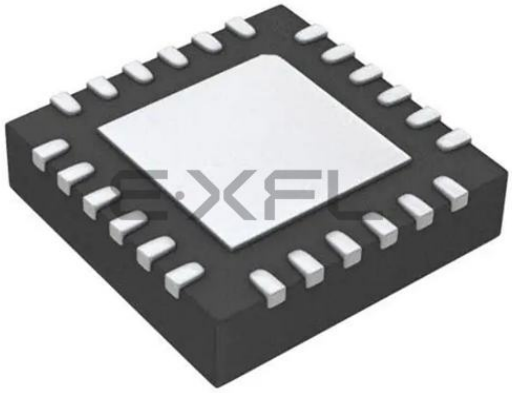

Welcome to E-XFL.COM

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	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	50MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	21
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 20x10b; D/A 2x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	24-WFQFN Exposed Pad
Supplier Device Package	24-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f374-a-gmr

1. System Overview

C8051F39x/37x devices are fully integrated mixed-signal System-on-a-Chip MCUs. Highlighted features are listed below. Refer to Section “2. Ordering Information” on page 20 for specific product feature selection and part ordering numbers.

- High-speed pipelined 8051-compatible microcontroller core (up to 50 MIPS)
- In-system, full-speed, non-intrusive debug interface (on-chip)
- True 10-bit 500 ksps 20 or 16-channel single-ended/differential ADC with analog multiplexer
- Two 10-bit Current Output DACs
- Precision temperature sensor with ±2 °C absolute accuracy
- Precision programmable 49 MHz internal oscillator
- Low-power, low-frequency oscillator
- 16 kB of on-chip Flash memory
- 1024 bytes of on-chip RAM
- Co-packaged with 512 bytes of EEPROM memory, accessible via I^2C (C8051F37x)
- Two SMBus/I^2C, UART, and SPI serial interfaces implemented in hardware
- Six general-purpose 16-bit timers
- Programmable Counter/Timer Array (PCA) with three capture/compare modules and Watchdog Timer function
- On-chip Power-On Reset, V_{DD} Monitor, and Temperature Sensor
- On-chip Voltage Comparator
- 21 or 17 Port I/O
- Low-power suspend mode with fast wake-up time

With on-chip Power-On Reset, V_{DD} monitor, Watchdog Timer, and clock oscillator, the C8051F39x/37x devices are truly stand-alone System-on-a-Chip solutions. The Flash memory can be reprogrammed even in-circuit, providing non-volatile data storage, and also allowing field upgrades of the 8051 firmware. User software has complete control of all peripherals, and may individually shut down any or all peripherals for power savings.

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

The C8051F37x devices are specified for 1.8 to 3.6 V operation over the industrial temperature range (–40 to +85 °C), while the C8051F39x devices operate over an extended temperature range (-40 to +105 °C). The C8051F392/3/6/7/8/9 are available in a 20-pin QFN package and the C8051F390/1/4/5 and C8051F37x are available in a 24-pin QFN package. Both package options are lead-free and RoHS compliant. See Section “2. Ordering Information” on page 20 for ordering information. Block diagrams are included in Figure 1.1, Figure 1.2 and Figure 1.3.

Table 7.4. Reset Electrical Characteristics

–40 to +105 °C (C8051F39x), –40 to +85 °C (C8051F37x), unless otherwise specified.

Parameter	Test Condition	Min	Typ	Max	Unit
$\overline{\text{RST}}$ Output Low Voltage	I_{OL} = 4 mA, V_{DD} = 1.8 to 3.6 V	—	—	0.6	V
$\overline{\text{RST}}$ Input Low Voltage		—	—	0.6	V
$\overline{\text{RST}}$ Input Pullup Current	$\overline{\text{RST}}$ = 0.0 V	—	20	100	µA
V_{DD} POR Threshold (V_{RST})	V_{RST_LOW}	1.7	1.75	1.8	V
	V_{RST_HIGH}	2.4	2.55	2.7	V
Missing Clock Detector Time-out	Time from last system clock rising edge to reset initiation	80	580	800	µs
Reset Time Delay	Delay between release of any reset source and code execution at location 0x0000	—	—	40	µs
Minimum $\overline{\text{RST}}$ Low Time to Generate a System Reset		15	—	—	µs
V_{DD} Monitor Turn-on Time		100	—	—	µs
V_{DD} Monitor Supply Current		—	20	50	µA

SFR Definition 9.1. ADC0CF: ADC0 Configuration

Bit	7	6	5	4	3	2	1	0
Name	AD0SC[4:0]					AD0LJST		
Type	R/W					R/W	R/W	
Reset	1	1	1	1	1	0	0	0

SFR Address = 0xBC; SFR Page = All Pages

Bit	Name	Function
7:3	AD0SC[4:0]	**ADC0 SAR Conversion Clock Period Bits.** SAR Conversion clock is derived from system clock by the following equation, where *AD0SC* refers to the 5-bit value held in bits AD0SC4–0. SAR Conversion clock requirements are given in the ADC specification Table 7.10. $AD0SC = \frac{SYSCLK}{CLK_{SAR}} - 1$
2	AD0LJST	**ADC0 Left Justify Select.** 0: Data in ADC0H:ADC0L registers are right-justified. 1: Data in ADC0H:ADC0L registers are left-justified.
1:0	Reserved	Must Write 00b.

