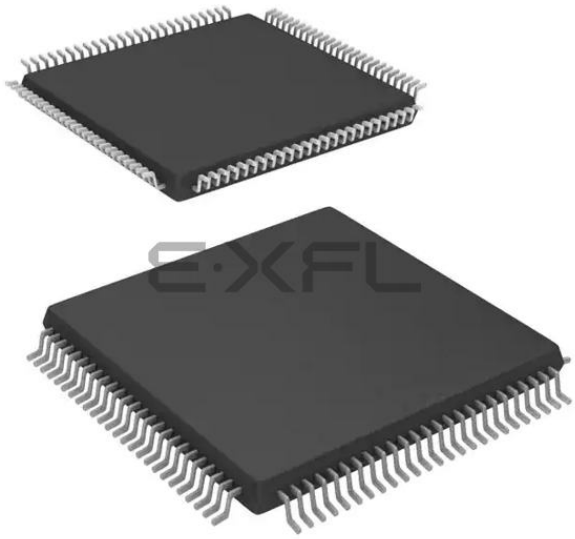


Welcome to E-XFL.COM

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	Not For New Designs
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	CANbus, EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	59
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 2x16b, 8x10b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f060-gq

Table 1.1. Product Selection Guide

	MIPS (Peak)	Flash Memory	RAM	External Memory Interface	SMBus/I2C and SPI	CAN	UARTS	Timers (16-bit)	Programmable Counter Array	Digital Port I/O's	16-bit 1 Msps ADC Typical INL (LSBs)	10-bit 200 ksps ADC Inputs	Voltage Reference	Temperature Sensor	DAC Resolution (bits)	DAC Outputs	Analog Comparators	Package
C8051F060	25	64 k	4352	✓	✓	✓	2	5	✓	59	±0.75	8	✓	✓	12	2	3	100 TQFP
C8051F061	25	64 k	4352	-	✓	✓	2	5	✓	24	±0.75	8	✓	✓	12	2	3	64 TQFP
C8051F062	25	64 k	4352	✓	✓	✓	2	5	✓	59	±1.5	8	✓	✓	12	2	3	100 TQFP
C8051F063	25	64 k	4352	-	✓	✓	2	5	✓	24	±1.5	8	✓	✓	12	2	3	64 TQFP
C8051F064	25	64 k	4352	✓	✓	-	2	5	✓	59	±0.75	-	✓	-	-	-	3	100 TQFP
C8051F065	25	64 k	4352	-	✓	-	2	5	✓	24	±0.75	-	✓	-	-	-	3	64 TQFP
C8051F066	25	32 k	4352	✓	✓	-	2	5	✓	59	±0.75	-	✓	-	-	-	3	100 TQFP
C8051F067	25	32 k	4352	-	✓	-	2	5	✓	24	±0.75	-	✓	-	-	-	3	64 TQFP

1.6. Controller Area Network

The C8051F060/1/2/3 devices feature a Controller Area Network (CAN) controller that implements serial communication using the CAN protocol. The CAN controller facilitates communication on a CAN network in accordance with the Bosch specification 2.0A (basic CAN) and 2.0B (full CAN). The CAN controller consists of a CAN Core, Message RAM (separate from the C8051 RAM), a message handler state machine, and control registers.

The CAN controller can operate at bit rates up to 1 Mbit/second. Silicon Labs CAN has 32 message objects each having its own identifier mask used for acceptance filtering of received messages. Incoming data, message objects and identifier masks are stored in the CAN message RAM. All protocol functions for transmission of data and acceptance filtering is performed by the CAN controller and not by the C8051 MCU. In this way, minimal CPU bandwidth is used for CAN communication. The C8051 configures the CAN controller, accesses received data, and passes data for transmission via Special Function Registers (SFR) in the C8051.

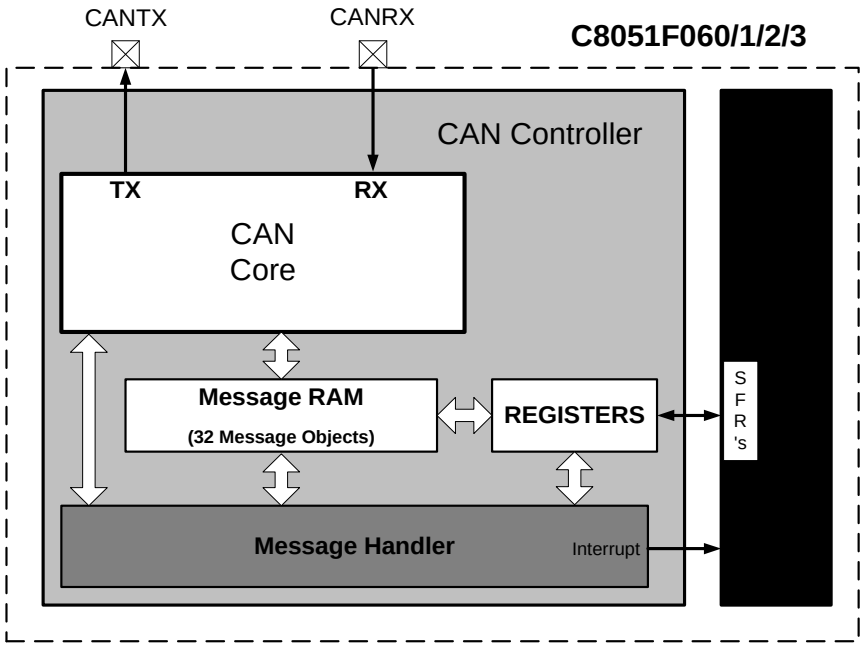


Figure 1.11. CAN Controller Overview

Table 4.1. Pin Definitions (Continued)

