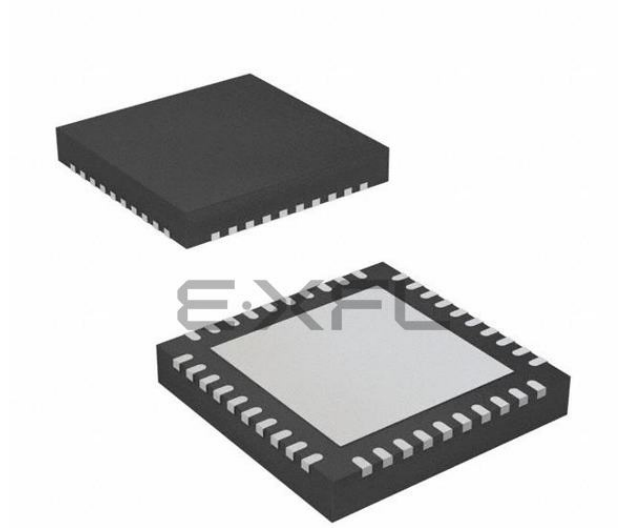


**Welcome to E-XFL.COM**

## 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), CANbus, LINbus, SPI, UART/USART |
| Peripherals | POR, PWM, Temp Sensor, WDT |
| Number of I/O | 33 |
| Program Memory Size | 128KB (128K 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 | 40-VFQFN Exposed Pad |
| Supplier Device Package | 40-QFN (6x6) |
| Purchase URL | https://www.e-xfl.com/product-detail/silicon-labs/c8051f588-imr |

# 1. System Overview

C8051F58x/F59x 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 (C8051F580/2/4/6/8-F590)
- LIN 2.1 peripheral (fully backwards compatible, master and slave modes) (C8051F580/2/4/6/8-F590)
- 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
- 128 kB (C8051F580/1/2/3/8/9) or 96 kB (C8051F584/5/6/7-F590/1) of on-chip Flash memory
- 8448 bytes of on-chip RAM
- SMBus/I2C, Two Enhanced UARTs, and Enhanced SPI serial interfaces implemented in hardware
- Six general-purpose 16-bit timers
- External Data Memory Interface (C8051F580/1/4/5) with 64 kB address space
- Two Programmable Counter/Timer Array (PCA) modules with six capture/compare modules each and one with a Watchdog Timer function
- Three Voltage Comparators
- On-chip Voltage Regulator
- On-chip Power-On Reset, $V_{DD}$ Monitor, and Temperature Sensor
- 40, 33 or 25 Port I/O (5 V push-pull)

With an on-chip Voltage Regulator, Power-On Reset and $V_{DD}$ monitors, Watchdog Timer, and clock oscillator, the C8051F58x/F59x 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 Port I/O and $\overline{RST}$ pins can interface to 5 V logic by setting the VIO pin to 5 V. The C8051F580/1/4/5 devices are available in 48-pin QFP and QFN packages, and the C8051F588/9-F590/1 devices are available in a 40-pin QFN package, and the C8051F582/3/6/7 devices are available in 32-pin QFP and 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 and Figure 1.3.

