

Welcome to [E-XFL.COM](https://www.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/I ² C), LINbus, SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	25
Program Memory Size	16KB (16K 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-VFQFN Exposed Pad
Supplier Device Package	32-QFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f540-imr

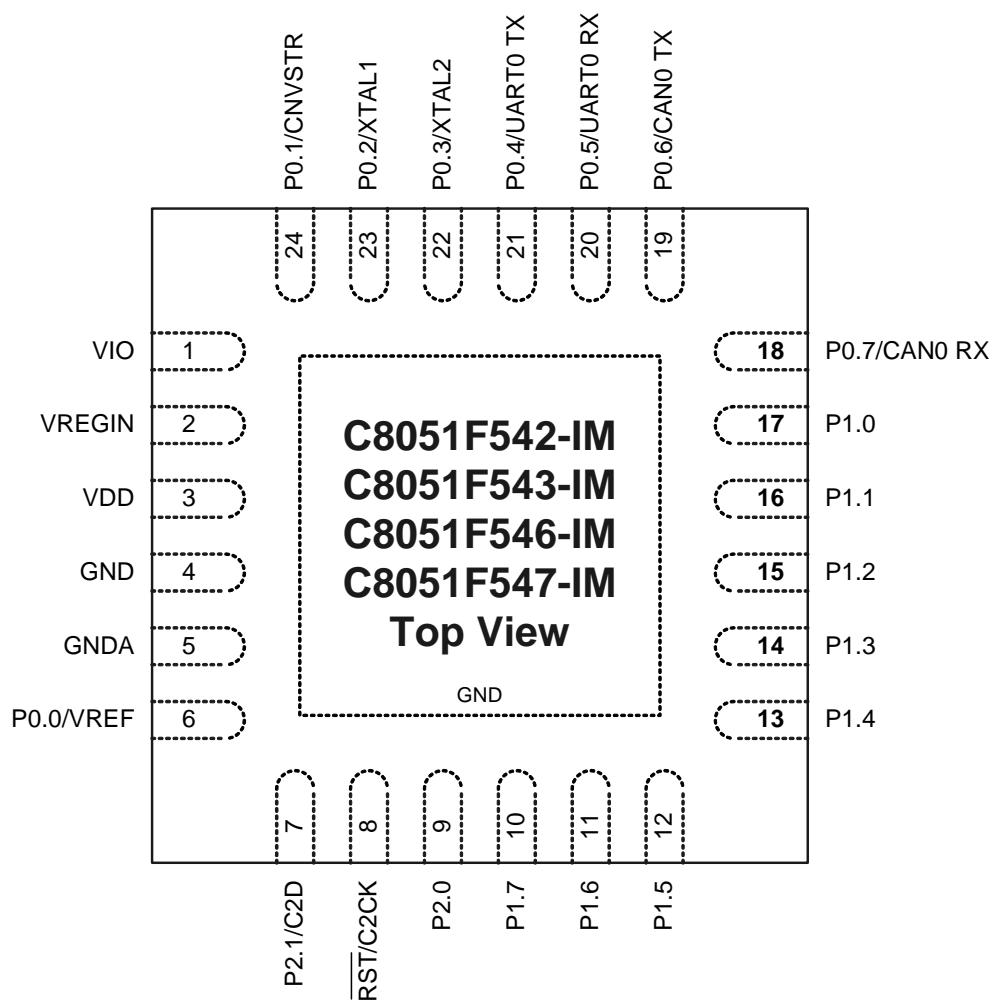


Figure 3.3. QFN-24 Pinout Diagram (Top View)

Table 6.12. Comparator 0 and Comparator 1 Electrical CharacteristicsV_{IO} = 1.8 to 5.25 V, –40 to +125 °C unless otherwise noted.