Name	Pin Numbers				Type	Description
	F060	F061	F064	F065		
	F062	F063	F066	F067		
P0.5	57	46	57	46	D I/O	Port 0.5. See Port Input/Output section for complete description.
P0.6	56	45	56	45	D I/O	Port 0.6. See Port Input/Output section for complete description.
P0.7	55	44	55	44	D I/O	Port 0.7. See Port Input/Output section for complete description.
P1.0/AIN2.0	36	33	36	33	D I/O A In	Port 1.0. See Port Input/Output section for complete description. ADC2 Input Channel 0 (C8051F060/1/2/3 Only).
P1.1/AIN2.1	35	32	35	32	D I/O A In	Port 1.1. See Port Input/Output section for complete description. ADC2 Input Channel 1 (C8051F060/1/2/3 Only).
P1.2/AIN2.2	34	31	34	31	D I/O A In	Port 1.2. See Port Input/Output section for complete description. ADC2 Input Channel 2 (C8051F060/1/2/3 Only).
P1.3/AIN2.3	33	30	33	30	D I/O A In	Port 1.3. See Port Input/Output section for complete description. ADC2 Input Channel 3 (C8051F060/1/2/3 Only).
P1.4/AIN2.4	32	29	32	29	D I/O A In	Port 1.4. See Port Input/Output section for complete description. ADC2 Input Channel 4 (C8051F060/1/2/3 Only).
P1.5/AIN2.5	31	28	31	28	D I/O A In	Port 1.5. See Port Input/Output section for complete description. ADC2 Input Channel 5 (C8051F060/1/2/3 Only).
P1.6/AIN2.6	30	25	30	25	D I/O A In	Port 1.6. See Port Input/Output section for complete description. ADC2 Input Channel 6 (C8051F060/1/2/3 Only).
P1.7/AIN2.7	29	24	29	24	D I/O A In	Port 1.7. See Port Input/Output section for complete description. ADC2 Input Channel 7 (C8051F060/1/2/3 Only).
P2.0	46	43	46	43	D I/O	Port 2.0. See Port Input/Output section for complete description.
P2.1	45	42	45	42	D I/O	Port 2.1. See Port Input/Output section for complete description.
P2.2	44	41	44	41	D I/O	Port 2.2. See Port Input/Output section for complete description.
P2.3	43	38	43	38	D I/O	Port 2.3. See Port Input/Output section for complete description.
P2.4	42	37	42	37	D I/O	Port 2.4. See Port Input/Output section for complete description.

Figure 7.10. ADC2CN: ADC2 Control Register

SFR Page: 2
SFR Address: 0xE8 (bit addressable)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
AD2EN	AD2TM	AD2INT	AD2BUSY	AD2CM1	AD2CM0	AD2WINT	AD2LJST	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: AD2EN: ADC2 Enable Bit.
0: ADC2 Disabled. ADC2 is in low-power shutdown.
1: ADC2 Enabled. ADC2 is active and ready for data conversions.

Bit6: AD2TM: ADC2 Track Mode Bit.
0: Normal Track Mode: When ADC2 is enabled, tracking is continuous unless a conversion is in progress.
1: Low-power Track Mode: Tracking Defined by AD2CM2-0 bits (see below).

Bit5: AD2INT: ADC2 Conversion Complete Interrupt Flag.
0: ADC2 has not completed a data conversion since the last time AD2INT was cleared.
1: ADC2 has completed a data conversion.

Bit 4: AD2BUSY: ADC2 Busy Bit.
Read:
0: ADC2 conversion is complete or a conversion is not currently in progress. AD2INT is set to logic 1 on the falling edge of AD2BUSY.
1: ADC2 conversion is in progress.
Write:
0: No Effect.
1: Initiates ADC2 Conversion if AD2CM2-0 = 000b

Bits 3-2: AD2CM1-0: ADC2 Start of Conversion Mode Select.
When AD2TM = 0:
00: ADC2 conversion initiated on every write of ‘1’ to AD2BUSY.
01: ADC2 conversion initiated on overflow of Timer 3.
10: ADC2 conversion initiated on rising edge of external CNVSTR2 pin.
11: ADC2 conversion initiated on overflow of Timer 2.
When AD2TM = 1:
00: Tracking initiated on write of ‘1’ to AD2BUSY and lasts 3 SAR clocks, followed by conversion.
01: Tracking initiated on overflow of Timer 3 and lasts 3 SAR clocks, followed by conversion.
10: ADC2 tracks only when CNVSTR2 input is logic low; conversion starts on rising CNVSTR2 edge.
11: Tracking initiated on overflow of Timer 2 and lasts 3 SAR clocks, followed by conversion.

Bit 1: AD2WINT: ADC2 Window Compare Interrupt Flag.
0: ADC2 Window Comparison Data match has not occurred since this flag was last cleared.
1: ADC2 Window Comparison Data match has occurred.

Bit 0: AD2LJST: ADC2 Left Justify Select.
0: Data in ADC2H:ADC2L registers are right-justified.
1: Data in ADC2H:ADC2L registers are left-justified.

The temperature sensor connects to the highest order input of the ADC2 input multiplexer (see Section “7. 10-Bit ADC (ADC2, C8051F060/1/2/3)” on page 87). The TEMPE bit within REF2CN enables and disables the temperature sensor. While disabled, the temperature sensor defaults to a high impedance state, and any A/D measurements performed on the sensor while disabled result in meaningless data.

