

## Features

- Multichip Module Containing Field Programmable System Level Integrated Circuit (FPSLIC™) and Secure Configuration EEPROM Memory
- 512 Kbits to 1 Mbit of Configuration Memory with Security Protection and In-System Programming (ISP)
- Field Programmable System Level Integrated Circuit (FPSLIC)
  - AT40K SRAM-based FPGA with Embedded High-performance RISC AVR® Core and Extensive Data and Instruction SRAM
- 5,000 to 40,000 Gates of Patented SRAM-based AT40K FPGA with FreeRAM™
  - 2 - 18.4 Kbits of Distributed Single/Dual Port FPGA User SRAM
  - High-performance DSP Optimized FPGA Core Cell
  - Dynamically Reconfigurable In-System – FPGA Configuration Access Available On-chip from AVR Microcontroller Core to Support Cache Logic® Designs
  - Very Low Static and Dynamic Power Consumption – Ideal for Portable and Handheld Applications
- Patented AVR Enhanced RISC Architecture
  - 120+ Powerful Instructions – Most Single Clock Cycle Execution
  - High-performance Hardware Multiplier for DSP-based Systems
  - Approaching 1 MIPS per MHz Performance
  - C Code Optimized Architecture with 32 x 8 General-purpose Internal Registers
  - Low-power Idle, Power-save, and Power-down Modes
  - 100 µA Standby and Typical 2-3 mA per MHz Active
- Up to 36 Kbytes of Dynamically Allocated Instruction and Data SRAM
  - Up to 16 Kbytes x 16 Internal 15 ns Instructions SRAM
  - Up to 16 Kbytes x 8 Internal 15 ns Data SRAM
- AVR Fixed Peripherals
  - Industry-standard 2-wire Serial Interface
  - Two Programmable Serial UARTs
  - Two 8-bit Timer/Counters with Separate Prescaler and PWM
  - One 16-bit Timer/Counter with Separate Prescaler, Compare, Capture Modes and Dual 8-, 9- or 10-bit PWM
- Support for FPGA Custom Peripherals
  - AVR Peripheral Control – Up to 16 Decoded AVR Address Lines Directly Accessible to FPGA
  - FPGA Macro Library of Custom Peripherals
- Up to 16 FPGA Supplied Internal Interrupts to AVR
- Up to Four External Interrupts to AVR
- 8 Global FPGA Clocks
  - Two FPGA Clocks Driven from AVR Logic
  - FPGA Global Clock Access Available from FPGA Core
- Multiple Oscillator Circuits
  - Programmable Watchdog Timer with On-chip Oscillator
  - Oscillator to AVR Internal Clock Circuit
  - Software-selectable Clock Frequency
  - Oscillator to Timer/Counter for Real-time Clock
- V<sub>CC</sub>: 3.0V - 3.6V
- 5V Tolerant I/O
- 3.3V 33 MHz PCI Compliant FPGA I/O
  - 20 mA Sink/Source High-performance I/O Structures
  - All FPGA I/O Individually Programmable
- High-performance, Low-power 0.35µ CMOS Five-layer Metal Process
- State-of-the-art Integrated PC-based Software Suite including Co-verification



**Secure  
5K - 40K Gates  
of AT40K FPGA  
with 8-bit AVR®  
Microcontroller,  
up to 36 Kbytes  
of SRAM and  
On-chip  
Program  
Storage  
EEPROM**

**AT94S  
Secure Series  
Programmable  
SLI**

Rev. 2314C-FPSLI-06/02





## Description

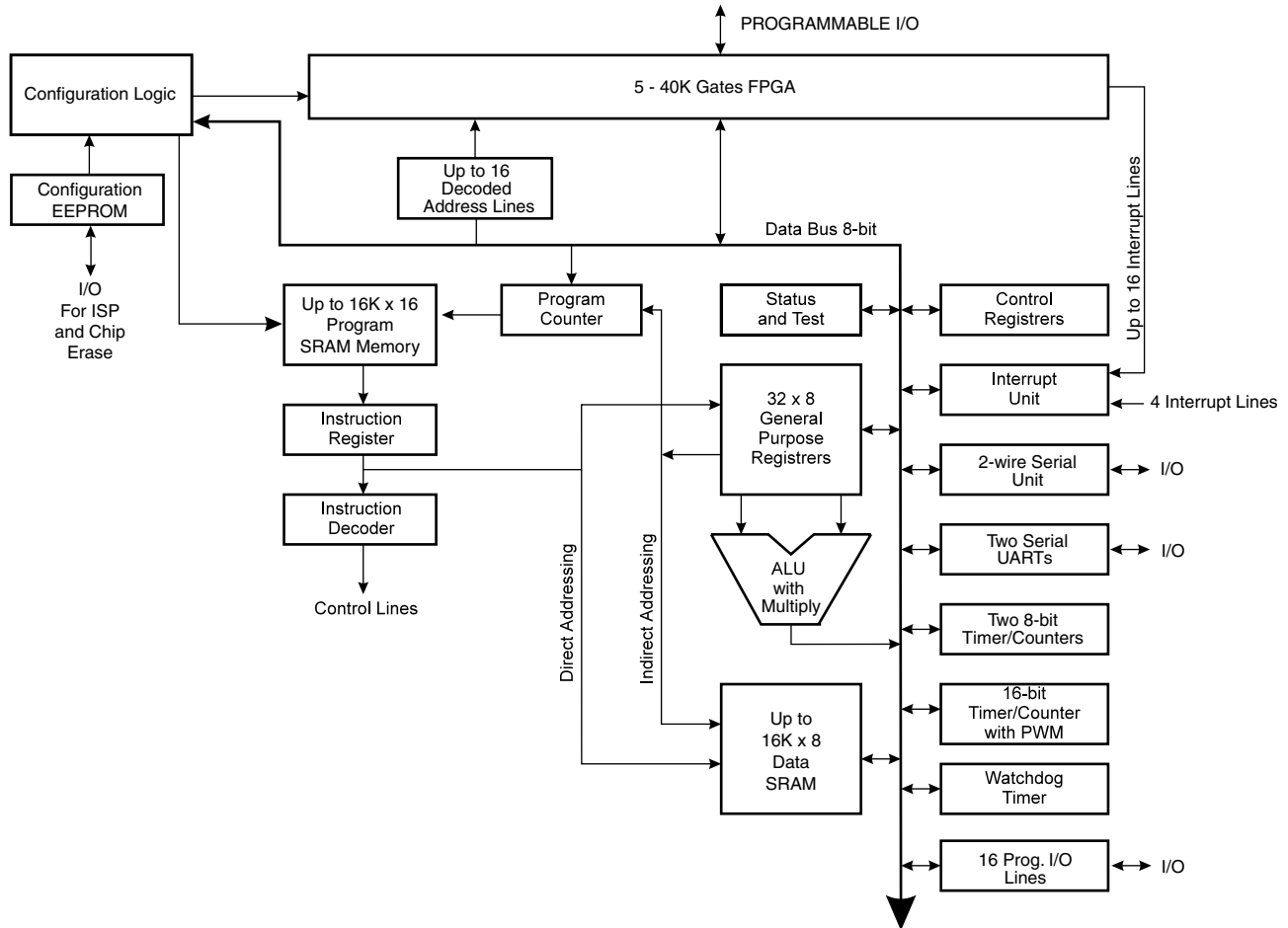
The AT94S Series (Secure FPSLIC family) shown in Table 1 is a combination of the popular Atmel AT40K Series SRAM FPGAs, the AT17 Series Configuration Memories and the high-performance Atmel AVR 8-bit RISC microcontroller with standard peripherals. Extensive data and instruction SRAM as well as device control and management logic are included in this multi-chip module (MCM).

The embedded AT40K FPGA core is a fully 3.3V PCI-compliant, SRAM-based FPGA with distributed 10 ns programmable synchronous/asynchronous, dual-port/single-port SRAM, 8 global clocks, Cache Logic ability (partially or fully reconfigurable without loss of data) and 5,000 to 40,000 usable gates.

**Table 1.** The AT94S Series Family

Device	AT94S05AL	AT94S10AL	AT94S40AL
Configuration Memory Size	512 Kbits	512 Kbits	1 Mbit
FPGA Gates	5K	10K	40K
FPGA Core Cells	256	576	2304
FPGA SRAM Bits	2048	4096	18432
FPGA Registers (Total)	436	846	2862
Maximum FPGA User I/O	95	143	287
AVR Programmable I/O Lines	8	16	16
Program SRAM Bytes	4K - 16K	20K - 32K	20K - 32K
Data SRAM Bytes	4K - 16K	4K - 16K	4K - 16K
Hardware Multiplier (8-bit)	Yes	Yes	Yes
2-wire Serial Interface	Yes	Yes	Yes
UARTs	2	2	2
Watchdog Timer	Yes	Yes	Yes
Timer/Counters	3	3	3
Real-time Clock	Yes	Yes	Yes
Typical AVR Throughput	@ 25 MHz	19 MIPS	19 MIPS
	@ 40 MHz	30 MIPS	30 MIPS
Operating Voltage	3.0 - 3.6V	3.0 - 3.6V	3.0 - 3.6V

Figure 1. AT94S Architecture



The embedded AVR core achieves throughputs approaching 1 MIPS per MHz by executing powerful instructions in a single-clock-cycle, and allows system designers to optimize power consumption versus processing speed. The AVR core is based on an enhanced RISC architecture that combines a rich instruction set with 32 general-purpose working registers. All 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code-efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers at the same clock frequency. The AVR executes out of on-chip SRAM. Both the FPGA configuration SRAM and AVR instruction code SRAM are automatically loaded at system power-up using Atmel's in-system programmable AT17 Series EEPROM configuration memories, which are part of the AT94S Multi-chip Module (MCM).

