



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-iqr

C8051F55x/56x/57x

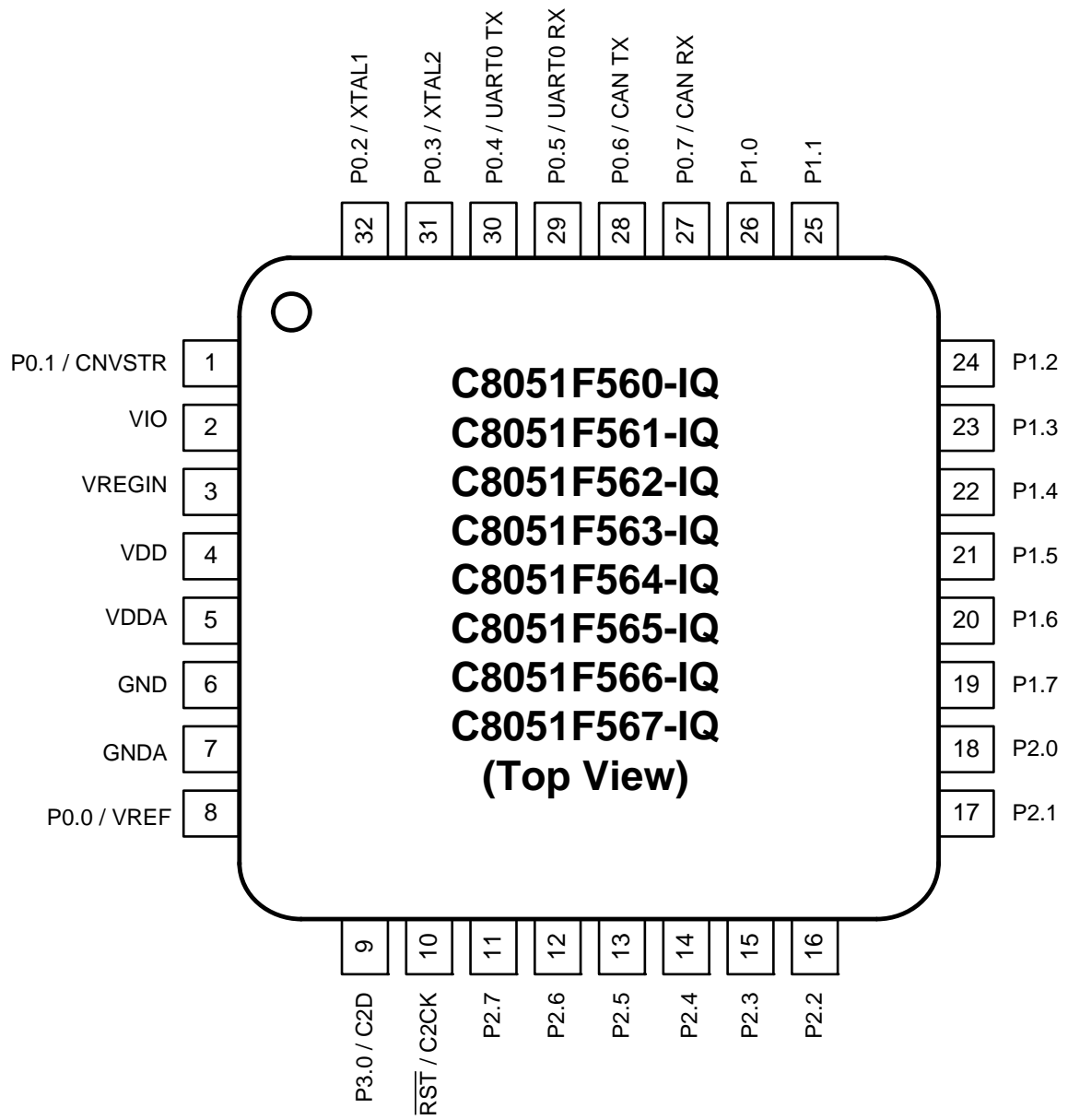


Figure 3.2. QFP-32 Pinout Diagram (Top View)

Table 5.2. Global Electrical Characteristics (Continued)

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

Parameter	Conditions	Min	Typ	Max	Units
I _{DD} Frequency Sensitivity ^{4,5}	V _{DD} = 2.1 V, F ≤ 12.5 MHz, T = 25 °C	—	0.43	—	mA/MHz
	V _{DD} = 2.1 V, F > 12.5 MHz, T = 25 °C	—	0.33	—	mA/MHz
	V _{DD} = 2.6 V, F ≤ 12.5 MHz, T = 25 °C	—	0.60	—	mA/MHz
	V _{DD} = 2.6 V, F > 12.5 MHz, T = 25 °C	—	0.42	—	mA/MHz
Digital Supply Current—CPU Inactive (Idle Mode, not fetching instructions from Flash)					
I _{DD} ⁴	V _{DD} = 2.1 V, F = 200 kHz	—	50	—	μA
	V _{DD} = 2.1 V, F = 1.5 MHz	—	410	—	μA
	V _{DD} = 2.1 V, F = 25 MHz	—	6.5	8.0	mA
	V _{DD} = 2.1 V, F = 50 MHz	—	13	16	mA
I _{DD} ⁴	V _{DD} = 2.6 V, F = 200 kHz	—	67	—	μA
	V _{DD} = 2.6 V, F = 1.5 MHz	—	530	—	μA
	V _{DD} = 2.6 V, F = 25 MHz	—	8.0	15	mA
	V _{DD} = 2.6 V, F = 50 MHz	—	16	25	mA
I _{DD} Supply Sensitivity ⁴	F = 25 MHz	—	55	—	%/V
	F = 1 MHz	—	58	—	
I _{DD} Frequency Sensitivity ^{4,6}	V _{DD} = 2.1V, F ≤ 12.5 MHz, T = 25 °C	—	0.26	—	mA/MHz
	V _{DD} = 2.1V, F > 12.5 MHz, T = 25 °C	—	0.26	—	
	V _{DD} = 2.6V, F ≤ 12.5 MHz, T = 25 °C	—	0.34	—	
	V _{DD} = 2.6V, F > 12.5 MHz, T = 25 °C	—	0.34	—	
Digital Supply Current ⁴ (Stop or Suspend Mode)	Oscillator not running, V _{DD} Monitor Disabled				μA
	Temp = 25 °C	—	1	—	
	Temp = 60 °C	—	6	—	
	Temp= 125 °C	—	70	—	
Notes:					
1. Given in Table 5.4 on page 41.					
2. V _{IO} should not be lower than the V _{DD} voltage.					
3. SYSCLK must be at least 32 kHz to enable debugging.					
4. Guaranteed by characterization. Does not include oscillator supply current.					
5. IDD estimation for different frequencies.					
6. Idle IDD estimation for different frequencies.					

