Migrating from AT89S52 to AT89LP52

New Features

- SIngle Clock Cycle per Byte Fetch with 20 MIPS Throughput at 20MHz Clock Frequency
- 12 Clock Instruction Compatibility Mode
- 4-level Interrupt Priority
- Enhanced Dual Data Pointers
- 256 Bytes of Flash Data Memory
- 256 Bytes of User Signature Array
- In-Application Programming of Program Memory
- Clock Out mode on Timer 0 and 1
- Shared Timer Prescaler
- Enhanced UART
  - Automatic Address Recognition
  - Framing Error Detection
  - SPI and TWI Master Emulation Modes
- Watchdog Timer Prescaler
- Software Reset
- Brown-out Detection (BOD) and Power-on Reset (POR) with Power-off Flag
- Selectable Polarity External Reset Pin
- Internal 1.8432 MHz Auxiliary Oscillator
- Selectable HIgh and Low Power Crystal Oscillator
- System Clock Divider
- Configurable I/O Port Modes per 8-bit Port
  - Quasi-bidirectional (80C51 Style)
  - Input-only (Tristate)
  - Push-pull CMOS Output
  - Open-drain
  - Enable Strong Pull-ups on Port 0
- Up to 36 Programmable I/O Lines
- 44-pad VQFN/MLF Package Option
- Wide Operating Voltage 2.4–5.5V

1. Introduction

The purpose of this application note is to help users convert existing designs from AT89S52/AT89LS52 to AT89LP52. AT89LP52 is meant as a drop-in replacement for AT89S52 or AT89LS52. The AT89LP52 is Atmel's first 8051 product to include both single-cycle (Fast) and classic 12-Clock (Compatibility) execution modes. Compatibility mode can ease the migration process for legacy designs with little or no software changes. For users looking for more performance, Fast mode and other new features are available to meet their needs. This application note describes the AT89LP52 memories, new and enhanced features, and SFR mapping and register differences. Some assembly code examples are provided. More detailed information can be found in the AT89LP52 datasheet.

8-bit Flash Microcontrollers

Application Note

3715A–MICRO–9/10

2. Flash Programming

The AT89LP52 microcontroller does not support high-voltage parallel programming. Users with third-party parallel programmers will need to update their programmer with the appropriate AT89LP52 serial driver if available. However, the In-System Programming (ISP) protocol is mostly compatible with the existing AT89S52. It is possible that some existing ISP programmers may be able to program the AT89LP52 with an AT89S52 driver, although it is recommended that drivers be updated to make use of the newer features on the AT89LP52. The flash programming differences are summarized in Table 2-1.

Table 2-1. Summary of Flash Programming Differences on AT89LP52

Feature	AT89S52	AT89LP52
High-Voltage Parallel Interface	YES	NO
3-wire Serial Interface	YES	YES
Reset Polarity	Active-High	Selectable
Byte Command Length	4 bytes	4 bytes
Page Command Length	258 bytes	67 bytes
Page Command Address Bytes	1	2
Page Size	256	128
Page Buffer Size	256	64
Device Signature Locations	0000H, 0100H, 0200H	0000H, 0001H, 0002H
Auto-Erase Commands	NO	YES
User Signature Array	0	256 bytes
Flash Data Memory	0	256 bytes
User Configuration Fuses	0	10

Users migrating from AT89S52 to AT89LP52 devices will notice the following changes to the ISP protocol:

1. All Byte mode commands are identical to AT89S52 with the exception that device signature bytes 1 and 2 are located at addresses 0001H and 0002H instead of 0100H and 0200H for the “Read Signature Bytes” command. Existing AT89S52 ISP drivers should be able to program the AT89LP52 in Byte mode provided they do not strictly check the signature.
2. The AT89LP52 has a half-page buffer of 64 bytes. Therefore Page mode commands accept only 64 bytes of data and require an additional address byte.
3. The AT89LP52 has a selectable polarity RST pin. Existing AT89S52 ISP drivers expect the reset to have active-high polarity. In most existing applications the $\overline{EA}$ pin is tied high, so dropping in an AT89LP52 will result in an active-high reset. See Section 5. “Reset” on page 10.

In addition to the above changes, the AT89LP52 also supports the following new features:

1. It is not necessary to provide a system clock during ISP; and the SCK frequency is independent of the system frequency when a clock is present.
2. It is not necessary to activate the Lock Bits in sequence as on AT89S52. The correct lock mode will be set; however, the lower order bits will not show as activated when reading back the Lock bits unless they were previously activated.

3. The AT89LP52 includes a User Signature Array for storing up to 256 bytes of user ID, revision, configuration or other nonvolatile information. This information can be read from the application code.
4. The AT89LP52 includes 256 bytes of Flash Data memory for storing additional nonvolatile data. This data can be read or written from the application code.
5. The AT89LP52 supports 10 User Configuration Fuses for configuring the default behavior of the device. See Section 2.1 for more information.
6. Each nonvolatile memory type includes a related Auto-Erase command that can erase and reprogram a page without erasing the entire memory first, provided the device is not write protected.
7. ISP can be disabled during warm resets by clearing the ISP Enable Fuse. When this fuse is disabled, programming is only available by asserting RST at power-up (cold reset).

2.1 Configuration Fuses

The AT89LP52 includes ten User Configuration Fuses for configuring the default behavior of the device. The fuses can only be changed by a device programmer that supports this feature. The default fuse settings are listed in Table 2-2. These settings were chosen to provide the greatest compatibility with the previous AT89S52 device. See the AT89LP52 datasheet for more detailed information.

For existing applications that use an external clock source instead of a crystal or resonator, it is recommended that the CS1 fuse (address 01H) be cleared to 00H to select the external clock configuration. This configuration provides better performance at higher frequencies than driving the on-chip crystal oscillator in open loop mode. Changes to this fuse will only take affect when the power is cycled off and on.

Compatibility/Fast Mode is selected by setting/clearing the Compatibility Mode Fuse at address 04H. Compatibility mode is enabled by default. Users wishing to migrate to Fast mode must clear this fuse.

Other features can be enabled/disabled by changing their respective fuses.

Table 2-2. Default Factory Fuse Settings

Address	Fuse Name	Default	Description
00H	Clock Source	FFH	High Speed Crystal Oscillator (XTAL)
01H		FFH	
02H	Start-up Time	FFH	16 ms Delay
03H		FFH	
04h	Compatibility Mode	FFH	CPU functions in 12-clock Compatibility mode
05h	ISP Enable	FFH	In-System Programming Enabled
06H	User Signature Programming	FFH	User Signature Programming Disabled
07H	Tristate Ports	00H	Ports default to Quasi-bidirectional mode
08H	In-Application Programming	FFH	In-Application Programming Disabled
09H	R1 Enable	FFH	Internal 5 MΩ resistor connected to XTAL1

3. Compatibility vs. Fast Mode

The AT89LP52 has a high performance, single-cycle CPU compatible with the 80C51 instruction set. For ease of migration the AT89LP52 can operate in a Compatibility execution mode. In Compatibility mode the AT89LP52 CPU uses the six-state machine cycle of the standard 8051 where instruction bytes are fetched every three system clock cycles. Execution times in this mode are identical to AT89S52 with instructions taking 1, 2 or 4 machine cycles. Sub-instruction level operations may not occur at exactly the same point within the instruction as in AT89S52; however, this should be transparent for most users. For greater performance the user can enable Fast mode by disabling the Compatibility fuse. In Fast mode the CPU fetches one code byte from memory every clock cycle instead of every three clock cycles. This greatly increases the throughput of the CPU. Each standard instruction executes in only 1 to 4 clock cycles. Any software delay loops or instruction-based timing operations may need to be retuned to achieve the desired results in Fast mode. Table 3-1 lists the major differences between Compatibility and Fast modes.

