



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	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I ² C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	16
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	0.9V ~ 3.6V
Data Converters	A/D 15x10/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	24-SSOP (0.154", 3.90mm Width)
Supplier Device Package	24-QSOP
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f902-gur

13.5.3. System Clock	138
13.6. Minimizing Flash Read Current	139
14. Power Management.....	143
14.1. Normal Mode	144
14.2. Idle Mode	145
14.3. Stop Mode	145
14.4. Suspend Mode	146
14.5. Sleep Mode	146
14.6. Configuring Wakeup Sources.....	147
14.7. Determining the Event that Caused the Last Wakeup.....	148
14.8. Power Management Specifications	151
15. Cyclic Redundancy Check Unit (CRC0)	152
15.1. CRC Algorithm.....	152
15.2. Preparing for a CRC Calculation	154
15.3. Performing a CRC Calculation	154
15.4. Accessing the CRC0 Result	154
15.5. CRC0 Bit Reverse Feature.....	159
16. On-Chip DC-DC Converter (DC0)	160
16.1. Startup Behavior.....	161
16.2. High Power Applications	162
16.3. Pulse Skipping Mode.....	162
16.4. Enabling the DC-DC Converter	163
16.5. Minimizing Power Supply Noise	164
16.6. Selecting the Optimum Switch Size.....	164
16.7. DC-DC Converter Clocking Options	164
16.8. DC-DC Converter Behavior in Sleep Mode	164
16.9. Bypass Mode (C8051F912/02 only)	165
16.10. Low Power Mode (C8051F912/02 only)	165
16.11. Passive Diode Mode (C8051F912/02 only).....	166
16.12. DC-DC Converter Register Descriptions	167
16.13. DC-DC Converter Specifications	169
17. Voltage Regulator (VREG0)	170
17.1. Voltage Regulator Electrical Specifications	170
18. Reset Sources.....	171
18.1. Power-On (VBAT Supply Monitor) Reset	172
18.2. Power-Fail (VDD/DC+ Supply Monitor) Reset.....	173
18.3. External Reset	176
18.4. Missing Clock Detector Reset	176
18.5. Comparator0 Reset	176
18.6. PCA Watchdog Timer Reset	176
18.7. Flash Error Reset	177
18.8. SmaRTClock (Real Time Clock) Reset	177
18.9. Software Reset.....	177
19. Clocking Sources	179
19.1. Programmable Precision Internal Oscillator	180

