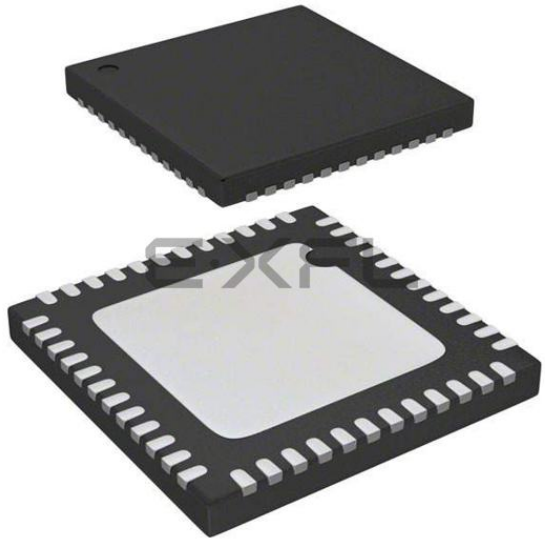

Welcome to E-XFL.COM

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	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/c8051f584-im

Table of Contents

List of Tables

Table 2.1. Product Selection Guide

Ordering Part Number	Flash Memory (kB)	CAN2.0B	LIN2.1	Digital Port I/Os	External Memory Interface	Package	Ordering Part Number	Flash Memory (kB)	CAN2.0B	LIN2.1	Digital Port I/Os	External Memory Interface	Package
C8051F580-IQ	128	✓	✓	40	✓	QFP-48	C8051F585-IQ	96	—	—	40	✓	QFP-48
C8051F580-IM	128	✓	✓	40	✓	QFN-48	C8051F585-IM	96	—	—	40	✓	QFN-48
C8051F581-IQ	128	—	—	40	✓	QFP-48	C8051F586-IQ	96	✓	✓	25	—	QFP-32
C8051F581-IM	128	—	—	40	✓	QFN-48	C8051F586-IM	96	✓	✓	25	—	QFN-32
C8051F582-IQ	128	✓	✓	25	—	QFP-32	C8051F587-IQ	96	—	—	25	—	QFP-32
C8051F582-IM	128	✓	✓	25	—	QFN-32	C8051F587-IM	96	—	—	25	—	QFN-32
C8051F583-IQ	128	—	—	25	—	QFP-32	C8051F588-IM	128	✓	✓	33	✓	QFN-40
C8051F583-IM	128	—	—	25	—	QFN-32	C8051F589-IM	128	—	—	33	✓	QFN-40
C8051F584-IQ	96	✓	✓	40	✓	QFP-48	C8051F590-IM	96	✓	✓	33	✓	QFN-40
C8051F584-IM	96	✓	✓	40	✓	QFN-48	C8051F591-IM	96	—	—	33	✓	QFN-40

Note: The suffix of the part number indicates the device rating and the package. All devices are RoHS compliant.

All of these devices are also available in an automotive version. For the automotive version, the -I in the ordering part number is replaced with -A. For example, the automotive version of the C8051F580-IM is the C8051F580-AM.

The -AM and -AQ devices receive full automotive quality production status, including AEC-Q100 qualification, registration with International Material Data System (IMDS) and Part Production Approval Process (PPAP) documentation. PPAP documentation is available at www.silabs.com with a registered and NDA approved user account. The -AM and -AQ devices enable high volume automotive OEM applications with their enhanced testing and processing. Please contact Silicon Labs sales for more information regarding –AM and -AQ devices for your automotive project.

Table 5.13. Comparator 0, 1 and 2 Electrical Characteristics

VIO = 1.8 to 5.25 V, –40 to +125 °C unless otherwise noted.

Parameter	Conditions	Min	Typ	Max	Units
Response Time: Mode 0, Vcm* = 1.5 V	CPn+ – CPn– = 100 mV	—	390	—	ns
	CPn+ – CPn– = –100 mV	—	430	—	ns
Response Time: Mode 1, Vcm* = 1.5 V	CPn+ – CPn– = 100 mV	—	620	—	ns
	CPn+ – CPn– = –100 mV	—	690	—	ns
Response Time: Mode 2, Vcm* = 1.5 V	CPn+ – CPn– = 100 mV	—	770	—	ns
	CP0+ – CP0– = –100 mV	—	860	—	ns
Response Time: Mode 3, Vcm* = 1.5 V	CPn+ – CPn– = 100 mV	—	3500	—	ns
	CPn+ – CPn– = –100 mV	—	3900	—	ns
Common-Mode Rejection Ratio		—	1.5	8.9	mV/V
Positive Hysteresis 1	CPnHYP1–0 = 00	-2	0	2	mV
Positive Hysteresis 2	CPnHYP1–0 = 01	2	6	10	mV
Positive Hysteresis 3	CPnHYP1–0 = 10	5	11	20	mV
Positive Hysteresis 4	CPnHYP1–0 = 11	13	22	40	mV
Negative Hysteresis 1	CPnHYN1–0 = 00	–2	0	2	mV
Negative Hysteresis 2	CPnHYN1–0 = 01	2	6	10	mV
Negative Hysteresis 3	CPnHYN1–0 = 10	5	11	20	mV
Negative Hysteresis 4	CPnHYN1–0 = 11	13	22	40	mV
Inverting or Non-Inverting Input Voltage Range		–0.25	—	V_{IO} + 0.25	V
Input Capacitance		—	8	—	pF
Input Offset Voltage		–10	—	+10	mV
Power Supply					
Power Supply Rejection		—	0.33	—	mV/V
Power-Up Time		—	3	—	µs
Supply Current at DC	Mode 0	—	6.1	20	µA
	Mode 1	—	3.2	10	µA
	Mode 2	—	2.5	7.5	µA
	Mode 3	—	0.5	3	µA

***Note:** Vcm is the common-mode voltage on CP0+ and CP0–.

If the internal voltage regulator is not used, the VREGIN input should be tied to VDD, as shown in Figure 10.2.

