDRx[7:1]** | | | **I2Cn Device’s Own Slave Address**  In master mode:  These bits have no effect.  In slave mode:  These 7 bits define the slave address of this I2C device by user. The master should address I2C device by sending the same address in the first byte data after a START or a repeated START condition. If the AA flag is set, this I2C device will acknowledge the master after receiving its own address and become an addressed slave. Otherwise, the addressing from the master will be ignored.  Note that I2CnADDRx[7:1] should not remain its default value of all 0, because address 0x00 is reserved for General Call. | | | | | |
| [0] | **GC** | | | **General Call Bit**  In master mode:  This bit has no effect.  In slave mode:  0 = The General Call is always ignored.  1 = The General Call is recognized if AA flag is 1; otherwise, it is ignored if AA is 0. | | | | | |

##### CKDIV – Clock Divider

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| CKDIV | C1H, Page 1 | 0000\_00xxb |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:0] | **CKDIV[7:0]** | **Clock Divider**  The system clock frequency FSYS follows the equation below according to CKDIV value.  , while CKDIV = 00H.,  , while CKDIV = 01H to FFH.  This register will be reload by CONFIG3[1:0] during reset  CONDIG3[1:0] = 00b, set CLKDIV = 0x03.  CONDIG3[1:0] = 01b, set CLKDIV = 0x02.  CONDIG3[1:0] = 10b, set CLKDIV = 0x01.  CONDIG3[1:0] = 11b, set CLKDIV = 0x00. |

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

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0C0H | D2H, Page 1 | 0000\_0000 b |
| PWM0C1H | D3H, Page 1 | 0000\_0000 b |
| PWM0C2H | D4H, Page 1 | 0000\_0000 b |
| PWM0C3H | D5H, Page 1 | 0000\_0000 b |
| PWM0C4H | C4H, Page 1 | 0000\_0000 b |
| PWM0C5H | C5H, Page 1 | 0000\_0000 b |

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

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

##### 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 1 Mode / Baud Rate Description for details. |
| [6] | **SMOD0\_1** | **Serial Port 1 Framing Error Access Enable**  0 = S1CON.7 accesses to SM0\_1 bit.  1 = S1CON.7 accesses to FE\_1 bit. |
| [5] | **BRCK** | **Serial Port 0 Baud Rate Clock Source**  This bit selects which Timer is used as the baud rate clock source when serial port 0 is in Mode 1 or 3.  0 = Timer 1.  1 = Timer 3. |
| [4] | **TF3** | **Timer 3 Overflow Flag**  This bit is set when Timer 3 overflows. It is automatically cleared by hardware when the program executes the Timer 3 interrupt service routine. This bit can be set or cleared by software. |
| [3] | **TR3** | **Timer 3 Run Control**  0 = Timer 3 is halted.  1 = Timer 3 starts running.  Note that the reload registers RH3 and RL3 can only be written when Timer 3 is halted (TR3 bit is 0). If any of RH3 or RL3 is written if TR3 is 1, result is unpredictable. |
| [2:0] | **T3PS[2:0]** | **Timer 3 Pre-Scalar**  These bits determine the scale of the clock divider for Timer 3.  000 = 1/1.  001 = 1/2.  010 = 1/4.  011 = 1/8.  100 = 1/16.  101 = 1/32.  110 = 1/64.  111 = 1/128. |

##### RL3 – Timer 3 Reload Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| RL3 | C5H, Page 0 | 0000\_0000 b |

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

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

##### RH3 – Timer 3 Reload High Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| RH3 | C6H, Page 0 | 0000\_0000 b |

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

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

##### PORDIS – POR Disable (TA Protected)

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PORDIS | C6H, Page 1, TA protected | 0000\_0000 b |

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

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

##### TA – Timed Access

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| TA | C7H, All Pages | 0000\_0000 b |

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

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

##### T2CON – Timer 2 Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| T2CON | C8H, All Pages, Bit addressable | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **TF2** | **-** | **-** | **TH2RF** | **TL2RF** | **TR2** | **-** | **CM\_RL2** |
| R/W | - | - | R | R | R/W | - | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **TF2** | **Timer 2 Overflow Flag**  This bit is set when Timer 2 overflows or a compare match occurs. If the Timer 2 interrupt and the global interrupt are enable, setting this bit will make CPU execute Timer 2 interrupt service routine. This bit is not automatically cleared via hardware and should be cleared via software. |
| [6:5] | **-** | Reserved |
| [4] | **TH2RF** | **TH2 Reading Status Flag**  When TH2 is written, software should check this bit first.  0 = TH2 reading is available.  1 = TH2 reading is not available. |
| [3] | **TL2RF** | **TL2 Reading Status Flag**  When TL2 is written, software should check this bit first.  0 = TL2 reading is available.  1 = TL2 reading is not available. |
| [2] | **TR2** | **Timer 2 Run Control**  0 = Timer 2 Disabled. Clearing this bit will halt Timer 2 and the current count will be preserved in TH2 and TL2.  1 = Timer 2 Enabled. |
| [1] | **-** | Reserved |
| [0] | **CM\_RL2** | **Timer 2 Compare or Auto-Reload Mode Select**  This bit selects Timer 2 functioning mode.  0 = Auto-reload mode.  1 = Compare mode. |

##### T2MOD – Timer 2 Mode

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| T2MOD | C9H, Page 0 | 0000\_0000 b |

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

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

##### AUXR1 – Auxiliary Register 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| AUXR1 | C9H, Page 1 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:3] | **-** | Reserved |
| [2] | **UART2PX** | **Serial Port 2 RX (SMC0 DATA) /TX (SMC0 CLK) Pin Exchange**  0 = Assign UART2 RXD (SMC0 DATA) to multiple I/O pin RXD.  UART2 TXD (SMC CLK) to multiple I/O pin TXD  1 = Assign UART2 RXD (SMC0 DATA) to multiple I/O pin TXD.  UART2 TXD (SMC CLK) to multiple I/O pin RXD  Note : that Pin direction is controlled by I/O type of relative pin.  RXD/TXD will exchange immediately once setting or clearing this bit. User should take care of not exchanging pins during transmission or receiving. Or it may cause unpredictable situation and no warning alarms. |
| [1] | **UART1PX** | **Serial Port 1 RX/TX Pin Exchange**  0 = Assign UART1 RXD to multiple I/O pin RXD.  UART1 TXD to multiple I/O pin TXD  1 = Assign UART1 RXD to multiple I/O pin TXD.  UART1 TXD to multiple I/O pin RXD  **Note:** that Pin direction is controlled by I/O type of relative pin.  RXD/TXD will exchange immediately once setting or clearing this bit. User should take care of not exchanging pins during transmission or receiving. Or it may cause unpredictable situation and no warning alarms. |
| [0] | **UART0PX** | **Serial Port 0 RX/TX Pin Exchange**  0 = Assign UART0 RXD to multiple I/O pin RXD.  UART0 TXD to multiple I/O pin TXD  1 = Assign UART0 RXD to multiple I/O pin TXD.  UART0 TXD to multiple I/O pin RXD  **Note:** that Pin direction is controlled by I/O type of relative pin.  RXD/TXD will exchange immediately once setting or clearing this bit. User should take care of not exchanging pins during transmission or receiving. Or it may cause unpredictable situation and no warning alarms. |

##### PRTHCON0 – Port Low Return High Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PRTHCON0 | C9H, Page 3 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P40RTH\_1** | **P40RTH\_0** | **P30RTH\_1** | **P30RTH\_0** | **P21RTH\_1** | **P30RTH\_0** | **P20RTH\_1** | **P20RTH\_0** |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| 7:6 | **P40RTH[1:0]** | **P4.0 Low Level Auto Return To High Delay Timing**  00 = Auto return to high function is disabled.  01 = 1 LIRC clock < auto return delay timing < 2 LIRC clock.  10 = 2 LIRC clock < auto return delay timing < 3 LIRC clock.  11 = 3 LIRC clock < auto return delay timing < 4 LIRC clock. |
| 5:4 | **P30RTH[1:0]** | **P3.0 Low Level To High Delay Timing**  00 = Auto return to high function is disabled.  01 = 1 LIRC clock < auto return delay timing < 2 LIRC clock.  10 = 2 LIRC clock < auto return delay timing < 3 LIRC clock.  11 = 3 LIRC clock < auto return delay timing < 4 LIRC clock. |
| 3:2 | **P21RTH[1:0]** | **P2.1 Low Level To High Delay Timing**  00 = Auto return to high function is disabled.  01 = 1 LIRC clock < auto return delay timing < 2 LIRC clock.  10 = 2 LIRC clock < auto return delay timing < 3 LIRC clock.  11 = 3 LIRC clock < auto return delay timing < 4 LIRC clock. |
| 1:0 | **P20RTH[1:0]** | **P2.0 Low Level To High Delay Timing**  00 = Auto return to high function is disabled.  01 = 1 LIRC clock < auto return delay timing < 2 LIRC clock.  10 = 2 LIRC clock < auto return delay timing < 3 LIRC clock.  11 = 3 LIRC clock < auto return delay timing < 4 LIRC clock. |

##### PIF – Pin Interrupt Flags

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PIF | CAH, Page 0 | 0000\_0000 b |

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

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

##### RCMP2L– Timer 2 Reload/Compare Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| RCMP2L | CAH, Page 1 | 0000\_0000 b |

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

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

##### RCMP2H – Timer 2 Reload/Compare High Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| RCMP2H | CBH, Page 1 | 0000\_0000 b |

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

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

##### TL2 – Timer 2 Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| TL2 | CCH, Page 0 | 0000\_0000 b |

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

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

##### TH2 – Timer 2 High Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| TH2 | CDH, Page 0 | 0000\_0000 b |

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

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

##### PWM0FBS – PWM Brake Source Select

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0FBS | CEH, Page 3 | 0000\_0000 b |

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

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

##### I2CnADDRM – I2Cn Address Mask

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| I2C0ADDRM | CFH, Page 2 | 0000\_0000 b |
| I2C1ADDRM | D7H, Page 2 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **I2CnADDRM[7:0]** | | | | | | | |
| R/W | | | | | | | |

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

##### PSW – Program Status Word

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PSW | D0H, All Pages, Bit addressable | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **CY** | **Carry Flag**  For a adding or subtracting operation, CY will be set when the previous operation resulted in a carry-out from or a borrow-in to the Most Significant bit, otherwise cleared.  If the previous operation is MUL or DIV, CY is always 0.  CY is affected by DA A instruction, which indicates that if the original BCD sum is greater than 100.  For a CJNE branch, CY will be set if the first unsigned integer value is less than the second one. Otherwise, CY will be cleared. |
| [6] | **AC** | **Auxiliary Carry**  Set when the previous operation resulted in a carry-out from or a borrow-in to the 4th bit of the low order nibble, otherwise cleared. |
| [5] | **F0** | **User Flag 0**  The general purpose flag that can be set or cleared by user. |
| [4] | **RS1** | **Register Bank Selection Bits**  These two bits select one of four banks in which R0 to R7 locate.   |  |  |  |  | | --- | --- | --- | --- | | RS1 | RS0 | Register Bank | RAM Address | | 0 | 0 | 0 | 00H to 07H | | 0 | 1 | 1 | 08H to 0FH | | 1 | 0 | 2 | 10H to 17H | | 1 | 1 | 3 | 18H to 1FH | |
| [3] | **RS0** | Check with bit 4 description. |
| [2] | **OV** | **Overflow Flag**  OV is used for a signed character operands. For a ADD or ADDC instruction, OV will be set if there is a carry out of bit 6 but not out of bit 7, or a carry out of bit 7 but not bit 6. Otherwise, OV is cleared. OV indicates a negative number produced as the sum of two positive operands or a positive sum from two negative operands. For a SUBB, OV is set if a borrow is needed into bit6 but not into bit 7, or into bit7 but not bit 6. Otherwise, OV is cleared. OV indicates a negative number produced when a negative value is subtracted from a positive value, or a positive result when a positive number is subtracted from a negative number.  For a MUL, if the product is greater than 255 (00FFH), OV will be set. Otherwise, it is cleared.  For a DIV, it is normally 0. However, if B had originally contained 00H, the values returned in A and B will be undefined. Meanwhile, the OV will be set. |
| [1] | **F1** | **User Flag 1**  The general purpose flag that can be set or cleared by user via software. |
| [0] | **P** | **Parity Flag**  Set to 1 to indicate an odd number of ones in the accumulator. Cleared for an even number of ones. It performs even parity check. |

##### PWM0CON0 – PWM Control Register0

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0CON0 | D1H, Page 0 | 0000\_0000 b |

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

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

##### PWMnCON0 – PWM Control Register0

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

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

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

##### PWMnPH – PWM Period High Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0PH | D1H, Page 1 | 0000\_0000 b |

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

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

##### ACMPCR0 – Analog Comparator Control Register 0

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| ACMPCR0 | D2H, Page 0 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **POSSEL** | **Comparator 0 Positive Input Selection**  00 = ACMP0\_P0 (P2.5) pin.  01 = ACMP0\_P1 (P2.3) pin.  10 = ACMP0\_P2 (P2.1) pin.  11 = ACMP0\_P3 (P3.0) pin. |
| [5:4] | **NEGSEL** | **Comparator 0 Negative Input Selection**  00 = ACMP0\_N0 (P2.4) pin.  01 = Internal comparator reference voltage (CRV).  10 = VBG (Band-gap).  11 = ACMP0\_N1 (P2.0)pin. |
| [3] | **WKEN** | **Comparator 0 Power-Down Wake-Up Enable Bit**  0 = Comparator 0 Wake-up function Disabled.  1 = Comparator 0 Wake-up function Enabled. |
| [2] | **HYSEN** | **Comparator 0 Hysteresis Enable Bit**  0 = Comparator 0 hysteresis Disabled.  1 = Comparator 0 hysteresis Enabled. |
| [1] | **ACMPIE** | **Comparator 0 Interrupt Enable Bit**  0 = Comparator 0 interrupt Disabled.  1 = Comparator 0 interrupt Enabled. If WKEN (ACMPCR1[3]) is set to 1, the wake-up interrupt function will be enabled as well. |
| [0] | **ACMPEN** | **Comparator 0 Enable Bit**  0 = Comparator 0 Disabled.  1 = Comparator 0 Enabled. |

##### ACMPCR1 – Analog Comparator Control Register 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| ACMPCR1 | D3H, Page 0 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **POSSEL** | **Comparator 1 Positive Input Selection**  00 = ACMP1\_P0 (P2.5) pin.  01 = ACMP1\_P1 (P2.3) pin.  10 = ACMP1\_P2 (P2.1) pin.  11 = ACMP1\_P3 (P3.0) pin. |
| [5:4] | **NEGSEL** | **Comparator 1 Negative Input Selection**  00 = ACMP1\_N0 (P2.2) pin.  01 = Internal comparator reference voltage (CRV).  10 = VBG (Band-gap).  11 = ACMP1\_N1 (P3.1)pin. |
| [3] | **WKEN** | **Comparator 1 Power-Down Wake-Up Enable Bit**  0 = Comparator 1 Wake-up function Disabled.  1 = Comparator 1 Wake-up function Enabled. |
| [2] | **HYSEN** | **Comparator 1 Hysteresis Enable Bit**  0 = Comparator 1 hysteresis Disabled.  1 = Comparator 1 hysteresis Enabled. |
| [1] | **ACMPIE** | **Comparator 1 Interrupt Enable Bit**  0 = Comparator 1 interrupt Disabled.  1 = Comparator 1 interrupt Enabled. If WKEN (ACMPCR2[3]) is set to 1, the wake-up interrupt function will be enabled as well. |
| [0] | **ACMPEN** | **Comparator 1 Enable Bit**  0 = Comparator 1 Disabled.  1 = Comparator 1 Enabled. |

##### ACMPSR – Analog Comparator Status Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| ACMPSR | D4H, Page 0 | 0000\_0000 b |

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

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

##### ACMPVREF – ACMP Reference Voltage Control Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| ACMPVREF | D5H, Page 0 | 0000\_0000 b |

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

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

##### SCnCR0 – SC Control Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SC0CR0 | D6H, Page 0 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **NSB** | **Stop Bit Length**  This field indicates the length of stop bit.  0 = The stop bit length is 2 ETU.  1= The stop bit length is 1 ETU.  **Note:** The default stop bit length is 2. SC and UART adopt NSB to program the stop bit length. |
| [6] | **T** | **T Mode**  0 = T0 (ISO7816-3 T = 0 mode).  1 = T1 (ISO7816-3 T = 1 mode).  The T mode controls the BGT (Block Guard Time). Block guard time means the minimum bit length between the leading edges of two consecutive characters between different transfer directions. This field indicates the counter for the bit length of block guard time. According to ISO7816-3, in T = 0 mode, the software must clear T bit to 0 for real block guard time = 16.5. In T = 1 mode, the software must set T bit to 1 for real block guard time = 22.5.  **Note:** In T = 0 mode, the receiver will generate the parity flag on PEF(SC0TSR[4]), if parity error is detected and also drive the parity error signal to transceiver. In T = 1 mode, the receiver will generate the parity flag on PEF(SC0TSR[4]), if parity error detected, but doesn’t drive the parity error signal to transceiver.  **Note:** The description please see section 6.10.5.2 Error Signal and Character Repetition |
| [5] | **RXBGTEN** | **Receiver Block Guard Time Function Enable Bit**  0 = Receiver block guard time function Disabled.  1 = Receiver block guard time function Enabled. |
| [4] | **CONSEL** | **Convention Selection**  0 = Direct convention.  1 = Inverse convention.  **Note 1:** This bit is auto clear to “0”, if AUTOCEN(SCnCR0[3]) is writing “1”  **Note 2:** If AUTOCEN(SCnCR0[3]) is enabled, hardware will decide the convention and change the CONSEL (SCnCR0[4]) bits automatically after SCEN (SCnCR0[0]) =”1”. |
| [3] | **AUTOCEN** | **Auto Convention Enable Bit**  0 = Auto-convention Disabled.  1 = Auto-convention Enabled. When hardware receives TS in answer to reset state and the TS is direct convention, CONSEL(SCnCR0[4]) will be set to 0 automatically, otherwise if the TS is inverse convention, and CONSEL (SCnCR0[4]) will be set to 1.  **Note:** If software enables auto convention function, the setting step must be done before Answer to Reset state and the first data must be 0x3B or 0x3F. After hardware received first data and stored it at buffer, hardware will decided the convention and change the CONSEL (SCnCR0[4]) bits automatically. |
| [2] | **TXOFF** | **TX Transition Disable Bit**  0 = The transceiver Enabled.  1 = The transceiver Disabled. |
| [1] | **RXOFF** | **RX Transition Disable Bit**  0 = The receiver Enabled.  1 = The receiver Disabled.  **Note:** If AUTOCEN (SCnCR0[3])is enabled, these fields must be ignored. |
| [0] | **SCEN** | **SC Engine Enable Bit**  Set this bit to 1 to enable SC operation. If this bit is cleared, SC will force all transition to IDLE state  **Note:** SCEN must be set to 1 before filling in other registers, or smart card will not work properly. |

##### PWM0NP – PWM Negative Polarity

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0NP | D6H, Page 1 | 0000\_0000 b |

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

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

##### SCnCR1 – SC Control Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SC0CR1 | D7H, Page 0 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **OPE** | **Odd Parity Enable Bit**  0 = Even number of logic 1’s are transmitted or check the data word and parity bits in receiving mode.  1 = Odd number of logic 1’s are transmitted or check the data word and parity bits in receiving mode.  **Note:** This bit has effect only when PBOFF bit is ‘0’. |
| [6] | **PBOFF** | **Parity Bit Disable Control**  0 = Parity bit is generated or checked between the “last data word bit” and “stop bit” of the serial data.  1 = Parity bit is not generated (transmitting data) or checked (receiving data) during transfer.  **Note:** In smart card mode, this field must be ‘0’ (default setting is with parity bit) |
| [5:4] | **WLS[1:0]** | **Word Length Selection**  00 = Word length is 8 bits.  01 = Word length is 7 bits.  10 = Word length is 6 bits.  11 = Word length is 5 bits.  **Note:** In smart card mode, this WLS must be ‘00’ |
| [3] | **TXDMAEN** | **SC/UART TX DMA Enable**  This bit enables the SC/UART TX operating by through PDMA transfer, TX data needs to be ready in XRAM before SC/UART TX starting.  0 = SC/UART TX DMA Disabled.  1 = SC/UART 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 = Smart Card mode.  1 = UART mode.  **Note 1:**When operating in UART mode, user must set CONSEL (SCnCR0[4]) = 0 and AUTOCEN(SCnCR0[3]) = 0.  **Note 2:**When operating in Smart Card mode, user must set UARTEN(SCnCR1 [0]) = 0.  **Note 3:**When UART is enabled, hardware will generate a reset to reset FIFO and internal state machine. |

##### PWM0FBD – PWM Fault Brake Data

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0FBD | D7H, Page 1 | 0000\_0000 b |

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

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

##### SCnDR – SC Data Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SC0DR | D9H, Page 0 | 0000\_0000 b |

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

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

##### PWMnPL – PWM Period Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0PL | D9H, Page 1 | 0000\_0000 b |

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

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

##### SCnEGT – SC Extra Guard Time Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SC0EGT | DAH, Page 0 | 0000\_0000 b |

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

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

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

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0C0L | DAH, Page 1 | 0000\_0000 b |
| PWM0C1L | DBH, Page 1 | 0000\_0000 b |
| PWM0C2L | DCH, Page 1 | 0000\_0000 b |
| PWM0C3L | DDH, Page 1 | 0000\_0000 b |
| PWM0C4L | CCH, Page 1 | 0000\_0000 b |
| PWM0C5L | CDH, Page 1 | 0000\_0000 b |

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

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

##### SCnETURD0 – SCn ETU Rate Divider Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SC0ETURD0 | DBH, Page 0 | 0111\_0011 b |

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

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

##### SCnETURD1 –SC ETU Rate Divider Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SC0ETURD1 | DCH, Page 0 | 0011\_0001 b |

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

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

##### SCnIE – SC Interrupt Enable Control Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SC0IE | DDH, Page 0 | 0000\_0000 b |

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

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

##### SCnIS – SC Interrupt Status Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SC0IS | DEH, Page 0 | 0000\_0010 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | **Tx\_Er** | **ACERRIF** | **BGTIF** | **TERRIF** | **TBEIF** | **RDAIF** |
| - | | R/W | R/W | R/W | R | R | R |

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

##### SCnTSR – SC Transfer Status Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SC0TSR | DFH, Page 0 | 0000\_1010 b |

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

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

##### PWMnCON1 – PWM Control 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0CON1 | DFH, Page 1 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **PWMMOD[1:0]** | **PWM Mode Select**  00 = Independent mode.  01 = Complementary mode.  10 = Synchronized mode.  11 = Reserved. |
| [5] | **GP** | **Group Mode Enable**  This bit enables the group mode. If enabled, the duty of first three pairs of PWM are decided by PWM01H and PWM01L rather than their original duty Register Description.  0 = Group mode Disabled.  1 = Group mode Enabled. |
| [4] | **PWMTYP** | **PWM Type Select**  0 = Edge-aligned PWM.  1 = Center-aligned PWM. |
| [3] | **FBINEN** | **FB Pin Input Enable**  0 = PWM0 output Fault Braked by FB pin input Disabled.  1 = PWM0 output Fault Braked by FB pin input Enabled. Once an edge, which matches FBINLS (PWM0FBD.6) selection, occurs on FB pin, PWM0CH0~5 output Fault Brake data in PWMnFBD register. PWMRUN (PWM0CON0.7) will also be automatically cleared by hardware. The PWM output resumes when PWMRUN is set again.  **Note:** This bit is only vaild in PWM0 |
| [2:0] | **PWMDIV[2:0]** | **PWM Clock Divider**  This field decides the pre-scale of PWM clock source.  000 = 1/1.  001 = 1/2.  010 = 1/4.  011 = 1/8.  100 = 1/16.  101 = 1/32.  110 = 1/64.  111 = 1/128. |

##### A or ACC – Accumulator (Bit-addressable)

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| ACC | E0H, All Pages, Bit addressable | 0000\_0000 b |

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

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

##### CAPCON0 – Input Capture Control 0

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| CAPCON0 | E1H, Page 1 | 0000\_0000b |

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

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

##### CAPCON1 – Input Capture Control 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| CAPCON1 | E2H, Page 1 | 0000\_0000b |

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

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

##### CAPCON2 – Input Capture Control 2

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| CAPCON2 | E3H, Page 1 | 0000\_0000b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **-** | Reserved |
| [6] | **ENF2** | **Enable Noise Filer on Input Capture 2**  0 = Noise filter on input capture channel 2 Disabled.  1 = Noise filter on input capture channel 2 Enabled. |
| [5] | **ENF1** | **Enable Noise Filer on Input Capture 1**  0 = Noise filter on input capture channel 1 Disabled.  1 = Noise filter on input capture channel 1 Enabled. |
| [4] | **ENF0** | **Enable Noise Filer on Input Capture 0**  0 = Noise filter on input capture channel 0 Disabled.  1 = Noise filter on input capture channel 0 Enabled. |
| [3:1] | **-** | Reserved |
| [0] | **CMOD** | **Capture Into Interrupt Mode Select Bit**  0 = Enter capture interrupt vector when each trig condition is met.  1 = The first time trig condition met not enter the interrupt vector. Then after the second trigger condition occurs, the capture interrupt is entered every subsequent time. |

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

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

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

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

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

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

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

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

##### DMAnTSR – PDMAn Transfer Status Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| DMA0TSR | E9H, Page 0 | 0000\_0000 b |
| DMA1TSR | F1H, Page 0 | 0000\_0000 b |

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

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

##### PICON – Pin Interrupt Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PICON | E9H, Page 1 | 0000 \_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **PIT7** | **Pin Interrupt Channel 7 Type Select**  This bit selects which type that pin interrupt channel 7 is triggered.  0 = Level triggered.  1 = Edge triggered. |
| [6] | **PIT6** | **Pin Interrupt Channel 6 Type Select**  This bit selects which type that pin interrupt channel 6 is triggered.  0 = Level triggered.  1 = Edge triggered. |
| [5] | **PIT5** | **Pin Interrupt Channel 5 Type Select**  This bit selects which type that pin interrupt channel 5 is triggered.  0 = Level triggered.  1 = Edge triggered. |
| [4] | **PIT4** | **Pin Interrupt Channel 4 Type Select**  This bit selects which type that pin interrupt channel 4 is triggered.  0 = Level triggered.  1 = Edge triggered. |
| [3] | **PIT3** | **Pin Interrupt Channel 3 Type Select**  This bit selects which type that pin interrupt channel 3 is triggered.  0 = Level triggered.  1 = Edge triggered. |
| [2] | **PIT2** | **Pin Interrupt Channel 2 Type Select**  This bit selects which type that pin interrupt channel 2 is triggered.  0 = Level triggered.  1 = Edge triggered. |
| [1] | **PIT1** | **Pin Interrupt Channel 1 Type Select**  This bit selects which type that pin interrupt channel 1 is triggered.  0 = Level triggered.  1 = Edge triggered. |
| [0] | **PIT0** | **Pin Interrupt Channel 0 Type Select**  This bit selects which type that pin interrupt channel 0 is triggered.  0 = Level triggered.  1 = Edge triggered. |

##### MTMnDA – Memory to Memory Destination Address Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| MTM0DA | EAH, Page 0 | 0000\_0000 b |
| MTM1DA | F2H, Page 0 | 0000\_0000 b |

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

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

##### PINEN – Pin Interrupt Negative Polarity Enable.

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PINEN | EAH, Page 1 | 0000\_0000 b |

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

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

##### PIPEN – Pin Interrupt Positive Polarity Enable.

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PIPEN | EBH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **PIPENn** | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

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

##### EIP0 – Extensive Interrupt Priority

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| EIP0 | EFH, Page 0 | 0000\_0000 b |

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

| Bit | Name | Description |
| --- | --- | --- |
| [7] | **PT2** | Timer 2 interrupt priority low bit |
| [6] | **PSPI0** | SPI0 interrupt priority low bit |
| [5] | **PFB** | Fault Brake interrupt priority low bit |
| [4] | **PWDT** | WDT interrupt priority low bit |
| [3] | **PPWM0** | PWM interrupt priority low bit |
| [2] | **PCAP** | Input capture interrupt priority low bit |
| [1] | **PPI** | Pin interrupt priority low bit |
| [0] | **PI2C0** | I2C interrupt priority low bit |
| **Note:** EIP0 is used in combination with the EIPH0 to determine the priority of each interrupt source. See Table 6.2‑5 Interrupt Priority Level Setting for correct interrupt priority configuration. | | |

##### 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. |

##### SPInCR0 – Serial Peripheral Control Register0

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SPI0CR0 | 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 (SPInSR.3) bit to determine the feature of SS pin as shown in Table 6.11‑3 Slave Select Pin Configurations. This bit takes effect only under MSTR = 1 and DISMODF = 1 condition.  0 = SS functions as a general purpose I/O pin.  1 = 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 6.11‑4 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 6.11‑4 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. |
| [1:0] | **SPR[1:0]** | **SPI Clock Rate Select**  These two bits select grades of SPI clock divider. The clock rates below are illustrated under FSYS condition find in Table 6.11‑1 SPI Master Clock Rate Define Table  SPR[3:0] are valid only under Master mode (MSTR = 1). If under Slave mode, the clock will automatically synchronize with the external clock on SPICLK pin from Master device up to FSYS/4 communication speed. |

##### SPInCR1 – Serial Peripheral Control Register1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SPI0CR1 | F3H, Page 1 | 0000\_0000 b |

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

| Bit | Name | Description |
| --- | --- | --- |
| [7:6] | **-** | Reserved. |
| [5:4] | **SPR[3:2]** | **SPI Clock Rate Select**  These two bits select grades of SPI clock divider. The clock rates below are illustrated under FSYS condition find in Table 6.11‑1 SPI Master Clock Rate Define Table  SPR[3:0] are valid only under Master mode (MSTR = 1). If under Slave mode, the clock will automatically synchronize with the external clock on SPICLK pin from Master device up to FSYS/4 communication speed. |
| [3] | **TXDMAEN** | **SPI TX DMA Enable**  This bit enables the SPI TX operating by through PDMA transfer, TX data needs to be ready in XRAM before SPI TX starting.  0 = SPI TX DMA Disabled.  1 = SPI TX DMA Enabled. |
| [2] | **RXDMAEN** | **SPI RX DMA Enable**  This bit enables the SPI RX operating by through PDMA transfer, RX data are saved in XRAM after SPI RX operation.  0 = SPI RX DMA Disabled.  1 = SPI RX DMA Enabled. |
| [1:0] | **SPIS[1:0]** | **SPI Interval Time Selection Between Adjacent Bytes**  SPIS[1:0] and CPHA select eight grades of SPI interval time selection between adjacent bytes. As below table:   |  |  |  |  | | --- | --- | --- | --- | | 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 |   SPIS[1:0] are valid only under Master mode (MSTR = 1). |

##### SPInSR – Serial Peripheral Status Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SPI0SR | F4H, Page 0 | 0000\_0000 b |

