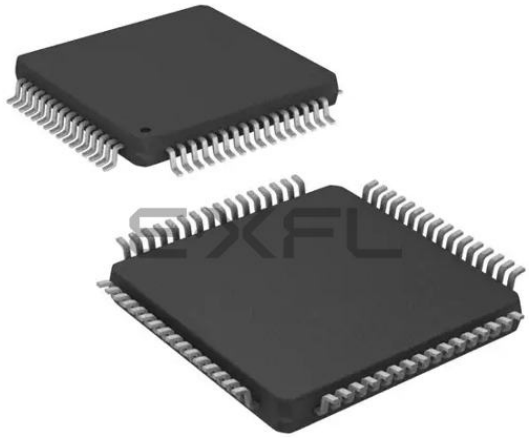


Welcome to E-XFL.COM

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	25MHz
Connectivity	EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	32
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 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	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f023

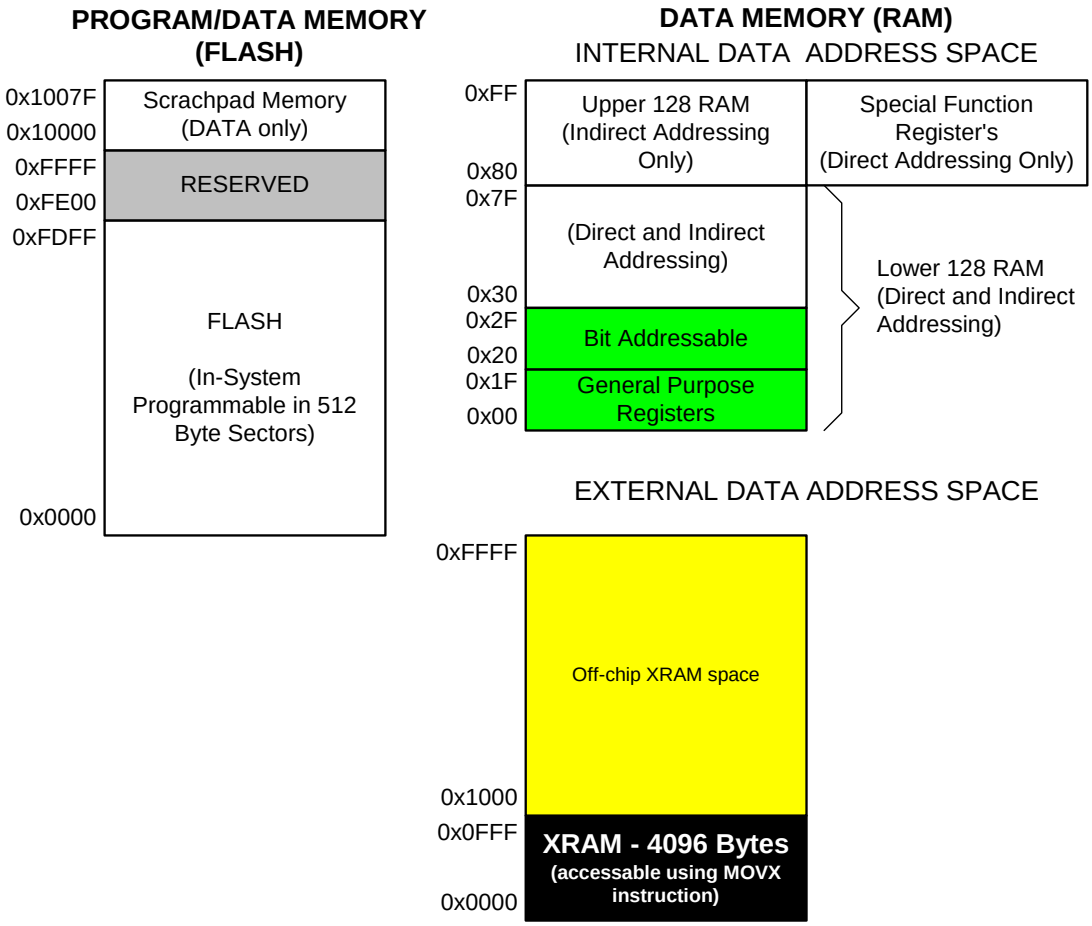
1.2. On-Chip Memory

The CIP-51 has a standard 8051 program and data address configuration. It includes 256 bytes of data RAM, with the upper 128 bytes dual-mapped. Indirect addressing accesses the upper 128 bytes of general purpose RAM, and direct addressing accesses the 128 byte SFR address space. The lower 128 bytes of RAM are accessible via direct and indirect addressing. The first 32 bytes are addressable as four banks of general purpose registers, and the next 16 bytes can be byte addressable or bit addressable.

The CIP-51 in the C8051F020/1/2/3 MCUs additionally has an on-chip 4k byte RAM block and an external memory interface (EMIF) for accessing off-chip data memory. The on-chip 4k byte block can be addressed over the entire 64k external data memory address range (overlapping 4k boundaries). External data memory address space can be mapped to on-chip memory only, off-chip memory only, or a combination of the two (addresses up to 4k directed to on-chip, above 4k directed to EMIF). The EMIF is also configurable for multiplexed or non-multiplexed address/data lines.

The MCU's program memory consists of 64k bytes of FLASH. This memory may be reprogrammed in-system in 512 byte sectors, and requires no special off-chip programming voltage. The 512 bytes from addresses 0xFE00 to 0xFFFF are reserved for factory use. There is also a single 128 byte sector at address 0x10000 to 0x1007F, which may be useful as a small table for software constants. See Figure 1.7 for the MCU system memory map.