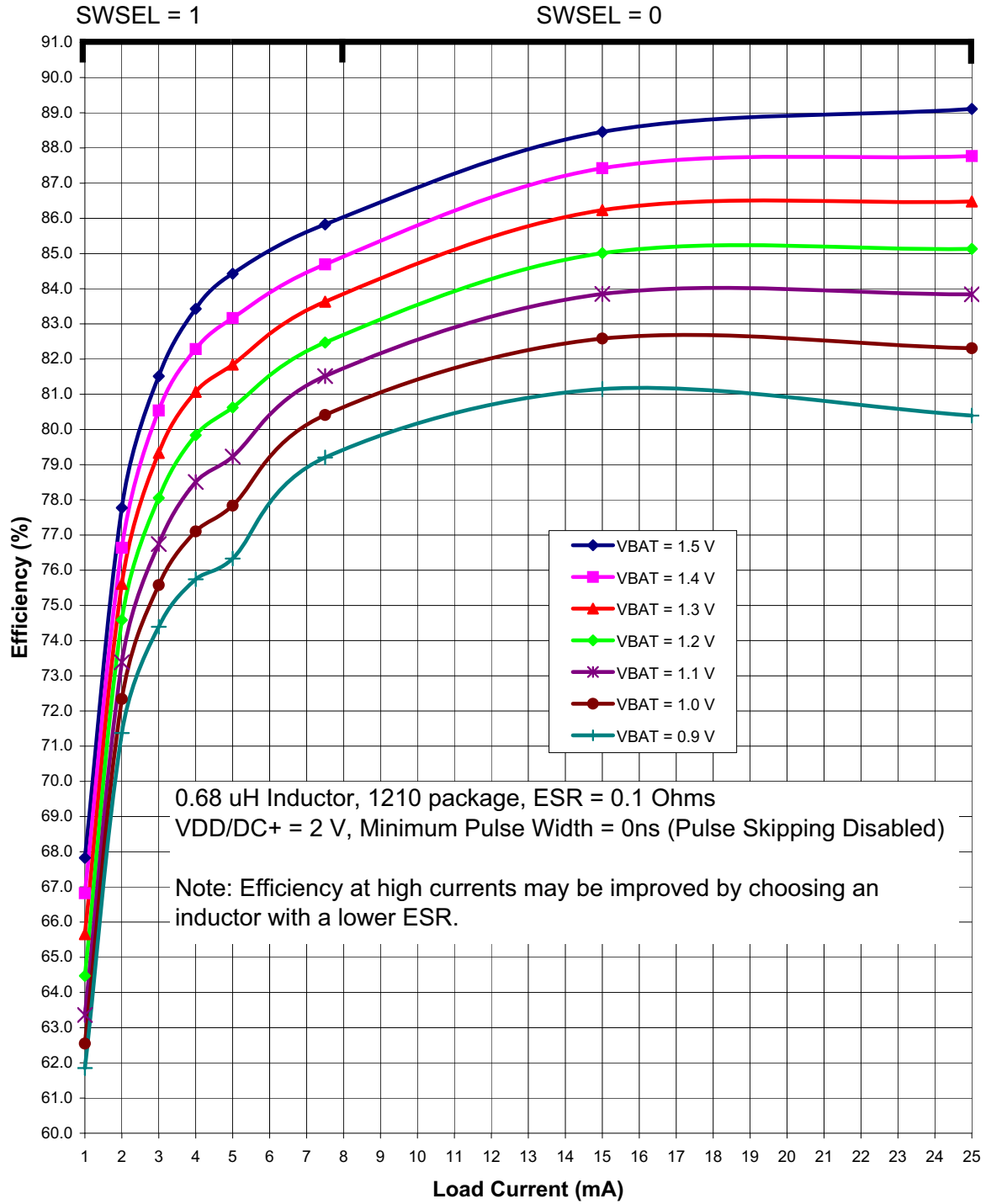


Figure 4.3. Typical DC-DC Converter Efficiency (High Current, VDD/DC+ = 2 V)

C8051F91x-C8051F90x

Table 4.4. Reset Electrical Characteristics

$V_{DD} = 1.8$ to 3.6 V, -40 to $+85$ °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
\overline{RST} Output Low Voltage	$I_{OL} = 1.4$ mA,	—	—	0.6	V
\overline{RST} Input High Voltage	$V_{DD} = 2.0$ to 3.6 V	$V_{DD} - 0.6$	—	—	V
	$V_{DD} = 0.9$ to 2.0 V	$0.7 \times V_{DD}$	—	—	V
\overline{RST} Input Low Voltage	$V_{DD} = 2.0$ to 3.6 V	—	—	0.6	V
	$V_{DD} = 0.9$ to 2.0 V	—	—	$0.3 \times V_{DD}$	V
\overline{RST} Input Pullup Current	$\overline{RST} = 0.0$ V, $V_{DD} = 1.8$ V	—	4	—	μ A
	$\overline{RST} = 0.0$ V, $V_{DD} = 3.6$ V	—	20	35	
VDD/DC+ Monitor Threshold (V_{RST})	Early Warning	1.8	1.85	1.9	V
	Reset Trigger (all power modes except Sleep)	1.7	1.75	1.8	
VBAT Ramp Time for Power On	VBAT Ramp from 0–0.9 V	—	—	3	ms
VBAT Monitor Threshold (V_{POR})	Initial Power-On (VBAT Rising)	—	0.75	—	V
	Early Warning	0.9	1.0	1.1	
	Brownout Condition (VBAT Falling)	0.7	0.8	0.9	
	Recovery from Brownout (VBAT Rising)	—	0.95	—	
Missing Clock Detector Timeout	Time from last system clock rising edge to reset initiation	100	525	1000	μ s
Minimum System Clock w/ Missing Clock Detector Enabled	System clock frequency which triggers a missing clock detector timeout	—	2	10	kHz
Reset Time Delay	Delay between release of any reset source and code execution at location 0x0000	—	10	—	μ s
Minimum \overline{RST} Low Time to Generate a System Reset		15	—	—	μ s
V_{DD} Monitor Turn-on Time		—	300	—	ns
V_{DD} Monitor Supply Current		—	10	—	μ A
*Note: Blue indicates a feature only available on 'F912 and 'F902 devices.					

C8051F91x-C8051F90x

SFR Definition 5.2. ADC0CF: ADC0 Configuration

Bit	7	6	5	4	3	2	1	0
Name	AD0SC[4:0]					AD08BE	AD0TM	AMP0GN
Type	R/W					R/W	R/W	R/W
Reset	1	1	1	1	1	0	0	0

SFR Page = 0x0; SFR Address = 0xBC

Bit	Name	Function
7:3	AD0SC[4:0]	ADC0 SAR Conversion Clock Divider. SAR Conversion clock is derived from FCLK by the following equation, where AD0SC refers to the 5-bit value held in bits AD0SC[4:0]. SAR Conversion clock requirements are given in Table 4.10. BURSTEN = 0: FCLK is the current system clock. BURSTEN = 1: FCLK is the 20 MHz low power oscillator, independent of the system clock. $AD0SC = \frac{FCLK}{CLK_{SAR}} - 1 *$ *Round the result up. or $CLK_{SAR} = \frac{FCLK}{AD0SC + 1}$
2	AD08BE	ADC0 8-Bit Mode Enable. 0: ADC0 operates in 10-bit mode (normal operation). 1: ADC0 operates in 8-bit mode.
1	AD0TM	ADC0 Track Mode. Selects between Normal or Delayed Tracking Modes. 0: Normal Track Mode: When ADC0 is enabled, conversion begins immediately following the start-of-conversion signal. 1: Delayed Track Mode: When ADC0 is enabled, conversion begins 3 SAR clock cycles following the start-of-conversion signal. The ADC is allowed to track during this time.
0	AMP0GN	ADC0 Gain Control. 0: The on-chip PGA gain is 0.5. 1: The on-chip PGA gain is 1.