C8051F55x/56x/57x

Gain Register Definition 6.1. ADC0GNH: ADC0 Selectable Gain High Byte

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

Indirect Address = 0x04;

Bit	Name	Function
7:0	GAINH[7:0]	ADC0 Gain High Byte. See Section 6.3.1 for details on calculating the value for this register.
Note: This register is accessed indirectly; See Section 6.3.2 for details for writing this register.		

Gain Register Definition 6.2. ADC0GNL: ADC0 Selectable Gain Low Byte

Bit	7	6	5	4	3	2	1	0
Name	GAINL[3:0]				Reserved	Reserved	Reserved	Reserved
Type	W				W	W	W	W
Reset	0	0	0	0	0	0	0	0

Indirect Address = 0x07;

Bit	Name	Function
7:4	GAINL[3:0]	ADC0 Gain Lower 4 Bits. See Figure 6.3.1 for details for setting this register. This register is only accessed indirectly through the ADC0H and ADC0L register.
3:0	Reserved	Must Write 0000b
Note: This register is accessed indirectly; See Section 6.3.2 for details for writing this register.		

9. Voltage Regulator (REG0)

C8051F55x/56x/57x devices include an on-chip low dropout voltage regulator (REG0). The input to REG0 at the V_{REGIN} pin can be as high as 5.25 V. The output can be selected by software to 2.1 V or 2.6 V. When enabled, the output of REG0 appears on the V_{DD} pin, powers the microcontroller core, and can be used to power external devices. On reset, REG0 is enabled and can be disabled by software.

The Voltage regulator can generate an interrupt (if enabled by EREG0, EIE2.0) that is triggered whenever the V_{REGIN} input voltage drops below the dropout threshold voltage. This dropout interrupt has no pending flag and the recommended procedure to use it is as follows:

1. Wait enough time to ensure the V_{REGIN} input voltage is stable
2. Enable the dropout interrupt (EREG0, EIE2.0) and select the proper priority (PREG0, EIP2.0)
3. If triggered, inside the interrupt disable it (clear EREG0, EIE2.0), execute all procedures necessary to protect your application (put it in a safe mode and leave the interrupt now disabled).
4. In the main application, now running in the safe mode, regularly checks the DROPOUT bit (REG0CN.0). Once it is cleared by the regulator hardware the application can enable the interrupt again (EREG0, EIE1.6) and return to the normal mode operation.

The input (V_{REGIN}) and output (V_{DD}) of the voltage regulator should both be bypassed with a large capacitor (4.7 μF + 0.1 μF) to ground as shown in Figure 9.1. This capacitor will eliminate power spikes and provide any immediate power required by the microcontroller. The settling time associated with the voltage regulator is shown in Table 5.8 on page 43.

Note: The output of the internal voltage regulator is calibrated by the MCU immediately after any reset event. The output of the un-calibrated internal regulator could be below the high threshold setting of the V_{DD} Monitor. If this is the case *and* the V_{DD} Monitor is set to the high threshold setting *and* if the MCU receives a non-power on reset (POR), the MCU will remain in reset until a POR occurs (i.e., V_{DD} Monitor will keep the device in reset). A POR will force the V_{DD} Monitor to the low threshold setting which is guaranteed to be below the un-calibrated output of the internal regulator. The device will then exit reset and resume normal operation. It is for this reason Silicon Labs strongly recommends that the V_{DD} Monitor is always left in the low threshold setting (i.e. default value upon POR).

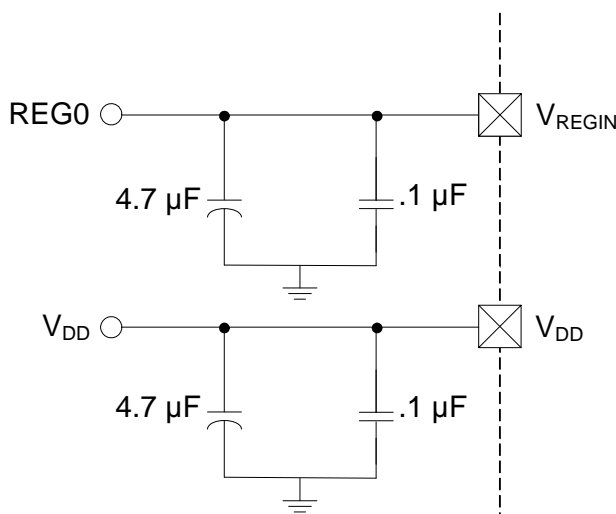


Figure 9.1. External Capacitors for Voltage Regulator Input/Output—Regulator Enabled

