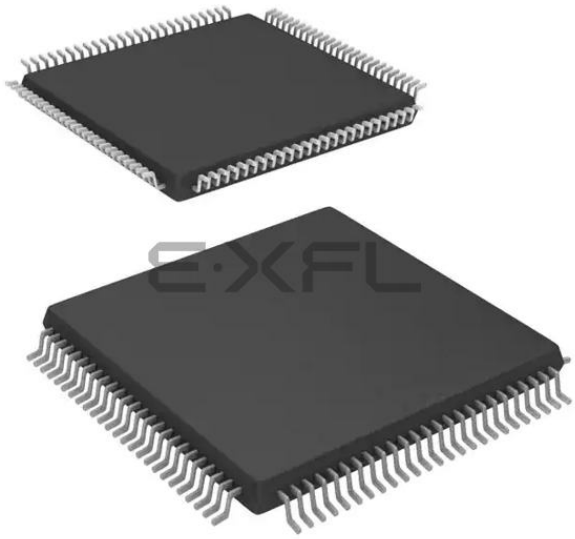


Welcome to E-XFL.COM

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	100MHz
Connectivity	EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	64
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8.25K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x8b, 8x10b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f122r

List of Figures

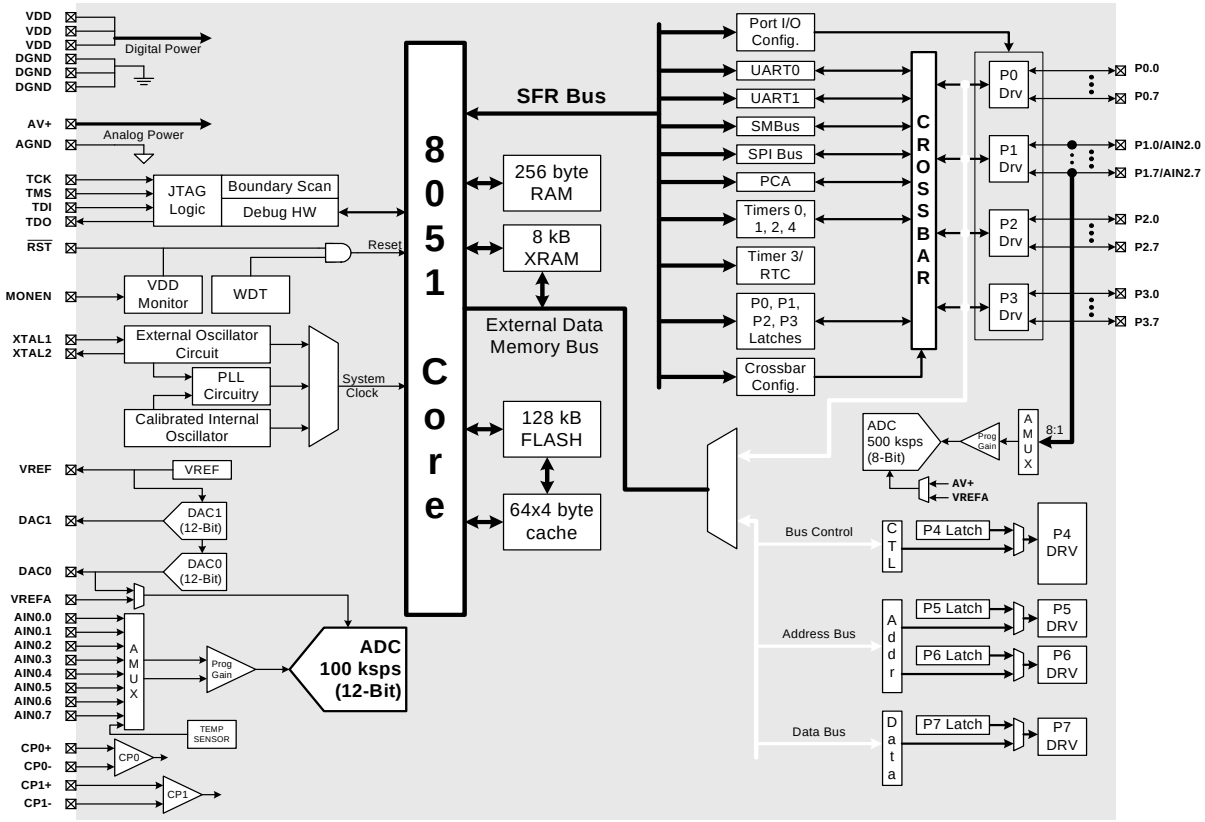


Figure 1.2. C8051F121/125 Block Diagram

1.1.3. Additional Features

Several key enhancements are implemented in the CIP-51 core and peripherals to improve overall performance and ease of use in end applications.

The extended interrupt handler provides 20 interrupt sources into the CIP-51 (as opposed to 7 for the standard 8051), allowing the numerous analog and digital peripherals to interrupt the controller. An interrupt driven system requires less intervention by the MCU, giving it more effective throughput. The extra interrupt sources are very useful when building multi-tasking, real-time systems.

There are up to seven reset sources for the MCU: an on-board V_{DD} monitor, a Watchdog Timer, a missing clock detector, a voltage level detection from Comparator0, a forced software reset, the CNVSTR0 input pin, and the $\overline{RST}$ pin. The $\overline{RST}$ pin is bi-directional, accommodating an external reset, or allowing the internally generated POR to be output on the $\overline{RST}$ pin. Each reset source except for the V_{DD} monitor and Reset Input pin may be disabled by the user in software; the V_{DD} monitor is enabled/disabled via the MONEN pin. The Watchdog Timer may be permanently enabled in software after a power-on reset during MCU initialization.

The MCU has an internal, stand alone clock generator which is used by default as the system clock after any reset. If desired, the clock source may be switched on the fly to the external oscillator, which can use a crystal, ceramic resonator, capacitor, RC, or external clock source to generate the system clock. This can be extremely useful in low power applications, allowing the MCU to run from a slow (power saving) external crystal source, while periodically switching to the 24.5 MHz internal oscillator as needed. Additionally, an on-chip PLL is provided to achieve higher system clock speeds for increased throughput.