State-of-the-art FPSLIC design tools, System Designer™, were developed in conjunction with the FPSLIC architecture to help reduce overall time-to-market by integrating microcontroller development and debugging, FPGA development, place and route, and complete system co-verification in one easy-to-use software tool.

## Internal Architecture

For details of the AT94S Secure FPSLIC architecture, please refer to the AT94K FPSLIC datasheet and the AT17 Series Configuration Memory datasheet, available on the Atmel web site at <http://www.atmel.com>. This document only describes the differences between the AT94S Secure FPSLIC and the AT94K FPSLIC.

## FPSLIC and Configurator Interface

- Fully In-System Programmable and Re-programmable
- When Security Bit Set:
  - Data Verification Disabled
  - Data Transfer to FPSLIC not Externally Visible
  - Secured EEPROM Will Only Boot the FPSLIC Device or Respond to a Chip Erase
- When Security Bit Cleared:
  - Entire Chip Erase Performed
  - In-System Programming Enabled
  - Data Verification Enabled

External Data pins allow for In-System Programming of the device and setting of the EEPROM-based security bit. When the security bit is set (active) this programming connection will only respond to a device erase command. Data cannot be read out of the external programming/data pins when the security bit is set. The part can be re-programmed, but only after first being erased.

## Programming and Configuration Timing Characteristics

Atmel's Configurator Programming Software (CPS), available from the Atmel web site (<http://www.atmel.com/atmel/products/prod185.htm>), creates the programming algorithm for the embedded configurator; however, if you are planning to write your own software or use other means to program the embedded configurator, the section below includes the algorithm and other details.

## The FPSLIC Configurator

The FPSLIC Configurator is a serial EEPROM memory which is used to load programmable devices. This document describes the features needed to program the Configurator from within its programming mode (i.e., when  $\overline{\text{SER\_EN}}$  is driven Low).

Reference schematics are supplied for ISP applications.

## Serial Bus Overview

The serial bus is a two-wire bus; one wire (cSCK) functions as a clock and is provided by the programmer, the second wire (cSDA) is a bi-directional signal and is used to provide data and control information.

Information is transmitted on the serial bus in messages. Each MESSAGE is preceded by a Start Condition and ends with a Stop Condition. The message consists of an integer number of bytes, each byte consisting of 8 bits of data, followed by a ninth Acknowledge Bit. This Acknowledge Bit is provided by the recipient of the transmitted byte. This is possible because devices may only drive the cSDA line Low. The system must provide a small pull-up current (1 k $\Omega$  equivalent) for the cSDA line.

The MESSAGE FORMAT for read and write instructions consists of the bytes shown in "Bit Format" on page 5.

While writing, the programmer is responsible for issuing the instruction and data. While reading, the programmer issues the instruction and acknowledges the data from the Configurator as necessary.

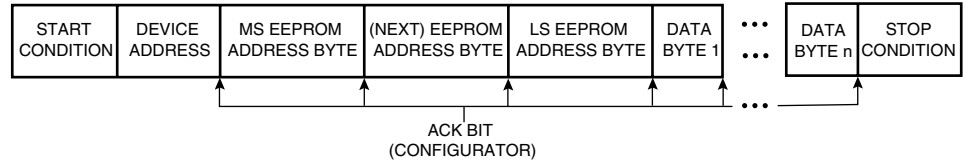
Again, the Acknowledge Bit is asserted on the cSDA line by the receiving device on a byte-by-byte basis.

The factory blanks devices to all zeros before shipping. The array cannot otherwise be “initialized” except by explicitly writing a known value to each location using the serial protocol described herein.

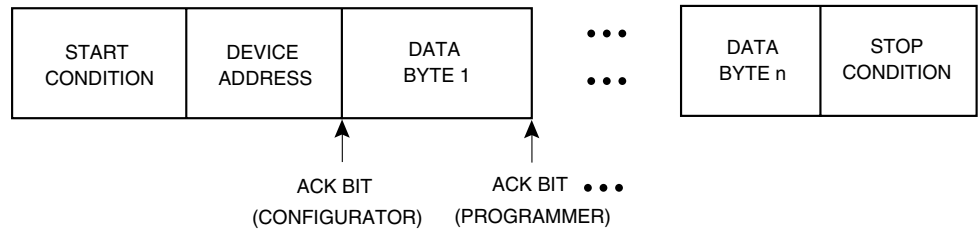
## Bit Format

Data on the cSDA pin may change only during the cSCK Low time; whereas Start and Stop Conditions are identified as transitions during the cSCK High time.

### Write Instruction Message Format



### Current Address Read (Extended to Sequential Read) Instruction Message Format



## Start and Stop Conditions

The Start Condition is indicated by a high-to-low transition of the cSDA line when the cSCK line is High. Similarly, the Stop Condition is generated by a low-to-high transition of the cSDA line when the cSCK line is High, as shown in Figure 2.

The Start Condition will return the device to the state where it is waiting for a Device Address (its normal quiescent mode).

The Stop Condition initiates an internally timed write signal whose maximum duration is  $t_{WR}$  (refer to AC Characteristics table for actual value). During this time, the Configurator must remain in programming mode (i.e.,  $\overline{SER\_EN}$  is driven Low). cSDA and cSCK lines are ignored until the cycle is completed. Since the write cycle typically completes in less than  $t_{WR}$  seconds, we recommend the use of “polling” as described in later sections. Input levels to all other pins should be held constant until the write cycle has been completed.

## Acknowledge Bit

The Acknowledge (ACK) Bit shown in Figure 2 is provided by the Configurator receiving the byte. The receiving Configurator can accept the byte by asserting a Low value on the cSDA line, or it can refuse the byte by asserting (allowing the signal to be externally pulled up to) a High value on the cSDA line. All bytes from accepted messages must be terminated by either an Acknowledge Bit or a Stop Condition. Following an ACK Bit, when the cSDA line is released during an exchange of control between the Configurator and the programmer, the cSDA line may be pulled High temporarily due to the open-collector output nature of the line. Control of the line must resume before the next rising edge of the clock.

## Bit Ordering Protocol

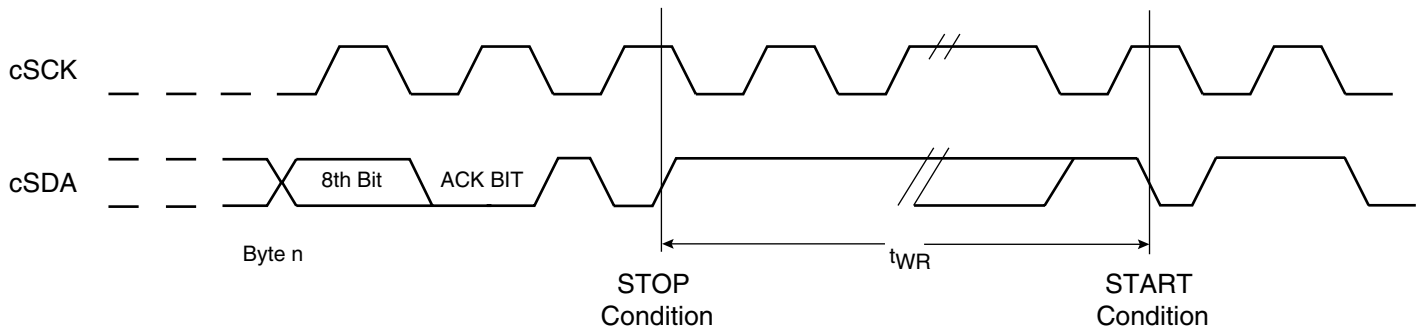
The most significant bit is the first bit of a byte transmitted on the cSDA line for the Device Address Byte and the EEPROM Address Bytes. It is followed by the lesser significant bits until the eighth bit, the least significant bit, is transmitted. However, for Data Bytes (both writing and reading), the first bit transmitted is the least significant bit. This protocol is shown in the diagrams below.

## Device Address Byte

The contents of the Device Address Byte are shown below, along with the order in which the bits are clocked into the device.

The  $\overline{CE}$  pin cannot be used for device selection in programming mode (i.e., when  $\overline{SER\_EN}$  is drive Low).