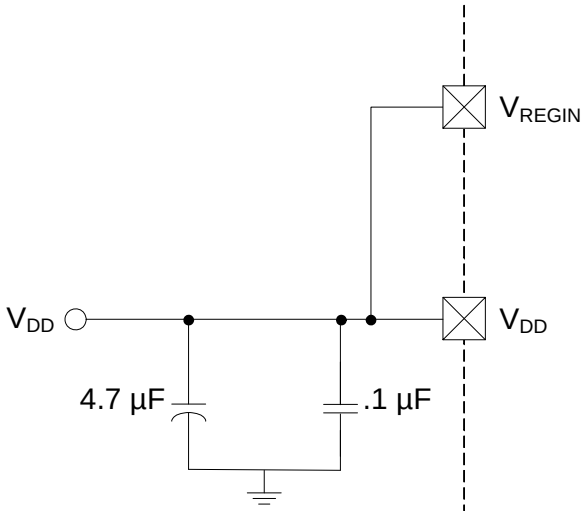

Figure 10.2. External Capacitors for Voltage Regulator Input/Output—Regulator Disabled

SFR Definition 10.1. 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.** 0: Voltage Regulator Enabled 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.** 0: Voltage Regulator Output is 2.1V. 1: Voltage Regulator Output is 2.6V.
3:1	Unused	Read = 000b. Write = Don't Care.
0	DROPOUT	**Voltage Regulator Dropout Indicator.** 0: Voltage Regulator is not in dropout 1: Voltage Regulator is in or near dropout.

SFR Definition 14.1. IE: Interrupt Enable

Bit	7	6	5	4	3	2	1	0
Name	EA	ESPI0	ET2	ES0	ET1	EX1	ET0	EX0
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 = 0xA8; Bit-Addressable; SFR Page = All Pages

Bit	Name	Function
7	EA	**Enable All Interrupts.** Globally enables/disables all interrupts. It overrides individual interrupt mask settings. 0: Disable all interrupt sources. 1: Enable each interrupt according to its individual mask setting.
6	ESPI0	**Enable Serial Peripheral Interface (SPI0) Interrupt.** This bit sets the masking of the SPI0 interrupts. 0: Disable all SPI0 interrupts. 1: Enable interrupt requests generated by SPI0.
5	ET2	**Enable Timer 2 Interrupt.** This bit sets the masking of the Timer 2 interrupt. 0: Disable Timer 2 interrupt. 1: Enable interrupt requests generated by the TF2L or TF2H flags.
4	ES0	**Enable UART0 Interrupt.** This bit sets the masking of the UART0 interrupt. 0: Disable UART0 interrupt. 1: Enable UART0 interrupt.
3	ET1	**Enable Timer 1 Interrupt.** This bit sets the masking of the Timer 1 interrupt. 0: Disable all Timer 1 interrupt. 1: Enable interrupt requests generated by the TF1 flag.
2	EX1	**Enable External Interrupt 1.** This bit sets the masking of External Interrupt 1. 0: Disable external interrupt 1. 1: Enable interrupt requests generated by the $\overline{\text{INT1}}$ input.
1	ET0	**Enable Timer 0 Interrupt.** This bit sets the masking of the Timer 0 interrupt. 0: Disable all Timer 0 interrupt. 1: Enable interrupt requests generated by the TF0 flag.
0	EX0	**Enable External Interrupt 0.** This bit sets the masking of External Interrupt 0. 0: Disable external interrupt 0. 1: Enable interrupt requests generated by the $\overline{\text{INT0}}$ input.

14.3. External Interrupts $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$

The $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ external interrupt sources are configurable as active high or low, edge or level sensitive. The IN0PL ($\overline{\text{INT0}}$ Polarity) and IN1PL ($\overline{\text{INT1}}$ Polarity) bits in the IT01CF register select active high or active low; the IT0 and IT1 bits in TCON (Section “27.1. Timer 0 and Timer 1” on page 287) select level or edge sensitive. The table below lists the possible configurations.

IT0	IN0PL	$\overline{\text{INT0}}$ Interrupt
1	0	Active low, edge sensitive
1	1	Active high, edge sensitive
0	0	Active low, level sensitive
0	1	Active high, level sensitive

IT1	IN1PL	$\overline{\text{INT1}}$ Interrupt
1	0	Active low, edge sensitive
1	1	Active high, edge sensitive
0	0	Active low, level sensitive
0	1	Active high, level sensitive

$\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ are assigned to Port pins as defined in the IT01CF register (see SFR Definition 14.7). Note that $\overline{\text{INT0}}$ and $\overline{\text{INT0}}$ Port pin assignments are independent of any Crossbar assignments. $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ will monitor their assigned Port pins without disturbing the peripheral that was assigned the Port pin via the Crossbar. To assign a Port pin only to $\overline{\text{INT0}}$ and/or $\overline{\text{INT1}}$, configure the Crossbar to skip the selected pin(s). This is accomplished by setting the associated bit in register XBR0 (see Section “20.3. Priority Crossbar Decoder” on page 192 for complete details on configuring the Crossbar).

IE0 (TCON.1) and IE1 (TCON.3) serve as the interrupt-pending flags for the $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ external interrupts, respectively. If an $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$ external interrupt is configured as edge-sensitive, the corresponding interrupt-pending flag is automatically cleared by the hardware when the CPU vectors to the ISR. When configured as level sensitive, the interrupt-pending flag remains logic 1 while the input is active as defined by the corresponding polarity bit (IN0PL or IN1PL); the flag remains logic 0 while the input is inactive. The external interrupt source must hold the input active until the interrupt request is recognized. It must then deactivate the interrupt request before execution of the ISR completes or another interrupt request will be generated.

