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	EBI/EMI, SMBus (2-Wire/I²C), CANbus, LINbus, SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	40
Program Memory Size	96KB (96K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8.25K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.25V
Data Converters	A/D 32x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TQFP
Supplier Device Package	48-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f584-iqr

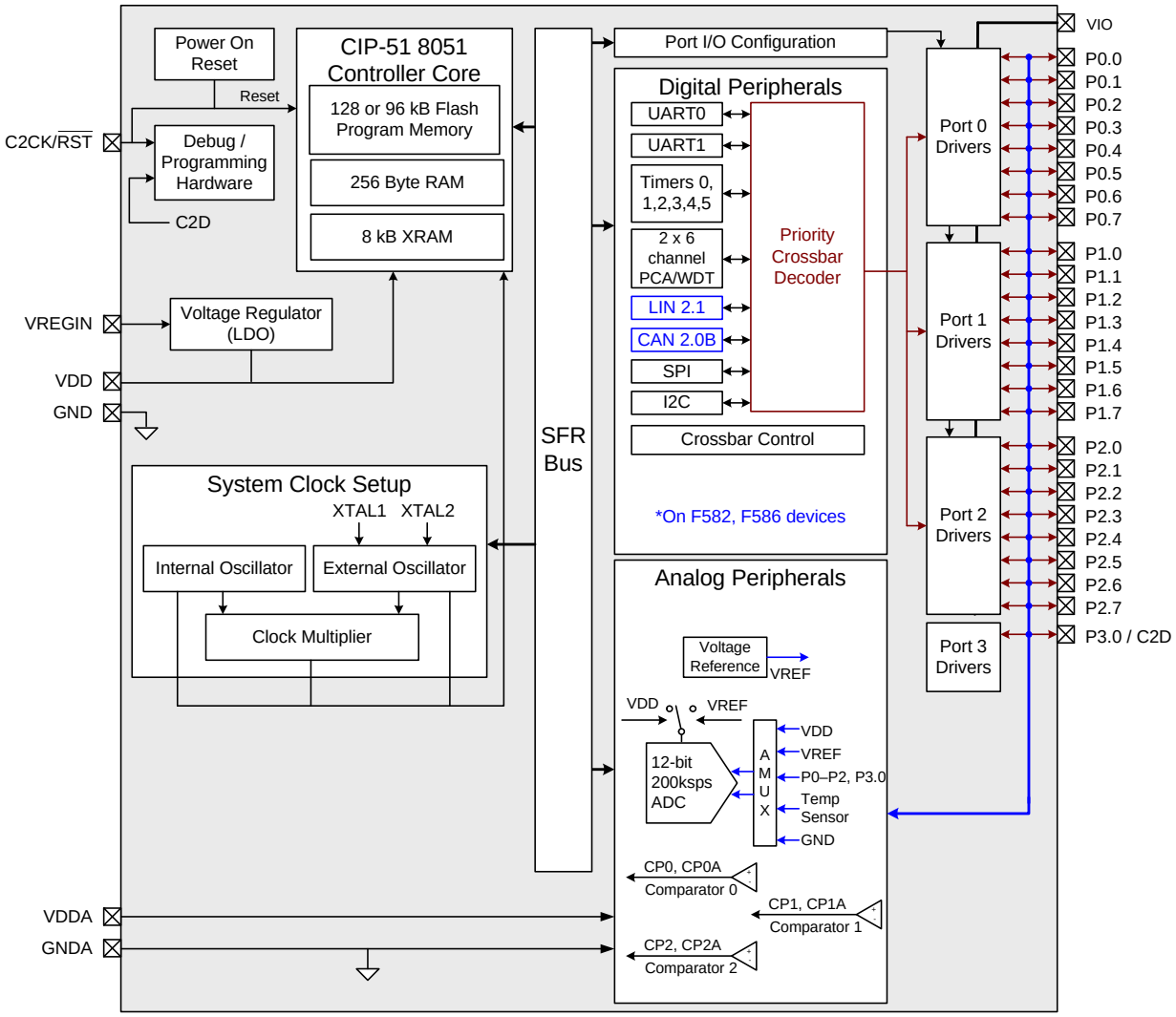

Figure 1.3. C8051F582/3/6/7 Block Diagram

4. Package Specifications

4.1. QFP-48 Package Specifications

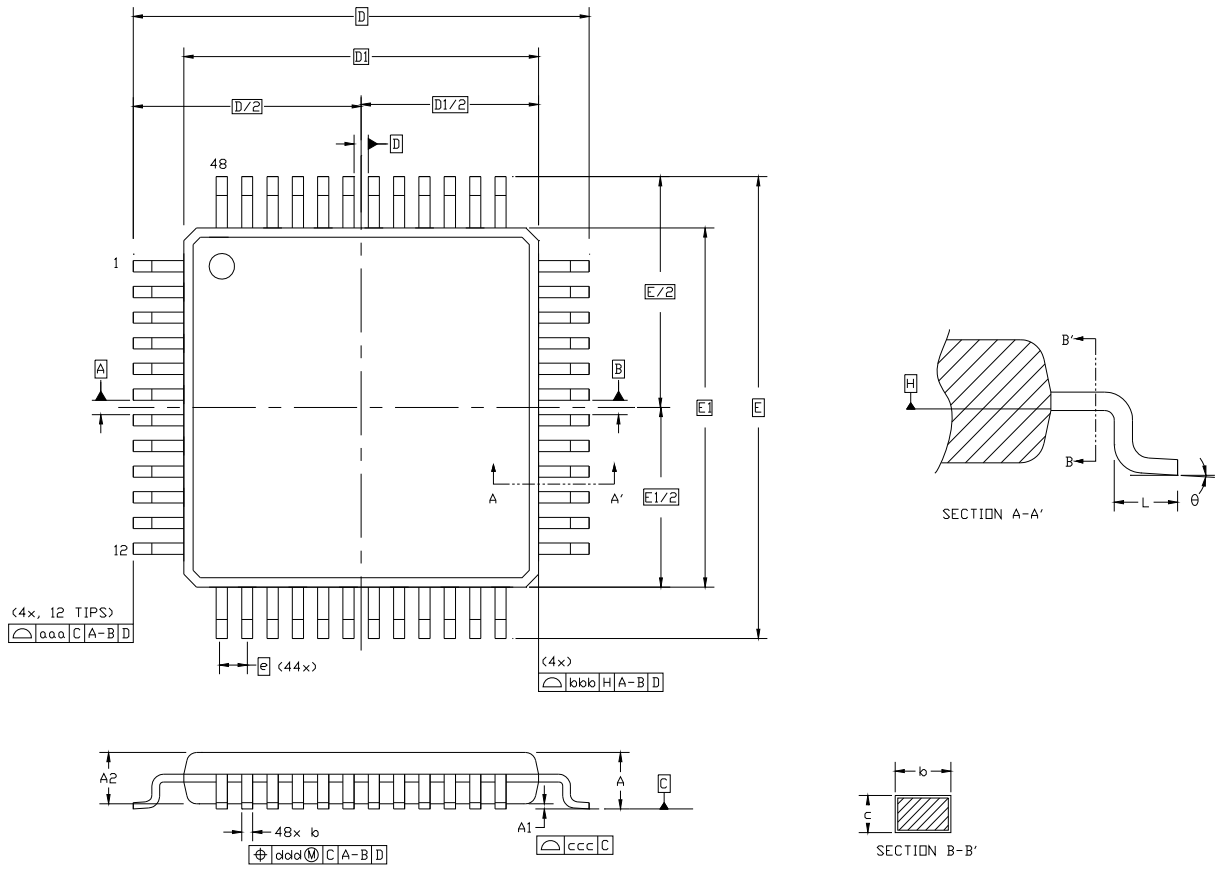

Figure 4.1. QFP-48 Package Drawing

Table 4.1. QFP-48 Package Dimensions

Dimension	Min	Typ	Max
A	—	—	1.20
A1	0.05	—	0.15
A2	0.95	1.00	1.05
b	0.17	0.22	0.27
c	0.09	—	0.20
D	9.00 BSC.		
D1	7.00 BSC.		
e	0.50 BSC.		

Dimension	Min	Typ	Max
E	9.00 BSC.		
E1	7.00 BSC.		
L	0.45	0.60	0.75
aaa	0.20		
bbb	0.20		
ccc	0.08		
ddd	0.08		
θ	0°	3.5°	7°

Notes:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to the JEDEC outline MS-026, variation ABC.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

5.2. Electrical Characteristics

Table 5.2. Global Electrical Characteristics

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

Parameter	Conditions	Min	Typ	Max	Units
Supply Input Voltage (V_{REGIN})		1.8	—	5.25	V
Digital Supply Voltage (V_{DD})	System Clock ≤ 25 MHz	V_{RST}[1]	—	2.75	V
	System Clock > 25 MHz	2		2.75	
Analog Supply Voltage (VDDA)	System Clock ≤ 25 MHz	V_{RST}[1]	—	2.75	V
(Must be connected to V_{DD})	System Clock > 25 MHz	2		2.75	
Digital Supply RAM Data Retention Voltage		—	1.5	—	
Port I/O Supply Voltage (V_{IO})	Normal Operation	1.8[2]	—	5.25	V
SYSCLK (System Clock)[3]		0	—	50	MHz
T_{SYSH} (SYSCLK High Time)		9	—	—	ns
T_{SYSL} (SYSCLK Low Time)		9	—	—	ns
Specified Operating Temperature Range		–40	—	+125	°C
Digital Supply Current—CPU Active (Normal Mode, fetching instructions from Flash)					
I_{DD}[4]	V_{DD} = 2.1 V, F = 200 kHz	—	150	—	µA
	V_{DD} = 2.1 V, F = 1.5 MHz	—	650	—	µA
	V_{DD} = 2.1 V, F = 25 MHz	—	8.5	11	mA
	V_{DD} = 2.1 V, F = 50 MHz	—	15	21	mA

Notes:

1. Given in Table 5.4 on page 48.
2. V_{IO} should not be lower than the V_{DD} voltage.
3. SYSCLK must be at least 32 kHz to enable debugging.
4. Based on device characterization data; Not production tested. Does not include oscillator supply current.
