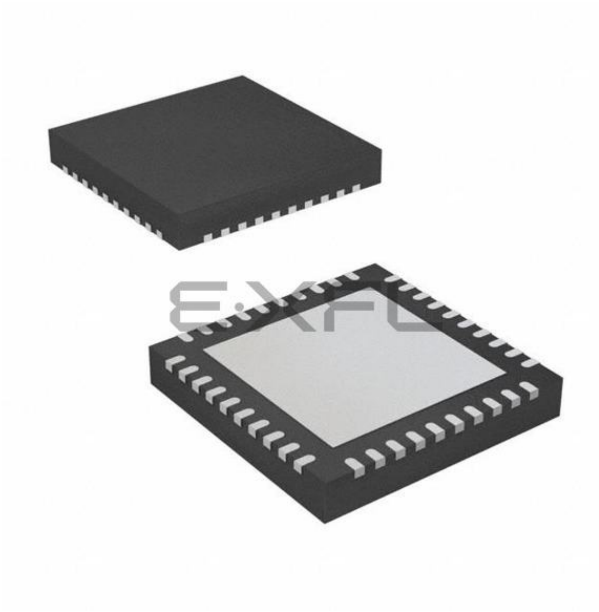
Silicon Labs - C8051F569-IMR Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	8051
Core Size	8-Bit
Speed	50MHz
Connectivity	CANbus, EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	33
Program Memory Size	32KB (32K 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 32x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	40-VFQFN Exposed Pad
Supplier Device Package	40-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f569-imr

4.4. QFN-24 Package Specifications

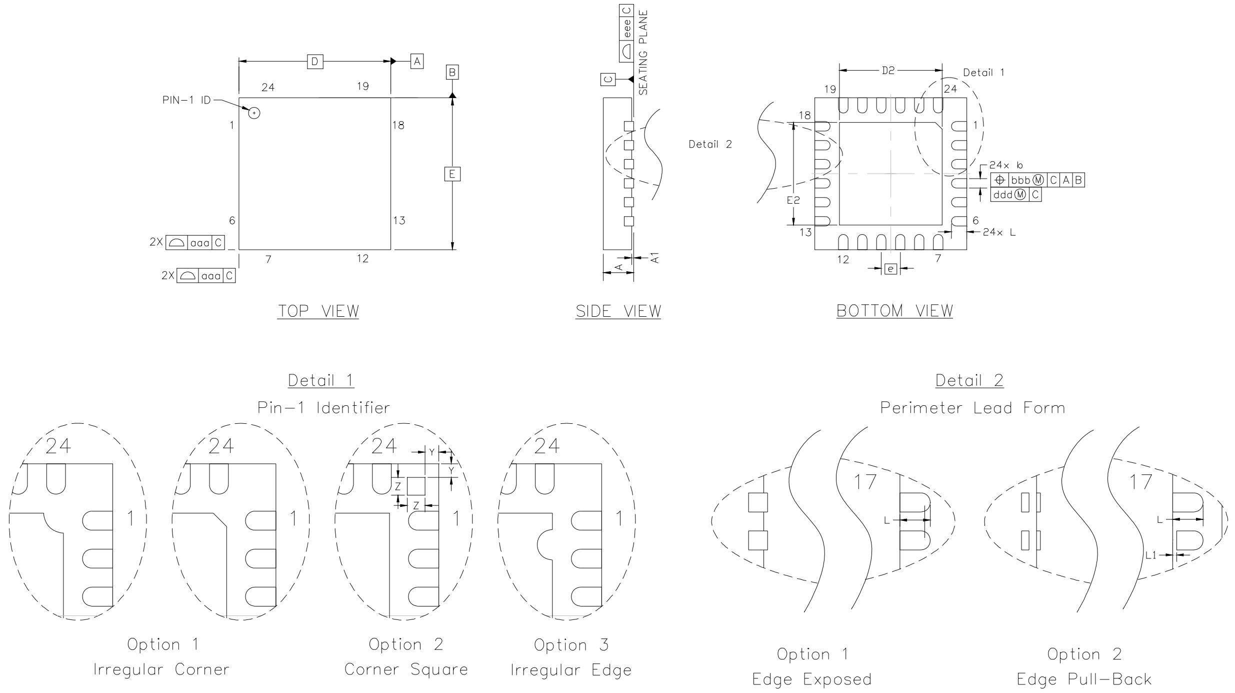


Figure 4.7. QFN-24 Package Drawing

Table 4.7. QFN-24 Package Dimensions

Dimension	Min	Typ	Max	Dimension	Min	Typ	Max
A	0.70	0.75	0.80	L	0.30	0.40	0.50
A1	0.00	0.02	0.05	L1	0.00		0.15
b	0.18	0.25	0.30	aaa			0.15
D	4.00 BSC			bbb			0.10
D2	2.55	2.70	2.80	ddd			0.05
e	0.50 BSC			eee			0.08
E	4.00 BSC			Z		0.24	
E2	2.55	2.70	2.80	Y		0.18	

Notes:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to JEDEC Solid State Outline MO-220, variation WGGD, except for custom features D2, E2, Z, Y, and L which are toleranced per supplier designation.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