Parameter	Conditions	Min	Typ	Max	Units
Response Time: Mode 0, V _{cm} * = 1.5 V	CPn+ – CPn– = 100 mV	—	330	—	ns
	CPn+ – CPn– = –100 mV	—	390	—	ns
Response Time: Mode 1, V _{cm} * = 1.5 V	CPn+ – CPn– = 100 mV	—	490	—	ns
	CPn+ – CPn– = –100 mV	—	610	—	ns
Response Time: Mode 2, V _{cm} * = 1.5 V	CPn+ – CPn– = 100 mV	—	590	—	ns
	CP0+ – CP0– = –100 mV	—	750	—	ns
Response Time: Mode 3, V _{cm} * = 1.5 V	CPn+ – CPn– = 100 mV	—	2300	—	ns
	CPn+ – CPn– = –100 mV	—	3100	—	ns
Common-Mode Rejection Ratio		—	2.1	13	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	21	40	mV
Negative Hysteresis 1	CPnHYN1–0 = 00	–2	0	2	mV
Negative Hysteresis 2	CPnHYN1–0 = 01	2	5	10	mV
Negative Hysteresis 3	CPnHYN1–0 = 10	5	11	20	mV
Negative Hysteresis 4	CPnHYN1–0 = 11	13	21	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		—	0.18	—	mV/V
Power-Up Time		—	3	—	μs
Supply Current at DC	Mode 0	—	6.3	20	μA
	Mode 1	—	3.4	10	μA
	Mode 2	—	2.6	7.5	μA
	Mode 3	—	0.6	3	μA
*Note: V _{cm} is the common-mode voltage on CP0+ and CP0–.					

C8051F54x

SFR Definition 8.5. CPT0MX: Comparator0 MUX Selection

Bit	7	6	5	4	3	2	1	0
Name	CMX0N[3:0]				CMX0P[3:0]			
Type	R/W				R/W			
Reset	0	1	1	1	0	1	1	1

SFR Address = 0x9C; SFR Page = 0x00

Bit	Name	Function
7:4	CMX0N[3:0]	Comparator0 Negative Input MUX Selection. 0000: P0.1 0001: P0.3 0010: P0.5 0011: P0.7 0100: P1.1 0101: P1.3 0110: P1.5 0111: P1.7 1000: P2.1 1001: P2.3 (only available on 32-pin devices) 1010: P2.5 (only available on 32-pin devices) 1011: P2.7 (only available on 32-pin devices) 1100–1111: None
3:0	CMX0P[3:0]	Comparator0 Positive Input MUX Selection. 0000: P0.0 0001: P0.2 0010: P0.4 0011: P0.6 0100: P1.0 0101: P1.2 0110: P1.4 0111: P1.6 1000: P2.0 1001: P2.2 (only available on 32-pin devices) 1010: P2.4 (only available on 32-pin devices) 1011: P2.6 (only available on 32-pin devices) 1100–1111: None

SFR Definition 8.6. CPT1MX: Comparator1 MUX Selection

Bit	7	6	5	4	3	2	1	0
Name	CMX1N[3:0]				CMX1P[3:0]			
Type	R/W				R/W			
Reset	0	1	1	1	0	1	1	1

SFR Address = 0x9F; SFR Page = 0x00

Bit	Name	Function
7:4	CMX1N[3:0]	Comparator1 Negative Input MUX Selection. 0000: P0.1 0001: P0.3 0010: P0.5 0011: P0.7 0100: P1.1 0101: P1.3 0110: P1.5 0111: P1.7 1000: P2.1 1001: P2.3 (only available on 32-pin devices) 1010: P2.5 (only available on 32-pin devices) 1011: P2.7 (only available on 32-pin devices) 1100–1111: None
3:0	CMX1P[3:0]	Comparator1 Positive Input MUX Selection. 0000: P0.0 0001: P0.2 0010: P0.4 0011: P0.6 0100: P1.0 0101: P1.2 0110: P1.4 0111: P1.6 1000: P2.0 1001: P2.2 (only available on 32-pin devices) 1010: P2.4 (only available on 32-pin devices) 1011: P2.6 (only available on 32-pin devices) 1100–1111: None