SFR Definition 9.4. ADC0CN: ADC0 Control

Bit	7	6	5	4	3	2	1	0
Name	AD0EN	AD0TM	AD0INT	AD0BUSY	AD0WINT	AD0CM[2:0]		
Type	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE8; SFR Page = All Pages; Bit-Addressable

Bit	Name	Function
7	AD0EN	**ADC0 Enable Bit.** 0: ADC0 Disabled. ADC0 is in low-power shutdown. 1: ADC0 Enabled. ADC0 is active and ready for data conversions.
6	AD0TM	**ADC0 Track Mode Bit.** 0: Normal Track Mode: When ADC0 is enabled, tracking is continuous unless a conversion is in progress. Conversion begins immediately on start-of-conversion event, as defined by AD0CM[2:0]. 1: Delayed Track Mode: When ADC0 is enabled, input is tracked when a conversion is not in progress. A start-of-conversion signal initiates three SAR clocks of additional tracking, and then begins the conversion. Note that there is not a tracking delay when CNVSTR is used (AD0CM[2:0] = 100).
5	AD0INT	**ADC0 Conversion Complete Interrupt Flag.** 0: ADC0 has not completed a data conversion since AD0INT was last cleared. 1: ADC0 has completed a data conversion.
4	AD0BUSY	**ADC0 Busy Bit.**
3	AD0WINT	**ADC0 Window Compare Interrupt Flag.** 0: ADC0 Window Comparison Data match has not occurred since this flag was last cleared. 1: ADC0 Window Comparison Data match has occurred.
2:0	AD0CM[2:0]	**ADC0 Start of Conversion Mode Select.** 000: ADC0 start-of-conversion source is write of 1 to AD0BUSY. 001: ADC0 start-of-conversion source is overflow of Timer 0. 010: ADC0 start-of-conversion source is overflow of Timer 2. 011: ADC0 start-of-conversion source is overflow of Timer 1. 100: ADC0 start-of-conversion source is rising edge of external CNVSTR. 101: ADC0 start-of-conversion source is overflow of Timer 3. 110: ADC0 start-of-conversion source is overflow of Timer 4. 111: ADC0 start-of-conversion source is overflow of Timer 5.

SFR Definition 11.1. IDA0CN: IDA0 Control

Bit	7	6	5	4	3	2	1	0
Name	IDA0EN	IDA0CM[2:0]				IDA0RP	IDA0OMD[1:0]	
Type	R/W	R/W			R	R/W	R/W	
Reset	0	1	1	1	0	Varies	1	0

SFR Address = 0xB9; SFR Page = 0

Bit	Name	Function
7	IDA0EN	**IDA0 Enable.** 0: IDA0 Disabled. 1: IDA0 Enabled.
6:4	IDA0CM[2:0]	**IDA0 Update Source Select bits.** 000: DAC output updates on Timer 0 overflow. 001: DAC output updates on Timer 1 overflow. 010: DAC output updates on Timer 2 overflow. 011: DAC output updates on Timer 3 overflow. 100: DAC output updates on rising edge of CNVSTR. 101: DAC output updates on falling edge of CNVSTR. 110: DAC output updates on any edge of CNVSTR. 111: DAC output updates on write to IDA0H.
3	Reserved	Write = 0b.
2	IDA0RP	**IDA0 Reset Persistence.** 0: IDA0 is disabled by any reset source. 1: IDA0 will remain enabled through any reset source except a power-on-reset. This bit is reset to 0 by a power on reset, but is sticky through all other reset sources. When setting IDA0RP to 1, IDA0EN must be set to 1 also in the same mov instruction.
1:0	IDA0OMD[1:0]	**IDA0 Output Mode Select bits.** 00: 0.5 mA full-scale output current. 01: 1.0 mA full-scale output current. 1x: 2.0 mA full-scale output current.