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

| Bit | Name | Description |
| --- | --- | --- |
| [7] | **SPIF** | **SPI Complete Flag**  This bit is set to logic 1 via hardware while an SPI data transfer is complete or an receiving data has been moved into the SPI read buffer. If ESPI and EA are enabled, an SPI interrupt will be required. This bit should be cleared via software. Attempting to write to SPInDR is inhibited if SPIF is set. |
| [6] | **WCOL** | **Write Collision Error Flag**  This bit indicates a write collision event. Once a write collision event occurs, this bit will be set. It should be cleared via software. |
| [5] | **SPIOVF** | **SPI Overrun Error Flag**  This bit indicates an overrun event. Once an overrun event occurs, this bit will be set. If ESPI and EA are enabled, an SPI interrupt will be required. This bit should be cleared via software. |
| [4] | **MODF** | **Mode Fault Error Flag**  This bit indicates a Mode Fault error event. If SS pin is configured as Mode Fault input (MSTR = 1 and DISMODF = 0) and SS is pulled low by external devices, a Mode Fault error occurs. Instantly MODF will be set as logic 1. If ESPI and EA are enabled, an SPI interrupt will be required. This bit should be cleared via software. |
| [3] | **DISMODF** | **Disable Mode Fault Error Detection**  This bit is used in combination with the SSOE (SPInCR.7) bit to determine the feature of SS pin as shown in Table 6.11‑3 Slave Select Pin Configurations. DISMODF is valid only in Master mode (MSTR = 1).  0 = Mode Fault detection Enabled. SS serves as input pin for Mode Fault detection disregard of SSOE.  1 = Mode Fault detection Disabled. The feature of SS follows SSOE bit. |
| [2] | **DISSPIF** | **Disable SPI Complete Interrupt**  This bit is used to disable SPI complete interrupt while an SPI data transfer is complete or an receiving data has been moved into the SPI read buffer. Especially in SPI DMA operation.  0 = SPI Complete Interrupt Enabled while ESPI and EA are enabled,.  1 = SPI Complete Interrupt Disabled. |
| [1] | **TXBFF** | **SPI TX Buffer Full Flag**  0 = SPI TX buffer is empty.  1 = SPI TX buffer is full. |

##### SPInDR – Serial Peripheral Data Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SPI0DR | F5H, Page 0 | 0000\_0000 b |

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

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

##### DMAnBAH – PDMAn XRAM Base Address High Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| DMA0BAH | F6H, Page 0 | 0000\_0000 b |
| DMA1BAH | FDH, Page 0 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **MDAH[3:0]** | | | | **MAH[3:0]** | | | |
| R/W | | | | R/W | | | |

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

##### EIPH0 – Extensive Interrupt Priority High

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| EIPH0 | F7H, Page 0 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **PT2H** | Timer 2 interrupt priority high bit |
| [6] | **PSPI0H** | SPI0 interrupt priority high bit |
| [5] | **PFBH** | Fault Brake interrupt priority high bit |
| [4] | **PWDTH** | WDT interrupt priority high bit |
| [3] | **PPWM0H** | PWM0 interrupt priority high bit |
| [2] | **PCAPH** | Input capture interrupt priority high bit |
| [1] | **PPIH** | Pin interrupt priority high bit |
| [0] | **PI2C0H** | I2C interrupt priority high bit |
| **Note：**EIPH0 is used in combination with the EIP0 to determine the priority of each interrupt source. See Table 6.2‑5 Interrupt Priority Level Setting for correct interrupt priority configuration. | | |

##### S1CON – Serial Port 1 Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| S1CON | F8H, All Pages, Bit addressable | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **SM0\_1/FE\_1** | **SM1\_1** | **SM2\_1** | **REN\_1** | **TB8\_1** | **RB8\_1** | **TI\_1** | **RI\_1** |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Name | Description |
| --- | --- | --- |
| [7] | **SM0\_1/FE\_1** | **Serial Port 1 Mode Select**  SMOD0\_1 (T3CON.6) = 0:.  See Table 6.9‑2 Serial Port 1 Mode / Baud Rate Description  for details.  SMOD0\_1 (T3CON.6) = 1:.  SM0\_1/FE\_1 bit is used as frame error (FE) status flag. It is cleared by software.  0 = Frame error (FE) did not occur.  1 = Frame error (FE) occurred and detected. |
| [6] | **SM1\_1** | Check with bit 7 description. |
| [5] | **SM2\_1** | **Multiprocessor Communication Mode Enable**  The function of this bit is dependent on the serial port 1 mode.  Mode 0:  No effect.  Mode 1:  This bit checks valid stop bit.  0 = Reception is always valid no matter the logic level of stop bit.  1 = Reception is valid only when the received stop bit is logic 1 and the received data matches “Given” or “Broadcast” address.  Mode 2 or 3:  For multiprocessor communication.  0 = Reception is always valid no matter the logic level of the 9th bit.  1 = Reception is valid only when the received 9th bit is logic 1 and the received data matches “Given” or “Broadcast” address. |
| [4] | **REN\_1** | **Receiving Enable**  0 = Serial port 1 reception Disabled.  1 = Serial port 1 reception Enabled in Mode 1,2, or 3. In Mode 0, reception is initiated by the condition REN\_1 = 1 and RI\_1 = 0. |
| [3] | **TB8\_1** | **9th Transmitted Bit**  This bit defines the state of the 9th transmission bit in serial port 1 Mode 2 or 3. It is not used in Mode 0 or 1. |
| [2] | **RB8\_1** | **9th Received Bit**  The bit identifies the logic level of the 9th received bit in serial port 1 Mode 2 or 3. In Mode 1, RB8\_1 is the logic level of the received stop bit. SM2\_1 bit as logic 1 has restriction for exception. RB8\_1 is not used in Mode 0. |
| [1] | **TI\_1** | **Transmission Interrupt Flag**  This flag is set by hardware when a data frame has been transmitted by the serial port 1 after the 8th bit in Mode 0 or the last data bit in other modes. When the serial port 1 interrupt is enabled, setting this bit causes the CPU to execute the serial port 1 interrupt service routine. This bit must be cleared manually via software. |
| [0] | **RI\_1** | **Receiving Interrupt Flag**  This flag is set via hardware when a data frame has been received by the serial port 1 after the 8th bit in Mode 0 or after sampling the stop bit in Mode 1, 2, or 3. SM2\_1 bit as logic 1 has restriction for exception. When the serial port 1 interrupt is enabled, setting this bit causes the CPU to execute to the serial port 1 interrupt service routine. This bit must be cleared manually via software. |

##### PWM0DTEN – PWM Dead-time Enable (TA Protected)

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0DTEN | F9H, Page 1, TA protected | 0000\_0000 b |

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

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

##### PWM0DTCNT – PWM Dead-time Counter (TA Protected)

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0DTCNT | FAH, Page 1, TA protected | 0000\_0000 b |

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

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

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

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0MEN | FBH, Page 1 | 0000\_0000 b |

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

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

##### PWMnMD – PWM Mask Data

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0MD | FCH, Page 1 | 0000\_0000 b |

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

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

##### 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** |
| **-** | | **PDMA0** | **PSMC** | **PHF** | **PWKT** | **PT3** | **PS1** |
| - | | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **-** | Reserved |
| [5] | **PDMA0** | PDMA0 interrupt priority low bit |
| [4] | **PSMC** | SMC interrupt priority low bit |
| [3] | **PHF** | Hard fault interrupt priority low bit |
| [2] | **PWKT** | WKT interrupt priority low bit |
| [1] | **PT3** | Timer 3 interrupt priority low bit |
| [0] | **PS1** | Serial port 1 interrupt priority low bit |
| **Note:** EIP1 is used in combination with the EIPH1 to determine the priority of each interrupt source. See Table 6.2‑5 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** |
| **-** | | **PDMA0H** | **PSMCH** | **PHFH** | **PWKTH** | **PT3H** | **PS1H** |
| - | | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **-** | Reserved |
| [5] | **PDMA0H** | PDMA0 interrupt priority high bit |
| [4] | **PSMCH** | SMC interrupt priority high bit |
| [3] | **PHFH** | Hard fault interrupt priority high bit |
| [2] | **PWKTH** | WKT interrupt priority high bit |
| [1] | **PT3H** | Timer 3 interrupt priority high bit |
| [0] | **PS1H** | Serial port 1 interrupt priority high bit |
| **Note:** EIPH1 is used in combination with the EIP1 to determine the priority of each interrupt source. See Table 6.2‑5 Interrupt Priority Level Setting for correct interrupt priority configuration. | | |

##### LVRDIS – LVR Disable (TA Protected)

|  |  |  |
| --- | --- | --- |
| **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. |

#### SFR Page Selection

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

##### SFRS – SFR Page Selection

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SFRS | 91H, All Pages | 0000\_0000b |

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

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

Switch SFR page demo code:

MOV SFRS,#00H ;switch to SFR Page 0

MOV SFRS,#01H ;switch to SFR Page 1

MOV SFRS,#02H ;switch to SFR page 2

MOV SFRS,#03H ;switch to SFR page 3

#### Timed Access Protection (TA)

The MUG51TBAE has several features such as WDT and Brown-out detection that are crucial to proper operation of the system. If leaving these Register Description unprotected, errant code may write undetermined value into them and results in incorrect operation and loss of control. To prevent this risk, the MUG51TBAE has a protection scheme, which limits the write access to critical SFR. This protection scheme is implemented using a timed access (TA). The following registers are related to the TA process.

##### TA – Timed Access

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| TA | C7H, All Pages | 0000\_0000 b |

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

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

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

(CLR EA) ;if any interrupt is enabled, disable temporally

MOV TA,#0AAH

MOV TA,#55H

(Instruction that writes a TA protected register)

(SETB EA) ;resume interrupts enabled

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

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

Example 1,

MOV TA,#0AAH ;3 clock cycles

MOV TA,#55H ;3 clock cycles

ORL WDCON,#data ;4 clock cycles

Example 2,

MOV TA,#0AAH ;3 clock cycles

MOV TA,#55H ;3 clock cycles

NOP ;1 clock cycle

ANL BODCON0,#data ;4 clock cycles

Example 3,

MOV TA,#0AAH ;3 clock cycles

MOV TA,#55H ;3 clock cycles

MOV WDCON,#data1 ;3 clock cycles

ORL BODCON0,#data2 ;4 clock cycles

Example 4,

MOV TA,#0AAH ;3 clock cycles

NOP ;1 clock cycle

MOV TA,#55H ;3 clock cycles

ANL BODCON0,#data ;4 clock cycles

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

#### Dual DPTRs

The original 8051 contains one DPTR (data pointer) only. With single DPTR, it is difficult to move data form one address to another with wasting code size and low performance. The MUG51TBAE provides two data pointers. Thus, software can load both a source and a destination address when doing a block move. Once loading, the software simply switches between DPTR and DPTR1 by the active data pointer selection DPS (AUXR0.0) bit.

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

MOV R0,#64 ;number of bytes to move

MOV DPTR,#D\_Addr ;load destination address

INC AUXR0 ;change active DPTR

MOV DPTR,#S\_Addr ;load source address

LOOP:

MOVX A,@DPTR ;read source data byte

INC AUXR0 ;change DPTR to destination

MOVX @DPTR,A ;write data to destination

INC DPTR ;next destination address

INC AUXR0 ;change DPTR to source

INC DPTR ;next source address

DJNZ R0,LOOP

INC AUXR0 ;(optional) restore DPS

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

##### DPL – Data Pointer Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| DPL | 82H, All Pages | 0000\_0000b |

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

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

##### DPH – Data Pointer High Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| DPH | 83H, All Pages | 0000\_0000b |

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

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

##### AUXR0 – Auxiliary Register 0

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

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

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

##### PSW – Program Status Word

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PSW | D0H, All Pages, Bit addressable | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **CY** | **Carry Flag**  For a adding or subtracting operation, CY will be set when the previous operation resulted in a carry-out from or a borrow-in to the Most Significant bit, otherwise cleared.  If the previous operation is MUL or DIV, CY is always 0.  CY is affected by DA A instruction, which indicates that if the original BCD sum is greater than 100.  For a CJNE branch, CY will be set if the first unsigned integer value is less than the second one. Otherwise, CY will be cleared. |
| [6] | **AC** | **Auxiliary Carry**  Set when the previous operation resulted in a carry-out from or a borrow-in to the 4th bit of the low order nibble, otherwise cleared. |
| [5] | **F0** | **User Flag 0**  The general purpose flag that can be set or cleared by user. |
| [4] | **RS1** | **Register Bank Selection Bits**  These two bits select one of four banks in which R0 to R7 locate.   |  |  |  |  | | --- | --- | --- | --- | | RS1 | RS0 | Register Bank | RAM Address | | 0 | 0 | 0 | 00H to 07H | | 0 | 1 | 1 | 08H to 0FH | | 1 | 0 | 2 | 10H to 17H | | 1 | 1 | 3 | 18H to 1FH | |
| [3] | **RS0** | Check with bit 4 description. |
| [2] | **OV** | **Overflow Flag**  OV is used for a signed character operands. For a ADD or ADDC instruction, OV will be set if there is a carry out of bit 6 but not out of bit 7, or a carry out of bit 7 but not bit 6. Otherwise, OV is cleared. OV indicates a negative number produced as the sum of two positive operands or a positive sum from two negative operands. For a SUBB, OV is set if a borrow is needed into bit6 but not into bit 7, or into bit7 but not bit 6. Otherwise, OV is cleared. OV indicates a negative number produced when a negative value is subtracted from a positive value, or a positive result when a positive number is subtracted from a negative number.  For a MUL, if the product is greater than 255 (00FFH), OV will be set. Otherwise, it is cleared.  For a DIV, it is normally 0. However, if B had originally contained 00H, the values returned in A and B will be undefined. Meanwhile, the OV will be set. |
| [1] | **F1** | **User Flag 1**  The general purpose flag that can be set or cleared by user via software. |
| [0] | **P** | **Parity Flag**  Set to 1 to indicate an odd number of ones in the accumulator. Cleared for an even number of ones. It performs even parity check. |

## System Manager

### Clock System

The MUG51TBAE 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 MUG51TBAE is embedded with two internal oscillators: one 38.4 kHz low-speed and one 7.3728 MHz median speed, which is factory trimmed to ±3% under 0C~70C conditions. A clock divider CKDIV is also available on MUG51TBAE for adjustment of the flexibility between power consumption and operating performance.

|  |
| --- |
|  |

Figure 6.2‑1 Clock System Block Diagram

##### A or ACC – Accumulator (Bit-addressable)

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| ACC | E0H, All Pages, Bit addressable | 0000\_0000 b |

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

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

##### 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 Flag**  The hardware will set this bit after IAPGO (IAPTRG.0) is set if any of the following condition is met:  (1) The accessing address is oversize.  (2) IAPCN command is invalid.  (3) IAP erases or programs updating un-enabled block.  (4) IAP erasing or programming operates under VBOD while BOIAP (CONFIG2.5) remains un-programmed 1 with BODEN (BODCON0.7) as 1 and BORST (BODCON0.2) as 0.  This bit should be cleared via software. |
| [5:2] | **-** | Reserved |
| [1] | **BS** | **Boot Select**  This bit defines from which block that MCU re-boots after all resets.  0 = MCU will re-boot from APROM after all resets.  1 = MCU will re-boot from LDROM after all resets. |
| [0] | **IAPEN** | **IAP Enable**  0 = IAP function Disabled.  1 = IAP function Enabled.  Once enabling IAP function, the MIRC 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. |

##### CKCON – Clock Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| CKCON | 8EH, Page 0 | 1000\_0010b |

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

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

##### 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 0 Mode / Baud Rate Description for details. |
| [6] | **SMOD0** | **Serial Port 0 Framing Error Flag Access Enable**  0 = SCON.7 accesses to SM0 bit.  1 = SCON.7 accesses to FE bit. |
| [5] | **-** | Reserved. |
| [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. |

##### AUXR0 – Auxiliary Register 0

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

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

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

##### PORDIS – POR Disable (TA Protected)

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PORDIS | C6H, Page 1, TA protected | 0000\_0000 b |

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

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

##### LVRDIS – LVR Disable (TA Protected)

|  |  |  |
| --- | --- | --- |
| **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. |

#### System Clock Sources

There are a total oftwo system clock sources selectable in the MUG51TBAE including Median-speed internal oscillator, and low-speed internal oscillator. Each of them can be the system clock source in the MUG51TBAE.

#### Internal Oscillators

There are two internal oscillators in the MUG51TBAE – one Median-speed internal oscillator (MIRC) and one 38.4 kHz low-speed (LIRC). Both of them can be selected as the system clock. MIRC can be enabled by setting MIRCEN (CKEN. 1). LIRC is enabled after device is powered up. User can set OSC[2:0] (CKSWT [2:0]) as [0,0, 0] to select the MIRC as the system clock. By setting OSC[2:0] as [1,0,x], LIRC will be selected as the system clock. Note that after the MUG51TBAE is powered, MIRC and LIRC will be both enabled and MIRC is default selected as the system clock source.

##### MIRCTRIM0 – Median Speed Internal Oscillator Trim 0

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| MIRCTRIM | 84H, Page 3, TA protected | XXXX\_XXXXb |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **MIRCTRIM[9:2]** | | | | | | | |
| R/W | | | | | | | |

|  |  |  |
| --- | --- | --- |
| **Bit** | **Name** | **Description** |
| [9:3] | **MIRCTRIM[9:2]** | **Median Speed Internal Oscillator Trim Value High Byte** |

##### MIRCTRIM1 – Median Speed Internal Oscillator Trim 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| MIRCTRIM1 | 85H, Page 3, TA protected | XXXX\_XXXXb |

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

|  |  |  |
| --- | --- | --- |
| **Bit** | **Name** | **Description** |
| [1:0] | **MIRCTRIM[1:0]** | **Median Speed Internal Oscillator Trim Value Low Byte** |

##### LIRCTRIM – Low Speed Internal Oscillator Trim (TA Protected)

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| LIRCTRIM | 84H, Page 1 | XXXX\_XXXXb |

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

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

#### System Clock Switching

The MUG51TBAE supports clock source switching on-the-fly by controlling CKSWT and CKEN registers via software. It provides a wide flexibility in application. Note that these SFR are writing TA protected for precaution. With this clock source control, the clock source can be switched between the external clock source and the internal oscillator, even between the high and low-speed internal oscillator. However, during clock source switching, the device requires some amount of warm-up period for an original disabled clock source. Therefore, use should follow steps below to ensure a complete clock source switching. User can enable the target clock source by writing proper value into CKEN register, wait for the clock source stable by polling its status bit in CKSWT register, and switch to the target clock source by changing OSC[2:0] (CKSWT[2:0]). After these step, the clock source switching is successful and then user can also disable the original clock source if power consumption is concerned. Note that if not following the steps above, the hardware will take certain actions to deal with such illegal operations as follows.

* 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.
* If user tries to switch the system clock source to a disabled one by changing OSC[2:0] value, OSC[2:0] value will be updated right away. But the system clock will remain the original one and CKSWTF flag will be set by hardware.
* 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 | 0001 \_1000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | **LIRCST** | **MIRCST** | **OSC[2:0]** | | |
| - | | | R | R | W | | |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:5] | **-** | Reserved |
| [4] | **LIRCST** | **Low Speed Internal Oscillator 38.4 kHz Status**  0 = Low speed internal oscillator is not stable or is disabled.  1 = Low speed internal oscillator is enabled and stable. |
| [3] | **MIRCST** | **Median Speed Internal Oscillator Status**  0 = Internal median speed oscillator is not stable or disabled.  1 = Internal median speed internal oscillator is enabled and stable. |
| [2:0] | **OSC[2:0]** | **Oscillator Selection Bits**  This field selects the system clock source.  000 = Internal MIRC oscillator. Defaul value accoding to MIRCEN(CKEN.1) enabled  10x = Internal 38.4 kHz oscillator according to LIRCEN(CKEN.4) enabled.  Others = Reseved.  Note that this field is write only. The read back value of this field may not correspond to the present system clock source. |

##### CKEN – Clock Enable (TA Protected)

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| CKEN | 97H, Page 0, TA protected | 0001\_0110 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | **LIRCEN** | **-** | | **MIRCEN** | **CKSWTF** |
| - | | | R/W | - | | R/W | R |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:5] | **-** | Reserved. |
| [4] | **LIRCEN** | **Low Speed Internal Oscillator 38.4 kHz Enable**  0 = The low speed internal oscillator Disabled.  1 = The low speed internal oscillator Enabled.  Note that when (1)WDT is enabled, (2)WKT is running by the clock source of the internal 38.4 kHz oscillator ,(3) BOD is enabled, or (4)LVR filter is enabled, a write 0 to LIRCEN will be ignored. LIRCEN is always 1 and the internal 38.4 kHz oscillator is always enabled. |
| [3:2] | **-** | Reserved. |
| [1] | **MIRCEN** | **Median Speed Internal Oscillator Enable**  0 = The medianspeed internal oscillator Disabled.  1 = The median speed internal oscillator Enabled.  Note that once IAP is enabled by setting IAPEN (CHPCON.0), the median speed internal oscillator will be enabled automatically. The hardware will also set MIRCEN and MIRCST bits. After IAPEN is cleared, MIRCEN and MIRCST resume the original values. |
| [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. |

#### System Clock Divider

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

##### CKDIV – Clock Divider

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| CKDIV | C1H, Page 1 | 0000\_00xxb |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:0] | **CKDIV[7:0]** | **Clock Divider**  The system clock frequency FSYS follows the equation below according to CKDIV value.  FSYS = FOSC, while CKDIV = 00H.  , while CKDIV = 01H to FFH.  CKDIV register will be reload and setting by CONFIG3[1:0] after reset |

#### System Clock Output

The MUG51TBAE provides a CLO pin that outputs the system clock. Its frequency is the same as FSYS. 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

|  |  |  |
| --- | --- | --- |
| **Part of Register Content** | **SFR Address** | **Reset Value** |
| CKCON | 8EH, Page 0 | 1000\_0010b |

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

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

### Power Management

The MUG51TBAE has several features that help user to control the power consumption of the device.

lists all power mode at MUG51TBAE to save the power consumption. For a stable current consumption, the state and mode of each pin should be taken care of. The minimum power consumption can be attained by giving the pin state just the same as the external pulls for example output 1 if pull-high is used or output 0 if pull-low. If the I/O pin is floating, user is recommended to leave it as quasi-bidirectional mode.

|  |  |  |
| --- | --- | --- |
| **Mode** | **Clock Source** | **Comment** |
| Normal mode | MIRC / LIRC | Only CPU clock is stoped. |
| Idle mode | MIRC / LIRC | - |
| Power-down mode (PD) | Any clock source | Most clocks are disabled except ACMP/LIRC, and only WDT/WKT peripheral clocks still enable if their clock sources are selected as LIRC.  Level 0 (PD0): all IP off / flash standby / MIRC off / LDO on (Default)  Level 1 (PD1): all IP off / flash standby / MIRC off / LDO off  Level 2 (PD2): all IP off / flash off / MIRC off / LDO off |

Table 6.2‑1 Power Mode and Clock Source

For each power mode, they have different entry setting and leaving condition. The Table 6.2‑2 shows the entry setting for each power mode. When chip power-on, chip is running in normal mode. User can enter each mode by setting IDL (PCON.0), and PD (PCON.1).

| **Register/Instruction Mode** | **PD (PCON.1)** | **IDL (PCON.0)** | **PDLS(PDL[1:0])** |
| --- | --- | --- | --- |
| Normal mode | 0 | 0 | - |
| Idle mode | 0 | 1 | - |
| Power-down mode (PD) Level 0 | 1 | X | 00 |
| Power-down mode (PD) Level 1 | 1 | X | 01 |
| Power-down mode (PD) Level 2 | 1 | X | 10 |

Table 6.2‑2 Entry Setting of Power-down Mode

#### 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.

##### 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** |
| --- | --- | --- |
| [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. |

#### Power-down Mode

Power-down mode is the lowest power state that the MUG51TBAE can enter. It remains the power consumption as A ”uA” level by stopping the system clock source. Both of CPU and peripheral functions like Timers or UART are frozen. Flash memory is put into its stop mode. All activity is completely stopped and the power consumption is reduced to the lowest possible value. The device can be put into Power-down mode by writing 1 to bit PD (PCON.1). The instruction that does this action will be the last instruction to be executed before the device enters Power-down mode. In the Power-down mode, RAM maintains its content. The port pins output the values held by their own state before Power-down respectively.

There are several ways to exit the MUG51TBAE 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 MUG51TBAE up from the Power-down mode is by an enabled external interrupt. The trigger on the external pin will asynchronously restart the system clock. After oscillator is stable, the device executes the interrupt service routine (ISR) for the corresponding external interrupt. After the ISR is completed, the program execution returns to the instruction after the one, which puts the device into Power-down mode and continues. Interrupts that allows to wake up CPU from Power-down mode includes external interrupt INT0 and INT1, pin interrupt, WDT interrupt, WKT interrupt, and brown-out interrupt.

##### PDL – Power Down Level Select

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PDL | 8FH, Page 3 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | **-** | **-** | **-** | **-** | **-** | **PDLS\_1** | **PDLS\_0** |
| - | - | - | - | - | - | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [1:0] | **PDLS[1:0]** | **Power Down Level Select**  00 = Power Down Level 0 (defaut).  01 = Power Down Level 1.  10 = Power Down Level 2.  Others = Reserved. |

##### 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** |
| --- | --- | --- |
| [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. |

### Reset

The MUG51TBAE 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.

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

The MUG51TBAE incorporates an internal power-on reset (POR) and a low voltage reset (LVR). During a power-on process of rising power supply voltage VDD, the POR or LVR will hold the MCU in reset mode when VDD is lower than the voltage reference thresholds. This design makes CPU not access program Flash while the VDD 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, VDD 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

|  |  |  |
| --- | --- | --- |
| **Part of Register Content** | **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 |

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

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

#### Brown-Out Reset (BOR)

The other power monitoring function brown-out detection (BOD) circuit is used for monitoring the VDD level during execution. There are eight CONFIG selectable brown-out trigger levels available for wide voltage applications. These eight nominal levels are 1.8V, 2.0V, 2.4V, 2.7V, 3.0V, 3.7V and 4.4V selected via setting CBOV[2:0] (CONFIG2[6:4]). BOD level can also be changed by setting BOV[2:0] (BODCON0[6:4]) after power-on. When VDD drops to the selected brown-out trigger level (VBOD), 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. VBOD 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 VDD drops below VBOD while BORST (BODCON0.2) is 0. In this case, BOF (BODCON0.3) will be set as 1. After user cleared this flag whereas VDD remains below VBOD, BOF will not set again. BOF just acknowledge user a power drop occurs. The BOF will also be set as 1 after VDD goes higher than VBOD to indicate a power resuming. The BOD circuit provides an useful status indicator BOS (BODCON0.0), which is helpful to tell a brown-out event or power resuming event occurrence. If the BORST bit is set as 1, this will enable brown-out reset function. After a brown-out reset, BORF (BODCON0.1) will be set as 1 via hardware. It will not be altered by reset other than power-on. This bit can be cleared by software. Note that all bits in BODCON0 is writing protected by timed access (TA).

|  |  |  |  |
| --- | --- | --- | --- |
| CBODEN (CONFIG2.7) | CBORST (CONFIG2.2) | VDD Level | BOF |
| 1 | 1 | > VBOD always | 0 |
| 1 | 0 | < VBOD | 1 |
| 1 | 0 | > VBOD | 0 |
| 0 | X | X | 0 |

Table 6.2‑3 BOF Reset Value

The MUG51TBAE 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 MUG51TBAE has a BOD filter which filters the power noise to avoid BOD event triggering unconsciously. The BOD filter is enabled by default and can be disabled by setting BODFLT (BODCON1.0) as 0 if user requires a rapid BOD response. The minimum brown-out detect pulse width is listed in LPBOD[1:0] (BODCON1)

**Note:** Since each level of BOD voltage is selectable no matter MUG51TBAE VDD range with 5.5V or 3.6V. The BOD enabled voltage value should be select base on the VDD.

|  |
| --- |
|  |

Figure 6.2‑2 Brown-out Detection Block Diagram

##### CONFIG2

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

Factory default value: 1111 1111b

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

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

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

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **BODEN** | **Brown-Out Detection Enable**  0 = Brown-out detection circuit ON.  1 = Brown-out detection circuit OFF.  Note that BOD output is not available until 2~3 LIRC clocks after enabling. |
| [6:4] | **BOV[2:0]** | **CONFIG Brown-Out Voltage Select**  111 = VBOD is 1.8V.  110 = VBOD is 1.8V.  101 = VBOD is 2.0V.  100 = VBOD is 2.4V.  011 = VBOD is 2.7V.  010 = VBOD is 3.0V.  001 = VBOD is 3.7V.  000 = VBOD is 4.4V. |
| [3] | **BOF** | **Brown-Out Interrupt Flag**  This flag will be set as logic 1 via hardware after a VDD dropping below or rising above VBOD event occurs. If both EBOD (I.E.5) and EA (I.E.7) are set, a brown-out interrupt requirement will be generated. This bit should be cleared via software. |
| [2] | **BORST** | **Brown-Out Reset Enable**  This bit decides whether a brown-out reset is caused by a power drop below VBOD.  0 = Brown-out reset when VDD drops below VBOD Disabled.  1 = Brown-out reset when VDD drops below VBOD Enabled. |
| [1] | **BORF** | **Brown-Out Reset Flag**  When the MCU is reset by brown-out event, this bit will be set via hardware. This flag is recommended to be cleared via software. |
| [0] | **BOS** | **Brown-Out Status**  This bit indicates the VDD voltage level comparing with VBOD while BOD circuit is enabled. It keeps 0 if BOD is not enabled.  0 = VDD voltage level is higher than VBOD or BOD is disabled.  1 = VDD voltage level is lower than VBOD.  Note that this bit is read-only. |
| **Note:**  1.BODEN, BOV[2:0], and BORST are initialized by being directly loaded from CONFIG2 bit 7, [6:4], and 2 after all resets.  2. BOF reset value depends on different setting of CONFIG2 and VDD voltage level. | | |

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

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

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

| **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 MIRC 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.) |

#### 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.

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

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

##### AUXR0 – Auxiliary Register 0

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

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

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

#### Watchdog Timer Reset

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

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

##### WDCON – Watchdog Timer Control (TA Protected)

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

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

| **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. |

#### Software Reset

The MUG51TBAE provides a software reset, which allows the software to reset the whole system just similar to an external reset, initializing the MCU as it reset state. The software reset is quite useful in the end of an ISP progress. For example, if an ISP of Boot Code updating User Code finishes, a software reset can be asserted to re-boot CPU to execute new User Code immediately. Writing 1 to SWRST (CHPCON.7) will trigger a software reset. Note that this bit is writing TA protection. The instruction that sets the SWRST bit is the last instruction that will be executed before the device reset. See demo code below.

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

##### CHPCON – Chip Control

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

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

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

##### AUXR0 – Auxiliary Register 0

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

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

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

The software demo code is listed below.

ANL AUXR0,#01111111b ;software reset flag clear

CLR EA

MOV TA,#0Aah

MOV TA,#55h

ORL CHPCON,#10000000b ;software reset

#### Boot Select

|  |
| --- |
|  |

Figure 6.2‑3 Boot Selecting Diagram

The MUG51TBAE provides user a flexible boot selection for variant application. The SFR bit BS in CHPCON.1 determines MCU booting from APROM or LDROM after any source of reset. If reset occurs and BS is 0, MCU will reboot from address 0000H of APROM. Else, the CPU will reboot from address 0000H of LDROM. Note that BS is loaded from the inverted value of CBS bit in CONFIG0.7 after all resets except software reset.