Figure 2. Start and Stop Conditions

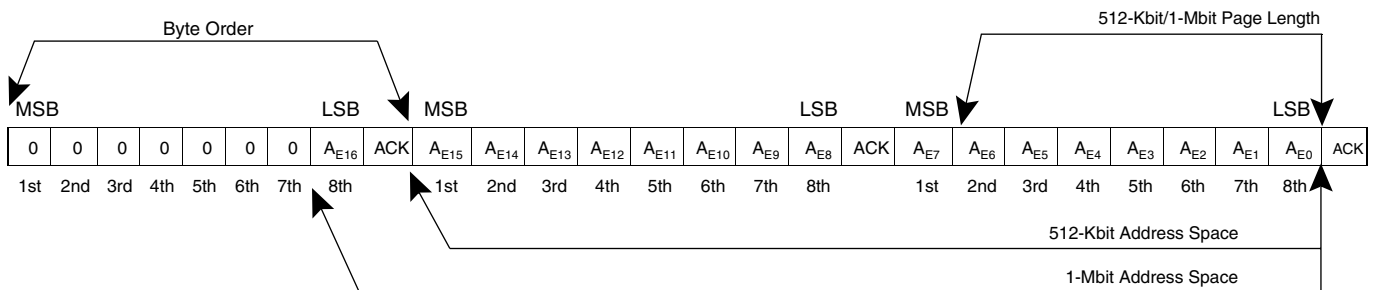


## Device Address Byte

MSB							LSB
1	0	1	0	0	1	1	R/ $\overline{W}$
1st	2nd	3rd	4th	5th	6th	7th	8th

Where: R/ $\overline{W}$  = 1 Read  
 = 0 Write

## EEPROM Address



The EEPROM Address consists of three bytes on the 512-Kbit and 1-Mbit parts. Each Address Byte is followed by an Acknowledge Bit (provided by the Configurator). These bytes define the normal address space of the Configurator. The order in which each byte is clocked into the Configurator is also indicated. Unused bits in an Address Byte must be set to "0". Exceptions to this are when reading Device and Manufacturer Codes.

## Programming Summary: Write to Whole Device

- Notes: 1. The 512-Kbit and 1-Mbit parts require three EEPROM address bytes; all three bytes must be individually ACK'd by the EEPROM.  
2. Data byte received/sent LSB to MSB.

### EEPROM Address is Defined as:

AT17LV512 0000 0000  $x_8x_7x_6x_5$   $x_4x_3x_2x_1$   $x_0$ 000 0000

AT17LV010 0000 000 $x_9$   $x_8x_7x_6x_5$   $x_4x_3x_2x_1$   $x_0$ 000 0000

Note: where  $X_n \dots X_0$  is (PAGE\_COUNT)\b

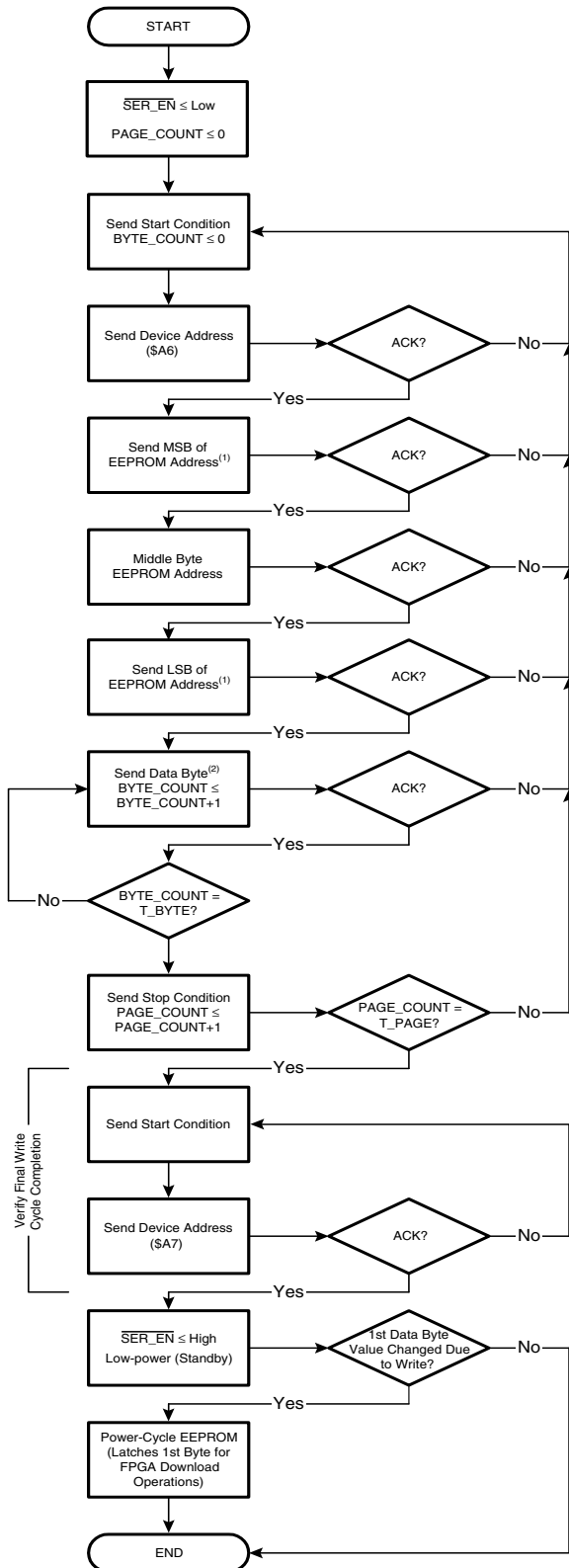
### T\_BYTE

AT17LV512/010 128

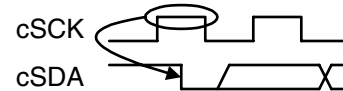
### T\_PAGE

AT17LV512 512

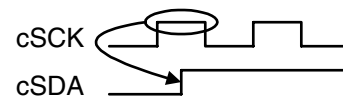
AT17LV010 1024



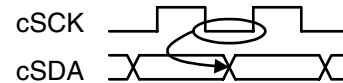
### START CONDITION



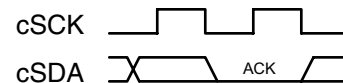
### STOP CONDITION



### DATA BIT



### ACK BIT



## Programming Summary: Read from Whole Device

- Notes:
1. The 512-Kbit and 1-Mbit parts require three EEPROM address bytes; all three bytes must be individually ACK'd by the EEPROM.
  2. Data byte received/sent LSB to MSB

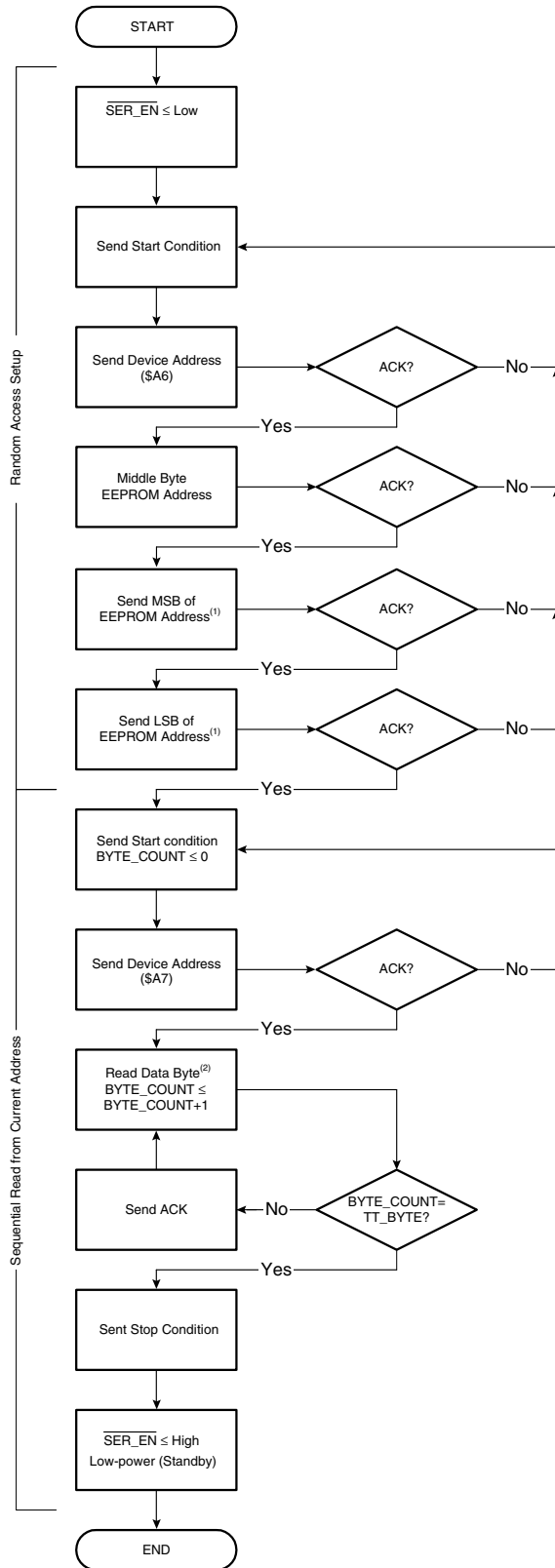
### EEPROM Address is Defined as:

AT17LV512/010 00 00 00 \h

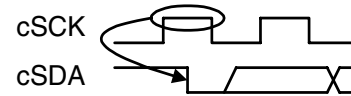
### TT\_BYTE

AT17LV512 65536 \d

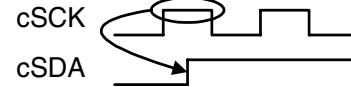
AT17LV010 131072 \d



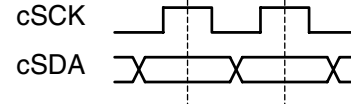
### START CONDITION



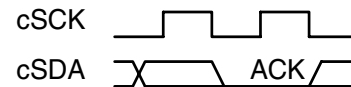
### STOP CONDITION



### SAMPLE DATA BIT



### ACK BIT





## Data Byte

LSB							MSB
D0	D1	D2	D3	D4	D5	D6	D7
1st	2nd	3rd	4th	5th	6th	7th	8th

The organization of the Data Byte is shown above. Note that in this case, the Data Byte is clocked into the device LSB first and MSB last.