Figure 1.7. On-Chip Memory Map



4. PINOUT AND PACKAGE DEFINITIONS

Table 4.1. Pin Definitions

Name	Pin Numbers		Type	Description
	F020	F021		
	F022	F023		
VDD	37, 64, 90	24, 41, 57		Digital Supply Voltage. Must be tied to +2.7 to +3.6 V.
DGND	38, 63, 89	25, 40, 56		Digital Ground. Must be tied to Ground.
AV+	11, 14	6		Analog Supply Voltage. Must be tied to +2.7 to +3.6 V.
AGND	10, 13	5		Analog Ground. Must be tied to Ground.
TMS	1	58	D In	JTAG Test Mode Select with internal pull-up.
TCK	2	59	D In	JTAG Test Clock with internal pull-up.
TDI	3	60	D In	JTAG Test Data Input with internal pull-up. TDI is latched on the rising edge of TCK.
TDO	4	61	D Out	JTAG Test Data Output with internal pull-up. Data is shifted out on TDO on the falling edge of TCK. TDO output is a tri-state driver.
/RST	5	62	D I/O	Device Reset. Open-drain output of internal VDD monitor. Is driven low when VDD is <2.7 V and MONEN is high. An external source can initiate a system reset by driving this pin low.
XTAL1	26	17	A In	Crystal Input. This pin is the return for the internal oscillator circuit for a crystal or ceramic resonator. For a precision internal clock, connect a crystal or ceramic resonator from XTAL1 to XTAL2. If overdriven by an external CMOS clock, this becomes the system clock.
XTAL2	27	18	A Out	Crystal Output. This pin is the excitation driver for a crystal or ceramic resonator.
MONEN	28	19	D In	VDD Monitor Enable. When tied high, this pin enables the internal VDD monitor, which forces a system reset when VDD is < 2.7 V. When tied low, the internal VDD monitor is disabled.
VREF	12	7	A I/O	Bandgap Voltage Reference Output (all devices). DAC Voltage Reference Input (F021/3 only).
VREFA		8	A In	ADC0 and ADC1 Voltage Reference Input.
VREF0	16		A In	ADC0 Voltage Reference Input.
VREF1	17		A In	ADC1 Voltage Reference Input.
VREFD	15		A In	DAC Voltage Reference Input.

7.2. ADC1 Modes of Operation

ADC1 has a maximum conversion speed of 500 ksps. The ADC1 conversion clock (SAR1 clock) is a divided version of the system clock, determined by the AD1SC bits in the ADC1CF register (system clock divided by (AD1SC + 1) for 0 ≤ AD1SC ≤ 31). The maximum ADC1 conversion clock is 6 MHz.

7.2.1. Starting a Conversion

A conversion can be initiated in one of five ways, depending on the programmed states of the ADC1 Start of Conversion Mode bits (AD1CM2-0) in register ADC1CN. Conversions may be initiated by:

1. Writing a '1' to the AD1BUSY bit of ADC1CN;
2. A Timer 3 overflow (i.e. timed continuous conversions);
3. A rising edge detected on the external ADC convert start signal, CNVSTR;
4. A Timer 2 overflow (i.e. timed continuous conversions);
5. Writing a '1' to the AD0BUSY of register ADC0CN (initiate conversion of ADC1 and ADC0 with a single software command).

During conversion, the AD1BUSY bit is set to logic 1 and restored to 0 when conversion is complete. The falling edge of AD1BUSY triggers an interrupt (when enabled) and sets the interrupt flag in ADC1CN. Converted data is available in the ADC1 data word, ADC1.

When a conversion is initiated by writing a '1' to AD1BUSY, it is recommended to poll AD1INT to determine when the conversion is complete. The recommended procedure is:

Step 1. Write a '0' to AD1INT;
Step 2. Write a '1' to AD1BUSY;
Step 3. Poll AD1INT for '1';
Step 4. Process ADC1 data.

7.2.2. Tracking Modes

The AD1TM bit in register ADC1CN controls the ADC1 track-and-hold mode. In its default state, the ADC1 input is continuously tracked, except when a conversion is in progress. When the AD1TM bit is logic 1, ADC1 operates in low-power track-and-hold mode. In this mode, each conversion is preceded by a tracking period of 3 SAR clocks (after the start-of-conversion signal). When the CNVSTR signal is used to initiate conversions in low-power tracking mode, ADC1 tracks only when CNVSTR is low; conversion begins on the rising edge of CNVSTR (see Figure 7.2). Tracking can also be disabled (shutdown) when the entire chip is in low power standby or sleep modes. Low-power Track-and-Hold mode is also useful when AMUX or PGA settings are frequently changed, due to the settling time requirements described in **Section "7.2.3. Settling Time Requirements" on page 78**.

Figure 7.7. ADC1: ADC1 Data Word Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9C

Bits7-0: ADC1 Data Word.

Figure 7.8. ADC1 Data Word Example

8-bit ADC Data Word appears in the ADC1 Data Word Register as follows:
Example: ADC1 Data Word Conversion Map, AIN1.0 Input (AMX1SL = 0x00)

AIN1.0-AGND (Volts)	ADC1
VREF * (255/256)	0xFF
VREF / 2	0x80
VREF * (127/256)	0x7F
0	0x00

$$Code = Vin \times \frac{Gain}{VREF} \times 256$$

Table 12.3. Special Function Registers

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

Register	Address	Description	Page No.
ADC0LTH	0xC7	ADC0 Less-Than High	page 53*, page 69**
ADC0LTL	0xC6	ADC0 Less-Than Low	page 53*, page 69**
ADC1CF	0xAB	ADC1 Analog Multiplexer Configuration	page 79
ADC1CN	0xAA	ADC1 Control	page 80
ADC1	0x9C	ADC1 Data Word	page 81
AMX0CF	0xBA	ADC0 Multiplexer Configuration	page 47*, page 63**
AMX0SL	0xBB	ADC0 Multiplexer Channel Select	page 48*, page 64**
AMX1SL	0xAC	ADC1 Analog Multiplexer Channel Select	page 79
B	0xF0	B Register	page 115
CKCON	0x8E	Clock Control	page 226
CPT0CN	0x9E	Comparator 0 Control	page 97
CPT1CN	0x9F	Comparator 1 Control	page 98
DAC0CN	0xD4	DAC0 Control	page 86
DAC0H	0xD3	DAC0 High	page 85
DAC0L	0xD2	DAC0 Low	page 85
DAC1CN	0xD7	DAC1 Control	page 88
DAC1H	0xD6	DAC1 High Byte	page 87
DAC1L	0xD5	DAC1 Low Byte	page 87
DPH	0x83	Data Pointer High	page 113
DPL	0x82	Data Pointer Low	page 113
EIE1	0xE6	Extended Interrupt Enable 1	page 121
EIE2	0xE7	Extended Interrupt Enable 2	page 122
EIP1	0xF6	External Interrupt Priority 1	page 123
EIP2	0xF7	External Interrupt Priority 2	page 124
EMI0CN	0xAF	External Memory Interface Control	page 147
EMI0CF	0xA3	EMIF Configuration	page 147
EMI0TC	0xA1	EMIF Timing Control	page 152
FLACL	0xB7	FLASH Access Limit	page 142
FLSCL	0xB6	FLASH Scale	page 143
IE	0xA8	Interrupt Enable	page 119
IP	0xB8	Interrupt Priority	page 120
OSCICN	0xB2	Internal Oscillator Control	page 136
OSCXCN	0xB1	External Oscillator Control	page 137
P0	0x80	Port 0 Latch	page 173
P0MDOUT	0xA4	Port 0 Output Mode Configuration	page 173
P1	0x90	Port 1 Latch	page 174
P1MDIN	0xBD	Port 1 Input Mode	page 174
P1MDOUT	0xA5	Port 1 Output Mode Configuration	page 175
P2	0xA0	Port 2 Latch	page 175
P2MDOUT	0xA6	Port 2 Output Mode Configuration	page 175
P3	0xB0	Port 3 Latch	page 176
P3IF	0xAD	Port 3 Interrupt Flags	page 177
P3MDOUT	0xA7	Port 3 Output Mode Configuration	page 176
†P4	0x84	Port 4 Latch	page 180†
†P5	0x85	Port 5 Latch	page 180†