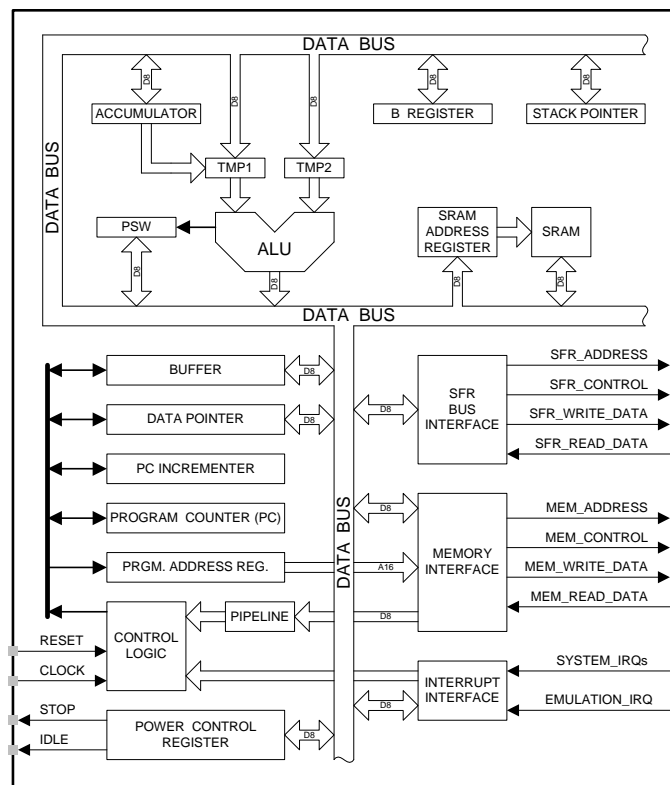


Figure 10.1. CIP-51 Block Diagram

With the CIP-51's maximum system clock at 50 MHz, it has a peak throughput of 50 MIPS. The CIP-51 has a total of 109 instructions. The table below shows the total number of instructions that require each execution time.

Clocks to Execute	1	2	2/3	3	3/4	4	4/5	5	8
Number of Instructions	26	50	5	14	7	3	1	2	1

Programming and Debugging Support

In-system programming of the Flash program memory and communication with on-chip debug support logic is accomplished via the Silicon Labs 2-Wire Development Interface (C2).

The on-chip debug support logic facilitates full speed in-circuit debugging, allowing the setting of hardware breakpoints, starting, stopping and single stepping through program execution (including interrupt service routines), examination of the program's call stack, and reading/writing the contents of registers and memory. This method of on-chip debugging is completely non-intrusive, requiring no RAM, Stack, timers, or other on-chip resources. C2 details can be found in Section "25. C2 Interface" on page 269.

The CIP-51 is supported by development tools from Silicon Labs and third party vendors. Silicon Labs provides an integrated development environment (IDE) including editor, debugger and programmer. The IDE's debugger and programmer interface to the CIP-51 via the C2 interface to provide fast and efficient in-system device programming and debugging. Third party macro assemblers and C compilers are also available.