5. IDD can be estimated for frequencies ≤ 15 MHz by simply multiplying the frequency of interest by the frequency sensitivity number for that range. When using these numbers to estimate I_{DD} for >15 MHz, the estimate should be the current at 50 MHz minus the difference in current indicated by the frequency sensitivity number. For example: V_{DD} = 2.6 V; F = 20 MHz, I_{DD} = 21 mA - (50 MHz - 20 MHz) * 0.46 mA/MHz = 7.2 mA.
6. Idle IDD can be estimated for frequencies ≤ 1 MHz by simply multiplying the frequency of interest by the frequency sensitivity number for that range. When using these numbers to estimate Idle I_{DD} for >1 MHz, the estimate should be the current at 50 MHz minus the difference in current indicated by the frequency sensitivity number.
   For example: V_{DD} = 2.6 V; F = 5 MHz, Idle I_{DD} = 19 mA – (50 MHz – 5 MHz) x 0.38 mA/MHz = 1.9 mA.

Post-Tracking Mode is selected when AD0TM is set to 01b. A programmable tracking time based on AD0TK is started immediately following the convert start signal. Conversions are started after the programmed tracking time ends. After a conversion is complete, ADC0 does not track the input. Rather, the sampling capacitor remains disconnected from the input making the input pin high-impedance until the next convert start signal.

Dual-Tracking Mode is selected when AD0TM is set to 11b. A programmable tracking time based on AD0TK is started immediately following the convert start signal. Conversions are started after the programmed tracking time ends. After a conversion is complete, ADC0 tracks continuously until the next conversion is started.

Depending on the output connected to the ADC input, additional tracking time, more than is specified in Table 5.10, may be required after changing MUX settings. See the settling time requirements described in Section “6.2.1. Settling Time Requirements” on page 59.

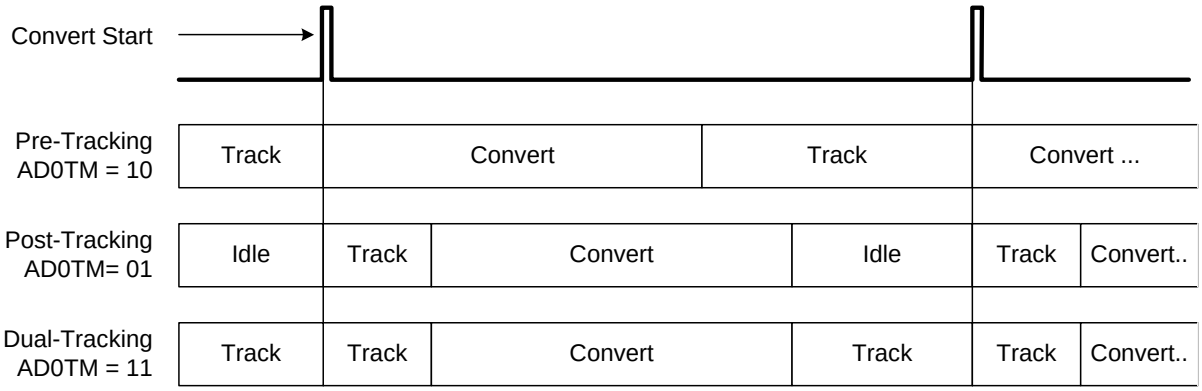

Figure 6.2. ADC0 Tracking Modes

6.1.3. Timing

ADC0 has a maximum conversion speed specified in Table 5.10. ADC0 is clocked from the ADC0 Subsystem Clock (FCLK). The source of FCLK is selected based on the BURSTEN bit. When BURSTEN is logic 0, FCLK is derived from the current system clock. When BURSTEN is logic 1, FCLK is derived from the Burst Mode Oscillator, an independent clock source with a maximum frequency of 25 MHz.

