Migrating from AT89C51RB2/RC2/RD2/ED2/IC2/ID2 to AT89LP51RB2/RC2/RD2/ED2/IC2/ID2

8051 Flash Microcontroller

Application Note

New Features

- **SIngle Clock Cycle per Byte Fetch with 20 MIPS Throughput at 20MHz Clock Frequency**
- **12 Clock Instruction Compatibility Mode**
- **DSP Extensions with 40-bit 16x16 MAC**
- **Enhanced Dual Data Pointers**
- **4K Bytes of EEPROM Data Memory with 32-byte Page Programming (AT89LP51ED2 and AT89LP51ID2 Only)**
- **512 Bytes of User Signature Array**
- **SPI-based In-System Programming in addition to Serial Bootloader**
- **Enhanced Modes on Timer 0 and 1, including PWM**
- **TWI on AT89LP51RB2/RC2/RD2/ED2**
- **Edge or Level Triggered Keyboard Interrupts**
- **Shared Peripheral Prescaler**
- **Software Reset**
- **Selectable Polarity External Reset Pin**
- **Internal 8.0 MHz Auxiliary Oscillator**
- **Selectable HIgh and Low Power Crystal Oscillator**
- **Use Internal Oscillator for OSCB Source (AT89LP51IC2/ID2 Only)**
- **Configurable I/O Port Modes per Port Pin**
  - **Quasi-bidirectional (80C51 Style)**
  - **Input-only (Tristate)**
  - **Push-pull CMOS Output**
  - **Open-drain**
  - **Enable Strong Pull-ups on Port 0**
- **10-bit ADC/DAC with Temperature Sensor**
- **Dual Analog Comparators with Internal Reference**
- **On-Chip Debug Interface**
- **Up to 40 Programmable I/O Lines**
- **40-pin PDIP Package Option for AT89LP51RD2/ED2/IC2**
- **44-pad QFN/MLF Package Option for all devices**
- **Wide Operating Voltage 2.4–5.5V**

1. Introduction

The purpose of this application note is to help users convert existing designs from the Atmel AT89C51RB2/RC2/RD2/ED2/IC2/ID2 family to the Atmel AT89LP51RB2/RC2/RD2/ED2/IC2/ID2 family. Within this document AT89LP51xx2 will refer to any one of AT89LP51RB2/RC2/RD2/ED2/IC2/ID2 while AT89C51xx2 will refer to the older AT89C51RB2/RC2/RD2/ED2/IC2/ID2. AT89LP51xx2 is meant as a drop-in replacement for AT89C51xx2. The AT89LP51xx2 product includes 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 AT89LP51xx2 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 AT89LP51RB2/RC2/IC2 and AT89LP51RD2/ED2/ID2 datasheets.

Note that the Two-Wire Interface originally found only on AT89C51IC2/ID2 is now also present on AT89LP51RB2/RC2/RD2/ED2. Users of AT89C51IC2/ID2 can migrate to AT89LP51RC2/ED2 instead of AT89LP51IC2/ID2 if they do not use the second oscillator (OSCB) feature.

2. Getting Started

To get started with an AT89C51xx2 to AT89LP51xx2 migration, follow these steps:

1. Obtain an updated programming driver
   - Check if your existing device programmer has a device driver for the AT89LP51xx2. If your programmer does not support the AT89LP51xx2 you may need to find an alternate solution.
   - If you are solely using a Bootloader-based programming solution, you may need to find a programmer using the SPI-based interface or an OCD Debug solution to take full advantage of the enhanced features. See Section 3.1 “Configuration Fuses” on page 4.
2. Determine if you want to use 12-Clock Compatibility mode or Single-Cycle Fast mode
   - Compatibility mode provides the easiest migration because instruction/peripheral timing is maintained. In most cases no changes to the software are required. See Section 4.1 “Migrating to Compatibility Mode” on page 6.
   - Fast mode provides increased performance or lower power, but at the expense of some software changes. See Section 4.2 “Migrating to Fast Mode” on page 6.
3. If migrating to Fast mode, update fixed timings such as baud rates or timer overflows
   - Refer to Section 7. “System Clock” on page 12.
   - Refer to Section 10. “Timers” on page 19.
   - Refer to Section 11. “Watchdog” on page 21.
   - Refer to Section 12. “Serial Interfaces” on page 22.
4. Reconfigure the I/O pin modes, if desired
   - By default I/O pins operate compatible to AT89C51xx2. The I/O mode can be updated per port pin in software for greater flexibility. See Section 8. “I/O Ports” on page 14.
5. Make sure the reset pin is active high
   - The POL pin (formerly $\overline{EA}$) must be pulled high to ensure an AT89C51xx2 compatible reset. See Section 6. “Reset” on page 12.
   - Applications with $\overline{EA}$ low are not supported. However, external execution is still possible for some addresses even on the 64KB AT89LP51RD2/ED2/ID2 without using $\overline{EA}$. See Section 5. “Memory Access” on page 9.
6. Configure the device, if desired
   - The AT89LP51xx2 supports 15 more fuse options than AT89C51xx2. An SPI-based programmer is required to change these fuses. See Section 3.1 “Configuration Fuses” on page 4.

3. Flash/EEPROM Programming

The AT89LP51xx2 microcontroller does not support high-voltage parallel programming. Instead an SPI-based interface compatible with other AT89LP devices is provided. Users with third-party parallel programmers will need to update their programmer with the appropriate AT89LP51xx2 serial driver if available. However, the serial bootloader protocol is compatible with the existing AT89C51xx2. Existing bootloader-based programmers will be able to program the AT89LP51xx2 with an AT89C51xx2 driver, although it is recommended that drivers be updated to make use of the new features on the AT89LP51xx2. The flash programming differences are summarized in Table 3-1.

Table 3-1. Summary of Programming Differences on AT89LP51xx2

Feature	AT89C51xx2	AT89LP51xx2
High-Voltage Parallel Interface	YES	NO
4-wire SPI Interface	NO	YES
Serial UART Bootloader	YES	YES
Reset Polarity	Active-High	Selectable
Flash Page Size	128	128
Flash Page Buffer Size	128	64
Flash Auto-Erase Programming	Byte-Level	Page-Level
Device Signature Locations[1]	30H, 31H, 60H, 61H	00H, 01H, 02H, 30H, 31H, 60H, 61H
User Signature Array	0	512 bytes
EEPROM Data Memory	2KB	4KB
EEPROM Page Size	1	32
User Configuration Fuses	4	19

Note: 1. The device signature is stored in the Atmel Signature Array on AT89LP51xx2 and in the XROW on AT89C51xx2. Locations 30-31H and 60-61H have the same signature value in both AT89C51xx2 and AT89LP51xx2. Locations 00-02H are provided for compatibility with other AT89LP devices.

