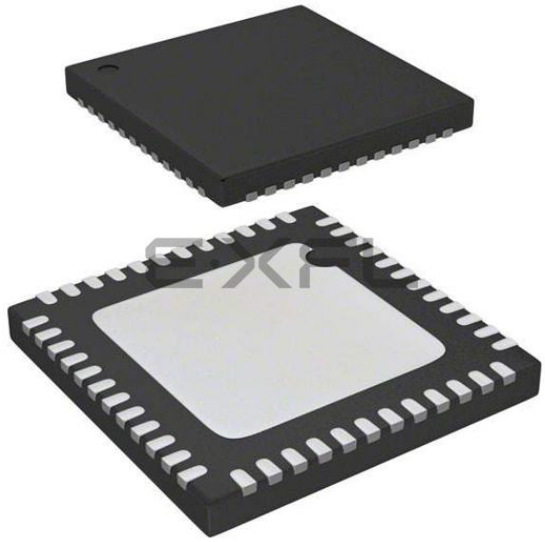

**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | 8051 |
| Core Size | 8-Bit |
| Speed | 50MHz |
| Connectivity | EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART |
| Peripherals | POR, PWM, Temp Sensor, WDT |
| Number of I/O | 40 |
| Program Memory Size | 96KB (96K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 8.25K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 5.25V |
| Data Converters | A/D 32x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 48-VFQFN Exposed Pad |
| Supplier Device Package | 48-QFN (7x7) |
| Purchase URL | https://www.e-xfl.com/product-detail/silicon-labs/c8051f585-im |

## 11.4. Serial Number Special Function Registers (SFRs)

The C8051F58x/F59x devices include four SFRs, SN0 through SN3, that are pre-programmed during production with a unique, 32-bit serial number. The serial number provides a unique identification number for each device and can be read from the application firmware. If the serial number is not used in the application, these four registers can be used as general purpose SFRs.

### SFR Definition 11.7. SNn: Serial Number n

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | SERNUMn[7:0] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | Varies—Unique 32-bit value | | | | | | | |

SFR Addresses: SN0 = 0xF9; SN1 = 0xFA; SN2 = 0xFB; SN3 = 0xFC; SFR Page = 0x0F;

| Bit | Name | Function |
|---|---|---|
| 7:0 | SERNUMn[7:0] | **Serial Number Bits.**<br>The four serial number registers form a 32-bit serial number, with SN3 as the most significant byte and SN0 as the least significant byte. |

## SFR Definition 13.2. SFRPAGE: SFR Page

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | SFRPAGE[7:0] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Address = 0xA7; SFR Page = All Pages

| Bit | Name | Function |
|---|---|---|
| 7:0 | SFRPAGE[7:0] | **SFR Page Bits.**<br>Represents the SFR Page the C8051 core uses when reading or modifying SFRs.<br>Write: Sets the SFR Page.<br>Read: Byte is the SFR page the C8051 core is using.<br>When enabled in the SFR Page Control Register (SFR0CN), the C8051 core will automatically switch to the SFR Page that contains the SFRs of the corresponding peripheral/function that caused the interrupt, and return to the previous SFR page upon return from interrupt (unless SFR Stack was altered before a returning from the interrupt). SFRPAGE is the top byte of the SFR Page Stack, and push/pop events of this stack are caused by interrupts (and not by reading/writing to the SFRPAGE register) |

**Table 13.3. Special Function Registers (Continued)**

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

| Register | Address | Description | Page |
|---|---|---|---|
| **FLSCL** | 0xB6 | Flash Scale | 147 |
| **IE** | 0xA8 | Interrupt Enable | 130 |
| **IP** | 0xB8 | Interrupt Priority | 131 |
| **IT01CF** | 0xE4 | INT0/INT1 Configuration | 137 |
| **LIN0ADR** | 0xD3 | LIN0 Address | 221 |
| **LIN0CF** | 0xC9 | LIN0 Configuration | 221 |
| **LIN0DAT** | 0xD2 | LIN0 Data | 222 |
| **OSCICN** | 0xA1 | Internal Oscillator Control | 179 |
| **OSCICRS** | 0xA2 | Internal Oscillator Coarse Control | 180 |
| **OSCIFIN** | 0x9E | Internal Oscillator Fine Calibration | 180 |
| **OSCXCN** | 0x9F | External Oscillator Control | 184 |
| **P0** | 0x80 | Port 0 Latch | 204 |
| **P0MASK** | 0xF2 | Port 0 Mask Configuration | 200 |
| **P0MAT** | 0xF1 | Port 0 Match Configuration | 200 |
| **P0MDIN** | 0xF1 | Port 0 Input Mode Configuration | 205 |
| **P0MDOUT** | 0xA4 | Port 0 Output Mode Configuration | 205 |
| **P0SKIP** | 0xD4 | Port 0 Skip | 206 |
| **P1** | 0x90 | Port 1 Latch | 206 |
| **P1MASK** | 0xF4 | Port 1 Mask Configuration | 201 |
| **P1MAT** | 0xF3 | Port 1 Match Configuration | 201 |
| **P1MDIN** | 0xF2 | Port 1 Input Mode Configuration | 207 |
| **P1MDOUT** | 0xA5 | Port 1 Output Mode Configuration | 207 |
| **P1SKIP** | 0xD5 | Port 1 Skip | 208 |
| **P2** | 0xA0 | Port 2 Latch | 208 |
| **P2MASK** | 0xB2 | Port 2 Mask Configuration | 202 |
| **P2MAT** | 0xB1 | Port 2 Match Configuration | 202 |
| **P2MDIN** | 0xF3 | Port 2 Input Mode Configuration | 209 |
| **P2MDOUT** | 0xA6 | Port 2 Output Mode Configuration | 209 |
| **P2SKIP** | 0xD6 | Port 2 Skip | 210 |
| **P3** | 0xB0 | Port 3 Latch | 210 |
| **P3MASK** | 0xAF | Port 3 Mask Configuration | 203 |
| **P3MAT** | 0xAE | Port 3 Match Configuration | 203 |
| **P3MDIN** | 0xF4 | Port 3 Input Mode Configuration | 211 |
| **P3MDOUT** | 0xAE | Port 3 Output Mode Configuration | 211 |
| **P3SKIP** | 0xD7 | Port 3 Skip | 212 |
| **P4** | 0xB5 | Port 4 Latch | 212 |