When ADC0 is performing a conversion, it requires a clock source that is typically slower than FCLK. The ADC0 SAR conversion clock (SAR clock) is a divided version of FCLK. The divide ratio can be configured using the AD0SC bits in the ADC0CF register. The maximum SAR clock frequency is listed in Table 5.10.

ADC0 can be in one of three states at any given time: tracking, converting, or idle. Tracking time depends on the tracking mode selected. For Pre-Tracking Mode, tracking is managed by software and ADC0 starts conversions immediately following the convert start signal. For Post-Tracking and Dual-Tracking Modes, the tracking time after the convert start signal is equal to the value determined by the AD0TK bits plus 2 FCLK cycles. Tracking is immediately followed by a conversion. The ADC0 conversion time is always 13 SAR clock cycles plus an additional 2 FCLK cycles to start and complete a conversion. Figure 6.4 shows timing diagrams for a conversion in Pre-Tracking Mode and tracking plus conversion in Post-Tracking or Dual-Tracking Mode. In this example, repeat count is set to one.

been accumulated. Similarly, the Window Comparator will not compare the result to the greater-than and less-than registers until “repeat count” conversions have been accumulated.

Note: When using Burst Mode, care must be taken to issue a convert start signal no faster than once every four SYSCLK periods. This includes external convert start signals.

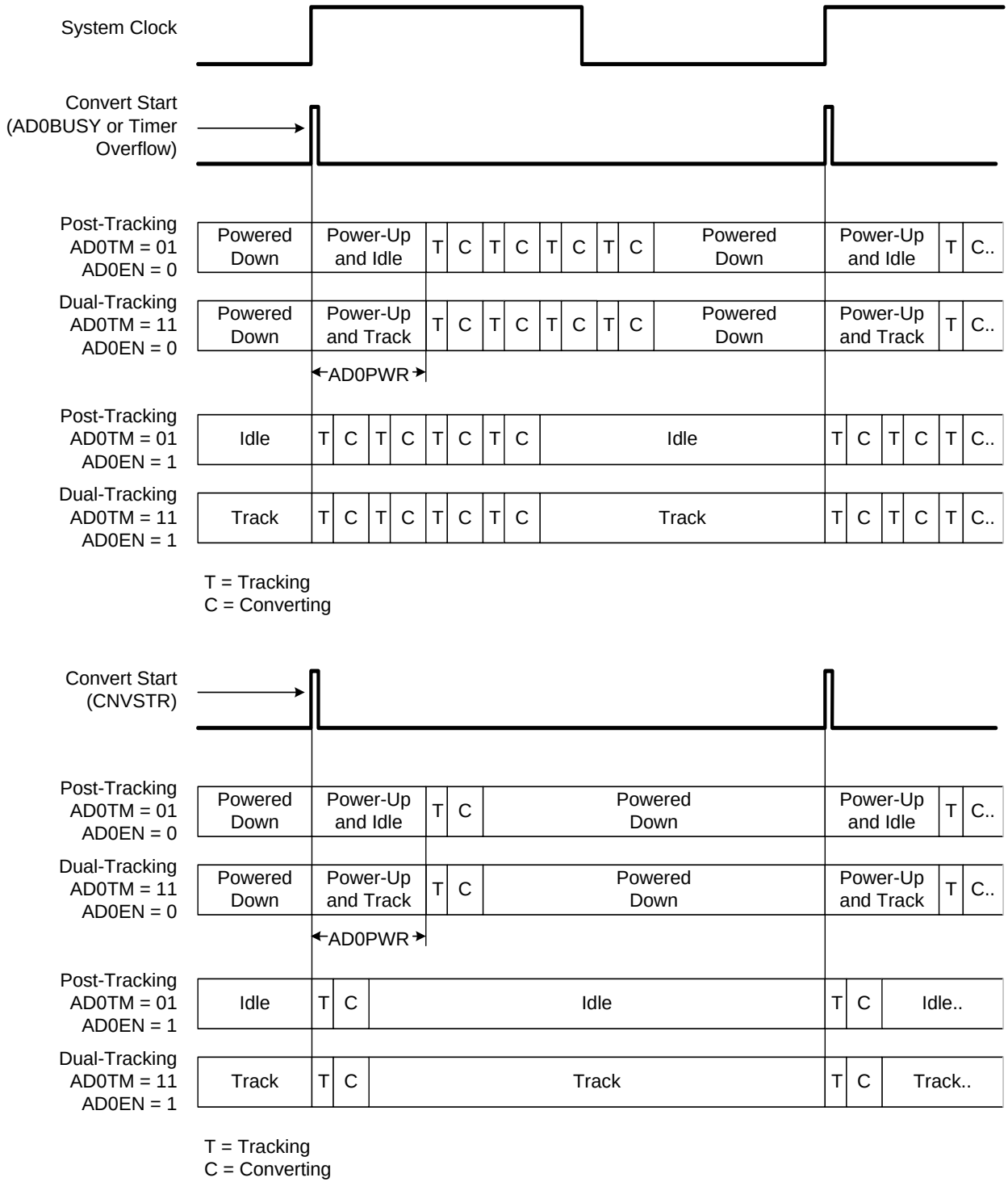

Figure 6.4. 12-Bit ADC Burst Mode Example With Repeat Count Set to 4

SFR Definition 6.8. ADC0TK: ADC0 Tracking Mode Select

Bit	7	6	5	4	3	2	1	0
Name	AD0PWR[3:0]				AD0TM[1:0]		AD0TK[1:0]	
Type	R/W				R/W		R/W	
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xBA; SFR Page = 0x00;

Bit	Name	Function
7:4	AD0PWR[3:0]	**ADC0 Burst Power-Up Time.** For BURSTEN = 0: ADC0 Power state controlled by AD0EN For BURSTEN = 1, AD0EN = 1: ADC0 remains enabled and does not enter the very low power state For BURSTEN = 1, AD0EN = 0: ADC0 enters the very low power state and is enabled after each convert start signal. The Power-Up time is programmed according the following equation: $AD0PWR = \frac{Tstartup}{200ns} - 1$ or $Tstartup = (AD0PWR + 1)200ns$
3:2	AD0TM[1:0]	**ADC0 Tracking Mode Enable Select Bits.** 00: Reserved. 01: ADC0 is configured to Post-Tracking Mode. 10: ADC0 is configured to Pre-Tracking Mode. 11: ADC0 is configured to Dual Tracking Mode.
1:0	AD0TK[1:0]	**ADC0 Post-Track Time.** 00: Post-Tracking time is equal to 2 SAR clock cycles + 2 FCLK cycles. 01: Post-Tracking time is equal to 4 SAR clock cycles + 2 FCLK cycles. 10: Post-Tracking time is equal to 8 SAR clock cycles + 2 FCLK cycles. 11: Post-Tracking time is equal to 16 SAR clock cycles + 2 FCLK cycles.