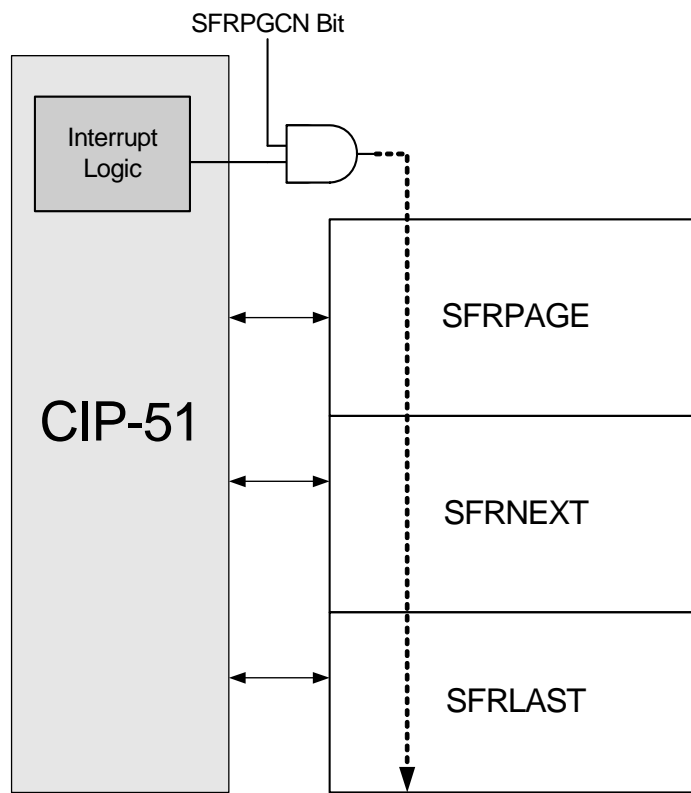


Figure 12.1. SFR Page Stack

Automatic hardware switching of the SFR Page on interrupts may be enabled or disabled as desired using the SFR Automatic Page Control Enable Bit located in the SFR Page Control Register (SFR0CN). This function defaults to “enabled” upon reset. In this way, the autoswitching function will be enabled unless disabled in software.

A summary of the SFR locations (address and SFR page) are provided in Table 12.3 in the form of an SFR memory map. Each memory location in the map has an SFR page row, denoting the page in which that SFR resides. Certain SFRs are accessible from ALL SFR pages, and are denoted by the “(ALL PAGES)” designation. For example, the Port I/O registers P0, P1, P2, and P3 all have the “(ALL PAGES)” designation, indicating these SFRs are accessible from all SFR pages regardless of the SFRPAGE register value.

While in the CAN0 ISR, a PCA interrupt occurs. Recall the PCA interrupt is configured as a high priority interrupt, while the CAN0 interrupt is configured as a *low* priority interrupt. Thus, the CIP-51 will now vector to the high priority PCA ISR. Upon doing so, the CIP-51 will automatically place the SFR page needed to access the PCA's special function registers into the SFRPAGE register, SFR Page 0x00. The value that was in the SFRPAGE register before the PCA interrupt (SFR Page 0x0C for CAN0) is pushed down the stack into SFRNEXT. Likewise, the value that was in the SFRNEXT register before the PCA interrupt (in this case SFR Page 0x00 for SPI0DAT) is pushed down to the SFRLAST register, the “bottom” of the stack. Note that a value stored in SFRLAST (via a previous software write to the SFRLAST register) will be overwritten. See Figure 12.4.

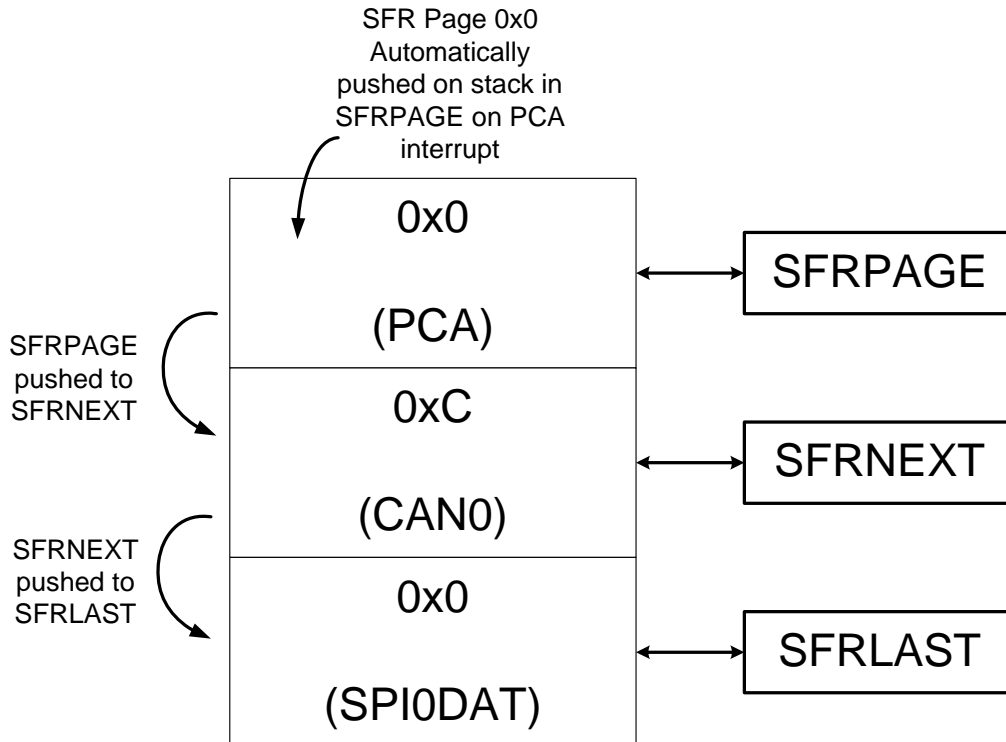


Figure 12.4. SFR Page Stack Upon PCA Interrupt Occurring During a CAN0 ISR

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>

Table 12.3. Special Function Registers (Continued)