Table 3-1. Compatibility Mode versus Fast Mode Summary

Feature	Compatibility	Fast
Instruction Fetch in System Clocks (Internal Flash)	3	1
Instruction Fetch in System Clocks (External Memory)	3	3
Instruction Execution Time in System Clocks	6, 12, 18 or 24	1, 2, 3, 4 or 5
Default System Clock Divisor	2	1
Default Timer Prescaler Divisor	6	1
Pin Sampling Rate ($\overline{INT0}$, $\overline{INT1}$, T0, T1, T2, T2EX)	Prescaler Rate	System Clock
ALE Toggle Rate	1/3 System Clock	1/2 System Clock
Minimum RST input pulse in System Clocks	12	2
WDIDLE and DISRTO bit locations	AUXR	WDTCON

3.1 Migrating to Compatibility Mode

Users migrating legacy software from AT89S52 to AT89LP52 in Compatibility mode should not need to make any changes to their software provided that the application code does not activate any of the new register bits in existing registers listed in Section 8.2 “Modified Registers” on page 14 in such a manner as to change the behavior of the application. In most cases this should not be an issue.

Users willing to update their software can take advantage of any of the new peripheral features found on the AT89LP52 in Compatibility mode while maintaining standard 8051 instruction timing.

3.2 Migrating to Fast Mode

Users migrating legacy software from AT89S52 to AT89LP52 in Fast mode generally must make a small number of changes to their application code and possibly a hardware-related change to the clock frequency to maintain the same behavior. These changes are described in the following sections.

3.2.1 System Frequency and Clock Division

One of the first decisions that users migrating designs from AT89S52 to AT89LP52 in Fast mode must make is at what frequency to run the AT89LP52. By default the AT89LP52 in Fast mode is at least 6 times faster than the AT89S52, meaning it fetches bytes from memory in one-sixth the time. Therefore the maximum operating frequency may be lower in Fast mode than if the AT89LP52 was operating in Compatibility mode. Users must check that the desired frequency of the crystal or clock driver does not exceed the maximum specification for AT89LP52. If so, the frequency must be scaled down.

Furthermore, while the AT89S52 always divides the external clock by two to generate the internal system clock, by default the AT89LP52 in Fast mode does not. Therefore if the same external clock frequency is maintained from AT89S52, the AT89LP52 will operate at an internal-system clock that is twice as fast as on the AT89S52. Features that depend only on the internal system clock rate include the UART in modes 0 and 2, Timer 2 in Baud Rate or Clock Out modes and the external memory interface timing. To maintain the same timing using the UART in these modes or when the compare values for Timer 2 cannot be updated, the user can do either of the following:

- Cut the external clock frequency in half, or
- Enable divide-by-two for the internal system clock in the CLKREG SFR (See Table 8-5).

For timing on the external memory interfaces, see Section 4.1 “Program Memory” on page 7 and Section 4.2.1 “External Data Memory” on page 8.

Another difference in Fast mode is that Timer 0, 1, 2 and the Watchdog count at a rate equal to the internal system clock instead of every sixth system clock. The counting behavior is made compatible with AT89S52 by setting the timer prescaler in CLKREG to divide-by-6. If one of the above steps was also taken, this will maintain any time-out periods or timer-generated baud rates without any further changes. Otherwise the prescaler must be set to another appropriate value.

- Example 1: AT89S52 at external 24 MHz

  The AT89S52 runs at an internal frequency of 12 MHz after the default divider of 2. The user chooses to run the AT89LP52 at external 12 MHz with the default divider of 1. This maintains the same internal system clock speed, therefore the UART in modes 0 and 2 or Timer 2 in Baud Rate or Clock Out modes will still function as expected. Timers 0 and 1 and the Watchdog will count six times faster, so the timer prescaler is set to divide by 6 to preserve any time-out periods or baud rates:

```
MOV  CLKREG, #50H     ; TPS to div-by-6
```

- Example 2: AT89S52 at external 11.0592 MHz

  The user chooses to maintain the existing external frequency of 11.0592 Mhz. The user enables the clock divider to maintain the UART and Timer 2 in the modes mentioned above. The user also wants to maintain the baud rate generated by Timer 1 and sets the prescaler to divide-by-6:

```
MOV  CLKREG, #052H    ; TPS to div-by-6, CDV to div-by-2
```

  Another user also maintains the existing external frequency, but the system is not using the UART or Timer 2 in any of the modes mentioned above, so the user leaves the clock divider off. The user wants to maintain the baud rate generated by Timer 1 and sets the prescaler to divide-by-12:

```
MOV  CLKREG, #0B0H    ; TPS to div-by-12
```

One advantage of Fast mode is that it allows the AT89LP52 to provide the same instruction throughput as the AT89S52 at a much lower frequency, thereby reducing overall power consumption. Assuming that the clock divider is left disabled, AT89LP52 Fast mode is guaranteed to have at least 6 times the throughput of AT89S52 at the same frequency. Therefore the frequency of the AT89LP52 can be reduced up to 6 times as compared to the AT89S52 without reducing performance. However, doing so will affect the behavior of the peripherals.

- Example 3: AT89S52 at external 24 MHz

  The user chooses to run the AT89LP52 at external 4 MHz with the default divider of 1. The minimum instruction throughput is the same as AT89S52. The UART in modes 0 and 2 will be slower and Timer 2 in Baud Rate or Clock Out modes will need a shorter period to maintain the same rate. Timers 0 and 1 and the Watchdog will only count twice as fast as AT89S52, so the timer prescaler is set to divide by 2 to preserve any time-out periods or baud rates:

```
MOV  CLKREG, #10H     ; TPS to div-by-2
```

In conclusion, the external system frequency, system clock divider and timer prescaler are all available as parameters for maintaining timing compatibility between AT89LP52 in Fast mode and AT89S52.

3.2.2 Software Delays

Delays generated in the application code by executing instructions will have different lengths between Fast and Compatibility modes. These type of routines must be updated.

- Example: DJNZ Wait Loop

  A simple example is the DJNZ wait loop, which waits for a specified count:

```
        MOV  R7, #N
WAITN:  DJNZ R7, WAITN     ; loop N times
```

  On AT89S52 and AT89LP52 in Compatibility mode, the DJNZ instruction requires 12 system clocks. In Fast mode, the AT89LP52 executes DJNZ in only 3 clocks, thus N would need to be 4 times larger to generate the same delay at the same system frequency.

