# FPSLIC™ System Designer™ 3.0

# Quickstart Tutorial

ATMEL®

# Table of Contents

# Section 1

# Introduction

This tutorial is intended for first time users. Software updates can be downloaded from the Atmel web site, at http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2752.

When running the FPGA and AVR stand-alone designs, the design must be combined at the end using System Designer. If this method is used, the interface must be clearly defined prior to the design stage, since the FPGA-AVR interface is fixed in silicon.

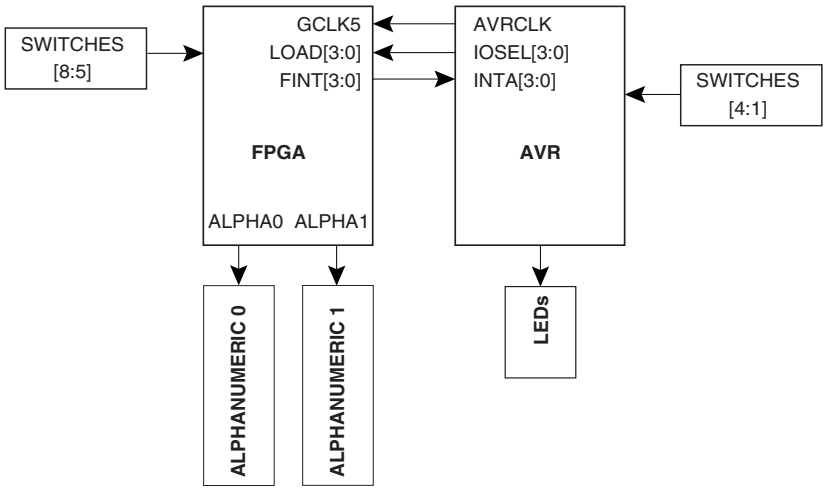
## 1.1 Design Description

The design in this tutorial shows the user how to use the switches to invoke different interrupt service routines.

Four switches (SW1 – SW4) are connected to the AVR external interrupts. Pressing the switch will send the signal to an AVR external interrupt. When an interrupt occurs, it jumps to the corresponding program segment (C or Assembly) to execute the code that will be displayed on the LEDs.

Four switches (SW5 – SW8) are connected to the FPGA I/Os. When the switch is pressed, the signal will be sent to the FPGA interrupt (INTA0 – INTA3). When the AVR receives an interrupt, it will jump to the interrupt service routine and send the load signal to the FPGA to display different characters on the Alphanumeric LEDs. The AVR CLK is connected to the FPGA Global Clock (GCLK 5), see Figure 1-1.

***Figure 1-1.*** FPGA-AVR Connections



## 1.2 Design Flow

The flow below provides a step-by-step explanation of `quickstart_lab`:

1. **Compile Assembly/C Code:** Wavrasm Assembler creates the `qs_avr.hex` file. Using ImageCraft Compiler creates the `qs_avrim.hex` (used for download) and `qs_avrim.cof` (used for debugging) files.
2. **AVR-FPGA Interface:** Defines the connections between the embedded FPGA and AVR Core; `qs_fpga.ict` is created.
3. **Pre-layout Co-verification:** Pre-layout co-verification provides a simulation of the FPGA design with the AVR code, including the timing information from the FPGA.
4. **Synthesize VHDL Code:** LeonardoSpectrum creates the `qs_fpga.edf` netlist file. This file is imported into the Figaro Place & Route tool.
5. **FPGA Place & Route Using Figaro:** `qs_fpga.edf` is imported into Figaro to place and route the design, `qs_fpga.bst` is created.
6. **Post-layout Co-verification:** Post-layout Co-verification provides a simulation of the FPGA design with the AVR code, including the timing information from the FPGA targeted to the architecture of your design.

## 1.3 Expected Results

There are four patterns that are shown on each set of LEDs depending on the switch pressed, see Table 1-1 and Table 1-2.

***Table 1-1.*** LED Patterns

| Switches | LED |
|---|---|
| 1 | Count Up |
| 2 | Count Down |
| 3 | Knightrider |
| 4 | Bounce |

***Table 1-2.*** Alphanumeric LED Patterns

| Switches | Alphanumeric LED |
|---|---|
| 5 | * |
| 6 | + |
| 7 | x |
| 8 | 0 |

# Section 2

# Step-by-Step Procedure

## 2.1 Setting Up the Example Files

The source files for this application note can be found within the `c:\SystemDesigner\Examples\AT94K\Coverify\2451.zip` file. Alternatively, the source files may be found in the FPSLIC System Designer Software card of the Atmel web site (http://www.atmel.com).

The contents of the zip file are shown in Table 1.

**Table 1.** 2451.zip Contents

| File | Description |
|---|---|
| `qs_fpga.vhd` | VHDL® Design File |
| `qs_fpga.pin` | FPGA Pin Assignments |
| `wave.do` | Command File for Hardware Simulation. This File is not Required for this Tutorial. |
| `qs_avr.asm` | Assembly File for the AVR Design |
| `qs_avrim.c` | C File for ImageCraft Compiler |
| `at94kdef.inc` | Include File Used with Assembly File |

Before starting the tutorial you need to set up a directory for training.

1. Create a directory under `c:\training\fpslic\lab1`.
2. Copy `2451.zip` to the `lab1` directory.

## 2.2 Installing ImageCraft™ Compiler

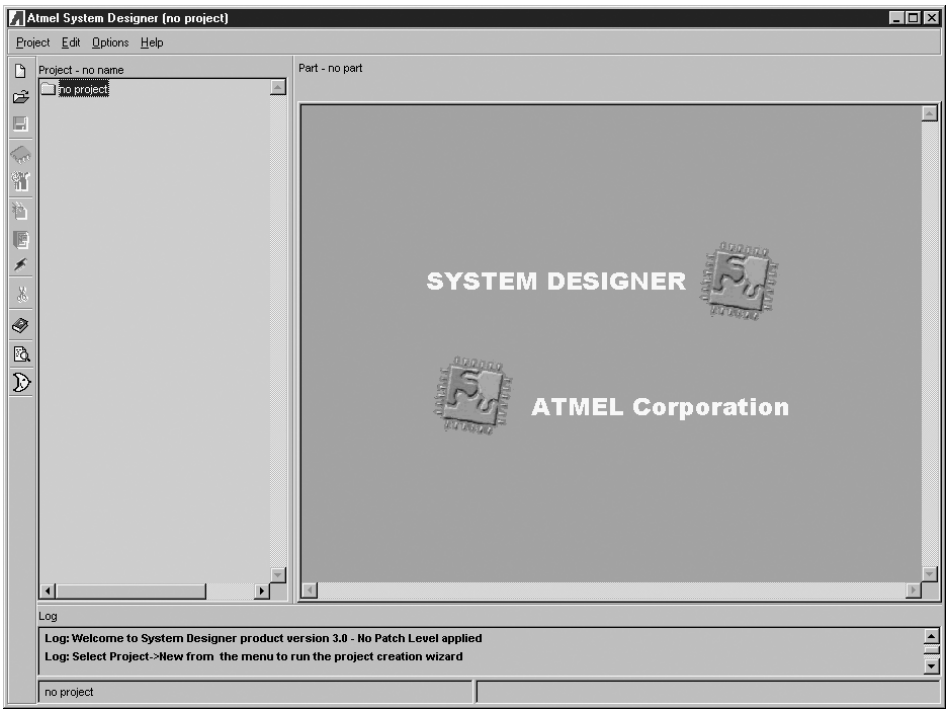
1. Insert the System Designer CD into the CD-ROM drive.
2. From the CD browser go to Install Products and select ImageCraft C Compiler.

The complete version of ImageCraft will run for 30 days after the installation. After the 30 days, it will ask you to register the application. If you do not register the application, ImageCraft can still be used, but it is limited to 2K of code space.

## 2.3 Setting Up the Project

1. Double-click on the System Designer icon on the desktop. System Designer opens, see Figure 2-1.

***Figure 2-1.*** System Designer Window

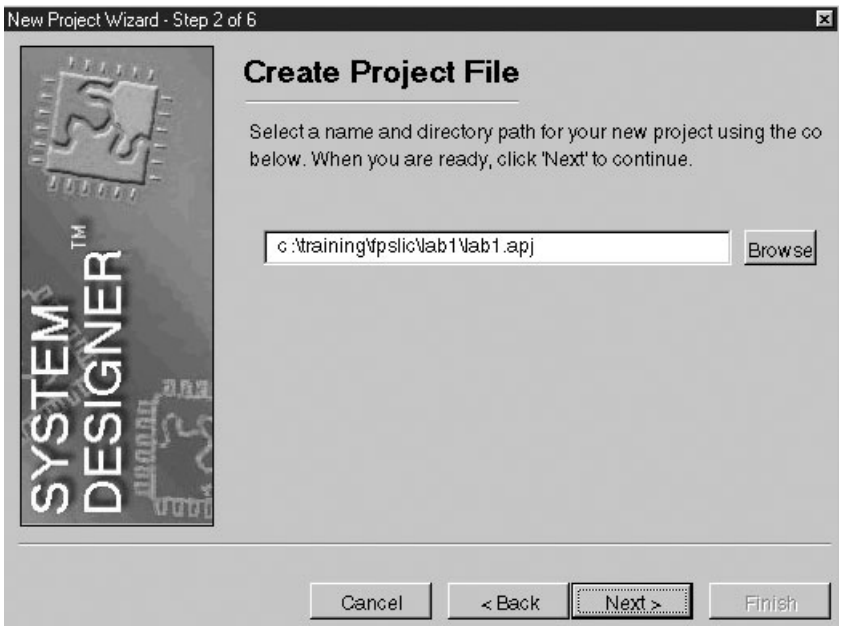



2. Go to the *Project* menu and select *New…* to create a new project. The *New Project Wizard* appears. The *New Project Wizard* allows you to select your project directory, select which part you want to target and set up the design tool flow, see Figure 2-2 to Figure 2-7.

***Figure 2-2.*** New Project Wizard – Step 1 of 6

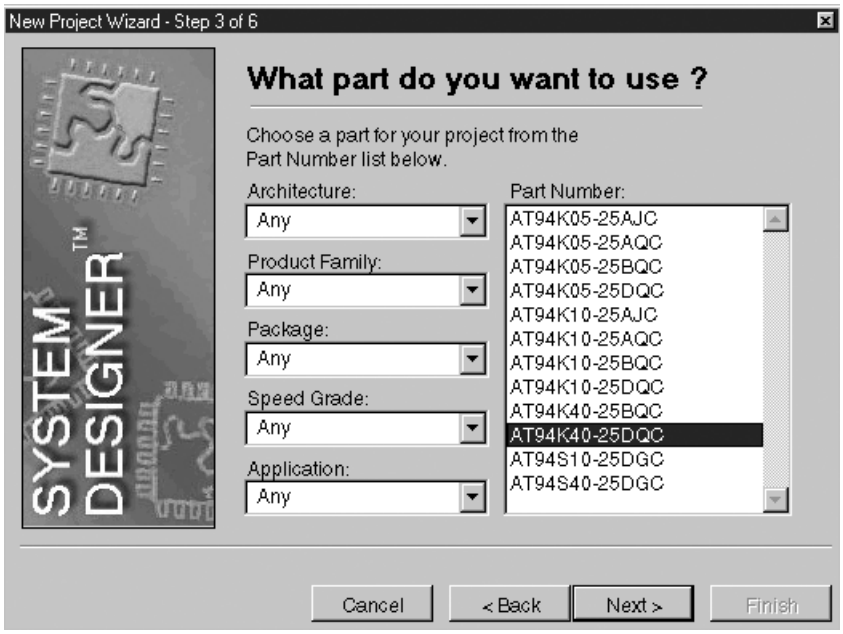
3. Press *Next >*. The *Create Project File* window opens, see Figure 2-3.

***Figure 2-3.*** New Project Wizard – Step 2 of 6




4. Change the directory to `c:\training\fpslic\lab1` and name the project `lab1.apj`. Press *Next >*. The part selection window appears, see Figure 2-4.

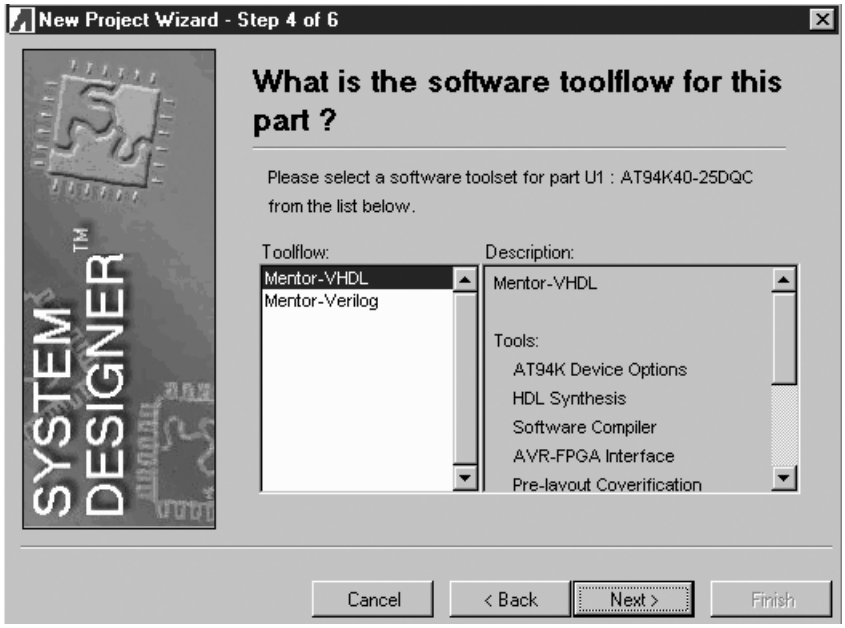
***Figure 2-4.*** New Project Wizard – Step 3 of 6




5. Select *AT94K40-25DQC* from the *Part Number* box. This is the part used in the ATSTK94 starter kit.

6. Press *Next >*. The software tool flow window appears, see Figure 2-5.

***Figure 2-5.*** New Project Wizard – Step 4 of 6




7. Select *Mentor-VHDL* from the tool flow box.
8. Press *Next >*. The add more parts window appears, see Figure 2-6.

***Figure 2-6.*** New Project Wizard – Step 5 of 6


It is possible to add multiple parts to work on, but for the purpose of this tutorial only one part will be used.