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

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

C8051F55x/56x/57x

SFR Definition 13.7. IT01CF: INT0/INT1 Configuration

Bit	7	6	5	4	3	2	1	0
Name	IN1PL	IN1SL[2:0]			IN0PL	IN0SL[2:0]		
Type	R/W	R/W			R/W	R/W		
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE4; SFR Page = 0x0F

Bit	Name	Function
7	IN1PL	INT1 Polarity. 0: $\overline{\text{INT1}}$ input is active low. 1: INT1 input is active high.
6:4	IN1SL[2:0]	INT1 Port Pin Selection Bits. These bits select which Port pin is assigned to $\overline{\text{INT1}}$. Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT1}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P1.0 001: Select P1.1 010: Select P1.2 011: Select P1.3 100: Select P1.4 101: Select P1.5 110: Select P1.6 111: Select P1.7
3	IN0PL	INT0 Polarity. 0: $\overline{\text{INT0}}$ input is active low. 1: INT0 input is active high.
2:0	IN0SL[2:0]	INT0 Port Pin Selection Bits. These bits select which Port pin is assigned to $\overline{\text{INT0}}$. Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT0}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P1.0 001: Select P1.1 010: Select P1.2 011: Select P1.3 100: Select P1.4 101: Select P1.5 110: Select P1.6 111: Select P1.7

14.4. Flash Write and Erase Guidelines

Any system which contains routines which write or erase Flash memory from software involves some risk that the write or erase routines will execute unintentionally if the CPU is operating outside its specified operating range of V_{DD} , system clock frequency, or temperature. This accidental execution of Flash modifying code can result in alteration of Flash memory contents causing a system failure that is only recoverable by re-Flashing the code in the device.

The following guidelines are recommended for any system which contains routines which write or erase Flash from code.

14.4.1. V_{DD} Maintenance and the V_{DD} monitor

1. If the system power supply is subject to voltage or current "spikes," add sufficient transient protection devices to the power supply to ensure that the supply voltages listed in the Absolute Maximum Ratings table are not exceeded.
2. Make certain that the minimum VREGIN rise time specification of 1 ms is met. If the system cannot meet this rise time specification, then add an external V_{DD} brownout circuit to the RST pin of the device that holds the device in reset until V_{DD} reaches the minimum threshold and re-asserts RST if V_{DD} drops below the minimum threshold.
3. Enable the on-chip V_{DD} monitor in the high setting and enable the V_{DD} monitor as a reset source as early in code as possible. This should be the first set of instructions executed after the Reset Vector. For C-based systems, this will involve modifying the startup code added by the C compiler. See your compiler documentation for more details. Make certain that there are no delays in software between enabling the V_{DD} monitor in the high setting and enabling the V_{DD} monitor as a reset source. Code examples showing this can be found in "AN201: Writing to Flash from Firmware", available from the Silicon Laboratories web site.
4. As an added precaution, explicitly enable the V_{DD} monitor in the high setting and enable the V_{DD} monitor as a reset source inside the functions that write and erase Flash memory. The V_{DD} monitor enable instructions should be placed just after the instruction to set PSWE to a 1, but before the Flash write or erase operation instruction.

Note: The output of the internal voltage regulator is calibrated by the MCU immediately after any reset event. The output of the un-calibrated internal regulator could be below the high threshold setting of the V_{DD} Monitor. If this is the case *and* the V_{DD} Monitor is set to the high threshold setting *and* if the MCU receives a non-power on reset (POR), the MCU will remain in reset until a POR occurs (i.e., V_{DD} Monitor will keep the device in reset). A POR will force the V_{DD} Monitor to the low threshold setting which is guaranteed to be below the un-calibrated output of the internal regulator. The device will then exit reset and resume normal operation. It is for this reason Silicon Labs strongly recommends that the V_{DD} Monitor is always left in the low threshold setting (i.e. default value upon POR). When programming the Flash in-system, the V_{DD} Monitor must be set to the high threshold setting. For the highest system reliability, the time the V_{DD} Monitor is set to the high threshold setting should be minimized (e.g., setting the V_{DD} Monitor to the high threshold setting just before the Flash write operation and then changing it back to the low threshold setting immediately after the Flash write operation).

5. Make certain that all writes to the RSTSRC (Reset Sources) register use direct assignment operators and explicitly DO NOT use the bit-wise operators (such as AND or OR). For example, "RSTSRC = 0x02" is correct. "RSTSRC |= 0x02" is incorrect.
6. Make certain that all writes to the RSTSRC register explicitly set the PORSF bit to a 1. Areas to check are initialization code which enables other reset sources, such as the Missing Clock Detector or Comparator, for example, and instructions which force a Software Reset. A global search on "RSTSRC" can quickly verify this.

SFR Definition 15.1. PCON: Power Control

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

SFR Address = 0x87; SFR Page = All Pages

Bit	Name	Function
7:2	GF[5:0]	General Purpose Flags 5–0. These are general purpose flags for use under software control.
1	STOP	Stop Mode Select. Setting this bit will place the CIP-51 in Stop mode. This bit will always be read as 0. 1: CPU goes into Stop mode (internal oscillator stopped).
0	IDLE	IDLE: Idle Mode Select. Setting this bit will place the CIP-51 in Idle mode. This bit will always be read as 0. 1: CPU goes into Idle mode. (Shuts off clock to CPU, but clock to Timers, Interrupts, Serial Ports, and Analog Peripherals are still active.)

18. Oscillators and Clock Selection

C8051F55x/56x/57x devices include a programmable internal high-frequency oscillator, an external oscillator drive circuit, and a clock multiplier. The internal oscillator can be enabled/disabled and calibrated using the OSCICN, OSCICRS, and OSCIFIN registers, as shown in Figure 18.1. The system clock can be sourced by the external oscillator circuit or the internal oscillator. The clock multiplier can produce three possible base outputs which can be scaled by a programmable factor of 1, 2/3, 2/4 (or 1/2), 2/5, 2/6 (or 1/3), or 2/7: Internal Oscillator x 2, Internal Oscillator x 4, External Oscillator x 2, or External Oscillator x 4.