## Writing

Writing to the normal address space takes place in pages. A page is 128-bytes long in the 512-Kbit and 1-Mbit parts. The page boundaries are, respectively, addresses where  $A_{E0}$  down to  $A_{E05}$  are all zero, and  $A_{E6}$  down to  $A_{E0}$  are all zero. Writing can start at any address within a page and the number of bytes written must be 128 for the 512-Kbit and 1-Mbit parts. The first byte is written at the transmitted address. The address is incremented in the Configurator following the receipt of each Data Byte. Only the lower bits of the address (6, 7 or 8, depending on the page length) are incremented. Thus, after writing to the last byte address within the given page, the address will roll over to the first byte address of the same page. A Write Instruction consists of:

```

a Start Condition
a Device Address Byte with  $R/\bar{W} = 0$ 
  An Acknowledge Bit from the Configurator
MS Byte of the EEPROM Address
  An Acknowledge Bit from the Configurator
Next Byte of the EEPROM Address
  An Acknowledge Bit from the Configurator
LS Byte of EEPROM Address
  An Acknowledge Bit from the Configurator
One or more Data Bytes (sent to the
Configurator)
  Each followed by an Acknowledge Bit from the
  Configurator
a Stop Condition
  
```

**WRITE POLLING:** On receipt of the Stop Condition, the Configurator enters an internally-timed write cycle. While the Configurator is busy with this write cycle, it will not acknowledge any transfers. The programmer can start the next page write by sending the Start Condition followed by the Device Address, in effect polling the Configurator. If this is not acknowledged, then the programmer should abandon the transfer without asserting a Stop Condition. The programmer can then repeatedly initiate a write instruction as above, until an acknowledge is received. When the Acknowledge Bit is received, the write instruction should continue by sending the first EEPROM Address Byte to the Configurator.

An alternative to write polling would be to wait a period of  $t_{WR}$  before sending the next page of data or exiting the programming mode. All signals must be maintained during the entire write cycle.

## Reading

Read instructions are initiated similarly to write instructions. However, with the  $R/\bar{W}$  bit in the Device Address set to one. There are three variants of the read instruction: current address read, random read and sequential read.

For all reads, it is important to understand that the internal Data Byte address counter maintains the last address accessed during the previous read or write operation, incremented by one. This address remains valid between operations as long as the chip power is maintained and the device remains in 2-wire access mode (i.e.,  $\overline{SER\_EN}$  is driven Low). If the last operation was a read at address  $n$ , then the current address would be  $n + 1$ . If the final operation was a write at address  $n$ , then the current address would again be  $n + 1$  with one exception. If address  $n$  was the last byte address in the page, the incremented address  $n + 1$  would “roll over” to the first byte address on the next page.

**CURRENT ADDRESS READ:** Once the Device Address (with the  $R/\bar{W}$  select bit set to High) is clocked in and acknowledged by the Configurator, the Data Byte at the current address is serially clocked out by the Configurator in response to the clock from the programmer. The programmer generates a Stop Condition to accept the single byte of data and terminate the read instruction.

```
A Current Address Read instruction consists of
  a Start Condition
  a Device Address with  $R/\bar{W} = 1$ 
    An Acknowledge Bit from the Configurator
  a Data Byte from the Configurator
  a Stop Condition from the programmer.
```

**RANDOM READ:** A Random Read is a Current Address Read preceded by an aborted write instruction. The write instruction is only initiated for the purpose of loading the EEPROM Address Bytes. Once the Device Address Byte and the EEPROM Address Bytes are clocked in and acknowledged by the Configurator, the programmer immediately initiates a Current Address Read.

A Random Address Read instruction consists of :

```
a Start Condition
a Device Address with  $R/\bar{W} = 0$ 
  An Acknowledge Bit from the Configurator
MS Byte of the EEPROM Address
  An Acknowledge Bit from the Configurator
Next Byte of the EEPROM Address
  An Acknowledge Bit from the Configurator
LS Byte of EEPROM Address
  An Acknowledge bit from the Configurator
a Start Condition
a Device Address with  $R/\bar{W} = 1$ 
  An Acknowledge Bit from the Configurator
a Data Byte from the Configurator
a Stop Condition from the programmer.
```

**SEQUENTIAL READ:** Sequential Reads follow either a Current Address Read or a Random Address Read. After the programmer receives a Data Byte, it may respond with an Acknowledge Bit. As long as the Configurator receives an Acknowledge Bit, it will continue to increment the Data Byte address and serially clock out sequential Data Bytes until the memory address limit is reached. The Sequential Read instruction is terminated when the programmer does not respond with an Acknowledge Bit but instead generates a Stop Condition following the receipt of a Data Byte.

## Programmer Functions

The following programmer functions are supported while the Configurator is in programming mode (i.e., when  $\overline{\text{SER\_EN}}$  is driven Low):

1. Read the Manufacturer's Code and the Device Code (optional for ISP).
2. Program the device.
3. Verify the device.

In the order given above, they are performed in the following manner.

## Reading Manufacturer's and Device Codes

On AT17LV512/010 Configurators, the sequential reading of these bytes are accomplished by performing a Random Read at EEPROM Address 040000H.

The correct codes are:

Manufacturers Code	-Byte 0	1E
Device Code	- Byte 1	37 AT17LV512
		F7 AT17LV010

Note: The Manufacturer's Code and Device Code are read using the byte ordering specified for Data Bytes; i.e., LSB first, MSB last.

## Programming the Device

All the bytes in a given page must be written. The page access order is not important but it is suggested that the Configurator be written sequentially from address 0. Writing is accomplished by using the cSDA and cSCK pins.

## Important Note on AT94S Series Configurators Programming

The first byte of data will not be cached for read back during FPGA Configuration (i.e., when  $\overline{\text{SER\_EN}}$  is driven High) until the Configurator is power-cycled.

## Verifying the Device

All bytes in the Configurator should be read and compared to their intended values. Reading is done using the cSDA and cSCK pins.

## In-System Programming Applications

The AT94S Series Configurators are in-system (re)programmable (ISP). The example shown on the following page supports the following programmer functions:

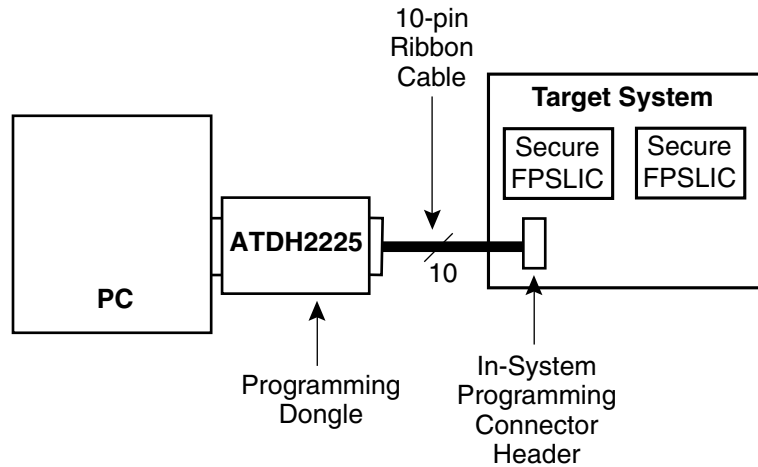
1. Read the Manufacturer's Code and the Device Code.
2. Program the device.
3. Verify the device data.

While Atmel's Secure FPSLIC Configurators can be programmed from various sources (e.g., on-board microcontrollers or PLDs), the applications shown here are designed to facilitate users of our ATDH2225 Configurator Programming Cable. The typical system setup is shown in Figure 3.

The pages within the configuration EEPROM can be selectively rewritten.