Figure 12.10. IP: Interrupt Priority

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	PT2	PS0	PT1	PX1	PT0	PX0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xB8

Bits7-6: UNUSED. Read = 11b, Write = don't care.
Bit5: PT2: Timer 2 Interrupt Priority Control.
This bit sets the priority of the Timer 2 interrupt.
0: Timer 2 interrupt priority determined by default priority order.
1: Timer 2 interrupts set to high priority level.
Bit4: PS0: UART0 Interrupt Priority Control.
This bit sets the priority of the UART0 interrupt.
0: UART0 interrupt priority determined by default priority order.
1: UART0 interrupts set to high priority level.
Bit3: PT1: Timer 1 Interrupt Priority Control.
This bit sets the priority of the Timer 1 interrupt.
0: Timer 1 interrupt priority determined by default priority order.
1: Timer 1 interrupts set to high priority level.
Bit2: PX1: External Interrupt 1 Priority Control.
This bit sets the priority of the External Interrupt 1 interrupt.
0: External Interrupt 1 priority determined by default priority order.
1: External Interrupt 1 set to high priority level.
Bit1: PT0: Timer 0 Interrupt Priority Control.
This bit sets the priority of the Timer 0 interrupt.
0: Timer 0 interrupt priority determined by default priority order.
1: Timer 0 interrupt set to high priority level.
Bit0: PX0: External Interrupt 0 Priority Control.
This bit sets the priority of the External Interrupt 0 interrupt.
0: External Interrupt 0 priority determined by default priority order.
1: External Interrupt 0 set to high priority level.

Figure 12.12. EIE2: Extended Interrupt Enable 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
EXVLD	ES1	EX7	EX6	EADC1	ET4	EADC0	ET3	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE7

Bit7: EXVLD: Enable External Clock Source Valid (XTLVLD) Interrupt.
This bit sets the masking of the XTLVLD interrupt.
0: Disable XTLVLD interrupt.
1: Enable interrupt requests generated by the XTLVLD flag (OSCXCN.7)

Bit6: ES1: Enable UART1 Interrupt.
This bit sets the masking of the UART1 interrupt.
0: Disable UART1 interrupt.
1: Enable UART1 interrupt.

Bit5: EX7: Enable External Interrupt 7.
This bit sets the masking of External Interrupt 7.
0: Disable External Interrupt 7.
1: Enable interrupt requests generated by the External Interrupt 7 input pin.

Bit4: EX6: Enable External Interrupt 6.
This bit sets the masking of External Interrupt 6.
0: Disable External Interrupt 6.
1: Enable interrupt requests generated by the External Interrupt 6 input pin.

Bit3: EADC1: Enable ADC1 End Of Conversion Interrupt.
This bit sets the masking of the ADC1 End of Conversion interrupt.
0: Disable ADC1 End of Conversion interrupt.
1: Enable interrupt requests generated by the ADC1 End of Conversion Interrupt.

Bit2: ET4: Enable Timer 4 Interrupt
This bit sets the masking of the Timer 4 interrupt.
0: Disable Timer 4 interrupt.
1: Enable interrupt requests generated by the TF4 flag (T4CON.7).

Bit1: EADC0: Enable ADC0 End of Conversion Interrupt.
This bit sets the masking of the ADC0 End of Conversion Interrupt.
0: Disable ADC0 Conversion Interrupt.
1: Enable interrupt requests generated by the ADC0 Conversion Interrupt.

Bit0: ET3: Enable Timer 3 Interrupt.
This bit sets the masking of the Timer 3 interrupt.
0: Disable all Timer 3 interrupts.
1: Enable interrupt requests generated by the TF3 flag (TMR3CN.7).

Figure 12.15. PCON: Power Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
SMOD0	SSTAT0	Reserved	SMOD1	SSTAT1	Reserved	STOP	IDLE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x87

Bit7: SMOD0: UART0 Baud Rate Doubler Enable.
This bit enables/disables the divide-by-two function of the UART0 baud rate logic for configurations described in the UART0 section.
0: UART0 baud rate divide-by-two enabled.
1: UART0 baud rate divide-by-two disabled.

Bit6: SSTAT0: UART0 Enhanced Status Mode Select.
This bit controls the access mode of the SM20-SM00 bits in register SCON0.
0: Reads/writes of SM20-SM00 access the SM20-SM00 UART0 mode setting.
1: Reads/writes of SM20-SM00 access the Framing Error (FE0), RX Overrun (RXOV0), and TX Collision (TXCOL0) status bits.

Bit5: Reserved. Read is undefined. Must write 0.

Bit4: SMOD1: UART1 Baud Rate Doubler Enable.
This bit enables/disables the divide-by-two function of the UART1 baud rate logic for configurations described in the UART1 section.
0: UART1 baud rate divide-by-two enabled.
1: UART1 baud rate divide-by-two disabled.

Bit3: SSTAT1: UART1 Enhanced Status Mode Select.
This bit controls the access mode of the SM21-SM01 bits in SCON1.
0: Reads/writes of SM21-SM01 access the SM21-SM01 UART1 mode setting.
1: Reads/writes of SM21-SM01 access the Framing Error (FE1), RX Overrun (RXOV1), and TX Collision (TXCOL1) status bits.

Bit2: Reserved. Read is undefined. Must write 0.

Bit1: STOP: STOP Mode Select.
Writing a ‘1’ to this bit will place the CIP-51 into STOP mode. This bit will always read ‘0’.
1: CIP-51 forced into power-down mode. (Turns off internal oscillator).

