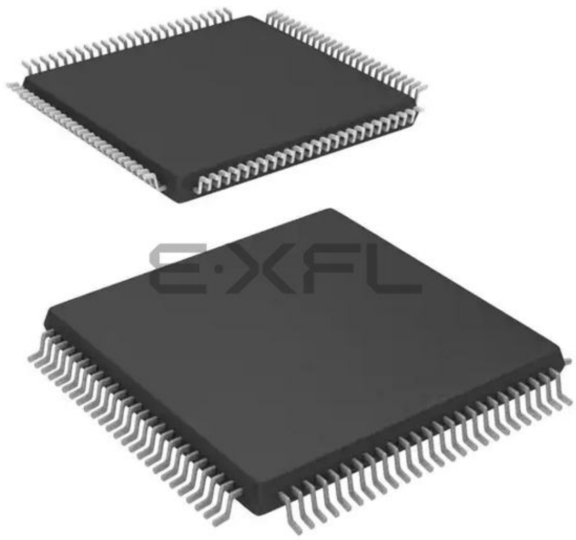


Welcome to E-XFL.COM

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/c8051f122

Table of Contents

List of Figures

11. CIP-51 Microcontroller

The MCU system controller core is the CIP-51 microcontroller. The CIP-51 is fully compatible with the MCS-51™ instruction set; standard 803x/805x assemblers and compilers can be used to develop software. The MCU family has a superset of all the peripherals included with a standard 8051. Included are five 16-bit counter/timers (see description in **Section 23**), two full-duplex UARTs (see description in **Section 21** and **Section 22**), 256 bytes of internal RAM, 128 byte Special Function Register (SFR) address space (see **Section 11.2.6**), and 8/4 byte-wide I/O Ports (see description in **Section 18**). The CIP-51 also includes on-chip debug hardware (see description in **Section 25**), and interfaces directly with the MCU's analog and digital subsystems providing a complete data acquisition or control-system solution in a single integrated circuit.

The CIP-51 Microcontroller core implements the standard 8051 organization and peripherals as well as additional custom peripherals and functions to extend its capability (see Figure 11.1 for a block diagram).

- Fully Compatible with MCS-51 Instruction Set
- 100 or 50 MIPS Peak Using the On-Chip PLL
- 256 Bytes of Internal RAM
- 8/4 Byte-Wide I/O Ports
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic
- Program and Data Memory Security

The CIP-51 includes the following features:

Performance

The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute, and usually have a maximum system clock of 12 MHz. By contrast, the CIP-51 core executes 70% of its instructions in one or two system clock cycles, with no instructions taking more than eight system clock cycles.

With the CIP-51's system clock running at 100 MHz, it has a peak throughput of 100 MIPS. The CIP-51 has a total of 109 instructions. The table below shows the total number of instructions that require each execution time.

Clocks to Execute	1	2	2/3	3	3/4	4	4/5	5	8
Number of Instructions	26	50	5	14	7	3	1	2	1

11.1. Instruction Set

The instruction set of the CIP-51 System Controller is fully compatible with the standard MCS-51™ instruction set; standard 8051 development tools can be used to develop software for the CIP-51. All CIP-51 instructions are the binary and functional equivalent of their MCS-51™ counterparts, including opcodes, addressing modes and effect on PSW flags. However, instruction timing is different than that of the standard 8051.

11.1.1. Instruction and CPU Timing

In many 8051 implementations, a distinction is made between machine cycles and clock cycles, with machine cycles varying from 2 to 12 clock cycles in length. However, the CIP-51 implementation is based solely on clock cycle timing. All instruction timings are specified in terms of clock cycles.

Due to the pipelined architecture of the CIP-51, most instructions execute in the same number of clock cycles as there are program bytes in the instruction. Conditional branch instructions take one less clock cycle to complete when the branch is not taken as opposed to when the branch is taken. Table 11.1 is the CIP-51 Instruction Set Summary, which includes the mnemonic, number of bytes, and number of clock cycles for each instruction.

11.1.2. MOVX Instruction and Program Memory

In the CIP-51, the MOVX instruction serves three purposes: accessing on-chip XRAM, accessing off-chip XRAM, and accessing on-chip program Flash memory. The Flash access feature provides a mechanism for user software to update program code and use the program memory space for non-volatile data storage (see **Section "15. Flash Memory" on page 199**). The External Memory Interface provides a fast access to off-chip XRAM (or memory-mapped peripherals) via the MOVX instruction. Refer to **Section "17. External Data Memory Interface and On-Chip XRAM" on page 219** for details.