15.1.2. Flash Lock and Key Functions

Flash writes and erases by user software are protected with a lock and key function. The Flash Lock and Key Register (FLKEY) must be written with the correct key codes, in sequence, before Flash operations may be performed. The key codes are: 0xA5, 0xF1. The timing does not matter, but the codes must be written in order. If the key codes are written out of order, or the wrong codes are written, Flash writes and erases will be disabled until the next system reset. Flash writes and erases will also be disabled if a Flash write or erase is attempted before the key codes have been written properly. The Flash lock resets after each write or erase; the key codes must be written again before a following Flash operation can be performed. The FLKEY register is detailed in SFR Definition 15.2.

15.1.3. Flash Erase Procedure

The Flash memory can be programmed by software using the MOVX write instruction with the address and data byte to be programmed provided as normal operands. Before writing to Flash memory using MOVX, Flash write operations must be enabled by doing the following: (1) setting the PSWE Program Store Write Enable bit (PSCTL.0) to logic 1 (this directs the MOVX writes to target Flash memory); and (2) Writing the Flash key codes in sequence to the Flash Lock register (FLKEY). The PSWE bit remains set until cleared by software.

A write to Flash memory can clear bits to logic 0 but cannot set them; only an erase operation can set bits to logic 1 in Flash. **A byte location to be programmed should be erased before a new value is written.** The Flash memory is organized in 512-byte pages. The erase operation applies to an entire page (setting all bytes in the page to 0xFF). To erase an entire 512-byte page, perform the following steps:

1. Disable interrupts (recommended).
2. If erasing a page in Banks 1, 2, or 3, set the COBANK[1:0] bits (register PSBANK) for the appropriate bank.
3. Set the FLEWT bit (register FLSCL).
4. Set the PSEE bit (register PSCTL).
5. Set the PSWE bit (register PSCTL).
6. Write the first key code to FLKEY: 0xA5.
7. Write the second key code to FLKEY: 0xF1.
8. Using the MOVX instruction, write a data byte to any location within the 512-byte page to be erased.
9. Clear the PSWE and PSEE bits.

15.1.4. Flash Write Procedure

Flash bytes are programmed by software with the following sequence:

1. Disable interrupts (recommended).
2. If writing to an address in Banks 1, 2, or 3, set the COBANK[1:0] (register PSBANK) for the appropriate bank.
3. Erase the 512-byte Flash page containing the target location, as described in Section 15.1.3.
4. Set the FLEWT bit (register FLSCL).
5. Set the PSWE bit (register PSCTL).
6. Clear the PSEE bit (register PSCTL).
7. Write the first key code to FLKEY: 0xA5.
8. Write the second key code to FLKEY: 0xF1.
9. Using the MOVX instruction, write a single data byte to the desired location within the 512-byte sector.
10. Clear the PSWE bit.

17.2. Power-Fail Reset/V_{DD} Monitor

When a power-down transition or power irregularity causes V_{DD} to drop below V_{RST}, the power supply monitor will drive the $\overline{RST}$ pin low and hold the CIP-51 in a reset state (see Figure 17.2). When V_{DD} returns to a level above V_{RST}, the CIP-51 will be released from the reset state. Note that even though internal data memory contents are not altered by the power-fail reset, it is impossible to determine if V_{DD} dropped below the level required for data retention. If the PORSF flag reads 1, the data may no longer be valid. The V_{DD} monitor is enabled after power-on resets. Its defined state (enabled/disabled) is not altered by any other reset source. For example, if the V_{DD} monitor is disabled by code and a software reset is performed, the V_{DD} monitor will still be disabled after the reset. **To protect the integrity of Flash contents, the V_{DD} monitor must be enabled to the higher setting (VDMLVL = 1) and selected as a reset source if software contains routines which erase or write Flash memory. If the V_{DD} monitor is not enabled and set to the high level, any erase or write performed on Flash memory will cause a Flash Error device reset.**

Important Note: If the V_{DD} monitor is being turned on from a disabled state, it should be enabled before it is selected as a reset source. Selecting the V_{DD} monitor as a reset source before it is enabled and stabilized may cause a system reset. Ensure that there are no delays between the time the V_{DD} monitor is enabled and when it is enabled as a reset source.

1. Enable the V_{DD} monitor (VDMEN bit in VDM0CN = 1). Ensure that there are no delays before step 2 is executed.
2. Select the V_{DD} monitor as a reset source (PORSF bit in RSTSRC = 1).

See Figure 17.2 for V_{DD} monitor timing; note that the power-on-reset delay is not incurred after a V_{DD} monitor reset. See Table 5.4 for complete electrical characteristics of the V_{DD} monitor.

When programming the Flash in-system, the V_{DD} Monitor must be set to the high threshold setting. For 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.

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 V_{DD} Monitor is always left in the low threshold setting (i.e. default value upon POR).

Note: The VDD Monitor may trigger on fast changes in voltage on the VDD pin, regardless of whether the voltage increased or decreased.

18.5. Memory Mode Selection

The external data memory space can be configured in one of four modes, shown in Figure 18.3, based on the EMIF Mode bits in the EMI0CF register (SFR Definition 18.2). These modes are summarized below. More information about the different modes can be found in Section “18.6. Timing” on page 167.