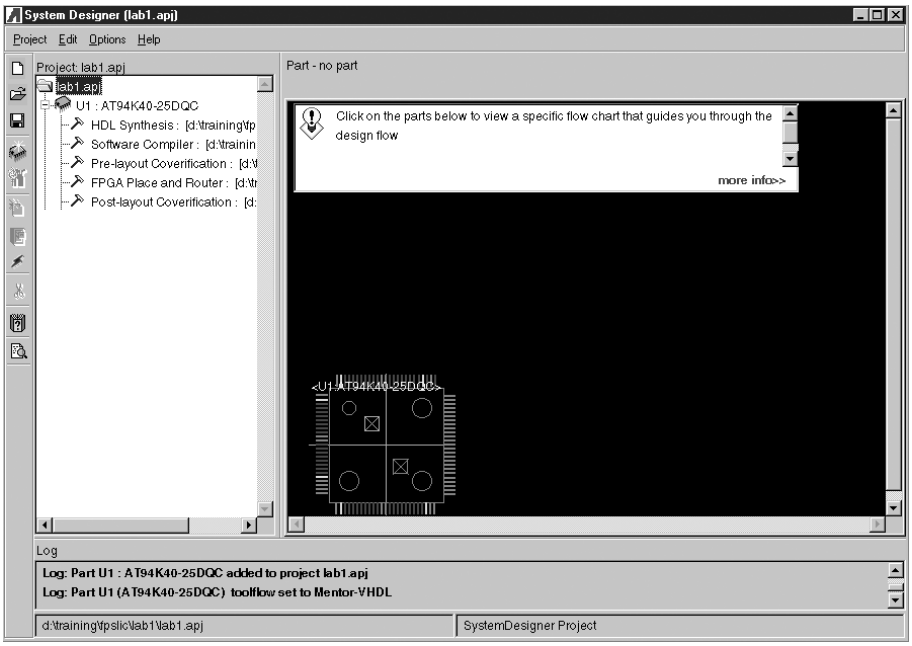
9. Press *Next >*. The last window of the wizard appears, see Figure 2-7.

***Figure 2-7.*** New Project Wizard – Step 6 of 6


10. Press *Finish* to exit the wizard.

    The project window now contains `lab1.apj` and the part window displays the part selected, see Figure 2-8.

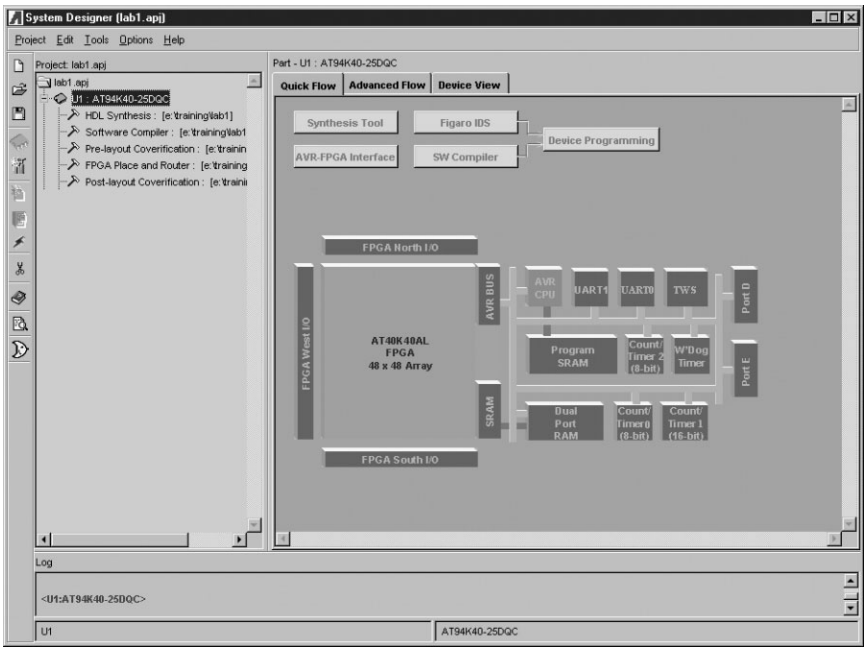
***Figure 2-8.*** System Designer Window – lab1.apj




Clicking on *more info >>* opens the online help.

11. Click on the graphic in the part window to display the *Quick Flow View*, see Figure 2-9.

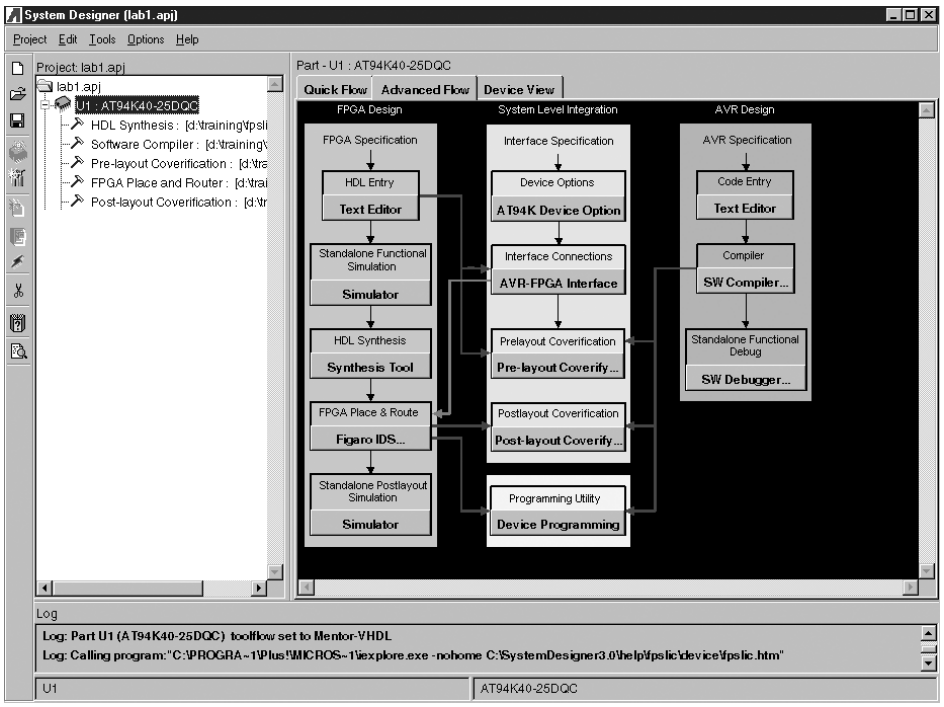
***Figure 2-9.*** Quick Flow View




## 2.4 Advanced Flow

The Advanced Flow view shows the steps available for designing with an FPSLIC device. The arrows on the diagram show the dependencies between the steps, see Figure 2-10.

***Figure 2-10.*** Advanced Flow



## 2.5 Compiling the AVR Assembly File

For design entry using assembly language refer to the AVR Instruction Set summary in the AT94K series datasheet. The instruction set summary describes the details for each of the supported functions in the FPSLIC.

1. Press the *SW Compiler* button to open *wavrasm*.
2. Go to the *File* menu and select *Open File…*
3. Navigate to the correct folder `c:\training\fpslic\lab1` and select `qs_avr.asm`.
4. Go to the Assemble menu and press *Assemble*. A report window opens.

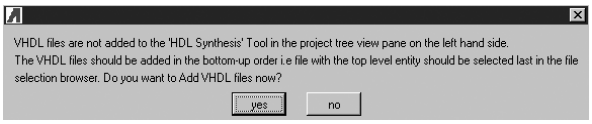
If successful, close the assembler and return to System Designer.

If the design is not assembled successfully, the message window will display the error messages. Check if `at94kdef.inc` is missing in your design directory; if this is the case, extract `at94kdef.inc` from `2451.zip` and save it to the lab1 folder.

## 2.6 Synthesizing the FPGA file

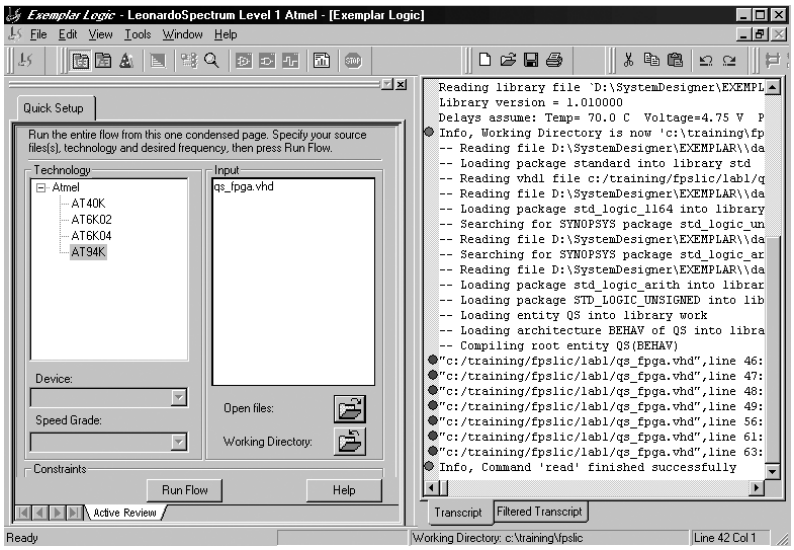
1. Press the *Synthesis Tool* button. A dialog box to add VHDL files appears, see Figure 2-11.

***Figure 2-11.*** Add VHDL Files Dialog Box



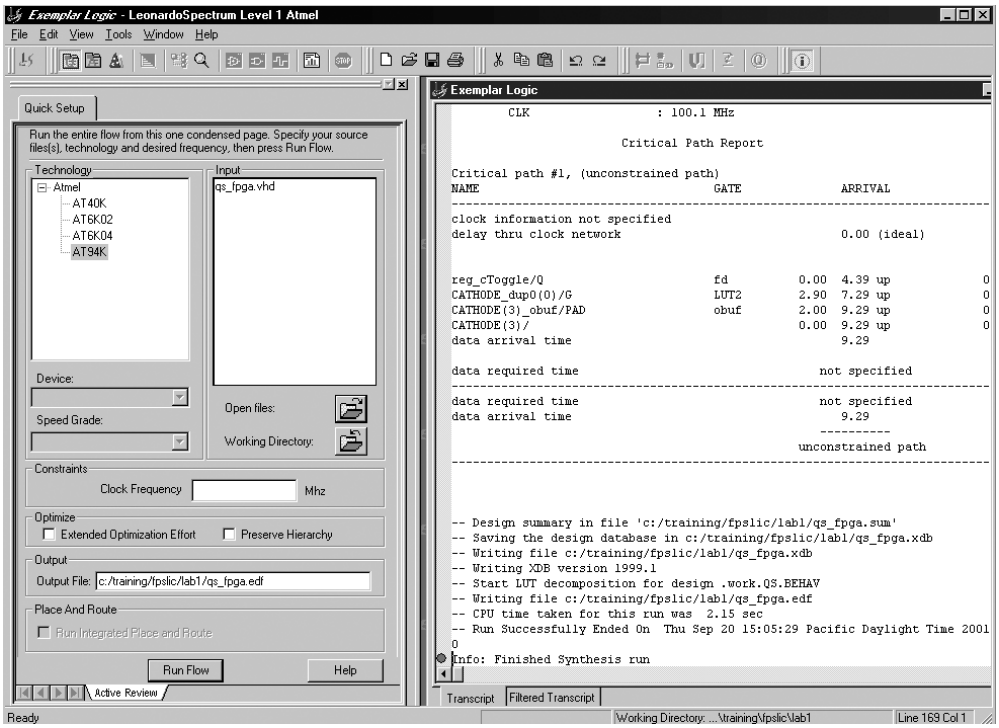

2. Press *Yes*. A file selection window appears.
3. Select `qs_fpga.vhd` and press *OK*. LeonardoSpectrum opens, see Figure 2-12

***Figure 2-12.*** LeonardoSpectrum




4. Select *AT94K* as the *Technology*. Leonardo® automatically lists `qs_fpga.vhd` under input files and lists the output file name as `qs_fpga.edf`. `qs_fpga.edf` will be imported into Atmel's Place & Route tool (IDS).
5. Press the *Run Flow* button. Leonardo shows the successful synthesis, see Figure 2-13.

