



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



**FPSLIC<sup>®</sup>** FPSLIC STK594

---

## **User Guide**





## Table of Contents

---

### Section 1

Introduction .....	1-1
--------------------	-----

---

### Section 2

Using the STK594 Top Module .....	2-1
2.1 Preparing the STK500 for Use with the STK594.....	2-1
2.1.1 Adjusting VTARGET for the AT94K Devices .....	2-1
2.1.2 Connecting the STK594 to the STK500 Starter Kit .....	2-1
2.2 PORT Connectors.....	2-2
2.2.1 PORT E .....	2-2
2.3 Programming the AT94K Devices.....	2-3
2.4 JTAG Connector .....	2-3
2.5 TOSC Switch.....	2-4
2.6 Universal Asynchronous Receiver Transmitter (UART).....	2-4
2.6.1 Second RS-232C Port .....	2-4
2.7 Two-Wire Serial Interface (TWSI) .....	2-5
2.7.1 Description of Configuration Memory Pins .....	2-5
2.8 External Interrupts.....	2-5
2.9 Split Power Supply Support .....	2-6
2.10 XTAL Switch .....	2-6
2.11 Reset Switches .....	2-6

---

### Section 3

Installing System Designer .....	3-1
3.1 System Requirements.....	3-1
3.3 Configuration Programming System (CPS) Installation .....	3-2
3.4 System Designer Licensing.....	3-2
3.4.1 Requesting a System Designer License.....	3-2
3.4.2 Configuring the System Designer License .....	3-3
3.4.3 Testing the System Designer License .....	3-3
3.4.4 Troubleshooting.....	3-3

---

### Section 4

Using System Designer .....	4-1
4.1 Preparing the Example Files.....	4-1
4.3 Design Flow .....	4-2
4.6 Synthesizing the FPGA Source File.....	4-8
4.8 FPGA Place and Route.....	4-11

4.10.1 Hardware Setup.....4-15

4.10.2 Software Setup .....4-16

---

**Section 5**

Technical Specifications ..... 5-1

---

**Section 6**

Complete Schematics ..... 6-1







# Section 1

## Introduction

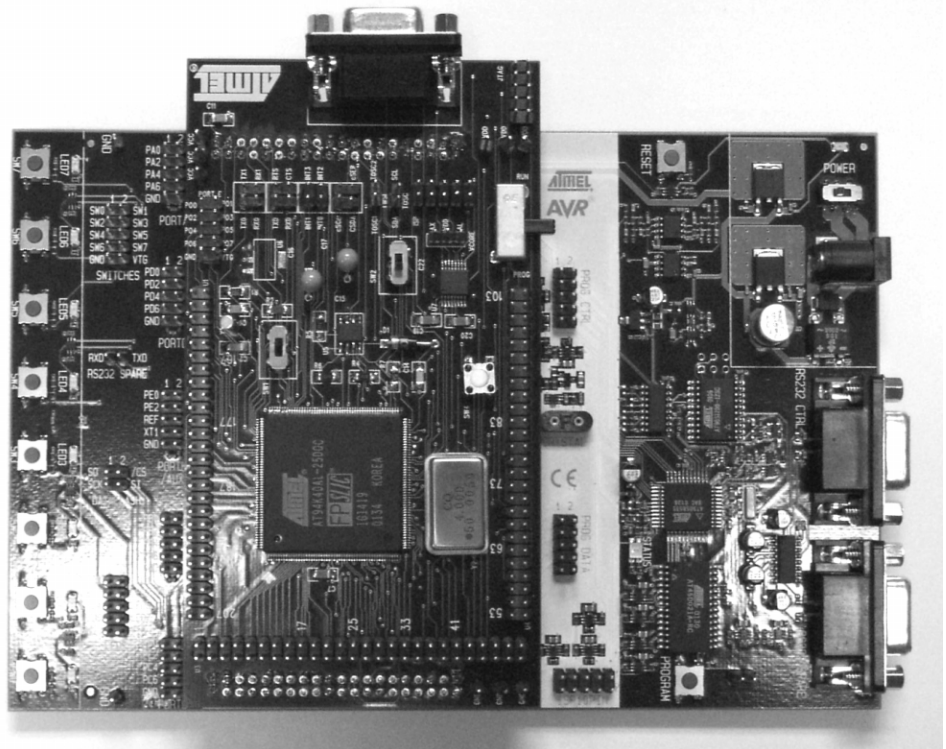
The STK594 board is a top module designed to add AT94K FPSLIC® support to the STK500 development board. With this board the STK500 is extended to support all current AT94K FPSLIC devices in a single development environment.

The STK594 includes connectors, jumpers and hardware allowing full utilization of the new features of the FPSLIC family, see Figure 1-1.

This user guide acts as a general getting started guide as well as a complete technical reference for advanced users.

In addition to adding support for new devices, it also adds new support for peripherals previously not supported by the STK500. An additional RS-232 port and a Two-Wire Serial Interface are among the new features.

**Figure 1-1.** STK594 Top Module for STK500



---

## 1.1 Features

- STK500 Compatible
- AVR Studio® and System Designer Compatible
- Supports AT94KAL and AT94KAX Devices
- Supports all Added Features in FPSLIC Devices
- JTAG Connector for On-chip Debugging Using JTAG ICE
- Additional RS-232C Port with Available RTS/CTS Handshake Lines
- On-board 32 kHz Crystal for Easy RTC Implementations