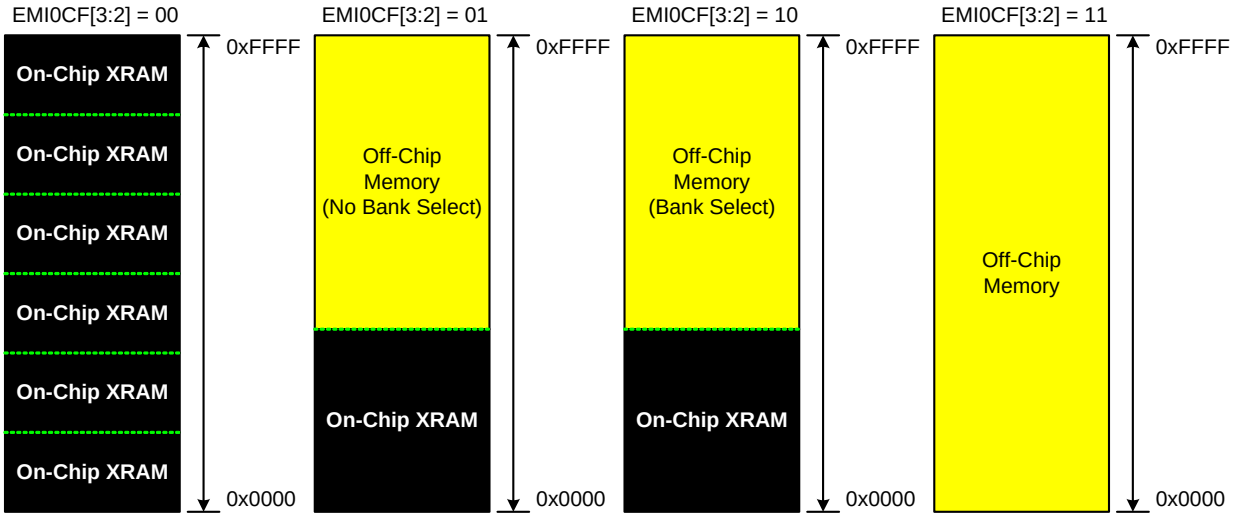

Figure 18.3. EMIF Operating Modes

18.5.1. Internal XRAM Only

When bits EMI0CF[3:2] are set to 00, all MOVX instructions will target the internal XRAM space on the device. Memory accesses to addresses beyond the populated space will wrap on 8 kB boundaries. As an example, the addresses 0x2000 and 0x4000 both evaluate to address 0x0000 in on-chip XRAM space.

- 8-bit MOVX operations use the contents of EMI0CN to determine the high-byte of the effective address and R0 or R1 to determine the low-byte of the effective address.
- 16-bit MOVX operations use the contents of the 16-bit DPTR to determine the effective address.

18.5.2. Split Mode without Bank Select

When bit EMI0CF.[3:2] are set to 01, the XRAM memory map is split into two areas, on-chip space and off-chip space.

- Effective addresses below the internal XRAM size boundary will access on-chip XRAM space.
- Effective addresses above the internal XRAM size boundary will access off-chip space.
- 8-bit MOVX operations use the contents of EMI0CN to determine whether the memory access is on-chip or off-chip. However, in the “No Bank Select” mode, an 8-bit MOVX operation will not drive the upper 8-bits A[15:8] of the Address Bus during an off-chip access. This allows the user to manipulate the upper address bits at will by setting the Port state directly via the port latches. This behavior is in contrast with “Split Mode with Bank Select” described below. The lower 8-bits of the Address Bus A[7:0] are driven, determined by R0 or R1.
- 16-bit MOVX operations use the contents of DPTR to determine whether the memory access is on-chip or off-chip, and unlike 8-bit MOVX operations, the full 16-bits of the Address Bus A[15:0] are driven during the off-chip transaction.

24. UART0

UART0 is an asynchronous, full duplex serial port offering a variety of data formatting options. A dedicated baud rate generator with a 16-bit timer and selectable prescaler is included, which can generate a wide range of baud rates (details in Section “24.1. Baud Rate Generator” on page 256). A received data FIFO allows UART0 to receive up to three data bytes before data is lost and an overflow occurs.

UART0 has six associated SFRs. Three are used for the Baud Rate Generator (SBCON0, SBRLH0, and SBRLL0), two are used for data formatting, control, and status functions (SCON0, SMOD0), and one is used to send and receive data (SBUF0). The single SBUF0 location provides access to both transmit and receive registers. **Writes to SBUF0 always access the Transmit register. Reads of SBUF0 always access the buffered Receive register; it is not possible to read data from the Transmit register.**

With UART0 interrupts enabled, an interrupt is generated each time a transmit is completed (TI0 is set in SCON0), or a data byte has been received (RI0 is set in SCON0). The UART0 interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine. They must be cleared manually by software, allowing software to determine the cause of the UART0 interrupt (transmit complete or receive complete). If additional bytes are available in the Receive FIFO, the RI0 bit cannot be cleared by software.