Users migrating from AT89C51xx2 to AT89LP51xx2 devices will notice the following changes to the programming interface:

1. The Hardware Security Byte (HSB) is accessible only through the Bootloader interface. The SPI-based interface accesses these bits individually.
2. The Bootloader configuration bytes (BSB, SSB and SBV) are located in the User Signature space at different addresses than AT89C51xx2. This affects only the SPI-based interface and not the Bootloader.
3. The AT89LP51xx2 has a half-page buffer of 64 bytes. Therefore Page mode commands accept only 64 bytes of data. However, Bootloader programming still allows a full 128 bytes to be sent.
4. Individual reprogramming of bytes is not possible. On AT89LP51xx2 one page is the smallest erasable quantity whereas on AT89C51xx2 it was one byte. Programming algorithms must erase and write an entire page when even only a single byte needs to change. This affects only the SPI-based interface and not the Bootloader; however, the bootloader is most efficient when working with full pages.

5. The Data[1] byte of the Program OSC Fuse command is changed from 20h on AT89C51IC2/ID2 to 10h on AT89LP51IC2/ID2.
6. The AT89LP51xx2 has a selectable polarity RST pin. Existing AT89C51xx2 drivers expect the reset to have active-high polarity. In most existing applications the $\overline{EA}$ pin is tied high, so dropping in an AT89LP51xx2 will result in an active-high reset. See Section 6. “Reset” on page 12.

In addition to the above changes, the AT89LP51xx2 also supports the following new features:

1. The AT89LP51xx2 includes a User Signature Array for storing up to 512 bytes of user ID, revision, configuration or other nonvolatile information. This information can be read/written from the application code. The BSB, SBV and SSB bytes are stored in this array. The full array is accessible to the SPI interface; however, only the first 256 bytes can be programmed through the Bootloader.
2. The AT89LP51xx2 supports 19 User Configuration Fuses for configuring the default behavior of the device. See Section 3.1 for more information. The fuses beyond the 4 used by the AT89C51xx2 are not accessible by the Bootloader. These fuses may require configuration before placing the device in system.
3. SPI-programming 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). This fuse does not affect the bootloader functionality.
4. The AT89LP51ED2/ID2 supports page programming of the EEPROM with up to 32 bytes written per operation.

3.1 Configuration Fuses

The AT89LP51xx2 includes nineteen User Configuration Fuses for configuring the default behavior of the device. Most fuses can only be changed by an SPI-based device programmer or OCD Debug solution that supports this feature. Four fuses are supported by the bootloader and Flash API. The default fuse settings are listed in Table 3-2. These settings were chosen to provide the greatest compatibility with the previous AT89C51xx2 device. See the AT89LP51RB2/RC2/IC2 and AT89LP51RD2/ED2/ID2 datasheets for more detailed information.

For existing applications that use an external clock source instead of a crystal or resonator, it is recommended that the CSA1 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.

For the AT89LP51IC2/ID2 devices the second oscillator source is also configurable with the $\text{CSB}_{1\text{-}0}$ fuses (addresses 11–12H).

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 3-2. Default Factory Fuse Settings

Address	Fuse Name	Default	Description
00H	Clock Source A	FFH	High Speed Crystal Oscillator (XTALA)
01H		FFH	

Table 3-2. Default Factory Fuse Settings

Address	Fuse Name	Default	Description
02H	Start-up Time	FFH	16 ms Delay
03H		FFH	
04h	Bootloader Jump Bit[1]	00H	Execute bootloader after reset (ENBOOT = 1)
05h	External RAM Enable[1]	FFH	External RAM Enabled (EXTRAM = 1)
06h	Compatibility Mode	FFH	CPU functions in 12-clock Compatibility mode
07h	ISP Enable	FFH	In-System Programming Enabled
08h	X1/X2 Mode[1]	FFH	X1 Mode (Divide-by-2)
09h	On-Chip Debug Enable	FFH	OCD Disabled
0AH	User Signature Programming	FFH	User Signature Programming Disabled
0BH	Tristate Ports	00H	Ports default to Quasi-bidirectional mode
0CH	EEPROM Erase	FFH	EEPROM is erased during chiperase
0DH	Low Power Mode	FFH	Low Power Mode
0EH		FFH	
0FH	R1 Enable	FFH	Internal 5 MΩ resistor connected to XTAL1A
10H	Oscillator Select[1]	FFH	Oscillator A Enabled (AT89LP51IC2/ID2 Only)
11H	Clock Source B	FFH	Low Frequency Crystal Oscillator (XTALB) (AT89LP51IC2/ID2 Only)
12H		FFH	

Note: 1. This fuse can be modified by the bootloader or through the Flash API

4. Compatibility vs. Fast Mode

The AT89LP51xx2 has a high performance, single-cycle CPU compatible with the 80C51 instruction set. For ease of migration the AT89LP51xx2 can operate in a Compatibility execution mode. In Compatibility mode the AT89LP51xx2 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 AT89C51xx2 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 AT89C51xx2; 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 system clock cycles. This greatly increases the throughput of the CPU. Each standard instruction executes in only 1 to 4 system 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 4-1 lists the major differences between Compatibility and Fast modes. This table assumes that both devices operate in the same X1/X2 clock mode.

Table 4-1. Compatibility Mode versus Fast Mode Summary[1]

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	X1/X2 Controlled	X1/X2 Controlled

Table 4-1. Compatibility Mode versus Fast Mode Summary[1]

Feature	Compatibility	Fast
Default Timer Prescaler Divisor	6	1
Pin Sampling Rate ($\overline{\text{INT0}}$, $\overline{\text{INT1}}$, 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

Note: 1. Cycle times are given in internal system clocks. These are the same as oscillator periods for X2 mode. For X1 mode, multiple system clocks by two to get equivalent oscillator periods.

4.1 Migrating to Compatibility Mode

Most users migrating legacy software from AT89C51xx2 to AT89LP51xx2 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 9.2 "Modified Registers" on page 16 in such a manner as to change the behavior of the application. In most cases this should not be an issue.

For users of AT89C51IC2/ID2 that manipulate the TWI lines directly through the port register, an update is required because these lines are mapped to P4.1-0 instead of PI2C.1-0.

In the rare case a user has a custom software bootloader or application that uses the low-level register interface (FCON) instead of the Flash API, changes to the code will be required. See Section 5.1.1 "In-Application Programming" on page 9.

Users willing to update their software can take advantage of any of the new peripheral features found on the AT89LP51xx2 in Compatibility mode while maintaining standard 8051 instruction timing.

4.2 Migrating to Fast Mode

Users migrating legacy software from AT89C51xx2 to AT89LP51xx2 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. Users willing to further update their software can take advantage of any of the new peripheral features found on the AT89LP51xx2 plus the greater performance of the Fast mode CPU.

For users of AT89C51IC2/ID2 that manipulate the TWI lines directly through the port register, an update is required because these lines are mapped to P4.1-0 instead of PI2C.1-0.

In the rare case a user has a custom software bootloader or application that uses the low-level register interface (FCON) instead of the Flash API, more changes to the code will be required. See Section 5.1.1 "In-Application Programming" on page 9.