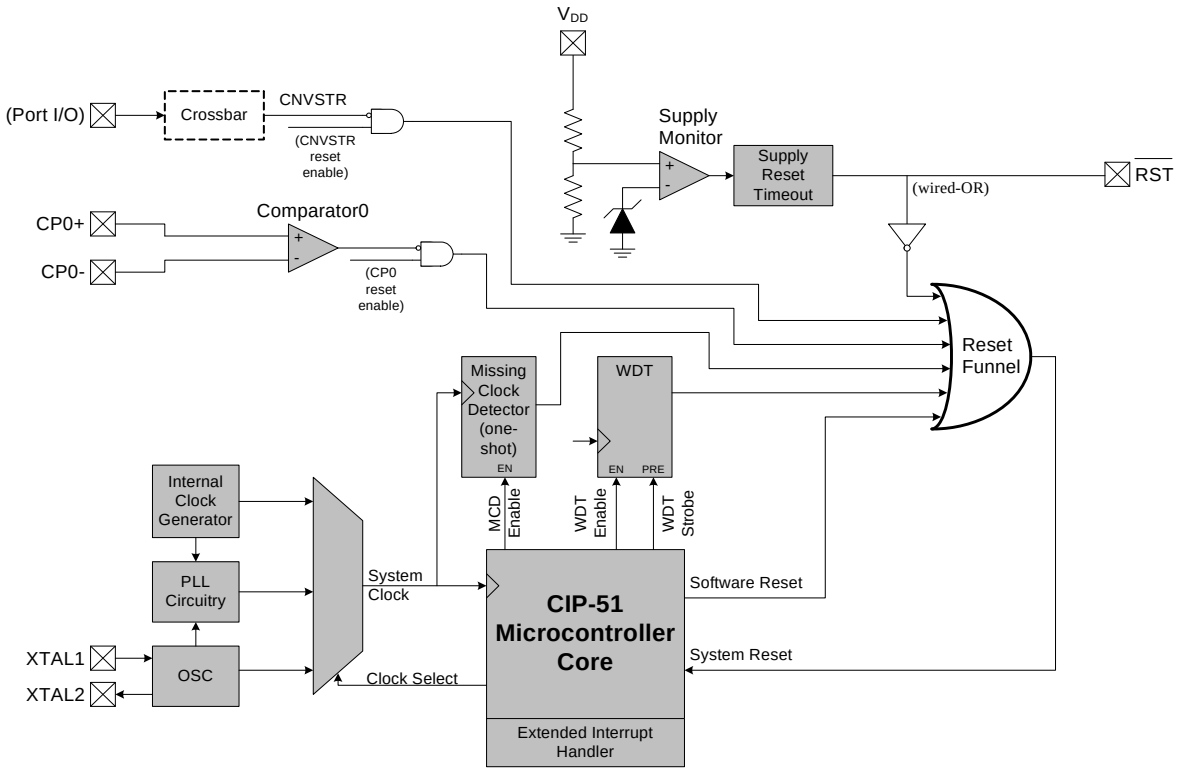


Figure 1.7. On-Board Clock and Reset

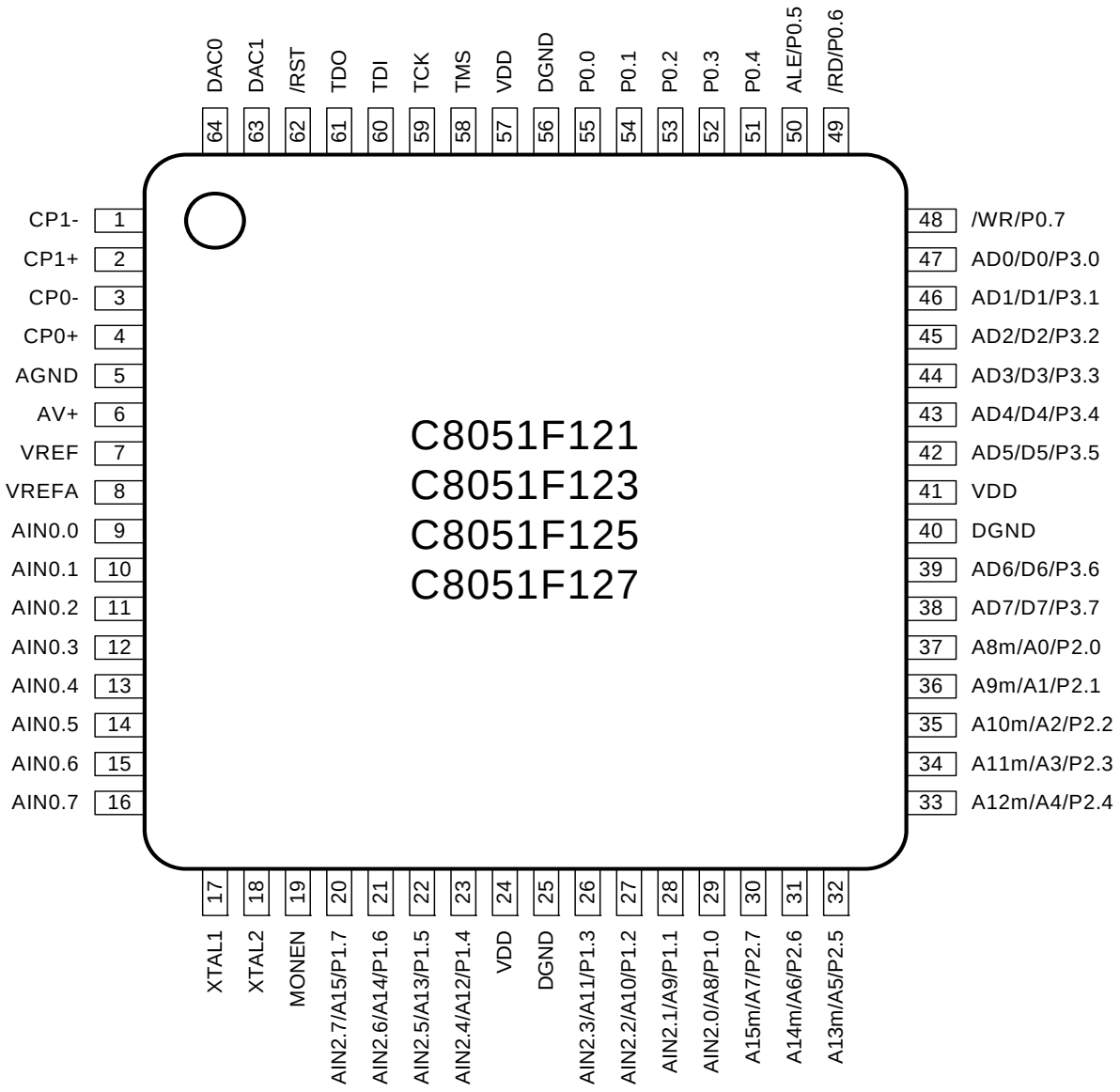


Figure 4.4. C8051F121/3/5/7 Pinout Diagram (TQFP-64)