3.2.3 ALE

In Fast mode the ALE signal toggles at a rate of half the system clock with 50% duty cycles as compared to one third of the system clock at 33% duty cycle in Compatibility mode. Applications that make use of ALE at board level may need to be adjusted accordingly. For applications that do not use ALE, or use it only for external memory, it is recommended that ALE be disabled by setting the DISALE bit in AUXR for reduced power consumption.

3.2.4 Timers

If the same timer rate as AT89S52 is not maintained as detailed above in Section 3.2.1, the timer reload or compare values will need to be scaled accordingly. See Section 9. on page 15

3.2.5 Watchdog

AT89S52 software that activates the WDIDLE or DISRTO bits in AUXR must be updated for Fast mode on the AT89LP52 as these bits are relocated from AUXR to WDTCON. If the same timer rate as AT89S52 is not maintained as detailed above in Section 3.2.1, the Watchdog may

need to either be reset more often or have its period lengthened with the watchdog prescaler settings in WDTCON. See Section 10. on page 16.

4. Memory Access

The AT89LP52 supports the same memory spaces found on the AT89S52. The program memory has support for 64K bytes of directly addressable application code, with 8K bytes of on-chip Flash program memory and support for up to 56K bytes of external program memory using the standard 80C51 interface. The data memory has 256 bytes of internal RAM and 128 bytes of Special Function Register I/O space, with support for up to 64K bytes of external data memory also using the standard 80C51 interface. Applications that use these memory spaces should not need to be updated in Compatibility mode unless they want to take advantage of the additional features on the AT89LP52.

4.1 Program Memory

The AT89LP52 supports both internal and external program memory. Applications that only use external program memory, by tying the $\overline{EA}$ pin low, are not supported on the AT89LP52. External program memory is only available at addresses 2000H–FFFFH. In Compatibility mode there is no difference when executing from internal versus external program memory. In Fast mode, two wait states must be inserted for every external fetch. Therefore instructions executed from external memory in Fast mode are exactly three times longer than instructions executed from internal memory. This may play an important role when selecting the operating frequency as detailed in Section 3.2.1 on page 5. Furthermore, by default the Fast mode external memory interface will fetch at twice the rate of the AT89S52 unless the clock is scaled by two. To maintain the same external memory timing the user can do either of the following:

- Cut the external frequency in half, or
- Enable divide-by-two for the system clock in the CLKREG SFR.

4.1.1 In-Application Programming

Read-only data access to program memory is supported exactly as in AT89S52 with the MOVC instructions. An additional feature on the AT89LP52 is write access through the In-Application Programming (IAP) interface in the MEMCON register. IAP allows reprogramming of the program memory from the application code by mapping the internal program memory to addresses 0000H–1FFFH of the external data memory space for access by MOVX. The CPU is placed in an idle state while any writes to program memory are in progress. See the AT89LP52 datasheet for more information. The IAP User Fuse must be programmed to activate the IAP feature. Applications that do not use IAP should keep the IAP Fuse disabled.

- Example: Write a simple pattern in code memory

```
      MOV  MEMCON, #0B0H  ; set IAP LDPG MWEN
      MOV  DPTR, #1000H   ; Page to program
      MOV  R1, #0         ; pattern start
      MOV  R0, #63        ; length-1
LOAD:
      MOV  A, R1
      MOVX @DPTR, A       ; load byte to buffer
      INC  DPTR           ; next address
      INC  R1             ; incrementing pattern
      DJNZ R0, LOAD
      ANL  MEMCON, #~20H  ; clear LDPG
      MOV  A, R1
```

```
MOVX @DPTR, A       ; load last byte and start write
MOV  MEMCON, #0     ; clear IAP and MWEN
```

4.1.2 Signature Array

The AT89LP52 makes both the 128-byte Atmel Signature and 256-byte User Signature readable from the application code by setting the SIGEN bit in AUXR1. This bit allows the MOVC instructions to access the signatures instead of the program memory. In addition the User Signature is writable through the IAP interface. The Atmel Signature is mapped to addresses 0000H–007FH and the User signature is mapped to addresses 0100H–01FFH. SIGEN must be cleared to access normal program memory again.

- Example: Read Device ID from Atmel Signature

```
ORL  AUXR1, #8     ; set SIGEN
MOV  DPTR, #0      ; Atmel base
CLR  A
MOVC A, @A+DPTR    ; read signature 0
MOV  A, #1
MOVC A, @A+DPTR    ; read signature 1
MOV  A, #2
MOVC A, @A+DPTR    ; read signature 2
ANL  AUXR1, #~8    ; clear SIGEN
```

- Example: Read User Signature

```
ORL  AUXR1, #8     ; set SIGEN
MOV  DPTR, #100H   ; User base
CLR  A
MOVC A, @A+DPTR    ; read signature 0
MOV  A, #1
MOVC A, @A+DPTR    ; read signature 1
MOV  A, #2
MOVC A, @A+DPTR    ; read signature 2
ANL  AUXR1, #~8    ; clear SIGEN
```

4.2 Data Memory

The 256 bytes of internal RAM and 128 bytes of Special Function Register I/O space are accessed in the normal manner. See Section 8. for a list of new or modified registers on the AT89LP52. The external data memory interface may require some tuning for operation in Fast mode. On-chip Flash data memory is a new feature available on the AT89LP52.

4.2.1 External Data Memory

The external data memory interface of the AT89LP52 is accessed in the same manner as on AT89S52. Some portions of the external memory address space can be redirected toward on-chip memories. By default this redirection is disabled so legacy code can access the entire address range. The EXRAM bit in AUXR can also be used to override any other settings and force access to external memory.

In Compatibility mode, MOVX timing for external memory access is identical to AT89S52. In Fast mode the MOVX instruction is faster, meaning a shorter access time to external memory. If an existing external memory can support the faster access, then nothing must be done when migrating to Fast mode. If not, the timing of MOVX must be lengthened by changing the system clock and/or wait state settings. The wait states are controlled by the WS bits in AUXR as listed in Table 4-1. The wait states can increase the length of the strobe and also the setup between ALE and the strobe. Note that each additional wait state adds one clock cycle to the instruction execution time.

Table 4-2 shows some possible configurations for MOVX timing in Fast mode. The only way to achieve identical timing with Compatibility mode is to cut the system frequency in third and add one wait state. In other configurations the user must select the settings that are nearest enough to meet the specifications of the system.