Figure 10.2. REF2CN: Reference Control Register 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	AD2VRS	TEMPE	BIASE	REFBE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xD1
SFR Page: 2

Bits7-4: UNUSED. Read = 0000b; Write = don’t care.
Bit3: AD2VRS: ADC2 Voltage Reference Select.
0: ADC2 voltage reference from VREF2 pin.
1: ADC2 voltage reference from AV+.
Bit2: TEMPE: Temperature Sensor Enable Bit.
0: Internal Temperature Sensor Off.
1: Internal Temperature Sensor On.
Bit1: BIASE: ADC/DAC Bias Generator Enable Bit. (Must be ‘1’ if using ADC2 or DACs).
0: Internal Bias Generator Off.
1: Internal Bias Generator On.
Bit0: REFBE: Internal Reference Buffer Enable Bit.
0: Internal Reference Buffer Off.
1: Internal Reference Buffer On. Internal voltage reference is driven on the VREF pin.

Table 10.1. Voltage Reference Electrical Characteristics

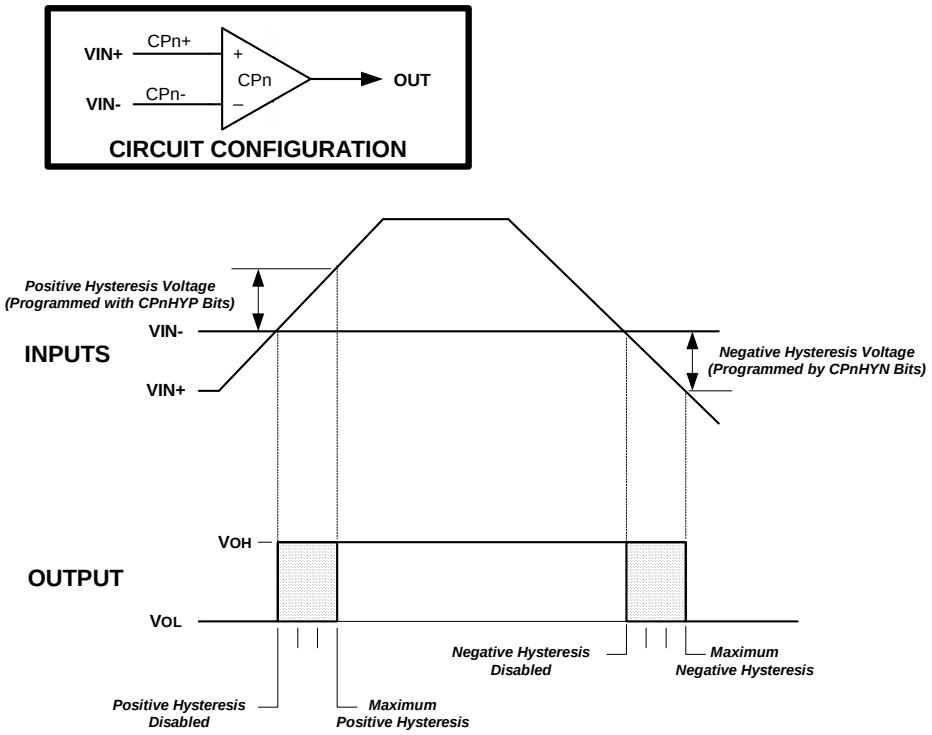
VDD = 3.0 V, AV+ = 3.0 V, -40 to +85 °C unless otherwise specified

Parameter	Conditions	Min	Typ	Max	Units
Internal Reference (REFBE = 1)					
Output Voltage	25 °C ambient	2.36	2.43	2.48	V
VREF Power Supply Current			50		µA
VREF Short-Circuit Current				30	mA
VREF Temperature Coefficient			15		ppm/°C
Load Regulation	Load = 0 to 200 µA to AGND		0.5		ppm/µA
VREF Turn-on Time 1	4.7 µF tantalum, 0.1 µF ceramic bypass		2		ms
VREF Turn-on Time 2	0.1 µF ceramic bypass		20		µs
VREF Turn-on Time 3	no bypass cap		10		µs
External Reference (REFBE = 0)					
Input Voltage Range		1.00		(AV+) - 0.3	V
Input Current			0	1	µA

complete electrical specifications for the Comparator are given in Table 12.1.

The Comparator response time may be configured in software using the CPnMD1-0 bits in register CPTnMD (see Figure 12.4). Selecting a longer response time reduces the amount of power consumed by the comparator. See Table 12.1 for complete timing and current consumption specifications.

Figure 12.2. Comparator Hysteresis Plot



The hysteresis of the Comparator is software-programmable via its Comparator Control register (CPTnCN). The user can program both the amount of hysteresis voltage (referred to the input voltage) and the positive and negative-going symmetry of this hysteresis around the threshold voltage.

The Comparator hysteresis is programmed using Bits3-0 in the Comparator Control Register CPTnCN (shown in Figure 12.3). The amount of negative hysteresis voltage is determined by the settings of the CPnHYN bits. As shown in Figure 12.2, the negative hysteresis can be programmed to three different settings, or negative hysteresis can be disabled. In a similar way, the amount of positive hysteresis is determined by the setting the CPnHYP bits.

