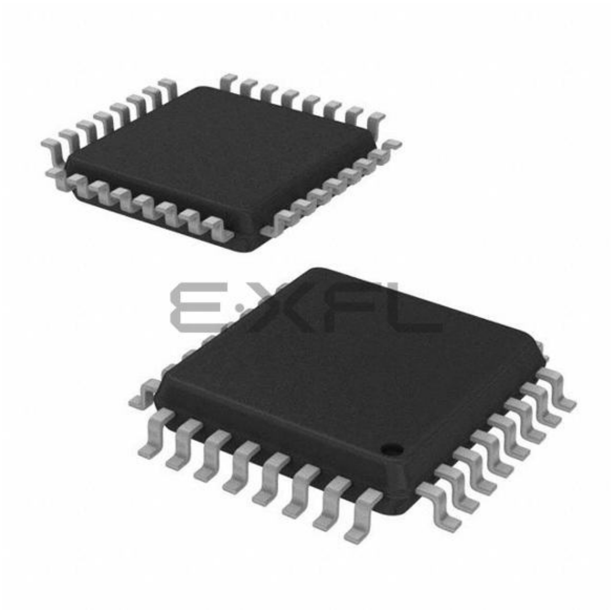


Welcome to E-XFL.COM

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), LINbus, SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	25
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.25K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.25V
Data Converters	A/D 25x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f544-iqr

5.1. Modes of Operation

In a typical system, ADC0 is configured using the following steps:

1. If a gain adjustment is required, refer to Section “5.3. Selectable Gain” on page 35.
2. Choose the start of conversion source.
3. Choose Normal Mode or Burst Mode operation.
4. If Burst Mode, choose the ADC0 Idle Power State and set the Power-Up Time.
5. Choose the tracking mode. Note that Pre-Tracking Mode can only be used with Normal Mode.
6. Calculate the required settling time and set the post convert-start tracking time using the AD0TK bits.
7. Choose the repeat count.
8. Choose the output word justification (Right-Justified or Left-Justified).
9. Enable or disable the End of Conversion and Window Comparator Interrupts.

5.1.1. Starting a Conversion

A conversion can be initiated in one of four ways, depending on the programmed states of the ADC0 Start of Conversion Mode bits (AD0CM1–0) in register ADC0CN. Conversions may be initiated by one of the following:

- Writing a 1 to the AD0BUSY bit of register ADC0CN
- A rising edge on the CNVSTR input signal (pin P0.1)
- A Timer 1 overflow (i.e., timed continuous conversions)
- A Timer 2 overflow (i.e., timed continuous conversions)

Writing a 1 to AD0BUSY provides software control of ADC0 whereby conversions are performed "on-demand.” During conversion, the AD0BUSY bit is set to logic 1 and reset to logic 0 when the conversion is complete. The falling edge of AD0BUSY triggers an interrupt (when enabled) and sets the ADC0 interrupt flag (AD0INT). Note: When polling for ADC conversion completions, the ADC0 interrupt flag (AD0INT) should be used. Converted data is available in the ADC0 data registers, ADC0H:ADC0L, when bit AD0INT is logic 1. Note that when Timer 2 overflows are used as the conversion source, Low Byte overflows are used if Timer2 is in 8-bit mode; High byte overflows are used if Timer 2 is in 16-bit mode. See Section “23. Timers” on page 227 for timer configuration.

Important Note About Using CNVSTR: The CNVSTR input pin also functions as Port pin P0.1. When the CNVSTR input is used as the ADC0 conversion source, Port pin P0.1 should be skipped by the Digital Crossbar. To configure the Crossbar to skip P0.1, set to 1 Bit1 in register P0SKIP. See Section “18. Port Input/Output” on page 147 for details on Port I/O configuration.

5.1.2. Tracking Modes

Each ADC0 conversion must be preceded by a minimum tracking time for the converted result to be accurate. ADC0 has three tracking modes: Pre-Tracking, Post-Tracking, and Dual-Tracking. Pre-Tracking Mode provides the minimum delay between the convert start signal and end of conversion by tracking continuously before the convert start signal. This mode requires software management in order to meet minimum tracking requirements. In Post-Tracking Mode, a programmable tracking time starts after the convert start signal and is managed by hardware. Dual-Tracking Mode maximizes tracking time by tracking before and after the convert start signal. Figure 5.2 shows examples of the three tracking modes.

Pre-Tracking Mode is selected when AD0TM is set to 10b. Conversions are started immediately following the convert start signal. ADC0 is tracking continuously when not performing a conversion. Software must allow at least the minimum tracking time between each end of conversion and the next convert start signal. The minimum tracking time must also be met prior to the first convert start signal after ADC0 is enabled.

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 6.9, may be required after changing MUX settings. See the settling time requirements described in Section "5.2.1. Settling Time Requirements" on page 34.