Table 11.1. CIP-51 Instruction Set Summary

Mnemonic	Description	Bytes	Clock Cycles
	Arithmetic Operations		
ADD A, Rn	Add register to A	1	1
ADD A, direct	Add direct byte to A	2	2
ADD A, @Ri	Add indirect RAM to A	1	2
ADD A, #data	Add immediate to A	2	2
ADDC A, Rn	Add register to A with carry	1	1
ADDC A, direct	Add direct byte to A with carry	2	2
ADDC A, @Ri	Add indirect RAM to A with carry	1	2
ADDC A, #data	Add immediate to A with carry	2	2
SUBB A, Rn	Subtract register from A with borrow	1	1
SUBB A, direct	Subtract direct byte from A with borrow	2	2
SUBB A, @Ri	Subtract indirect RAM from A with borrow	1	2
SUBB A, #data	Subtract immediate from A with borrow	2	2
INC A	Increment A	1	1
INC Rn	Increment register	1	1
INC direct	Increment direct byte	2	2
INC @Ri	Increment indirect RAM	1	2

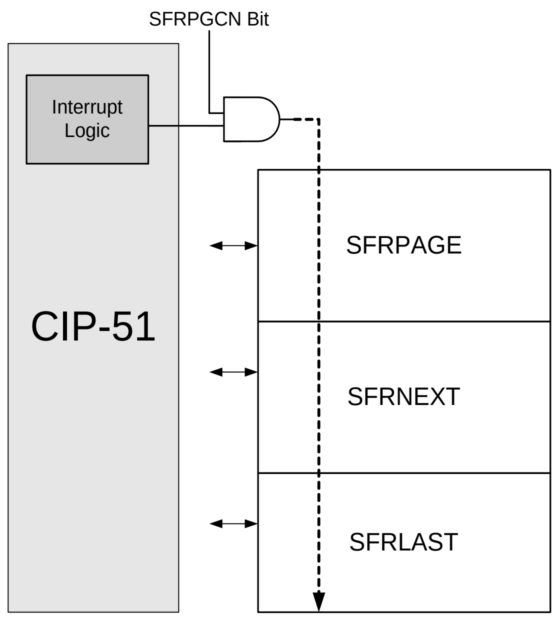


Figure 11.4. SFR Page Stack

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 (SFRPGCN). 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) is provided in Table 11.2. 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. Note that certain SFR's 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 SFR's are accessible from all SFR pages regardless of the SFRPAGE register value.

SFR Definition 11.17. EIP2: Extended Interrupt Priority 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	PS1	-	PADC2	PWADC2	PT4	PADC0	PT3	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xF7
SFR Page: All Pages

Bit7: UNUSED. Read = 0b, Write = don't care.
Bit6: ES1: UART1 Interrupt Priority Control.
This bit sets the priority of the UART1 interrupt.
0: UART1 interrupt set to low priority.
1: UART1 interrupt set to high priority.
Bit5: UNUSED. Read = 0b, Write = don't care.
Bit4: PADC2: ADC2 End Of Conversion Interrupt Priority Control.
This bit sets the priority of the ADC2 End of Conversion interrupt.
0: ADC2 End of Conversion interrupt set to low priority.
1: ADC2 End of Conversion interrupt set to high priority.
Bit3: PWADC2: ADC2 Window Compare Interrupt Priority Control.
This bit sets the priority of the ADC2 Window Compare interrupt.
0: ADC2 Window Compare interrupt set to low priority.
1: ADC2 Window Compare interrupt set to high priority.
Bit2: PT4: Timer 4 Interrupt Priority Control.
This bit sets the priority of the Timer 4 interrupt.
0: Timer 4 interrupt set to low priority.
1: Timer 4 interrupt set to high priority.
Bit1: PADC0: ADC0 End of Conversion Interrupt Priority Control.
This bit sets the priority of the ADC0 End of Conversion Interrupt.
0: ADC0 End of Conversion interrupt set to low priority.
1: ADC0 End of Conversion interrupt set to high priority.
Bit0: PT3: Timer 3 Interrupt Priority Control.
This bit sets the priority of the Timer 3 interrupts.
0: Timer 3 interrupt set to low priority.
1: Timer 3 interrupt set to high priority.