7.3. Comparator Response Time

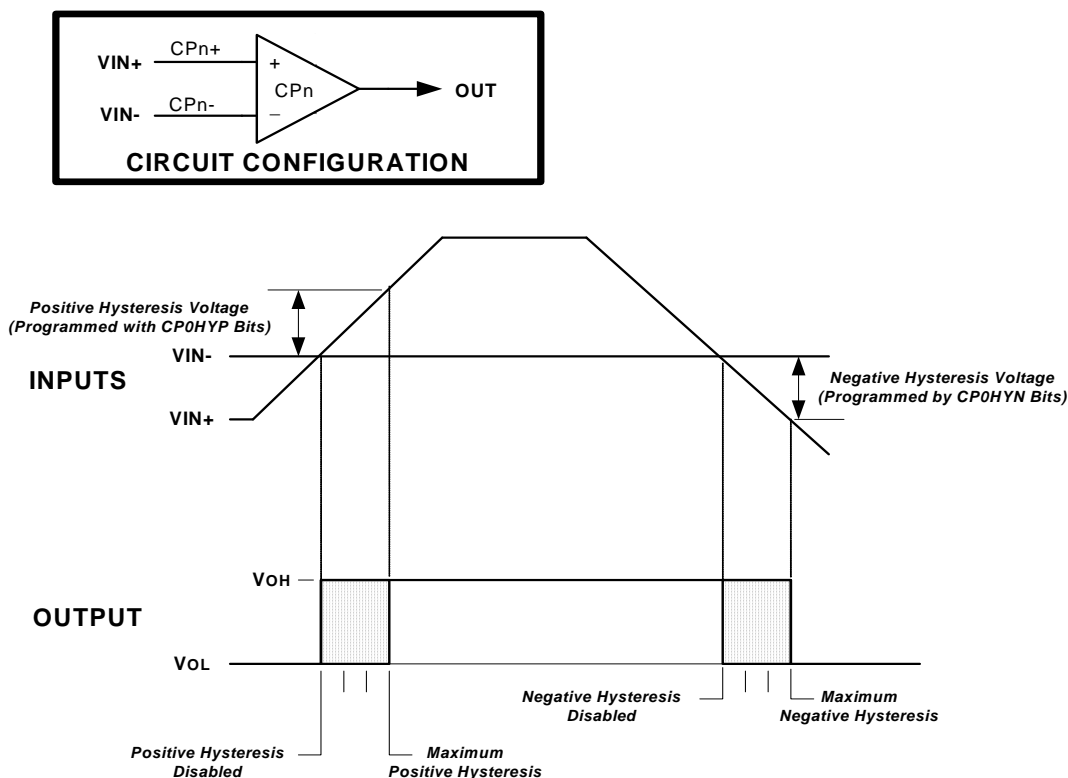
Comparator response time may be configured in software via the CPTnMD registers described on “CPT0MD: Comparator 0 Mode Selection” on page 92 and “CPT1MD: Comparator 1 Mode Selection” on page 94. Four response time settings are available: Mode 0 (Fastest Response Time), Mode 1, Mode 2, and Mode 3 (Lowest Power). Selecting a longer response time reduces the Comparator active supply current. The Comparators also have low power shutdown state, which is entered any time the comparator is disabled. Comparator rising edge and falling edge response times are typically not equal. See Table 4.14 on page 58 for complete comparator timing and supply current specifications.

7.4. Comparator Hysteresis

The Comparators feature software-programmable hysteresis that can be used to stabilize the comparator output while a transition is occurring on the input. Using the CPTnCN registers, the user can program both the amount of hysteresis voltage (referred to the input voltage) and the positive and negative-going symmetry of this hysteresis around the threshold voltage (i.e., the comparator negative input).

Figure 7.3 shows that when positive hysteresis is enabled, the comparator output does not transition from logic 0 to logic 1 until the comparator positive input voltage has exceeded the threshold voltage by an amount equal to the programmed hysteresis. It also shows that when negative hysteresis is enabled, the comparator output does not transition from logic 1 to logic 0 until the comparator positive input voltage has fallen below the threshold voltage by an amount equal to the programmed hysteresis.

The amount of positive hysteresis is determined by the settings of the CPnHYP bits in the CPTnCN register and the amount of negative hysteresis voltage is determined by the settings of the CPnHYN bits in the same register. Settings of 20, 10, 5, or 0 mV can be programmed for both positive and negative hysteresis. See Section “Table 4.14. Comparator Electrical Characteristics” on page 58 for complete comparator hysteresis specifications.



C8051F91x-C8051F90x

9.2.1.3. Stack

A programmer's stack can be located anywhere in the 256-byte data memory. The stack area is designated using the Stack Pointer (SP) SFR. The SP will point to the last location used. The next value pushed on the stack is placed at SP+1 and then SP is incremented. A reset initializes the stack pointer to location 0x07. Therefore, the first value pushed on the stack is placed at location 0x08, which is also the first register (R0) of register bank 1. Thus, if more than one register bank is to be used, the SP should be initialized to a location in the data memory not being used for data storage. The stack depth can extend up to 256 bytes.

9.2.2. External RAM

There are 512 bytes of on-chip RAM mapped into the external data memory space. All of these address locations may be accessed using the external move instruction (MOVX) and the data pointer (DPTR), or using MOVX indirect addressing mode (such as @R1) in combination with the EMI0CN register.

12. Interrupt Handler

The C8051F91x-C8051F90x microcontroller family includes an extended interrupt system supporting multiple interrupt sources and two priority levels. The allocation of interrupt sources between on-chip peripherals and external input pins varies according to the specific version of the device. Refer to Table 12.1, “Interrupt Summary,” on page 122 for a detailed listing of all interrupt sources supported by the device. Refer to the data sheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

Each interrupt source has one or more associated interrupt-pending flag(s) located in an SFR or an indirect register. When a peripheral or external source meets a valid interrupt condition, the associated interrupt-pending flag is set to logic 1. If both global interrupts and the specific interrupt source is enabled, a CPU interrupt request is generated when the interrupt-pending flag is set.

