



Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	8051
Core Size	8-Bit
Speed	50MHz
Connectivity	SMBus (2-Wire/I ² C), LINbus, SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	25
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	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/c8051f566-iq

List of Tables

Table 2.1. Product Selection Guide	21
Table 3.1. Pin Definitions for the C8051F55x/56x/57x	22
Table 4.1. QFN-40 Package Dimensions	28
Table 4.2. QFN-40 Landing Diagram Dimensions	29
Table 4.3. QFP-32 Package Dimensions	30
Table 4.4. QFP-32 Landing Diagram Dimensions	31
Table 4.5. QFN-32 Package Dimensions	32
Table 4.6. QFN-32 Landing Diagram Dimensions	33
Table 4.7. QFN-24 Package Dimensions	34
Table 4.8. QFN-24 Landing Diagram Dimensions	35
Table 5.1. Absolute Maximum Ratings	36
Table 5.2. Global Electrical Characteristics	37
Table 5.3. Port I/O DC Electrical Characteristics	40
Table 5.4. Reset Electrical Characteristics	41
Table 5.5. Flash Electrical Characteristics	41
Table 5.6. Internal High-Frequency Oscillator Electrical Characteristics	42
Table 5.7. Clock Multiplier Electrical Specifications	43
Table 5.8. Voltage Regulator Electrical Characteristics	43
Table 5.9. ADC0 Electrical Characteristics	44
Table 5.10. Temperature Sensor Electrical Characteristics	45
Table 5.11. Voltage Reference Electrical Characteristics	45
Table 5.12. Comparator 0 and Comparator 1 Electrical Characteristics	46
Table 10.1. CIP-51 Instruction Set Summary	84
Table 12.1. Special Function Register (SFR) Memory Map for Pages 0x00 and 0x0F 106	
Table 12.2. Special Function Register (SFR) Memory Map for Page 0x0C	107
Table 12.3. Special Function Registers	108
Table 13.1. Interrupt Summary	114
Table 14.1. Flash Security Summary	128
Table 17.1. EMIF Pinout (C8051F568-9 and 'F570-5)	146
Table 17.2. AC Parameters for External Memory Interface	156
Table 19.1. Port I/O Assignment for Analog Functions	171
Table 19.2. Port I/O Assignment for Digital Functions	172
Table 19.3. Port I/O Assignment for External Digital Event Capture Functions	172
Table 20.1. Baud Rate Calculation Variable Ranges	194
Table 20.2. Manual Baud Rate Parameters Examples	196
Table 20.3. Autobaud Parameters Examples	197
Table 20.4. LIN Registers* (Indirectly Addressable)	202
Table 21.1. Background System Information	212
Table 21.2. Standard CAN Registers and Reset Values	215
Table 22.1. SMBus Clock Source Selection	222
Table 22.2. Minimum SDA Setup and Hold Times	223

