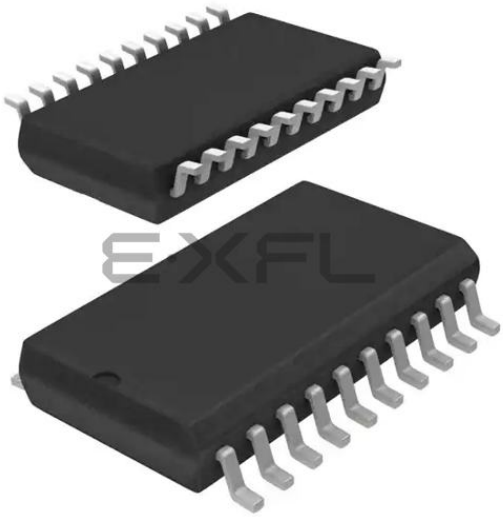


**Welcome to E-XFL.COM**

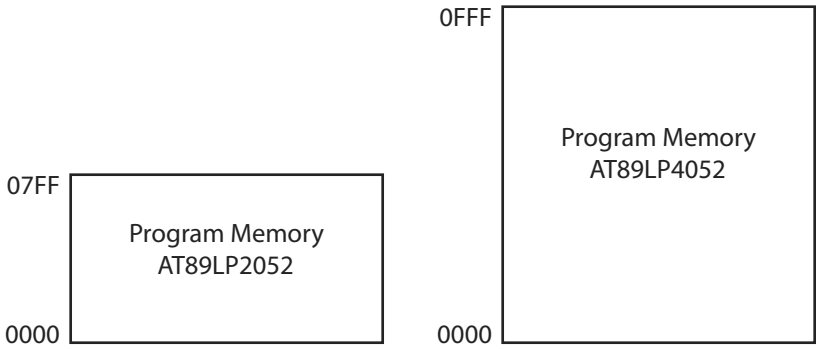
## What is "Embedded - Microcontrollers"?

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

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | 8051 |
| Core Size | 8-Bit |
| Speed | 20MHz |
| Connectivity | SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 15 |
| Program Memory Size | 4KB (4K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 256 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.4V ~ 5.5V |
| Data Converters | - |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 20-SOIC (0.295", 7.50mm Width) |
| Supplier Device Package | 20-SOIC |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/at89lp4052-20su |

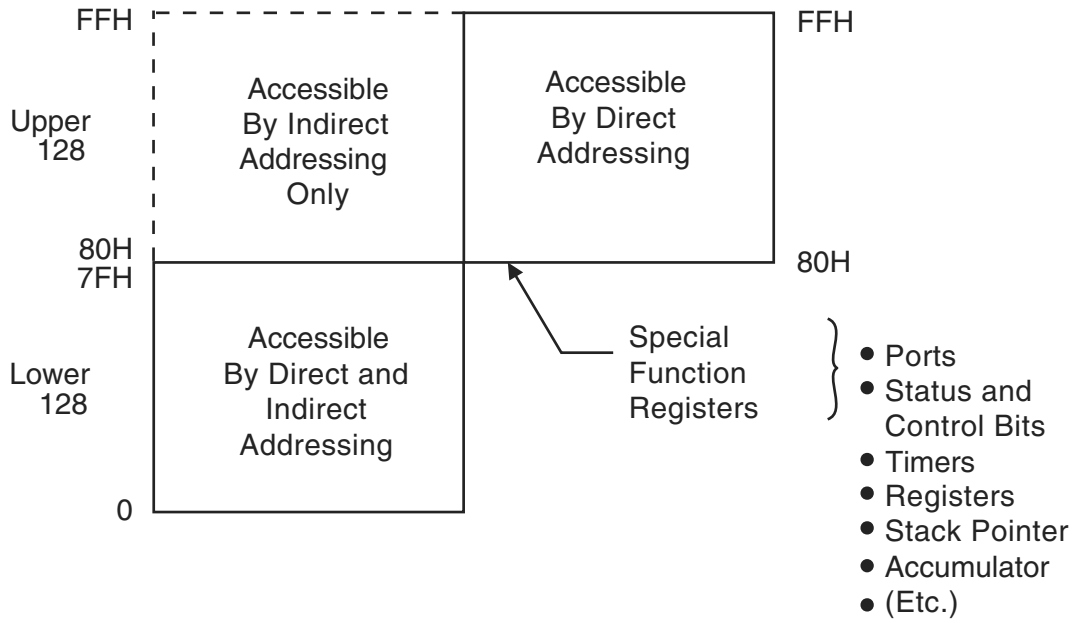
**Figure 5-1.** Program Memory Map




## 5.2 Data Memory

The AT89LP2052/LP4052 contains 256 bytes of general SRAM data memory plus 128 bytes of I/O memory. The lower 128 bytes of data memory may be accessed through both direct and indirect addressing. The upper 128 bytes of data memory and the 128 bytes of I/O memory share the same address space (see Figure 5-2). The upper 128 bytes of data memory may only be accessed using indirect addressing. The I/O memory can only be accessed through direct addressing and contains the Special Function Registers (SFRs). The lowest 32 bytes of data memory are grouped into 4 banks of 8 registers each. The RS0 and RS1 bits (PSW.3 and PSW.4) select which register bank is in use. Instructions using register addressing will only access the currently specified bank. The AT89LP2052/LP4052 does not support external data memory.

**Figure 5-2.** Data Memory Map



# 6. Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 6-1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect. User software should not write to these unlisted locations, since they may be used in future products to invoke new features.

**Table 6-1.** AT89LP2052/LP4052 SFR Map and Reset Values

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0F8H | | | | | | | | | 0FFH |
| 0F0H | B*<br>0000 0000 | | | | | | | | 0F7H |
| 0E8H | | | | | | | | | 0EFH |
| 0E0H | ACC*<br>0000 0000 | | | | | | | | 0E7H |
| 0D8H | | | | | | | | | 0DFH |
| 0D0H | PSW*<br>0000 0000 | | | | | SPCR<br>0000 0000 | | | 0D7H |
| 0C8H | | | | | | | | | 0CFH |
| 0C0H | | | P1M0<br>1111 1111 | P1M1<br>0000 0000 | | | P3M0<br>1111 1111 | P3M1<br>0000 0000 | 0C7H |
| 0B8H | IP*<br>x0x0 0000 | SADEN<br>0000 0000 | | | | | | | 0BFH |
| 0B0H | P3*<br>1111 1111 | | | | | | | IPH<br>x0x0 0000 | 0B7H |
| 0A8H | IE*<br>00x0 0000 | SADDR<br>0000 0000 | SPSR<br>000x xx00 | | | | | | 0AFH |
| 0A0H | | | | | | | WDTRST<br>(write-only) | WDTCON<br>0000 x000 | 0A7H |
| 98H | SCON*<br>0000 0000 | SBUF<br>xxxx xxxx | | | | | | | 9FH |
| 90H | P1*<br>1111 1111 | TCONB<br>0010 0100 | RL0<br>0000 0000 | RL1<br>0000 0000 | RH0<br>0000 0000 | RH1<br>0000 0000 | | ACSR<br>xx00 0000 | 97H |
| 88H | TCON*<br>0000 0000 | TMOD<br>0000 0000 | TL0<br>0000 0000 | TL1<br>0000 0000 | TH0<br>0000 0000 | TH1<br>0000 0000 | | | 8FH |
| 80H | | SP<br>0000 0111 | DPL<br>0000 0000 | DPH<br>0000 0000 | | | SPDR<br>xxxx xxxx | PCON<br>000x 0000 | 87H |

Note: *These SFRs are bit-addressable.