As soon as execution of the current instruction is complete, the CPU generates an LCALL to a predetermined address to begin execution of an interrupt service routine (ISR). Each ISR must end with an RETI instruction, which returns program execution to the next instruction that would have been executed if the interrupt request had not occurred. If interrupts are not enabled, the interrupt-pending flag is ignored by the hardware and program execution continues as normal. (The interrupt-pending flag is set to logic 1 regardless of the interrupt's enable/disable state.)

Some interrupt-pending flags are automatically cleared by hardware when the CPU vectors to the ISR. However, most are not cleared by the hardware and must be cleared by software before returning from the ISR. If an interrupt-pending flag remains set after the CPU completes the return-from-interrupt (RETI) instruction, a new interrupt request will be generated immediately and the CPU will re-enter the ISR after the completion of the next instruction.

12.1. Enabling Interrupt Sources

Each interrupt source can be individually enabled or disabled through the use of an associated interrupt enable bit in the Interrupt Enable and Extended Interrupt Enable SFRs. However, interrupts must first be globally enabled by setting the EA bit (IE.7) to logic 1 before the individual interrupt enables are recognized. Setting the EA bit to logic 0 disables all interrupt sources regardless of the individual interrupt-enable settings. Note that interrupts which occur when the EA bit is set to logic 0 will be held in a pending state, and will not be serviced until the EA bit is set back to logic 1.

12.2. MCU Interrupt Sources and Vectors

The CPU services interrupts by generating an LCALL to a predetermined address (the interrupt vector address) to begin execution of an interrupt service routine (ISR). The interrupt vector addresses associated with each interrupt source are listed in Table 12.1 on page 122. Software should ensure that the interrupt vector for each enabled interrupt source contains a valid interrupt service routine.

Software can simulate an interrupt by setting any interrupt-pending flag to logic 1. If interrupts are enabled for the flag, an interrupt request will be generated and the CPU will vector to the ISR address associated with the interrupt-pending flag.

13. Flash Memory

On-chip, re-programmable Flash memory is included for program code and non-volatile data storage. The Flash memory can be programmed in-system through the C2 interface or by software using the MOVX write instruction. Once cleared to logic 0, a Flash bit must be erased to set it back to logic 1. Flash bytes would typically be erased (set to 0xFF) before being reprogrammed. The write and erase operations are automatically timed by hardware for proper execution; data polling to determine the end of the write/erase operations is not required. Code execution is stalled during Flash write/erase operations. Refer to Table 4.6 for complete Flash memory electrical characteristics.

13.1. Programming The Flash Memory

The simplest means of programming the Flash memory is through the C2 interface using programming tools provided by Silicon Laboratories or a third party vendor. This is the only means for programming a non-initialized device. For details on the C2 commands to program Flash memory, see Section “27. C2 Interface” on page 312.

The Flash memory can be programmed by software using the MOVX write instruction with the address and data byte to be programmed provided as normal operands. Before programming Flash memory using MOVX, Flash programming operations must be enabled by: (1) setting the PSWE Program Store Write Enable bit (PSCTL.0) to logic 1 (this directs the MOVX writes to target Flash memory); and (2) Writing the Flash key codes in sequence to the Flash Lock register (FLKEY). The PSWE bit remains set until cleared by software. For detailed guidelines on programming Flash from firmware, please see Section “13.5. Flash Write and Erase Guidelines” on page 137.

To ensure the integrity of the Flash contents, the on-chip VDD Monitor must be enabled and enabled as a reset source in any system that includes code that writes and/or erases Flash memory from software. Furthermore, there should be no delay between enabling the VDD Monitor and enabling the VDD Monitor as a reset source. Any attempt to write or erase Flash memory while the VDD Monitor is disabled, or not enabled as a reset source, will cause a Flash Error device reset.

13.1.1. Flash Lock and Key Functions

Flash writes and erases by user software are protected with a lock and key function. The Flash Lock and Key Register (FLKEY) must be written with the correct key codes, in sequence, before Flash operations may be performed. The key codes are: 0xA5, 0xF1. The timing does not matter, but the codes must be written in order. If the key codes are written out of order, or the wrong codes are written, Flash writes and erases will be disabled until the next system reset. Flash writes and erases will also be disabled if a Flash write or erase is attempted before the key codes have been written properly. The Flash lock resets after each write or erase; the key codes must be written again before a following Flash operation can be performed. The FLKEY register is detailed in SFR Definition 13.2.

C8051F91x-C8051F90x

13.3. Security Options

The CIP-51 provides security options to protect the Flash memory from inadvertent modification by software as well as to prevent the viewing of proprietary program code and constants. The Program Store Write Enable (bit PSWE in register PSCTL) and the Program Store Erase Enable (bit PSEE in register PSCTL) bits protect the Flash memory from accidental modification by software. PSWE must be explicitly set to 1 before software can modify the Flash memory; both PSWE and PSEE must be set to 1 before software can erase Flash memory. Additional security features prevent proprietary program code and data constants from being read or altered across the C2 interface.