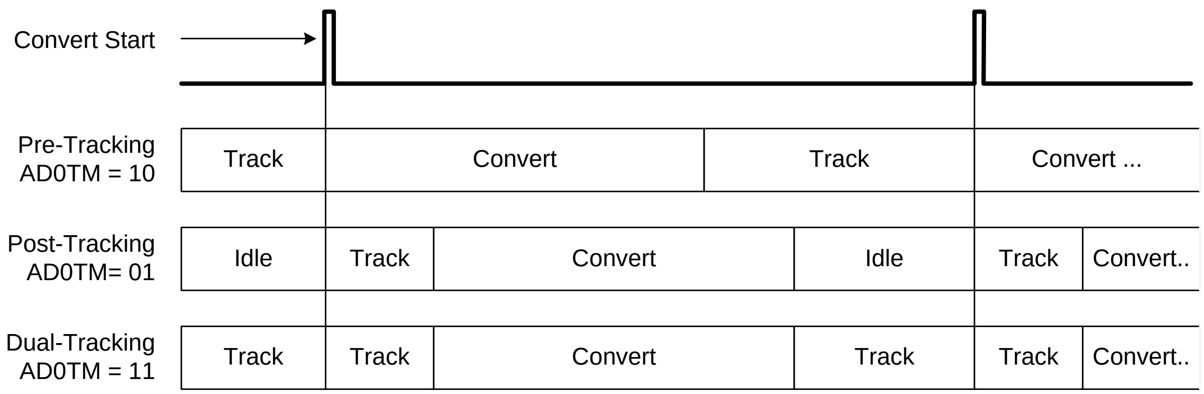


Figure 5.2. ADC0 Tracking Modes

5.1.3. Timing

ADC0 has a maximum conversion speed specified in Table 6.9. 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 6.9.

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 5.3 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.

10.4. Serial Number Special Function Registers (SFRs)

The C8051F54x devices include four SFRs, SN0 through SN3, that are pre-programmed during production with a unique, 32-bit serial number. The serial number provides a unique identification number for each device and can be read from the application firmware. If the serial number is not used in the application, these four registers can be used as general purpose SFRs.

SFR Definition 10.7. SNn: Serial Number n

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

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

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

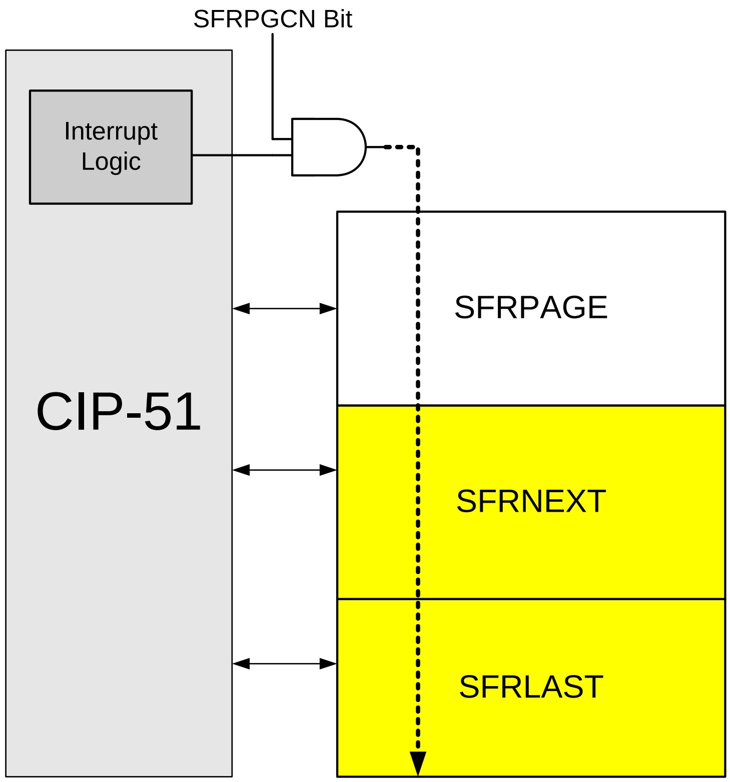


Figure 12.1. SFR Page Stack

Automatic hardware switching of the SFR Page on interrupts may be enabled or disabled as desired using the SFR Automatic Page Control Enable Bit located in the SFR Page Control Register (SFR0CN). This function defaults to “enabled” upon reset. In this way, the autoswitching function will be enabled unless disabled in software.

A summary of the SFR locations (address and SFR page) are provided in Table 12.2 in the form of an SFR memory map. Each memory location in the map has an SFR page row, denoting the page in which that SFR resides. Certain SFRs are accessible from ALL SFR pages, and are denoted by the “(ALL PAGES)” designation. For example, the Port I/O registers P0, P1, P2, and P3 all have the “(ALL PAGES)” designation, indicating these SFRs are accessible from all SFR pages regardless of the SFRPAGE register value.

SFR Definition 12.1. SFR0CN: SFR Page Control

Bit	7	6	5	4	3	2	1	0
Name								SFRPGEN
Type	R	R	R	R	R	R	R	R/W
Reset	0	0	0	0	0	0	0	1

SFR Address = 0x84; SFR Page = 0x0F

Bit	Name	Function
7:1	Unused	Read = 0000000b; Write = Don’t Care
0	SFRPGEN	**SFR Automatic Page Control Enable.** Upon interrupt, the C8051 Core will vector to the specified interrupt service routine and automatically switch the SFR page to the corresponding peripheral or function’s SFR page. This bit is used to control this autopaging function. 0: SFR Automatic Paging disabled. The C8051 core will not automatically change to the appropriate SFR page (i.e., the SFR page that contains the SFRs for the peripheral/function that was the source of the interrupt). 1: SFR Automatic Paging enabled. Upon interrupt, the C8051 will switch the SFR page to the page that contains the SFRs for the peripheral or function that is the source of the interrupt.