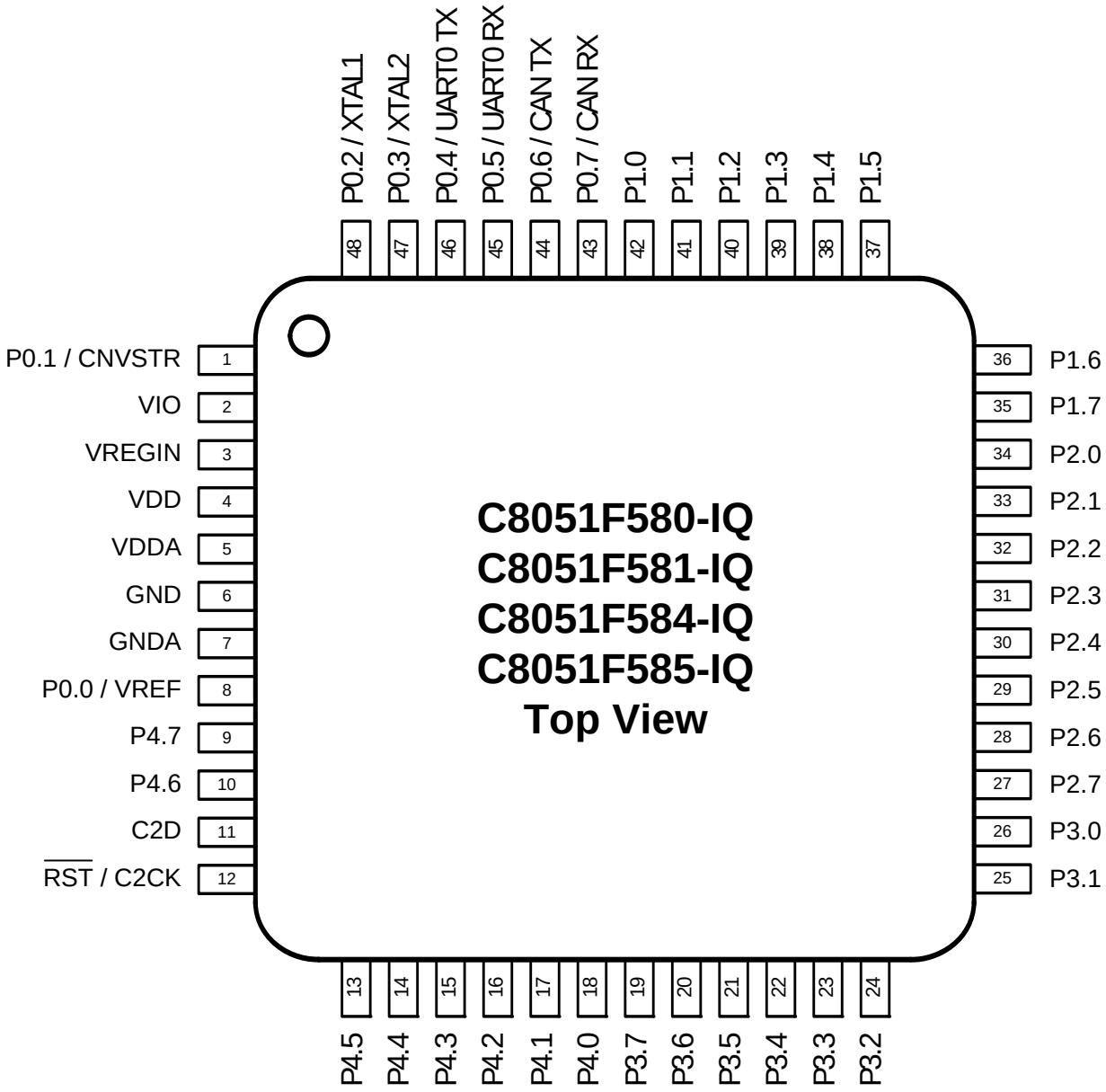



**Figure 3.1. QFP-48 Pinout Diagram (Top View)**

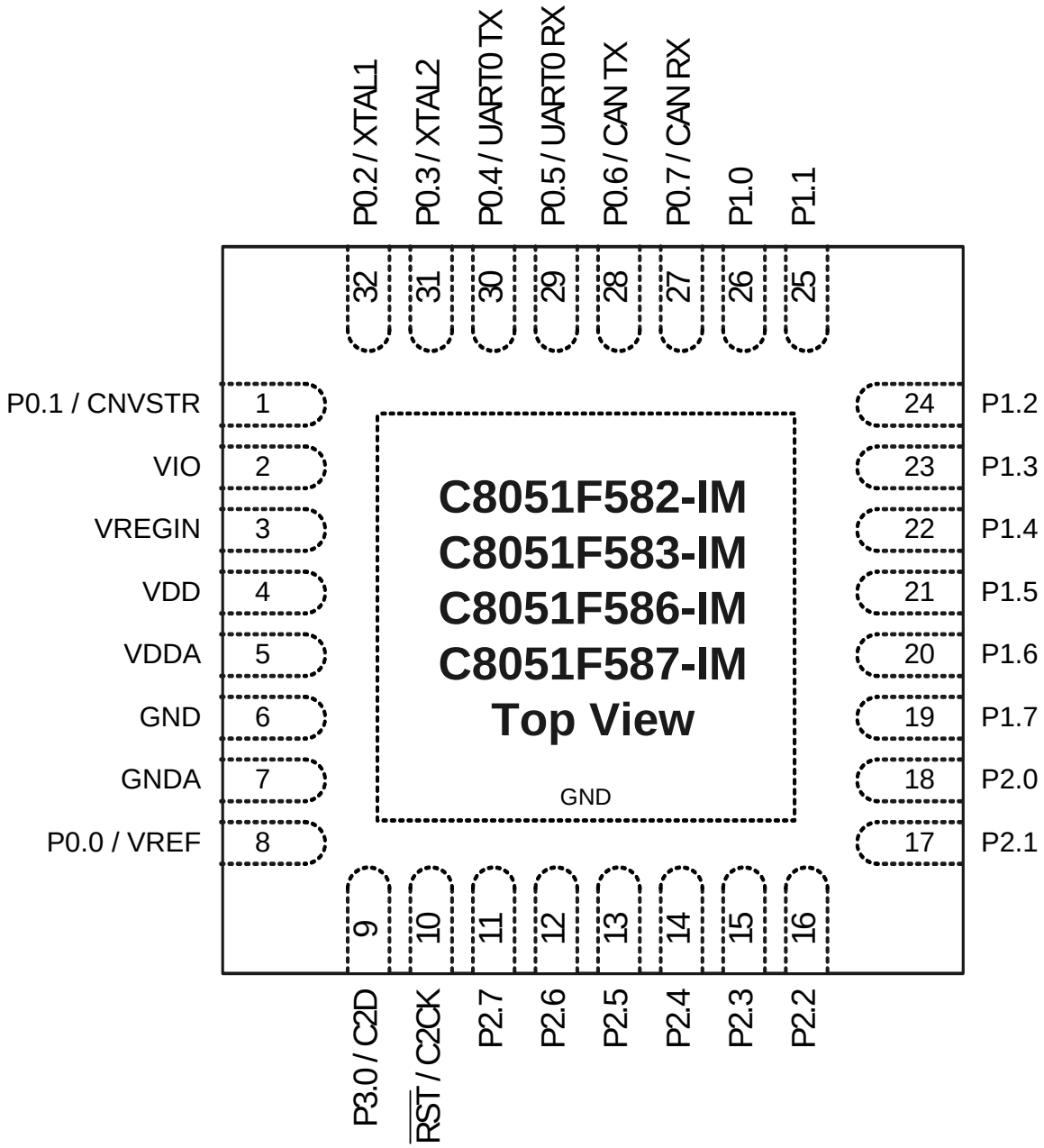



**Figure 3.5. QFN-32 Pinout Diagram (Top View)**

**Table 5.2. Global Electrical Characteristics (Continued)**
–40 to +125 °C, 24 MHz system clock unless otherwise specified.