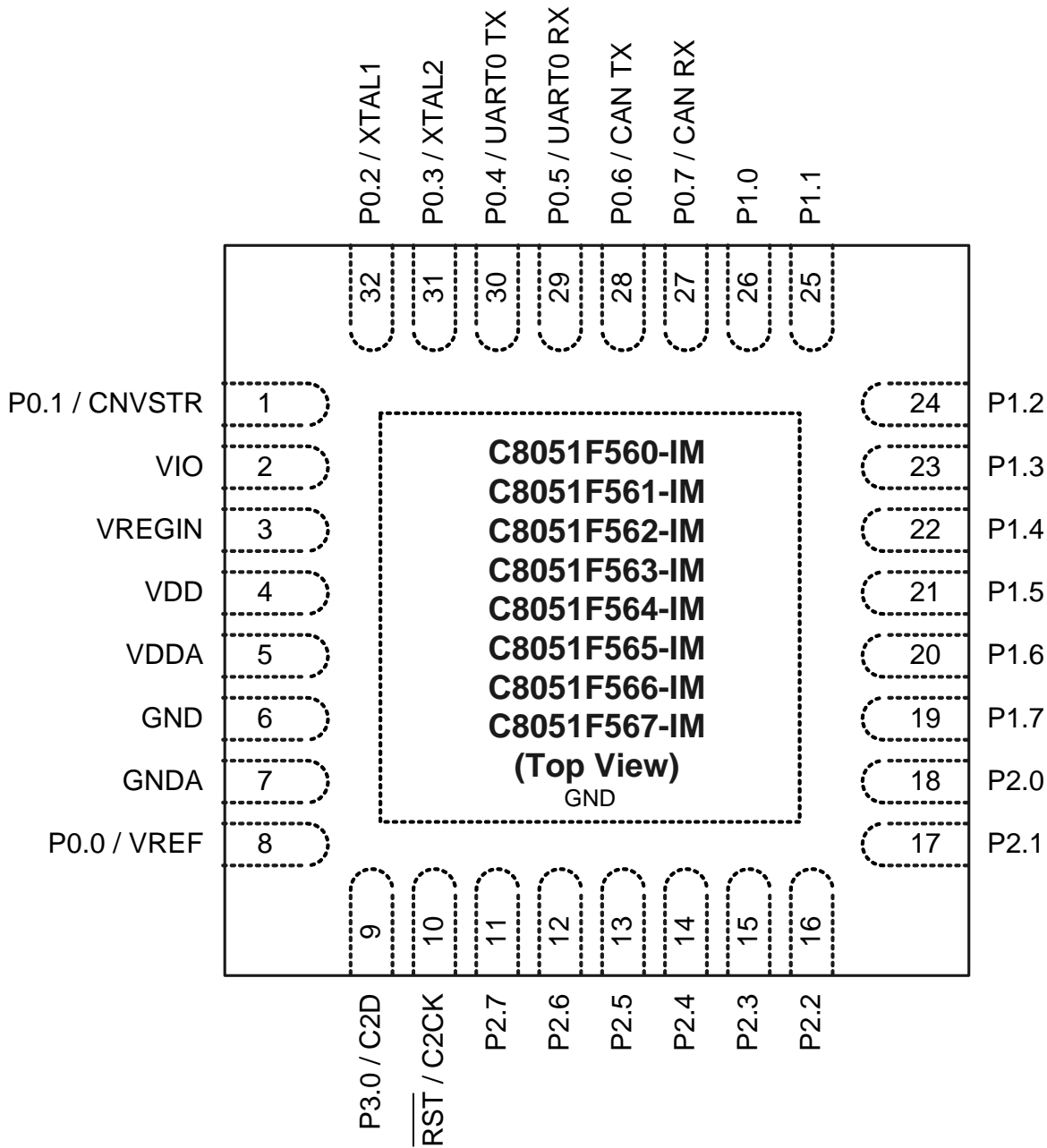


Figure 3.3. QFN-32 Pinout Diagram (Top View)

6.1. Modes of Operation

In a typical system, ADC0 is configured using the following steps:

1. If a gain adjustment is required, refer to Section “6.3. Selectable Gain” on page 53.
2. Choose the start of conversion source.
3. Choose Normal Mode or Burst Mode operation.
4. If Burst Mode, choose the ADC0 Idle Power State and set the Power-up Time.
5. Choose the tracking mode. Note that Pre-Tracking Mode can only be used with Normal Mode.
6. Calculate the required settling time and set the post convert-start tracking time using the AD0TK bits.
7. Choose the repeat count.
8. Choose the output word justification (Right-Justified or Left-Justified).
9. Enable or disable the End of Conversion and Window Comparator Interrupts.

6.1.1. Starting a Conversion

A conversion can be initiated in one of four ways, depending on the programmed states of the ADC0 Start of Conversion Mode bits (AD0CM1–0) in register ADC0CN. Conversions may be initiated by one of the following:

- Writing a 1 to the AD0BUSY bit of register ADC0CN
- A rising edge on the CNVSTR input signal (pin P0.1)
- A Timer 1 overflow (i.e., timed continuous conversions)
- A Timer 2 overflow (i.e., timed continuous conversions)

Writing a 1 to AD0BUSY provides software control of ADC0 whereby conversions are performed “on-demand.” During conversion, the AD0BUSY bit is set to logic 1 and reset to logic 0 when the conversion is complete. The falling edge of AD0BUSY triggers an interrupt (when enabled) and sets the ADC0 interrupt flag (AD0INT). Note: When polling for ADC conversion completions, the ADC0 interrupt flag (AD0INT) should be used. Converted data is available in the ADC0 data registers, ADC0H:ADC0L, when bit AD0INT is logic 1. Note that when Timer 2 overflows are used as the conversion source, Low Byte overflows are used if Timer2 is in 8-bit mode; High byte overflows are used if Timer 2 is in 16-bit mode. See Section “25. Timers” on page 259 for timer configuration.

Important Note About Using CNVSTR: The CNVSTR input pin also functions as Port pin P0.1. When the CNVSTR input is used as the ADC0 conversion source, Port pin P0.1 should be skipped by the Digital Crossbar. To configure the Crossbar to skip P0.1, set to 1 Bit1 in register P0SKIP. See Section “19. Port Input/Output” on page 169 for details on Port I/O configuration.

6.1.2. Tracking Modes

Each ADC0 conversion must be preceded by a minimum tracking time for the converted result to be accurate. ADC0 has three tracking modes: Pre-Tracking, Post-Tracking, and Dual-Tracking. Pre-Tracking Mode provides the minimum delay between the convert start signal and end of conversion by tracking continuously before the convert start signal. This mode requires software management in order to meet minimum tracking requirements. In Post-Tracking Mode, a programmable tracking time starts after the convert start signal and is managed by hardware. Dual-Tracking Mode maximizes tracking time by tracking before and after the convert start signal. Figure 6.2 shows examples of the three tracking modes.

Pre-Tracking Mode is selected when AD0TM is set to 10b. Conversions are started immediately following the convert start signal. ADC0 is tracking continuously when not performing a conversion. Software must allow at least the minimum tracking time between each end of conversion and the next convert start signal. The minimum tracking time must also be met prior to the first convert start signal after ADC0 is enabled.