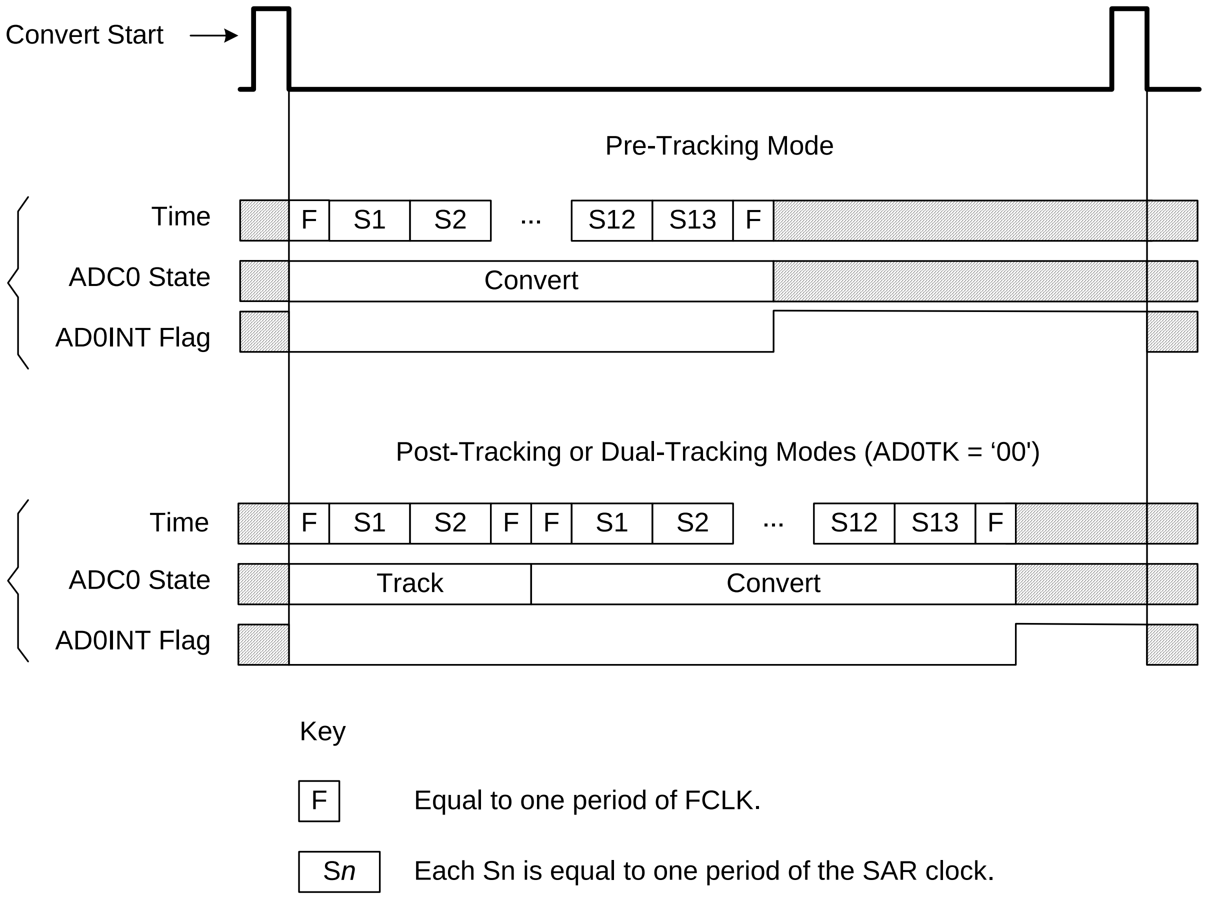


Figure 6.3. 12-Bit ADC Tracking Mode Example

6.1.4. Burst Mode

Burst Mode is a power saving feature that allows ADC0 to remain in a very low power state between conversions. When Burst Mode is enabled, ADC0 wakes from a very low power state, accumulates 1, 4, 8, or 16 samples using an internal Burst Mode clock (approximately 25 MHz), then re-enters a very low power state. Since the Burst Mode clock is independent of the system clock, ADC0 can perform multiple conversions then enter a very low power state within a single system clock cycle, even if the system clock is slow (e.g., 32.768 kHz), or suspended.

Burst Mode is enabled by setting BURSTEN to logic 1. When in Burst Mode, AD0EN controls the ADC0 idle power state (i.e. the state ADC0 enters when not tracking or performing conversions). If AD0EN is set to logic 0, ADC0 is powered down after each burst. If AD0EN is set to logic 1, ADC0 remains enabled after each burst. On each convert start signal, ADC0 is awakened from its Idle Power State. If ADC0 is powered down, it will automatically power up and wait the programmable Power-up Time controlled by the AD0PWR bits. Otherwise, ADC0 will start tracking and converting immediately. Figure 6.4 shows an example of Burst Mode Operation with a slow system clock and a repeat count of 4.

Important Note: When Burst Mode is enabled, only Post-Tracking and Dual-Tracking modes can be used.

When Burst Mode is enabled, a single convert start will initiate a number of conversions equal to the repeat count. When Burst Mode is disabled, a convert start is required to initiate each conversion. In both modes, the ADC0 End of Conversion Interrupt Flag (AD0INT) will be set after “repeat count” conversions have

11. Memory Organization

The memory organization of the CIP-51 System Controller is similar to that of a standard 8051. There are two separate memory spaces: program memory and data memory. Program and data memory share the same address space but are accessed via different instruction types. The memory organization is shown in Figure 11.1

