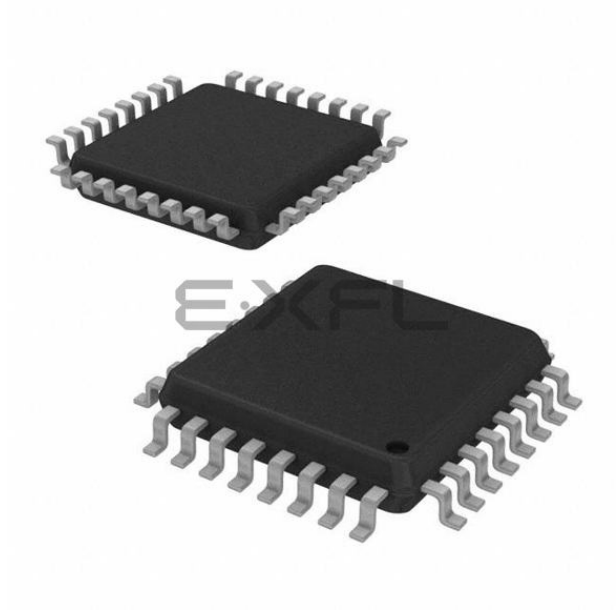

Welcome to E-XFL.COM

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), CANbus, 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	2.25K 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/c8051f565-iqr

1. System Overview

C8051F55x/56x/57x devices are fully integrated mixed-signal System-on-a-Chip MCUs. Highlighted features are listed below. Refer to Table 2.1 for specific product feature selection and part ordering numbers.

- High-speed pipelined 8051-compatible microcontroller core (up to 50 MIPS)
- In-system, full-speed, non-intrusive debug interface (on-chip)
- Controller Area Network (CAN 2.0B) Controller with 32 message objects, each with its own indentifier mask (C8051F550/1/4/5, ‘F560/1/4/5/8/9, and ‘F572/3)
- LIN 2.1 peripheral (fully backwards compatible, master and slave modes) (C8051F550/2/4/6, ‘F560/2/4/6/8, and ‘F570/2/4)
- True 12-bit 200 ksps 32-channel single-ended ADC with analog multiplexer
- Precision programmable 24 MHz internal oscillator that is within ±0.5% across the temperature range and for VDD voltages greater than or equal to the on-chip voltage regulator minimum output at the low setting. The oscillator is within ±1.0% for VDD voltages below this minimum output setting.
- On-chip Clock Multiplier to reach up to 50 MHz
- 32 kB (C8051F550-3, ‘F560-3, ‘F568-9, and ‘F570-1) or 16 kB (C8051F554-7, ‘F564-7, and ‘F572-5) of on-chip Flash memory
- 2304 bytes of on-chip RAM
- SMBus/I2C, Enhanced UART, and Enhanced SPI serial interfaces implemented in hardware
- Four general-purpose 16-bit timers
- External Data Memory Interface (C8051F568-9 and ‘F570-5) with 64 kB address space
- Programmable Counter/Timer Array (PCA) with six capture/compare modules and Watchdog Timer function
- On-chip Voltage Regulator
- On-chip Power-On Reset, V_{DD} Monitor, and Temperature Sensor
- On-chip Voltage Comparator
- 33, 25, or 18 Port I/O (5 V push-pull)

With on-chip Voltage Regulator, Power-On Reset, V_{DD} monitor, Watchdog Timer, and clock oscillator, the C8051F55x/56x/57x devices are truly stand-alone System-on-a-Chip solutions. The Flash memory can be reprogrammed even in-circuit, providing non-volatile data storage, and also allowing field upgrades of the 8051 firmware. User software has complete control of all peripherals, and may individually shut down any or all peripherals for power savings.

The on-chip Silicon Labs 2-Wire (C2) Development Interface allows non-intrusive (uses no on-chip resources), full speed, in-circuit debugging using the production MCU installed in the final application. This debug logic supports inspection and modification of memory and registers, setting breakpoints, single stepping, run and halt commands. All analog and digital peripherals are fully functional while debugging using C2. The two C2 interface pins can be shared with user functions, allowing in-system debugging without occupying package pins.

The devices are specified for 1.8 V to 5.25 V operation over the automotive temperature range (–40 to +125 °C). The C8051F568-9 and ‘F570-5 are available in 40-pin QFN packages, the C8051F560-7 devices are available in 32-pin QFP and QFN packages, and the C8051F550-7 are available in 24-pin QFN packages. All package options are lead-free and RoHS compliant. See Table 2.1 for ordering information. Block diagrams are included in Figure 1.1, Figure 1.2, and Figure 1.3.