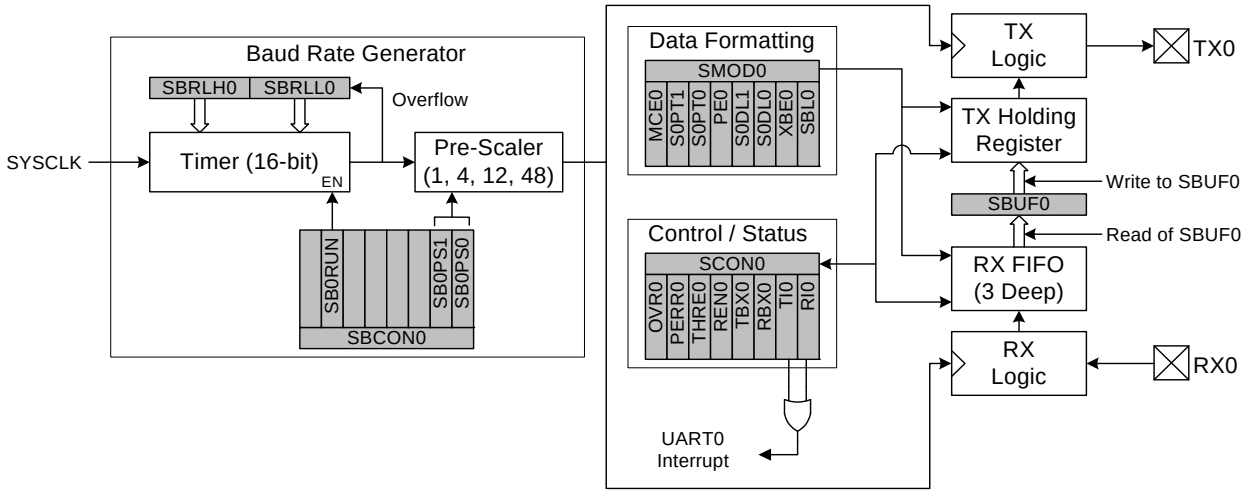

Figure 24.1. UART0 Block Diagram

24.1. Baud Rate Generator

The UART0 baud rate is generated by a dedicated 16-bit timer which runs from the controller’s core clock (SYSCLK) and has prescaler options of 1, 4, 12, or 48. The timer and prescaler options combined allow for a wide selection of baud rates over many clock frequencies.

The baud rate generator is configured using three registers: SBCON0, SBRLH0, and SBRLL0. The UART0 Baud Rate Generator Control Register (SBCON0, SFR Definition 24.4) enables or disables the baud rate generator, selects the clock source for the baud rate generator, and selects the prescaler value for the timer. The baud rate generator must be enabled for UART0 to function. Registers SBRLH0 and SBRLL0 contain a 16-bit reload value for the dedicated 16-bit timer. The internal timer counts up from the reload value on every clock tick. On timer overflows (0xFFFF to 0x0000), the timer is reloaded. The baud rate for UART0 is defined in Equation 24.1, where “BRG Clock” is the baud rate generator’s selected clock source. For reliable UART operation, it is recommended that the UART baud rate is not configured for baud rates faster than SYSCLK/16.

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.

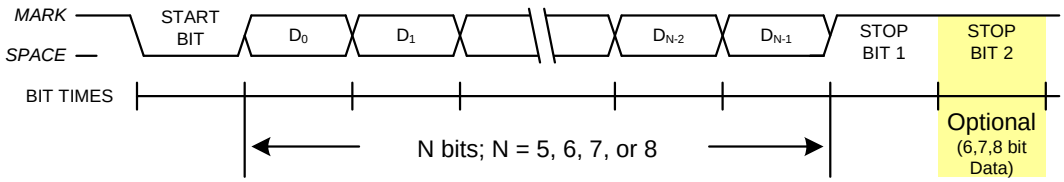

Figure 24.2. UART0 Timing Without Parity or Extra Bit

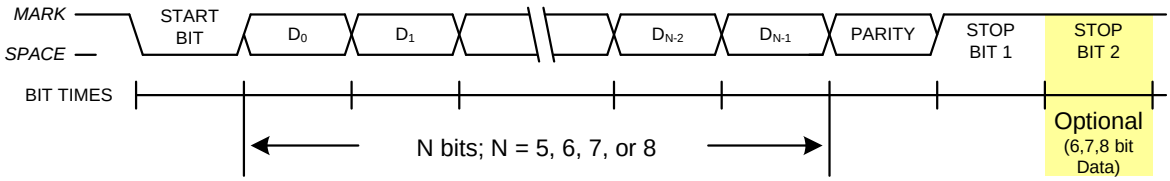

Figure 24.3. UART0 Timing With Parity

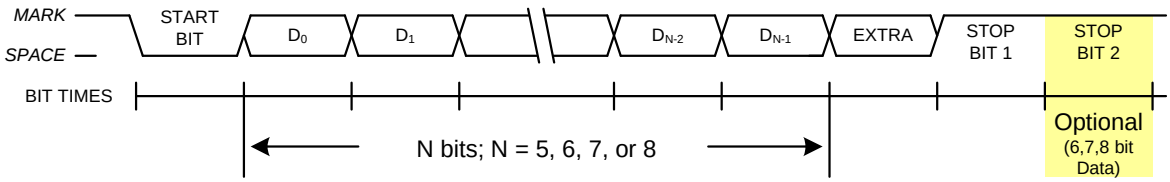

Figure 24.4. UART0 Timing With Extra Bit

Data transmission begins when an instruction writes a data byte to the SBUF1 register. The TI1 Transmit Interrupt Flag (SCON1.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN1 Receive Enable bit (SCON1.4) is set to 1. After the stop bit is received, the data byte will be loaded into the SBUF1 receive register if the following conditions are met: (1) RI1 must be logic 0, and (2) if MCE1 is logic 1, the 9th bit must be logic 1 (when MCE1 is logic 0, the state of the ninth data bit is unimportant). If these conditions are met, the eight bits of data are stored in SBUF1, the ninth bit is stored in RB81, and the RI1 flag is set to 1. If the above conditions are not met, SBUF1 and RB81 will not be loaded and the RI1 flag will not be set to 1. A UART1 interrupt will occur if enabled when either TI1 or RI1 is set to '1'.