## Section 2

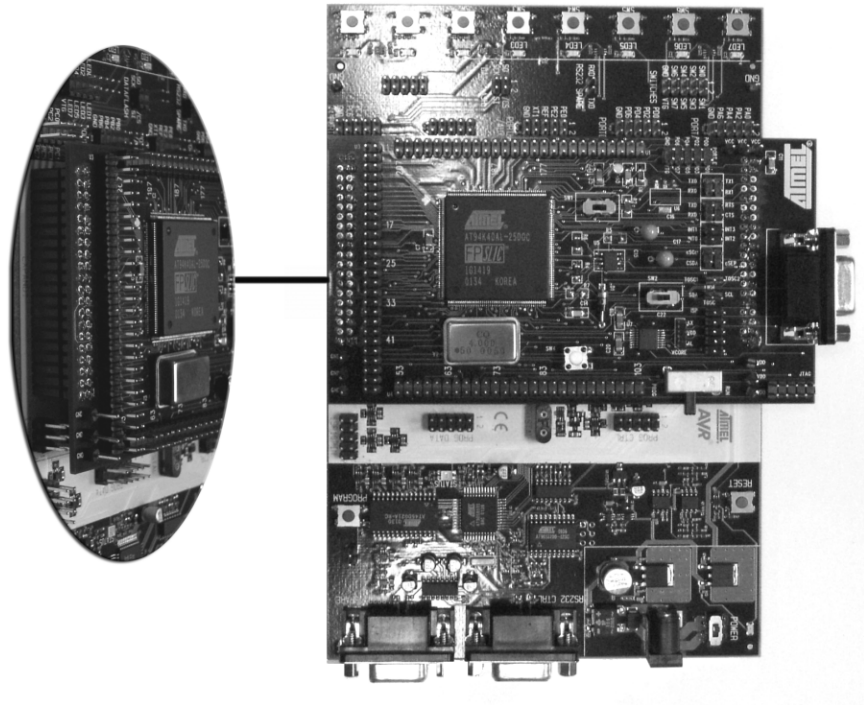
---

# Using the STK594 Top Module

---

- 2.1 Preparing the STK500 for Use with the STK594** Prior to using the STK594 with the STK500, it is necessary to make a few adjustments to the STK500 Starter Kit to allow for proper operation of Atmel's AT94K FPSLIC® devices.
- 2.1.1 Adjusting VTARGET for the AT94K Devices** According to the AT94K Series datasheet, the  $V_{CC}$  operating voltage is specified where  $\{V_{CC} \mid 3.0 < V_{CC} = 3.6\}$  Volts, with respect to ground. The STK594 board requires that the STK500 board supplies a  $V_{CC}$  within the operating range for the AT94K devices. Prior to using the STK594 board, it is necessary to adjust the VTARGET to a value between 3.0 and 3.6V. For more information on adjusting VTARGET from within AVR Studio®, consult section 5.3.5.1 of the STK500 User Guide, available on the Atmel web site ([www.atmel.com](http://www.atmel.com)).
- Note:** It may be necessary to adjust the  $V_{DD}$  voltage, see "Split Power Supply Support" on page 6 of this section for more information.
- 2.1.2 Connecting the STK594 to the STK500 Starter Kit** The STK594 should be connected to the STK500 expansion header 0 and 1. It is important that the top module is connected in the correct orientation as shown in Figure 2-1 on page 2. The EXPAND0 written on the STK594 top module should match the EXPAND0 written beside the expansion header on the STK500 board.

**Figure 2-1.** Connecting the STK594 to the STK500 Board



**Note:** Connecting the STK594 with the wrong orientation may damage the boards.

**Note:** Do not mount the STK594 at the same time an AVR<sup>®</sup> is mounted on the STK500 board.

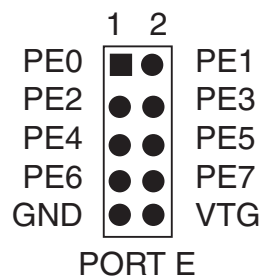
## 2.2 PORT Connectors

Since the AT94K devices have additional ports not available on the STK500, these ports are located on the STK594 board. The STK594 ports have the same pinout and functionality as the ports on the STK500 board. Since Port A to Port D are already present on the STK500 board, they are not duplicated on the STK594.

### 2.2.1 PORT E

Figure 2-2 shows the pinout for the I/O port headers Port E.

**Figure 2-2.** General I/O Ports



**Note:** Port E is also present on the STK500, but only PE0 to PE2 (3 least significant bits) are accessible. To access all Port E bits the connector on the STK594 must be used.



**2.3 Programming the AT94K Devices** The FPSLIC configuration process involves configuring the FPGA, the AVR<sup>®</sup> program code and the FPSLIC data memory. This configuration requires a single bitstream that configures the FPGA, the embedded AVR Program SRAM and the FPSLIC Data SRAM. The combined bitstream is automatically generated by the Bitstream Generator, a System Designer software utility.

After a reset and the internal clearing of the configuration data, the FPSLIC device self-initiates configuration. The Master mode uses an internal oscillator to provide the Configuration Clock (CCLK) for clocking the external EEPROM (configurator), which contains the configuration data. After auto-configuration is complete, re-configuration can be initiated manually by the user, if needed.

**Note:** The AT94K devices also support Self-Programming. For more information on this topic, refer to the “Code-Self Modify” application note available on the Atmel web site.

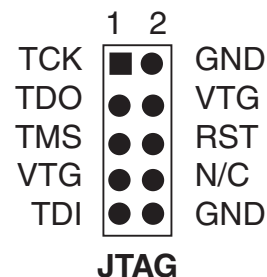
**Note:** The AT94K devices also support Cache Logic<sup>®</sup> Configuration. For more information on this topic, refer to the “Cache Logic Configuration” application note available on the Atmel web site.

For more details on programming procedures, refer to Section 4.10.

**2.4 JTAG Connector** The JTAG connector is intended for the AT94K devices that have a built-in JTAG interface. The pinout of the JTAG connector is shown in Figure 2-3 and is compliant with the pinout of the JTAG ICE available from Atmel. Connecting a JTAG ICE to this connector allows On-chip Debugging of the AT94K devices.

More information about the JTAG ICE and On-chip Debugging can be found in the AVR JTAG ICE user guide, available on the Atmel web site.

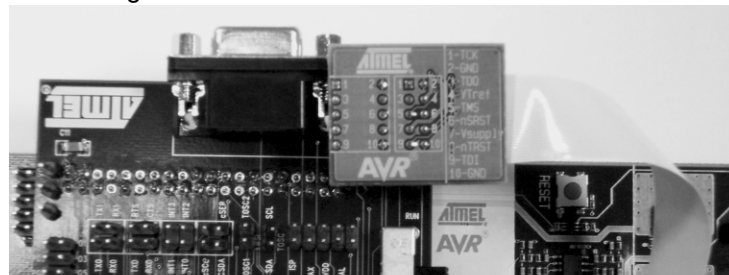
**Figure 2-3.** JTAG Connector



**Note:** To determine if your AT94K device supports JTAG Debug, examine the date code. Any parts with a J after their date code support JTAG. Example, 4201J.

Figure 2-4 shows how to connect the JTAG ICE probe on the STK594 board.

**Figure 2-4.** Connecting JTAG ICE to the STK594

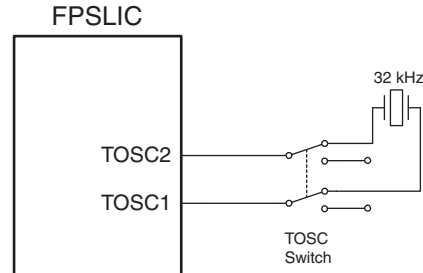


## 2.5 TOSC Switch

The AT94K device provides dedicated I/O pins for TOSC1 and TOSC2, rather than sharing with the general purpose I/O pins. The TOSC switch selects whether or not the 32 kHz crystal is connected to the pins of the device.

Figure 2-5 shows a simplified block schematic on how this is implemented.

**Figure 2-5.** TOSC Block Schematic

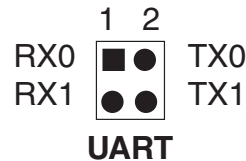


## 2.6 Universal Asynchronous Receiver Transmitter (UART)

Unlike traditional AVR microcontrollers, the AT94K device provides the option of having separate I/O pins for the UARTs rather than sharing with the general purpose I/O pins.