##### CONFIG0

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

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

##### CHPCON – Chip Control (TA Protected)

|  |  |  |
| --- | --- | --- |
| **Part of Register Content** | **SFR Address** | **Reset Value** |
| CHPCON | 9FH, Page 0, TA protected | Software 0000\_00U0 b  Others 0000\_00C0 b |

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

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

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

#### 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.

### Interrupt System

#### Interrupt Overview

The purpose of the interrupt is to make the software deal with unscheduled or asynchronous events. The MUG51TBAE has a four-priority-level interrupt structure with 31 interrupt sources. Each of the interrupt sources has an individual priority setting bits, interrupt vector and enable bit. In addition, the interrupts can be globally enabled or disabled. When an interrupt occurs, the CPU is expected to service the interrupt. This service is specified as an Interrupt Service Routine (ISR). The ISR resides at a predetermined address as shown in Table 6.2‑4 Interrupt Vectors. When the interrupt occurs if enabled, the CPU will vector to the respective location depending on interrupt source, execute the code at this location, stay in an interrupt service state until the ISR is done. Once an ISR has begun, it can be interrupted only by a higher priority interrupt. The ISR should be terminated by a return from interrupt instruction RETI. This instruction will force the CPU return to the instruction that would have been next when the interrupt occurred.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Source** | **Vector Addess** | **Vector Number** | **Source** | **Vector Address** | **Vector Number** |
| Reset | 0000H | - | Serial port 1 interrupt | 007BH | 15 |
| External interrupt 0 | 0003H | 0 | Timer 3 overflow | 0083H | 16 |
| Timer 0 overflow | 000BH | 1 | Self Wake-up Timer interrupt | 008BH | 17 |
| External interrupt 1 | 0013H | 2 | CPU Hard Fault Interrupt | 0093H | 18 |
| Timer 1 overflow | 001BH | 3 | SMC0 Interrupt | 009BH | 19 |
| Serial port 0 interrupt | 0023H | 4 | PDMA0 Interrupt | 00A3H | 20 |
| Timer 2 event | 002BH | 5 | PDMA1 Interrupt | 00ABH | 21 |
| I2C0 status/timer-out interrupt | 0033H | 6 |  |  |  |
| Pin interrupt | 003BH | 7 | ACMP Interrupt | 00BBH | 23 |
| Brown-out detection interrupt | 0043H | 8 | I2C1 status/timer-out interrupt | 00C3H | 24 |
| SPI0 interrupt | 004BH | 9 |  |  |  |
| WDT interrupt | 0053H | 10 |  |  |  |
| ADC interrupt | 005BH | 11 |  |  |  |
| Input capture interrupt | 0063H | 12 |  |  |  |
| PWM0 interrupt | 006BH | 13 |  |  |  |
| Fault Brake0 interrupt | 0073H | 14 |  |  |  |

Table 6.2‑4 Interrupt Vectors

#### Enabling Interrupts

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

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

##### IE – Interrupt Enable

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| IE | A8H, All Pages, Bit addressable | 0000 \_0000 b |

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

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

##### EIE0 – Extensive Interrupt Enable

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| EIE0 | 9BH, Page 0 | 0000 \_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **ET2** | **Enable Timer 2 Interrupt**  0 = Timer 2 interrupt Disabled.  1 = Timer 2 interrupt Enable. When interrupt generated, TF2 (T2CON.7) set 1 |
| [6] | **ESPI0** | **Enable SPI Interrupt**  0 = SPI interrupt Disabled.  1 = SPI interrupt Enable. When interrupt generated SPIF (SPInSR.7), SPIOVF (SPInSR.5), or MODF (SPInSR.4) set 1 . |
| [5] | **EFB0** | **Enable Fault Brake Interrupt**  0 = Fault Brake interrupt Disabled.  1 = Fault Brake interrupt Enable. When interrupt generated FBF (PWM0FBD.7) set 1. |
| [4] | **EWDT** | **Enable WDT Interrupt**  0 = WDT interrupt Disabled.  1 = WDT interrupt Enable. When interrupt generated WDTF (WDCON.5) set 1. |
| [3] | **EPWM0** | **Enable PWM0 Interrupt**  0 = PWM interrupt Disabled.  1 = PWM interrupt Enable. When interrupt generated PWMF (PWMnCON0.5) set 1. |
| [2] | **ECAP** | **Enable Input Capture Interrupt**  0 = Input capture interrupt Disabled.  1 = Input capture interrupt Enable. When interrupt generated CAPF[2:0] (CAPCON0[2:0]) set 1. |
| [1] | **EPI** | **Enable Pin Interrupt**  0 = Pin interrupt Disabled.  1 = Pin interrupt Enable. When interrupt generated PIF related bit set 1. |
| [0] | **EI2C0** | **Enable I2C0 Interrupt**  0 = I2C interrupt Disabled.  1 = I2C interrupt Enable. When interrupt generated SI (I2C0CON.3) or I2TOF (I2C0TOC.0) set 1. |

##### EIE1 – Extensive Interrupt Enable 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| EIE1 | 9CH, Page 0 | 0000 \_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **-** | Rreserved |
| [5] | **EI2C1** | **Enable I2C1 Interrupt**  0 = I2C1 interrupt Disabled.  1 = I2C1 interrupt Enable. When interrupt generated SI (I2C1CON.3) or I2TOF (I2C1TOC.0) set 1. |
| [4] | **ESPI1** | **Enable SPI1 Interrupt**  0 = SPI1 interrupt Disabled.  1 = SPI1 interrupt Enable. When interrupt generated SPIF (SP2SR.7), MODF (SP2SR.4) or SPIOVF (SP2SR.5) set 1 |
| [3] | **EHFI** | **Enable Hard Fault Interrupt**  0 = hard fault interrupt Disabled and hard fault reset is Enabled.  1 = hard fault interrupt Enable. When interrupt generated HFIF (AUXR0.4) set 1. |
| [2] | **EWKT** | **Enable WKT Interrupt**  0 = WKT interrupt Disabled.  1 = WKT interrupt Enable. When interrupt generated WKTF (WKCON.4) set 1. |
| [1] | **ET3** | **Enable Timer 3 Interrupt**  0 = Timer 3 interrupt Disabled.  1 = Timer 3interrupt Enable. When interrupt generated TF3 (T3CON.4) set 1. |
| [0] | **ES1** | **Enable Serial Port 1 Interrupt**  0 = Serial port 1 interrupt Disabled.  1 = Serial port 1 interrupt Enable. When interrupt generated TI\_1 (S1CON.1) or RI\_1 (S1CON.0) set 1. |

##### SCnIE – SC Interrupt Enable Control Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SC0IE | DDH, Page 0 | 0000\_0000 b |

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

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

##### DMAnCR – PDMAn Control Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| DMA0CR0 | 92H, Page 0 | 0000\_0000 b |
| DMA1CR0 | EBH, Page 0 | 0000\_0000 b |

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

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

#### 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 6.2‑5 Interrupt Priority Level Setting lists four priority setting. Naturally, a low level priority interrupt can itself be interrupted by a high level priority interrupt, but not by any same level interrupt or lower level. In addition, there exists a pre-defined natural priority among the interrupts themselves. The natural priority comes into play when the interrupt controller has to resolve simultaneous requests having the same priority level.

In case of multiple interrupts, the following rules apply:

1. While a low priority interrupt handler is running, if a high priority interrupt arrives, the handler will be interrupted and the high priority handler will run. When the high priority handler does “RETI”, the low priority handler will resume. When this handler does “RETI”, control is passed back to the main program.

2. If a high priority interrupt is running, it cannot be interrupted by any other source – even if it is a high priority interrupt which is higher in natural priority.

3. A low-priority interrupt handler will be invoked only if no other interrupt is already executing. Again, the low priority interrupt cannot preempt another low priority interrupt, even if the later one is higher in natural priority.

4. If two interrupts occur at the same time, the interrupt with higher priority will execute first. If both interrupts are of the same priority, the interrupt which is higher in natural priority will be executed first. This is the only context in which the natural priority matters.

|  |  |  |
| --- | --- | --- |
| **Interrupt Priority Control Bits** | | **Interrupt Priority Level** |
| **IPH/EIPH0/EIPH1/EIPH2** | **IP/EIP0/EIP1/EIP2** |
| 0 | 0 | Level 0 (lowest) |
| 0 | 1 | Level 1 |
| 1 | 0 | Level 2 |
| 1 | 1 | Level 3 (highest) |

Table 6.2‑5 Interrupt Priority Level Setting

This natural priority is defined as shown on Table 6.2‑6 Characteristics of Each Interrupt SourceIt also summarizes the interrupt sources, flag bits, vector addresses, enable bits, priority bits, natural priority and the permission to wake up the CPU from Power-down mode. For details of waking CPU up from Power-down mode, please see Section 6.2.2.2 Power-down Mode

| **Interrupt Source** | **Vector Address** | **Interrupt Flag** | **Enable Bit** | **Natural Priority** | **Priority Control Bits** | **PD**  **Wake-Up** |
| --- | --- | --- | --- | --- | --- | --- |
| Reset | 0000H | - | Always Enabled | Highest | - | Yes |
| CPU Hard Fault | 0093H | HFIF (RSR.5) | EHFI (EIE1.3) | 1 | PHF, PHFH | No |
| External interrupt 0 | 0003H | IE0 (TCON.1) | EX0 (I.E.0) | 2 | PX0, PX0H | Yes |
| Brown-out | 0043H | BOF (BODCON0.3) | EBOD (I.E.5) | 3 | PBOD, PBODH | Yes |
| Watchdog Timer | 0053H | WDTF (WDCON.5) | EWDT (EIE0.4) | 4 | PWDT, PWDTH | Yes |
| Timer 0 | 000BH | TF0 (TCON.5) | ET0 (I.E.1) | 5 | PT0, PT0H | No |
| I2C0 status/time-out | 0033H | SI (I2C0CON.3)  I2TOF (I2C0TOC.0) | EI2C0 (EIE0.0) | 6 | PI2C0, PI2C0H | No |
| External interrupt 1 | 0013H | IE1 (TCON.3) | EX1 (I.E.2) | 8 | PX1, PX1H | Yes |
| Pin interrupt | 003BH | PIF0 (PIF.0)  PIF1 (PIF.1)  PIF2 (PIF.2)  PIF3 (PIF.3)  PIF4 (PIF.4)  PIF5 (PIF.5)  PIF6 (PIF.6)  PIF7 (PIF.7) | EPI (EIE0.1) | 9 | PPI, PPIH | Yes |
| Timer 1 | 001BH | TF1 (TCON.7) | ET1 (I.E.3) | 10 | PT1, PT1H | No |
| Serial port 0 | 0023H | RI (SCON.0)  TI (SCON.1) | ES (I.E.4) | 11 | PS, PSH | No |
| PWM0 Fault Brake event | 0073H | FBF (PWM0FBD.7) | EFB0 (EIE0.5) | 12 | PFB, PFBH | No |
| SPI0 | 004BH | SPIF (SPI0SR.7)  MODF (SPI0SR.4)  SPIOVF (SPI0SR.5) | ESPI0 (EIE0.6) | 13 | PSPI, PSPIH | No |
| Timer 2 | 002BH | TF2 (T2CON.7) | ET2 (EIE0.7) | 14 | PT2, PT2H | No |
| Input capture | 0063H | CAPF0 (CAPCON0.0)  CAPF1 (CAPCON0.1)  CAPF2 (CAPCON0.2) | ECAP (EIE0.2) | 15 | PCAP, PCAPH | No |
| PWM0 interrupt | 006BH | PWMF (PWM0CON0.5) | EPWM0 (EIE0.3) | 16 | PPWM0, PPWM0H | No |
| Serial port 1 | 007BH | RI\_1 (S1CON.0)  TI\_1 (S1CON.1) | ES1 (EIE1.0) | 17 | PS1, PS1H | No |
| Timer 3 | 0083H | TF3 (T3CON.4) | ET3 (EIE1.1) | 18 | PT3, PT3H | No |
| Self Wake-up Timer | 008BH | WKTF (WKCON.4) | EWKT (EIE1.2) | 19 | PWKT, PWKTH | Yes |
| SMC0 | 009BH | RDAIF (SC0IS.0)  TBEIF (SC0IS.1)  TERRIF (SC0IS.2)  BGTIF (SC0IS.3)  ACERRIF (SC0IS.4) | RDAIEN (SC0I.E.0)  TBEIEN (SC0I.E.1)  TERRIEN (SC0I.E.2)  BGTIEN (SC0I.E.3)  ACERRIEN (SC0I.E.4) | 20 | SMC0, SMC0H | No |
| PDMA0 | 00A3H | FDONE (DMA0TSR.0)  HDONE (DMA0TSR.1) | FIE (DMA0CR0.2)  HIE (DMA0CR0.3) | 21 | PDMA0, PDMA0H | No |
| PDMA1 | 00ABH | FDONE (DMA1TSR.0)  HDONE (DMA1TSR.1) | FIE (DMA1CR0.2)  HIE (DMA1CR0.3) | 22 | PDMA1, PDMA1H | No |
| ACMP | 00BBH | ACMP0IF (ACMPSR.0)  ACMP1IF (ACMPSR.2) | CMPIE (ACMPCR0.1) CMPIE (ACMPCR1.1) | 24 | PACMP, PACMPH | Yes |
| I2C1 status/time-out | 00C3h | SI (I2C1CON.3)  I2TOF (I2C1TOC.0) | EI2C1 (EIE1.5) | 25 | PI2C1, PI2C1H | No |
| **Note:**  1. While the external interrupt pin is set as edge triggered (Itx = 1), its own flag Iex will be automatically cleared if the interrupt service routine (ISR) is executed. While as level triggered (Itx = 0), Iex follows the inverse of respective pin state. It is not controlled via software.  2. TF0, TF1, or TF3 is automatically cleared if the interrupt service routine (ISR) is executed. On the contrary, be aware that TF2 is not.  3. If level triggered is selected for pin interrupt channel n, PIFn flag reflects the respective channel state. It is not controlled via software. | | | | | | |

Table 6.2‑6 Characteristics of Each Interrupt Source

##### IP – Interrupt Priority

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| IP | B8H, All Pages, Bit addressable | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **-** | Reserved |
| [6] | **PADC** | ADC interrupt priority low bit |
| [5] | **PBOD** | Brown-out detection interrupt priority low bit |
| [4] | **PS** | Serial port 0 interrupt priority low bit |
| [3] | **PT1** | Timer 1 interrupt priority low bit |
| [2] | **PX1** | External interrupt 1 priority low bit |
| [1] | **PT0** | Timer 0 interrupt priority low bit |
| [0] | **PX0** | External interrupt 0 priority low bit |
| **Note:** used in combination with the IPH to determine the priority of each interrupt source. See Table 6.2‑5 Interrupt Priority Level Setting for correct interrupt priority configuration. | | |

##### IPH – Interrupt Priority High

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| IPH | B7H, Page 0 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | **PADCH** | **PBODH** | **PSH** | **PT1H** | **PX1H** | **PT0H** | **PX0H** |
| - | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **-** | Rreserved |
| [6] | **PADCH** | ADC interrupt priority high bit |
| [5] | **PBODH** | Brown-out detection interrupt priority high bit |
| [4] | **PSH** | Serial port 0 interrupt priority high bit |
| [3] | **PT1H** | Timer 1 interrupt priority high bit |
| [2] | **PX1H** | External interrupt 1 priority high bit |
| [1] | **PT0H** | Timer 0 interrupt priority high bit |
| [0] | **PX0H** | External interrupt 0 priority high bit |

##### EIP0 – Extensive Interrupt Priority

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| EIP0 | EFH, Page 0 | 0000\_0000 b |

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

| Bit | Name | Description |
| --- | --- | --- |
| [7] | **PT2** | Timer 2 interrupt priority low bit |
| [6] | **PSPI0** | SPI0 interrupt priority low bit |
| [5] | **PFB** | Fault Brake interrupt priority low bit |
| [4] | **PWDT** | WDT interrupt priority low bit |
| [3] | **PPWM0** | PWM interrupt priority low bit |
| [2] | **PCAP** | Input capture interrupt priority low bit |
| [1] | **PPI** | Pin interrupt priority low bit |
| [0] | **PI2C0** | I2C interrupt priority low bit |
| **Note:** EIP0 is used in combination with the EIPH0 to determine the priority of each interrupt source. See Table 6.2‑5 Interrupt Priority Level Setting for correct interrupt priority configuration. | | |

##### EIPH0 – Extensive Interrupt Priority High

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| EIPH0 | F7H, Page 0 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **PT2H** | Timer 2 interrupt priority high bit |
| [6] | **PSPI0H** | SPI0 interrupt priority high bit |
| [5] | **PFBH** | Fault Brake interrupt priority high bit |
| [4] | **PWDTH** | WDT interrupt priority high bit |
| [3] | **PPWM0H** | PWM0 interrupt priority high bit |
| [2] | **PCAPH** | Input capture interrupt priority high bit |
| [1] | **PPIH** | Pin interrupt priority high bit |
| [0] | **PI2C0H** | I2C interrupt priority high bit |
| **Note:** EIPH0 is used in combination with the EIP0 to determine the priority of each interrupt source. See Table 6.2‑5 Interrupt Priority Level Setting for correct interrupt priority configuration. | | |

##### EIP1 – Extensive Interrupt Priority 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| EIP1 | FEH, Page 0 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **-** | Rreserved |
| [6] | **PDMA1** | PDMA1 interrupt priority low bit |
| [5] | **PDMA0** | PDMA0 interrupt priority low bit |
| [4] | **PSMC** | SMC interrupt priority low bit |
| [3] | **PHF** | Hard fault interrupt priority low bit |
| [2] | **PWKT** | WKT interrupt priority low bit |
| [1] | **PT3** | Timer 3 interrupt priority low bit |
| [0] | **PS1** | Serial port 1 interrupt priority low bit |
| **Note:** EIP1 is used in combination with the EIPH1 to determine the priority of each interrupt source. See Table 6.2‑5 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** |
| **-** | **PDMA1H** | **PDMA0H** | **PSMCH** | **PHFH** | **PWKTH** | **PT3H** | **PS1H** |
| - | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **-** | Rreserved |
| [6] | **PDMA1H** | PDMA1 interrupt priority high bit |
| [5] | **PDMA0H** | PDMA0 interrupt priority high bit |
| [4] | **PSMCH** | SMC interrupt priority high bit |
| [3] | **PHFH** | Hard fault interrupt priority high bit |
| [2] | **PWKTH** | WKT interrupt priority high bit |
| [1] | **PT3H** | Timer 3 interrupt priority high bit |
| [0] | **PS1H** | Serial port 1 interrupt priority high bit |
| **Note:** EIPH1 is used in combination with the EIP1 to determine the priority of each interrupt source. See Table 6.2‑5 Interrupt Priority Level Setting for correct interrupt priority configuration. | | |

##### EIP2 – Extensive Interrupt Priority 2

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| EIP2 | ACH, Page 0 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | | | **PI2C1** | **PACMP** |
| - | | | | | | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:2] | **-** | Reserved |
| [1] | **PI2C1** | I2C interrupt priority low bit |
| [0] | **PACMP** | ACMP interrupt priority low bit |
| **Note:** EIP2 is used in combination with the EIPH2 to determine the priority of each interrupt source. See Table 6.2‑5 Interrupt Priority Level Setting for correct interrupt priority configuration. | | |

##### EIPH2 – Extensive Interrupt Priority High 2

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| EIPH2 | ADH, Page 0 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | | | **PI2C1H** | **PACMPH** |
| - | | | | | | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:2] | **-** | Reserved |
| [1] | **PI2C1H** | I2C interrupt priority high bit |
| [0] | **PACMPH** | ACMP interrupt priority high bit |
| **Note:** EIPH2 is used in combination with the EIP2 to determine the priority of each interrupt source. See Table 6.2‑5 Interrupt Priority Level Setting for correct interrupt priority configuration. | | |

##### AUXR0 – Auxiliary Register 0

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

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **SWRF** | **Software Reset Flag**  When the MCU is reset via software reset, this bit will be set via hardware. It is recommended that the flag be cleared via software. |
| [6] | **RSTPINF** | **External Reset Flag**  When the MCU is reset by the external reset, this bit will be set via hardware. It is recommended that the flag be cleared via software. |
| [5] | **HFRF** | **Hard Fault Reset Flag**  Once CPU fetches instruction address over Flash size while EHFI (EIE1.4)=0, MCU will reset and this bit will be set via hardware. It is recommended that the flag be cleared via software.  **Note:** If MCU run in OCD debug mode and OCDEN = 0, Hard fault reset will disable. Only HFRF flag be asserted. |
| [4] | **HFIF** | **Hard Fault Interrupt Flag**  Once CPU fetches instruction address over Flash size while EHFI (EIE1.4)=1. MCU will be interrupt and this bit will be set via hardware. It is recommended that the flag be cleared via software. |

#### 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.

#### 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.

## Flash Memory Control

### 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 MUG51TBAE carried out the Flash operation with convenient mechanism to help user re-programming the Flash content by In-Application-Programming (IAP). IAP is an in-circuit electrical erasure and programming method through software.

After IAP enabling by setting IAPEN (CHPCON.0 with TA protected) and setting the enable bit in IAPUEN that allows the target block to be updated, user can easily fill the 16-bit target address in IAPAH and IAPAL, data in IAPFD, and command in IAPCN. Then the IAP is ready to begin by setting a triggering bit IAPGO (IAPTRG.0). Note that IAPTRG is also TA protected. At this moment, the CPU holds the Program Counter and the built-in IAP automation takes over to control the internal charge-pump for high voltage and the detail signal timing. The erase and program time is internally controlled disregard of the operating voltage and frequency. Nominally, a page-erase time is 5 ms and a byte-program time is 23.5 μs. After IAP action completed, the Program Counter continues to run the following instructions. The IAPGO bit will be automatically cleared. An IAP failure flag, IAPFF (CHPCON.6), can be check whether the previous IAP operation was successful or not. Through this progress, user can easily erase, program, and verify the Flash Memory by just taking care of pure software.

The following registers are related to IAP processing.

**CONFIG2**

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

Factory default value: 1111 1111b

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [3] | **BOIAP** | **Brown-Out Inhibiting IAP**  This bit decide whether IAP erasing or programming is inhibited by brown-out status. This bit is valid only when brown-out detection is enabled.  1 = IAP erasing or programming is inhibited if VDD is lower than VBOD.  0 = IAP erasing or programming is allowed under any workable VDD. |

##### CHPCON – Chip Control

|  |  |  |
| --- | --- | --- |
| **Part of Register Content** | **SFR Address** | **Reset Value** |
| CHPCON | 9FH, Page 0, TA protected | Software 0000\_00U0 b  Others 0000\_00C0 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [6] | **IAPFF** | **IAP Fault Flag**  The hardware will set this bit after IAPGO (IAPTRG.0) is set if any of the following condition is met:  (1) The accessing address is oversize.  (2) IAPCN command is invalid.  (3) IAP erases or programs updating un-enabled block.  (4) IAP erasing or programming operates under VBOD while BOIAP (CONFIG2.5) remains un-programmed 1 with BODEN (BODCON0.7) as 1 and BORST (BODCON0.2) as 0.  This bit should be cleared via software. |
| [0] | **IAPEN** | **IAP Enable**  0 = IAP function Disabled.  1 = IAP function Enabled.  Once enabling IAP function, the MIRC will be turned on for timing control. To clear IAPEN should always be the last instruction after IAP operation to stop internal oscillator if reducing power consumption is concerned. |

##### IAPUEN – IAP Updating Enable (TA Protected)

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| IAPUEN | A5H, Page 0, TA protected | 0000 \_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:5] | **-** | Reserved |
| [4] | **SPMEN** | **SPROM Memory Space Mapping Enable**  0 = CPU memory address 0xff80~0xffff is mapping to APROM memory.  1 = CPU memory address 0xff80~0xffff is mapping to SPROM memory. |
| [3] | **SPUEN** | **SPROM Memory Space Updated Enable(TA Protected)**  0 = Inhibit erasing or programming SPRO 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 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **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** | **This Byte is Used for IAP Command** For details, seeTable 6.3‑1 IAP Modes and Command Codes. |
| [4] | **FCEN** | **This Byte is Used for IAP Command** For details, seeTable 6.3‑1 IAP Modes and Command Codes. |
| [3:0] | **FCTRL[3:0]** | **This Byte is Used for IAP Command** For details, seeTable 6.3‑1 IAP Modes and Command Codes. |

##### IAPAH – IAP Address High Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| IAPAH | A7H, Page 0 | 0000 \_0000 b |

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

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

##### IAPAL – IAP Address Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| IAPAL | A6H, Page 0 | 0000 \_0000 b |

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

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

##### IAPFD – IAP Flash Data

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| IAPFD | AEH, Page 0 | 0000 \_0000 b |

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

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

##### IAPTRG – IAP Trigger

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| IAPTRG | A4H, Page 0, TA protected | 0000 \_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:1] | **-** | Reserved |
| [0] | **IAPGO** | **IAP Go**  IAP begins by setting this bit as logic 1. After this instruction, the CPU holds the Program Counter (PC) and the IAP hardware automation takes over to control the progress. After IAP action completed, the Program Counter continues to run the following instruction. The IAPGO bit will be automatically cleared and always read as logic 0.  Before triggering an IAP action, interrupts (if enabled) should be temporary disabled for hardware limitation.  The program process should follows below.  CLR EA  MOV TA,#0AAH  MOV TA,#55H  ORL IAPTRG,#01H  (SETB EA) |

#### IAP Commands

The MUG51TBAE provides a wide range of applications to perform IAP to APROM, LDROM, or CONFIG bytes. The IAP action mode and the destination of the Flash block are defined by IAP control register IAPCN.

| **IAP Mode** | **IAPCN** | | | | | **IAPA[15:0]**  **{IAPAH, IAPAL}** | **IAPFD[7:0]** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **IAPB [1:0]** | **FOEN** | **FCEN** | | **FCTRL**  **[3:0]** |
| Company ID read | XX[1] | 0 | | 0 | 1011 | X | DAH |
| Device ID read | XX | 0 | | 0 | 1100 | Low-byte DID: 0000H  High-byte DID: 0001H | Low-byte DID  High-byte DID |
| PID Read | x, x | 0 | | 0 | 1100 | A[15:0]=0002H  for low-byte ID  A[15:0]=0003H  for high-byte ID | Data out  D[7:0]=PID |
| 96-bit Unique Code read | x, x | 0 | | 0 | 0100 | A[15:0]  (0x0000~0x000B) | Data out  D[7:0]=UID |
| 128-bit Die Record read | x, x | 0 | | 0 | 0100 | A[15:0]  (0x0010~0x001F) | Data out  D[7:0]=DR |
| 128-bit UCID read | x, x | 0 | | 0 | 0100 | A[15:0]  (0x0020~0x02FB) | Data out  D[7:0]=UCID |
| APROM page-erase | 00 | 1 | 0 | | 0010 | Address in[2] | FFH |
| APROM byte-program | 00 | 1 | 0 | | 0001 | Address in | Data in |
| APROM byte-read | 00 | 0 | 0 | | 0000 | Address in | Data out |
| APROM Checksum Run | 00 | 1 | 0 | | 1110 | IAPAL=0x00  IAPAH  (starting address) | Data in  (size x256bytes) |
| APROM Checksum Read | 00 | 0 | 0 | | 1110 | IAPAL=0x00  IAPAH  (starting address) | Checksum |
| LDROM page-erase | 01 | 1 | 0 | | 0010 | Address in[2] | FFH |
| LDROM byte-read | 01 | 0 | 0 | | 0000 | Address in | Data out |
| LDROM Checksum Run | 01 | 1 | 0 | | 1110 | IAPAL=0x00  IAPAH  (starting address) | Data in  (size x256bytes) |
| LDROM Checksum Read | 01 | 0 | 0 | | 1110 | IAPAL=0x00  IAPAH  (starting address) | Checksum |
|  |  |  |  | |  |  |  |
| All CONFIG bytes erase | 11 | 1 | 0 | | 0010 | 0000H | FFH |
| CONFIG byte-program | 11 | 1 | 0 | | 0001 | CONFIG0: 0000H  CONFIG1: 0001H  CONFIG2: 0002H  CONFIG4: 0004H  CONFIG6: 0005H | Data in |
| CONFIG byte-read | 11 | 0 | 0 | | 0000 | CONFIG0: 0000H  CONFIG1: 0001H  CONFIG2: 0002H  CONFIG4: 0004H  CONFIG6: 0005H | Data out |
| SPROM page-erase | 10 | 1 | 0 | | 0010 | 0180H | FFH |
| SPROM byte-program | 10 | 1 | 0 | | 0001 | 0180H~01FFH | Data in |
| SPROM byte-read | 10 | 0 | 0 | | 0000 | 0180H~01FFH | Data out |
| SPROM Erase | 1, 0 | 1 | 0 | | 0010 | A[15:0]=0x0180 | FFH |
| SPROM Program | 1, 0 | 1 | 0 | | 0001 | A[15:0]  (0x0180~0x01FF) | Data in  D[7:0] |
| SPROM Read | 1, 0 | 0 | 0 | | 0000 | A[15:0]  (0x0180~0x01FF) | Data in  D[7:0] |
| SPROM Checksum Run | 10 | 1 | 0 | | 1110 | IAPAL=0x80  IAPAH =0x01  (starting address) | Data in |
| SPROM Checksum Read | 10 | 0 | 0 | | 1110 | IAPAL=0x00  IAPAH =0x01  (starting address) | Checksum |
| **Note:**  [1] “X” means “don’t care”.  [2] Each page is 128 bytes size. Therefore, the address should be the address pointed to the target page. | | | | | | | |

Table 6.3‑1 IAP Modes and Command Codes

#### CRC-8 for Flash check

|  |
| --- |
|  |

Figure 6.3‑1. CRC-8 Block Diagram

#### 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 MIRC running. 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.

***Note:***

***During IAP progress, interrupts (if enabled) should be disabled temporally by clearing EA bit for implement limitation.***

***Do not attempt to erase or program to a page that the code is currently executing. This will cause unpredictable program behavior and may corrupt program data.***

#### Using Flash Memory as Data Storage

In general application, there is a need of data storage, which is non-volatile so that it remains its content even after the power is off. Therefore, in general application user can read back or update the data, which rules as parameters or constants for system control. The Flash Memory array of the MUG51TBAE 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.

#### 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 MUG51TBAE. 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/NuvotonMOSS/Community/ProductInfo.aspx?tp_GUID=670aaf31-5d5c-45d3-8a9e-040e148d55cf). A simple ISP demo code is given below.

### 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, nRESET , ICPDA, and ICPCK, involved in ICP function. nRESET is used to enter or exit ICP mode. ICPDA is the data input and output pin. ICPCK is the clock input pin, which synchronizes the data shifted in to or out from MCU under programming. User should leave these three pins plus VDD and GND pins on the circuit board to make ICP possible.

Nuvoton provides ICP tool for MUG51TBAE, 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](http://www.nuvoton.com/hq/products/microcontrollers/8bit-8051-mcus/Software/?__locale=zh_TW&resourcePage=Y).

### On-Chip-Debugger (ICE)

#### Functional Description

The MUG51TBAE is embedded in an on-chip-debugger (OCD) providing developers with a low cost method for debugging user code, which is available on each package. The OCD gives debug capability of complete program flow control with eight hardware address breakpoints, single step, free running, and non-intrusive commands for memory access. The OCD system does not occupy any locations in the memory map and does not share any on-chip peripherals.

When the OCDEN (CONFIG0.4) is programmed as 0 and LOCK (CONFIG0.1) remains un-programmed as 1, the OCD is activated. The OCD cannot operate if chip is locked. The OCD system uses a two-wire serial interface, OCDDA and OCDCK, to establish communication between the target device and the controlling debugger host. OCDDA is an input/output pin for debug data transfer and OCDCK is an input pin for synchronization with OCDDA data. The nRESETpin is also necessary for OCD mode entry and exit. The MUG51TBAE supports OCD with Flash Memory control path by ICP writer mode, which shares the same three pins of OCD interface.