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| **Digital Supply Current—CPU Inactive (Idle Mode, not fetching instructions from Flash)** | | | | | |
| $I_{DD}$[4] | $V_{DD}$ = 2.1 V, F = 200 kHz | — | 130 | — | µA |
| | $V_{DD}$ = 2.1 V, F = 1.5 MHz | — | 440 | — | µA |
| | $V_{DD}$ = 2.1 V, F = 25 MHz | — | 5.8 | 8.0 | mA |
| | $V_{DD}$ = 2.1 V, F = 50 MHz | — | 11 | 16 | mA |
| $I_{DD}$[4] | $V_{DD}$ = 2.6 V, F = 200 kHz | — | 170 | — | µA |
| | $V_{DD}$ = 2.6 V, F = 1.5 MHz | — | 570 | — | µA |
| | $V_{DD}$ = 2.6 V, F = 25 MHz | — | 7.3 | 15 | mA |
| | $V_{DD}$ = 2.6 V, F = 50 MHz | — | 15 | 25 | mA |
| $I_{DD}$ Supply Sensitivity[4] | F = 25 MHz | — | 53 | — | %/V |
| | F = 1 MHz | — | 60 | — | |
| $I_{DD}$ Frequency Sensitivity [4.6] | $V_{DD}$ = 2.1V, F ≤ 12.5 MHz, T = 25 °C | — | 0.28 | — | mA/MHz |
| | $V_{DD}$ = 2.1V, F > 12.5 MHz, T = 25 °C | — | 0.28 | — | |
| | $V_{DD}$ = 2.6V, F ≤ 12.5 MHz, T = 25 °C | — | 0.35 | — | |
| | $V_{DD}$ = 2.6V, F > 12.5 MHz, T = 25 °C | — | 0.35 | — | |
| Digital Supply Current[4] (Stop or Suspend Mode) | Oscillator not running, $V_{DD}$ Monitor Disabled | | | | µA |
| | Temp = 25 °C | — | 230 | — | |
| | Temp = 60 °C | — | 230 | — | |
| | Temp= 125 °C | — | 330 | — | |

**Notes:**
1. Given in Table 5.4 on page 48.
2. $V_{IO}$ should not be lower than the $V_{DD}$ voltage.
3. SYSCLK must be at least 32 kHz to enable debugging.
4. Based on device characterization data; Not production tested. Does not include oscillator supply current.
5. IDD can be estimated for frequencies ≤ 15 MHz by simply multiplying the frequency of interest by the frequency sensitivity number for that range. When using these numbers to estimate $I_{DD}$ for >15 MHz, the estimate should be the current at 50 MHz minus the difference in current indicated by the frequency sensitivity number. For example: $V_{DD}$ = 2.6 V; F = 20 MHz, $I_{DD}$ = 21 mA - (50 MHz - 20 MHz) * 0.46 mA/MHz = 7.2 mA.
6. Idle IDD can be estimated for frequencies ≤ 1 MHz by simply multiplying the frequency of interest by the frequency sensitivity number for that range. When using these numbers to estimate Idle $I_{DD}$ for >1 MHz, the estimate should be the current at 50 MHz minus the difference in current indicated by the frequency sensitivity number.
   For example: $V_{DD}$ = 2.6 V; F = 5 MHz, Idle $I_{DD}$ = 19 mA – (50 MHz – 5 MHz) x 0.38 mA/MHz = 1.9 mA.

## 6.2. Output Code Formatting

The registers ADC0H and ADC0L contain the high and low bytes of the output conversion code. When the repeat count is set to 1, conversion codes are represented in 12-bit unsigned integer format and the output conversion code is updated after each conversion. Inputs are measured from 0 to $V_{REF}$ x 4095/4096. Data can be right-justified or left-justified, depending on the setting of the AD0LJST bit (ADC0CN.2). Unused bits in the ADC0H and ADC0L registers are set to 0. Example codes are shown below for both right-justified and left-justified data.

| Input Voltage | Right-Justified ADC0H:ADC0L (AD0LJST = 0) | Left-Justified ADC0H:ADC0L (AD0LJST = 1) |
|---|---|---|
| VREF x 4095/4096 | 0x0FFF | 0xFFF0 |
| VREF x 2048/4096 | 0x0800 | 0x8000 |
| VREF x 2047/4096 | 0x07FF | 0x7FF0 |
| 0 | 0x0000 | 0x0000 |

When the ADC0 Repeat Count is greater than 1, the output conversion code represents the accumulated result of the conversions performed and is updated after the last conversion in the series is finished. Sets of 4, 8, or 16 consecutive samples can be accumulated and represented in unsigned integer format. The repeat count can be selected using the AD0RPT bits in the ADC0CF register. The value must be right-justified (AD0LJST = 0), and unused bits in the ADC0H and ADC0L registers are set to 0. The following example shows right-justified codes for repeat counts greater than 1. Notice that accumulating $2^n$ samples is equivalent to left-shifting by *n* bit positions when all samples returned from the ADC have the same value.

| Input Voltage | Repeat Count = 4 | Repeat Count = 8 | Repeat Count = 16 |
|---|---|---|---|
| $V_{REF}$ x 4095/4096 | 0x3FFC | 0x7FF8 | 0xFFF0 |
| $V_{REF}$ x 2048/4096 | 0x2000 | 0x4000 | 0x8000 |
| $V_{REF}$ x 2047/4096 | 0x1FFC | 0x3FF8 | 0x7FF0 |
| 0 | 0x0000 | 0x0000 | 0x0000 |

### 6.2.1. Settling Time Requirements

A minimum tracking time is required before an accurate conversion is performed. This tracking time is determined by any series impedance, including the AMUX0 resistance, the ADC0 sampling capacitance, and the accuracy required for the conversion.

Figure 6.5 shows the equivalent ADC0 input circuit. The required ADC0 settling time for a given settling accuracy (SA) may be approximated by Equation 6.1. When measuring the Temperature Sensor output, use the tracking time specified in Table 5.11 on page 52. When measuring $V_{DD}$ with respect to GND, $R_{TOTAL}$ reduces to $R_{MUX}$. See Table 5.10 for ADC0 minimum settling time requirements as well as the mux impedance and sampling capacitor values.

$$t = \ln\left(\frac{2^n}{SA}\right) \times R_{TOTAL} C_{SAMPLE}$$

**Equation 6.1. ADC0 Settling Time Requirements**

Where:
*SA* is the settling accuracy, given as a fraction of an LSB (for example, 0.25 to settle within 1/4 LSB)
*t* is the required settling time in seconds
$R_{TOTAL}$ is the sum of the AMUX0 resistance and any external source resistance.
*n* is the ADC resolution in bits (10).

## 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.

# 11. CIP-51 Microcontroller

The MCU system controller core is the CIP-51 microcontroller. The CIP-51 is fully compatible with the MCS-51™ instruction set; standard 803x/805x assemblers and compilers can be used to develop software. The MCU family has a superset of all the peripherals included with a standard 8051. The CIP-51 also includes on-chip debug hardware (see description in “C2 Interface” on page 351), and interfaces directly with the analog and digital subsystems providing a complete data acquisition or control-system solution in a single integrated circuit.