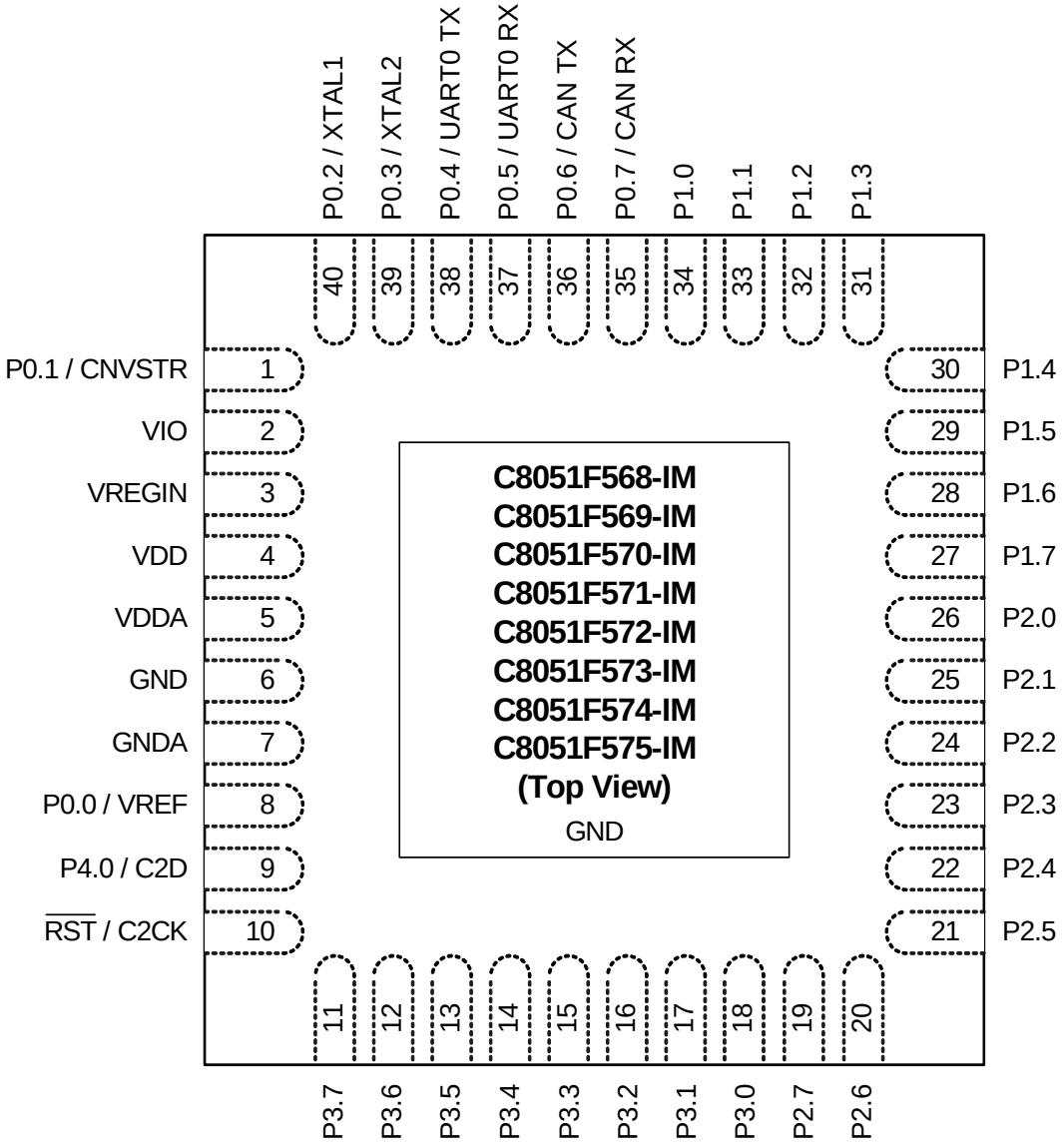

Figure 3.1. QFN-40 Pinout Diagram (Top View)

6.3.2. Setting the Gain Value

The three programmable gain registers are accessed indirectly using the ADC0H and ADC0L registers when the GAINEN bit (ADC0CF.0) bit is set. ADC0H acts as the address register, and ADC0L is the data register. The programmable gain registers can only be written to and cannot be read. See Gain Register Definition 6.1, Gain Register Definition 6.2, and Gain Register Definition 6.3 for more information.

The gain is programmed using the following steps:

1. Set the GAINEN bit (ADC0CF.0)
2. Load the ADC0H with the ADC0GNH, ADC0GNL, or ADC0GNA address.
3. Load ADC0L with the desired value for the selected gain register.
4. Reset the GAINEN bit (ADC0CF.0)

Notes:

1. An ADC conversion should not be performed while the GAINEN bit is set.
2. Even with gain enabled, the maximum input voltage must be less than V_{REGIN} and the maximum voltage of the signal after gain must be less than or equal to V_{REF}.

In code, changing the value to 0.44 gain from the previous example looks like:

```
// in 'C':
ADC0CF |= 0x01;          // GAINEN = 1
ADC0H = 0x04;            // Load the ADC0GNH address
ADC0L = 0x6C;            // Load the upper byte of 0x6CA to ADC0GNH
ADC0H = 0x07;            // Load the ADC0GNL address
ADC0L = 0xA0;            // Load the lower nibble of 0x6CA to ADC0GNL
ADC0H = 0x08;            // Load the ADC0GNA address
ADC0L = 0x01;            // Set the GAINADD bit
ADC0CF &= ~0x01;         // GAINEN = 0

; in assembly
ORL ADC0CF,#01H          ; GAINEN = 1
MOV ADC0H,#04H           ; Load the ADC0GNH address
MOV ADC0L,#06CH          ; Load the upper byte of 0x6CA to ADC0GNH
MOV ADC0H,#07H           ; Load the ADC0GNL address
MOV ADC0L,#0A0H          ; Load the lower nibble of 0x6CA to ADC0GNL
MOV ADC0H,#08H           ; Load the ADC0GNA address
MOV ADC0L,#01H           ; Set the GAINADD bit
ANL ADC0CF,#0FEH         ; GAINEN = 0
```

SFR Definition 6.7. ADC0CN: ADC0 Control

Bit	7	6	5	4	3	2	1	0
Name	AD0EN	BURSTEN	AD0INT	AD0BUSY	AD0WINT	AD0LJST	AD0CM[1:0]	
Type	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 = 0xE8; SFR Page = 0x00; Bit-Addressable