SFR Definition 12.1. REF0CN: Reference Control

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

SFR Address = 0xD1; SFR Page = All Pages

Bit	Name	Function
7	REFBGS	**Reference Buffer Gain Select.** This bit selects between 1x and 2x gain for the on-chip voltage reference buffer. 0: 2x Gain 1: 1x Gain
6:5	Unused	Read = 00b; Write = Don't care.
4	REGOVR	**Regulator Reference Override.** This bit "overrides" the REFSL bit, and allows the internal regulator to be used as a reference source. 0: The voltage reference source is selected by the REFSL bit. 1: The internal regulator is used as the voltage reference.
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.
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.

With the CIP-51's maximum system clock at 48 MHz, it has a peak throughput of 48 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	2/4	3	3/5	4	5	4/6	6	8
Number of Instructions	26	50	5	10	7	5	2	1	2	1

Programming and Debugging Support
In-system programming of the EPROM 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 Section “33. C2 Interface” on page 297.

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.

15.1. Instruction Set

The instruction set of the CIP-51 System Controller is fully compatible with the standard MCS-51™ instruction set. Standard 8051 development tools can be used to develop software for the CIP-51. All CIP-51 instructions are the binary and functional equivalent of their MCS-51™ counterparts, including opcodes, addressing modes and effect on PSW flags. However, instruction timing is different than that of the standard 8051.

15.1.1. Instruction and CPU Timing

In many 8051 implementations, a distinction is made between machine cycles and clock cycles, with machine cycles varying from 2 to 12 clock cycles in length. However, the CIP-51 implementation is based solely on clock cycle timing. All instruction timings are specified in terms of clock cycles.

Due to the pipelined architecture of the CIP-51, most instructions execute in the same number of clock cycles as there are program bytes in the instruction. Conditional branch instructions take one less clock cycle to complete when the branch is not taken as opposed to when the branch is taken. Table 15.1 is the CIP-51 Instruction Set Summary, which includes the mnemonic, number of bytes, and number of clock cycles for each instruction.

SFR Definition 15.3. 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 15.4. 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.

21.4. 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.

21.4.1. 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 V_{DD} 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 $\overline{RST}$ pin of the device that holds the device in reset until V_{DD} reaches 2.7 V and re-asserts $\overline{RST}$ if V_{DD} drops below 2.7 V.
3. Enable the on-chip V_{DD} monitor 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 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 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.
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.

21.4.2. PSWE Maintenance

7. Reduce the number of places in code where the PSWE bit (b0 in PSCTL) is set to a '1'. There should be exactly one routine in code that sets PSWE to a '1' to write Flash bytes and one routine in code that sets PSWE and PSEE both to a '1' to erase Flash pages.
8. Minimize the number of variable accesses while PSWE is set to a '1'. Handle pointer address updates and loop variable maintenance outside the "PSWE = 1;... PSWE = 0;" area. Code examples showing this can be found in “AN201: Writing to Flash from Firmware", available from the Silicon Laboratories web site.
9. Disable interrupts prior to setting PSWE to a '1' and leave them disabled until after PSWE has been reset to 0. Any interrupts posted during the Flash write or erase operation will be serviced in priority order after the Flash operation has been completed and interrupts have been re-enabled by software.
10. Make certain that the Flash write and erase pointer variables are not located in XRAM. See your compiler documentation for instructions regarding how to explicitly locate variables in different memory areas.
11. Add address bounds checking to the routines that write or erase Flash memory to ensure that a routine called with an illegal address does not result in modification of the Flash.

23.2. Preparing for a CRC Calculation

To prepare CRC0 for a CRC calculation, software should set the initial value of the result. The polynomial used for the CRC computation is 0x1021. The CRC0 result may be initialized to one of two values: 0x0000 or 0xFFFF. The following steps can be used to initialize CRC0.

1. Select the initial result value (Set CRC0VAL to 0 for 0x0000 or 1 for 0xFFFF).
2. Set the result to its initial value (Write 1 to CRC0INIT).

23.3. Performing a CRC Calculation

Once CRC0 is initialized, the input data stream is sequentially written to CRC0IN, one byte at a time. The CRC0 result is automatically updated after each byte is written. The CRC engine may also be configured to automatically perform a CRC on one or more 256 byte blocks read from Flash. The following steps can be used to automatically perform a CRC on Flash memory.

1. Prepare CRC0 for a CRC calculation as shown above.
2. Write the index of the starting page to CRC0AUTO.
3. Set the AUTOEN bit to 1 in CRC0AUTO.
4. Write the number of 256 byte blocks to perform in the CRC calculation to CRC0CNT.
5. Write any value to CRC0CN (or OR its contents with 0x00) to initiate the CRC calculation. The CPU will not execute code any additional code until the CRC operation completes. **See the note in SFR Definition 23.1. CRC0CN: CRC0 Control for more information on how to properly initiate a CRC calculation.**
6. Clear the AUTOEN bit in CRC0AUTO.
7. Read the CRC result using the procedure below.