The CIP-51 Microcontroller core implements the standard 8051 organization and peripherals as well as additional custom peripherals and functions to extend its capability (see Figure 11.1 for a block diagram). The CIP-51 includes the following features:

- Fully Compatible with MCS-51 Instruction Set
- 50 MIPS Peak Throughput with 50 MHz Clock
- 0 to 50 MHz Clock Frequency
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic
- Program and Data Memory Security

## 11.1. Performance

The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute, and usually have a maximum system clock of 12 MHz. By contrast, the CIP-51 core executes 70% of its instructions in one or two system clock cycles, with no instructions taking more than eight system clock cycles.

## SFR Definition 11.3. SP: Stack Pointer

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

SFR Address = 0x81; SFR Page = All Pages

| Bit | Name | Function |
|---|---|---|
| 7:0 | SP[7:0] | **Stack Pointer.**<br>The Stack Pointer holds the location of the top of the stack. The stack pointer is incremented before every PUSH operation. The SP register defaults to 0x07 after reset. |

## SFR Definition 11.4. ACC: Accumulator

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

SFR Address = 0xE0; SFR Page = All Pages; Bit-Addressable

| Bit | Name | Function |
|---|---|---|
| 7:0 | ACC[7:0] | **Accumulator.**<br>This register is the accumulator for arithmetic operations. |

## SFR Definition 11.5. B: B Register

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

SFR Address = 0xF0; SFR Page = All Pages; Bit-Addressable

| Bit | Name | Function |
|---|---|---|
| 7:0 | B[7:0] | **B Register.**<br>This register serves as a second accumulator for certain arithmetic operations. |

## SFR Definition 12.1. PSBANK: Program Space Bank Select

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

SFR Address = 0xF5; SFR Page = All Pages

| Bit | Name | Function |
|---|---|---|
| 7:6 | Reserved | Read = 00b, Must Write = 00b. |
| 5:4 | COBANK[1:0] | **Constant Operations Bank Select.**<br>These bits select which Flash bank is targeted during constant operations (MOVC and Flash MOVX) involving address 0x8000 to 0xFFFF.<br>00: Constant Operations Target Bank 0 (note that Bank 0 is also mapped between 0x0000 to 0x7FFF).<br>01: Constant operations target Bank 1.<br>10: Constant operations target Bank 2.<br>11: Constant operations target Bank 3. |
| 3:2 | Reserved | Read = 00b, Must Write = 00b. |
| 1:0 | IFBANK[1:0] | **Instruction Fetch Operations Bank Select.**<br>These bits select which Flash bank is used for instruction fetches involving address 0x8000 to 0xFFFF. These bits can only be changed from code in Bank 0.<br>00: Instructions fetch from Bank 0 (note that Bank 0 is also mapped between 0x0000 to 0x7FFF).<br>01: Instructions fetch from Bank 1.<br>10: Instructions fetch from Bank 2.<br>11: Instructions fetch from Bank 3. |
| **Note:** COBANK[1:0] and IFBANK[1:0] should not be set to select Bank 3 (11b) on the C8051F584/5/6/7-F590/1 devices. | | |

### 12.1.1. MOVX Instruction and Program Memory

The MOVX instruction in an 8051 device is typically used to access external data memory. On the C8051F58x/F59x devices, the MOVX instruction is normally used to read and write on-chip XRAM, but can be re-configured to write and erase on-chip Flash memory space. MOVC instructions are always used to read Flash memory, while MOVX write instructions are used to erase and write Flash. This Flash access feature provides a mechanism for the C8051F58x/F59x to update program code and use the program memory space for non-volatile data storage. Refer to Section “15. Flash Memory” on page 138 for further details.

## 12.2. Data Memory

The C8051F58x/F59x devices include 8448 bytes of RAM data memory. 256 bytes of this memory is mapped into the internal RAM space of the 8051. The other 8192 bytes of this memory is on-chip “external” memory. The data memory map is shown in Figure 12.1 for reference.

## SFR Definition 14.5. EIE2: Extended Interrupt Enable 2

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | ET5 | ET4 | ECP2 | EPCA1 | ES1 | EMAT | ECAN0 | EREG0 |
| 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 = 0xE7; SFR Page = All Pages

| Bit | Name | Function |
|---|---|---|
| 7 | ET5 | **Enable Timer 5 Interrupt.**<br>This bit sets the masking of the Timer 5 interrupt.<br>0: Disable Timer 5 interrupts.<br>1: Enable interrupt requests generated by the TF5L or TF5H flags. |
| 6 | ET4 | **Enable Timer 4 Interrupt.**<br>This bit sets the masking of the Timer 4 interrupt.<br>0: Disable Timer 4 interrupts.<br>1: Enable interrupt requests generated by the TF4L or TF4H flags. |
| 5 | ECP2 | **Enable Comparator2 (CP2) Interrupt.**<br>This bit sets the masking of the CP2 interrupt.<br>0: Disable CP2 interrupts.<br>1: Enable interrupt requests generated by the CP2RIF or CP2FIF flags. |
| 4 | EPCA1 | **Enable Programmable Counter Array (PCA1) Interrupt.**<br>This bit sets the masking of the PCA1 interrupts.<br>0: Disable all PCA1 interrupts.<br>1: Enable interrupt requests generated by PCA1 |
| 3 | ES1 | **Enable UART1 Interrupt.**<br>This bit sets the masking of the UART1 interrupt.<br>0: Disable UART1 interrupt.<br>1: Enable UART1 interrupt |
| 2 | EMAT | **Enable Port Match Interrupt.**<br>This bit sets the masking of the Port Match interrupt.<br>0: Disable all Port Match interrupts.<br>1: Enable interrupt requests generated by a Port Match |
| 1 | ECAN0 | **Enable CAN0 Interrupts.**<br>This bit sets the masking of the CAN0 interrupt.<br>0: Disable all CAN0 interrupts.<br>1: Enable interrupt requests generated by CAN0. |
| 0 | EREG0 | **Enable Voltage Regulator Dropout Interrupt.**<br>This bit sets the masking of the Voltage Regulator Dropout interrupt.<br>0: Disable the Voltage Regulator Dropout interrupt.<br>1: Enable the Voltage Regulator Dropout interrupt. |