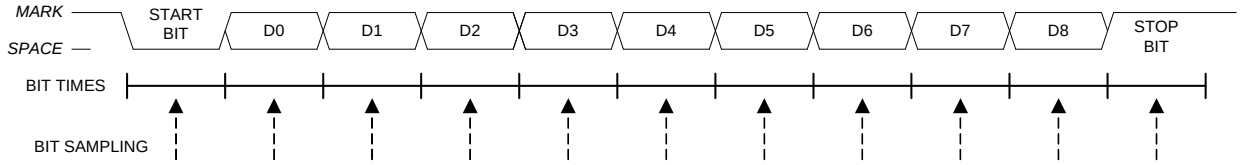

Figure 25.5. 9-Bit UART Timing Diagram

25.3. Multiprocessor Communications

9-Bit UART mode supports multiprocessor communication between a master processor and one or more slave processors by special use of the ninth data bit. When a master processor wants to transmit to one or more slaves, it first sends an address byte to select the target(s). An address byte differs from a data byte in that its ninth bit is logic 1; in a data byte, the ninth bit is always set to logic 0.

Setting the MCE1 bit (SCON1.5) of a slave processor configures its UART such that when a stop bit is received, the UART will generate an interrupt only if the ninth bit is logic 1 (RB81 = 1) signifying an address byte has been received. In the UART interrupt handler, software will compare the received address with the slave's own assigned 8-bit address. If the addresses match, the slave will clear its MCE1 bit to enable interrupts on the reception of the following data byte(s). Slaves that weren't addressed leave their MCE1 bits set and do not generate interrupts on the reception of the following data bytes, thereby ignoring the data. Once the entire message is received, the addressed slave resets its MCE1 bit to ignore all transmissions until it receives the next address byte.

Multiple addresses can be assigned to a single slave and/or a single address can be assigned to multiple slaves, thereby enabling "broadcast" transmissions to more than one slave simultaneously. The master processor can be configured to receive all transmissions or a protocol can be implemented such that the master/slave role is temporarily reversed to enable half-duplex transmission between the original master and slave(s).

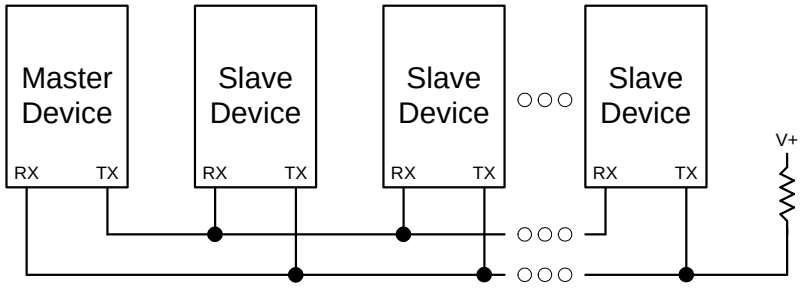

Figure 25.6. UART Multi-Processor Mode Interconnect Diagram

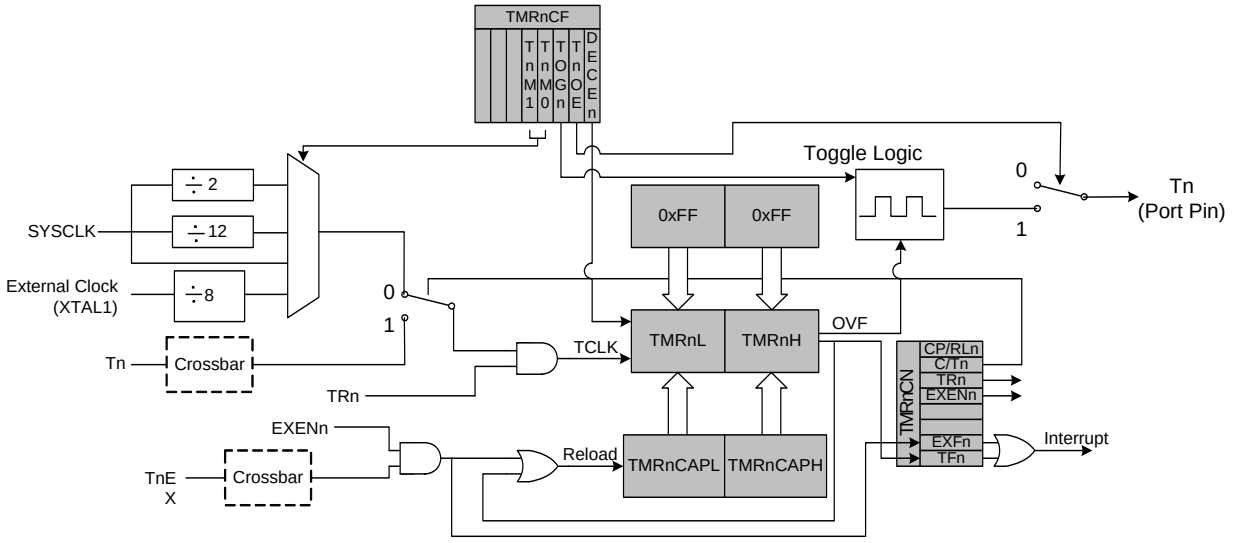

Figure 27.11. Timer 4 and 5 Auto Reload and Toggle Mode Block Diagram

27.4.4. Toggle Output Mode

Timers 4 and 5 have the capability to toggle the state of their respective output port pins (T4 or T5) to produce a 50% duty cycle waveform output. The port pin state will change upon the overflow or underflow of the respective timer (depending on whether the timer is counting *up* or *down*). The toggle frequency is determined by the clock source of the timer and the values loaded into TMRnCAPH and TMRnCAPL. When counting down, the auto-reload value for the timer is 0xFFFF, and underflow will occur when the value in the timer matches the value stored in TMRnCAPH:TMRCAPL. When counting up, the auto-reload value for the timer is TMRnCAPH:TMRCAPL, and overflow will occur when the value in the timer transitions from 0xFFFF to the reload value.

