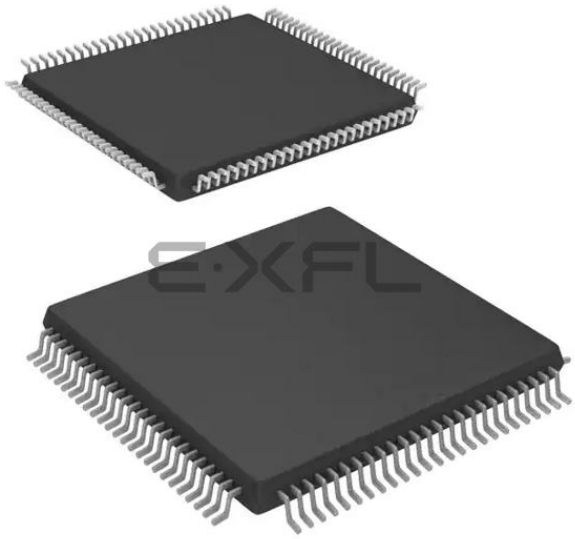


Welcome to E-XFL.COM

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	CANbus, 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	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, 13x10b; D/A 2x10b, 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/c8051f042r

Table 4.1. Pin Definitions (Continued)

Name	Pin Numbers		Type	Description
	F040/2/4/6	F041/3/5/7		
P1.0/AIN2.0/A8	36	29	A In D I/O	ADC1 Input Channel 0 (See ADC1 Specification for complete description). Bit 8 External Memory Address bus (Non-multiplexed mode) Port 1.0 See Port Input/Output section for complete description.
P1.1/AIN2.1/A9	35	28	A In D I/O	Port 1.1. See Port Input/Output section for complete description.
P1.2/AIN2.2/A10	34	27	A In D I/O	Port 1.2. See Port Input/Output section for complete description.
P1.3/AIN2.3/A11	33	26	A In D I/O	Port 1.3. See Port Input/Output section for complete description.
P1.4/AIN2.4/A12	32	23	A In D I/O	Port 1.4. See Port Input/Output section for complete description.
P1.5/AIN2.5/A13	31	22	A In D I/O	Port 1.5. See Port Input/Output section for complete description.
P1.6/AIN2.6/A14	30	21	A In D I/O	Port 1.6. See Port Input/Output section for complete description.
P1.7/AIN2.7/A15	29	20	A In D I/O	Port 1.7. See Port Input/Output section for complete description.
P2.0/A8m/A0	46	37	D I/O	Bit 8 External Memory Address bus (Multiplexed mode) Bit 0 External Memory Address bus (Non-multiplexed mode) Port 2.0 See Port Input/Output section for complete description.
P2.1/A9m/A1	45	36	D I/O	Port 2.1. See Port Input/Output section for complete description.
P2.2/A10m/A2	44	35	D I/O	Port 2.2. See Port Input/Output section for complete description.
P2.3/A11m/A3	43	34	D I/O	Port 2.3. See Port Input/Output section for complete description.
P2.4/A12m/A4	42	33	D I/O	Port 2.4. See Port Input/Output section for complete description.
P2.5/A13m/A5	41	32	D I/O	Port 2.5. See Port Input/Output section for complete description.
P2.6/A14m/A6	40	31	D I/O	Port 2.6. See Port Input/Output section for complete description.
P2.7/A15m/A7	39	30	D I/O	Port 2.7. See Port Input/Output section for complete description.