NOTES:

Electrical specifications for the precision internal oscillator are given in Table 14.1. Note that the system clock may be derived from the programmed internal oscillator divided by 1, 2, 4, or 8, as defined by the IFCN bits in register OSCICN.

SFR Definition 14.1. OSCICL: Internal Oscillator Calibration.

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

SFR Address: 0x8B
SFR Page: F

Bits 7–0: OSCICL: Internal Oscillator Calibration Register.
This register calibrates the internal oscillator period. The reset value for OSCICL defines the internal oscillator base frequency. The reset value is factory calibrated to generate an internal oscillator frequency of 24.5 MHz.

SFR Definition 14.2. OSCICN: Internal Oscillator Control

R/W	R	R/W	R	R/W	R/W	R/W	R/W	Reset Value
IOSCEN	IFRDY	-	-	-	-	IFCN1	IFCN0	11000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x8A
SFR Page: F

Bit 7: IOSCEN: Internal Oscillator Enable Bit.
0: Internal Oscillator Disabled.
1: Internal Oscillator Enabled.
Bit 6: IFRDY: Internal Oscillator Frequency Ready Flag.
0: Internal Oscillator not running at programmed frequency.
1: Internal Oscillator running at programmed frequency.
Bits 5–2: Reserved.
Bits 1–0: IFCN1-0: Internal Oscillator Frequency Control Bits.
00: Internal Oscillator is divided by 8.
01: Internal Oscillator is divided by 4.
10: Internal Oscillator is divided by 2.
11: Internal Oscillator is divided by 1.

14.7. Phase-Locked Loop (PLL)

A Phase-Locked-Loop (PLL) is included, which is used to multiply the internal oscillator or an external clock source to achieve higher CPU operating frequencies. The PLL circuitry is designed to produce an output frequency between 25 MHz and 100 MHz, from a divided reference frequency between 5 MHz and 30 MHz. A block diagram of the PLL is shown in Figure 14.2.

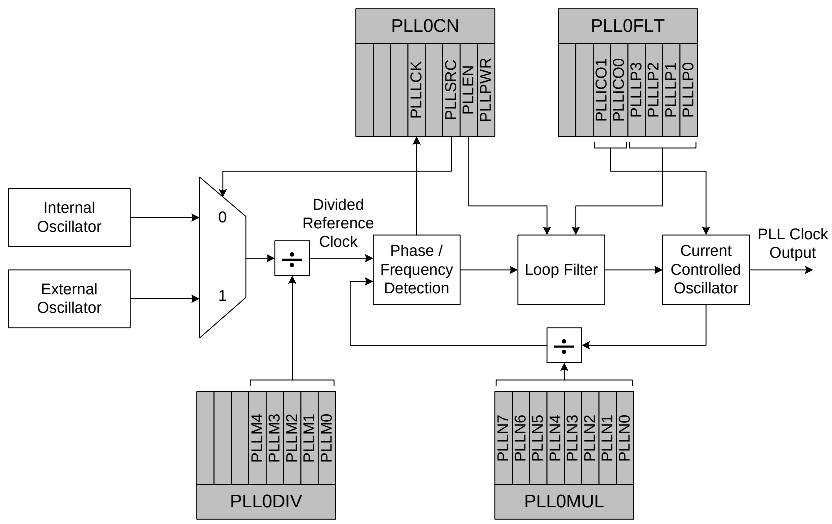


Figure 14.2. PLL Block Diagram

14.7.1. PLL Input Clock and Pre-divider

The PLL circuitry can derive its reference clock from either the internal oscillator or an external clock source. The PLLSRC bit (PLL0CN.2) controls which clock source is used for the reference clock (see SFR Definition 14.5). If PLLSRC is set to ‘0’, the internal oscillator source is used. Note that the internal oscillator divide factor (as specified by bits IFCN1-0 in register OSCICN) will also apply to this clock. When PLLSRC is set to ‘1’, an external oscillator source will be used. The external oscillator should be active and settled before it is selected as a reference clock for the PLL circuit. The reference clock is divided down prior to the PLL circuit, according to the contents of the PLLM4-0 bits in the PLL Pre-divider Register (PLL0DIV), shown in SFR Definition 14.6.