4.2.1 System Frequency and Clock Division

One of the first decisions that users migrating designs from AT89C51xx2 to AT89LP51xx2 in Fast mode must make is at what frequency to run the AT89LP51xx2. By default the AT89LP51xx2 in Fast mode is at least 3 times faster than the AT89C51xx2 in the same X1/X2 mode, meaning it fetches bytes from memory in one-third the time. Therefore the maximum operating frequency may be lower in Fast mode than if the AT89LP51xx2 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 AT89LP51xx2. If so, the frequency must be scaled down.

The AT89LP51xx2 supports the X1/X2 feature of AT89C51xx2. This feature determines if the internal system clock is divided by two or not. The easiest migration path is to maintain the same mode on AT89LP51xx2 as was used on AT89C51xx2. This will preserve the timing of all peripheral features that depend only on the system clock such as Mode 2 of the UART. However, the best performance in terms of throughput and power is achieved only in X2 mode.

When migrating from X1 to X2 mode the system clock is no longer divided-by-2. Therefore if the same external clock frequency is maintained from AT89C51xx2 in X1, the AT89LP51xx2 in X2 mode will operate at an internal system clock that is twice as fast as on the AT89C51xx2. 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
- Set the Timer Prescaler for divide-by-two in the CLKREG SFR and enable prescaling on the UART, Timer 2, etc. through the CKCON0 and CKCON1 registers.

For timing on the external memory interfaces, see Section 5.1 “Program Memory” on page 9 and Section 5.2.1 “External Data Memory” on page 10.

Another difference in Fast mode is that the PCA, Timer 0, 1, 2 and the Watchdog by default count at a rate equal to the internal system clock instead of every sixth system clock. The counting behavior is made compatible with AT89C51xx2 by setting the timer prescaler in CLKREG to divide-by-6 (the default value) and then enabling the prescaler for a specific timer in CKCON0. If the mode is also changed from X1 to X2, then either the frequency must be halved or the prescaler should be doubled to divide-by-12. If one of the these steps is taken, any time-out periods or timer-generated baud rates will be maintained without any further changes.

- Example 1: AT89C51xx2 at external 24 MHz X1 or 12 MHz X2

  The AT89C51xx2 runs at an internal frequency of 12 MHz after the default divider of 2. The user chooses to run the AT89LP51xx2 at external 12 MHz X2. 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
MOV  CKCON0, #6FH     ; Set X2, T0X2, T1X2, T2X2, PCAX2, WDTX2
```

- Example 2: AT89C51xx2 at external 11.0592 MHz X1

  The user chooses to maintain the existing external frequency of 11.0592 Mhz. The user keeps the X1 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, #050H    ; TPS to div-by-6
MOV  CKCON0, #6EH     ; Clear X2; Set T0X2, T1X2, T2X2, PCAX2, WDTX2
```

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 disables the clock divider (X2). 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
MOV   CKCON0, #6FH     ; Set X2, T0X2, T1X2, T2X2, PCAX2, WDTX2
```

One advantage of Fast mode is that it allows the AT89LP51xx2 to provide the same instruction throughput as the AT89C51xx2 at a much lower frequency, thereby reducing overall power consumption. AT89LP51xx2 Fast mode is guaranteed to have at least 3 times the throughput of AT89C51xx2 at the same frequency and X1/X2 mode. Therefore the frequency of the AT89LP51xx2 can be reduced up to 3 times as compared to the AT89C51xx2 without reducing performance. However, doing so will affect the behavior of the peripherals. If the clock divider is disabled (X1 to X2), Fast mode is guaranteed to have at least 6 times the throughput, e.g. can operate 6 times slower than AT89C51xx2 in X1 mode.

- Example 3: AT89C51xx2 at external 24 MHz X1 or 12 MHz X2

  The user chooses to run the AT89LP51xx2 at external 4 MHz with no divider (X2). The minimum instruction throughput is the same as AT89C51xx2. 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 AT89C51xx2, 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
MOV   CKCON0, #6FH     ; Set X2, T0X2, T1X2, T2X2, PCAX2, WDTX2
```

In conclusion, the external system frequency, system clock divider and timer prescaler are all available as parameters for maintaining timing compatibility between AT89LP51xx2 in Fast mode and AT89C51xx2.

4.2.2 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 AT89C51xx2 and AT89LP51xx2 in Compatibility mode, the DJNZ instruction requires 12 system clocks. In Fast mode, the AT89LP51xx2 executes DJNZ in only 3 system clocks, thus N would need to be 4 times larger to generate the same delay at the same system frequency and X1/X2 mode.

4.2.3 ALE

In Fast mode the ALE signal toggles at a rate of half the system clock with 50% duty cycle 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 AO bit in AUXR for reduced power consumption.

4.2.4 Timers

If the same timer rate as AT89C51xx2 is not maintained as detailed above in Section 4.2.1, the timer reload or compare values will need to be scaled accordingly. See Section 10. on page 19

4.2.5 Watchdog

If the same timer rate as AT89C51xx2 is not maintained as detailed above in Section 4.2.1, the Watchdog may need to either be reset more often or have its period lengthened with the watchdog prescaler settings in WDTPRG. See Section 11. on page 21.

4.2.6 UART/SPI/TWI

If the same baud rate as AT89C51xx2 is not maintained as detailed above in Section 4.2.1, the baud rate setting may need to be scaled accordingly. See Section 12. on page 22.

5. Memory Access

The AT89LP51xx2 supports the same memory spaces found on the AT89C51xx2. The program memory has support for 24/32/64K bytes of on-chip Flash program memory and support for up to 40/32K 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 AT89LP51xx2.

5.1 Program Memory

The AT89LP51xx2 supports both internal and external program memory. Applications that only use external program memory, by tying the $\overline{\text{EA}}$ pin low, are not supported on the AT89LP51xx2. External program memory is only available at addresses 8000H–FFFFH (6000H–FFFFH) for AT89LP51RC2/IC2 (AT89LP51RB2). For AT89LP51RD2/ED2/ID2 external program memory is allowed at 8000H–FFFFH only when bank switching is enabled by setting the FBS bit in BMSEL.

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 4.2.1 on page 6. Furthermore, if switching from X1 to X2 mode, the external memory interface will fetch at twice the rate of X1 mode at the same frequency. 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 (X1) for the system clock in the CKCON0 SFR before accessing external memory.

5.1.1 In-Application Programming

Self-programming of the Flash Code memory, fuses and configuration bytes is supported through the same Flash API as found on AT89C51xx2. Legacy code that uses the Flash API does not require updating. In the rare case a user has a custom software bootloader or application that uses the low-level register interface (FCON) instead of the Flash API, changes to the code will be required as there are some differences between AT89LP51xx2 and AT89C51xx2 in this aspect. The low-level interface is described in the AT89LP51RB2/RC2/IC2 and AT89LP51 RD2/ED2/ID2 datasheets whereas it was omitted from the AT89C51xx2 documentation. This type of migration is beyond the scope of this document. Please contact the Atmel support team directly if questions arise.