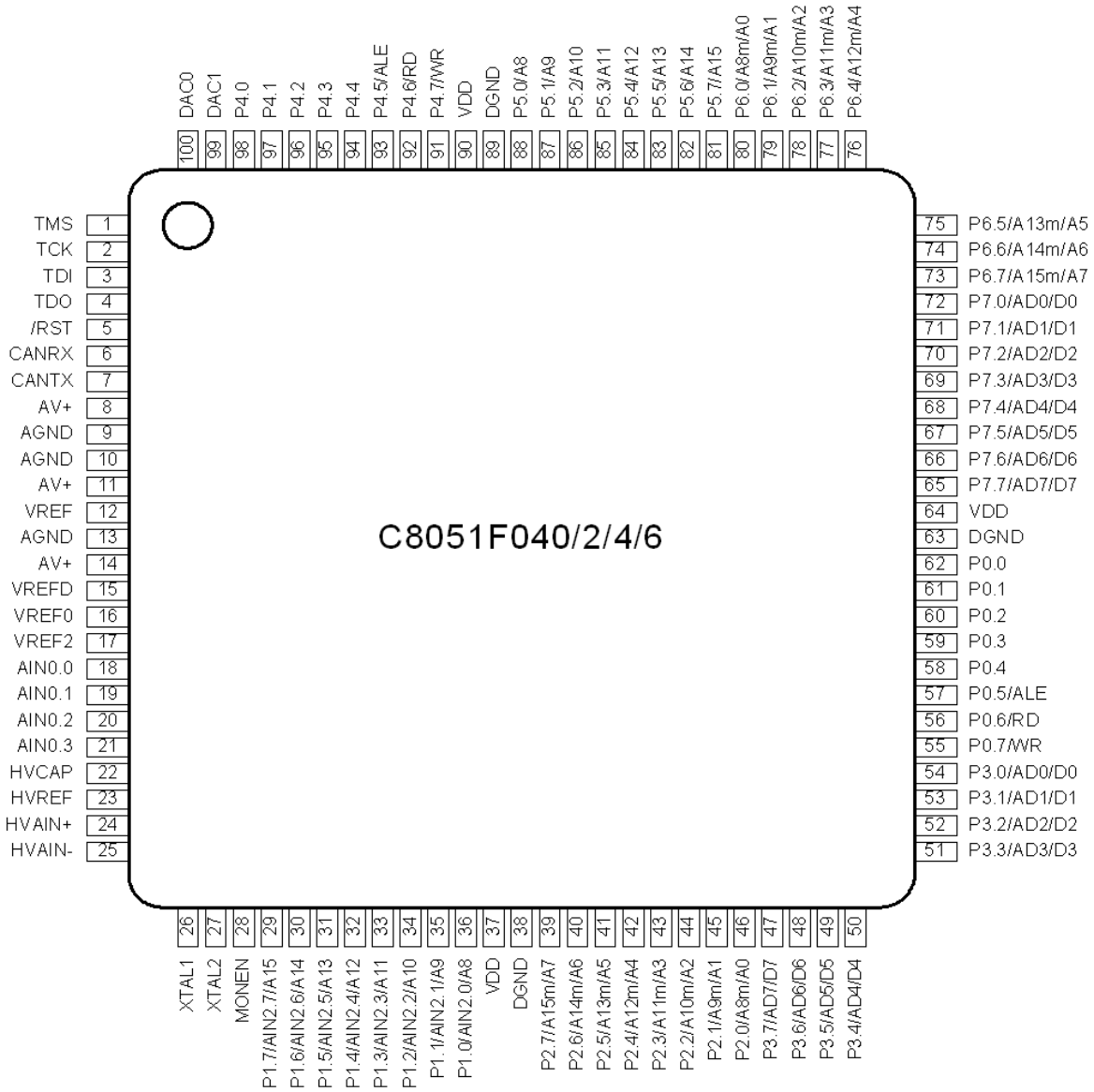


Figure 4.1. TQFP-100 Pinout Diagram

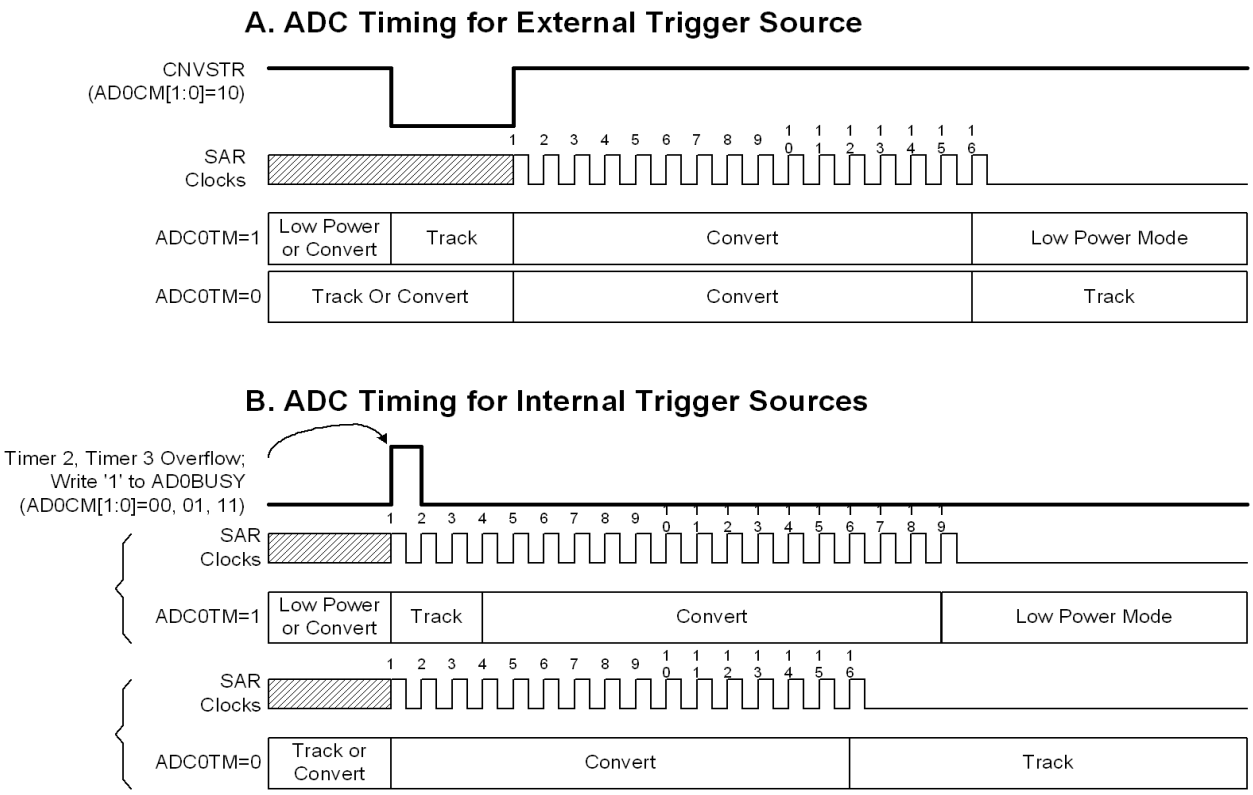
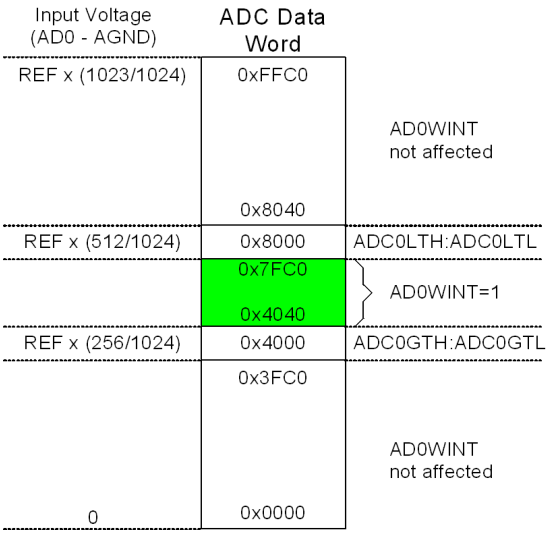
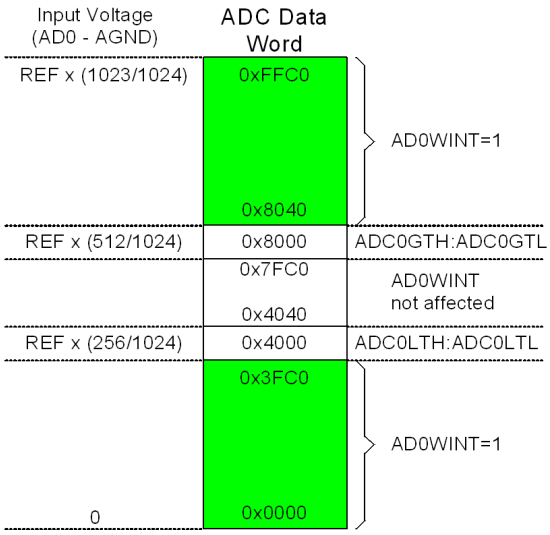


Figure 5.4. 12-Bit ADC Track and Conversion Example Timing



Given:
AMX0SL = 0x00, AMX0CF = 0x00, ADLJST = 1,
ADC0LTH:ADC0LTL = 0x8000,
ADC0GTH:ADC0GTL = 0x4000.
An ADC End of Conversion will cause an ADC Window Compare Interrupt (ADWINT=1) if the resulting ADC Data Word is < 0x8000 and > 0x4000.
Given:



AMX0SL = 0x00, AMX0CF = 0x00, ADLJST = 1,
ADC0LTH:ADC0LTL = 0x4000,
ADC0GTH:ADC0GTL = 0x8000.
An ADC End of Conversion will cause an ADC Window Compare Interrupt (ADWINT=1) if the resulting ADC Data Word is < 0x4000 or > 0x8000.