### 7.7 I/O Ports

The I/O ports of the AT89LP2052/LP4052 may be configured in four different modes. On the AT89LP2052/LP4052, all the I/O ports revert to input-only (tri-stated) mode at power-up or reset. In the standard 8051, all ports are weakly pulled high during power-up or reset. To enable 8051-like ports, the ports must be put into quasi-bidirectional mode by clearing the P1M0 and P3M0 SFRs.

### 7.8 Reset

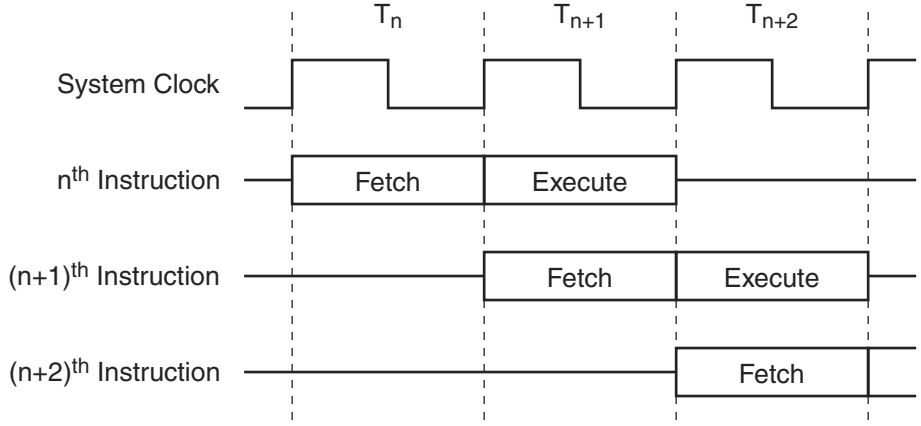
The RST pin in the AT89LP2052/LP4052 has different pulse width requirements than the standard 8051. The RST pin is sampled every clock cycle and must be held **high** for a minimum of two clock cycles, instead of 24 clock cycles, to be recognized as a valid reset pulse

## 8. Enhanced CPU

The AT89LP2052/LP4052 uses an enhanced 8051 CPU that runs at 6 to 12 times the speed of standard 8051 devices (or 3 to 6 times the speed of X2-mode 8051 devices). The increase in performance is due to two factors. First, the CPU fetches one instruction byte from the code memory every clock cycle. Second, the CPU uses a simple two-stage pipeline to fetch and execute instructions in parallel. This basic pipelining concept allows the CPU to obtain up to 1 MIPS per MHz. A simple example is shown in Figure 8-1.

The MCS-51 instruction set allows for instructions of variable length from 1 to 3 bytes. In a single-clock-per-byte-fetch system this means each instruction takes at least as many clocks as it has bytes to execute. A majority of the instructions in the AT89LP2052/LP4052 follow this rule: the instruction execution time in clock cycles equals the number of bytes per instruction with a few exceptions. Branches and Calls require an additional cycle to compute the target address and some other complex instructions require multiple cycles. See Section 22. “Instruction Set Summary” on page 52 for more detailed information on individual instructions. Figures 8-2 and 8-3 show examples of one- and two-byte instructions.

**Figure 8-1.** Parallel Instruction Fetches and Executions



## 13.1 Idle Mode

Setting the IDL bit in PCON enters Idle mode. Idle mode halts the internal CPU clock. The CPU state is preserved in its entirety, including the RAM, stack pointer, program counter, program status word, and accumulator. The Port pins hold the logic states they had at the time that Idle was activated. Idle mode leaves the peripherals running in order to allow them to wake up the CPU when an interrupt is generated. The Timer, UART and SPI blocks continue to function during Idle. The comparator and watchdog may be selectively enabled or disabled during Idle. Any enabled interrupt source or reset may terminate Idle mode. When exiting Idle mode with an interrupt, the interrupt will immediately be serviced, and following RETI the next instruction to be executed will be the one following the instruction that put the device into Idle.

## 13.2 Power-down Mode

Setting the Power-down (PD) bit in PCON enters Power-down mode. Power-down mode stops the oscillator and powers down the Flash memory in order to minimize power consumption. Only the power-on circuitry will continue to draw power during Power-down. During Power-down, the power supply voltage may be reduced to the RAM keep-alive voltage. The RAM contents will be retained, but the SFR contents are not guaranteed once $V_{CC}$ has been reduced. Power-down may be exited by external reset, power-on reset, or certain interrupts.

The user should not attempt to enter (or re-enter) the power-down mode for a minimum of 4 µs until after one of the following conditions has occurred: Start of code execution (after any type of reset), or Exit from power-down mode.

### 13.2.1 Interrupt Recovery from Power-down

Two external interrupts may be configured to terminate Power-down mode. Pins P3.2 and P3.3 may be used to exit Power-down through external interrupts $\overline{INT0}$ and $\overline{INT1}$. To wake up by external interrupts $\overline{INT0}$ or $\overline{INT1}$, that interrupt must be enabled and configured for level-sensitive operation. If configured as inputs, $\overline{INT0}$ and $\overline{INT1}$ should not be left floating during Power-down even if interrupt recovery is not used.

When terminating Power-down by an interrupt, two different wake-up modes are available. When PWDEX in PCON is zero, the wake-up period is internally timed. At the falling edge on the interrupt pin, Power-down is exited, the oscillator is restarted, and an internal timer begins counting. The internal clock will not be allowed to propagate to the CPU until after the timer has counted for nominally 2 ms. After the time-out period the interrupt service routine will begin. The interrupt pin may be held low until the device has timed out and begun executing, or it may return high before the end of the time-out period. If the pin remains low, the service routine should disable the interrupt before returning to avoid re-triggering the interrupt.

When PWDEX = “1”, the wake-up period is controlled externally by the interrupt. Again, at the falling edge on the interrupt pin, Power-down is exited and the oscillator is restarted. However, the internal clock will not propagate until the rising edge of the interrupt pin. The interrupt should be held low long enough for the selected clock source to stabilize.

### 13.2.2 Reset Exit from Power-down

The wake-up from Power-down through an external reset is similar to the interrupt with PWDEX = “0”. At the rising edge of RST, Power-down is exited, the oscillator is restarted, and an internal timer begins counting. The internal clock will not be allowed to propagate to the CPU until after the timer has counted for nominally 2 ms. The RST pin must be held high for longer than the time-out period to ensure that the device is reset properly. The device will begin executing once RST is brought back low.