Bit0: IDLE: IDLE Mode Select.
Writing a ‘1’ to this bit will place the CIP-51 into IDLE mode. This bit will always read ‘0’.
1: CIP-51 forced into idle mode. (Shuts off clock to CPU, but clock to Timers, Interrupts, and all peripherals remain active.)

16.5. Memory Mode Selection

The external data memory space can be configured in one of four modes, shown in Figure 16.5, based on the EMIF Mode bits in the EMI0CF register (Figure 16.2). These modes are summarized below. More information about the different modes can be found in **Section " ." on page 152**.

16.5.1. Internal XRAM Only

When 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 4k boundaries. As an example, the addresses 0x1000 and 0x2000 both evaluate to address 0x0000 in on-chip XRAM space.

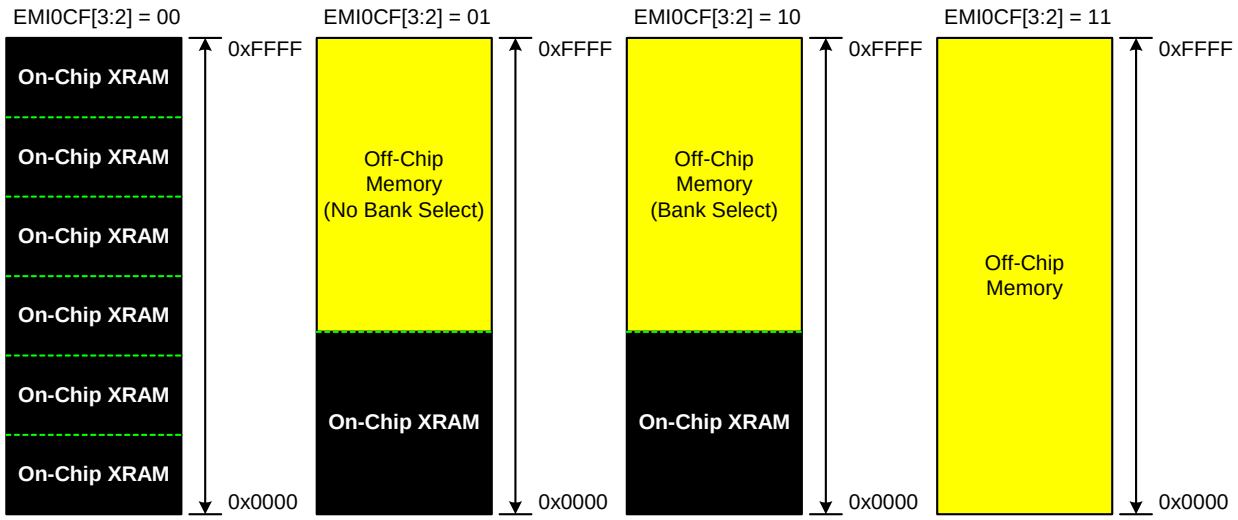
- 8-bit MOVX operations use the contents of EMI0CN to determine the high-byte of the effective address and R0 or R1 to determine the low-byte of the effective address.
- 16-bit MOVX operations use the contents of the 16-bit DPTR to determine the effective address.

16.5.2. Split Mode without Bank Select

When 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 4k boundary will access on-chip XRAM space.
- Effective addresses beyond the 4k 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. The lower 8-bits of the Address Bus A[7:0] are driven as defined by R0 or R1. 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. This behavior is in contrast with "Split Mode with Bank Select" described below.
- 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.

Figure 16.5. EMIF Operating Modes



.

Figure 16.6. EMI0TC: External Memory Timing Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
EAS1	EAS0	EWR3	EWR2	EWR1	EWR0	EAH1	EAH0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xA1

Bits7-6: EAS1-0: EMIF Address Setup Time Bits.
00: Address setup time = 0 SYSCLK cycles.
01: Address setup time = 1 SYSCLK cycle.
10: Address setup time = 2 SYSCLK cycles.
11: Address setup time = 3 SYSCLK cycles.

Bits5-2: EWR3-0: EMIF /WR and /RD Pulse-Width Control Bits.
0000: /WR and /RD pulse width = 1 SYSCLK cycle.
0001: /WR and /RD pulse width = 2 SYSCLK cycles.
0010: /WR and /RD pulse width = 3 SYSCLK cycles.
0011: /WR and /RD pulse width = 4 SYSCLK cycles.
0100: /WR and /RD pulse width = 5 SYSCLK cycles.
0101: /WR and /RD pulse width = 6 SYSCLK cycles.
0110: /WR and /RD pulse width = 7 SYSCLK cycles.
0111: /WR and /RD pulse width = 8 SYSCLK cycles.
1000: /WR and /RD pulse width = 9 SYSCLK cycles.
1001: /WR and /RD pulse width = 10 SYSCLK cycles.
1010: /WR and /RD pulse width = 11 SYSCLK cycles.
1011: /WR and /RD pulse width = 12 SYSCLK cycles.
1100: /WR and /RD pulse width = 13 SYSCLK cycles.
1101: /WR and /RD pulse width = 14 SYSCLK cycles.
1110: /WR and /RD pulse width = 15 SYSCLK cycles.
1111: /WR and /RD pulse width = 16 SYSCLK cycles.

Bits1-0: EAH1-0: EMIF Address Hold Time Bits.
00: Address hold time = 0 SYSCLK cycles.
01: Address hold time = 1 SYSCLK cycle.
10: Address hold time = 2 SYSCLK cycles.
11: Address hold time = 3 SYSCLK cycles.

to a Port pin without assigning RX0 as well. Each combination of enabled peripherals results in a unique device pinout.

All Port pins on Ports 0 through 3 that are not allocated by the Crossbar can be accessed as General-Purpose I/O (GPIO) pins by reading and writing the associated Port Data registers (See Figure 17.10, Figure 17.12, Figure 17.15, and Figure 17.17), a set of SFRs which are both byte- and bit-addressable. The output states of Port pins that are allocated by the Crossbar are controlled by the digital peripheral that is mapped to those pins. Writes to the Port Data registers (or associated Port bits) will have no effect on the states of these pins.

A Read of a Port Data register (or Port bit) will always return the logic state present at the pin itself, regardless of whether the Crossbar has allocated the pin for peripheral use or not. An exception to this occurs during the execution of a *read-modify-write* instruction (ANL, ORL, XRL, CPL, INC, DEC, DJNZ, JBC, CLR, SET, and the bitwise MOV operation). During the *read* cycle of the *read-modify-write* instruction, it is the contents of the Port Data register, not the state of the Port pins themselves, which is read.