**Table 13.3. Special Function Registers (Continued)**

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

| Register | Address | Description | Page |
|---|---|---|---|
| **PCA1CPL11** | 0xCE | PCA1 Capture 11 Low | 350 |
| **PCA1CPM6** | 0xDA | PCA1 Module 6 Mode Register | 348 |
| **PCA1CPM7** | 0xDB | PCA1 Module 7 Mode Register | 348 |
| **PCA1CPM8** | 0xDC | PCA1 Module 8 Mode Register | 348 |
| **PCA1CPM9** | 0xDD | PCA1 Module 9 Mode Register | 348 |
| **PCA1CPM10** | 0xDE | PCA1 Module 10 Mode Register | 348 |
| **PCA1CPM11** | 0xDF | PCA1 Module 11 Mode Register | 348 |
| **PCA1H** | 0xFA | PCA1 Counter High | 349 |
| **PCA1L** | 0xF9 | PCA1 Counter Low | 349 |
| **PCA1MD** | 0xD9 | PCA1 Mode | 346 |
| **PCA1PWM** | 0xDA | PCA1 PWM Configuration | 347 |
| **PCON** | 0x87 | Power Control | 151 |
| **PSBANK** | 0xF5 | Program Space Bank Select | 104 |
| **PSCTL** | 0x8F | Program Store R/W Control | 145 |
| **PSW** | 0xD0 | Program Status Word | 100 |
| **REF0CN** | 0xD1 | Voltage Reference Control | 76 |
| **REG0CN** | 0xD1 | Voltage Regulator Control | 90 |
| **RSTSRC** | 0xEF | Reset Source Configuration/Status | 157 |
| **SBCON0** | 0xAB | UART0 Baud Rate Generator Control | 263 |
| **SBRLH0** | 0xAD | UART0 Baud Rate Reload High Byte | 264 |
| **SBRLL0** | 0xAC | UART0 Baud Rate Reload Low Byte | 264 |
| **SBUF0** | 0x99 | UART0 Data Buffer | 263 |
| **SCON0** | 0x98 | UART0 Control | 261 |
| **SBUF1** | 0x99 | UART1 Data Buffer | 270 |
| **SCON1** | 0x98 | UART1 Control | 269 |
| **SFR0CN** | 0x84 | SFR Page Control | 113 |
| **SFRLAST** | 0x86 | SFR Stack Last Page | 116 |
| **SFRNEXT** | 0x85 | SFR Stack Next Page | 115 |
| **SFRPAGE** | 0xA7 | SFR Page Select | 114 |
| **SMB0CF** | 0xC1 | SMBus0 Configuration | 245 |
| **SMB0CN** | 0xC0 | SMBus0 Control | 247 |
| **SMB0DAT** | 0xC2 | SMBus0 Data | 249 |
| **SMOD0** | 0xA9 | UART0 Mode | 262 |
| **SN0** | 0xF9 | Serial Number 0 | 101 |
| **SN1** | 0xFA | Serial Number 1 | 101 |
| **SN2** | 0xFB | Serial Number 2 | 101 |

### 14.1.1. Interrupt Priorities

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. Each interrupt has an associated interrupt priority bit in an SFR (IE, EIP1, or EIP2) used to configure its priority level. Low priority is the default. If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate, given in Table 14.1.

### 14.1.2. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 5 system clock cycles: 1 clock cycle to detect the interrupt and 4 clock cycles to complete the LCALL to the ISR. If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs when the CPU is performing an RETI instruction followed by a DIV as the next instruction. In this case, the response time is 18 system clock cycles: 1 clock cycle to detect the interrupt, 5 clock cycles to execute the RETI, 8 clock cycles to complete the DIV instruction and 4 clock cycles to execute the LCALL to the ISR. If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the RETI and following instruction.

## 15.4. Flash Write and Erase Guidelines

Any system which contains routines which write or erase Flash memory from software involves some risk that the write or erase routines will execute unintentionally if the CPU is operating outside its specified operating range of $V_{DD}$, system clock frequency, or temperature. This accidental execution of Flash modifying code can result in alteration of Flash memory contents causing a system failure that is only recoverable by re-Flashing the code in the device.

The following guidelines are recommended for any system which contains routines which write or erase Flash from code.

### 15.4.1. $V_{DD}$ Maintenance and the $V_{DD}$ monitor

1. If the system power supply is subject to voltage or current "spikes," add sufficient transient protection devices to the power supply to ensure that the supply voltages listed in the Absolute Maximum Ratings table are not exceeded.
2. Make certain that the minimum $V_{REGIN}$ rise time specification of 1 ms is met. If the system cannot meet this rise time specification, then add an external $V_{DD}$ brownout circuit to the $\overline{RST}$ pin of the device that holds the device in reset until $V_{DD}$ reaches the minimum threshold and re-asserts $\overline{RST}$ if $V_{DD}$ drops below the minimum threshold.
3. Enable the on-chip $V_{DD}$ monitor to the high setting and enable the $V_{DD}$ monitor as a reset source as early in code as possible. This should be the first set of instructions executed after the Reset Vector. For C-based systems, this will involve modifying the startup code added by the C compiler. See your compiler documentation for more details. Make certain that there are no delays in software between enabling the $V_{DD}$ monitor and enabling the $V_{DD}$ monitor as a reset source. Code examples showing this can be found in “AN201: Writing to Flash from Firmware", available from the Silicon Laboratories web site.
4. As an added precaution, explicitly enable the $V_{DD}$ monitor and enable the $V_{DD}$ monitor as a reset source inside the functions that write and erase Flash memory. The $V_{DD}$ monitor enable instructions should be placed just after the instruction to set PSWE to a 1, but before the Flash write or erase operation instruction.