**Table 14-4.** IPH – Interrupt Priority High Register

IPH = B7H — Reset Value = X0X0 0000B

Not Bit Addressable

| | – | PCH | – | PSH | PT1H | PX1H | PT0H | PX0H |
|---|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Symbol | Function |
|---|---|
| PCH | Comparator Interrupt Priority High |
| PSH | Serial Port Interrupt Priority High |
| PT1H | Timer 1 Interrupt Priority High |
| PX1H | External Interrupt 1 Priority High |
| PT0H | Timer 0 Interrupt Priority High |
| PX0H | External Interrupt 0 Priority High |

# 15. I/O Ports

All 15 port pins on the AT89LP2052/LP4052 may be configured to one of four modes: quasi-bidirectional (standard 8051 port outputs), push-pull output, open-drain output, or input-only. Port modes may be assigned in software on a pin-by-pin basis as shown in Table 15-1. All port pins default to input-only mode after reset. Each port pin also has a Schmitt-triggered input for improved input noise rejection. During Power-down all the Schmitt-triggered inputs are disabled with the exception of P3.2 and P3.3, which may be used to wake-up the device. Therefore P3.2 and P3.3 should not be left floating during Power-down.

**Table 15-1.** Configuration Modes for Port x, Bit y

| PxM0.y | PxM1.y | Port Mode |
|---|---|---|
| 0 | 0 | Quasi-bidirectional |
| 0 | 1 | Push-pull Output |
| 1 | 0 | Input Only (High Impedance) |
| 1 | 1 | Open-Drain Output |

## 15.1 Quasi-bidirectional Output

Port pins in quasi-bidirectional output mode function similar to standard 8051 port pins. A Quasi-bidirectional port can be used both as an input and output without the need to reconfigure the port. This is possible because when the port outputs a logic high, it is weakly driven, allowing an external device to pull the pin low. When the pin is driven low, it is driven strongly and able to sink a large current. There are three pull-up transistors in the quasi-bidirectional output that serve different purposes.

One of these pull-ups, called the “very weak” pull-up, is turned on whenever the port register for the pin contains a logic “1”. This very weak pull-up sources a very small current that will pull the pin high if it is left floating.

**Table 16-3.** TCONB – Timer/Counter Control Register B

TCONB = 91H — Reset Value = 0010 0100B

Not Bit Addressable

| Bit | PWM1EN | PWM0EN | PSC12 | PSC11 | PSC10 | PSC02 | PSC01 | PSC00 |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

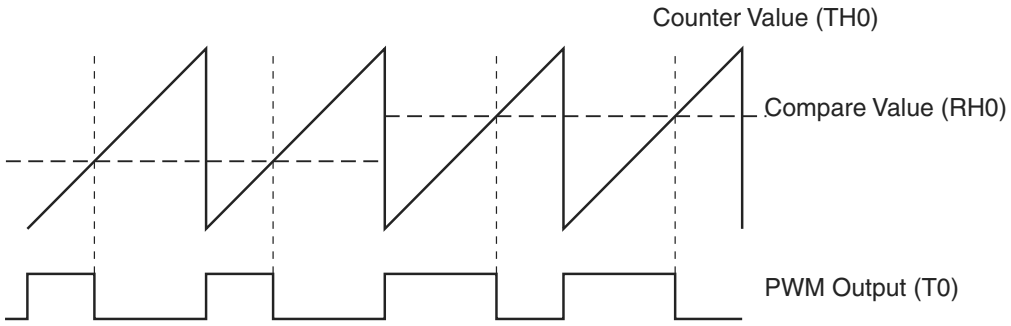
| Symbol | Function |
|---|---|
| PWM1EN | Configures Timer 1 for Pulse Width Modulation output on T1 (P3.5). |
| PWM0EN | Configures Timer 0 for Pulse Width Modulation output on T0 (P3.4). |
| PSC12<br>PSC11<br>PSC10 | Prescaler for Timer 1 Mode 0. The number of active bits in TL1 equals PSC1 + 1. After reset PSC1 = 100B which enables 5 bits of TL1 for compatibility with the 13-bit Mode 0 in AT89S2051. |
| PSC02<br>PSC01<br>PSC00 | Prescaler for Timer 0 Mode 0. The number of active bits in TL0 equals PSC0 + 1. After reset PSC0 = 100B which enables 5 bits of TL0 for compatibility with the 13-bit Mode 0 in AT89C52. |

## 16.5 Pulse Width Modulation

Timer 0 and Timer 1 may be independently configured as 8-bit asymmetrical pulse width modulators (PWM) by setting the PWM0EN or PWM1EN bits in TCONB, respectively. In PWM Mode the generated waveform is output on the timer's pin, T0 or T1. C/$\overline{T}$ must be set to “0” when in PWM mode. In Timer 0's PWM mode, TH0 acts as an 8-bit counter while RH0 stores the 8-bit compare value. When TH0 is 00H the PWM output is set high. When the TH0 count reaches the value stored in RH0 the PWM output is set low. Therefore, the pulse width is proportional to the value in RH0. To prevent glitches writes to RH0 only take effect on the FFH to 00H overflow of TH0. Timer 1 has the same behavior using TH1 and RH1. See Figure 16-5 for PWM waveform example. Setting RH0 to 00H will keep the PWM output low.

The PWM will only function if the timer is in Mode 0 or Mode 1. In Mode 0, TL0 acts as a logarithmic prescaler driving TH0 (see Figure 16-6). The PSC0 bits in TCONB control the prescaler value. In Mode 1, TL0 provides linear prescaling with an 8-bit auto-reload from RL0 (see Figure 16-7).

**Figure 16-5.** Asymmetrical Pulse Width Modulation



and a low in the next cycle, interrupt request flag IEx in TCON is set. Flag bit IEx then requests the interrupt. Since the external interrupt pins are sampled once each clock cycle, an input high or low should hold for at least 2 oscillator periods to ensure sampling. If the external interrupt is transition-activated, the external source has to hold the request pin high for at least two clock cycles, and then hold it low for at least two clock cycles to ensure that the transition is seen so that interrupt request flag IEx will be set. IEx will be automatically cleared by the CPU when the service routine is called if generated in edge-triggered mode. If the external interrupt is level-activated, the external source has to hold the request active until the requested interrupt is actually generated. Then the external source must deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.

# 18. Serial Interface

The serial port is full-duplex, which means it can transmit and receive simultaneously. It is also receive-buffered, which means it can begin receiving a second byte before a previously received byte has been read from the receive register. (However, if the first byte still has not been read when reception of the second byte is complete, the first byte will be lost.) The serial port receive and transmit registers are both accessed at Special Function Register SBUF. Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register. The serial port can operate in the following four modes.

**Mode 0:** Half-Duplex serial data enters or exits through RXD. TXD outputs the shift clock. Eight data bits are transmitted/received, with the LSB first. The baud rate is fixed at 1/2 the oscillator frequency.

**Mode 1:** 10 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in Special Function Register SCON. The baud rate is variable based on Timer 1 overflow.

**Mode 2:** 11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable ninth data bit, and a stop bit (1). On transmit, the 9th data bit (TB8 in SCON) can be assigned the value of “0” or “1”. For example, the parity bit (P, in the PSW) can be moved into TB8. On receive, the 9th data bit goes into RB8 in Special Function Register SCON, while the stop bit is ignored. The baud rate is programmable to either 1/16 or 1/32 the oscillator frequency.

**Mode 3:** 11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable ninth data bit, and a stop bit (1). In fact, Mode 3 is the same as Mode 2 in all respects except the baud rate, which in Mode 3 is variable based on Timer 1 overflow.

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1.