14.7.2. PLL Multiplication and Output Clock

The PLL circuitry will multiply the divided reference clock by the multiplication factor stored in the PLL0MUL register shown in SFR Definition 14.7. To accomplish this, it uses a feedback loop consisting of a phase/frequency detector, a loop filter, and a current-controlled oscillator (ICO). It is important to configure the loop filter and the ICO for the correct frequency ranges. The PLLLP3–0 bits (PLL0FLT.3–0) should be set according to the divided reference clock frequency. Likewise, the PLLICO1–0 bits (PLL0FLT.5–4) should be set according to the desired output frequency range. SFR Definition 14.8 describes the proper settings to use for the PLLLP3–0 and PLLICO1–0 bits. When the PLL is locked and stable at the desired frequency, the PLLLCK bit (PLL0CN.5) will be set to a ‘1’. The resulting PLL frequency will be set according to the equation:
Where “Reference Frequency” is the selected source clock frequency, PLLN is the PLL Multiplier, and PLLM is the PLL Pre-divider.

Table 18.1. Port I/O DC Electrical Characteristics

V_{DD} = 2.7 to 3.6 V, –40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Output High Voltage (V_{OH})	I_{OH} = -3 mA, Port I/O Push-Pull I_{OH} = -10 µA, Port I/O Push-Pull I_{OH} = -10 mA, Port I/O Push-Pull	V_{DD} – 0.7 V_{DD} – 0.1	 V_{DD} – 0.8		V
Output Low Voltage (V_{OL})	I_{OL} = 8.5 mA I_{OL} = 10 µA I_{OL} = 25 mA		 1.0	0.6 0.1	V
Input High Voltage (VIH)		0.7 x V_{DD}			
Input Low Voltage (VIL)				0.3 x V_{DD}	
Input Leakage Current	DGND < Port Pin < V_{DD}, Pin Tri-state Weak Pullup Off Weak Pullup On		 10	 ± 1	µA
Input Capacitance			5		pF

SFR Definition 18.6. P1: Port1 Data

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0x90
SFR Page: All Pages

Bits7–0: P1.[7:0]: Port1 Output Latch Bits.
(Write - Output appears on I/O pins per XBR0, XBR1, and XBR2 Registers)
0: Logic Low Output.
1: Logic High Output (open if corresponding P1MDOUT.n bit = 0).
(Read - Regardless of XBR0, XBR1, and XBR2 Register settings).
0: P1.n pin is logic low.
1: P1.n pin is logic high.

Notes:

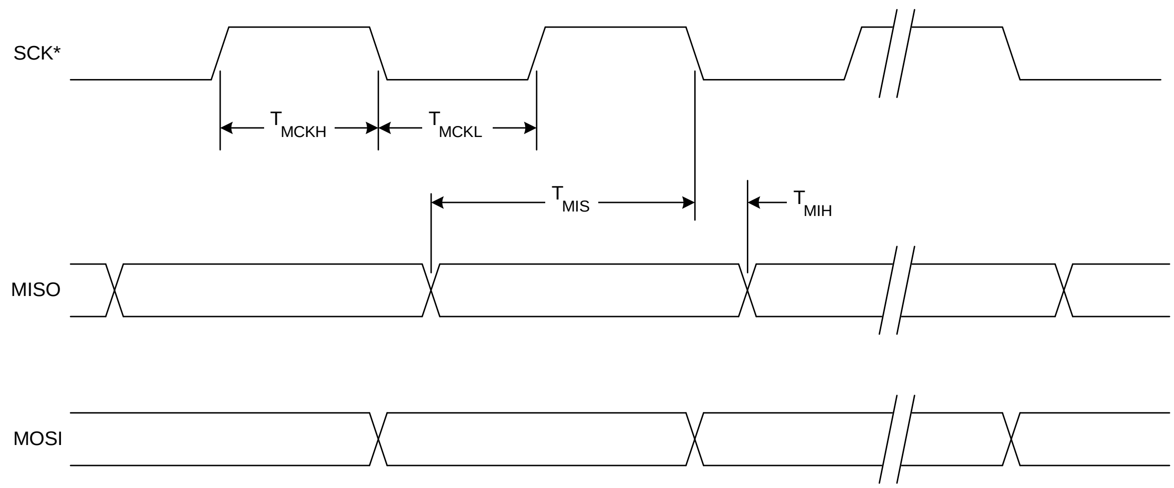
1. On C8051F12x devices, P1.[7:0] can be configured as inputs to ADC2 as AIN2.[7:0], in which case they are 'skipped' by the Crossbar assignment process and their digital input paths are disabled, depending on P1MDIN (See SFR Definition 18.7**)**. Note that in analog mode, the output mode of the pin is determined by the Port 1 latch and P1MDOUT (SFR Definition 18.8). See **Section "7. ADC2 (8-Bit ADC, C8051F12x Only)" on page 91** for more information about ADC2.
2. P1.[7:0] can be driven by the External Data Memory Interface (as Address[15:8] in Non-multiplexed mode). See **Section "17. External Data Memory Interface and On-Chip XRAM" on page 219** for more information about the External Memory Interface.