***Figure 2-13.*** Synthesis Results

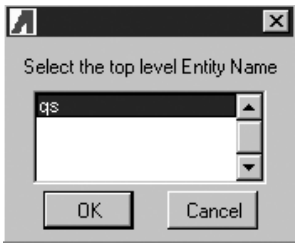



6. It is not required to save the project. Close Leonardo and return to System Designer.

## 2.7 AVR-FPGA Interface

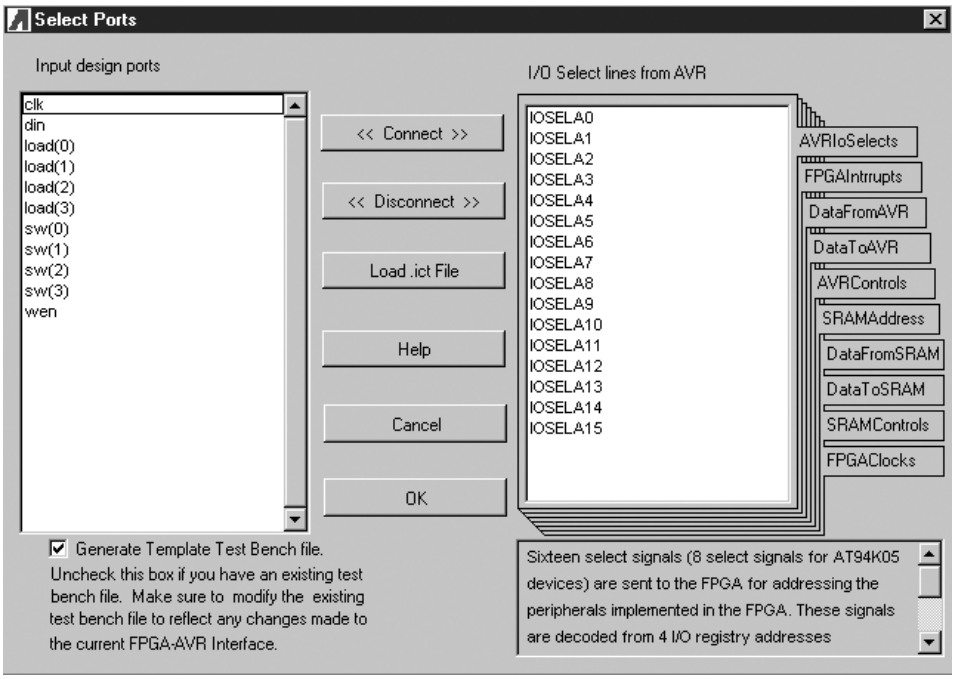
1. Press the *AVR-FPGA Interface* button. A dialog box to select the top-level entity name appears, see Figure 2-14.

***Figure 2-14.*** Top-level Entity Name Dialog Box




2. Press *OK* to accept *qs* as the top-level entity name. The *Select Ports* window appears, see Figure 2-15.

***Figure 2-15.*** Select Ports Window



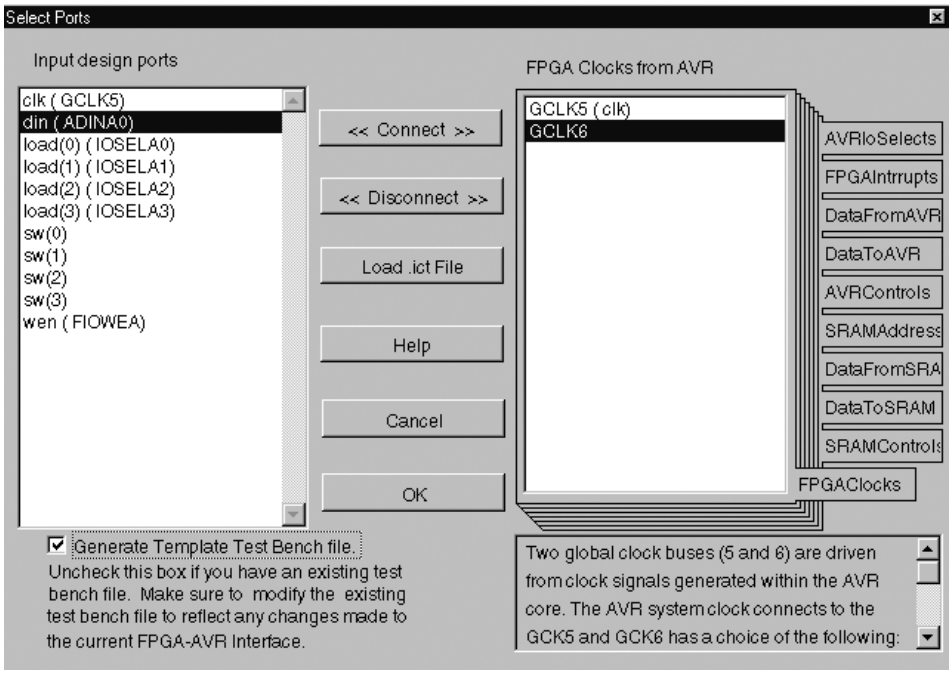

3. Select the *AVRIoSelects* tab on the right.
4. Select *load(0)* from the *Input Design Ports* list and *IOSELA0* from the *AVRIOSelects* list.
5. Press *<<Connect>>*. Table 2-1 shows all input and output connections for the design.

***Table 2-1.*** Input and Output Connections

| Tab | Input design port | I/O Select Lines from AVR |
|---|---|---|
| AVRIoSelects | load0 – load3 | IOSELA0 – IOSELA3 |
| FPGAInterrups | fints0 – fints3 | INTA0 – INTA3 |
| DataFromAVR | Din | ADINA0 |
| AVRControls | wen | FIOWEA |
| FPGAClocks | clk | GCLK5 |

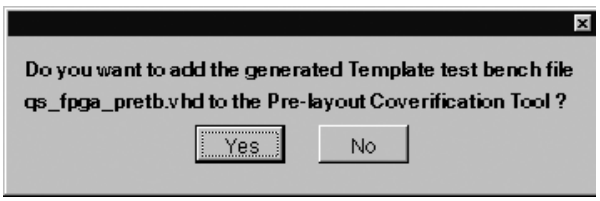
6. Perform all the connections above, the resulting window is shown in Figure 2-16.

***Figure 2-16.*** Select Ports Window – Global Clock Buses




7. The *Generate Template Test Bench file* option generates the pre-layout test bench file for co-verification.
8. Press *OK*. A dialog box to add the generated template test bench file appears, see Figure 2-17.

***Figure 2-17.*** Add Generated Template Dialog Box

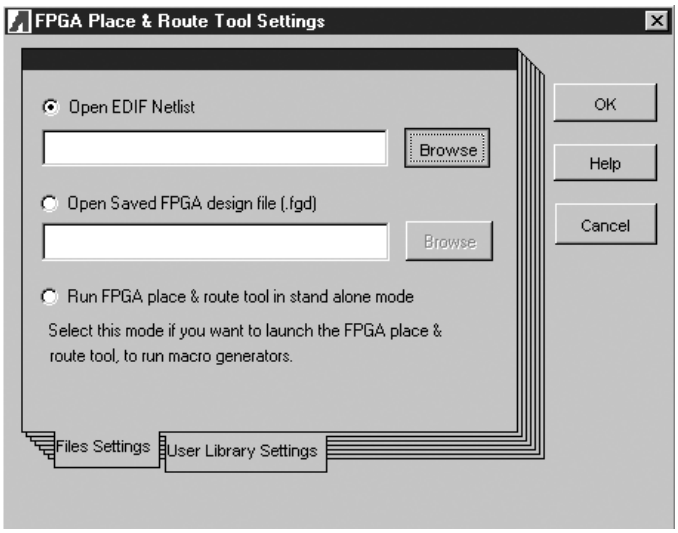



9. Select *Yes*.

## 2.8 FPGA Place & Route

1. Press the *Figaro IDS* button. The *FPGA Place & Route Tool Settings* dialog box opens, see Figure 2-18.

***Figure 2-18.*** FPGA Place & Route Settings Dialog Box – Files Settings

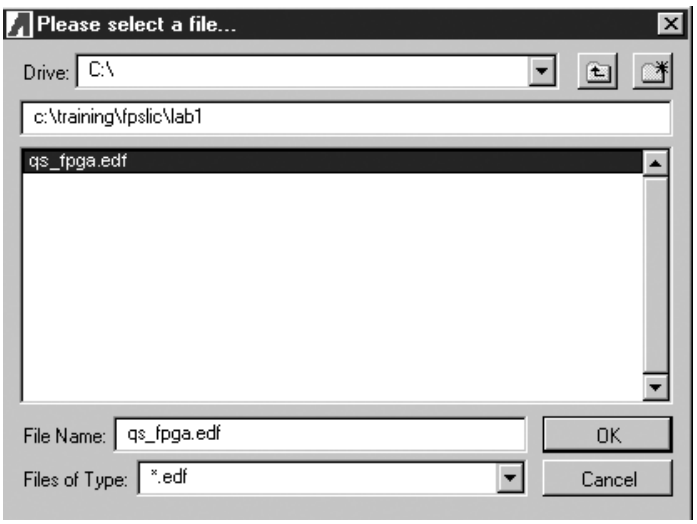



The *Files Settings* tab allows one of three options:

– Open EDIF Netlist
– Open the saved FPGA design file (`*.fgd`)
– Run FPGA Place & Route tool in stand-alone mode

2. Select the default option (*Open EDIF Netlist*) from the *FPGA Place & Route Tool Settings* dialog box and press the *Browse* button. A file selection window opens, see Figure 2-19.

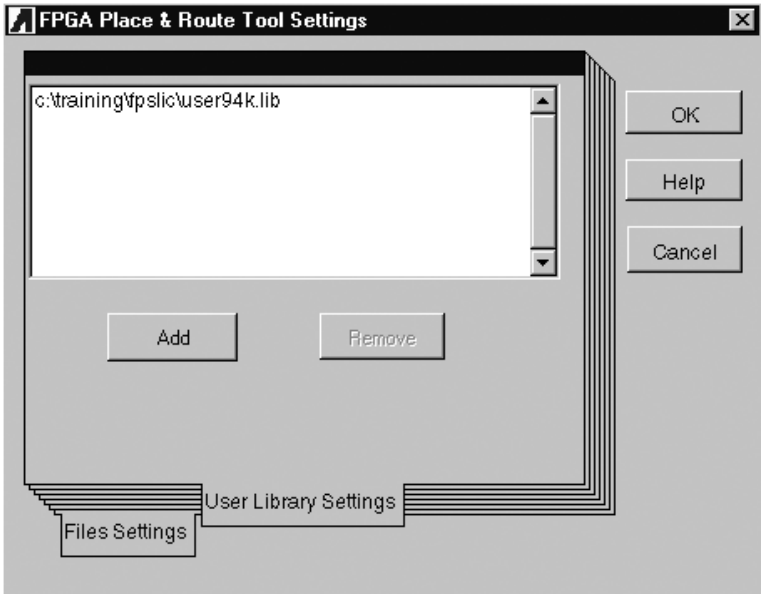
***Figure 2-19.*** File Selection Window




3. Select `qs_fpga.edf` and press *OK* to return to the *FPGA Place & Route Tool Settings* dialog box.

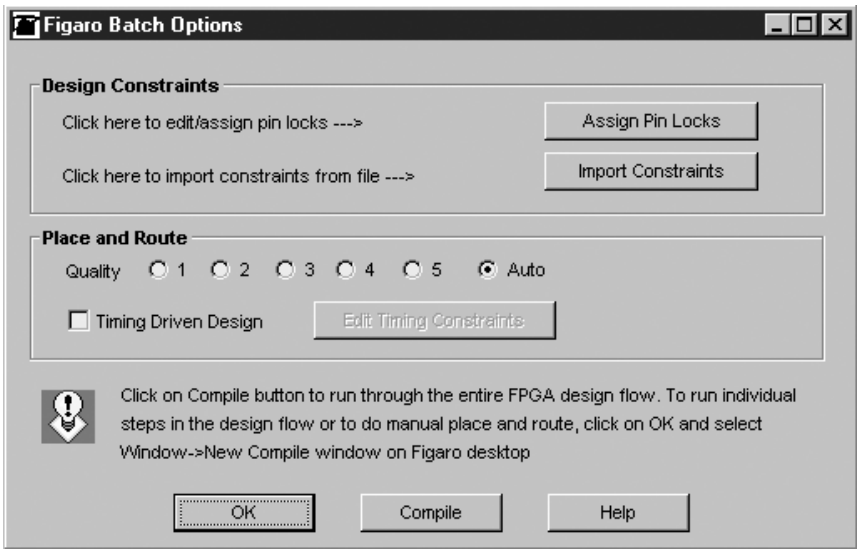
4. Click on the *User Library Settings* tab, see Figure 2-20. This tab allows the user to set up user libraries for the FPGA Place & Route tool.

***Figure 2-20.*** FPGA Place & Route Settings Dialog Box – User Library Settings




5. Press *OK* in the *FPGA Place & Route Tool Settings* dialog box.
6. Figaro runs through *Open*, *Map* and *Add Parts* automatically. A *Figaro Batch Options* dialog box opens, see Figure 2-21.

***Figure 2-21.*** Figaro Batch Options Dialog Box




The *Design Constraints* box has two options available: *Assign Pin Locks* and *Import Constraints*; we will use the latter.

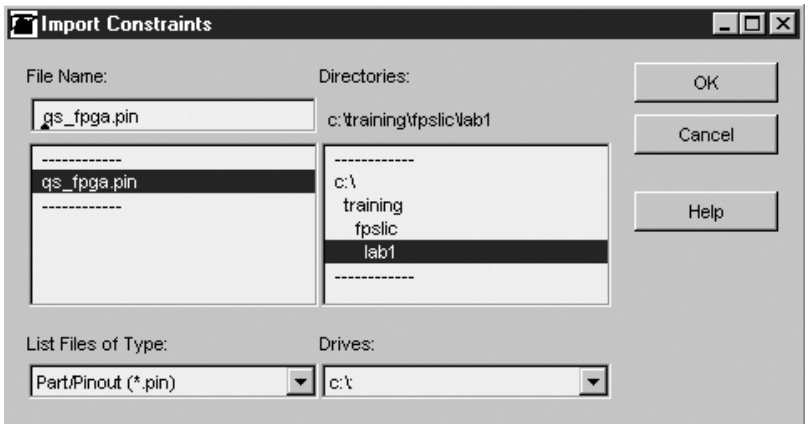
- *Assign Pin Locks* – The design constraints allow you to assign pin locks to the specified I/Os of the device. If you select *Assign Pin Locks*, follow the steps below:
  - Select the *Assign Pin Locks* tab in the Figaro Batch Options.
  - Select the *Design I/Os and Usable Pins* one by one and press *Lock*.