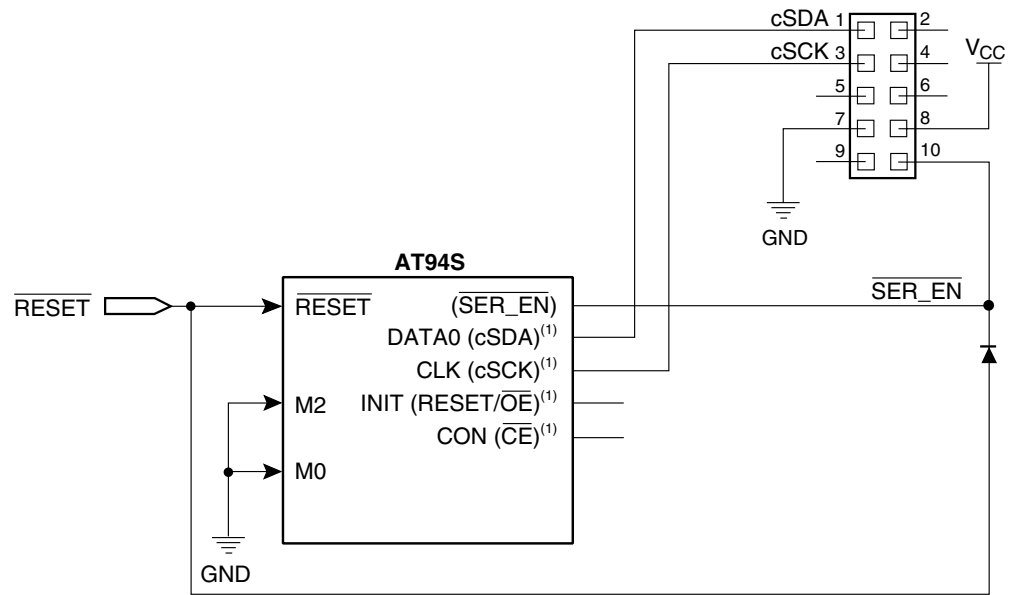
This document is limited to example implementations for Atmel's AT94S application.

**Figure 3.** Typical System Setup



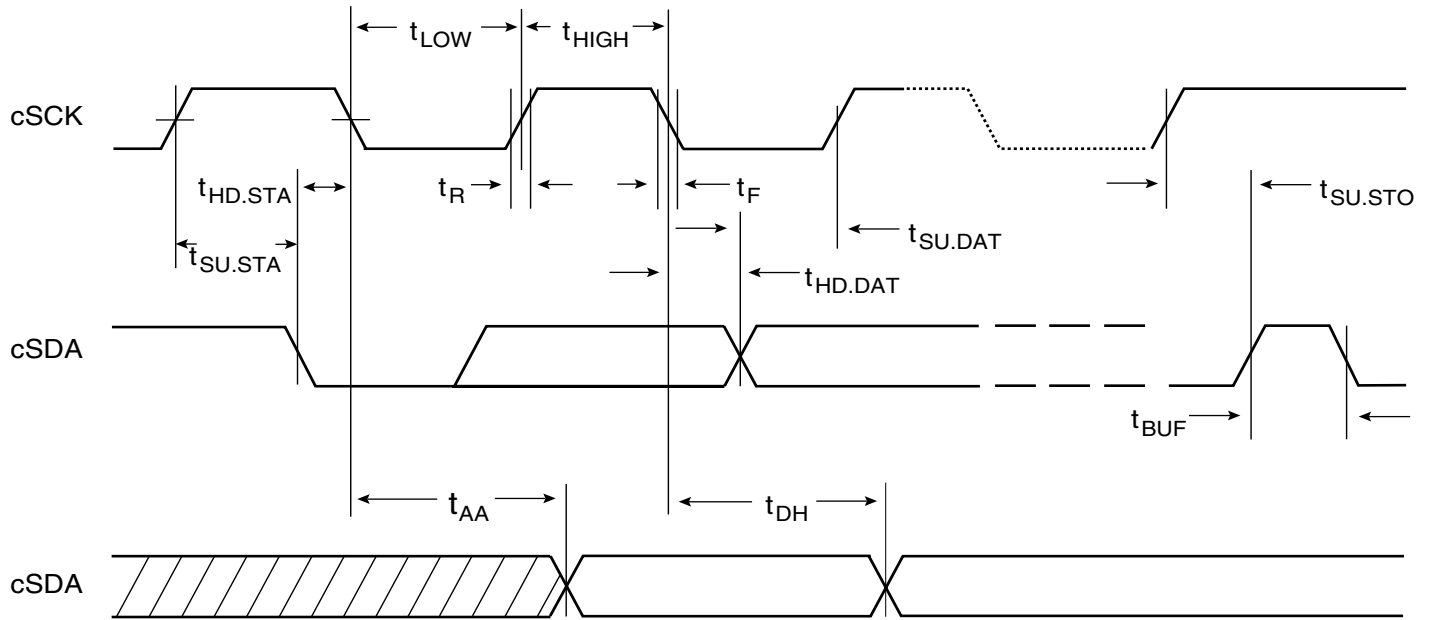
The diode connection between the AT94S'  $\overline{\text{RESET}}$  pin and the  $\overline{\text{SER\_EN}}$  signal allows the external programmer to force the FPGA into a reset state during ISP. This eliminates the potential for contention on the cSCK line. The pull-up resistors required on the lines to  $\overline{\text{RESET}}$ , CON and INIT are present on the inputs (internally) to the AT94S FPSLIC, see Figure 4.

**Figure 4.** ISP of the AT17LV512/010 in an AT94S FPSLIC Application



Note: 1. Configurator signal names are shown in parenthesis.

**Figure 5. Serial Data Timing Diagram**



## DC Characteristics<sup>(1)</sup>

$$V_{CC} = 3.3V \pm 5\%, T_A = -40^{\circ}C - 80^{\circ}C^{(2)(3)(4)}$$

Symbol	Parameter	Test Condition	Min	Typ	Max	Units
$V_{CC}$	Supply Voltage		3.0	3.3	3.6	V
$I_{CC}$	Supply Current	$V_{CC} = 3.6$		2	3	mA
$I_{LL}$	Input Leakage Current	$V_{IN} = V_{CC}$ or $V_{SS}$		0.10	10	$\mu A$
$I_{LO}$	Output Leakage Current	$V_{OUT} = V_{CC}$ or $V_{SS}$		0.05	10	$\mu A$
$V_{IH}$	High-level Input Voltage		$V_{CC} \times 0.7$		$V_{CC} + 0.5$	V
$V_{IL}$	Low-level Input Voltage		-0.5		0.2	V
$V_{OL}$	Output Low-level Voltage	$I_{OL} = 2.1$ mA			0.4	V

- Notes:
1. Specific to programming mode (i.e., when  $\overline{SER\_EN}$  is driven Low)
  2. Commercial temperature range  $0^{\circ}C - 70^{\circ}C$
  3. Industrial temperature range  $-40^{\circ}C - 80^{\circ}C$
  4. This parameter is characterized and is not 100% tested.

## AC Characteristics<sup>(1)</sup>

$$V_{CC} = 3.3V \pm 5\%, T_A = -40^{\circ}C - 80^{\circ}C^{(2)(3)(4)}$$

Symbol	Parameter	Min	Max	Units
$f_{CLOCK}$	Clock Frequency, Clock		100	KHz
$t_{LOW}$	Clock Pulse Width Low	4		$\mu s$
$t_{HIGH}$	Clock Pulse Width High	4		$\mu s$
$t_{AA}$	Clock Low to Data Out Valid	0.1	1	$\mu s$
$t_{BUF}$	Time the Bus Must Be Free Before a New Transmission Can Start	4.5		$\mu s$
$t_{HD,STA}$	Start Hold Time	2		$\mu s$
$t_{SU,STA}$	Start Setup Time	2		$\mu s$
$t_{HD,DAT}$	Data In Hold Time	0		$\mu s$
$t_{SU,DAT}$	Data In Setup Time	0.2		$\mu s$
$t_R$	Inputs Rise Time		0.3	$\mu s$
$t_F$	Inputs Fall Time		0.3	$\mu s$
$t_{SU,STO}$	Stop Setup Time	2		$\mu s$
$t_{DH}$	Data Out Hold Time	0.1		$\mu s$
$t_{WR}$	Write Cycle Time		20	ms

- Notes:
1. Specific to programming mode (i.e., when  $\overline{SER\_EN}$  is driven Low)
  2. Commercial temperature range  $0^{\circ}C - 70^{\circ}C$
  3. Industrial temperature range  $-40^{\circ}C - 80^{\circ}C$
  4. This parameter is characterized and is not 100% tested.

## Secure FPSLIC Configurator Pin Configurations

256-pin CABGA	Name	I/O	Description
D16	cSDA	I/O	Three-state DATA output for configuration. Open-collector bi-directional pin for programming.
C16	cSCK	I/O	CLOCK input/output. Used to increment the internal address and bit counter for reading and programming.
K9	RESET/ $\overline{OE}$	I	RESET/ $\overline{OE}$ input (when $\overline{SER\_EN}$ is High). A Low level on both the $\overline{CE}$ and RESET/ $\overline{OE}$ inputs enables the data output driver. A High level on RESET/ $\overline{OE}$ resets both the address and bit counters. The logic polarity of this input is programmable as either RESET/ $\overline{OE}$ or $\overline{RESET/OE}$ . This document describes the pin as RESET/ $\overline{OE}$ .
N16	$\overline{CE}$	I	Chip Enable input. Used for device selection only when $\overline{SER\_EN}$ is High. A Low level on both $\overline{CE}$ and $\overline{OE}$ enables the data output driver. A High level on $\overline{CE}$ disables both the address and bit counters and forces the device into a low-power mode. Note this pin will not enable/disable the device in the 2-wire Serial mode (i.e., when $\overline{SER\_EN}$ is driven Low).
M5	$\overline{SER\_EN}$	I	Serial enable is normally High during FPGA loading operations. Bringing $\overline{SER\_EN}$ Low enables the programming mode.