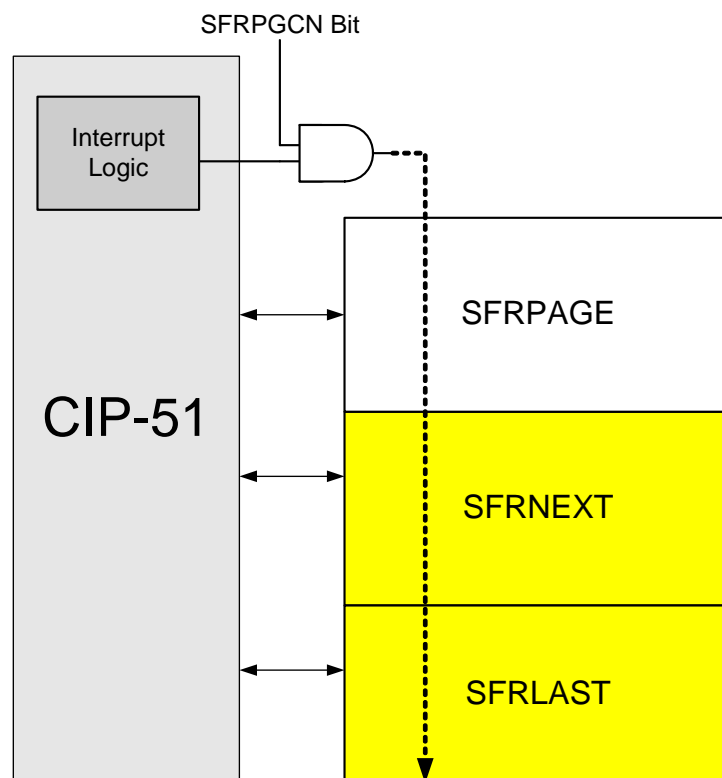


Figure 12.1. SFR Page Stack

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.

C8051F54x

Table 12.2. Special Function Registers (Continued)

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	Description	Page
SN0	0xF9	Serial Number 0	84
SN1	0xFA	Serial Number 1	84
SN2	0xFB	Serial Number 2	84
SN3	0xFC	Serial Number 3	84
SP	0x81	Stack Pointer	82
SPI0CFG	0xA1	SPI0 Configuration	221
SPI0CKR	0xA2	SPI0 Clock Rate Control	223
SPI0CN	0xF8	SPI0 Control	222
SPI0DAT	0xA3	SPI0 Data	223
TCON	0x88	Timer/Counter Control	233
TH0	0x8C	Timer/Counter 0 High	236
TH1	0x8D	Timer/Counter 1 High	236
TL0	0x8A	Timer/Counter 0 Low	235
TL1	0x8B	Timer/Counter 1 Low	235
TMOD	0x89	Timer/Counter Mode	234
TMR2CN	0xC8	Timer/Counter 2 Control	240
TMR2H	0xCD	Timer/Counter 2 High	242
TMR2L	0xCC	Timer/Counter 2 Low	242
TMR2RLH	0xCB	Timer/Counter 2 Reload High	241
TMR2RLL	0xCA	Timer/Counter 2 Reload Low	241
TMR3CN	0x91	Timer/Counter 3 Control	246
TMR3H	0x95	Timer/Counter 3 High	248
TMR3L	0x94	Timer/Counter 3 Low	248
TMR3RLH	0x93	Timer/Counter 3 Reload High	247
TMR3RLL	0x92	Timer/Counter 3 Reload Low	247
VDM0CN	0xFF	V _{DD} Monitor Control	132
XBR0	0xE1	Port I/O Crossbar Control 0	154
XBR1	0xE2	Port I/O Crossbar Control 1	155
XBR2	0xC7	Port I/O Crossbar Control 2	156

13.1.1. Interrupt Priorities

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. Each interrupt has an associated interrupt priority bit in an SFR (IE, EIP1, or EIP2) used to configure its priority level. Low priority is the default. If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate, given in Table 13.1.

13.1.2. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 5 system clock cycles: 1 clock cycle to detect the interrupt and 4 clock cycles to complete the LCALL to the ISR. If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs when the CPU is performing an RETI instruction followed by a DIV as the next instruction. In this case, the response time is 18 system clock cycles: 1 clock cycle to detect the interrupt, 5 clock cycles to execute the RETI, 8 clock cycles to complete the DIV instruction and 4 clock cycles to execute the LCALL to the ISR. If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the RETI and following instruction.

17.3. Clock Multiplier

The Clock Multiplier generates an output clock which is 4 times the input clock frequency scaled by a programmable factor of 1, 2/3, 2/4 (or 1/2), 2/5, 2/6 (or 1/3), or 2/7. The Clock Multiplier's input can be selected from the external oscillator, or the internal or external oscillators divided by 2. This produces three possible base outputs which can be scaled by a programmable factor: Internal Oscillator x 2, External Oscillator x 2, or External Oscillator x 4. See Section 17.1 on page 135 for details on system clock selection.