**Note:** The output of the internal voltage regulator is calibrated by the MCU immediately after any reset event. The output of the un-calibrated internal regulator could be below the high threshold setting of the VDD Monitor. If this is the case, and the MCU receives a non-power on reset (POR) when the VDD Monitor is set to the high threshold setting, the MCU will remain in reset until a POR occurs (i.e. VDD Monitor will keep the device in reset). A POR will force the VDD Monitor to the low threshold setting, which is guaranteed to be below the un-calibrated output of the internal regulator. The device will then exit reset and resume normal operation. It is for this reason Silicon Labs strongly recommends that the VDD Monitor is always left in the low threshold setting (i.e. default value upon POR). When programming the Flash in-system, the VDD Monitor must be set to the high threshold setting. To prevent this issue from happening and ensure the highest system reliability, firmware can change the $V_{DD}$ Monitor high threshold, and the system must use an external supply monitor. For instructions on how to do this, see “Reprogramming the VDD Monitor High Threshold” on page 138.

5. Make certain that all writes to the RSTSRC (Reset Sources) register use direct assignment operators and explicitly DO NOT use the bit-wise operators (such as AND or OR). For example, "RSTSRC = 0x02" is correct. "RSTSRC |= 0x02" is incorrect.
6. Make certain that all writes to the RSTSRC register explicitly set the PORSF bit to a 1. Areas to check are initialization code which enables other reset sources, such as the Missing Clock Detector or Comparator, for example, and instructions which force a Software Reset. A global search on "RSTSRC" can quickly verify this.

## SFR Definition 19.5. CLKMUL: Clock Multiplier

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | MULEN | MULINIT | MULRDY | MULDIV[2:0] | | | MULSEL[1:0] | |
| Type | R/W | R/W | R | R/W | | | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Address = 0x97; SFR Page = 0x0F;

| Bit | Name | Function |
|---|---|---|
| 7 | MULEN | **Clock Multiplier Enable.**<br>0: Clock Multiplier disabled.<br>1: Clock Multiplier enabled. |
| 6 | MULINIT | **Clock Multiplier Initialize.**<br>This bit is 0 when the Clock Multiplier is enabled. Once enabled, writing a 1 to this bit will initialize the Clock Multiplier. The MULRDY bit reads 1 when the Clock Multiplier is stabilized. |
| 5 | MULRDY | **Clock Multiplier Ready.**<br>0: Clock Multiplier is not ready.<br>1: Clock Multiplier is ready (PLL is locked). |
| 4:2 | MULDIV[2:0] | **Clock Multiplier Output Scaling Factor.**<br>000: Clock Multiplier Output scaled by a factor of 1.<br>001: Clock Multiplier Output scaled by a factor of 1.<br>010: Clock Multiplier Output scaled by a factor of 1.<br>011: Clock Multiplier Output scaled by a factor of 2/3*.<br>100: Clock Multiplier Output scaled by a factor of 2/4 (1/2).<br>101: Clock Multiplier Output scaled by a factor of 2/5*.<br>110: Clock Multiplier Output scaled by a factor of 2/6 (1/3).<br>111: Clock Multiplier Output scaled by a factor of 2/7*.<br>***Note:** The Clock Multiplier output duty cycle is not 50% for these settings. |
| 1:0 | MULSEL[1:0] | **Clock Multiplier Input Select.**<br>These bits select the clock supplied to the Clock Multiplier |

| MULSEL[1:0] | Selected Input Clock | Clock Multiplier Output for MULDIV[2:0] = 000b |
|---|---|---|
| 00 | Internal Oscillator | Internal Oscillator x 2 |
| 01 | External Oscillator | External Oscillator x 2 |
| 10 | Internal Oscillator | Internal Oscillator x 4 |
| 11 | External Oscillator | External Oscillator x 4 |

**Notes:** The maximum system clock is 50 MHz, and so the Clock Multiplier output should be scaled accordingly. If Internal Oscillator x 2 or External Oscillator x 2 is selected using the MULSEL bits, MULDIV[2:0] is ignored.