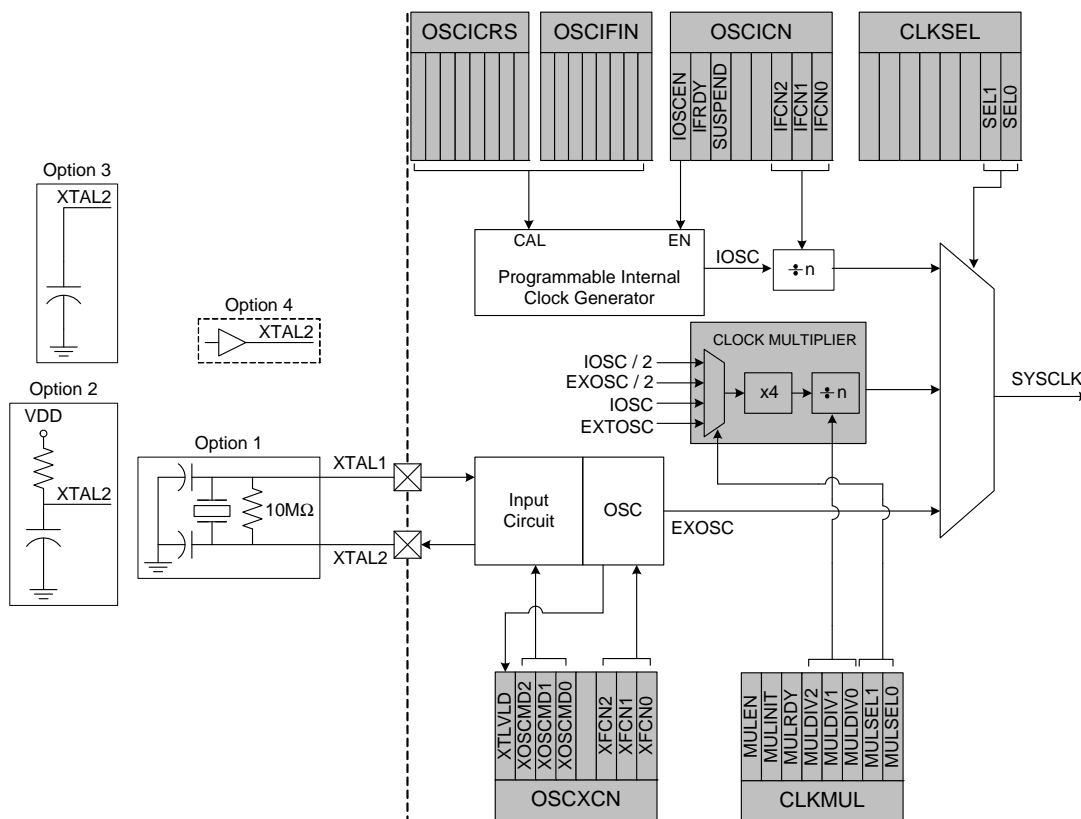


Figure 18.1. Oscillator Options

18.1. System Clock Selection

The CLKSL[1:0] bits in register CLKSEL select which oscillator source is used as the system clock. CLKSL[1:0] must be set to 01b for the system clock to run from the external oscillator; however the external oscillator may still clock certain peripherals (timers, PCA) when the internal oscillator is selected as the system clock. The system clock may be switched on-the-fly between the internal oscillator, external oscillator, and Clock Multiplier so long as the selected clock source is enabled and has settled.

The internal oscillator requires little start-up time and may be selected as the system clock immediately following the register write which enables the oscillator. The external RC and C modes also typically require no startup time.

External crystals and ceramic resonators however, typically require a start-up time before they are settled and ready for use. The Crystal Valid Flag (XTLVLD in register OSCXCN) is set to 1 by hardware when the external crystal or ceramic resonator is settled. **In crystal mode, to avoid reading a false XTLVLD, software should delay at least 1 ms between enabling the external oscillator and checking XTLVLD.**

When a digital I/O cell is placed in the high impedance state, a weak pull-up transistor pulls the Port pad to the VIO supply voltage to ensure the digital input is at a defined logic state. Weak pull-ups are disabled when the I/O cell is driven to GND to minimize power consumption and may be globally disabled by setting WEAKPUD to 1. The user should ensure that digital I/O are always internally or externally pulled or driven to a valid logic state to minimize power consumption. Port pins configured for digital I/O always read back the logic state of the Port pad, regardless of the output logic value of the Port pin.

19.1.3. Interfacing Port I/O in a Multi-Voltage System

All Port I/O are capable of interfacing to digital logic operating at a supply voltage higher than VDD and less than 5.25 V. Connect the VIO pin to the voltage source of the interface logic.

19.2. Assigning Port I/O Pins to Analog and Digital Functions

Port I/O pins P0.0–P3.7 can be assigned to various analog, digital, and external interrupt functions. P4.0 can be assigned to only digital functions. The Port pins assigned to analog functions should be configured for analog I/O, and Port pins assigned to digital or external interrupt functions should be configured for digital I/O.

19.2.1. Assigning Port I/O Pins to Analog Functions

Table 19.1 shows all available analog functions that require Port I/O assignments. **Port pins selected for these analog functions should have their corresponding bit in PnSKIP set to 1.** This reserves the pin for use by the analog function and does not allow it to be claimed by the Crossbar. Table 19.1 shows the potential mapping of Port I/O to each analog function.

Table 19.1. Port I/O Assignment for Analog Functions

Analog Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
ADC Input	P0.0–P3.7 ¹	ADC0MX, PnSKIP
Comparator0 or Comparator1 Input	P0.0–P2.7 ¹	CPT0MX, CPT1MX, PnSKIP
Voltage Reference (VREF0) ²	P0.0	REF0CN, PnSKIP
External Oscillator in Crystal Mode (XTAL1)	P0.2	OSCXCN, PnSKIP
External Oscillator in RC, C, or Crystal Mode (XTAL2)	P0.3	OSCXCN, PnSKIP
Notes: <ol style="list-style-type: none"> P3.1–P3.7 are available on the 40-pin packages. P2.2–P3.0 are available 40-pin and 32-pin packages. If VDD is selected as the voltage reference in the REF0CN register 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. 		

