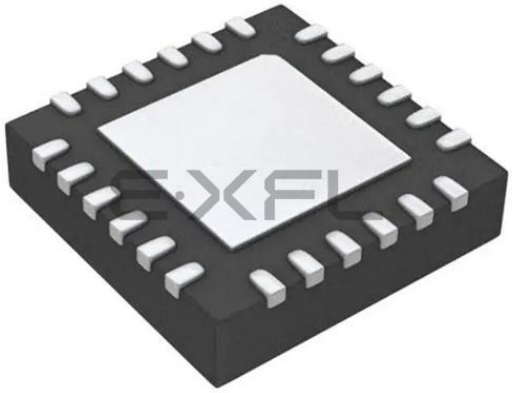


**Welcome to E-XFL.COM**

## 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 | SMBus (2-Wire/I²C), CANbus, LINbus, SPI, UART/USART |
| Peripherals | POR, PWM, Temp Sensor, WDT |
| Number of I/O | 18 |
| Program Memory Size | 32KB (32K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 2.25K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 5.25V |
| Data Converters | A/D 18x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 24-WFQFN Exposed Pad |
| Supplier Device Package | 24-QFN (4x4) |
| Purchase URL | https://www.e-xfl.com/product-detail/silicon-labs/c8051f550-imr |

# 1. System Overview

C8051F55x/56x/57x devices are fully integrated mixed-signal System-on-a-Chip MCUs. Highlighted features are listed below. Refer to Table 2.1 for specific product feature selection and part ordering numbers.

- High-speed pipelined 8051-compatible microcontroller core (up to 50 MIPS)
- In-system, full-speed, non-intrusive debug interface (on-chip)
- Controller Area Network (CAN 2.0B) Controller with 32 message objects, each with its own indentifier mask (C8051F550/1/4/5, ‘F560/1/4/5/8/9, and ‘F572/3)
- LIN 2.1 peripheral (fully backwards compatible, master and slave modes) (C8051F550/2/4/6, ‘F560/2/4/6/8, and ‘F570/2/4)
- True 12-bit 200 ksps 32-channel single-ended ADC with analog multiplexer
- Precision programmable 24 MHz internal oscillator that is within ±0.5% across the temperature range and for VDD voltages greater than or equal to the on-chip voltage regulator minimum output at the low setting. The oscillator is within ±1.0% for VDD voltages below this minimum output setting.
- On-chip Clock Multiplier to reach up to 50 MHz
- 32 kB (C8051F550-3, ‘F560-3, ‘F568-9, and ‘F570-1) or 16 kB (C8051F554-7, ‘F564-7, and ‘F572-5) of on-chip Flash memory
- 2304 bytes of on-chip RAM
- SMBus/I2C, Enhanced UART, and Enhanced SPI serial interfaces implemented in hardware
- Four general-purpose 16-bit timers
- External Data Memory Interface (C8051F568-9 and ‘F570-5) with 64 kB address space
- Programmable Counter/Timer Array (PCA) with six capture/compare modules and Watchdog Timer function
- On-chip Voltage Regulator
- On-chip Power-On Reset, $V_{DD}$ Monitor, and Temperature Sensor
- On-chip Voltage Comparator
- 33, 25, or 18 Port I/O (5 V push-pull)

With on-chip Voltage Regulator, Power-On Reset, $V_{DD}$ monitor, Watchdog Timer, and clock oscillator, the C8051F55x/56x/57x devices are truly stand-alone System-on-a-Chip solutions. The Flash memory can be reprogrammed even in-circuit, providing non-volatile data storage, and also allowing field upgrades of the 8051 firmware. User software has complete control of all peripherals, and may individually shut down any or all peripherals for power savings.

The on-chip Silicon Labs 2-Wire (C2) Development Interface allows non-intrusive (uses no on-chip resources), full speed, in-circuit debugging using the production MCU installed in the final application. This debug logic supports inspection and modification of memory and registers, setting breakpoints, single stepping, run and halt commands. All analog and digital peripherals are fully functional while debugging using C2. The two C2 interface pins can be shared with user functions, allowing in-system debugging without occupying package pins.

The devices are specified for 1.8 V to 5.25 V operation over the automotive temperature range (–40 to +125 °C). The C8051F568-9 and ‘F570-5 are available in 40-pin QFN packages, the C8051F560-7 devices are available in 32-pin QFP and QFN packages, and the C8051F550-7 are available in 24-pin QFN packages. All package options are lead-free and RoHS compliant. See Table 2.1 for ordering information. Block diagrams are included in Figure 1.1, Figure 1.2, and Figure 1.3.

# 6. 12-Bit ADC (ADC0)

The ADC0 on the C8051F55x/56x/57x consists of an analog multiplexer (AMUX0) with 33, 25, or 18 total input selections and a 200 ksps, 12-bit successive-approximation-register (SAR) ADC with integrated track-and-hold, programmable window detector, programmable attenuation (1:2), and hardware accumulator. The ADC0 subsystem has a special Burst Mode which can automatically enable ADC0, capture and accumulate samples, then place ADC0 in a low power shutdown mode without CPU intervention. The AMUX0, data conversion modes, and window detector are all configurable under software control via the Special Function Registers shows in Figure 6.1. ADC0 inputs are single-ended and may be configured to measure P0.0-P3.7, the Temperature Sensor output, $V_{DD}$, or GND with respect to GND. The voltage reference for ADC0 is selected as described in Section "6.6. Temperature Sensor" on page 67. ADC0 is enabled when the AD0EN bit in the ADC0 Control register (ADC0CN) is set to logic 1, or when performing conversions in Burst Mode. ADC0 is in low power shutdown when AD0EN is logic 0 and no Burst Mode conversions are taking place.