6.4. Programmable Window Detector

The ADC Programmable Window Detector continuously compares the ADC0 output registers to user-programmed limits, and notifies the system when a desired condition is detected. This is especially effective in an interrupt-driven system, saving code space and CPU bandwidth while delivering faster system response times. The window detector interrupt flag (AD0WINT in register ADC0CN) can also be used in polled mode. The ADC0 Greater-Than (ADC0GTH, ADC0GTL) and Less-Than (ADC0LTH, ADC0LTL) registers hold the comparison values. The window detector flag can be programmed to indicate when measured data is inside or outside of the user-programmed limits, depending on the contents of the ADC0 Less-Than and ADC0 Greater-Than registers.

8. Voltage Reference

The Voltage reference multiplexer on the C8051F58x/F59x devices is configurable to use an externally connected voltage reference, the on-chip reference voltage generator routed to the VREF pin, or the V_{DD} power supply voltage (see Figure 8.1). The REFSL bit in the Reference Control register (REF0CN, SFR Definition 8.1) selects the reference source for the ADC. For an external source or the on-chip reference, REFSL should be set to 0 to select the VREF pin. To use V_{DD} as the reference source, REFSL should be set to 1.

The BIASE bit enables the internal voltage bias generator, which is used by the ADC, Temperature Sensor, and internal oscillator. This bias is automatically enabled when any peripheral which requires it is enabled, and it does not need to be enabled manually. The bias generator may be enabled manually by writing a 1 to the BIASE bit in register REF0CN. The electrical specifications for the voltage reference circuit are given in Table 5.12.

The on-chip voltage reference circuit consists of a temperature stable bandgap voltage reference generator and a gain-of-two output buffer amplifier. The output voltage is selectable between 1.5 V and 2.25 V. The on-chip voltage reference can be driven on the VREF pin by setting the REFBE bit in register REF0CN to a 1. The maximum load seen by the VREF pin must be less than 200 µA to GND. Bypass capacitors of 0.1 µF and 4.7 µF are recommended from the VREF pin to GND. If the on-chip reference is not used, the REFBE bit should be cleared to 0. Electrical specifications for the on-chip voltage reference are given in Table 5.12.

Important Note about the VREF Pin: When using either an external voltage reference or the on-chip reference circuitry, the VREF pin should be configured as an analog pin and skipped by the Digital Crossbar. Refer to Section “20. Port Input/Output” on page 188 for the location of the VREF pin, as well as details of how to configure the pin in analog mode and to be skipped by the crossbar.

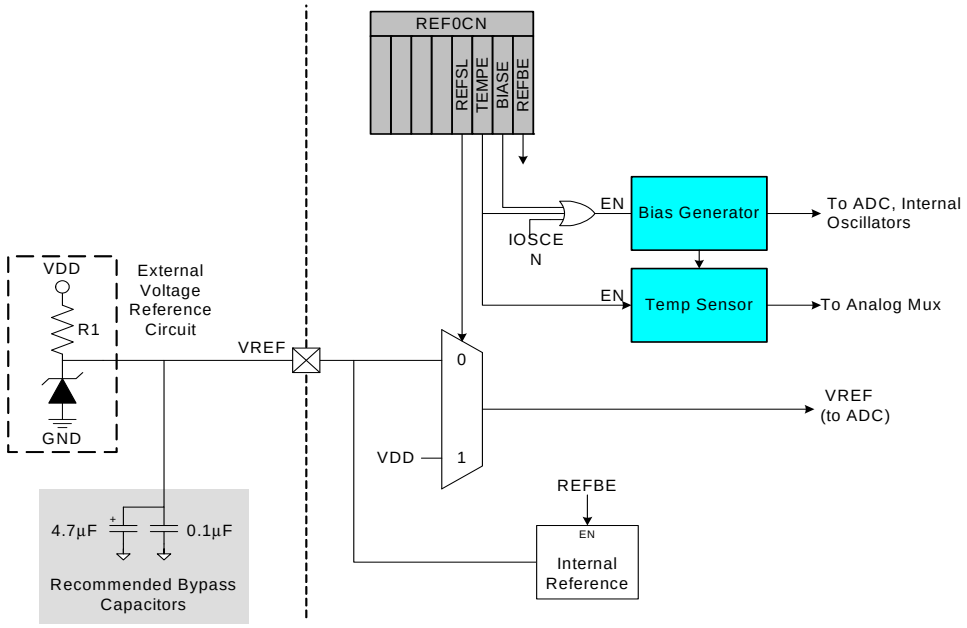

Figure 8.1. Voltage Reference Functional Block Diagram

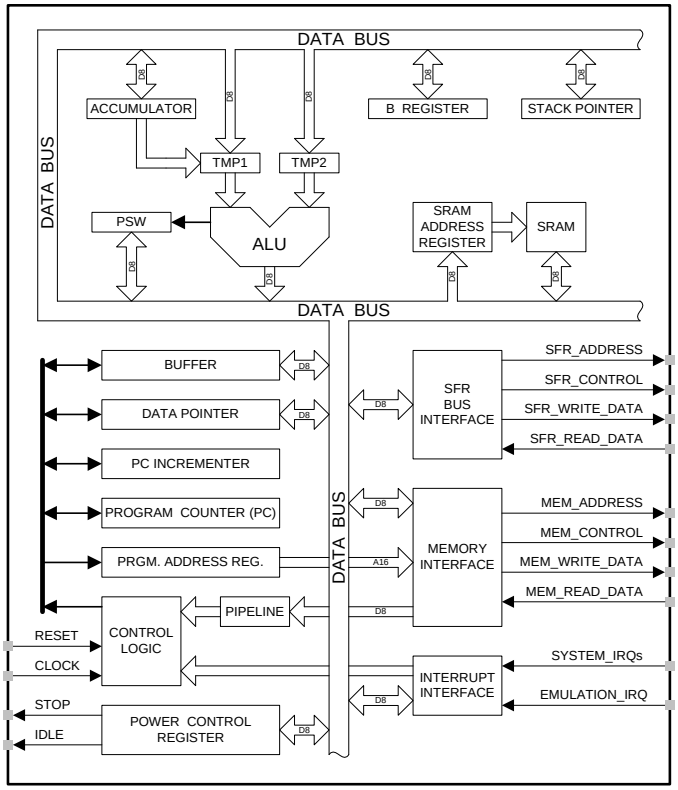

Figure 11.1. CIP-51 Block Diagram

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 “C2 Interface” on page 351.

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.

SFR Definition 11.3. SP: Stack Pointer

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

SFR Address = 0x81; SFR Page = All Pages

Bit	Name	Function
7:0	SP[7:0]	**Stack Pointer.** The Stack Pointer holds the location of the top of the stack. The stack pointer is incremented before every PUSH operation. The SP register defaults to 0x07 after reset.