8. DACs, 12-Bit Voltage Mode (C8051F12x Only)

The C8051F12x devices include two on-chip 12-bit voltage-mode Digital-to-Analog Converters (DACs). Each DAC has an output swing of 0 V to (VREF-1LSB) for a corresponding input code range of 0x000 to 0xFFF. The DACs may be enabled/disabled via their corresponding control registers, DAC0CN and DAC1CN. While disabled, the DAC output is maintained in a high-impedance state, and the DAC supply current falls to 1 µA or less. The voltage reference for each DAC is supplied at the VREFD pin (C8051F120/2/4/6 devices) or the VREF pin (C8051F121/3/5/7 devices). Note that the VREF pin on C8051F121/3/5/7 devices may be driven by the internal voltage reference or an external source. If the internal voltage reference is used it must be enabled in order for the DAC outputs to be valid. See **Section "9. Voltage Reference" on page 113** for more information on configuring the voltage reference for the DACs.

8.1. DAC Output Scheduling

Each DAC features a flexible output update mechanism which allows for seamless full-scale changes and supports jitter-free updates for waveform generation. The following examples are written in terms of DAC0, but DAC1 operation is identical.

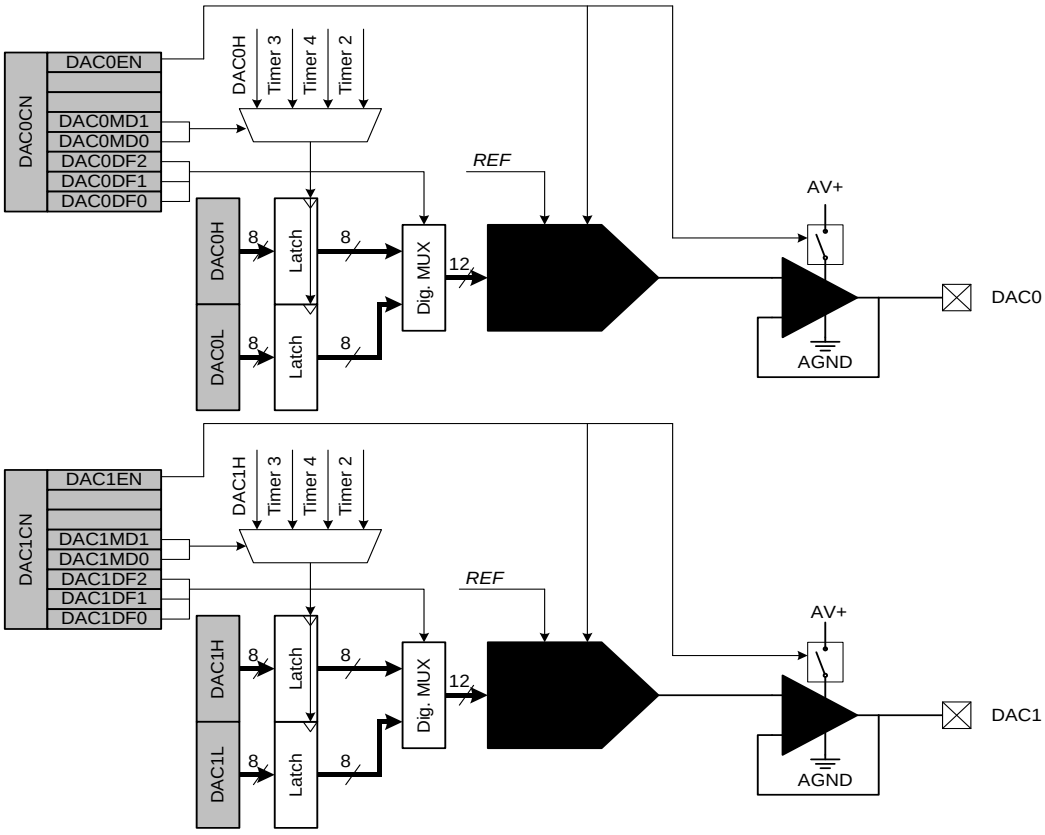


Figure 8.1. DAC Functional Block Diagram

8.1.1. Update Output On-Demand

In its default mode (DAC0CN.[4:3] = ‘00’) the DAC0 output is updated “on-demand” on a write to the high-byte of the DAC0 data register (DAC0H). It is important to note that writes to DAC0L are held, and have no effect on the DAC0 output until a write to DAC0H takes place. If writing a full 12-bit word to the DAC data registers, the 12-bit data word is written to the low byte (DAC0L) and high byte (DAC0H) data registers. Data is latched into DAC0 after a write to the corresponding DAC0H register, **so the write sequence should be DAC0L followed by DAC0H** if the full 12-bit resolution is required. The DAC can be used in 8-bit mode by initializing DAC0L to the desired value (typically 0x00), and writing data to only DAC0H (also see **Section 8.2** for information on formatting the 12-bit DAC data word within the 16-bit SFR space).

8.1.2. Update Output Based on Timer Overflow

Similar to the ADC operation, in which an ADC conversion can be initiated by a timer overflow independently of the processor, the DAC outputs can use a Timer overflow to schedule an output update event. This feature is useful in systems where the DAC is used to generate a waveform of a defined sampling rate by eliminating the effects of variable interrupt latency and instruction execution on the timing of the DAC output. When the DAC0MD bits (DAC0CN.[4:3]) are set to ‘01’, ‘10’, or ‘11’, writes to both DAC data registers (DAC0L and DAC0H) are held until an associated Timer overflow event (Timer 3, Timer 4, or Timer 2, respectively) occurs, at which time the DAC0H:DAC0L contents are copied to the DAC input latches allowing the DAC output to change to the new value.

8.2. DAC Output Scaling/Justification

In some instances, input data should be shifted prior to a DAC0 write operation to properly justify data within the DAC input registers. This action would typically require one or more load and shift operations, adding software overhead and slowing DAC throughput. To alleviate this problem, the data-formatting feature provides a means for the user to program the orientation of the DAC0 data word within data registers DAC0H and DAC0L. The three DAC0DF bits (DAC0CN.[2:0]) allow the user to specify one of five data word orientations as shown in the DAC0CN register definition.

DAC1 is functionally the same as DAC0 described above. The electrical specifications for both DAC0 and DAC1 are given in Table 8.1.

NOTES:

13. 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 configuration.
- 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 are not altered.

The I/O port latches are reset to 0xFF (all logic 1's), activating internal weak pullups during and after the reset. For V_{DD} Monitor resets, the $\overline{RST}$ pin is driven low until the end of the V_{DD} reset timeout.