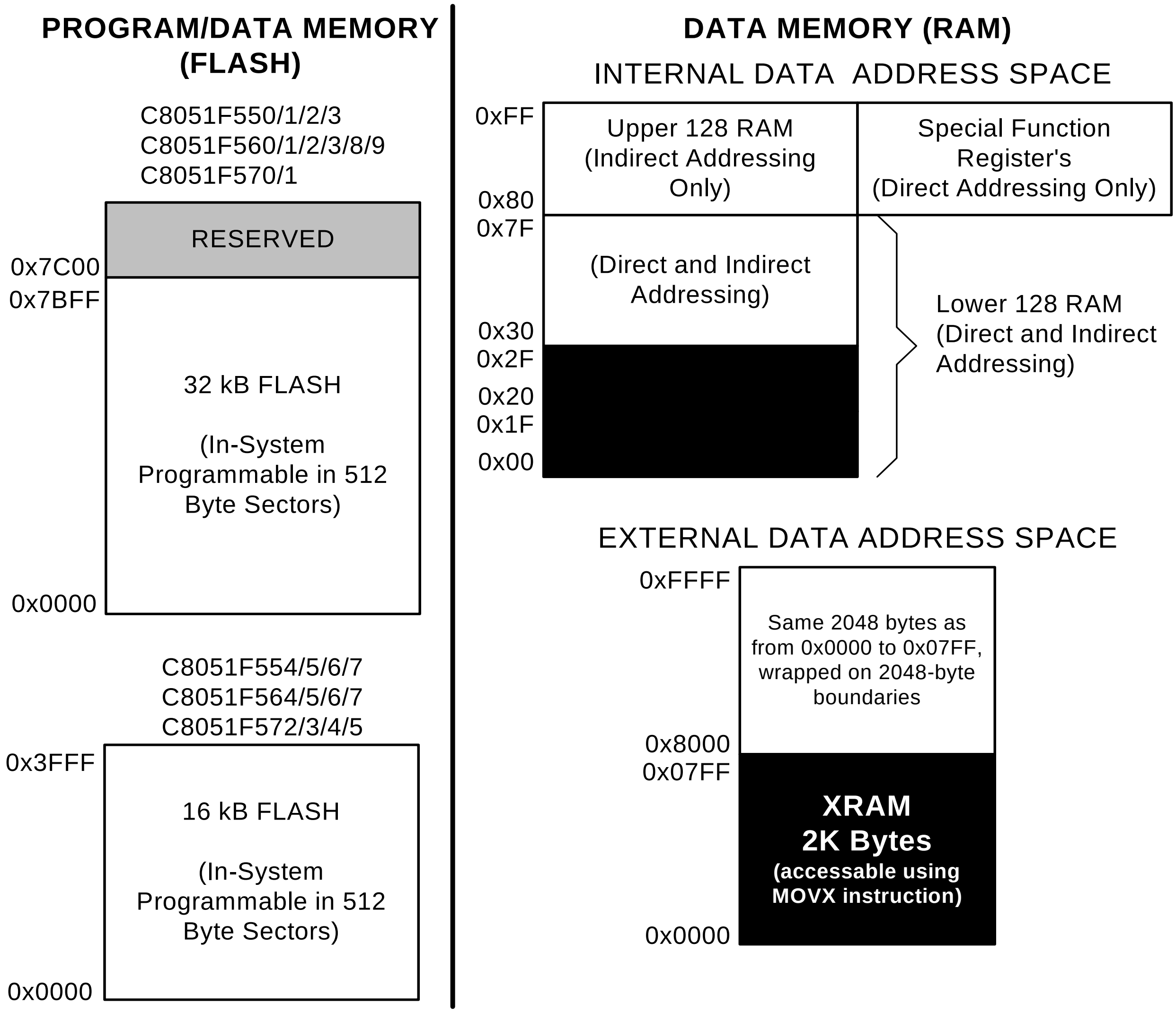

PROGRAM/DATA MEMORY (FLASH)
C8051F550/1/2/3
C8051F560/1/2/3/8/9
C8051F570/1
RESERVED
0x7C00
0x7BFF
32 kB FLASH
(In-System Programmable in 512 Byte Sectors)
0x0000
C8051F554/5/6/7
C8051F564/5/6/7
C8051F572/3/4/5
0x3FFF
16 kB FLASH
(In-System Programmable in 512 Byte Sectors)
0x0000
DATA MEMORY (RAM)
INTERNAL DATA ADDRESS SPACE
0xFF
Upper 128 RAM (Indirect Addressing Only)
Special Function Register's (Direct Addressing Only)
0x80
0x7F
(Direct and Indirect Addressing)
0x30
0x2F
Bit Addressable
0x20
0x1F
General Purpose Registers
0x00
Lower 128 RAM (Direct and Indirect Addressing)
EXTERNAL DATA ADDRESS SPACE
0xFFFF
Same 2048 bytes as from 0x0000 to 0x07FF, wrapped on 2048-byte boundaries
0x8000
0x07FF
XRAM
2K Bytes
(accessable using MOVX instruction)
0x0000

Figure 11.1. C8051F55x/56x/57x Memory Map

11.1. Program Memory

The CIP-51 core has a 64 kB program memory space. The C8051F55x/56x/57x devices implement 32 kB or 16 kB of this program memory space as in-system, re-programmable Flash memory, organized in a contiguous block from addresses 0x0000 to 0x7FFF in 32 kB devices and addresses 0x0000 to 0x3FFF in 16 kB devices. The address 0x7BFF in 32 kB devices and 0x3FFF in 16 kB devices serves as the security lock byte for the device. Addresses above 0x7BFF are reserved in the 32 kB devices.

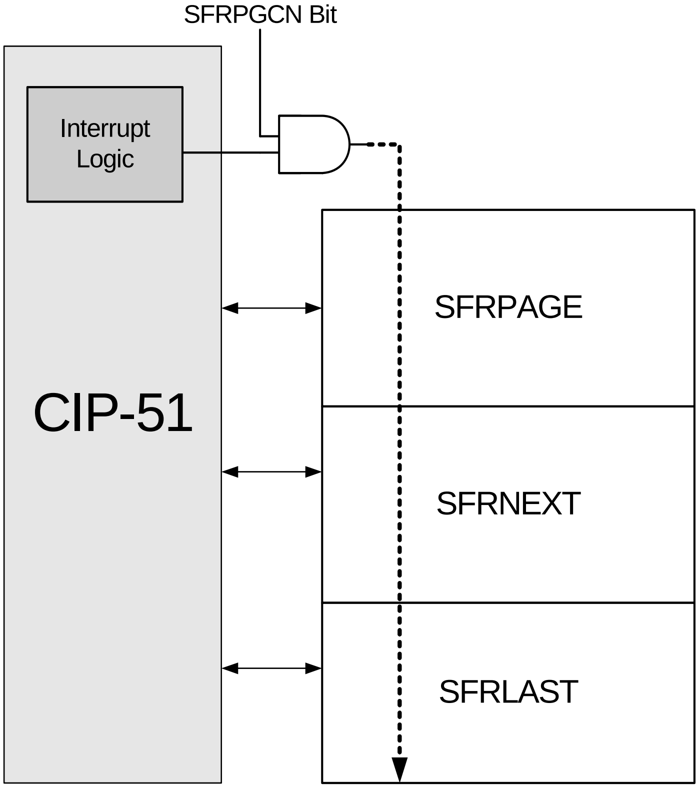


Figure 12.1. SFR Page Stack

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.

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]	**SFR Page Bits.** This is the value that will go to the SFR Page register upon a return from interrupt. 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. Read: Returns the value of the SFR page contained in the second byte of the SFR stack. 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.

13.1.1. Interrupt Priorities

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. Each interrupt has an associated interrupt priority bit in an SFR (IE, EIP1, or EIP2) used to configure its priority level. Low priority is the default. If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate, given in Table 13.1.

13.1.2. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 5 system clock cycles: 1 clock cycle to detect the interrupt and 4 clock cycles to complete the LCALL to the ISR. If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs when the CPU is performing an RETI instruction followed by a DIV as the next instruction. In this case, the response time is 18 system clock cycles: 1 clock cycle to detect the interrupt, 5 clock cycles to execute the RETI, 8 clock cycles to complete the DIV instruction and 4 clock cycles to execute the LCALL to the ISR. If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the RETI and following instruction.

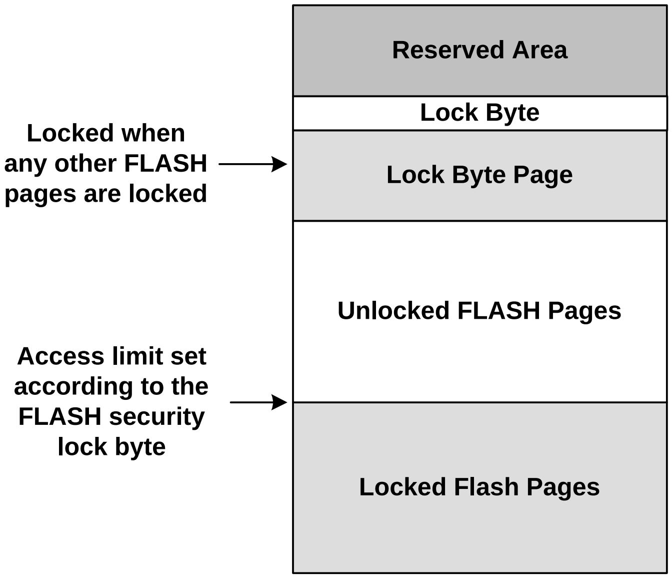
14.2. Non-volatile Data Storage

The Flash memory can be used for non-volatile data storage as well as program code. This allows data such as calibration coefficients to be calculated and stored at run time. Data is written using the MOVX write instruction and read using the MOVC instruction. Note: MOVX read instructions always target XRAM.