Table 4-1. Wait State Effects on MOVX Timing (in System Clocks)

WS1	WS0	$\overline{RD}$ or $\overline{WR}$ Strobe Width		ALE low to $\overline{RD}$ or $\overline{WR}$ low Setup		Access Time (setup + width)	
		Compatibility	Fast	Compatibility	Fast	Compatibility	Fast
0	0	3	1	1.5	1	4.5	2
0	1	15	2	1.5	1	16.5	3
1	0	3	2	1.5	2	4.5	4
1	1	15	3	1.5	2	16.5	5

Table 4-2. Configuration Settings for MOVX Timing in Fast Mode

Description	Configuration Settings		Access Time
	CDV	WS	
AT89S52	–	–	9 x T_{OSC}
AT89LP52 Compatibility Mode Default	1	0	9 x T_{OSC}
AT89LP52 Fast Mode Default	0	0	2 x T_{OSC}
Nearest to AT89S52 in Fast Mode at Same Frequency	0	3	5 x T_{OSC}
	1	2	8 x T_{OSC}
		3	10 x T_{OSC}
	2	0	8 x T_{OSC}
Nearest to AT89S52 in Fast Mode at 1/2 Frequency	0	2	4 x (2 x T_{OSC})
		3	5 x (2 x T_{OSC})
	1	0	4 x (2 x T_{OSC})
Compatible to AT89S52 in Fast Mode at 1/3 Frequency	0	1	3 x (3 x T_{OSC})
Nearest to AT89S52 in Fast Mode at 1/6 Frequency	0	0	2 x (6 x T_{OSC})

Note: T_{OSC} is the external oscillator period of the AT89S52

4.2.2 Flash Data Memory

In addition to the 64K bytes of external data memory, addresses 0000H–00FFH of the external data memory space are implemented on chip as 256 bytes of nonvolatile Flash data memory. These bytes are only accessible when the DMEN bit is set in MEMCON. The memory can be read using MOVX just like external data memory. Writing Flash data is similar to IAP for the program memory. The CPU is placed in an idle state while any writes to flash data memory are in progress. See the AT89LP52 datasheet or the “AT89LP Flash Data Memory” application note for more information.

- Example: Write a simple pattern in flash data memory

```
      MOV  MEMCON, #038H  ; set LDPG MWEN DMEN
      MOV  DPTR, #080H    ; Page to program
      MOV  R1, #0         ; pattern start
      MOV  R0, #63        ; length-1
LOAD:
      MOV  A, R1
      MOVX @DPTR, A       ; load byte to buffer
      INC  DPTR           ; next address
      INC  R1             ; incrementing pattern
      DJNZ R0, LOAD
      ANL  MEMCON, #~20H  ; clear LDPG
      MOV  A, R1
      MOVX @DPTR, A       ; load last byte and start write
      MOV  MEMCON, #0     ; clear DMEN and MWEN
```

5. Reset

The AT89LP52 has a user-selectable external reset pin. To support this feature the former External Access pin ($\overline{EA}$) of the AT89S52 is replaced by the Polarity pin (POL). When this pin is at VCC, the RST pin is active-HIGH with a pull-down resistor; and when this pin is at GND, the RST pin is active-LOW with a pull-up resistor. As a consequence the external access feature is NOT supported on the AT89LP52; however, external execution is still allowed for addresses 2000H–FFFFH. The majority of legacy AT89S52 applications have $\overline{EA}$ tied high for internal execution. Dropping an AT89LP52 into these applications will result in POL tied high for an AT89S52- compatible active-HIGH reset. If an AT89S52 application has $\overline{EA}$ tied low, the user must either modify the board to connect POL to high or disconnect the RST pin from any board-level signals.

The POL pin must be driven high or low at all times. It does not have an internal pull-up or pull-down. The level of the POL pin is sampled during power-up. It is not possible to change the polarity once the device is operational. An active-low reset is recommended for all new applications.

If In-System Programming is disabled, the only way to program the AT89LP52 is if RST is active during power-up. ISP is always enabled at power-up and will remain active until the first deactivation of RST. Users wishing to further program the device in such a state must have a means of connecting RST to VCC or GND at power-up, depending on the polarity.

5.1 BOD and POR

The AT89LP52 includes an on-chip Brown-Out Detector (BOD) and Power-On Reset (POR). The POR has a trip point of ~1.4V and is always on. The BOD has a trip point of ~2.0V and is shutdown during powerdown. Both the BOD and the POR can reset the device. As a consequence, external RC circuits are usually not required on the RST pin for proper startup, and can be removed to reduce system cost. However, if a trigger level greater than 2.0V is necessary, an external BOD circuit connected to RST is required. This may be the case for some high speed 5V applications that may execute incorrectly at lower voltages.

6. System Clock

The system clock source of the AT89LP52 is selectable between the crystal oscillator, an externally driven clock and an internal 1.8432 MHz auxiliary oscillator. In addition the crystal oscillator can operate in either high-power or low-power mode and optionally have an on-chip 5 MΩ resistor connected between XTAL1 and GND for improved startup. The clock source and options are controlled by the User Fuses. Section 2.1 “Configuration Fuses” on page 3.

The A89LP52 is factory-configured to use the crystal oscillator in high-power mode. Users wanting to switch to low-power mode may need to remove or reduce capacitors on the XTAL1 and XTAL2 pins. Applications that use an external clock source should select the external clock configuration instead of driving the oscillator in open-loop mode. In external clock mode, the XTAL2 pin is available as a general purpose I/O, P4.7.

The AT89LP52 includes an on-chip 1.8432 MHz auxiliary RC oscillator that is used for some internal functions. It is also available as a system clock source. The oscillator has accuracy of ±2% to enable UART communications. In fact, 1.8432 MHz is the UART frequency of 11.0592 MHz divided by six, which allows the AT89LP52 running at 1.8432 MHz in fast mode to provide the same or better throughput as the AT89S52 at 11.0592 MHz with much lower power consumption. In internal oscillator mode, XTAL1 and XTAL2 pins are available as a general purpose I/Os P4.6 and P4.7, respectively.

6.1 Clock Divider

The System Clock Divider scales the internal system clock versus the oscillator clock source. In Compatibility mode the system clock frequency is divided by 2 from the externally supplied XTAL1 frequency by default for compatibility with standard 8051s (12 clocks per machine cycle). The divide-by-2 can be disabled to operate in X2 mode (6 clocks per machine cycle) or the clock may be further divided to reduce the operating frequency. If the divide-by-2 is disabled, make sure that the new system clock frequency is still within the valid operating range. In Fast mode the clock divider defaults to divide by 1. See “System Frequency and Clock Division” on page 5 for more information.

$$f_{SYS} = \frac{f_{OSC}}{2^{CDV}}$$

- Example: Disable Divide-by-2 (Enable X2 Mode) in Compatibly Mode

```
ANL  CLKREG, #0F0H    ; clear CDV2-0
```

6.2 Timer Prescaler

A common prescaler is available to divide the time base for Timer 0, Timer 1, Timer 2 and the Watchdog. The TPS_{3-0} bits in the CLKREG SFR control the prescaler. In Compatibility mode TPS_{3-0} defaults to 0101B, which causes the timers to count once every machine cycle. The counting rate can be adjusted linearly from the system clock rate to 1/16 of the system clock rate by changing TPS_{3-0}. In Fast mode TPS_{3-0} defaults to 0000B, or the system clock rate. TPS does not affect Timer 2 in Clock Out or Baud Rate modes.

$$f_{TIMER} = \frac{f_{SYS}}{TPS + 1}$$

- Example: Scale Timers by 6

```
MOV  CLKREG, #050H    ; set TPS3-0 = 5; CDV2-0 = 0
```

7. I/O Ports

The P0, P1, P2 and P3 I/O ports of the AT89LP52 may be configured in four different modes:

- Quasi-bidirectional
  Standard 80C51 port with strong, medium and weak pull-ups good for general bidirectional use.
- Input Only
  The port is tristated for high impedance inputs. This mode reduces power for low-level inputs by removing the pull-ups; however, the input should not be left floating or higher power consumption may occur. A port in this mode cannot have any pin acting as an output.
- Push-Pull Output
  The port provides a full CMOS output driver with larger current sourcing capabilities than quasi-bidirectional. A port in this mode cannot have any pin acting as an input.
- Open-Drain
  The port pull-ups are disabled. External pull-ups are required to use the pins to output a high level. Open-Drain is most useful for wired-AND type buses.