5.1.2 Signature Array & Fuses

The AT89LP51xx2 makes the 128-byte Atmel Signature, 512-byte User Signature and 19 User Fuses readable and writable from the application code (Atmel Signature is read-only). For compatibility reasons only a subset of these are accessible through the bootloader and Flash API. Users willing to update their code can take full advantage of these resources through the low-level interface.

5.2 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 9. for a list of new or modified registers on the AT89LP51xx2. The external data memory interface may require some tuning for operation in Fast mode. EEPROM Page Mode is a new feature available on the AT89LP51ED2/ID2.

5.2.1 External Data Memory

The external data memory interface of the AT89LP51xx2 is accessed in the same manner as on AT89C51xx2. 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 EXTRAM bit in AUXR can also be used to override any other settings and force access to external memory. The default value is controlled by the XRAM fuse.

In Compatibility mode, MOVX timing for external memory access is identical to AT89C51xx2. 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 5-1. 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 5-2 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 5-1. Wait State Effects on MOVX Timing (in System Clocks)

		$\overline{\text{RD}}$ or $\overline{\text{WR}}$ Strobe Width		ALE low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ low Setup		Access Time (setup + width)	
WS1	**WS0**	**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 5-2. Configuration Settings for MOVX Timing in Fast Mode

Description	Configuration Settings		Access Time
	X2	WS	
AT89C51xx2	0	M0 = 0	9 x T_{OSC}
AT89LP51xx2 Compatibility Mode Default	0	0	9 x T_{OSC}
AT89LP51xx2 Fast Mode Default	1	0	2 x T_{OSC}
Nearest to AT89C51xx2 in Fast Mode at Same Frequency	1	3	5 x T_{OSC}
	0	2	8 x T_{OSC}
		3	10 x T_{OSC}
Nearest to AT89C51xx2 in Fast Mode at 1/2 Frequency	1	2	4 x (2 x T_{OSC})
		3	5 x (2 x T_{OSC})
	0	0	4 x (2 x T_{OSC})
Compatible to AT89C51xx2 in Fast Mode at 1/3 Frequency	1	1	3 x (3 x T_{OSC})
Nearest to AT89C51xx2 in Fast Mode at 1/6 Frequency	1	0	2 x (6 x T_{OSC})

Note: T_{OSC} is the external oscillator period of the AT89C51xx2

5.2.2 Internal Extra RAM

The AT89LP51RB2/RC2/IC2 (AT89LP51RD2/ED2/ID2) supports 1152 (2048) bytes of internal Extra RAM, 128 (256) bytes more than AT89C51RB2/RC2/IC2 (AT89LP51RD2/ED2/ID2). This additional RAM is accessible by changing the XRS bits in AUXR.

5.2.2.1 Paged Access

On the AT89C51xx2, only the first 256 bytes of Extra RAM are accessible with the MOVX @Ri instructions. On the AT89LP51xx2 the entire Extra RAM is accessible in a paged manner by using the PAGE register to select a 256-byte block. Furthermore, the PAGE register can force MOVX @Ri to use external memory without setting the EXTRAM bit. PAGE addressing is independent of XRS.

5.2.2.2 Extended Stack

The AT89LP51xx2 allows the hardware stack to be located anywhere in Extra RAM by setting the XSTK bit in AUXR1. The Extended Stack requires the use of the Extended Stack Pointer (SPX) in addition to SP. All stack operations incur a one cycle delay when operating on the extended stack.

5.2.3 EEPROM Data Memory

In addition to the 64K bytes of external data memory, addresses 0000H–0FFFH of the external data memory space are implemented on chip as 4K bytes of nonvolatile EEPROM data memory on the AT89LP51ED2/ID2. These bytes are only accessible when the EEE bit is set in EECON. This is twice the amount of EEPROM as present on AT89C51ED2/ID2. Note that unlike AT89C51ED2/ID2, addresses above the EEPROM range will still access external memory while EEE is set.

A new feature of the AT89LP51ED2/ID2 is page programming of the EEPROM. Page programming is controlled by the LDPG bit in EECON. While this bit is set, a write to the EEPROM space will not start a programming sequence, but will only load the page buffer. LDPG should be cleared before writing the final byte in a page. Up to 32 bytes can be loaded at a time.

- Example: Write a simple pattern in EEPROM data memory

```
        MOV  EECON, #022H   ; set LDPG EEE
        MOV  DPTR, #080H    ; Page to program
        MOV  R1, #0         ; pattern start
        MOV  R0, #31        ; 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  EECON, #~20H   ; clear LDPG
        MOV  A, R1
        MOVX @DPTR, A       ; load last byte and start write
        MOV  EECON, #0      ; clear EEE
```

6. Reset

The AT89LP51xx2 has a user-selectable external reset pin. To support this feature the former External Access pin ($\overline{\text{EA}}$) of the AT89C51xx2 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 AT89LP51xx2; however, external execution is still allowed for addresses 8000H–FFFFH (6000H–FFFFH) on AT89LP51RC2/IC2 (AT89LP51RB2). External execution on AT89LP51RD2/ED2/ID2 is allowed for addresses 8000H–FFFFH when FBS is set in BMSEL. The majority of legacy AT89C51xx2 applications have $\overline{\text{EA}}$ tied high for internal execution. Dropping an AT89LP51xx2 into these applications will result in POL tied high for an AT89C51xx2- compatible active-HIGH reset. If an AT89C51xx2 application has $\overline{\text{EA}}$ 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 AT89LP51xx2 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.

7. System Clock

The system clock source of the AT89LP51xx2 is selectable between the crystal oscillator, an externally driven clock and an internal 8.0 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. On the AT89LP51IC2/ID2 a second oscillator source is also available and is selectable between a low frequency (32KHz) crystal oscillator, an externally driven clock and the internal 8.0 MHz auxiliary oscillator. The clock source and options are controlled by the User Fuses. Section 3.1 “Configuration Fuses” on page 4.

The AT89LP51xx2 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 AT89LP51xx2 includes an on-chip 8.0 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 ±2.5% to enable UART communications and may be user calibrated to other frequencies near 8.0 MHz. In internal oscillator mode, XTAL1 and XTAL2 pins are available as a general purpose I/Os P4.6 and P4.7, respectively.