  You can export a pin file after choosing the option to assign pin locks by going to the *File* menu and choosing *Export* - the export option is only available if you are in the Parts window. The pin locks are automatically reloaded through `yourfile-name.rct` file when you are finished. If the file was not created, go to the *Options* menu and select *Options > Design Constraints*, check the *Auto-import Repeat Constraints (*.ict)* file and press *OK*.

- *Import Constraints* – Figaro can read pinout constraints from a constraints file named `yourfilename.pin`. Since we have provided the pin constraints file, we import them:

7. Click on *Import Constraints*. The *Import Constraints* dialog box opens, see Figure 2-22.

***Figure 2-22.*** Import Constraints Dialog Box




8. Select *Part/Pinout (*.pin)* under *List File Types*.
9. Select `qs_fpga.pin` to import the pin constraint file.
10. Press *OK*.

The *Place & Route* box has two options are available: *Quality* and *Timing Driven Design*.

- *Quality* – Use the quality buttons to set the trade-off between Figaro's speed and the efficiency of the result, see the online help for more details.
- *Timing Driven Design* – Check this box if you want Figaro placement and routing to take account of critical paths in the design, refer to the technical reference guide under `c:/SystemDesigner/doc/Tutorial.pdf`.

11. Take the default quality settings and timing-driven options.
12. Press the *Compile* button, this will take you through the placing and routing.
13. When the *Compile* button turns green, go to the *Window* menu and select *New Compile Window*. This displays the contents of the selected part and you can view layout details of the device at the logic module and interconnect level.

14. Maximize the compile window.

    Choose one of these options to zoom in:

    – Go to the *View* menu and select *Zoom to Area*, or

    – Press *F7*

    Choose one of these options to zoom out:

    – Go to the *View* menu and select *Zoom Out*, or

    – Press *F8*

15. Go to the *File* menu and select *Exit*.

16. When asked, choose *yes* to save the design. Figaro will close.

This allows you to return to the existing layout and setup again later. You could do this to check or modify the design timing if any issues were found during post-layout co-verification or hardware testing.

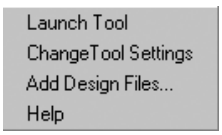
## 2.9 Co-verification

This section provides complete instructions on how to perform co-verification of the FPGA hardware and the AVR software using the System Designer tool suite. It also explains how to change the compiler settings from the default assembler to the ImageCraft compiler.

### 2.9.1 Changing the Compiler Tool Settings

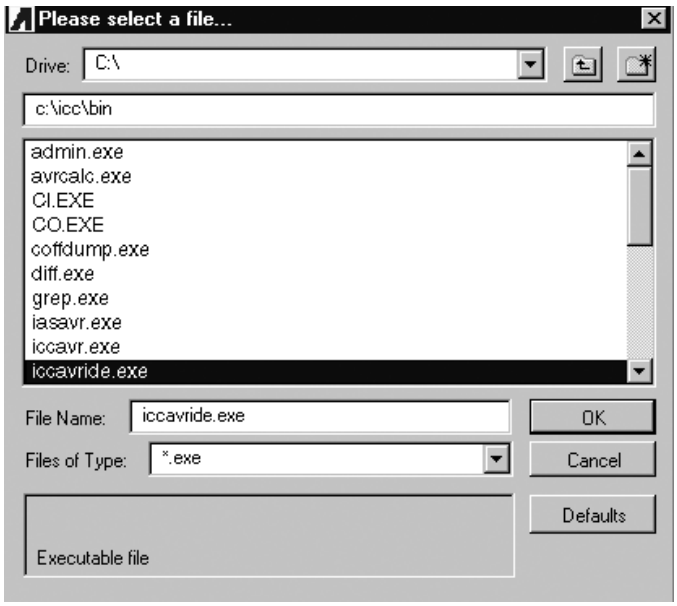
1. Right-click on the *SW Compiler* button, a pop-up menu appears, see Figure 2-23.

***Figure 2-23.*** SE Compiler Right-click Menu



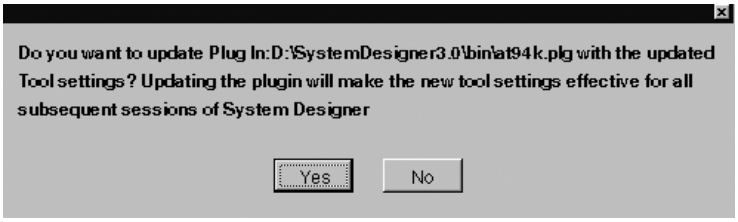

2. Select *Change Tool Settings*. A file browser window appears. Navigate to `c:\icc\bin\Iccavride.exe`, see Figure 2-24.

***Figure 2-24.*** File Browser Window



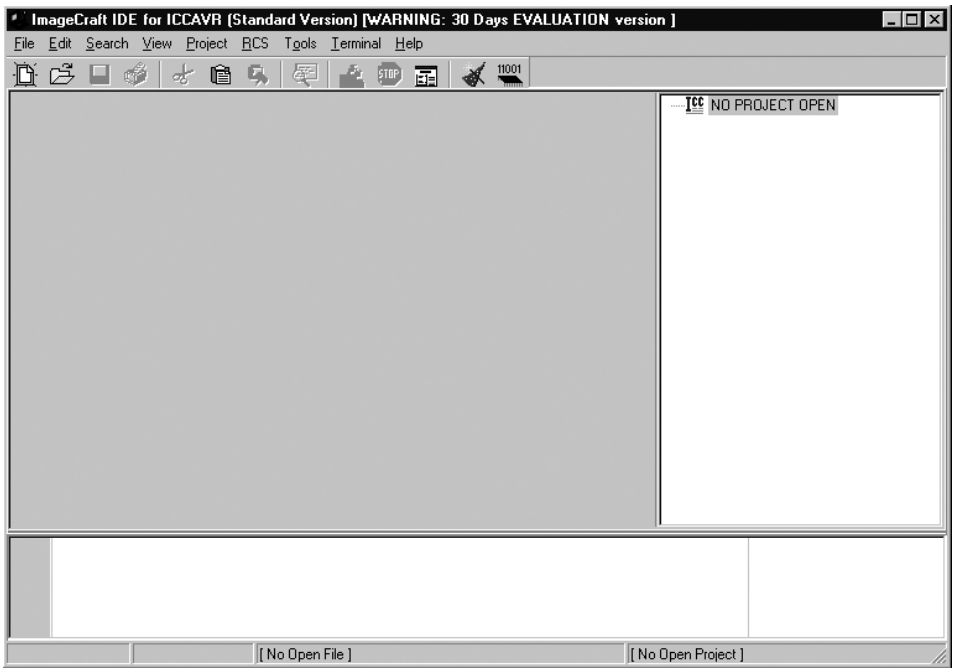
3. Press *OK*. An update *Plug-In* dialog box appears, see Figure 2-25.

***Figure 2-25.*** Update Plug-In Dialog Box



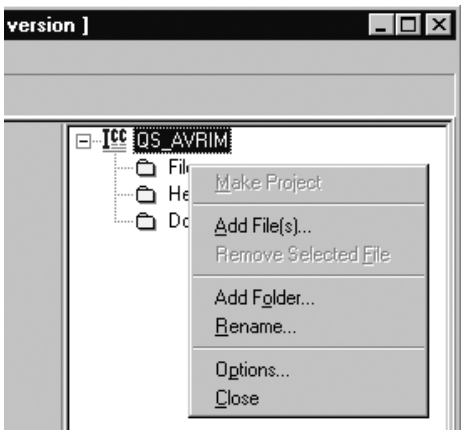

4. Choose *Yes* if you would like to have ImageCraft C compiler as default compiler for all your projects. Choose *No* if you would like to use ImageCraft C compiler for the current project only.
5. Press the *SW Compiler* button to open ImageCraft, see Figure 2-26.

***Figure 2-26.*** ImageCraft Window



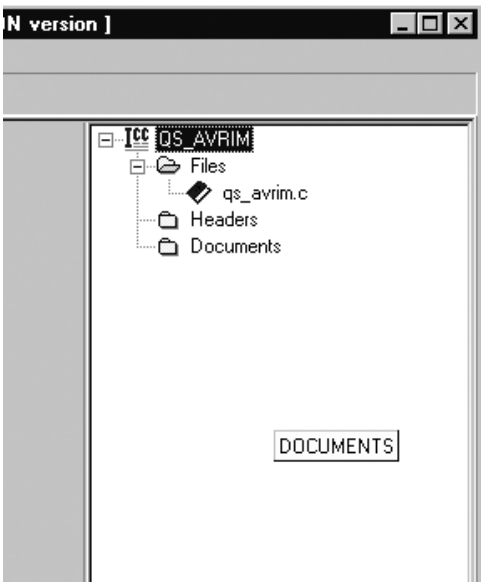

6. Go to the *Project* menu and select *New*. Use the *Browser* button to navigate to the correct folder `c:\training\fpslic\lab1` and type `qs_avrim` as your project name.
7. Press *Save*.
8. On the project window, right-click on *Files* and select Add File(s)… The *Add Files...* pop-up menu appears, see Figure 2-27.

***Figure 2-27.*** Add File(s)... Pop-up Menu



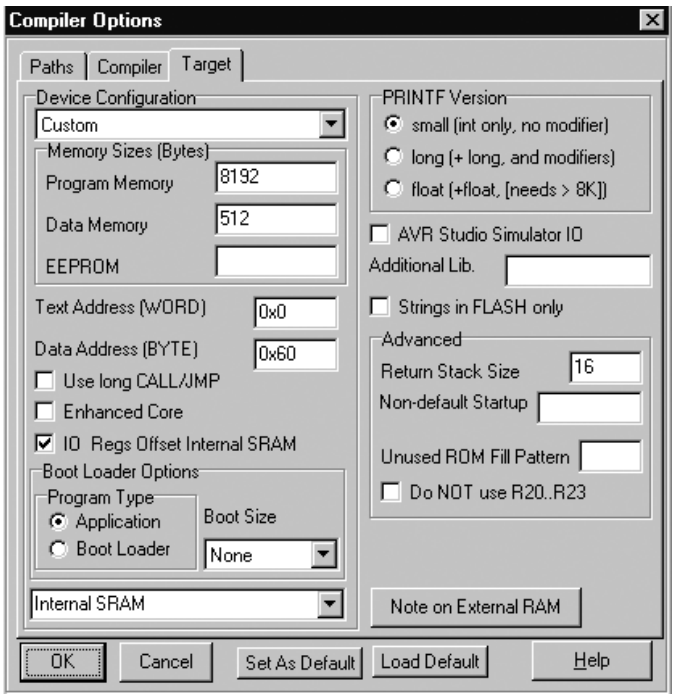

9. Select `qs_avrim.c` and press *Open*.
10. Press ⊞ to see the added file; `qs_avrim.c` now appears on the *File* window, see Figure 2-28.

***Figure 2-28.*** qs_avrim.c



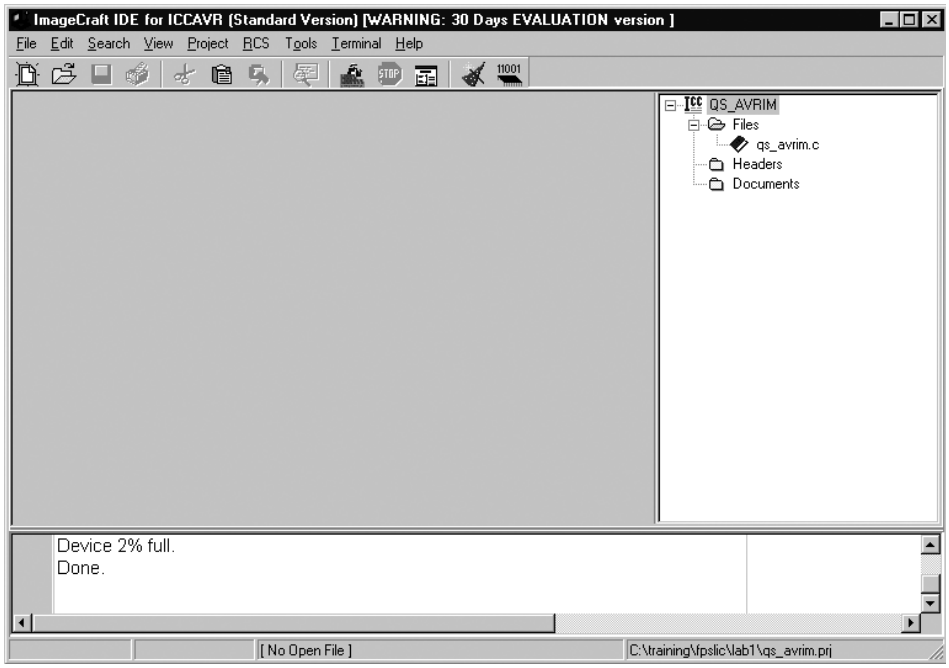

11. Go to the *Project* menu and select *Options…*. A *Compiler Options* dialog box appears, see Figure 2-29.

***Figure 2-29.*** Compiler Options Dialog Box




12. Click on the *Target* tab and select *AT94K40* as *Device Configuration*.
13. Select *32K Code/4K Data* as *FPSLIC Memory*.
14. Press *OK*.
15. Go to the *Project* menu and select *Rebuild All*; if successful, the status window will show *Done*, see Figure 2-30. If it is not successful, an error will appear in the status window.

