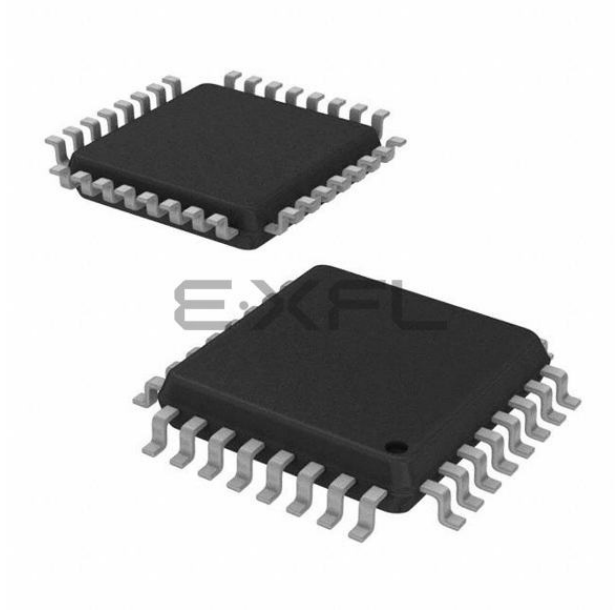

Welcome to E-XFL.COM

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/c8051f545-iqr

Table of Contents

1. System Overview 13
2. Ordering Information 16
3. Pin Definitions 18
4. Package Specifications 23
4.1. QFP-32 Package Specifications 23
4.2. QFN-32 Package Specifications 25
4.3. QFN-24 Package Specifications 27
5. 12-Bit ADC (ADC0) 29
5.1. Modes of Operation 30
5.2. Output Code Formatting 34
5.3. Selectable Gain 35
5.4. Programmable Window Detector 43
6. Electrical Characteristics 47
6.1. Absolute Maximum Specifications 47
6.2. Electrical Characteristics 48
6.1. ADC0 Analog Multiplexer 58
6.2. Temperature Sensor 60
7. Voltage Reference 61
8. Comparators 63
8.1. Comparator Multiplexer 69
9. Voltage Regulator (REG0) 72
10. CIP-51 Microcontroller 74
10.1. Performance 74
10.2. Instruction Set 76
10.3. CIP-51 Register Descriptions 80
10.4. Serial Number Special Function Registers (SFRs) 84
11. Memory Organization 85
11.1. Program Memory 85
11.2. Data Memory 86
11.3. External RAM 87
12. Special Function Registers 89
12.1. SFR Paging 89
12.2. Interrupts and SFR Paging 89
12.3. SFR Page Stack Example 91
13. Interrupts 105
13.1. MCU Interrupt Sources and Vectors 105
13.2. Interrupt Register Descriptions 108
13.3. External Interrupts INT0 and INT1 115
14. Flash Memory 117
14.1. Programming the Flash Memory 117
14.2. Non-volatile Data Storage 119
14.3. Security Options 119
14.4. Flash Write and Erase Guidelines 121

Figure 24.3. PCA Interrupt Block Diagram 252
Figure 24.4. PCA Capture Mode Diagram 254
Figure 24.5. PCA Software Timer Mode Diagram 255
Figure 24.6. PCA High-Speed Output Mode Diagram 256
Figure 24.7. PCA Frequency Output Mode 257
Figure 24.8. PCA 8-Bit PWM Mode Diagram 258
Figure 24.9. PCA 9, 10 and 11-Bit PWM Mode Diagram 259
Figure 24.10. PCA 16-Bit PWM Mode 260
Figure 24.11. PCA Module 2 with Watchdog Timer Enabled 261
Figure 25.1. Typical C2 Pin Sharing 272