Figure 6.10. 10-Bit ADC0 Window Interrupt Example: Left Justified Single-Ended Data

Table 8.1. DAC Electrical Characteristics

V_{DD} = 3.0 V, AV+ = 3.0 V, V_{REF} = 2.40 V (REFBE = 0), No Output Load unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Static Performance					
Resolution			12		bits
Integral Nonlinearity		—	±2	—	LSB
Differential Nonlinearity		—		±1	LSB
Output Noise	No Output Filter 100 kHz Output Filter 10 kHz Output Filter	— — —	250 128 41	— — —	µVrms
Offset Error	Data Word = 0x014	—	±3	±30	mV
Offset Tempco		—	6	—	ppm/°C
Full-Scale Error		—	±20	±60	mV
Full-Scale Error Tempco		—	10	—	ppm/°C
V_{DD} Power Supply Rejection Ratio		—	–60	—	dB
Output Impedance in Shutdown Mode	DACnEN = 0	—	100	—	kΩ
Output Sink Current		—	300	—	µA
Output Short-Circuit Current	Data Word = 0xFFF	—	15	—	mA
Dynamic Performance					
Voltage Output Slew Rate	Load = 40 pF	—	0.44	—	V/µs
Output Settling Time to 1/2 LSB	Load = 40 pF, Output swing from code 0xFFF to 0x014	—	10	—	µs
Output Voltage Swing		0	—	VREF – LSB	V
Startup Time		—	10	—	µs
Analog Outputs					
Load Regulation	I_L = 0.01 mA to 0.3 mA at code 0xFFF	—	60	—	ppm
Power Consumption (each DAC)					
Power Supply Current (AV+ supplied to DAC)	Data Word = 0x7FF	—	110	400	µA

SFR Definition 9.1. REF0CN: Reference Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	AD0VRS	AD2VRS	TEMPE	BIASE	REFBE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xD1
SFR Page: 0

Bits7-5: UNUSED. Read = 000b; Write = don't care.
Bit4: AD0VRS: ADC0 Voltage Reference Select
0: ADC0 voltage reference from VREF0 pin.
1: ADC0 voltage reference from DAC0 output (C8051F040/2 only).
Bit3: AD2VRS: ADC2 Voltage Reference Select (C8051F040/2 only).
0: ADC2 voltage reference from VREF2 pin.
1: ADC2 voltage reference from AV+.
Bit2: TEMPE: Temperature Sensor Enable Bit.
0: Internal Temperature Sensor Off.
1: Internal Temperature Sensor On.
Bit1: BIASE: ADC/DAC Bias Generator Enable Bit. (Must be '1' if using ADC or DAC).
0: Internal Bias Generator Off.
1: Internal Bias Generator On.
Bit0: REFBE: Internal Reference Buffer Enable Bit.
0: Internal Reference Buffer Off.
1: Internal Reference Buffer On. Internal voltage reference is driven on the VREF pin.

Table 12.3. Special Function Registers (Continued)

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

Register	Address	SFR Page	Description	Page No.
SPI0CFG	0x9A	0	SPI Configuration	page 261
SPI0CKR	0x9D	0	SPI Clock Rate Control	page 263
SPI0CN	0xF8	0	SPI Control	page 262
SPI0DAT	0x9B	0	SPI Data	page 264
SSTA0	0x91	0	UART0 Status and Clock Selection	page 275
TCON	0x88	0	Timer/Counter Control	page 293
TH0	0x8C	0	Timer/Counter 0 High	page 296
TH1	0x8D	0	Timer/Counter 1 High	page 296
TL0	0x8A	0	Timer/Counter 0 Low	page 295
TL1	0x8B	0	Timer/Counter 1 Low	page 296
TMOD	0x89	0	Timer/Counter Mode	page 294
TMR2CF	0xC9	0	Timer/Counter 2 Configuration	page 302
TMR2CN	0xC8	0	Timer/Counter 2 Control	page 301
TMR2H	0xCD	0	Timer/Counter 2 High	page 304
TMR2L	0xCC	0	Timer/Counter 2 Low	page 303
TMR3CF	0xC9	1	Timer/Counter 3 Configuration	page 302
TMR3CN	0xC8	1	Timer 3 Control	page 301
TMR3H	0xCD	1	Timer/Counter 3 High	page 304
TMR3L	0xCC	1	Timer/Counter 3 Low	page 303
TMR4CF	0xC9	2	Timer/Counter 4 Configuration	page 302
TMR4CN	0xC8	2	Timer/Counter 4 Control	page 301
TMR4H	0xCD	2	Timer/Counter 4 High	page 304
TMR4L	0xCC	2	Timer/Counter 4 Low	page 303
WDTCN	0xFF	All Pages	Watchdog Timer Control	page 169
XBR0	0xE1	F	Port I/O Crossbar Control 0	page 212
XBR1	0xE2	F	Port I/O Crossbar Control 1	page 213
XBR2	0xE3	F	Port I/O Crossbar Control 2	page 214
XBR3	0xE4	F	Port I/O Crossbar Control 3	page 215
0x97, 0xA2, 0xB3, 0xB4, 0xCE, 0xDF			Reserved	

Notes:

1. Refers to a register in the C8051F040 only.
2. Refers to a register in the C8051F041 only.
3. Refers to a register in C8051F040/1/2/3 only.
4. Refers to a register in the C8051F040/2/4/6 only.
5. Refers to a register in the C8051F041/3/5/7 only.

lup and/or decoupling of the /RST pin to avoid erroneous noise-induced resets. The MCU will remain in reset until at least 12 clock cycles after the active-low /RST signal is removed. The PINRSF flag (RSTSRC.0) is set on exit from an external reset.

13.4. Missing Clock Detector Reset

The Missing Clock Detector is essentially a one-shot circuit that is triggered by the MCU system clock. If the system clock goes away for more than 100 µs, the one-shot will time out and generate a reset. After a Missing Clock Detector reset, the MCDRSF flag (RSTSRC.2) will be set, signifying the MCD as the reset source; otherwise, this bit reads '0'. The state of the /RST pin is unaffected by this reset. Setting the MCDRSF bit, RSTSRC.2 (see Section "**14. Oscillators**" on page **173**) enables the Missing Clock Detector.

