Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

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
- JTAG (IEEE Std. 1149.1 Compliant) Interface
  - Extensive On-chip Debug Support
  - Limited Boundary-scan Capabilities According to the JTAG Standards (AVR Ports)
- 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

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

2314E–FPSLI–6/05

- **V_{CC}: 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**

1. Description

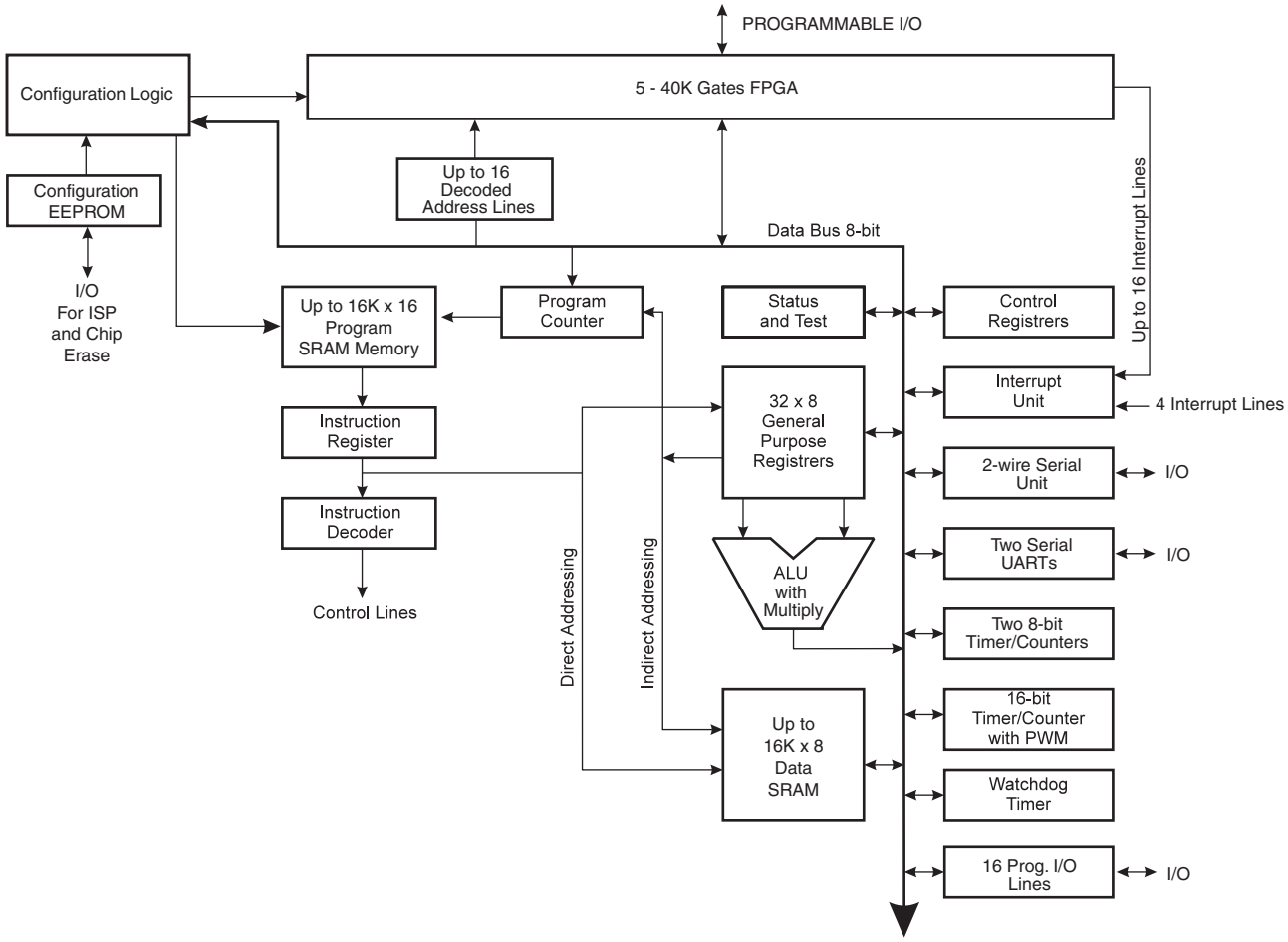
The AT94S Series (Secure FPSLIC family) shown in Table 1-1 is a combination of the popular Atmel AT40K Series SRAM FPGAs, the AT17 Series Configuration Memories and the high-performance Atmel AVR 8-bit RISC microcontroller with standard peripherals. Extensive data and instruction SRAM as well as device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

Device		AT94S05AL	AT94S10AL	AT94S40AL
Configuration Memory Size		1 Mbit	1 Mbit	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		93	137	162
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
JTAG ICE		Yes	Yes	Yes
Typical AVR Throughput	@ 25 MHz	19 MIPS	19 MIPS	19 MIPS
	@ 40 MHz	30 MIPS	30 MIPS	30 MIPS
Operating Voltage		3.0 - 3.6V	3.0 - 3.6V	3.0 - 3.6V

Figure 1-1. 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.

2. 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.

3. 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.

4. Programming and Configuration Timing Characteristics

Atmel's Configurator Programming Software (CPS), available from the Atmel web site (http://www.atmel.com/dyn/products/tools_card.asp?tool_id=3191), 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.