19.2.2. Assigning Port I/O Pins to Digital Functions

Any Port pins not assigned to analog functions may be assigned to digital functions or used as GPIO. Most digital functions rely on the Crossbar for pin assignment; however, some digital functions bypass the Crossbar in a manner similar to the analog functions listed above. **Port pins used by these digital functions and any Port pins selected for use as GPIO should have their corresponding bit in PnSKIP set to 1.** Table 19.2 shows all available digital functions and the potential mapping of Port I/O to each digital function.

21. Controller Area Network (CAN0)

Important Documentation Note: The Bosch CAN Controller is integrated in the C8051F550/1/4/5, 'F560/1/4/5/8/9, and 'F572/3 devices. This section of the data sheet gives a description of the CAN controller as an overview and offers a description of how the Silicon Labs CIP-51 MCU interfaces with the on-chip Bosch CAN controller. In order to use the CAN controller, refer to Bosch's C_CAN User's Manual as an accompanying manual to the Silicon Labs' data sheet.

The C8051F550/1/4/5, 'F560/1/4/5/8/9, and 'F572/3 devices feature a Control Area Network (CAN) controller that enables serial communication using the CAN protocol. Silicon Labs CAN 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 CIP-51 RAM), a message handler state machine, and control registers. Silicon Labs CAN is a protocol controller and does not provide physical layer drivers (i.e., transceivers). Figure 21.1 shows an example typical configuration on a CAN bus.

Silicon Labs' CAN operates at bit rates of up to 1 Mbit/second, though this can be limited by the physical layer chosen to transmit data on the CAN bus. The CAN processor has 32 Message Objects that can be configured to transmit or receive data. Incoming data, message objects and their 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 CIP-51 MCU. In this way, minimal CPU bandwidth is needed to use CAN communication. The CIP-51 configures the CAN controller, accesses received data, and passes data for transmission via Special Function Registers (SFRs) in the CIP-51.

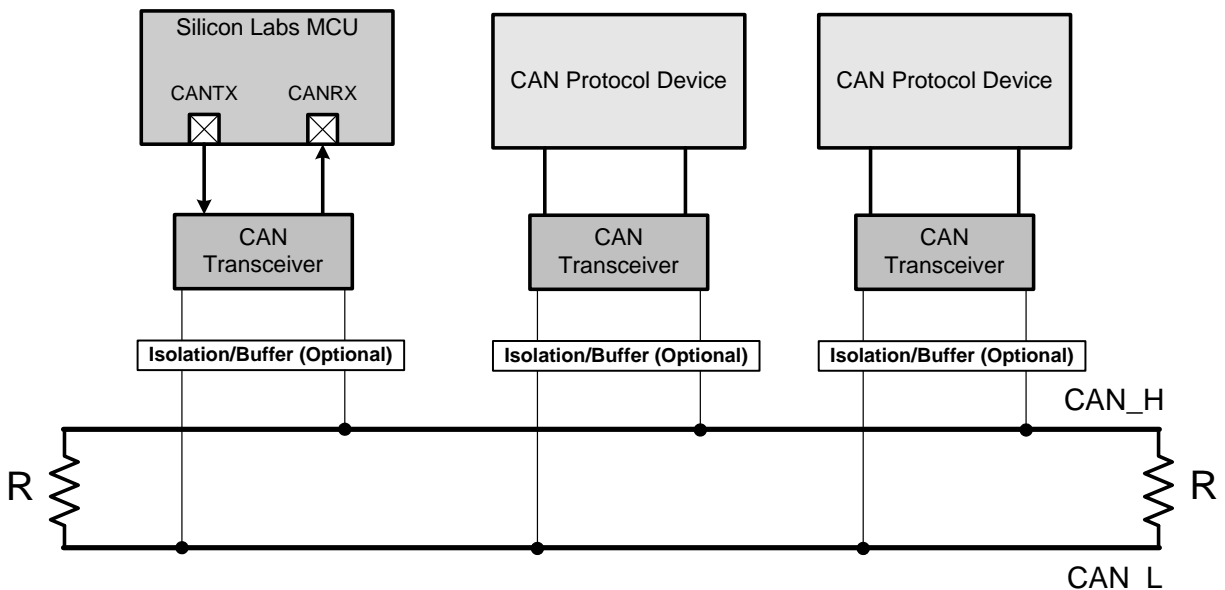


Figure 21.1. Typical CAN Bus Configuration

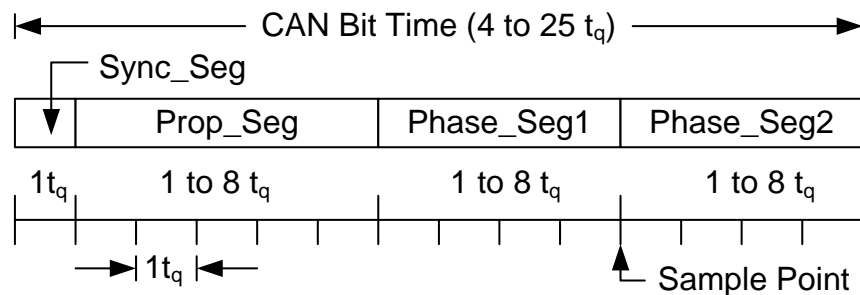


Figure 21.3. Four segments of a CAN Bit

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 $24 t_q$ (1000.008 ns), yielding a bit rate of 0.999992 Mbit/sec. The Sync_Seg is a constant $1 t_q$. The Prop_Seg must be greater than or equal to the propagation delay of 400 ns and so the choice is $10 t_q$ (416.67 ns).