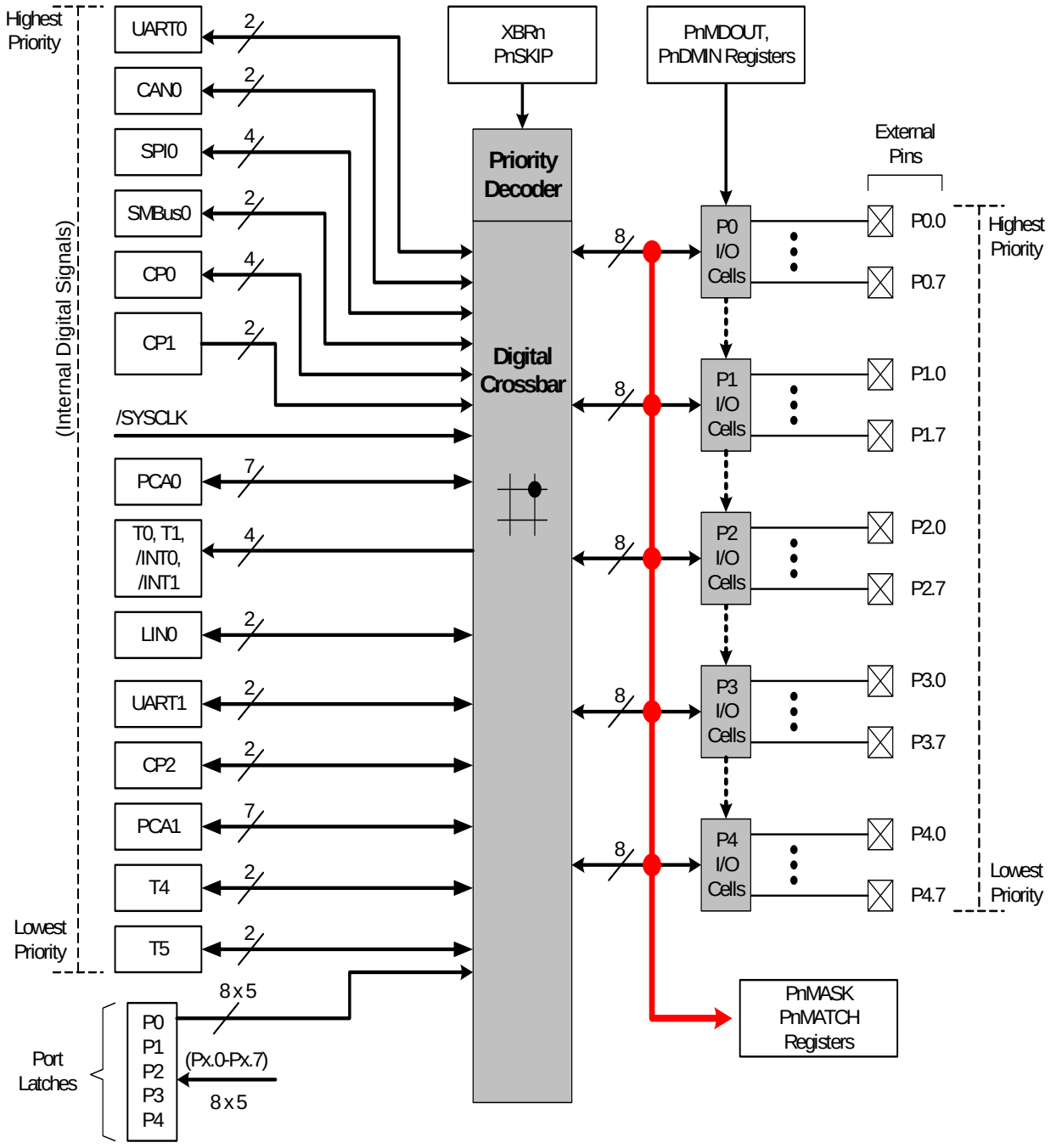


**Figure 20.1. Port I/O Functional Block Diagram**

## SFR Definition 20.4. XBR3: Port I/O Crossbar Register 3

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | T5EXE | T5E | T4EXE | T4E | ECI1E | PCA1ME[2:0] | | |
| Type | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Address = 0xC6; SFR Page = 0x0F

| Bit | Name | Function |
|---|---|---|
| 7 | T5EXE | **T5EX Enable.**<br>0: T5EX unavailable at Port pin.<br>1: T5EX routed to Port pin |
| 6 | T5EX | **T5E Enable.**<br>0: T5E unavailable at Port pin.<br>1: T5E routed to Port pin |
| 5 | T4EXE | **T4EX Enable.**<br>0: T4EX unavailable at Port pin.<br>1: T4EX routed to Port pin |
| 4 | T5EX | **T4E Enable.**<br>0: T4E unavailable at Port pin.<br>1: T4E routed to Port pin |
| 3 | ECI1E | **PCA1 External Counter Input Enable.**<br>0: ECI1 unavailable at Port pin.<br>1: ECI1 routed to Port pin. |
| 2:0 | PCA1ME[2:0] | **PCA1 Module I/O Enable Bits.**<br>000: All PCA1 I/O unavailable at Port pins.<br>001: CEX6 routed to Port pin.<br>010: CEX6, CEX7 routed to Port pins.<br>011: CEX6, CEX7, CEX8 routed to Port pins.<br>100: CEX6, CEX7, CEX8, CEX9 routed to Port pins.<br>101: CEX6, CEX7, CEX8, CEX9, CEX10 routed to Port pins.<br>110: CEX6, CEX7, CEX8, CEX9, CEX10, CEX11 routed to Port pins.<br>111: RESERVED |

## SFR Definition 20.28. P3SKIP: Port 3Skip

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | P3SKIP[7:0] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Address = 0xD7; SFR Page = 0x0F

| Bit | Name | Function |
|---|---|---|
| 7:0 | P3SKIP[7:0] | **Port 3 Crossbar Skip Enable Bits.**<br>These bits select Port 3 pins to be skipped by the Crossbar Decoder. Port pins used for analog, special functions or GPIO should be skipped by the Crossbar.<br>0: Corresponding P3.n pin is not skipped by the Crossbar.<br>1: Corresponding P3.n pin is skipped by the Crossbar. |
| **Note:** Port P3.1–P3.7 are only available on the 48-pin and 40-pin packages. | | |

## SFR Definition 20.29. P4: Port 4

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | P4[7:0] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

SFR Address = 0xB5; SFR Page = All Pages

| Bit | Name | Description | Write | Read |
|---|---|---|---|---|
| 7:0 | P4[7:0] | **Port 4 Data.**<br>Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O. | 0: Set output latch to logic LOW.<br>1: Set output latch to logic HIGH. | 0: P4.n Port pin is logic LOW.<br>1: P4.n Port pin is logic HIGH. |
| **Note:** Port 4.0 is only available on the 48-pin and 40-pin packages.; P4.1-P4.7 is only available on the 48-pin packages. | | | | |