Comparator interrupts can be generated on either rising-edge and falling-edge output transitions. (For Interrupt enable and priority control, see Section “13.3. Interrupt Handler” on page 151). The rising and/or falling -edge interrupts are enabled using the comparator’s Rising/Falling Edge Interrupt Enable Bits (CPnRIE and CPnFIE) in their respective Comparator Mode Selection Register (CPTnMD), shown in Figure 12.4. These bits allow the user to control which edge (or both) will cause a comparator interrupt. However, the comparator interrupt must also be enabled in the Extended Interrupt Enable Register (EIE1). The CPnFIF flag is set to logic 1 upon a Comparator falling-edge interrupt, and the CPnRIF flag is set to logic 1 upon the Comparator rising-edge interrupt. Once set, these bits remain set until cleared by software. The output state of a Comparator can be obtained at any time by reading the CPnOUT bit. A Comparator is enabled by setting its respective CPnEN bit to logic 1, and is disabled by clearing this bit to logic 0.Upon enabling a comparator, the output of the comparator is not immediately valid. Before using a comparator as an interrupt or reset source, software should wait for a minimum of the specified “Power-up time” as specified in Table 12.1, “Comparator Electrical Characteristics,” on page 122.

12.1. Comparator Inputs

The Port pins selected as comparator inputs should be configured as analog inputs in the Port 2 Input Configuration Register (for details on Port configuration, see Section “18.1.3. Configuring Port Pins as Digital Inputs” on page 207). The inputs for Comparator are on Port 2 as follows:

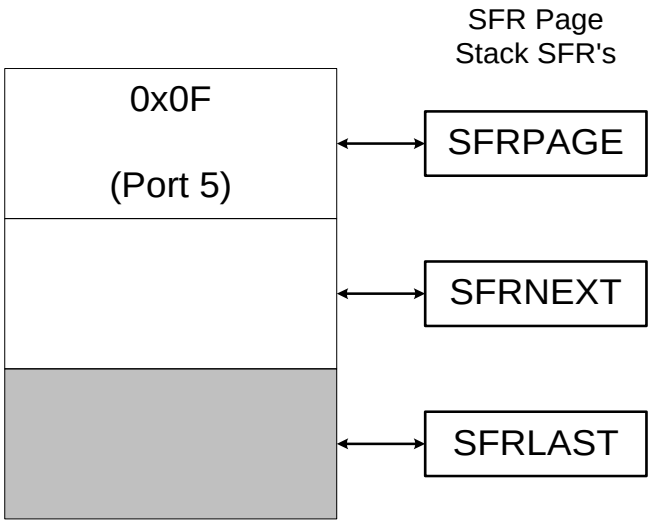
Comparator Input	Port PIN
CP0 +	P2.6
CP0 -	P2.7
CP1 +	P2.2
CP1 -	P2.3
CP2 +	P2.4
CP2 -	P2.5

13.2.6.3.SFR Page Stack Example

The following is an example that shows the operation of the SFR Page Stack during interrupts.

In this example, the SFR Page Control is left in the default enabled state (i.e., SFRPGEN = 1), and the CIP-51 is executing in-line code that is writing values to Port 5 (SFR “P5”, located at address 0xD8 on SFR Page 0x0F). The device is also using the Programmable Counter Array (PCA) and the 10-bit ADC (ADC2) window comparator to monitor a voltage. The PCA is timing a critical control function in its interrupt service routine (ISR), so its interrupt is enabled and is set to *high* priority. The ADC2 is monitoring a voltage that is less important, but to minimize the software overhead its window comparator is being used with an associated ISR that is set to *low* priority. At this point, the SFR page is set to access the Port 5 SFR (SFRPAGE = 0x0F). See Figure 13.4 below.

Figure 13.4. SFR Page Stack While Using SFR Page 0x0F To Access Port 5



17.6.2. Multiplexed Mode

17.6.2.1.16-bit MOVX: EMI0CF[4:2] = ‘001’, ‘010’, or ‘011’.