On exit from the reset state, the program counter (PC) is reset, and the system clock defaults to the internal oscillator running at its lowest frequency. Refer to Section "**14. Oscillators**" on page **185** for information on selecting and configuring the system clock source. The Watchdog Timer is enabled using its longest timeout interval (see Section "**13.7. Watchdog Timer Reset**" on page **179**). Once the system clock source is stable, program execution begins at location 0x0000.

There are seven sources for putting the MCU into the reset state: power-on, power-fail, external $\overline{RST}$ pin, external CNVSTR0 signal, software command, Comparator0, Missing Clock Detector, and Watchdog Timer. Each reset source is described in the following sections.

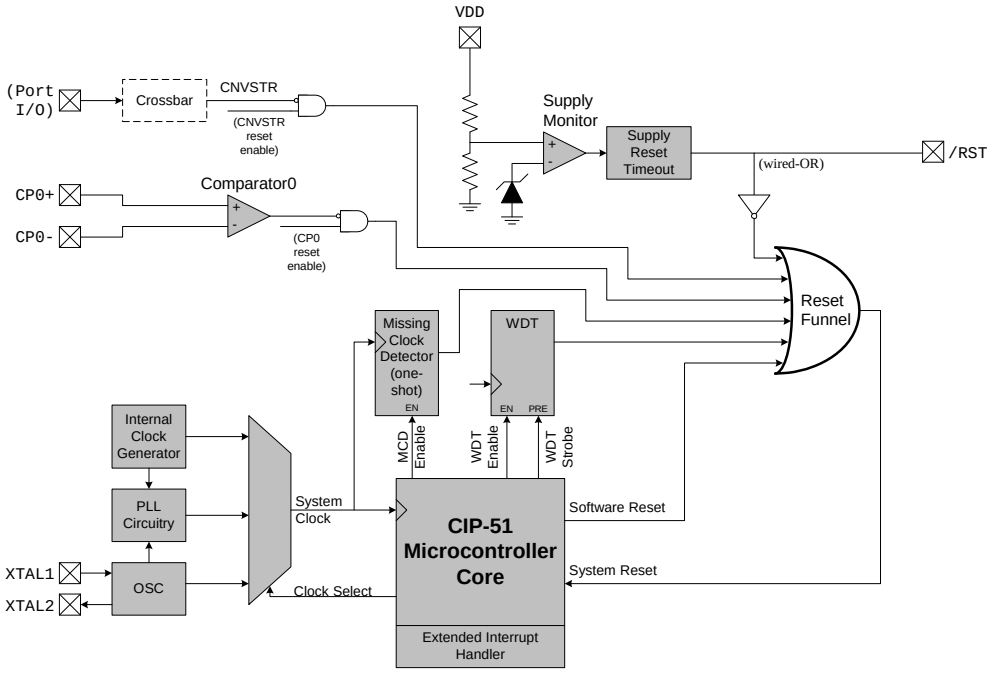


Figure 13.1. Reset Sources

Table 13.1. Reset Electrical Characteristics

–40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
$\overline{\text{RST}}$ Output Low Voltage	I_{OL} = 8.5 mA, V_{DD} = 2.7 to 3.6 V	—	—	0.6	V
$\overline{\text{RST}}$ Input High Voltage		0.7 x V_{DD}	—	—	V
$\overline{\text{RST}}$ Input Low Voltage		—	—	0.3 x V_{DD}	
$\overline{\text{RST}}$ Input Leakage Current	$\overline{\text{RST}}$ = 0.0 V	—	50	—	µA
V_{DD} for $\overline{\text{RST}}$ Output Valid		1.0	—	—	V
AV+ for $\overline{\text{RST}}$ Output Valid		1.0	—	—	V
V_{DD} POR Threshold (V_{RST})*		2.40	2.55	2.70	V
Minimum $\overline{\text{RST}}$ Low Time to Generate a System Reset		10	—	—	ns
Reset Time Delay	$\overline{\text{RST}}$ rising edge after V_{DD} crosses V_{RST} threshold	80	100	120	ms
Missing Clock Detector Timeout	Time from last system clock to reset initiation	100	220	500	µs
***Note:** When operating at frequencies above 50 MHz, minimum V_{DD} supply Voltage is 3.0 V.					

Table 14.3. PLL Lock Timing Characteristics

–40 to +85 °C unless otherwise specified

Input Frequency	Multiplier (Pll0mul)	Pll0flt Setting	Output Frequency	Min	Typ	Max	Units
5 MHz	20	0x0F	100 MHz		202		µs
	13	0x0F	65 MHz		115		µs
	16	0x1F	80 MHz		241		µs
	9	0x1F	45 MHz		116		µs
	12	0x2F	60 MHz		258		µs
	6	0x2F	30 MHz		112		µs
	10	0x3F	50 MHz		263		µs
	5	0x3F	25 MHz		113		µs
25 MHz	4	0x01	100 MHz		42		µs
	2	0x01	50 MHz		33		µs
	3	0x11	75 MHz		48		µs
	2	0x11	50 MHz		17		µs
	2	0x21	50 MHz		42		µs
	1	0x21	25 MHz		33		µs
	2	0x31	50 MHz		60		µs
	1	0x31	25 MHz		25		µs

5. Select the memory mode (on-chip only, split mode without bank select, split mode with bank select, or off-chip only).
6. Set up timing to interface with off-chip memory or peripherals.

Each of these five steps is explained in detail in the following sections. The Port selection, Multiplexed mode selection, and Mode bits are located in the EMI0CF register shown in SFR Definition 17.2.

17.3. Port Selection and Configuration

The External Memory Interface can appear on Ports 3, 2, 1, and 0 (All Devices) or on Ports 7, 6, 5, and 4 (100-pin TQFP devices only), depending on the state of the PRTSEL bit (EMI0CF.5). If the lower Ports are selected, the EMIFLE bit (XBR2.1) must be set to a '1' so that the Crossbar will skip over P0.7 (/WR), P0.6 (/RD), and if multiplexed mode is selected P0.5 (ALE). For more information about the configuring the Crossbar, see **Section "18.1. Ports 0 through 3 and the Priority Crossbar Decoder" on page 238**.

The External Memory Interface claims the associated Port pins for memory operations ONLY during the execution of an off-chip MOVX instruction. Once the MOVX instruction has completed, control of the Port pins reverts to the Port latches or to the Crossbar (on Ports 3, 2, 1, and 0). See **Section "18. Port Input/ Output" on page 235** for more information about the Crossbar and Port operation and configuration. **The Port latches should be explicitly configured to 'park' the External Memory Interface pins in a dormant state, most commonly by setting them to a logic 1**.