SFR Definition 18.7. P1MDIN: Port1 Input Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

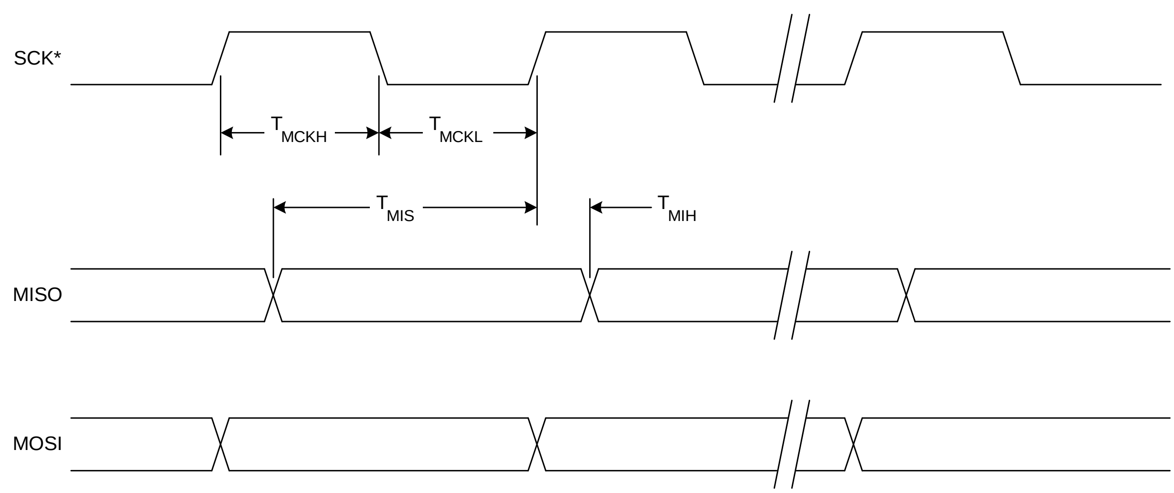
SFR Address: 0xAD
SFR Page: F

Bits7–0: P1MDIN.[7:0]: Port 1 Input Mode Bits.
0: Port Pin is configured in Analog Input mode. The digital input path is disabled (a read from the Port bit will always return '0'). The weak pullup on the pin is disabled.
1: Port Pin is configured in Digital Input mode. A read from the Port bit will return the logic level at the Pin. When configured as a digital input, the state of the weak pullup for the port pin is determined by the WEAKPUD bit (XBR2.7, see SFR Definition 18.3).



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 20.8. SPI Master Timing (CKPHA = 0)



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 20.9. SPI Master Timing (CKPHA = 1)

21.1.2. Mode 1: 8-Bit UART, Variable Baud Rate

Mode 1 provides standard asynchronous, full duplex communication using a total of 10 bits per data byte: one start bit, eight data bits (LSB first), and one stop bit. Data are transmitted from the TX0 pin and received at the RX0 pin. On receive, the eight data bits are stored in SBUF0 and the stop bit goes into RB80 (SCON0.2).

Data transmission begins when an instruction writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to logic 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: RI0 must be logic 0, and if SM20 is logic 1, the stop bit must be logic 1.