Because the Crossbar registers affect the pinout of the peripherals of the device, they are typically configured in the initialization code of the system before the peripherals themselves are configured. Once configured, the Crossbar registers are typically left alone.

Once the Crossbar registers have been properly configured, the Crossbar is enabled by setting XBARE (XBR2.6) to a logic 1. **Until XBARE is set to a logic 1, the output drivers on Ports 0 through 3 are explicitly disabled in order to prevent possible contention on the Port pins while the Crossbar registers and other registers which can affect the device pinout are being written**.

The output drivers on Crossbar-assigned input signals (like RX0, for example) are explicitly disabled; thus the values of the Port Data registers and the PnMDOUT registers have no effect on the states of these pins.

17.1.2. Configuring the Output Modes of the Port Pins

The output drivers on Ports 0 through 3 remain disabled until the Crossbar is enabled by setting XBARE (XBR2.6) to a logic 1.

The output mode of each port pin can be configured as either Open-Drain or Push-Pull; the default state is Open-Drain. In the Push-Pull configuration, writing a logic 0 to the associated bit in the Port Data register will cause the Port pin to be driven to GND, and writing a logic 1 will cause the Port pin to be driven to VDD. In the Open-Drain configuration, writing a logic 0 to the associated bit in the Port Data register will cause the Port pin to be driven to GND, and a logic 1 will cause the Port pin to assume a high-impedance state. The Open-Drain configuration is useful to prevent contention between devices in systems where the Port pin participates in a shared interconnection in which multiple outputs are connected to the same physical wire (like the SDA signal on an SMBus connection).

The output modes of the Port pins on Ports 0 through 3 are determined by the bits in the associated PnMDOUT registers (See Figure 17.11, Figure 17.14, Figure 17.16, and Figure 17.18). For example, a logic 1 in P3MDOUT.7 will configure the output mode of P3.7 to Push-Pull; a logic 0 in P3MDOUT.7 will configure the output mode of P3.7 to Open-Drain. All Port pins default to Open-Drain output.

The PnMDOUT registers control the output modes of the port pins regardless of whether the Crossbar has allocated the Port pin for a digital peripheral or not. The exceptions to this rule are: the Port pins connected to SDA, SCL, RX0 (if UART0 is in Mode 0), and RX1 (if UART1 is in Mode 0) are always configured as Open-Drain outputs, regardless of the settings of the associated bits in the PnMDOUT registers.

Port Data register will cause the Port pin to be driven to GND, and a logic 1 will cause the Port pin to assume a high-impedance state. The Open-Drain configuration is useful to prevent contention between devices in systems where the Port pin participates in a shared interconnection in which multiple outputs are connected to the same physical wire.

The output modes of the Port pins on Ports 4 through 7 are determined by the bits in the P74OUT register (see Figure 17.20). Each bit in P74OUT controls the output mode of a 4-bit bank of Port pins on Ports 4, 5, 6, and 7. A logic 1 in P74OUT.7 will configure the output modes of 4 most-significant bits of Port 7, P7.[7:4], to Push-Pull; a logic 0 in P74OUT.7 will configure the output modes of P7.[7:4] to Open-Drain.

17.2.3. Configuring Port Pins as Digital Inputs

A Port pin is configured as a digital input by setting its output mode to “Open-Drain” and writing a logic 1 to the associated bit in the Port Data register. For example, P7.7 is configured as a digital input by setting P74OUT.7 to a logic 0 and P7.7 to a logic 1.

17.2.4. Weak Pull-ups

By default, each Port pin has an internal weak pull-up device enabled which provides a resistive connection (about 100 kΩ) between the pin and VDD. The weak pull-up devices can be globally disabled by writing a logic 1 to the Weak Pull-up Disable bit, (WEAKPUD, XBR2.7). The weak pull-up is automatically deactivated on any pin that is driving a logic 0; that is, an output pin will not contend with its own pull-up device.

17.2.5. External Memory Interface

If the External Memory Interface (EMIF) is enabled on the High ports (Ports 4 through 7), EMIFLE (XBR2.1) should be set to a logic 0.

If the External Memory Interface is enabled on the High ports and an off-chip MOVX operation occurs, the External Memory Interface will control the output states of the affected Port pins during the execution phase of the MOVX instruction, regardless of the settings of the Port Data registers. The output configuration of the Port pins is not affected by the EMIF operation, except that Read operations will explicitly disable the output drivers on the Data Bus during the MOVX execution. See **Section “16. EXTERNAL DATA MEMORY INTERFACE AND ON-CHIP XRAM” on page 145** for more information about the External Memory Interface.

Figure 17.20. P74OUT: Ports 7 - 4 Output Mode Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P7H	P7L	P6H	P6L	P5H	P5L	P4H	P4L	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xB5

Bit7: P7H: Port7 Output Mode High Nibble Bit.
0: P7.[7:4] configured as Open-Drain.
1: P7.[7:4] configured as Push-Pull.

Bit6: P7L: Port7 Output Mode Low Nibble Bit.
0: P7.[3:0] configured as Open-Drain.
1: P7.[3:0] configured as Push-Pull.

Bit5: P6H: Port6 Output Mode High Nibble Bit.
0: P6.[7:4] configured as Open-Drain.
1: P6.[7:4] configured as Push-Pull.

Bit4: P6L: Port6 Output Mode Low Nibble Bit.
0: P6.[3:0] configured as Open-Drain.
1: P6.[3:0] configured as Push-Pull.

Bit3: P5H: Port5 Output Mode High Nibble Bit.
0: P5.[7:4] configured as Open-Drain.
1: P5.[7:4] configured as Push-Pull.

Bit2: P5L: Port5 Output Mode Low Nibble Bit.
0: P5.[3:0] configured as Open-Drain.
1: P5.[3:0] configured as Push-Pull.

Bit1: P4H: Port4 Output Mode High Nibble Bit.
0: P4.[7:4] configured as Open-Drain.
1: P4.[7:4] configured as Push-Pull.

Bit0: P4L: Port4 Output Mode Low Nibble Bit.
0: P4.[3:0] configured as Open-Drain.
1: P4.[3:0] configured as Push-Pull.

19.4. 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 section.

Figure 19.5. SPI0CFG: SPI0 Configuration Register

R/W	R/W	R	R	R	R/W	R/W	R/W	Reset Value
CKPHA	CKPOL	BC2	BC1	BC0	SPIFRS2	SPIFRS1	SPIFRS0	00000111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9A

Bit7: CKPHA: SPI0 Clock Phase.
This bit controls the SPI0 clock phase.
0: Data sampled on first edge of SCK period.
1: Data sampled on second edge of SCK period.