The MUG51TBAE uses OCDDA, OCDCK, and 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. nRESET cannot be connected directly to VDD and any external capacitors connected must be removed.

2. All external reset sources must be disconnected.

3. Any external component connected on OCDDA and OCDCK must be isolated.

#### Limitation of OCD

The MUG51TBAE 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 nRESET pin needs to be used for OCD mode selection.

2. The OCDDA pin is physically located on the same pin P5.0. Therefore, neither its I/O function nor shared multi-functions can be emulated.

3. The OCDCK pin is physically located on the same pin as P5.1. Therefore, neither its I/O function nor shared multi-functions can be emulated.

4. When the system is in Idle or Power-down mode, it is invalid to perform any accesses because parts of the device may not be clocked. A read access could return garbage or a write access might not succeed.

5. MIRC cannot be turned off because OCD uses this clock to monitor its internal status. The instruction that turns off MIRC affects nothing if executing under debug mode. When CPU enters its Power-down mode under debug mode, MIRC keeps turning on.

The MUG51TBAE OCD system has another limitation that non-intrusive commands cannot be executed at any time while the user’s program is running. Non-intrusive commands allow a user to read or write MCU memory locations or access status and Register Description with the debug controller. A reading or writing memory or control register space is allowed only when MCU is under halt condition after a matching of the hardware address breakpoint or a single step running.

##### CONFIG0

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

Factory default value: 1111 1111b

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [5] | **OCDPWM** | **PWM Output State Under OCD Halt**  This bit decides the output state of PWM when OCD halts CPU.  1 = Tri-state pins those are used as PWM outputs.  0 = PWM continues. |
| [4] | **OCDEN** | **OCD Enable**  1 = OCD Disabled.  0 = OCD Enabled.  **Note:** If MCU run in OCD debug mode and OCDEN = 0, hard fault reset will disable. Only HardF flag be asserted. |

Following is the OCD Relation multi-function pin define list. As default when CONFIG define OCD enabled, P5.1 and P5.0 pin auto setting as ICE\_CLK and ICE\_DAT function.

| **Group** | **Pin Name** | **GPIO** | **MFP** | **Type** | **Description** |
| --- | --- | --- | --- | --- | --- |
| ICE | ICE\_CLK | P5.1 | MFP14 | I | Serial wired debugger clock pin. |
| ICE\_DAT | P5.0 | MFP14 | O | Serial wired debugger data pin. |

## GPIO Port Structure and Operation

### GPIO Mode

The MUG51TBAE has a maximum of 24 general purpose I/O pins which 18 bit-addressable general I/O pins grouped as 5 ports, P0 to P4, and general I/O pins grouped as P5. Each port has its port control register (Px register). The writing and reading of a port control register have different meanings. A write to port control register sets the port output latch logic value, where as a read gets the port pin logic state. These four modes are quasi-bidirectional (standard 8051 port structure), push-pull, input-only, and open-drain modes. Each port spends two special function registers PxM1 and PxM2 to select the I/O mode of port Px. The list below illustrates how to select the I/O mode of Px.n. Note that the default configuration of is input-only (high-impedance) after any reset.

| PnM1.X[1] | PnM2.X[1] | I/O Type |
| --- | --- | --- |
| 0 | 0 | Quasi-bidirectional |
| 0 | 1 | Push-pull |
| 1 | 0 | Input-only (high-impedance) |
| 1 | 1 | Open-drain |
| **Note:** N = 0~5, x = 0~7 | | |

Table 6.4‑1 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 Register Description are PxSR. By default, the slew rate is slow. If user would like to increase the I/O output speed, setting the corresponding bit in PxSR, the slew rate is selected in a faster level.

For example:

P0M1 |= 0x40;

P0M2 &= 0xBF; //Set P0.6 as input only mode

* + - * 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

* + - * 1. 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

* + - * 1. 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 VDD 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

* + - * 1. 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 VDD. 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

#### 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.

##### P0 – Port 0

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P0 | 80H, All Pages, Bit-addressable | 1111\_1111 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | **P0\_3** | **P0\_2** | **P0\_1** | **P0\_0** |
| - | | | | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [3:0] | **n[3:0]** | **Port 0**  Port 0 is an maximum 4-bit general purpose I/O port. |

##### P1 – Port 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P1 | 90H, All Pages, Bit-addressable | 1111\_1111 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P1\_7** | **P1\_6** | **-** | | | | | |
| R/W | R/W | - | | | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **n[7:6]** | **Port 1**  Port 1 is an maximum 2-bit general purpose I/O port. |

##### P2 – Port 2

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P2 | A0H, All Pages, Bit-addressable | 1111\_1111 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | **P2\_5** | **P2\_4** | **P2\_3** | **P2\_2** | **P2\_1** | **P2\_0** |
| - | | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [5:0] | **n[5:0]** | **Port 2**  Port 2 is an maximum 6-bit general purpose I/O port. |

##### P3 – Port 3

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P3 | B0H, All Pages, Bit-addressable | 1111\_1111 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | **P3\_3** | **P3\_2** | **P3\_1** | **P3\_1** |
| - | | | | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [3:0] | **n[3:0]** | **Port 3**  Port 3 is an maximum 4-bit general purpose I/O port. |

##### P4 – Port 4

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P4 | D8H, All Pages, Bit-addressable | 1111\_1111 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | | | **P4\_1** | **P4\_0** |
| - | | | | | | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:0] | **n[7:0]** | **Port 4**  Port 4 is an maximum 8-bit general purpose I/O port. |

##### P5 – Port 5

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P5 | B1H, Page 0 | 1111\_1111 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | **P5\_5** | **P5\_4** | **P5\_3** | **P5\_2** | **P5\_1** | **P5\_0** |
| - | | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [5:0] | **n[5:0]** | **Port 5**  Port 5 is an maximum 6-bit general purpose I/O port. |

#### 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.

##### P0M1 – Port Mode Select 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P0M1 | B1H, Page 1 | 1111\_1111 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | **P0M1\_3** | **P0M1\_2** | **P0M1\_1** | **P0M1\_0** |
| - | | | | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [3:0] | **n[3:0]** | **Port 0 mode select 1**  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. |

##### P1M1 – Port Mode Select 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P1M1 | B3H, Page 1 | 1111\_1111 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P1M1\_7** | **P1M1\_6** | **-** | | | | | |
| R/W | R/W | - | | | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **n[7:6]** | Port 1 mode select 1 |

##### P2M1 – Port Mode Select 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P2M1 | B5H, Page 1 | 1111\_1111 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | **P2M1\_5** | **P2M1\_4** | **P2M1\_3** | **P2M1\_2** | **P2M1\_1** | **P2M1\_0** |
| - | | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [5:0] | **n[5:0]** | Port 2 mode select 1 |

##### P3M1 – Port Mode Select 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P3M1 | C2H, Page 1 | 1111\_1111 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | **P3M1\_3** | **P3M1\_2** | **P3M1\_1** | **P3M1\_0** |
| - | | | | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [4:0] | **n[4:0]** | Port 3 mode select 1 |

##### P4M1 – Port Mode Select 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P4M1 | B9H, Page 1 | 1111\_1111 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | | | **P4M1\_1** | **P4M1\_0** |
| - | | | | | | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [1:0] | **n[1:0]** | Port 4 mode select 1 |

##### P5M1 – Port Mode Select 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P5M1 | BDH, Page 1 | 1111\_1111 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | **P5M1\_5** | **P5M1\_4** | **P5M1\_3** | **P5M1\_2** | **P5M1\_1** | **P5M1\_0** |
| - | | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| 76 | **-** | Reserved |
| [5:0] | **n[5:0]** | Port 5 mode select 1 |

##### P0M2 – Port Mode Select 2

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P0M2 | B2H, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | **P0M2\_3** | **P0M2\_2** | **P0M2\_1** | **P0M2\_0** |
| - | | | | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [3:0] | **n[3:0]** | **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. |

##### P1M2 – Port Mode Select 2

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P1M2 | B4H, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P1M2\_7** | **P1M2\_6** | **-** | | | | | |
| R/W | R/W | - | | | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **n[7:6]** | **Port 1 Mode Select 2** |
| [5:0] | **-** | **Reserved** |

##### P2M2 – Port Mode Select 2

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P2M2 | B6H, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | **P2M2\_5** | **P2M2\_4** | **P2M2\_3** | **P2M2\_2** | **P2M2\_1** | **P2M2\_0** |
| - | | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| 76 | **-** | Reserved |
| [5:0] | **n[5:0]** | **Port 2 Mode Select 2** |

##### P3M2 – Port Mode Select 2

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P3M2 | C3H, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | **P3M2\_3** | **P3M2\_2** | **P3M2\_1** | **P3M2\_0** |
| - | | | | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:4] | **-** | Reserved |
| [3:0] | **n[3:0]** | **Port 3 Mode Select 2** |

##### P4M2 – Port Mode Select 2

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P4M2 | BAH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | | | **P4M2\_1** | **P4M2\_0** |
| - | | | | | | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:2] | **-** | Reserved |
| [1:0] | **n[1:0]** | **Port 4 Mode Select 2** |

##### P5M2 – Port Mode Select 2

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P5M2 | BEH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | **P5M2\_5** | **P5M2\_4** | **P5M2\_3** | **P5M2\_2** | **P5M2\_1** | **P5M2\_0** |
| - | | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **-** | Reserved |
| [5:0] | **n[5:0]** | **Port 5 Mode Select 2** |

#### GPIO Multi-Function Select

##### P0MF10 – Pn.1 and Pn.0 Multi-function Select

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P0MF10 | F9H, Page 2 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P0MF1[3:0]** | | | | **P0MF0[3:0]** | | | |
| R/W | | | | R/W | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:4] | **P0MF1[3:0]** | P0.1 multi-function select |
| [3:0] | **P0MF0[3:0]** | P0.0 multi-function select |

##### P2MF10 – Pn.1 and Pn.0 Multi-function Select

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P2MF10 | F2H, Page 2 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P2MF1[3:0]** | | | | **P2MF0[3:0]** | | | |
| R/W | | | | R/W | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:4] | **P2MF1[3:0]** | P2.1 multi-function select |
| [3:0] | **P2MF0[3:0]** | P2.0 multi-function select |

##### P3MF10 – Pn.1 and Pn.0 Multi-function Select

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P3MF10 | F6H, Page 2 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P3MF1[3:0]** | | | | **P3MF0[3:0]** | | | |
| R/W | | | | R/W | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:4] | **P3MF1[3:0]** | P3.1 multi-function select |
| [3:0] | **P3MF0[3:0]** | P3.0 multi-function select |

##### P4MF10 – Pn.1 and Pn.0 Multi-function Select

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P4MF10 | EBH, Page 2 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P4MF1[3:0]** | | | | **P4MF0[3:0]** | | | |
| R/W | | | | R/W | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:4] | **P4MF1[3:0]** | P4.1 multi-function select |
| [3:0] | **P4MF0[3:0]** | P4.0 multi-function select |

##### P5MF10 – Pn.1 and Pn.0 Multi-function Select

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P5MF10 | EFH, Page 2 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P5MF1[3:0]** | | | | **P5MF0[3:0]** | | | |
| R/W | | | | R/W | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:4] | **P5MF1[3:0]** | P5.1 multi-function select |
| [3:0] | **P5MF0[3:0]** | P5.0 multi-function select |

##### P0MF32 – Pn.3 and Pn.2 Multi-function Select

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P0MF32 | FAH, Page 2 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P0MF3[3:0]** | | | | **P0MF2[3:0]** | | | |
| R/W | | | | R/W | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:4] | **P0MF3[3:0]** | P0.3 multi-function select |
| [3:0] | **P0MF2[3:0]** | P0.2 multi-function select |

##### P2MF32 – Pn.3 and Pn.2 Multi-function Select

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P2MF32 | F3H, Page 2 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P2MF3[3:0]** | | | | **P2MF2[3:0]** | | | |
| R/W | | | | R/W | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:4] | **P2MF3[3:0]** | P2.3 multi-function select |
| [3:0] | **P2MF2[3:0]** | P2.2 multi-function select |

##### P3MF32 – Pn.3 and Pn.2 Multi-function Select

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P3MF32 | F7H, Page 2 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P3MF3[3:0]** | | | | **P3MF2[3:0]** | | | |
| R/W | | | | R/W | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:4] | **P3MF3[3:0]** | P3.3 multi-function select |
| [3:0] | **P3MF2[3:0]** | P3.2 multi-function select |

##### P5MF32 – Pn.3 and Pn.2 Multi-function Select

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P5MF32 | E1H, Page 2 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P5MF3[3:0]** | | | | **P5MF2[3:0]** | | | |
| R/W | | | | R/W | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:4] | **P5MF3[3:0]** | P5.3 multi-function select |
| [3:0] | **P5MF2[3:0]** | P5.2 multi-function select |

##### P2MF54 – Pn.5 and Pn.4 Multi-function Select

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P2MF54 | F4H, Page 2 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P2MF5[3:0]** | | | | **P2MF4[3:0]** | | | |
| R/W | | | | R/W | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:4] | **P2MF5[3:0]** | P2.5 multi-function select |
| [3:0] | **P2MF4[3:0]** | P2.4 multi-function select |

##### P5MF54 – Pn.5 and Pn.4 Multi-function Select

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P5MF54 | E2H, Page 2 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P5MF5[3:0]** | | | | **P5MF4[3:0]** | | | |
| R/W | | | | R/W | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:4] | **P5MF5[3:0]** | P5.5 multi-function select |
| [3:0] | **P5MF4[3:0]** | P5.4 multi-function select |

##### P1MF76 – Pn.7 and Pn.6 Multi-function Select

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P1MF76 | F1H, Page 2 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P1MF7[3:0]** | | | | **P1MF6[3:0]** | | | |
| R/W | | | | R/W | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:4] | **P1MF7[3:0]** | P1.7 multi-function select |
| [3:0] | **P1MF6[3:0]** | P1.6 multi-function select |

#### Input Type

Each I/O pin can be configured individually as TTL input or Schmitt triggered input. Note that all of PxS registers are accessible by switching SFR page to Page 1.

##### P0S – Port n Schmitt Triggered Input

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P0S | 99H, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | **P0S\_3** | **P0S\_2** | **P0S\_1** | **P0S\_0** |
| - | | | | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [3:0] | **P0S[3:0]** | **P0 Schmitt Triggered Input**  0 = TTL level input of Pn.x.  1 = Schmitt triggered input of Pn.x. |

##### P1S – Port n Schmitt Triggered Input

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P1S | 9BH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P1S\_7** | **P1S\_6** | **-** | | | | | |
| R/W | R/W | - | | | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **P1S[7:6]** | **P1Schmitt Triggered Input**  0 = TTL level input of Pn.x.  1 = Schmitt triggered input of Pn.x. |
| [5:0] | **-** | Reserved |

##### P2S – Port n Schmitt Triggered Input

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P2S | 9DH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | **P2S\_5** | **P2S\_4** | **P2S\_3** | **P2S\_2** | **P2S\_1** | **P2S\_0** |
| - | | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [5:0] | **P2S[5:0]** | **P2 Schmitt Triggered Input**  0 = TTL level input of Pn.x.  1 = Schmitt triggered input of Pn.x. |

##### P3S – Port n Schmitt Triggered Input

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P3S | ACH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | **P3S\_3** | **P3S\_2** | **P3S\_1** | **P3S\_0** |
| - | | | | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [3:0] | **P3S[3:0]** | **P3 Schmitt Triggered Input**  0 = TTL level input of Pn.x.  1 = Schmitt triggered input of Pn.x. |

##### P4S – Port n Schmitt Triggered Input

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P4S | BBH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | | | **P4S\_1** | **P4S\_0** |
| - | | | | | | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [1:0] | **P4S[1:0]** | **P4 Schmitt Triggered Input**  0 = TTL level input of Pn.x.  1 = Schmitt triggered input of Pn.x. |

##### P5S – Port n Schmitt Triggered Input

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P5S | BFH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | **P5S\_5** | **P5S\_4** | **P5S\_3** | **P5S\_2** | **P5S\_1** | **P5S\_0** |
| - | | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [5:0] | **P5S[5:0]** | **P5 Schmitt Triggered Input**  0 = TTL level input of Pn.x.  1 = Schmitt triggered input of Pn.x. |

#### 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.

##### P0SR –Port n Slew Rate Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P0SR | 9AH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | **P0SR\_3** | **P0SR\_2** | **P0SR\_1** | **P0SR\_0** |
| - | | | | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [3:0] | **n[3:0]** | **P0 Slew Rate**  0 = Pn.x normal output slew rate.  1 = Pn.x high-speed output slew rate. |

##### P1SR –Port n Slew Rate Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P1SR | 9CH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P1SR\_7** | **P1SR\_6** | **-** | | | | | |
| R/W | R/W | - | | | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **n[7:6]** | **P1 Slew Rate**  0 = Pn.x normal output slew rate.  1 = Pn.x high-speed output slew rate. |

##### P2SR –Port n Slew Rate Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P2SR | 9EH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | **P2SR\_5** | **P2SR\_4** | **P2SR\_3** | **P2SR\_2** | **P2SR\_1** | **P2SR\_0** |
| - | | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [5:0] | **n[5:0]** | **P2 Slew Rate**  0 = Pn.x normal output slew rate.  1 = Pn.x high-speed output slew rate. |

##### P3SR –Port n Slew Rate Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P3SR | ADH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | **P3SR\_3** | **P3SR\_2** | **P3SR\_1** | **P3SR\_0** |
| - | | | | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [3:0] | **n[3:0]** | **P3 Slew Rate**  0 = Pn.x normal output slew rate.  1 = Pn.x high-speed output slew rate. |

##### P4SR –Port n Slew Rate Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P4SR | BCH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | | | **P4SR\_1** | **P4SR\_0** |
| - | | | | | | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [3:0] | **n[3:0]** | **P4 Slew Rate**  0 = Pn.x normal output slew rate.  1 = Pn.x high-speed output slew rate. |

##### P5SR –Port n Slew Rate Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P5SR | AEH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | **P5SR\_5** | **P5SR\_4** | **P5SR\_3** | **P5SR\_2** | **P5SR\_1** | **P5SR\_0** |
| - | | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [5:0] | **n[5:0]** | **P5 Slew Rate**  0 = Pn.x normal output slew rate.  1 = Pn.x high-speed output slew rate. |

#### Pull-Up Resister Control

Pull up resister for each I/O pin is configurable individually. But even enabled the pull up resister only effect when GPIO setting as input mode. By default, after reset each pin pull high resister is disabled.

##### P0UP – Port n Pull-up Resister Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P0UP | 92H, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | **P0UP\_3** | **P0UP\_2** | **P0UP\_1** | **P0UP\_0** |
| - | | | | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [3:0] | **n[3:0]** | **P0 Pull-Up Enable**  0 = P0.n pull-up Disabled.  1 = P0.n pull-up Enabled. |

##### P1UP – Port n Pull-up Resister Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P1UP | 93H, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P1UP\_7** | **P1UP\_6** | **-** | | | | | |
| R/W | R/W | - | | | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **n[7:6]** | **P1 Pull-Up Enable**  0 = P1.n pull-up Disabled.  1 = P1.n pull-up Enabled. |

##### P2UP – Port n Pull-up Resister Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P2UP | 94H, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | **P2UP\_5** | **P2UP\_4** | **P2UP\_3** | **P2UP\_2** | **P2UP\_1** | **P2UP\_0** |
| - | | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [5:0] | **n[5:0]** | **P2 Pull-Up Enable**  0 = P2.n pull-up Disabled.  1 = P2.n pull-up Enabled. |

##### P3UP – Port n Pull-up Resister Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P3UP | 95H, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | **P3UP\_3** | **P3UP\_2** | **P3UP\_1** | **P3UP\_0** |
| - | | | | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [3:0] | **n[3:0]** | **P3 Pull-Up Enable**  0 = P3.n pull-up Disabled.  1 = P3.n pull-up Enabled. |

##### P4UP – Port n Pull-up Resister Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P4UP | 96H, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | | | **P4UP\_1** | **P4UP\_0** |
| - | | | | | | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [1:0] | **P4UP[1:0]** | **P4 Pull-Up Enable**  0 = P4.n pull-up Disabled.  1 = P4.n pull-up Enabled. |

##### P5UP – Port n Pull-up Resister Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| P5UP | 97H, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | **P5UP\_5** | **P5UP\_4** | **P5UP\_3** | **P5UP\_2** | **P5UP\_1** | **P5UP\_0** |
| - | | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [5:0] | **P5UP[5:0]** | **P5.n Pull-Up Enable**  0 = P5.n pull-up Disabled.  1 = P5.n pull-up Enabled. |

#### Pull-Down Resister Control

Pull down resister for each I/O pin is configurable individually. Even enabled the pull down resister only effect when GPIO setting as input mode. By default, after reset each pin pull high resister is disabled.

##### P0DW – Port n Pull-down Resister Control

| **Register** | **SFR Address** | **Reset Value** |
| --- | --- | --- |
| P0DW | 8AH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | **P0DW\_3** | **P0DW\_2** | **P0DW\_1** | **P0DW\_0** |
| - | | | | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [3:0] | **P0DW[3:0]** | **P0 Pull-Down Enable**  0 = P0.n pull-down Disabled.  1 = P0.n pull-down Enabled. |

##### P1DW – Port n Pull-down Resister Control

| **Register** | **SFR Address** | **Reset Value** |
| --- | --- | --- |
| P1DW | 8BH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P1DW\_7** | **P1DW\_6** | **-** | | | | | |
| R/W | R/W | - | | | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **P1DW[7:6]** | **P1 Pull-Down Enable**  0 = P1.n pull-down Disabled.  1 = P1.n pull-down Enabled. |

##### P2DW – Port n Pull-down Resister Control

| **Register** | **SFR Address** | **Reset Value** |
| --- | --- | --- |
| P2DW | 8CH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | **P2DW\_5** | **P2DW\_4** | **P2DW\_3** | **P2DW\_2** | **P2DW\_1** | **P2DW\_0** |
| - | | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [5:0] | **P2DW[5:0]** | **P2 Pull-Down Enable**  0 = P2.n pull-down Disabled.  1 = P2.n pull-down Enabled. |

##### P3DW – Port n Pull-down Resister Control

| **Register** | **SFR Address** | **Reset Value** |
| --- | --- | --- |
| P3DW | 8DH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | **P3DW\_3** | **P3DW\_2** | **P3DW\_1** | **P3DW\_0** |
| - | | | | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [3:0] | **P3DW[3:0]** | **P3 Pull-Down Enable**  0 = P3.n pull-down Disabled.  1 = P3.n pull-down Enabled. |

##### P4DW – Port n Pull-down Resister Control

| **Register** | **SFR Address** | **Reset Value** |
| --- | --- | --- |
| P4DW | 8EH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | | | | **P4DW\_1** | **P4DW\_0** |
| - | | | | | | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [1:0] | **P4DW[1:0]** | **P4 Pull-Down Enable**  0 = P4.n pull-down Disabled.  1 = P4.n pull-down Enabled. |

##### P5DW – Port n Pull-down Resister Control

| **Register** | **SFR Address** | **Reset Value** |
| --- | --- | --- |
| P5DW | 8FH, Page 1 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | **P5DW\_5** | **P5DW\_4** | **P5DW\_3** | **P5DW\_2** | **P5DW\_1** | **P5DW\_0** |
| - | | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [5:0] | **P5DW[5:0]** | **P5 Pull-Down Enable**  0 = P5.n pull-down Disabled.  1 = P5.n pull-down Enabled. |

#### GPIO Low Level Change To High Delay Timing Control

Following example timing diagram base on P4.0 all other GPIO define in the register behavior mode are the same . When P40RTH set as 00, all states of P4.0 are determined by SFR P4 bit 0. When P40RTH set as any other value then once P4 bit 0 set as 0, P4.0 will auto return to high and the delay timing base on P40RTH defined. But the delay timing is not a definite value but a range. The range value is based on the P4.0 = 0 trig start point.

|  |
| --- |
|  |

Figure 6.4‑5 GPIO Auto Return Timing And Setting

##### PRTHCON0 – Port Low Return High Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PRTHCON0 | C9H, Page 3 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **P40RTH[1:0]** | | **P30RTH[1:0]** | | **P21RTH[1:0]** | | **P20RTH[1:0]** | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **P40RTH[1:0]** | **P4.0 Low Level Auto Return To High Delay Timing**  00 = Auto return to high function is disabled.  01 = 1 LIRC clock < auto return delay timing < 2 LIRC clock.  10 = 2 LIRC clock < auto return delay timing < 3 LIRC clock.  11 = 3 LIRC clock < auto return delay timing < 4 LIRC clock. |
| [5:4] | **P30RTH[1:0]** | **P3.0 Low Level To High Delay Timing**  00 = Auto return to high function is disabled.  01 = 1 LIRC clock < auto return delay timing < 2 LIRC clock.  10 = 2 LIRC clock < auto return delay timing < 3 LIRC clock.  11 = 3 LIRC clock < auto return delay timing < 4 LIRC clock. |
| [3:2] | **P21RTH[1:0]** | **P2.1 Low Level To High Delay Timing**  00 = Auto return to high function is disabled.  01 = 1 LIRC clock < auto return delay timing < 2 LIRC clock.  10 = 2 LIRC clock < auto return delay timing < 3 LIRC clock.  11 = 3 LIRC clock < auto return delay timing < 4 LIRC clock. |
| [1:0] | **P20RTH[1:0]** | **P2.0 Low Level To High Delay Timing**  00 = Auto return to high function is disabled.  01 = 1 LIRC clock < auto return delay timing < 2 LIRC clock.  10 = 2 LIRC clock < auto return delay timing < 3 LIRC clock.  11 = 3 LIRC clock < auto return delay timing < 4 LIRC clock. |

#### Internal Inverter and Buffer Control

MUG51TBAE GPIO embedded inverter and buffer function. Following shows the internal Inverter and buffer MFP control bit.

|  |
| --- |
|  |

Figure 6.4‑6 GPIO Internal Inverter And Buffer Structure

|  |
| --- |
|  |

Figure 6.4‑7 GPIO Internal Invert Active Application

|  |
| --- |
|  |

Figure 6.4‑8 GPIO Internal Buffer Application Example

### External Interrupt Pins

The external interrupt INT0 and INT1 can be used as interrupt sources. They are selectable to be either edge or level triggered depending on bits IT0 (TCON.0) and IT1 (TCON.2). The bits IE0 (TCON.1) and IE1 (TCON.3) are the flags those are checked to generate the interrupt. In the edge triggered mode, the INT0 or INT1 inputs are sampled every system clock cycle. If the sample is high in one cycle and low in the next, then a high to low transition is detected and the interrupts request flag IE0 or IE1 will be set. Since the external interrupts are sampled every system clock, they have to be held high or low for at least one system clock cycle. The IE0 and IE1 are automatically cleared when the interrupt service routine is called. If the level triggered mode is selected, then the requesting source has to hold the pin low till the interrupt is serviced. The IE0 and IE1 will not be cleared by the hardware on entering the service routine. In the level triggered mode, IE0 and IE1 follows the inverse value of INT0 and INT1 pins. If interrupt pins continue to be held low even after the service routine is completed, the processor will acknowledge another interrupt request from the same source. Both INT0 and INT1 can wake up the device from the Power-down mode.

##### TCON – Timer 0 and 1 Control

|  |  |  |
| --- | --- | --- |
| **Part of Register Content** | **SFR Address** | **Reset Value** |
| TCON | 88H, All Pages, Bit-addressable | 0000\_0000b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [3] | **IE1** | **External Interrupt 1 Edge Flag**  If IT1 = 1 (falling edge trigger), this flag will be set by hardware when a falling edge is detected. It remain set until cleared via software or cleared by hardware in the beginning of its interrupt service routine.  If IT1 = 0 (low level trigger), this flag follows the inverse of the INT1 input signal’s logic level. Software cannot control it. |
| [2] | **IT1** | **External Interrupt 1 Type Select**  This bit selects by which type that INT1 is triggered.  0 = INT1 is low level triggered.  1 = INT1 is falling edge triggered. |
| [1] | **IE0** | **External Interrupt 0 Edge Flag**  If IT0 = 1 (falling edge trigger), this flag will be set by hardware when a falling edge is detected. It remain set until cleared via software or cleared by hardware in the beginning of its interrupt service routine.  If IT0 = 0 (low level trigger), this flag follows the inverse of the INT0 input signal’s logic level. Software cannot control it. |
| [0] | **IT0** | **External Interrupt 0 Type Select**  This bit selects by which type that INT0 is triggered.  0 = INT0 is low level triggered.  1 = INT0 is falling edge triggered. |

### Pin Interrupt (PIT)

The MUG51TBAE provides pin interrupt input for each I/O pin to detect pin state if button or keypad set is used. A maximum 8-channel pin interrupt detection can be assigned by I/O port sharing. The pin interrupt is generated when any key is pressed on a keyboard or keypad, which produces an edge or level triggering event. Pin interrupt may be used to wake the CPU up from Idle or Power-down mode.

Each channel of pin interrupt can be enabled and polarity controlled independently by PIPEN and PINEN register. PICON selects which port that the pin interrupt is active. It also defines which type of pin interrupt is used – level detect or edge detect. Each channel also has its own interrupt flag. There are total eight pin interrupt flags located in PIF register. The respective flags for each pin interrupt channel allow the interrupt service routine to poll on which channel on which the interrupt event occurs. All flags in PIF register are set by hardware and should be cleared by software.

|  |
| --- |
|  |

Figure 6.4‑9 Pin Interface Block Diagram

Pin interrupt is generally used to detect an edge transient from peripheral devices like keyboard or keypad. During idle state, the system prefers to enter Power-down mode to minimize power consumption and waits for event trigger. Pin interrupt can wake up the device from Power-down mode.

##### PICON – Pin Interrupt Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PICON | E9H, Page 1 | 0011 \_0100 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **PITn[7:0]** | **Pin Interrupt Channel nType Select**  This bit selects which type that pin interrupt channel 7 is triggered.  0 = Level triggered.  1 = Edge triggered. |

##### PINEN – Pin Interrupt Negative Polarity Enable

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PINEN | EAH, Page 1 | 0000\_0000 b |

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

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

##### PIPEN – Pin Interrupt Positive Polarity Enable

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PIPEN | EBH, Page 1 | 0000\_0000 b |

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

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

##### PIF – Pin Interrupt Flags

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PIF | CAH, Page 0 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| R (level)  R/W (edge) | R (level)  R/W (edge) | R (level)  R/W (edge) | R (level)  R/W (edge) | R (level)  R/W (edge) | R (level)  R/W (edge) | R (level)  R/W (edge) | R (level)  R/W (edge) |

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

##### PIPSn – Pin Interrupt Control

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

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

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

## Timer

### Overview

MUG51TBAE provides following 16-bit Timer. Two 16-bit Timers/Counters 0 and 1 compatible with standard 8051. One 16-bit Timer 2 with three-channel input capture module and 9 input pin can be selected. One 16-bit auto-reload Timer 3, which can be the baud rate clock source of UARTs.

### Features

* Two 16-bit Timers/Counters 0 and 1 compatible with standard 8051.
* One 16-bit Timer 2 with three-channel input capture module and 9 input pin can be selected.
* One 16-bit auto-reload Timer 3, which can be the baud rate clock source of UARTs.