Bit	Name	Function		
7	AD0EN	**ADC0 Enable Bit.** 0: ADC0 Disabled. ADC0 is in low-power shutdown. 1: ADC0 Enabled. ADC0 is active and ready for data conversions.		
6	BURSTEN	**ADC0 Burst Mode Enable Bit.** 0: Burst Mode Disabled. 1: Burst Mode Enabled.		
5	AD0INT	**ADC0 Conversion Complete Interrupt Flag.** 0: ADC0 has not completed a data conversion since AD0INT was last cleared. 1: ADC0 has completed a data conversion.		
4	AD0BUSY	**ADC0 Busy Bit.**	**Read:** 0: ADC0 conversion is not in progress. 1: ADC0 conversion is in progress.	**Write:** 0: No Effect. 1: Initiates ADC0 Conversion if AD0CM[1:0] = 00b
3	AD0WINT	**ADC0 Window Compare Interrupt Flag.** This bit must be cleared by software 0: ADC0 Window Comparison Data match has not occurred since this flag was last cleared. 1: ADC0 Window Comparison Data match has occurred.		
2	AD0LJST	**ADC0 Left Justify Select Bit.** 0: Data in ADC0H:ADC0L registers is right-justified 1: Data in ADC0H:ADC0L registers is left-justified. This option should not be used with a repeat count greater than 1 (when AD0RPT[1:0] is 01b, 10b, or 11b).		
1:0	AD0CM[1:0]	**ADC0 Start of Conversion Mode Select.** 00: ADC0 start-of-conversion source is write of 1 to AD0BUSY. 01: ADC0 start-of-conversion source is overflow of Timer 1. 10: ADC0 start-of-conversion source is rising edge of external CNVSTR. 11: ADC0 start-of-conversion source is overflow of Timer 2.		

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.

Table 12.1. Special Function Register (SFR) Memory Map for Pages 0x00 and 0x0F

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 EMI0CF			P4 (All Pages)	FLSCL (All Pages)	FLKEY (All Pages)
A8	0F	IE (All Pages)	SMOD0	EMI0CN EMI0TC	SBCON0	SBRLL0	SBRLH0	P3MAT P3MDOUT	P3MASK P4MDOUT
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)

Table 12.3. Special Function Registers (Continued)

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

Register	Address	Description	Page
SMB0CF	0xC1	SMBus0 Configuration	224
SMB0CN	0xC0	SMBus0 Control	226
SMB0DAT	0xC2	SMBus0 Data	228
SMOD0	0xA9	UART0 Mode	243
SN0	0xF9	Serial Number 0	91
SN1	0xFA	Serial Number 1	91
SN2	0xFB	Serial Number 2	91
SN3	0xFC	Serial Number 3	91
SP	0x81	Stack Pointer	89
SPI0CFG	0xA1	SPI0 Configuration	253
SPI0CKR	0xA2	SPI0 Clock Rate Control	255
SPI0CN	0xF8	SPI0 Control	254
SPI0DAT	0xA3	SPI0 Data	255
TCON	0x88	Timer/Counter Control	265
TH0	0x8C	Timer/Counter 0 High	268
TH1	0x8D	Timer/Counter 1 High	268
TL0	0x8A	Timer/Counter 0 Low	267
TL1	0x8B	Timer/Counter 1 Low	267
TMOD	0x89	Timer/Counter Mode	266
TMR2CN	0xC8	Timer/Counter 2 Control	272
TMR2H	0xCD	Timer/Counter 2 High	274
TMR2L	0xCC	Timer/Counter 2 Low	274
TMR2RLH	0xCB	Timer/Counter 2 Reload High	273
TMR2RLL	0xCA	Timer/Counter 2 Reload Low	273
TMR3CN	0x91	Timer/Counter 3 Control	278
TMR3H	0x95	Timer/Counter 3 High	280
TMR3L	0x94	Timer/Counter 3 Low	280
TMR3RLH	0x93	Timer/Counter 3 Reload High	279
TMR3RLL	0x92	Timer/Counter 3 Reload Low	279
VDM0CN	0xFF	V_{DD} Monitor Control	141
XBR0	0xE1	Port I/O Crossbar Control 0	176
XBR1	0xE2	Port I/O Crossbar Control 1	177
XBR2	0xC7	Port I/O Crossbar Control 2	178

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 5.4 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 $\overline{\text{RST}}$ pin provides a means for external circuitry to force the device into a reset state. Asserting an active-low signal on the $\overline{\text{RST}}$ pin generates a reset; an external pullup and/or decoupling of the $\overline{\text{RST}}$ pin may be necessary to avoid erroneous noise-induced resets. See Table 5.4 for complete $\overline{\text{RST}}$ 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 value specified in Table 5.4, "Reset Electrical Characteristics," on page 41, 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 $\overline{\text{RST}}$ pin is unaffected by this reset.