***Figure 2-30.*** ImageCraft Status Window

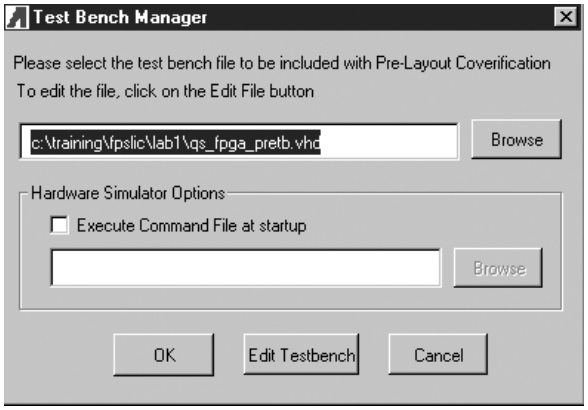



16. Go to the *File* menu, select *Exit* and return to System Designer.

### 2.9.2 Pre-layout Co-verification

1. Press the *Pre-layout Coverify…* button. The *Test Bench Manager* dialog box appears, see Figure 2-31. You now need to edit the test bench file to include the required stimulus and the signals.

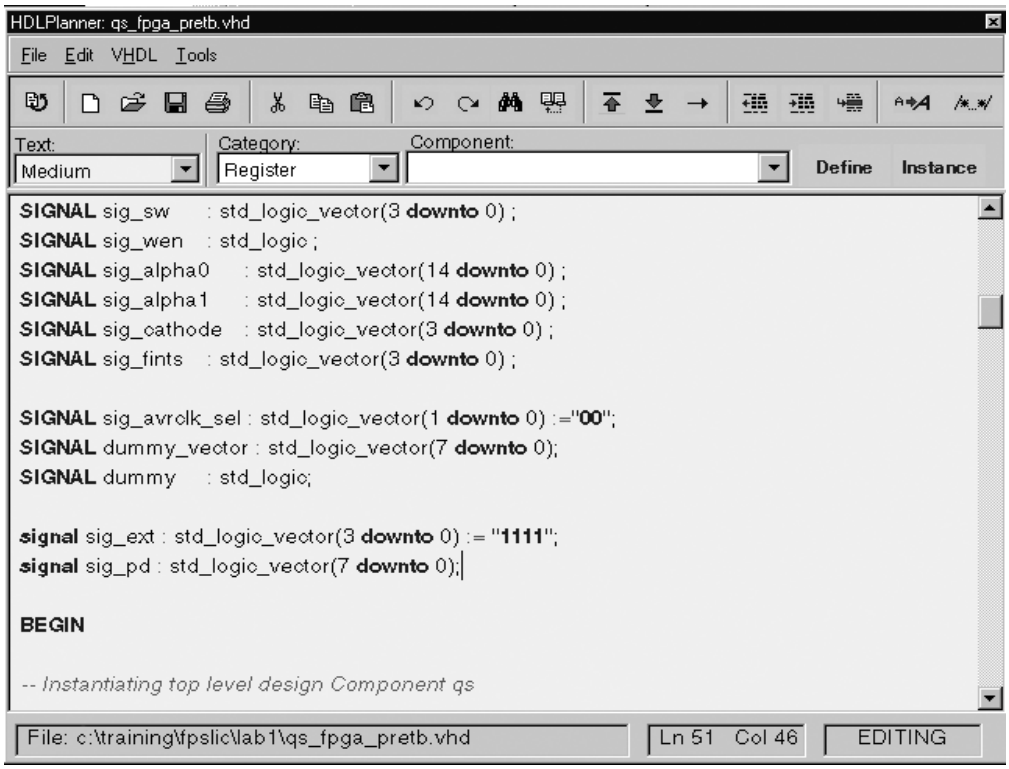
***Figure 2-31.*** Test Bench Manager Dialog Box




2. Select `qs_fpga_pretb.vhd`. If you are new to VHDL, refer to the Doulos VHDL tutorial provided with the System Designer CD.
3. Click *Edit Testbench*. HDLPlanner opens `qs_fpga_pretb.vhd`.
4. Add the following signals as shown in Figure 2-32. Scroll to the bottom of the window to find these signals and change them to the code below:

```
signal sig_ext : std_logic_vector(3 downto 0) := "1111";
signal sig_pd : std_logic_vector(7 downto 0);
```

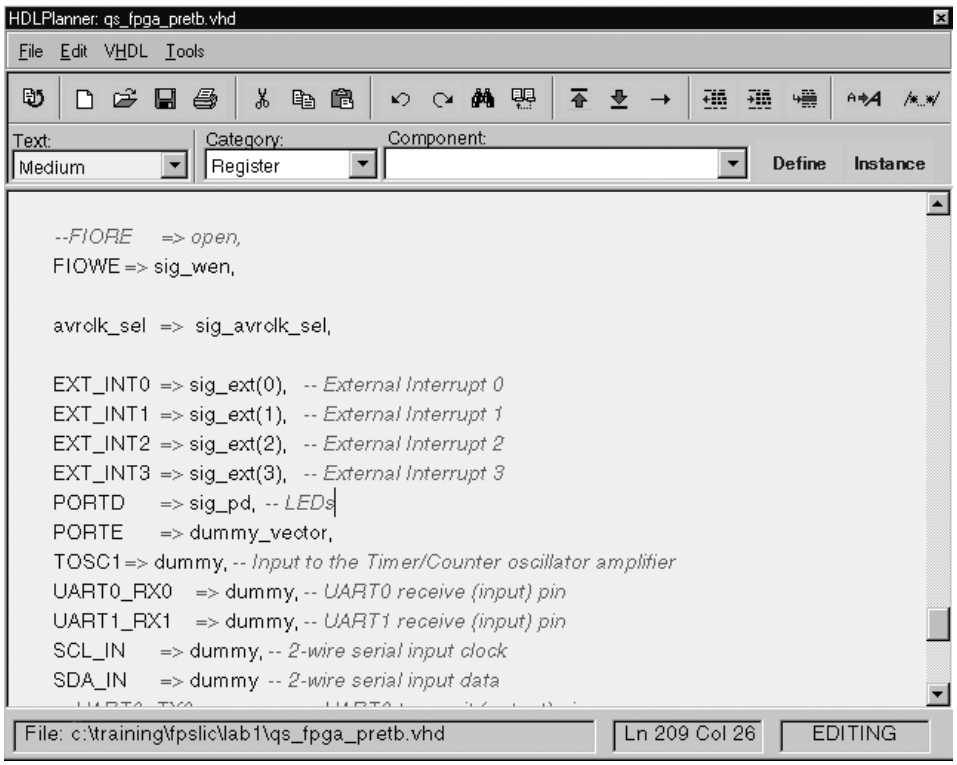
***Figure 2-32.*** Adding Signals



5. Connect the signals appropriately by editing existing lines of codes shown in Figure 2-33. Scroll to the bottom of the window to find these signals and change them to the code below:

```
EXT_INT0  => sig_ext(0),-- External Interrupt 0
EXT_INT1  => sig_ext(1),-- External Interrupt 1
EXT_INT2  => sig_ext(2),-- External Interrupt 2
EXT_INT3  => sig_ext(3),-- External Interrupt 3
PORTD     => sig_pd, -- LEDs
```

***Figure 2-33.*** Connecting External Interrupt and PortD Signals

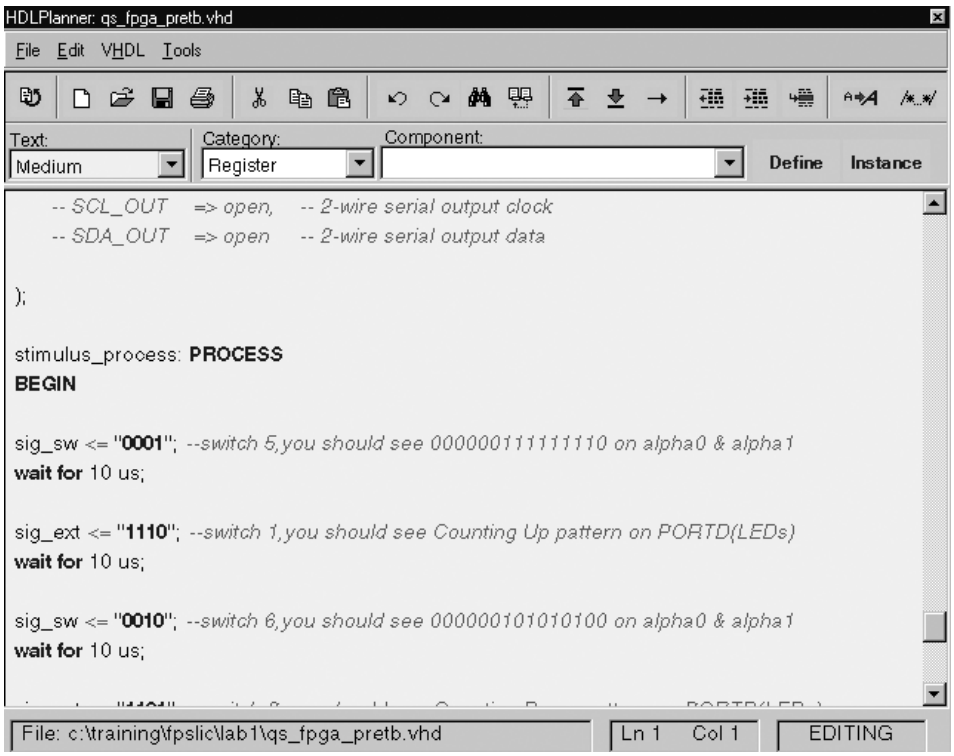



6. Scroll down to “stimulus process” and add the following stimulus as shown inFigure 2-34:

```
stimulus_process: PROCESS
BEGIN

sig_sw <= "0001"; --switch 5,you should see 000000111111110 on alpha0 & alpha1
wait for 10 us;

sig_ext <= "1110"; --switch 1,you should see Counting Up pattern on
PORTD(LEDs)
wait for 10 us;

sig_sw <= "0010"; --switch 6,you should see 000000101010100 on alpha0 & alpha1
wait for 10 us;

sig_ext <= "1101"; --switch 2,you should see Counting Down pattern on
PORTD(LEDs)
wait for 10 us;
```

```
sig_sw <= "0100"; --switch 7,you should see 000000010101010 on alpha0 & alpha1
wait for 10 us;

sig_ext <= "1011"; --switch 3,you should see Knightrider pattern on
PORTD(LEDs)
wait for 10 us;

sig_sw <= "1000"; --switch 8,you should see 111111000000000 on alpha0 & alpha1
wait for 10 us;

sig_ext <= "0111"; --switch 4,you should see Bounce pattern on PORTD(LEDs)
wait for 10 us;

END PROCESS stimulus_process;

END arch_test_bench;
```

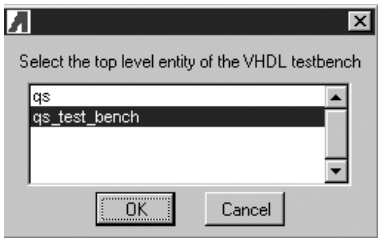
***Figure 2-34.*** Adding Stimulus




***Note:*** The order of stimulus is not in sequence (from SW1–SW8) just so both LEDs and the Alphanumeric values are visible in the same screen. Otherwise, by the time the Alphanumeric values are displayed, it will be about 50 µs that can not be captured in the same screen.

7. Save the file and close HDLPlanner.
8. Press *OK* to close the *Test Bench Manager* dialog box. A *Top Level Entity* dialog box appears, see Figure 2-35

***Figure 2-35.*** Top Level Entity Dialog Box




9. Select *qs_test_bench* from the given list and press *OK*.

   If you receive the error below, there may be a syntax error in your `*.vhd` file(s).

   ***ERROR** There were errors in compiling the VHDL files with ModelSim. Please refer to the log file for more details on the errors*.

   Follow the procedure below to find and fix the problem:

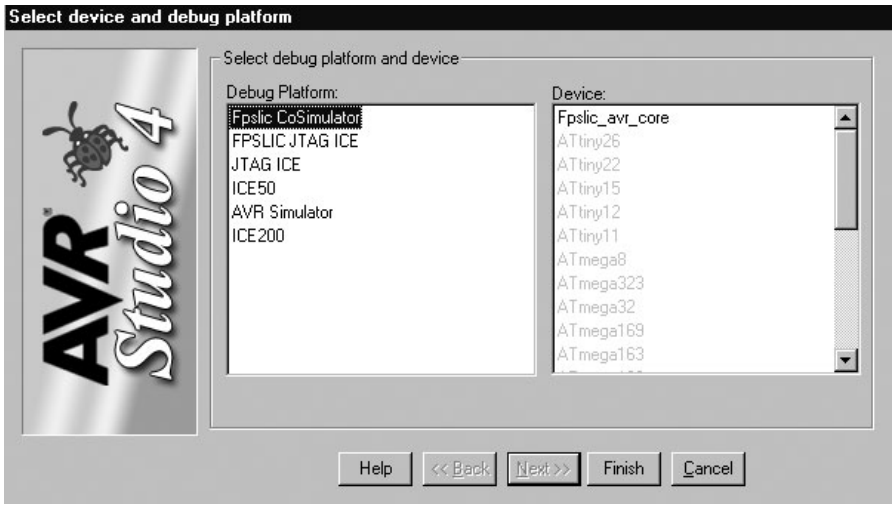
   – Open the log file by pressing the button located on the left side of the System Designer window.
   – The log file will display the errors with the file name and the line number. Note the error and the line number and close the log window.
   – Press the *Pre-layout Co-verification* button, the *Test Bench Manager* dialog box opens with the `qs_fpga_pretb.vhd` in the path.

   If the error appears again, press *Edit Testbench* to fix the error, otherwise press the *Browse* button to add the right file and follow the same procedure above to edit the test bench file.

   If successful, a DOS window, ModelSim and AVR Studio will open. To run Co-verification you have to run two simulators at the same time. Because both programs expect to be in control when they are operating, you have to switch between the two systems, so only one of them is in control at a time.