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

$$\text{Phase_Seg1} + \text{Phase_Seg2} = \text{Bit_Time} - (\text{Sync_Seg} + \text{Prop_Seg})$$

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 21.1. 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.

$$\text{SJW} = \text{minimum}(4, \text{Phase_Seg1})$$

Equation 21.2. 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.

$$\text{BRPE} = \text{BRP} - 1 = \text{BRP Extension Register} = 0x0000$$

$$\text{SJWp} = \text{SJW} - 1 = \text{minimum}(4, 6) - 1 = 3$$

$$\text{TSEG1} = \text{Prop_Seg} + \text{Phase_Seg1} - 1 = 10 + 6 - 1 = 15$$

$$\text{TSEG2} = \text{Phase_Seg2} - 1 = 6$$

$$\text{Bit Timing Register} = (\text{TSEG2} \times 0x1000) + (\text{TSEG1} \times 0x0100)$$

$$\text{Bit Timing Register} = (\text{TSEG2} \times 0x1000) + (\text{TSEG1} \times 0x0100) + (\text{SJWp} \times 0x0040) + \text{BRPE} = 0x6FC0$$

Equation 21.3. Calculating the Bit Timing Register Value

Table 21.2. Standard CAN Registers and Reset Values

CAN Addr.	Name	SFR Name (High)	SFR Addr.	SFR Name (Low)	SFR Addr.	16-bit SFR	Reset Value
0x50	IF2 Data A 2	CAN0IF2DA2H	0xFB	CAN0IF2DA2L	0xFA	CAN0IF2DA2	0x0000
0x52	IF2 Data B 1	CAN0IF2DB1H	0xFD	CAN0IF2DB1L	0xFC	CAN0IF2DB1	0x0000
0x54	IF2 Data B 2	CAN0IF2DB2H	0xFF	CAN0IF2DB2L	0xFE	CAN0IF2DB2	0x0000
0x80	Transmission Request 1 ¹	CAN0TR1H	0xA3	CAN0TR1L	0xA2	CAN0TR1	0x0000
0x82	Transmission Request 2 ¹	CAN0TR2H	0xA5	CAN0TR2L	0xA4	CAN0TR2	0x0000
0x90	New Data 1 ¹	CAN0ND1H	0xAB	CAN0ND1L	0xAA	CAN0ND1	0x0000
0x92	New Data 2 ¹	CAN0ND2H	0xAD	CAN0ND2L	0xAC	CAN0ND2	0x0000
0xA0	Interrupt Pending 1 ¹	CAN0IP1H	0xAF	CAN0IP1L	0xAE	CAN0IP1	0x0000
0xA2	Interrupt Pending 2 ¹	CAN0IP2H	0xB3	CAN0IP2L	0xB2	CAN0IP2	0x0000
0xB0	Message Valid 1 ¹	CAN0MV1H	0xBB	CAN0MV1L	0xBA	CAN0MV1	0x0000
0xB2	Message Valid 2 ¹	CAN0MV2H	0xBD	CAN0MV2L	0xBC	CAN0MV2	0x0000

Notes:

1. Read-only register.
2. Write-enabled by CCE.
3. The reset value of CAN0TST could also be r0000000b, where r signifies the value of the CAN RX pin.
4. Write-enabled by Test.

22. SMBus

The SMBus I/O interface is a two-wire, bi-directional serial bus. The SMBus is compliant with the System Management Bus Specification, version 1.1, and compatible with the I2C serial bus. Reads and writes to the interface by the system controller are byte oriented with the SMBus interface autonomously controlling the serial transfer of the data. Data can be transferred at up to 1/20th of the system clock as a master or slave (this can be faster than allowed by the SMBus specification, depending on the system clock used). A method of extending the clock-low duration is available to accommodate devices with different speed capabilities on the same bus.

The SMBus interface may operate as a master and/or slave, and may function on a bus with multiple masters. The SMBus provides control of SDA (serial data), SCL (serial clock) generation and synchronization, arbitration logic, and START/STOP control and generation. A block diagram of the SMBus peripheral and the associated SFRs is shown in Figure 22.1.