Bit6: 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.

Bits5-3: BC2-BC0: SPI0 Bit Count.
Indicates which of the up to 8 bits of the SPI0 word have been transmitted.

BC2-BC0			BIT Transmitted
0	0	0	Bit 0 (LSB)
0	0	1	Bit 1
0	1	0	Bit 2
0	1	1	Bit 3
1	0	0	Bit 4
1	0	1	Bit 5
1	1	0	Bit 6
1	1	1	Bit 7 (MSB)

Bits2-0: SPIFRS2-SPIFRS0: SPI0 Frame Size.
These three bits determine the number of bits to shift in/out of the SPI0 shift register during a data transfer in master mode. They are ignored in slave mode.

SPIFRS			Bits Shifted
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

21. UART1

UART1 is an enhanced serial port with frame error detection and address recognition hardware. UART1 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 UART1 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 previous received byte is read.

UART1 is accessed via its associated SFRs, Serial Control (SCON1) and Serial Data Buffer (SBUF1). The single SBUF1 location provides access to both transmit and receive registers. Reads access the Receive register and writes access the Transmit register automatically.

UART1 may be operated in polled or interrupt mode. UART1 has two sources of interrupts: a Transmit Interrupt flag, TI1 (SCON1.1) set when transmission of a data byte is complete, and a Receive Interrupt flag, RI1 (SCON1.0) set when reception of a data byte is complete. UART1 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 UART1 interrupt (transmit complete or receive complete).

Figure 21.1. UART1 Block Diagram

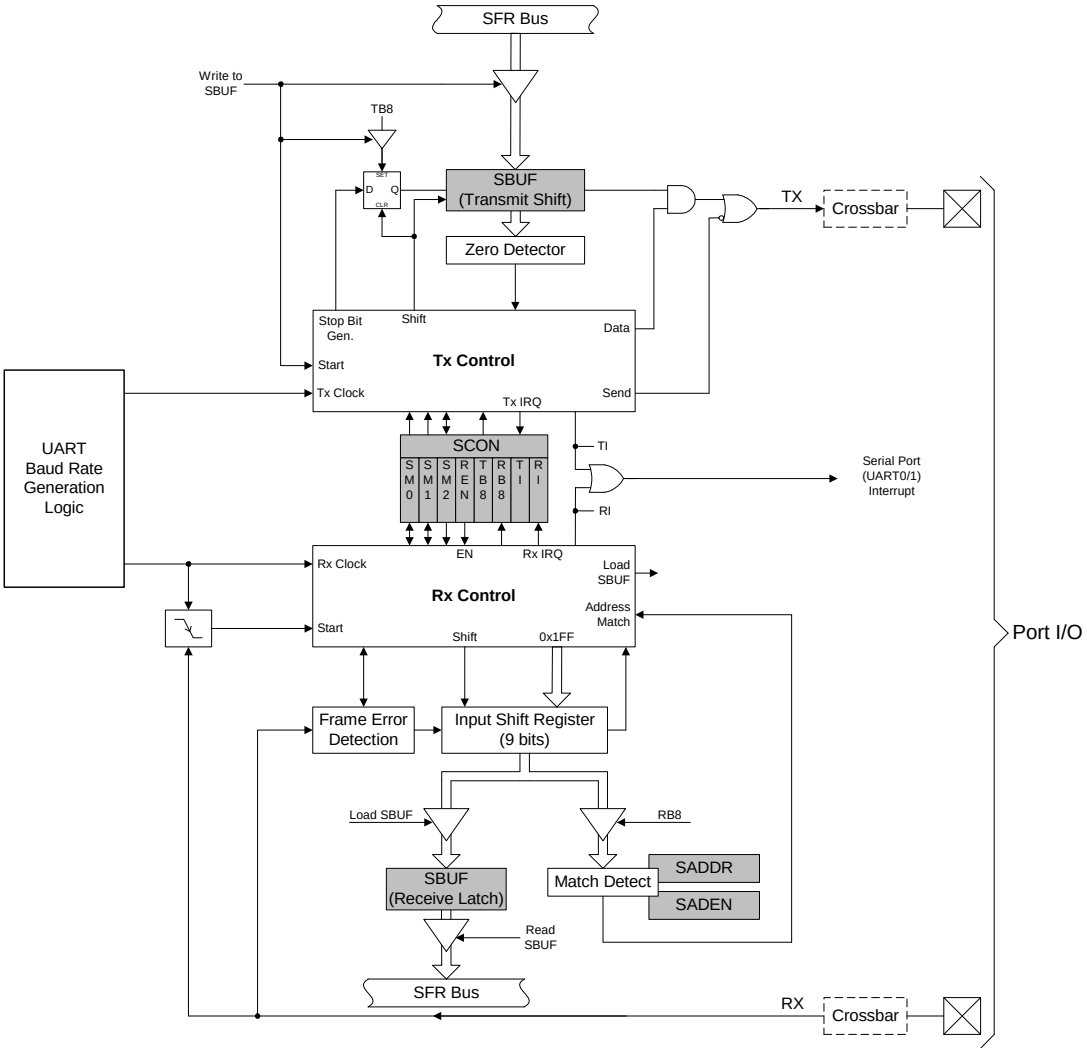


Table 21.2. Oscillator Frequencies for Standard Baud Rates

Oscillator frequency (MHz)	Divide Factor	Timer 1 Load Value*	Resulting Baud Rate (Hz)**
25.0	434	0xE5	57600 (57870)
25.0	868	0xCA	28800
24.576	320	0xEC	76800
24.576	848	0xCB	28800 (28921)
24.0	208	0XF3	115200 (115384)
24.0	833	0xCC	28800 (28846)
23.592	205	0xF3	115200 (113423)
23.592	819	0xCD	28800 (28911)
22.1184	192	0xF4	115200
22.1184	768	0xD0	28800
18.432	160	0xF6	115200
18.432	640	0xD8	28800
16.5888	144	0xF7	115200
16.5888	576	0xDC	28800
14.7456	128	0xF8	115200
14.7456	512	0xE0	28800
12.9024	112	0xF9	115200
12.9024	448	0xE4	28800
11.0592	96	0xFA	115200
11.0592	348	0xE8	28800
9.216	80	0xFB	115200
9.216	320	0xEC	28800
7.3728	64	0xFC	115200
7.3728	256	0xF0	28800
5.5296	48	0xFD	115200
5.5296	192	0xF4	28800
3.6864	32	0xFE	115200
3.6864	128	0xF8	28800
1.8432	16	0xFF	115200
1.8432	64	0xFC	28800