## 16.2. Stop Mode

Setting the Stop Mode Select bit (PCON.1) causes the controller core to enter Stop mode as soon as the instruction that sets the bit completes execution. In Stop mode the internal oscillator, CPU, and all digital peripherals are stopped; the state of the external oscillator circuit is not affected. Each analog peripheral (including the external oscillator circuit) may be shut down individually prior to entering Stop Mode. Stop mode can only be terminated by an internal or external reset. On reset, the device performs the normal reset sequence and begins program execution at address 0x0000.

If enabled, the Missing Clock Detector will cause an internal reset and thereby terminate the Stop mode. The Missing Clock Detector should be disabled if the CPU is to be put to in STOP mode for longer than the MCD timeout of 100 µs.

## 16.3. Suspend Mode

Setting the SUSPEND bit (OSCICN.5) causes the hardware to halt the CPU and the high-frequency internal oscillator, and go into Suspend mode as soon as the instruction that sets the bit completes execution. All internal registers and memory maintain their original data. Most digital peripherals are not active in Suspend mode. The exception to this is the Port Match feature.

Suspend mode can be terminated by three types of events, a port match (described in Section “20.5. Port Match” on page 200), a Comparator low output (if enabled), or a device reset event. When Suspend mode is terminated, the device will continue execution on the instruction following the one that set the SUSPEND bit. If the wake event was configured to generate an interrupt, the interrupt will be serviced upon waking the device. If Suspend mode is terminated by an internal or external reset, the CIP-51 performs a normal reset sequence and begins program execution at address 0x0000.

Note: When entering Suspend mode, firmware must set the ZTCEN bit in REF0CN (SFR Definition 8.1).

input voltage (on CP0+) is less than the inverting input voltage (on CP0–), the device is put into the reset state. After a Comparator0 reset, the C0RSEF flag (RSTSRC.5) will read 1 signifying Comparator0 as the reset source; otherwise, this bit reads 0. The state of the RST pin is unaffected by this reset.

## 17.6. PCA Watchdog Timer Reset

The programmable Watchdog Timer (WDT) function of the Programmable Counter Array (PCA0) can be used to prevent software from running out of control during a system malfunction. The PCA WDT function can be enabled or disabled by software as described in Section “28.4. Watchdog Timer Mode” on page 324; the WDT is enabled and clocked by SYSCLK/12 following any reset. If a system malfunction prevents user software from updating the WDT, a reset is generated and the WDTRSF bit (RSTSRC.5) is set to 1. The state of the RST pin is unaffected by this reset.

## 17.7. Flash Error Reset

If a Flash read/write/erase or program read targets an illegal address, a system reset is generated. This may occur due to any of the following:

- A Flash write or erase is attempted above user code space. This occurs when PSWE is set to 1 and a MOVX write operation targets an address above address 0xFBFF in Bank 3 on C8051F580/1/2/3/8/9 or any address in Bank 3 on C8051F584/5/6/7-F590/1 devices.
- A Flash read is attempted above user code space. This occurs when a MOVC operation targets an address above address 0xFBFF in Bank 3 on C8051F580/1/2/3/8/9 or any address in Bank 3 on C8051F584/5/6/7-F590/1 devices.
- A Program read is attempted above user code space. This occurs when user code attempts to branch to an address above 0xFBFF in Bank 3 on C8051F580/1/2/3/8/9 or any address in Bank 3 on C8051F584/5/6/7-F590/1 devices.
- A Flash read, write or erase attempt is restricted due to a Flash security setting (see Section “15.3. Security Options” on page 141).
- A Flash read, write, or erase is attempted when the VDD Monitor is not enabled to the high threshold and set as a reset source

The FERROR bit (RSTSRC.6) is set following a Flash error reset. The state of the RST pin is unaffected by this reset.

## 17.8. Software Reset

Software may force a reset by writing a 1 to the SWRSF bit (RSTSRC.4). The SWRSF bit will read 1 following a software forced reset. The state of the RST pin is unaffected by this reset.

## 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.

# 21. Local Interconnect Network (LIN0)

**Important Note:** This chapter assumes an understanding of the Local Interconnect Network (LIN) protocol. For more information about the LIN protocol, including specifications, please refer to the LIN consortium (http://www.lin-subbus.org).

LIN is an asynchronous, serial communications interface used primarily in automotive networks. The Silicon Laboratories LIN controller is compliant to the 2.1 Specification, implements a complete hardware LIN interface and includes the following features:

- Selectable Master and Slave modes.
- Automatic baud rate option in slave mode.
- The internal oscillator is accurate to within 0.5% of 24 MHz across the entire temperature range and for VDD voltages greater than or equal to the minimum output of the on-chip voltage regulator, so an external oscillator is not necessary for master mode operation for most systems.