To output a square wave, the timer is placed in reload mode (the Capture/Reload Select Bit in TMRnCN and the Timer/Counter Select Bit in TMRnCN are cleared to 0). The timer output is enabled by setting the Timer Output Enable Bit in TMRnCF to 1. The timer should be configured via the timer clock source and reload/underflow values such that the timer overflow/underflows at 1/2 the desired output frequency. The port pin assigned by the crossbar as the timer's output pin should be configured as a digital output (see **Section "20. Port Input/Output" on page 188**). Setting the timer's Run Bit (TRn) to 1 will start the toggle of the pin. A Read/Write of the Timer's Toggle Output State Bit (TMRnCF.2) is used to read the state of the toggle output, or to force a value of the output. This is useful when it is desired to start the toggle of a pin in a known state, or to force the pin into a desired state when the toggle mode is halted.

$$F_{sq} = \frac{F_{TCLK}}{2 \times (65536 - TMRnCAP)}$$

Equation 27.1. Square Wave Frequency

SFR Definition 27.20. TMRnCAPL: Timer 4 and 5 Capture Register Low Byte

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

TMR4CAPL SFR Address = 0xCA; TMR5CAPL SFR Address = 0x92; SFR Page = 0x10

Bit	Name	Function
7:0	TMRnCAPL[7:0]	**Timer n Reload Register Low Byte.** TMRnCAPL captures the low byte of Timer 4 and 5 when Timer 4 and 5 are configured in capture mode. When Timer 4 and 5 are configured in auto-reload mode, it holds the low byte of the reload value.

SFR Definition 27.21. TMRnCAPH: Timer 4 and 5 Capture Register High Byte

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

TMR4CAPH SFR Address = 0xCB; TMR5CAPH SFR Address = 0x93; SFR Page = 0x10

Bit	Name	Function
7:0	TMRnCAPH[7:0]	**Timer n Reload Register High Byte.** TMRnCAPH captures the high byte of Timer 4 and 5 when Timer 4 and 5 are configured in capture mode. When Timer 4 and 5 are configured in auto-reload mode, it holds the high byte of the reload value.

28.5. Register Descriptions for PCA0

Following are detailed descriptions of the special function registers related to the operation of the PCA.

SFR Definition 28.1. PCA0CN: PCA0 Control

Bit	7	6	5	4	3	2	1	0
Name	CF	CR	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0
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 = 0xD8; Bit-Addressable; SFR Page = 0x00

Bit	Name	Function
7	CF	**PCA0 Counter/Timer Overflow Flag.** Set by hardware when the PCA0 Counter/Timer overflows from 0xFFFF to 0x0000. When the Counter/Timer Overflow (CF) interrupt is enabled, setting this bit causes the CPU to vector to the PCA0 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
6	CR	**PCA0 Counter/Timer Run Control.** This bit enables/disables the PCA0 Counter/Timer. 0: PCA0 Counter/Timer disabled. 1: PCA0 Counter/Timer enabled.
5	CCF5	**PCA0 Module 5 Capture/Compare Flag.** This bit is set by hardware when a match or capture occurs. When the CCF5 interrupt is enabled, setting this bit causes the CPU to vector to the PCA0 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
4	CCF4	**PCA0 Module 4 Capture/Compare Flag.** This bit is set by hardware when a match or capture occurs. When the CCF4 interrupt is enabled, setting this bit causes the CPU to vector to the PCA0 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
3	CCF3	**PCA0 Module 3 Capture/Compare Flag.** This bit is set by hardware when a match or capture occurs. When the CCF3 interrupt is enabled, setting this bit causes the CPU to vector to the PCA0 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
2	CCF2	**PCA0 Module 2 Capture/Compare Flag.** This bit is set by hardware when a match or capture occurs. When the CCF2 interrupt is enabled, setting this bit causes the CPU to vector to the PCA0 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
1	CCF1	**PCA0 Module 1 Capture/Compare Flag.** This bit is set by hardware when a match or capture occurs. When the CCF1 interrupt is enabled, setting this bit causes the CPU to vector to the PCA0 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
0	CCF0	**PCA0 Module 0 Capture/Compare Flag.** This bit is set by hardware when a match or capture occurs. When the CCF0 interrupt is enabled, setting this bit causes the CPU to vector to the PCA0 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

29.3.1. Edge-triggered Capture Mode

In this mode, a valid transition on the CEXn pin causes PCA1 to capture the value of the PCA1 counter/timer and load it into the corresponding module's 16-bit capture/compare register (PCA1CPLn and PCA1CPHn). The CAPP1n and CAPN1n bits in the PCA1CPMn 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 PCA1CN 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 CAPP1n and CAPN1n 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.