## 18.1 Multiprocessor Communications

Modes 2 and 3 have a special provision for multiprocessor communications. In these modes, 9 data bits are received, followed by a stop bit. The ninth bit goes into RB8. Then comes a stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt is activated only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON.

The following example shows how to use the serial interrupt for multiprocessor communications. When the master processor must transmit a block of data to one of several slaves, it first sends

out an address byte that identifies the target slave. An address byte differs from a data byte in that the 9th bit is “1” in an address byte and “0” in a data byte. With SM2 = 1, no slave is interrupted by a data byte. An address byte, however, interrupts all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave clears its SM2 bit and prepares to receive the data bytes that follows. The slaves that are not addressed set their SM2 bits and ignore the data bytes.

The SM2 bit has no effect in Mode 0 but can be used to check the validity of the stop bit in Mode 1. In a Mode 1 reception, if SM2 = 1, the receive interrupt is not activated unless a valid stop bit is received.

**Table 18-1.** SCON – Serial Port Control Register

SCON Address = 98H — Reset Value = 0000 0000B

Bit Addressable

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SM0/FE | SM1 | SM2 | REN | TB8 | RB8 | T1 | RI |

(SMOD0 = 0/1)[1]

| Symbol | Function |
|---|---|
| FE | Framing error bit. This bit is set by the receiver when an invalid stop bit is detected. The FE bit is not cleared by valid frames but should be cleared by software. The SMOD0 bit must be set to enable access to the FE bit. FE will be set regardless of the state of SMOD0. |
| SM0 | Serial Port Mode Bit 0, (SMOD0 must = 0 to access bit SM0) |
| SM1 | Serial Port Mode Bit 1<br>**SM0** **SM1** **Mode** **Description** **Baud Rate**[2]<br>0 0 0 shift register $f_{osc}/2$<br>0 1 1 8-bit UART variable<br>1 0 2 9-bit UART $f_{osc}/32$ or $f_{osc}/16$<br>1 1 3 9-bit UART variable |
| SM2 | Enables the Automatic Address Recognition feature in Modes 2 or 3. If SM2 = 1 then RI will not be set unless the received 9th data bit (RB8) is 1, indicating an address, and the received byte is a Given or Broadcast Address. In Mode 1, if SM2 = 1 then RI will not be activated unless a valid stop bit was received, and the received byte is a Given or Broadcast Address. In Mode 0, SM2 should be 0. |
| REN | Enables serial reception. Set by software to enable reception. Clear by software to disable reception. |
| TB8 | The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired. |
| RB8 | In Modes 2 and 3, the 9th data bit that was received. In Mode 1, if SM2 = 0, RB8 is the stop bit that was received. In Mode 0, RB8 is not used. |
| TI | Transmit interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software. |
| RI | Receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software. |

Notes: 1. SMOD0 is located at PCON.6.
2. $f_{osc}$ = oscillator frequency.

**Figure 18-2.** Serial Port Mode 1

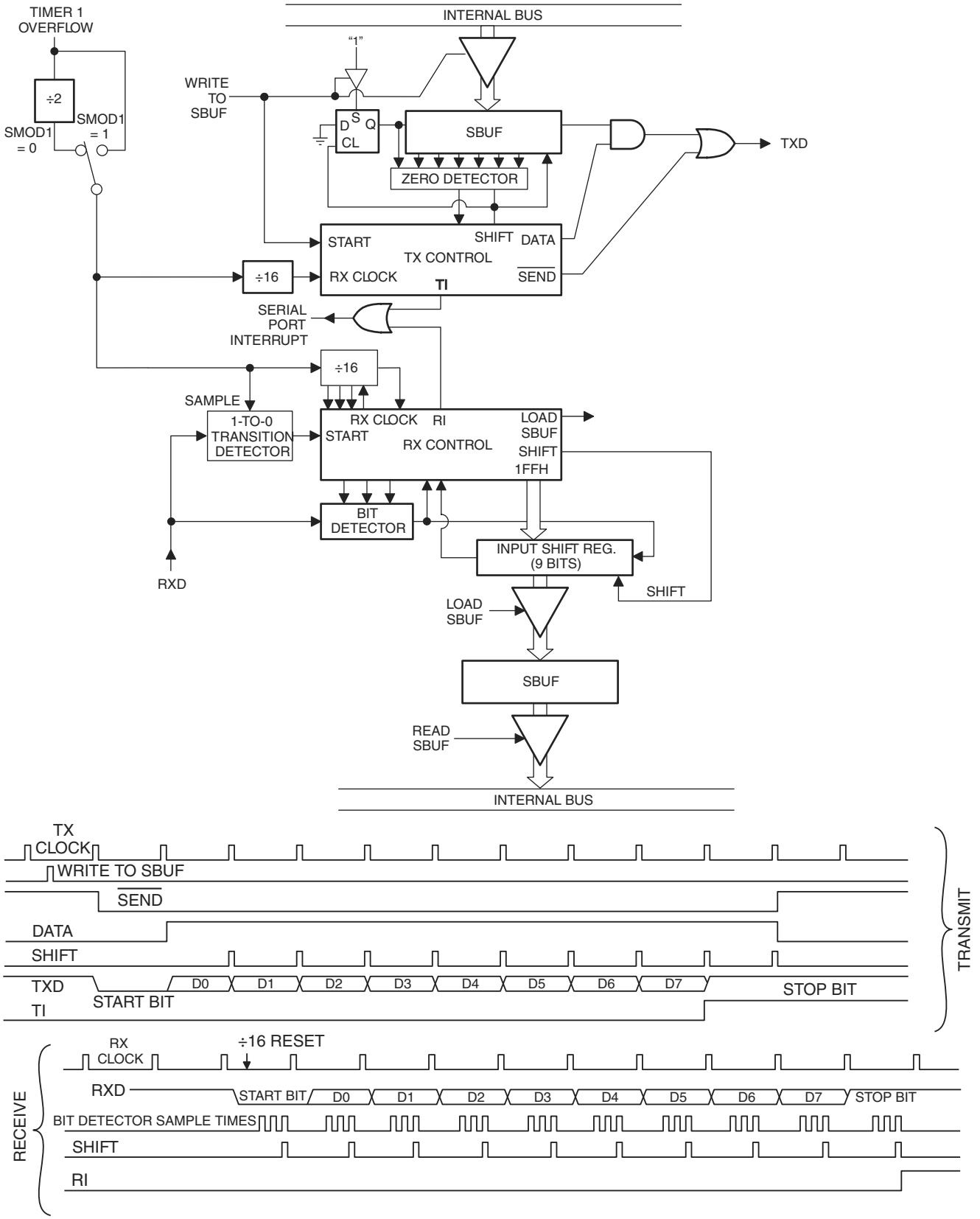
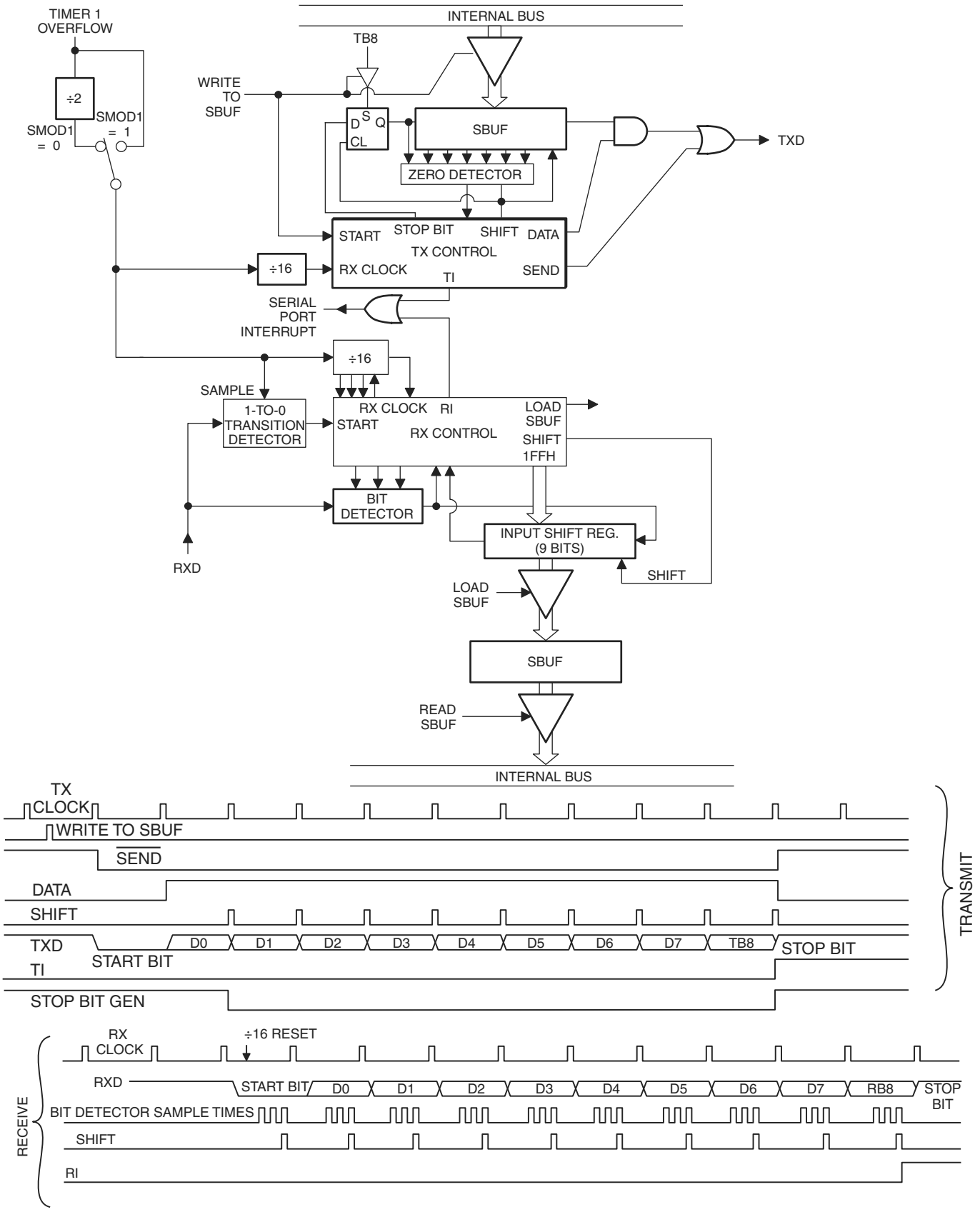