13.5. Comparator0 Reset

Comparator0 can be configured as a reset input by writing a '1' to the C0RSEF flag (RSTSRC.5). Comparator0 should be enabled using CPT0CN.7 (see Section "**11. Comparators**" on page **121**) prior to writing to C0RSEF to prevent any turn-on chatter on the output from generating an unwanted reset. The Comparator0 reset is active-low: if the non-inverting input voltage (CP0+ pin) is less than the inverting input voltage (CP0- pin), the MCU is put into the reset state. After a Comparator0 Reset, the C0RSEF flag (RSTSRC.5) will read '1' signifying Comparator0 as the reset source; otherwise, this bit reads '0'. The state of the /RST pin is unaffected by this reset.

13.6. External CNVSTR0 Pin Reset

The external CNVSTR0 signal can be configured as a reset input by writing a '1' to the CNVRSEF flag (RSTSRC.6). The CNVSTR0 signal can appear on any of the P0, P1, P2 or P3 I/O pins as described in Section "**17.1. Ports 0 through 3 and the Priority Crossbar Decoder**" on page **204**. Note that the Crossbar must be configured for the CNVSTR0 signal to be routed to the appropriate Port I/O. The Crossbar should be configured and enabled before the CNVRSEF is set. When configured as a reset, CNVSTR0 is active-low and level sensitive. After a CNVSTR0 reset, the CNVRSEF flag (RSTSRC.6) will read '1' signifying CNVSTR0 as the reset source; otherwise, this bit reads '0'. The state of the /RST pin is unaffected by this reset.

13.7. Watchdog Timer Reset

The MCU includes a programmable Watchdog Timer (WDT) running off the system clock. A WDT overflow will force the MCU into the reset state. To prevent the reset, the WDT must be restarted by application software before overflow. If the system experiences a software or hardware malfunction preventing the software from restarting the WDT, the WDT will overflow and cause a reset. This should prevent the system from running out of control.

Following a reset the WDT is automatically enabled and running with the default maximum time interval. If desired the WDT can be disabled by system software or locked on to prevent accidental disabling. Once locked, the WDT cannot be disabled until the next system reset. The state of the /RST pin is unaffected by this reset.

The WDT consists of a 21-bit timer running from the programmed system clock. The timer measures the period between specific writes to its control register. If this period exceeds the programmed limit, a WDT reset is generated. The WDT can be enabled and disabled as needed in software, or can be permanently enabled if desired. Watchdog features are controlled via the Watchdog Timer Control Register (WDTCN) shown in SFR Definition 13.1.

16.4. Multiplexed and Non-multiplexed Selection

The External Memory Interface is capable of acting in a Multiplexed mode or a Non-multiplexed mode, depending on the state of the EMD2 (EMI0CF.4) bit.

16.4.1. Multiplexed Configuration

In Multiplexed mode, the Data Bus and the lower 8-bits of the Address Bus share the same Port pins: AD[7:0]. In this mode, an external latch (74HC373 or equivalent logic gate) is used to hold the lower 8-bits of the RAM address. The external latch is controlled by the ALE (Address Latch Enable) signal, which is driven by the External Memory Interface logic. An example of a Multiplexed Configuration is shown in Figure 16.1.

In Multiplexed mode, the external MOVX operation can be broken into two phases delineated by the state of the ALE signal. During the first phase, ALE is high and the lower 8-bits of the Address Bus are presented to AD[7:0]. During this phase, the address latch is configured such that the 'Q' outputs reflect the states of the 'D' inputs. When ALE falls, signaling the beginning of the second phase, the address latch outputs remain fixed and are no longer dependent on the latch inputs. Later in the second phase, the Data Bus controls the state of the AD[7:0] port at the time /RD or /WR is asserted.

See **Section "16.6.2. Multiplexed Mode" on page 199** for more information.

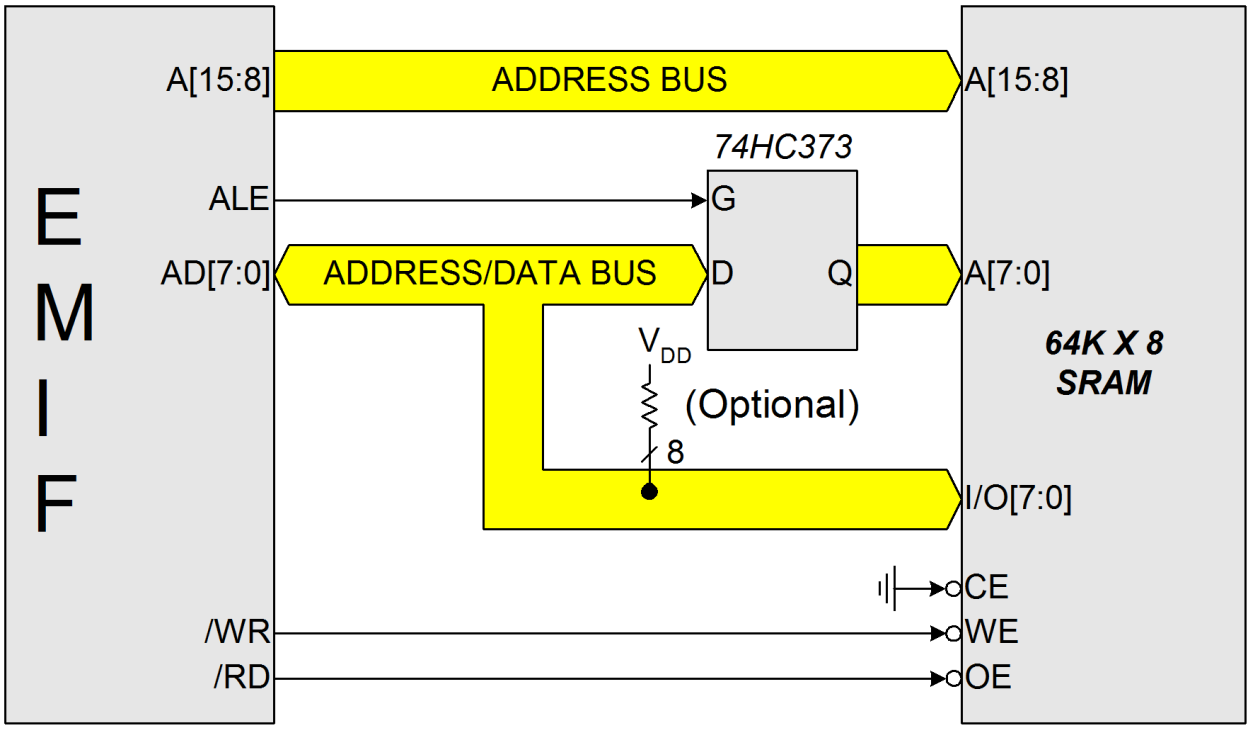


Figure 16.1. Multiplexed Configuration Example

16.6.2.2.8-bit MOVX without Bank Select: EMI0CF[4:2] = ‘001’ or ‘011’.