23.4. Accessing the CRC0 Result

The internal CRC0 result is 16 bits. The CRC0PNT bits select the byte that is targeted by read and write operations on CRC0DAT and increment after each read or write. The calculation result will remain in the internal CR0 result register until it is set, overwritten, or additional data is written to CRC0IN.

23.5. CRC0 Bit Reverse Feature

CRC0 includes hardware to reverse the bit order of each bit in a byte as shown in Figure 23.2. Each byte of data written to CRC0FLIP is read back bit reversed. For example, if 0xC0 is written to CRC0FLIP, the data read back is 0x03. Bit reversal is a useful mathematical function used in algorithms such as the FFT.

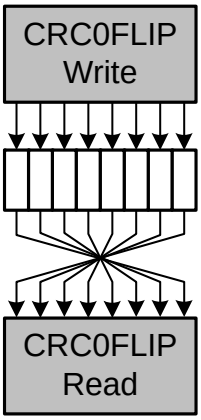

Figure 23.2. Bit Reverse Register

24. Reset Sources

Reset circuitry allows the controller to be easily placed in a predefined default condition. Upon entering this reset state, the following events occur:

- CIP-51 halts program execution
- Special Function Registers (SFRs) are initialized to their defined reset values
- External Port pins are forced to a known state
- Interrupts and timers are disabled.

All SFRs are reset to the predefined values noted in the SFR detailed descriptions. The contents of internal data memory are unaffected during a reset; any previously stored data is preserved. However, since the stack pointer SFR is reset, the stack is effectively lost, even though the data on the stack is not altered.

The Port I/O latches are reset to 0xFF (all logic ones) in open-drain mode. Weak pullups are enabled during and after the reset. For V_{DD} Monitor and power-on resets, the $\overline{RST}$ pin is driven low until the device exits the reset state.

On exit from the reset state, the program counter (PC) is reset, and the system clock defaults to the internal oscillator. The Watchdog Timer is enabled with the system clock divided by 12 as its clock source. Program execution begins at location 0x0000.