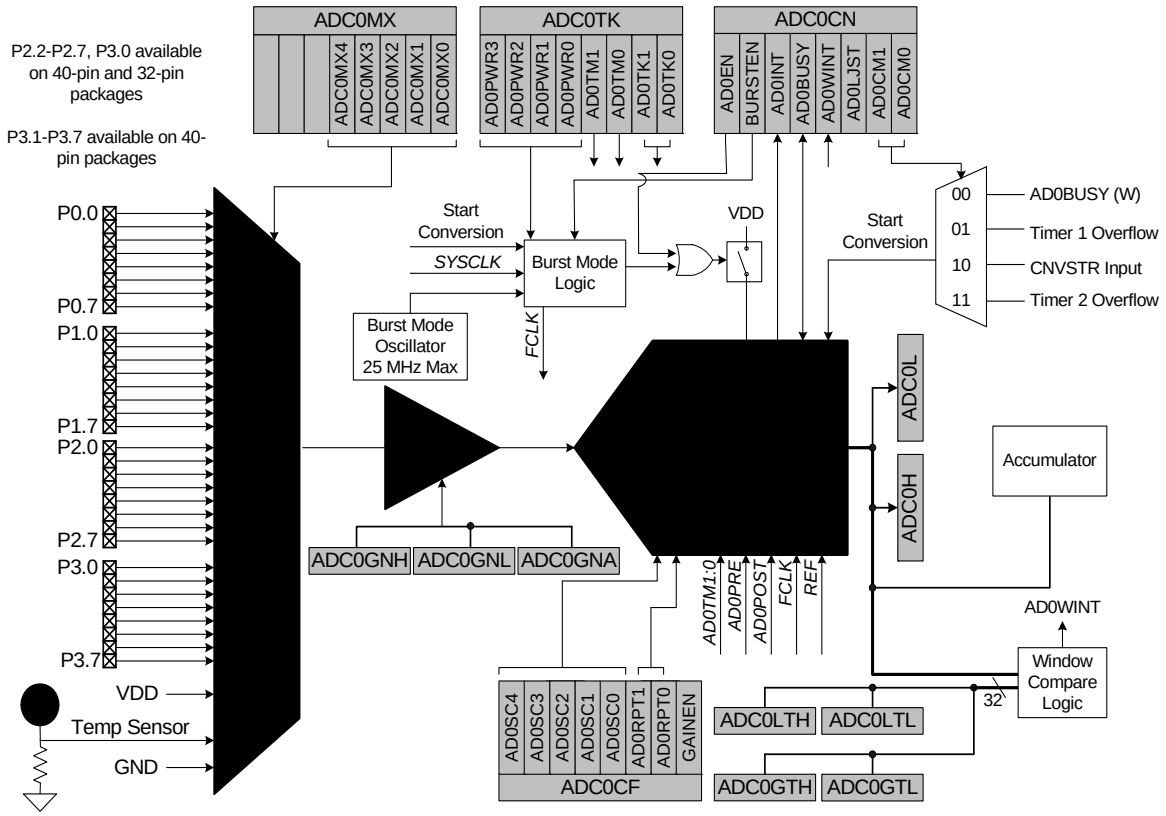



**Figure 6.1. ADC0 Functional Block Diagram**

Post-Tracking Mode is selected when AD0TM is set to 01b. A programmable tracking time based on AD0TK is started immediately following the convert start signal. Conversions are started after the programmed tracking time ends. After a conversion is complete, ADC0 does not track the input. Rather, the sampling capacitor remains disconnected from the input making the input pin high-impedance until the next convert start signal.

Dual-Tracking Mode is selected when AD0TM is set to 11b. A programmable tracking time based on AD0TK is started immediately following the convert start signal. Conversions are started after the programmed tracking time ends. After a conversion is complete, ADC0 tracks continuously until the next conversion is started.

Depending on the output connected to the ADC input, additional tracking time, more than is specified in Table 5.9, may be required after changing MUX settings. See the settling time requirements described in Section “6.2.1. Settling Time Requirements” on page 52.

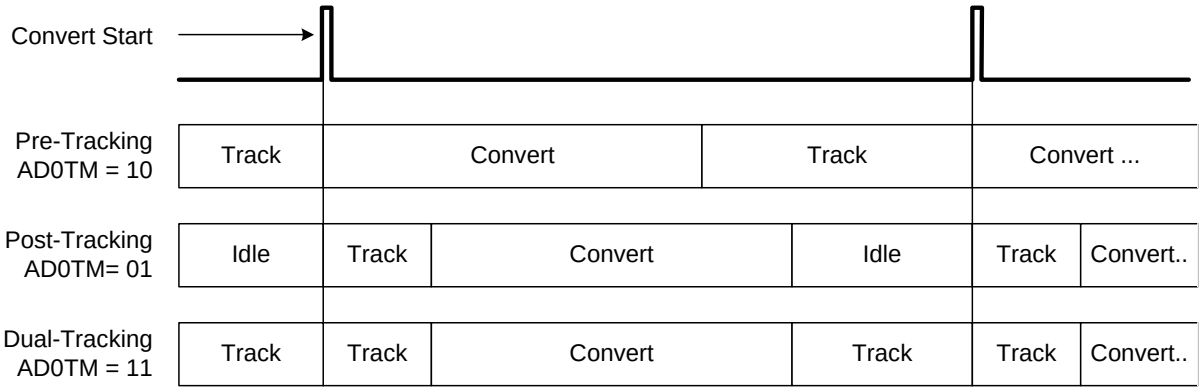



**Figure 6.2. ADC0 Tracking Modes**

### 6.1.3. Timing

ADC0 has a maximum conversion speed specified in Table 5.9. ADC0 is clocked from the ADC0 Subsystem Clock (FCLK). The source of FCLK is selected based on the BURSTEN bit. When BURSTEN is logic 0, FCLK is derived from the current system clock. When BURSTEN is logic 1, FCLK is derived from the Burst Mode Oscillator, an independent clock source with a maximum frequency of 25 MHz.

When ADC0 is performing a conversion, it requires a clock source that is typically slower than FCLK. The ADC0 SAR conversion clock (SAR clock) is a divided version of FCLK. The divide ratio can be configured using the AD0SC bits in the ADC0CF register. The maximum SAR clock frequency is listed in Table 5.9.