C8051F55x/56x/57x

SFR Definition 7.1. REF0CN: Reference Control

Bit	7	6	5	4	3	2	1	0
Name			ZTCEN	REFLV	REFSL	TEMPE	BIASE	REFBE
Type	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD1; SFR Page = 0x00

Bit	Name	Function
7:6	Unused	Read = 00b; Write = don't care.
5	ZTCEN	Zero Temperature Coefficient Bias Enable Bit. This bit must be set to 1b before entering oscillator suspend mode. 0: ZeroTC Bias Generator automatically enabled when required. 1: ZeroTC Bias Generator forced on.
4	REFLV	Voltage Reference Output Level Select. This bit selects the output voltage level for the internal voltage reference 0: Internal voltage reference set to 1.5 V. 1: Internal voltage reference set to 2.20 V.
3	REFSL	Voltage Reference Select. This bit selects the ADCs voltage reference. 0: V_{REF} pin used as voltage reference. 1: V_{DD} used as voltage reference. If V_{DD} is selected as the voltage reference and the ADC is enabled in the ADC0CN register, the P0.0/VREF pin cannot operate as a general purpose I/O pin in open-drain mode. With the above settings, this pin can operate in push-pull output mode or as an analog input.
2	TEMPE	Temperature Sensor Enable Bit. 0: Internal Temperature Sensor off. 1: Internal Temperature Sensor on.
1	BIASE	Internal Analog Bias Generator Enable Bit. 0: Internal Bias Generator off. 1: Internal Bias Generator on.
0	REFBE	On-chip Reference Buffer Enable Bit. 0: On-chip Reference Buffer off. 1: On-chip Reference Buffer on. Internal voltage reference driven on the V_{REF} pin.

SFR Definition 8.6. CPT1MX: Comparator1 MUX Selection

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

SFR Address = 0x9F; SFR Page = 0x00

Bit	Name	Function
7:4	CMX1N[3:0]	Comparator1 Negative Input MUX Selection. 0000: P0.1 0001: P0.3 0010: P0.5 0011: P0.7 0100: P1.1 0101: P1.3 0110: P1.5 0111: P1.7 1000: P2.1 1001: P2.3 (only available on 40-pin and 32-pin devices) 1010: P2.5 (only available on 40-pin and 32-pin devices) 1011: P2.7 (only available on 40-pin and 32-pin devices) 1100–1111: None
3:0	CMX1P[3:0]	Comparator1 Positive Input MUX Selection. 0000: P0.0 0001: P0.2 0010: P0.4 0011: P0.6 0100: P1.0 0101: P1.2 0110: P1.4 0111: P1.6 1000: P2.0 1001: P2.2 (only available on 40-pin and 32-pin devices) 1010: P2.4 (only available on 40-pin and 32-pin devices) 1011: P2.6 (only available on 40-pin and 32-pin devices) 1100–1111: None

C8051F55x/56x/57x

SFR Definition 12.3. SFRNEXT: SFR Next

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

SFR Address = 0x85; SFR Page = All Pages

Bit	Name	Function
7:0	SFRNEXT[7:0]	<p>SFR Page Bits.</p> <p>This is the value that will go to the SFR Page register upon a return from interrupt.</p> <p>Write: Sets the SFR Page contained in the second byte of the SFR Stack. This will cause the SFRPAGE SFR to have this SFR page value upon a return from interrupt.</p> <p>Read: Returns the value of the SFR page contained in the second byte of the SFR stack.</p> <p>SFR page context is retained upon interrupts/return from interrupts in a 3 byte SFR Page Stack: SFRPAGE is the first entry, SFRNEXT is the second, and SFRLAST is the third entry. The SFR stack bytes may be used alter the context in the SFR Page Stack, and will not cause the stack to “push” or “pop”. Only interrupts and return from interrupts cause pushes and pops of the SFR Page Stack.</p>

15. Power Management Modes

The C8051F55x/56x/57x devices have three software programmable power management modes: Idle, Stop, and Suspend. Idle mode and Stop mode are part of the standard 8051 architecture, while Suspend mode is an enhanced power-saving mode implemented by the high-speed oscillator peripheral.

Idle mode halts the CPU while leaving the peripherals and clocks active. In Stop mode, the CPU is halted, all interrupts and timers (except the Missing Clock Detector) are inactive, and the internal oscillator is stopped (analog peripherals remain in their selected states; the external oscillator is not affected). Suspend mode is similar to Stop mode in that the internal oscillator and CPU are halted, but the device can wake on events such as a Port Match or Comparator low output. Since clocks are running in Idle mode, power consumption is dependent upon the system clock frequency and the number of peripherals left in active mode before entering Idle. Stop mode and Suspend mode consume the least power because the majority of the device is shut down with no clocks active. SFR Definition 15.1 describes the Power Control Register (PCON) used to control the C8051F55x/56x/57x devices' Stop and Idle power management modes. Suspend mode is controlled by the SUSPEND bit in the OSCICN register (SFR Definition 18.2).

Although the C8051F55x/56x/57x has Idle, Stop, and Suspend modes available, more control over the device power can be achieved by enabling/disabling individual peripherals as needed. Each analog peripheral can be disabled when not in use and placed in low power mode. Digital peripherals, such as timers or serial buses, draw little power when they are not in use. Turning off oscillators lowers power consumption considerably, at the expense of reduced functionality.

15.1. Idle Mode

Setting the Idle Mode Select bit (PCON.0) causes the hardware to halt the CPU and enter Idle mode as soon as the instruction that sets the bit completes execution. All internal registers and memory maintain their original data. All analog and digital peripherals can remain active during Idle mode.