4.1 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.

4.2 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 kΩ 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.

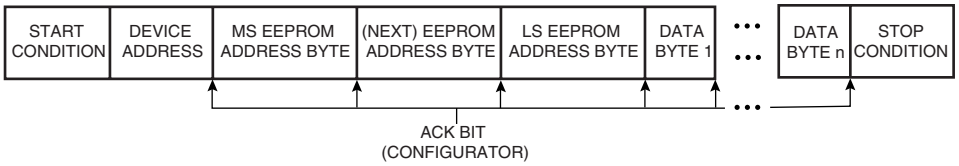
Again, the Acknowledge Bit is asserted on the cSDA line by the receiving device on a byte-by-byte basis.

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.

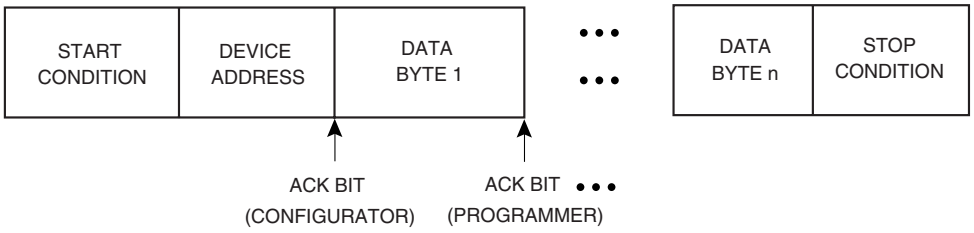
4.3 Bit Format

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



4.4 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 4-1.

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., $\overline{\text{SER\_EN}}$ 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.

4.5 Acknowledge Bit

The Acknowledge (ACK) Bit shown in Figure 4-1 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.

4.6 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.

4.7 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 4-1. Start and Stop Conditions

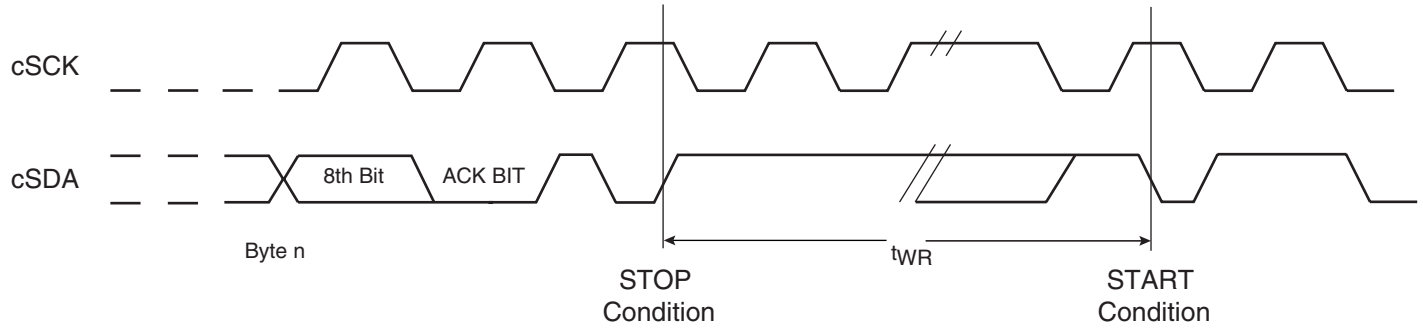
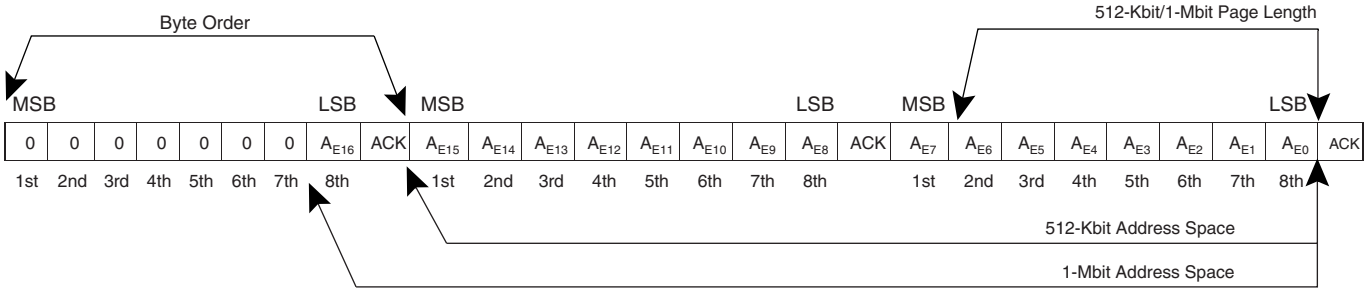