ADC0 can be in one of three states at any given time: tracking, converting, or idle. Tracking time depends on the tracking mode selected. For Pre-Tracking Mode, tracking is managed by software and ADC0 starts conversions immediately following the convert start signal. For Post-Tracking and Dual-Tracking Modes, the tracking time after the convert start signal is equal to the value determined by the AD0TK bits plus 2 FCLK cycles. Tracking is immediately followed by a conversion. The ADC0 conversion time is always 13 SAR clock cycles plus an additional 2 FCLK cycles to start and complete a conversion. Figure 6.3 shows timing diagrams for a conversion in Pre-Tracking Mode and tracking plus conversion in Post-Tracking or Dual-Tracking Mode. In this example, repeat count is set to one.

## Gain Register Definition 6.3. ADC0GNA: ADC0 Additional Selectable Gain

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | GAINADD |
| Type | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Indirect Address = 0x08;

| Bit | Name | Function |
|---|---|---|
| 7:1 | Reserved | Must Write 0000000b. |
| 0 | GAINADD | **ADC0 Additional Gain Bit.**<br>Setting this bit add 1/64 (0.016) gain to the gain value in the ADC0GNH and ADC0GNL registers. |
| **Note:** This register is accessed indirectly; See Section 6.3.2 for details for writing this register. | | |

## SFR Definition 6.4. ADC0CF: ADC0 Configuration

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | AD0SC[4:0] | | | | | AD0RPT[1:0] | | GAINEN |
| Type | R/W | | | | | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

SFR Address = 0xBC; SFR Page = 0x00

| Bit | Name | Function |
|---|---|---|
| 7:3 | AD0SC[4:0] | **ADC0 SAR Conversion Clock Period Bits.**<br>SAR Conversion clock is derived from system clock by the following equation, where *AD0SC* refers to the 5-bit value held in bits AD0SC4–0. SAR Conversion clock requirements are given in the ADC specification table<br>BURSTEN = 0: FCLK is the current system clock<br>BURSTEN = 1: FCLK is a maximum of 30 MHz, independent of the current system clock..<br>$AD0SC = \frac{FCLK}{CLK_{SAR}} - 1$<br>**Note:** Round up the result of the calculation for AD0SC |
| 2:1 | A0RPT[1:0] | **ADC0 Repeat Count.**<br>Controls the number of conversions taken and accumulated between ADC0 End of Conversion (ADCINT) and ADC0 Window Comparator (ADCWINT) interrupts. A convert start is required for each conversion unless Burst Mode is enabled. In Burst Mode, a single convert start can initiate multiple self-timed conversions. Results in both modes are accumulated in the ADC0H:ADC0L register. **When AD0RPT1–0 are set to a value other than '00', the AD0LJST bit in the ADC0CN register must be set to '0' (right justified).**<br>00: 1 conversion is performed.<br>01: 4 conversions are performed and accumulated.<br>10: 8 conversions are performed and accumulated.<br>11: 16 conversions are performed and accumulated. |
| 0 | GAINEN | **Gain Enable Bit.**<br>Controls the gain programming. Refer to Section “6.3. Selectable Gain” on page 53 for information about using this bit. |

### SFR Definition 6.8. ADC0TK: ADC0 Tracking Mode Select

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | AD0PWR[3:0] | | | | AD0TM[1:0] | | AD0TK[1:0] | |
| Type | R/W | | | | R/W | | R/W | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

SFR Address = 0xBA; SFR Page = 0x00

| Bit | Name | Function |
|---|---|---|
| 7:4 | AD0PWR[3:0] | **ADC0 Burst Power-up Time.**<br>For BURSTEN = 0: ADC0 Power state controlled by AD0EN<br>For BURSTEN = 1, AD0EN = 1: ADC0 remains enabled and does not enter the very low power state<br>For BURSTEN = 1, AD0EN = 0: ADC0 enters the very low power state and is enabled after each convert start signal. The Power-up time is programmed according the following equation:<br>$AD0PWR = \frac{Tstartup}{200ns} - 1$ or $Tstartup = (AD0PWR + 1)200ns$ |
| 3:2 | AD0TM[1:0] | **ADC0 Tracking Mode Enable Select Bits.**<br>00: Reserved.<br>01: ADC0 is configured to Post-Tracking Mode.<br>10: ADC0 is configured to Pre-Tracking Mode.<br>11: ADC0 is configured to Dual Tracking Mode. |
| 1:0 | AD0TK[1:0] | **ADC0 Post-Track Time.**<br>00: Post-Tracking time is equal to 2 SAR clock cycles + 2 FCLK cycles.<br>01: Post-Tracking time is equal to 4 SAR clock cycles + 2 FCLK cycles.<br>10: Post-Tracking time is equal to 8 SAR clock cycles + 2 FCLK cycles.<br>11: Post-Tracking time is equal to 16 SAR clock cycles + 2 FCLK cycles. |

## 6.4. Programmable Window Detector

The ADC Programmable Window Detector continuously compares the ADC0 output registers to user-programmed limits, and notifies the system when a desired condition is detected. This is especially effective in an interrupt-driven system, saving code space and CPU bandwidth while delivering faster system response times. The window detector interrupt flag (AD0WINT in register ADC0CN) can also be used in polled mode. The ADC0 Greater-Than (ADC0GTH, ADC0GTL) and Less-Than (ADC0LTH, ADC0LTL) registers hold the comparison values. The window detector flag can be programmed to indicate when measured data is inside or outside of the user-programmed limits, depending on the contents of the ADC0 Less-Than and ADC0 Greater-Than registers.

If the internal voltage regulator is not used, the $V_{REGIN}$ input should be tied to $V_{DD}$, as shown in Figure 9.2.