SFR Definition 16.1. VDM0CN: VDD Monitor Control 132
SFR Definition 16.2. RSTSRC: Reset Source 134
SFR Definition 17.1. CLKSEL: Clock Select 136
SFR Definition 17.2. OSCICN: Internal Oscillator Control 138
SFR Definition 17.3. OSCICRS: Internal Oscillator Coarse Calibration 139
SFR Definition 17.4. OSCIFIN: Internal Oscillator Fine Calibration 139
SFR Definition 17.5. CLKMUL: Clock Multiplier 141
SFR Definition 17.6. OSCXCN: External Oscillator Control 143
SFR Definition 18.1. XBR0: Port I/O Crossbar Register 0 154
SFR Definition 18.2. XBR1: Port I/O Crossbar Register 1 155
SFR Definition 18.3. XBR2: Port I/O Crossbar Register 1 156
SFR Definition 18.4. P0MASK: Port 0 Mask Register 157
SFR Definition 18.5. P0MAT: Port 0 Match Register 157
SFR Definition 18.6. P1MASK: Port 1 Mask Register 158
SFR Definition 18.7. P1MAT: Port 1 Match Register 158
SFR Definition 18.8. P2MASK: Port 2 Mask Register 159
SFR Definition 18.9. P2MAT: Port 2 Match Register 159
SFR Definition 18.10. P3MASK: Port 3 Mask Register 160
SFR Definition 18.11. P3MAT: Port 3 Match Register 160
SFR Definition 18.12. P0: Port 0 161
SFR Definition 18.13. P0MDIN: Port 0 Input Mode 162
SFR Definition 18.14. P0MDOUT: Port 0 Output Mode 162
SFR Definition 18.15. P0SKIP: Port 0 Skip 163
SFR Definition 18.16. P1: Port 1 163
SFR Definition 18.17. P1MDIN: Port 1 Input Mode 164
SFR Definition 18.18. P1MDOUT: Port 1 Output Mode 164
SFR Definition 18.19. P1SKIP: Port 1 Skip 165
SFR Definition 18.20. P2: Port 2 165
SFR Definition 18.21. P2MDIN: Port 2 Input Mode 166
SFR Definition 18.22. P2MDOUT: Port 2 Output Mode 166
SFR Definition 18.23. P2SKIP: Port 2 Skip 167
SFR Definition 18.24. P3: Port 3 167
SFR Definition 18.25. P3MDIN: Port 3 Input Mode 168
SFR Definition 18.26. P3MDOUT: Port 3 Output Mode 168
SFR Definition 18.27. P3SKIP: Port 3Skip 169
SFR Definition 19.1. LIN0ADR: LIN0 Indirect Address Register 177
SFR Definition 19.2. LIN0DAT: LIN0 Indirect Data Register 177
SFR Definition 19.3. LIN0CF: LIN0 Control Mode Register 178
SFR Definition 20.1. SMB0CF: SMBus Clock/Configuration 193
SFR Definition 20.2. SMB0CN: SMBus Control 195
SFR Definition 20.3. SMB0DAT: SMBus Data 197
SFR Definition 21.1. SCON0: Serial Port 0 Control 210
SFR Definition 21.2. SMOD0: Serial Port 0 Control 211
SFR Definition 21.3. SBUF0: Serial (UART0) Port Data Buffer 212
SFR Definition 21.4. SBCON0: UART0 Baud Rate Generator Control 212

6.2. Electrical Characteristics

Table 6.2. Global Electrical Characteristics

–40 to +125 °C, 24 MHz system clock unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Supply Input Voltage (V_{REGIN})		1.8	—	5.25	V
Digital Supply Voltage (V_{DD})	System Clock ≤ 25 MHz	V_{RST}[1]	—	2.75	V
	System Clock > 25 MHz	2	—	2.75	
Analog Supply Voltage (VDDA)	System Clock ≤ 25 MHz	V_{RST}[1]	—	2.75	V
(Must be connected to V_{DD})	System Clock > 25 MHz	2		2.75	
Port I/O Supply Voltage (V_{IO})	Normal Operation	1.8[2]	—	5.25	V
Digital Supply RAM Data Retention Voltage		—	1.5	—	V
SYSCLK (System Clock)[3]		0	—	50	MHz
T_{SYSH} (SYSCLK High Time)		9	—	—	ns
T_{SYSL} (SYSCLK Low Time)		9	—	—	ns
Specified Operating Temperature Range		–40	—	+125	°C
Digital Supply Current—CPU Active (Normal Mode, fetching instructions from Flash)					
I_{DD}[4]	V_{DD} = 2.1 V, F = 200 kHz	—	85	—	µA
	V_{DD} = 2.1 V, F = 1.5 MHz	—	600	—	µA
	V_{DD} = 2.1 V, F = 25 MHz	—	9.2	11	mA
	V_{DD} = 2.1 V, F = 50 MHz	—	17	21	mA
I_{DD}[4]	V_{DD} = 2.6 V, F = 200 kHz	—	120	—	µA
	V_{DD} = 2.6 V, F = 1.5 MHz	—	920	—	µA
	V_{DD} = 2.6 V, F = 25 MHz	—	13	21	mA
	V_{DD} = 2.6 V, F = 50 MHz	—	22	33	mA
I_{DD} Supply Sensitivity[4]	F = 25 MHz	—	68	—	%/V
	F = 1 MHz	—	77	—	%/V
I_{DD} Frequency Sensitivity [4,5]	V_{DD} = 2.1 V, F ≤ 12.5 MHz, T = 25 °C	—	0.43	—	mA/MHz
	V_{DD} = 2.1 V, F > 12.5 MHz, T = 25 °C	—	0.33	—	mA/MHz
	V_{DD} = 2.6 V, F ≤ 12.5 MHz, T = 25 °C	—	0.60	—	mA/MHz
	V_{DD} = 2.6 V, F > 12.5 MHz, T = 25 °C	—	0.42	—	mA/MHz

Notes:

1. Given in Table 6.4 on page 52.
2. V_{IO} should not be lower than the V_{DD} voltage.
3. SYSCLK must be at least 32 kHz to enable debugging.
4. Guaranteed by characterization. Does not include oscillator supply current.
5. IDD estimation for different frequencies.
6. Idle IDD estimation for different frequencies.