Table 4-1. Device Address Byte

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

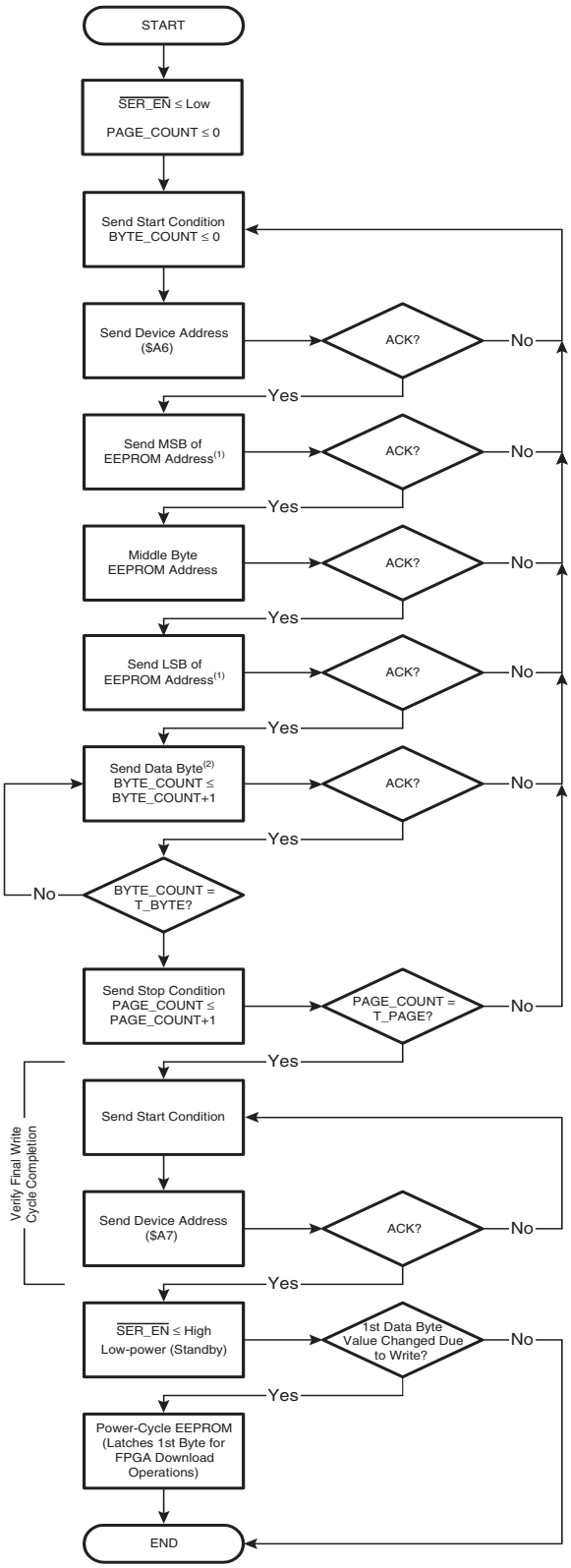
Where: R/$\overline{W}$ = 1 Read
= 0 Write

4.7.1 EEPROM Address



The EEPROM Address consists of three bytes on the 1-Mbit part. 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.

4.8 Programming Summary: Write to Whole Device



Notes: 1. The 1-Mbit part requires three EEPROM address bytes; all three bytes must be individually ACK'd by the EEPROM.