Figure 2-6 shows the pinout of a header for the dedicated UART pins.

**Figure 2-6.** UART Header

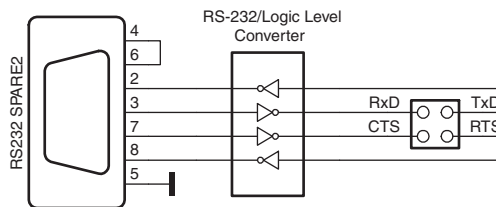


### 2.6.1 Second RS-232C Port

The AT94K device has an additional UART. The RS-232 port on the STK594 board has in addition to the RXD and TXD lines support for RTS and CTS flow control. Figure 2-7 shows a simplified block schematic on how this is implemented.

**Note:** The UART in AT94K devices does not support hardware RTS or CTS control. If such functionality is needed, it must be implemented in software.

**Figure 2-7.** UART Block Schematic



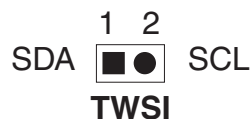
This UART can also be used from devices placed in the STK500 board. Simply connect the appropriate port pins to RXD and TXD on the STK594 board.

**Note:** If no software RTS/CTS flow control is implemented, a jumper shorting RTS and CTS will ensure correct communication with an external application that uses such flow control.

- 2.7 Two-Wire Serial Interface (TWSI)** The AT94K device includes dedicated I/O pins for the TWSI rather than sharing with the general purpose I/O pins.

Figure 2-8 shows the pinout of a header for the dedicated TWSI pins.

**Figure 2-8.** TWSI Header

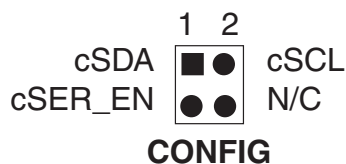


- 2.7.1 Description of Configuration Memory Pins** An AT17LV010-10CC 1-Mbit Configuration Memory is included on the STK594 for supplying the AT94K FPSLIC device with its configuration data, as well as for non-volatile data storage. The configurator is a high-density EEPROM with a TWSI interface. A detailed datasheet of the Configuration Memory can be obtained from the Atmel web site.

The configurator can be connected to the I/O pins of the embedded AVR microcontroller. The 4-pin header marked *CONFIG* can be used for connecting the TWSI interface of the configurator to the I/O pins of the target AVR microcontroller. Two-wire cables are included with the STK500 for connecting the configurator to the I/O pins.

Figure 2-9 shows the pinout of a header for the Configuration Memory pins.

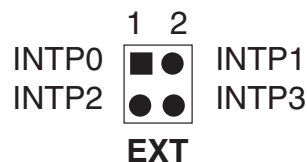
**Figure 2-9.** Configuration Memory Header



- 2.8 External Interrupts** Unlike traditional AVR microcontrollers, the AT94K device provides the option of having separate I/O pins for the External Interrupts rather than sharing with the general purpose I/O pins.

Figure 2-10 shows the pinout of a header for the dedicated External Interrupt pins.

**Figure 2-10.** External Interrupt Header



**2.9 Split Power Supply Support**

The AT94K FPSLIC devices exist in two different variations, the AL and AX. The AL variation is a 3.3V device manufactured on a 0.35μ process, while the AX variation has a 1.8V core manufactured on a 0.18μ process. The primary difference between the two variations, is that the AX device requires a split power supply, as the I/Os are still powered from a 3.3V supply, while the core operates at 1.8V.

The STK594 supports both AT94K variations. If an AX variant is being used it is necessary to supply the proper core to the AT94K device. Figure 2-11 shows how to set the jumper to select the core voltage.

**Figure 2-11.** AT94K Core Voltage Selector

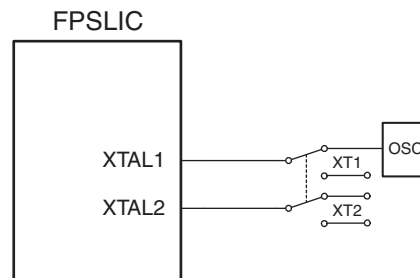


**2.10 XTAL Switch**

An oscillator is included on the STK594 for supplying an additional clock to the AT94K FPSLIC device. The XTAL switch selects if the oscillator is connected to the XTAL1 pin, or whether the clock is provided by the STK500.

Figure 2-12 shows a simplified block schematic on how this is implemented.

**Figure 2-12.** XTAL Block Schematic



**2.11 Reset Switches**

The reset switch found on the STK594 is connected to the AT94K's  $\overline{\text{RESET}}$  pin. When pressed, the AT94K device will reset and initiate a configuration download from the configuration memory.

The reset switch found on the STK500 is connected to the  $\overline{\text{AVRRESET}}$ . When pressed, the embedded AVR microcontroller resets and begins execution at location \$0000.



## Section 3

---

# Installing System Designer

System Designer is the ideal software platform for all AT94K FPSLIC<sup>®</sup> development. It includes an Editor, an Assembler and a Debugger as its development tools for the embedded AVR<sup>®</sup> development, and also includes a Simulator, Synthesizer and a Place and Route tool for FPGA development. System Designer also includes a Co-Verification suite powered by Mentor Graphics<sup>®</sup>, allowing for step-by-step simulation of the FPGA and AVR design concurrently.

---

### 3.1 System Requirements

For a single-user system, System Designer requires a personal computer equipped as follows:

- CD-ROM Drive
- 250-Mbyte Minimum Hard Drive
- 128-Mbyte RAM
- Parallel Interface Port
- Windows<sup>®</sup> 95/98/2000/Me, or WindowsNT<sup>®</sup> 4.0
- Network Interface Card or Security Dongle

The software security dongle is used to generate a unique HOSTID for systems without a network interface card. The security dongle is connected to the PC through the parallel port interface. It is possible to configure a floating network license through the security dongle. The security dongle allows users to use the software dongle on different machines by removing and placing the dongle on other machines.



### 3.2 System Designer Installation

This installation assumes that you have no previous version of System Designer installed in your machine. If you have a previous version of the software installed or you need step-by-step installation instructions, please refer to the System Designer “Installation, Licensing and Troubleshooting” tutorial available on the Atmel web site.

1. Insert the supplied System Designer CD-ROM into the computer. If the CD does not automatically start, execute `SETUP.EXE` from the CD.
2. From the CD Browser, select *Install Products* and select *System Designer*. The System Designer installation will perform a full installation, as there are no optional components.

**Note:** Do not install System Designer to a directory name that contains spaces, otherwise improper operation will occur.