17.6.1.2. 8-bit MOVX without Bank Select: EMI0CF[4:2] = 001 or 011

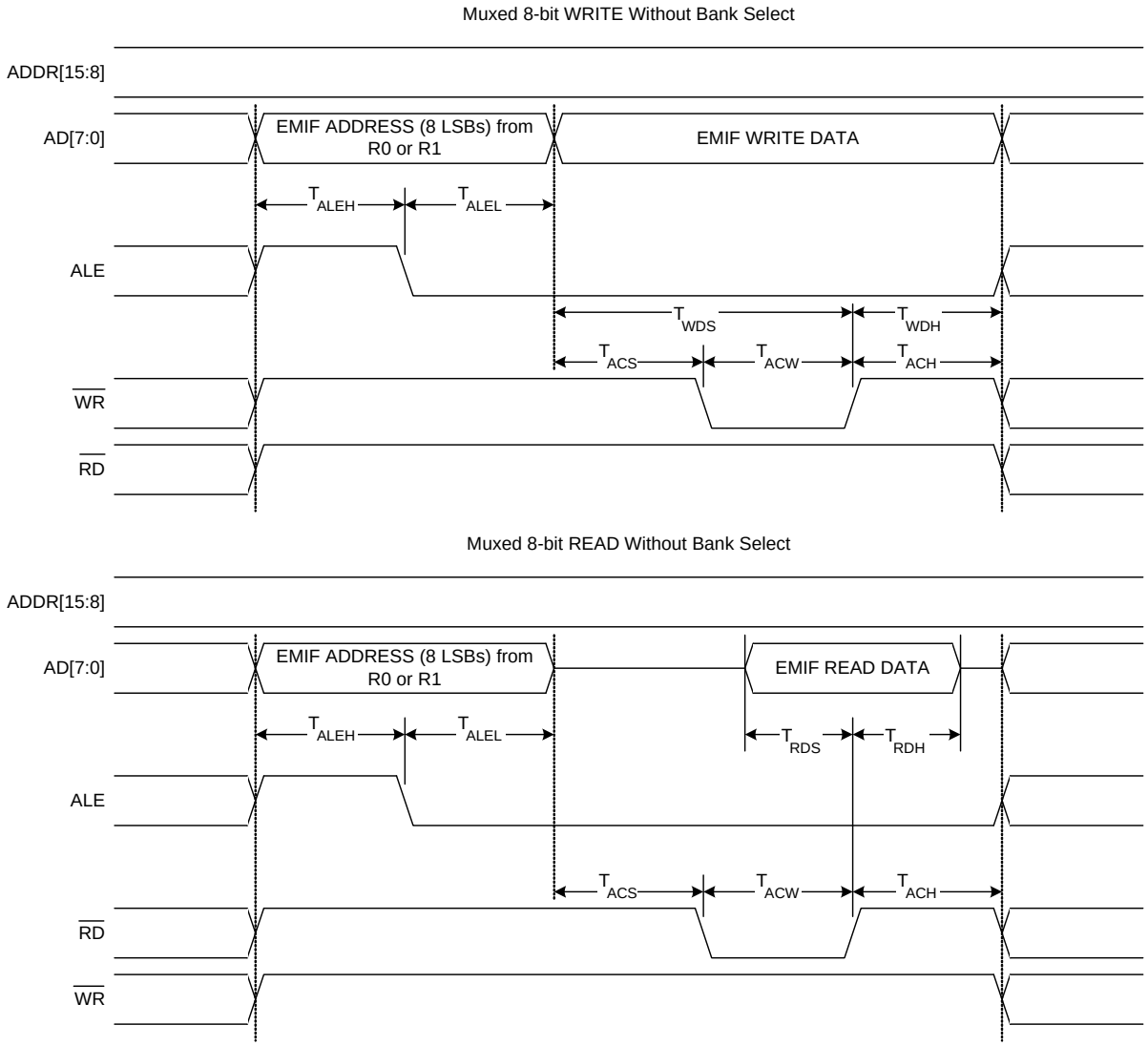

Figure 17.4. Multiplexed 8-bit MOVX without Bank Select Timing

SFR Definition 19.3. XBR2: Port I/O Crossbar Register 1

Bit	7	6	5	4	3	2	1	0
Name	WEAKPUD	XBARE	Reserved					LIN0E
Type	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC7; SFR Page = 0x0F

Bit	Name	Function
7	WEAKPUD	**Port I/O Weak Pullup Disable.** 0: Weak Pullups enabled (except for Ports whose I/O are configured for analog mode). 1: Weak Pullups disabled.
6	XBARE	**Crossbar Enable.** 0: Crossbar disabled. 1: Crossbar enabled.
5:1	Reserved	Always Write to 00000b.
0	LIN0E	**LIN I/O Output Enable.** 0: LIN I/O unavailable at Port pin. 1: LIN_TX, LIN_RX routed to Port pins.

LIN Register Definition 20.8. 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.

22.1. Supporting Documents

It is assumed the reader is familiar with or has access to the following supporting documents:

1. The I^2C-Bus and How to Use It (including specifications), Philips Semiconductor.
2. The I^2C-Bus Specification—Version 2.0, Philips Semiconductor.
3. System Management Bus Specification—Version 1.1, SBS Implementers Forum.

22.2. SMBus Configuration

Figure 22.2 shows a typical SMBus configuration. The SMBus specification allows any recessive voltage between 3.0 V and 5.0 V; different devices on the bus may operate at different voltage levels. The bi-directional SCL (serial clock) and SDA (serial data) lines must be connected to a positive power supply voltage through a pullup resistor or similar circuit. Every device connected to the bus must have an open-drain or open-collector output for both the SCL and SDA lines, so that both are pulled high (recessive state) when the bus is free. The maximum number of devices on the bus is limited only by the requirement that the rise and fall times on the bus not exceed 300 ns and 1000 ns, respectively.

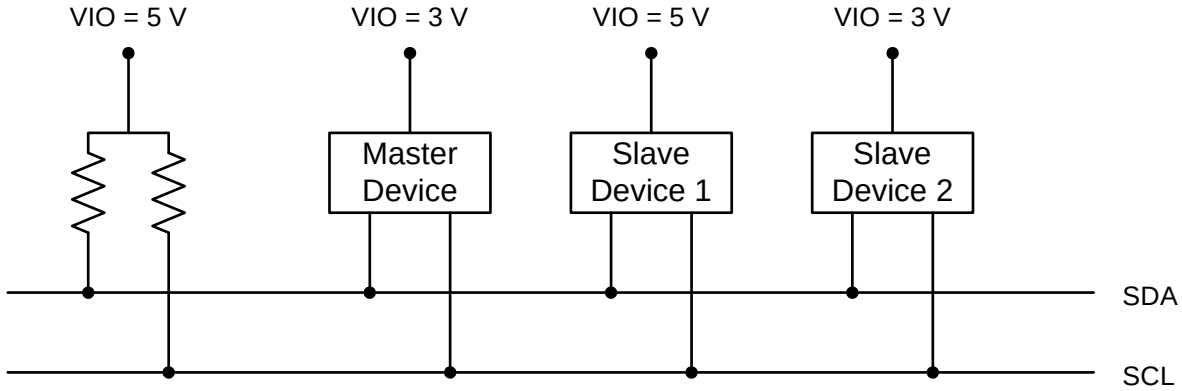

Figure 22.2. Typical SMBus Configuration

22.3. SMBus Operation

Two types of data transfers are possible: data transfers from a master transmitter to an addressed slave receiver (WRITE), and data transfers from an addressed slave transmitter to a master receiver (READ). The master device initiates both types of data transfers and provides the serial clock pulses on SCL. The SMBus interface may operate as a master or a slave, and multiple master devices on the same bus are supported. If two or more masters attempt to initiate a data transfer simultaneously, an arbitration scheme is employed with a single master always winning the arbitration. It is not necessary to specify one device as the Master in a system; any device who transmits a START and a slave address becomes the master for the duration of that transfer.