Idle mode is terminated when an enabled interrupt is asserted or a reset occurs. The assertion of an enabled interrupt will cause the Idle Mode Selection bit (PCON.0) to be cleared and the CPU to resume operation. The pending interrupt will be serviced and the next instruction to be executed after the return from interrupt (RETI) will be the instruction immediately following the one that set the Idle Mode Select bit. If Idle mode is terminated by an internal or external reset, the CIP-51 performs a normal reset sequence and begins program execution at address 0x0000.

Note: If the instruction following the write of the IDLE bit is a single-byte instruction and an interrupt occurs during the execution phase of the instruction that sets the IDLE bit, the CPU may not wake from Idle mode when a future interrupt occurs. Therefore, instructions that set the IDLE bit should be followed by an instruction that has two or more opcode bytes, for example:

```
// in 'C':
PCON |= 0x01;           // set IDLE bit
PCON = PCON;           // ... followed by a 3-cycle dummy instruction

; in assembly:
ORL PCON, #01h          ; set IDLE bit
MOV PCON, PCON          ; ... followed by a 3-cycle dummy instruction
```

If enabled, the Watchdog Timer (WDT) will eventually cause an internal watchdog reset and thereby terminate the Idle mode. This feature protects the system from an unintended permanent shutdown in the event of an inadvertent write to the PCON register. If this behavior is not desired, the WDT may be disabled by software prior to entering the Idle mode if the WDT was initially configured to allow this operation. This provides the opportunity for additional power savings, allowing the system to remain in the Idle mode indefinitely, waiting for an external stimulus to wake up the system. Refer to Section “16.6. PCA Watchdog Timer Reset” on page 142 for more information on the use and configuration of the WDT.

15.2. Stop Mode

Setting the Stop Mode Select bit (PCON.1) causes the controller core to enter Stop mode as soon as the instruction that sets the bit completes execution. In Stop mode the internal oscillator, CPU, and all digital peripherals are stopped; the state of the external oscillator circuit is not affected. Each analog peripheral (including the external oscillator circuit) may be shut down individually prior to entering Stop Mode. Stop mode can only be terminated by an internal or external reset. On reset, the device performs the normal reset sequence and begins program execution at address 0x0000.

If enabled, the Missing Clock Detector will cause an internal reset and thereby terminate the Stop mode. The Missing Clock Detector should be disabled if the CPU is to be put to in STOP mode for longer than the MCD timeout of 100 μ s.

15.3. Suspend Mode

Setting the SUSPEND bit (OSCIEN.5) causes the hardware to halt the CPU and the high-frequency internal oscillator, and go into Suspend mode as soon as the instruction that sets the bit completes execution. All internal registers and memory maintain their original data. Most digital peripherals are not active in Suspend mode. The exception to this is the Port Match feature.

Suspend mode can be terminated by three types of events, a port match (described in Section “19.5. Port Match” on page 179), a Comparator low output (if enabled), or a device reset event. When Suspend mode is terminated, the device will continue execution on the instruction following the one that set the SUSPEND bit. If the wake event was configured to generate an interrupt, the interrupt will be serviced upon waking the device. If Suspend mode is terminated by an internal or external reset, the CIP-51 performs a normal reset sequence and begins program execution at address 0x0000.

Note: Before entering suspend mode, firmware must set the ZTCEN bit in REF0CN (SFR Definition 7.1).

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.

C8051F55x/56x/57x

Port	P 0								P 1								P 2								P 3								P 4
																	P 2.2-P2.7, P3.0 available on 40-pin and 32-pin packages								P3.1-P3.7, P4.0 available on 40-pin packages								
PIN I/O	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0
UART_TX																																	
UART_RX																																	
CAN_TX																																	
CAN_RX																																	
SCK																																	
MISO																																	
MOSI																																	
NSS																																	
SDA																																	
SCL																																	
CP0																																	
CP0A																																	
CP1																																	
CP1A																																	
SYCLK																																	
CEX0																																	
CEX1																																	
CEX2																																	
CEX3																																	
CEX4																																	
CEX5																																	
ECI																																	
T0																																	
T1																																	
LIN_TX																																	
LIN_RX																																	
	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	P0SKIP[0:7]								P1SKIP[0:7]								P2SKIP[0:7]								P3SKIP[0:7]								

Figure 19.4. Crossbar Priority Decoder in Example Configuration

19.4. Port I/O Initialization

Port I/O initialization consists of the following steps:

1. Select the input mode (analog or digital) for all Port pins, using the Port Input Mode register (PnMDIN).
2. Select the output mode (open-drain or push-pull) for all Port pins, using the Port Output Mode register (PnMDOUT).
3. Select any pins to be skipped by the I/O Crossbar using the Port Skip registers (PnSKIP).
4. Assign Port pins to desired peripherals.
5. Enable the Crossbar (XBARE = 1).

All Port pins must be configured as either analog or digital inputs. Port 4 C8051F568-9 and 'F570-5 is a digital-only Port. Any pins to be used as Comparator or ADC inputs should be configured as an analog inputs. When a pin is configured as an analog input, its weak pullup, digital driver, and digital receiver are disabled. This process saves power and reduces noise on the analog input. Pins configured as digital inputs may still be used by analog peripherals; however this practice is not recommended.

Additionally, all analog input pins should be configured to be skipped by the Crossbar (accomplished by setting the associated bits in PnSKIP). Port input mode is set in the PnMDIN register, where a 1 indicates a digital input, and a 0 indicates an analog input. All pins default to digital inputs on reset. See SFR Definition 19.13 for the PnMDIN register details.