**Note:** The minimum system clock (SYSCLK) required when using the LIN controller is 8 MHz.

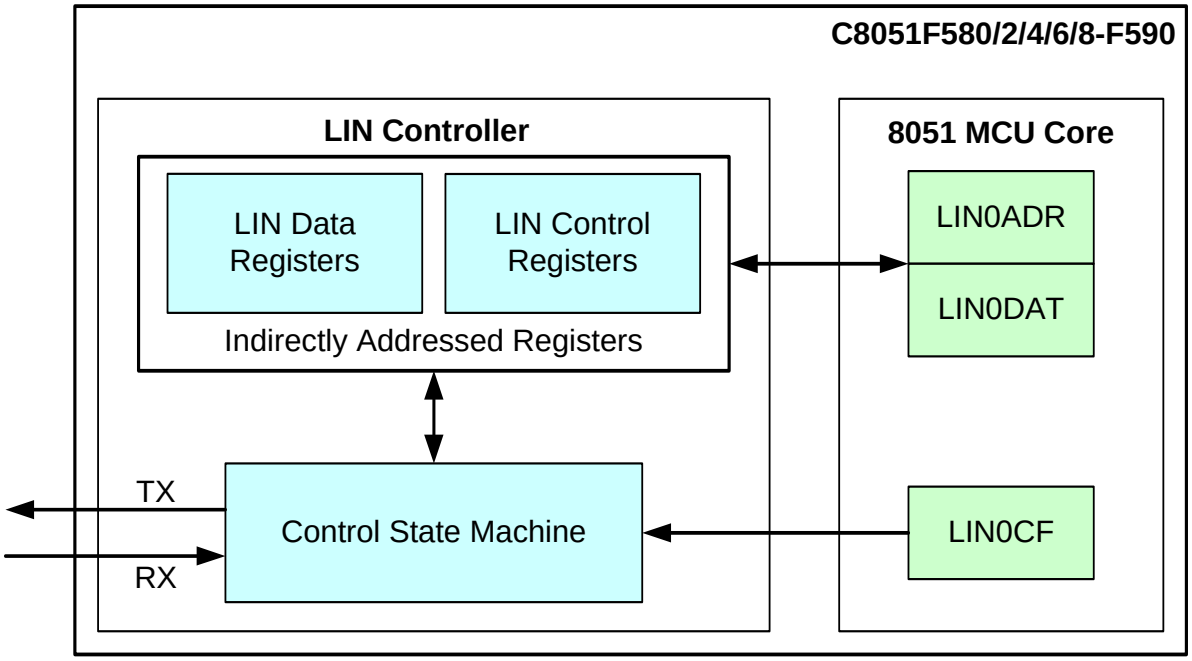



**Figure 21.1. LIN Block Diagram**

The LIN controller has four main components:

- LIN Access Registers—Provide the interface between the MCU core and the LIN controller.
- LIN Data Registers—Where transmitted and received message data bytes are stored.
- LIN Control Registers—Control the functionality of the LIN interface.
- Control State Machine and Bit Streaming Logic—Contains the hardware that serializes messages and controls the bus timing of the controller.

## LIN Register Definition 21.11. LIN0ID: LIN0 Identifier Register

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

Indirect Address = 0x0E

| Bit | Name | Function |
|---|---|---|
| 7:6 | Unused | Read = 00b; Write = Don't Care. |
| 5:0 | ID[5:0] | **LIN Identifier Bits.**<br>These bits form the data identifier.<br><br>If the LINSIZE bits (LIN0SIZE[3:0]) are 1111b, bits ID[5:4] are used to determine the data size and are interpreted as follows:<br>00: 2 bytes<br>01: 2 bytes<br>10: 4 bytes<br>11: 8 bytes |

# 23. SMBus

The SMBus I/O interface is a two-wire, bi-directional serial bus. The SMBus is compliant with the System Management Bus Specification, version 1.1, and compatible with the I2C serial bus. Reads and writes to the interface by the system controller are byte oriented with the SMBus interface autonomously controlling the serial transfer of the data. Data can be transferred at up to 1/20th of the system clock as a master or slave (this can be faster than allowed by the SMBus specification, depending on the system clock used). A method of extending the clock-low duration is available to accommodate devices with different speed capabilities on the same bus.

The SMBus interface may operate as a master and/or slave, and may function on a bus with multiple masters. The SMBus provides control of SDA (serial data), SCL (serial clock) generation and synchronization, arbitration logic, and START/STOP control and generation. A block diagram of the SMBus peripheral and the associated SFRs is shown in Figure 23.1.