**Figure 18-4.** Serial Port Mode 3



# 20. Analog Comparator

A single analog comparator is provided on the AT89LP2052/LP4052. Comparator operation is such that the output is a logic “1” when the positive input AIN0 (P1.0) is greater than the negative input AIN1 (P1.1). Otherwise, the output is a zero. Setting the CEN bit in ACSR enables the comparator. When the comparator is first enabled, the comparator output and interrupt flag are guaranteed to be stable only after 10 µs. The corresponding comparator interrupt should not be enabled during that time, and the comparator interrupt flag must be cleared before the interrupt is enabled in order to prevent an immediate interrupt service. Before enabling the comparator the analog inputs should be tri-stated by putting P1.0 and P1.1 into input-only mode. See Section 15.5 “Port 1 Analog Functions” on page 22.

The comparator output is internally tied to the P3.6 pin. Instructions which read the pins of P3 will also read the comparator output directly. Read-Modify-Write instructions or Write instructions to P3.6 will access bit 6 of the Port 3 register without affecting the comparator.

The comparator may be configured to cause an interrupt under a variety of output value conditions by setting the CM bits in ACSR. The comparator interrupt flag CF in ACSR is set whenever the comparator output matches the condition specified by CM. The flag may be polled by software or may be used to generate an interrupt and must be cleared by software. The EC bit in IE must be set before CF will generate an interrupt.

## 20.1 Comparator Interrupt with Debouncing

The comparator output is sampled every clock cycle. The conditions on the analog inputs may be such that the comparator output will toggle excessively. This is especially true if applying slow moving analog inputs. Three debouncing modes are provided to filter out this noise. In debouncing mode, the comparator uses Timer 1 to modulate its sampling time. When a relevant transition occurs, the comparator waits until two Timer 1 overflows have occurred before resampling the output. If the new sample agrees with the expected value, CF is set. Otherwise the event is ignored. The filter may be tuned by adjusting the time-out period of Timer 1. Because Timer 1 is free running, the debouncer must wait for two overflows to guarantee that the sampling delay is at least 1 time-out period. Therefore after the initial edge event, the interrupt may occur between 1 and 2 time-out periods later. See Figure 20-1.

By default the comparator is disabled during Idle mode. To allow the comparator to function during Idle, the CIDL bit in ACSR must be set. When CIDL is set, the comparator can be used to wake-up the CPU from Idle if the comparator interrupt is enabled. The comparator is always disabled during Power-down mode.

**Figure 20-1.** Negative Edge with Debouncing Example

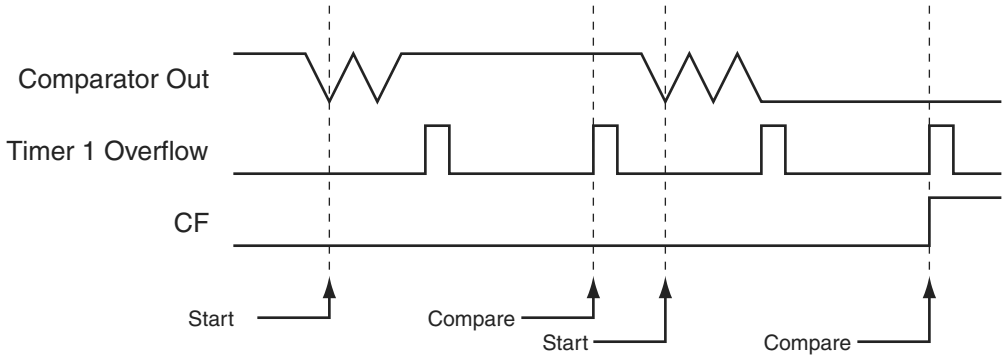


**Table 22-5.** Detailed Bit Instruction Summary

| Bit Instruction | Bytes | Clock Cycles | | Hex Code |
|---|---|---|---|---|
| | | 8051 | LP2052 | |
| SETB bit | 2 | 12 | 2 | D2 |
| CPL C | 1 | 12 | 1 | B3 |
| CPL bit | 2 | 12 | 2 | B2 |
| ANL C, bit | 2 | 24 | 2 | 82 |
| ANL C, /bit | 2 | 24 | 2 | B0 |
| ORL C, bit | 2 | 24 | 2 | 72 |
| ORL C, /bit | 2 | 24 | 2 | A0 |
| MOV C, bit | 2 | 12 | 2 | A2 |
| MOV bit, C | 2 | 24 | 2 | 92 |