## LIN Register Definition 21.7. LIN0ERR: LIN0 Error Register

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | | | | SYNCH | PRTY | TOUT | CHK | BITERR |
| Type | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Indirect Address = 0x0A

| Bit | Name | Function |
|---|---|---|
| 7:5 | Unused | Read = 000b; Write = Don't Care |
| 4 | SYNCH | **Synchronization Error Bit (slave mode only).**<br>0: No error with the SYNCH FIELD has been detected.<br>1: Edges of the SYNCH FIELD are outside of the maximum tolerance. |
| 3 | PRTY | **Parity Error Bit (slave mode only).**<br>0: No parity error has been detected.<br>1: A parity error has been detected. |
| 2 | TOUT | **Timeout Error Bit.**<br>0: A timeout error has not been detected.<br>1: A timeout error has been detected. This error is detected whenever one of the following conditions is met:<br>• The master is expecting data from a slave and the slave does not respond.<br>• The slave is expecting data but no data is transmitted on the bus.<br>• A frame is not finished within the maximum frame length.<br>• The application does not set the DTACK bit (LIN0CTRL.4) or STOP bit (LIN0CTRL.7) until the end of the reception of the first byte after the identifier. |
| 1 | CHK | **Checksum Error Bit.**<br>0: Checksum error has not been detected.<br>1: Checksum error has been detected. |
| 0 | BITERR | **Bit Transmission Error Bit.**<br>0: No error in transmission has been detected.<br>1: The bit value monitored during transmission is different than the bit value sent. |

# 26. Enhanced Serial Peripheral Interface (SPI0)

The Enhanced Serial Peripheral Interface (SPI0) provides access to a flexible, full-duplex synchronous serial bus. SPI0 can operate as a master or slave device in both 3-wire or 4-wire modes, and supports multiple masters and slaves on a single SPI bus. The slave-select (NSS) signal can be configured as an input to select SPI0 in slave mode, or to disable Master Mode operation in a multi-master environment, avoiding contention on the SPI bus when more than one master attempts simultaneous data transfers. NSS can also be configured as a chip-select output in master mode, or disabled for 3-wire operation. Additional general purpose port I/O pins can be used to select multiple slave devices in master mode.

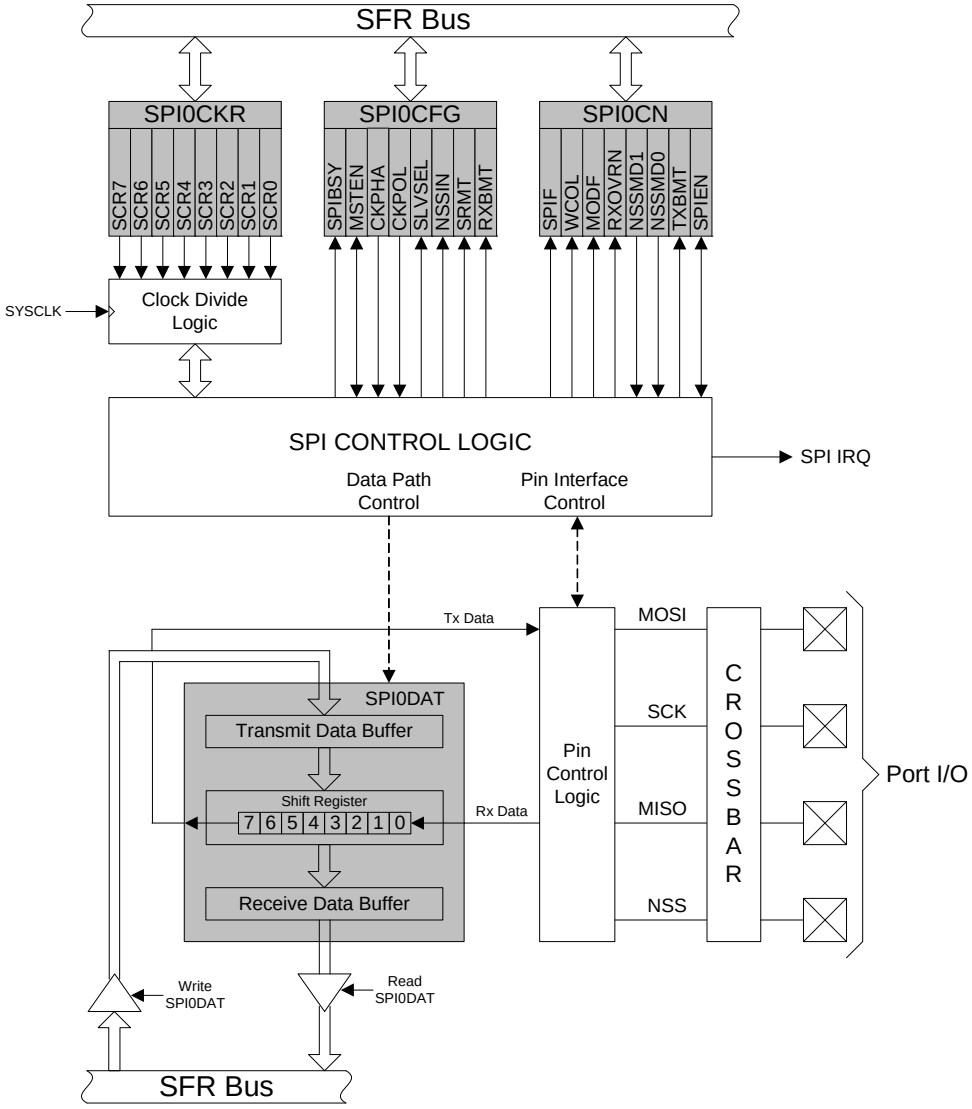


**Figure 26.1. SPI Block Diagram**

## 26.2. SPI0 Master Mode Operation

A SPI master device initiates all data transfers on a SPI bus. SPI0 is placed in master mode by setting the Master Enable flag (MSTEN, SPI0CN.6). Writing a byte of data to the SPI0 data register (SPI0DAT) when in master mode writes to the transmit buffer. If the SPI shift register is empty, the byte in the transmit buffer is moved to the shift register, and a data transfer begins. The SPI0 master immediately shifts out the data serially on the MOSI line while providing the serial clock on SCK. The SPIF (SPI0CN.7) flag is set to logic 1 at the end of the transfer. If interrupts are enabled, an interrupt request is generated when the SPIF flag is set. While the SPI0 master transfers data to a slave on the MOSI line, the addressed SPI slave device simultaneously transfers the contents of its shift register to the SPI master on the MISO line in a full-duplex operation. Therefore, the SPIF flag serves as both a transmit-complete and receive-data-ready flag. The data byte received from the slave is transferred MSB-first into the master's shift register. When a byte is fully shifted into the register, it is moved to the receive buffer where it can be read by the processor by reading SPI0DAT.