SFR Definition 11.4. ACC: Accumulator

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

SFR Address = 0xE0; SFR Page = All Pages; Bit-Addressable

Bit	Name	Function
7:0	ACC[7:0]	**Accumulator.** This register is the accumulator for arithmetic operations.

SFR Definition 11.5. B: B Register

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

SFR Address = 0xF0; SFR Page = All Pages; Bit-Addressable

Bit	Name	Function
7:0	B[7:0]	**B Register.** This register serves as a second accumulator for certain arithmetic operations.

Table 13.1. Special Function Register (SFR) Memory Map for Pages 0x00, 0x10, and 0x0F

90	00 10 0F	P1 (All Pages)	TMR3CN TMR5CN	TMR3RLL TMR5CAPL	TMR3RLH TMR5CAPH	TMR3L TMR5L	TMR3H TMR5H	TMR5CF	CLKMUL
88	00 10 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	00 10 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)

15.1.2. Flash Lock and Key Functions

Flash writes and erases by user software are protected with a lock and key function. The Flash Lock and Key Register (FLKEY) must be written with the correct key codes, in sequence, before Flash operations may be performed. The key codes are: 0xA5, 0xF1. The timing does not matter, but the codes must be written in order. If the key codes are written out of order, or the wrong codes are written, Flash writes and erases will be disabled until the next system reset. Flash writes and erases will also be disabled if a Flash write or erase is attempted before the key codes have been written properly. The Flash lock resets after each write or erase; the key codes must be written again before a following Flash operation can be performed. The FLKEY register is detailed in SFR Definition 15.2.

15.1.3. Flash Erase Procedure

The Flash memory can be programmed by software using the MOVX write instruction with the address and data byte to be programmed provided as normal operands. Before writing to Flash memory using MOVX, Flash write operations must be enabled by doing the following: (1) setting the PSWE Program Store Write Enable bit (PSCTL.0) to logic 1 (this directs the MOVX writes to target Flash memory); and (2) Writing the Flash key codes in sequence to the Flash Lock register (FLKEY). The PSWE bit remains set until cleared by software.

A write to Flash memory can clear bits to logic 0 but cannot set them; only an erase operation can set bits to logic 1 in Flash. **A byte location to be programmed should be erased before a new value is written.** The Flash memory is organized in 512-byte pages. The erase operation applies to an entire page (setting all bytes in the page to 0xFF). To erase an entire 512-byte page, perform the following steps:

1. Disable interrupts (recommended).
2. If erasing a page in Banks 1, 2, or 3, set the COBANK[1:0] bits (register PSBANK) for the appropriate bank.
3. Set the FLEWT bit (register FLSCL).
4. Set the PSEE bit (register PSCTL).
5. Set the PSWE bit (register PSCTL).
6. Write the first key code to FLKEY: 0xA5.
7. Write the second key code to FLKEY: 0xF1.
8. Using the MOVX instruction, write a data byte to any location within the 512-byte page to be erased.
9. Clear the PSWE and PSEE bits.

15.1.4. Flash Write Procedure

Flash bytes are programmed by software with the following sequence:

1. Disable interrupts (recommended).
2. If writing to an address in Banks 1, 2, or 3, set the COBANK[1:0] (register PSBANK) for the appropriate bank.
3. Erase the 512-byte Flash page containing the target location, as described in Section 15.1.3.
4. Set the FLEWT bit (register FLSCL).
5. Set the PSWE bit (register PSCTL).
6. Clear the PSEE bit (register PSCTL).
7. Write the first key code to FLKEY: 0xA5.
8. Write the second key code to FLKEY: 0xF1.
9. Using the MOVX instruction, write a single data byte to the desired location within the 512-byte sector.
10. Clear the PSWE bit.