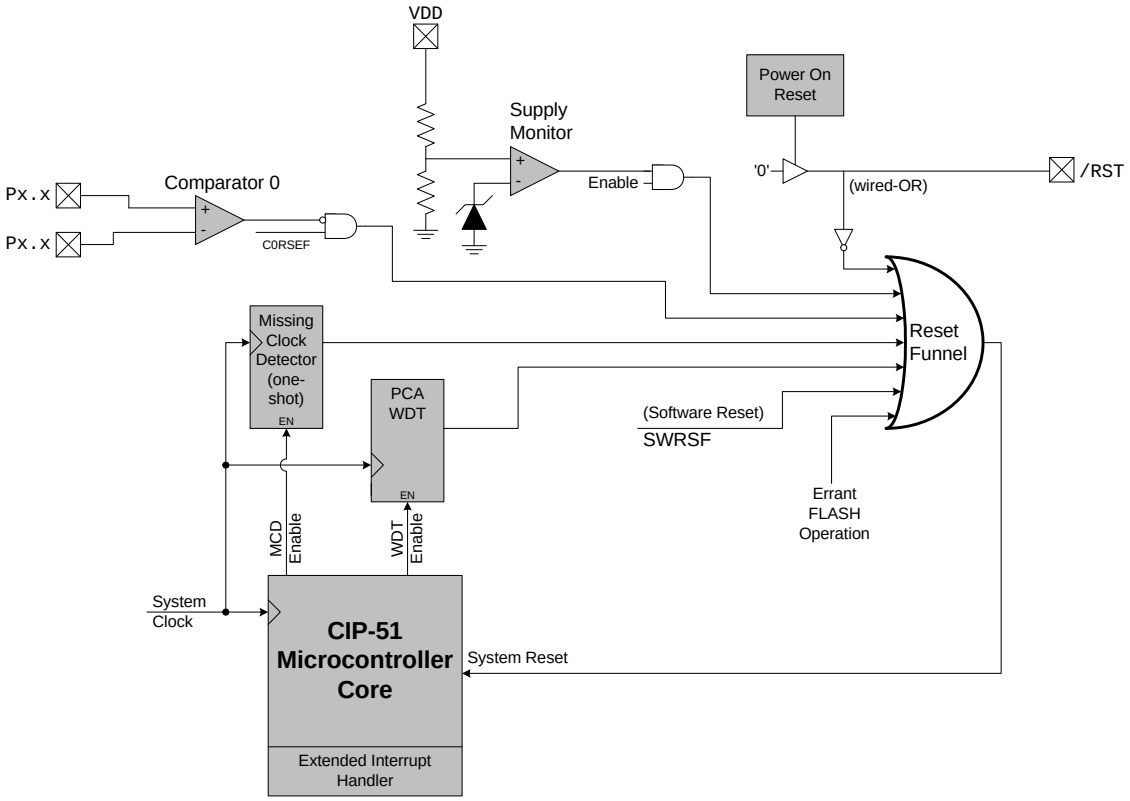

Figure 24.1. Reset Sources

SFR Definition 24.2. RSTSRC: Reset Source

Bit	7	6	5	4	3	2	1	0
Name		FERROR	C0RSEF	SWRSF	WDTRSF	MCDRSF	PORSF	PINRSF
Type	R	R	R/W	R/W	R	R/W	R/W	R
Reset	0	Varies	Varies	Varies	Varies	Varies	Varies	Varies

SFR Address = 0xEF; SFR Page = All Pages

Bit	Name	Description	Write	Read
7	Unused	**Unused.**	Don't care.	0
6	FERROR	**Flash Error Reset Flag.**	N/A	Set to '1' if Flash read/ write/erase error caused the last reset.
5	C0RSEF	**Comparator0 Reset Enable and Flag.**	Writing a '1' enables Comparator0 as a reset source (active-low).	Set to '1' if Comparator0 caused the last reset.
4	SWRSF	**Software Reset Force and Flag.**	Writing a '1' forces a system reset.	Set to '1' if last reset was caused by a write to SWRSF.
3	WDTRSF	**Watchdog Timer Reset Flag.**	N/A	Set to '1' if Watchdog Timer overflow caused the last reset.
2	MCDRSF	**Missing Clock Detector Enable and Flag.**	Writing a '1' enables the Missing Clock Detector. The MCD triggers a reset if a missing clock condition is detected.	Set to '1' if Missing Clock Detector timeout caused the last reset.
1	PORSF	**Power-On / V_{DD} Monitor Reset Flag, and V_{DD} monitor Reset Enable.**	Writing a '1' enables the V_{DD} monitor as a reset source. **Writing '1' to this bit before the V_{DD} monitor is enabled and stabilized may cause a system reset.**	Set to '1' anytime a power-on or V_{DD} monitor reset occurs. **When set to '1' all other RSTSRC flags are indeterminate.**
0	PINRSF	**HW Pin Reset Flag.**	N/A	Set to '1' if $\overline{\text{RST}}$ pin caused the last reset.

Note: Do not use read-modify-write operations on this register

26.1. System Clock Selection

The CLKSL[2:0] bits in register CLKSEL select which oscillator source is used as the system clock. CLKSL[2:0] must be set to 001b 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 any of the oscillator sources so long as the selected clock source is enabled and has settled.

The internal high-frequency and low-frequency oscillators require 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.**

SFR Definition 26.1. CLKSEL: Clock Select

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

SFR Address = 0xA9; SFR Page = All Pages

Bit	Name	Function
7:3	Unused	Read = 00000b; Write = Don't Care
2:0	CLKSL[2:0]	**System Clock Source Select Bits.** 000: SYSCLK derived from the Internal High-Frequency Oscillator and scaled per the IFCN bits in register OSCICN. 001: SYSCLK derived from the External Oscillator circuit.* 010: SYSCLK derived from the Internal Low-Frequency Oscillator and scaled per the OSCLD bits in register OSCLCN. 011: SYSCLK derived directly from the Internal High-Frequency Oscillator.* 100: Reserved. 101: SYSCLK derived from the Internal Low-Power Oscillator. 110: Reserved. 111: Reserved.
Note: Prior to switching to a system clock frequency > 25 MHz, ensure that the FLRT bit in SFR Definition 21.3. FLSCL: Flash Scale has been set appropriately to ensure proper flash read timing.		

SFR Definition 28.7. SMB0ADM: SMBus0 Slave Address Mask

Bit	7	6	5	4	3	2	1	0
Name	SLVM0[6:0]							EHACK0
Type	R/W							R/W
Reset	1	1	1	1	1	1	1	0

SFR Address = 0xE7; SFR Page = 0

Bit	Name	Function
7:1	SLVM0[6:0]	**SMBus0 Slave Address Mask.** Defines which bits of register SMB0ADR are compared with an incoming address byte, and which bits are ignored. Any bit set to 1 in SLVM0[6:0] enables comparisons with the corresponding bit in SLV0[6:0]. Bits set to 0 are ignored (can be either 0 or 1 in the incoming address).
0	EHACK0	**Hardware Acknowledge Enable.** Enables hardware acknowledgement of slave address and received data bytes. 0: Firmware must manually acknowledge all incoming address and data bytes. 1: Automatic Slave Address Recognition and Hardware Acknowledge is Enabled.