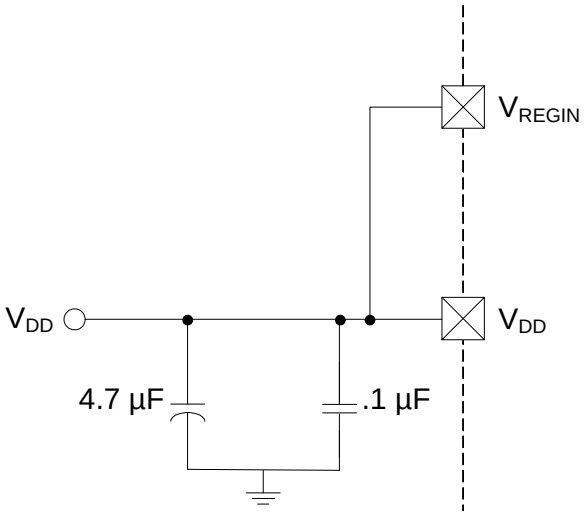



**Figure 9.2. External Capacitors for Voltage Regulator Input/Output—Regulator Disabled**

## SFR Definition 9.1. REG0CN: Regulator Control

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | REGDIS | Reserved | | REG0MD | | | | DROPOUT |
| Type | R/W | R/W | R | R/W | R | R | R | R |
| Reset | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |

SFR Address = 0xC9; SFR Page = 0x00

| Bit | Name | Function |
|---|---|---|
| 7 | REGDIS | **Voltage Regulator Disable Bit.**<br>0: Voltage Regulator Enabled<br>1: Voltage Regulator Disabled |
| 6 | Reserved | Read = 1b; Must Write 1b. |
| 5 | Unused | Read = 0b; Write = Don't Care. |
| 4 | REG0MD | **Voltage Regulator Mode Select Bit.**<br>0: Voltage Regulator Output is 2.1 V.<br>1: Voltage Regulator Output is 2.6 V. |
| 3:1 | Unused | Read = 000b. Write = Don't Care. |
| 0 | DROPOUT | **Voltage Regulator Dropout Indicator.**<br>0: Voltage Regulator is not in dropout.<br>1: Voltage Regulator is in or near dropout. |

## SFR Definition 14.2. FLKEY: Flash Lock and Key

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

SFR Address = 0xB7; SFR Page = All Pages

| Bit | Name | Function |
|---|---|---|
| 7:0 | FLKEY[7:0] | **Flash Lock and Key Register.**<br>Write:<br>This register provides a lock and key function for Flash erasures and writes. Flash writes and erases are enabled by writing 0xA5 followed by 0xF1 to the FLKEY register. Flash writes and erases are automatically disabled after the next write or erase is complete. If any writes to FLKEY are performed incorrectly, or if a Flash write or erase operation is attempted while these operations are disabled, the Flash will be permanently locked from writes or erasures until the next device reset. If an application never writes to Flash, it can intentionally lock the Flash by writing a non-0xA5 value to FLKEY from software.<br>Read:<br>When read, bits 1–0 indicate the current Flash lock state.<br>00: Flash is write/erase locked.<br>01: The first key code has been written (0xA5).<br>10: Flash is unlocked (writes/erases allowed).<br>11: Flash writes/erases disabled until the next reset. |

## SFR Definition 18.5. 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.

The CAN controller clock must be less than or equal to 25 MHz. If the CIP-51 system clock is above 25 MHz, the divider in the CAN0CFG register must be set to divide the CAN controller clock down to an appropriate speed.

### 21.1.2. CAN Register Access

The CAN controller clock divider selected in the CAN0CFG SFR affects how the CAN registers can be accessed. If the divider is set to 1, then a CAN SFR can immediately be read after it is written. If the divider is set to a value other than 1, then a read of a CAN SFR that has just been written must be delayed by a certain number of cycles. This delay can be performed using a NOP or some other instruction that does not attempt to read the register. This access limitation applies to read and read-modify-write instructions that occur immediately after a write. The full list of affected instructions is ANL, ORL, MOV, XCH, and XRL.

For example, with the CAN0CFG divider set to 1, the CAN0CN SFR can be accessed as follows:

```
MOV CAN0CN, #041          ; Enable access to Bit Timing Register
MOV R7, CAN0CN            ; Copy CAN0CN to R7
```

With the CAN0CFG divider set to /2, the same example code requires an additional NOP:

```
MOV CAN0CN, #041          ; Enable access to Bit Timing Register
NOP                       ; Wait for write to complete
MOV R7, CAN0CN            ; Copy CAN0CN to R7
```

The number of delay cycles required is dependent on the divider setting. With a divider of 2, the read must wait for 1 system clock cycle. With a divider of 4, the read must wait 3 system clock cycles, and with the divider set to 8, the read must wait 7 system clock cycles. The delay only needs to be applied when reading the same register that was written. The application can write and read other CAN SFRs without any delay.

### 21.1.3. Example Timing Calculation for 1 Mbit/Sec Communication

This example shows how to configure the CAN controller timing parameters for a 1 Mbit/Sec bit rate. Table 21.1 shows timing-related system parameters needed for the calculation.

**Table 21.1. Background System Information**

| Parameter | Value | Description |
|---|---|---|
| CIP-51 system clock (SYSCLK) | 24 MHz | Internal Oscillator Max |
| CAN controller clock ($f_{sys}$) | 24 MHz | CAN0CFG divider set to 1 |
| CAN clock period ($t_{sys}$) | 41.667 ns | Derived from $1/f_{sys}$ |
| CAN time quantum ($t_q$) | 41.667 ns | Derived from $t_{sys}$ x BRP[1,2] |
| CAN bus length | 10 m | 5 ns/m signal delay between CAN nodes |
| Propogation delay time[3] | 400 ns | 2 x (transceiver loop delay + bus line delay) |

**Notes:**
1. The CAN time quantum is the smallest unit of time recognized by the CAN controller. Bit timing parameters are specified in integer multiples of the time quantum.
2. The Baud Rate Prescaler (BRP) is defined as the value of the BRP Extension Register plus 1. The BRP extension register has a reset value of 0x0000. The BRP has a reset value of 1.
3. Based on an ISO-11898 compliant transceiver. CAN does not specify a physical layer.