18.6.2.3. 8-bit MOVX with Bank Select: EMI0CF[4:2] = 010

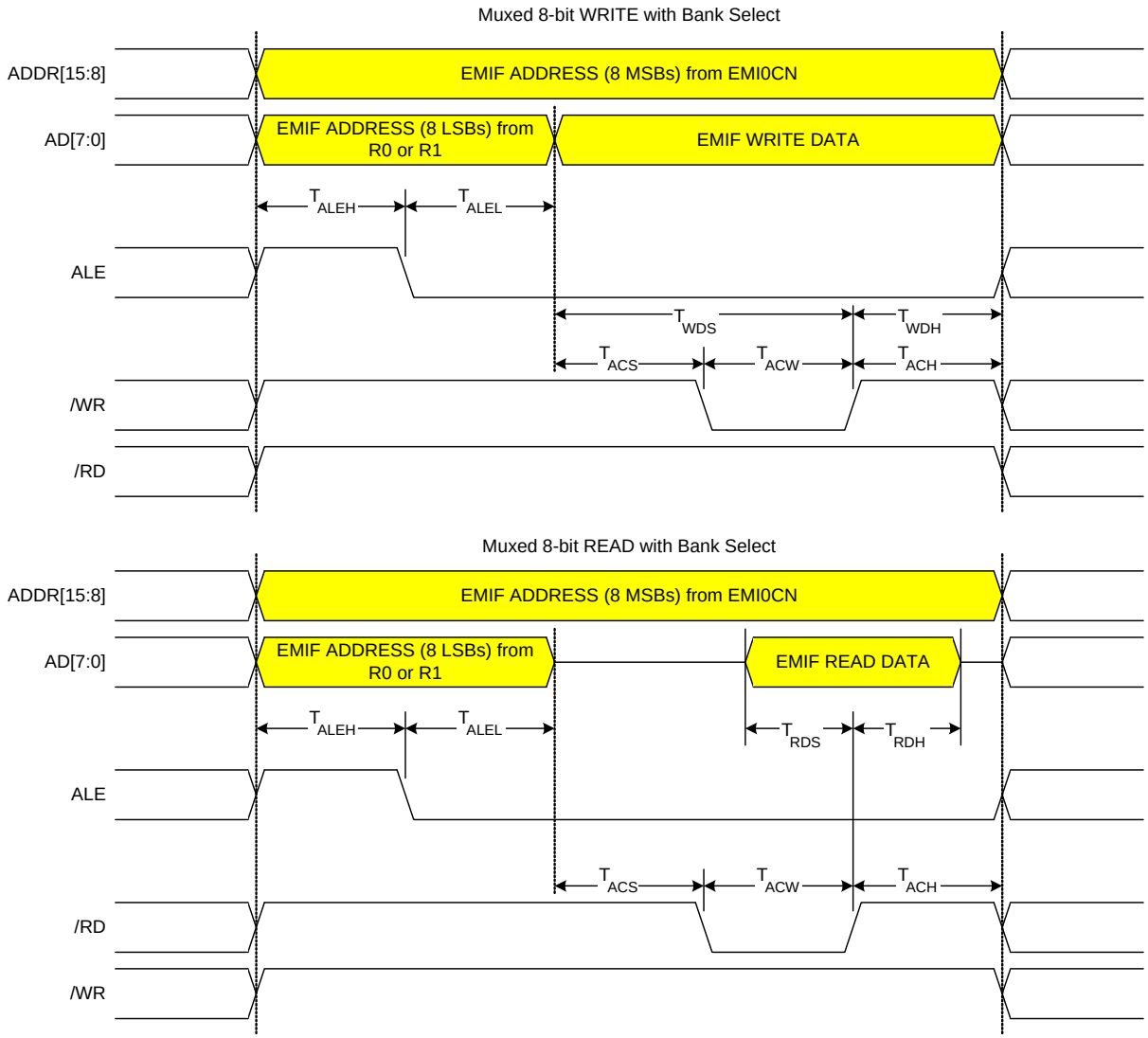

Figure 18.9. Multiplexed 8-bit MOVX with Bank Select Timing

SFR Definition 19.6. OSCXCN: External Oscillator Control

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

SFR Address = 0x9F; SFR Page = 0x0F;

Bit	Name	Function
7	XTLVLD	**Crystal Oscillator Valid Flag.** (Read only when XOSCMD = 11x.) 0: Crystal Oscillator is unused or not yet stable. 1: Crystal Oscillator is running and stable.
6:4	XOSCMD[2:0]	**External Oscillator Mode Select.** 00x: External Oscillator circuit off. 010: External CMOS Clock Mode. 011: External CMOS Clock Mode with divide by 2 stage. 100: RC Oscillator Mode. 101: Capacitor Oscillator Mode. 110: Crystal Oscillator Mode. 111: Crystal Oscillator Mode with divide by 2 stage.
3	Unused	Read = 0b; Write =0b
2:0	XFCN[2:0]	**External Oscillator Frequency Control Bits.** Set according to the desired frequency for Crystal or RC mode. Set according to the desired K Factor for C mode.

XFCN	Crystal Mode	RC Mode	C Mode
000	f ≤ 32 kHz	f ≤ 25 kHz	K Factor = 0.87
001	32 kHz < f ≤ 84 kHz	25 kHz < f ≤ 50 kHz	K Factor = 2.6
010	84 kHz < f ≤ 225 kHz	50 kHz < f ≤ 100 kHz	K Factor = 7.7
011	225 kHz < f ≤ 590 kHz	100 kHz < f ≤ 200 kHz	K Factor = 22
100	590 kHz < f ≤ 1.5 MHz	200 kHz < f ≤ 400 kHz	K Factor = 65
101	1.5 MHz < f ≤ 4 MHz	400 kHz < f ≤ 800 kHz	K Factor = 180
110	4 MHz < f ≤ 10 MHz	800 kHz < f ≤ 1.6 MHz	K Factor = 664
111	10 MHz < f ≤ 30 MHz	1.6 MHz < f ≤ 3.2 MHz	K Factor = 1590

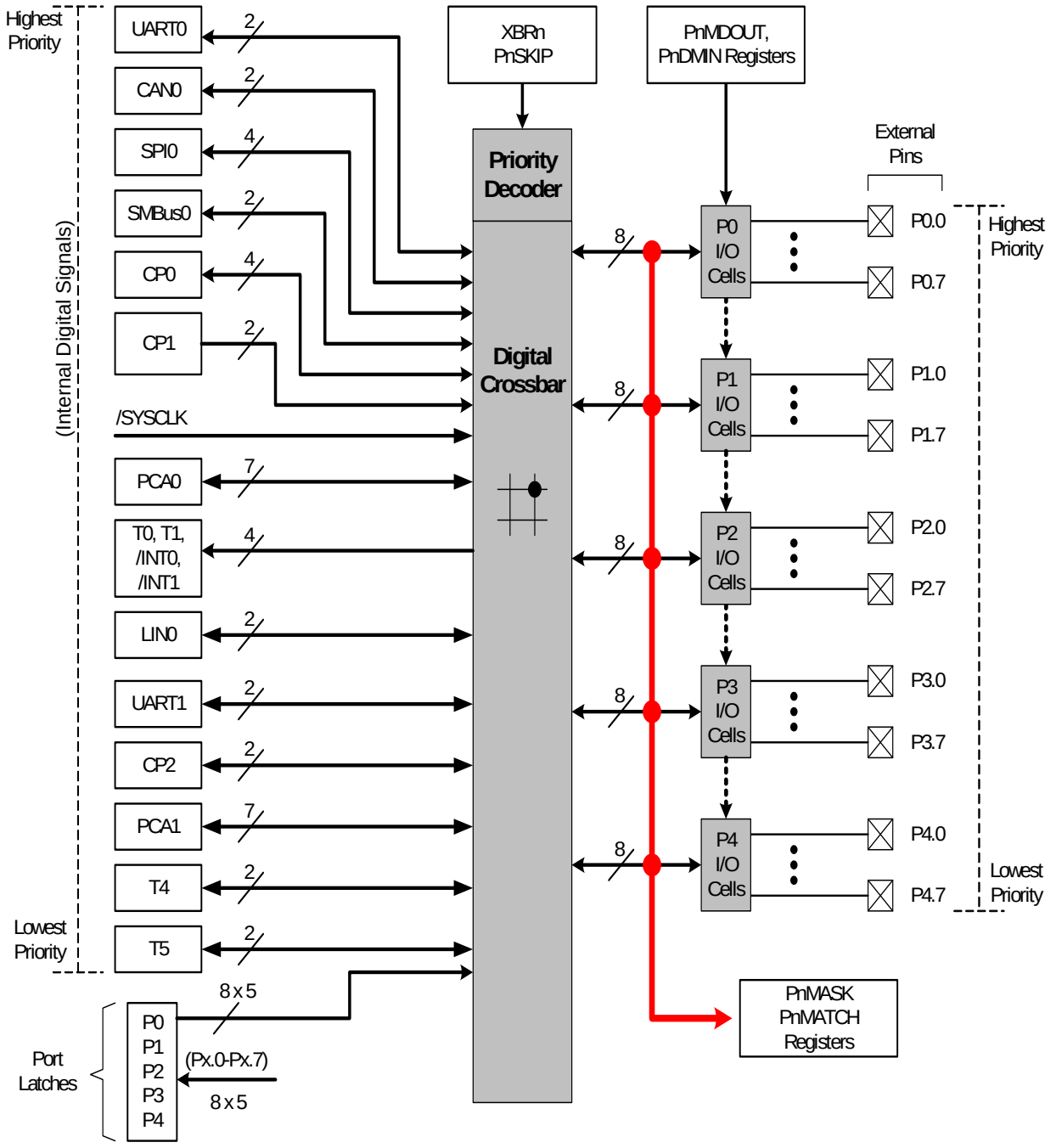

Figure 20.1. Port I/O Functional Block Diagram

SFR Definition 20.20. P1SKIP: Port 1 Skip

Bit	7	6	5	4	3	2	1	0
Name	P1SKIP[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD5; SFR Page = 0x0F

Bit	Name	Function
7:0	P1SKIP[7:0]	**Port 1 Crossbar Skip Enable Bits.** These bits select Port 1 pins to be skipped by the Crossbar Decoder. Port pins used for analog, special functions or GPIO should be skipped by the Crossbar. 0: Corresponding P1.n pin is not skipped by the Crossbar. 1: Corresponding P1.n pin is skipped by the Crossbar.