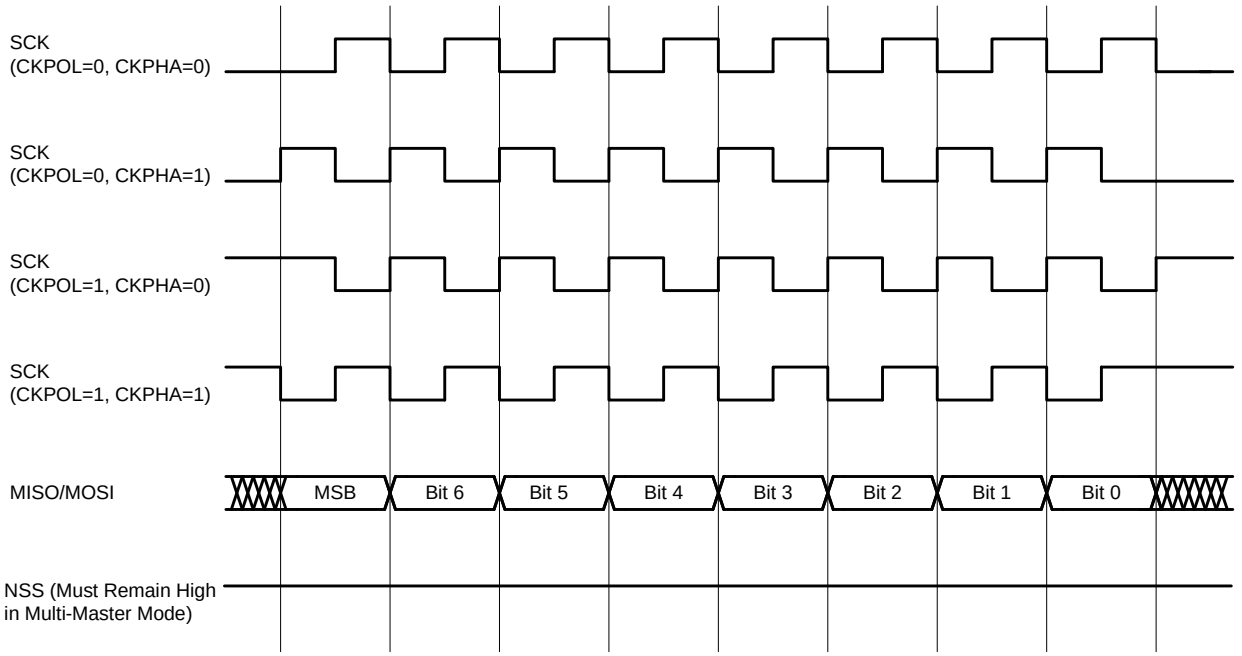

Figure 30.5. Master Mode Data/Clock Timing

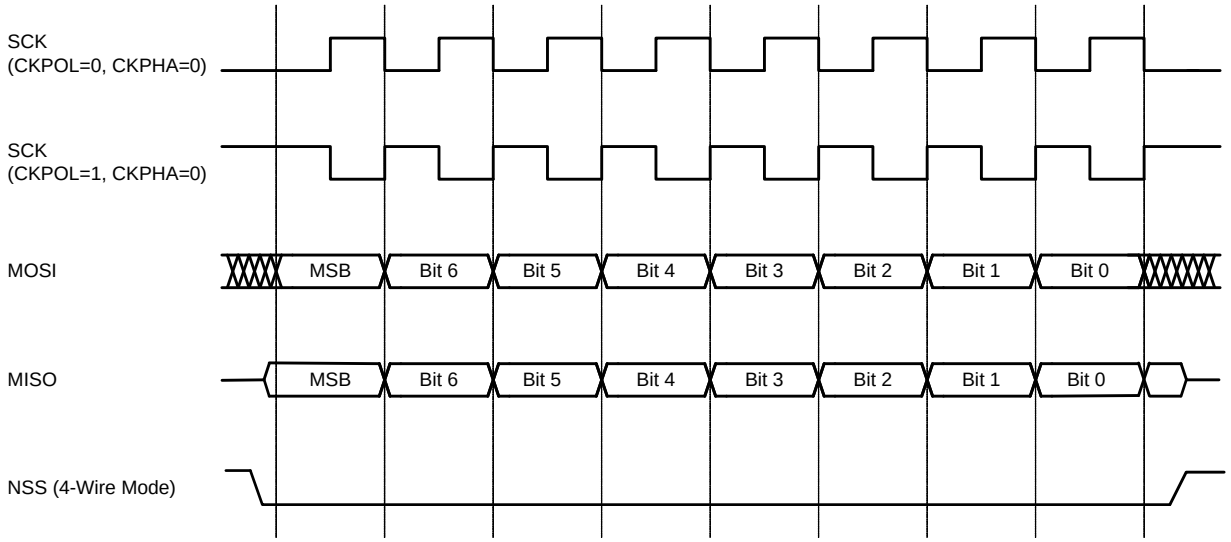

Figure 30.6. Slave Mode Data/Clock Timing (CKPHA = 0)