Table 6.2. 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	—	50	—	µA
	V_{DD} = 2.1 V, F = 1.5 MHz	—	410	—	µA
	V_{DD} = 2.1 V, F = 25 MHz	—	6.5	8.0	mA
	V_{DD} = 2.1 V, F = 50 MHz	—	13	16	mA
I_{DD}[4]	V_{DD} = 2.6 V, F = 200 kHz	—	67	—	µA
	V_{DD} = 2.6 V, F = 1.5 MHz	—	530	—	µA
	V_{DD} = 2.6 V, F = 25 MHz	—	8.0	15	mA
	V_{DD} = 2.6 V, F = 50 MHz	—	16	25	mA
I_{DD} Supply Sensitivity[4]	F = 25 MHz	—	55	—	%/V
	F = 1 MHz	—	58	—	
I_{DD} Frequency Sensitivity [4.6]	V_{DD} = 2.1V, F ≤ 12.5 MHz, T = 25 °C	—	0.26	—	mA/MHz
	V_{DD} = 2.1V, F > 12.5 MHz, T = 25 °C	—	0.26	—	
	V_{DD} = 2.6V, F ≤ 12.5 MHz, T = 25 °C	—	0.34	—	
	V_{DD} = 2.6V, F > 12.5 MHz, T = 25 °C	—	0.34	—	
Digital Supply Current[4] (Stop or Suspend Mode)	Oscillator not running, V_{DD} Monitor Disabled				µA
	Temp = 25 °C	—	1	—	
	Temp = 60 °C	—	6	—	
	Temp= 125 °C	—	70	—	

Notes:
1. Given in Table 6.4 on page 52.
2. V_{IO} should not be lower than the V_{DD} voltage.
3. SYSCLK must be at least 32 kHz to enable debugging.
4. Guaranteed by characterization. Does not include oscillator supply current.
5. IDD estimation for different frequencies.
6. Idle IDD estimation for different frequencies.