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

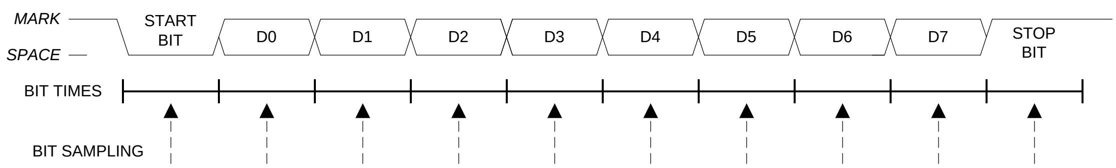


Figure 21.4. UART0 Mode 1 Timing Diagram

The baud rate generated in Mode 1 is a function of timer overflow. UART0 can use Timer 1 operating in *8-Bit Auto-Reload Mode*, or Timer 2, 3, or 4 operating in *Auto-reload Mode* to generate the baud rate (note that the TX and RX clocks are selected separately). On each timer overflow event (a rollover from all ones - (0xFF for Timer 1, 0xFFFF for Timer 2, 3, or 4) - to zero) a clock is sent to the baud rate logic.

Timers 1, 2, 3, or 4 are selected as the baud rate source with bits in the SSTA0 register (see SFR Definition 21.2). The transmit baud rate clock is selected using the S0TCLK1 and S0TCLK0 bits, and the receive baud rate clock is selected using the S0RCLK1 and S0RCLK0 bits.

When Timer 1 is selected as a baud rate source, the SMOD0 bit (SSTA0.4) selects whether or not to divide the Timer 1 overflow rate by two. On reset, the SMOD0 bit is logic 0, thus selecting the lower speed baud rate by default. The SMOD0 bit affects the baud rate generated by Timer 1 as shown in Equation 21.1.

The Mode 1 baud rate equations are shown below, where T1M is bit4 of register CKCON, TH1 is the 8-bit reload register for Timer 1, and [RCAPnH , RCAPnL] is the 16-bit reload register for Timer 2, 3, or 4.

Equation 21.1. Mode 1 Baud Rate using Timer 1

When SMOD0 = 0:

$$\text{Mode1\_BaudRate} = 1/32 \cdot \text{Timer1\_OverflowRate}$$

When SMOD0 = 1:

$$\text{Mode1\_BaudRate} = 1/16 \cdot \text{Timer1\_OverflowRate}$$

SFR Definition 21.2. SSTA0: UART0 Status and Clock Selection

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
FE0	RXOV0	TXCOL0	SMOD0	S0TCLK1	S0TCLK0	S0RCLK1	S0RCLK0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x91
SFR Page: 0

Bit7: FE0: Frame Error Flag.*
This flag indicates if an invalid (low) STOP bit is detected.
0: Frame Error has not been detected
1: Frame Error has been detected.

Bit6: RXOV0: Receive Overrun Flag.*
This flag indicates new data has been latched into the receive buffer before software has read the previous byte.
0: Receive overrun has not been detected.
1: Receive Overrun has been detected.

Bit5: TXCOL0: Transmit Collision Flag.*
This flag indicates user software has written to the SBUF0 register while a transmission is in progress.
0: Transmission Collision has not been detected.
1: Transmission Collision has been detected.

Bit4: 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.

Bits3–2: UART0 Transmit Baud Rate Clock Selection Bits

S0TCLK1	S0TCLK0	**Serial Transmit Baud Rate Clock Source**
0	0	Timer 1 generates UART0 TX Baud Rate
0	1	Timer 2 Overflow generates UART0 TX baud rate
1	0	Timer 3 Overflow generates UART0 TX baud rate
1	1	Timer 4 Overflow generates UART0 TX baud rate

Bits1–0: UART0 Receive Baud Rate Clock Selection Bits

S0RCLK1	S0RCLK0	**Serial Receive Baud Rate Clock Source**
0	0	Timer 1 generates UART0 RX Baud Rate
0	1	Timer 2 Overflow generates UART0 RX baud rate
1	0	Timer 3 Overflow generates UART0 RX baud rate
1	1	Timer 4 Overflow generates UART0 RX baud rate

***Note:** FE0, RXOV0, and TXCOL0 are flags only, and no interrupt is generated by these conditions.

SFR Definition 21.3. SBUF0: UART0 Data Buffer

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: 0x99
SFR Page: 0

Bits7–0: SBUF0.[7:0]: UART0 Buffer Bits 7–0 (MSB–LSB)
This is actually two registers; a transmit and a receive buffer register. When data is moved to SBUF0, it goes to the transmit buffer and is held for serial transmission. Moving a byte to SBUF0 is what initiates the transmission. When data is moved from SBUF0, it comes from the receive buffer.