During the execution of the MOVX instruction, the External Memory Interface will explicitly disable the drivers on all Port pins that are acting as Inputs (Data[7:0] during a READ operation, for example). The Output mode of the Port pins (whether the pin is configured as Open-Drain or Push-Pull) is unaffected by the External Memory Interface operation, and remains controlled by the PnMDOUT registers. In most cases, the output modes of all EMIF pins should be configured for push-pull mode. See"Configuring the Output Modes of the Port Pins" on page 239.

SFR Definition 17.1. EMI0CN: External Memory Interface Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
PGSEL7	PGSEL6	PGSEL5	PGSEL4	PGSEL3	PGSEL2	PGSEL1	PGSEL0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xA2
SFR Page: 0

Bits7–0: PGSEL[7:0]: XRAM Page Select Bits.
The XRAM Page Select Bits provide the high byte of the 16-bit external data memory address when using an 8-bit MOVX command, effectively selecting a 256-byte page of RAM.
0x00: 0x0000 to 0x00FF
0x01: 0x0100 to 0x01FF
...
0xFE: 0xFE00 to 0xFEFF
0xFF: 0xFF00 to 0xFFFF

17.6.2. Multiplexed Mode

17.6.2.1.16-bit MOVX: EMI0CF[4:2] = ‘001’, ‘010’, or ‘011’