### Timer/Counter 0 and 1

Timer/Counter 0 and 1 on MUG51TBAE 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/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 (FSYS) for standard 8051 capability or direct the system clock for enhancement, which is selected by T0M (CKCON.3) bit for Timer 0 and T1M (CKCON.4) bit for Timer 1. In the “Counter” mode, the countering register increases on the falling edge of the external input pin T0. If the sampled value is high in one clock cycle and low in the next, a valid 1-to-0 transition is recognized on T0 or T1 pin.

The Timers 0 and 1 can be configured to automatically to toggle output whenever a timer overflow occurs. The same device pins that are used for the T0 and T1 count inputs are also used for the timer toggle outputs. This function is enabled by control bits T0OE and T1OE in the CKCON register, and apply to Timer 0 and Timer 1 respectively. The port outputs will be logic 1 prior to the first timer overflow when this mode is turned on. In order for this mode to function, the bit should be cleared selecting the system clock as the clock source for the timer.

Note that the TH0 (TH1) and TL0 (TL1) are accessed separately. It is strongly recommended that in mode 0 or 1, user should stop Timer temporally by clearing TR0 (TR1) bit before reading from or writing to TH0 (TH1) and TL0 (TL1). The free-running reading or writing may cause unpredictable result.

* + - * 1. Mode 0 (13-Bit Timer)

In Mode 0, the Timer/Counter is a 13-bit counter. The 13-bit counter consists of TH0 (TH1) and the five lower bits of TL0 (TL1). The upper three bits of TL0 (TL1) are ignored. The Timer/Counter is enabled when TR0 (TR1) is set and either GATE is 0 or INT0 (INT1) is 1. Gate setting as 1 allows the Timer to calculate the pulse width on external input pin INT0 (INT1). When the 13-bit value moves from 1FFFH to 0000H, the Timer overflow flag TF0 (TF1) is set and an interrupt occurs if enabled.

|  |
| --- |
|  |

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

* + - * 1. 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

* + - * 1. Mode 2 (8-Bit Auto-Reload Timer)

In Mode 2, the Timer/Counter is in auto-reload mode. In this mode, TL0 (TL1) acts as an 8-bit count register whereas TH0 (TH1) holds the reload value. When the TL0 (TL1) register overflow, the TF0 (TF1) bit in TCON is set and TL0 (TL1) is reloaded with the contents of TH0 (TH1) and the counting process continues from here. The reload operation leaves the contents of the TH0 (TH1) register unchanged. This feature is best suitable for UART baud rate generator for it runs without continuous software intervention. Note that only Timer1 can be the baud rate source for UART. Counting is enabled by setting the TR0 (TR1) bit as 1 and proper setting of GATE and INT0 (INT1) pins. The functions of GATE and INT0 (INT1) pins are just the same as Mode 0 and 1.

|  |
| --- |
|  |

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

* + - * 1. Mode 3 (Two Separate 8-Bit Timers)

Mode 3 has different operating methods for Timer 0 and Timer 1. For Timer/Counter 1, Mode 3 simply freezes the counter. Timer/Counter 0, however, configures TL0 and TH0 as two separate 8 bit count registers in this mode. TL0 uses the Timer/Counter 0 control bits C/T, GATE, TR0, INT0, and TF0. The TL0 also can be used as a 1-to-0 transition counter on pin T0 as determined by C/T(TMOD.2). TH0 is forced as a clock cycle counter and takes over the usage of TR1 and TF1 from Timer/Counter 1. Mode 3 is used in case that an extra 8 bit timer is needed. If Timer/Counter 0 is configured in Mode 3, Timer/Counter 1 can be turned on or off by switching it out of or into its own Mode 3. It can still be used in Modes 0, 1 and 2 although its flexibility is restricted. It no longer has control over its overflow flag TF1 and the enable bit TR1. However Timer 1 can still be used as a Timer/Counter and retains the use of GATE, INT1 pin and T1M. It can be used as a baud rate generator for the serial port or other application not requiring an interrupt.

|  |
| --- |
|  |

Figure 6.5‑4 Timer/Counter 0 in Mode 3

#### Register Description

##### TMOD – Timer 0 and 1 Mode

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| TMOD | 89H, All Pages | 0000\_0000b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **GATE\_T1** | **CT\_T1** | **M1\_T1** | **M0\_T1** | **GATE\_T0** | **CT\_T0** | **M1\_T0** | **M0\_T0** |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **GATE\_T1** | **Timer 1 Gate Control**  0 = Timer 1 will clock when TR1 is 1 regardless of INT1 logic level.  1 = Timer 1 will clock only when TR1 is 1 and INT1 is logic 1. |
| [6] | **CT\_T1** | **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\_T1** | **Timer 1 Mode Select**   |  |  |  | | --- | --- | --- | | M1 | M0 | Timer 1 Mode | | 0 | 0 | Mode 0: 13-bit Timer/Counter | | 0 | 1 | Mode 1: 16-bit Timer/Counter | | 1 | 0 | Mode 2: 8-bit Timer/Counter with auto-reload from TH1 | | 1 | 1 | Mode 3: Timer 1 halted | |
| [4] | **M0\_T1** | Check with bit 5 description. |
| [3] | **GATE\_T0** | **Timer 0 Gate Control**  0 = Timer 0 will clock when TR0 is 1 regardless of INT0 logic level.  1 = Timer 0 will clock only when TR0 is 1 and INT0 is logic 1. |
| [2] | **CT\_T0** | **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\_T0** | **Timer 0 Mode Select**   |  |  |  | | --- | --- | --- | | M1 | M0 | Timer 0 Mode | | 0 | 0 | Mode 0: 13-bit Timer/Counter | | 0 | 1 | Mode 1: 16-bit Timer/Counter | | 1 | 0 | Mode 2: 8-bit Timer/Counter with auto-reload from TH0 | | 1 | 1 | Mode 3: TL0 as a 8-bit Timer/Counter and TH0 as a 8-bit Timer | |
| [0] | **M0\_T0** | Check with bit 1 description |

##### TCON – Timer 0 and 1 Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| TCON | 88H, All Pages, Bit-addressable | 0000\_0000b |

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

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

##### TL0 – Timer 0 Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| TL0 | 8AH, Page 0 | 0000\_0000b |

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

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

##### TH0 – Timer 0 High Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| TH0 | 8CH, Page 0 | 0000\_0000b |

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

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

##### TL1 – Timer 1 Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| TL1 | 8BH, Page 0 | 0000\_0000b |

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

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

##### TH1 – Timer 1 High Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| TH1 | 8DH, Page 0 | 0000\_0000b |

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

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

##### CKCON – Clock Control

|  |  |  |
| --- | --- | --- |
| **Part of Register Content** | **SFR Address** | **Reset Value** |
| CKCON | 8EH, Page 0 | 1000\_0000b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [5] | **T1OE** | **Timer 1 Output Enable**  0 = Timer 1 output Disabled.  1 = Timer 1 output Enabled from T1 pin.  Note that Timer 1 output should be enabled only when operating in its “Timer” mode. |
| [4] | **T1M** | **Timer 1 Clock Mode Select**  0 = The clock source of Timer 1 is the system clock divided by 12. It maintains standard 8051 compatibility.  1 = The clock source of Timer 1 is direct the system clock. |
| [3] | **T0M** | **Timer 0 Clock Mode Select**  0 = The clock source of Timer 0 is the system clock divided by 12. It maintains standard 8051 compatibility.  1 = The clock source of Timer 0 is direct the system clock. |
| [2] | **T0OE** | **Timer 0 Output Enable**  0 = Timer 0 output Disabled.  1 = Timer 0 output Enabled from T0 pin.  Note that Timer 0 output should be enabled only when operating in its “Timer” mode. |

### Timer 2 and Input Capture

#### 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 (T2CON.0). Timer 2 with two clock select from system clock or 3 GPIO input as source. 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.

#### Functional Description

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

* + - * 1. Auto-Reload

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

* + - * 1. Compare Mode

Timer 2 can also be configured as the compare mode by setting . 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 6.5‑5 Timer 2 Compare Mode Functional Block Diagram

* + - * 1. Input Capture Mode

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.

|  |
| --- |
|  |

Figure 6.5‑6 Timer 2 Input Capture Module Functional Block Diagram

#### Register Description

##### 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** | **-** | **-** | **TH2RF** | **TL2RF** | **TR2** | **-** | **CM\_RL2** |
| R/W | - | - | R | R | R/W | - | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **TF2** | **Timer 2 Overflow Flag**  This bit is set when Timer 2 overflows or a compare match occurs. If the Timer 2 interrupt and the global interrupt are enable, setting this bit will make CPU execute Timer 2 interrupt service routine. This bit is not automatically cleared via hardware and should be cleared via software. |
| [6:5] | **-** | Reserved |
| [4] | **TH2RF** | **TH2 Reading Status Flag**  When TH2 is written, software should check this bit first.  0 = TH2 reading is available.  1 = TH2 reading is not available. |
| [3] | **TL2RF** | **TL2 Reading Status Flag**  When TL2 is written, software should check this bit first.  0 = TL2 reading is available.  1 = TL2 reading is not available. |
| [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. |

##### TH2 – Timer 2 High Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| TH2 | CDH, Page 0 | 0000\_0000 b |

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

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

##### TL2 – Timer 2 Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| TL2 | CCH, Page 0 | 0000\_0000 b |

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

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

##### RCMP2H – Timer 2 Reload/Compare High Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| RCMP2H | CBH, Page 1 | 0000\_0000 b |

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

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

##### RCMP2L– Timer 2 Reload/Compare Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| RCMP2L | CAH, Page 1 | 0000\_0000 b |

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

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

##### CAPCON0 – Input Capture Control 0

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| CAPCON0 | E1H, Page 1 | 0000\_0000b |

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

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

##### CAPCON1 – Input Capture Control 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| CAPCON1 | E2H, Page 1 | 0000\_0000b |

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

Address: E2H, Page 1 Reset value: 0000 0000b

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

##### CAPCON2 – Input Capture Control 2

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| CAPCON2 | E3H, Page 1 | 0000\_0000b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [6] | **ENF2** | **Enable Noise Filer on Input Capture 2**  0 = Noise filter on input capture channel 2 Disabled.  1 = Noise filter on input capture channel 2 Enabled. |
| [5] | **ENF1** | **Enable Noise Filer on Input Capture 1**  0 = Noise filter on input capture channel 1 Disabled.  1 = Noise filter on input capture channel 1 Enabled. |
| [4] | **ENF0** | **Enable Noise Filer on Input Capture 0**  0 = Noise filter on input capture channel 0 Disabled.  1 = Noise filter on input capture channel 0 Enabled. |
| [3:1] | **-** | Reserved |
| [0] | **CMOD** | **Capture Into Interrupt Mode Select Bit**  0 = Enter capture interrupt vector when each trig condition is met.  1 = The first time trig condition met not enter the interrupt vector. Then after the second trigger condition occurs, the capture interrupt is entered every subsequent time. |

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

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

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

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

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

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

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

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

##### T2ACMP – TIMER2 and ACMP Output Connection

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| T2ACMP | 8EH, Page 3 | 0000 \_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **T2CKS1** | **T2CKS0** | **IC2S1** | **IC2S0** | **IC1S1** | **IC1S0** | **IC0S1** | **IC0S0** |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **T2CKS[1:0]** | **Timer2 Clock Source Select**  00 = Timer 2 Clock source from FSYS.  01 = Timer 2 Clock source from external P4.0.  10 = Timer 2 Clock source from external pin P5.4.  11 = Timer 2 Clock source from external pin P3.2. |
| [5:4] | **IC2S[1:0]** | **Input Capture 2 Source Select**  00 = IC2.  01 = ACMP\_O.  10 = ACMP1\_O.  11 = Reserved. |
| [3:2] | **IC1S[1:0]** | **Input Capture 1 Source Select**  00 = external IC1 source connects to internal IC1.  01 = ACMP0\_O.  10 = ACMP1\_O.  11 = external IC1 source connects to internal IC1. |
| [1:0] | **IC0S[1:0]** | **Input Capture 0 Source Select**  00 = IC0.  01 = ACMP0\_O.  10 = ACMP1\_O.  11 = IC0. |

### Timer 3

#### 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 6.9.3.2“Baud Rate”.

#### Block Diagram

|  |
| --- |
|  |

Figure 6.5‑7 Timer 3 Block Diagram

#### Register Description

##### T3CON – Timer 3 Control

|  |  |  |
| --- | --- | --- |
| **Part of Register Content** | **SFR Address** | **Reset Value** |
| T3CON | C4H, Page 0 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **SMOD\_1** | **SMOD0\_1** | **BRCK** | **TF3** | **TR3** | **T3PS[2:0]** | | |
| R/W | R/W | R/W | R/W | R/W | R/W | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [4] | **TF3** | **Timer 3 Overflow Flag**  This bit is set when Timer 3 overflows. It is automatically cleared by hardware when the program executes the Timer 3 interrupt service routine. This bit can be set or cleared by software. |
| [3] | **TR3** | **Timer 3 Run Control**  0 = Timer 3 is halted.  1 = Timer 3 starts running.  Note that the reload registers RH3 and RL3 can only be written when Timer 3 is halted (TR3 bit is 0). If any of RH3 or RL3 is written if TR3 is 1, result is unpredictable. |
| [2:0] | **T3PS[2:0]** | **Timer 3 Pre-Scalar**  These bits determine the scale of the clock divider for Timer 3.  000 = 1/1.  001 = 1/2.  010 = 1/4.  011 = 1/8.  100 = 1/16.  101 = 1/32.  110 = 1/64.  111 = 1/128. |

##### RL3 – Timer 3 Reload Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| RL3 | C5H, Page 0 | 0000\_0000 b |

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

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

##### RH3 – Timer 3 Reload High Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| RH3 | C6H, Page 0 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **RH3[7:0]** | | | | | | | |
| R/W | | | | | | | |

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

## Watchdog Timer (WDT)

The MUG51TBAE 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 , where FLIRC is the frequency of internal 38.4 kHz oscillator. The following table shows an example of the Watchdog time-out interval with different pre-scales.

| **WDPS.3** | **WDPS.2** | **WDPS.1** | **WDPS.0** | **Clock Divider Scale** | **WDT Time-Out Timing[1]** |
| --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 1/1 | 1.66 ms |
| 0 | 0 | 0 | 1 | 1/4 | 6.64 ms |
| 0 | 0 | 1 | 0 | 1/8 | 13.31 ms |
| 0 | 0 | 1 | 1 | 1/16 | 26.62 ms |
| 0 | 1 | 0 | 0 | 1/32 | 53.25 ms |
| 0 | 1 | 0 | 1 | 1/64 | 106.66 ms |
| 0 | 1 | 1 | 0 | 1/128 | 213.12 ms |
| 0 | 1 | 1 | 1 | 1/256 | 426.64 ms |
| 1 | 0 | 0 | 0 | 1/512 | 853.28ms |
| 1 | 0 | 0 | 1 | 1/1024 | 1706.56ms |
| 1 | 0 | 1 | 0 | 1/2048 | 3413.12ms |
| Others | | | | 1/2048 | 3413.12ms |
| **Note:** This is an approximate value since the deviation of LIRC. | | | | | |

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

Since the limitation of the maxima vaule of WDT timer delay. To wake up MUG51TBAE from idle mode or Power-down mode suggest use WKT function seeChapter 6.7 **.**

The WDT is implemented with a set of divider that divides the low-speed internal oscillator clock nominal 38.4 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.

### 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.

### 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.

### Register Description

##### CONFIG4

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

Factory default value: 1111 1111b

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

##### WDCON – Watchdog Timer Control

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

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **WDTR** | **WDT Run**  This bit is valid only when control bits in WDTEN[3:0] (CONFIG4[7:4]) are all 1. At this time, WDT works as a general purpose timer.  0 = WDT Disabled.  1 = WDT Enabled. The WDT counter starts running. |
| [6] | **WDCLR** | **WDT Clear**  Setting this bit will reset the WDT count to 00H. It puts the counter in a known state and prohibit the system from unpredictable reset. The meaning of writing and reading WDCLR bit is different.  Writing:  0 = No effect.  1 = Clearing WDT counter.  Reading:  0 = WDT counter is completely cleared.  1 = WDT counter is not yet cleared. |
| [5] | **WDTF** | **WDT Time-Out Flag**  This bit indicates an overflow of WDT counter. This flag should be cleared by software. |
| [4] | **WIDPD** | **WDT Running in Idle or Power-Down Mode**  This bit is valid only when control bits in WDTEN[3:0] (CONFIG4[7:4]) are all 1. It decides whether WDT runs in Idle or Power-down mode when WDT works as a general purpose timer.  0 = WDT stops running during Idle or Power-down mode.  1 = WDT keeps running during Idle or Power-down mode. |
| [3] | **WDTRF** | **WDT Reset Flag**  When the CPU is reset by WDT time-out event, this bit will be set via hardware. This flag is recommended to be cleared via software after reset. |
| [2:0] | **WDPS[2:0]** | **WDT Clock Pre-Scalar Select**  These bits determine the pre-scale of WDT clock from 1/1 through 1/256. SeeTable 6.6‑1 Watchdog Timer-out Interval Under Different Pre-scalars. 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[3:0] are all set after power-on reset and keep unchanged after any reset other than power-on reset. | | |

##### WDCON1 – Watchdog Timer Control Register 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| WDCON1 | AAH, Page 0, TA protected | POR 0000\_0001 b  WDT 0000\_000U b  Others 0000\_000U b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | **-** | **-** | **-** | **-** | **-** | **-** | **WDPS.3** |
|  |  |  |  |  |  |  | R/W |

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

### Typical Structure of WDT Service Routine

In some application of low power consumption, the CPU usually stays in Idle mode when nothing needs to be served to save power consumption. After a while the CPU will be woken up to check if anything needs to be served at an interval of programmed period implemented by Timer 0~3. However, the current consumption of Idle mode still keeps at a "mA” level. To further reducing the current consumption to “uA” level, the CPU should stay in Power-down mode when nothing needs to be served, and has the ability of waking up at a programmable interval. The MUG51TBAE is equipped with this useful function by WDT waking up. It provides a very low power internal oscillator 38.4 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

## Self Wake-up Timer (WKT)

### Overview

The MUG51TBAE has a dedicated Self Wake-up Timer (WKT), which serves for a periodic wake-up timer in low power mode or for general purpose timer. WKT remains counting in Idle or Power-down mode. When WKT is being used as a wake-up timer, a start of WKT can occur just prior to entering a power management mode. WKT has two clock source, internal LIRC 38.4 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 16-bit auto-reload, up-counting timer with pre-scale 1/1 to 1/2048 selected by WKPS[2:0] (WKCON[2:0]). User fills the reload value into RWK register to determine its overflow rate. The RWK can reloadable when counter is count to overflow. The CWK can read current count value. The WKTR (WKCON.3) can be set to start counting. When the counter rolls over FFH, WKTF (WKCON.4) is set as 1 and a reload is generated and causes the contents of the RWK register to be reloaded into the internal 8-bit counter. If EWKT (EIE1.2) is set as 1, WKT interrupt service routine will be served.

### Block Diagram

|  |
| --- |
|  |

Figure 6.7‑1 Self Wake-Up Timer (WKT) Block Diagram

### Register Description

##### WKCON – Self Wake-up Timer Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| WKCON | 8FH, Page 0 | 0000\_0000b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | | **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. |

##### RWKH – Self Wake-up Timer Reload High Byte

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

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

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

##### RWKL – Self Wake-up Timer Reload Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| RWKL | 86H, Page 0 | 0000 0000b |

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

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

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

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| CWKH | BEH, Page 2 | 0000 0000b |

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

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

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

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| CWKL | 86H, Page 1 | 0000 0000b |

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

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

## Pulse Width Modulated (PWM)

### Overview

The PWM (Pulse Width Modulation) signal is a useful control solution in wide application field. It can used on motor driving, fan control, backlight brightness tuning, LED light dimming, or simulating as a simple digital to analog converter output through a low pass filter circuit.

The MUG51TBAE 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.

### Features

* Up To 6 output pins can be selected
* Supports maximum clock source frequency up to FSYS
* Supports independent mode for PWM output
* Supports complementary mode for 3 complementary paired PWM output channels
* Dead-time insertion with 8-bit resolution
* Supports 16-bit resolution PWM counter
* Supports mask function and tri-state enable for each PWM pin
* Supports brake function

### Block Diagram

#### PWM0 Block Diagram

|  |
| --- |
|  |

Figure 6.8‑1 PWM0 Block Diagram

### Functional Description

#### PWM Generator

The PWM generator is clocked by the system clock or Timer 1 overflow divided by a PWM clock pre-scalar selectable from 1/1~1/128. The PWM0period is defined by effective 16-bit period registers, {PWMnPH, PWMnPL}. The period is the same for all PWM0 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. These PWMs output can be generated independently with different duty cycles. The interval and duty of PWM0 signal is generated by a 16-bit counter comparing with the period and duty registers.

Only PWM0 has group mode to facilitate the three-phase motor control, a group mode can be used by setting GP (PWM0CON1.5), which makes {PWM0C0H, PWM0C0L} and {PWM0C1H, PWM0C1L} duty register decide duties of the PWM outputs. In a three-phase motor control application, two-group PWM outputs generally are given the same duty cycle. When the group mode is enabled, {PWM0C2H, PWM0C2L}, {PWM0C3H, PWM0C3L}, {PWM0C4H, PWM0C4L} and {PWM0C5H, PWM0C5L} registers have no effect. This mean is {PWM0C2H, PWM0C2L} and {PWM0C4H, PWM0C4L} both as same as {PWM0C0H, PWM0C0L}. Also {PWM0C3H, PWM0C3L} and {PWM0C5H, PWM0C5L} are same as {PWM0C1H, PWM0C1L}.Note that enabling PWM0 does not configure the I/O pins into their output mode automatically. User should configure I/O output mode via software manually.

The PWM0 counter generates six PWM0 signals called P0G0, P0G1, P0G2, P0G3, P0G4, and P0G5. These signals will go through the PWM0 and Fault Brake output control circuit. It generates real PWM0 outputs on I/O pins. The output control circuit determines the PWM mode, dead-time insertion, mask output, Fault Brake control, and PWM polarity. The last stage is a multiplexer of PWM0 output or I/O function.

|  |
| --- |
|  |

Figure 6.8‑2 PWM0 and Fault Brake Output Control Block Diagram

User should follow the initialization steps below to start generating the PWM signal output. In the first step by setting CLRPWM (PWMnCON0.4), it ensures the 16-bit up counter reset for the accuracy of the first duration. After initialization and setting {PWMnPH, PWMnPL} and all {PWMnH, PWMnL} registers, PWMRUN (PWMnCON0.7) can be set as logic 1 to trigger the 16-bit counter running. PWM starts to generate waveform on its output pins. The hardware for all period and duty Register Description are double buffered designed. Therefore, {PWMnPH, PWMnPL} and all {PWMnH, PWMnL} registers can be written to at any time, but the period and duty cycle of PWM will not be updated immediately until the LOAD (PWMnCON0.6) is set and previous period is complete. This prevents glitches when updating the PWM period or duty.

Note: A loading of new period and duty by setting LOAD should be ensured complete by monitoring it and waiting for a hardware automatic clearing LOAD bit. Any updating of PWM Register Description during LOAD bit as logic 1 will cause unpredictable output

#### PWM Types

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

* + - * 1. Edge-Aligned Type

In edge-aligned mode, the 16-bit counter uses single slop operation by counting up from 0000H to {PWMnPH, PWMnPL} and then starting from 0000H. The PWM generator signal (PGn before PWM and Fault Brake output control) is cleared on the compare match of 16-bit counter and the duty register {PWMnH, PWMnL} and set at the 16-bit counter is 0000H. The result PWM output waveform is left-edge aligned.

|  |
| --- |
|  |

Figure 6.8‑3 PWM Edge-aligned Type Waveform

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

PWM frequency =  (FPWM is the PWM clock source frequency divided by PWMDIV).

PWM high level duty = .

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

|  |
| --- |
|  |

Figure 6.8‑4 PWM Center-aligned Type Waveform

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

PWM frequency =  (FPWM is the PWM clock source frequency divided by PWMDIV).

PWM high level duty = .

#### Operation Modes

After PGn signals pass through the first stage of the PWM and Fault Brake output control circuit. The PWM mode selection circuit generates different kind of PWM output modes with six-channel, three-pair signal PG0~PG5 . It supports independent mode, complementary mode, and synchronous mode.

* + - * 1. Independent Mode

Independent mode is enabled when PWMMOD[1:0] (PWMnCON1[7:6]) is [0:0]. It is the default mode of PWM. PG0, PG1, PG2, PG3, PG4 and PG5 output PWM signals independently.

* + - * 1. Complementary Mode with Dead-Time Insertion

Complementary mode is enabled when PWMMOD[1:0] = [0:1]. In this mode, PG0/2/4 output PWM signals the same as the independent mode. However, PG1/3/5 output the out-phase PWM signals of PG0/2/4 correspondingly, and ignore PG1/3/5 Duty register {PWMnH, PWMnL} (n:1/3/5). This mode makes PG0/PG1 a PWM complementary pair and so on PG2/PG3 and PG4/PG5.

In a real motor application, a complementary PWM output always has a need of “dead-time” insertion to prevent damage of the power switching device like GPIBs due to being active on simultaneously of the upper and lower switches of the half bridge, even in a “μs” duration. For a power switch device physically cannot switch on/off instantly. For the MUG51TBAE PWM, each PWM pair share a 9-bit dead-time down-counter PWM0DTCNT used to produce the off time between two PWM signals in the same pair. On implementation, a 0-to-1 signal edge delays after PWM0DTCNT timer underflows. The timing diagram illustrates the complementary mode with dead-time insertion of PG0/PG1 pair. Pairs of PG2/PG3 and PG4/PG5 have the same dead-time circuit. Each pair has its own dead-time enabling bit in the field of PWMnDTEN [3:0].

**Note** that the PWM0DTCNT and PWMnDTEN registers are all TA write protection. The dead-time control are also valid only when the PWM is configured in its complementary mode.

|  |
| --- |
|  |

Figure 6.8‑5 PWM Complementary Mode with Dead-time Insertion

* + - * 1. Synchronous Mode

Synchronous mode is enabled when PWMMOD[1:0] = [1:0]. In this mode, PG0/2/4 output PWM signals the same as the independent mode. PG1/3/5 output just the same in-phase PWM signals of PG02/4 correspondingly.

#### Mask Output Control

Each PWM signal can be software masked by driving a specified level of PWM signal. The PWM mask output function is quite useful when controlling Electrical Commutation Motor like a BLDC. PWMnMEN contains six bits, those determine which channel of PWM signal will be masked. PWMnMD set the individual mask level of each PWM channel. The default value of PWMnMEN is 00H, which makes all outputs of PWM channels follow signals from PWM generator. Note that the masked level is reversed or not by PWM0NP setting on PWM output pins.

#### Fault Brake

The Fault Brake function is usually implemented in conjunction with an enhanced PWM circuit. It rules as a fault detection input to protect the motor system from damage. Fault Brake pin input (FB) is valid when FBINEN (PWMnCON1.3) is set. When Fault Brake is asserted PWM signals will be individually overwritten by PWMnFBD corresponding bits. PWMRUN (PWMnCON0.7) will also be automatically cleared by hardware to stop PWM generating. The PWM 16-bit counter will also be reset as 0000H. A indicating flag FBF will be set by hardware to assert a Fault Brake interrupt if enabled. PWMnFBD data output remains even after the FBF is cleared by software. User should resume the PWM output only by setting PWMRUN again. Meanwhile the Fault Brake state will be released and PWM waveform outputs on pins as usual. Fault Brake input has a polarity selection by FBINLS (PWMnFBD.6) bit. Note that the Fault Brake signal feed in FB pin should be longer than eight-system-clock time for FB pin input has a permanent 8/FSYS 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.

|  |
| --- |
|  |

Figure 6.8‑6 Fault Brake Function Block Diagram

#### Polarity Control

Each PWM output channel has its independent polarity control bit, PNP0~PNP5. The default is high active level on all control fields implemented with positive logic. It means the power switch is ON when PWM outputs high level and OFF when low level. User can easily configure all setting with positive logic and then set PWMnNP bit to make PWM actually outputs according to the negative logic.

#### 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 6.8‑7.

|  |
| --- |
|  |

Figure 6.8‑7 PWM Interrupt Type

Fault Brake event requests another interrupt, Fault Brake interrupt. It has different interrupt vector from PWM interrupt. When either Fault Brake pin input event or ADC compare event occurs, FBF (PWMnFBD.7) will be set by hardware. It generates Fault Brake interrupt if enabled. The Fault Brake interrupt enable bit is EFB0 (EIE0.5). FBF Is cleared via software.

### Register Description

##### PWM0CON0 – PWM Control Register0

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0CON0 | D1H, Page 0 | 0000\_0000 b |

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

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

##### PWMnCON1 – PWM Control 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0CON1 | DFH, Page 1 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **PWMMOD[1:0]** | **PWM Mode Select**  00 = Independent mode.  01 = Complementary mode.  10 = Synchronized mode.  11 = Reserved. |
| [5] | **GP** | **Group Mode Enable**  This bit enables the group mode. If enabled, the duty of first three pairs of PWM are decided by PWM01H and PWM01L rather than their original duty Register Description.  0 = Group mode Disabled.  1 = Group mode Enabled. |
| [4] | **PWMTYP** | **PWM Type Select**  0 = Edge-aligned PWM.  1 = Center-aligned PWM. |
| [3] | **FBINEN** | **FB Pin Input Enable**  0 = PWM0 output Fault Braked by FB pin input Disabled.  1 = PWM0 output Fault Braked by FB pin input Enabled. Once an edge, which matches FBINLS (PWM0FBD.6) selection, occurs on FB pin, PWM0CH0~5 output Fault Brake data in PWMnFBD register. PWMRUN (PWM0CON0.7) will also be automatically cleared by hardware. The PWM output resumes when PWMRUN is set again.  **Note:** This bit is only vaild in PWM0 |
| [2:0] | **PWMDIV[2:0]** | **PWM Clock Divider**  This field decides the pre-scale of PWM clock source.  000 = 1/1.  001 = 1/2.  010 = 1/4.  011 = 1/8.  100 = 1/16.  101 = 1/32.  110 = 1/64.  111 = 1/128. |

##### PWMnPL – PWM Period Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0PL | D9H, Page 1 | 0000\_0000 b |

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

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

##### PWMnPH – PWM Period High Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0PH | D1H, Page 1 | 0000\_0000 b |

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

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

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

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0C0H | D2H, Page 1 | 0000\_0000 b |
| PWM0C1H | D3H, Page 1 | 0000\_0000 b |
| PWM0C2H | D4H, Page 1 | 0000\_0000 b |
| PWM0C3H | D5H, Page 1 | 0000\_0000 b |
| PWM0C4H | C4H, Page 1 | 0000\_0000 b |
| PWM0C5H | C5H, Page 1 | 0000\_0000 b |

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

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

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

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0C0L | DAH, Page 1 | 0000\_0000 b |
| PWM0C1L | DBH, Page 1 | 0000\_0000 b |
| PWM0C2L | DCH, Page 1 | 0000\_0000 b |
| PWM0C3L | DDH, Page 1 | 0000\_0000 b |
| PWM0C4L | CCH, Page 1 | 0000\_0000 b |
| PWM0C5L | CDH, Page 1 | 0000\_0000 b |

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

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

##### PWM0DTEN – PWM Dead-time Enable (TA Protected)

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0DTEN | F9H, Page 1, TA protected | 0000\_0000 b |

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

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

##### PWM0DTCNT – PWM Dead-time Counter (TA Protected)

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0DTCNT | FAH, Page 1, TA protected | 0000\_0000 b |

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

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

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

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0MEN | FBH, Page 1 | 0000\_0000 b |

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

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

##### PWMnMD – PWM Mask Data

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0MD | FCH, Page 1 | 0000\_0000 b |

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

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

##### PWMnFBD – PWM Fault Brake Data

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0FBD | D7H, Page 1 | 0000\_0000 b |

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

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

##### PWM0NP – PWM Negative Polarity

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0NP | D6H, Page 1 | 0000\_0000 b |

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

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

##### PWMnINTC – PWM Interrupt Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PWM0INTC | B7H, Page 1 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **-** | Reserved |
| [5:4] | **INTTYPn[1:0]** | **PWM Interrupt Type Select**  These bit select PWM interrupt type.  00 = Falling edge on PWMn\_CH0/1/2/3/4/5 pin.  01 = Rising edge on PWMn\_CH0/1/2/3/4/5 pin.  10 = Central point of a PWM period.  11 = End point of a PWM period.  Note that the central point interrupt or the end point interrupt is only available while PWM operates in center-aligned type. |
| [3] | **-** | Reserved |
| [2:0] | **INTSELn[2:0]** | **PWM Interrupt Pair Select**  These bits select which PWM channel asserts PWM interrupt when PWM interrupt type is selected as falling or rising edge on PWM0/1/2/3/4/5 pin..  000 = PWMn\_CH0.  001 = PWMn\_CH1.  010 = PWMn\_CH2.  011 = PWMn\_CH3.  100 = PWMn\_CH4.  101 = PWMn\_CH5.  Others = PWMn\_CH0. |

## Serial Port (UART0 & UART1)

### Overview

The MUG51TBAE 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.