14.3. 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 ones 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).** See example in Figure 14.1.


Reserved Area
Lock Byte
Lock Byte Page
Unlocked FLASH Pages
Locked Flash Pages
Locked when any other FLASH pages are locked
Access limit set according to the FLASH security lock byte

Security Lock Byte:	11111101b
1s Complement:	00000010b
Flash pages locked:	3 (First two Flash pages + Lock Byte Page)

Figure 14.1. Flash Program Memory Map

16. Reset Sources

Reset circuitry allows the controller to be easily placed in a predefined default condition. On entry to this reset state, the following 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 $\overline{\text{RST}}$ pin is driven low until the device exits the reset state.

Note: When VIO rises faster than VDD, which can happen when VREGIN and VIO are tied together, a delay created between GPIO power (VIO) and the logic controlling GPIO (VDD) results in a temporary unknown state at the GPIO pins. When VIO rises faster than VDD, the GPIO may enter the following states: floating, glitch low, or glitch high. Cross coupling VIO and VDD with a 4.7 µF capacitor mitigates the root cause of the problem by allowing VIO and VDD to rise at the same rate.

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.

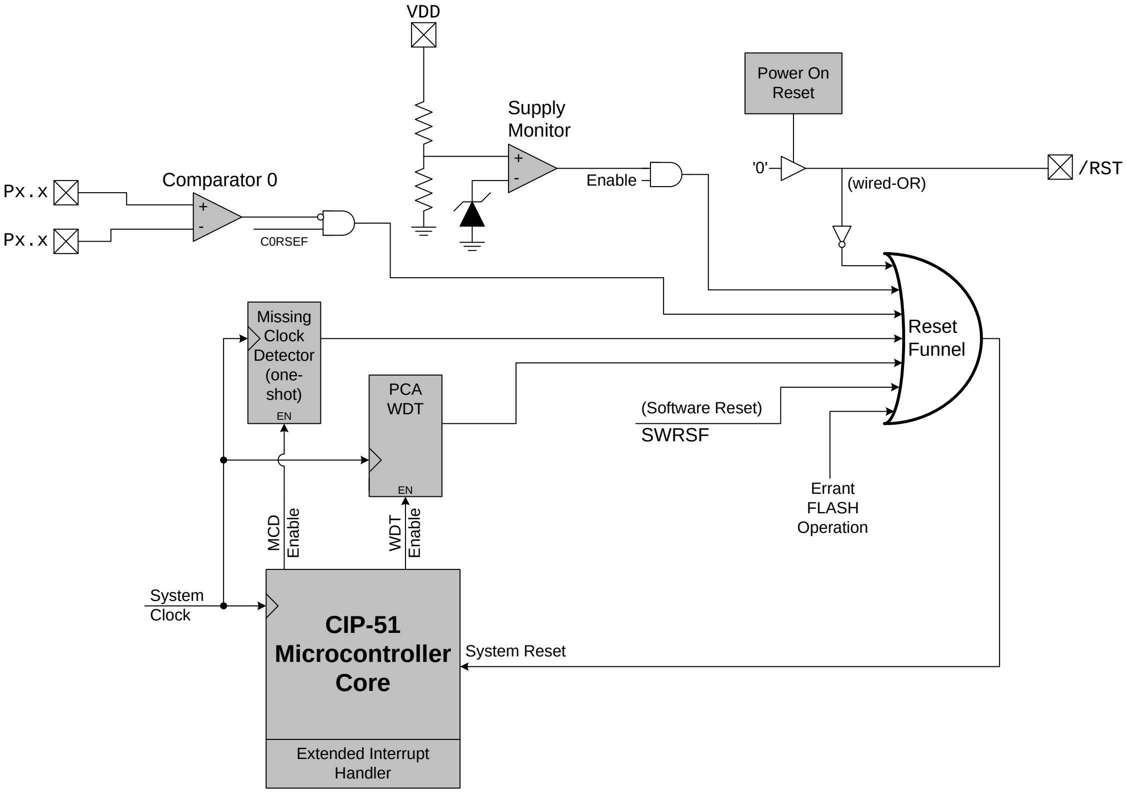


Figure 16.1. Reset Sources

16.1. Power-On Reset

During power-up, the device is held in a reset state and the $\overline{\text{RST}}$ pin is driven low until V_{DD} settles above V_{RST}. A delay occurs before the device is released from reset; the delay decreases as the V_{DD} ramp time increases (V_{DD} ramp time is defined as how fast V_{DD} ramps from 0 V to V_{RST}). Figure 16.2. plots the power-on and V_{DD} monitor reset timing.

On exit from a power-on reset, the PORSF flag (RSTSRC.1) is set by hardware to logic 1. When PORSF is set, all of the other reset flags in the RSTSRC Register are indeterminate (PORSF is cleared by all other resets). Since all resets cause program execution to begin at the same location (0x0000) software can read the PORSF flag to determine if a power-up was the cause of reset. The content of internal data memory should be assumed to be undefined after a power-on reset. The V_{DD} monitor is enabled following a power-on reset.

Note: For devices with a date code before year 2011, work week 24 (1124), if the /RST pin is held low for more than 1 second while power is applied to the device, and then /RST is released, a percentage of devices may lock up and fail to execute code. Toggling the /RST pin does not clear the condition. The condition is cleared by cycling power. Most devices that are affected will show the lock up behavior only within a narrow range of temperatures (a 5 to 10 °C window). Parts with a date code of year 2011, work week 24 (1124) or later do not have any restrictions on /RST low time. The date code of a device is a four-digit number on the bottom-most line of each device with the format YYWW, where YY is the two-digit calendar year and WW is the two digit work week.