Each bit transmitted on a CAN network has 4 segments (Sync_Seg, Prop_Seg, Phase_Seg1, and Phase_Seg2), as shown in Figure 18.3. The sum of these segments determines the CAN bit time (1/bit rate). In this example, the desired bit rate is 1 Mbit/sec; therefore, the desired bit time is 1000 ns.

## 21.2. CAN Registers

CAN registers are classified as follows:

1. **CAN Controller Protocol Registers**: CAN control, interrupt, error control, bus status, test modes.
2. **Message Object Interface Registers**: Used to configure 32 Message Objects, send and receive data to and from Message Objects. The CIP-51 MCU accesses the CAN message RAM via the Message Object Interface Registers. Upon writing a message object number to an IF1 or IF2 Command Request Register, the contents of the associated Interface Registers (IF1 or IF2) will be transferred to or from the message object in CAN RAM.
3. **Message Handler Registers**: These read only registers are used to provide information to the CIP-51 MCU about the message objects (MSGVLD flags, Transmission Request Pending, New Data Flags) and Interrupts Pending (which Message Objects have caused an interrupt or status interrupt condition).

For the registers other than CAN0CFG, refer to the Bosch CAN User's Guide for information on the function and use of the CAN Control Protocol Registers.

### 21.2.1. CAN Controller Protocol Registers

The CAN Control Protocol Registers are used to configure the CAN controller, process interrupts, monitor bus status, and place the controller in test modes.

The registers are: CAN Control Register (CAN0CN), CAN Clock Configuration (CAN0CFG), CAN Status Register (CAN0STA), CAN Test Register (CAN0TST), Error Counter Register, Bit Timing Register, and the Baud Rate Prescaler (BRP) Extension Register.

### 21.2.2. Message Object Interface Registers

There are two sets of Message Object Interface Registers used to configure the 32 Message Objects that transmit and receive data to and from the CAN bus. Message objects can be configured for transmit or receive, and are assigned arbitration message identifiers for acceptance filtering by all CAN nodes.

Message Objects are stored in Message RAM, and are accessed and configured using the Message Object Interface Registers.

### 21.2.3. Message Handler Registers

The Message Handler Registers are read only registers. The message handler registers provide interrupt, error, transmit/receive requests, and new data information.

SFR Address = 0x98; Bit-Addressable; SFR Page = 0x00

| Bit | Name | Function |
|---|---|---|
| 7 | OVR0 | **Receive FIFO Overrun Flag.**<br>0: Receive FIFO Overrun has not occurred<br>1: Receive FIFO Overrun has occurred; A received character has been discarded due to a full FIFO. |
| 6 | PERR0 | **Parity Error Flag.**<br>When parity is enabled, this bit indicates that a parity error has occurred. It is set to 1 when the parity of the oldest byte in the FIFO does not match the selected Parity Type.<br>0: Parity error has not occurred<br>1: Parity error has occurred.<br>This bit must be cleared by software. |
| 5 | THRE0 | **Transmit Holding Register Empty Flag.**<br>THRE0 can have a momentary glitch high when the UART Transmit Holding Register is not empty. The glitch will occur some time after SBUF0 was written with the previous byte and does not occur if THRE0 is checked in the instruction(s) immediately following the write to SBUF0. When firmware writes SBUF0 and SBUF0 is not empty, TX0 will be stuck low until the next device reset. Firmware should use or poll on TI0 rather than THRE0 for asynchronous UART writes that may have a random delay in between transactions.<br>0: Transmit Holding Register not Empty—do not write to SBUF0.<br>1: Transmit Holding Register Empty—it is safe to write to SBUF0. |
| 4 | REN0 | **Receive Enable.**<br>This bit enables/disables the UART receiver. When disabled, bytes can still be read from the receive FIFO.<br>0: UART1 reception disabled.<br>1: UART1 reception enabled. |
| 3 | TBX0 | **Extra Transmission Bit.**<br>The logic level of this bit will be assigned to the extra transmission bit when XBE0 is set to 1. This bit is not used when Parity is enabled. |
| 2 | RBX0 | **Extra Receive Bit.**<br>RBX0 is assigned the value of the extra bit when XBE1 is set to 1. If XBE1 is cleared to 0, RBX1 will be assigned the logic level of the first stop bit. This bit is not valid when Parity is enabled. |
| 1 | TI0 | **Transmit Interrupt Flag.**<br>Set to a 1 by hardware after data has been transmitted, at the beginning of the STOP bit. When the UART0 interrupt is enabled, setting this bit causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software. |
| 0 | RI0 | **Receive Interrupt Flag.**<br>Set to 1 by hardware when a byte of data has been received by UART0 (set at the STOP bit sampling time). When the UART0 interrupt is enabled, setting this bit to 1 causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software. Note that RI0 will remain set to ‘1’ as long as there is data still in the UART FIFO. After the last byte has been shifted from the FIFO to SBUF0, RI0 can be cleared. |

## SFR Definition 23.5. SBRLH0: UART0 Baud Rate Generator Reload High Byte

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | SBRLH0[7:0] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Address = 0xAD; SFR Page = 0x0F