### Features

* Supports up to 2 UARTs: UART0, UART1
* Supports 1 Smart Card configuration as UART function as UART2 List in Smart Card Interface Chapter
* UART baud rate clock from MIRC.
* Full-duplex asynchronous communications
* Programmable 9th bit.
* TXD and RXD pins of UART0/1 exchangeable via software.

### Functional Description

#### Operation Mode

* + - * 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 FSYS/12 if SM2 (SCON.5) is 0 or as FSYS/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.

|  |
| --- |
| 4T UART 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 received. Then the received flag RI will be set as 1. User can clear RI to triggering the next byte reception.

* + - * 1. Mode 1

Mode 1 supports asynchronous, full duplex serial communication. The asynchronous mode is commonly used for communication with PCs, modems or other similar interfaces. In Mode 1, 10 bits are transmitted through TXD or received through RXD including a start bit (logic 0), 8 data bits (LSB first) and a stop bit (logic 1). The baud rate is determined by the Timer 1. SMOD (PCON.7) setting 1 makes the baud rate double. Figure 6.9‑2 shows the associated timings of the serial port in Mode 1 for transmitting and receiving.

|  |
| --- |
| UART mode 1 |

Figure 6.9‑2 Serial Port Mode 1 Timing Diagram

Transmission is initiated by any writing instructions to SBUF. Transmission takes place on TXD pin. First the start bit comes out, the 8-bit data follows to be shifted out and then ends with a stop bit. After the stop bit appears, TI (SCON.1) will be set to indicate one byte transmission complete. All bits are shifted out depending on the rate determined by the baud rate generator.

Once the baud rate generator is activated and REN (SCON.4) is 1, the reception can begin at any time. Reception is initiated by a detected 1-to-0 transition at RXD. Data will be sampled and shifted in at the selected baud rate. In the midst of the stop bit, certain conditions should be met to load SBUF with the received data:

1. RI (SCON.0) = 0, and

2. Either SM2 (SCON.5) = 0, or the received stop bit = 1 while SM2 = 1 and the received data matches “Given” or “Broadcast” address. (For enhancement function, see Section 6.9.3.4“Multiprocessor Communication” and Section 6.9.3.5“Automatic Address Recognition”.)

If these conditions are met, then the SBUF will be loaded with the received data, the RB8 (SCON.2) with stop bit, and RI will be set. If these conditions fail, there will be no data loaded and RI will remain 0. After above receiving progress, the serial control will look forward another 1-to-0 transition on RXD pin to start next data reception.

* + - * 1. Mode 2

Mode 2 supports asynchronous, full duplex serial communication. Different from Mode1, there are 11 bits to be transmitted or received. They are a start bit (logic 0), 8 data bits (LSB first), a programmable 9th bit TB8 or RB8 bit and a stop bit (logic 1). The most common use of 9th bit is to put the parity bit in it or to label address or data frame for multiprocessor communication. The baud rate is fixed as 1/32 or 1/64 the system clock frequency depending on SMOD (PCON.7) bit. Figure 6.9‑3 shows the associated timings of the serial port in Mode 2 for transmitting and receiving.

|  |
| --- |
| 12T UART mode 2 and 3 FE |

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 6.9.3.4“Multiprocessor Communication” and Section 6.9.3.5“Automatic Address Recognition”.)

If these conditions are met, the SBUF will be loaded with the received data, the RB8(SCON.2) with the received 9th bit and RI will be set. If these conditions fail, there will be no data loaded and RI will remain 0. After above receiving progress, the serial control will look forward another 1-to-0 transition on RXD pin to start next data reception.

* + - * 1. 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.

#### 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 0 Mode / Baud Rate DescriptionThe user should calculate the baud rate according to their system configuration.

In Mode 1 or 3, the baud rate clock source of UART0 can be selected from Timer 1 or Timer 3. User can select the baud rate clock source by BRCK (T3CON.5). For UART1, its baud rate clock comes only from Timer 3 as its unique clock source.

When using Timer 1 as the baud rate clock source, note that the Timer 1 interrupt should be disabled. Timer 1 itself can be configured for either “Timer” or “Counter” operation. It can be in any of its three running modes. However, in the most typical applications, it is configured for “Timer” operation, in the auto-reload mode (Mode 2). If using Timer 3 as the baud rate generator, its interrupt should also be disabled.

Following shows all UART mode and baudrate fomula:

**Note:** That the maximum of the baud rate value is FSYS/32

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Mode** | **SM0 / SM1 (SCON[7:6])** | **SM2 (SCON[5])** | **SMOD (PCON[7])** | **Frame Bits** | **Baud Rate** | | |
| 0 | 00 | 0 | - | 8 | FSYS divided by 12 | | |
| 1 | FSYS divided by 2 | | |
| 1 | 01 | - | 0 | 10 | **Time1**  **TM1 (CKCON[3]) = 0** | |  |
| **Time1**  **TM1 (CKCON[3]) = 1** | |  |
| **Timer 3** |  | |
| 1 | **Time1**  **TM1 (CKCON[3]) = 0** | |  |
| **Time1**  **TM1 (CKCON[3]) = 1** | |  |
| **Timer** 3 |  | |
| 2 | 10 | - | 0 | 11 | FSYS divided by 64 | | |
| 1 | FSYS divided by 32 | | |
| 3 | 11 | - | 0 | 11 | **Time 1**  **TM1 (CKCON[3]) = 0** | |  |
| **Time 1**  **TM1 (CKCON[3]) = 1** | |  |
| **Timer 3** |  | |
| 1 | **Time1**  **TM1 (CKCON[3]) = 0** | |  |
| **Time1**  **TM1 (CKCON[3]) = 1** | |  |
| **Timer 3** |  | |
| **Note:** Timer 1 should configured as a timer in auto-reload mode (Mode 2). | | | | | | | |

Table 6.9‑1 Serial Port 0 Mode / Baud Rate Description

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ode | SM0\_1 / SM1\_1 (S1CON[7:6]) | SMOD\_1 (T3CON[7]) | Frame Bits | Baud Rate | |
| 0 | 00 | - | 8 | FSYS divided by 12 | |
| 1 | 01 | 0 | 10 | **Timer 3** |  |
| 1 | **Timer 3** |  |
| 2 | 10 | 0 | 11 | FSYS divided by 64 | |
| 1 | FSYS divided by 32 | |
| 3 | 11 | 0 | 11 | **Timer 3** |  |
| 1 | **Timer 3** |  |

Table 6.9‑2 Serial Port 1 Mode / Baud Rate Description

Sample code: we list the most popular UART setting Mode 1 initial step as following：

Serial port 0 (**UART0**) use **timer 1** as baudrate generator: Formula is 

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 

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) uses Timer 3 as baud rate generator: Fomula is 

SCON\_1 = 0x52; //UART1 Mode1,REN\_1=1,TI\_1=1

T3CON = 0xF8; //T3PS2=0,T3PS1=0,T3PS0=0(Prescale=1),

RH3 = *value high byte*

RL3 = *value low byte*

T3CON|= 0x08;

Following lists some popular baudrate value based on different Fsys and the deviation value:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fsys Value | Baud Rate | TH1 Value (Hex)  SMOD = 0 | RH3,RL3 Value (Hex) | Baudrate Deviation |
| 7372800 | 4800 | A0 | FFA0 | 0.000% |
| 9600 | D0 | FFD0 | 0.000% |
| 19200 | E8 | FFE8 | 0.000% |
| 38400 | F4 | FFF4 | 0.000% |
| 57600 | F8 | FFF8 | 0.000% |
| 64000 | F9 | FFF9 | 0.000% |
| 72000 | FA | FFFA | 0.000% |
| 96000 | FB | FFFB | 0.000% |
| 115200 | FC | FFFC | 0.000% |
| 144000 (Max) | FD | FFFD | 0.000% |

Table 6.9‑3 Timer Define Value for Baud Rate

#### 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.

#### Multiprocessor Communication

The MUG51TBAE 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:

Set all devices (masters and slaves) to UART Mode 2 or 3.

Write the SM2 bit of all the slave devices to 1.

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).

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.

After all data bytes have been received, set SM2 back to 1 to wait for next address.

SM2 has no effect in Mode 0, and in Mode 1 can be used to check the validity of the stop bit. For Mode 1 reception, if SM2 is 1, the receiving interrupt will not be issue unless a valid stop bit is received.

#### 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.

* + - * 1. Example 1, slave 0:

SADDR = 11000000b

SADEN = 11111101b

Given = 110000X0b

* + - * 1. Example 2, slave 1:

SADDR = 11000000b

SADEN = 11111110b

Given = 1100000Xb

In the above example SADDR is the same and the SADEN data is used to differentiate between the two slaves. Slave 0 requires 0 in bit 0 and it ignores bit 1. Slave 1 requires 0 in bit 1 and bit 0 is ignored. A unique address for Slave 0 would be 1100 0010 since slave 1 requires 0 in bit 1. A unique address for slave 1 would be 11000001b since 1 in bit 0 will exclude slave 0. Both slaves can be selected at the same time by an address, which has bit 0 = 0 (for slave 0) and bit 1 = 0 (for slave 1). Thus, both could be addressed with 11000000b as their “Broadcast” address.

In a more complex system, the following could be used to select slaves 1 and 2 while excluding slave 0:

* + - * 1. Example 1, slave 0:

SADDR = 11000000b

SADEN = 11111001b

Given = 11000XX0b

* + - * 1. Example 2, slave 1:

SADDR = 11100000b

SADEN = 11111010b

Given = 11100X0Xb

* + - * 1. Example 3, slave 2:

SADDR = 11000000b

SADEN = 11111100b

Given = 110000XXb

In the above example the differentiation among the 3 slaves is in the lower 3 address bits. Slave 0 requires that bit 0 = 0 and it can be uniquely addressed by 11100110b. Slave 1 requires that bit 1 = 0 and it can be uniquely addressed by 11100101b. Slave 2 requires that bit 2 = 0 and its unique address is 11100011b. To select Slaves 0 and 1 and exclude Slave 2 use address 11100100b, since it is necessary to make bit 2 = 1 to exclude slave 2.

The “Broadcast” address for each slave is created by taking the logical OR of SADDR and SADEN. Zeros in this result are treated as “don’t-cares”, e.g.:

SADDR = 01010110b

SADEN = 11111100b

Broadcast = 1111111Xb

The use of don’t-care bits provides flexibility in defining the Broadcast address, however in most applications, interpreting the “don’t-cares” as all ones, the broadcast address will be FFH.

On reset, SADDR and SADEN are initialized to 00H. This produces a “Given” address of all “don’t cares” as well as a “Broadcast” address of all XXXXXXXXb (all “don’t care” bits). This ensures that the serial port will reply to any address, and so that it is backwards compatible with the standard 80C51 Serial Port Register Description

### Register Description

##### SCON – Serial Port Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SCON | 98H, All Pages, Bit addressable | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **SM0/FE** | **Serial Port Mode Select**  SMOD0 (PCON.6) = 0:.  See Table 6.9‑1 Serial Port 0 Mode / Baud Rate Description for details.  SMOD0 (PCON.6) = 1:.  SM0/FE bit is used as frame error (FE) status flag. It is cleared by software.  0 = Frame error (FE) did not occur.  1 = Frame error (FE) occurred and detected. |
| [6] | **SM1** | Check with bit 7 description. |
| [5] | **SM2** | **Multiprocessor Communication Mode Enable**  The function of this bit is dependent on the serial port 0 mode.  Mode 0:  This bit select the baud rate between FSYS/12 and FSYS/2.  0 = The clock runs at FSYS/12 baud rate. It maintains standard 8051compatibility.  1 = The clock runs at FSYS/2 baud rate for faster serial communication.  Mode 1:  This bit checks valid stop bit.  0 = Reception is always valid no matter the logic level of stop bit.  1 = Reception is valid only when the received stop bit is logic 1 and the received data matches “Given” or “Broadcast” address.  Mode 2 or 3:  For multiprocessor communication.  0 = Reception is always valid no matter the logic level of the 9th bit.  1 = Reception is valid only when the received 9th bit is logic 1 and the received data matches “Given” or “Broadcast” address. |
| [4] | **REN** | **Receiving Enable**  0 = Serial port 0 reception Disabled.  1 = Serial port 0 reception Enabled in Mode 1,2, or 3. In Mode 0, reception is initiated by the condition REN = 1 and RI = 0. |
| [3] | **TB8** | **9th Transmitted Bit**  This bit defines the state of the 9th transmission bit in serial port 0 Mode 2 or 3. It is not used in Mode 0 or 1. |
| [2] | **RB8** | **9th Received Bit**  The bit identifies the logic level of the 9th received bit in serial port 0 Mode 2 or 3. In Mode 1, RB8 is the logic level of the received stop bit. SM2 bit as logic 1 has restriction for exception. RB8 is not used in Mode 0. |
| [1] | **TI** | **Transmission Interrupt Flag**  This flag is set by hardware when a data frame has been transmitted by the serial port 0 after the 8th bit in Mode 0 or the last data bit in other modes. When the serial port 0 interrupt is enabled, setting this bit causes the CPU to execute the serial port 0 interrupt service routine. This bit should be cleared manually via software. |
| [0] | **RI** | **Receiving Interrupt Flag**  This flag is set via hardware when a data frame has been received by the serial port 0 after the 8th bit in Mode 0 or after sampling the stop bit in Mode 1, 2, or 3. SM2 bit as logic 1 has restriction for exception. When the serial port 0 interrupt is enabled, setting this bit causes the CPU to execute to the serial port 0 interrupt service routine. This bit should be cleared manually via software. |

##### S1CON – Serial Port 1 Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| S1CON | F8H, All Pages, Bit addressable | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **SM0\_1/FE\_1** | **SM1\_1** | **SM2\_1** | **REN\_1** | **TB8\_1** | **RB8\_1** | **TI\_1** | **RI\_1** |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Name | Description |
| --- | --- | --- |
| [7] | **SM0\_1/FE\_1** | **Serial Port 1 Mode Select**  SMOD0\_1 (T3CON.6) = 0:.  See Table 6.9‑2 Serial Port 1 Mode / Baud Rate Description  for details.  SMOD0\_1 (T3CON.6) = 1:.  SM0\_1/FE\_1 bit is used as frame error (FE) status flag. It is cleared by software.  0 = Frame error (FE) did not occur.  1 = Frame error (FE) occurred and detected. |
| [6] | **SM1\_1** | Check with bit 7 description. |
| [5] | **SM2\_1** | **Multiprocessor Communication Mode Enable**  The function of this bit is dependent on the serial port 1 mode.  Mode 0:  No effect.  Mode 1:  This bit checks valid stop bit.  0 = Reception is always valid no matter the logic level of stop bit.  1 = Reception is valid only when the received stop bit is logic 1 and the received data matches “Given” or “Broadcast” address.  Mode 2 or 3:  For multiprocessor communication.  0 = Reception is always valid no matter the logic level of the 9th bit.  1 = Reception is valid only when the received 9th bit is logic 1 and the received data matches “Given” or “Broadcast” address. |
| [4] | **REN\_1** | **Receiving Enable**  0 = Serial port 1 reception Disabled.  1 = Serial port 1 reception Enabled in Mode 1,2, or 3. In Mode 0, reception is initiated by the condition REN\_1 = 1 and RI\_1 = 0. |
| [3] | **TB8\_1** | **9th Transmitted Bit**  This bit defines the state of the 9th transmission bit in serial port 1 Mode 2 or 3. It is not used in Mode 0 or 1. |
| [2] | **RB8\_1** | **9th Received Bit**  The bit identifies the logic level of the 9th received bit in serial port 1 Mode 2 or 3. In Mode 1, RB8\_1 is the logic level of the received stop bit. SM2\_1 bit as logic 1 has restriction for exception. RB8\_1 is not used in Mode 0. |
| [1] | **TI\_1** | **Transmission Interrupt Flag**  This flag is set by hardware when a data frame has been transmitted by the serial port 1 after the 8th bit in Mode 0 or the last data bit in other modes. When the serial port 1 interrupt is enabled, setting this bit causes the CPU to execute the serial port 1 interrupt service routine. This bit must be cleared manually via software. |
| [0] | **RI\_1** | **Receiving Interrupt Flag**  This flag is set via hardware when a data frame has been received by the serial port 1 after the 8th bit in Mode 0 or after sampling the stop bit in Mode 1, 2, or 3. SM2\_1 bit as logic 1 has restriction for exception. When the serial port 1 interrupt is enabled, setting this bit causes the CPU to execute to the serial port 1 interrupt service routine. This bit must be cleared manually via software. |

##### PCON – Power Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| PCON | 87H, All Pages | POR: 0001\_0000b  Others: 000U \_0000b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **SMOD** | **SMOD0** | **-** | **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 0 Mode / Baud Rate Description for details. |
| [6] | **SMOD0** | **Serial Port 0 Framing Error Flag Access Enable**  0 = SCON.7 accesses to SM0 bit.  1 = SCON.7 accesses to FE bit. |

##### T3CON – Timer 3 Control

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| T3CON | C4H, Page 0 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **SMOD\_1** | **SMOD0\_1** | **BRCK** | **TF3** | **TR3** | **T3PS[2:0]** | | |
| R/W | R/W | R/W | R/W | R/W | R/W | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **SMOD\_1** | **Serial Port 1 Double Baud Rate Enable**  Setting this bit doubles the serial port baud rate when UART1 is in Mode 2. See Table 6.9‑2 Serial Port 1 Mode / Baud Rate Description  for details. |
| [6] | **SMOD0\_1** | **Serial Port 1 Framing Error Access Enable**  0 = S1CON.7 accesses to SM0\_1 bit.  1 = S1CON.7 accesses to FE\_1 bit. |

##### SBUF – Serial Port 0 Data Buffer

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SBUF | 99H, Page 0 | 0000\_0000 b |

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

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

##### SBUF1 – Serial Port 1 Data Buffer

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SBUF1 | 9AH, Page 0 | 0000 \_0000 b |

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

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

##### IE – Interrupt Enable (Bit-addressable)

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| IE | A8H, All Pages, Bit addressable | 0000 \_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [4] | **ES** | **Enable Serial Port 0 Interrupt**  0 = Serial port 0 interrupt Disabled.  1 = Interrupt generated by TI (SCON.1) or RI (SCON.0) Enabled. |

##### EIE1 – Extensive Interrupt Enable 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| EIE1 | 9CH, Page 0 | 0000 \_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [0] | **ES1** | **Enable Serial Port 1 Interrupt**  0 = Serial port 1 interrupt Disabled.  1 = Serial port 1Interrupt Enable. When interrupt generated TI\_1 (S1CON.1) or RI\_1 (S1CON.0) set 1. |

##### SADDR0 – Slave 0 Address

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SADDR0 | A9H, Page 0 | 0000 \_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **SADDR0[7:0]** | | | | | | | |
| R/W | | | | | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:0] | **SADDR0[7:0]** | **Slave 0 Address**  This byte specifies the microcontroller’s own slave address for UATR0 multi-processor communication. |

##### SADEN0 – Slave 0 Address Mask

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SADEN0 | B9H, Page 0 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **SADEN0[7:0]** | | | | | | | |
| R/W | | | | | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:0] | **SADEN0[7:0]** | **Slave 0 Address Mask**  This byte is a mask byte of UART0 that contains “don’t-care” bits (defined by zeros) to form the device’s “Given” address. The don’t-care bits provide the flexibility to address one or more slaves at a time. |

##### SADDR1 – Slave 1 Address

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SADDR1 | BBH, Page 0 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **SADDR1[7:0]** | | | | | | | |
| R/W | | | | | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:0] | **SADDR1[7:0]** | **Slave 1 Address**  This byte specifies the microcontroller’s own slave address for UART1 multi-processor communication. |

##### SADEN1 – Slave 1 Address Mask

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SADEN1 | BAH, Page 0 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **SADEN1[7:0]** | | | | | | | |
| R/W | | | | | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:0] | **SADEN1[7:0]** | **Slave 1 Address Mask**  This byte is a mask byte of UART1 that contains “don’t-care” bits (defined by zeros) to form the device’s “Given” address. The don’t-care bits provide the flexibility to address one or more slaves at a time. |

##### AUXR1 – Auxiliary Register 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| AUXR1 | C9H, Page 1 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:3] | **-** | Reserved |
| [2] | **UART2PX** | **Serial Port 2 RX (SMC0 DATA) /TX (SMC0 CLK) Pin Exchange**  0 = Assign UART2 RXD (SMC0 DATA) to multiple I/O pin RXD.  UART2 TXD (SMC CLK) to multiple I/O pin TXD  1 = Assign UART2 RXD (SMC0 DATA) to multiple I/O pin TXD.  UART2 TXD (SMC CLK) to multiple I/O pin RXD  Note : that Pin direction is controlled by I/O type of relative pin.  RXD/TXD will exchange immediately once setting or clearing this bit. User should take care of not exchanging pins during transmission or receiving. Or it may cause unpredictable situation and no warning alarms. |
| [1] | **UART1PX** | **Serial Port 1 RX/TX Pin Exchange**  0 = Assign UART1 RXD to multiple I/O pin RXD.  UART1 TXD to multiple I/O pin TXD  1 = Assign UART1 RXD to multiple I/O pin TXD.  UART1 TXD to multiple I/O pin RXD  **Note:** that Pin direction is controlled by I/O type of relative pin.  RXD/TXD will exchange immediately once setting or clearing this bit. User should take care of not exchanging pins during transmission or receiving. Or it may cause unpredictable situation and no warning alarms. |
| [0] | **UART0PX** | **Serial Port 0 RX/TX Pin Exchange**  0 = Assign UART0 RXD to multiple I/O pin RXD.  UART0 TXD to multiple I/O pin TXD  1 = Assign UART0 RXD to multiple I/O pin TXD.  UART0 TXD to multiple I/O pin RXD  **Note:** that Pin direction is controlled by I/O type of relative pin.  RXD/TXD will exchange immediately once setting or clearing this bit. User should take care of not exchanging pins during transmission or receiving. Or it may cause unpredictable situation and no warning alarms. |

##### AUXR3 – Auxiliary Register 3

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| AUXR3 | CFH, Page 3 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:3] | **-** | Reserved |
| [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 |

## Smart Card Interface (SC)

### Overview

The MUG51TBAE provides Smart Card Interface controller (SC controller) with asynchronous protocal based on ISO/IEC 7816-3 standard. Software controls GPIO pins as the smartcard reset function and card detection function. This controller also provides UART emulation for high precision baud rate communication.

### 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

### Block Diagram

|  |
| --- |
|  |

Figure 6.10‑1 Figure SC Controller Block Diagram

### Operating Modes

#### Smart Card Mode

The Smart Card Interface controller supports activation, cold reset, warm reset and deactivation sequence by software control. The activation, cold reset, warm reset and deactivation and sequence are shown as follows.

* + - * 1. SC Interface Connection

The SC interface connection is shown **in** Figure 6.10‑2

SC\_CLK / UART2\_TXD : SC clock pin (output from MCU)

SC\_DAT / UART2\_RXD : SC data pin (bi-directional)

SC\_RST: SC reset pin (output from MCU, firmware assigned GPIO)

SC\_PWR: SC power pin (output from MCU, firmware assigned GPIO)

\*: SC\_PWR is used for power control function to turn ON/OFF the power for Smart Card. Do not use SC\_PWR as the direct power supply for Smart Card.

SC\_CD: SC card detect pin (input to MCU, detect card by a card insert mechanism)

|  |
| --- |
|  |

Figure 6.10‑2 SC Interface Connection

* + - * 1. Activation and Cold Reset

The activation and cold reset sequence is shown in Figure 6.10‑3

Set SC\_RST to low by software programming to ‘0’

Set SC\_PWR at high level by software programming to ‘1’ before timing T1 and SC\_DAT at high level (reception mode) by software programming to ‘1’ period of timing T1.

Enable SC\_CLK clock by programming CLKKEEP (SCCR2[1]) to ‘1’ after timing T1.

De-assert SC\_RST to high by software programming to ‘1’ after timing T2.

Smart Card host controller read the card ATR period of timing T3.

|  |
| --- |
|  |

Figure 6.10‑3 SC Activation and Cold Reset Sequence

* + - * 1. Warm Reset

The warm reset sequence is showed in Figure 6.10‑4

Set SC\_RST to low by software programming to ‘0’ before timing T4.

Set SC\_DAT to high by software programming to ‘1’ period of timing T4.

Set SC\_RST to high by software programming to ‘1’ after timing T5.

Smart Card host controller read the card ATR period of timing T6.

|  |
| --- |
|  |

Figure 6.10‑4 SC Warm Reset Sequence

* + - * 1. Deactivation

The deactivation sequence is showedin Figure 6.10‑5

Set SC\_RST to low by software programming to ‘0’ period of timing T7.

Stop SC\_CLK by programming CLKKEEP (SCCR2[1]) to ‘0’ period of timing T8.

Set SC\_DAT to low by software programming to ‘0’ period of timing T8.

Deactivate SC\_PWR by software programming to ‘0’ period of timing T9.

|  |
| --- |
|  |

Figure 6.10‑5 SC Deactivation Sequence

#### UART Mode

When the UARTEN (SCCR2[0]) bit is set, the Smart Card Interface controller can also be used as basic UART function. The following is the program example for UART mode.

Programming example:

Set UARTEN (SCCR2[0]) bit to enter UART mode.

Fill “0” to CONSEL (SCnCR1[4]) and AUTOCEN (SCnCR1[3]) field. (In UART mode, those fields must be “0”)

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).

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])).

Write the SCnDR (SCnDR[7:0]) (TX) register or read the SCnDR (SCnDR[7:0]) (RX) register can perform UART function.

### Smart Card Data Transfer

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

|  |
| --- |
|  |

Figure 6.10‑6 SC Data Character

#### 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 AUTOCEN (SCnCR1[3]) and then the operating convention will be decided by hardware. Software can also set the CONSEL (SCnCR1[4]) register (set to ‘0’ or ‘1’) to change the operating convention after SC received TS of answer to request (ATR).

If auto convention function is enabled by setting AUTOCEN (SCnCR1[3]) register, the setting step must be done before Answer to Request state and the first data must be 0x3B or 0x3F. After hardware received first data and stored it at buffer, the hardware will decided the convention and change the CONSEL (SCnCR1[4]) register automatically. If the first data is neither 0x3B nor 0x3F, the hardware will generate an interrupt (if ACERRIEN (ScnIE[4] = ‘1’) to CPU.

|  |
| --- |
|  |

Figure 6.10‑7 Initial Character TS

#### Error Signal and Character Repetition

According to ISO7816-3 T=0 mode description, as shown in Figure 6.10‑8 ,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

#### Block Guard Time and Extra Guard Time

Block guard time means the minimum bit length between the leading edges of two consecutive characters between different transfer directions. This field indicates the counter for the bit length of block guard time.

According to ISO7816-3, in T = 0 mode, software must fill T bit = 0 (real block guard time = 16.5) to this field; in T = 1 mode, software must fill T bit = 1 (real block guard time = 22.5) to it.