7.1 Clock Divider

The System Clock Divider scales the internal system clock versus the oscillator clock source. The divider is controlled by the CKRL register. In addition the system clock frequency may be divided by 2 from the externally supplied XTAL1 frequency for compatibility with standard 8051s in X1 mode (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. The X1/X2 feature is maintain in Fast mode. See “System Frequency and Clock Division” on page 6 for more information.

$$f_{SYS} = \frac{f_{OSC} \times 2^{X2}}{4 \times (255 - \mathrm{CKRL})} \qquad (\mathrm{CKRL} < 255)$$

$$f_{SYS} = \frac{f_{OSC} \times 2^{X2}}{2} \qquad (\mathrm{CKRL} = 255)$$

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

```
ORL   CKCON0, #01H     ; set X2
```

7.2 Timer Prescaler

A common prescaler is available to divide the time base for Timer 0, Timer 1, Timer 2, PCA 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} also defaults to 0101B; however, the prescaler must be enabled for a particular peripheral by settings its ?X2 bit in CKCON0/1. Note that in Fast mode the prescaler is also available for the SPI and TWI.

$$f_{PERIPHERAL} = \frac{f_{CPU}}{2^{?X2} \times (TPS + 1)} \qquad \text{Compatibility Mode}$$

$$f_{PERIPHERAL} = \frac{f_{CPU}}{TPS + 1} \qquad \text{Fast Mode and ?X2 = 1}$$

- Example: Scale Timers by 6

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

8. I/O Ports

The P0, P1, P2, P3 and P4 I/O port pins of the AT89LP51xx2 may be independently configured in four different modes:

- Quasi-bidirectional
  Standard 80C51 I/O with strong, medium and weak pull-ups good for general bidirectional use.
- Input Only
  The I/O is tristated for high impedance input. 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.
- Push-Pull Output
  The I/O provides a full CMOS output driver with larger current sourcing capabilities than quasi-bidirectional.
- Open-Drain
  The I/O pull-ups are disabled. An external pull-up is required to use the I/O 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 enabled, all port pins on P1, P2 and P3 default to input-only mode after reset. When the fuse is disabled, all port pins on P1, P2 and P3 default to quasi-bidirectional mode after reset and are weakly pulled high. P0 always defaults to open-drain mode. P4.4–5 always default to quasi-bidirectional mode. P4.0–1 always default to open-drain. The other pins of P4 obey the fuse. 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  P0M0, #00H       ; all P0 pins quasi-bidirectional
```

Port configuration is set by the PxM0 and PxM1 registers and is independently configurable for each pin. Ports pins using bidirectional signals should not be set to Input Only or Push-Pull Output modes as these modes are unidirectional.

Table 8-1. Configuration Modes for Port x Pin.y

PxM1.y	PxM0.y	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 default to 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, either by clearing the Tristate-Port Fuse or writing the correct P3M0 and P3M1 bits.

The AT89LP51xx2 provides an additional I/O port P4. Note that this P4 differs from the P4 originally offered on the obsoleted PLCC68 and VQFP64 packages of AT89C51RD2/ED2. Pins P4.2 and P4.4–P4.7 replace the normally dedicated XTAL2B, ALE, $\overline{\text{PSEN}}$, XTAL1(A) and XTAL2(A) pins of the AT89LP51xx2. These pins can be used as additional I/Os depending on the configuration of the clock and external memory as listed in Table 8-2. Pins P4.1-0 are the TWI lines that were previously called PI2.1-0 on AT89C51IC2/ID2. P4.3 is the data line for the On-Chip Debug database.

Table 8-2. Configurations for Additional I/Os

Pin	Function	Configuration Required for General I/O Use
P4.0	SCL	None. (Replaces PI2.0 in AT89C51IC2/ID2)
P4.1	SDA	None. (Replaces PI2.1 in AT89C51IC2/ID2)
P4.2	XTAL2B	Internal 8.0 MHz oscillator or external clock selected for OSCB; or OSCB disabled (AT89LP51IC2/ID2 Only)
P4.3	DDA	OCD Disabled
P4.4	ALE	No external memory and AO = 1
P4.5	$\overline{\text{PSEN}}$	No external program memory
P4.6	XTAL1(A)	Internal 8.0 MHz oscillator selected for OSCA[1]
P4.7	XTAL2(A)	External clock or internal 8.0 MHz oscillator selected for OSCA[1]

Note: 1. For AT89LP51IC2/ID2 the OSCA pins dependent only on the OSCA source selection and not on whether OSCA is enabled or not.

9. Special Function Registers

This section lists the Special Function Registers (SFRs) that are new or modified in AT89LP51xx2 from those in AT89C51xx2.

9.1 SFR Mapping

The highlighted SFR locations are new registers for the AT89LP51xx2 device.

Table 9-1. SFR Mapping in AT89LP51xx2

0F8H		CH	CCAP0H	CCAP1H	CCAP2H	CCAP3H	CCAP4H		0FFH
0F0H	B		RL0	RL1	RH0	RH1	PAGE	BX	0F7H
0E8H		CL	CCAP0L	CCAP0L	CCAP2L	CCAP3L	CCAP4L	SPX	0EFH
0E0H	ACC	AX	DSPR	FIRD	MACL	MACH	P0M0	P0M1	0E7H
0D8H	CCON	CMOD	CCAPM0	CCAPM1	CCAPM2	CCAPM3	CCAPM4		0DFH
0D0H	PSW	FCON	EECON		DPLB	DPHB	P1M0	P1M1	0D7H
0C8H	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2	P2M0	P2M1	0CFH
0C0H	P4			SPCON	SPSTA	SPDAT	P3M0	P3M1	0C7H
0B8H	IPL0	SADEN				AREF	P4M0	P4M1	0BFH
0B0H	P3	IEN1	IPL1	IPH1				IPH0	0B7H
0A8H	IEN0	SADDR		ACSRA	DADL	DADH	CLKREG	CKCON1	0AFH
0A0H	P2	DPCF	AUXR1	ACSRA	DADC	DADI	WDTRST	WDTCON	0A7H
98H	SCON	SBUF	BRL	BDRCON	KBLS	KBE	KBF	KBMOD	9FH
90H	P1	TCONB	BMSEL	SSCON	SSCS	SSDAT	SSADR	CKRL	97H
88H	TCON	TMOD	TL0	TL1	TH0	TH1	AUXR	CKCON0	8FH
80H	P0	SP	DP0L	DP0H		CKSEL	OSCCON	PCON	87H

9.2 Modified Registers

The following registers are modified in their bit definitions from AT89C51xx2. For more detailed information see the AT89LP51RB2/RC2/IC2 and AT89LP51RD2/ED2/ID2 datasheets.

Table 9-2. AUXR Register Bits

AUXR (8EH)	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AT89C51xx2	DPU	–	M0	XRS2	XRS1	XRS0	EXTRAM	AO
AT89LP51xx2	DPU	WS1	WS0	XRS2	XRS1	XRS0	EXTRAM	AO

Table 9-3. AUXR1 Register Bits

AUXR1 (A2H)	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AT89C51xx2	–	–	–	ENBOOT	GF3	0	–	DPS
AT89LP51xx2	–	–	XSTK	ENBOOT	GF3	0	–	DPS

Table 9-4. EECON Register Bits

EECON (D2H)	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AT89C51xx2	–	–	–	–	–	–	EEE	EEBUSY
AT89LP51xx2	FOUT	AERS	LDPG	FLGE	$\overline{\text{INHIBIT}}$	ERR	EEE	EEBUSY

Table 9-5. PCON Register Bits

PCON (87H)	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AT89C51xx2	SMOD1	SMOD0	–	POF	GF1	GF0	PD	IDL
AT89LP51xx2	SMOD1	SMOD0	PWDEX	POF	GF1	GF0	PD	IDL

Table 9-6. SPSTA Register Bits

SPSTA (C4H)	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AT89C51xx2	SPIF	WCOL	SSERR	MODF	–	–	–	–
AT89LP51xx2	SPIF	WCOL	SSERR	MODF	TXE	DORD	REMAP	TXIE

Table 9-7. WDTPRG Register Bits

WDTPRG (A7H)	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AT89C51xx2	–	–	–	–	–	WTO2	WTO1	WTO0
AT89LP51xx2	OVF	SWRST	WDTEN	WDIDLE	DISRTO	WTO2	WTO1	WTO0

9.3 New Registers

The following registers are additional registers not found in AT89C51xx2. For more detailed information see the AT89LP51RB2/RC2/IC2 and AT89LP51RD2/ED2/ID2 datasheets.

Table 9-8. ACSRA Register Bits

ACSRA (A3H)	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reset = 0000 0000	CSA1	CSA0	CONA	CFA	CENA	CMA2	CMA1	CMA0

Table 9-9. ACSRBA Register Bits

ACSRB (ABH)	7	6	5	4	3	2	1	0
Reset = 0000 0000	CSB1	CSB0	CONB	CFB	CENB	CMB2	CMB1	CMB0

Table 9-10. AREF Register Bits

AREF (BDH)	7	6	5	4	3	2	1	0
Reset = 0000 0000	CMPB	CMPA	RFB1	RFB0	CCS1	CCS0	RFA1	RFA0

Table 9-11. AX Register Bits

AX (E1H)	7	6	5	4	3	2	1	0
Reset = 0000 0000	ACC.15	ACC.14	ACC.13	ACC.12	ACC.11	ACC.10	ACC.9	ACC.8

Table 9-12. BMSEL Register Bits (AT89LP51RD2/ED2/ID2 Only)

BMSEL (92H)	7	6	5	4	3	2	1	0
Reset = 0000 0000	–	–	–	–	–	–	–	FBS

Table 9-13. BX Register Bits

BX (F7H)	7	6	5	4	3	2	1	0
Reset = 0000 0000	B.15	B.14	B.13	B.12	B.11	B.10	B.9	B.8

Table 9-14. CLKREG Register Bits

CLKREG (AEH)	7	6	5	4	3	2	1	0
Reset = 0101 xxxx	TPS3	TPS2	TPS1	TPS0	–	–	–	–

Table 9-15. DADC Register Bits

DADC (A4H)	7	6	5	4	3	2	1	0
Reset = 0000 0000	ADIF	GO/BSY	DAC	ADCE	LADJ	ACK2	ACK1	ACK0

Table 9-16. DADI Register Bits

DADI (A5H)	7	6	5	4	3	2	1	0
Reset = 0000 0000	ACON	IREF	TRG1	TRG0	DIFF	ACS2	ACS1	ACS0

Table 9-17. DADH Register Bits

DADH (ADH)	7	6	5	4	3	2	1	0
Reset = 0000 0000	ADC.15	ADC.14	ADC.13	ADC.12	ADC.11	ADC.10	ADC.9	ADC.8

Table 9-18. DADL Register Bits

DADL (ACH)	7	6	5	4	3	2	1	0
Reset = 0000 0000	ADC.7	ADC.6	ADC.5	ADC.4	ADC.3	ADC.2	ADC.1	ADC.0

Table 9-19. DPCF Register Bits

DPCF (A1H)	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reset = 0000 xxxx	DPU1	DPU0	DPD1	DPD0	LADJ	ACK2	ACK1	ACK0

Table 9-20. DPHB Register Bits

DPHB (D5H)	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reset = 0000 0000	DPB.15	DPB.14	DPB.13	DPB.12	DPB.11	DPB.10	DPB.9	DPB.8

Table 9-21. DPLB Register Bits

DPLB (D4H)	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reset = 0000 0000	DPB.7	DPB.6	DPB.5	DPB.4	DPB.3	DPB.2	DPB.1	DPB.0

Table 9-22. DSPR Register Bits

DSPR (E2H)	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reset = 0000 0000	MRW1	MRW0	SMLB	SMLA	CBE1	CBE0	MVCD	DPRB

Table 9-23. FIRD Register Bits