2. Data byte received/sent LSB to MSB.

4.8.1 EEPROM Address is Defined as:

AT17LV010 0000 $000x_9$ $x_8x_7x_6x_5$ $x_4x_3x_2x_1$ x_0000 0000

Note: where X_n ... X_0 is (PAGE_COUNT)\b

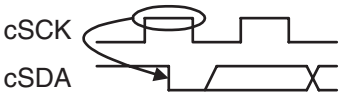
4.8.2 T_BYTE

AT17LV010 128

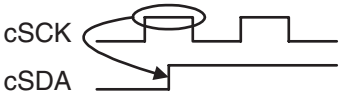
4.8.3 T_PAGE

AT17LV010 1024

START CONDITION

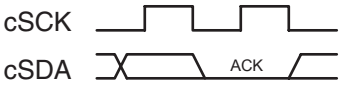


STOP CONDITION

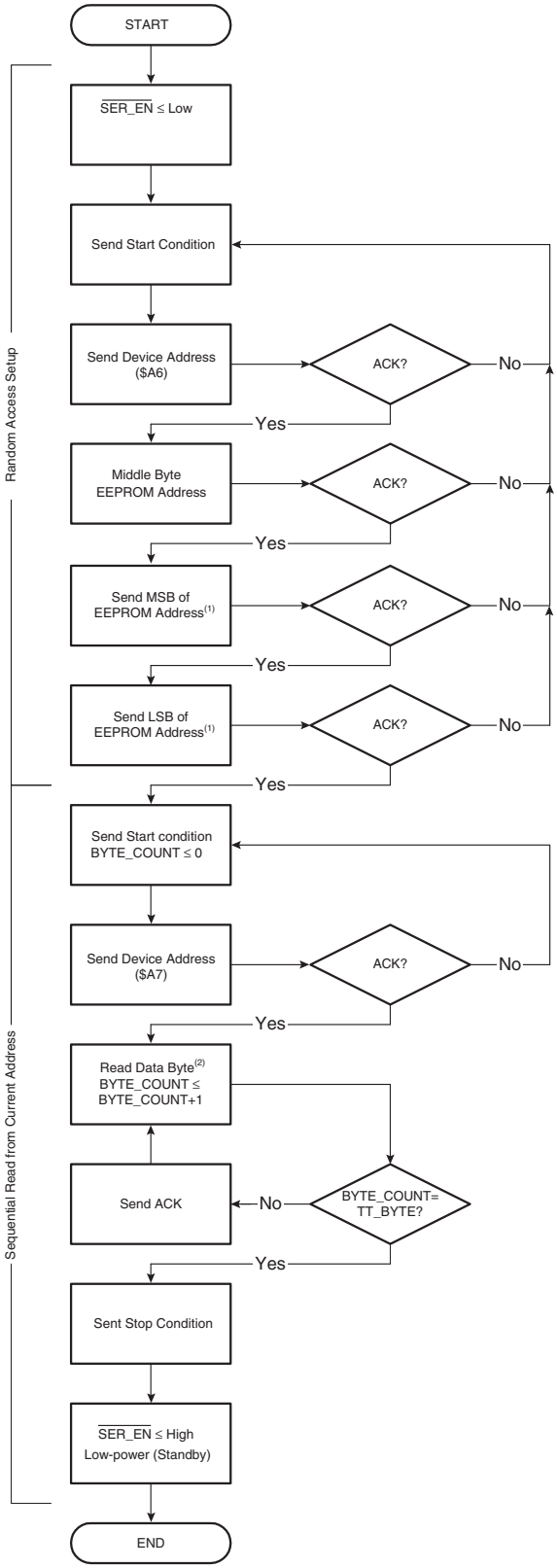


DATA BIT

ACK BIT



4.9 Programming Summary: Read from Whole Device



Notes: 1. The 1-Mbit part requires three EEPROM address bytes; all three bytes must be individually ACK'd by the EEPROM.

2. Data byte received/sent LSB to MSB

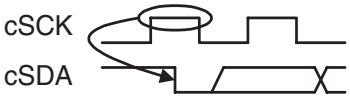
4.9.1 EEPROM Address is Defined as:

AT17LV010 00 00 00 \h

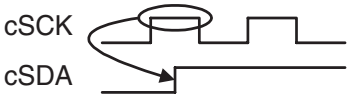
4.9.2 TT_BYTE

AT17LV010 131072 \d

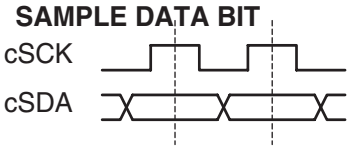
START CONDITION



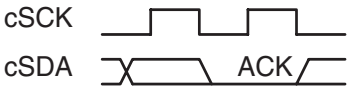
STOP CONDITION



SAMPLE DATA BIT



ACK BIT



4.9.3 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.