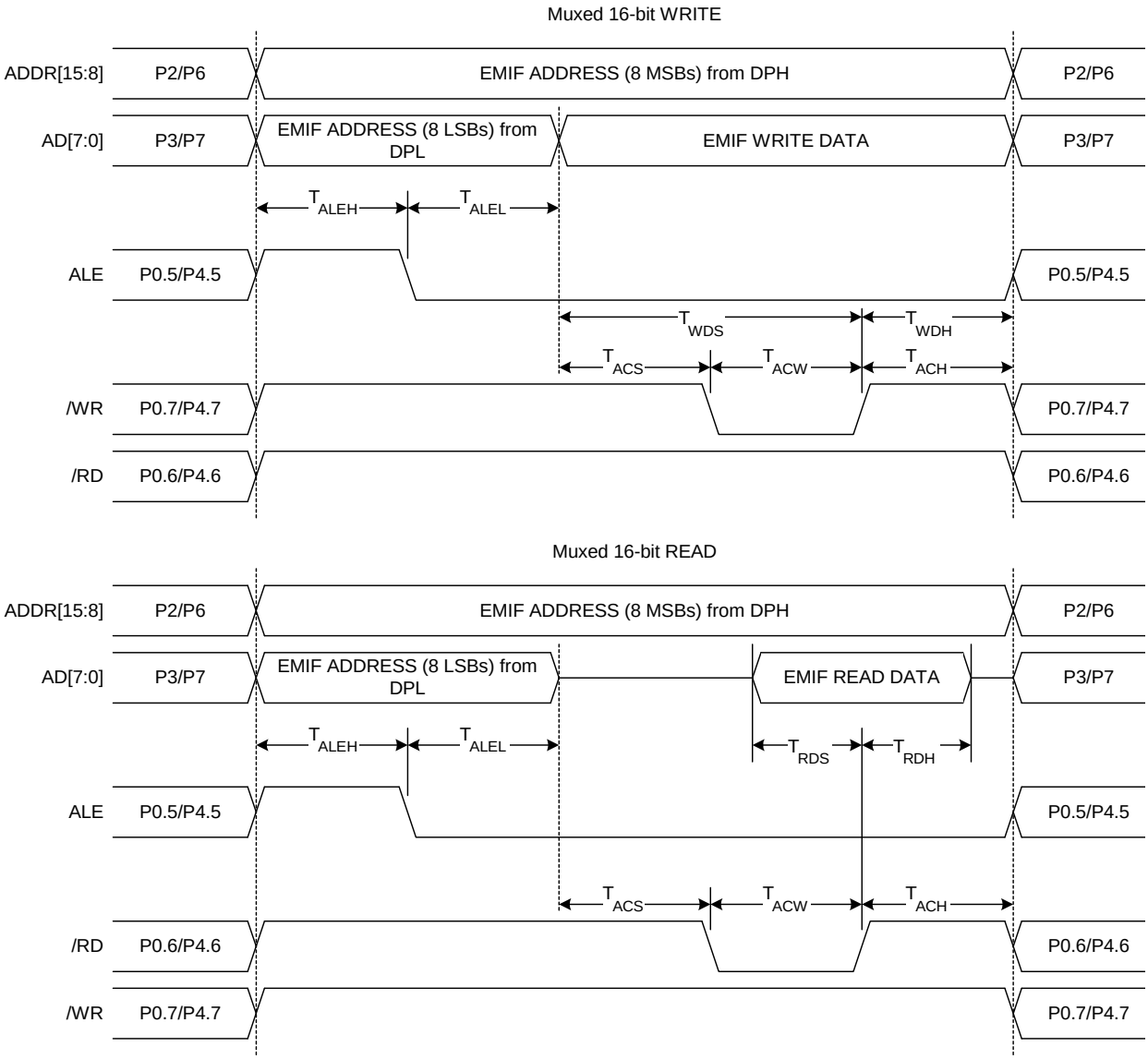


Figure 17.7. Multiplexed 16-bit MOVX Timing

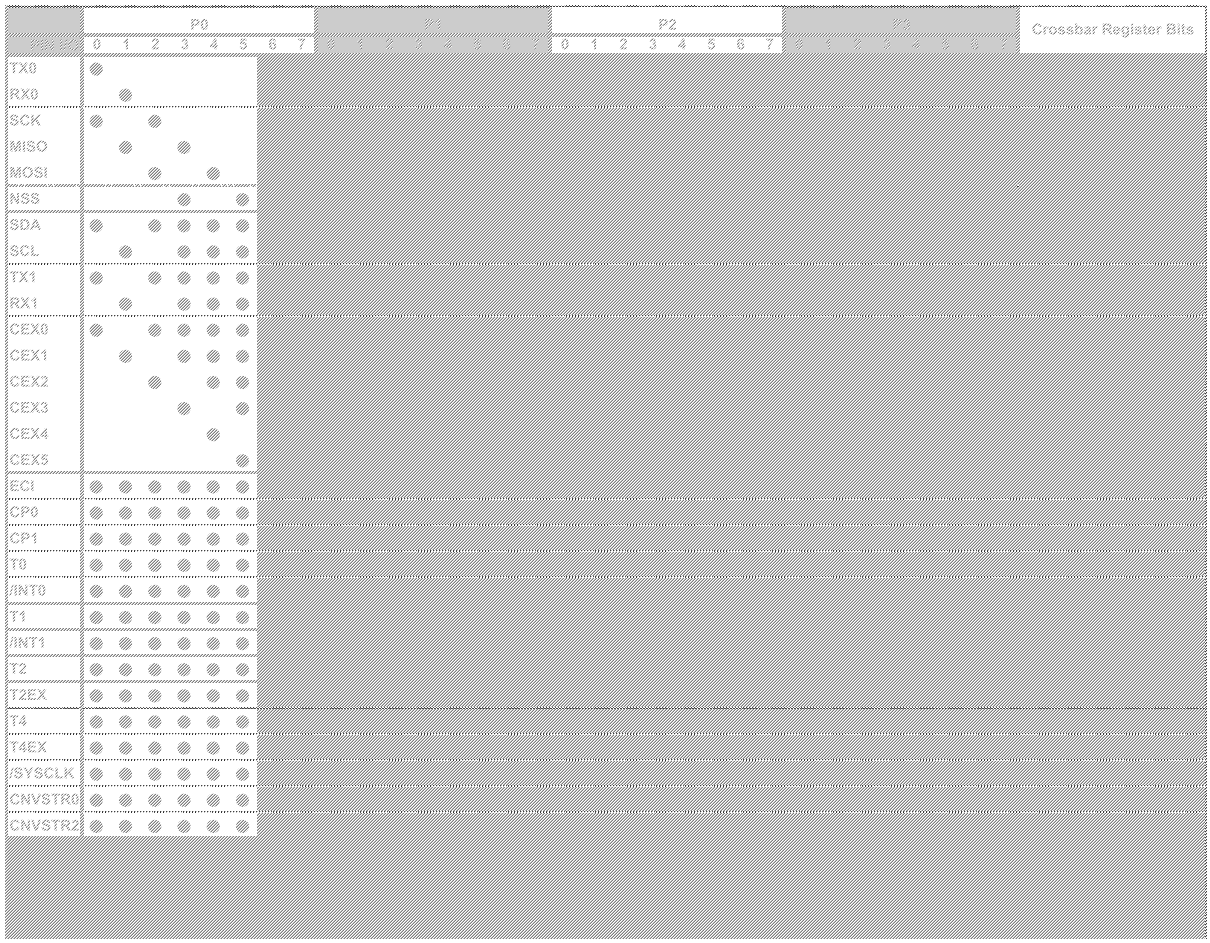


Figure 18.5. Priority Crossbar Decode Table (EMIFLE = 1; EMIF in Non-Multiplexed Mode; P1MDIN = 0xFF)

18.1.7. Crossbar Pin Assignment Example

In this example (Figure 18.6), we configure the Crossbar to allocate Port pins for UART0, the SMBus, UART1, /INT0, and /INT1 (8 pins total). Additionally, we configure the External Memory Interface to operate in Multiplexed mode and to appear on the Low ports. Further, we configure P1.2, P1.3, and P1.4 for Analog Input mode so that the voltages at these pins can be measured by ADC2. The configuration steps are as follows:

1. XBR0, XBR1, and XBR2 are set such that UART0EN = 1, SMB0EN = 1, INT0E = 1, INT1E = 1, and EMIFLE = 1. Thus: XBR0 = 0x05, XBR1 = 0x14, and XBR2 = 0x02.
2. We configure the External Memory Interface to use Multiplexed mode and to appear on the Low ports. PRTSEL = 0, EMD2 = 0.
3. We configure the desired Port 1 pins to Analog Input mode by setting P1MDIN to 0xE3 (P1.4, P1.3, and P1.2 are Analog Inputs, so their associated P1MDIN bits are set to logic 0).
4. We enable the Crossbar by setting XBARE = 1: XBR2 = 0x42.
   - UART0 has the highest priority, so P0.0 is assigned to TX0, and P0.1 is assigned to RX0.
   - The SMBus is next in priority order, so P0.2 is assigned to SDA, and P0.3 is assigned to SCL.
   - UART1 is next in priority order, so P0.4 is assigned to TX1. Because the External Memory Interface is selected on the lower Ports, EMIFLE = 1, which causes the Crossbar to skip P0.6 (/RD) and P0.7 (/WR). Because the External Memory Interface is configured in Multiplexed mode, the Crossbar will also skip P0.5 (ALE). RX1 is assigned to the next non-skipped pin, which in this case is P1.0.
   - /INT0 is next in priority order, so it is assigned to P1.1.
   - P1MDIN is set to 0xE3, which configures P1.2, P1.3, and P1.4 as Analog Inputs, causing the Crossbar to skip these pins.
   - /INT1 is next in priority order, so it is assigned to the next non-skipped pin, which is P1.5.
   - The External Memory Interface will drive Ports 2 and 3 (denoted by red dots in Figure 18.6) during the execution of an off-chip MOVX instruction.
5. We set the UART0 TX pin (TX0, P0.0) and UART1 TX pin (TX1, P0.4) outputs to Push-Pull by setting P0MDOUT = 0x11.
6. We configure all EMIF-controlled pins to push-pull output mode by setting P0MDOUT |= 0xE0; P2MDOUT = 0xFF; P3MDOUT = 0xFF.
7. We explicitly disable the output drivers on the 3 Analog Input pins by setting P1MDOUT = 0x00 (configure outputs to Open-Drain) and P1 = 0xFF (a logic 1 selects the high-impedance state).