When configured as a master, SPI0 can operate in one of three different modes: multi-master mode, 3-wire single-master mode, and 4-wire single-master mode. The default, multi-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In this mode, NSS is an input to the device, and is used to disable the master SPI0 when another master is accessing the bus. When NSS is pulled low in this mode, MSTEN (SPI0CN.6) and SPIEN (SPI0CN.0) are set to 0 to disable the SPI master device, and a Mode Fault is generated (MODF, SPI0CN.5 = 1). Mode Fault will generate an interrupt if enabled. SPI0 must be manually re-enabled in software under these circumstances. In multi-master systems, devices will typically default to being slave devices while they are not acting as the system master device. In multi-master mode, slave devices can be addressed individually (if needed) using general-purpose I/O pins. Figure 26.2 shows a connection diagram between two master devices in multiple-master mode.

3-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. In this mode, NSS is not used, and is not mapped to an external port pin through the crossbar. Any slave devices that must be addressed in this mode should be selected using general-purpose I/O pins. Figure 26.3 shows a connection diagram between a master device in 3-wire master mode and a slave device.

4-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 1. In this mode, NSS is configured as an output pin, and can be used as a slave-select signal for a single SPI device. In this mode, the output value of NSS is controlled (in software) with the bit NSSMD0 (SPI0CN.2). Additional slave devices can be addressed using general-purpose I/O pins. Figure 26.4 shows a connection diagram for a master device in 4-wire master mode and two slave devices.

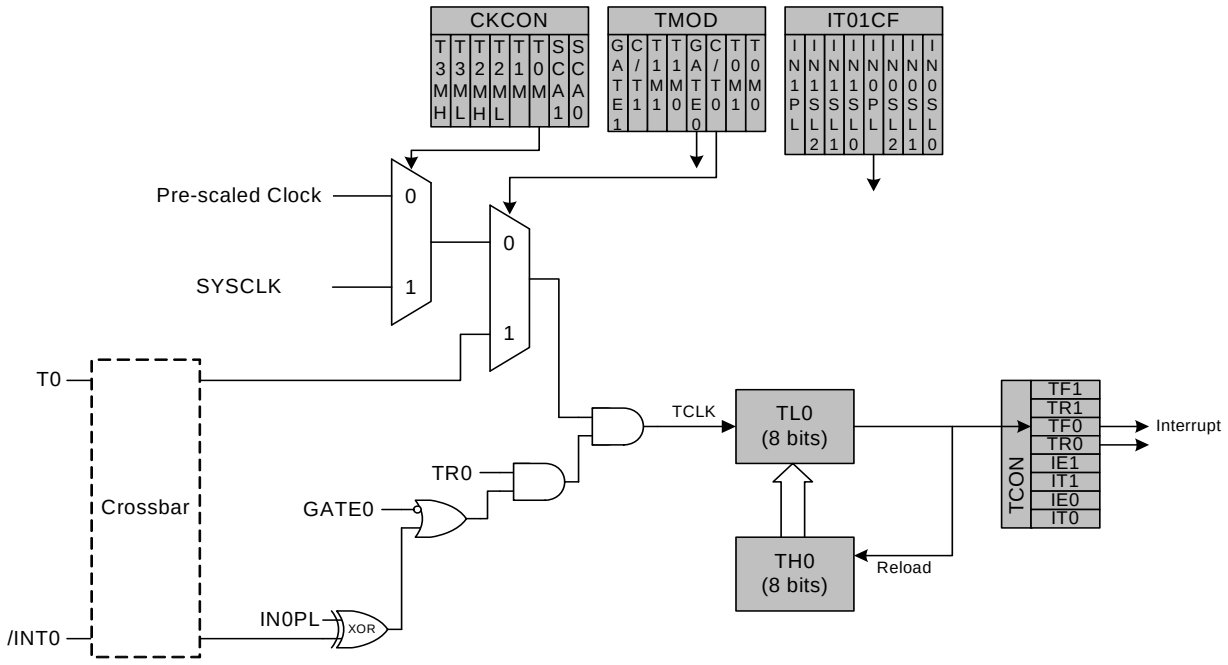


**Figure 27.2. T0 Mode 2 Block Diagram**

### 27.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only)

In Mode 3, Timer 0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. The counter/timer in TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0, C/T0, GATE0 and TF0. TL0 can use either the system clock or an external input signal as its timebase. The TH0 register is restricted to a timer function sourced by the system clock or prescaled clock. TH0 is enabled using the Timer 1 run control bit TR1. TH0 sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

Timer 1 is inactive in Mode 3. When Timer 0 is operating in Mode 3, Timer 1 can be operated in Modes 0, 1 or 2, but cannot be clocked by external signals nor set the TF1 flag and generate an interrupt. However, the Timer 1 overflow can be used to generate baud rates for the SMBus and/or UART, and/or initiate ADC conversions. While Timer 0 is operating in Mode 3, Timer 1 run control is handled through its mode settings. To run Timer 1 while Timer 0 is in Mode 3, set the Timer 1 Mode as 0, 1, or 2. To disable Timer 1, configure it for Mode 3.