**Note:** When you reach the portion of the System Designer installation prompting about licensing, you should select one of two options, either *Custom* or *Skip*. Select *Custom* if you already have a valid license and follow the instructions outlined in Section 3.4.2 “Configuring the System Designer License”. Choose *Skip* if you do not have a license and follow the instructions in Section 3.4.1 “Requesting a System Designer License”.

**Note:** AVR Studio® version 3.2 or higher is required for STK500/STK594 support.

### 3.3 Configuration Programming System (CPS) Installation

From the CD Browser, select *Install Products* and select *CPS for AT17 Devices*. This will install the CPS utility, which is used to program the AT17 and ATFS series configuration memories found on the STK594.

**Note:** When installing the CPS utility, it is necessary to install the software in an account with Administrator privileges if the operating system is WindowsNT or Windows 2000/XP.

### 3.4 System Designer Licensing

The licensing of System Designer is for the Mentor Graphics tools. You can use System Designer without a valid license, however you will not be able to use ModelSim®, LeonardoSpectrum™ or Co-Verification.

The typical license is based on the hostID of your Network Interface Card (NIC). If you prefer to use a dongle-based license, it is necessary that you purchase a Security Dongle from Atmel (Atmel Part Number: **ATDH94DNG**).

The instructions below describe the configuration of a NIC-based license. If you request a dongle-based license, you will receive instructions on how to configure the license with the dongle. If you wish to use a single license for multiple machines, it is necessary to purchase a Security Dongle.

#### 3.4.1 Requesting a System Designer License

Prior to obtaining a license for System Designer, it is necessary to first install the System Designer software suite. During installation, System Designer creates the file `LMUTIL.TXT`, which is found in the `C:\SystemDesigner\ETC` directory, assuming a default installation. The `LMUTIL.TXT` file contains the hostID of your NIC, and is composed of a combination of twelve alphanumeric characters.

Once you have installed System Designer, proceed to the FPSLIC section of the Atmel web site and click on the *Request License* button, alternatively the direct link for the license request page is <http://www.atmel.com/atmel/products/prod39r.htm>.

**Note:** The Serial Number is located on the white sticker on the underside of the STK594 board or on the System Designer case.



### 3.4.2 Configuring the System Designer License

1. Once you have received your System Designer license from Atmel, place the file in the `C:\SystemDesigner` directory.
2. Launch the Mentor Graphics License Configuration Utility from *Start > Programs > Atmel > Mentor Graphics Licensing > Configure Licensing*.
3. Follow the on-screen instructions. When it prompts you to select *Configuration Option*, choose *3*.
4. Define the *Product License Location* and press *Next*. Use the full path and file-name when defining the license location, for example `C:\SystemDesigner\fpslic.dat`. If you used a different path and/or file-name, make the necessary changes.

**Note:** If you are using Windows 95/98/Me it is necessary to reboot the machine prior to running the System Designer software.

### 3.4.3 Testing the System Designer License

Once you have configured your license, you can test it by invoking the Mentor Graphics programs that require a license.

1. Launch ModelSim from *Start > Programs > Atmel > ModelTech > ModelSim*. If ModelSim launches without any licensing errors, the ModelSim license has been successfully installed.
2. Launch LeonardoSpectrum from *Start > Programs > Atmel > Leonardo Spectrum > Leonardo Spectrum*. If LeonardoSpectrum launches without any licensing errors, then the LeonardoSpectrum license has been successfully installed.

### 3.4.4 Troubleshooting

You can access the trouble shooting guide from *Start > Programs > Atmel > Trouble Shooting Guide*.





## Section 4

# Using System Designer

This tutorial will guide you through the required steps for designing and programming AT94K series devices using System Designer.

### 4.1 Preparing the Example Files

Before starting the tutorial, a few preparations need to be performed:

1. Download `STK594.ZIP` from the FPSLIC® software page of the Atmel web site and copy `STK594.ZIP` to `C:\SystemDesigner\Designs`.
2. Extract the contents of the `STK594.ZIP` file to `C:\SystemDesigner\Designs`. The contents of the zip file are shown in Table 4-1.

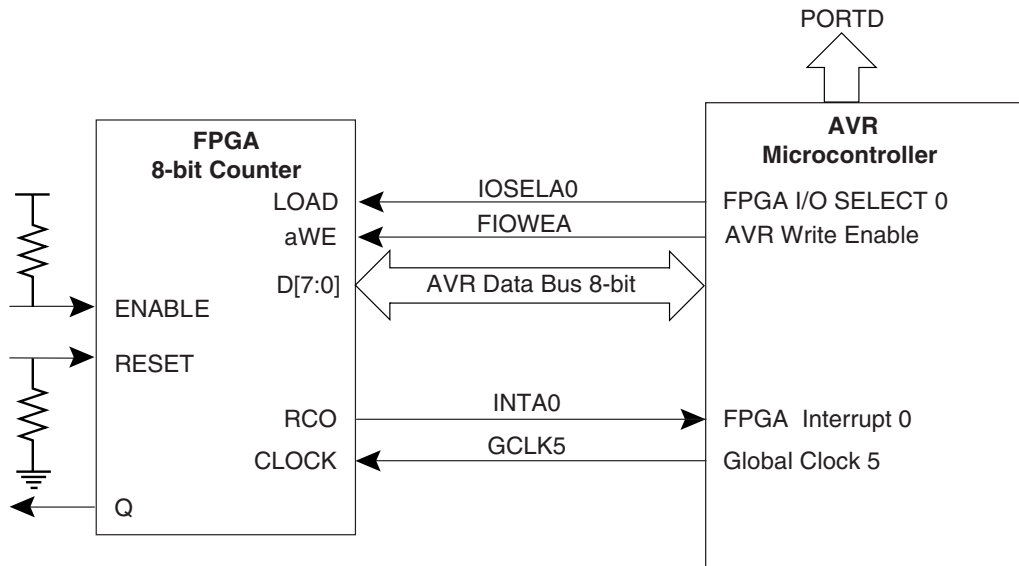
**Table 4-1.** `STK594.ZIP`

File	Description
<code>AT94KDEF.INC</code>	Atmel AVR Assembler AT94K FPSLIC Include File
<code>COUNTER.PIN</code>	FPGA Pin Lock File
<code>COUNTER.V</code>	Top Level FPGA Verilog® Counter Source File
<code>COUNTER.VHD</code>	Top Level FPGA VHDL Counter Source File
<code>STK594.ASM</code>	Atmel AVR Assembler Source File
<code>COUNTER.ATT</code>	FPGA I/O Attribute File

## 4.2 Description

The design in this tutorial is composed of a simple AVR<sup>®</sup> microcontroller program and a loadable counter implemented in the FPGA. When the counter reaches the terminal value, an interrupt to the microcontroller will be generated using the counter's carry-out (RCO) signal. The interrupt is active Low and must be held for three clock cycles prior to its acknowledgement by the microcontroller. During the Interrupt Service Routine (ISR) the microcontroller increments the count of interrupt occurrences and places the incremented data on PORTD and the AVR-FPGA Data Bus, triggering the counter's LOAD signal. Once the counter has been loaded, counting will commence and the process will be repeated. Figure 4-1 shows a simplified block diagram of the tutorial design.