The default port settings depend on the Tristate-Port User Fuse. When the fuse is set all the I/O ports revert to input-only (tristated) mode at power-up or reset. When the fuse is not active, ports P1, P2 and P3 start in quasi-bidirectional mode and P0 starts in open-drain mode. P4 always operates in quasi-bidirectional mode. P0 can be configured to have internal pull-ups by placing it in quasi-bidirectional or output modes. This can reduce system cost by removing the need for external pull-ups on Port 0.

- Example: Enable Pull-ups on Port 0

```
MOV  PMOD, #00H       ; all ports quasi-bidirectional
```

Port configuration is set by the PMOD register and is independently configurable for P0, P1, P2 and P3. Each configuration applies to the entire 8-bit port. Ports using bidirectional signals should not be set to Input Only or Push-Pull Output modes as these modes are unidirectional.

Table 7-1. Configuration Modes for Port x

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

P0 and P2 do not require configuration to use the external memory interface. Addresses will automatically be output in push-pull mode and P0 is automatically tristated during instruction or data read. ALE and $\overline{\text{PSEN}}$ always operate in quasi-bidirectional mode. The $\overline{\text{RD}}$ and $\overline{\text{WR}}$ strobes; however, will be configured with P3. To use external data memory P3 must be configured accordingly.

The AT89LP52 provides additional I/Os, pins P4.4–P4.7, that replace the normally dedicated ALE, $\overline{PSEN}$, XTAL1 and XTAL2 pins of the AT89S52. These pins can be used as additional I/Os depending on the configuration of the clock and external memory as listed in Table 7-2.

Table 7-2. Configurations for Additional I/Os

Pin	Function	Configuration Required for General I/O Use
P4.4	ALE	No external memory and DISALE = 1
P4.5	$\overline{PSEN}$	No external program memory
P4.6	XTAL1	Internal 1.8432 MHz oscillator selected
P4.7	XTAL2	External clock or internal 1.8432 MHz oscillator selected

8. Special Function Registers

This section lists the Special Function Registers (SFRs) that are new or modified in AT89LP52 from those in AT89S52.

8.1 SFR Mapping

The highlighted SFR locations are new registers for the AT89LP52 device.

Table 8-1. SFR Mapping in AT89LP52

0F8H									0FFH
0F0H	B								0F7H
0E8H									0EFH
0E0H	ACC								0E7H
0D8H									0DFH
0D0H	PSW								0D7H
0C8H	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2			0CFH
0C0H	P4	PMOD							0C7H
0B8H	IP	SADEN							0BFH
0B0H	P3							IPH	0B7H
0A8H	IE	SADDR							0AFH
0A0H			AUXR1				WDTRST	WDTCON	0A7H
98H	SCON	SBUF							9FH
90H	P1	TCONB					MEMCON		97H
88H	TCON	TMOD	TL0	TL1	TH0	TH1	AUXR	CLKREG	8FH
80H		SP	DP0L	DP0H	DP1L	DP1H		PCON	87H

8.2 Modified Registers

The following registers are modified in their bit definitions from AT89S52. For more detailed information see the AT89LP52 datasheet.

Table 8-2. AUXR Register Bits

AUXR (8EH)	7	6	5	4	3	2	1	0
AT89S52	–	–	–	WDIDLE	DISRTO	–	–	DISALE
AT89LP52 (Compatibility Mode)	–	–	–	WDIDLE	DISRTO	WS0	EXRAM	DISALE
AT89LP52 (Fast Mode)	–	–	–	(1)	WS1(1)	WS0	EXRAM	DISALE

Note: 1. In Fast Mode the WDIDLE and DISRTO bits are located in the WDTCON register. In Compatibility Mode WS1 = 0.

Table 8-3. AUXR1 Register Bits

AUXR1 (A2H)	7	6	5	4	3	2	1	0
AT89S52	–	–	–	–	–	–	–	DPS
AT89LP52	DPU1	DPU0	DPD1	DPD0	SIGEN	0	–	DPS

Table 8-4. PCON Register Bits

PCON (87H)	7	6	5	4	3	2	1	0
AT89S52	SMOD	–	–	–	GF1	GF0	PD	IDL
AT89LP52	SMOD1	SMOD0	PWDEX	POF	GF1	GF0	PD	IDL

8.3 New Registers

The following registers are additional registers not found in AT89S52. For more detailed information see the AT89LP52 datasheet

Table 8-5. CLKREG Register Bits

CLKREG (8FH)	7	6	5	4	3	2	1	0
Reset = (1)	TPS3	TPS2	TPS1	TPS0	CDV2	CDV1	CDV0	–

Note: 1. The reset value of CLKREG is 0000000B in Fast mode and 01010010B in Compatibility mode.

Table 8-6. IPH Register Bits

IPH (B7H)	7	6	5	4	3	2	1	0
Reset = XX00 0000	–	–	PT2H	PSH	PT1H	PX1H	PT0H	PX0H

Table 8-7. MEMCON Register Bits

MEMCON (96H)	7	6	5	4	3	2	1	0
Reset = 0000 0XXX	IAP	AERS	LDPG	MWEN	DMEN	ERR	$\overline{\text{BUSY}}$	$\overline{\text{WRTINH}}$