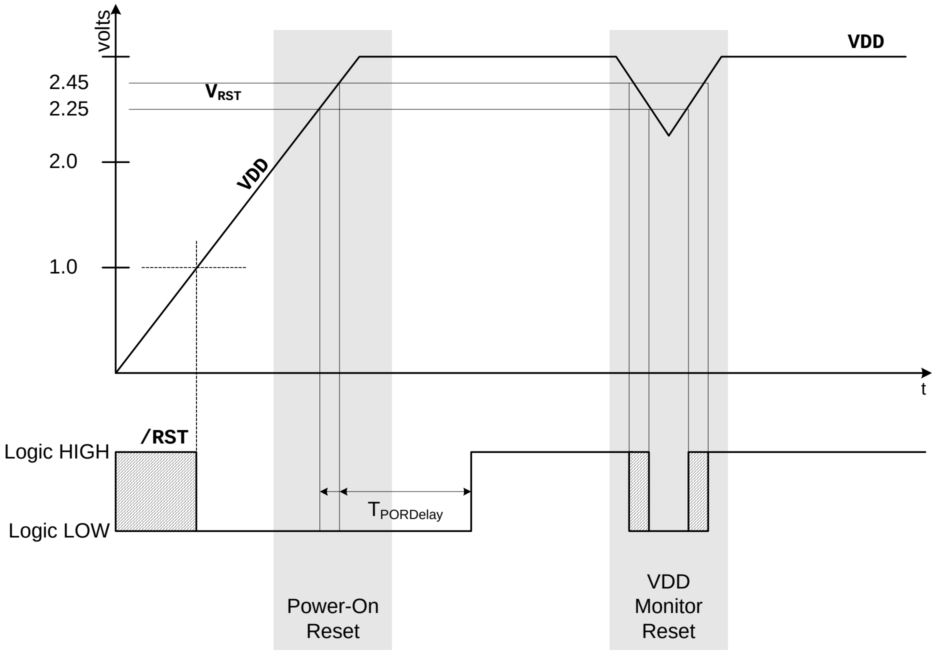


Figure 16.2. Power-On and V_{DD} Monitor Reset Timing

17. External Data Memory Interface and On-Chip XRAM

For C8051F55x/56x/57x devices, 2 kB of RAM are included on-chip and mapped into the external data memory space (XRAM). Additionally, an External Memory Interface (EMIF) is available on the C8051F568-9 and ‘F570-5 devices, which can be used to access off-chip data memories and memory-mapped devices connected to the GPIO ports. The external memory space may be accessed using the external move instruction (MOVX) and the data pointer (DPTR), or using the MOVX indirect addressing mode using R0 or R1. If the MOVX instruction is used with an 8-bit address operand (such as @R1), then the high byte of the 16-bit address is provided by the External Memory Interface Control Register (EMI0CN, shown in SFR Definition 17.1).

Note: The MOVX instruction can also be used for writing to the Flash memory. See Section “14. Flash Memory” on page 124 for details. The MOVX instruction accesses XRAM by default.

17.1. Accessing XRAM

The XRAM memory space is accessed using the MOVX instruction. The MOVX instruction has two forms, both of which use an indirect addressing method. The first method uses the Data Pointer, DPTR, a 16-bit register which contains the effective address of the XRAM location to be read from or written to. The second method uses R0 or R1 in combination with the EMI0CN register to generate the effective XRAM address. Examples of both of these methods are given below.

17.1.1. 16-Bit MOVX Example

The 16-bit form of the MOVX instruction accesses the memory location pointed to by the contents of the DPTR register. The following series of instructions reads the value of the byte at address 0x1234 into the accumulator A:

```
MOV    DPTR, #1234h        ; load DPTR with 16-bit address to read (0x1234)
MOVX   A, @DPTR            ; load contents of 0x1234 into accumulator A
```

The above example uses the 16-bit immediate MOV instruction to set the contents of DPTR. Alternately, the DPTR can be accessed through the SFR registers DPH, which contains the upper 8-bits of DPTR, and DPL, which contains the lower 8-bits of DPTR.

17.1.2. 8-Bit MOVX Example

The 8-bit form of the MOVX instruction uses the contents of the EMI0CN SFR to determine the upper 8-bits of the effective address to be accessed and the contents of R0 or R1 to determine the lower 8-bits of the effective address to be accessed. The following series of instructions read the contents of the byte at address 0x1234 into the accumulator A.

```
MOV    EMI0CN, #12h        ; load high byte of address into EMI0CN
MOV    R0, #34h            ; load low byte of address into R0 (or R1)
MOVX   a, @R0              ; load contents of 0x1234 into accumulator A
```

17.5. Memory Mode Selection

The external data memory space can be configured in one of four modes, shown in Figure 17.2, based on the EMIF Mode bits in the EMI0CF register (SFR Definition 17.2). These modes are summarized below. More information about the different modes can be found in Section “17.6. Timing” on page 151.

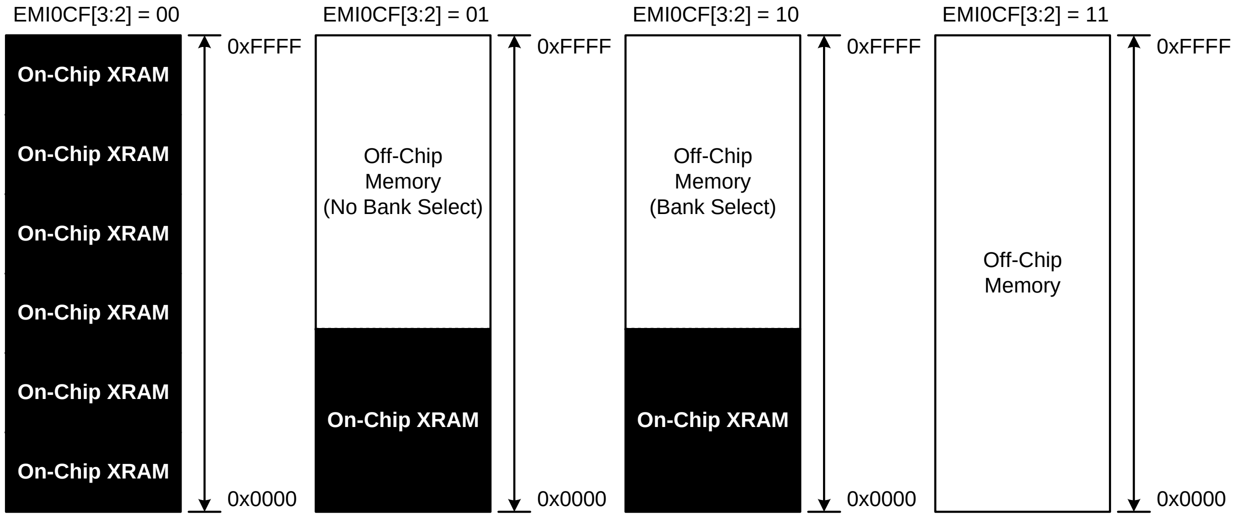


Figure 17.2. EMIF Operating Modes

17.5.1. Internal XRAM Only

When bits EMI0CF[3:2] are set to 00, all MOVX instructions will target the internal XRAM space on the device. Memory accesses to addresses beyond the populated space will wrap on 2 kB boundaries. As an example, the addresses 0x800 and 0x1000 both evaluate to address 0x0000 in on-chip XRAM space.