**Table 22-6.** Detailed Branching Instruction Summary

| Branching Instruction | Bytes | Clock Cycles | | Hex Code |
|---|---|---|---|---|
| | | 8051 | LP2052 | |
| JC rel | 2 | 24 | 3 | 40 |
| JNC rel | 2 | 24 | 3 | 50 |
| JB bit, rel | 3 | 24 | 4 | 20 |
| JNB bit, rel | 3 | 24 | 4 | 30 |
| JBC bit, rel | 3 | 24 | 4 | 10 |
| JZ rel | 2 | 24 | 3 | 60 |
| JNZ rel | 2 | 24 | 3 | 70 |
| SJMP rel | 2 | 24 | 3 | 80 |
| ACALL addr11 | 2 | 24 | 3 | 11,31,51,71,91,B1,D1,F1 |
| LCALL addr16 | 3 | 24 | 4 | 12 |
| RET | 1 | 24 | 4 | 22 |
| RETI | 1 | 24 | 4 | 32 |
| AJMP addr11 | 2 | 24 | 3 | 01,21,41,61,81,A1,C1,E1 |
| LJMP addr16 | 3 | 24 | 4 | 02 |
| JMP @A+DPTR | 1 | 24 | 2 | 73 |
| CJNE A, direct, rel | 3 | 24 | 4 | B5 |
| CJNE A, #data, rel | 3 | 24 | 4 | B4 |
| CJNE Rn, #data, rel | 3 | 24 | 4 | B8-BF |
| CJNE @Ri, #data, rel | 3 | 24 | 4 | B6-B7 |
| DJNZ Rn, rel | 2 | 24 | 3 | D8-DF |
| DJNZ direct, rel | 3 | 24 | 4 | D5 |
| NOP | 1 | 12 | 1 | 00 |

# 23. Programming the Flash Memory

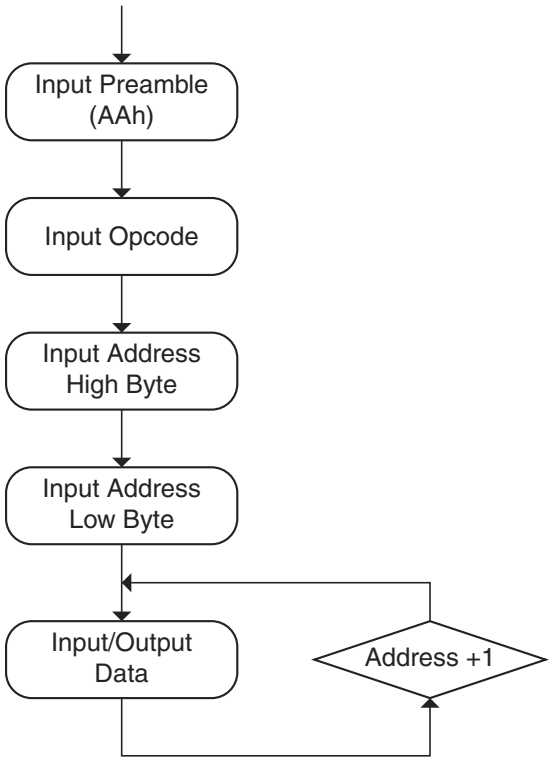
The AT89LP2052/LP4052 offers 2/4K bytes of In-System Programmable (ISP) non-volatile Flash code memory. In addition, the device contains a 32-byte User Signature Row and a 32-byte read-only Atmel Signature Row. The memory organization is shown in Table 23-1. The Memory is divided into pages of 32 bytes each. A single read or write command may only access a single page in the memory.

**Table 23-1.** Memory Organization

| Device # | Code Size | Page Size | # Pages | Address Range |
|---|---|---|---|---|
| AT89LP2052 | 2K bytes | 32 bytes | 64 | 0000H - 07FFH |
| AT89LP4052 | 4K bytes | 32 bytes | 128 | 0000H - 0FFFH |

The AT89LP2052/LP4052 provides two flexible interfaces for programming the Flash memory: a parallel interface which uses 10 pins; and a serial interface which uses the 4 SPI pins. The parallel and serial programming algorithms are identical. Both interfaces support the same command format where each command is issued to the device one byte at a time. Commands consist of a preamble byte for noise immunity, an opcode byte, two address bytes, and from 1 to 32 data bytes. Figure 23-1 shows a simplified flow chart of a command sequence.

**Figure 23-1.** Command Sequence Flow Chart



## 23.1 Programming Command Summary

| Command | Preamble | Opcode | Addr High | Addr Low | Data 0 | Data n |
|---|---|---|---|---|---|---|
| Program Enable[1] | 1010 1010 | 1010 1100 | 0101 0011 | | | |
| Chip Erase | 1010 1010 | 1000 1010 | | | | |
| Read Status | 1010 1010 | 0110 0000 | xxxx xxxx | xxxx xxxx | Status Out | |
| Load Code **Page Buffer**[2] | 1010 1010 | 0101 0001 | xxxx xxxx | xxxB BBBB | DataIn 0 ... DataIn n | |
| Write Code **Page**[2] | 1010 1010 | 0101 0000 | xxxx AAAA | AAAB BBBB | DataIn 0 ... DataIn n | |
| Read Code **Page**[2] | 1010 1010 | 0011 0000 | xxxx AAAA | AAAB BBBB | DataOut 0 ... DataOut n | |
| Write User Fuses[3] | 1010 1010 | 1110 0001 | xxxx xxxx | xxxx xxxx | xxxx FFFF | |
| Read User Fuses[3] | 1010 1010 | 0110 0001 | xxxx xxxx | xxxx xxxx | xxxx FFFF | |
| Write Lock Bits[4] | 1010 1010 | 1110 0100 | xxxx xxxx | xxxx xxxx | xxxx xxLL | |
| Read Lock Bits[4] | 1010 1010 | 0110 0100 | xxxx xxxx | xxxx xxxx | xxxx xxLL | |
| Write User Signature Page[2] | 1010 1010 | 0101 0010 | xxxx xxxx | xxxB BBBB | DataIn 0 ... DataIn n | |
| Read User Signature Page[2] | 1010 1010 | 0011 0010 | xxxx xxxx | xxxB BBBB | DataOut 0 ... DataOut n | |
| Read Atmel Signature Page[2][5] | 1010 1010 | 0011 1000 | xxxx xxxx | xxxB BBBB | DataOut 0 ... DataOut n | |

Notes:
1. Program Enable must be the **first** command issued after entering into programming mode.
2. Any number of Data bytes from 1 to 32 may be written/read. The internal address is incremented between each byte.
3. **Fuse Bit Definitions**:

| | | |
|---|---|---|
| Bit 0 | ISP Enable* | Enable = 0/Disable = 1 |
| Bit 1 | XTAL Osc Bypass | Enable = 0/Disable = 1 |
| Bit 2 | User Row Programming | Enable = 0/Disable = 1 |
| Bit 3 | System Clock Out | Enable = 0/Disable = 1 |