A Security Lock Byte located at the last byte of Flash user space offers protection of the Flash program memory from access (reads, writes, or erases) by unprotected code or the C2 interface. The Flash security mechanism allows the user to lock n 512-byte Flash pages, starting at page 0 (addresses 0x0000 to 0x01FF), where n is the 1s complement number represented by the Security Lock Byte. **Note that the page containing the Flash Security Lock Byte is unlocked when no other Flash pages are locked (all bits of the Lock Byte are 1) and locked when any other Flash pages are locked (any bit of the Lock Byte is 0).**

Security Lock Byte:	1111 1011b
ones Complement:	0000 0100b
Flash pages locked:	5 (First four Flash pages + Lock Byte Page)
Addresses locked:	0x0000 to 0x07FF (first four Flash pages) and 0x3A00 to 0x3BFF (Lock Byte Page)

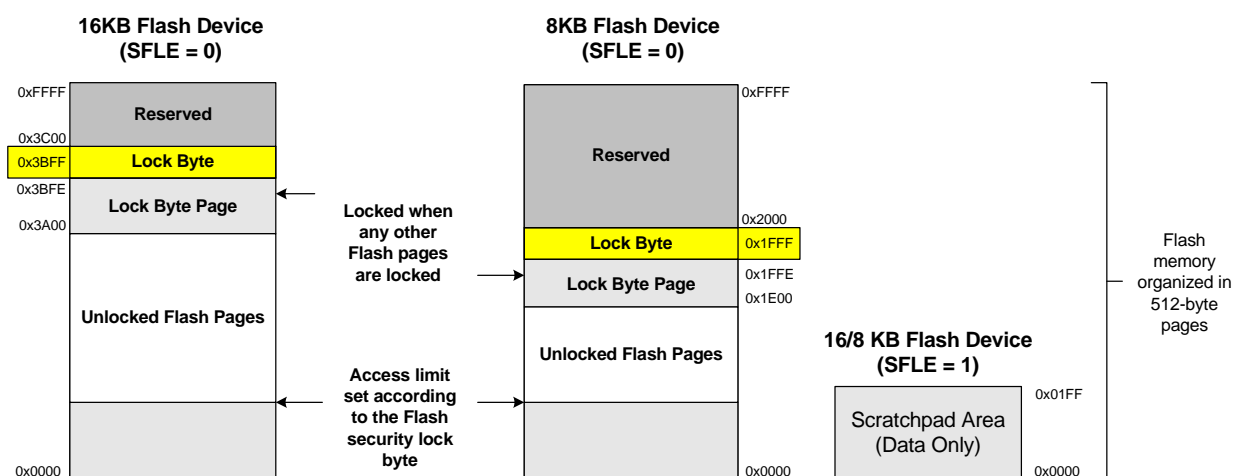


Figure 13.1. Flash Program Memory Map (16 kB and 8 kB devices)

The level of Flash security depends on the Flash access method. The three Flash access methods that can be restricted are reads, writes, and erases from the C2 debug interface, user firmware executing on unlocked pages, and user firmware executing on locked pages. Table 13.1 summarizes the Flash security features of the C8051F91x-C8051F90x devices.

Table 13.1. Flash Security Summary

Action	C2 Debug Interface	User Firmware executing from:	
		an unlocked page	a locked page
Read, Write or Erase unlocked pages (except page with Lock Byte)	Permitted	Permitted	Permitted
Read, Write or Erase locked pages (except page with Lock Byte)	Not Permitted	FEDR	Permitted
Read or Write page containing Lock Byte (if no pages are locked)	Permitted	Permitted	Permitted
Read or Write page containing Lock Byte (if any page is locked)	Not Permitted	FEDR	Permitted
Read contents of Lock Byte (if no pages are locked)	Permitted	Permitted	Permitted
Read contents of Lock Byte (if any page is locked)	Not Permitted	FEDR	Permitted
Erase page containing Lock Byte (if no pages are locked)	Permitted	FEDR	FEDR
Erase page containing Lock Byte - Unlock all pages (if any page is locked)	Only by C2DE	FEDR	FEDR
Lock additional pages (change 1s to 0s in the Lock Byte)	Not Permitted	FEDR	FEDR
Unlock individual pages (change 0s to 1s in the Lock Byte)	Not Permitted	FEDR	FEDR
Read, Write or Erase Reserved Area	Not Permitted	FEDR	FEDR
<p>C2DE—C2 Device Erase (Erases all Flash pages including the page containing the Lock Byte) FEDR—Not permitted; Causes Flash Error Device Reset (FERROR bit in RSTSRC is 1 after reset)</p> <ul style="list-style-type: none"> • All prohibited operations that are performed via the C2 interface are ignored (do not cause device reset). • Locking any Flash page also locks the page containing the Lock Byte. • Once written to, the Lock Byte cannot be modified except by performing a C2 Device Erase. • If user code writes to the Lock Byte, the Lock does not take effect until the next device reset. • The scratchpad is locked when all other Flash pages are locked. • The scratchpad is erased when a Flash Device Erase command is performed. 			

14.2. Idle Mode

Setting the Idle Mode Select bit (PCON.0) causes the CIP-51 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.

Note: To ensure the MCU enters a low power state upon entry into Idle Mode, the one-shot circuit should be enabled by clearing the BYPASS bit (FLSCL.6).

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.

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 “18.6. PCA Watchdog Timer Reset” on page 176 for more information on the use and configuration of the WDT.

14.3. Stop Mode

Setting the Stop Mode Select bit (PCON.1) causes the CIP-51 to enter stop mode as soon as the instruction that sets the bit completes execution. In stop mode the precision internal oscillator and CPU are stopped; the state of the low power oscillator and 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 CIP-51 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.

Stop mode is a legacy 8051 power mode; it will not result in optimal power savings. Sleep or suspend mode will provide more power savings if the MCU needs to be inactive for a long period of time.

Note: To ensure the MCU enters a low power state upon entry into Stop Mode, the one-shot circuit should be enabled by clearing the BYPASS bit (FLSCL.6).