## Security Bit

Once the security bit is programmed, data will no longer output from the normal data pad. Once the fuse is set, any attempt to erase the fuse will cause the configurator to erase all of its contents.

### AT17LV512/010 Security Bit Programming

#### *Disabling the Security Bit*

Write 4 bytes "00 00 00 00" to addresses 800000-800003 twice, without a power cycle in between, using the previously defined 2-wire write algorithm.

#### *Enabling the Security Bit*

Write 4 bytes "FF FF FF FF" to addresses 800000-800003 using the previously defined 2-wire write algorithm.

#### *Verifying the Security Bit*

Read 4 bytes of data to addresses 800000-800003 twice using the previously defined 2-wire Random Read algorithm. If the data is "FF FF FF FF", the security bit has been enabled. If the data is "00 00 00 00", the security bit has been disabled.

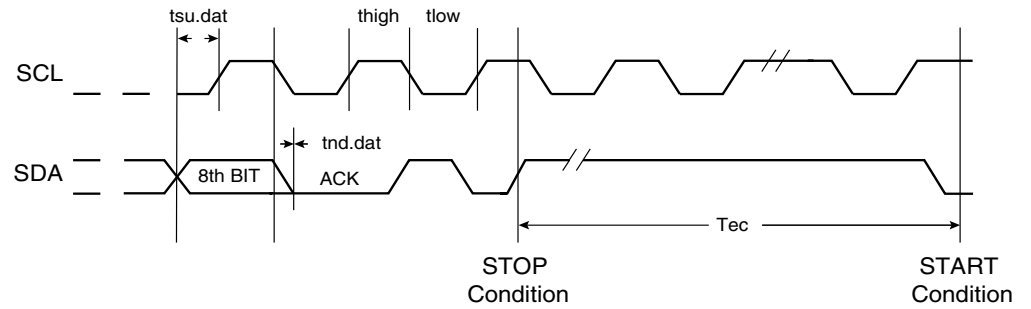
## Chip Erase Timing

The entire device can be erased at once by writing to a specific address. This operation will erase the entire array. See Table 2 for specifics on the write algorithm.

**Table 2.** Chip Erase Cycle Characteristics

Symbol	Parameter
Tec	Chip Erase Cycle Time (25 ms)

**Figure 6.** Chip Erase Timing Diagram





## Packaging and Pin List information

**Table 3.** Part and Package Combinations Available

Part #	Package	AT94S05	AT94S10	AT94S40
BG256	DG	93	137	162

**Table 4.** AT94S Pin List

AT94S05 <sup>(1)</sup> 96 FPGA I/O	AT94S10 144 FPGA I/O	AT94S40 288 FPGA I/O	Package
			Chip Array 256 CABGA
<b>FPSLIC Array</b>			
I/O1, GCK1 (A16)	I/O1, GCK1 (A16)	I/O1, GCK1 (A16)	A1
I/O2 (A17)	I/O2 (A17)	I/O2 (A17)	D4
I/O3	I/O3	I/O3	D3
I/O4	I/O4	I/O4	B1
I/O5 (A18)	I/O5 (A18)	I/O5 (A18)	C2
I/O6 (A19)	I/O6 (A19)	I/O6 (A19)	C1
		I/O7	
		I/O8	
NC	NC	I/O9	D2
NC	NC	I/O10	D1
		I/O11	
		I/O12	
		I/O13	
		I/O14	
I/O7	I/O7	I/O15	E3
I/O8	I/O8	I/O16	E4
NC	I/O9	I/O17	E2
NC	I/O10	I/O18	E1
		I/O19	
		I/O20	
NC	I/O11	I/O21	F4
NC	I/O12	I/O22	F3
		I/O23	
		I/O24	
I/O9, FCK1	I/O13, FCK1	I/O25, FCK1	F1
I/O10	I/O14	I/O26	G7
I/O11 (A20)	I/O15 (A20)	I/O27 (A20)	G6



**Table 4. AT94S Pin List (Continued)**

AT94S05 <sup>(1)</sup> 96 FPGA I/O	AT94S10 144 FPGA I/O	AT94S40 288 FPGA I/O	Package
			Chip Array 256 CABGA
I/O12 (A21)	I/O16 (A21)	I/O28 (A21)	G4
NC	I/O17	I/O29	G5
NC	I/O18	I/O30	G2
		I/O31	
		I/O32	
		I/O33	
		I/O34	
NC	NC	I/O35	G1
NC	NC	I/O36	H7
		I/O37	
		I/O38	
NC	NC	I/O39	H6
NC	NC	I/O40	H5
NC	I/O19	I/O41	H3
NC	I/O20	I/O42	H4
I/O13	I/O21	I/O43	H2
I/O14	I/O22	I/O44	H1
		I/O45	
		I/O46	
I/O15 (A22)	I/O23 (A22)	I/O47 (A22)	J7
I/O16 (A23)	I/O24 (A23)	I/O48 (A23)	J1
I/O17 (A24)	I/O25 (A24)	I/O49 (A24)	J4
I/O18 (A25)	I/O26 (A25)	I/O50 (A25)	J5
		I/O51	
		I/O52	
I/O19	I/O27	I/O53	J6
I/O20	I/O28	I/O54	J8
NC	I/O29	I/O55	K1
NC	I/O30	I/O56	K2
		I/O57	
		I/O58	
		I/O59	
		I/O60	
NC	NC	I/O61	K4

**Table 4. AT94S Pin List (Continued)**

AT94S05 <sup>(1)</sup> 96 FPGA I/O	AT94S10 144 FPGA I/O	AT94S40 288 FPGA I/O	Package
			Chip Array 256 CABGA
NC	NC	I/O62	K5
		I/O63	
		I/O64	
NC	NC	I/O65	K6
NC	NC	I/O66	L1
NC	I/O31	I/O67	L2
NC	I/O32	I/O68	L5
I/O21 (A26)	I/O33 (A26)	I/O69 (A26)	L4
I/O22 (A27)	I/O34 (A27)	I/O70 (A27)	M1
I/O23	I/O35	I/O71	M2
I/O24, FCK2	I/O36, FCK2	I/O72, FCK2	N1
		I/O73	
		I/O74	
	I/O37	I/O75	
	I/O38	I/O76	
		I/O77	
		I/O78	
		I/O79	
		I/O80	
NC	I/O39	I/O81	M3
NC	I/O40	I/O82	N2
I/O25	I/O41	I/O83	
I/O26	I/O42	I/O84	
		I/O85	
		I/O86	
		I/O87	
		I/O88	
I/O27 (A28)	I/O43 (A28)	I/O89 (A28)	P1
I/O28	I/O44	I/O90	P2
		I/O91	
		I/O92	
I/O29	I/O45	I/O93	R1
I/O30	I/O46	I/O94	N3
I/O31 ( $\overline{\text{OTS}}$ )	I/O47 ( $\overline{\text{OTS}}$ )	I/O95 ( $\overline{\text{OTS}}$ )	T1



**Table 4. AT94S Pin List (Continued)**

AT94S05 <sup>(1)</sup> 96 FPGA I/O	AT94S10 144 FPGA I/O	AT94S40 288 FPGA I/O	Package
			Chip Array 256 CABGA
I/O32, GCK2 (A29)	I/O48, GCK2 (A29)	I/O96, GCK2 (A29)	P3
$\overline{\text{AVRRESET}}$	$\overline{\text{AVRRESET}}$	$\overline{\text{AVRRESET}}$	R2
M0	M0	M0	R3
<b>FPSLIC Array</b>			
M2	M2	M2	T3
I/O33, GCK3	I/O49, GCK3	I/O97, GCK3	R4
I/O34 (HDC/TDI)	I/O50 (HDC/TDI)	I/O98 (HDC/TDI)	T4
I/O35	I/O51	I/O99	N5
I/O36	I/O52	I/O100	P5
$\overline{\text{SER\_EN}}$	$\overline{\text{SER\_EN}}$	$\overline{\text{SER\_EN}}$	M5
I/O38 (LDC/TDO)	I/O54 (LDC/TDO)	I/O102 (LDC/TDO)	R5
		I/O103	
		I/O104	
		I/O105	
		I/O106	
NC	NC	I/O107	T5
NC	NC	I/O108	M6
I/O39	I/O55	I/O109	P6
I/O40	I/O56	I/O110	R6
NC	I/O57	I/O111	L6
NC	I/O58	I/O112	T6
		I/O113	
		I/O114	
		I/O115	
		I/O116	
	I/O59	I/O117	
	I/O60	I/O118	
		I/O119	
		I/O120	
I/O41	I/O61	I/O121	M7
I/O42	I/O62	I/O122	N7
I/O43 (TMS)	I/O63 (TMS)	I/O123 (TMS)	P7
I/O44 (TCK)	I/O64 (TCK)	I/O124 (TCK)	R7
NC	I/O65	I/O125	K7