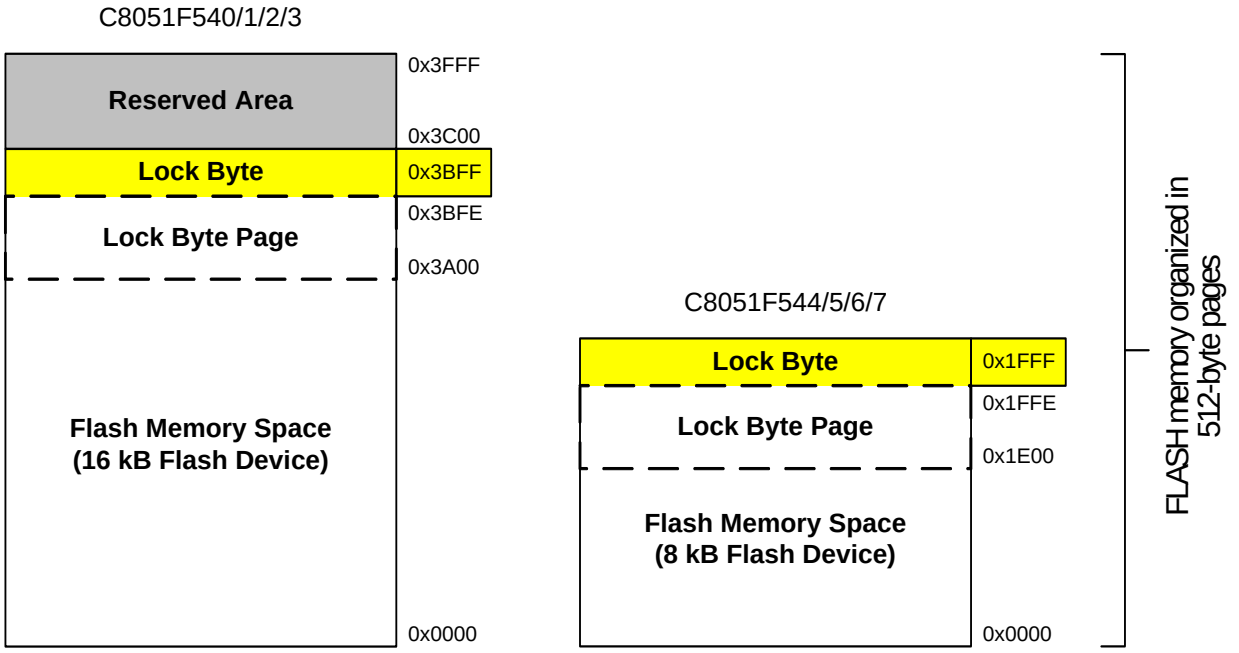

Figure 11.2. Flash Program Memory Map

11.1.1. MOVX Instruction and Program Memory

The MOVX instruction in an 8051 device is typically used to access external data memory. On the C8051F54x 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 C8051F54x to update program code and use the program memory space for non-volatile data storage. Refer to Section “14. Flash Memory” on page 117 for further details.

11.2. Data Memory

The C8051F54x devices include 1280 bytes of RAM data memory. 256 bytes of this memory is mapped into the internal RAM space of the 8051. The other 1024 bytes of this memory is on-chip “external” memory. The data memory map is shown in Figure 11.1 for reference.

11.2.1. Internal RAM

There are 256 bytes of internal RAM mapped into the data memory space from 0x00 through 0xFF. The lower 128 bytes of data memory are used for general purpose registers and scratch pad memory. Either direct or indirect addressing may be used to access the lower 128 bytes of data memory. Locations 0x00 through 0x1F are addressable as four banks of general purpose registers, each bank consisting of eight byte-wide registers. The next 16 bytes, locations 0x20 through 0x2F, may either be addressed as bytes or as 128 bit locations accessible with the direct addressing mode.

The upper 128 bytes of data memory are accessible only by indirect addressing. This region occupies the same address space as the Special Function Registers (SFR) but is physically separate from the SFR space. The addressing mode used by an instruction when accessing locations above 0x7F determines whether the CPU accesses the upper 128 bytes of data memory space or the SFRs. Instructions that use direct addressing will access the SFR space. Instructions using indirect addressing above 0x7F access the upper 128 bytes of data memory. Figure 11.1 illustrates the data memory organization of the C8051F54x.

Table 12.1. Special Function Register (SFR) Memory Map for Pages 0x0 and 0xF

Address	Page	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
F8	0F	SPI0CN	PCA0L SN0	PCA0H SN1	PCA0CPL0 SN2	PCA0CPH0 SN3	PCACPL4	PCACPH4	VDM0CN
F0	0F	B (All Pages)	P0MAT P0MDIN	P0MASK P1MDIN	P1MAT P2MDIN	P1MASK P3MDIN		EIP1 EIP1	EIP2 EIP2
E8	0F	ADC0CN	PCA0CPL1	PCA0CPH1	PCA0CPL2	PCA0CPH2	PCA0CPL3	PCA0CPL3	RSTSRC
E0	0F	ACC (All Pages)	XBR0	XBR1	CCH0CN	IT01CF		EIE1 (All Pages)	EIE2 (All Pages)
D8	0F	PCA0CN	PCA0MD PCA0PWM	PCA0CPM0	PCA0CPM1	PCA0CPM2	PCA0CPM3	PCA0CPM4	PCA0CPM5
D0	0F	PSW (All Pages)	REF0CN	LIN0DATA	LIN0ADDR	P0SKIP	P1SKIP	P2SKIP	P3SKIP
C8	0F	TMR2CN	REG0CN LIN0CF	TMR2RLL	TMR2RLH	TMR2L	TMR2H	PCA0CPL5	PCA0CPH5
C0	0F	SMB0CN	SMB0CF	SMB0DAT	ADC0GTL	ADC0GTH	ADC0LTL	ADC0LTH	XBR2
B8	0F	IP (All Pages)		ADC0TK	ADC0MX	ADC0CF	ADC0L	ADC0H	
B0	0F	P3 (All Pages)	P2MAT	P2MASK				FLSCL (All Pages)	FLKEY (All Pages)
A8	0F	IE (All Pages)	SMOD0	EMI0CN	SBCON0	SBRLL0	SBRLH0	P3MAT P3MDOUT	P3MASK
A0	0F	P2 (All Pages)	SPI0CFG OSCICN	SPI0CKR OSCICRS	SPI0DAT	P0MDOUT	P1MDOUT	P2MDOUT	SFRPAGE (All Pages)
98	0F	SCON0	SBUF0	CPT0CN	CPT0MD	CPT0MX	CPT1CN	CPT1MD OSCIFIN	CPT1MX OSCXCN
90	0F	P1 (All Pages)	TMR3CN	TMR3RLL	TMR3RLH	TMR3L	TMR3H		CLKMUL
88	0F	TCON (All Pages)	TMOD (All Pages)	TL0 (All Pages)	TL1 (All Pages)	TH0 (All Pages)	TH1 (All Pages)	CKCON (All Pages)	PSCTL CLKSEL
80	0F	P0 (All Pages)	SP (All Pages)	DPL (All Pages)	DPH (All Pages)	SFR0CN	SFRNEXT (All Pages)	SFRLAST (All Pages)	PCON (All Pages)
		0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