22.1. Supporting Documents

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

1. The I²C-Bus and How to Use It (including specifications), Philips Semiconductor.
2. The I²C-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.

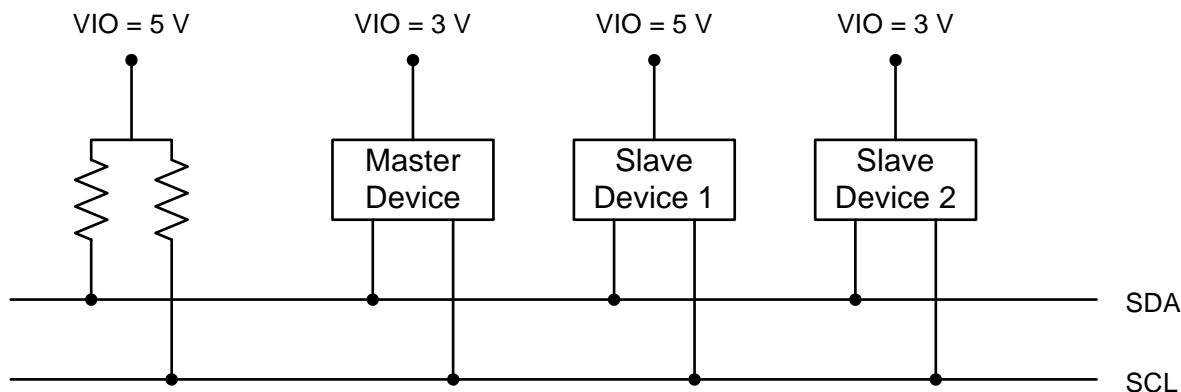


Figure 22.2. Typical SMBus Configuration

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.

All transactions are initiated by a master, with one or more addressed slave devices as the target. The master generates the START condition and then transmits the slave address and direction bit. If the transaction is a WRITE operation from the master to the slave, the master transmits the data a byte at a time waiting for an ACK from the slave at the end of each byte. For READ operations, the slave transmits the data waiting for an ACK from the master at the end of each byte. At the end of the data transfer, the master generates a STOP condition to terminate the transaction and free the bus. Figure 22.3 illustrates a typical SMBus transaction.

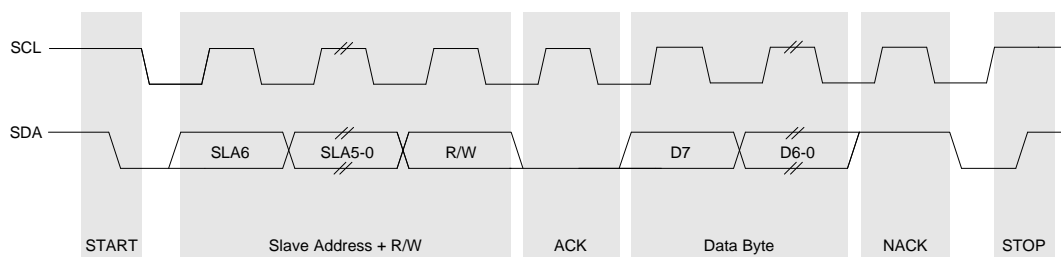


Figure 22.3. SMBus Transaction

22.3.1. Transmitter Vs. Receiver

On the SMBus communications interface, a device is the “transmitter” when it is sending an address or data byte to another device on the bus. A device is a “receiver” when an address or data byte is being sent to it from another device on the bus. The transmitter controls the SDA line during the address or data byte. After each byte of address or data information is sent by the transmitter, the receiver sends an ACK or NACK bit during the ACK phase of the transfer, during which time the receiver controls the SDA line.

22.3.2. Arbitration

A master may start a transfer only if the bus is free. The bus is free after a STOP condition or after the SCL and SDA lines remain high for a specified time (see Section “22.3.5. SCL High (SMBus Free) Timeout” on page 221). In the event that two or more devices attempt to begin a transfer at the same time, an arbitration scheme is employed to force one master to give up the bus. The master devices continue transmitting until one attempts a HIGH while the other transmits a LOW. Since the bus is open-drain, the bus will be pulled LOW. The master attempting the HIGH will detect a LOW SDA and lose the arbitration. The winning master continues its transmission without interruption; the losing master becomes a slave and receives the rest of the transfer if addressed. This arbitration scheme is non-destructive: one device always wins, and no data is lost.

22.3.3. Clock Low Extension

SMBus provides a clock synchronization mechanism, similar to I²C, which allows devices with different speed capabilities to coexist on the bus. A clock-low extension is used during a transfer in order to allow slower slave devices to communicate with faster masters. The slave may temporarily hold the SCL line LOW to extend the clock low period, effectively decreasing the serial clock frequency.

22.3.4. SCL Low Timeout

If the SCL line is held low by a slave device on the bus, no further communication is possible. Furthermore, the master cannot force the SCL line high to correct the error condition. To solve this problem, the SMBus protocol specifies that devices participating in a transfer must detect any clock cycle held low longer than 25 ms as a “timeout” condition. Devices that have detected the timeout condition must reset the communication no later than 10 ms after detecting the timeout condition.