Figure 17.10. Multiplexed 16-bit MOVX Timing

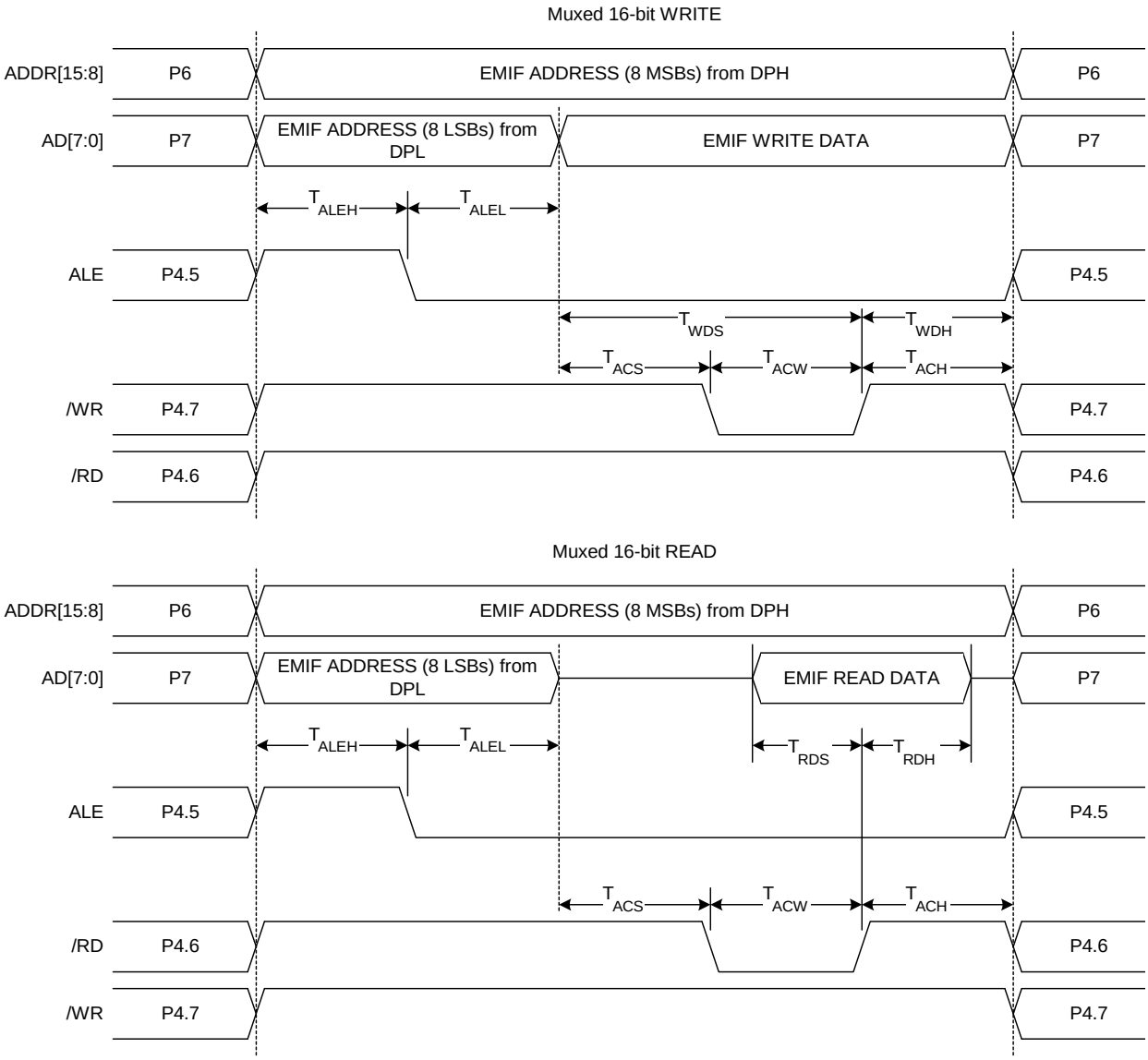


Figure 18.8. XBR3: Port I/O Crossbar Register 3

R	R	R	R/W	R/W	R/W	R/W	R/W	Reset Value
CTXOUT	-	-		CP2E	CNVST2E	T3EXE	T3E	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xE4
SFR Page: F

Bit7: CTXOUT: CAN Transmit Pin (CTX) Output Mode.
0: CTX pin output mode is configured as open-drain.
1: CTX pin output mode is configured as push-pull.
Bit6-4: Reserved
Bit3: CP2E: CP2 Output Enable Bit.
0: CP2 unavailable at Port pin.
1: CP2 routed to Port pin.
Bit2: CNVST2E: ADC2 External Convert Start Input Enable Bit.
0: CNVST2 for ADC2 unavailable at Port pin.
1: CNVST2 for ADC2 routed to Port pin.
Bit1: T3EXE: T3EX Input Enable Bit.
0: T3EX unavailable at Port pin.
1: T3EX routed to Port pin.
Bit0: T3E: T3 Input Enable Bit.
0: T3 unavailable at Port pin.
1: T3 routed to Port pin.

Figure 18.13. P1MDOUT: Port1 Output Mode Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xA5
SFR Page: F

Bits7-0: P1MDOUT.[7:0]: Port1 Output Mode Bits.
0: Port Pin output mode is configured as Open-Drain.
1: Port Pin output mode is configured as Push-Pull.

Note: SDA, SCL, and RX0 (when UART0 is in Mode 0) and RX1 (when UART1 is in Mode 0) are always configured as Open-Drain when they appear on Port pins.

Figure 18.14. P2: Port2 Data Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xA0
SFR Page: All Pages

Bits7-0: P2.[7:0]: Port2 Output Latch Bits.
(Write - Output appears on I/O pins per XBR0, XBR1, XBR2, and XBR3 Registers)
0: Logic Low Output.
1: Logic High Output (open if corresponding P2MDOUT.n bit = 0).
(Read - Regardless of XBR0, XBR1, XBR2, and XBR3 Register settings).
0: P2.n pin is logic low.
1: P2.n pin is logic high.

18.2. Ports 4 through 7 (C8051F060/2/4/6 only)

All Port pins on Ports 4 through 7 can be accessed as General-Purpose I/O (GPIO) pins by reading and writing the associated Port Data registers (See Figure 18.19, Figure 18.21, Figure 18.23, and Figure 18.25), a set of SFRs which are byte-addressable. Note that Port 4 has only three pins: P4.5, P4.6, and P4.7. Note also that the Port 4, 5, 6, and 7 registers are located on SFR Page F. The SFRPAGE register must be set to 0x0F to access these Port registers.

A Read of a Port Data register (or Port bit) will always return the logic state present at the pin itself, regardless of whether the Crossbar has allocated the pin for peripheral use or not. An exception to this occurs during the execution of a *read-modify-write* instruction (ANL, ORL, XRL, CPL, INC, DEC, DJNZ, JBC, CLR, SETB, and the bitwise MOV write operation). During the *read* cycle of the *read-modify-write* instruction, it is the contents of the Port Data register, not the state of the Port pins themselves, which is read.