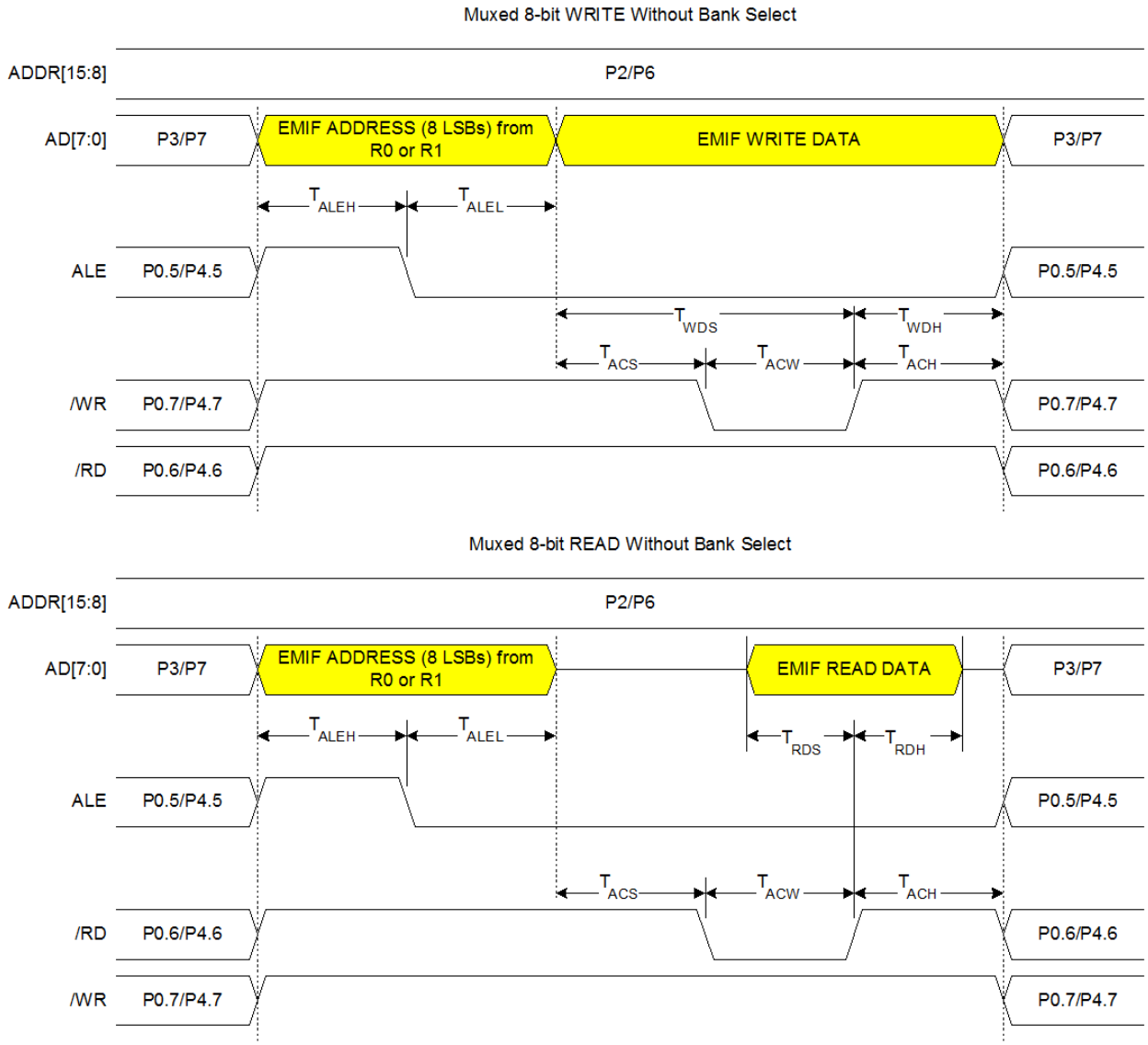


Figure 16.8. Multiplexed 8-bit MOVX without Bank Select Timing

18.2.3. Message Handler Registers

The Message Handler Registers are *read only* registers. Their flags can be read via the indexed access method with CAN0ADR, CAN0DATH, and CAN0DATL. The message handler registers provide interrupt, error, transmit/receive requests, and new data information.

Please refer to the Bosch CAN User's Guide for information on the function and use of the Message Handler Registers.

18.2.4. CIP-51 MCU Special Function Registers

C8051F04x family peripherals are modified, monitored, and controlled using Special Function Registers (SFR's). Only three of the CAN Controller's registers may be accessed directly with SFR's. However, all CAN Controller registers can be accessed indirectly using three CIP-51 MCU SFR's: the CAN Data Registers (CAN0DATH and CAN0DATL) and CAN Address Register (CAN0ADR).

18.2.5. Using CAN0ADR, CAN0DATH, and CANDATL to Access CAN Registers

Each CAN Controller Register has an index number (see Table 18.2). The CAN register address space is 128 words (256 bytes). A CAN register is accessed via the CAN Data Registers (CAN0DATH and CAN0DATL) when a CAN register's index number is placed into the CAN Address Register (CAN0ADR). For example, if the Bit Timing Register is to be configured with a new value, CAN0ADR is loaded with 0x03. The low byte of the desired value is accessed using CAN0DATL and the high byte of the bit timing register is accessed using CAN0DATH. CAN0DATL is bit addressable for convenience. To load the value 0x2304 into the Bit Timing Register:

```
CAN0ADR = 0x03;     // Load Bit Timing Register's index (Table 18.1)
CAN0DATH = 0x23;    // Move the upper byte into data reg high byte
CAN0DATL = 0x04;    // Move the lower byte into data reg low byte
```

Note: CAN0CN, CAN0STA, and CAN0TST may be accessed either by using the index method, or by direct access with the CIP-51 MCU SFR's. CAN0CN is located at SFR location 0xF8/SFR page 1 (SFR Definition 18.3), CAN0TST at 0xDB/SFR page 1 (SFR Definition 18.4), and CAN0STA at 0xC0/SFR page 1 (SFR Definition 18.5).

18.2.6. CAN0ADR Autoincrement Feature

For ease of programming message objects, CAN0ADR features autoincrementing for the index ranges 0x08 to 0x12 (Interface Registers 1) and 0x20 to 0x2A (Interface Registers 2). When the CAN0ADR register has an index in these ranges, **the CAN0ADR will autoincrement by 1 to point to the next CAN register 16-bit word upon a read/write of CAN0DATL**. This speeds programming of the frequently-accessed interface registers when configuring message objects.