When the SMBTOE bit in SMB0CF is set, Timer 3 is used to detect SCL low timeouts. Timer 3 is forced to reload when SCL is high, and allowed to count when SCL is low. With Timer 3 enabled and configured to

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.

22.3.5. SCL High (SMBus Free) Timeout

The SMBus specification stipulates that if the SCL and SDA lines remain high for more than 50 μ s, 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.

22.4. 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 22.5 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 22.4.2; Table 22.4 provides a quick SMB0CN decoding reference.

22.4.1. 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).

23.2. Data Format

UART0 has a number of available options for data formatting. Data transfers begin with a start bit (logic low), followed by the data bits (sent LSB-first), a parity or extra bit (if selected), and end with one or two stop bits (logic high). The data length is variable between 5 and 8 bits. A parity bit can be appended to the data, and automatically generated and detected by hardware for even, odd, mark, or space parity. The stop bit length is selectable between 1 and 2 bit times, and a multi-processor communication mode is available for implementing networked UART buses. All of the data formatting options can be configured using the SMOD0 register, shown in SFR Definition 23.2. Figure 23.2 shows the timing for a UART0 transaction without parity or an extra bit enabled. Figure 23.3 shows the timing for a UART0 transaction with parity enabled ($PE0 = 1$). Figure 23.4 is an example of a UART0 transaction when the extra bit is enabled ($XBE0 = 1$). Note that the extra bit feature is not available when parity is enabled, and the second stop bit is only an option for data lengths of 6, 7, or 8 bits.

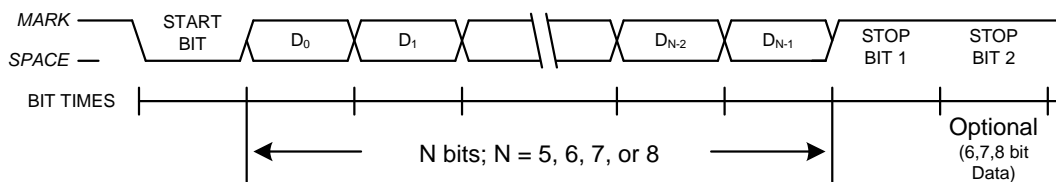


Figure 23.2. UART0 Timing Without Parity or Extra Bit

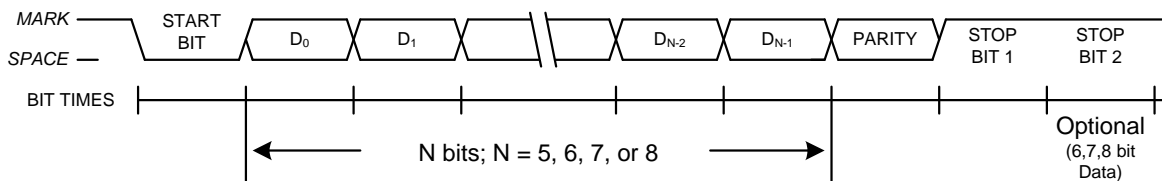


Figure 23.3. UART0 Timing With Parity

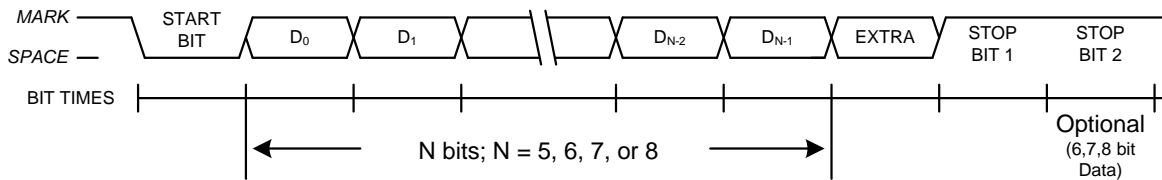


Figure 23.4. UART0 Timing With Extra Bit

SFR Definition 25.4. TL0: Timer 0 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TL0[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x8A; SFR Page = All Pages

Bit	Name	Function
7:0	TL0[7:0]	Timer 0 Low Byte. The TL0 register is the low byte of the 16-bit Timer 0.

SFR Definition 25.5. TL1: Timer 1 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TL1[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x8B; SFR Page = All Pages

Bit	Name	Function
7:0	TL1[7:0]	Timer 1 Low Byte. The TL1 register is the low byte of the 16-bit Timer 1.

25.3. Timer 3

Timer 3 is a 16-bit timer formed by two 8-bit SFRs: TMR3L (low byte) and TMR3H (high byte). Timer 3 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T3SPLIT bit (TMR3CN.3) defines the Timer 3 operation mode.

Timer 3 may be clocked by the system clock, the system clock divided by 12, or the external oscillator source divided by 8. The external clock mode is ideal for real-time clock (RTC) functionality, where the internal oscillator drives the system clock while Timer 3 (and/or the PCA) is clocked by an external precision oscillator. Note that the external oscillator source divided by 8 is synchronized with the system clock.