The Clock Multiplier is configured via the CLKMUL register (SFR Definition 17.5). The procedure for configuring and enabling the Clock Multiplier is as follows:

1. Reset the Multiplier by writing 0x00 to register CLKMUL.
2. Select the Multiplier input source via the MULSEL bits.
3. Select the Multiplier output scaling factor via the MULDIV bits
4. Enable the Multiplier with the MULEN bit (CLKMUL | = 0x80).
5. Delay for >5 μ s.
6. Initialize the Multiplier with the MULINIT bit (CLKMUL | = 0xC0).
7. Poll for MULRDY ≥ 1 .

Important Note: When using an external oscillator as the input to the Clock Multiplier, the external source must be enabled and stable before the Multiplier is initialized. See “17.4. External Oscillator Drive Circuit” on page 142 for details on selecting an external oscillator source.

The Clock Multiplier allows faster operation of the CIP-51 core and is intended to generate an output frequency between 25 and 50 MHz. The clock multiplier can also be used with slow input clocks. However, if the clock is below the minimum Clock Multiplier input frequency (F_{CMmin}), the generated clock will consist of four fast pulses followed by a long delay until the next input clock rising edge. The average frequency of the output is equal to 4x the input, but the instantaneous frequency may be faster. See Figure 17.2 below for more information.

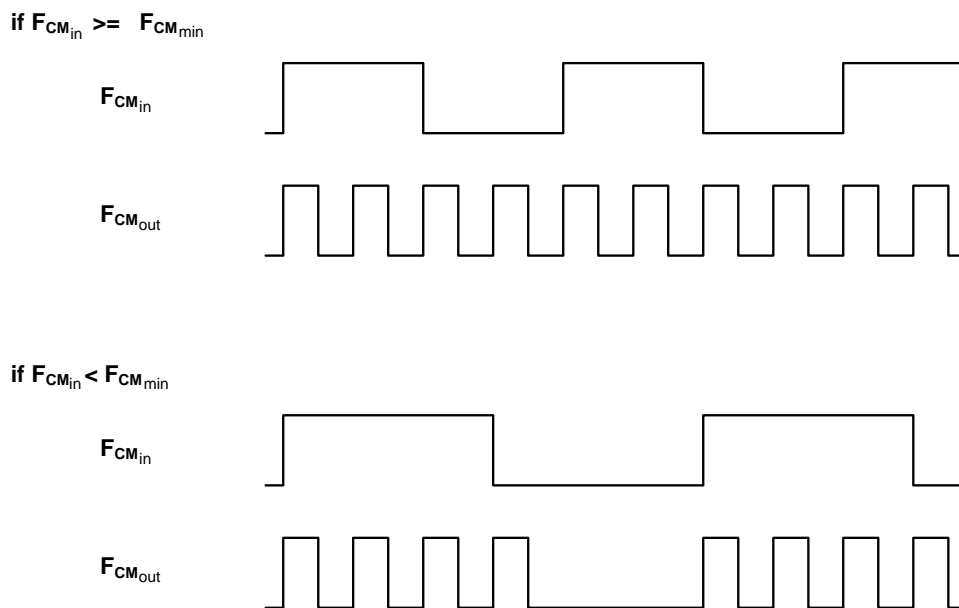


Figure 17.2. Example Clock Multiplier Output

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 M Ω 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.