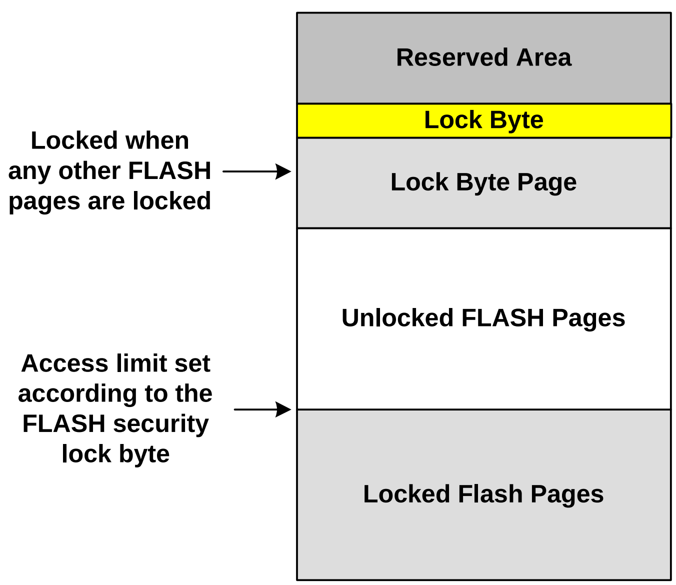
14.2. Non-volatile Data Storage

The Flash memory can be used for non-volatile data storage as well as program code. This allows data such as calibration coefficients to be calculated and stored at run time. Data is written using the MOVX write instruction and read using the MOVC instruction. Note: MOVX read instructions always target XRAM.

14.3. Security Options

The CIP-51 provides security options to protect the Flash memory from inadvertent modification by software as well as to prevent the viewing of proprietary program code and constants. The Program Store Write Enable (bit PSWE in register PSCTL) and the Program Store Erase Enable (bit PSEE in register PSCTL) bits protect the Flash memory from accidental modification by software. PSWE must be explicitly set to 1 before software can modify the Flash memory; both PSWE and PSEE must be set to 1 before software can erase Flash memory. Additional security features prevent proprietary program code and data constants from being read or altered across the C2 interface.

A Security Lock Byte located at the last byte of Flash user space offers protection of the Flash program memory from access (reads, writes, or erases) by unprotected code or the C2 interface. The Flash security mechanism allows the user to lock *n* 512-byte Flash pages, starting at page 0 (addresses 0x0000 to 0x01FF), where *n* is the ones complement number represented by the Security Lock Byte. **Note that the page containing the Flash Security Lock Byte is unlocked when no other Flash pages are locked (all bits of the Lock Byte are 1) and locked when any other Flash pages are locked (any bit of the Lock Byte is 0).** See example in Figure 14.1.



Security Lock Byte:	11111101b
1s Complement:	00000010b
Flash pages locked:	3 (First two Flash pages + Lock Byte Page)

Figure 14.1. Flash Program Memory Map

14.4. Flash Write and Erase Guidelines

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

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

14.4.1. V_{DD} Maintenance and the V_{DD} monitor

1. If the system power supply is subject to voltage or current "spikes," add sufficient transient protection devices to the power supply to ensure that the supply voltages listed in the Absolute Maximum Ratings table are not exceeded.
2. Enable the on-chip V_{DD} monitor and enable the V_{DD} monitor as a reset source as early in code as possible. This should be the first set of instructions executed after the Reset Vector. For C-based systems, this will involve modifying the startup code added by the C compiler. See your compiler documentation for more details. Make certain that there are no delays in software between enabling the V_{DD} monitor and enabling the V_{DD} monitor as a reset source. Code examples showing this can be found in “AN201: Writing to Flash from Firmware", available from the Silicon Laboratories web site.
3. As an added precaution, explicitly enable the V_{DD} monitor and enable the V_{DD} monitor as a reset source inside the functions that write and erase Flash memory. The V_{DD} monitor enable instructions should be placed just after the instruction to set PSWE to a 1, but before the Flash write or erase operation instruction.
4. Make certain that all writes to the RSTSRC (Reset Sources) register use direct assignment operators and explicitly DO NOT use the bit-wise operators (such as AND or OR). For example, "RSTSRC = 0x02" is correct. "RSTSRC |= 0x02" is incorrect.
5. Make certain that all writes to the RSTSRC register explicitly set the PORSF bit to a 1. Areas to check are initialization code which enables other reset sources, such as the Missing Clock Detector or Comparator, for example, and instructions which force a Software Reset. A global search on "RSTSRC" can quickly verify this.

14.4.2. PSWE Maintenance

1. Reduce the number of places in code where the PSWE bit (b0 in PSCTL) is set to a 1. There should be exactly one routine in code that sets PSWE to a 1 to write Flash bytes and one routine in code that sets PSWE and PSEE both to a 1 to erase Flash pages.
2. Minimize the number of variable accesses while PSWE is set to a 1. Handle pointer address updates and loop variable maintenance outside the "PSWE = 1;... PSWE = 0;" area. Code examples showing this can be found in ”AN201: Writing to Flash from Firmware" available from the Silicon Laboratories web site.
3. Disable interrupts prior to setting PSWE to a 1 and leave them disabled until after PSWE has been reset to '0'. Any interrupts posted during the Flash write or erase operation will be serviced in priority order after the Flash operation has been completed and interrupts have been re-enabled by software.
4. Make certain that the Flash write and erase pointer variables are not located in XRAM. See your compiler documentation for instructions regarding how to explicitly locate variables in different memory areas.
5. Add address bounds checking to the routines that write or erase Flash memory to ensure that a routine called with an illegal address does not result in modification of the Flash.

15. Power Management Modes

The C8051F54x devices have three software programmable power management modes: Idle, Stop, and Suspend. Idle mode and Stop mode are part of the standard 8051 architecture, while Suspend mode is an enhanced power-saving mode implemented by the high-speed oscillator peripheral.