Table 8-8. P4 Register Bits

P4 (C0H)	7	6	5	4	3	2	1	0
Reset = 1111 XXXX	P4.7	P4.6	P4.5	P4.4	–	–	–	–

Table 8-9. PMOD Register Bits

PMOD (C1H)	7	6	5	4	3	2	1	0
Reset = [2]	P3M1	P3M0	P2M1	P2M0	P1M1	P1M0	P0M1	P0M0

2. The reset value of PMOD is 0101 0101B when Tristate-Port Fuse is enabled and 0000 0011B when disabled

Table 8-10. SADDR Register Bits

SADDR (A9H)	7	6	5	4	3	2	1	0
Reset = 0000 0000	SADDR.7	SADDR.6	SADDR.5	SADDR.4	SADDR.3	SADDR.2	SADDR.1	SADDR.0

Table 8-11. SADEN Register Bits

SADEN (B9H)	7	6	5	4	3	2	1	0
Reset = 0000 0000	SADEN.7	SADEN.6	SADEN.5	SADEN.4	SADEN.3	SADEN.2	SADEN.1	SADEN.0

Table 8-12. TCONB Register Bits

TCONB (91H)	7	6	5	4	3	2	1	0
Reset = 000X XXXX	T1OE	T0OE	SPEN	–	–	–	–	–

Table 8-13. WDTCON Register Bits

WDTCON (A6H)	7	6	5	4	3	2	1	0
Reset = 0000 0XX0	PS2	PS1	PS0	WDIDLE	DISRTO[3]	SWRST	WDTOVF	WDTEN

3. WDTCON.4 and WDTCON.3 function as WDIDLE and DISRTO only in Fast mode. In Compatibility mode these bits are in AUXR.

9. Timers

The timer counting rate of Timer 0, Timer 1 and Timer 2 in Auto-Reload or Capture modes is controlled by the timer prescaler. In Compatibility mode this defaults to one sixth of the system clock, the same rate as AT89S52. No software changes to the timers should be required when migrating legacy software to Compatibility Mode. In Fast mode the timers count at the system clock rate. See “System Frequency and Clock Division” on page 5. The user can control the behavior of the timers with the timer prescaler (TPS) and clock divider (CDV) settings in CLKREG. Timer 2 in Baud Rate or Clock-Out modes is dependent only on the system clock. The user can control the behavior of Timer 2 in these modes only with the clock divider settings. Table 9-1 lists several possible configurations for the Timers.

Table 9-1. Configuration Settings for Compatible Timer Rates

Timer 0 or Timer 1 Timer 2 in Auto-Reload or Capture Modes	Configuration Settings		Timer Rate
	CDV	TPS	
AT89S52	–	–	f_{OSC} / 12
AT89LP52 Compatibility Mode Default	1	5	f_{OSC} / 12
AT89LP52 Fast Mode Default	0	0	f_{OSC} / 1
Compatible to AT89S52 in Fast Mode at Same Frequency	0	11	f_{S52} / 12
	1	5	
Compatible to AT89S52 in Fast Mode at Half Frequency	0	5	(f_{S52}÷ 2) / 6
Compatible to AT89S52 in Fast Mode at One-Sixth Frequency	0	1	(f_{S52}÷ 6) / 2
Timer 2 in Baud Rate or Clock-Out Modes	**CDV**	**TPS**	**Timer Rate**
AT89S52	–	–	f_{OSC} / 2
AT89LP52 Compatibility Mode Default	1	5	f_{OSC} / 2
AT89LP52 Fast Mode Default	0	0	f_{OSC} / 1
Compatible to AT89S52 in Fast Mode at Same Frequency	1	–	f_{S52} / 2
Compatible to AT89S52 in Fast Mode at Half Frequency	0	–	(f_{S52}÷ 2) / 1

Note: f_{S52} is the XTAL1 frequency of the AT89S52, i.e f_{S52}÷ 2 means set f_{OSC} of the AT89LP52 to f_{S52}/2.

Full timer rate compatibility with AT89S52 for Timer 0 and 1, and Timer 2 in Baud Rate or Clock-Out modes, can only be maintained in AT89LP52 Fast mode in the following two cases:

- If AT89LP52 operates at the same frequency as AT89S52, then CDV = 1 and TPS = 5.
- If AT89LP52 operates at half the frequency of AT89S52, then CDV = 0 and TPS = 5.

Otherwise the reload values may need to be scaled accordingly. For example, if the AT89LP52 operates at the same frequency as AT89S52 but CDV = 0, the reload value for Timer 2 in Baud Rate or Clock-Out mode must be twice as long to overflow at the same rate.

In Compatibility mode the sampling of the external Timer/Counter pins: T0, T1, T2 and T2EX; and the external interrupt pins, $\overline{INT0}$ and $\overline{INT1}$, is also controlled by the prescaler. In Fast mode these pins are always sampled at the system clock rate.

An additional feature on AT89LP52 is that both Timer 0 and Timer 1 can toggle their respective counter pins, T0 and T1, when they overflow by setting the output enable bits in TCONB. This is most useful in Mode 2 to generate a signal of varying frequency similar to Timer 2 in Clock-Out mode.

10. Watchdog

The AT89LP52 adds a new register, WDTCON, for controlling the Watchdog. This register holds the prescaler settings and watchdog flags. The flags allow the user to determine if the watchdog is currently running, if the watchdog previously overflowed, and if a software reset occurred. In Fast Mode the WDIDLE and DISRTO bits are also located in WDTCON and not in AUXR. Code migrated from AT89S52 to AT89LP52 Fast mode must be updated to use these bits. Other than these bits, no software changes to the watchdog should be required when migrating legacy software to Compatibility Mode.

The Watchdog Timer on the AT89LP52 includes a 7-bit logarithmic prescaler for longer time-out periods than the AT89S52. The counting rate of the watchdog is controlled by the timer prescaler. In Compatibility mode this defaults to one sixth of the system clock, the same rate as AT89S52. In Fast mode the watchdog counts at the system clock rate. See “System Frequency and Clock Division” on page 5. The user can control the behavior of the Watchdog with the timer prescaler (TPS) and clock divider (CDV) settings in CLKREG or the watchdog prescaler (PSC) settings in WDTCON. Table 10-1 lists several possible configurations for the Watchdog.

Table 10-1. Configuration Settings for Compatible Time-out Periods

Description	Configuration Settings			Time-out
	CDV	TPS	PSC	
AT89S52	–	–	–	196608 x T_{OSC}
AT89LP52 Compatibility Mode Default	1	5	0	196608 x T_{OSC}
AT89LP52 Fast Mode Default	0	0	0	16384 x T_{OSC}
Compatible to AT89S52 in Fast Mode at Same Frequency	0	11	0	196608 x T_{OSC}
	1	5	0	
Compatible to AT89S52 in Fast Mode at Half Frequency	0	5	0	98304 x (2 x T_{OSC})
Compatible to AT89S52 in Fast Mode at One-Sixth Frequency	0	1	0	32768 x (6 x T_{OSC})
Nearest to AT89S52 in Fast Mode at Same Frequency without Timer Prescaler	0	0	3	131072 x T_{OSC}
	0	0	4	262144 x T_{OSC}

10.1 Software Reset

Software reset is a new feature of the AT89LP52 that immediately resets the device. Software reset is triggered by writing the sequence 5AH/A5H to WDTRST.

- Example: Software Reset

```
MOV  WDTRST, #05AH
MOV  WDTRST, #0A5H
```

11. Serial Port

The UART in Compatibility mode behaves identically to AT89S52 by default. The timer prescaler increases the range of achievable baud rates when using Timer 1 to generate the baud rate in UART Modes 1 or 3, including an increase in the maximum baud rate available in Compatibility mode. Additional features include automatic address recognition and framing error detection. The shift register mode (Mode 0) has been enhanced with more control of the polarity, phase and frequency of the clock and full-duplex operation. This allows emulation of master serial peripheral (SPI) and two-wire (TWI) interfaces.