(bit addressable)

13.2. Interrupt Register Descriptions

The SFRs used to enable the interrupt sources and set their priority level are described in this section. Refer to the data sheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

14.1.3. Flash Write Procedure

Flash bytes are programmed by software with the following sequence:

1. Disable interrupts (recommended).
2. Erase the 512-byte Flash page containing the target location, as described in Section 14.1.2.
3. Set the PSWE bit (register PSCTL).
4. Clear the PSEE bit (register PSCTL).
5. Write the first key code to FLKEY: 0xA5.
6. Write the second key code to FLKEY: 0xF1.
7. Using the MOVX instruction, write a single data byte to the desired location within the 512-byte sector.
8. Clear the PSWE bit.

Steps 5–7 must be repeated for each byte to be written. After Flash writes are complete, PSWE should be cleared so that MOVX instructions do not target program memory.

14.1.4. Flash Write Optimization

The Flash write procedure includes a block write option to optimize the time to perform consecutive byte writes. When block write is enabled by setting the CHBLKW bit (CCH0CN.0), writes to two consecutive bytes in Flash require the same amount of time as a single byte write. This is performed by caching the first byte that is written to Flash and then committing both bytes to Flash when the second byte is written. When block writes are enabled, if the second write does not occur, the first data byte written is not actually written to Flash. Flash bytes with block write enabled are programmed by software with the following sequence:

1. Disable interrupts (recommended).
2. Erase the 512-byte Flash page containing the target location, as described in Section 14.1.2.
3. Set the CHBLKW bit (register CCH0CN).
4. Set the PSWE bit (register PSCTL).
5. Clear the PSEE 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 the first data byte to the desired location within the 512-byte sector.
9. Write the first key code to FLKEY: 0xA5.
10. Write the second key code to FLKEY: 0xF1.
11. Using the MOVX instruction, write the second data byte to the desired location within the 512-byte sector. The location of the second byte must be the next higher address from the first data byte.
12. Clear the PSWE bit.
13. Clear the CHBLKW bit.

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.

15.2. 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.

15.3. 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 “18.5. Port Match” on page 157), 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 7.1).

17.4. External Oscillator Drive Circuit

The external oscillator circuit may drive an external crystal, ceramic resonator, capacitor, or RC network. A CMOS clock may also provide a clock input. For a crystal or ceramic resonator configuration, the crystal/resonator must be wired across the XTAL1 and XTAL2 pins as shown in Option 1 of Figure 17.1. A 10 MΩ resistor also must be wired across the XTAL2 and XTAL1 pins for the crystal/resonator configuration. In RC, capacitor, or CMOS clock configuration, the clock source should be wired to the XTAL2 pin as shown in Option 2, 3, or 4 of Figure 17.1. The type of external oscillator must be selected in the OSCXCN register, and the frequency control bits (XFCN) must be selected appropriately (see SFR Definition 17.6).

Important Note on External Oscillator Usage: Port pins must be configured when using the external oscillator circuit. When the external oscillator drive circuit is enabled in crystal/resonator mode, Port pins P0.2 and P0.3 are used as XTAL1 and XTAL2 respectively. When the external oscillator drive circuit is enabled in capacitor, RC, or CMOS clock mode, Port pin P0.3 is used as XTAL2. The Port I/O Crossbar should be configured to skip the Port pins used by the oscillator circuit; see Section “18.3. Priority Crossbar Decoder” on page 150 for Crossbar configuration. Additionally, when using the external oscillator circuit in crystal/resonator, capacitor, or RC mode, the associated Port pins should be configured as **analog inputs**. In CMOS clock mode, the associated pin should be configured as a **digital input**. See Section “18.4. Port I/O Initialization” on page 152 for details on Port input mode selection.