4.9.4 Writing

Writing to the normal address space takes place in pages. A page is 128-bytes long in the 1-Mbit part. The page boundaries are, respectively, addresses where A_{E0} down to A_{EOS} 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 1-Mbit part. 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 7 bits of the address 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/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
```

4.9.4.1 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.

4.9.5 Reading

Read instructions are initiated similarly to write instructions. However, with the R/$\overline{W}$ 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.

4.9.5.1 Current Address Read

Once the Device Address (with the R/$\overline{W}$ 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/$\overline{W}$ = 1
  - An Acknowledge Bit from the Configurator
- a Data Byte from the Configurator
- a Stop Condition from the programmer.

4.9.5.2 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/$\overline{W}$ = 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/$\overline{W}$ = 1
  - An Acknowledge Bit from the Configurator
- a Data Byte from the Configurator
- a Stop Condition from the programmer.

4.9.5.3 *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.(1) 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.

Note: 1. If an ACK is sent by the programmer after the data in the last memory address is sent by the configurator, the internal address counter will "rollover" to the first byte address of the memory array and continue to send data as long as an ACK is sent by the programmer.

4.9.6 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.

4.9.7 Reading Manufacturer's and Device Codes

On AT17LV010 Configurator, 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 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.

4.9.8 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.

4.9.8.1 *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.

4.9.9 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.

4.10 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:

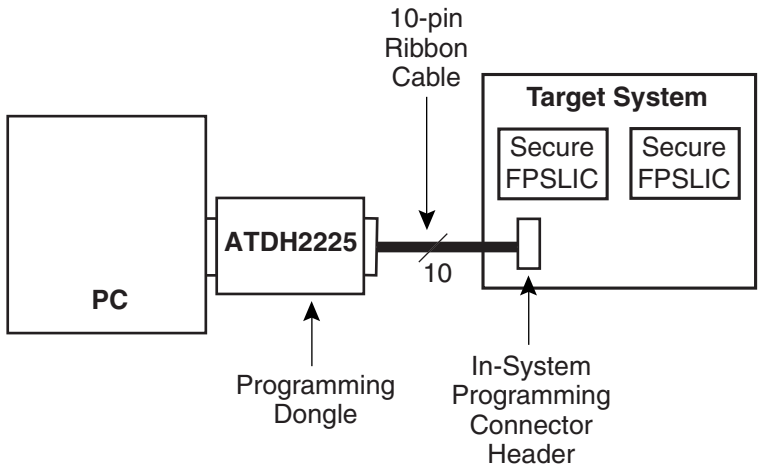
1. Read the Manufacturer's Code and the Device Code.
2. Program the device.
3. Verify the device data.

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 4-2.

The pages within the configuration EEPROM can be selectively rewritten.

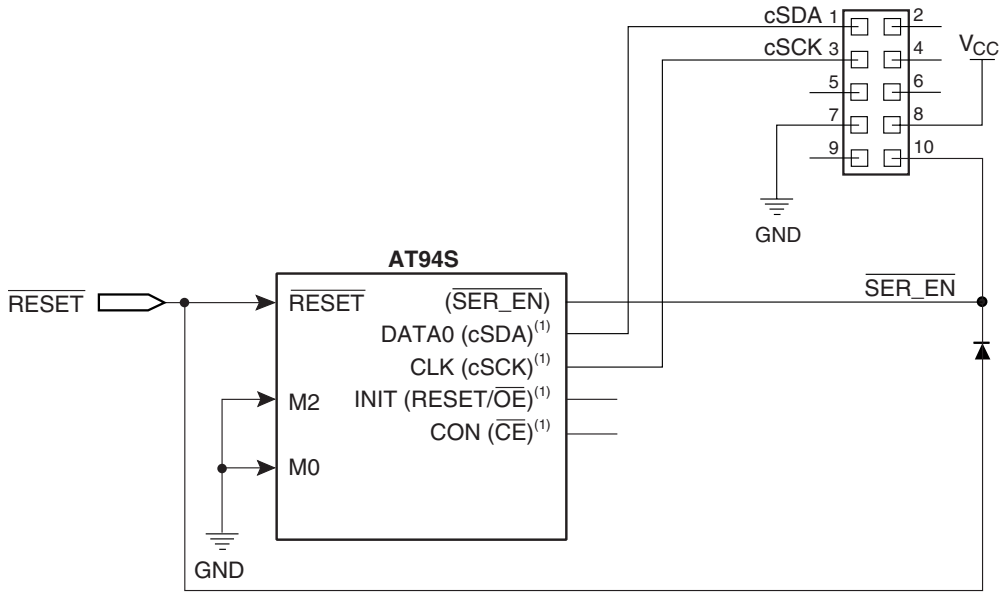
This document is limited to example implementations for Atmel's AT94S application.

Figure 4-2. Typical System Setup



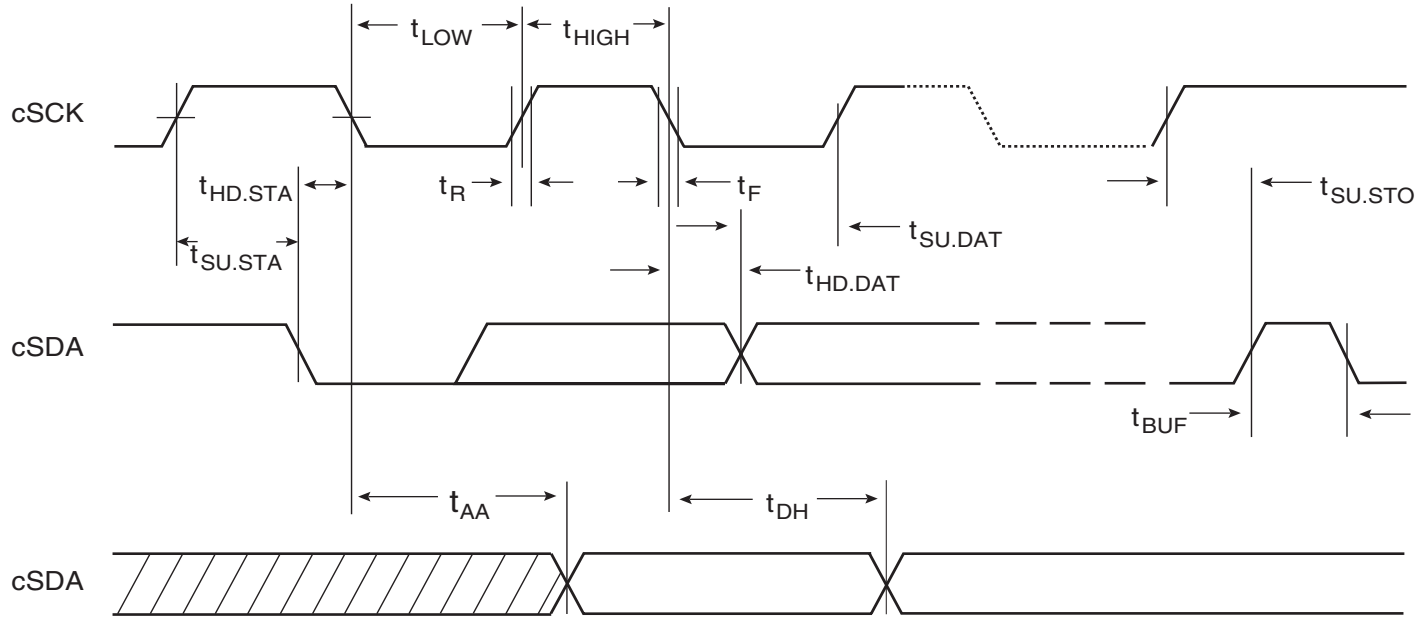
The diode connection between the AT94S' $\overline{\text{RESET}}$ pin and the $\overline{\text{SER\_EN}}$ signal allows the external programmer to force the FPGA into a reset state during ISP. This eliminates the potential for contention on the cSCK line. The pull-up resistors required on the lines to $\overline{\text{RESET}}$, CON and INIT are present on the inputs (internally) to the AT94S FPSLIC, see Figure 4-3.

Figure 4-3. ISP of the AT17LV512/010 in an AT94S FPSLIC Application