11.1 UART under Fast Mode

The baud rate of the UART in Modes 0 or 2 can only be maintained in AT89LP52 Fast mode in the following two cases:

- If AT89LP52 operates at the same frequency as AT89S52 and CDV = 1.
- If AT89LP52 operates at half the frequency of AT89S52 and CDV = 0.

When the baud rate is provided by Timer 2 for Modes 1 or 3, the same baud rate is also maintained by the above cases without updating the reload value. In other cases the Timer 2 reload value would need to be updated. When Timer 1 provides the baud rate there are several cases where the rate can be maintained by using the timer prescaler. Other cases would require an update of the Timer 1 reload value. Table 11-1 lists several possible configurations for the baud rate generation in Fast mode that do not require updates to the timer values.

Table 11-1. Configuration Settings for Compatible Baud Rates in Fast Mode

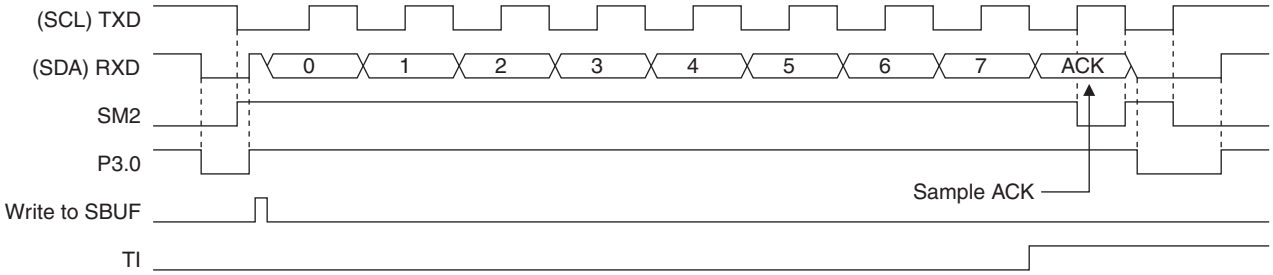
Mode	Baud Source	System Frequency	Configuration Settings	
			CDV	TPS
0 or 2	–	$f_{LP52} = f_{S52}$	1	–
		$f_{LP52} = f_{S52} / 2$	0	–
1 or 3	Timer 1	$f_{LP52} = f_{S52}$	0	11
			1	5
		$f_{LP52} = f_{S52} / 2$	0	5
		$f_{LP52} = f_{S52} / 3$	0	3
		$f_{LP52} = f_{S52} / 4$	0	2
		$f_{LP52} = f_{S52} / 6$	0	1
	Timer 2	$f_{LP52} = f_{S52}$	1	–
		$f_{LP52} = f_{S52} / 2$	0	–

- Example: AT89S52 at external 11.0592 MHz with Baud Rate of 19.2Kbps
  - A user chooses to maintain the existing frequency of 11.0592 Mhz. The user enables the clock divider for divide-by-2 (CDV = 1). If Timer 2 is the baud generator, RCAP2H/RCAP2L can stay equal to FFEEH. If Timer 1 is the baud generator in Mode 2, setting TPS = 5 will allow TH1 to stay at FDH and SMOD1 to stay 1. If TPS is left at 0, then TH1 must be updated to EEH for SMOD1 = 1.
  - Another user chooses to maintain the existing frequency of 11.0592 Mhz, but does not enable the clock divider (CDV = 0). If Timer 2 is the baud generator, RCAP2H/RCAP2L must be updated to FFDCH. If Timer 1 is the baud generator in Mode 2, setting TPS = 11 will allow TH1 to stay at FDH and SMOD1 to stay 1. If TPS is left at 0, then TH1 must be updated to DCH for SMOD1 = 1.
  - A third user chooses to scale the external frequency to 1.8432 Mhz and does not enable the clock divider (CDV = 0). If Timer 2 is the baud generator, RCAP2H/RCAP2L must be updated to FFFAH. If Timer 1 is the baud generator in Mode 2, setting TPS = 1 will allow TH1 to stay at FDH and SMOD1 to stay 1. If TPS is left at 0, then TH1 must be updated to FAH for SMOD1 = 1.

11.2 Two-Wire Interface Emulation

The UART Mode 0 on the AT89LP52 can act as a hardware accelerator for software emulation of a serial Two-Wire Interface (TWI) in master mode. An example of Mode 0 emulating a TWI master device is shown in Figure 11-1. The TXD pin connects to the SCL line of the TWI bus and the RXD pin connects to the SDA line.

Figure 11-1. UART Mode 0 TWI Emulation (SMOD1 = 1)



In this example, the start, stop, and acknowledge are handled in software while the byte transmission is done in hardware. Falling/rising edges on TXD/SCL are created by setting/clearing SM2. Rising/falling edges on RXD/SDA are forced by setting/clearing the P3.0 register bit. SM2, SMOD1 and P3.0 must be 1 while the byte is being transferred. The following code examples demonstrate how to generate start and stop conditions and send and receive acknowledges:

- Example: TWI Start, Stop and Acknowledge Routines

```
STA: CLR  SM2
     CLR  P3.0
     SETB SM2
     CLR  P3.0
     RET
STO: SETB SM2
     CLR  P3.0
     CLR  SM2
     SETB P3.0
     RET
TX_ACK:
     CLR  P3.0
     CLR  SM2
     SETB SM2
     SETB P3.0
     RET
TX_NACK:
     SETB P3.0
     CLR  SM2
     SETB SM2
     SETB P3.0
     RET
RX_ACK:
     CLR  SM2
     MOV  C, P3.0     ; sample ACK
     SETB SM2
     RET
```

Mode 0 transfers data LSB first whereas TWI is generally MSB first, requiring a bit reversal of the transferred data bytes. The following code example reverses the bits in the accumulator:

- Example: Bit Reversal Routine

```
REVERSE:
     MOV  R7, #8
REVR_LOOP:
     RLC  A             ; C << msb (ACC)
     XCH  A, R6
     RRC  A             ; msb (ACC) >> R6
     XCH  A, R6
     DJNZ R7, REVR_LOOP
     XCH  A, R6
     RET
```

The following sample routines show how to send and receive a byte over TWI. Appropriate delays may need to be added, especially in the start, stop and acknowledge routines. Setting TB8 in SCON will allow the baud rate to be controlled by Timer 1. TWI emulation only works when the AT89LP52 is the only master on the bus.