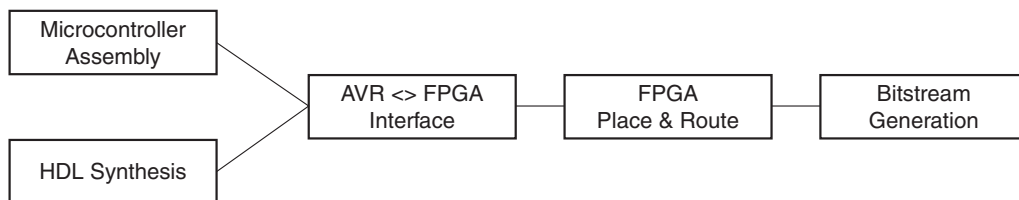
**Figure 4-1.** Tutorial Design Block Diagram



## 4.3 Design Flow

The design presented in this tutorial, only performs the required steps for designing and programming an AT94K series device. For more information on the optional steps (i.e. Simulation and Co-verification) please consult the FPSLIC application notes available on the Atmel web site. Figure 4-2 outlines the design flow followed in this tutorial. For more information on a specific step, consult the appropriate section within this tutorial.

**Figure 4-2.** Design Flow





## 4.4 Creating a Project

The New Project Wizard allows you to choose your Project Directory, Target Device, and desired Tool Flow.

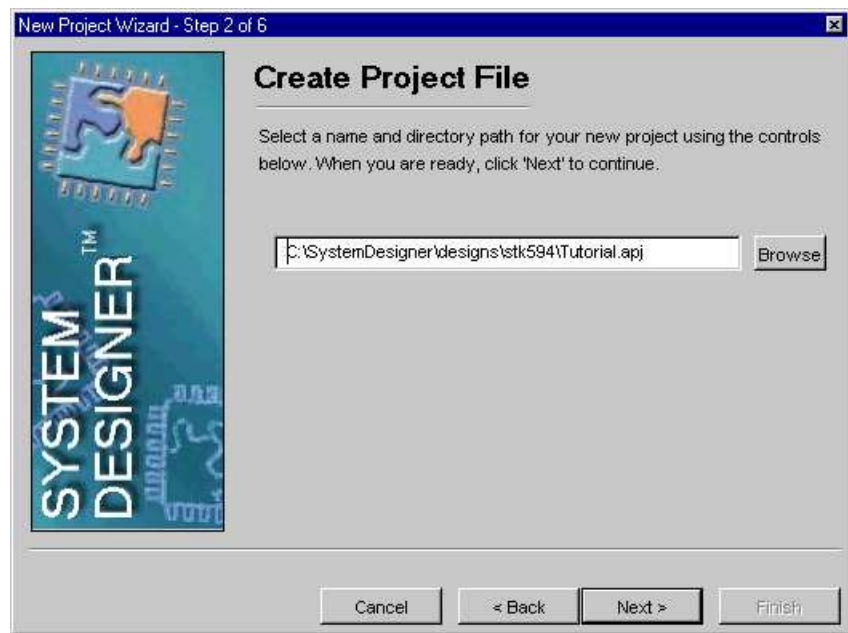
1. Launch System Designer from the desktop icon, or by pointing to *Start > Programs > Atmel > SystemDesigner*.
2. Create a new project by selecting *New* from the *Project* menu and then pressing the *New Project Wizard* button, see Figure 4-3.

**Figure 4-3.** New Project Wizard Window - Step 1 of 6



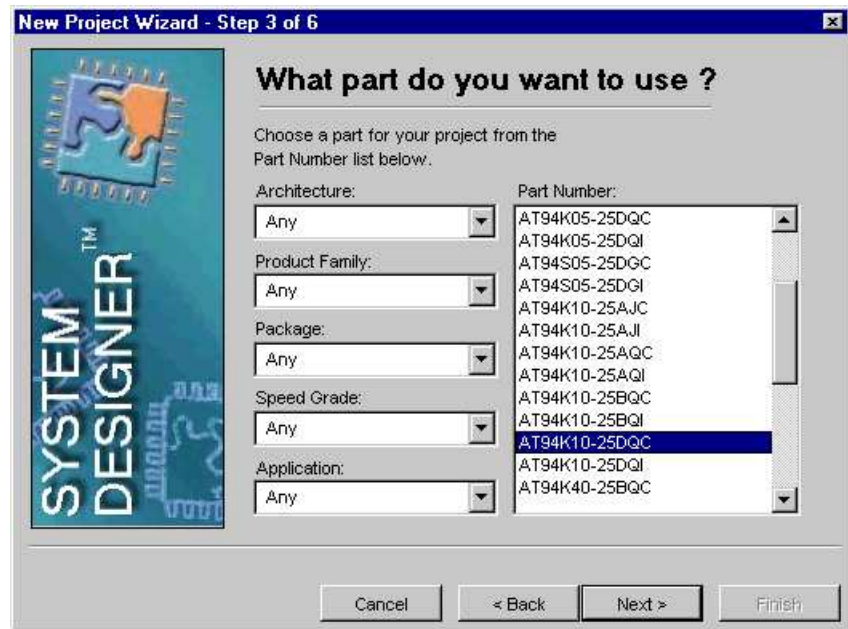
3. Press *Next*. The window to create a project file opens, see Figure 4-4.

**Figure 4-4.** New Project Wizard Window - Step 2 of 6



- Set the Project Directory to C:\SystemDesigner\Designs\STK594, name the project TUTORIAL and press *Next*. The part selection window appears, see Figure 4-5.

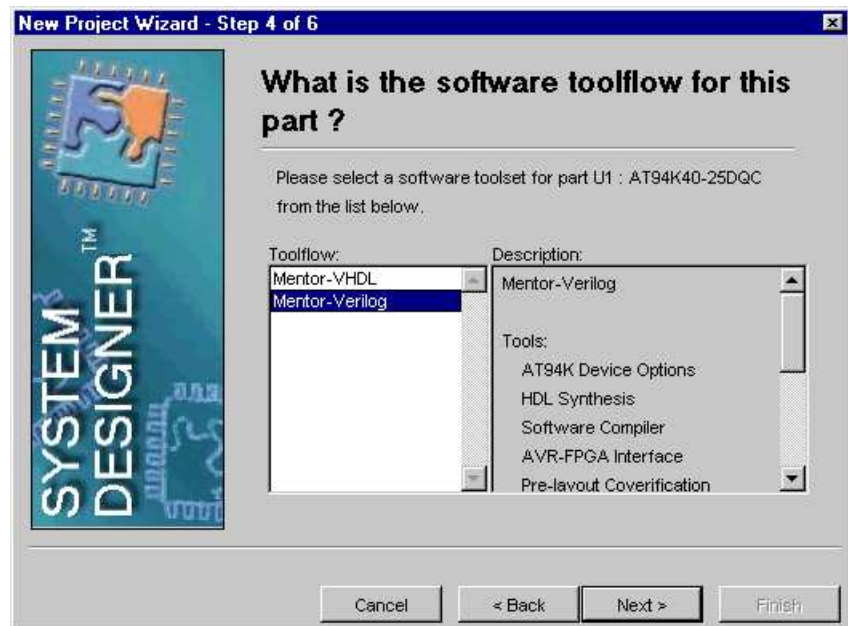
Figure 4-5. New Project Wizard Window - Step 3 of 6



- Select *AT94K10AL-25DQC*<sup>(1)</sup> from the parts list as this is the part found on the STK594 development board, and press *Next*. The software tool flow window opens, see Figure 4-6.