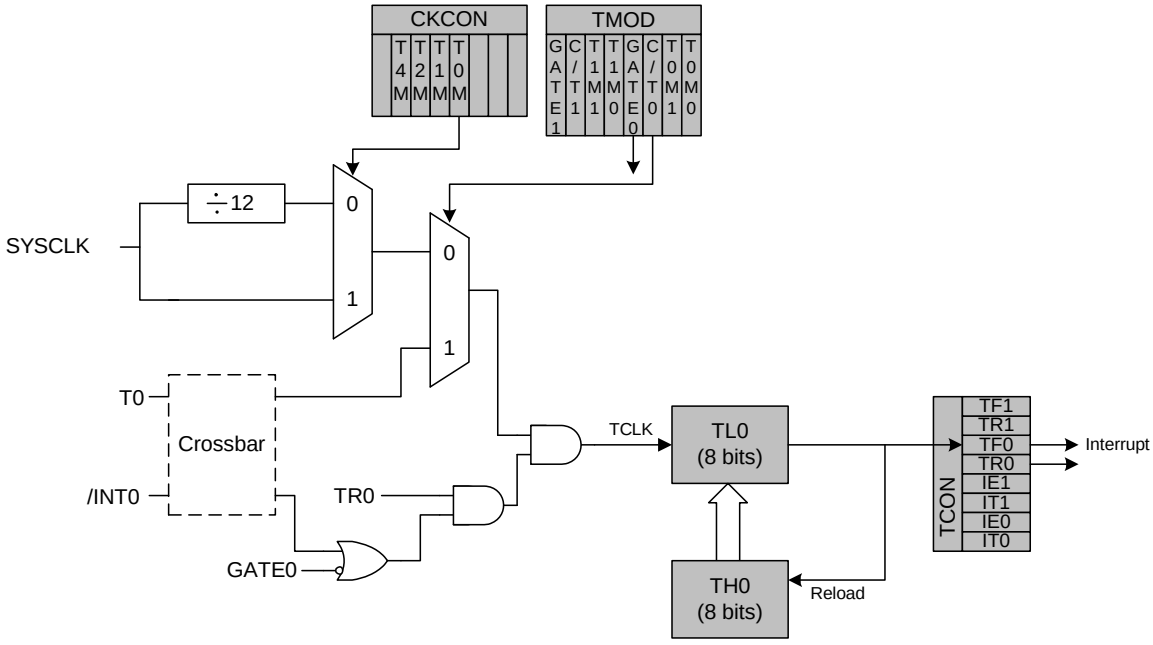
* Assumes SMOD1=1 and T1M=1.

** Numbers in parenthesis show the actual baud rate.

22.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload

Mode 2 configures Timer 0 and Timer 1 to operate as 8-bit counter/timers with automatic reload of the start value. The TL0 holds the count and TH0 holds the reload value. When the counter in TL0 overflows from all ones to 0x00, the timer overflow flag TF0 (TCON.5) is set and the counter value in TL0 is reloaded from TH0. If enabled, an interrupt will occur when the TF0 flag is set. The reload value in TH0 is not changed. TL0 must be initialized to the desired value before enabling the timer for the first count to be correct. When in Mode 2, Timer 1 operates identically to Timer 0. Both counter/timers are enabled and configured in Mode 2 in the same manner as Mode 0.

Figure 22.3. T0 Mode 2 (8-bit Auto-Reload) Block Diagram

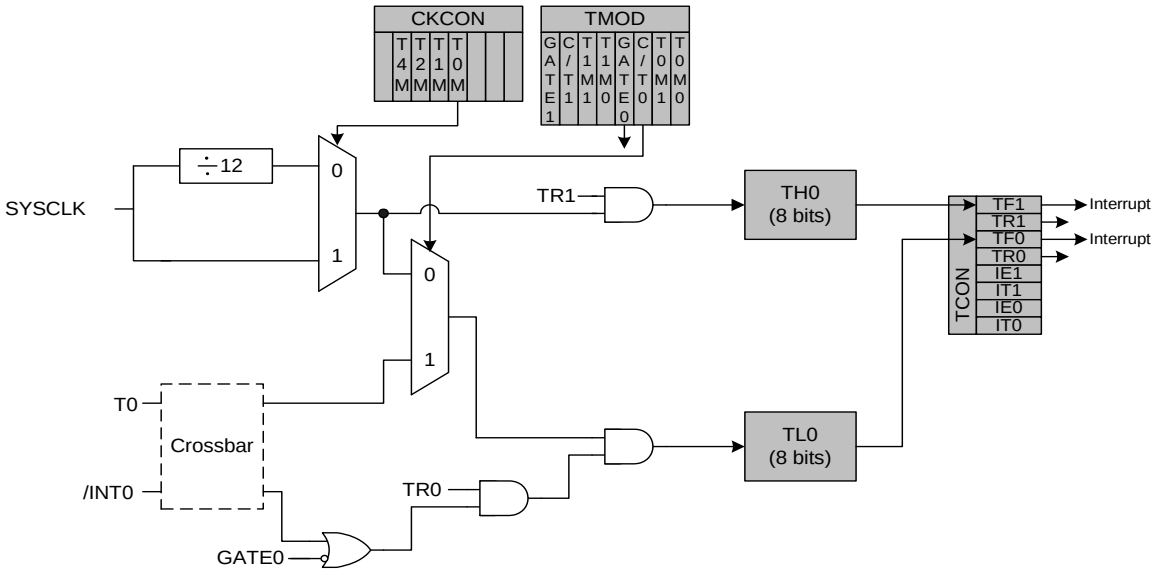


22.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only)

Timer 0 and Timer 1 behave differently in Mode 3. Timer 0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. The counter/timer in TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0, C/T0, GATE0 and TF0. It can use either the system clock or an external input signal as its timebase. The timer in the TH0 register is restricted to a timer function sourced by the system clock. TH0 is enabled using the Timer 1 run control bit TR1. TH0 sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

Timer 1 is inactive in Mode 3, so with Timer 0 in Mode 3, Timer 1 can be turned off and on by switching it into and out of its Mode 3. When Timer 0 is in Mode 3, Timer 1 can be operated in Modes 0, 1 or 2, but cannot be clocked by external signals nor set the TF1 flag and generate an interrupt. However, the Timer 1 overflow can be used to generate the baud clock for UART0 and/or UART1. Refer to **Section "20. UART0" on page 205** and **Section "21. UART1" on page 215** for information on configuring Timer 1 for baud rate generation.