- 8-bit MOVX operations use the contents of EMI0CN to determine the high-byte of the effective address and R0 or R1 to determine the low-byte of the effective address.
- 16-bit MOVX operations use the contents of the 16-bit DPTR to determine the effective address.

17.5.2. Split Mode without Bank Select

When bit EMI0CF.[3:2] are set to 01, the XRAM memory map is split into two areas, on-chip space and off-chip space.

- Effective addresses below the internal XRAM size boundary will access on-chip XRAM space.
- Effective addresses above the internal XRAM size boundary will access off-chip space.
- 8-bit MOVX operations use the contents of EMI0CN to determine whether the memory access is on-chip or off-chip. However, in the “No Bank Select” mode, an 8-bit MOVX operation will not drive the upper 8-bits A[15:8] of the Address Bus during an off-chip access. This allows the user to manipulate the upper address bits at will by setting the Port state directly via the port latches. This behavior is in contrast with “Split Mode with Bank Select” described below. The lower 8-bits of the Address Bus A[7:0] are driven, determined by R0 or R1.
- 16-bit MOVX operations use the contents of DPTR to determine whether the memory access is on-chip or off-chip, and unlike 8-bit MOVX operations, the full 16-bits of the Address Bus A[15:0] are driven during the off-chip transaction.

18.4. External Oscillator Drive Circuit

The external oscillator circuit may drive an external crystal, ceramic resonator, capacitor, or RC network. A CMOS clock may also provide a clock input. For a crystal or ceramic resonator configuration, the crystal/resonator must be wired across the XTAL1 and XTAL2 pins as shown in Option 1 of Figure 18.1. A 10 MΩ resistor also must be wired across the XTAL2 and XTAL1 pins for the crystal/resonator configuration. In RC, capacitor, or CMOS clock configuration, the clock source should be wired to the XTAL2 pin as shown in Option 2, 3, or 4 of Figure 18.1. The type of external oscillator must be selected in the OSCXCN register, and the frequency control bits (XFCN) must be selected appropriately (see SFR Definition 18.6).

Important Note on External Oscillator Usage: Port pins must be configured when using the external oscillator circuit. When the external oscillator drive circuit is enabled in crystal/resonator mode, Port pins P0.2 and P0.3 are used as XTAL1 and XTAL2 respectively. When the external oscillator drive circuit is enabled in capacitor, RC, or CMOS clock mode, Port pin P0.3 is used as XTAL2. The Port I/O Crossbar should be configured to skip the Port pins used by the oscillator circuit; see Section “19.3. Priority Crossbar Decoder” on page 172 for Crossbar configuration. Additionally, when using the external oscillator circuit in crystal/resonator, capacitor, or RC mode, the associated Port pins should be configured as **analog inputs**. In CMOS clock mode, the associated pin should be configured as a **digital input**. See Section “19.4. Port I/O Initialization” on page 174 for details on Port input mode selection.

SFR Definition 19.3. XBR2: Port I/O Crossbar Register 1

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	WEAKPUD	XBARE	Reserved					LIN0E
Type	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC7; SFR Page = 0x0F

Bit	Name	Function
7	WEAKPUD	**Port I/O Weak Pullup Disable.** 0: Weak Pullups enabled (except for Ports whose I/O are configured for analog mode). 1: Weak Pullups disabled.
6	XBARE	**Crossbar Enable.** 0: Crossbar disabled. 1: Crossbar enabled.
5:1	Reserved	Always Write to 00000b.
0	LIN0E	**LIN I/O Output Enable.** 0: LIN I/O unavailable at Port pin. 1: LIN_TX, LIN_RX routed to Port pins.

SFR Definition 19.6. P1MASK: Port 1 Mask Register

Bit	7	6	5	4	3	2	1	0
Name	P1MASK[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xF4; SFR Page = 0x00

Bit	Name	Function
7:0	P1MASK[7:0]	**Port 1 Mask Value.** Selects P1 pins to be compared to the corresponding bits in P1MAT. 0: P1.n pin logic value is ignored and cannot cause a Port Mismatch event. 1: P1.n pin logic value is compared to P1MAT.n.

SFR Definition 19.7. P1MAT: Port 1 Match Register

Bit	7	6	5	4	3	2	1	0
Name	P1MAT[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xF3; SFR Page = 0x00

Bit	Name	Function
7:0	P1MAT[7:0]	**Port 1 Match Value.** Match comparison value used on Port 1 for bits in P1MAT which are set to 1. 0: P1.n pin logic value is compared with logic LOW. 1: P1.n pin logic value is compared with logic HIGH.

SFR Definition 19.17. P1MDIN: Port 1 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	P1MDIN[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xF2; SFR Page = 0x0F

Bit	Name	Function
7:0	P1MDIN[7:0]	**Analog Configuration Bits for P1.7–P1.0 (respectively).** Port pins configured for analog mode have their weak pull-up and digital receiver disabled. For analog mode, the pin also needs to be configured for open-drain mode in the P1MDOUT register. 0: Corresponding P1.n pin is configured for analog mode. 1: Corresponding P1.n pin is not configured for analog mode.

SFR Definition 19.18. P1MDOUT: Port 1 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	P1MDOUT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xA5; SFR Page = 0x0F

Bit	Name	Function
7:0	P1MDOUT[7:0]	**Output Configuration Bits for P1.7–P1.0 (respectively).** These bits are ignored if the corresponding bit in register P1MDIN is logic 0. 0: Corresponding P1.n Output is open-drain. 1: Corresponding P1.n Output is push-pull.

SFR Definition 19.29. P4MDOUT: Port 4 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	P4MDOUT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xAF; SFR Page = 0x0F

Bit	Name	Function
7:0	P4MDOUT[7:0]	**Output Configuration Bits for P4.7–P4.0 (respectively).** 0: Corresponding P4.n Output is open-drain. 1: Corresponding P4.n Output is push-pull.
Note: Port 4.0 is available on 40-pin packages.		