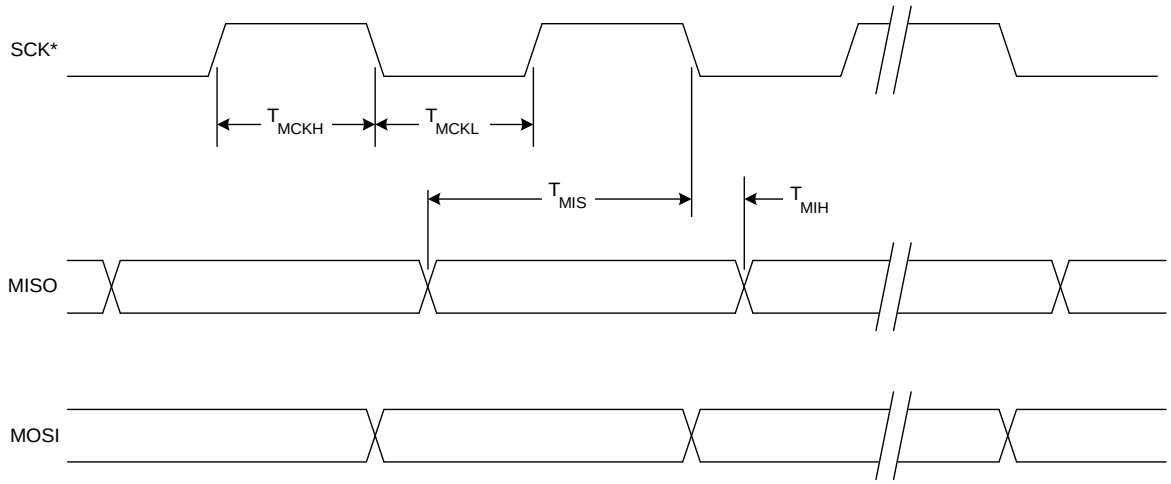
| Bit | Name | Function |
|---|---|---|
| 7:0 | SBRLH0[7:0] | **High Byte of Reload Value for UART0 Baud Rate Generator.**<br>This value is loaded into the high byte of the UART0 baud rate generator when the counter overflows from 0xFFFF to 0x0000. |

## SFR Definition 23.6. SBRLL0: UART0 Baud Rate Generator Reload Low Byte

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | SBRLL0[7:0] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

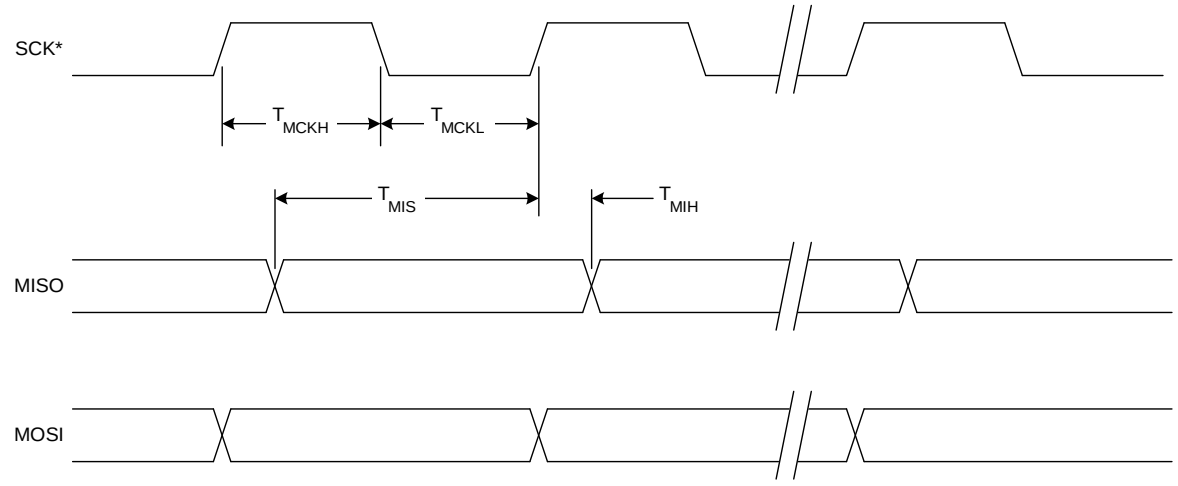
SFR Address = 0xAC; SFR Page = 0x0F

| Bit | Name | Function |
|---|---|---|
| 7:0 | SBRLL0[7:0] | **Low Byte of Reload Value for UART0 Baud Rate Generator.**<br>This value is loaded into the low byte of the UART0 baud rate generator when the counter overflows from 0xFFFF to 0x0000. |




* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

**Figure 24.8. SPI Master Timing (CKPHA = 0)**




* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

**Figure 24.9. SPI Master Timing (CKPHA = 1)**

| T2MH | T2XCLK | TMR2H Clock Source |
|---|---|---|
| 0 | 0 | SYSCLK/12 |
| 0 | 1 | External Clock/8 |
| 1 | X | SYSCLK |

| T2ML | T2XCLK | TMR2L Clock Source |
|---|---|---|
| 0 | 0 | SYSCLK/12 |
| 0 | 1 | External Clock/8 |
| 1 | X | SYSCLK |

The TF2H bit is set when TMR2H overflows from 0xFF to 0x00; the TF2L bit is set when TMR2L overflows from 0xFF to 0x00. When Timer 2 interrupts are enabled (IE.5), an interrupt is generated each time TMR2H overflows. If Timer 2 interrupts are enabled and TF2LEN (TMR2CN.5) is set, an interrupt is generated each time either TMR2L or TMR2H overflows. When TF2LEN is enabled, software must check the TF2H and TF2L flags to determine the source of the Timer 2 interrupt. The TF2H and TF2L interrupt flags are not cleared by hardware and must be manually cleared by software.