**Table 4. AT94S Pin List (Continued)**

AT94S05 <sup>(1)</sup> 96 FPGA I/O	AT94S10 144 FPGA I/O	AT94S40 288 FPGA I/O	Package
			Chip Array 256 CABGA
NC	I/O66	I/O126	K8
		I/O127	
		I/O128	
		I/O129	
		I/O130	
		I/O131	
		I/O132	
		I/O133	
		I/O134	
NC	I/O67	I/O135	M8
NC	I/O68	I/O136	R8
I/O45	I/O69	I/O137	P8
I/O46	I/O70	I/O138	N8
		I/O139	
		I/O140	
		I/O141	
		I/O142	
I/O47 (TD7)	I/O71 (TD7)	I/O143 (TD7)	L8
I/O48 (InitErr) RESET/ŌE	I/O72 (InitErr) RESET/ŌE	I/O144 (InitErr) RESET/ŌE	K9
I/O49 (TD6)	I/O73 (TD6)	I/O145 (TD6)	P9
I/O50 (TD5)	I/O74 (TD5)	I/O146 (TD5)	N9
		I/O147	
		I/O148	
		I/O149	
		I/O150	
I/O51	I/O75	I/O151	M9
I/O52	I/O76	I/O152	L9
NC	I/O77	I/O153	J9
NC	I/O78	I/O154	T10
		I/O155	
		I/O156	
		I/O157	
		I/O158	
		I/O159	



**Table 4. AT94S Pin List (Continued)**

AT94S05 <sup>(1)</sup> 96 FPGA I/O	AT94S10 144 FPGA I/O	AT94S40 288 FPGA I/O	Package
			Chip Array 256 CABGA
		I/O160	
		I/O161	
		I/O162	
NC	I/O79	I/O163	P10
NC	I/O80	I/O164	N10
I/O53 (TD4)	I/O81 (TD4)	I/O165 (TD4)	L10
I/O54 (TD3)	I/O82 (TD3)	I/O166 (TD3)	T11
I/O55	I/O83	I/O167	R11
I/O56	I/O84	I/O168	M11
NC	NC	I/O169	N11
NC	NC	I/O170	T12
NC	I/O85	I/O171	R12
NC	I/O86	I/O172	T13
		I/O173	
		I/O174	
		I/O175	
		I/O176	
NC	I/O87	I/O177	N12
NC	I/O88	I/O178	P12
I/O57	I/O89	I/O179	R13
I/O58	I/O90	I/O180	T14
NC	NC	I/O181	N13
NC	NC	I/O182	P13
I/O59 (TD2)	I/O91 (TD2)	I/O183 (TD2)	T16
I/O60 (TD1)	I/O92 (TD1)	I/O184 (TD1)	P14
		I/O185	
		I/O186	
		I/O187	
		I/O188	
I/O61	I/O93	I/O189	R16
I/O62	I/O94	I/O190	P15
I/O63 (TD0)	I/O95 (TD0)	I/O191 (TD0)	N14
I/O64, GCK4	I/O96, GCK4	I/O192, GCK4	P16
CON/ $\overline{\text{CE}}$	CON/ $\overline{\text{CE}}$	CON/ $\overline{\text{CE}}$	N16

**Table 4. AT94S Pin List (Continued)**

AT94S05 <sup>(1)</sup> 96 FPGA I/O	AT94S10 144 FPGA I/O	AT94S40 288 FPGA I/O	Package
			Chip Array 256 CABGA
<b>FPSLIC Array</b>			
$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	M14
PE0	PE0	PE0	M12
PE1	PE1	PE1	M15
PD0	PD0	PD0	M16
PD1	PD1	PD1	L12
PE2	PE2	PE2	L15
PD2	PD2	PD2	L11
NC	NC	NC	E12
PD3	PD3	PD3	K11
PD4	PD4	PD4	K12
PE3	PE3	PE3	K14
$\overline{\text{CS0}}$	$\overline{\text{CS0}}$	$\overline{\text{CS0}}$	K15
SDA	SDA	SDA	J10
SCL	SCL	SCL	J12
PD5	PD5	PD5	J14
PD6	PD6	PD6	J13
PE4	PE4	PE4	J16
PE5	PE5	PE5	J11
PE6	PE6	PE6	H15
PE7 ( $\overline{\text{CHECK}}$ )	PE7 ( $\overline{\text{CHECK}}$ )	PE7 ( $\overline{\text{CHECK}}$ )	H14
PD7	PD7	PD7	H13
INTP0	INTP0	INTP0	H12
XTAL1	XTAL1	XTAL1	G15
XTAL2	XTAL2	XTAL2	G14
RX0	RX0	RX0	G12
TX0	TX0	TX0	G11
INTP1	INTP1	INTP1	F15
INTP2	INTP2	INTP2	F14
TOSC1	TOSC1	TOSC1	E16
TOSC2	TOSC2	TOSC2	E15
RX1	RX1	RX1	E14
TX1	TX1	TX1	E13
DATA0/cSDA	DATA0/cSDA	DATA0/cSDA	D16

**Table 4. AT94S Pin List (Continued)**

AT94S05 <sup>(1)</sup> 96 FPGA I/O	AT94S10 144 FPGA I/O	AT94S40 288 FPGA I/O	Package
			Chip Array 256 CABGA
INTP3 ( $\overline{\text{CSOUT}}$ )	INTP3 ( $\overline{\text{CSOUT}}$ )	INTP3 ( $\overline{\text{CSOUT}}$ )	D15
CCLK/cSCK	CCLK/cSCK	CCLK/cSCK	C16
I/O65:96 Are Unbonded	I/O97:144 Are Unbonded	I/O193:288 Are Unbonded	
<b>FPSLIC Array</b>			
Testclock	Testclock	Testclock	C15
I/O97 (A0)	I/O145 (A0)	I/O289 (A0)	C14
I/O98, GCK7 (A1)	I/O146, GCK7 (A1)	I/O290, GCK7 (A1)	B15
I/O99	I/O147	I/O291	A16
I/O100	I/O148	I/O292	D13
		I/O293	
		I/O294	
NC	NC	I/O295	C13
NC	NC	I/O296	B14
I/O101 ( $\overline{\text{CS1}}$ , A2)	I/O149 ( $\overline{\text{CS1}}$ , A2)	I/O297 ( $\overline{\text{CS1}}$ , A2)	A15
I/O102 (A3)	I/O150 (A3)	I/O298 (A3)	A14
		I/O299	
		I/O300	
I/O104	I/O151	I/O301	Shared with Testclock
NC	I/O152	I/O302	D12
I/O103	I/O153	I/O303	C12
NC	I/O154	I/O304	A13
NC	NC	I/O305	B12
		I/O306	
		I/O307	
		I/O308	
NC	I/O155	I/O309	A12
NC	I/O156	I/O310	E11
NC	NC	I/O311	C11
NC	NC	I/O312	D11
I/O105	I/O157	I/O313	A11
I/O106	I/O158	I/O314	F10
NC	I/O159	I/O315	E10
NC	I/O160	I/O316	D10



**Table 4. AT94S Pin List (Continued)**

AT94S05 <sup>(1)</sup> 96 FPGA I/O	AT94S10 144 FPGA I/O	AT94S40 288 FPGA I/O	Package
			Chip Array 256 CABGA
NC	NC	I/O317	C10
NC	NC	I/O318	B10
		I/O319	
		I/O320	
		I/O321	
		I/O322	
		I/O323	
		I/O324	
I/O107 (A4)	I/O161 (A4)	I/O325 (A4)	A10
I/O108 (A5)	I/O162 (A5)	I/O326 (A5)	G10
NC	I/O163	I/O327	G9
NC	I/O164	I/O328	F9
I/O109	I/O165	I/O329	E9
I/O110	I/O166	I/O330	C9
		I/O331	
		I/O332	
		I/O333	
		I/O334	
I/O111 (A6)	I/O167 (A6)	I/O335 (A6)	B9
I/O112 (A7)	I/O168 (A7)	I/O336 (A7)	A9
I/O113 (A8)	I/O169 (A8)	I/O337 (A8)	A8
I/O114 (A9)	I/O170 (A9)	I/O338 (A9)	B8
		I/O339	
		I/O340	
		I/O341	
		I/O342	
I/O115	I/O171	I/O343	C8
I/O116	I/O172	I/O344	D8
NC	I/O173	I/O345	E8
NC	I/O174	I/O346	F8
I/O117 (A10)	I/O175 (A10)	I/O347 (A10)	H8
I/O118 (A11)	I/O176 (A11)	I/O348 (A11)	A7
NC	NC	I/O349	C7
NC	NC	I/O350	D7