- Example: Simple TWI Transmit Byte Routine

```
TRANSMIT:               ; Address in ACC, data in B
     ORL    PCON, #80H  ; set SMOD1
     LCALL  STA         ; send Start
     LCALL  REVERSE
     MOV    SBUF, A     ; send address
     JNB    TI, $
     CLR    TI
     LCALL  RX_ACK      ; get ACK
     JC     TX_DONE
     MOV    A, B
     LCALL  REVERSE
     MOV    SBUF, A     ; send data
     JNB    TI, $
     CLR    TI
     LCALL  RX_ACK
TX_DONE:
     LCALL  STO
     RET                ; ACK state in C
```

- Example: Simple TWI Receive Byte Routine

```
RECEIVE:                ; Address in ACC
     ORL    PCON, #80H  ; set SMOD1
     LCALL  STA         ; send Start
     LCALL  REVERSE
     MOV    SBUF, A     ; send address
     JNB    TI, $
     CLR    TI
     LCALL  RX_ACK      ; get ACK
     JC     RX_DONE
     SETB   REN
     JNB    RI, $
     CLR    REN
     CLR    RI
     MOV    A, SBUF
     LCALL  REVERSE
     LCALL  TX_NACK
     CLR    C
RX_DONE:
     LCALL  STO
     RET                ; ACK state in C, Data in ACC
```

11.3 Serial Peripheral Interface Emulation

The UART Mode 0 on the AT89LP52 can act as a full-duplex Serial Peripheral Interface (SPI) in master mode. SPI Mode is enabled by setting the SPEN bit in TCONB. SPEN causes the serial clock to output on P1.7. Serial data is input on P1.6 and output on P1.5. The UART can emulate the SPI modes listed in Table 11-2. The bit rate is the system clock divided-by-2 when SMOD1 = 1 and the system clock divided-by-4 when SMOD1 = 0. For other bit rates, set TB8 to use Timer 1 as the bit rate generator.

In SPI mode the transmit line (P1.5) is always active. Transfers only start when data is written to SBUF. Unlike normal Mode 0, setting REN will not start a transfer. REN only determines if the receive line (P1.6) is sampled and shifted into the input buffer and RI is set.

Table 11-2. Mode 0 Clock and Data Modes for SPI Emulation

SM2	SMOD1	Mode	Clock Idle	Data Changes	Data Sampled
0	0	(1,1)	High	While clock is high	Positive edge of clock
0	1	(1,1)	High	Negative edge of clock	Positive edge of clock
1	0	(0,1)	Low	While clock is low	Negative edge of clock
1	1	(0,0)	Low	Negative edge of clock	Positive edge of clock

SPI mode uses different I/Os from Two-Wire mode and the standard UART. It is possible to time share the UART hardware between SPI slaves connected on P1 and UART or TWI devices on P3 with the caveat that any asynchronous UART receptions on the RXD pin will be ignored while the UART is in Mode 0. If the UART receiver is not used, or the system can guarantee that reception will not occur during a certain mode, the AT89LP52 can act as if it had two communication interfaces.

12. Additional Features

In addition to the features discussed above, the AT89LP52 includes the following new features.

12.1 Interrupts

With the addition of the IPH register, the AT89LP52 provides four levels of interrupt priority for greater flexibility in handling multiple interrupts. IPH is identical to the IP register and holds the higher order priority bit. Also, Fast mode allows for faster interrupt response due to the shorter instruction execution times at a given frequency.

12.2 Dual Data Pointers

The AT89LP52 adds instruction level support for dual data pointers. The five new instructions listed in Table 12-1 allow access to the currently non-selected data pointer without needing to change the DPS bit in AUXR1. For single data pointer routines these instructions are not necessary, other than for convenience, as toggling DPS and using the standard instructions is more efficient. For routines requiring two data pointers, they can improve the performance by removing the need to toggle DPS repeatedly.

Table 12-1. New Dual Data Pointer Instructions

Mnemonic	Opcode	Bytes	Clock Cycles	
			Compatibility	Fast
MOV /DPTR, #data16	A5 90	4	18	4
MOVC A, @A+/DPTR	A5 93	2	18	4
INC /DPTR	A5 A3	2	18	3
MOVX A, @/DPTR	A5 E0	2	18	5
MOVX @/DPTR, A	A5 F0	2	18	5

Note: The JMP @A+DPTR instruction is not part of the dual data pointer extensions. Adding A5H to this instruction results in the JMP @A+PC instruction. See "New Instructions" on page 22.

- Example: Block Copy Routine

```
        MOV  AUXR1, #00H    ; DPS = 0
        MOV  DPTR, #SRC     ; load source address to dptr0
        MOV  /DPTR, #DST    ; load destination address to dptr1
        MOV  R7, #BLKSIZE   ; number of bytes to copy
COPY:   MOVX A, @DPTR       ; read source (dptr0)
        INC  DPTR           ; next src (dptr0+1)
        MOVX @/DPTR, A      ; write destination (dptr1)
        INC  /DPTR          ; next dst (dptr1+1)
        DJNZ R7, COPY
```

Third party tools may not directly support these instructions. Contact your tool vendor for more information. If a tool does not support them, they can be forced in assembly code by explicitly inserting the 0A5H escape code before the equivalent standard instruction:

- Example: Inserting New Data Pointer Instructions

```
        DB   0A5H
        INC  DPTR           ; equivalent to INC /DPTR

        DB   0A5H
        MOVX A, @DPTR       ; equivalent to MOVX A, @/DPTR
```

Applications written in C will need to use the above as in-line assembly or link assembly source objects with the C source code.

The AT89LP52 also includes new update modes for the data pointers. Each data pointer can be configured to increment or decrement during the INC, MOVC and MOVX instructions. The update behavior is controlled by the DPD and DPU bits in AUXR1. For more information see the AT89LP52 datasheet.

- Example: Block Copy Routine with Auto Increment

```
        MOV  AUXR1, #0C0H   ; DPS = 0 DPU1=1 DPU0 = 1
        MOV  DPTR, #SRC     ; load source address to dptr0
        MOV  /DPTR, #DST    ; load destination address to dptr1
        MOV  R7, #BLKSIZE   ; number of bytes to copy
COPY:   MOVX A, @DPTR       ; read source (dptr0) and dptr0++
        MOVX @/DPTR, A      ; write destination (dptr1) and dptr1++
        DJNZ R7, COPY
```

12.3 New Instructions

The AT89LP52 includes 8 new instructions listed in Table 12-2. Third party tools may not directly support these instructions. Contact your tool vendor for more information. If a tool does not support them, the data pointer and jump instructions can be forced in assembly code by explicitly inserting the 0A5H escape code before the equivalent standard instruction:

- Example: Inserting JMP @A+PC

```
        DB   0A5H
        JMP  @A+DPTR        ; equivalent to JMP @A+PC
```

The new compare and jump instructions must be emulated in assembly:

- Example: Emulating CJNE A, R0, rel

```
DB   0A5H, 0B6H, LABEL-($+3)
```

For ease of use these can be wrapped in macros. Applications written in C will need to use the above as in-line assembly or link assembly source objects with the C source code.

Table 12-2. New Instructions

Mnemonic	Opcode	Bytes	Clock Cycles	
			Compatibility	Fast
JMP @A+PC	A5 73	2	12	3
MOV /DPTR, #data16	A5 90	4	18	4
MOVC A, @A+/DPTR	A5 93	2	18	4
INC /DPTR	A5 A3	2	18	3
CJNE A, @R0, rel	A5 B6	3	18	4
CJNE A, @R1, rel	A5 B7	3	18	4
MOVX A, @/DPTR	A5 E0	2	18	5
MOVX @/DPTR, A	A5 F0	2	18	5

13. Revision History

Revision No.	History
Revision A – Sept. 2010	• Initial Release

Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
Tel: (852) 2245-6100
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
8051@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Requests
www.atmel.com/literature