Idle mode halts the CPU while leaving the peripherals and clocks active. In Stop mode, the CPU is halted, all interrupts and timers (except the Missing Clock Detector) are inactive, and the internal oscillator is stopped (analog peripherals remain in their selected states; the external oscillator is not affected). Suspend mode is similar to Stop mode in that the internal oscillator and CPU are halted, but the device can wake on events such as a Port Match or Comparator low output. Since clocks are running in Idle mode, power consumption is dependent upon the system clock frequency and the number of peripherals left in active mode before entering Idle. Stop mode and Suspend mode consume the least power because the majority of the device is shut down with no clocks active. SFR Definition 15.1 describes the Power Control Register (PCON) used to control the C8051F54x devices' Stop and Idle power management modes. Suspend mode is controlled by the SUSPEND bit in the OSCICN register (SFR Definition 17.2).

Although the C8051F54x has Idle, Stop, and Suspend modes available, more control over the device power can be achieved by enabling/disabling individual peripherals as needed. Each analog peripheral can be disabled when not in use and placed in low power mode. Digital peripherals, such as timers or serial buses, draw little power when they are not in use. Turning off oscillators lowers power consumption considerably, at the expense of reduced functionality.

15.1. Idle Mode

Setting the Idle Mode Select bit (PCON.0) causes the hardware to halt the CPU and enter Idle mode as soon as the instruction that sets the bit completes execution. All internal registers and memory maintain their original data. All analog and digital peripherals can remain active during Idle mode.

Idle mode is terminated when an enabled interrupt is asserted or a reset occurs. The assertion of an enabled interrupt will cause the Idle Mode Selection bit (PCON.0) to be cleared and the CPU to resume operation. The pending interrupt will be serviced and the next instruction to be executed after the return from interrupt (RETI) will be the instruction immediately following the one that set the Idle Mode Select bit. If Idle 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: If the instruction following the write of the IDLE bit is a single-byte instruction and an interrupt occurs during the execution phase of the instruction that sets the IDLE bit, the CPU may not wake from Idle mode when a future interrupt occurs. Therefore, instructions that set the IDLE bit should be followed by an instruction that has two or more opcode bytes, for example:

```
// in ‘C’:
PCON |= 0x01;                    // set IDLE bit
PCON = PCON;                     // ... followed by a 3-cycle dummy instruction

; in assembly:
ORL PCON, #01h                   ; set IDLE bit
MOV PCON, PCON                   ; ... followed by a 3-cycle dummy instruction
```

If enabled, the Watchdog Timer (WDT) will eventually cause an internal watchdog reset and thereby terminate the Idle mode. This feature protects the system from an unintended permanent shutdown in the event of an inadvertent write to the PCON register. If this behavior is not desired, the WDT may be disabled by software prior to entering the Idle mode if the WDT was initially configured to allow this operation. This provides the opportunity for additional power savings, allowing the system to remain in the Idle mode indefinitely, waiting for an external stimulus to wake up the system. Refer to Section “16.6. PCA Watchdog Timer Reset” on page 133 for more information on the use and configuration of the WDT.

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. In some applications, this reset may be undesirable. If this is not desirable in the application, a delay should be introduced between enabling the monitor and selecting it as a reset source. The procedure for enabling the V_{DD} monitor and configuring it as a reset source from a disabled state is as follows:

1. Enable the V_{DD} monitor (VDMEN bit in VDM0CN = 1).
2. If necessary, wait for the V_{DD} monitor to stabilize (see Table 6.4 for the V_{DD} Monitor turn-on time).
   Note: This delay should be omitted if software contains routines that erase or write Flash memory.
3. Select the V_{DD} monitor as a reset source (PORSF bit in RSTSRC = 1).

See Figure 16.2 for V_{DD} monitor timing; note that the power-on-reset delay is not incurred after a V_{DD} monitor reset. See Table 6.4 for complete electrical characteristics of the V_{DD} monitor.

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 V_{DD} Monitor is set to the high threshold setting *and* if the MCU receives a non-power on reset (POR), the MCU will remain in reset until a POR occurs (i.e., V_{DD} Monitor will keep the device in reset). A POR will force the V_{DD} 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).

When programming the Flash in-system, the V_{DD} Monitor must be set to the high threshold setting. For the highest system reliability, the time the V_{DD} Monitor is set to the high threshold setting should be minimized (e.g., setting the V_{DD} Monitor to the high threshold setting just before the Flash write operation and then changing it back to the low threshold setting immediately after the Flash write operation).

SFR Definition 16.1. VDM0CN: V_{DD} Monitor Control

Bit	7	6	5	4	3	2	1	0
Name	VDMEN	VDDSTAT	VDMLVL					
Type	R/W	R	R/W	R	R	R	R	R
Reset	Varies	Varies	0	0	0	0	0	0

SFR Address = 0xFF; SFR Page = 0x00

Bit	Name	Function
7	VDMEN	**V_{DD} Monitor Enable.** This bit turns the V_{DD} monitor circuit on/off. The V_{DD} Monitor cannot generate system resets until it is also selected as a reset source in register RSTSRC (SFR Definition 16.2). Selecting the V_{DD} monitor as a reset source before it has stabilized may generate a system reset. In systems where this reset would be undesirable, a delay should be introduced between enabling the V_{DD} Monitor and selecting it as a reset source. See Table 6.4 for the minimum V_{DD} Monitor turn-on time. 0: V_{DD} Monitor Disabled. 1: V_{DD} Monitor Enabled.
6	VDDSTAT	**V_{DD} Status.** This bit indicates the current power supply status (V_{DD} Monitor output). 0: V_{DD} is at or below the V_{DD} monitor threshold. 1: V_{DD} is above the V_{DD} monitor threshold.
5	VDMLVL	**V_{DD} Monitor Level Select.** 0: V_{DD} Monitor Threshold is set to VRST-LOW 1: V_{DD} Monitor Threshold is set to VRST-HIGH. This setting is required for any system includes code that writes to and/or erases Flash.
4:0	Unused	Read = 00000b; Write = Don't care.