C8051F91x-C8051F90x

Important Notes:

- The Power-on Reset (POR) delay is not incurred after a VDD/DC+ supply monitor reset. See Section “4. Electrical Characteristics” on page 36 for complete electrical characteristics of the VDD/DC+ monitor.
- Software should take care not to inadvertently disable the V_{DD} Monitor as a reset source when writing to RSTSRC to enable other reset sources or to trigger a software reset. All writes to RSTSRC should explicitly set PORSF to 1 to keep the V_{DD} Monitor enabled as a reset source.
- The VDD/DC+ supply monitor must be enabled before selecting it as a reset source. Selecting the VDD/DC+ supply 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 VDD/DC+ supply monitor and selecting it as a reset source. See Section “4. Electrical Characteristics” on page 36 for minimum VDD/DC+ Supply Monitor turn-on time. **No delay should be introduced in systems where software contains routines that erase or write Flash memory.** The procedure for enabling the VDD/DC+ supply monitor and selecting it as a reset source is shown below:
 1. Enable the VDD/DC+ Supply Monitor (VDMEN bit in VDM0CN = 1).
 2. Wait for the VDD/DC+ Supply Monitor to stabilize (optional).
 3. Select the VDD/DC+ Supply Monitor as a reset source (PORSF bit in RSTSRC = 1).

Internal Register Definition 20.6. RTC0XCF: SmaRTClock Oscillator Configuration

Bit	7	6	5	4	3	2	1	0
Name	AUTOSTP	LOADRDY			LOADCAP			
Type	R/W	R	R	R	R/W			
Reset	0	0	0	0	Varies	Varies	Varies	Varies

SmaRTClock Address = 0x06

Bit	Name	Function
7	AUTOSTP	Automatic Load Capacitance Stepping Enable. Enables/disables automatic load capacitance stepping. 0: Load capacitance stepping disabled. 1: Load capacitance stepping enabled.
6	LOADRDY	Load Capacitance Ready Indicator. Set by hardware when the load capacitance matches the programmed value. 0: Load capacitance is currently stepping. 1: Load capacitance has reached its programmed value.
5:4	Unused	Unused. Read = 00b; Write = Don't Care.
3:0	LOADCAP	Load Capacitance Programmed Value. Holds the user's desired value of the load capacitance. See Table 20.2 on page 196.

Internal Register Definition 20.7. RTC0PIN: SmaRTClock Pin Configuration

Bit	7	6	5	4	3	2	1	0
Name	RTC0PIN							
Type	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	Varies	Varies	Varies	Varies	Varies	Varies	Varies

SmaRTClock Address = 0x07

Bit	Name	Function
7	RTC0PIN	SmaRTClock Pin Configuration. 0: XTAL3 and XTAL4 in their normal configuration. 1: XTAL3 and XTAL4 internally shorted for use with Self Oscillate Mode.
6:0	Reserved	Reserved. Read = Varies. Software should not modify the value of these bits. To change the RTC0PIN setting, the entire register contents should be read, modified, then rewritten.

21.3. Priority Crossbar Decoder

The Priority Crossbar Decoder assigns a Port I/O pin to each software selected digital function using the fixed peripheral priority order shown in Figure 21.3. The registers XBR0, XBR1, and XBR2 defined in SFR Definition 21.1, SFR Definition 21.2, and SFR Definition 21.3 are used to select digital functions in the Crossbar. The Port pins available for assignment by the Crossbar include all Port pins (P0.0–P1.6) which have their corresponding bit in PnSKIP set to 0.

From Figure 21.3, the highest priority peripheral is UART0. If UART0 is selected in the Crossbar (using the XBRn registers), then P0.4 and P0.5 will be assigned to UART0. The next highest priority peripheral is SPI1. If SPI1 is selected in the Crossbar, then P1.0–P1.3 will be assigned to SPI1. The user should ensure that the pins to be assigned by the Crossbar have their PnSKIP bits set to 0.

For all remaining digital functions selected in the Crossbar, starting at the top of Figure 21.3 going down, the least-significant unskipped, unassigned Port pin(s) are assigned to that function. If a Port pin is already assigned (e.g. UART0 or SPI1 pins), or if its PnSKIP bit is set to 1, then the Crossbar will skip over the pin and find next available unskipped, unassigned Port pin. All Port pins used for analog functions, GPIO, or dedicated digital functions such as the EMIF should have their PnSKIP bit set to 1.

Figure 21.3 shows the Crossbar Decoder priority with no Port pins skipped (P0SKIP, P1SKIP = 0x00); Figure 21.4 shows the Crossbar Decoder priority with the External Oscillator pins (XTAL1 and XTAL2) skipped (P0SKIP = 0x0C).

Notes:

- The Crossbar must be enabled (XBARE = 1) before any Port pin is used as a digital output. Port output drivers are disabled while the Crossbar is disabled.
- When SMBus is selected in the Crossbar, the pins associated with SDA and SCL will automatically be forced into open-drain output mode regardless of the PnMDOUT setting.
- SPI0 can be operated in either 3-wire or 4-wire modes, depending on the state of the NSSMD1-NSSMD0 bits in register SPI0CN. The NSS signal is only routed to a Port pin when 4-wire mode is selected. When SPI0 is selected in the Crossbar, the SPI0 mode (3-wire or 4-wire) will affect the pinout of all digital functions lower in priority than SPI0.
- For given XBRn, PnSKIP, and SPInCN register settings, one can determine the I/O pin-out of the device using Figure 21.3 and Figure 21.4.

23.2. Operational Modes

UART0 provides standard asynchronous, full duplex communication. The UART mode (8-bit or 9-bit) is selected by the S0MODE bit (SCON0.7). Typical UART connection options are shown below.