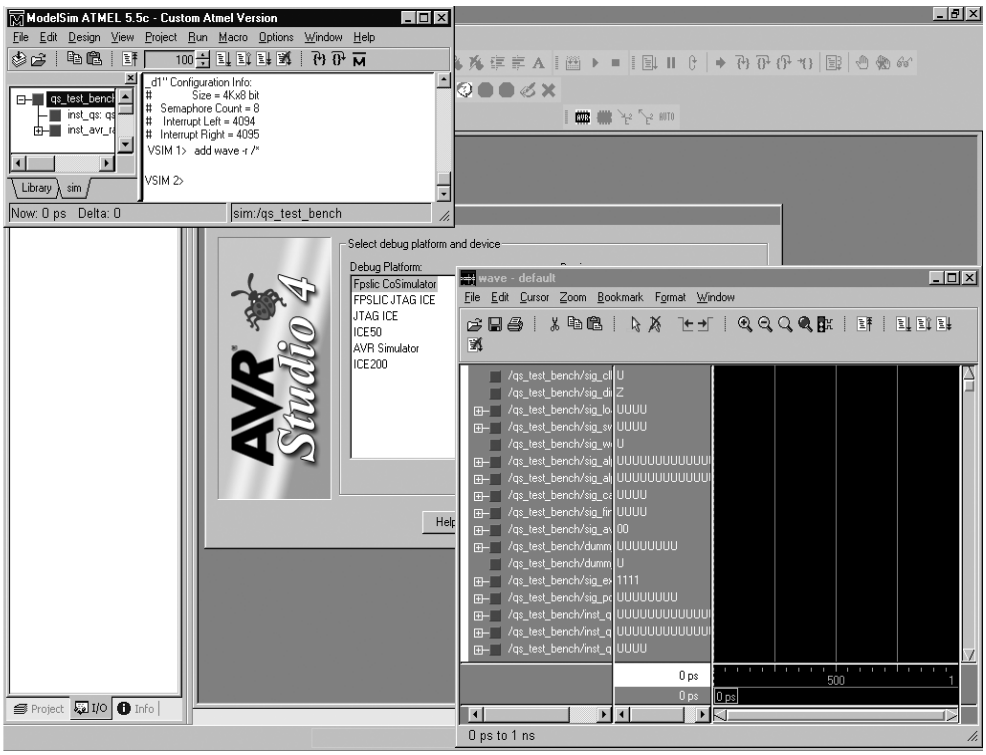
10. In AVR Studio, go to the *File* menu, chose *Open File...* and select *qs_avrim.cof*. The *Select Device and Debugging Platform* dialog box will appear, see Figure 2-36. The Select Device and Debugging Platform dialog box displays the list of the devices supported by the debugging platforms. Fpslic_avr_core is the only device available for the FPSLIC Co-simulator debugging platform.

***Figure 2-36.*** Select Device and Debugging Platform Dialog Box



11. Press *Finish*. AVR Studio will show an hourglass but will not complete loading the file until ModelSim has been set up. The AVR Studio window will no longer respond, ModelSim now holds control.
12. In ModelSim, type in `add wave -r /*`. Press *Enter*. A wave window showing all the signals in the design appears, see Figure 2-37.

***Figure 2-37.*** Wave Window



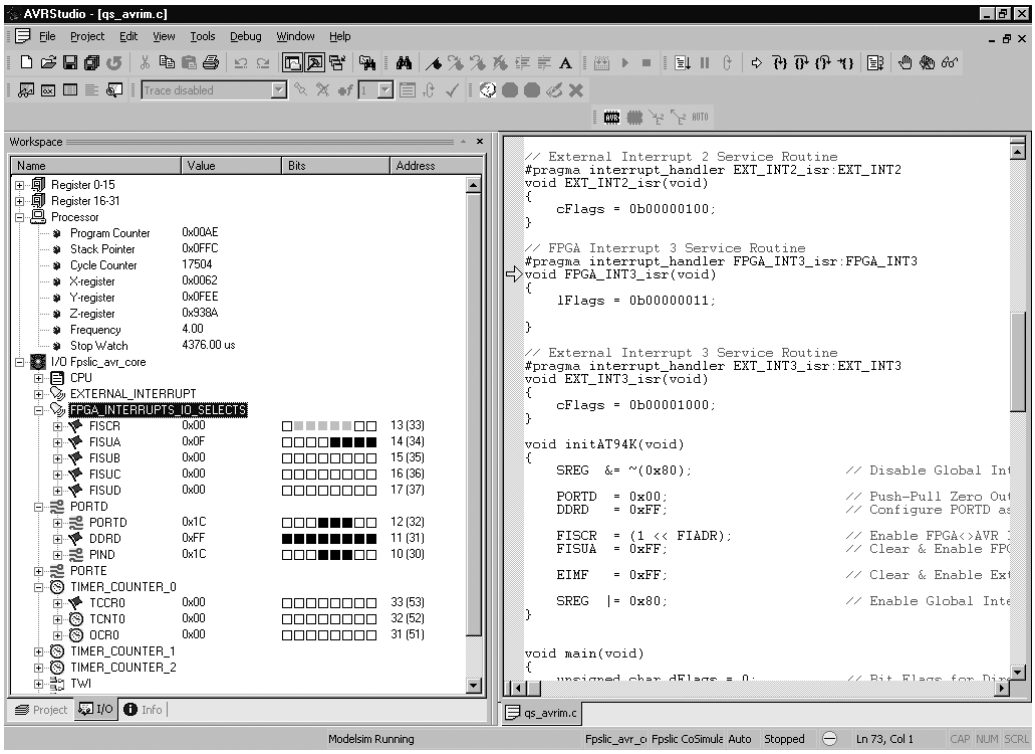

13. In ModelSim, type in `run -all` or press the button.

    **Co-verification Control**: If AVR Studio is not responding, you have to go back and check ModelSim and type in `run -all` to get back into AVR Studio.

    If ModelSim is not responding, AVR Studio is in control and you have to press the *Auto Step* button or *Run* for program execution. To view the waveforms, press the *Break* command from the *Debug* menu to stop program execution, followed by a hardware break button to return control to ModelSim.

14. In AVR Studio, maximize the AVR Studio window. The *Workspace* window allows you to view the registers, I/Os and processor details. Press on ⊞ to view the details of External Interrupts and PORTD, see Figure 2-38.

***Figure 2-38.*** External Interrupts and PORTD Details




15. Press the *Auto Step* button.

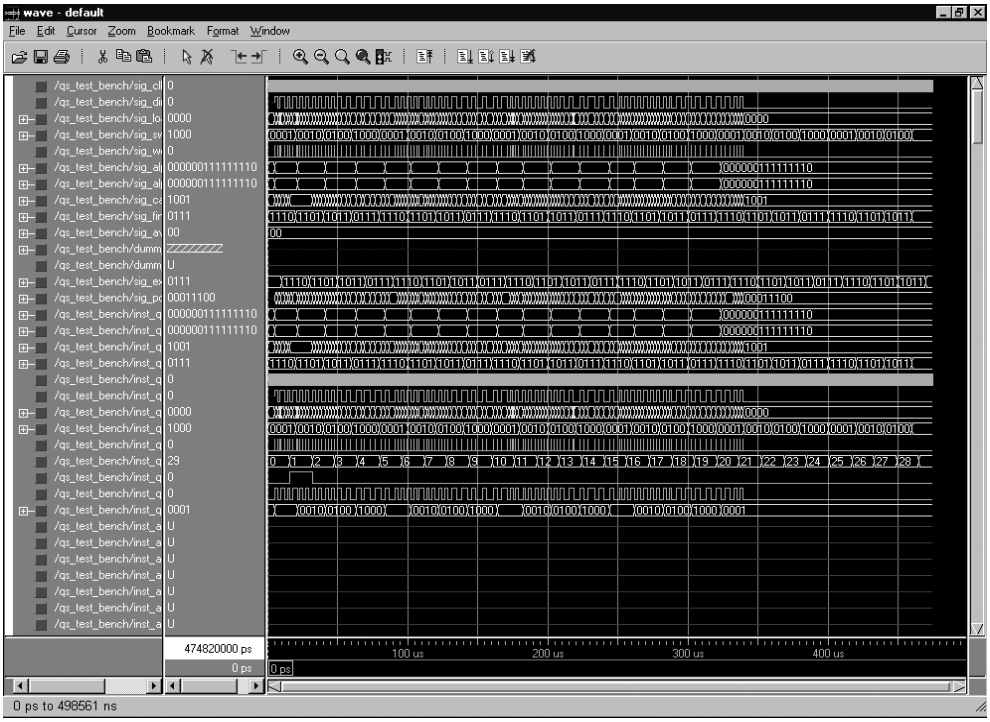
    Notice the patterns on PORTD and External Interrupt fields in the I/O View. SW1–SW4 are connected to External Interrupts, the patterns (Counting Up, Counting Down, Knightrider and Bounce) can be viewed on the Port D data fields. The Interrupt Mask shows the active External Interrupt.

    Let it run for about 30 µs to see different patterns. The Time Elapsed field on the *Processor* window reports the time.

16. Press the *Debug Break* button followed by the *Hardware Break* button to take control over ModelSim.

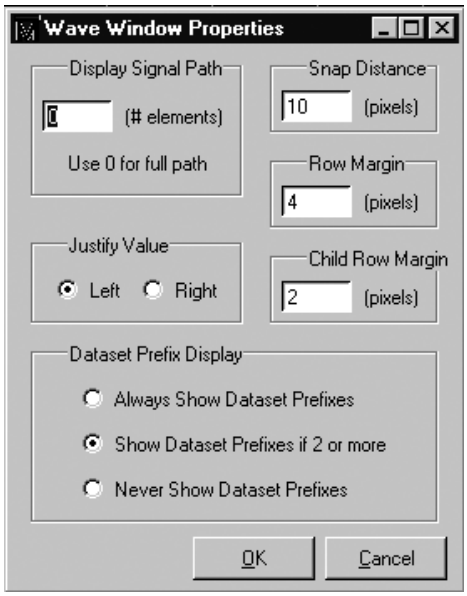
17. Click on the wave window and maximize it. Go to the *Zoom* menu and select *Zoom Full*, see Figure 2-39.

***Figure 2-39.*** Zoomed Wave Window




18. Go to the *Edit* menu and select *Display Properties*. The *Wave Window Properties* dialog box appears, see Figure 2-40.

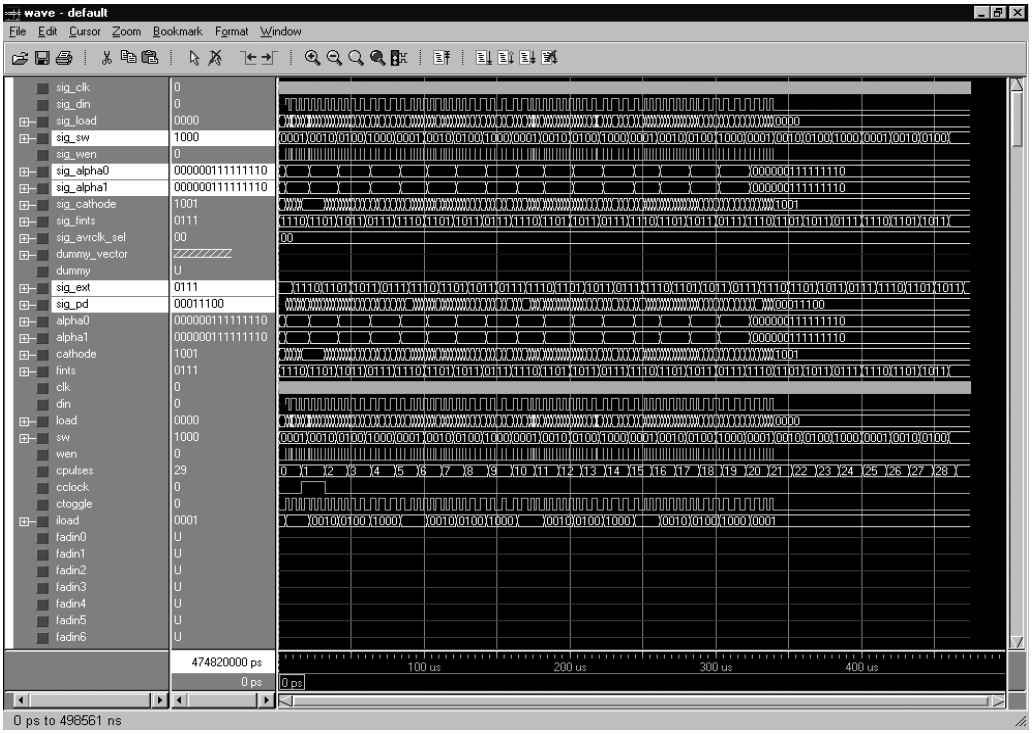
***Figure 2-40.*** Wave Window Properties




19. Under *Display Signal Path,* change 0 to 1 to view the signals without any path associated with them and press *OK*.

If AVR Studio is not responding switch to ModelSim. If ModelSim is not responding switch to AVR Studio. The resulting signals appear in the Pre-layout Co-verification Waveform, see Figure 2-41.

***Figure 2-41.*** Pre-layout Co-verification Waveform Window




The highlighted signals are the ones that we have to look at. In the test bench file we give the stimulus for signals sig_sw (SW5–SW8) and sig_ext (SW1–SW4), and we expect the results on signals alpha0, alpha1 and sig_pd. The radix of the signal can be changed.