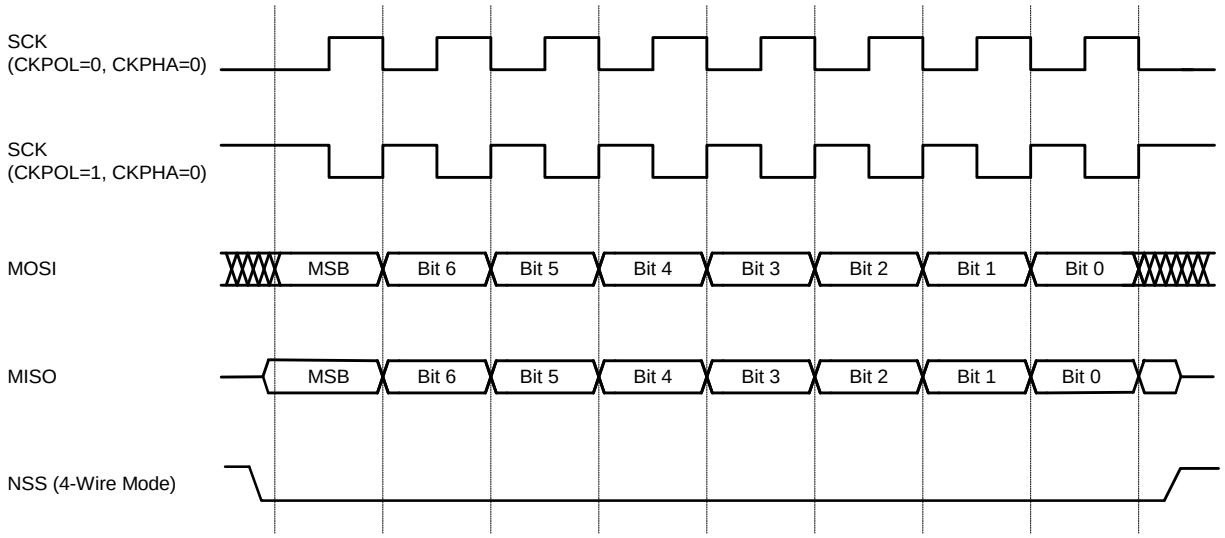


Figure 20.6. Slave Mode Data/Clock Timing (CKPHA = 0)

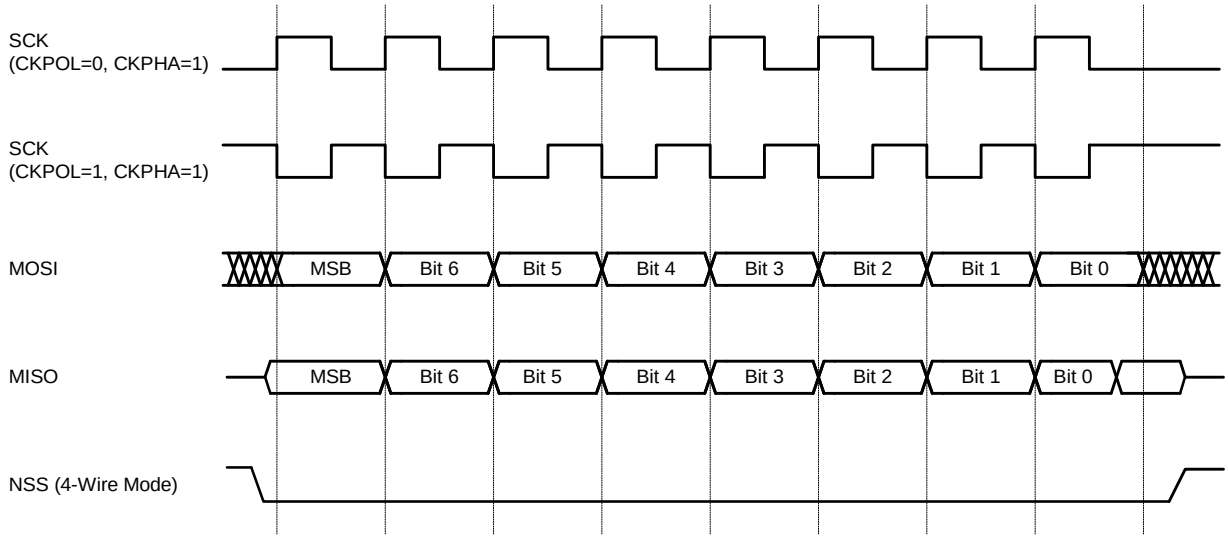


Figure 20.7. Slave Mode Data/Clock Timing (CKPHA = 1)

20.6. SPI Special Function Registers

SPI0 is accessed and controlled through four special function registers in the system controller: SPI0CN Control Register, SPI0DAT Data Register, SPI0CFG Configuration Register, and SPI0CKR Clock Rate Register. The four special function registers related to the operation of the SPI0 Bus are described in the following figures.

SFR Definition 20.1. SPI0CFG: SPI0 Configuration

R	R/W	R/W	R/W	R	R	R	R	Reset Value
SPIBSY	MSTEN	CKPHA	CKPOL	SLVSEL	NSSIN	SRMT	RXBMT	00000111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x9A
SFR Page: 0

Bit 7: SPIBSY: SPI Busy (read only).
This bit is set to logic 1 when a SPI transfer is in progress (Master or slave Mode).

Bit 6: MSTEN: Master Mode Enable.
0: Disable master mode. Operate in slave mode.
1: Enable master mode. Operate as a master.

Bit 5: CKPHA: SPI0 Clock Phase.
This bit controls the SPI0 clock phase.
0: Data centered on first edge of SCK period.*
1: Data centered on second edge of SCK period.*

Bit 4: CKPOL: SPI0 Clock Polarity.
This bit controls the SPI0 clock polarity.
0: SCK line low in idle state.
1: SCK line high in idle state.

Bit 3: SLVSEL: Slave Selected Flag (read only).
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.

Bit 2: NSSIN: NSS Instantaneous Pin Input (read only).
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.

Bit 1: SRMT: Shift Register Empty (Valid in Slave Mode, read 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.
NOTE: SRMT = 1 when in Master Mode.

Bit 0: RXBMT: Receive Buffer Empty (Valid in Slave Mode, read 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.
NOTE: 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 20.1 for timing parameters.

21. UART0

UART0 is an enhanced serial port with frame error detection and address recognition hardware. UART0 may operate in full-duplex asynchronous or half-duplex synchronous modes, and mutiproccessor communication is fully supported. Receive data is buffered in a holding register, allowing UART0 to start reception of a second incoming data byte before software has finished reading the previous data byte. A Receive Overrun bit indicates when new received data is latched into the receive buffer before the previously received byte has been read.

UART0 is accessed via its associated SFR's, Serial Control (SCON0) and Serial Data Buffer (SBUF0). The single SBUF0 location provides access to both transmit and receive registers. Reading SCON0 accesses the Receive register and writing SCON0 accesses the Transmit register.

UART0 may be operated in polled or interrupt mode. UART0 has two sources of interrupts: a Transmit Interrupt flag, TI0 (SCON0.1) set when transmission of a data byte is complete, and a Receive Interrupt flag, RI0 (SCON0.0) set when reception of a data byte is complete. UART0 interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine; they must be cleared manually by software. This allows software to determine the cause of the UART0 interrupt (transmit complete or receive complete).