A typical SMBus transaction consists of a START condition followed by an address byte (Bits7–1: 7-bit slave address; Bit0: R/W direction bit), one or more bytes of data, and a STOP condition. Bytes that are received (by a master or slave) are acknowledged (ACK) with a low SDA during a high SCL (see Figure 22.3). If the receiving device does not ACK, the transmitting device will read a NACK (not acknowledge), which is a high SDA during a high SCL.

The direction bit (R/W) occupies the least-significant bit position of the address byte. The direction bit is set to logic 1 to indicate a "READ" operation and cleared to logic 0 to indicate a "WRITE" operation.

SFR Definition 22.2. SMB0CN: SMBus Control

Bit	7	6	5	4	3	2	1	0
Name	MASTER	TXMODE	STA	STO	ACKRQ	ARBLOST	ACK	SI
Type	R	R	R/W	R/W	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC0; Bit-Addressable; SFR Page =0x00

Bit	Name	Description	Read	Write
7	MASTER	**SMBus Master/Slave Indicator.** This read-only bit indicates when the SMBus is operating as a master.	0: SMBus operating in slave mode. 1: SMBus operating in master mode.	N/A
6	TXMODE	**SMBus Transmit Mode Indicator.** This read-only bit indicates when the SMBus is operating as a transmitter.	0: SMBus in Receiver Mode. 1: SMBus in Transmitter Mode.	N/A
5	STA	**SMBus Start Flag.**	0: No Start or repeated Start detected. 1: Start or repeated Start detected.	0: No Start generated. 1: When Configured as a Master, initiates a START or repeated START.
4	STO	**SMBus Stop Flag.**	0: No Stop condition detected. 1: Stop condition detected (if in Slave Mode) or pending (if in Master Mode).	0: No STOP condition is transmitted. 1: When configured as a Master, causes a STOP condition to be transmitted after the next ACK cycle. Cleared by Hardware.
3	ACKRQ	**SMBus Acknowledge Request.**	0: No Ack requested 1: ACK requested	N/A
2	ARBLOST	**SMBus Arbitration Lost Indicator.**	0: No arbitration error. 1: Arbitration Lost	N/A
1	ACK	**SMBus Acknowledge.**	0: NACK received. 1: ACK received.	0: Send NACK 1: Send ACK
0	SI	**SMBus Interrupt Flag.** This bit is set by hardware under the conditions listed in Table 15.3. SI must be cleared by software. While SI is set, SCL is held low and the SMBus is stalled.	0: No interrupt pending 1: Interrupt Pending	0: Clear interrupt, and initiate next state machine event. 1: Force interrupt.

Table 22.4. SMBus Status Decoding (Continued)

Mode	Values Read				Current SMbus State	Typical Response Options	Values to Write			Next Status Vector Expected
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
Slave Transmitter	0100	0	0	0	A slave byte was transmitted; NACK received.	No action required (expecting STOP condition).	0	0	X	0001
		0	0	1	A slave byte was transmitted; ACK received.	Load SMB0DAT with next data byte to transmit.	0	0	X	0100
		0	1	X	A Slave byte was transmitted; error detected.	No action required (expecting Master to end transfer).	0	0	X	0001
	0101	0	X	X	An illegal STOP or bus error was detected while a Slave Transmission was in progress.	Clear STO.	0	0	X	—
Slave Receiver	0010	1	0	X	A slave address + R/W was received; ACK requested.	If Write, Acknowledge received address	0	0	1	0000
						If Read, Load SMB0DAT with data byte; ACK received address	0	0	1	0100
						NACK received address.	0	0	0	—
		1	1	X	Lost arbitration as master; slave address + R/W received; ACK requested.	If Write, Acknowledge received address	0	0	1	0000
						If Read, Load SMB0DAT with data byte; ACK received address	0	0	1	0100
						NACK received address.	0	0	0	—
						Reschedule failed transfer; NACK received address.	1	0	0	1110
	0001	0	0	X	A STOP was detected while addressed as a Slave Transmitter or Slave Receiver.	Clear STO.	0	0	X	—
		1	1	X	Lost arbitration while attempting a STOP.	No action required (transfer complete/aborted).	0	0	0	—
	0000	1	0	X	A slave byte was received; ACK requested.	Acknowledge received byte; Read SMB0DAT.	0	0	1	0000
						NACK received byte.	0	0	0	—
Bus Error Condition	0010	0	1	X	Lost arbitration while attempting a repeated START.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110
	0001	0	1	X	Lost arbitration due to a detected STOP.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110
	0000	1	1	X	Lost arbitration while transmitting a data byte as master.	Abort failed transfer.	0	0	0	—
						Reschedule failed transfer.	1	0	0	1110

23. UART0

UART0 is an asynchronous, full duplex serial port offering a variety of data formatting options. A dedicated baud rate generator with a 16-bit timer and selectable prescaler is included, which can generate a wide range of baud rates (details in Section “23.1. Baud Rate Generator” on page 235). A received data FIFO allows UART0 to receive up to three data bytes before data is lost and an overflow occurs.