Table 20.2. Manual Baud Rate Parameters Examples

	Baud (bits/sec)														
	20 K			**19.2 K**			**9.6 K**			**4.8 K**			**1 K**		
SYSCLK (MHz)	**Mult.**	**Pres.**	**Div.**	**Mult.**	**Pres.**	**Div.**	**Mult.**	**Pres.**	**Div.**	**Mult.**	**Pres.**	**Div.**	**Mult.**	**Pres.**	**Div.**
25	0	1	312	0	1	325	1	1	325	3	1	325	19	1	312
24.5	0	1	306	0	1	319	1	1	319	3	1	319	19	1	306
24	0	1	300	0	1	312	1	1	312	3	1	312	19	1	300
22.1184	0	1	276	0	1	288	1	1	288	3	1	288	19	1	276
16	0	1	200	0	1	208	1	1	208	3	1	208	19	1	200
12.25	0	0	306	0	0	319	1	0	319	3	0	319	19	0	306
12	0	0	300	0	0	312	1	0	312	3	0	312	19	0	300
11.0592	0	0	276	0	0	288	1	0	288	3	0	288	19	0	276
8	0	0	200	0	0	208	1	0	208	3	0	208	19	0	200

20.2.4. Baud Rate Calculations—Automatic Mode

If the LIN controller is configured for slave mode, only the prescaler and divider need to be calculated:

$$\text{prescaler} = \ln\left[\frac{\text{SYSCLK}}{4000000}\right] \times \frac{1}{\ln 2} - 1$$

$$\text{divider} = \frac{\text{SYSCLK}}{2^{(\text{prescaler} + 1)} \times 20000}$$

The following example calculates the values of these variables for a 24 MHz system clock:

$$\text{prescaler} = \ln\left[\frac{24000000}{4000000}\right] \times \frac{1}{\ln 2} - 1 = 1.585 \cong 1$$

$$\text{divider} = \frac{24000000}{2^{(1 + 1)} \times 20000} = 300$$

Table 20.3 presents some typical values of system clock and baud rate along with their factors.

23.3.3. Multiprocessor Communications

UART0 supports multiprocessor communication between a master processor and one or more slave processors by special use of the extra data bit. When a master processor wants to transmit to one or more slaves, it first sends an address byte to select the target(s). An address byte differs from a data byte in that its extra bit is logic 1; in a data byte, the extra bit is always set to logic 0.

Setting the MCE0 bit (SMOD0.7) of a slave processor configures its UART such that when a stop bit is received, the UART will generate an interrupt only if the extra bit is logic 1 (RBX0 = 1) signifying an address byte has been received. In the UART interrupt handler, software will compare the received address with the slave's own assigned address. If the addresses match, the slave will clear its MCE0 bit to enable interrupts on the reception of the following data byte(s). Slaves that weren't addressed leave their MCE0 bits set and do not generate interrupts on the reception of the following data bytes, thereby ignoring the data. Once the entire message is received, the addressed slave resets its MCE0 bit to ignore all transmissions until it receives the next address byte.

Multiple addresses can be assigned to a single slave and/or a single address can be assigned to multiple slaves, thereby enabling "broadcast" transmissions to more than one slave simultaneously. The master processor can be configured to receive all transmissions or a protocol can be implemented such that the master/slave role is temporarily reversed to enable half-duplex transmission between the original master and slave(s).

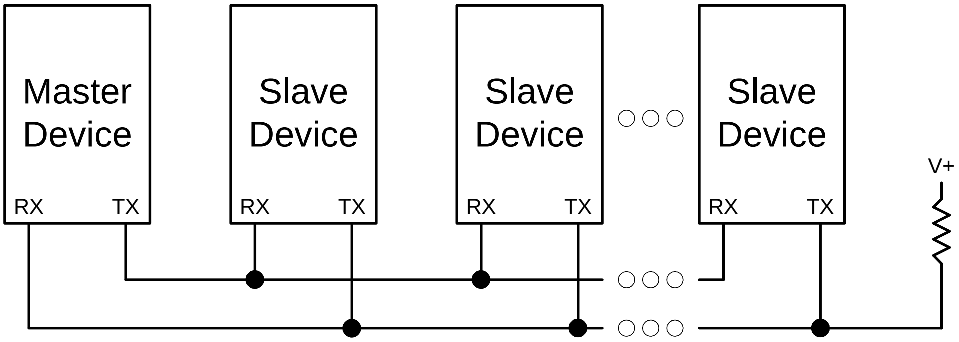


Figure 23.6. UART Multi-Processor Mode Interconnect Diagram

26. Programmable Counter Array

The Programmable Counter Array (PCA0) provides enhanced timer functionality while requiring less CPU intervention than the standard 8051 counter/timers. The PCA consists of a dedicated 16-bit counter/timer and six 16-bit capture/compare modules. Each capture/compare module has its own associated I/O line (CEXn) which is routed through the Crossbar to Port I/O when enabled. The counter/timer is driven by a programmable timebase that can select between six sources: system clock, system clock divided by four, system clock divided by twelve, the external oscillator clock source divided by 8, Timer 0 overflows, or an external clock signal on the ECI input pin. Each capture/compare module may be configured to operate independently in one of six modes: Edge-Triggered Capture, Software Timer, High-Speed Output, Frequency Output, 8 to 11-Bit PWM, or 16-Bit PWM (each mode is described in Section “26.3. Capture/Compare Modules” on page 283). The external oscillator clock option is ideal for real-time clock (RTC) functionality, allowing the PCA to be clocked by a precision external oscillator while the internal oscillator drives the system clock. The PCA is configured and controlled through the system controller's Special Function Registers. The PCA block diagram is shown in Figure 26.1

Important Note: The PCA Module 5 may be used as a watchdog timer (WDT), and is enabled in this mode following a system reset. **Access to certain PCA registers is restricted while WDT mode is enabled**. See Section 26.4 for details.