In transmit direction, the smart card sends data to smart card host controller, first. After the period is greater than (16.5 or 22.5, by T bit setting), the smart card host controller begin to send the data.

|  |
| --- |
|  |

Figure 6.10‑9 Transmit Direction Block Guard Time Operation

In receive direction, the smart card host controller sends data to smart card, first. If the smart card sends data to smart card host controller at the time which is less than (16.5 or 22.5, by T bit setting),the block guard time interrupt BGTIF (ScnIS[3]) is generated when RXBGTEN (SCnCR1[5]) and BGTIEN (ScnIE[3]) are enabled.

|  |
| --- |
|  |

Figure 6.10‑10 Receive Direction Block Guard Time Operation

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

|  |
| --- |
|  |

Figure 6.10‑11 Extra Guard Time Operation

### Register Description

##### SCnCR0 – SC Control Register 0

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SC0CR0 | D6H, Page 0 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **NSB** | **Stop Bit Length**  This field indicates the length of stop bit.  0 = The stop bit length is 2 ETU.  1= The stop bit length is 1 ETU.  **Note:** The default stop bit length is 2. SC and UART adopt NSB to program the stop bit length. |
| [6] | **T** | **T Mode**  0 = T0 (ISO7816-3 T = 0 mode).  1 = T1 (ISO7816-3 T = 1 mode).  The T mode controls the BGT (Block Guard Time). Block guard time means the minimum bit length between the leading edges of two consecutive characters between different transfer directions. This field indicates the counter for the bit length of block guard time. According to ISO7816-3, in T = 0 mode, the software must clear T bit to 0 for real block guard time = 16.5. In T = 1 mode, the software must set T bit to 1 for real block guard time = 22.5.  **Note:** In T = 0 mode, the receiver will generate the parity flag on PEF(SC0TSR[4]), if parity error is detected and also drive the parity error signal to transceiver. In T = 1 mode, the receiver will generate the parity flag on PEF(SC0TSR[4]), if parity error detected, but doesn’t drive the parity error signal to transceiver.  **Note:** The description please see section 6.10.5.2 Error Signal and Character Repetition |
| [5] | **RXBGTEN** | **Receiver Block Guard Time Function Enable Bit**  0 = Receiver block guard time function Disabled.  1 = Receiver block guard time function Enabled. |
| [4] | **CONSEL** | **Convention Selection**  0 = Direct convention.  1 = Inverse convention.  **Note 1:** This bit is auto clear to “0”, if AUTOCEN(SCnCR0[3]) is writing “1”  **Note 2:** If AUTOCEN(SCnCR0[3]) is enabled, hardware will decide the convention and change the CONSEL (SCnCR0[4]) bits automatically after SCEN (SCnCR0[0]) =”1”. |
| [3] | **AUTOCEN** | **Auto Convention Enable Bit**  0 = Auto-convention Disabled.  1 = Auto-convention Enabled. When hardware receives TS in answer to reset state and the TS is direct convention, CONSEL(SCnCR0[4]) will be set to 0 automatically, otherwise if the TS is inverse convention, and CONSEL (SCnCR0[4]) will be set to 1.  **Note:** If software enables auto convention function, the setting step must be done before Answer to Reset state and the first data must be 0x3B or 0x3F. After hardware received first data and stored it at buffer, hardware will decided the convention and change the CONSEL (SCnCR0[4]) bits automatically. |
| [2] | **TXOFF** | **TX Transition Disable Bit**  0 = The transceiver Enabled.  1 = The transceiver Disabled. |
| [1] | **RXOFF** | **RX Transition Disable Bit**  0 = The receiver Enabled.  1 = The receiver Disabled.  **Note:** If AUTOCEN (SCnCR0[3])is enabled, these fields must be ignored. |
| [0] | **SCEN** | **SC Engine Enable Bit**  Set this bit to 1 to enable SC operation. If this bit is cleared, SC will force all transition to IDLE state  **Note:** SCEN must be set to 1 before filling in other registers, or smart card will not work properly. |

##### SCnCR1 – SC Control Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SC0CR1 | D7H, Page 0 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **OPE** | **Odd Parity Enable Bit**  0 = Even number of logic 1’s are transmitted or check the data word and parity bits in receiving mode.  1 = Odd number of logic 1’s are transmitted or check the data word and parity bits in receiving mode.  **Note:** This bit has effect only when PBOFF bit is ‘0’. |
| [6] | **PBOFF** | **Parity Bit Disable Control**  0 = Parity bit is generated or checked between the “last data word bit” and “stop bit” of the serial data.  1 = Parity bit is not generated (transmitting data) or checked (receiving data) during transfer.  **Note:** In smart card mode, this field must be ‘0’ (default setting is with parity bit) |
| [5:4] | **WLS[1:0]** | **Word Length Selection**  00 = Word length is 8 bits.  01 = Word length is 7 bits.  10 = Word length is 6 bits.  11 = Word length is 5 bits.  **Note:** In smart card mode, this WLS must be ‘00’ |
| [3] | **TXDMAEN** | **SC/UART TX DMA Enable**  This bit enables the SC/UART TX operating by through PDMA transfer, TX data needs to be ready in XRAM before SC/UART TX starting.  0 = SC/UART TX DMA Disabled.  1 = SC/UART 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 = Smart Card mode.  1 = UART mode.  **Note 1:**When operating in UART mode, user must set CONSEL (SCnCR0[4]) = 0 and AUTOCEN(SCnCR0[3]) = 0.  **Note 2:**When operating in Smart Card mode, user must set UARTEN(SCnCR1 [0]) = 0.  **Note 3:**When UART is enabled, hardware will generate a reset to reset FIFO and internal state machine. |

##### SCnDR – SC Data Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SC0DR | D9H, Page 0 | 0000\_0000 b |

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

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

##### SCnEGT – SC0~1 Extra Guard Time Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SC0EGT | DAH, Page 0 | 0000\_0000 b |

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

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

##### SCnETURD0 – SCn ETU Rate Divider Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SC0ETURD0 | DBH, Page 0 | 0111\_0011 b |

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

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

##### SCnETURD1 –SC ETU Rate Divider Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SC0ETURD1 | DCH, Page 0 | 0011\_0001 b |

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

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

##### SCnIE – SC Interrupt Enable Control Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SC0IE | DDH, Page 0 | 0000\_0000 b |

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

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

##### SCnIS – SC Interrupt Status Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SC0IS | DEH, Page 0 | 0000\_0010 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | | **Tx\_Er** | **ACERRIF** | **BGTIF** | **TERRIF** | **TBEIF** | **RDAIF** |
| - | | R/W | R/W | R/W | R | R | R |

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

##### SCnTSR – SC Transfer Status Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SC0TSR | DFH, Page 0 | 0000\_1010 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **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] | **FEF** | **Receiver Frame Error Status Flag (Read Only)**  This bit is set to logic 1 whenever the received character does not have a valid “stop bit” (that is, the stop bit following the last data bit or parity bit is detected as logic 0).  **Note:** This bit is read only, but it can be cleared by writing 0 to it. |
| [4] | **PEF** | **Receiver Parity Error Status Flag (Read Only)**  This bit is set to logic 1 whenever the received character does not have a valid  “parity bit”.  **Note:** This bit is read only, but it can be cleared by writing 0 to it. |
| [3] | **TXEMPTY** | **Transmit Buffer Empty Status Flag (Read Only)**  This bit indicates TX buffer empty or not.  **Note:** When TX buffer has been transferred to Transmitter Shift Register, hardware sets this bit high. It will be cleared when writing data into DAT(SCnDR[7:0]) (TX buffer not empty). |
| [2] | **TXOV** | **TX Overflow Error Interrupt Status Flag (Read Only)**  If TX buffer is full, an additional write to DAT(SCnDR[7:0]) will cause this bit be set to “1” by hardware.  **Note:** This bit is read only, but it can be cleared by writing 0 to it. |
| [1] | **RXEMPTY** | **Receiver Buffer Empty Status Flag(Read Only)**  This bit indicates RX buffer empty or not.  **Note:** When Rx buffer has been read by CPU, hardware sets this bit high. It will be cleared when SC receives any new data. |
| [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. |

## Serial Peripheral Interface (SPI)

### Overview

The MUG51TBAE 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. It provides either Master or Slave mode, high-speed rate up to FSYS/4, transfer complete and write collision flag. For a multi-master system, SPI supports Master Mode Fault to protect a multi-master conflict.

### Features

* Supports Master or Slave mode operation
* Supports MSB first or LSB first transfer sequence
* Slave mode up to 4 Mhz

### Block Diagram

|  |
| --- |
|  |

Figure 6.11‑1 SPI Block Diagram

### Functional Description

SPI block diagram provides an overview of SPI architecture in this device. The main blocks of SPI are the SPI control register logic, SPI status logic, clock rate control logic, and pin control logic. For a serial data transfer or receiving, The SPI block exists a shift register and a read data buffer. It is double buffered in the receiving and transmit directions. Transmit data cannot be written to the shifter until the previous transfer is complete. Receiving logic consists of parallel read data buffer so the shift register is free to accept a second data, as the first received data will be transferred to the read data buffer.

The 2 set, 16 pins of SPI interface and 4 pins of SPI0 interface which are Master-In/Slave-Out (MISO), Master-Out/Slave-In (MOSI), Shift Clock (SPCLK), and Slave Select (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 (SS). The signal should stay low for any Slave access. When 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 SS pin does not function and it can be configured as a general purpose I/O. However, SS can be used as Master Mode Fault detection see Section 6.11.4.6 Mode Fault Detection via software setting if multi-master environment exists. The MUG51TBAE also provides auto-activating function to toggle SS between each byte-transfer.

|  |
| --- |
|  |

Figure 6.11‑2 SPI Multi-Master, Multi-Slave Interconnection

Figure 6.11‑2 SPI Multi-Master, Multi-Slave Interconnection 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 SS pins. MCU1 and MCU2 play either Master or Slave mode. The SS should be configured as Master Mode Fault detection to avoid multi-master conflict.

|  |
| --- |
|  |

Figure 6.11‑3 SPI Single-Master / Single-Slave Interconnection

Figure 6.11‑3 SPI Single-Master / Single-Slave Interconnection shows the simplest SPI system interconnection, single-master and signal-slave. During a transfer, the Master shifts data out to the Slave via MOSI line. While simultaneously, the Master shifts data in from the Slave via MISO line. The two shift registers in the Master MCU and the Slave MCU can be considered as one 16-bit circular shift register. Therefore, while a transfer data pushed from Master into Slave, the data in Slave will also be pulled in Master device respectively. The transfer effectively exchanges the data, which was in the SPI shift registers of the two MCUs.

* + - * 1. LSB/MSB First

By default, SPI data is transferred MSB first. If the LSBFE (SPInCR.5) is set, SPI data shifts LSB first. This bit does not affect the position of the MSB and LSB in the data register. Note that all the following description and figures are under the condition of LSBFE logic 0. MSB is transmitted and received first.

#### Operating Modes

* + - * 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.

* + - * 1. 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 SS pin also becomes input. The Master device cannot exchange data with the Slave device until the SS pin of the Slave device is externally pulled low. Before data transmissions occurs, the SS of the Slave device should be pulled and remain low until the transmission is complete. If SS goes high, the SPI is forced into idle state. If the 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.

#### SPI Master Clock Configuration

These four bits select four grades of SPI clock divider. The clock rates below are illustrated under FSYS condition.SPR[3:0] are valid only under Master mode (MSTR = 1). If under Slave mode, the clock will automatically synchronize with the external clock on SPICLK pin from Master device up to FSYS/4 communication speed. Following is the SPI clock rate define table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SPR3 | SPR2 | SPR1 | SPR0 | Divider of system |
| 0 | 0 | 0 | 0 | 2 |
| 0 | 0 | 0 | 1 | 4 |
| 0 | 0 | 1 | 0 | 8 |
| 0 | 0 | 1 | 1 | 16 |
| 0 | 1 | 0 | 0 | 32 |
| 0 | 1 | 0 | 1 | 64 |
| 0 | 1 | 1 | 0 | 128 |
| 0 | 1 | 1 | 1 | 256 |
| 1 | 0 | 0 | 0 | 3 |
| 1 | 0 | 0 | 1 | 6 |
| 1 | 0 | 1 | 0 | 12 |
| 1 | 0 | 1 | 1 | 24 |
| 1 | 1 | 0 | 0 | 48 |
| 1 | 1 | 0 | 1 | 96 |
| 1 | 1 | 1 | 0 | 192 |
| 1 | 1 | 1 | 1 | 384 |

Table 6.11‑1 SPI Master Clock Rate Define Table

SPIS[1:0] (SPI0CR1 [1:0]) provides a configurable suspend interval, 0.5 ~ 2.5 SPI clock periods, between two successive transaction words in Master mode. The definition of the suspend interval is the interval between the last clock edge of the preceding transaction word and the first clock edge of the following transaction word. Following shows the suspend interval select define.

|  |  |  |  |
| --- | --- | --- | --- |
| CPHA | SPIS1 | SPIS0 | SPI clock |
| 0 | 0 | 0 | 0.0 |
| 0 | 0 | 1 | 0.5 |
| 0 | 1 | 0 | 1.5 |
| 0 | 1 | 1 | 2.0 |
| 1 | 0 | 0 | 0.0 |
| 1 | 0 | 1 | 1.0 |
| 1 | 1 | 0 | 2.0 |
| 1 | 1 | 1 | 2.5 |

Table 6.11‑2 SPI Clock Suspend Interval Select

#### 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 6.11‑4 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 6.11‑4 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 SS signal needs to be taken care. As shown in Figure 6.11‑4 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 SS is used for preparing the MSB on MISO line. The 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 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 SS falling edge. Therefore, the SS line can remain low between successive transfers. This format may be preferred in systems having single fixed Master and single fixed Slave. The SS line of the unique Slave device can be tied to GND as long as only CPHA = 1 clock mode is used.

The SPI should be configured before it is enabled (SPIEN = 1), or a change of LSBFE, MSTR, CPOL, CPHA and SPR[1:0] will abort a transmission in progress and force the SPI system into idle state. Prior to any configuration bit changed, SPIEN must be disabled first.

|  |
| --- |
|  |

Figure 6.11‑5 SPI Clock and Data Format with CPHA = 0

|  |
| --- |
|  |

Figure 6.11‑6 SPI Clock and Data Format with CPHA = 1

#### Slave Select Pin Configuration

The MUG51TBAE SPI gives a flexible pin feature for different system requirements. When the SPI operates as a Slave, pin always rules as Slave select input. When the Master mode is enabled, 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. is configured as a input pin to check if the Mode Fault appears. On the contrary, if DISMODF is 1, Mode Fault is inactivated and the SSOE bit takes over to control the function of the pin. While SSOE is 1, it means the Slave select signal will generate automatically to select a Slave device. The as output pin of the Master usually connects with the input pin of the Slave device. The 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, is no more used by the SPI and reverts to be a general purpose I/O pin.

|  |  |  |  |
| --- | --- | --- | --- |
| **DISMODF** | **SSOE** | **Master Mode (MSTR = 1)** | **Slave Mode (MSTR = 0)** |
| 0 | X | input for Mode Fault | Input for Slave select |
| 1 | 0 | General purpose I/O |
| 1 | 1 | Automatic output |

Table 6.11‑3 Slave Select Pin Configurations

#### 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 input line is configured for Mode Fault input depending on SPInCR0, a Mode Fault error occurs once the 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.

* + - * 1. 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.

* + - * 1. Overrun Error

For receiving data, the SPI is double buffered in the receiving direction. The received data is transferred into a parallel read data buffer so the shifter is free to accept a second serial byte. However, the received data should be read from SPInDR before the next data has been completely shifted in. As long as the first byte is read out of the read data buffer and SPIF is cleared before the next byte is ready to be transferred, no overrun error condition occurs. Otherwise the overrun error occurs. In this condition, the second byte data will not be successfully received into the read data register and the previous data will remains. If overrun occur, SPIOVF (SPInSR.5) will be set via hardware. An SPIOVF setting will also require an interrupt if enabled. Figure 6.11‑7 SPI Overrun Waveform shows the relationship between the data receiving and the overrun error.

|  |
| --- |
|  |

Figure 6.11‑7 SPI Overrun Waveform

#### 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 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 6.11‑8 SPI Interrupt Request

### Register Description

##### SPInCR0 – Serial Peripheral Control Register0

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SPI0CR0 | 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 (SPInSR.3) bit to determine the feature of pin as shown inTable 6.11‑3 Slave Select Pin Configurations. This bit takes effect only under MSTR = 1 and DISMODF = 1 condition.  0 = functions as a general purpose I/O pin.  1 = 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 6.11‑4 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 6.11‑4 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. |
| [1:0] | **SPR[1:0]** | **SPI Clock Rate Select**  These four bits select four grades of SPI clock divider. The clock rates below are illustrated under FSYS condition. See Table 6.11‑1 SPI Master Clock Rate Define Table  SPR[3:0] are valid only under Master mode (MSTR = 1). If under Slave mode, the clock will automatically synchronize with the external clock on SPICLK pin from Master device up to FSYS/4 communication speed. |

##### SPInCR1 – Serial Peripheral Control Register1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SPI0CR1 | F3H, Page 1 | 0000\_0000 b |

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

| Bit | Name | Description |
| --- | --- | --- |
| [7:6] | **-** | Reserved. |
| [5:4] | **SPRn[3:2]** | **SPI Clock Rate Select**  These two bits select grades of SPI clock divider. The clock rates below are illustrated under FSYS condition find in Table 6.11‑1 SPI Master Clock Rate Define Table  SPR[3:0] are valid only under Master mode (MSTR = 1). If under Slave mode, the clock will automatically synchronize with the external clock on SPICLK pin from Master device up to FSYS/4 communication speed. |
| [3] | **TXDMAEN** | **SPI TX DMA Enable**  This bit enables the SPI TX operating by through PDMA transfer, TX data needs to be ready in XRAM before SPI TX starting.  0 = SPI TX DMA Disabled.  1 = SPI TX DMA Enabled. |
| [2] | **RXDMAEN** | **SPI RX DMA Enable**  This bit enables the SPI RX operating by through PDMA transfer, RX data are saved in XRAM after SPI RX operation.  0 = SPI RX DMA Disabled.  1 = SPI RX DMA Enabled. |
| [1:0] | **SPISn[1:0]** | **SPI Interval Time Selection Between Adjacent Bytes**  SPIS[1:0] and CPHA select eight grades of SPI interval time selection between adjacent bytes. As see Table 6.11‑2 SPI Clock Suspend Interval Select  SPIS[1:0] are valid only under Master mode (MSTR = 1). |

##### SPInSR – Serial Peripheral Status Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SPI0SR | F4H, Page 0 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **SPIF** | **SPI Complete Flag**  This bit is set to logic 1 via hardware while an SPI data transfer is complete or an receiving data has been moved into the SPI read buffer. If ESPI and EA are enabled, an SPI interrupt will be required. This bit should be cleared via software. Attempting to write to SPInDR is inhibited if SPIF is set. |
| [6] | **WCOL** | **Write Collision Error Flag**  This bit indicates a write collision event. Once a write collision event occurs, this bit will be set. It should be cleared via software. |
| [5] | **SPIOVF** | **SPI Overrun Error Flag**  This bit indicates an overrun event. Once an overrun event occurs, this bit will be set. If ESPI and EA are enabled, an SPI interrupt will be required. This bit should be cleared via software. |
| [4] | **MODF** | **Mode Fault Error Flag**  This bit indicates a Mode Fault error event. If pin is configured as Mode Fault input (MSTR = 1 and DISMODF = 0) and is pulled low by external devices, a Mode Fault error occurs. Instantly MODF will be set as logic 1. If ESPI and EA are enabled, an SPI interrupt will be required. This bit should be cleared via software. |
| [3] | **DISMODF** | **Disable Mode Fault Error Detection**  This bit is used in combination with the SSOE (SPInCR.7) bit to determine the feature of pin as shown in Table 6.11‑3 Slave Select Pin Configurations. DISMODF is valid only in Master mode (MSTR = 1).  0 = Mode Fault detection Enabled. serves as input pin for Mode Fault detection disregard of SSOE.  1 = Mode Fault detection Disabled. The feature of follows SSOE bit. |
| [2] | **DISSPIF** | **Disable SPI Complete Interrupt**  This bit is used to disable SPI complete interrupt while an SPI data transfer is complete or an receiving data has been moved into the SPI read buffer. Especially in SPI PDMA operation.  0 = SPI Complete Interrupt Enabled while ESPI and EA are enabled,.  1 = SPI Complete Interrupt Disabled. |
| [1] | **TXBFF** | **SPI TX Buffer Full Flag**  0 = SPI TX buffer is empty.  1 = SPI TX buffer is full. |

##### SPInDR – Serial Peripheral Data Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| SPI0DR | F5H, Page 0 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:0] | **SPIDR[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. |

## Inter-Integrated Circuit (I2C)

### Overview

The MUG51TBAE provides two Inter-Integrated Circuit (I2C) bus to serves as an serial interface between the microcontrollers and the I2C devices. The I2C bus used two wires design (a serial data line SDA and a serial clock line 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.

### Features

* 2 sets of I2C devices
* Master/Slave mode
* Bidirectional data transfer between masters and slaves
* Multi-master bus (no central master)
* 7-bit addressing mode
* Standard mode (100 kbps) and Fast mode (400 kbps).
* Supports 8-bit time-out counter requesting the I2C interrupt if the I2C bus hangs up and timer-out counter overflows
* Multiple address recognition (four slave addresses with mask option)
* Supports hold time programmable

### Functional Description

For a bi-directional transfer operation, the SDA and SCL pins should be open-drain pads. This implements a wired-AND function, which is essential to the operation of the interface. A low level on a I2C bus line is generated when one or more I2C devices output a “0”. A high level is generated when all I2C devices output “1”, allowing the pull-up resistors to pull the line high. In MUG51TBAE, user should set output latches of SCL and SDA. As logic 1 before enabling the I2C function by setting I2CEN.

|  |
| --- |
|  |

Figure 6.12‑1 I2C Bus Interconnection

The I2C is considered free when both lines are high. Meanwhile, any device, which can operate as a master can occupy the bus and generate one transfer after generating a START condition. The bus now is considered busy before the transfer ends by sending a STOP condition. The master generates all of the serial clock pulses and the START and STOP condition. However if there is no START condition on the bus, all devices serve as not addressed slave. The hardware looks for its own slave address or a General Call address. (The General Call address detection may be enabled or disabled by GC (I2CnADDRx.0).) If the matched address is received, an interrupt is requested.

Every transaction on the I2C bus is 9 bits long, consisting of 8 data bits (MSB first) and a single acknowledge bit. The number of bytes per transfer (defined as the time between a valid START and STOP condition) is unrestricted but each byte has to be followed by an acknowledge bit. The master device generates 8 clock pulse to send the 8-bit data. After the 8th falling edge of the SCL line, the device outputting data on the SDA changes that pin to an input and reads in an acknowledge value on the 9th clock pulse. After 9th clock pulse, the data receiving device can hold SCL line stretched low if next receiving is not prepared ready. It forces the next byte transaction suspended. The data transaction continues when the receiver releases the SCL line.

|  |
| --- |
|  |

Figure 6.12‑2 I2C Bus Protocol

#### START and STOP Condition

The protocol of the I2C bus defines two states to begin and end a transfer, START (S) and STOP (P) conditions. A START condition is defined as a high-to-low transition on the SDA line while SCL line is high. The STOP condition is defined as a low-to-high transition on the SDA line while SCL line is high. A START or a STOP condition is always generated by the master and I2C bus is considered busy after a START condition and free after a STOP condition. After issuing the STOP condition successful, the original master device will release the control authority and turn back as a not addressed slave. Consequently, the original addressed slave will become a not addressed slave. The 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 6.12‑3 START, Repeated START, and STOP Conditions

#### 7-Bit Address with Data Format

Following the START condition is generated, one byte of special data should be transmitted by the master. It includes a 7-bit long slave address (SLA) following by an 8th bit, which is a data direction bit (R/W), to address the target slave device and determine the direction of data flow. If R/W bit is 0, it indicates that the master will write information to a selected slave. Also, if R/W bit is 1, it indicates that the master will read information from the addressed slave. An address packet consisting of a slave address and a read I or a write (W) bit is called SLA+R or SLA+W, respectively. A transmission basically consists of a START condition, a SLA+W/R, one or more data packets and a STOP condition. After the specified slave is addressed by SLA+W/R, the second and following 8-bit data bytes issue by the master or the slave devices according to the R/W bit configuration.

Figure 6.12‑4 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 6.12‑4 Master Transmits Data to Slave by 7-bit

Figure 6.12‑5 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 6.12‑5 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 6.12‑6 Data Format of One I2C Transfer

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

#### Acknowledge

The 9th SCL pulse for any transferred byte is dedicated as an Acknowledge (ACK). It allows receiving devices (which can be the master or slave) to respond back to the transmitter (which also can be the master or slave) by pulling the SDA line low. The acknowledge-related clock pulse is generated by the master. The transmitter should release control of SDA line during the acknowledge clock pulse. The ACK is an active-low signal, pulling the SDA line low during the clock pulse high duty, indicates to the transmitter that the device has received the transmitted data. Commonly, a receiver, which has been addressed is requested to generate an ACK after each byte has been received. When a slave receiver does not acknowledge (NACK) the slave address, the SDA line should be left high by the slave so that the mater can generate a STOP or a repeated START condition.

If a slave-receiver does acknowledge the slave address, it switches itself to not addressed slave mode and cannot receive any more data bytes. This slave leaves the SDA line high. The master should generate a STOP or a repeated START condition.

If a master-receiver is involved in a transfer, because the master controls the number of bytes in the transfer, it should signal the end of data to the slave-transmitter by not generating an acknowledge on the last byte. The slave-transmitter then switches to not addressed mode and releases the SDA line to allow the master to generate a STOP or a repeated START condition.

|  |
| --- |
|  |

Figure 6.12‑7 Acknowledge Bit

#### Arbitration

A master may start a transfer only if the bus is free. It is possible for two or more masters to generate a START condition. In these situations, an arbitration scheme takes place on the SDA line, while SCL is high. During arbitration, the first of the competing master devices to place‘a’’1’ (high) on SDA while another master transmits‘a’’0’ (low) switches off its data output stage because the level on the bus does not match its own level. The arbitration lost master switches to the not addressed slave immediately to detect its own slave address in the same serial transfer whether it is being addressed by the winning master. It also releases SDA line to high level for not affecting the data transfer continued by the winning master. However, the arbitration lost master continues generating clock pulses on SCL line until the end of the byte in which it loses the arbitration.

Arbitration is carried out by all masters continuously monitoring the SDA line after outputting data. If the value read from the SDA line does not match the value that the master has to output, it has lost the arbitration. Note that a master can only lose arbitration when it outputs a high SDA value while another master outputs a low value. Arbitration will continue until only one master remains, and this may take many bits. Its first stage is a comparison of address bits, and if both masters are trying to address the same device, arbitration continues on to the comparison of data bits or acknowledge bit.

|  |
| --- |
|  |

Figure 6.12‑8 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.

#### 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 6.12‑9 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 6.12‑9 Control I2C Bus According To The Current I2C Status

* + - * 1. 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 6.12‑10 Flow and Status of Master Transmitter Mode

* + - * 1. 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 6.12‑11 Flow and Status of Master Receiver Mode

* + - * 1. Slave Receiver

In the slave receiver mode, several bytes of data are received form a master transmitter. Before a transmission is commenced, I2CnADDRx should be loaded with the address to which the device will respond when addressed by a master. I2CnCLK does not affect in slave mode. The AA bit should be set to enable acknowledging its own slave address. After the initialization above, the 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.

* + - * 1. 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 6.12‑12 Flow and Status of Slave Receiver Mode

* + - * 1. 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 6.12‑13 Flow and Status of General Call Mode

#### 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 SDA and SCL lines, sets the SI flag, and loads I2CnSTAT as 00H. To recover from a bus error, the STO bit should be set and then SI should be cleared. After that, STO is cleared by hardware and release the 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 SDA line e.g. a slave device out of bit synchronization, the problem can be solved by transmitting additional clock pulses on the SCL line. The I2C hardware transmits additional clock pulses when the STA bit is set, but no START condition can be generated because the SDA line is pulled low. When the SDA line is eventually released, a normal START condition is transmitted, state 08H is entered, and the serial transaction continues. If a repeated START condition is transmitted while SDA is obstructed low, the 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. | | |

Table 6.12‑1 Status Display In I2STAT Register

#### 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 6.12‑14 I2C Time-Out Counter

#### 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.

### Register Description

##### I2CnCON – I2C Control

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

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7] | **I** | **I2Cn Hold Time Extend Enable**  0 = I2C DATA to SCL hold time extend disabled.  1 = I2C DATA to SCL hold time extend enabled, extend 8 system clock. |
| [6] | **I2CEN** | **I2Cn Bus Enable**  0 = I2C bus Disabled.  1 = I2C bus Enabled.  Before enabling the I2C, SCL and SDA port latches should be set to logic 1. |
| [5] | **STA** | **START Flag**  When STA is set, the I2C generates a START condition if the bus is free. If the bus is busy, the I2C waits for a STOP condition and generates a START condition following.  If STA is set while the I2C is already in the master mode and one or more bytes have been transmitted or received, the I2C generates a repeated START condition.  Note that STA can be set anytime even in a slave mode, but STA is not hardware automatically cleared after START or repeated START condition has been detected. User should take care of it by clearing STA manually. |
| [4] | **STO** | **STOP Flag**  When STO is set if the I2C is in the master mode, a STOP condition is transmitted to the bus. STO is automatically cleared by hardware once the STOP condition has been detected on the bus.  The STO flag setting is also used to recover the I2C device from the bus error state (I2CnSTAT as 00H). In this case, no STOP condition is transmitted to the I2C bus.  If the STA and STO bits are both set and the device is original in the master mode, the I2C bus will generate a STOP condition and immediately follow a START condition. If the device is in slave mode, STA and STO simultaneous setting should be avoid from issuing illegal I2C frames. |
| [3] | **SI** | **I2Cn Interrupt Flag**  SI flag is set by hardware when one of 26 possible I2C status (besides F8H status) is entered. After SI is set, the software should read I2CnSTAT register to determine which step has been passed and take actions for next step.  SI is cleared by software. Before the SI is cleared, the low period of SCL line is stretched. The transaction is suspended. It is useful for the slave device to deal with previous data bytes until ready for receiving the next byte.  The serial transaction is suspended until SI is cleared by software. After SI is cleared, I2C bus will continue to generate START or repeated START condition, STOP condition, 8-bit data, or so on depending on the software configuration of controlling byte or bits. Therefore, user should take care of it by preparing suitable setting of registers before SI is software cleared. |
| [2] | **AA** | **Acknowledge Assert Flag**  If the AA flag is set, an ACK (low level on SDA) will be returned during the acknowledge clock pulse of the SCL line while the I2C device is a receiver or an own-address-matching slave.  If the AA flag is cleared, a NACK (high level on SDA) will be returned during the acknowledge clock pulse of the SCL line while the I2C device is a receiver or an own-address-matching slave. A device with its own AA flag cleared will ignore its own salve address and the General Call. Consequently, SI will note be asserted and no interrupt is requested.  Note that if an addressed slave does not return an ACK under slave receiver mode or not receive an ACK under slave transmitter mode, the slave device will become a not addressed slave. It cannot receive any data until its AA flag is set and a master addresses it again.  There is a special case of I2CnSTAT value C8H occurs under slave transmitter mode. Before the slave device transmit the last data byte to the master, AA flag can be cleared as 0. Then after the last data byte transmitted, the slave device will actively switch to not addressed slave mode of disconnecting with the master. The further reading by the master will be all FFH. |
| [1:0] | **-** | Reserved |

|  |
| --- |
|  |

Figure 6.12 Hold Time Extend Enable

##### I2CnSTAT – I2C Status

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| I2C0STAT | BDH, Page 0 | 1111\_1000 b |
| I2C1STAT | B4H, Page 0 | 1111\_1000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **I2CSTAT[7:3]** | | | | | **0** | **0** | **0** |
| R | | | | | R | R | R |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:3] | **I2CSTAT[7:3]** | **I2Cn Status Code**  The MSB five bits of I2CnSTAT contains the status code. There are 27 possible status codes. When I2CnSTAT is F8H, no relevant state information is available and SI flag keeps 0. All other 26 status codes correspond to the I2C states. When each of these status is entered, SI will be set as logic 1 and a interrupt is requested. |
| [2:0] | **0** | **Reserved**  The least significant three bits of I2CnSTAT are always read as 0. |

##### I2CnDAT – I2C Data

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| I2C0DAT | BCH, Page 0 | 0000\_0000 b |
| I2C1DAT | B3H, Page 0 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **I2CDAT[7:0]** | | | | | | | |
| R/W | | | | | | | |

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:0] | **I2CDAT[7:0]** | **I2Cn Data**  I2CnDAT contains a byte of the I2C data to be transmitted or a byte, which has just received. Data in I2CnDAT remains as long as SI is logic 1. The result of reading or writing I2CnDAT during I2C transceiver progress is unpredicted.  While data in I2CnDAT is shifted out, data on the bus is simultaneously being shifted in to update I2CnDAT. I2CnDAT always shows the last byte that presented on the I2C bus. Thus the event of lost arbitration, the original value of I2CnDAT changes after the transaction. |

I2C Data Shifting Direction.

|  |
| --- |
|  |

##### I2CnADDRM – I2Cn Address Mask

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| I2C0ADDRM | CFH, Page 2 | 0000\_0000 b |
| I2C1ADDRM | D7H, Page 2 | 0000\_0000 b |

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

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

##### I2CnADDRx – I2Cn Own Slave Address

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

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:1] | **I2CADDRx[7:1]** | **I2Cn Device’s Own Slave Address**  In master mode:  These bits have no effect.  In slave mode:  These 7 bits define the slave address of this I2C device by user. The master should address I2C device by sending the same address in the first byte data after a START or a repeated START condition. If the AA flag is set, this I2C device will acknowledge the master after receiving its own address and become an addressed slave. Otherwise, the addressing from the master will be ignored.  Note that I2CnADDRx[7:1] should not remain its default value of all 0, because address 0x00 is reserved for General Call. |
| [0] | **GC** | **General Call Bit**  In master mode:  This bit has no effect.  In slave mode:  0 = The General Call is always ignored.  1 = The General Call is recognized if AA flag is 1; otherwise, it is ignored if AA is 0. |

##### I2CnCLK – I2C Clock

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| I2C0CLK | BEH, Page 0 | 0000\_1001 b |
| I2C1CLK | B5H, Page 0 | 0000\_1001 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:0] | **I2CCLK[7:0]** | **I2Cn Clock Setting**  In master mode:  This register determines the clock rate of I2C bus when the device is in a master mode. The clock rate follows the equation,  .  Note that the I2CnCLK value of 00H and 01H are not valid. This is an implement limitation.  In slave mode:  This byte has no effect. In slave mode, the I2C device will automatically synchronize with any given clock rate up to 400k bps. |

##### I2CnTOC – I2C Time-out Counter

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| I2C0TOC | BFH, Page 0 | 0000\_0000 b |
| I2C1TOC | B6H, Page 0 | 0000\_0000 b |

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

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

### Typical Structure of I2C Interrupt Service Routine

The following software example in C language for KEILTM C51 compiler shows the typical structure of the I2C interrupt service routine including the 26 state service routines and may be used as a base for user applications. User can follow or modify it for their own application. If one or more of the five modes are not used, the associated state service routines may be removed, but care should be taken that a deleted routine can never be invoked.