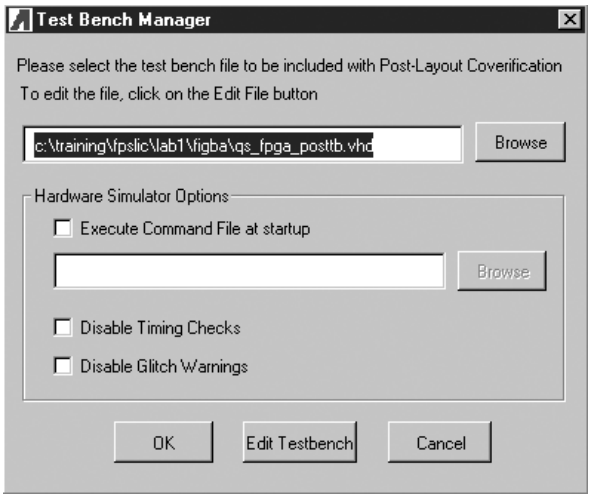
Select the signal and right-click to choose the desired radix.

20. To continue cosimulation, press the run -all button in ModelSim or the co-verification Run button in AVRStudio followed by the debug Run or autostepping button.
21. To exit co-verification, press the *Exit* button in AVRStudio or go to the *File* menu and select *Exit*.

### 2.9.3 Post-layout Co-verification

1. Press the *Post-layout Coverify…* button. The *Test Bench Manager* window appears, see Figure 2-42.

***Figure 2-42.*** Test Bench Manager Window

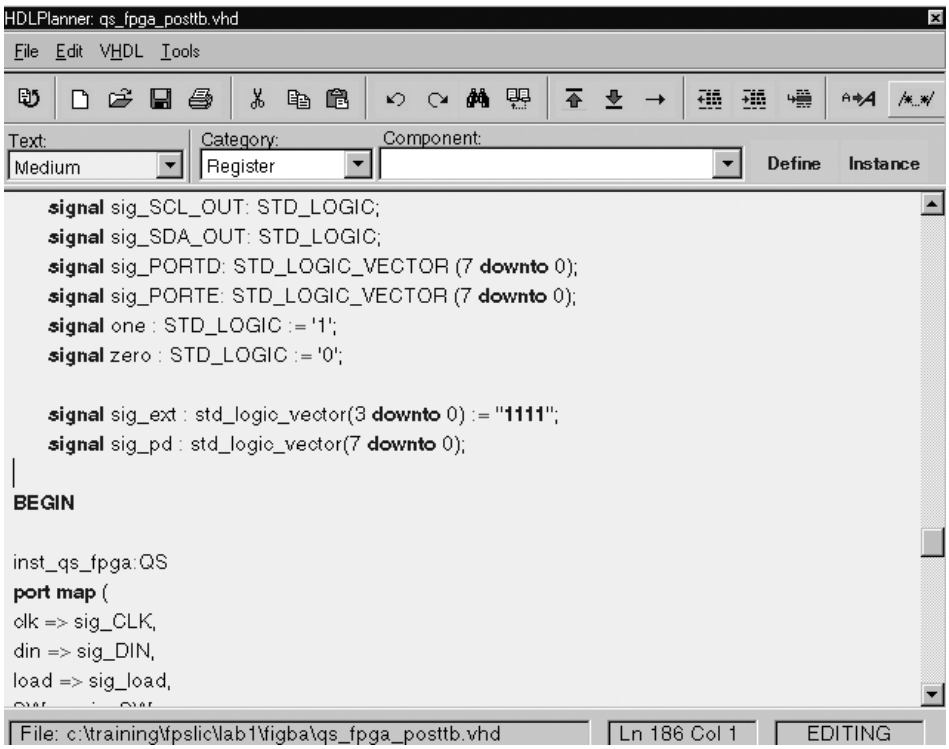



The changes made to the test bench file in pre-layout co-verification should be applied to post-layout test bench file `qs_fpga_posttb.vhd` as well. Again, there are three places that need to be edited.

2. Press `Edit Testbench`.
3. Add the following signals as shown in Figure 2-43:

```
signal sig_ext : std_logic_vector(3 downto 0) := "1111";
signal sig_pd : std_logic_vector(7 downto 0);
```

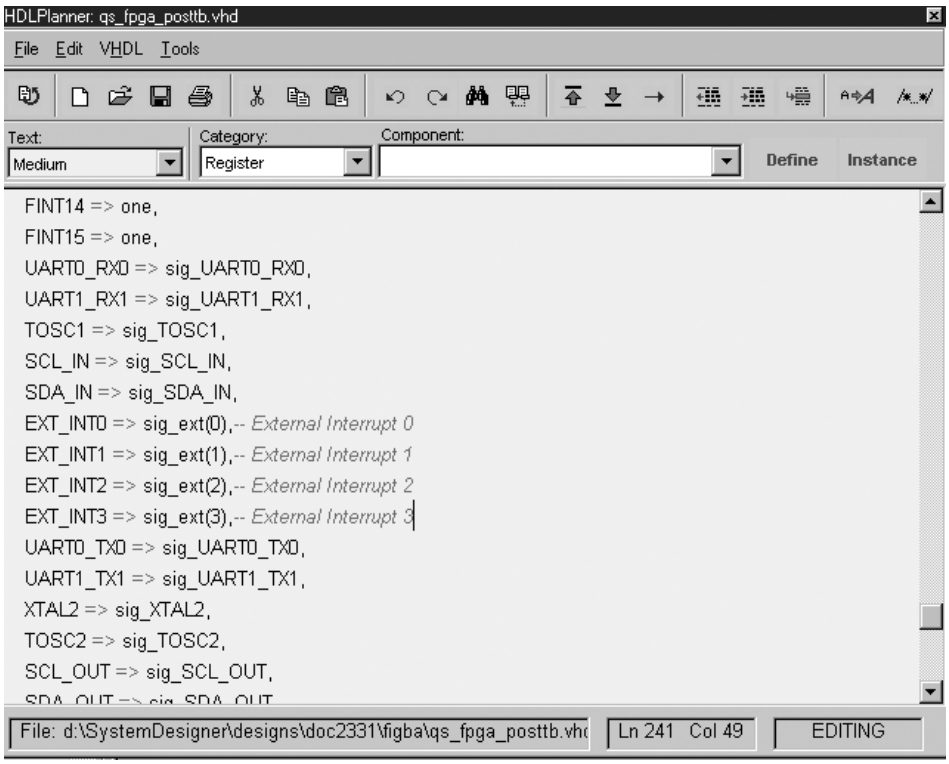
***Figure 2-43.*** Adding Signals



4. Connect the signals below as shown in Figure 2-44:

```
EXT_INT0 => sig_ext(0),
EXT_INT1 => sig_ext(1),
EXT_INT2 => sig_ext(2),
EXT_INT3 => sig_ext(3),
```

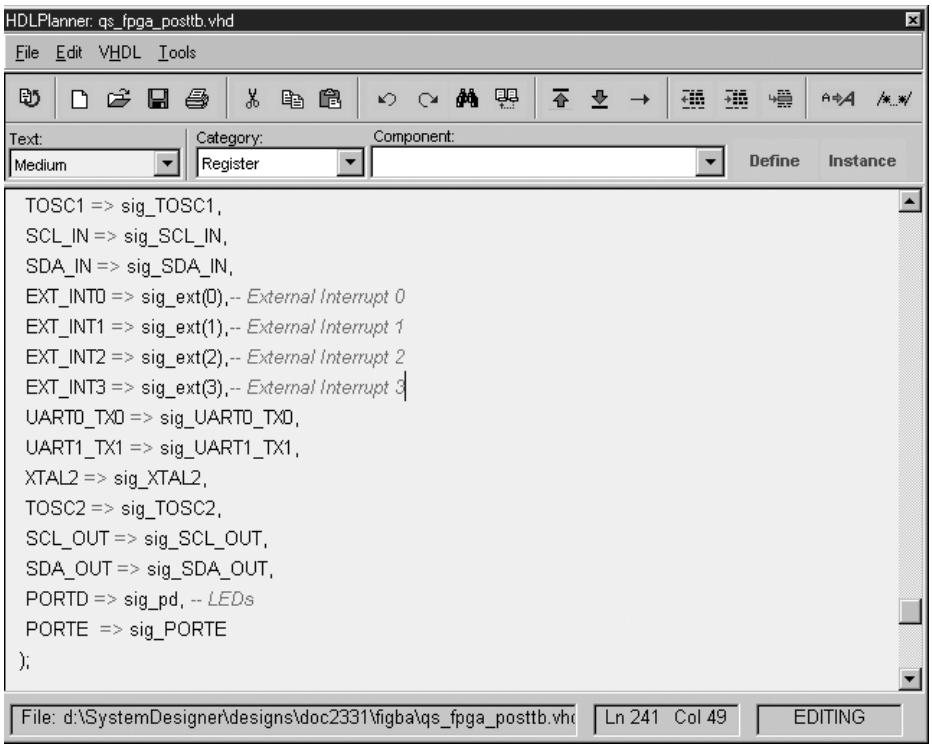
***Figure 2-44.*** Connecting Signals




5. Connect the signal below as shown in Figure 2-45.

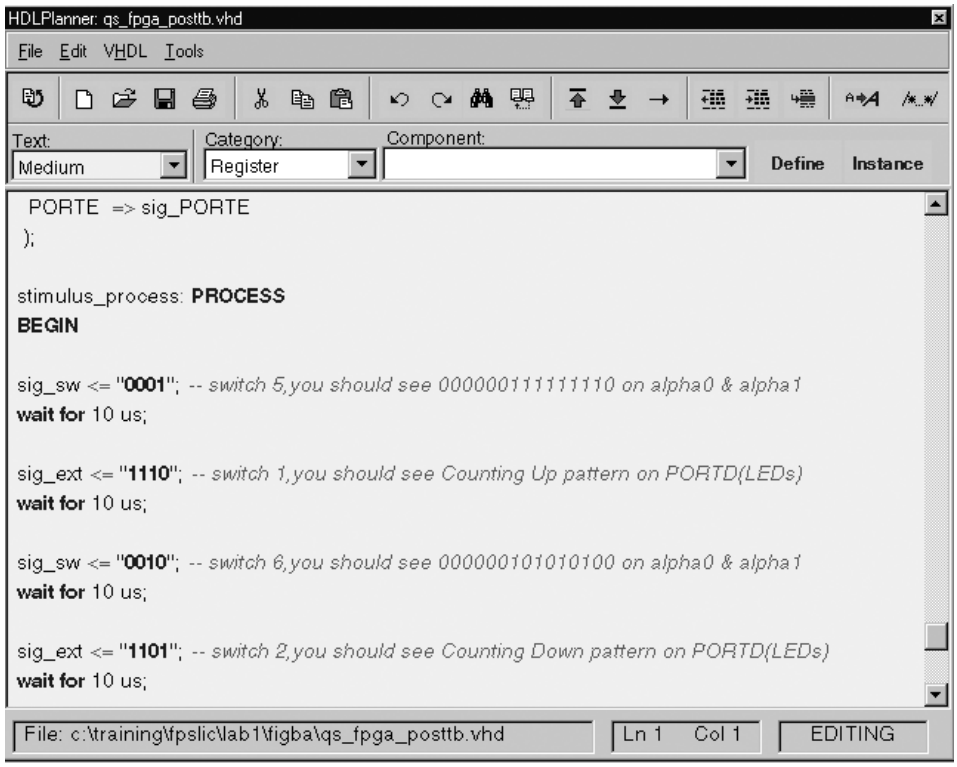
```
PORTD => sig_PD
```

***Figure 2-45.*** Connecting Signals for PORTD



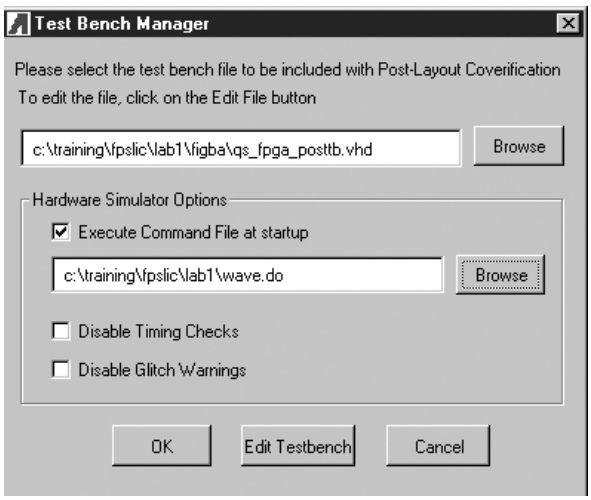

6. Copy and paste the stimulus from pre-layout test bench file to the post-layout test bech file, see Figure 2-46.

***Figure 2-46.*** Copied Stimulus



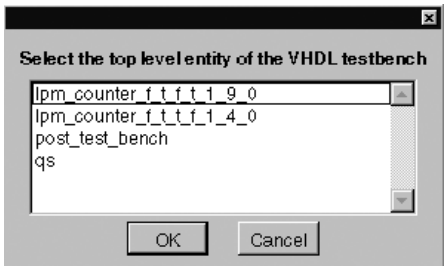
7. Save the edited file and close HDLPlanner.
8. Check the *Execute Command File at Startup* box and press the *Browse* button to add the `wave.do` file, see Figure 2-47.

***Figure 2-47.*** Test Bench Manager Window



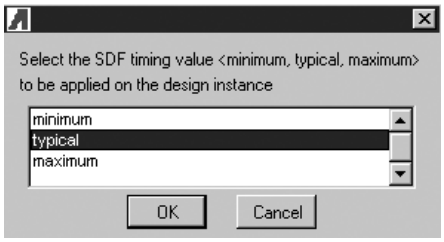

9. Press *OK* on the *Test Bench Manager* dialog box. A dialog box to select the top-level entity of the VHDL test bench appears, see Figure 2-48.

***Figure 2-48.*** Top-level Entity of the VHDL Testbench Dialog Box




10. Select *post_test_bench* from the given list and press *OK*. A dialog box to select the SDF timing value appears, see Figure 2-49.