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

Figure 4-4. Serial Data Timing Diagram



4.11 DC Characteristics[1]

V_{CC} = 3.3V ± 10%, T_A = -40°C - 85°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	µA
I_{LO}	Output Leakage Current	V_{OUT} = V_{CC} or V_{SS}		0.05	10	µA
V_{IH}	High-level Input Voltage		V_{CC} x 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 $\overline{\text{SER\_EN}}$ is driven Low)
2. Commercial temperature range 0°C - 70°C
3. Industrial temperature range -40°C - 85°C
4. This parameter is characterized and is not 100% tested.

4.12 AC Characteristics[1]

V_{CC} = 3.3V ± 10%, T_A = -40°C - 85°C[2][3][4]

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

Notes: 1. Specific to programming mode (i.e., when $\overline{\text{SER\_EN}}$ is driven Low)
2. Commercial temperature range 0°C - 70°C
3. Industrial temperature range -40°C - 85°C
4. This parameter is characterized and is not 100% tested.

4.13 Secure FPSLIC Configurator Pin Configurations

144-pin LQFP	256-pin CABGA	Name	I/O	Description
105	D16	cSDA	I/O	Three-state DATA output for configuration. Open-collector bi-directional pin for programming.
107	C16	cSCK	O	CLOCK output. Used to increment the internal address and bit counter for reading and programming.
53	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/$\overline{OE}$.
72	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).
81	M5	$\overline{SER\_EN}$	I	Serial enable is normally High during FPGA loading operations. Bringing $\overline{SER\_EN}$ Low enables the programming mode.

4.14 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 it contents.

4.14.1 AT17LV512/010 Security Bit Programming

4.14.1.1 Disabling the Security Bit

Write 4 bytes “00 00 00 00” to addresses 800000-800003 two consecutive times, using the previously defined 2-wire write algorithm. Thereafter, either cycle the power or toggle (HI-LO-HI) the $\overline{SER\_EN}$ pin in order to disable the security.

4.14.1.2 Enabling the Security Bit

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

4.14.1.3 Verifying the Security Bit

Read 4 bytes of data from addresses 800000-800003 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.

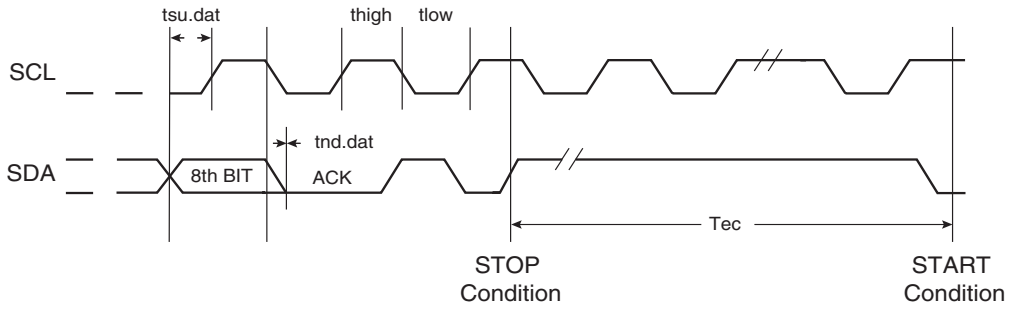
4.15 Chip Erase Timing

The entire device can be erased at once by writing to a specific address. This operation will erase the entire array. See Table 4-2 for specifics on the write algorithm.

Table 4-2. Chip Erase Cycle Characteristics

Symbol	Parameter
Tec	Chip Erase Cycle Time (25 ms)

Figure 4-5. Chip Erase Timing Diagram



5. Packaging and Pin List information

Table 5-1. Part and Package Combinations Available

Part #	Package	AT94S05	AT94S10	AT94S40
BG256	DG	93	137	162
LQ144	BQ	–	84	84

Table 5-2. AT94K JTAG ICE Pin List

Pin	AT94S05 96 FPGA I/O	AT94S10 192 FPGA I/O	AT94S40 384 FPGA I/O
TDI	IO34	IO50	IO98
TDO	IO38	IO54	IO102
TMS	IO43	IO63	IO123
TCK	IO44	IO64	IO124

Table 5-3. AT94S Pin List

AT94S05 96 FPGA I/O	AT94S10 144 FPGA I/O	AT94S40 288 FPGA I/O	Package	
			Chip Array 256 CABGA	LQ144[1]
FPSLIC Array				
I/O1, GCK1 (A16)	I/O1, GCK1 (A16)	I/O1, GCK1 (A16)	A1	2