16.3. External Reset

The external $\overline{RST}$ pin provides a means for external circuitry to force the device into a reset state. Asserting an active-low signal on the $\overline{RST}$ pin generates a reset; an external pullup and/or decoupling of the $\overline{RST}$ pin may be necessary to avoid erroneous noise-induced resets. See Table 6.4 for complete $\overline{RST}$ pin specifications. The PINRSF flag (RSTSRC.0) is set on exit from an external reset.

16.4. Missing Clock Detector Reset

The Missing Clock Detector (MCD) is a one-shot circuit that is triggered by the system clock. If the system clock remains high or low for more than the time specified in Table 6.4, "Reset Electrical Characteristics," on page 52, the one-shot will time out and generate a reset. After a MCD reset, the MCDRSF flag (RSTSRC.2) will read 1, signifying the MCD as the reset source; otherwise, this bit reads 0. Writing a 1 to the MCDRSF bit enables the Missing Clock Detector; writing a 0 disables it. The state of the $\overline{RST}$ pin is unaffected by this reset.

SFR Definition 19.3. LIN0CF: LIN0 Control Mode Register

Bit	7	6	5	4	3	2	1	0
Name	LINEN	MODE	ABAUD					
Type	R/W	R/W	R/W	R	R	R	R	R
Reset	0	1	1	0	0	0	0	0

SFR Address = 0xC9; SFR Page = 0x0F

Bit	Name	Function
7	LINEN	**LIN Interface Enable Bit.** 0: LIN0 is disabled. 1: LIN0 is enabled.
6	MODE	**LIN Mode Selection Bit.** 0: LIN0 operates in slave mode. 1: LIN0 operates in master mode.
5	ABAUD	**LIN Mode Automatic Baud Rate Selection.** This bit only has an effect when the MODE bit is configured for slave mode. 0: Manual baud rate selection is enabled. 1: Automatic baud rate selection is enabled.
4:0	Unused	Read = 00000b; Write = Don't Care

SFR Definition 21.2. SMOD0: Serial Port 0 Control

Bit	7	6	5	4	3	2	1	0
Name	MCE0	S0PT[1:0]		PE0	S0DL[1:0]		XBE0	SBL0
Type	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	1	0	0

SFR Address = 0xA9; SFR Page = 0x00

Bit	Name	Function
7	MCE0	**Multiprocessor Communication Enable.** 0: RI0 will be activated if stop bit(s) are 1. 1: RI0 will be activated if stop bit(s) and extra bit are 1. Extra bit must be enabled using XBE0.
6:5	S0PT[1:0]	**Parity Type Select Bits.** 00: Odd Parity 01: Even Parity 10: Mark Parity 11: Space Parity.
4	PE0	**Parity Enable.** This bit enables hardware parity generation and checking. The parity type is selected by bits S0PT[1:0] when parity is enabled. 0: Hardware parity is disabled. 1: Hardware parity is enabled.
3:2	S0DL[1:0]	**Data Length.** 00: 5-bit data 01: 6-bit data 10: 7-bit data 11: 8-bit data
1	XBE0	**Extra Bit Enable.** When enabled, the value of TBX0 will be appended to the data field 0: Extra Bit is disabled. 1: Extra Bit is enabled.
0	SBL0	**Stop Bit Length.** 0: Short—stop bit is active for one bit time 1: Long—stop bit is active for two bit times (data length = 6, 7, or 8 bits), or 1.5 bit times (data length = 5 bits).

23.1. Timer 0 and Timer 1

Each timer is implemented as a 16-bit register accessed as two separate bytes: a low byte (TL0 or TL1) and a high byte (TH0 or TH1). The Counter/Timer Control register (TCON) is used to enable Timer 0 and Timer 1 as well as indicate status. Timer 0 interrupts can be enabled by setting the ET0 bit in the IE register (Section “13.2. Interrupt Register Descriptions” on page 108); Timer 1 interrupts can be enabled by setting the ET1 bit in the IE register (Section “13.2. Interrupt Register Descriptions” on page 108). Both counter/timers operate in one of four primary modes selected by setting the Mode Select bits T1M1–T0M0 in the Counter/Timer Mode register (TMOD). Each timer can be configured independently. Each operating mode is described below.

23.1.1. Mode 0: 13-bit Counter/Timer

Timer 0 and Timer 1 operate as 13-bit counter/timers in Mode 0. The following describes the configuration and operation of Timer 0. However, both timers operate identically, and Timer 1 is configured in the same manner as described for Timer 0.

The TH0 register holds the eight MSBs of the 13-bit counter/timer. TL0 holds the five LSBs in bit positions TL0.4–TL0.0. The three upper bits of TL0 (TL0.7–TL0.5) are indeterminate and should be masked out or ignored when reading. As the 13-bit timer register increments and overflows from 0x1FFF (all ones) to 0x0000, the timer overflow flag TF0 (TCON.5) is set and an interrupt will occur if Timer 0 interrupts are enabled.