NOTE: Table 18.2 below supersedes Figure 5 in Section 3, "Programmer's Model" of the Bosch CAN User's Guide.

SFR Definition 18.1. CAN0DATL: CAN Data Access Register Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000001
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xD8
SFR Page: 1

Bit7-0: CAN0DATL: CAN Data Access Register Low Byte.
The CAN0DAT Registers are used to read/write register values and data to and from the CAN Registers pointed to with the index number in the CAN0ADR Register.
The CAN0ADR Register is used to point the [CAN0DATH:CAN0DATL] to a desired CAN Register. The desired CAN Register's index number is moved into CAN0ADR. The CAN0DAT Register can then read/write to and from the CAN Register.

SFR Definition 18.2. CAN0ADR: CAN Address Index

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: 0xDA
SFR Page: 1

Bit7-0: CAN0ADR: CAN Address Index Register.
The CAN0ADR Register is used to point the [CAN0DATH:CAN0DATL] to a desired CAN Register. The desired CAN Register's index number is moved into CAN0ADR. The CAN0DAT Register can then read/write to and from the CAN Register.

Note: When the value of CAN0ADR is 0x08-0x12 and 0x20-0x2A (IF1 and IF2 registers), this register will autoincrement by 1 upon a write to CAN0DATL. See **Section "18.2.6. CAN0ADR Autoincrement Feature" on page 232**.

All CAN registers' functions/definitions are listed and described in the Bosch CAN User's Guide.

20.3. SPI0 Slave Mode Operation

When SPI0 is enabled and not configured as a master, it will operate as a SPI slave. As a slave, bytes are shifted in through the MOSI pin and out through the MISO pin by a master device controlling the SCK signal. A bit counter in the SPI0 logic counts SCK edges. When 8 bits have been shifted through the shift register, the SPIF flag is set to logic 1, and the byte is copied into the receive buffer. Data is read from the receive buffer by reading SPI0DAT. A slave device cannot initiate transfers. Data to be transferred to the master device is pre-loaded into the shift register by writing to SPI0DAT. Writes to SPI0DAT are double-buffered, and are placed in the transmit buffer first. If the shift register is empty, the contents of the transmit buffer will immediately be transferred into the shift register. When the shift register already contains data, the SPI will wait until the byte is transferred before loading it with the transmit buffer's contents.

When configured as a slave, SPI0 can be configured for 4-wire or 3-wire operation. The default, 4-wire slave mode, is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In 4-wire mode, the NSS signal is routed to a port pin and configured as a digital input. SPI0 is enabled when NSS is logic 0, and disabled when NSS is logic 1. The bit counter is reset on a falling edge of NSS. Note that the NSS signal must be driven low at least 2 system clocks before the first active edge of SCK for each byte transfer. Figure 20.4 shows a connection diagram between two slave devices in 4-wire slave mode and a master device.

3-wire slave mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. NSS is not used in this mode, and does not get mapped to an external port pin through the crossbar. Since there is no way of uniquely addressing the device in 3-wire slave mode, SPI0 must be the only slave device present on the bus. It is important to note that in 3-wire slave mode there is no external means of resetting the bit counter that determines when a full byte has been received. The bit counter can only be reset by disabling and re-enabling SPI0 with the SPIEN bit. Figure 20.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

20.4. SPI0 Interrupt Sources

When SPI0 interrupts are enabled, the following four flags will generate an interrupt when they are set to logic 1:

Note: All of the following interrupt bits must be cleared by software.

1. The SPI Interrupt Flag, SPIF (SPI0CN.7) is set to logic 1 at the end of each byte transfer. This flag can occur in all SPI0 modes.
2. The Write Collision Flag, WCOL (SPI0CN.6) is set to logic 1 if a write to SPI0DAT is attempted when the transmit buffer has not been emptied to the SPI shift register. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written.This flag can occur in all SPI0 modes.
3. The Mode Fault Flag MODF (SPI0CN.5) is set to logic 1 when SPI0 is configured as a master, and for multi-master mode and the NSS pin is pulled low. When a Mode Fault occurs, the MSTEN and SPIEN bits in SPI0CN are set to logic 0 to disable SPI0 and allow another master device to access the bus.
4. The Receive Overrun Flag RXOVRN (SPI0CN.4) is set to logic 1 when configured as a slave, and a transfer is completed and the receive buffer still holds an unread byte from a previous transfer. The new byte is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte which caused the overrun is lost.

SFR Definition 20.3. SPI0CKR: SPI0 Clock Rate

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
SCR7	SCR6	SCR5	SCR4	SCR3	SCR2	SCR1	SCR0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x9D
SFR Page: 0

Bits 7-0: SCR7-SCR0: SPI0 Clock Rate
These bits determine the frequency of the SCK output when the SPI0 module is configured for master mode operation. The SCK clock frequency is a divided version of the system clock, and is given in the following equation, where *SYSCLK* is the system clock frequency and *SPI0CKR* is the 8-bit value held in the SPI0CKR register.

$$f_{SCK} = \frac{SYSCLK}{2 \times (SPI0CKR + 1)}$$

for 0 <= SPI0CKR <= 255

Example: If SYSCLK = 2 MHz and SPI0CKR = 0x04,

$$f_{SCK} = \frac{2000000}{2 \times (4 + 1)}$$

$$f_{SCK} = 200kHz$$

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 SFRs, 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).