I/O2 (A17)	I/O2 (A17)	I/O2 (A17)	D4	3
I/O3	I/O3	I/O3	D3	4
I/O4	I/O4	I/O4	B1	5
I/O5 (A18)	I/O5 (A18)	I/O5 (A18)	C2	6
I/O6 (A19)	I/O6 (A19)	I/O6 (A19)	C1	7
		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	9
I/O10	I/O14	I/O26	G7	10
I/O11 (A20)	I/O15 (A20)	I/O27 (A20)	G6	11
I/O12 (A21)	I/O16 (A21)	I/O28 (A21)	G4	12
NC	I/O17	I/O29	G5	
NC	I/O18	I/O30	G2	
		I/O31		
		I/O32		

Table 5-3. AT94S Pin List (Continued)

AT94S05 96 FPGA I/O	AT94S10 144 FPGA I/O	AT94S40 288 FPGA I/O	Package	
			Chip Array 256 CABGA	LQ144[1]
		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	13
I/O14	I/O22	I/O44	H1	14
		I/O45		
		I/O46		
I/O15 (A22)	I/O23 (A22)	I/O47 (A22)	J7	15
I/O16 (A23)	I/O24 (A23)	I/O48 (A23)	J1	16
I/O17 (A24)	I/O25 (A24)	I/O49 (A24)	J4	19
I/O18 (A25)	I/O26 (A25)	I/O50 (A25)	J5	20
		I/O51		
		I/O52		
I/O19	I/O27	I/O53	J6	21
I/O20	I/O28	I/O54	J8	22
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	
NC	NC	I/O62	K5	
		I/O63		
		I/O64		
NC	NC	I/O65	K6	
NC	NC	I/O66	L1	

Table 5-3. AT94S Pin List (Continued)

AT94S05 96 FPGA I/O	AT94S10 144 FPGA I/O	AT94S40 288 FPGA I/O	Package	
			Chip Array 256 CABGA	LQ144[1]
NC	I/O31	I/O67	L2	
NC	I/O32	I/O68	L5	
I/O21 (A26)	I/O33 (A26)	I/O69 (A26)	L4	23
I/O22 (A27)	I/O34 (A27)	I/O70 (A27)	M1	24
I/O23	I/O35	I/O71	M2	25
I/O24, FCK2	I/O36, FCK2	I/O72, FCK2	N1	26
		I/O73		
		I/O74		
	I/O37	I/O75		
	I/O38	I/O76		
		I/O77		
		I/O78		
		I/O79		
		I/O80		
I/O25	I/O39	I/O81	M3	
I/O26	I/O40	I/O82	N2	
	I/O41	I/O83		
	I/O42	I/O84		
		I/O85		
		I/O86		
		I/O87		
		I/O88		
I/O27 (A28)	I/O43 (A28)	I/O89 (A28)	P1	28
I/O28	I/O44	I/O90	P2	29
		I/O91		
		I/O92		
I/O29	I/O45	I/O93	R1	30
I/O30	I/O46	I/O94	N3	31
I/O31 ($\overline{\text{OTS}}$)	I/O47 ($\overline{\text{OTS}}$)	I/O95 ($\overline{\text{OTS}}$)	T1	32
I/O32, GCK2 (A29)	I/O48, GCK2 (A29)	I/O96, GCK2 (A29)	P3	33
AVRRESET	$\overline{\text{AVRRESET}}$	$\overline{\text{AVRRESET}}$	R2	34
M0	M0	M0	R3	36
FPSLIC Array				
M2	M2	M2	T3	38

Table 5-3. AT94S Pin List (Continued)

AT94S05 96 FPGA I/O	AT94S10 144 FPGA I/O	AT94S40 288 FPGA I/O	Package	
			Chip Array 256 CABGA	LQ144[1]
I/O33, GCK3	I/O49, GCK3	I/O97, GCK3	R4	39
I/O34 (HDC/TDI)	I/O50 (HDC/TDI)	I/O98 (HDC/TDI)	T4	40
I/O35	I/O51	I/O99	N5	41
I/O36	I/O52	I/O100	P5	42
	I/O53	I/O101		43
SER_EN	SER_EN	SER_EN	M5	81
I/O38 (LDC/TDO)	I/O54 (LDC/TDO)	I/O102 (LDC/TDO)	R5	44
		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	46
I/O42	I/O62	I/O122	N7	47
I/O43 (TMS)	I/O63 (TMS)	I/O123 (TMS)	P7	48
I/O44 (TCK)	I/O64 (TCK)	I/O124 (TCK)	R7	49
NC	I/O65	I/O125	K7	
NC	I/O66	I/O126	K8	
		I/O127		
		I/O128		
		I/O129		

Table 5-3. AT94S Pin List (Continued)

AT94S05 96 FPGA I/O	AT94S10 144 FPGA I/O	AT94S40 288 FPGA I/O	Package	
			Chip Array 256 CABGA	LQ144[1]
		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	50
I/O46	I/O70	I/O138	N8	51
		I/O139		
		I/O140		
		I/O141		
		I/O142		
I/O47 (TD7)	I/O71 (TD7)	I/O143 (TD7)	L8	52
I/O48 (InitErr) RESET/$\overline{OE}$	I/O72 (InitErr) RESET/$\overline{OE}$	I/O144 (InitErr) RESET/$\overline{OE}$	K9	53
I/O49 (TD6)	I/O73 (TD6)	I/O145 (TD6)	P9	56
I/O50 (TD5)	I/O74 (TD5)	I/O146 (TD5)	N9	57
		I/O147		