*The AT89LP2052/LP4052 has ISP enabled by default from the factory. However, if ISP is later disabled, the ISP Enable Fuse must be enabled by using Parallel Programming before entering ISP mode.
**When disabling the ISP fuse during ISP, the current ISP session will remain active until RST is brought low.**

4. **Lock Bit Definitions**:

| | | |
|---|---|---|
| Bit 0 | Lock Bit 1 | Locked = 0/Unlocked = 1 |
| Bit 1 | Lock Bit 2 | Locked = 0/Unlocked = 1 |

5. **Atmel Signature Byte**:

AT89LP2052: Address 00H = 1EH
01H = 25H
02H = FFH

AT89LP4052: Address 00H = 1EH
01H = 45H
02H = FFH

6. **Symbol Key**:

A: Page Address Bit
B: Byte Address Bit
F: Fuse Bit Data
L: Lock Bit Data
x: Don't Care

## 23.2 Status Register

The current state of the memory may be accessed by reading the status register. The status register is shown in Table 23-2.

**Table 23-2.** Status Register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | – | – | – | – | $\overline{\text{LOAD}}$ | SUCCESS | $\overline{\text{WRTINH}}$ | $\overline{\text{BUSY}}$ |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Symbol | Function |
|---|---|
| $\overline{\text{LOAD}}$ | Load flag. Cleared low by the load page buffer command and set high by the next memory write. This flag signals that the page buffer was previously loaded with data by the load page buffer command. |
| SUCCESS | Success flag. Cleared low at the start of a programming cycle and will only be set high if the programming cycle completes without interruption from the brownout detector. |
| $\overline{\text{WRTINH}}$ | Write Inhibit flag. Cleared low by the brownout detector (BOD) whenever programming is inhibited due to $V_{CC}$ falling below the minimum required programming voltage. If a BOD episode occurs during programming, the SUCCESS flag will remain low after the cycle is complete. $\overline{\text{WRTINH}}$ low also forces $\overline{\text{BUSY}}$ low. |
| $\overline{\text{BUSY}}$ | Busy flag. Cleared low whenever the memory is busy programming or if write is currently inhibited. |

## 23.3 $\overline{\text{DATA}}$ Polling

The AT89LP2052/LP4052 implements $\overline{\text{DATA}}$ polling to indicate the end of a programming cycle. While the device is busy, any attempted read of the last byte written will return the data byte with the MSB complemented. Once the programming cycle has completed, the true value will be accessible. During Erase the data is assumed to be FFH and $\overline{\text{DATA}}$ polling will return 7FH. When writing multiple bytes in a page, the $\overline{\text{DATA}}$ value will be the last data byte loaded before programming begins, not the written byte with the highest physical address within the page.

## 23.4 Parallel Programming

Parallel Programming Mode is enabled by applying $V_{PP}$ to the RST pin. The connections required during parallel mode are shown in Figure 23-2. During parallel programming, Port 1 is configured as an 8-bit wide bidirectional command bus. Data on P1 is strobed by a positive pulse on the XTAL1 pin. No other clock is required. The interface is enabled by pulling $\overline{\text{CS}}$ (P3.2) low. P3.1 acts as RDY/$\overline{\text{BSY}}$, and will be pulled low to indicate that the device is busy regardless of the state of $\overline{\text{CS}}$.

### 23.4.5 Load Code Page Buffer

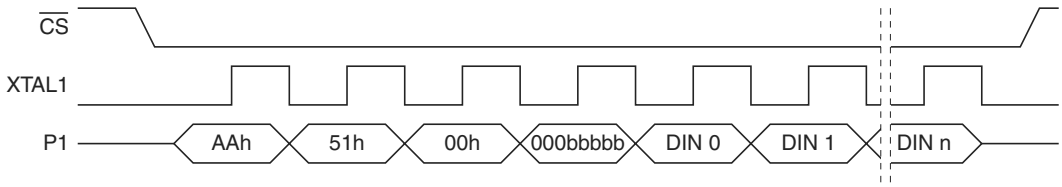
**Function**:

- Loads 1 page (1 to 32 bytes) of data into the temporary page buffer but does not start programming.
- Use for interruptible loads or loading non-contiguous bytes to a page.
- The byte address (offset in page) is initialized to bits [4:0] of the low address byte. One byte of data is loaded from P1 for the current address by the positive edge of a XTAL1 pulse. The internal address is incremented by one on the negative edge of the XTAL1 pulse. The address will wrap around to the 1st byte of the page when incremented past 31, however previously loaded bytes should not be re-loaded.
- The Load Page Buffer command needs to be followed by a write command as the internal buffer is not cleared until either the next write has completed or the programming session ends.
- Clears Bit 3 of the status byte to signal that the buffer contains data.

**Usage**:

1. Bring $\overline{CS}$ (P3.2) low.
2. Drive P1 to AAh and pulse XTAL1 high.
3. Drive P1 to 51h and pulse XTAL1 high.
4. Drive P1 to 00h and pulse XTAL1 high.
5. Drive P1 with bits [4:0] of address and pulse XTAL1 high.
6. To load data bytes, drive data on P1 and pulse XTAL1 high to load one byte and increment to the next address. Repeat for additional bytes. Only 1-32 bytes may be programmed at one time, including any bytes loaded by a previous load page buffer command. Bytes should not be loaded more than once.
7. Bring $\overline{CS}$ high.

**Figure 23-9.** Load Page Buffer Sequence




Note: The waveform on this page is not to scale.

### 23.4.13 Write User Fuses

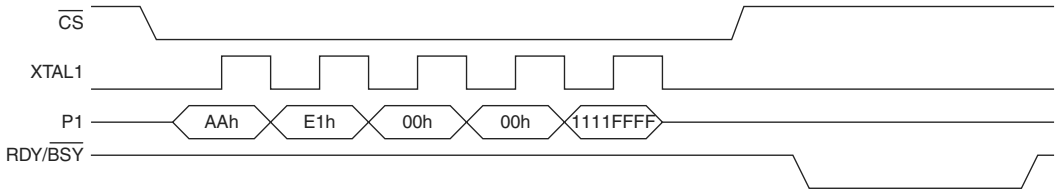
**Function**:

- Program User Fuses.
- Unimplemented bits should always be written with 1s.

**Usage**:

1. Bring $\overline{CS}$ (P3.2) low.
2. Drive P1 to AAh and pulse XTAL1 high.
3. Drive P1 to E1h and pulse XTAL1 high.
4. Drive P1 to 00h and pulse XTAL1 high.
5. Drive P1 to 00h and pulse XTAL1 high.
6. Drive data on P1 and pulse XTAL1 high.
7. Drive $\overline{CS}$ high.
8. Wait 4 ms, monitor P3.1, or poll data/status.