**Note:** 1. Some boards use AT94K40AL-25DQC devices.

Figure 4-6. New Project Wizard Window - Step 4 of 6



6. Select either *Mentor - VHDL* or *Mentor - Verilog* as the Tool Flow and press *Next*. The add parts window opens, see Figure 4-7. For this tutorial, the instructions will assume *Mentor - Verilog* has been selected.

**Figure 4-7.** New Project Wizard Window - Step 5 of 6

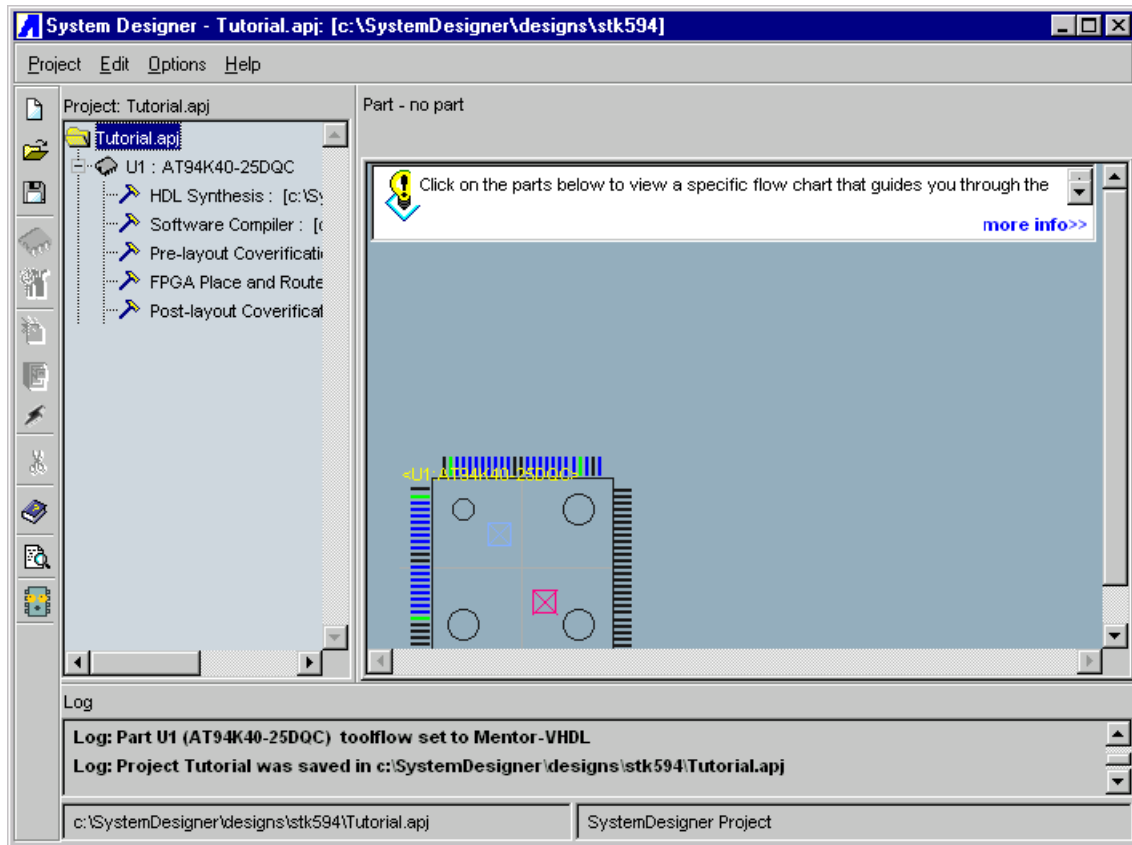


**Figure 4-8.** New Project Wizard Window - Step 6 of 6



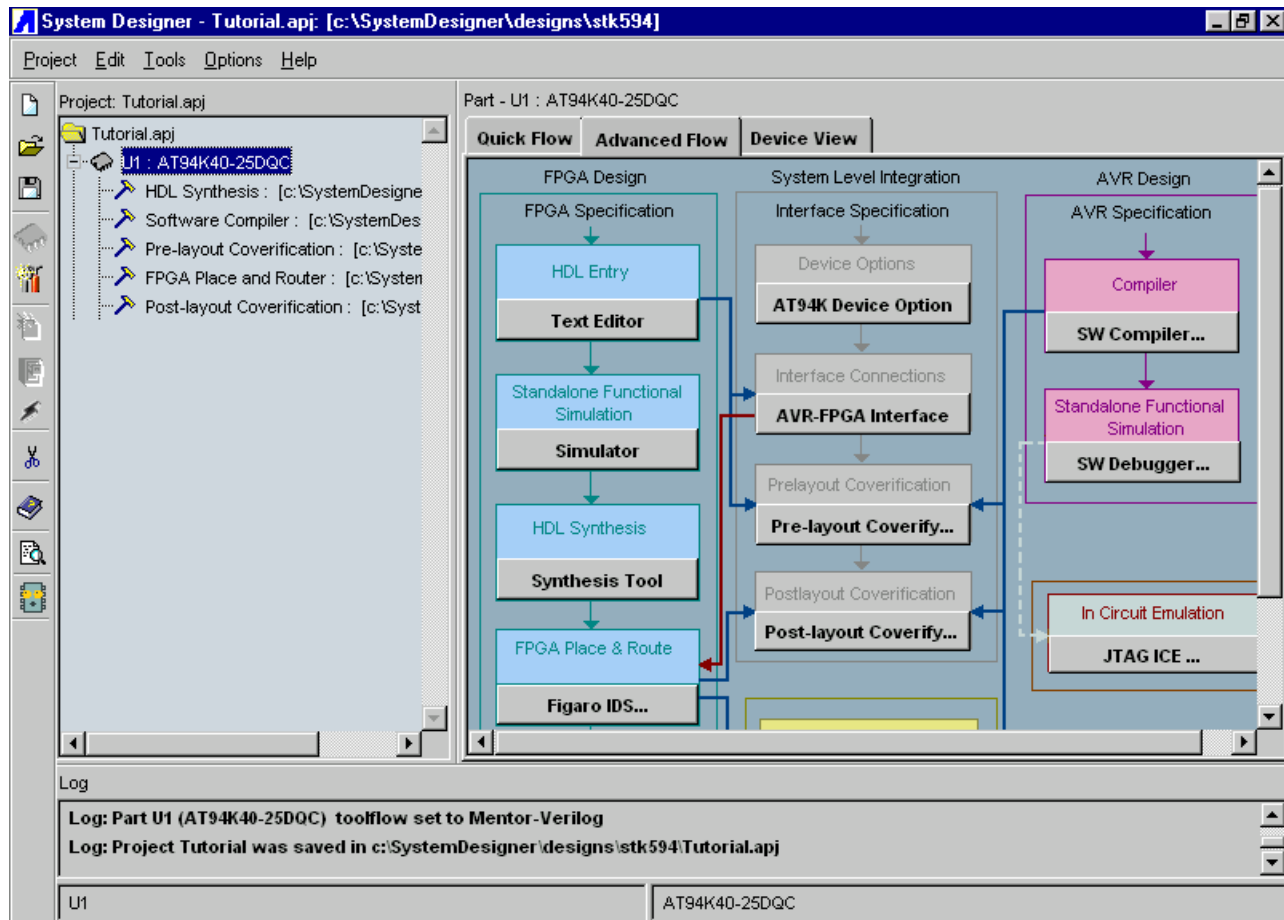
7. Press *Finish* to exit the *New Project Wizard*. The project window now contains the `TUTORIAL` design, see Figure 4-9.