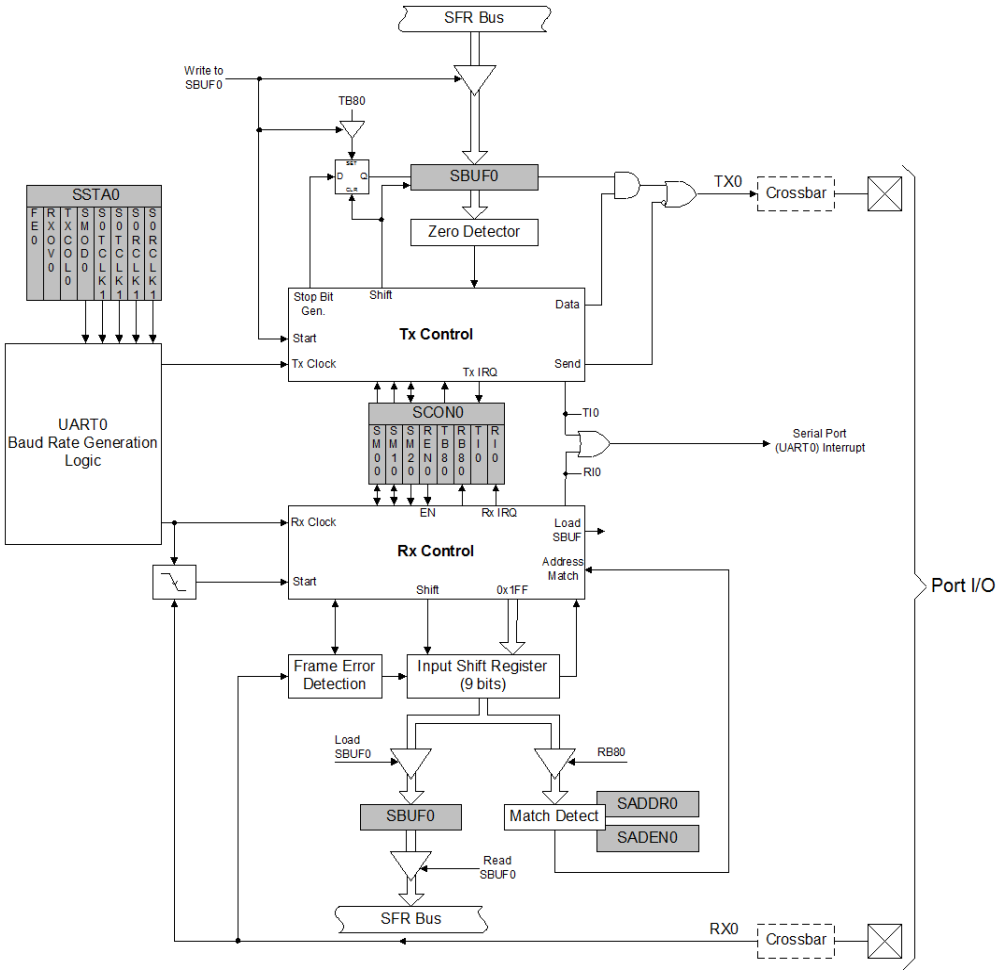


Figure 21.1. UART0 Block Diagram

21.1.4. Mode 3: 9-Bit UART, Variable Baud Rate

Mode 3 uses the Mode 2 transmission protocol with the Mode 1 baud rate generation. Mode 3 operation transmits 11 bits: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. The baud rate is derived from Timer 1 or Timer 2, 3, or 4 overflows, as defined by Equation 21.1 and Equation 21.3. Multiprocessor communications and hardware address recognition are supported, as described in **Section 21.2**.

21.2. Multiprocessor Communications

Modes 2 and 3 support multiprocessor communication between a master processor and one or more slave processors by special use of the ninth data bit and the built-in UART0 address recognition hardware. 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 ninth bit is logic 1; in a data byte, the ninth bit is always set to logic 0. UART0 will recognize as “valid” (i.e., capable of causing an interrupt) **two** types of addresses: (1) a *masked* address and (2) a *broadcast* address **at any given time**. Both are described below.

22. UART1

UART1 is an asynchronous, full duplex serial port offering modes 1 and 3 of the standard 8051 UART. Enhanced baud rate support allows a wide range of clock sources to generate standard baud rates (details in **Section "22.1. Enhanced Baud Rate Generation" on page 278**). Received data buffering allows UART1 to start reception of a second incoming data byte before software has finished reading the previous data byte.

UART1 has two associated SFRs: Serial Control Register 1 (SCON1) and Serial Data Buffer 1 (SBUF1). The single SBUF1 location provides access to both transmit and receive registers. Reading SBUF1 accesses the buffered Receive register; writing SBUF1 accesses the Transmit register.

With UART1 interrupts enabled, an interrupt is generated each time a transmit is completed (TI1 is set in SCON1), or a data byte has been received (RI1 is set in SCON1). The UART1 interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine. They must be cleared manually by software, allowing software to determine the cause of the UART1 interrupt (transmit complete or receive complete).

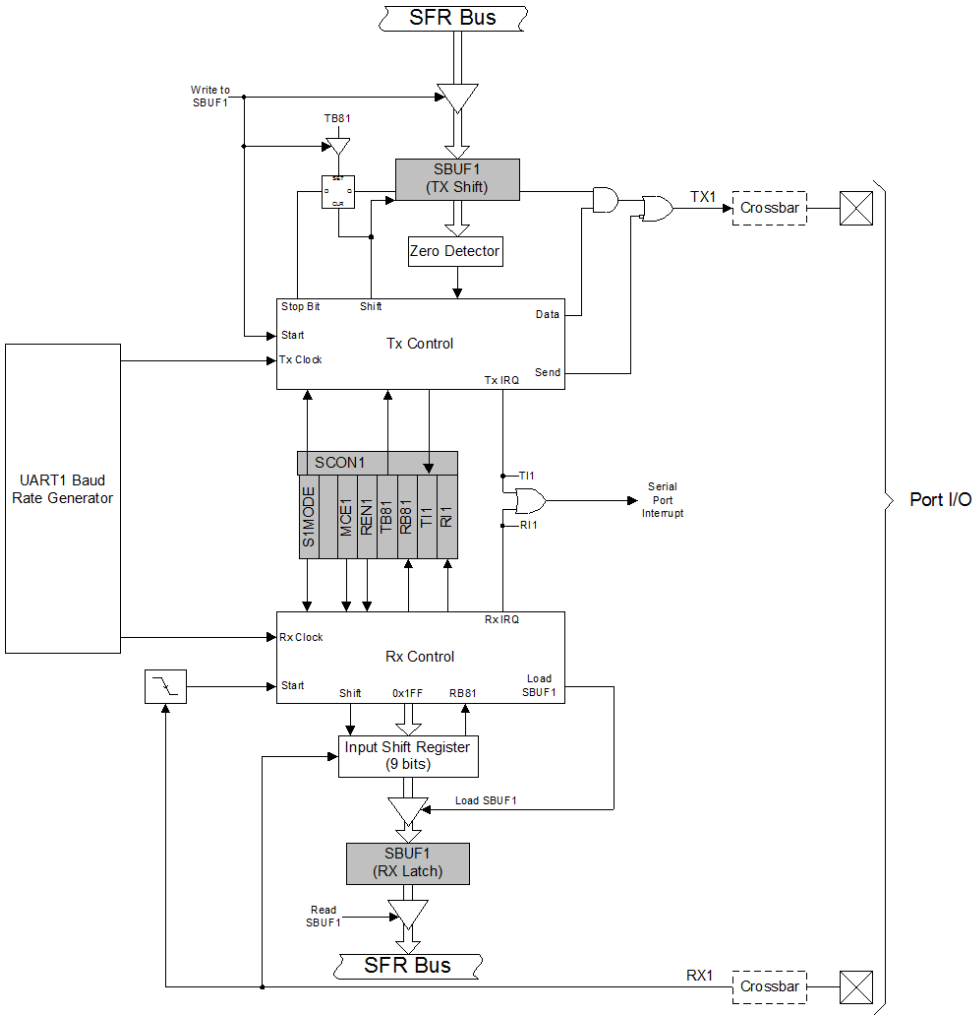


Figure 22.1. UART1 Block Diagram

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

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. TL0 can use either the system clock or an external input signal as its timebase. The TH0 register is restricted to a timer function sourced by the system clock or prescaled 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. When Timer 0 is operating 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 baud rates for the SMBus and/or UART, and/or initiate ADC conversions. While Timer 0 is operating in Mode 3, Timer 1 run control is handled through its mode settings. To run Timer 1 while Timer 0 is in Mode 3, set the Timer 1 Mode as 0, 1, or 2. To disable Timer 1, configure it for Mode 3.