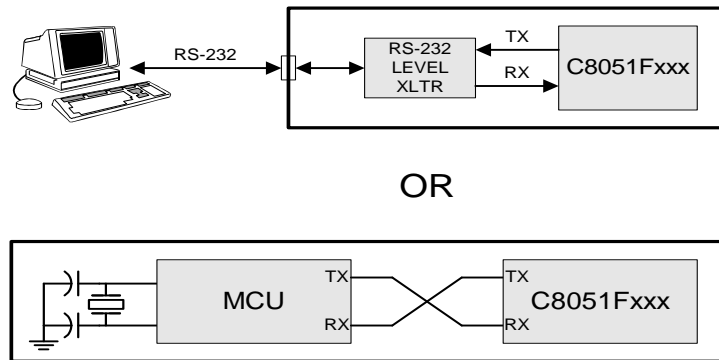


Figure 23.3. UART Interconnect Diagram

23.2.1. 8-Bit UART

8-Bit UART mode uses a total of 10 bits per data byte: one start bit, eight data bits (LSB first), and one stop bit. Data are transmitted LSB first from the TX0 pin and received at the RX0 pin. On receive, the eight data bits are stored in SBUF0 and the stop bit goes into RB80 (SCON0.2).

Data transmission begins when software writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to logic 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: RI0 must be logic 0, and if MCE0 is logic 1, the stop bit must be logic 1. In the event of a receive data overrun, the first received 8 bits are latched into the SBUF0 receive register and the following overrun data bits are lost.

If these conditions are met, the eight bits of data is stored in SBUF0, the stop bit is stored in RB80 and the RI0 flag is set. If these conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set. An interrupt will occur if enabled when either TI0 or RI0 is set.

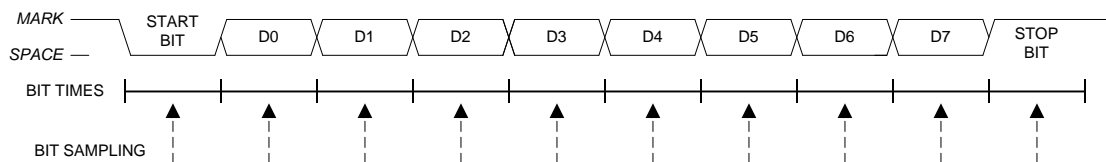


Figure 23.4. 8-Bit UART Timing Diagram

24.3. SPI Slave Mode Operation

When SPIn is enabled and not configured as a master, it will operate as a SPI slave. As a slave, bytes are shifted in through the MOSI pin and out through the MISO pin by a master device controlling the SCK signal. A bit counter in the SPIn logic counts SCK edges. When 8 bits have been shifted through the shift register, the SPIF flag is set to logic 1, and the byte is copied into the receive buffer. Data is read from the receive buffer by reading SPInDAT. A slave device cannot initiate transfers. Data to be transferred to the master device is pre-loaded into the shift register by writing to SPInDAT. Writes to SPInDAT are double-buffered, and are placed in the transmit buffer first. If the shift register is empty, the contents of the transmit buffer will immediately be transferred into the shift register. When the shift register already contains data, the SPI will load the shift register with the transmit buffer's contents after the last SCK edge of the next (or current) SPI transfer.

When configured as a slave, SPIn can be configured for 4-wire or 3-wire operation. The default, 4-wire slave mode, is active when NSSnMD1 (SPInCN.3) = 0 and NSSnMD0 (SPInCN.2) = 1. In 4-wire mode, the NSS signal is routed to a port pin and configured as a digital input. SPIn is enabled when NSS is logic 0, and disabled when NSS is logic 1. The bit counter is reset on a falling edge of NSS. Note that the NSS signal must be driven low at least 2 system clocks before the first active edge of SCK for each byte transfer. Figure 24.4 shows a connection diagram between two slave devices in 4-wire slave mode and a master device.

3-wire slave mode is active when NSSnMD1 (SPInCN.3) = 0 and NSSnMD0 (SPInCN.2) = 0. NSS is not used in this mode, and is not mapped to an external port pin through the crossbar. Since there is no way of uniquely addressing the device in 3-wire slave mode, SPIn must be the only slave device present on the bus. It is important to note that in 3-wire slave mode there is no external means of resetting the bit counter that determines when a full byte has been received. The bit counter can only be reset by disabling and re-enabling SPIn with the SPIEN bit. Figure 24.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

24.4. SPI Interrupt Sources

When SPIn interrupts are enabled, the following four flags will generate an interrupt when they are set to logic 1:

All of the following bits must be cleared by software.

1. The SPI Interrupt Flag, SPIFn (SPInCN.7) is set to logic 1 at the end of each byte transfer. This flag can occur in all SPIn modes.
2. The Write Collision Flag, WCOLn (SPInCN.6) is set to logic 1 if a write to SPInDAT is attempted when the transmit buffer has not been emptied to the SPI shift register. When this occurs, the write to SPInDAT will be ignored, and the transmit buffer will not be written. This flag can occur in all SPIn modes.
3. The Mode Fault Flag MODFn (SPInCN.5) is set to logic 1 when SPIn is configured as a master, and for multi-master mode and the NSS pin is pulled low. When a Mode Fault occurs, the MSTENn and SPIENn bits in SPI0CN are set to logic 0 to disable SPIn and allow another master device to access the bus.
4. The Receive Overrun Flag RXOVRNn (SPInCN.4) is set to logic 1 when configured as a slave, and a transfer is completed and the receive buffer still holds an unread byte from a previous transfer. The new byte is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte which caused the overrun is lost.