**Figure 23-17.** Write User Fuses Sequence




### 23.4.14 Read User Fuses

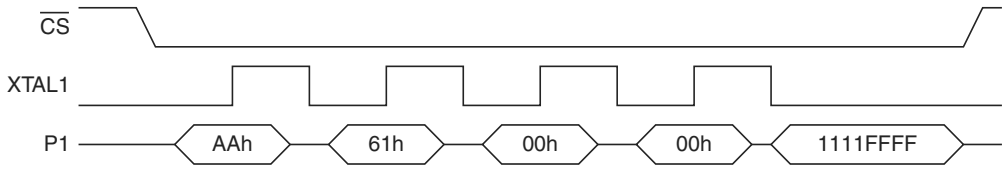
**Function**:

- Read status of User Fuses

**Usage**:

1. Bring $\overline{CS}$ (P3.2) low.
2. Drive P1 to 0xAA and pulse XTAL1 high.
3. Drive P1 to 0x61 and pulse XTAL1 high.
4. Drive P1 to 0x00 and pulse XTAL1 high.
5. Drive P1 to 0x00 and bring XTAL1 high.
6. Tri-state P1.
7. Bring XTAL1 low.
8. Read data from P1.
9. Drive $\overline{CS}$ high.

**Figure 23-18.** Read User Fuses Sequence




Note: The waveforms on this page are not to scale.

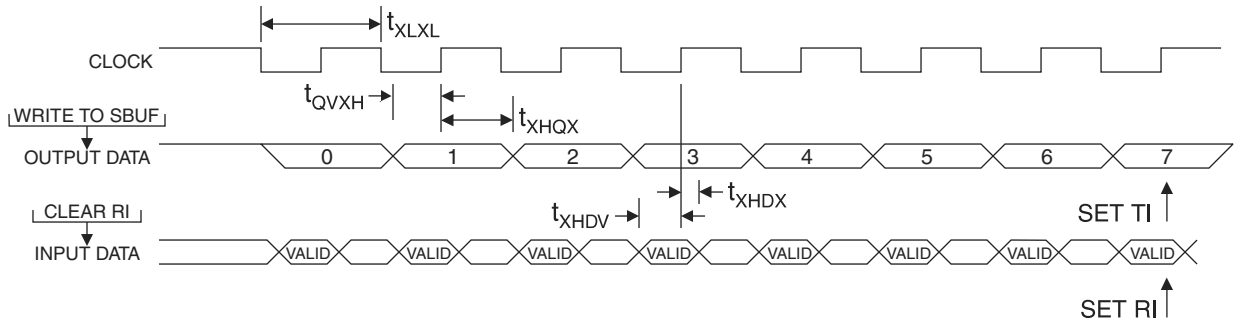
## 24.5 Serial Port Timing: Shift Register Mode

**Table 24-4.** Serial Port Shift Register Timing Parameters (1)

| Symbol | Parameter | Variable Oscillator | | Units |
|---|---|---|---|---|
| | | Min | Max | |
| $t_{XLXL}$ | Serial Port Clock Cycle Time | $2t_{CLCL}$ -15 | | µs |
| $t_{QVXH}$ | Output Data Setup to Clock Rising Edge | $t_{CLCL}$ -15 | | ns |
| $t_{XHQX}$ | Output Data Hold after Clock Rising Edge | $t_{CLCL}$ -15 | | ns |
| $t_{XHDX}$ | Input Data Hold after Clock Rising Edge | 0 | | ns |
| $t_{XHDV}$ | Input Data Valid to Clock Rising Edge | 15 | | ns |

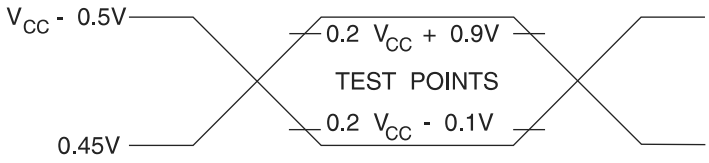
Note: 1. The values in this table are valid for $V_{CC}$ = 2.4V to 5.5V and Load Capacitance = 80 pF.

**Figure 24-6.** Shift Register Mode Timing Waveforms



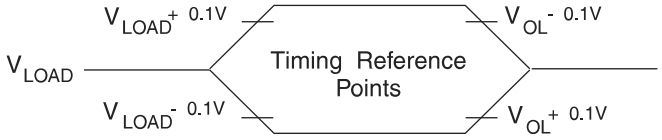

## 24.6 Test Conditions

### 24.6.1 AC Testing Input/Output Waveforms(1)



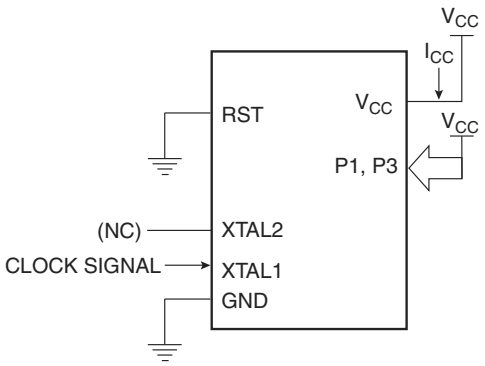

Note: 1. AC Inputs during testing are driven at $V_{CC}$ - 0.5V for a logic “1” and 0.45V for a logic “0”. Timing measurements are made at $V_{IH}$ min. for a logic “1” and $V_{IL}$ max. for a logic “0”.

### 24.6.2 Float Waveforms(1)

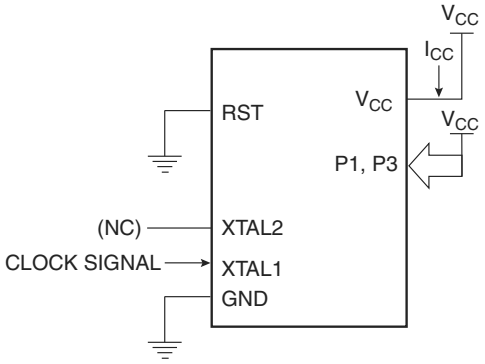



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when 100 mV change from the loaded $V_{OH}/V_{OL}$ level occurs.

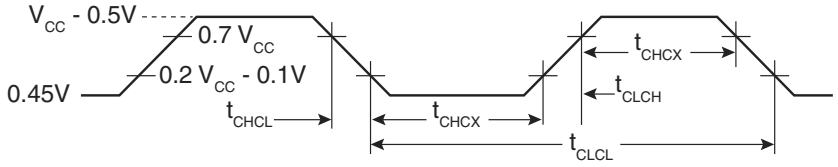
### 24.6.3 $I_{CC}$ Test Condition, Active Mode, All Other Pins are Disconnected



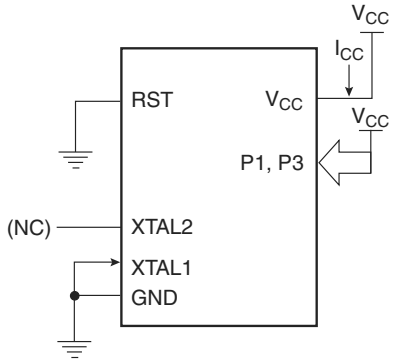

### 24.6.4 $I_{CC}$ Test Condition, Idle Mode, All Other Pins are Disconnected




### 24.6.5 Clock Signal Waveform for $I_{CC}$ Tests in Active and Idle Modes, $t_{CLCH} = t_{CHCL} = 5$ ns




### 24.6.6 $I_{CC}$ Test Condition, Power-down Mode, All Other Pins are Disconnected, $V_{CC}$ = 2V to 5.5V



## Table of Contents (Continued)