SFR Definition 30.1. SPI0CFG: SPI0 Configuration

Bit	7	6	5	4	3	2	1	0
Name	SPIBSY	MSTEN	CKPHA	CKPOL	SLVSEL	NSSIN	SRMT	RXBMT
Type	R	R/W	R/W	R/W	R	R	R	R
Reset	0	0	0	0	0	1	1	1

SFR Address = 0xA1; SFR Page = All Pages

Bit	Name	Function
7	SPIBSY	**SPI Busy.** This bit is set to logic 1 when a SPI transfer is in progress (master or slave mode).
6	MSTEN	**Master Mode Enable.** 0: Disable master mode. Operate in slave mode. 1: Enable master mode. Operate as a master.
5	CKPHA	**SPI0 Clock Phase.** 0: Data centered on first edge of SCK period.* 1: Data centered on second edge of SCK period.*
4	CKPOL	**SPI0 Clock Polarity.** 0: SCK line low in idle state. 1: SCK line high in idle state.
3	SLVSEL	**Slave Selected Flag.** This bit is set to logic 1 whenever the NSS pin is low indicating SPI0 is the selected slave. It is cleared to logic 0 when NSS is high (slave not selected). This bit does not indicate the instantaneous value at the NSS pin, but rather a de-glitched version of the pin input.
2	NSSIN	**NSS Instantaneous Pin Input.** This bit mimics the instantaneous value that is present on the NSS port pin at the time that the register is read. This input is not de-glitched.
1	SRMT	**Shift Register Empty (valid in slave mode only).** This bit will be set to logic 1 when all data has been transferred in/out of the shift register, and there is no new information available to read from the transmit buffer or write to the receive buffer. It returns to logic 0 when a data byte is transferred to the shift register from the transmit buffer or by a transition on SCK. SRMT = 1 when in Master Mode.
0	RXBMT	**Receive Buffer Empty (valid in slave mode only).** This bit will be set to logic 1 when the receive buffer has been read and contains no new information. If there is new information available in the receive buffer that has not been read, this bit will return to logic 0. RXBMT = 1 when in Master Mode.

Note: In slave mode, data on MOSI is sampled in the center of each data bit. In master mode, data on MISO is sampled one SYSCLK before the end of each data bit, to provide maximum settling time for the slave device. See Table 30.1 for timing parameters.

31.2.2. 8-bit Timers with Auto-Reload

When T2SPLIT is set, Timer 2 operates as two 8-bit timers (TMR2H and TMR2L). Both 8-bit timers operate in auto-reload mode as shown in Figure 31.5. TMR2RLL holds the reload value for TMR2L; TMR2RLH holds the reload value for TMR2H. The TR2 bit in TMR2CN handles the run control for TMR2H. TMR2L 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 2 Clock Select bits (T2MH and T2ML in CKCON) select either SYSCLK or the clock defined by the Timer 2 External Clock Select bit (T2XCLK in TMR2CN), as follows:

T2MH	T2XCLK	TMR2H Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

T2ML	T2XCLK	TMR2L Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

The TF2H bit is set when TMR2H overflows from 0xFF to 0x00; the TF2L bit is set when TMR2L overflows from 0xFF to 0x00. When Timer 2 interrupts are enabled (IE.5), an interrupt is generated each time TMR2H overflows. If Timer 2 interrupts are enabled and TF2LEN (TMR2CN.5) is set, an interrupt is generated each time either TMR2L or TMR2H overflows. When TF2LEN is enabled, software must check the TF2H and TF2L flags to determine the source of the Timer 2 interrupt. The TF2H and TF2L interrupt flags are not cleared by hardware and must be manually cleared by software.