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

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

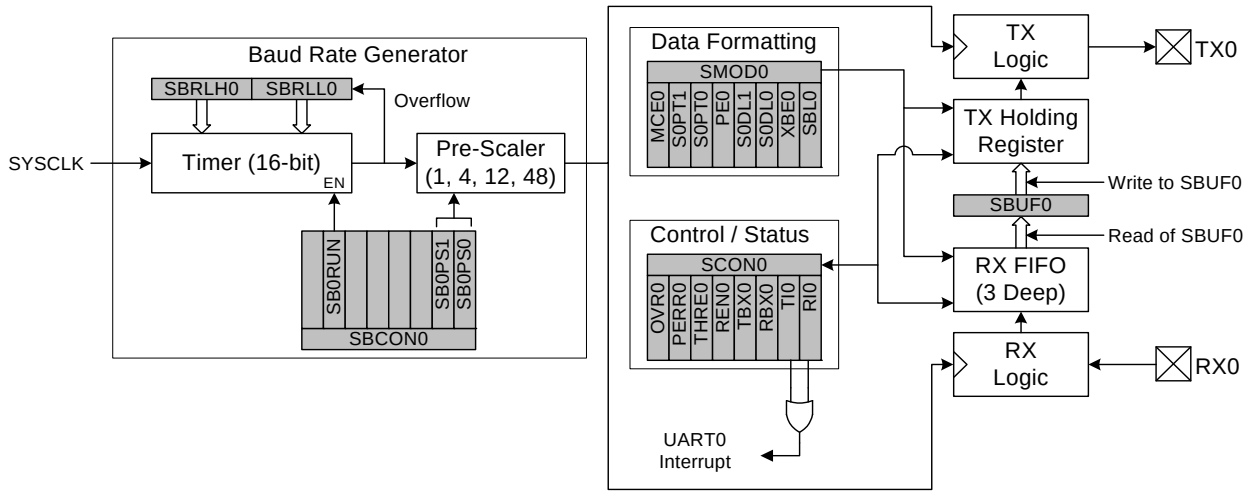

Figure 23.1. UART0 Block Diagram

23.1. Baud Rate Generator

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

The baud rate generator is configured using three registers: SBCON0, SBRLH0, and SBRLL0. The UART0 Baud Rate Generator Control Register (SBCON0, SFR Definition 23.4) enables or disables the baud rate generator and selects the prescaler value for the timer. The baud rate generator must be enabled for UART0 to function. Registers SBRLH0 and SBRLL0 contain a 16-bit reload value for the dedicated 16-bit timer. The internal timer counts up from the reload value on every clock tick. On timer overflows (0xFFFF to 0x0000), the timer is reloaded. The baud rate for UART0 is defined in Equation 23.1, where “BRG Clock” is the baud rate generator’s selected clock source. For reliable UART operation, it is recommended that the UART baud rate is not configured for baud rates faster than SYSCLK/16.

SFR Definition 25.2. TCON: Timer Control

Bit	7	6	5	4	3	2	1	0
Name	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
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 = 0x88; Bit-Addressable; SFR Page = All Pages

Bit	Name	Function
7	TF1	**Timer 1 Overflow Flag.** Set to 1 by hardware when Timer 1 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 1 interrupt service routine.
6	TR1	**Timer 1 Run Control.** Timer 1 is enabled by setting this bit to 1.
5	TF0	**Timer 0 Overflow Flag.** Set to 1 by hardware when Timer 0 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 0 interrupt service routine.
4	TR0	**Timer 0 Run Control.** Timer 0 is enabled by setting this bit to 1.
3	IE1	**External Interrupt 1.** This flag is set by hardware when an edge/level of type defined by IT1 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 1 service routine in edge-triggered mode.
2	IT1	**Interrupt 1 Type Select.** This bit selects whether the configured $\overline{\text{INT1}}$ interrupt will be edge or level sensitive. $\overline{\text{INT1}}$ is configured active low or high by the IN1PL bit in the IT01CF register (see SFR Definition 13.7). 0: $\overline{\text{INT1}}$ is level triggered. 1: $\overline{\text{INT1}}$ is edge triggered.
1	IE0	**External Interrupt 0.** This flag is set by hardware when an edge/level of type defined by IT1 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 0 service routine in edge-triggered mode.
0	IT0	**Interrupt 0 Type Select.** This bit selects whether the configured $\overline{\text{INT0}}$ interrupt will be edge or level sensitive. $\overline{\text{INT0}}$ is configured active low or high by the IN0PL bit in register IT01CF (see SFR Definition 13.7). 0: $\overline{\text{INT0}}$ is level triggered. 1: $\overline{\text{INT0}}$ is edge triggered.

T3MH	T3XCLK	TMR3H Clock Source
0	0	SYSCLK/12
0	1	External Clock/8
1	X	SYSCLK

T3ML	T3XCLK	TMR3L Clock Source
0	0	SYSCLK/12
0	1	External Clock/8
1	X	SYSCLK

The TF3H bit is set when TMR3H overflows from 0xFF to 0x00; the TF3L bit is set when TMR3L overflows from 0xFF to 0x00. When Timer 3 interrupts are enabled, an interrupt is generated each time TMR3H overflows. If Timer 3 interrupts are enabled and TF3LEN (TMR3CN.5) is set, an interrupt is generated each time either TMR3L or TMR3H overflows. When TF3LEN is enabled, software must check the TF3H and TF3L flags to determine the source of the Timer 3 interrupt. The TF3H and TF3L interrupt flags are not cleared by hardware and must be manually cleared by software.