18.2.1. Configuring Ports which are not Pinned Out

Although P3, P4, P5, P6, and P7 are not brought out to pins on the C8051F061/3/5/7 devices, the Port Data registers are still present and can be used by software. Because the digital input paths also remain active, it is recommended that these pins not be left in a ‘floating’ state in order to avoid unnecessary power dissipation arising from the inputs floating to non-valid logic levels. This condition can be prevented by any of the following:

1. Leave the weak pull-up devices enabled by setting WEAKPUD (XBR2.7) to a logic 0.
2. Configure the output modes of P3, P4, P5, P6, and P7 to “Push-Pull” by writing 0xFF to the associated output mode register (PnMDOUT).
3. Force the output states of P3, P4, P5, P6, and P7 to logic 0 by writing zeros to the Port Data registers: P3 = 0x00, P4 = 0x00, P5 = 0x00, P6= 0x00, and P7 = 0x00.

18.2.2. Configuring the Output Modes of the Port Pins

The output mode of each port pin can be configured to be either Open-Drain or Push-Pull. In the Push-Pull configuration, a logic 0 in the associated bit in the Port Data register will cause the Port pin to be driven to GND, and a logic 1 will cause the Port pin to be driven to VDD. In the Open-Drain configuration, a logic 0 in the associated bit in the Port Data register will cause the Port pin to be driven to GND, and a logic 1 will cause the Port pin to assume a high-impedance state. The Open-Drain configuration is useful to prevent contention between devices in systems where the Port pin participates in a shared interconnection in which multiple outputs are connected to the same physical wire.

The output modes of the Port pins on Ports 4 through 7 are determined by the bits in their respective PnMDOUT Output Mode Registers. Each bit in PnMDOUT controls the output mode of its corresponding port pin (see Figure 18.20, Figure 18.22, Figure 18.24, and Figure 18.26). For example, to place Port pin 5.3 in push-pull mode (digital output), set P5MDOUT.3 to logic 1. All port pins default to open-drain mode upon device reset.

18.2.3. Configuring Port Pins as Digital Inputs

A Port pin is configured as a digital input by setting its output mode to “Open-Drain” and writing a logic 1 to the associated bit in the Port Data register. For example, P7.7 is configured as a digital input by setting P7MDOUT.7 to a logic 0 and P7.7 to a logic 1.

18.2.4. Weak Pull-ups

By default, each Port pin has an internal weak pull-up device enabled which provides a resistive connection (about 100 kΩ) between the pin and VDD. The weak pull-up devices can be globally disabled by writ-

19.2. CAN Registers

CAN registers are classified as follows:

1. CAN Controller Protocol Registers: CAN control, interrupt, error control, bus status, test modes.
2. Message Object Interface Registers: Used to configure 32 Message Objects, send and receive data to and from Message Objects. The C8051 MCU accesses the CAN message RAM via the Message Object Interface Registers. Upon writing a message object number to an IF1 or IF2 Command Request Register, the contents of the associated Interface Registers (IF1 or IF2) will be transferred to or from the message object in CAN RAM.
3. Message Handler Registers: These read only registers are used to provide information to the CIP-51 MCU about the message objects (MSGVLD flags, Transmission Request Pending, New Data Flags) and Interrupts Pending (which Message Objects have caused an interrupt or status interrupt condition).
4. C8051 MCU Special Function Registers (SFR): Five registers located in the C8051 MCU memory map that allow direct access to certain CAN Controller Protocol Registers, and Indexed indirect access to all CAN registers.

19.2.1. CAN Controller Protocol Registers

The CAN Control Protocol Registers are used to configure the CAN controller, process interrupts, monitor bus status, and place the controller in test modes. The CAN controller protocol registers are accessible using C8051 MCU SFRs by an indexed method, and some can be accessed directly by addressing the SFRs in the C8051 SFR map for convenience.

The registers are: CAN Control Register (CAN0CN), CAN Status Register (CAN0STA), CAN Test Register (CAN0TST), Error Counter Register, Bit Timing Register, and the Baud Rate Prescaler (BRP) Extension Register. CAN0STA, CAN0CN, and CAN0TST can be accessed via C8051 MCU SFRs. All others are accessed indirectly using the CAN address indexed method via CAN0ADR, CAN0DATH, and CAN0DATL.

Please refer to the Bosch CAN User's Guide for information on the function and use of the CAN Control Protocol Registers.

19.2.2. Message Object Interface Registers

There are two sets of Message Object Interface Registers used to configure the 32 Message Objects that transmit and receive data to and from the CAN bus. Message objects can be configured for transmit or receive, and are assigned arbitration message identifiers for acceptance filtering by all CAN nodes.

Message Objects are stored in Message RAM, and are accessed and configured using the Message Object Interface Registers. These registers are accessed via the C8051's CAN0ADR and CAN0DAT registers using the indirect indexed address method.

Please refer to the Bosch CAN User's Guide for information on the function and use of the Message Object Interface Registers.

19.2.3. Message Handler Registers

The Message Handler Registers are *read only* registers. Their flags can be read via the indexed access method with CAN0ADR, CAN0DATH, and CAN0DATL. The message handler registers provide interrupt, error, transmit/receive requests, and new data information.

20.4. SMBus Special Function Registers

The SMBus0 serial interface is accessed and controlled through five SFRs: SMB0CN Control Register, SMB0CR Clock Rate Register, SMB0ADR Address Register, SMB0DAT Data Register and SMB0STA Status Register. The five special function registers related to the operation of the SMBus0 interface are described in the following sections.

20.4.1. Control Register

The SMBus0 Control register SMB0CN is used to configure and control the SMBus0 interface. All of the bits in the register can be read or written by software. Two of the control bits are also affected by the SMBus0 hardware. The Serial Interrupt flag (SI, SMB0CN.3) is set to logic 1 by the hardware when a valid serial interrupt condition occurs. It can only be cleared by software. The Stop flag (STO, SMB0CN.4) is set to logic 1 by software. It is cleared to logic 0 by hardware when a STOP condition is detected on the bus.