19. Local Interconnect Network (LIN)

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

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

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

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

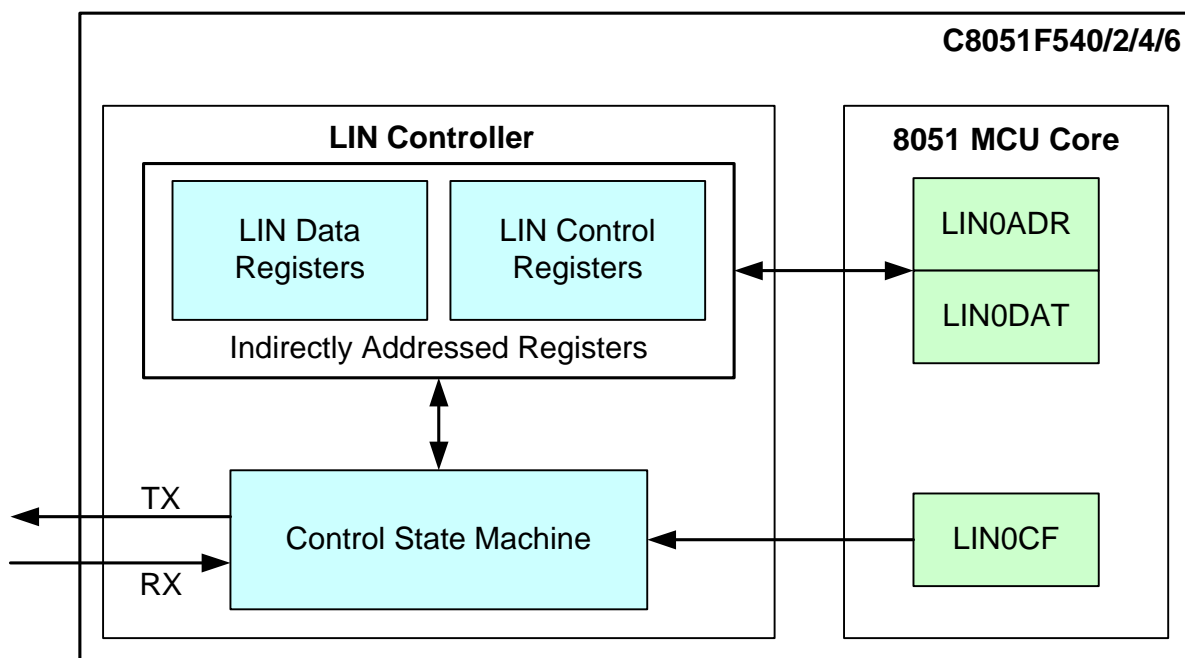


Figure 19.1. LIN Block Diagram

The LIN controller has four main components:

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

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

LIN Register Definition 19.6. LIN0ST: LIN0 Status Register

Bit	7	6	5	4	3	2	1	0
Name	ACTIVE	IDLTOU	ABORT	DTREQ	LININT	ERROR	WAKEUP	DONE
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Indirect Address = 0x09

Bit	Name	Function
7	ACTIVE	LIN Active Indicator Bit. 0: No transmission activity detected on the LIN bus. 1: Transmission activity detected on the LIN bus.
6	IDLTOU	Bus Idle Timeout Bit. (slave mode only) 0: The bus has not been idle for four seconds. 1: No bus activity has been detected for four seconds, but the bus is not yet in Sleep mode.
5	ABORT	Aborted Transmission Bit. (slave mode only) 0: The current transmission has not been interrupted or stopped. This bit is reset to 0 after receiving a SYNCH BREAK that does not interrupt a pending transmission. 1: New SYNCH BREAK detected before the end of the last transmission or the STOP bit (LIN0CTRL.7) has been set.
4	DTREQ	Data Request Bit. (slave mode only) 0: Data identifier has not been received. 1: Data identifier has been received.
3	LININT	Interrupt Request Bit. 0: An interrupt is not pending. This bit is cleared by setting RSTINT (LIN0CTRL.3) 1: There is a pending LIN0 interrupt.
2	ERROR	Communication Error Bit. 0: No error has been detected. This bit is cleared by setting RSTERR (LIN0CTRL.2) 1: An error has been detected.
1	WAKEUP	Wakeup Bit. 0: A wakeup signal is not being transmitted and has not been received. 1: A wakeup signal is being transmitted or has been received
0	DONE	Transmission Complete Bit. 0: A transmission is not in progress or has not been started. This bit is cleared at the start of a transmission. 1: The current transmission is complete.