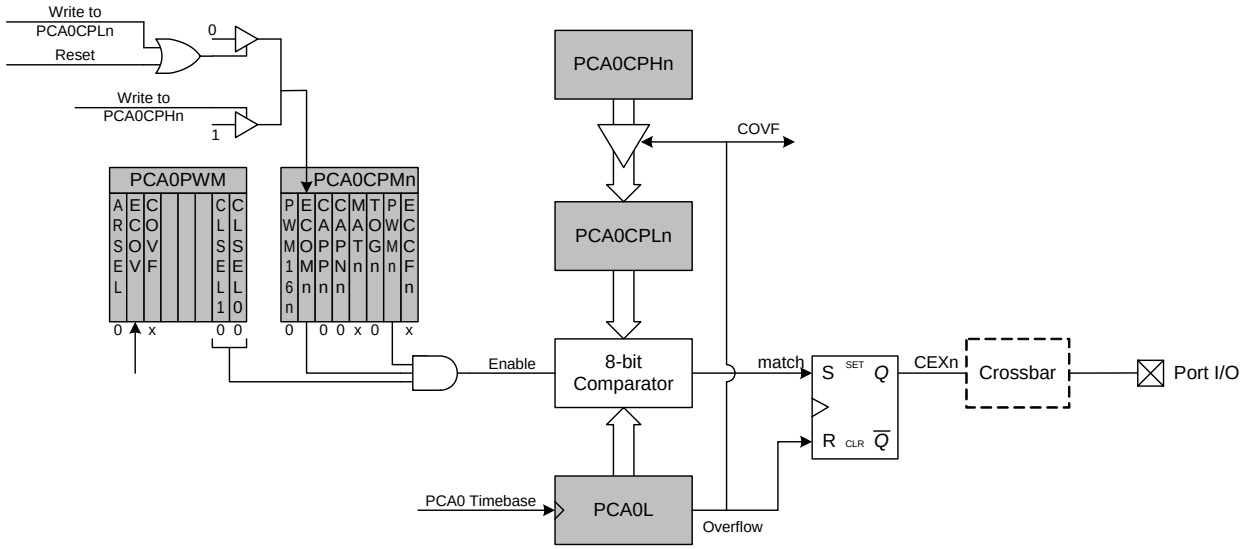


**Figure 28.8. PCA0 8-Bit PWM Mode Diagram**

### 28.3.5.2. 9/10/11-bit Pulse Width Modulator Mode

The duty cycle of the PWM output signal in 9/10/11-bit PWM mode should be varied by writing to an "Auto-Reload" Register, which is dual-mapped into the PCA0CPHn and PCA0CPLn register locations. The data written to define the duty cycle should be right-justified in the registers. The auto-reload registers are accessed (read or written) when the bit ARSEL in PCA0PWM is set to 1. The capture/compare registers are accessed when ARSEL is set to 0.

When the least-significant N bits of the PCA0 counter match the value in the associated module's capture/compare register (PCA0CPn), the output on CEXn is asserted high. When the counter overflows from the Nth bit, CEXn is asserted low (see Figure 28.9). Upon an overflow from the Nth bit, the COVF flag is set, and the value stored in the module's auto-reload register is loaded into the capture/compare register. The value of N is determined by the CLSEL bits in register PCA0PWM.

The 9, 10 or 11-bit PWM mode is selected by setting the ECOMn and PWMn bits in the PCA0CPMn register, and setting the CLSEL bits in register PCA0PWM to the desired cycle length (other than 8-bits). If the MATn bit is set to 1, the CCFn flag for the module will be set each time a comparator match (rising edge) occurs. The COVF flag in PCA0PWM can be used to detect the overflow (falling edge), which will occur every 512 (9-bit), 1024 (10-bit) or 2048 (11-bit) PCA0 clock cycles. The duty cycle for 9/10/11-Bit PWM Mode is given in Equation 28.2, where N is the number of bits in the PWM cycle.

**Important Note About PCA0CPHn and PCA0CPLn Registers**: When writing a 16-bit value to the PCA0CPn registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

$$\text{Duty Cycle} = \frac{(2^N - \text{PCA0CPn})}{2^N}$$

**Equation 28.3. 9, 10, and 11-Bit PWM Duty Cycle**

A 0% duty cycle may be generated by clearing the ECOMn bit to 0.

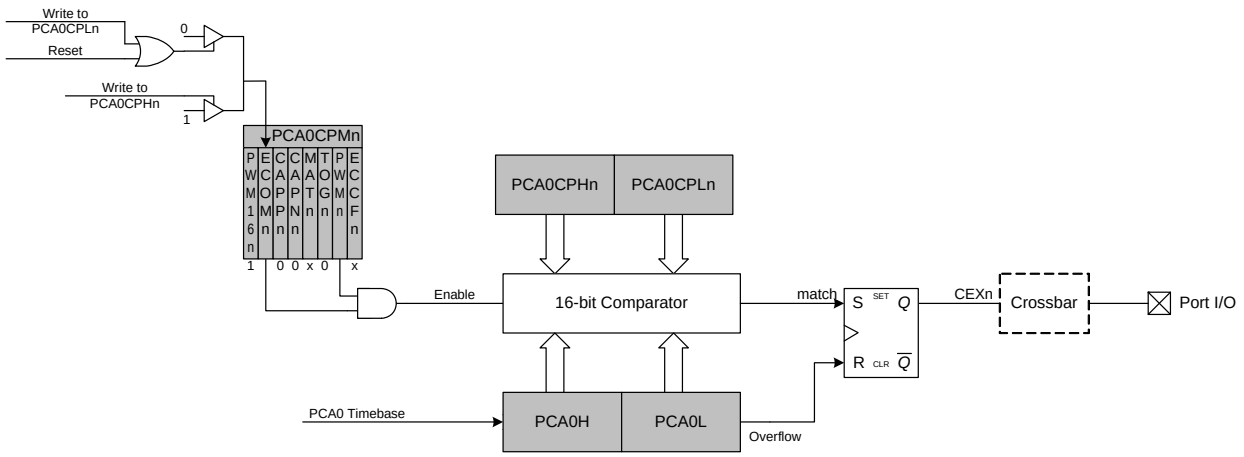


**Figure 28.10. PCA0 16-Bit PWM Mode**

## 28.4. Watchdog Timer Mode

A programmable watchdog timer (WDT) function is available through the PCA0 Module 5. The WDT is used to generate a reset if the time between writes to the WDT update register (PCA0CPH2) exceed a specified limit. The WDT can be configured and enabled/disabled as needed by software.