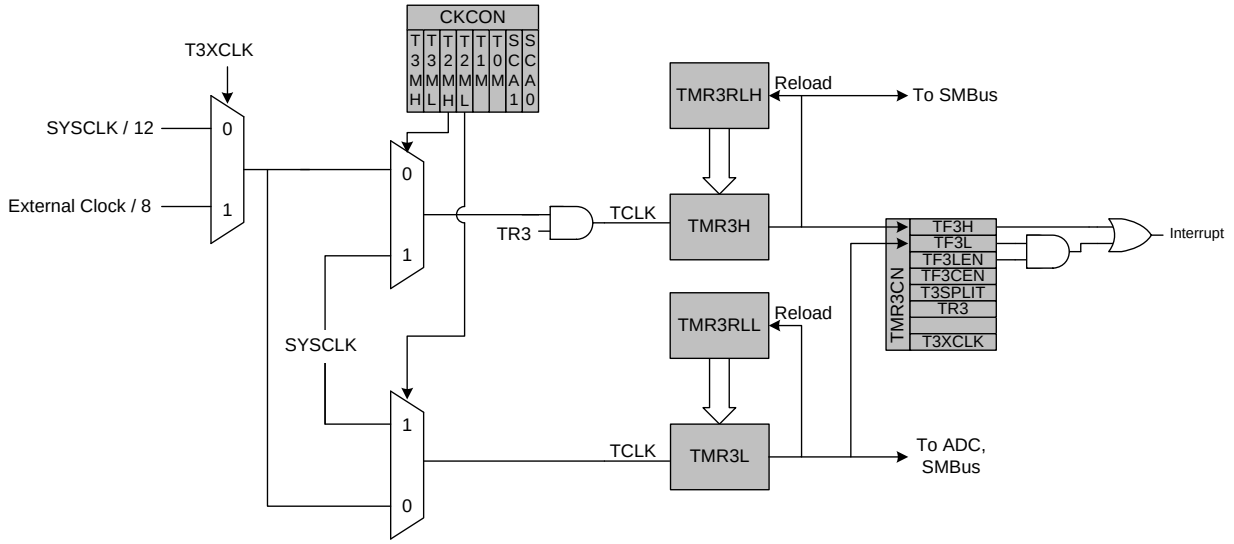

Figure 25.8. Timer 3 8-Bit Mode Block Diagram

25.3.3. External Oscillator Capture Mode

Capture Mode allows the external oscillator to be measured against the system clock. Timer 3 can be clocked from the system clock, or the system clock divided by 12, depending on the T3ML (CKCON.6), and T3XCLK bits. When a capture event is generated, the contents of Timer 3 (TMR3H:TMR3L) are loaded into the Timer 3 reload registers (TMR3RLH:TMR3RLL) and the TF3H flag is set. A capture event is generated by the falling edge of the clock source being measured, which is the external oscillator/8. By recording the difference between two successive timer capture values, the external oscillator frequency can be determined with respect to the Timer 3 clock. The Timer 3 clock should be much faster than the capture clock to achieve an accurate reading. Timer 3 should be in 16-bit auto-reload mode when using Capture Mode.

If the SYSCLK is 24 MHz and the difference between two successive captures is 5861, then the external clock frequency is as follows:

24 MHz/(5861/8) = 0.032754 MHz or 32.754 kHz

This mode allows software to determine the external oscillator frequency when an RC network or capacitor is used to generate the clock source.

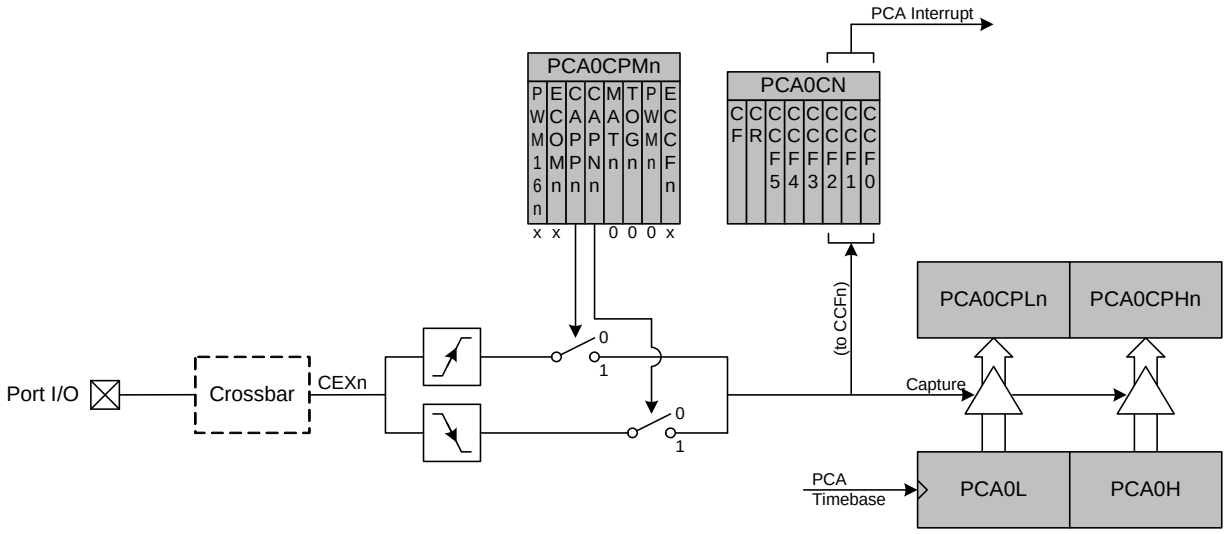

Figure 26.4. PCA Capture Mode Diagram

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

26.3.2. 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.