LIN Register Definition 19.8. LIN0SIZE: LIN0 Message Size Register

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

Indirect Address = 0x0B

Bit	Name	Function
7	ENHCHK	Checksum Selection Bit. 0: Use the classic, specification 1.3 compliant checksum. Checksum covers the data bytes. 1: Use the enhanced, specification 2.0 compliant checksum. Checksum covers data bytes and protected identifier.
6:4	Unused	Read = 000b; Write = Don't Care
3:0	LINSIZE[3:0]	Data Field Size. 0000: 0 data bytes 0001: 1 data byte 0010: 2 data bytes 0011: 3 data bytes 0100: 4 data bytes 0101: 5 data bytes 0110: 6 data bytes 0111: 7 data bytes 1000: 8 data bytes 1001-1110: RESERVED 1111: Use the ID[1:0] bits (LIN0ID[5:4]) to determine the data length.

LIN Register Definition 19.11. LIN0ID: LIN0 Identifier Register

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

Indirect Address = 0x0E

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

20.4.3. Data Register

The SMBus Data register SMB0DAT holds a byte of serial data to be transmitted or one that has just been received. Software may safely read or write to the data register when the SI flag is set. Software should not attempt to access the SMB0DAT register when the SMBus is enabled and the SI flag is cleared to logic 0, as the interface may be in the process of shifting a byte of data into or out of the register.

Data in SMB0DAT is always shifted out MSB first. After a byte has been received, the first bit of received data is located at the MSB of SMB0DAT. While data is being shifted out, data on the bus is simultaneously being shifted in. SMB0DAT always contains the last data byte present on the bus. In the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data or address in SMB0DAT.

SFR Definition 20.3. SMB0DAT: SMBus Data

Bit	7	6	5	4	3	2	1	0
Name	SMB0DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC2; SMB0DAT = 0x00

Bit	Name	Function
7:0	SMB0DAT[7:0]	<p>SMBus Data.</p> <p>The SMB0DAT register contains a byte of data to be transmitted on the SMBus serial interface or a byte that has just been received on the SMBus serial interface. The CPU can read from or write to this register whenever the SI serial interrupt flag (SMB0CN.0) is set to logic 1. The serial data in the register remains stable as long as the SI flag is set. When the SI flag is not set, the system may be in the process of shifting data in/out and the CPU should not attempt to access this register.</p>

20.5. SMBus Transfer Modes

The SMBus interface may be configured to operate as master and/or slave. At any particular time, it will be operating in one of the following four modes: Master Transmitter, Master Receiver, Slave Transmitter, or Slave Receiver. The SMBus interface enters Master Mode any time a START is generated, and remains in Master Mode until it loses an arbitration or generates a STOP. An SMBus interrupt is generated at the end of all SMBus byte frames. As a receiver, the interrupt for an ACK occurs **before** the ACK. As a transmitter, interrupts occur **after** the ACK.

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).

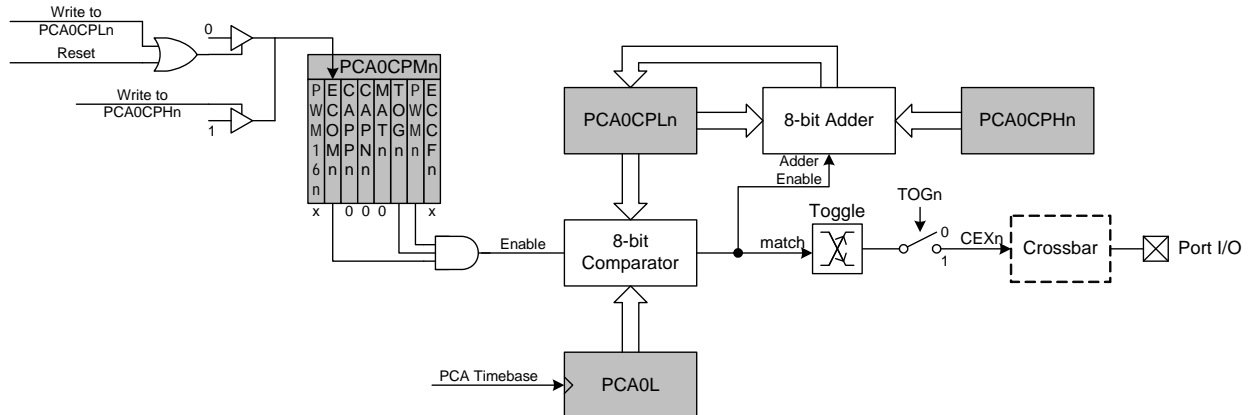


Figure 24.7. PCA Frequency Output Mode

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 24.5. PCA0L: PCA Counter/Timer Low Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xF9; SFR Page = 0x00