FIRD (E3H)	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reset = 0000 0000	FIRD.7	FIRD.6	FIRD.5	FIRD.4	FIRD.3	FIRD.2	FIRD.1	FIRD.0

Table 9-24. KBMOD Register Bits

IPH (9FH)	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reset = 0000 0000	KBM7	KBM6	KBM5	KBM4	KBM3	KBM2	KBM1	KBM0

Table 9-25. PAGE Register Bits

PAGE (F6H)	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reset = xxxx 0000	–	–	–	–	PAGE.3	PAGE.2	PAGE.1	PAGE.0

Table 9-26. SPX Register Bits

SPX (EFH)	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reset = 0000 0000	–	–	–	–	–	SP.10	SP.9	SP.8

Table 9-27. TCONB Register Bits

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

Table 9-28. Port Configuration Registers

Port	Register	Address	Reset	Register	Address	Reset
P0	P0M0	E6H	FFH	P0M1	E7H	FFH
P1	P1M0	D6H	[1]	P1M1	D7H	00H
P2	P2M0	CEH	[1]	P2M1	CFH	00H
P3	P3M0	C6H	[1]	P3M1	C7H	00H
P4	P4M0	BEH	[2]	P4M1	BFH	03H

Notes: 1. Reset value is FFH when Tristate-Port Fuse is enabled and 00H when disabled

2. Reset value is C7H when Tristate-Port Fuse is enabled and 03H when disabled

Table 9-29. Timer 0/1 Reload Registers

Timer	Register	Address	Reset	Register	Address	Reset
Timer 0	RL0	F2H	00H	RH0	F4H	00H
Timer 1	RL1	F3H	00H	RH1	F5H	00H

10. Timers

The timer counting rate of Timer 0, Timer 1, Timer 2 and the PCA is controlled by the timer prescaler. The timer prescaler defaults to one sixth of the system clock, the same rate as AT89C51xx2. In Compatibility Mode the prescaler always affects the timer operation. No software changes to the timers should be required when migrating legacy software to Compatibility Mode. In Fast mode the prescaler is disabled for all timers and the timers count at the system clock rate. See “System Frequency and Clock Division” on page 6. The user can control the behavior of the timers with the timer prescaler (TPS) settings in CLKREG, the clock divider setting (CKRL) and the individual clock control bits in CKCON0. In Fast mode each CKCON0 bit (T0X2, T1X2, T2X2 or PCAX2) enables/disables the prescaler for each timer individually. See “Timer Prescaler” on page 13.

Table 10-1 lists several possible configurations for the Timers.

Full timer rate compatibility with AT89C51xx2 for Timer 2 in Baud Rate or Clock-Out modes or PCA with CPS=1 can only be maintained in AT89LP51xx2 Fast mode in the following two cases:

- If AT89LP51xx2 operates at the same frequency as AT89C51xx2
  - If AT89C51xx2 uses X1 mode, then on AT89LP51xx2 T2X2/PCAX2 = X2
  - If AT89C51xx2 uses X2 mode, then on AT89LP51xx2 T2X2/PCAX2 = 0 and X2 = 1
- If AT89LP51xx2 operates at half the frequency of AT89C51xx2
  - Then T2X2/PCAX2 = 0 and X2 = 1.

Otherwise the reload values may need to be scaled accordingly. For example, if the AT89LP51xx2 operates at the same frequency as AT89C51xx2 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.

Table 10-1. Configuration Settings for Compatible Timer Rates (CKRL=255)

Timer 0 or Timer 1	**Configuration Settings**			
Timer 2 in Auto-Reload or Capture Modes **PCA with CPS=0**	**TnX2 / PCAX2**	**X2**	**TPS**	**Timer Rate**
AT89C51xx2	0	0	–	$f_{C51} / 12$
		1	–	$f_{C51} / 6$
	1	1	–	$f_{C51} / 12$
AT89LP51xx2 Compatibility Mode Default	0	0	5	$f_{OSC} / 12$
		1	5	$f_{OSC} / 6$
	1	1	5	$f_{OSC} / 12$
AT89LP51xx2 Fast Mode Default	0	0	–	$f_{OSC} / 2$
		1	–	$f_{OSC} / 1$
	1	0	5	$f_{OSC} / 12$
		1	5	$f_{OSC} / 6$
AT89LP51xx2 Fast Mode Compatible to AT89C51xx2 X1 at Same Frequency ($f_{OSC} = f_{C51}$)	1	0	5	$f_{OSC} / 12 = f_{C51} / 12$
	1	1	11	
AT89LP51xx2 Fast Mode Compatible to AT89C51xx2 X2 at Same Frequency ($f_{OSC} = f_{C51}$)	1	0	2	$f_{OSC} / 6 = f_{C51} / 6$
	1	1	5	
AT89LP51xx2 Fast Mode Compatible to AT89C51xx2 X1 at Half Frequency ($f_{OSC} = f_{C51} \div 2$)	1	0	2	$f_{OSC} / 6 = f_{C51} / 12$
	1	1	5	
AT89LP51xx2 Fast Mode Compatible to AT89C51xx2 X2 at Half Frequency ($f_{OSC} = f_{C51} \div 2$)	1	1	2	$f_{OSC} / 3 = f_{C51} / 6$
AT89LP51xx2 Fast Mode Compatible to AT89C51xx2 X1 at One-Sixth Frequency ($f_{OSC} = f_{C51} \div 6$)	0	0	–	$f_{OSC} / 2 = f_{C51} / 12$
	1	1	1	
AT89LP51xx2 Fast Mode Compatible to AT89C51xx2 X2 at One-Sixth Frequency ($f_{OSC} = f_{C51} \div 6$)	0	1	–	$f_{OSC} / 1 = f_{C51} / 6$
Timer 2 in Baud Rate or Clock-Out Modes **PCA with CPS=1**	**TnX2 / PCAX2**	**X2**	**TPS**	**Timer Rate**
AT89C51xx2	0	0	–	$f_{C51} / 2$
		1	–	$f_{C51} / 1$
	1	1	–	$f_{C51} / 2$
AT89LP51xx2 Compatibility Mode Default	0	0	–	$f_{OSC} / 2$
		1	–	$f_{OSC} / 1$
	1	1	–	$f_{OSC} / 2$
AT89LP51xx2 Fast Mode Default	0	0	–	$f_{OSC} / 2$
		1	–	$f_{OSC} / 1$
	1	0	–	$f_{OSC} / 4$
		1	–	$f_{OSC} / 2$

Table 10-1. Configuration Settings for Compatible Timer Rates (CKRL=255)

Timer 0 or Timer 1 Timer 2 in Auto-Reload or Capture Modes PCA with CPS=0	Configuration Settings			Timer Rate
	TnX2 / PCAX2	X2	TPS	
AT89LP51xx2 Fast Mode Compatible to AT89C51xx2 X1 at Same Frequency ($f_{OSC} = f_{C51}$)	0	0	–	f_{OSC} / 2 = f_{C51} / 2
	1	1	–	
AT89LP51xx2 Fast Mode Compatible to AT89C51xx2 X2 at Same Frequency ($f_{OSC} = f_{C51}$)	0	1	–	f_{OSC} / 1= f_{C51} / 1
AT89LP51xx2 Fast Mode Compatible to AT89C51xx2 X1 at Half Frequency ($f_{OSC} = f_{C51} \div 2$)	0	1	–	f_{OSC} / 1= f_{C51} / 2

Note: f_{C51} is the XTAL1 frequency of the AT89C51xx2.
fosc is the XTAL1 frequency of the AT89LP51xx2.

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

An additional feature on AT89LP51xx2 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.

11. Watchdog

The AT89LP51xx2 adds new flags to the WDTPRG register. The flags allow the user to determine if the watchdog is currently running, if the watchdog previously overflowed, and if a software reset occurred. Code migrated from AT89C51xx2 to AT89LP51xx2 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 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 AT89C51xx2. In Fast mode the watchdog counts at the system clock rate. See “System Frequency and Clock Division” on page 6. The user can control the behavior of the Watchdog with the timer prescaler (TPS) settings in CLKREG, clock divider (X2)in CKCON0 or the watchdog prescaler (WTO) settings in WDTCON. Table 11-1 lists several possible configurations for the Watchdog.

Table 11-1. Configuration Settings for Compatible Time-out Periods (WTO = 0)

Description	Configuration Settings			Time-out
	X2	WDX2	TPS	
AT89C51xx2	0	0	–	196608 x T_{C51}
	1	0	–	98304 x T_{C51}
		1	–	196608 x T_{C51}
AT89LP51xx2 Compatibility Mode Default	0	0	5	196608 x T_{OSC}
	1	0	5	98304 x T_{OSC}
		1	5	196608 x T_{OSC}

Table 11-1. Configuration Settings for Compatible Time-out Periods (WTO = 0)

	Configuration Settings			
Description	X2	WDX2	TPS	Time-out
AT89LP51xx2 Fast Mode Default	0	0	5	32768 x T_{OSC}
	1	0	5	16384 x T_{OSC}
		1	5	98034 x T_{OSC}
AT89LP51xx2 in Fast Mode Compatible to AT89C51xx2 X1 at Same Frequency	0	1	5	196608 x T_{OSC}
	1	1	11	
AT89LP51xx2 in Fast Mode Compatible to AT89C51xx2 X2 at Same Frequency	1	1	5	98304 x T_{OSC}
AT89LP51xx2 in Fast Mode Compatible to AT89C51xx2 at Half Frequency $T_{OSC} = T_{C51}$÷2)	1	1	5	98304 x (2 x T_{OSC})
AT89LP51xx2 in Fast Mode Compatible to AT89C51xx2 at One-Sixth Frequency	0		1	32768 x (6 x T_{OSC})
AT89LP51xx2Nearest to AT89C51xx2 in Fast Mode at Same Frequency without Timer Prescaler	0	0	0	131072 x T_{OSC}
	0	0	0	262144 x T_{OSC}

11.1 Software Reset

Software reset is a new feature of the AT89LP51xx2 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
```