Setting the ENSMB flag to logic 1 enables the SMBus0 interface. Clearing the ENSMB flag to logic 0 disables the SMBus0 interface and removes it from the bus. Momentarily clearing the ENSMB flag and then resetting it to logic 1 will reset SMBus0 communication. However, ENSMB should not be used to temporarily remove a device from the bus since the bus state information will be lost. Instead, the Assert Acknowledge (AA) flag should be used to temporarily remove the device from the bus (see description of AA flag below).

Setting the Start flag (STA, SMB0CN.5) to logic 1 will put SMBus0 in a master mode. If the bus is free, SMBus0 will generate a START condition. If the bus is not free, SMBus0 waits for a STOP condition to free the bus and then generates a START condition after a 5 µs delay per the SMB0CR value (In accordance with the SMBus protocol, the SMBus0 interface also considers the bus free if the bus is idle for 50 µs and no STOP condition was recognized). If STA is set to logic 1 while SMBus0 is in master mode and one or more bytes have been transferred, a repeated START condition will be generated.

When the Stop flag (STO, SMB0CN.4) is set to logic 1 while the SMBus0 interface is in master mode, the interface generates a STOP condition. In a slave mode, the STO flag may be used to recover from an error condition. In this case, a STOP condition is not generated on the bus, but the SMBus hardware behaves as if a STOP condition has been received and enters the "not addressed" slave receiver mode. Note that this simulated STOP will not cause the bus to appear free to SMBus0. The bus will remain occupied until a STOP appears on the bus or a Bus Free Timeout occurs. Hardware automatically clears the STO flag to logic 0 when a STOP condition is detected on the bus.

The Serial Interrupt flag (SI, SMB0CN.3) is set to logic 1 by hardware when the SMBus0 interface enters one of 27 possible states. If interrupts are enabled for the SMBus0 interface, an interrupt request is generated when the SI flag is set. The SI flag must be cleared by software.

Important Note: If SI is set to logic 1 while the SCL line is low, the clock-low period of the serial clock will be stretched and the serial transfer is suspended until SI is cleared to logic 0. A high level on SCL is not affected by the setting of the SI flag.

The Assert Acknowledge flag (AA, SMB0CN.2) is used to set the level of the SDA line during the acknowledge clock cycle on the SCL line. Setting the AA flag to logic 1 will cause an ACK (low level on SDA) to be sent during the acknowledge cycle if the device has been addressed. Setting the AA flag to logic 0 will cause a NACK (high level on SDA) to be sent during acknowledge cycle. After the transmission of a byte in slave mode, the slave can be temporarily removed from the bus by clearing the AA flag. The slave's own address and general call address will be ignored. To resume operation on the bus, the AA flag must be reset to logic 1 to allow the slave's address to be recognized.

SMBus0 is operating in master mode.

Figure 20.11. SMB0ADR: SMBus0 Address Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
SLV6	SLV5	SLV4	SLV3	SLV2	SLV1	SLV0	GC	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xC3
SFR Page: 0

Bits7-1: SLV6-SLV0: SMBus0 Slave Address.
These bits are loaded with the 7-bit slave address to which SMBus0 will respond when operating as a slave transmitter or slave receiver. SLV6 is the most significant bit of the address and corresponds to the first bit of the address byte received.

Bit0: GC: General Call Address Enable.
This bit is used to enable general call address (0x00) recognition.
0: General call address is ignored.
1: General call address is recognized.

20.4.5. Status Register

The SMB0STA Status register holds an 8-bit status code indicating the current state of the SMBus0 interface. There are 28 possible SMBus0 states, each with a corresponding unique status code. The five most significant bits of the status code vary while the three least-significant bits of a valid status code are fixed at zero when SI = ‘1’. Therefore, all possible status codes are multiples of eight. This facilitates the use of status codes in software as an index used to branch to appropriate service routines (allowing 8 bytes of code to service the state or jump to a more extensive service routine).

For the purposes of user software, the contents of the SMB0STA register is only defined when the SI flag is logic 1. Software should never write to the SMB0STA register; doing so will yield indeterminate results. The

Table 20.1. SMB0STA Status Codes and States

	0x60	Own slave address + W received. ACK transmitted.	Wait for data.
	0x68	Arbitration lost in sending SLA + R/W as master. Own address + W received. ACK transmitted.	Save current data for retry when bus is free. Wait for data.
	0x70	General call address received. ACK transmitted.	Wait for data.
	0x78	Arbitration lost in sending SLA + R/W as master. General call address received. ACK transmitted.	Save current data for retry when bus is free.
	0x80	Data byte received. ACK transmitted.	Read SMB0DAT. Wait for next byte or STOP.
	0x88	Data byte received. NACK transmitted.	Set STO to reset SMBus.
	0x90	Data byte received after general call address. ACK transmitted.	Read SMB0DAT. Wait for next byte or STOP.
	0x98	Data byte received after general call address. NACK transmitted.	Set STO to reset SMBus.
	0xA0	STOP or repeated START received.	No action necessary.
	0xA8	Own address + R received. ACK transmitted.	Load SMB0DAT with data to transmit.
	0xB0	Arbitration lost in transmitting SLA + R/W as master. Own address + R received. ACK transmitted.	Save current data for retry when bus is free. Load SMB0DAT with data to transmit.
	0xB8	Data byte transmitted. ACK received.	Load SMB0DAT with data to transmit.
	0xC0	Data byte transmitted. NACK received.	Wait for STOP.
	0xC8	Last data byte transmitted (AA=0). ACK received.	Set STO to reset SMBus.
	0xD0	SCL Clock High Timer per SMB0CR timed out	Set STO to reset SMBus.
	0x00	Bus Error (illegal START or STOP)	Set STO to reset SMBus.
	0xF8	Idle	State does not set SI.