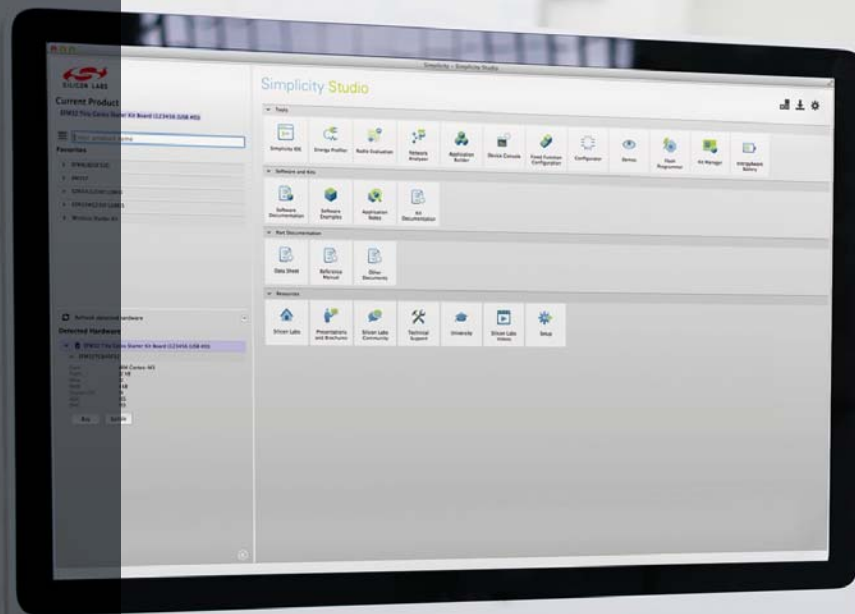
Bit	Name	Function
7:0	PCA0[7:0]	PCA Counter/Timer Low Byte. The PCA0L register holds the low byte (LSB) of the 16-bit PCA Counter/Timer.
Note: When the WDTE bit is set to 1, the PCA0L register cannot be modified by software. To change the contents of the PCA0L register, the Watchdog Timer must first be disabled.		

SFR Definition 24.6. PCA0H: PCA Counter/Timer High Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xFA; SFR Page = 0x00

Bit	Name	Function
7:0	PCA0[15:8]	PCA Counter/Timer High Byte. The PCA0H register holds the high byte (MSB) of the 16-bit PCA Counter/Timer. Reads of this register will read the contents of a “snapshot” register, whose contents are updated only when the contents of PCA0L are read (see Section 24.1).
Note: When the WDTE bit is set to 1, the PCA0H register cannot be modified by software. To change the contents of the PCA0H register, the Watchdog Timer must first be disabled.		



Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!



IoT Portfolio
www.silabs.com/IoT



SW/HW
www.silabs.com/simplicity



Quality
www.silabs.com/quality



Support and Community
community.silabs.com

Disclaimer

Silicon Laboratories intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Laboratories products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Laboratories reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Laboratories shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products must not be used within any Life Support System without the specific written consent of Silicon Laboratories. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Laboratories products are generally not intended for military applications. Silicon Laboratories products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

Trademark Information

Silicon Laboratories Inc., Silicon Laboratories, Silicon Labs, SiLabs and the Silicon Labs logo, CMEMS®, EFM, EFM32, EFR, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZMac®, EZRadio®, EZRadioPRO®, DSPLL®, ISOmodem®, Precision32®, ProSLIC®, SiPHY®, USBXpress® and others are trademarks or registered trademarks of Silicon Laboratories Inc. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.



Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

<http://www.silabs.com>