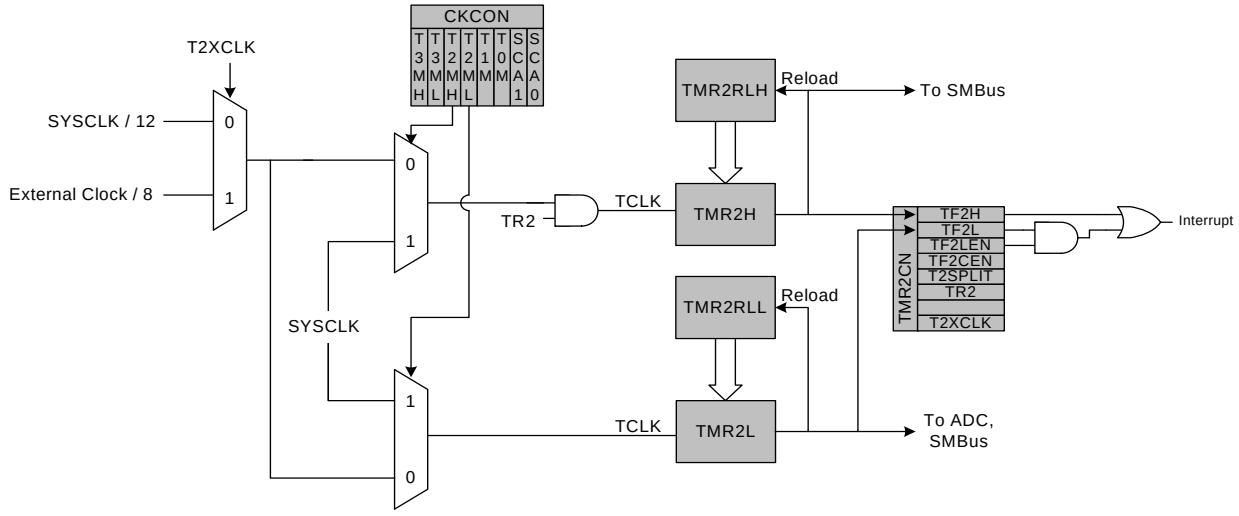

Figure 31.5. Timer 2 8-Bit Mode Block Diagram

SFR Definition 31.9. TMR2CN: Timer 2 Control

Bit	7	6	5	4	3	2	1	0
Name	TF2H	TF2L	TF2LEN	TF2CEN	T2SPLIT	TR2		T2XCLK
Type	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC8; SFR Page = 0; Bit-Addressable

Bit	Name	Function
7	TF2H	**Timer 2 High Byte Overflow Flag.** Set by hardware when the Timer 2 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 2 overflows from 0xFFFF to 0x0000. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 interrupt service routine. This bit is not automatically cleared by hardware.
6	TF2L	**Timer 2 Low Byte Overflow Flag.** Set by hardware when the Timer 2 low byte overflows from 0xFF to 0x00. TF2L will be set when the low byte overflows regardless of the Timer 2 mode. This bit is not automatically cleared by hardware.
5	TF2LEN	**Timer 2 Low Byte Interrupt Enable.** When set to 1, this bit enables Timer 2 Low Byte interrupts. If Timer 2 interrupts are also enabled, an interrupt will be generated when the low byte of Timer 2 overflows.
4	TF2CEN	**Timer 2 Low-Frequency Oscillator Capture Enable.** When set to 1, this bit enables Timer 2 Low-Frequency Oscillator Capture Mode. If TF2CEN is set and Timer 2 interrupts are enabled, an interrupt will be generated on a falling edge of the low-frequency oscillator output, and the current 16-bit timer value in TMR2H:TMR2L will be copied to TMR2RLH:TMR2RLL.
3	T2SPLIT	**Timer 2 Split Mode Enable.** When this bit is set, Timer 2 operates as two 8-bit timers with auto-reload. 0: Timer 2 operates in 16-bit auto-reload mode. 1: Timer 2 operates as two 8-bit auto-reload timers.
2	TR2	**Timer 2 Run Control.** Timer 2 is enabled by setting this bit to 1. In 8-bit mode, this bit enables/disables TMR2H only; TMR2L is always enabled in split mode.
1	Unused	Unused. Read = 0b; Write = Don't Care
0	T2XCLK	**Timer 2 External Clock Select.** This bit selects the external clock source for Timer 2. If Timer 2 is in 8-bit mode, this bit selects the external oscillator clock source for both timer bytes. However, the Timer 2 Clock Select bits (T2MH and T2ML in register CKCON) may still be used to select between the external clock and the system clock for either timer. 0: Timer 2 clock is the system clock divided by 12. 1: Timer 2 clock is the external clock divided by 8 (synchronized with SYSCLK).

SFR Definition 31.28. TMR5H Timer 5 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR5H[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCD; SFR Page = F

Bit	Name	Function
7:0	TMR5H[7:0]	**Timer 5 High Byte.** In 16-bit mode, the TMR5H register contains the high byte of the 16-bit Timer 5. In 8-bit mode, TMR5H contains the 8-bit high byte timer value.