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 21.1. SCON0: Serial Port 0 Control

Bit	7	6	5	4	3	2	1	0
Name	OVR0	PERR0	THRE0	REN0	TBX0	RBX0	TI0	RI0
Type	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	0	0	0

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

Bit	Name	Function
7	OVR0	**Receive FIFO Overrun Flag.** 0: Receive FIFO Overrun has not occurred 1: Receive FIFO Overrun has occurred; A received character has been discarded due to a full FIFO.
6	PERR0	**Parity Error Flag.** When parity is enabled, this bit indicates that a parity error has occurred. It is set to 1 when the parity of the oldest byte in the FIFO does not match the selected Parity Type. 0: Parity error has not occurred 1: Parity error has occurred. This bit must be cleared by software.
5	THRE0	**Transmit Holding Register Empty Flag.** 0: Transmit Holding Register not Empty—do not write to SBUF0. 1: Transmit Holding Register Empty—it is safe to write to SBUF0.
4	REN0	**Receive Enable.** This bit enables/disables the UART receiver. When disabled, bytes can still be read from the receive FIFO. 0: UART1 reception disabled. 1: UART1 reception enabled.
3	TBX0	**Extra Transmission Bit.** The logic level of this bit will be assigned to the extra transmission bit when XBE0 is set to 1. This bit is not used when Parity is enabled.
2	RBX0	**Extra Receive Bit.** RBX0 is assigned the value of the extra bit when XBE1 is set to 1. If XBE1 is cleared to 0, RBX1 will be assigned the logic level of the first stop bit. This bit is not valid when Parity is enabled.
1	TI0	**Transmit Interrupt Flag.** Set to a 1 by hardware after data has been transmitted, at the beginning of the STOP bit. When the UART0 interrupt is enabled, setting this bit causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.
0	RI0	**Receive Interrupt Flag.** Set to 1 by hardware when a byte of data has been received by UART0 (set at the STOP bit sampling time). When the UART0 interrupt is enabled, setting this bit to 1 causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by soft-ware. Note that RI0 will remain set to ‘1’ as long as there is data still in the UART FIFO. After the last byte has been shifted from the FIFO to SBUF0, RI0 can be cleared.

22.3. SPI0 Slave Mode Operation

When SPI0 is enabled and not configured as a master, it will operate as a SPI slave. As a slave, bytes are shifted in through the MOSI pin and out through the MISO pin by a master device controlling the SCK signal. A bit counter in the SPI0 logic counts SCK edges. When 8 bits have been shifted through the shift register, the SPIF flag is set to logic 1, and the byte is copied into the receive buffer. Data is read from the receive buffer by reading SPI0DAT. A slave device cannot initiate transfers. Data to be transferred to the master device is pre-loaded into the shift register by writing to SPI0DAT. Writes to SPI0DAT are double-buffered, and are placed in the transmit buffer first. If the shift register is empty, the contents of the transmit buffer will immediately be transferred into the shift register. When the shift register already contains data, the SPI will load the shift register with the transmit buffer's contents after the last SCK edge of the next (or current) SPI transfer.

When configured as a slave, SPI0 can be configured for 4-wire or 3-wire operation. The default, 4-wire slave mode, is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In 4-wire mode, the NSS signal is routed to a port pin and configured as a digital input. SPI0 is enabled when NSS is logic 0, and disabled when NSS is logic 1. The bit counter is reset on a falling edge of NSS. Note that the NSS signal must be driven low at least 2 system clocks before the first active edge of SCK for each byte transfer. Figure 22.4 shows a connection diagram between two slave devices in 4-wire slave mode and a master device.

3-wire slave mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. NSS is not used in this mode, and is not mapped to an external port pin through the crossbar. Since there is no way of uniquely addressing the device in 3-wire slave mode, SPI0 must be the only slave device present on the bus. It is important to note that in 3-wire slave mode there is no external means of resetting the bit counter that determines when a full byte has been received. The bit counter can only be reset by disabling and re-enabling SPI0 with the SPIEN bit. Figure 22.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

22.4. SPI0 Interrupt Sources

When SPI0 interrupts are enabled, the following four flags will generate an interrupt when they are set to logic 1:

All of the following bits must be cleared by software.

1. The SPI Interrupt Flag, SPIF (SPI0CN.7) is set to logic 1 at the end of each byte transfer. This flag can occur in all SPI0 modes.
2. The Write Collision Flag, WCOL (SPI0CN.6) is set to logic 1 if a write to SPI0DAT is attempted when the transmit buffer has not been emptied to the SPI shift register. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written.This flag can occur in all SPI0 modes.
3. The Mode Fault Flag MODF (SPI0CN.5) is set to logic 1 when SPI0 is configured as a master, and for multi-master mode and the NSS pin is pulled low. When a Mode Fault occurs, the MSTEN and SPIEN bits in SPI0CN are set to logic 0 to disable SPI0 and allow another master device to access the bus.
4. The Receive Overrun Flag RXOVRN (SPI0CN.4) is set to logic 1 when configured as a slave, and a transfer is completed and the receive buffer still holds an unread byte from a previous transfer. The new byte is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte which caused the overrun is lost.

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).