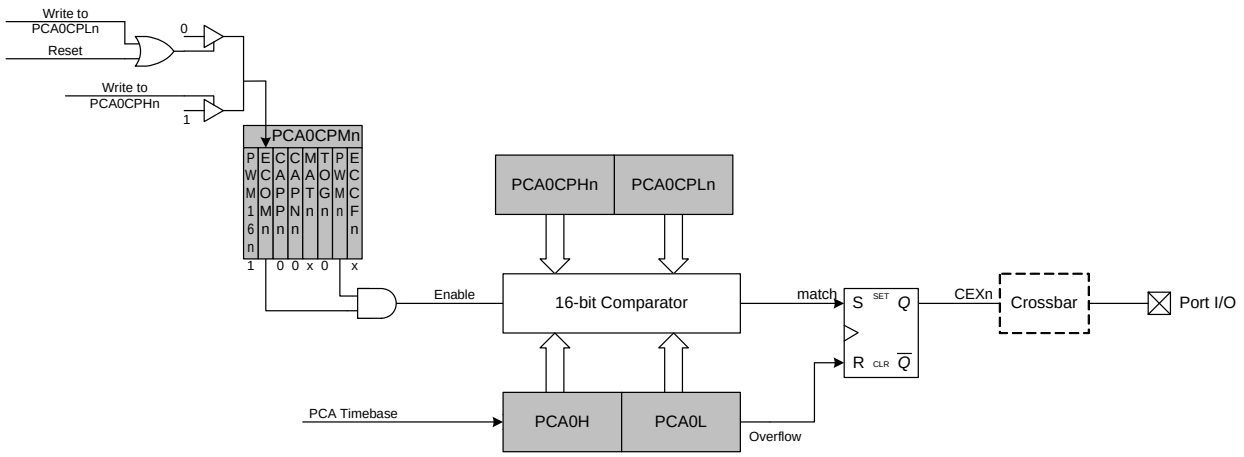

Figure 26.10. PCA 16-Bit PWM Mode

26.4. Watchdog Timer Mode

A programmable watchdog timer (WDT) function is available through the PCA Module 5. The WDT is used to generate a reset if the time between writes to the WDT update register (PCA0CPH5) exceed a specified limit. The WDT can be configured and enabled/disabled as needed by software.

With the WDTE bit set in the PCA0MD register, Module 5 operates as a watchdog timer (WDT). The Module 5 high byte is compared to the PCA counter high byte; the Module 5 low byte holds the offset to be used when WDT updates are performed. **The Watchdog Timer is enabled on reset. Writes to some PCA registers are restricted while the Watchdog Timer is enabled.** The WDT will generate a reset shortly after code begins execution. To avoid this reset, the WDT should be explicitly disabled (and optionally re-configured and re-enabled if it is used in the system).

26.4.1. Watchdog Timer Operation

While the WDT is enabled:

- PCA counter is forced on.
- Writes to PCA0L and PCA0H are not allowed.
- PCA clock source bits (CPS[2:0]) are frozen.
- PCA Idle control bit (CIDL) is frozen.
- Module 5 is forced into software timer mode.
- Writes to the Module 5 mode register (PCA0CPM5) are disabled.

While the WDT is enabled, writes to the CR bit will not change the PCA counter state; the counter will run until the WDT is disabled. The PCA counter run control bit (CR) will read zero if the WDT is enabled but user software has not enabled the PCA counter. If a match occurs between PCA0CPH5 and PCA0H while the WDT is enabled, a reset will be generated. To prevent a WDT reset, the WDT may be updated with a write of any value to PCA0CPH5. Upon a PCA0CPH5 write, PCA0H plus the offset held in PCA0CPL5 is loaded into PCA0CPH5 (See Figure 26.11).

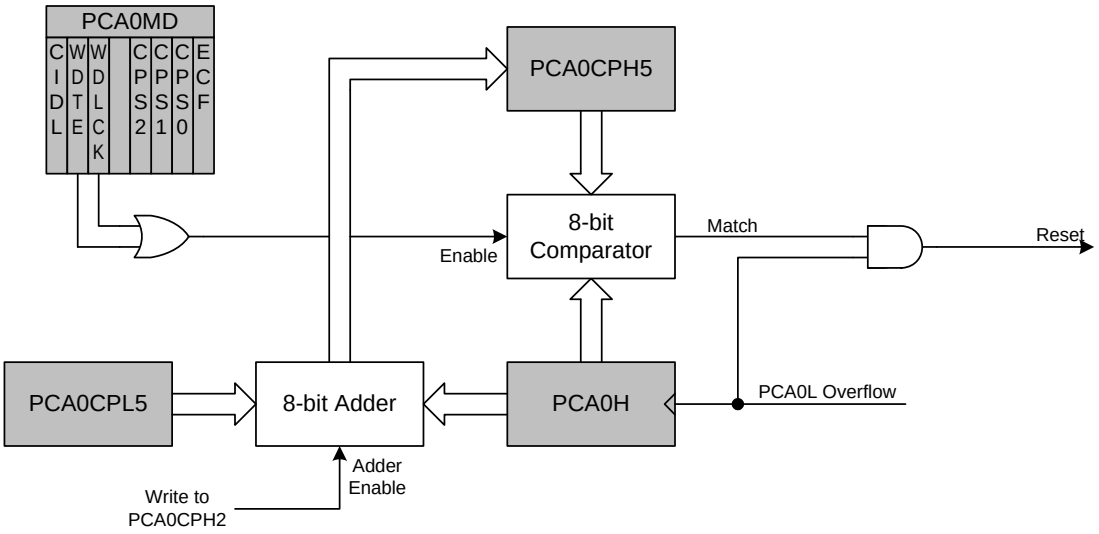

Figure 26.11. PCA Module 2 with Watchdog Timer Enabled

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 26.5, where PCA0L is the value of the PCA0L register at the time of the update.

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

Equation 26.5. 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.

26.4.2. 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 26.5, this results in a WDT timeout interval of 256 PCA clock cycles, or 3072 system clock cycles. Table 26.3 lists some example timeout intervals for typical system clocks.