Void I2C\_ISR (void) interrupt 6

{

switch (I2STAT)

{

//===============================================

//Bus Error, always put in ISR for noise handling

//===============================================

case 0x00: /\*00H, bus error occurs\*/

STO = 1; //recover from bus error

break;

//===========

//Master Mode

//===========

case 0x08: /\*08H, a START transmitted\*/

STA = 0; //STA bit should be cleared by software

I2DAT = SLA\_ADDR1; //load SLA+W/R

break;

case 0x10: /\*10H, a repeated START transmitted\*/

STA = 0;

I2DAT = SLA\_ADDR2;

break;

//=======================

//Master Transmitter Mode

//=======================

case 0x18: /\*18H, SLA+W transmitted, ACK received\*/

I2DAT = NEXT\_SEND\_DATA1; //load DATA

break;

case 0x20: /\*20H, SLA+W transmitted, NACK received\*/

STO = 1; //transmit STOP

AA = 1; //ready for ACK own SLA+W/R or General Call

break;

case 0x28: /\*28H, DATA transmitted, ACK received\*/

if (Conti\_TX\_Data) //if continuing to send DATA

I2DAT = NEXT\_SEND\_DATA2;

else //if no DATA to be sent

{

STO = 1;

AA = 1;

}

break;

case 0x30: /\*30H, DATA transmitted, NACK received\*/

STO = 1;

AA = 1;

break;

//===========

//Master Mode

//===========

case 0x38: /\*38H, arbitration lost\*/

STA = 1; //retry to transmit START if bus free

break;

//====================

//Master Receiver Mode

//====================

case 0x40: /\*40H, SLA+R transmitted, ACK received\*/

AA = 1; //ACK next received DATA

break;

case 0x48: /\*48H, SLA+R transmitted, NACK received\*/

STO = 1;

AA = 1;

break;

case 0x50: /\*50H, DATA received, ACK transmitted\*/

DATA\_RECEIVED1 = I2DAT; //store received DATA

if (To\_RX\_Last\_Data1) //if last DATA will be received

AA = 0; //not ACK next received DATA

else //if continuing receiving DATA

AA = 1;

break;

case 0x58: /\*58H, DATA received, NACK transmitted\*/

DATA\_RECEIVED\_LAST1 = I2DAT;

STO = 1;

AA = 1;

break;

//====================================

//Slave Receiver and General Call Mode

//====================================

case 0x60: /\*60H, own SLA+W received, ACK returned\*/

AA = 1;

break;

case 0x68: /\*68H, arbitration lost in SLA+W/R

own SLA+W received, ACK returned \*/

AA = 0; //not ACK next received DATA after

//arbitration lost

STA = 1; //retry to transmit START if bus free

break;

case 0x70: //\*70H, General Call received, ACK returned

AA = 1;

break;

case 0x78: /\*78H, arbitration lost in SLA+W/R

General Call received, ACK returned\*/

AA = 0;

STA = 1;

break;

case 0x80: /\*80H, previous own SLA+W, DATA received,

ACK returned\*/

DATA\_RECEIVED2 = I2DAT;

if (To\_RX\_Last\_Data2)

AA = 0;

else

AA = 1;

break;

case 0x88: /\*88H, previous own SLA+W, DATA received,

NACK returned, not addressed SLAVE mode

entered\*/

DATA\_RECEIVED\_LAST2 = I2DAT;

AA = 1; //wait for ACK next Master addressing

break;

case 0x90: /\*90H, previous General Call, DATA received,

ACK returned\*/

DATA\_RECEIVED3 = I2DAT;

if (To\_RX\_Last\_Data3)

AA = 0;

else

AA = 1;

break;

case 0x98: /\*98H, previous General Call, DATA received,

NACK returned, not addressed SLAVE mode

entered\*/

DATA\_RECEIVED\_LAST3 = I2DAT;

AA = 1;

break;

//==========

//Slave Mode

//==========

case 0Xa0: /\*A0H, STOP or repeated START received while

still addressed SLAVE mode\*/

AA = 1;

break;

//======================

//Slave Transmitter Mode

//======================

case 0Xa8: /\*A8H, own SLA+R received, ACK returned\*/

I2DAT = NEXT\_SEND\_DATA3;

AA = 1; //when AA is “1”, not last data to be

//transmitted

break;

case 0Xb0: /\*B0H, arbitration lost in SLA+W/R

own SLA+R received, ACK returned \*/

I2DAT = DUMMY\_DATA;

AA = 0; //when AA is “0”, last data to be

//transmitted

STA = 1; //retry to transmit START if bus free

break;

case 0Xb8: /\*B8H, previous own SLA+R, DATA transmitted,

ACK received\*/

I2DAT = NEXT\_SEND\_DATA4;

if (To\_TX\_Last\_Data) //if last DATA will be transmitted

AA = 0;

else

AA = 1;

break;

case 0Xc0: /\*C0H, previous own SLA+R, DATA transmitted,

NACK received, not addressed SLAVE mode

entered\*/

AA = 1;

break;

case 0Xc8: /\*C8H, previous own SLA+R, last DATA trans-

mitted, ACK received, not addressed SLAVE

AA = 1; mode entered\*/

break;

}//end of switch (I2STAT)

SI = 0; //SI should be the last command of I2C ISR

while(STO); //wait for STOP transmitted or bus error

//free, STO is cleared by hardware

}//end of I2C\_ISR

## Analog Comparator Controller (ACMP)

### Overview

The MUG51TBAE contains two comparators. The comparator output is logic 1 when positive input is greater than negative input; otherwise, the output is 0. The comparator can be configured to generate an interrupt when the comparator output value changes.

### Features

* Analog input voltage range: 0 ~ AVDD (voltage of AVDD pin)
* Supports hysteresis function
* Supports wake-up function
* Selectable input sources of negative input
* Comparator ACMP0 supports:
  + - * 4 positive sources
* P2.5 (ACMPn\_P0)
* P2.3 (ACMPn\_P1)
* P2.1 (ACMPn\_P2)
* P3.1 (ACMPn\_P3)
  + - * 4 negative sources
* P2.4 (ACMP0\_N0)
* Comparator Reference Voltage (CRV)
* VBG (BAND-GAP voltage)
* P2.0 (ACMP0\_N1)
* Comparator ACMP1 supports:
  + - * 4 positive sources
* P2.5 (ACMPn\_P0)
* P2.3 (ACMPn\_P1)
* P2.1 (ACMPn\_P2)
* P3.1 (ACMPn\_P3)
  + - * 4 negative sources
* P2.2 (ACMP1\_N0)
* Comparator Reference Voltage (CRV)
* VBG (BAND-GAP voltage)
* P3.2 (ACMP1\_N1)

### Block Diagram

|  |
| --- |
|  |

Figure 6.13‑1 Analog Comparator Block Diagram

### Functional Description

#### Hysteresis Function

The analog comparator provides the hysteresis function to make the comparator to have a stable output transition. If comparator output is 0, it will not be changed to 1 until the positive input voltage exceeds the negative input voltage by a high threshold voltage. Similarly, if comparator output is 1, it will not be changed to 0 until the positive input voltage drops below the negative input voltage by a low threshold voltage.

|  |
| --- |
|  |

Figure 6.13‑2 Comparator Hysteresis Function

#### Comparator Reference Voltage (CRV)

The comparator reference voltage (CRV) module is responsible for generating reference voltage for comparators. The CRV module consists of resisters ladder and analog switch. User can set the CRV output voltage by setting the CRVnCTL(ACMPVREF). The CRV output voltage can be selected as the negative input of comparator by setting NEGSEL (ACMPCR0[5:4]).

Features:

1. User selectable references voltage source by setting the CRVSSEL(ACMPCR2[1]) register.
2. User selectable references voltage by setting the CRVnCTL(ACMPVREF) register.

Comparator reference voltage = VIN \* (1/6+CRVnCTL/12); VIN = AVDD or VREF.

|  |
| --- |
|  |

Figure 6.13‑3 Comparator Reference Voltage Block Diagram

Note that If CRVEN = 0, CRV0 is equal to 0 and CRV1 is equal to Band-gap.

#### Interrupt Sources

The comparator generates an output ACMPnO (ACMPSR). If the ACMPIE (ACMPCR0[1]) bit in ACMPCR0 is set, a state change on the comparator output ACMPnO (ACMPSR) will cause comparator flag ACMPnIF (ACMPSR) be set and the comparator interrupt requested. User can write 1 to ACMPnIF (ACMPSR) through software to stop interrupt request.

|  |
| --- |
|  |

Figure 6.13‑4 Analog Comparator Interrupt Sources

### Register Description

##### ACMPCR0 – Analog Comparator Control Register 0

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| ACMPCR0 | D2H, Page 0 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **POSSEL** | **Comparator 0 Positive Input Selection**  00 = ACMP0\_P0 (P2.5) pin.  01 = ACMP0\_P1 (P2.3) pin.  10 = ACMP0\_P2 (P2.1) pin.  11 = ACMP0\_P3 (P3.1) pin. |
| [5:4] | **NEGSEL** | **Comparator 0 Negative Input Selection**  00 = ACMP0\_N0 (P2.4) pin.  01 = Internal comparator reference voltage (CRV).  10 = VBG (Band-gap).  11 = ACMP0\_N1 (P2.0)pin. |
| [3] | **WKEN** | **Comparator 0 Power-Down Wake-Up Enable Bit**  0 = Comparator 0 Wake-up function Disabled.  1 = Comparator 0 Wake-up function Enabled. |
| [2] | **HYSEN** | **Comparator 0 Hysteresis Enable Bit**  0 = Comparator 0 hysteresis Disabled.  1 = Comparator 0 hysteresis Enabled. |
| [1] | **ACMPIE** | **Comparator 0 Interrupt Enable Bit**  0 = Comparator 0 interrupt Disabled.  1 = Comparator 0 interrupt Enabled. If WKEN (ACMPCR1[3]) is set to 1, the wake-up interrupt function will be enabled as well. |
| [0] | **ACMPEN** | **Comparator 0 Enable Bit**  0 = Comparator 0 Disabled.  1 = Comparator 0 Enabled. |

##### ACMPCR1 – Analog Comparator Control Register 1

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| ACMPCR1 | D3H, Page 0 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **POSSEL** | **Comparator 1 Positive Input Selection**  00 = ACMP1\_P0 (P2.5) pin.  01 = ACMP1\_P1 (P2.3) pin.  10 = ACMP1\_P2 (P2.1) pin.  11 = ACMP1\_P3 (P3.1) pin. |
| [5:4] | **NEGSEL** | **Comparator 1 Negative Input Selection**  00 = ACMP1\_N0 (P2.2) pin.  01 = Internal comparator reference voltage (CRV).  10 = VBG (Band-gap).  11 = ACMP1\_N1 (P3.2)pin. |
| [3] | **WKEN** | **Comparator 1 Power-Down Wake-Up Enable Bit**  0 = Comparator 1 Wake-up function Disabled.  1 = Comparator 1 Wake-up function Enabled. |
| [2] | **HYSEN** | **Comparator 1 Hysteresis Enable Bit**  0 = Comparator 1 hysteresis Disabled.  1 = Comparator 1 hysteresis Enabled. |
| [1] | **ACMPIE** | **Comparator 1 Interrupt Enable Bit**  0 = Comparator 1 interrupt Disabled.  1 = Comparator 1 interrupt Enabled. If WKEN (ACMPCR2[3]) is set to 1, the wake-up interrupt function will be enabled as well. |
| [0] | **ACMPEN** | **Comparator 1 Enable Bit**  0 = Comparator 1 Disabled.  1 = Comparator 1 Enabled. |

##### ACMPCR2 – Analog Comparator Control Register 2

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| ACMPCR2 | ABH, Page 1 | 0000\_0000 b |

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

| **Bit** | **Name** | **Description** |
| --- | --- | --- |
| [7:6] | **SPEED1** | **Analog Comparator 1 Speed Control**  00 = slow speed, propagation delay : 4.5us, 1.2uA (typ.).  01 = slow+ speed, propagation delay : 2.0us, 3uA (typ.).  10 = fast speed, propagation delay : 0.6us, 10uA (typ.).  11 = fast+ speed, propagation delay : 0.2us, 75uA (typ.). |
| [5] | **POE1** | **Analog Comparator 1 Polarity Output Enable**  0 = ACMP1 output directly.  1 = ACMP1 output inversely. |
| [4] | **POE0** | **Analog Comparator 0 Polarity Output Enable**  0 = ACMP0 outputs directly.  1 = ACMP0 outputs inversely. |
| [3:2] | **SPEED0** | **Analog Comparator 0 Speed Control**  00 = slow speed, propagation delay : 4.5us, 1.2uA (typ.).  01 = slow+ speed, propagation delay : 2.0us, 3uA (typ.).  10 = fast speed, propagation delay : 0.6us, 10uA (typ.).  11 = fast+ speed, propagation delay : 0.2us, 75uA (typ.). |
| [1] | **-** | **Reserved** |
| [0] | **CRVEN** | **CRV Enable Bit**  0 = CRV Disabled.  1 = CRV Enabled. |

##### ACMPSR – Analog Comparator Status Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| ACMPSR | D4H, Page 0 | 0000\_0000 b |

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

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

##### ACMPVREF – ACMP Reference Voltage Control Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| ACMPVREF | D5H, Page 0 | 0000\_0000 b |

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

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

## PDMA Controller (PDMA)

### Overview

The MUG51TBAE provides peripheral direct memory access (PDMA) controller. The PDMA controller is used to provide high-speed data transfer between memory and peripherals or between memory and memory. The PDMA controller can transfer data from one address to another without CPU intervention. This has the benefit of reducing the workload of CPU and keeps CPU resources free for other applications.

### Features

* Supports transfer data width of 8 bits
* Supports software and SPI and SMC/UART request
* Supports source and destination address increment size can be byte
* Supports transfer done and half done interrupt
* Supports using PDMA to write data to perform CRC operation

### Block Diagram

|  |
| --- |
|  |

Figure 6.14‑1 PDMA Interface Diagram

### Functional Description

|  |
| --- |
|  |

Figure 6.14‑2 PDMA Controller Block Diagram

#### Operating Modes

Each PDMA channel behavior is not pre-defined, user must configure the channel service settings of PSSEL[3:0] registers before starting the related PDMA channel operation.

User must set EN DMAnCR[0] bit to enable PDMA channel. Then write a valid source address to the DMAnMA and DMAnBAH[3:0] register, a destination address to the MTMnDA and DMAnBAH[7:4]register if use memory to memory, and a transfer count to the DMAnCNT register. Next, trigger the RUN DMAnCR[1]. If the source address and destination are not in wrap around mode, the PDMA will continue the transfer until DMAnCCNT counts down to 0. In wrap around mode, when DMAnCCNT counts down to 0, the PDMA will reload DMAnCCNT and work around until user clears EN DMAnCR[0] bit to disable PDMA channel.

A programing sequence example is described below.

* + - * 1. SPI peripheral to XRAM memory

Configure DMAnCR register to set EN DMAnCR[0] bit to enable PDMA channel.

Set PSSEL[3:0] = 0001 SPI0 RX (, 0011 SPI1 RX, 0101 SPI0 TX or 0111 SPI1 TX) (DMAnCR (n-1~2)) register to configure the channel service setting.

Set DMAnMA/DMAnBAH[3:0] registers to configure destination address.

Set DMAnCNT register to configure PDMA transfer count.

Set HIE/FIE DMAnCR[3]/[2] to enable PDMA transfer done interrupt (optional) then enable global interrupt.

Set RUN DMAnCR[0] bit to enable PDMA transfer.

Write “0” to HDONE and FDONE DMAnTSR[1][0] bit to clear interrupt flag if PDMA transfer done interrupt is generated.

Set RUN DMAnCR[0] bit to enable next PDMA transfer.

If an error occurs during the PDMA operation, the channel operation stops until user clears the error condition, , and then clears the EN DMAnCR[0] bit to disable the PDMA channel, then sets EN DMAnCR[0] bit and RUN DMAnCR[1] bit to start operation again.

* + - * 1. SMC/UART peripheral to XRAM memory

Configure DMAnCR register to set EN DMAnCR[0] bit to enable PDMA channel.

Set PSSEL[3:0] = 0010 SMC/UART RX ( or 0110 SMC/UART TX) (DMAnCR (n-1~2)) register to configure the channel service setting.

Set DMAnMA/DMAnBAH[3:0] registers to configure destination address.

Set DMAnCNT register to configure PDMA transfer count.

Set HIE/FIE DMAnCR[3]/[2] to enable PDMA transfer done interrupt (optional) then enable global interrupt.

Set RUN DMAnCR[0] bit to enable PDMA transfer.

Write “0” to HDONE and FDONE DMAnTSR[1][0] bit to clear interrupt flag if PDMA transfer done interrupt is generated.

Set RUN DMAnCR[0] bit to enable next PDMA transfer.

If an error occurs during the PDMA operation, the channel operation stops until user clears the error condition, , and then clears the EN DMAnCR[0] bit to disable the PDMA channel, then sets EN DMAnCR[0] bit and RUN DMAnCR[1] bit to start operation again.

* + - * 1. Memory to Memory (XRAM) Transfer

Configure DMAnCR register to set EN DMAnCR[0] bit to enable PDMA channel.

Set PSSEL[3:0] = 0000 (XRAM to XRAM) (DMAnCR (n-1~2)) register to configure the channel service setting.

Set DMAnMA/DMAnBAH[3:0] registers to configure source address.

Set DMAnDA/DMAnBAH[7:4] registers to configure destination address.

Set DMAnCNT register to configure PDMA transfer count.

Set HIE/FIE DMAnCR[3]/[2] to enable PDMA transfer done interrupt (optional) then enable global interrupt.

Set RUN DMAnCR[0] bit to enable PDMA transfer.

Write “0” to HDONE and FDONE DMAnTSR[1][0] bit to clear interrupt flag if PDMA transfer done interrupt is generated.

Set RUN DMAnCR[0] bit to enable next PDMA transfer.

If an error occurs during the PDMA operation, the channel operation stops until user clears the error condition and then clears the EN DMAnCR[0] bit to disable the PDMA channel, then sets EN DMAnCR[0] bit and RUN DMAnCR[1] bit to start operation again.

#### CRC-8 Function for PDMA

|  |
| --- |
|  |

Figure 6.14‑3 CRC-8 Block Diagram

### Register Description

##### DMAnCR – PDMAn Control Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| DMA0CR0 | 92H, Page 0 | 0000\_0000 b |
| DMA1CR0 | EBH, Page 0 | 0000\_0000 b |

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

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

##### DMAnMAL – PDMA XRAM Base Address Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| DMA0MAL | 93H, Page 0 | 0000\_0000 b |
| DMA1MAL | ECH, Page 0 | 0000\_0000 b |

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

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

##### DMAnBAH – PDMAn XRAM Base and Memory to Memory Destination Address High Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| DMA0BAH | F6H, Page 0 | 0000\_0000 b |
| DMA1BAH | FDH, Page 0 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **MDAH[3:0]** | | | | **MAH[3:0]** | | | |
| R/W | | | | R/W | | | |

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

##### DMAnCNT – PDMA Transfer Count

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| DMA0CNT | 94H, Page 0 | 0000\_0000 b |
| DMA1CNT | EDH, Page 0 | 0000\_0000 b |

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

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

##### DMAnCCNT – PDMA Current Transfer Count

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| DMA0CCNT | 95H, Page 0 | 0000\_0000 b |
| DMA1CCNT | EEH, Page 0 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **CCNT[7:0]** | | | | | | | |
| R | | | | | | | |

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

##### DMAnTSR – PDMAn Transfer Status Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| DMA0TSR | E9H, Page 0 | 0000\_0000 b |
| DMA1TSR | F1H, Page 0 | 0000\_0000 b |

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

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

##### MTMnDAL– Memory to Memory Destination Address Low Byte

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| MTM0DAL | EAH, Page 0 | 0000\_0000 b |
| MTM1DAL | F2H, Page 0 | 0000\_0000 b |

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

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

##### DMAnCR1 – PDMAn Control 1 Register

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| DMA0CR1 | 8AH, Page 3 | 0000\_0000 b |
| DMA1CR1 | 8BH, Page 3 | 0000\_0000 b |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **-** | **-** | **-** | **-** | **XOROUT** | **REFOUT** | **REFIN** | **CRCEN** |
| - | - | - | - | R/W | R/W | R/W | R/W |

| Bit | Name | Description |
| --- | --- | --- |
| [7:4] | **-** | Reserved |
| [3] | **XOROUT** | **PDMA CRC OUT Reflect Enable Bit**  0 = CRC OUT exclusive-ored Disabled when PDMA is running.  1 = CRC OUT exclusive-ored Enabled when PDMA is running, the final value is exclusive-ored with 0x55 . |
| [2] | **REFOUT** | **PDMA CRC OUT Reflect Enable Bit**  0 = CRC OUT reflect Disabled when PDMA is running.  1 = CRC OUT reflect Enabled when PDMA is running, the output data will be bit order revised. |
| [1] | **REFIN** | **PDMA CRC IN Reflect Enable Bit**  0 = CRC IN reflect Disabled when PDMA is running.  1 = CRC IN reflect Enabled when PDMA is running, the input data will be bit order revised. |
| [0] | **CRCEN** | **PDMA CRC Checksum Enable Bit**  0 = CRC checksum Disabled when PDMA is running, DMAnCRC[7:0] is set to 0x00.  1 = CRC checksum Enabled when PDMA is running. |

##### DMAnCRC – PDMA CRC Checksum

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| DMA0CRC | 92H, Page 3 | 0000\_0000 b |
| DMA1CRC | 93H, Page 3 | 0000\_0000 b |

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

| Bit | Name | Description |
| --- | --- | --- |
| [7:0] | **CRC[7:0]** | **PDMA CRC Checksum**  The checksum of the Cyclic Redundancy Check (CRC-8) calculation  The CRC-8 polynomial is below  CRC-8: X8 + X2 + X + 1 |

##### DMAnSEED – PDMA CRC SEED

|  |  |  |
| --- | --- | --- |
| **Register** | **SFR Address** | **Reset Value** |
| DMA0SEED | 9AH, Page 3 | 0000\_0000 b |
| DMA1SEED | 9BH, Page 3 | 0000\_0000 b |

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

| Bit | Name | Description |
| --- | --- | --- |
| [7:0] | **SEED[7:0]** | **PDMA CRC SEED**  The seed of the Cyclic Redundancy Check (CRC-8) calculation  The CRC-8 polynomial is below  CRC-8: X8 + X2 + X + 1 |

Note :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | XOROUT | REFOUT | REFIN | SEED |
| **CRC-8** | 0 | 0 | 0 | 0x00 |
| **CRC-8/ITU** | 1 | 0 | 0 | 0x00 |
| **CRC-8/ROHC** | 0 | 1 | 1 | 0xFF |

## Instruction Set

### Instruction Set And Addressing Modes

The MUG51TBAE 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 MUG51TBAE uses one clock per machine-cycle. It leads to performance improvement of rate 8.1 (in terms of MIPS) with respect to traditional 12T 80C51 device working at the same clock frequency. However, the real speed improvement seen in any system will depend on the instruction mix.

All instructions are coded within an 8-bit field called an OPCODE. This single byte should be fetched from Program Memory. The OPCODE is decoded by the CPU. It determines what action the CPU will take and whether more operation data is needed from memory. If no other data is needed, then only one byte was required. Thus the instruction is called a one byte instruction. In some cases, more data is needed, which is two or three byte instructions.

Following lists all instructions in detail. The note of the instruction set and addressing modes are shown below.

|  |  |
| --- | --- |
| Rn (N = 0~7) | Register R0 To R7 Of The Currently Selected Register Bank. |
| Direct | 8-bit internal data location’s address. It could be an internal data RAM location (00H to 7FH) or an SFR (80H to FFH). |
| @RI (I = 0, 1) | 8-bit internal data RAM location (00H to FFH) addressed indirectly through register R0 or R1. |
| #data | 8-bit constant included in the instruction. |
| #data16 | 16-bit constant included in the instruction. |
| Addr16 | 16-bit destination address. Used by LCALL and LJMP. A branch can be any-where within the Program Memory address space. |
| Addr11 | 11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2K-Byte page of Program Memory as the first byte of the following instruction. |
| Rel | Signed (2’s complement) 8-bit offset Byte. Used by SJMP and all conditional branches. The range is -128 to +127 bytes relative to first byte of the following instruction. |
| Bit | Direct addressed bit in internal data RAM or SFR. |

Table 6.15‑1 Instruction Set And Addressing Modes

| **Instruction** | **CY** | **OV** | **AC** | **Instruction** | **CY** | **OV** | **AC** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ADD | X[1] | X | X | CLR C | 0 |  |  |
| ADDC | X | X | X | CPL C | X |  |  |
| SUBB | X | X | X | ANL C, bit | X |  |  |
| MUL | 0 | X |  | ANL C, /bit | X |  |  |
| DIV | 0 | X |  | ORL C, bit | X |  |  |
| DA A | X |  |  | ORL C, /bit | X |  |  |
| RRC A | X |  |  | MOV C, bit | X |  |  |
| RLC A | X |  |  | CJNE | X |  |  |
| SETB C | 1 |  |  |  |  |  |  |
| **Note:** X indicates the modification depends on the result of the instruction. | | | | | | | |

Table 6.15‑2 Instructions Affect Flag Settings

### Read-Modify-Write Instructions

Instructions that read a byte from SFR or internal RAM, modify it, and rewrite it back, are called “Read-Modify-Write” instructions. When the destination is an I/O port or a port bit, these instructions read the internal output latch rather than the external pin state. This kind of instructions read the port SFR value, modify it and write back to the port SFR. All “Read-Modify-Write” instructions are listed as follows.

Instruction Description

ANL Logical AND. (ANL direct, A and ANL direct, #data)

ORL Logical OR. (ORL direct, A and ORL direct, #data)

XRL Logical exclusive OR. (XRL direct, A and XRL direct, #data)

JBC Jump if bit = 1 and clear it. (JBC bit, rel)

CPL Complement bit. (CPL bit)

INC Increment. (INC direct)

DEC Decrement. (DEC direct)

DJNZ Decrement and jump if not zero. (DJNZ direct, rel)

MOV bit, C Move carry to bit. (MOV bit, C)

CLR bit Clear bit. (CLR bit)

SETB bit Set bit. (SETB bit)

The last three seem not obviously “Read-Modify-Write” instructions but actually they are. They read the entire port latch value, modify the changed bit, and then write the new value back to the port latch.

### Instruction Set

| **Instruction** | **OPCODE** | **Bytes** | **Clock Cycles** | **MUG51TBAE VS. Tradition 80C51 Speed Ratio** |
| --- | --- | --- | --- | --- |
| NOP | 00 | 1 | 1 | 12 |
| ADD A, Rn | 28~2F | 1 | 2 | 6 |
| ADD A, direct | 25 | 2 | 3 | 4 |
| ADD A, @Ri | 26, 27 | 1 | 4 | 3 |
| ADD A, #data | 24 | 2 | 2 | 6 |
| ADDC A, Rn | 38~3F | 1 | 2 | 6 |
| ADDC A, direct | 35 | 2 | 3 | 4 |
| ADDC A, @Ri | 36, 37 | 1 | 4 | 3 |
| ADDC A, #data | 34 | 2 | 2 | 6 |
| SUBB A, Rn | 98~9F | 1 | 2 | 6 |
| SUBB A, direct | 95 | 2 | 3 | 4 |
| SUBB A, @Ri | 96, 97 | 1 | 4 | 3 |
| SUBB A, #data | 94 | 2 | 2 | 6 |
| 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 |
| 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 |
| 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, direct, rel | B5 | 3 | 5 | 4.8 |
| CJNE A, #data, rel | B4 | 3 | 4 | 6 |
| CJNE Rn, #data, rel | B8~BF | 3 | 4 | 6 |
| CJNE @Ri, #data, rel | B6, B7 | 3 | 6 | 4 |
| DJNZ Rn, rel | D8~DF | 2 | 4 | 6 |
| DJNZ direct, rel | D5 | 3 | 5 | 4.8 |
| **Note:**  1. The MUG51TBAE does not have external memory bus. MOVX instructions are used to access internal XRAM.  2. The most three significant bits in the 11-bit address [A10:A8] decide the ACALL hex code. The code will be [A10, A9, A8, 1, 0, 0, 0, 1].  3. The most three significant bits in the 11-bit address [A10:A8] decide the AJMP hex code. The code will be [A10, A9, A8, 0, 0, 0, 0, 1]. | | | | |

Table 6.15‑3 Instruction Set

# APPLICATION CIRCUIT

## Power Supply Scheme

|  |
| --- |
|  |

Figure 7.1 MUG51TBAE Power Supply Circuit

## Peripheral Application Scheme

|  |
| --- |
|  |
| **Note 1:** It is recommended to use 100 kΩ pull-up resistor on both ICE\_DAT and ICE\_CLK pin.  **Note 2:** It is recommended to use 10 kΩ pull-up resistor and 10 uF capacitor on nRESET pin.  **Note 3:** It is suggest add 100ohm series resistor between ICE\_DAT/ICE\_CLK and writer pin to filter the disturb of noise on the circuit.  **Note 4 :** the LDO capacitor value tolerance is 20% |

Figure 7.2 MUG51TBAE Peripheral Interface Circuit

# ELECTRICAL CHARACTERISTICS

Please refer to the relative Datasheet for detailed information about the MUG51TBAE electrical characteristics.

# PACKAGE DIMENSIONS

Package is Halogen-free, RoHS-compliant and TSCA-compliant.

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

|  |
| --- |
|  |

Figure 9.1‑1 QFN-33 Package Dimension

# ABBREVIATIONS

## Abbreviations

|  |  |
| --- | --- |
| **Acronym** | **Description** |
| ACMP | Analog Comparator |
| ADC | Analog-to-Digital Converter |
| BOD | Brown-out Detection |
| GPIO | General-Purpose Input/Output |
| FSYS | Frequency of system clock |
| I2C | Inter-Integrated Circuit Bus |
| IAP | In Application Programming |
| ICP | In Circuit Programming |
| ISP | In System Programming |
| LDO | Low Dropout Regulator |
| LIRC | Internal low speed RC oscillator (LIRC) |
| LVR | Low Voltage $eset |
| MIRC | Internal Median Speed RC Oscillator |
| PDMA | Peripheral Direct Memory Access |
| PINV | GPIO Internal Inverter output enable |
| PBUF | GPIO Intenral Buffer output enable |
| 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

# REVISION HISTORY

|  |  |  |
| --- | --- | --- |
| Date | Revision | Description |
| 2022.07.25 | 1.00 | * Initial version. |

**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.**