Figure 22.4. T0 Mode 3 (Two 8-bit Timers) Block Diagram



22.1.3. Mode 2: Baud Rate Generator

Timer 2 can be used as a baud rate generator for UART0 when UART0 is operated in modes 1 or 3 (refer to Section "20.1. UART0 Operational Modes" on page 206 for more information on the UART0 operational modes). In Baud Rate Generator mode, Timer 2 works similarly to the auto-reload mode. On overflow, the 16-bit value held in the two capture registers (RCAP2H, RCAP2L) is automatically loaded into the counter/timer register. However, the TF2 overflow flag is not set and no interrupt is generated. Instead, the overflow event is used as the input to the UART's shift clock. Timer 2 overflows can be selected to generate baud rates for transmit and/or receive independently.

The Baud Rate Generator mode is selected by setting RCLK0 (T2CON.5) and/or TCLK0 (T2CON.2) to '1'. When RCLK0 or TCLK0 is set to logic 1, Timer 2 operates in the auto-reload mode regardless of the state of the CP/RL2 bit. Note that in Baud Rate Generator mode, the Timer 2 timebase is the system clock divided by two. When selected as the UART0 baud clock source, Timer 2 defines the UART0 baud rate as follows:

*Baud Rate = SYSCLK / ((65536 - [RCAP2H, RCAP2L]) * 32)*

If a different time base is required, setting the C/T2 bit to logic 1 will allow the timebase to be derived from the external input pin T2. In this case, the baud rate for the UART is calculated as:

*Baud Rate = F_{CLK} / ((65536 - [RCAP2H, RCAP2L]) * 16)*

Where F_{CLK} is the frequency of the signal (TCLK) supplied to Timer 2 and [RCAP2H, RCAP2L] is the 16-bit value held in the capture registers.

As explained above, in Baud Rate Generator mode, Timer 2 does not set the TF2 overflow flag and therefore cannot generate an interrupt. However, if EXEN2 is set to logic 1, a high-to-low transition on the T2EX input pin will set the EXF2 flag and a Timer 2 interrupt will occur if enabled. Therefore, the T2EX input may be used as an additional external interrupt source.

Figure 22.13. T2 Mode 2 Block Diagram

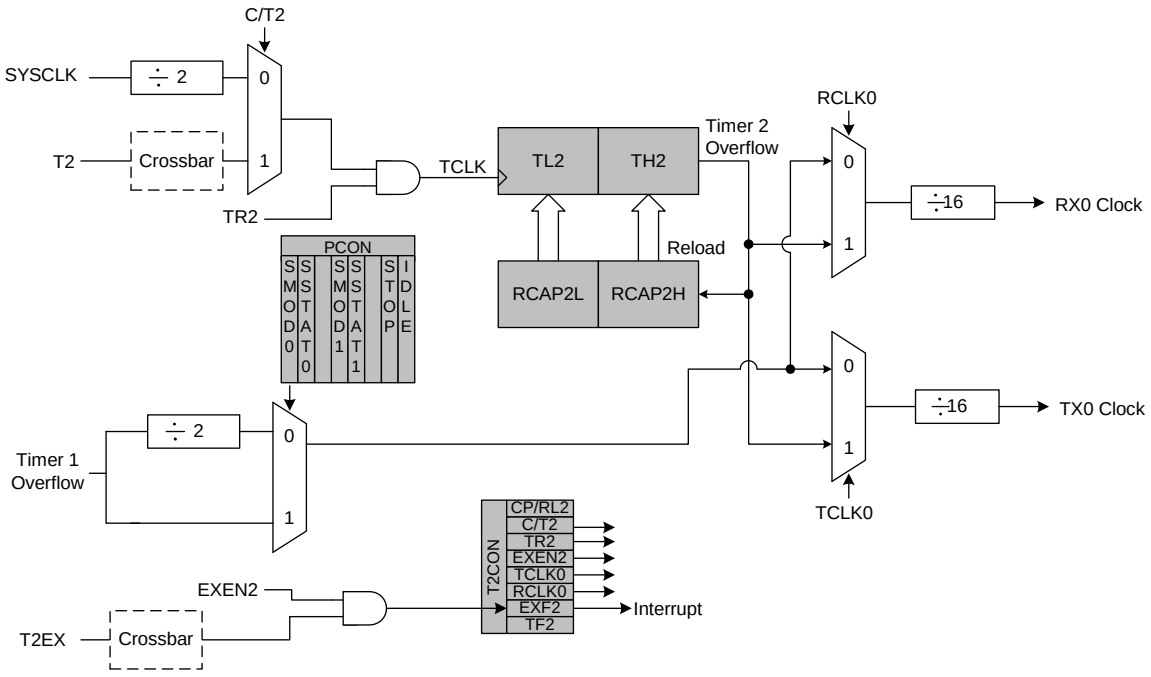
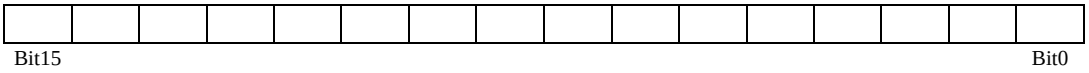


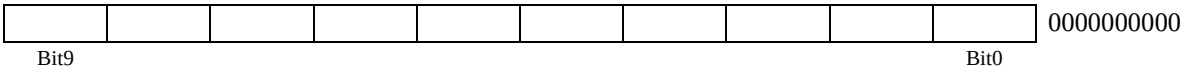
Figure 24.4. FLASHADR: JTAG Flash Address Register



This register holds the address for all JTAG Flash read, write, and erase operations. This register autoincrements after each read or write, regardless of whether the operation succeeded or failed.

Bits15-0: Flash Operation 16-bit Address.

Figure 24.5. FLASHDAT: JTAG Flash Data Register



This register is used to read or write data to the Flash memory across the JTAG interface.

Bits9-2: DATA7-0: Flash Data Byte.
Bit1: FAIL: Flash Fail Bit.
0: Previous Flash memory operation was successful.
1: Previous Flash memory operation failed. Usually indicates the associated memory location was locked.
Bit0: BUSY: Flash Busy Bit.
0: Flash interface logic is not busy.
1: Flash interface logic is processing a request. Reads or writes while BUSY = 1 will not initiate another operation