25.3.1. 16-Bit Timer with Auto-Reload

When T3SPLIT (TMR3CN.3) is zero, Timer 3 operates as a 16-bit timer with auto-reload. Timer 3 can be clocked by SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. As the 16-bit timer register increments and overflows from 0xFFFF to 0x0000, the 16-bit value in the Timer 3 reload registers (TMR3RLH and TMR3RLL) is loaded into the Timer 3 register as shown in Figure 25.7, and the Timer 3 High Byte Overflow Flag (TMR3CN.7) is set. If Timer 3 interrupts are enabled, an interrupt will be generated on each Timer 3 overflow. Additionally, if Timer 3 interrupts are enabled and the TF3LEN bit is set (TMR3CN.5), an interrupt will be generated each time the lower 8 bits (TMR3L) overflow from 0xFF to 0x00.

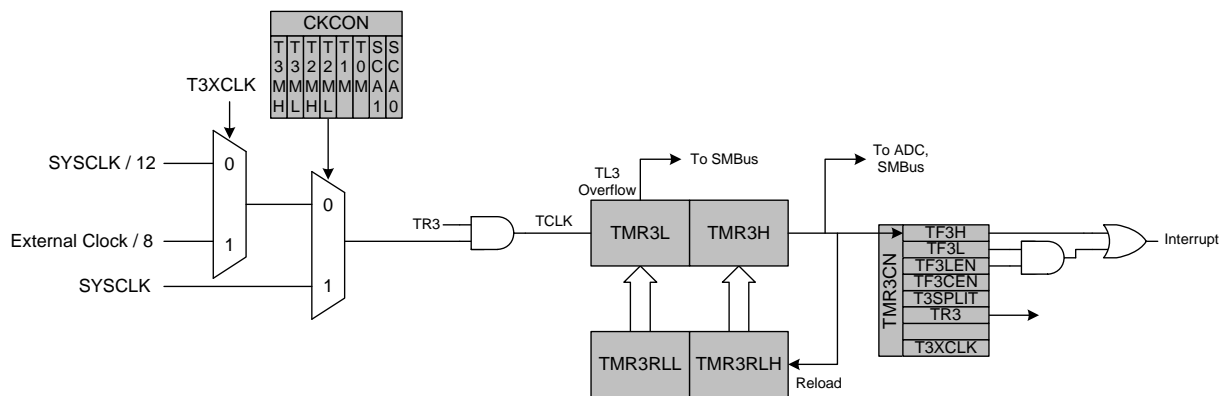


Figure 25.7. Timer 3 16-Bit Mode Block Diagram

25.3.2. 8-Bit Timers with Auto-Reload

When T3SPLIT is set, Timer 3 operates as two 8-bit timers (TMR3H and TMR3L). Both 8-bit timers operate in auto-reload mode as shown in Figure 25.8. TMR3RLL holds the reload value for TMR3L; TMR3RLH holds the reload value for TMR3H. The TR3 bit in TMR3CN handles the run control for TMR3H. TMR3L is always running when configured for 8-bit Mode.

Each 8-bit timer may be configured to use SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. The Timer 3 Clock Select bits (T3MH and T3ML in CKCON) select either SYSCLK or the clock defined by the Timer 3 External Clock Select bit (T3XCLK in TMR3CN), as follows:

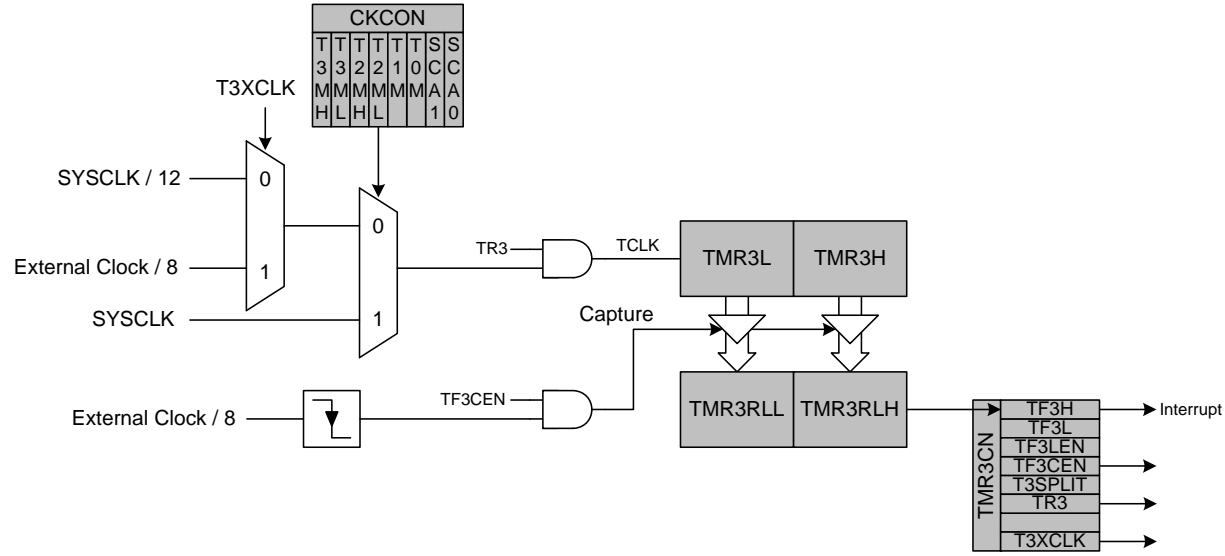


Figure 25.9. Timer 3 External Oscillator Capture Mode Block Diagram

Table 26.2. PCA0CPM and PCA0PWM Bit Settings for PCA Capture/Compare Modules

Operational Mode Bit Number	PCA0CPMn								PCA0PWM				
	7	6	5	4	3	2	1	0	7	6	5	4–2	1–0
Capture triggered by positive edge on CEXn	X	X	1	0	0	0	0	A	0	X	B	XXX	XX
Capture triggered by negative edge on CEXn	X	X	0	1	0	0	0	A	0	X	B	XXX	XX
Capture triggered by any transition on CEXn	X	X	1	1	0	0	0	A	0	X	B	XXX	XX
Software Timer	X	C	0	0	1	0	0	A	0	X	B	XXX	XX
High Speed Output	X	C	0	0	1	1	0	A	0	X	B	XXX	XX
Frequency Output	X	C	0	0	0	1	1	A	0	X	B	XXX	XX
8-Bit Pulse Width Modulator (7)	0	C	0	0	E	0	1	A	0	X	B	XXX	00
9-Bit Pulse Width Modulator (7)	0	C	0	0	E	0	1	A	D	X	B	XXX	01
10-Bit Pulse Width Modulator (7)	0	C	0	0	E	0	1	A	D	X	B	XXX	10
11-Bit Pulse Width Modulator (7)	0	C	0	0	E	0	1	A	D	X	B	XXX	11
16-Bit Pulse Width Modulator	1	C	0	0	E	0	1	A	0	X	B	XXX	XX
Notes: <ol style="list-style-type: none"> 1. X = Don't Care (no functional difference for individual module if 1 or 0). 2. A = Enable interrupts for this module (PCA interrupt triggered on CCFn set to 1). 3. B = Enable 8th, 9th, 10th or 11th bit overflow interrupt (Depends on setting of CLSEL[1:0]). 4. C = When set to 0, the digital comparator is off. For high speed and frequency output modes, the associated pin will not toggle. In any of the PWM modes, this generates a 0% duty cycle (output = 0). 5. D = Selects whether the Capture/Compare register (0) or the Auto-Reload register (1) for the associated channel is accessed via addresses PCA0CPHn and PCA0CPLn. 6. E = When set, a match event will cause the CCFn flag for the associated channel to be set. 7. All modules set to 8, 9, 10 or 11-bit PWM mode use the same cycle length setting. 													

26.3.1. Edge-triggered Capture Mode

In this mode, a valid transition on the CEXn pin causes the PCA to capture the value of the PCA counter/timer and load it into the corresponding module's 16-bit capture/compare register (PCA0CPLn and PCA0CPHn). The CAPPn and CAPNn bits in the PCA0CPMn register are used to select the type of transition that triggers the capture: low-to-high transition (positive edge), high-to-low transition (negative edge), or either transition (positive or negative edge). When a capture 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. If both CAPPn and CAPNn bits are set to logic 1, then the state of the Port pin associated with CEXn can be read directly to determine whether a rising-edge or falling-edge caused the capture.

Revision 1.1 to Revision 1.2

- Updated the note in “Power-Fail Reset/VDD Monitor” on page 140 to use a larger font.
- Added the note regarding the voltage regulator and VDD monitor in the high setting from “Power-Fail Reset/VDD Monitor” on page 140 to “Voltage Regulator (REG0)” on page 79 and “V_{DD} Maintenance and the V_{DD} monitor” on page 129.
- Updated the steps in “V_{DD} Maintenance and the V_{DD} monitor” on page 129 to mention using the VDD monitor in the high setting during flash write/erase operations.
- Updated the SUSPEND bit description in OSCICN (SFR Definition 18.2) to mention that firmware must set the ZTCEN bit in REF0CN (SFR Definition 7.1) before entering suspend.
- Added a note to the IFRDY flag in the OSCICN register (SFR Definition 18.2) that the flag may not accurately reflect the state of the oscillator.
- Added VREGIN Ramp Time for Power On spec to Table 5.4, “Reset Electrical Characteristics,” on page 41.
- Updated “V_{DD} Maintenance and the V_{DD} monitor” on page 129 to refer to V_{REGIN} ramp time instead of V_{DD} ramp time.
- Added a note regarding programming at cold temperatures on –I devices to “Programming The Flash Memory” on page 124 and added Temperature during Programming Operations specification to Table 5.5, “Flash Electrical Characteristics,” on page 41.
- Added a note regarding P0.0/VREF when VDD is used as the reference to Table 19.1, “Port I/O Assignment for Analog Functions,” on page 171 and to the description of the REFSL bit in REF0CN (SFR Definition 7.1).
- Added a note regarding a potential unknown state on GPIO during power up if VIO ramps significantly before VDD to “Port Input/Output” on page 169 and “Reset Sources” on page 138.
- Added steps to set the FLEWT bit in the FLSCCL register (SFR Definition 14.3) in the flash write/erase procedures in “Flash Erase Procedure” on page 125, “Flash Write Procedure” on page 125, and “Flash Write Optimization” on page 126.
- Added a note regarding fast changes on VDD causing the V_{DD} Monitor to trigger to “Power-Fail Reset/VDD Monitor” on page 140.
- Added notes regarding UART TX and RX behavior in “Data Transmission” on page 238, “Data Reception” on page 238, and the THRE0 description in the SCON0 register (SFR Definition 23.1).
- Added a note regarding an issue with /RST low time on some older devices to “16.1. Power-On Reset” .