The C/T0 bit (TMOD.2) selects the counter/timer's clock source. When C/T0 is set to logic 1, high-to-low transitions at the selected Timer 0 input pin (T0) increment the timer register (Refer to Section “18.3. Priority Crossbar Decoder” on page 150 for information on selecting and configuring external I/O pins). Clearing C/T selects the clock defined by the T0M bit (CKCON.3). When T0M is set, Timer 0 is clocked by the system clock. When T0M is cleared, Timer 0 is clocked by the source selected by the Clock Scale bits in CKCON (see SFR Definition 23.1).

Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3) is logic 0 or the input signal $\overline{\text{INT0}}$ is active as defined by bit IN0PL in register IT01CF (see SFR Definition 13.7). Setting GATE0 to 1 allows the timer to be controlled by the external input signal $\overline{\text{INT0}}$ (see Section “13.2. Interrupt Register Descriptions” on page 108), facilitating pulse width measurements.

TR0	GATE0	$\overline{\text{INT0}}$	Counter/Timer
0	X	X	Disabled
1	0	X	Enabled
1	1	0	Disabled
1	1	1	Enabled
Note: X = Don't Care			

Setting TR0 does not force the timer to reset. The timer registers should be loaded with the desired initial value before the timer is enabled.

TL1 and TH1 form the 13-bit register for Timer 1 in the same manner as described above for TL0 and TH0. Timer 1 is configured and controlled using the relevant TCON and TMOD bits just as with Timer 0. The input signal $\overline{\text{INT1}}$ is used with Timer 1; the $\overline{\text{INT1}}$ polarity is defined by bit IN1PL in register IT01CF (see SFR Definition 13.7).