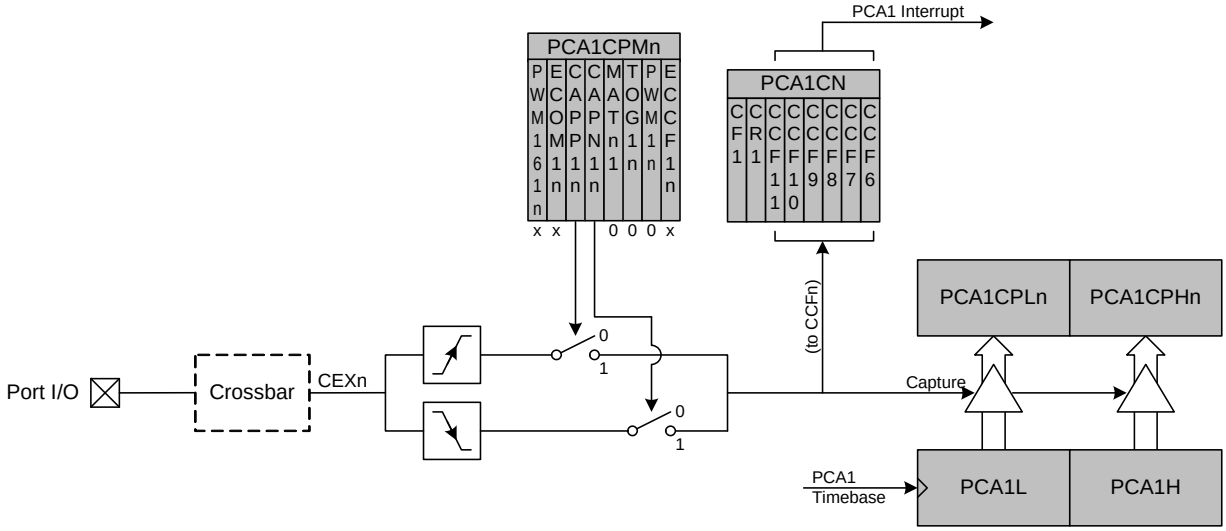

Figure 29.4. PCA1 Capture Mode Diagram

Note: The CEXn input signal must remain high or low for at least 2 system clock cycles to be recognized by the hardware.

29.3.4. Frequency Output Mode

Frequency Output Mode produces a programmable-frequency square wave on the module's associated CEXn pin. The capture/compare module high byte holds the number of PCA1 clocks to count before the output is toggled. The frequency of the square wave is then defined by Equation 29.1.

$$F_{CEXn} = \frac{F_{PCA}}{2 \times PCA1CPHn}$$

Note: A value of 0x00 in the PCA1CPHn register is equal to 256 for this equation.

Equation 29.1. Square Wave Frequency Output

Where F_{PCA} is the frequency of the clock selected by the CPS12–0 bits in the PCA1 mode register, PCA1MD. The lower byte of the capture/compare module is compared to the PCA1 counter low byte; on a match, CEXn is toggled and the offset held in the high byte is added to the matched value in PCA1CPLn. Frequency Output Mode is enabled by setting the ECOM1n, TOG1n, and PWM1n bits in the PCA1CPMn register. Note that the MAT1n bit should normally be set to 0 in this mode. If the MAT1n bit is set to 1, the CCFn flag for the channel will be set when the 16-bit PCA1 counter and the 16-bit capture/compare register for the channel are equal.

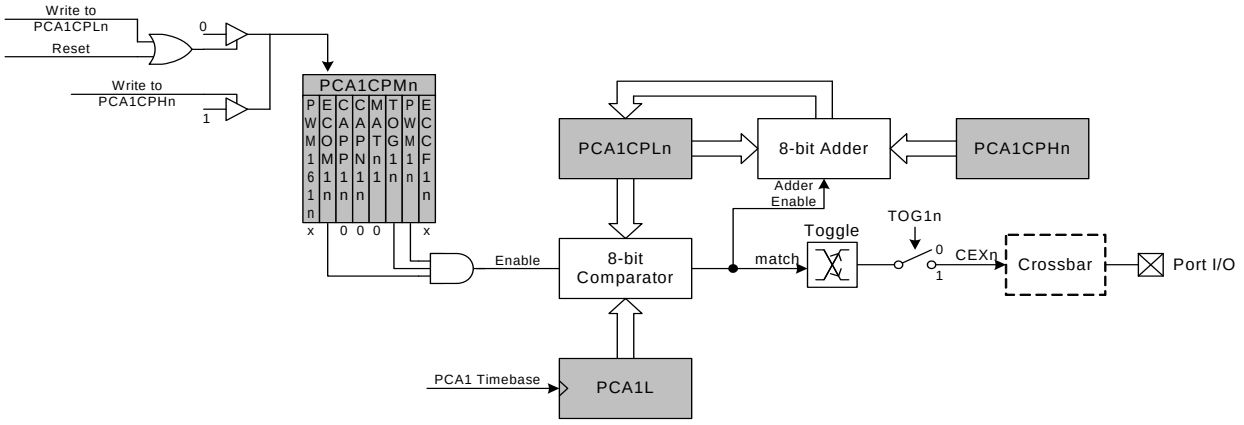

Figure 29.7. PCA1 Frequency Output Mode

SFR Definition 29.3. PCA1PWM: PCA1 PWM Configuration

Bit	7	6	5	4	3	2	1	0
Name	ARSEL1	ECOV1	COVF1				CLSEL1[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 = 0xDA; SFR Page = 0x0F

Bit	Name	Function
7	ARSEL1	**Auto-Reload Register Select.** This bit selects whether to read and write the normal PCA1 capture/compare registers (PCA1CPn), 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. 0: Read/Write Capture/Compare Registers at PCA1CPHn and PCA1CPLn. 1: Read/Write Auto-Reload Registers at PCA1CPHn and PCA1CPLn.
6	ECOV1	**Cycle Overflow Interrupt Enable.** This bit sets the masking of the Cycle Overflow Flag (COVF1) interrupt. 0: COVF1 will not generate PCA1 interrupts. 1: A PCA1 interrupt will be generated when COVF1 is set.
5	COVF1	**Cycle Overflow Flag.** This bit indicates an overflow of the 8th, 9th, 10th, or 11th bit of the main PCA1 counter (PCA1). 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. 0: No overflow has occurred since the last time this bit was cleared. 1: An overflow has occurred since the last time this bit was cleared.
4:2	Unused	Read = 000b; Write = Don't care.
1:0	CLSEL1[1:0]	**Cycle Length Select.** 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. 00: 8 bits. 01: 9 bits. 10: 10 bits. 11: 11 bits.

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.** 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.** This read-only register returns the 8-bit revision ID. For example: 0x00 = Revision A.