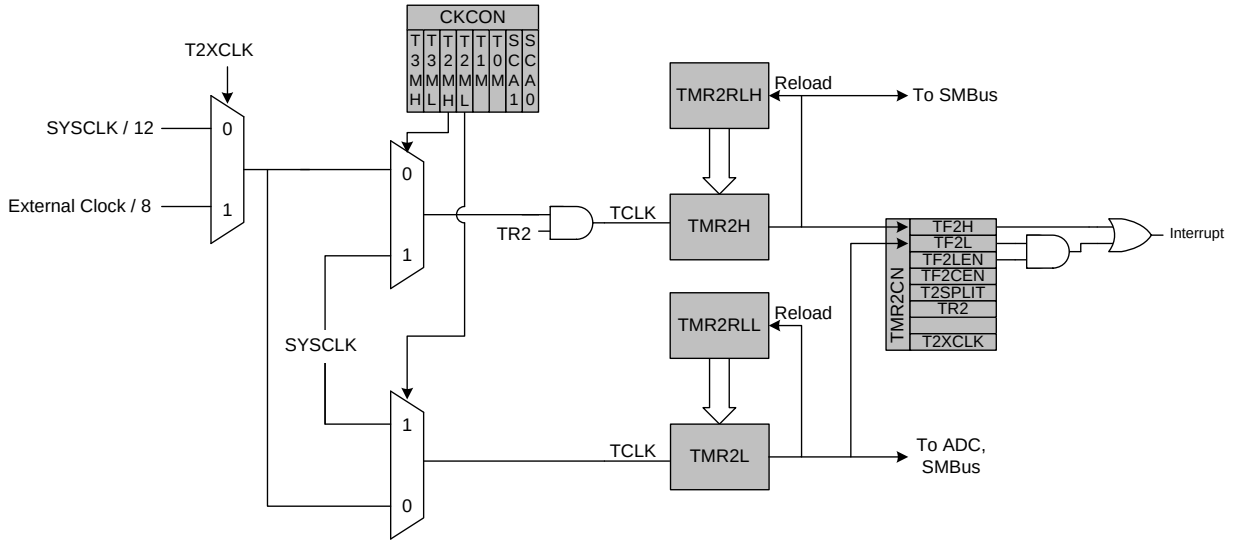



**Figure 25.5. Timer 2 8-Bit Mode Block Diagram**

### 25.2.3. External Oscillator Capture Mode

Capture Mode allows the external oscillator to be measured against the system clock. Timer 2 can be clocked from the system clock, or the system clock divided by 12, depending on the T2ML (CKCON.4), and T2XCLK bits. When a capture event is generated, the contents of Timer 2 (TMR2H:TMR2L) are loaded into the Timer 2 reload registers (TMR2RLH:TMR2RLL) and the TF2H flag is set. A capture event is generated by the falling edge of the clock source being measured, which is the external oscillator / 8. By recording the difference between two successive timer capture values, the external oscillator frequency can be determined with respect to the Timer 2 clock. The Timer 2 clock should be much faster than the capture clock to achieve an accurate reading. Timer 2 should be in 16-bit auto-reload mode when using Capture Mode.

For example, if T2ML = 1b and TF2CEN = 1b, Timer 2 will clock every SYSCLK and capture every external clock divided by 8. If the SYSCLK is 24 MHz and the difference between two successive captures is 5984, then the external clock frequency is as follows:

24 MHz/(5984/8) = 0.032086 MHz or 32.086 kHz

## SFR Definition 25.8. TMR2CN: Timer 2 Control

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | TF2H | TF2L | TF2LEN | TF2CEN | T2SPLIT | TR2 | | T2XCLK |
| Type | R/W | R/W | R/W | R/W | R/W | R/W | R | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

SFR Address = 0xC8; Bit-Addressable; SFR Page = 0x00

| Bit | Name | Function |
|---|---|---|
| 7 | TF2H | **Timer 2 High Byte Overflow Flag.**<br>Set by hardware when the Timer 2 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 2 overflows from 0xFFFF to 0x0000. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 interrupt service routine. This bit is not automatically cleared by hardware. |
| 6 | TF2L | **Timer 2 Low Byte Overflow Flag.**<br>Set by hardware when the Timer 2 low byte overflows from 0xFF to 0x00. TF2L will be set when the low byte overflows regardless of the Timer 2 mode. This bit is not automatically cleared by hardware. |
| 5 | TF2LEN | **Timer 2 Low Byte Interrupt Enable.**<br>When set to 1, this bit enables Timer 2 Low Byte interrupts. If Timer 2 interrupts are also enabled, an interrupt will be generated when the low byte of Timer 2 overflows. |
| 4 | TF2CEN | **Timer 2 Capture Mode Enable.**<br>0: Timer 2 Capture Mode is disabled.<br>1: Timer 2 Capture Mode is enabled. |
| 3 | T2SPLIT | **Timer 2 Split Mode Enable.**<br>When this bit is set, Timer 2 operates as two 8-bit timers with auto-reload.<br>0: Timer 2 operates in 16-bit auto-reload mode.<br>1: Timer 2 operates as two 8-bit auto-reload timers. |
| 2 | TR2 | **Timer 2 Run Control.**<br>Timer 2 is enabled by setting this bit to 1. In 8-bit mode, this bit enables/disables TMR2H only; TMR2L is always enabled in split mode. |
| 1 | Unused | Read = 0b; Write = Don’t Care |
| 0 | T2XCLK | **Timer 2 External Clock Select.**<br>This bit selects the external clock source for Timer 2. If Timer 2 is in 8-bit mode, this bit selects the external oscillator clock source for both timer bytes. However, the Timer 2 Clock Select bits (T2MH and T2ML in register CKCON) may still be used to select between the external clock and the system clock for either timer.<br>0: Timer 2 clock is the system clock divided by 12.<br>1: Timer 2 clock is the external clock divided by 8 (synchronized with SYSCLK). |

**Table 26.2. PCA0CPM and PCA0PWM Bit Settings for PCA Capture/Compare Modules**

| Operational Mode | PCA0CPMn | | | | | | | | PCA0PWM | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit Number | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4–2 | 1–0 |
| Capture triggered by positive edge on CEXn | X | X | 1 | 0 | 0 | 0 | 0 | A | 0 | X | B | XXX | XX |
| Capture triggered by negative edge on CEXn | X | X | 0 | 1 | 0 | 0 | 0 | A | 0 | X | B | XXX | XX |
| Capture triggered by any transition on CEXn | X | X | 1 | 1 | 0 | 0 | 0 | A | 0 | X | B | XXX | XX |
| Software Timer | X | C | 0 | 0 | 1 | 0 | 0 | A | 0 | X | B | XXX | XX |
| High Speed Output | X | C | 0 | 0 | 1 | 1 | 0 | A | 0 | X | B | XXX | XX |
| Frequency Output | X | C | 0 | 0 | 0 | 1 | 1 | A | 0 | X | B | XXX | XX |
| 8-Bit Pulse Width Modulator (7) | 0 | C | 0 | 0 | E | 0 | 1 | A | 0 | X | B | XXX | 00 |
| 9-Bit Pulse Width Modulator (7) | 0 | C | 0 | 0 | E | 0 | 1 | A | D | X | B | XXX | 01 |
| 10-Bit Pulse Width Modulator (7) | 0 | C | 0 | 0 | E | 0 | 1 | A | D | X | B | XXX | 10 |
| 11-Bit Pulse Width Modulator (7) | 0 | C | 0 | 0 | E | 0 | 1 | A | D | X | B | XXX | 11 |
| 16-Bit Pulse Width Modulator | 1 | C | 0 | 0 | E | 0 | 1 | A | 0 | X | B | XXX | XX |