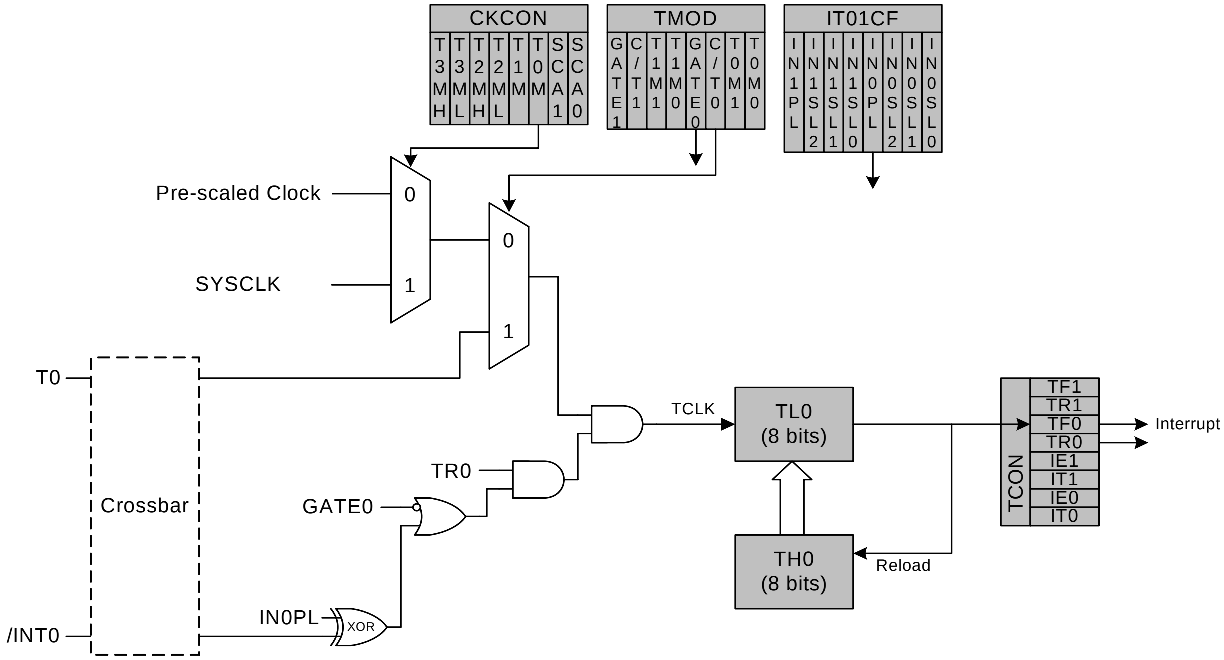


Figure 23.2. T0 Mode 2 Block Diagram

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

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

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

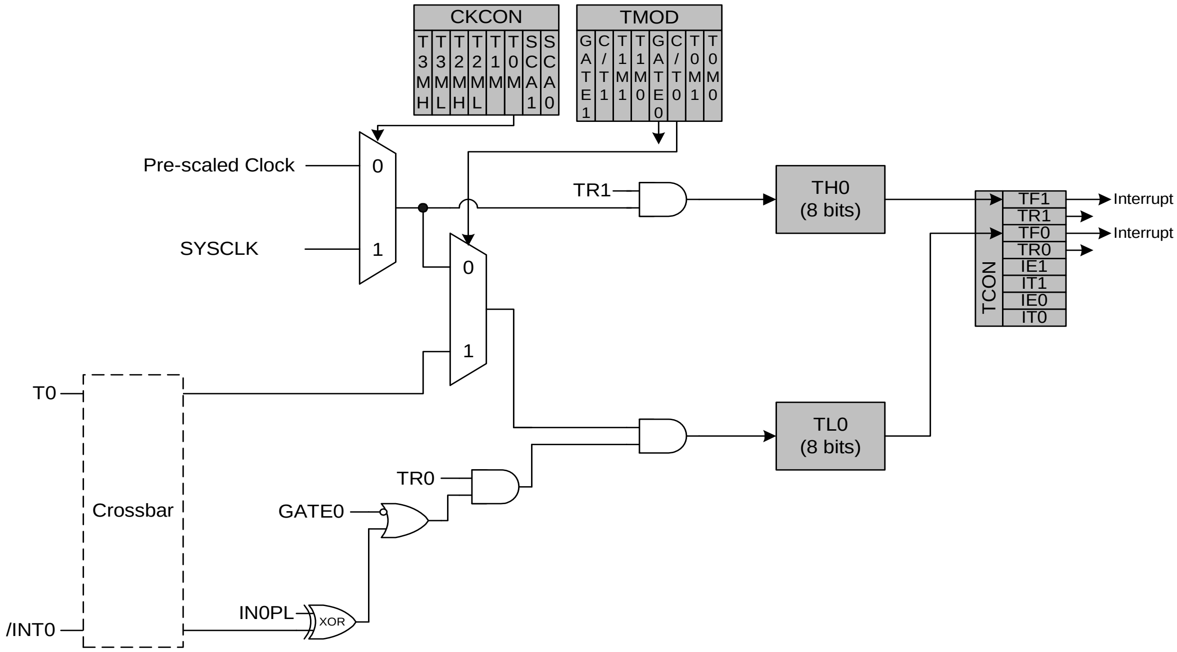


Figure 23.3. T0 Mode 3 Block Diagram

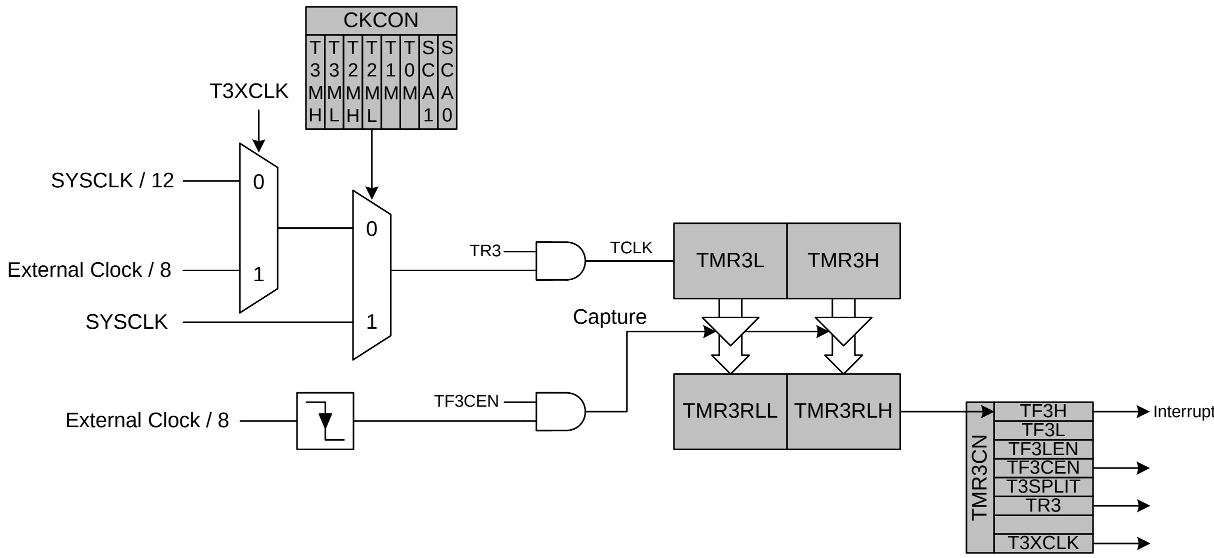


Figure 23.9. Timer 3 External Oscillator Capture Mode Block Diagram

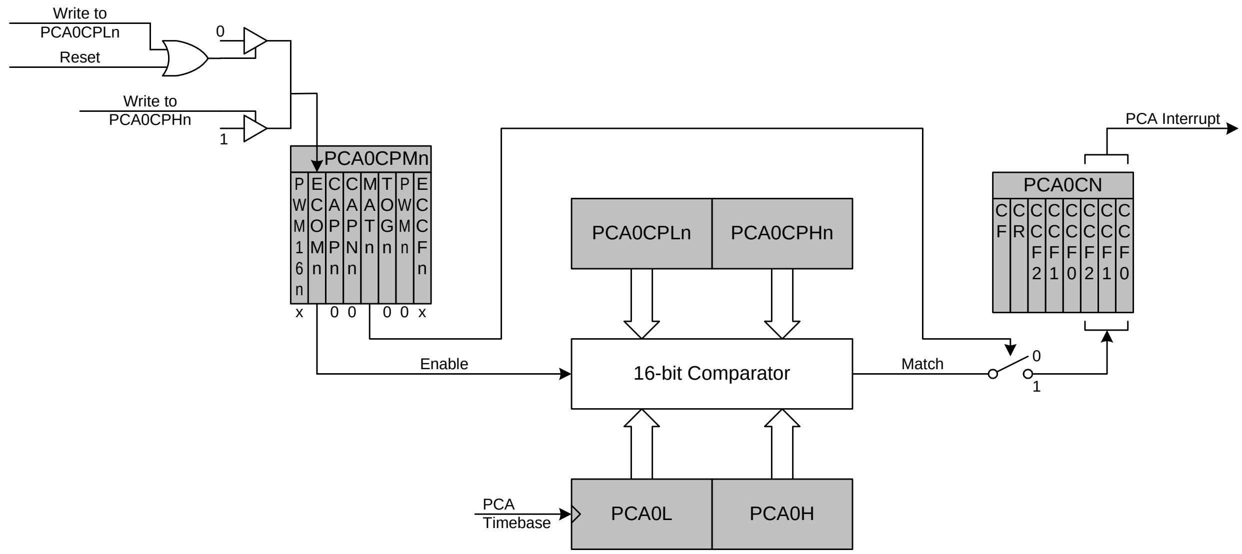


Figure 24.5. PCA Software Timer Mode Diagram

24.3.3. High-Speed Output Mode

In High-Speed Output mode, a module's associated CEXn pin is toggled each time a match occurs between the PCA Counter and the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn). When a match 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. Setting the TOGn, MATn, and ECOMn bits in the PCA0CPMn register enables the High-Speed Output mode. If ECOMn is cleared, the associated pin will retain its state, and not toggle on the next match event.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