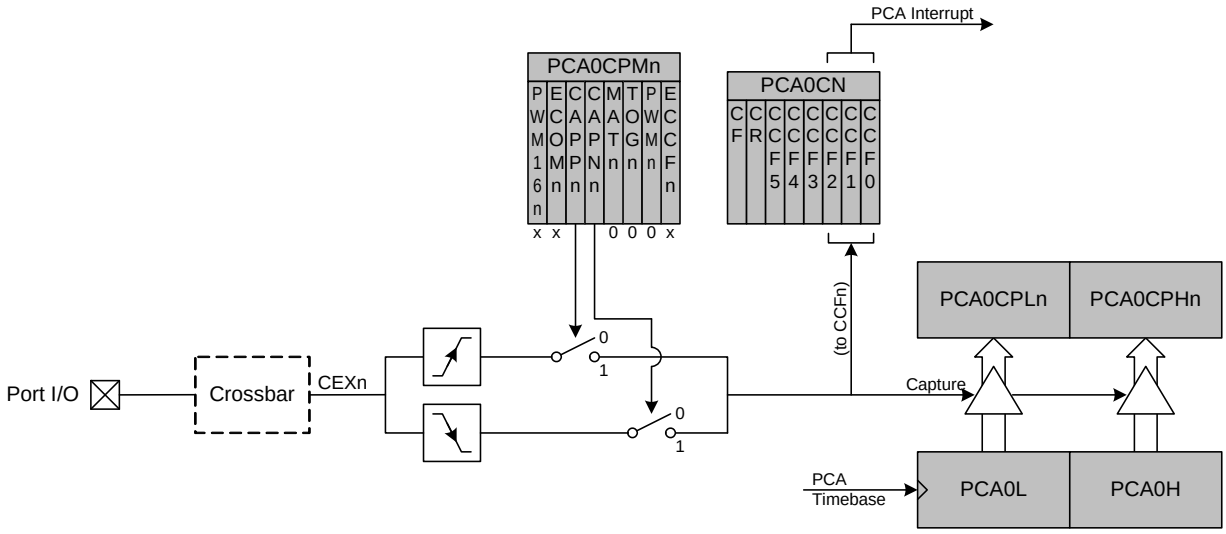

Figure 24.4. PCA Capture Mode Diagram

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

24.3.2. Software Timer (Compare) Mode

In Software Timer mode, the PCA counter/timer value is compared to 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 ECOMn and MATn bits in the PCA0CPMn register enables Software Timer mode.

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.

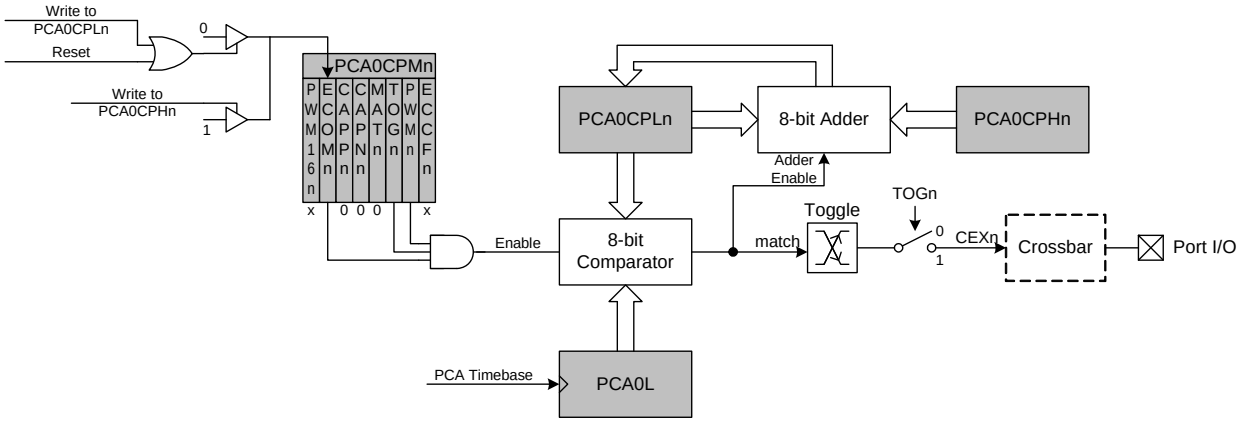

Figure 24.7. PCA Frequency Output Mode

24.3.5. 8-bit, 9-bit, 10-bit and 11-bit Pulse Width Modulator Modes

Each module can be used independently to generate a pulse width modulated (PWM) output on its associated CEXn pin. The frequency of the output is dependent on the timebase for the PCA counter/timer, and the setting of the PWM cycle length (8, 9, 10 or 11-bits). For backwards-compatibility with the 8-bit PWM mode available on other devices, the 8-bit PWM mode operates slightly different than 9, 10 and 11-bit PWM modes. **It is important to note that all channels configured for 8/9/10/11-bit PWM mode will use the same cycle length.** It is not possible to configure one channel for 8-bit PWM mode and another for 11-bit mode (for example). However, other PCA channels can be configured to Pin Capture, High-Speed Output, Software Timer, Frequency Output, or 16-bit PWM mode independently.