12. Serial Interfaces

The UART, SPI and TWI in Compatibility mode behave identically to AT89C51xx2 by default and should not require any changes for most applications. 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.

For users of AT89C51IC2/ID2 that manipulate the TWI lines directly through the port register, an update is required because these lines are mapped to P4.1-0 instead of PI2C.1-0.

12.1 UART under Fast Mode

The baud rate of the UART in Modes 0 or 2 can only be maintained in AT89LP51xx2 Fast mode in the following three cases:

- AT89LP51xx2 operates at the same frequency and X1/X2 settings as AT89C51xx2 and SX2 in CKCON0 is cleared
- AT89LP51xx2 X2 operates at the same frequency as AT89C51xx2 X1 and SX2 is set
- AT89LP51xx2 X2 operates at half the frequency of AT89C51xx2 X1 and SX2 is cleared

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 12-1 lists several possible configurations for the baud rate generation in Fast mode that do not require updates to the timer values.

Table 12-1. Configuration Settings for Compatible Baud Rates in Fast Mode

Mode	Baud Source	System Frequency	Configuration Settings		
			X2	**SX2**	**TPS**
0 or 2	–	$f_{LP51} = f_{C51}$ (1)	$X2_{LP51} = X2_{C51}$	0	–
		$f_{LP51} = f_{C51}$ (2)	1	1	–
		$f_{LP51} = f_{C51} / 2$ (2)	1	0	–
1 or 3	Timer 1	$f_{LP51} = f_{C51}$ (1)	$X2_{LP51} = X2_{C51}$	1	5
		$f_{LP51} = f_{C51}$ (2)	1	1	11
		$f_{LP51} = f_{C51} / 2$ (2)	1	1	5
		$f_{LP51} = f_{C51} / 3$ (1)	$X2_{LP51} = X2_{C51}$	1	1
		$f_{LP51} = f_{C51} / 4$ (2)	1	1	2
		$f_{LP51} = f_{C51} / 6$ (1)	$X2_{LP51} = X2_{C51}$	0	–
	Timer 2	$f_{LP51} = f_{C51}$ (1)	$X2_{LP51} = X2_{C51}$	0	–
		$f_{LP51} = f_{C51}$ (2)	1	1	–
		$f_{LP51} = f_{C51} / 2$ (2)	1	0	–

Notes: 1. AT89C51xx2 operates in X1 or X2 mode

2. AT89C51xx2 operates in X1 mode

- Example: AT89C51xx2 X1 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 (X2 = 0). 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 SX2 = 1 and TPS = 5 will allow TH1 to stay at FDH and SMOD1 to stay 1. If SX2 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 (X2 = 1). If Timer 2 is the baud generator, setting T2X2 = 1 will maintain the baud rate; otherwise RCAP2H/RCAP2L must be updated to FFDCH. If Timer 1 is the baud generator in Mode 2, setting T1X2 = 1 and TPS = 11 will allow TH1 to stay at FDH and SMOD1 to stay 1. If T1X2 is left at 0, then TH1 must be updated to DCH for SMOD1 = 1.

12.2 SPI under Fast Mode

The baud rate of the SPI can only be maintained in AT89LP51xx2 Fast mode in the following four cases:

- AT89LP51xx2 operates at the same frequency and X1/X2 settings as AT89C51xx2 and SPIX2 in CKCON1 is cleared
- AT89LP51xx2 X2 operates at the same frequency as AT89C51xx2 X1 and SPIX2 is set and TPS = 1
- AT89LP51xx2 X2 operates at half the frequency of AT89C51xx2 X1 and SPIX2 is cleared

- AT89LP51xx2 operates at the same X1/X2 settings as AT89C51xx2, but at a frequency divided by 2–16 with SPIX2 = 1 and TPS set to the division factor minus 1

In other cases the SPI baud rate setting must be updated to achieve a similiar rate.

12.3 TWI under Fast Mode

The baud rate of the TWI can only be maintained in AT89LP51xx2 Fast mode in the following four cases:

- AT89LP51xx2 operates at the same frequency and X1/X2 settings as AT89C51xx2 and TWX2 in CKCON0 is cleared
- AT89LP51xx2 X2 operates at the same frequency as AT89C51xx2 X1 and TWX2 is set and TPS = 1;
- AT89LP51xx2 X2 operates at half the frequency of AT89C51xx2 X1 and TWX2 is cleared
- AT89LP51xx2 operates at the same X1/X2 settings as AT89C51xx2, but at a frequency divided by 2–16 with TWX2 = 1 and TPS set to the division factor minus 1

In other cases the TWI baud rate setting must be updated to achieve a similiar rate.

13. Additional Features

In addition to the features discussed above, the AT89LP51xx2 includes the following new features.

13.1 Interrupts

In addition to the interrupt sources on the AT89C51xx2, the AT89LP51xx2 supports interrupts for the Analog Comparators and ADC/DAC. Also, Fast mode allows for faster interrupt response due to the shorter instruction execution times at a given frequency.

13.2 Dual Data Pointers

The AT89LP51xx2 adds instruction level support for dual data pointers. The five new instructions listed in Table 13-1 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 13-1. New Dual Data Pointer Instructions

			Clock Cycles	
Mnemonic	**Opcode**	**Bytes**	**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 26.

- 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 AT89LP51xx2 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 DPCF. For more information see the AT89LP51xx2 datasheet.

- Example: Block Copy Routine with Auto Increment

```
        MOV  AUXR1, #00H    ; DPS = 0
        MOV  DPCF, #0C0H    ; 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
```

13.3 New Instructions

The AT89LP51xx2 includes 12 new instructions listed in Table 13-2. 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 13-2. New Instructions

			Clock Cycles	
Mnemonic	**Opcode**	**Bytes**	**Compatibility**	**Fast**
ASR M	A5 03	2	12	2
LSL M	A5 23	2	12	2
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
MAC AB	A5 A4	2	30	9
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
CLR M	A5 E4	2	12	2
MOVX @/DPTR, A	A5 F0	2	18	5

14. Revision History

Revision No.	History
Revision A – October 2011	• Initial Release

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: (+1) (408) 441-0311
Fax: (+1) (408) 487-2600
www.atmel.com
8051@atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Packring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

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

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.