**Table 4. AT94S Pin List (Continued)**

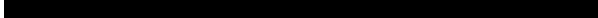
AT94S05 <sup>(1)</sup> 96 FPGA I/O	AT94S10 144 FPGA I/O	AT94S40 288 FPGA I/O	Package
			Chip Array 256 CABGA
		I/O351	
		I/O352	
		I/O353	
		I/O354	
		I/O355	
		I/O356	
NC	I/O177	I/O357	F7
NC	I/O178	I/O358	A6
I/O119	I/O179	I/O359	F6
I/O120	I/O180	I/O360	B6
		I/O361	
		I/O362	
NC	I/O181	I/O363	D6
NC	I/O182	I/O364	E6
		I/O365	
		I/O366	
		I/O367	
		I/O368	
I/O121	I/O183	I/O369	A5
I/O122	I/O184	I/O370	B5
I/O123 (A12)	I/O185 (A12)	I/O371 (A12)	E5
I/O124 (A13)	I/O186 (A13)	I/O372 (A13)	C5
		I/O373	
		I/O374	
		I/O375	
		I/O376	
		I/O377	
		I/O378	
NC	I/O187	I/O379	A4
NC	I/O188	I/O380	B4
I/O125	I/O189	I/O381	A3
I/O126	I/O190	I/O382	C4
I/O127 (A14)	I/O191 (A14)	I/O383 (A14)	B3
I/O128, GCK8 (A15)	I/O192, GCK8 (A15)	I/O384, GCK8 (A15)	A2

Note: 1. Available Q2/2002.

**Table 5.** 256 CABGA  $V_{DD}$ ,  $V_{CC}$  and GND Pins<sup>(1)</sup>

$V_{DD}$	$V_{CC}$	GND
B2, G13, R14, G8, H10, J3, K13, L3, M10, T7	D14, F12, P4, G3, H9, E7, K10, L13, M13, T9	B11, B13, B16, B7, C3, C6, D5, D9, F11, F13, T15, F16, F2, F5, G16, H11, H16, J15, J2, K16, K3, T2, L14, L16, L7, M4, N15, N4, N6, P11, R9, R10, R15, T8

Note: 1. For power rail support for product migration to lower-power devices, refer to the “Designing in Split Power Supply Support for AT94KAL/AX and AT94SAL/AX Devices” application note (doc2308.pdf), available on the Atmel web site, at <http://www.atmel.com/atmel/products/prod318.htm>.



**Thermal Coefficient Table**

Package Style	Lead Count	Theta J-A [ $^{\circ}$ C/W] 0 LFPM	Theta J-A [ $^{\circ}$ C/W] 225 LFPM	Theta J-A [ $^{\circ}$ C/W] 500 LFPM
CABGA	256	27	23	20



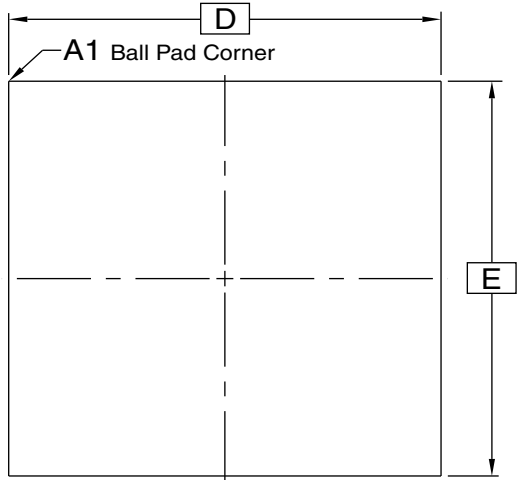
## Ordering Information

Usable Gates	Speed Grade	Ordering Code	Package	Operation Range
5,000	-25 MHz	AT94S05AL-25DGC	256ZA	Commercial (0°C - 70°C)
		AT94S05AL-25DGI	256ZA	Industrial (-40°C - 85°C)
10,000	-25 MHz	AT94S10AL-25DGC	256ZA	Commercial (0°C - 70°C)
		AT94S10AL-25DGI	256ZA	Industrial (-40°C - 85°C)
40,000	-25 MHz	AT94S40AL-25DGC	256ZA	Commercial (0°C - 70°C)
		AT94S40AL-25DGI	256ZA	Industrial (-40°C - 85°C)

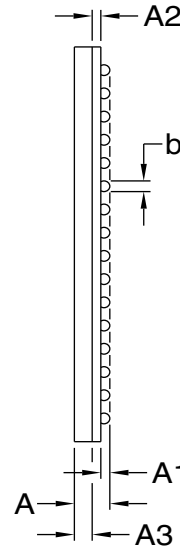
Package Type	
<b>256ZA</b>	256-ball, Chip Array Ball Grid Array Package (CABGA)

# Packaging Information

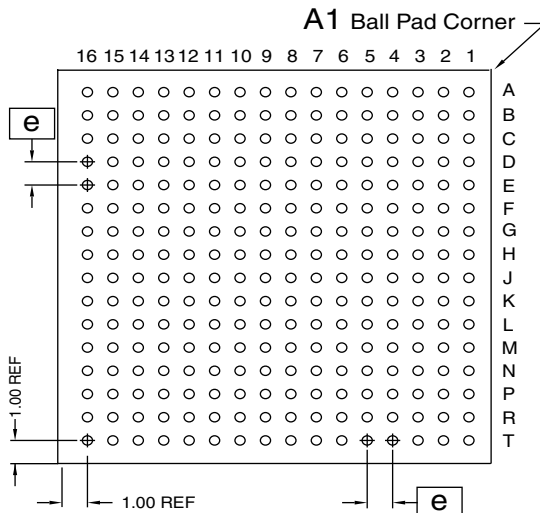
## 256ZA – CABGA



Top View



Side View



Bottom View  
(256 SOLDER BALLS)

**COMMON DIMENSIONS**  
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
D	-	17 BSC	-	
E	-	17 BSC	-	
A	1.30	1.40	1.50	
A1	0.31	0.36	0.41	
A2	0.29	0.34	0.39	
A3	0.65	0.70	0.75	
e	1.00 BSC			
b	0.46 REF			

- Notes: 1. This drawing is for general information only. Refer to JEDEC Drawing MO-205 for proper dimensions, tolerances, datums, etc.  
2. Array as seen from the bottom of the package.

11/07/01



2325 Orchard Parkway  
San Jose, CA 95131

**TITLE**

**256ZA**, 256-ball (16 x 16 Array), 17 x 17 mm Body,  
Chip Array Ball Grid Array (CABGA) Package

**DRAWING NO.**

256ZA

**REV.**

A



## Atmel Headquarters

### *Corporate Headquarters*

2325 Orchard Parkway  
San Jose, CA 95131  
TEL 1(408) 441-0311  
FAX 1(408) 487-2600

### *Europe*

Atmel Sarl  
Route des Arsenaux 41  
Case Postale 80  
CH-1705 Fribourg  
Switzerland  
TEL (41) 26-426-5555  
FAX (41) 26-426-5500

### *Asia*

Room 1219  
Chinachem Golden Plaza  
77 Mody Road Tsimshatsui  
East Kowloon  
Hong Kong  
TEL (852) 2721-9778  
FAX (852) 2722-1369

### *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

## Atmel Operations

### *Memory*

2325 Orchard Parkway  
San Jose, CA 95131  
TEL 1(408) 441-0311  
FAX 1(408) 436-4314

### *Microcontrollers*

2325 Orchard Parkway  
San Jose, CA 95131  
TEL 1(408) 441-0311  
FAX 1(408) 436-4314

La Chantrerie  
BP 70602  
44306 Nantes Cedex 3, France  
TEL (33) 2-40-18-18-18  
FAX (33) 2-40-18-19-60

### *ASIC/ASSP/Smart Cards*

Zone Industrielle  
13106 Rousset Cedex, France  
TEL (33) 4-42-53-60-00  
FAX (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.  
Colorado Springs, CO 80906  
TEL 1(719) 576-3300  
FAX 1(719) 540-1759

Scottish Enterprise Technology Park  
Maxwell Building  
East Kilbride G75 0QR, Scotland  
TEL (44) 1355-803-000  
FAX (44) 1355-242-743

### *RF/Automotive*

Theresienstrasse 2  
Postfach 3535  
74025 Heilbronn, Germany  
TEL (49) 71-31-67-0  
FAX (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.  
Colorado Springs, CO 80906  
TEL 1(719) 576-3300  
FAX 1(719) 540-1759

### *Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Datacom*

Avenue de Rochepleine  
BP 123  
38521 Saint-Egreve Cedex, France  
TEL (33) 4-76-58-30-00  
FAX (33) 4-76-58-34-80

---

*Atmel Programmable SLI Hotline*  
(408) 436-4119

*Atmel Programmable SLI e-mail*  
fpslic@atmel.com

*FAQ*  
Available on web site

*e-mail*  
literature@atmel.com

*Web Site*  
<http://www.atmel.com>

## © Atmel Corporation 2002.

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Atmel®, AVR® and Cache Logic® are the registered trademarks of Atmel. FPSLIC™, Secure FPSLIC™, FreeRAM™, System Designer™ and megaAVR™ are the trademarks of Atmel. Other terms and product names may be trademarks of others.



Printed on recycled paper.