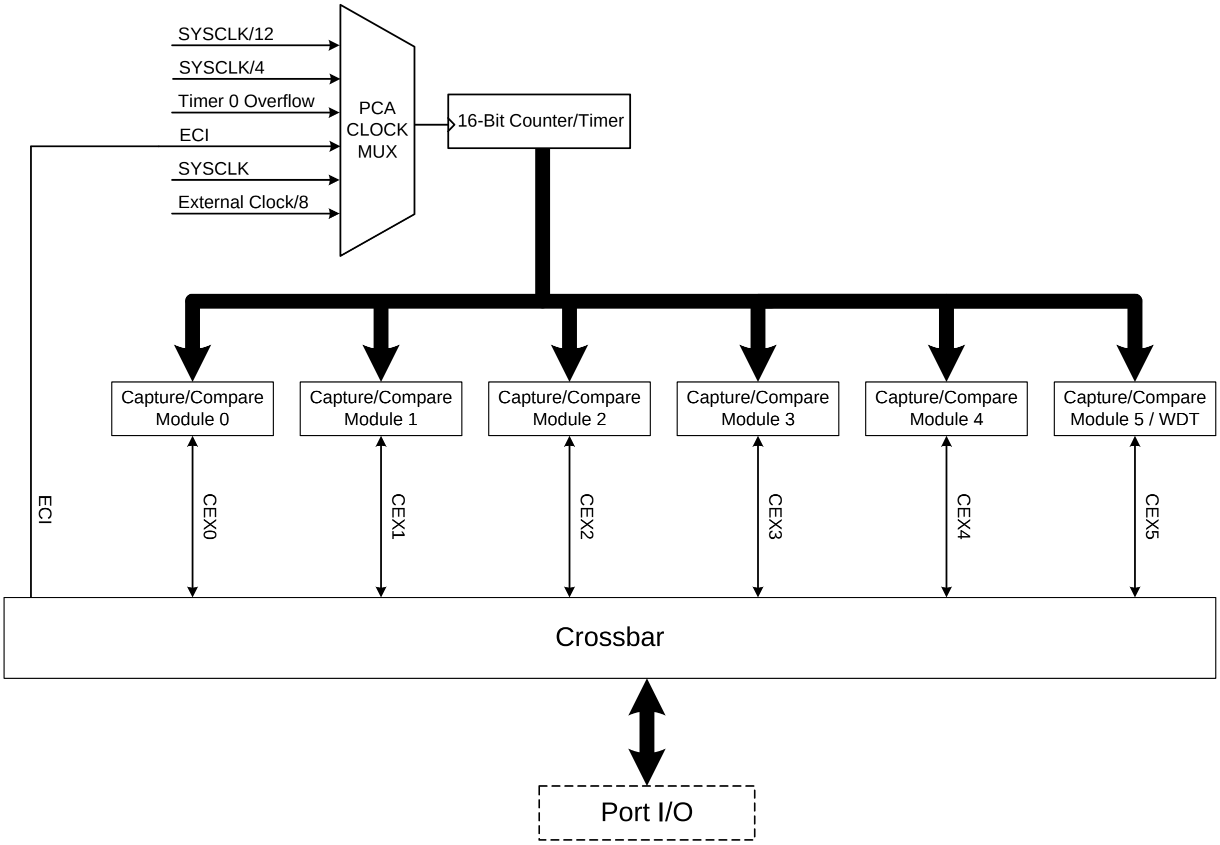


Figure 26.1. PCA Block Diagram

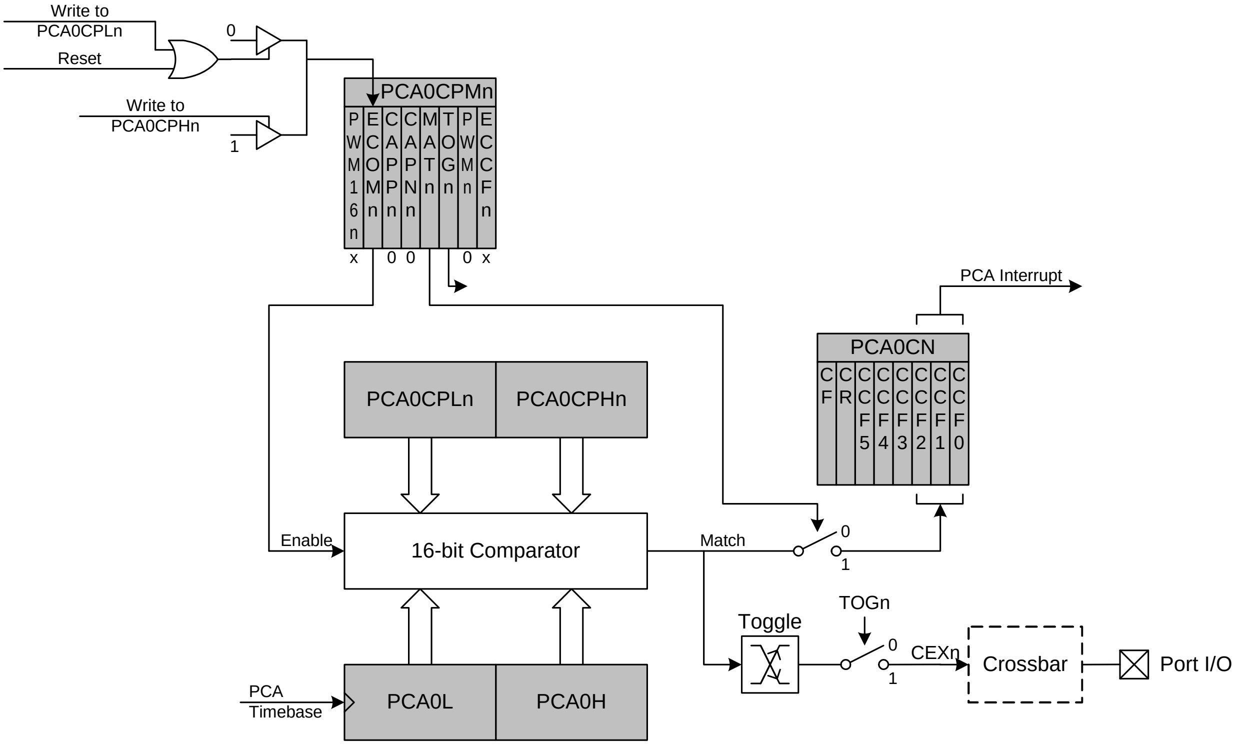


Figure 26.6. PCA High-Speed Output Mode Diagram

26.3.4. Frequency Output Mode

Frequency Output Mode produces a programmable-frequency square wave on the module's associated CEXn pin. The capture/compare module high byte holds the number of PCA clocks to count before the output is toggled. The frequency of the square wave is then defined by Equation 26.1.

$$F_{CEXn} = \frac{F_{PCA}}{2 \times PCA0CPHn}$$

Note: A value of 0x00 in the PCA0CPHn register is equal to 256 for this equation.

Equation 26.1. Square Wave Frequency Output

Where F_{PCA} is the frequency of the clock selected by the CPS[2:0] bits in the PCA mode register, PCA0MD. The lower byte of the capture/compare module is compared to the PCA counter low byte; on a match, CEXn is toggled and the offset held in the high byte is added to the matched value in PCA0CPLn. Frequency Output Mode is enabled by setting the ECOMn, TOGn, and PWMn bits in the PCA0CPMn register. Note that the MATn bit should normally be set to 0 in this mode. If the MATn bit is set to 1, the CCFn flag for the channel will be set when the 16-bit PCA0 counter and the 16-bit capture/compare register for the channel are equal.