Figure 4-9. Project Window



8. From the System Designer desktop, click on the *Part Graphic* (see Figure 4-9) to switch to the *Design Flow Manager*, see Figure 4-10.

Figure 4-10. Design Flow Manager



The Design Flow Manager shows the steps available for designing with Atmel FPSLIC devices. The red and blue arrows show the dependencies between the various stages of development. This tutorial will only show the minimum steps in order to complete a design. The remainder of the steps involves simulation and co-verification. For further information on simulation and co-verification, please consult the “Quick Start Tutorial” available on the Atmel web site.

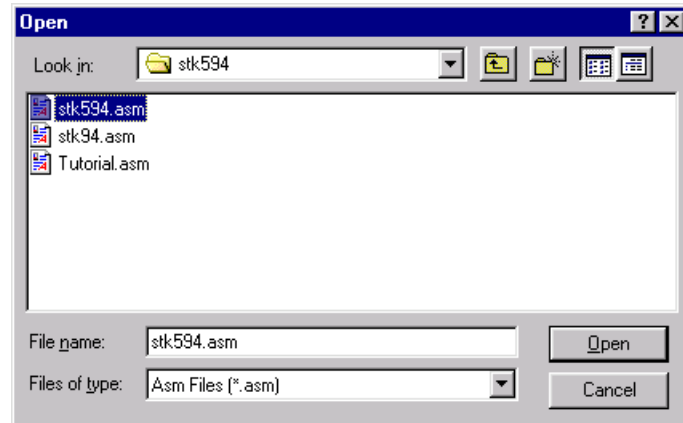


## 4.5 Assembling the Microcontroller Source Code

The Atmel AVR Assembler translates assembly source code into object code. The generated object code can then be used as an input to a simulator, emulator such as the Atmel AVR JTAG In-Circuit Emulator (ICE), or used to program the target device. The Assembler generates fixed code allocations, therefore no linking is necessary.

1. Press the *SW Compiler* button to open the Atmel AVR Studio®.
2. Click on *NO* when prompted to create a new file. The window to open an existing file opens up.
3. Browse to the `C:\SystemDesigner\Designs\STK594` directory and select `STK594.ASM`.

**Figure 4-11.** Open File stk594.asm



4. Press *Build* and close the Atmel AVR Studio if assembly was successful.

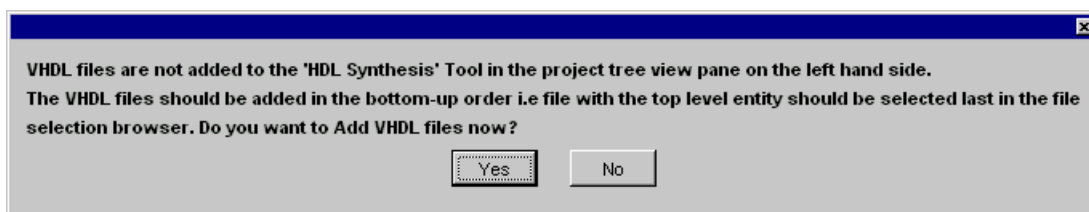
**Note:** If assembly was not successful, make sure the include file `AT94KDEF.INC` is in the design directory. Only AT94K devices with a “J” label support JTAG ICE debugging.

For design entry using assembly language, consult the AT94K datasheet for a summary of instructions supported by the FPSLIC devices. The complete “AVR Instruction Set Nomenclature” describes each instruction in detail and has been installed as part of the System Designer Tool. The AVR Instruction Set Nomenclature and FPSLIC datasheet can be accessed from the Help menu (from the System Designer window) and choosing *Online Resources*.