		I/O148		
		I/O149		
		I/O150		
I/O51	I/O75	I/O151	M9	58
I/O52	I/O76	I/O152	L9	59
NC	I/O77	I/O153	J9	
NC	I/O78	I/O154	T10	
		I/O155		
		I/O156		
		I/O157		
		I/O158		
		I/O159		
		I/O160		
		I/O161		
		I/O162		
NC	I/O79	I/O163	P10	

Table 5-3. AT94S Pin List (Continued)

AT94S05 96 FPGA I/O	AT94S10 144 FPGA I/O	AT94S40 288 FPGA I/O	Package	
			Chip Array 256 CABGA	LQ144[1]
NC	I/O80	I/O164	N10	
I/O53 (TD4)	I/O81 (TD4)	I/O165 (TD4)	L10	60
I/O54 (TD3)	I/O82 (TD3)	I/O166 (TD3)	T11	61
I/O55	I/O83	I/O167	R11	62
I/O56	I/O84	I/O168	M11	63
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	65
I/O60 (TD1)	I/O92 (TD1)	I/O184 (TD1)	P14	66
		I/O185		
		I/O186		
		I/O187		
		I/O188		
I/O61	I/O93	I/O189	R16	67
I/O62	I/O94	I/O190	P15	68
I/O63 (TD0)	I/O95 (TD0)	I/O191 (TD0)	N14	69
I/O64, GCK4	I/O96, GCK4	I/O192, GCK4	P16	70
$\overline{\text{CON/CE}}$	$\overline{\text{CON/CE}}$	$\overline{\text{CON/CE}}$	N16	72
FPSLIC Array				
$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	M14	74
PE0	PE0	PE0	M12	75
PE1	PE1	PE1	M15	76

Table 5-3. AT94S Pin List (Continued)

AT94S05 96 FPGA I/O	AT94S10 144 FPGA I/O	AT94S40 288 FPGA I/O	Package	
			Chip Array 256 CABGA	LQ144[1]
PD0	PD0	PD0	M16	77
PD1	PD1	PD1	L12	78
PE2	PE2	PE2	L15	79
PD2	PD2	PD2	L11	80
NC	NC	NC	E12	
$\overline{\text{SER\_EN}}$	$\overline{\text{SER\_EN}}$	$\overline{\text{SER\_EN}}$	M5	81
PD3	PD3	PD3	K11	82
PD4	PD4	PD4	K12	83
PE3	PE3	PE3	K14	84
$\overline{\text{CS0}}$	$\overline{\text{CS0}}$	$\overline{\text{CS0}}$	K15	85
SDA	SDA	SDA	J10	
SCL	SCL	SCL	J12	
PD5	PD5	PD5	J14	86
PD6	PD6	PD6	J13	87
PE4	PE4	PE4	J16	88
PE5	PE5	PE5	J11	89
PE6	PE6	PE6	H15	92
PE7 ($\overline{\text{CHECK}}$)	PE7 ($\overline{\text{CHECK}}$)	PE7 ($\overline{\text{CHECK}}$)	H14	93
PD7	PD7	PD7	H13	94
INTP0	INTP0	INTP0	H12	95
XTAL1	XTAL1	XTAL1	G15	96
XTAL2	XTAL2	XTAL2	G14	97
RX0	RX0	RX0	G12	98
TX0	TX0	TX0	G11	99
INTP1	INTP1	INTP1	F15	
INTP2	INTP2	INTP2	F14	
TOSC1	TOSC1	TOSC1	E16	101
TOSC2	TOSC2	TOSC2	E15	102
RX1	RX1	RX1	E14	103
TX1	TX1	TX1	E13	104
DATA0/cSDA	DATA0/cSDA	DATA0/cSDA	D16	105
INTP3 ($\overline{\text{CSOUT}}$)	INTP3 ($\overline{\text{CSOUT}}$)	INTP3 ($\overline{\text{CSOUT}}$)	D15	106
CCLK/cSCK	CCLK/cSCK	CCLK/cSCK	C16	107
I/O65:96 Are Unbonded	I/O97:144 Are Unbonded	I/O193:288 Are Unbonded		

Table 5-3. AT94S Pin List (Continued)

AT94S05 96 FPGA I/O	AT94S10 144 FPGA I/O	AT94S40 288 FPGA I/O	Package	
			Chip Array 256 CABGA	LQ144[1]
FPSLIC Array				
Testclock	Testclock	Testclock	C15	109
I/O97 (A0)	I/O145 (A0)	I/O289 (A0)	C14	111
I/O98, GCK7 (A1)	I/O146, GCK7 (A1)	I/O290, GCK7 (A1)	B15	112
I/O99	I/O147	I/O291	A16	113
I/O100	I/O148	I/O292	D13	114
		I/O293		
		I/O294		
NC	NC	I/O295	C13	
NC	NC	I/O296	B14	
I/O101 ($\overline{CS1}$, A2)	I/O149 ($\overline{CS1}$, A2)	I/O297 ($\overline{CS1}$, A2)	A15	115
I/O102 (A3)	I/O150 (A3)	I/O298 (A3)	A14	116
		I/O299		
		I/O300		
I/O104	I/O151	I/O301	Shared with Test clock	
NC	I/O152	I/O302	D12	
I/O103	I/O153	I/O303	C12	117
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	119
I/O106	I/O158	I/O314	F10	120
NC	I/O159	I/O315	E10	
NC	I/O160	I/O316	D10	
NC	NC	I/O317	C10	
NC	NC	I/O318	B10	
		I/O319		
		I/O320		