24.3.5.1. 8-bit Pulse Width Modulator Mode

The duty cycle of the PWM output signal in 8-bit PWM mode is varied using the module's PCA0CPLn capture/compare register. When the value in the low byte of the PCA counter/timer (PCA0L) is equal to the value in PCA0CPLn, the output on the CEXn pin will be set. When the count value in PCA0L overflows, the CEXn output will be reset (see Figure 24.8). Also, when the counter/timer low byte (PCA0L) overflows from 0xFF to 0x00, PCA0CPLn is reloaded automatically with the value stored in the module's capture/compare high byte (PCA0CPHn) without software intervention. Setting the ECOMn and PWMn bits in the PCA0CPMn register, and setting the CLSEL bits in register PCA0PWM to 00b enables 8-Bit Pulse Width Modulator mode. If the MATn bit is set to 1, the CCFn flag for the module will be set each time an 8-bit comparator match (rising edge) occurs. The COVF flag in PCA0PWM can be used to detect the overflow (falling edge), which will occur every 256 PCA clock cycles. The duty cycle for 8-Bit PWM Mode is given in Equation 24.2.

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.

$$\text{Duty Cycle} = \frac{(256 - \text{PCA0CPHn})}{256}$$

Equation 24.2. 8-Bit PWM Duty Cycle

Using Equation 24.2, the largest duty cycle is 100% (PCA0CPHn = 0), and the smallest duty cycle is 0.39% (PCA0CPHn = 0xFF). A 0% duty cycle may be generated by clearing the ECOMn bit to 0.

SFR Definition 24.4. PCA0CPMn: PCA Capture/Compare Mode

Bit	7	6	5	4	3	2	1	0
Name	PWM16n	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn
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: PCA0CPM0 = 0xDA, PCA0CPM1 = 0xDB, PCA0CPM2 = 0xDC; PCA0CPM3 = 0xDD, PCA0CPM4 = 0xDE, PCA0CPM5 = 0xDF, SFR Page (all registers) = 0x00

Bit	Name	Function
7	PWM16n	**16-bit Pulse Width Modulation Enable.** This bit enables 16-bit mode when Pulse Width Modulation mode is enabled. 0: 8 to 11-bit PWM selected. 1: 16-bit PWM selected.
6	ECOMn	**Comparator Function Enable.** This bit enables the comparator function for PCA module n when set to 1.
5	CAPPn	**Capture Positive Function Enable.** This bit enables the positive edge capture for PCA module n when set to 1.
4	CAPNn	**Capture Negative Function Enable.** This bit enables the negative edge capture for PCA module n when set to 1.
3	MATn	**Match Function Enable.** This bit enables the match function for PCA module n when set to 1. When enabled, matches of the PCA counter with a module's capture/compare register cause the CCFn bit in PCA0MD register to be set to logic 1.
2	TOGn	**Toggle Function Enable.** This bit enables the toggle function for PCA module n when set to 1. When enabled, matches of the PCA counter with a module's capture/compare register cause the logic level on the CEXn pin to toggle. If the PWMn bit is also set to logic 1, the module operates in Frequency Output Mode.
1	PWMn	**Pulse Width Modulation Mode Enable.** This bit enables the PWM function for PCA module n when set to 1. When enabled, a pulse width modulated signal is output on the CEXn pin. 8 to 11-bit PWM is used if PWM16n is cleared; 16-bit mode is used if PWM16n is set to logic 1. If the TOGn bit is also set, the module operates in Frequency Output Mode.
0	ECCFn	**Capture/Compare Flag Interrupt Enable.** This bit sets the masking of the Capture/Compare Flag (CCFn) interrupt. 0: Disable CCFn interrupts. 1: Enable a Capture/Compare Flag interrupt request when CCFn is set.
Note: When the WDTE bit is set to 1, the PCA0CPM5 register cannot be modified, and module 5 acts as the watchdog timer. To change the contents of the PCA0CPM5 register or the function of module 5, the Watchdog Timer must be disabled.		

25. C2 Interface

C8051F54x devices include an on-chip Silicon Labs 2-Wire (C2) debug interface to allow Flash programming and in-system debugging with the production part installed in the end application. The C2 interface uses a clock signal (C2CK) and a bi-directional C2 data signal (C2D) to transfer information between the device and a host system. See the C2 Interface Specification for details on the C2 protocol.

25.1. C2 Interface Registers

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

C2 Register Definition 25.1. C2ADD: C2 Address

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

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