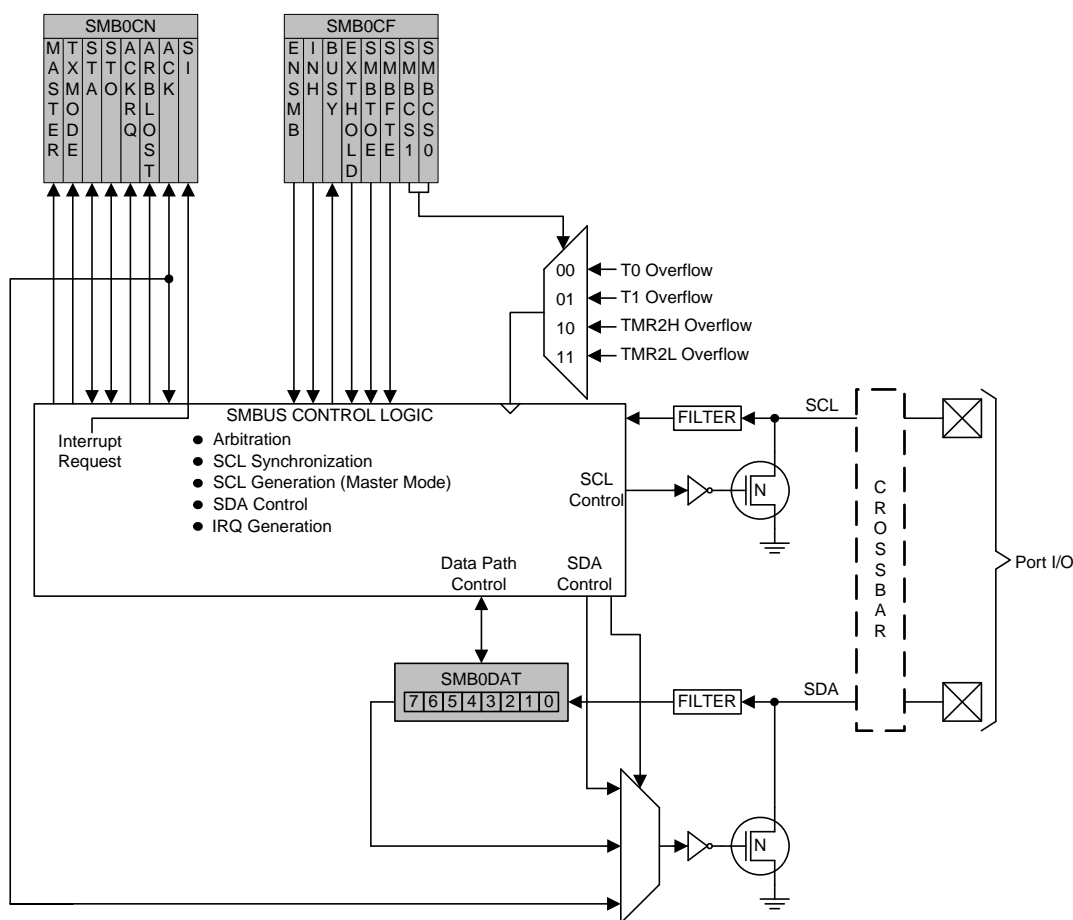


Figure 22.1. SMBus Block Diagram

SFR Definition 23.1. SCON0: Serial Port 0 Control

Bit	7	6	5	4	3	2	1	0
Name	OVR0	PERR0	THRE0	REN0	TBX0	RBX0	TI0	RI0
Type	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	0	0	0

Table 24.1. SPI Slave Timing Parameters

Parameter	Description	Min	Max	Units
Master Mode Timing * (See Figure 24.8 and Figure 24.9)				
T_{MCKH}	SCK High Time	$1 \times T_{SYSCLK}$	—	ns
T_{MCKL}	SCK Low Time	$1 \times T_{SYSCLK}$	—	ns
T_{MIS}	MISO Valid to SCK Shift Edge	$1 \times T_{SYSCLK} + 20$	—	ns
T_{MIH}	SCK Shift Edge to MISO Change	0	—	ns
Slave Mode Timing * (See Figure 24.10 and Figure 24.11)				
T_{SE}	NSS Falling to First SCK Edge	$2 \times T_{SYSCLK}$	—	ns
T_{SD}	Last SCK Edge to NSS Rising	$2 \times T_{SYSCLK}$	—	ns
T_{SEZ}	NSS Falling to MISO Valid	—	$4 \times T_{SYSCLK}$	ns
T_{SDZ}	NSS Rising to MISO High-Z	—	$4 \times T_{SYSCLK}$	ns
T_{CKH}	SCK High Time	$5 \times T_{SYSCLK}$	—	ns
T_{CKL}	SCK Low Time	$5 \times T_{SYSCLK}$	—	ns
T_{SIS}	MOSI Valid to SCK Sample Edge	$2 \times T_{SYSCLK}$	—	ns
T_{SIH}	SCK Sample Edge to MOSI Change	$2 \times T_{SYSCLK}$	—	ns
T_{SOH}	SCK Shift Edge to MISO Change	—	$4 \times T_{SYSCLK}$	ns
T_{SLH}	Last SCK Edge to MISO Change (CKPHA = 1 ONLY)	$6 \times T_{SYSCLK}$	$8 \times T_{SYSCLK}$	ns
*Note: T_{SYSCLK} is equal to one period of the device system clock (SYSCLK).				

C8051F55x/56x/57x

SFR Definition 25.3. TMOD: Timer Mode

Bit	7	6	5	4	3	2	1	0
Name	GATE1	C/T1	T1M[1:0]		GATE0	C/T0	T0M[1:0]	
Type	R/W	R/W	R/W		R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x89; SFR Page = All Pages

Bit	Name	Function
7	GATE1	Timer 1 Gate Control. 0: Timer 1 enabled when TR1 = 1 irrespective of $\overline{\text{INT1}}$ logic level. 1: Timer 1 enabled only when TR1 = 1 AND $\overline{\text{INT1}}$ is active as defined by bit IN1PL in register IT01CF (see SFR Definition 13.7).
6	C/T1	Counter/Timer 1 Select. 0: Timer: Timer 1 incremented by clock defined by T1M bit in register CKCON. 1: Counter: Timer 1 incremented by high-to-low transitions on external pin (T1).
5:4	T1M[1:0]	Timer 1 Mode Select. These bits select the Timer 1 operation mode. 00: Mode 0, 13-bit Counter/Timer 01: Mode 1, 16-bit Counter/Timer 10: Mode 2, 8-bit Counter/Timer with Auto-Reload 11: Mode 3, Timer 1 Inactive
3	GATE0	Timer 0 Gate Control. 0: Timer 0 enabled when TR0 = 1 irrespective of $\overline{\text{INT0}}$ logic level. 1: Timer 0 enabled only when TR0 = 1 AND $\overline{\text{INT0}}$ is active as defined by bit IN0PL in register IT01CF (see SFR Definition 13.7).
2	C/T0	Counter/Timer 0 Select. 0: Timer: Timer 0 incremented by clock defined by T0M bit in register CKCON. 1: Counter: Timer 0 incremented by high-to-low transitions on external pin (T0).
1:0	T0M[1:0]	Timer 0 Mode Select. These bits select the Timer 0 operation mode. 00: Mode 0, 13-bit Counter/Timer 01: Mode 1, 16-bit Counter/Timer 10: Mode 2, 8-bit Counter/Timer with Auto-Reload 11: Mode 3, Two 8-bit Counter/Timers