With the WDTE bit set in the PCA0MD register, Module 5 operates as a watchdog timer (WDT). The Module 2 high byte is compared to the PCA0 counter high byte; the Module 2 low byte holds the offset to be used when WDT updates are performed. **The Watchdog Timer is enabled on reset. Writes to some PCA0 registers are restricted while the Watchdog Timer is enabled.** The WDT will generate a reset shortly after code begins execution. To avoid this reset, the WDT should be explicitly disabled (and optionally re-configured and re-enabled if it is used in the system).

### 28.4.1. Watchdog Timer Operation

While the WDT is enabled:

- PCA0 counter is forced on.
- Writes to PCA0L and PCA0H are not allowed.
- PCA0 clock source bits (CPS2–CPS0) are frozen.
- PCA0 Idle control bit (CIDL) is frozen.
- PCA0 Module 5 is forced into software timer mode.
- Writes to the Module 5 mode register (PCA0CPM5) are disabled.

While the WDT is enabled, writes to the CR bit will not change the PCA0 counter state; the counter will run until the WDT is disabled. The PCA0 counter run control bit (CR) will read zero if the WDT is enabled but user software has not enabled the PCA0 counter. If a match occurs between PCA0CPH5 and PCA0H while the WDT is enabled, a reset will be generated. To prevent a WDT reset, the WDT may be updated with a write of any value to PCA0CPH5. Upon a PCA0CPH5 write, PCA0H plus the offset held in PCA0CPL5 is loaded into PCA0CPH5 (See Figure 28.11).

## SFR Definition 28.2. PCA0MD: PCA0 Mode

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | CIDL | WDTE | WDLCK | | CPS2 | CPS1 | CPS0 | ECF |
| Type | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W |
| Reset | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Address = 0xD9; SFR Page = 0x00

| Bit | Name | Function |
|---|---|---|
| 7 | CIDL | **PCA0 Counter/Timer Idle Control.**<br>Specifies PCA0 behavior when CPU is in Idle Mode.<br>0: PCA0 continues to function normally while the system controller is in Idle Mode.<br>1: PCA0 operation is suspended while the system controller is in Idle Mode. |
| 6 | WDTE | **Watchdog Timer Enable**<br>If this bit is set, PCA0 Module 5 is used as the watchdog timer.<br>0: Watchdog Timer disabled.<br>1: PCA0 Module 5 enabled as Watchdog Timer. |
| 5 | WDLCK | **Watchdog Timer Lock**<br>This bit locks/unlocks the Watchdog Timer Enable. When WDLCK is set, the Watchdog Timer may not be disabled until the next system reset.<br>0: Watchdog Timer Enable unlocked.<br>1: Watchdog Timer Enable locked. |
| 4 | Unused | Read = 0b, Write = Don't care. |
| 3:1 | CPS[2:0] | **PCA0 Counter/Timer Pulse Select.**<br>These bits select the timebase source for the PCA0 counter<br>000: System clock divided by 12<br>001: System clock divided by 4<br>010: Timer 0 overflow<br>011: High-to-low transitions on ECI (max rate = system clock divided by 4)<br>100: System clock<br>101: External clock divided by 8 (synchronized with the system clock)<br>110: Timer 4 overflow<br>111: Timer 5 overflow |
| 0 | ECF | **PCA0 Counter/Timer Overflow Interrupt Enable.**<br>This bit sets the masking of the PCA0 Counter/Timer Overflow (CF) interrupt.<br>0: Disable the CF interrupt.<br>1: Enable a PCA0 Counter/Timer Overflow interrupt request when CF (PCA0CN.7) is set. |
| **Note:** When the WDTE bit is set to 1, the other bits in the PCA0MD register cannot be modified. To change the contents of the PCA0MD register, the Watchdog Timer must first be disabled. | | |

## SFR Definition 28.3. PCA0PWM: PCA0 PWM Configuration

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | ARSEL | ECOV | COVF | | | | CLSEL[1:0] | |
| Type | R/W | R/W | R/W | R | R | R | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Address = 0xD9; SFR Page = 0x0F

| Bit | Name | Function |
|---|---|---|
| 7 | ARSEL | **Auto-Reload Register Select.**<br>This bit selects whether to read and write the normal PCA0 capture/compare registers (PCA0CPn), or the Auto-Reload registers at the same SFR addresses. This function is used to define the reload value for 9, 10, and 11-bit PWM modes. In all other modes, the Auto-Reload registers have no function.<br>0: Read/Write Capture/Compare Registers at PCA0CPHn and PCA0CPLn.<br>1: Read/Write Auto-Reload Registers at PCA0CPHn and PCA0CPLn. |
| 6 | ECOV | **Cycle Overflow Interrupt Enable.**<br>This bit sets the masking of the Cycle Overflow Flag (COVF) interrupt.<br>0: COVF will not generate PCA0 interrupts.<br>1: A PCA0 interrupt will be generated when COVF is set. |
| 5 | COVF | **Cycle Overflow Flag.**<br>This bit indicates an overflow of the 8th, 9th, 10th, or 11th bit of the main PCA0 counter (PCA0). The specific bit used for this flag depends on the setting of the Cycle Length Select bits. The bit can be set by hardware or software, but must be cleared by software.<br>0: No overflow has occurred since the last time this bit was cleared.<br>1: An overflow has occurred since the last time this bit was cleared. |
| 4:2 | Unused | Read = 000b; Write = Don't care. |
| 1:0 | CLSEL[1:0] | **Cycle Length Select.**<br>When 16-bit PWM mode is not selected, these bits select the length of the PWM cycle, between 8, 9, 10, or 11 bits. This affects all channels configured for PWM which are not using 16-bit PWM mode. These bits are ignored for individual channels configured to16-bit PWM mode.<br>00: 8 bits.<br>01: 9 bits.<br>10: 10 bits.<br>11: 11 bits. |