SFR Definition 24.1. SPInCFG: SPI 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 Addresses: SPI0CFG = 0xA1, SPI1CFG = 0x84

SFR Pages: SPI0CFG = 0x0, SPI1CFG = 0x0

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	SPI Clock Phase. 0: Data centered on first edge of SCK period.* 1: Data centered on second edge of SCK period.*
4	CKPOL	SPI Clock Polarity. 0: SCK line low in idle state. 1: SCK line high in idle state.
3	SLVSEL	Slave Selected Flag. 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). Set to logic 1 when data has been transferred in/out of the shift register, and there is no data is available to read from the transmit buffer or write to the receive buffer. Set to logic 0 when a data byte is transferred to the shift register from the transmit buffer or by a transition on SCK. Note: SRMT = 1 in Master Mode.
0	RXBMT	Receive Buffer Empty (valid in slave mode only). 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. Note: RXBMT = 1 in Master Mode.
<p>*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 24.1 for timing parameters.</p>		

C8051F91x-C8051F90x

25. Timers

Each MCU includes four counter/timers: two are 16-bit counter/timers compatible with those found in the standard 8051, and two are 16-bit auto-reload timer for use with the ADC, SMBus, or for general purpose use. These timers can be used to measure time intervals, count external events and generate periodic interrupt requests. Timer 0 and Timer 1 are nearly identical and have four primary modes of operation. Timer 2 and Timer 3 offer 16-bit and split 8-bit timer functionality with auto-reload. Additionally, Timer 2 and Timer 3 have a Capture Mode that can be used to measure the SmaRTClock or a Comparator period with respect to another oscillator. This is particularly useful when using Capacitive Touch Switches.

Timer 0 and Timer 1 Modes:	Timer 2 Modes:	Timer 3 Modes:
13-bit counter/timer	16-bit timer with auto-reload	16-bit timer with auto-reload
16-bit counter/timer		
8-bit counter/timer with auto-reload	Two 8-bit timers with auto-reload	Two 8-bit timers with auto-reload
Two 8-bit counter/timers (Timer 0 only)		

Timers 0 and 1 may be clocked by one of five sources, determined by the Timer Mode Select bits (T1M–T0M) and the Clock Scale bits (SCA1–SCA0). The Clock Scale bits define a pre-scaled clock from which Timer 0 and/or Timer 1 may be clocked (See SFR Definition 25.1 for pre-scaled clock selection).

Timer 0/1 may then be configured to use this pre-scaled clock signal or the system clock. Timer 2 and Timer 3 may be clocked by the system clock, the system clock divided by 12. Timer 2 may additionally be clocked by the SmaRTClock divided by 8 or the Comparator0 output. Timer 3 may additionally be clocked by the external oscillator clock source divided by 8 or the Comparator1 output.

Timer 0 and Timer 1 may also be operated as counters. When functioning as a counter, a counter/timer register is incremented on each high-to-low transition at the selected input pin (T0 or T1). Events with a frequency of up to one-fourth the system clock frequency can be counted. The input signal need not be periodic, but it should be held at a given level for at least two full system clock cycles to ensure the level is properly sampled.

SFR Definition 25.6. TH0: Timer 0 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TH0[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x8C

Bit	Name	Function
7:0	TH0[7:0]	Timer 0 High Byte. The TH0 register is the high byte of the 16-bit Timer 0.

SFR Definition 25.7. TH1: Timer 1 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TH1[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x8D

Bit	Name	Function
7:0	TH1[7:0]	Timer 1 High Byte. The TH1 register is the high byte of the 16-bit Timer 1.

SFR Definition 26.2. PCA0MD: PCA Mode

Bit	7	6	5	4	3	2	1	0
Name	CIDL	WDTE	WDLCK		CPS2	CPS1	CPS0	ECF
Type	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xD9

Bit	Name	Function
7	CIDL	PCA Counter/Timer Idle Control. Specifies PCA behavior when CPU is in Idle Mode. 0: PCA continues to function normally while the system controller is in Idle Mode. 1: PCA operation is suspended while the system controller is in Idle Mode.
6	WDTE	Watchdog Timer Enable. If this bit is set, PCA Module 2 is used as the watchdog timer. 0: Watchdog Timer disabled. 1: PCA Module 2 enabled as Watchdog Timer.
5	WDLCK	Watchdog Timer Lock. This bit locks/unlocks the Watchdog Timer Enable. When WDLCK is set, the Watchdog Timer may not be disabled until the next system reset. 0: Watchdog Timer Enable unlocked. 1: Watchdog Timer Enable locked.
4	Unused	Read = 0b, Write = don't care.
3:1	CPS[2:0]	PCA Counter/Timer Pulse Select. These bits select the timebase source for the PCA counter 000: System clock divided by 12 001: System clock divided by 4 010: Timer 0 overflow 011: High-to-low transitions on ECI (max rate = system clock divided by 4) 100: System clock 101: External clock divided by 8 (synchronized with the system clock) 110: SmARTClock divided by 8 (synchronized with the system clock and only available on 'F912 and 'F902 devices -- this setting is reserved on all other devices) 111: Reserved
0	ECF	PCA Counter/Timer Overflow Interrupt Enable. This bit sets the masking of the PCA Counter/Timer Overflow (CF) interrupt. 0: Disable the CF interrupt. 1: Enable a PCA Counter/Timer Overflow interrupt request when CF (PCA0CN.7) is set.

Note: When the WDTE bit is set to 1, the other bits in the PCA0MD register cannot be modified. To change the contents of the PCA0MD register, the Watchdog Timer must first be disabled.