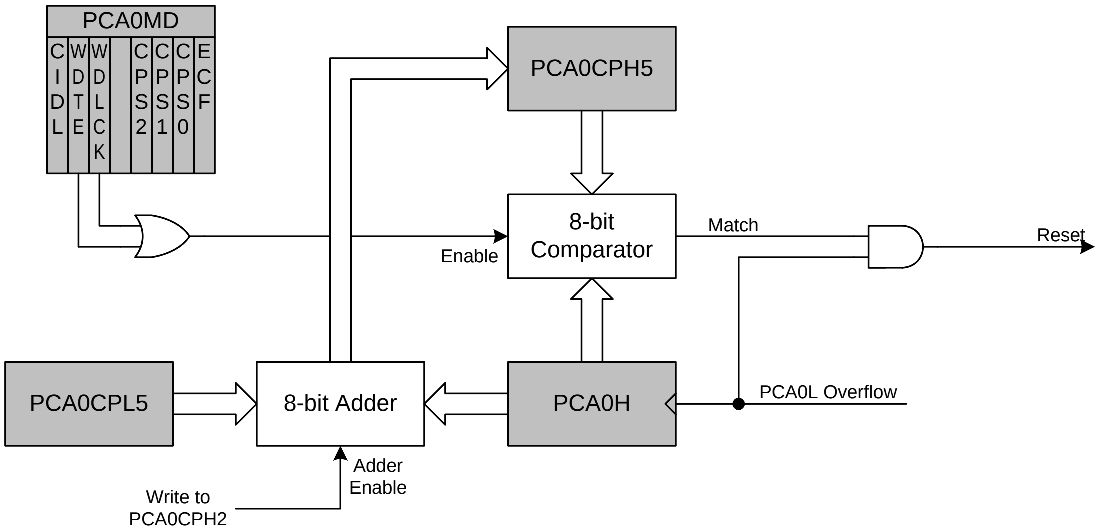


Figure 24.11. PCA Module 2 with Watchdog Timer Enabled

Note that the 8-bit offset held in PCA0CPH5 is compared to the upper byte of the 16-bit PCA counter. This offset value is the number of PCA0L overflows before a reset. Up to 256 PCA clocks may pass before the first PCA0L overflow occurs, depending on the value of the PCA0L when the update is performed. The total offset is then given (in PCA clocks) by Equation 24.5, where PCA0L is the value of the PCA0L register at the time of the update.

$$\text{Offset} = (256 \times \text{PCA0CPL5}) + (256 - \text{PCA0L})$$

Equation 24.5. Watchdog Timer Offset in PCA Clocks

The WDT reset is generated when PCA0L overflows while there is a match between PCA0CPH5 and PCA0H. Software may force a WDT reset by writing a 1 to the CCF5 flag (PCA0CN.5) while the WDT is enabled.

24.4.2. Watchdog Timer Usage

To configure the WDT, perform the following tasks:

- Disable the WDT by writing a 0 to the WDTE bit.
- Select the desired PCA clock source (with the CPS[2:0] bits).
- Load PCA0CPL5 with the desired WDT update offset value.
- Configure the PCA Idle mode (set CIDL if the WDT should be suspended while the CPU is in Idle mode).
- Enable the WDT by setting the WDTE bit to 1.
- Reset the WDT timer by writing to PCA0CPH5.

The PCA clock source and Idle mode select cannot be changed while the WDT is enabled. The watchdog timer is enabled by setting the WDTE or WDLCK bits in the PCA0MD register. When WDLCK is set, the WDT cannot be disabled until the next system reset. If WDLCK is not set, the WDT is disabled by clearing the WDTE bit.

The WDT is enabled following any reset. The PCA0 counter clock defaults to the system clock divided by 12, PCA0L defaults to 0x00, and PCA0CPL5 defaults to 0x00. Using Equation 24.5, this results in a WDT timeout interval of 256 PCA clock cycles, or 3072 system clock cycles. Table 24.3 lists some example timeout intervals for typical system clocks.

C2 Register Definition 25.4. FPCTL: C2 Flash Programming Control

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

C2 Address: 0x02

Bit	Name	Function
7:0	FPCTL[7:0]	**Flash Programming Control Register.** This register is used to enable Flash programming via the C2 interface. To enable C2 Flash programming, the following codes must be written in order: 0x02, 0x01. Note that once C2 Flash programming is enabled, a system reset must be issued to resume normal operation.

C2 Register Definition 25.5. FPDAT: C2 Flash Programming Data

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

C2 Address: 0xB4

Bit	Name	Function	
7:0	FPDAT[7:0]	**C2 Flash Programming Data Register.** This register is used to pass Flash commands, addresses, and data during C2 Flash accesses. Valid commands are listed below.	
		Code	**Command**
		0x06	Flash Block Read
		0x07	Flash Block Write
		0x08	Flash Page Erase
		0x03	Device Erase