SFR Definition 20.21. P2: Port 2

Bit	7	6	5	4	3	2	1	0
Name	P2[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xA0; SFR Page = All Pages; Bit-Addressable

Bit	Name	Description	Write	Read
7:0	P2[7:0]	**Port 2Data.** Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P2.n Port pin is logic LOW. 1: P2.n Port pin is logic HIGH.

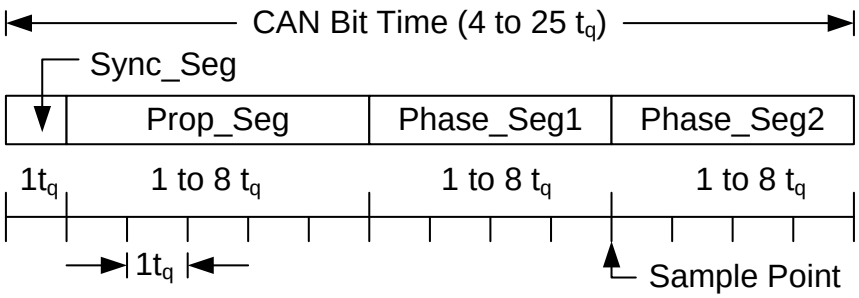

Figure 22.3. Four segments of a CAN Bit

The length of the 4 bit segments must be adjusted so that their sum is as close as possible to the desired bit time. Since each segment must be an integer multiple of the time quantum (t_q), the closest achievable bit time is 24 tq (1000.008 ns), yielding a bit rate of 0.999992 Mbit/sec. The Sync_Seg is a constant 1 tq. The Prop_Seg must be greater than or equal to the propagation delay of 400 ns and so the choice is 10 tq (416.67 ns).

The remaining time quanta (13 tq) in the bit time are divided between Phase_Seg1 and Phase_Seg2 as shown in. Based on this equation, Phase_Seg1 = 6 tq and Phase_Seg2 = 7 tq.

Phase_Seg1 + Phase_Seg2 = Bit_Time – (Synch_Seg + Prop_Seg)

1. If Phase_Seg1 + Phase_Seg2 is even, then Phase_Seg2 = Phase_Seg1. If the sum is odd, Phase_Seg2 = Phase_Seg1 + 1.
2. Phase_Seg2 should be at least 2 t_q.

Equation 22.1. Assigning the Phase Segments

The Synchronization Jump Width (SJW) timing parameter is defined by. It is used for determining the value written to the Bit Timing Register and for determining the required oscillator tolerance. Since we are using a quartz crystal as the system clock source, an oscillator tolerance calculation is not needed.

SJW = minimum (4, Phase_Seg1)

Equation 22.2. Synchronization Jump Width (SJW)

The value written to the Bit Timing Register can be calculated using Equation 18.3. The BRP Extension register is left at its reset value of 0x0000.

BRPE = BRP – 1 = BRP Extension Register = 0x0000

SJWp = SJW – 1 = minimum (4, 6) – 1 = 3

TSEG1 = Prop_Seg + Phase_Seg1 - 1 = 10 + 6 – 1 = 15

TSEG2 = Phase_Seg2 – 1 = 6

Bit Timing Register = (TSEG2 x 0x1000) + (TSEG1 x 0x0100)

Bit Timing Register = (TSEG2 x 0x1000) + (TSEG1 x 0x0100) + (SJWp x 0x0040) + BRPE = 0x6FC0

Equation 22.3. Calculating the Bit Timing Register Value

23.1. 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.

23.2. SMBus Configuration

Figure 23.2 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.

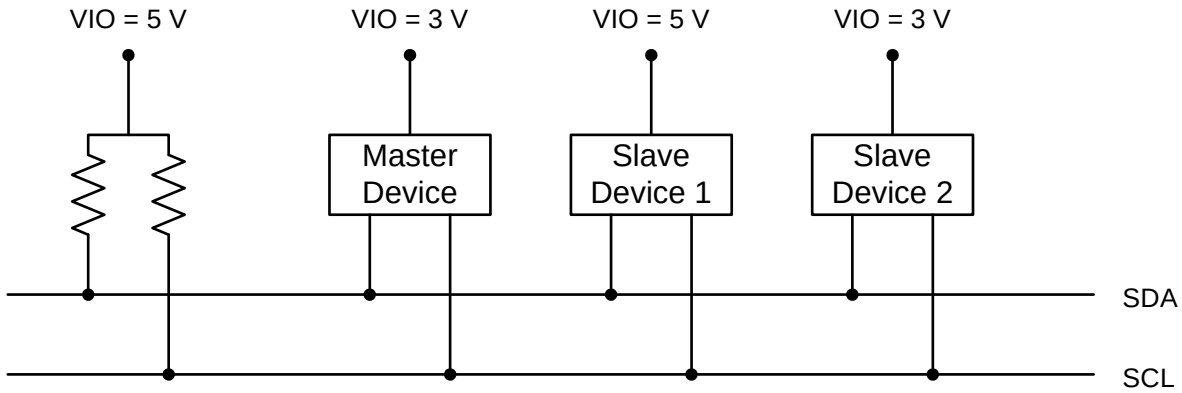

Figure 23.2. Typical SMBus Configuration

23.3. 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 23.3). 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.

overflow after 25 ms (and SMBTOE set), the Timer 3 interrupt service routine can be used to reset (disable and re-enable) the SMBus in the event of an SCL low timeout.

23.3.5. SCL High (SMBus Free) Timeout

The SMBus specification stipulates that if the SCL and SDA lines remain high for more that 50 µs, the bus is designated as free. When the SMBFTE bit in SMB0CF is set, the bus will be considered free if SCL and SDA remain high for more than 10 SMBus clock source periods (as defined by the timer configured for the SMBus clock source). If the SMBus is waiting to generate a Master START, the START will be generated following this timeout. Note that a clock source is required for free timeout detection, even in a slave-only implementation.