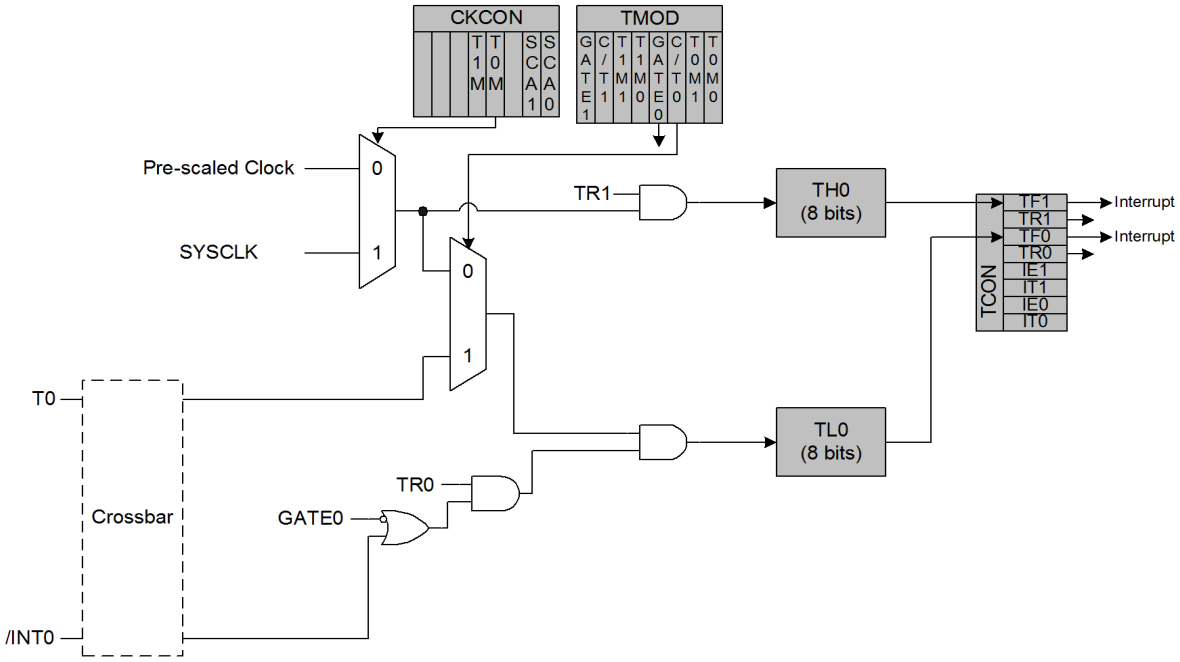


Figure 23.3. T0 Mode 3 Block Diagram

24.2.3. High-Speed Output Mode

In High-Speed Output mode, a module's associated CEXn pin is toggled each time a match occurs between the PCA Counter and the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn). Setting the TOGn, MATn, and ECOMn bits in the PCA0CPMn register enables the High-Speed Output mode.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to '0'; writing to PCA0CPHn sets ECOMn to '1'.

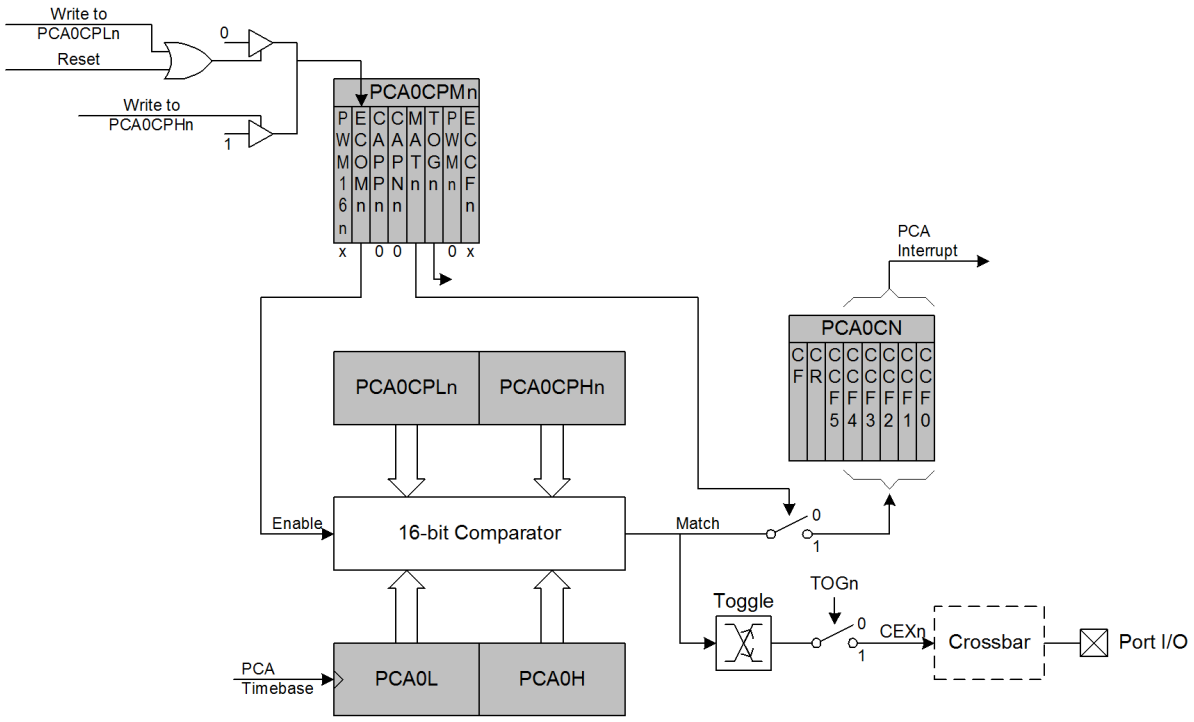


Figure 24.6. PCA High-Speed Output Mode Diagram