**Notes:**
1. X = Don't Care (no functional difference for individual module if 1 or 0).
2. A = Enable interrupts for this module (PCA interrupt triggered on CCFn set to 1).
3. B = Enable 8th, 9th, 10th or 11th bit overflow interrupt (Depends on setting of CLSEL[1:0]).
4. C = When set to 0, the digital comparator is off. For high speed and frequency output modes, the associated pin will not toggle. In any of the PWM modes, this generates a 0% duty cycle (output = 0).
5. D = Selects whether the Capture/Compare register (0) or the Auto-Reload register (1) for the associated channel is accessed via addresses PCA0CPHn and PCA0CPLn.
6. E = When set, a match event will cause the CCFn flag for the associated channel to be set.
7. All modules set to 8, 9, 10 or 11-bit PWM mode use the same cycle length setting.

### 26.3.1. Edge-triggered Capture Mode

In this mode, a valid transition on the CEXn pin causes the PCA to capture the value of the PCA counter/timer and load it into the corresponding module's 16-bit capture/compare register (PCA0CPLn and PCA0CPHn). The CAPPn and CAPNn bits in the PCA0CPMn register are used to select the type of transition that triggers the capture: low-to-high transition (positive edge), high-to-low transition (negative edge), or either transition (positive or negative edge). When a capture occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1. An interrupt request is generated if the CCFn interrupt for that module is enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. If both CAPPn and CAPNn bits are set to logic 1, then the state of the Port pin associated with CEXn can be read directly to determine whether a rising-edge or falling-edge caused the capture.

# 27. C2 Interface

C8051F55x/56x/57x devices include an on-chip Silicon Labs 2-Wire (C2) debug interface to allow Flash programming and in-system debugging with the production part installed in the end application. The C2 interface uses a clock signal (C2CK) and a bi-directional C2 data signal (C2D) to transfer information between the device and a host system. See the C2 Interface Specification for details on the C2 protocol.

## 27.1. C2 Interface Registers

The following describes the C2 registers necessary to perform Flash programming through the C2 interface. All C2 registers are accessed through the C2 interface as described in the C2 Interface Specification.

### C2 Register Definition 27.1. C2ADD: C2 Address

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | C2ADD[7:0] | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Name | Function | |
|---|---|---|---|
| 7:0 | C2ADD[7:0] | **C2 Address.**<br>The C2ADD register is accessed via the C2 interface to select the target Data register for C2 Data Read and Data Write commands. | |
| | | **Address** | **Description** |
| | | 0x00 | Selects the Device ID register for Data Read instructions |
| | | 0x01 | Selects the Revision ID register for Data Read instructions |
| | | 0x02 | Selects the C2 Flash Programming Control register for Data Read/Write instructions |
| | | 0xB4 | Selects the C2 Flash Programming Data register for Data Read/Write instructions |

## Revision 1.1 to Revision 1.2

- Updated the note in “Power-Fail Reset/VDD Monitor” on page 140 to use a larger font.
- Added the note regarding the voltage regulator and VDD monitor in the high setting from “Power-Fail Reset/VDD Monitor” on page 140 to “Voltage Regulator (REG0)” on page 79 and “$V_{DD}$ Maintenance and the $V_{DD}$ monitor” on page 129.
- Updated the steps in “$V_{DD}$ Maintenance and the $V_{DD}$ monitor” on page 129 to mention using the VDD monitor in the high setting during flash write/erase operations.
- Updated the SUSPEND bit description in OSCICN (SFR Definition 18.2) to mention that firmware must set the ZTCEN bit in REF0CN (SFR Definition 7.1) before entering suspend.
- Added a note to the IFRDY flag in the OSCICN register (SFR Definition 18.2) that the flag may not accurately reflect the state of the oscillator.
- Added VREGIN Ramp Time for Power On spec to Table 5.4, “Reset Electrical Characteristics,” on page 41.
- Updated “$V_{DD}$ Maintenance and the $V_{DD}$ monitor” on page 129 to refer to $V_{REGIN}$ ramp time instead of $V_{DD}$ ramp time.
- Added a note regarding programming at cold temperatures on –I devices to “Programming The Flash Memory” on page 124 and added Temperature during Programming Operations specification to Table 5.5, “Flash Electrical Characteristics,” on page 41.
- Added a note regarding P0.0/VREF when VDD is used as the reference to Table 19.1, “Port I/O Assignment for Analog Functions,” on page 171 and to the description of the REFSL bit in REF0CN (SFR Definition 7.1).
- Added a note regarding a potential unknown state on GPIO during power up if VIO ramps significantly before VDD to “Port Input/Output” on page 169 and “Reset Sources” on page 138.
- Added steps to set the FLEWT bit in the FLSCL register (SFR Definition 14.3) in the flash write/erase procedures in “Flash Erase Procedure” on page 125, “Flash Write Procedure” on page 125, and “Flash Write Optimization” on page 126.
- Added a note regarding fast changes on VDD causing the $V_{DD}$ Monitor to trigger to “Power-Fail Reset/VDD Monitor” on page 140.
- Added notes regarding UART TX and RX behavior in “Data Transmission” on page 238, “Data Reception” on page 238, and the THRE0 description in the SCON0 register (SFR Definition 23.1).
- Added a note regarding an issue with /RST low time on some older devices to “16.1. Power-On Reset” .