23.4. Using the SMBus

The SMBus can operate in both Master and Slave modes. The interface provides timing and shifting control for serial transfers; higher level protocol is determined by user software. The SMBus interface provides the following application-independent features:

- Byte-wise serial data transfers
- Clock signal generation on SCL (Master Mode only) and SDA data synchronization
- Timeout/bus error recognition, as defined by the SMB0CF configuration register
- START/STOP timing, detection, and generation
- Bus arbitration
- Interrupt generation
- Status information

SMBus interrupts are generated for each data byte or slave address that is transferred. The point at which the interrupt is generated depends on whether the hardware is acting as a data transmitter or receiver. When a transmitter (i.e. sending address/data, receiving an ACK), this interrupt is generated after the ACK cycle so that software may read the received ACK value; when receiving data (i.e. receiving address/data, sending an ACK), this interrupt is generated before the ACK cycle so that software may define the outgoing ACK value. See Section 23.5 for more details on transmission sequences.

Interrupts are also generated to indicate the beginning of a transfer when a master (START generated), or the end of a transfer when a slave (STOP detected). Software should read the SMB0CN (SMBus Control register) to find the cause of the SMBus interrupt. The SMB0CN register is described in Section 23.4.2; Table 23.4 provides a quick SMB0CN decoding reference.

23.4.1. SMBus Configuration Register

The SMBus Configuration register (SMB0CF) is used to enable the SMBus Master and/or Slave modes, select the SMBus clock source, and select the SMBus timing and timeout options. When the ENSMB bit is set, the SMBus is enabled for all master and slave events. Slave events may be disabled by setting the INH bit. With slave events inhibited, the SMBus interface will still monitor the SCL and SDA pins; however, the interface will NACK all received addresses and will not generate any slave interrupts. When the INH bit is set, all slave events will be inhibited following the next START (interrupts will continue for the duration of the current transfer).

Setting the EXTHOLD bit extends the minimum setup and hold times for the SDA line. The minimum SDA setup time defines the absolute minimum time that SDA is stable before SCL transitions from low-to-high. The minimum SDA hold time defines the absolute minimum time that the current SDA value remains stable after SCL transitions from high-to-low. EXTHOLD should be set so that the minimum setup and hold times meet the SMBus Specification requirements of 250 ns and 300 ns, respectively. Table 23.2 shows the minimum setup and hold times for the two EXTHOLD settings. Setup and hold time extensions are typically necessary when SYSCLK is above 10 MHz.

Table 23.2. Minimum SDA Setup and Hold Times

EXTHOLD	Minimum SDA Setup Time	Minimum SDA Hold Time
0	T_{low} – 4 system clocks or 1 system clock + s/w delay*	3 system clocks
1	11 system clocks	12 system clocks
***Note:** Setup Time for ACK bit transmissions and the MSB of all data transfers. When using software acknowledgement, the s/w delay occurs between the time SMB0DAT or ACK is written and when SI is cleared. Note that if SI is cleared in the same write that defines the outgoing ACK value, s/w delay is zero.		

With the SMBTOE bit set, Timer 3 should be configured to overflow after 25 ms in order to detect SCL low timeouts (see Section “23.3.4. SCL Low Timeout” on page 241). The SMBus interface will force Timer 3 to reload while SCL is high, and allow Timer 3 to count when SCL is low. The Timer 3 interrupt service routine should be used to reset SMBus communication by disabling and re-enabling the SMBus.

SMBus Free Timeout detection can be enabled by setting the SMBFTE bit. When this bit is set, the bus will be considered free if SDA and SCL remain high for more than 10 SMBus clock source periods (see Figure 23.4).

29.3.6. 16-Bit Pulse Width Modulator Mode

A PCA1 module may also be operated in 16-Bit PWM mode. 16-bit PWM mode is independent of the other (8/9/10/11-bit) PWM modes. In this mode, the 16-bit capture/compare module defines the number of PCA1 clocks for the low time of the PWM signal. When the PCA1 counter matches the module contents, the output on CEXn is asserted high; when the 16-bit counter overflows, CEXn is asserted low. To output a varying duty cycle, new value writes should be synchronized with PCA1 CCFn match interrupts. 16-Bit PWM Mode is enabled by setting the ECOM1n, PWM1n, and PWM161n bits in the PCA1CPMn register. For a varying duty cycle, match interrupts should be enabled (ECCF1n = 1 AND MAT1n = 1) to help synchronize the capture/compare register writes. If the MAT1n bit is set to 1, the CCFn flag for the module will be set each time a 16-bit comparator match (rising edge) occurs. The CF1 flag in PCA1CN can be used to detect the overflow (falling edge). The duty cycle for 16-Bit PWM Mode is given by Equation 29.4.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA1 Capture/Compare registers, the low byte should always be written first. Writing to PCA1CPLn clears the ECOM1n bit to 0; writing to PCA1CPHn sets ECOM1n to 1.

$$\text{Duty Cycle} = \frac{(65536 - \text{PCA1CPn})}{65536}$$

Equation 29.4. 16-Bit PWM Duty Cycle

Using Equation 29.4, the largest duty cycle is 100% (PCA1CPn = 0), and the smallest duty cycle is 0.0015% (PCA1CPn = 0xFFFF). A 0% duty cycle may be generated by clearing the ECOM1n bit to 0.

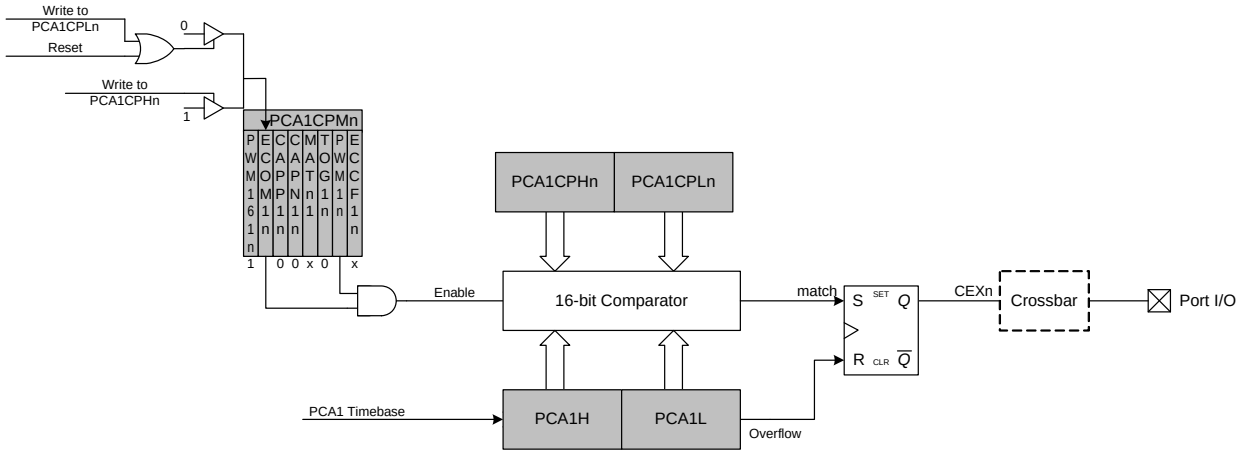

Figure 29.10. PCA1 16-Bit PWM Mode