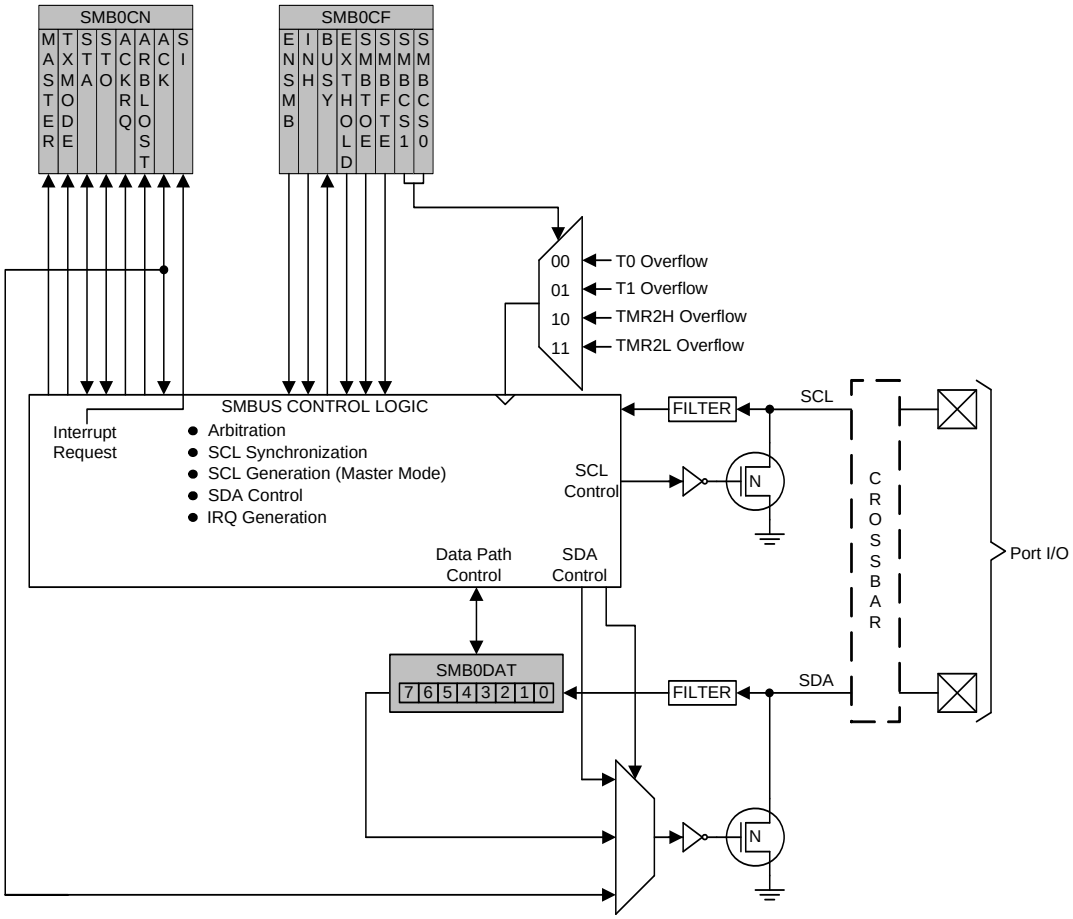



**Figure 23.1. SMBus Block Diagram**

## 24.2. Data Format

UART0 has a number of available options for data formatting. Data transfers begin with a start bit (logic low), followed by the data bits (sent LSB-first), a parity or extra bit (if selected), and end with one or two stop bits (logic high). The data length is variable between 5 and 8 bits. A parity bit can be appended to the data, and automatically generated and detected by hardware for even, odd, mark, or space parity. The stop bit length is selectable between 1 and 2 bit times, and a multi-processor communication mode is available for implementing networked UART buses. All of the data formatting options can be configured using the SMOD0 register, shown in SFR Definition 24.2. Figure 24.2 shows the timing for a UART0 transaction without parity or an extra bit enabled. Figure 24.3 shows the timing for a UART0 transaction with parity enabled (PE0 = 1). Figure 24.4 is an example of a UART0 transaction when the extra bit is enabled (XBE0 = 1). Note that the extra bit feature is not available when parity is enabled, and the second stop bit is only an option for data lengths of 6, 7, or 8 bits.