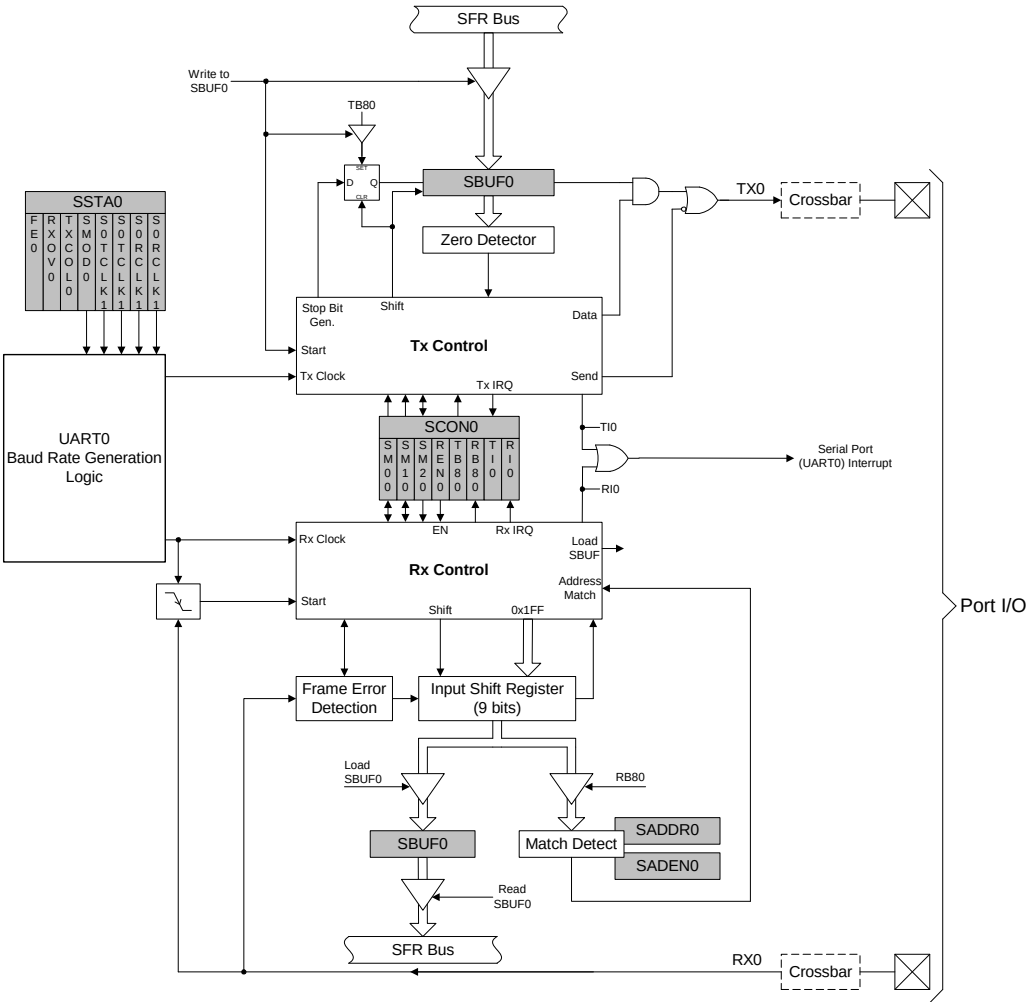


Figure 21.1. UART0 Block Diagram

SFR Definition 24.2. PCA0MD: PCA0 Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CIDL	-	-	-	CPS2	CPS1	CPS0	ECF	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xD9
SFR Page: 0

Bit7: CIDL: PCA0 Counter/Timer Idle Control.
Specifies PCA0 behavior when CPU is in Idle Mode.
0: PCA0 continues to function normally while the system controller is in Idle Mode.
1: PCA0 operation is suspended while the system controller is in Idle Mode.

Bits6–4: UNUSED. Read = 000b, Write = don't care.

Bits3–1: CPS2-CPS0: PCA0 Counter/Timer Pulse Select.
These bits select the timebase source for the PCA0 counter

CPS2	CPS1	CPS0	Timebase
0	0	0	System clock divided by 12
0	0	1	System clock divided by 4
0	1	0	Timer 0 overflow
0	1	1	High-to-low transitions on ECI (max rate = system clock divided by 4)
1	0	0	System clock
1	0	1	External clock divided by 8 (synchronized with system clock)
1	1	0	Reserved
1	1	1	Reserved

Bit0: ECF: PCA Counter/Timer Overflow Interrupt Enable.
This bit sets the masking of the PCA0 Counter/Timer Overflow (CF) interrupt.
0: Disable the CF interrupt.
1: Enable a PCA0 Counter/Timer Overflow interrupt request when CF (PCA0CN.7) is set.

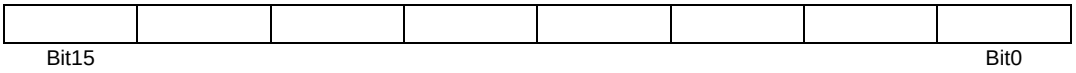
25. JTAG (IEEE 1149.1)

Each MCU has an on-chip JTAG interface and logic to support boundary scan for production and in-system testing, Flash read/write operations, and non-intrusive in-circuit debug. The JTAG interface is fully compliant with the IEEE 1149.1 specification. Refer to this specification for detailed descriptions of the Test Interface and Boundary-Scan Architecture. Access of the JTAG Instruction Register (IR) and Data Registers (DR) are as described in the Test Access Port and Operation of the IEEE 1149.1 specification.

The JTAG interface is accessed via four dedicated pins on the MCU: TCK, TMS, TDI, and TDO.

Through the 16-bit JTAG Instruction Register (IR), any of the eight instructions shown in Figure 25.1 can be commanded. There are three DR's associated with JTAG Boundary-Scan, and four associated with Flash read/write operations on the MCU.

JTAG Register Definition 25.1. IR: JTAG Instruction Register



IR Value	Instruction	Description
0x0000	EXTEST	Selects the Boundary Data Register for control and observability of all device pins
0x0002	SAMPLE/ PRELOAD	Selects the Boundary Data Register for observability and presetting the scan-path latches
0x0004	IDCODE	Selects device ID Register
0xFFFF	BYPASS	Selects Bypass Data Register
0x0082	Flash Control	Selects FLASHCON Register to control how the interface logic responds to reads and writes to the FLASHDAT Register
0x0083	Flash Data	Selects FLASHDAT Register for reads and writes to the Flash memory
0x0084	Flash Address	Selects FLASHADR Register which holds the address of all Flash read, write, and erase operations
0x0085	Flash Scale	Selects FLASHSCL Register which controls the Flash one-shot timer and read-always enable