SFR Definition 21.4. SADDR0: UART0 Slave Address

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: 0xA9
SFR Page: 0

Bits7–0: SADDR0.[7:0]: UART0 Slave Address
The contents of this register are used to define the UART0 slave address. Register SADEN0 is a bit mask to determine which bits of SADDR0 are checked against a received address: corresponding bits set to logic 1 in SADEN0 are checked; corresponding bits set to logic 0 are “don’t cares”.

SFR Definition 21.5. SADEN0: UART0 Slave Address Enable

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: 0xB9
SFR Page: 0

Bits7–0: SADEN0.[7:0]: UART0 Slave Address Enable
Bits in this register enable corresponding bits in register SADDR0 to determine the UART0 slave address.
0: Corresponding bit in SADDR0 is a “don’t care”.
1: Corresponding bit in SADDR0 is checked against a received address.

23.2. Timer 2, Timer 3, and Timer 4

Timers 2, 3, and 4 are 16-bit counter/timers, each formed by two 8-bit SFR's: TMRnL (low byte) and TMRnH (high byte) where n = 2, 3, and 4 for timers 2, 3, and 4 respectively. Timers 2 and 4 feature auto-reload, capture, and toggle output modes with the ability to count up or down. Timer 3 features auto-reload and capture modes, with the ability to count up or down. Capture Mode and Auto-reload mode are selected using bits in the Timer 2, 3, and 4 Control registers (TMRnCN). Toggle output mode is selected using the Timer 2 or 4 Configuration registers (TMRnCF). These timers may also be used to generate a square-wave at an external pin. As with Timers 0 and 1, Timers 2, 3, and 4 can use either the system clock (divided by one, two, or twelve), external clock (divided by eight) or transitions on an external input pin as its clock source. Timer 2 and 3 can be used to start an ADC Data Conversion and Timers 2, 3, and 4 can schedule DAC outputs. Timers 1, 2, 3, or 4 may be used to generate baud rates for UART 0. Only Timer 1 can be used to generate baud rates for UART 1.

The Counter/Timer Select bit C/Tn bit (TMRnCN.1) configures the peripheral as a counter or timer. Clearing C/Tn configures the Timer to be in a timer mode (i.e., the system clock or transitions on an external pin as the input for the timer). When C/Tn is set to 1, the timer is configured as a counter (i.e., high-to-low transitions at the Tn input pin increment (or decrement) the counter/timer register. Timer 3 and Timer 2 share the T2 input pin. Refer to **Section "18.1. Ports 0 through 3 and the Priority Crossbar Decoder" on page 238** for information on selecting and configuring external I/O pins for digital peripherals, such as the Tn pin.

Timer 2, 3, and 4 can use either SYSCLK, SYSCLK divided by 2, SYSCLK divided by 12, an external clock divided by 8, or high-to-low transitions on the Tn input pin as its clock source when operating in Counter/Timer with Capture mode. Clearing the C/Tn bit (TMRnCN.1) selects the system clock/external clock as the input for the timer. The Timer Clock Select bits TnM0 and TnM1 in TMRnCF can be used to select the system clock undivided, system clock divided by two, system clock divided by 12, or an external clock provided at the XTAL1/XTAL2 pins divided by 8 (see SFR Definition 23.13). When C/Tn is set to logic 1, a high-to-low transition at the Tn input pin increments the counter/timer register (i.e., configured as a counter).

23.2.1. Configuring Timer 2, 3, and 4 to Count Down

Timers 2, 3, and 4 have the ability to count down. When the timer's Decrement Enable Bit (DCENn) in the Timer Configuration Register (See SFR Definition 23.13) is set to '1', the timer can then count *up* or *down*. When DCENn = 1, the direction of the timer's count is controlled by the TnEX pin's logic level (Timer 3 shares the T2EX pin with Timer 2). When TnEX = 1, the counter/timer will count up; when TnEX = 0, the counter/timer will count down. To use this feature, TnEX must be enabled in the digital crossbar and configured as a digital input.

Note: When DCENn = 1, other functions of the TnEX input (i.e., capture and auto-reload) are not available. TnEX will only control the direction of the timer when DCENn = 1.