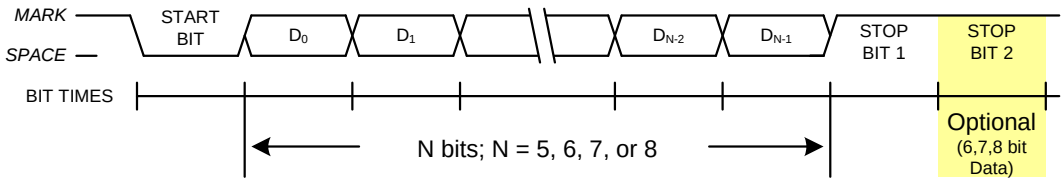



**Figure 24.2. UART0 Timing Without Parity or Extra Bit**

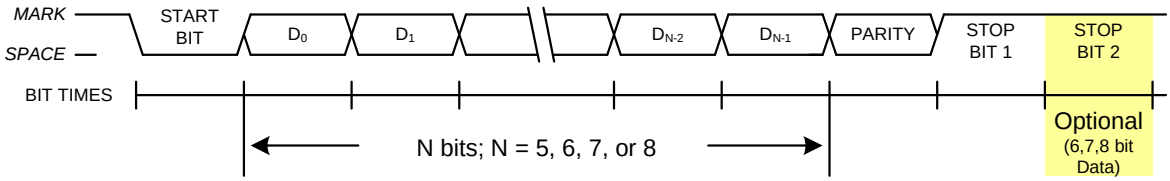



**Figure 24.3. UART0 Timing With Parity**

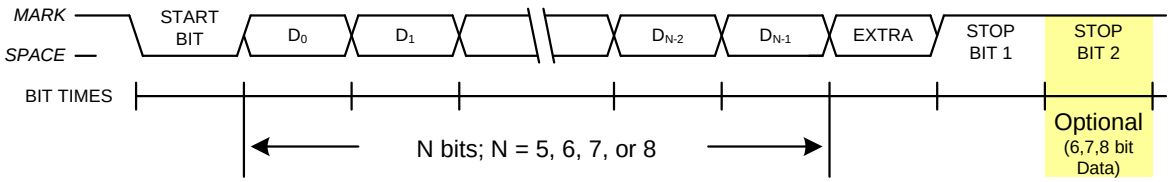
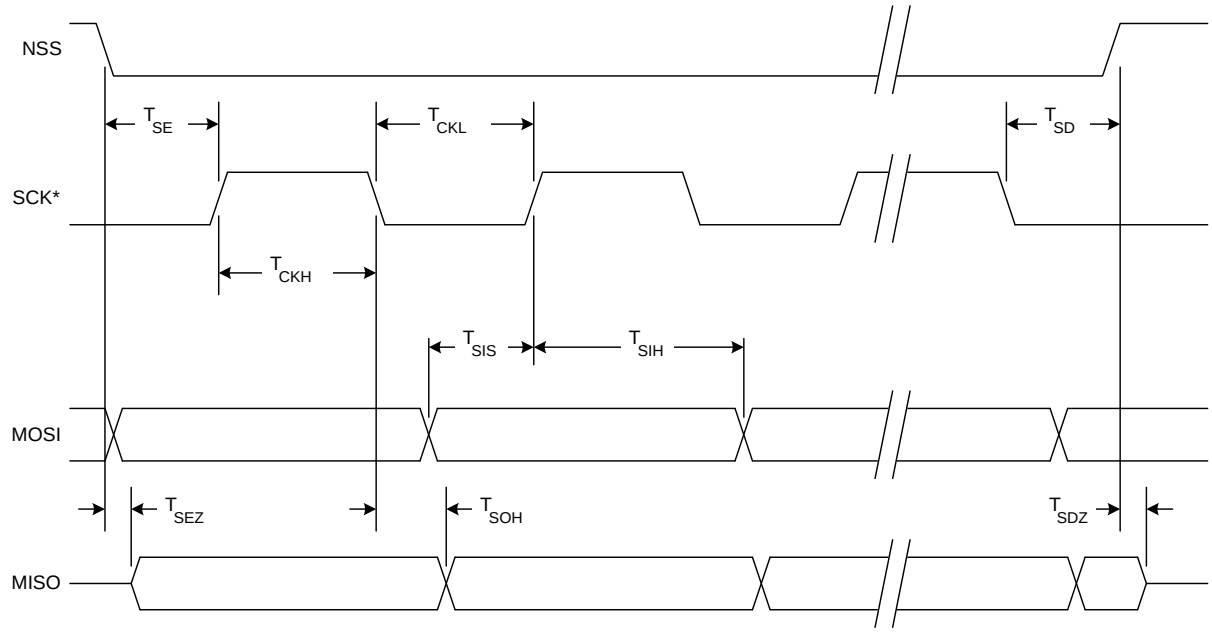
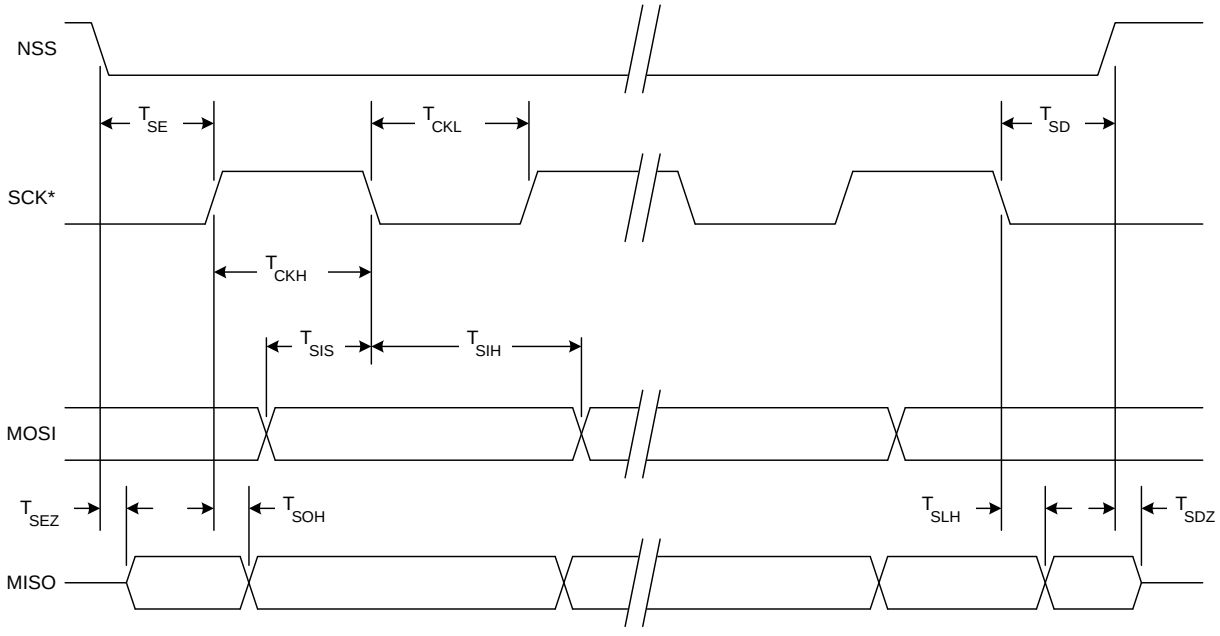



**Figure 24.4. UART0 Timing With Extra Bit**




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

**Figure 26.10. SPI Slave Timing (CKPHA = 0)**




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

**Figure 26.11. SPI Slave Timing (CKPHA = 1)**

## SFR Definition 28.7. PCA0CPLn: PCA0 Capture Module Low Byte

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | PCA0CPn[7:0] | | | | | | | |
| 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 Addresses: PCA0CPL0 = 0xFB, PCA0CPL1 = 0xE9, PCA0CPL2 = 0xEB, PCA0CPL3 = 0xED, PCA0CPL4 = 0xFD, PCA0CPL5 = 0xCE; SFR Page (all registers) = 0x00

| Bit | Name | Function |
|---|---|---|
| 7:0 | PCA0CPn[7:0] | **PCA0 Capture Module Low Byte.**<br>The PCA0CPLn register holds the low byte (LSB) of the 16-bit capture module n. This register address also allows access to the low byte of the corresponding PCA0 channel's auto-reload value for 9, 10, or 11-bit PWM mode. The ARSEL bit in register PCA0PWM controls which register is accessed. |
| **Note:** A write to this register will clear the module's ECOMn bit to a 0. | | |

## SFR Definition 28.8. PCA0CPHn: PCA0 Capture Module High Byte

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | PCA0CPn[15:8] | | | | | | | |
| 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 Addresses: PCA0CPH0 = 0xFC, PCA0CPH1 = 0xEA, PCA0CPH2 = 0xEC, PCA0CPH3 = 0xEE, PCA0CPH4 = 0xFE, PCA0CPH5 = 0xCF; SFR Page (all registers) = 0x00

| Bit | Name | Function |
|---|---|---|
| 7:0 | PCA0CPn[15:8] | **PCA0 Capture Module High Byte.**<br>The PCA0CPHn register holds the high byte (MSB) of the 16-bit capture module n. This register address also allows access to the high byte of the corresponding PCA0 channel's auto-reload value for 9, 10, or 11-bit PWM mode. The ARSEL bit in register PCA0PWM controls which register is accessed. |
| **Note:** A write to this register will set the module's ECOMn bit to a 1. | | |

## C2 Register Definition 30.2. DEVICEID: C2 Device ID

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

C2 Address = 0xFD; SFR Address = 0xFD; SFR Page = 0x0F

| Bit | Name | Function |
|---|---|---|
| 7:0 | DEVICEID[7:0] | **Device ID.**<br>This read-only register returns the 8-bit device ID: 0x20 (C8051F58x/F59x). |

## C2 Register Definition 30.3. REVID: C2 Revision ID

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

C2 Address = 0xFE; SFR Address = 0xFE; SFR Page = 0x0F

| Bit | Name | Function |
|---|---|---|
| 7:0 | REVID[7:0] | **Revision ID.**<br>This read-only register returns the 8-bit revision ID. For example: 0x00 = Revision A. |