## 4.6 Synthesizing the FPGA Source File

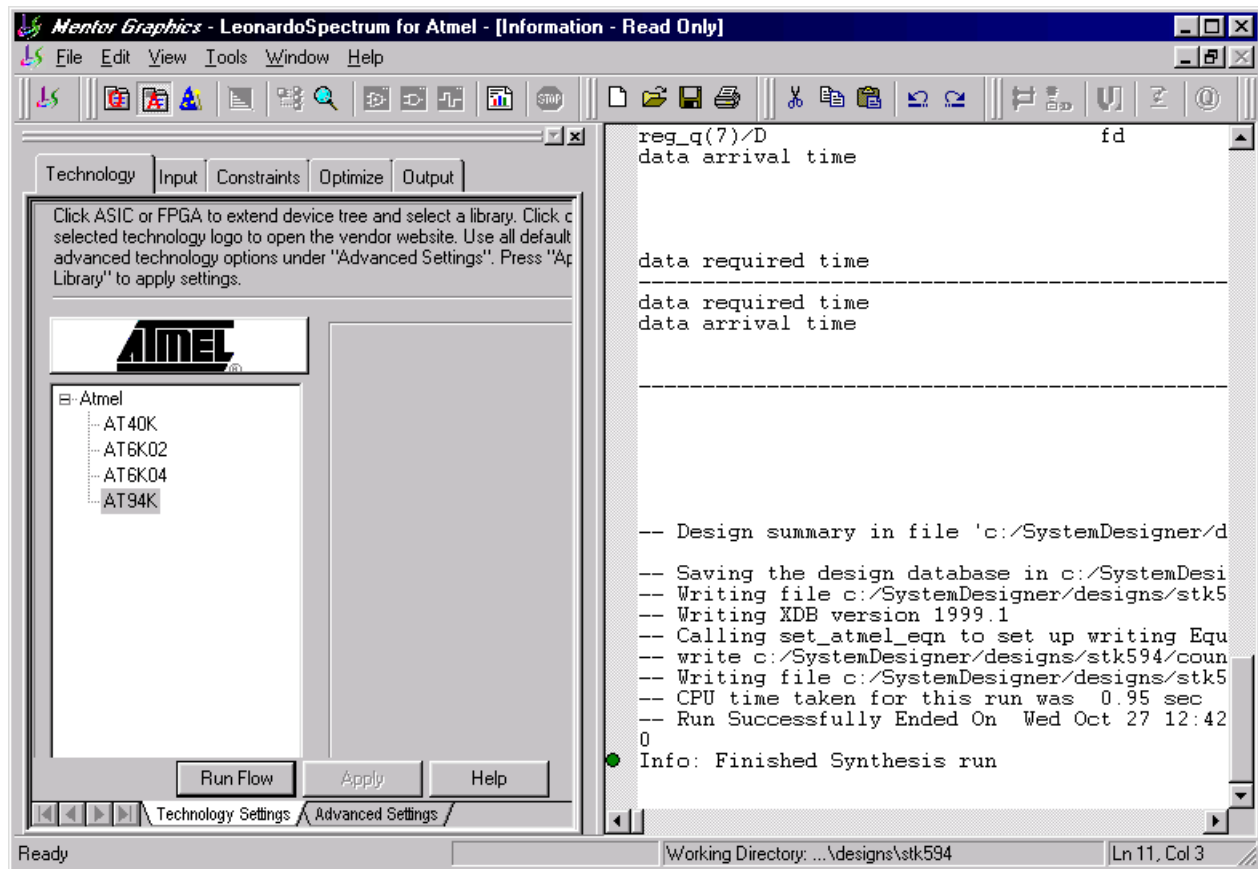
1. Synthesis translates the VHDL or Verilog source code into gate-level technology-specific file for use with the target FPGA Place and Route tool.
2. Press the **Synthesis Tool** button. A dialog box to add VHDL files appears, see Figure 4-12.

**Figure 4-12.** Add VHDL Files Dialog Box



3. Press **yes**. A file selection window appears.
4. Select `COUNTER.V` and press *Open*. LeonardoSpectrum opens.
5. Leonardo® automatically selects Atmel AT94K as the *Technology* and lists `COUNTER.V` under *Input*. Leonardo also lists `COUNTER.edf` under *Output*.
6. Press *Run Flow*. Figure 4-13 shows a successful synthesis.

**Figure 4-13.** Leonardo Spectrum, Successful Synthesis



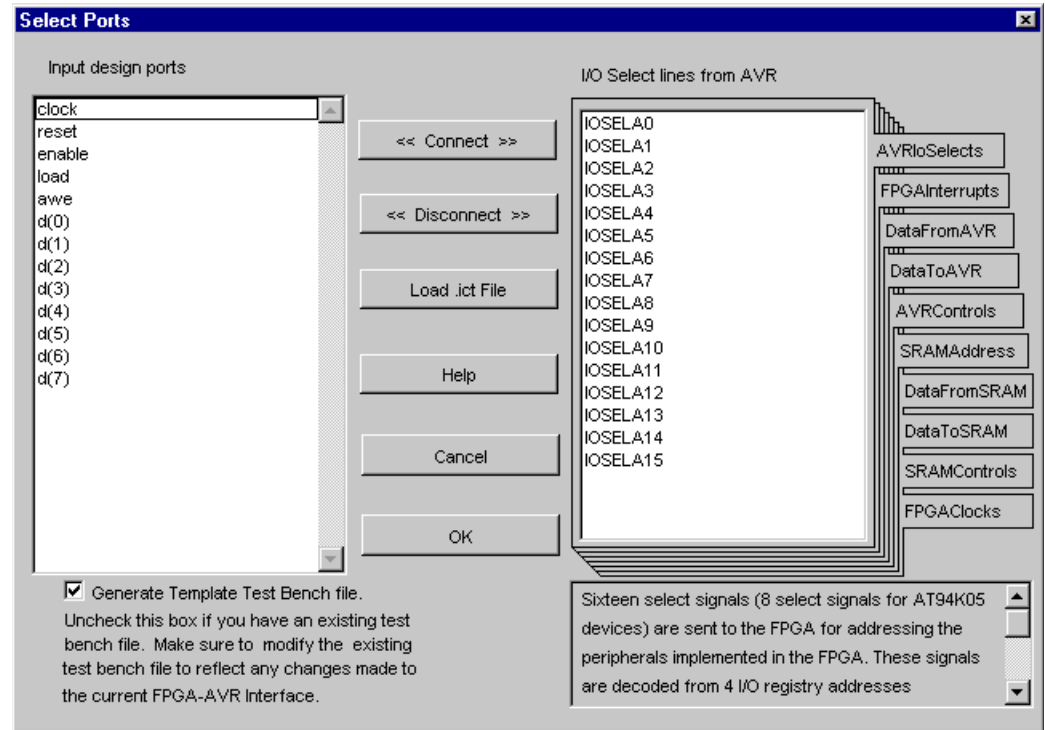
7. Close Leonardo Spectrum, when prompted to save your project press *No*.

## 4.7 AVR-FPGA Interface

The AVR-FPGA Interface dialog provides a means for making the connections between the embedded FPGA and AVR microcontroller.

1. Press the *AVR-FPGA* Interface button.
2. Select *counter* and press *OK* when prompted for Top-Level Entity. The *Select Ports* dialog appears, see Figure 4-14.

**Figure 4-14.** Select Ports Dialog



3. Select the *AVRloSelects* tab on the right-hand side of the dialog box.
4. Select the *LOAD* signal from the *Input Design Ports* and then select *IOSELA0* from the *AVRloSelects*.
5. Press *Connect* to connect the counter's *LOAD* signal to *FPGA-AVR I/O Select 0*.
6. Connect the remaining inputs and outputs as shown in Table 4-2.

**Table 4-2.** FPGA-AVR Interface Connections

FPGA I/O	FPGA-AVR I/O	Select Ports Tab
LOAD	IOSELA0	AVRloSelects
RCO	INTA0	FPGAInterrupts
D(7:0)	ADINA(7:0)	DataFromAVR
aWE	FIOWEA	AVRControls
Clock	GCLK5	FPGAClocks

7. Uncheck *Generate Template Test Bench File* on the bottom left-hand side of the *Select Ports* dialog. Since we are not performing co-verification, it is not necessary to generate the pre-layout test bench file.
8. Press *OK*.