Figure 21.9. SPI0CN: SPI0 Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	Reset Value
SPIF	WCOL	MODF	RXOVRN	NSSMD1	NSSMD0	TXBMT	SPIEN	00000110
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xF8
SFR Page: 0

Bit 7: SPIF: SPI0 Interrupt Flag.
This bit is set to logic 1 by hardware at the end of a data transfer. If interrupts are enabled, setting this bit causes the CPU to vector to the SPI0 interrupt service routine. This bit is not automatically cleared by hardware. It must be cleared by software.

Bit 6: WCOL: Write Collision Flag.
This bit is set to logic 1 by hardware (and generates a SPI0 interrupt) to indicate a write to the SPI0 data register was attempted while a data transfer was in progress. It must be cleared by software.

Bit 5: MODF: Mode Fault Flag.
This bit is set to logic 1 by hardware (and generates a SPI0 interrupt) when a master mode collision is detected (NSS is low, MSTEN = 1, and NSSMD[1:0] = 01). This bit is not automatically cleared by hardware. It must be cleared by software.

Bit 4: RXOVRN: Receive Overrun Flag (Slave Mode only).
This bit is set to logic 1 by hardware (and generates a SPI0 interrupt) when the receive buffer still holds unread data from a previous transfer and the last bit of the current transfer is shifted into the SPI0 shift register. This bit is not automatically cleared by hardware. It must be cleared by software.

Bits 3-2: NSSMD1-NSSMD0: Slave Select Mode.
Selects between the following NSS operation modes:
(See Section “21.2. SPI0 Master Mode Operation” on page 253 and Section “21.3. SPI0 Slave Mode Operation” on page 255).
00: 3-Wire Slave or 3-wire Master Mode. NSS signal is not routed to a port pin.
01: 4-Wire Slave or Multi-Master Mode (Default). NSS is always an input to the device.
1x: 4-Wire Single-Master Mode. NSS signal is mapped as an output from the device and will assume the value of NSSMD0.

Bit 1: TXBMT: Transmit Buffer Empty.
This bit will be set to logic 0 when new data has been written to the transmit buffer. When data in the transmit buffer is transferred to the SPI shift register, this bit will be set to logic 1, indicating that it is safe to write a new byte to the transmit buffer.

Bit 0: SPIEN: SPI0 Enable.
This bit enables/disables the SPI.
0: SPI disabled.
1: SPI enabled.

Table 21.1. SPI Slave Timing Parameters

Parameter	Description	Min	Max	Units
Master Mode Timing† (See Figure 21.12 and Figure 21.13)				
T_{MCKH}	SCK High Time	$1*T_{SYSCLK}$		ns
T_{MCKL}	SCK Low Time	$1*T_{SYSCLK}$		ns
T_{MIS}	MISO Valid to SCK Shift Edge	$1*T_{SYSCLK}$ + 20		ns
T_{MIH}	SCK Shift Edge to MISO Change	0		ns
Slave Mode Timing† (See Figure 21.14 and Figure 21.15)				
T_{SE}	NSS Falling to First SCK Edge	$2*T_{SYSCLK}$		ns
T_{SD}	Last SCK Edge to NSS Rising	$2*T_{SYSCLK}$		ns
T_{SEZ}	NSS Falling to MISO Valid		$4*T_{SYSCLK}$	ns
T_{SDZ}	NSS Rising to MISO High-Z		$4*T_{SYSCLK}$	ns
T_{CKH}	SCK High Time	$5*T_{SYSCLK}$		ns
T_{CKL}	SCK Low Time	$5*T_{SYSCLK}$		ns
T_{SIS}	MOSI Valid to SCK Sample Edge	$2*T_{SYSCLK}$		ns
T_{SIH}	SCK Sample Edge to MOSI Change	$2*T_{SYSCLK}$		ns
T_{SOH}	SCK Shift Edge to MISO Change		$4*T_{SYSCLK}$	ns
T_{SLH}	Last SCK Edge to MISO Change (CKPHA = 1 ONLY)	$6*T_{SYSCLK}$	$8*T_{SYSCLK}$	ns
†T_{SYSCLK} is equal to one period of the device system clock (SYSCLK).				

25.2.6. 16-Bit Pulse Width Modulator Mode

Each PCA0 module may also be operated in 16-Bit PWM mode. In this mode, the 16-bit capture/compare module defines the number of PCA0 clocks for the low time of the PWM signal. When the PCA0 counter matches the module contents, the output on CEXn is asserted high; when the counter overflows, CEXn is asserted low. To output a varying duty cycle, new value writes should be synchronized with PCA0 CCFn match interrupts. 16-Bit PWM Mode is enabled by setting the ECOMn, PWMn, and PWM16n bits in the PCA0CPMn register. For a varying duty cycle, CCFn should also be set to logic 1 to enable match interrupts. The duty cycle for 16-Bit PWM Mode is given by Equation 25.3.

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’.

Equation 25.3. 16-Bit PWM Duty Cycle

$$DutyCycle = \frac{(65536 - PCA0CPn)}{65536}$$

Figure 25.9. PCA 16-Bit PWM Mode