***Figure 2-49.*** SDF Timing Value Dialog Box




11. Select *typical* and press *OK*.

    A DOS window, ModelSim and AVR Studio will open, this time the wave window opens with all the signals since we added the command file.

    To run Co-verification you have to run two simulators at the same time. Because both programs expect to be in control when they are operating, you have to switch between the two systems, so only one of them is in control at a time.

12. In AVR Studio, go to the *File* menu, select *Open File...* and select `qs_avrim.cof`.

    Since the *Processor* and *New I/O View* windows were opened during pre-layout co-verification, AVR Studio will open them automatically.

    **Co-verification Control**: If AVR Studio is not responding, you have to go back and check ModelSim and type in `run -all` to get back into AVR Studio.

    If ModelSim is not responding, AVR Studio is in control and you have to press the *Auto Step* button or *Run* for program execution. To view the waveforms, press the *Break* command from the *Debug* menu to stop program execution, followed by a hardware break button to return control to ModelSim.

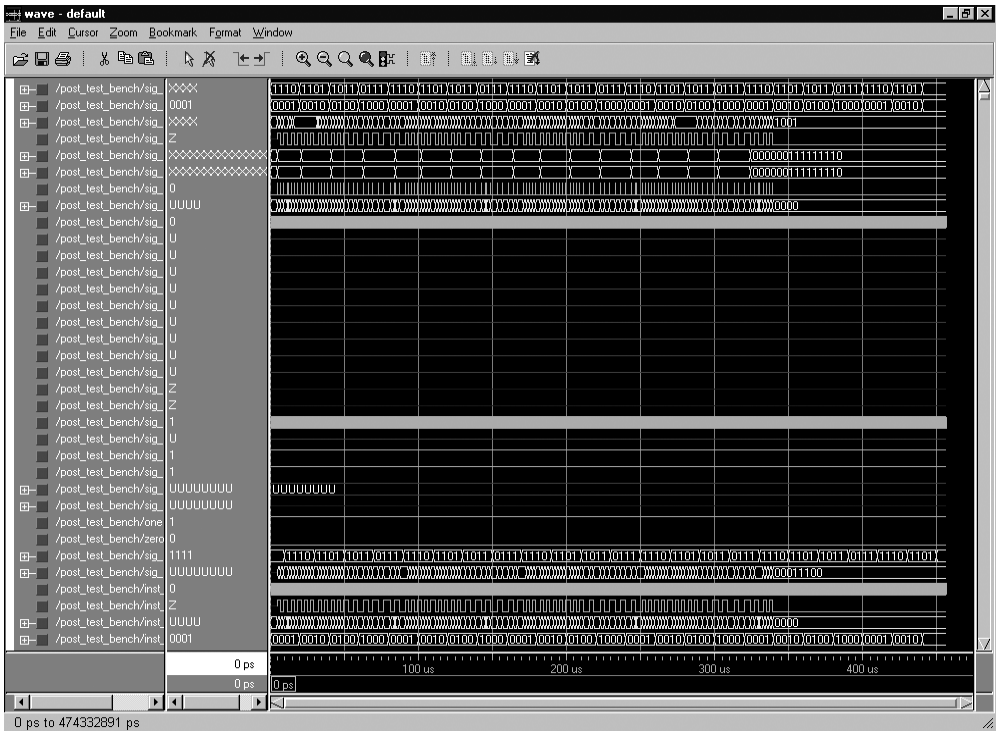
13. In AVR Studio, maximize the AVR Studio window.
14. Press the *Auto Step* button.

    Notice the patterns on PORTD and External Interrupt fields in the New I0 View. SW1–SW4 are connected to External Interrupts, the patterns (Counting Up, Counting Down, Knightrider and Bounce) can be viewed on the Port D data fields. The Interrupt Mask shows the active External Interrupt.

    Let it run for about 30 µs to see different patterns. The Time Elapsed field on the *Processor* window reports the time.

15. Press the *Debug Break* button tfollowed by the *Hardware Break* button to take control over ModelSim.
16. Click on the wave window. Maximize the window, go to the *Zoom* menu and select *Zoom Full*. The resulting signals appear in the post-layout co-verification waveform, see Figure 2-50.

***Figure 2-50.*** Post-layout Co-verification Waveform Window



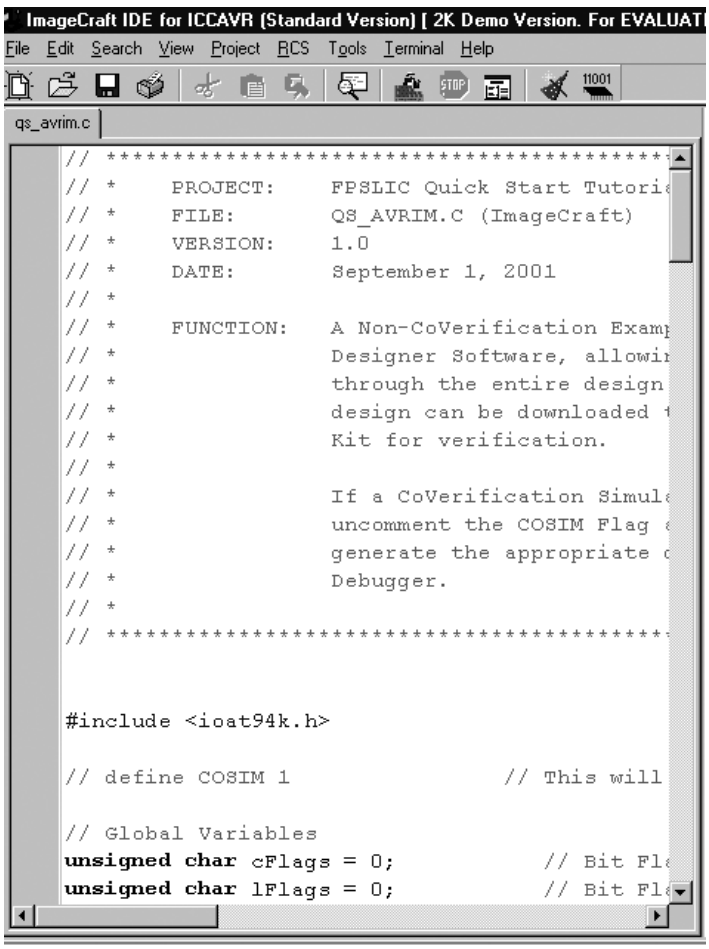
17. To continue cosimulation, press the run -all button in ModelSim or the co-verification Run button in AVRStudio followed by the debug Run or autostepping button.
18. To exit co-verification, press the *Exit* button in AVRStudio or go to the *File* menu and select *Exit*.

### 2.9.4 Introducing the Delay

The define statement will remove the delay during co-verification. While downloading to the starter kit we need this delay. Comment this line and generate the bitstream to download to the starter kit.

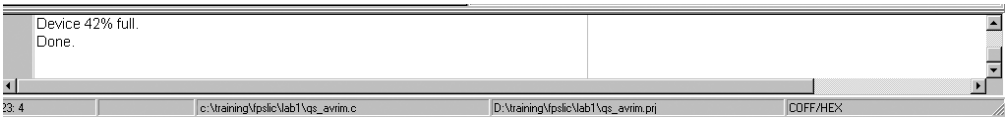
1. Press the *SW Compiler* button to open the ImageCraft compiler to edit and build the code.
2. On the right window, under *Files*, double-click on `qs_avrim.c` to open and edit the file.
3. Comment the line, `#define COSIM 1` by inserting two forward slashes, see Figure 2-51.

***Figure 2-51.*** : Commenting the #define COSIM 1 Line



4. Go to the *Project* menu and select *Rebuild All*. If successful, the status window will show *Done*, see Figure 2-52.

***Figure 2-52.***

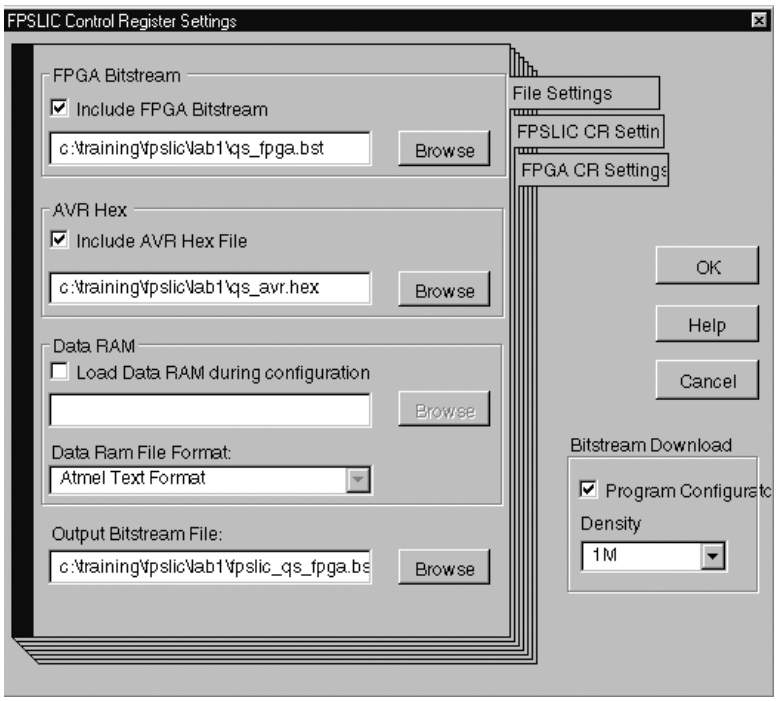



5. Go to the *File* menu and select *Exit*.

## 2.10 Bit Stream Generation

1. Press the *Device Programming* button. A dialog box with the bit stream settings opens, see Figure 2-53.

***Figure 2-53.*** File Settings




If you are testing code for the AVR you may only want to program the AVR portion without having the FPGA side of your design. Similarly you can program just the FPGA portion.

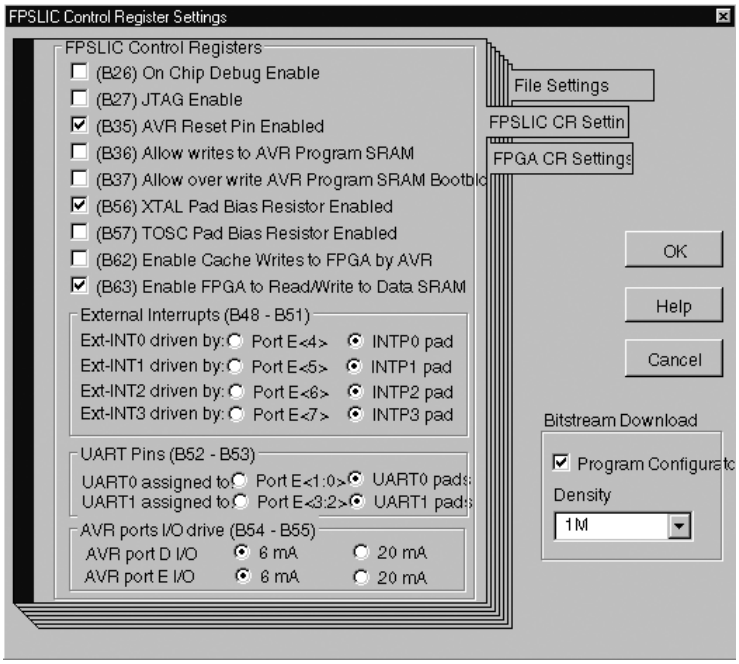
- To add the AVR file, enable *Include AVR Hex File* by checking the box and use the *Browse* button to add the file.
- To add the FPGA file, enable *Include FPGA Bit Stream* by checking the box and use the *Browse* button to add the file.

  If your design is dependent on signals from the AVR-FPGA Interface, you need to combine both the FPGA bit stream file and the AVR hex file.

2. Check the *Include FPGA Bit Stream* box to enable this option and press the Browse button to add the `qs_fpga.bst` file.
3. Check the *Include AVR Hex File* box to enable this option and press the *Browse* button to add the `qs_avr.hex` file.

4. Check the *Program Configurator Box* and select *1M* as the *Density.*
5. Click on the *FPSLIC Control Register Settings* tab. Use the default settings, see Figure 2-54.

***Figure 2-54.*** FPSLIC Control Register Settings



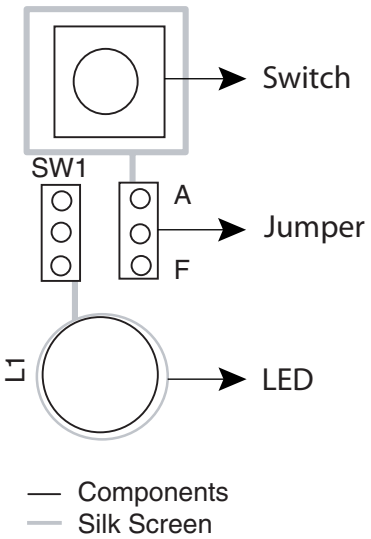

6. Click on the *FPGA Control Register Settings* tab. Use the default settings, see Figure 2-55.

***Figure 2-55.*** FPGA Control Settings


***Note:*** Before you press *OK*, some hardware connections need to be performed.

7. Connect the 25-pin parallel cable to the 25-pin male connector of the ATDH2225 download cable. The 10-pin female header plugs into the 10-pin male header (J1) on the ATDH94STKB board.
8. Connect the power supply from an AC outlet to the 9V DC connector (P3) on the ATSTK94 board.
9. Make sure to set the jumpers located between the LEDs and Switches appropriately. LED1 to LED8 should be connected to AVR side. Switches SW1 to SW4 should be connected to the AVR side, and switches SW5 to SW8 should be connected to the FPGA side.
10. Adjust *SW10* to the *PROG* position and *SW14* to the *ON* position.

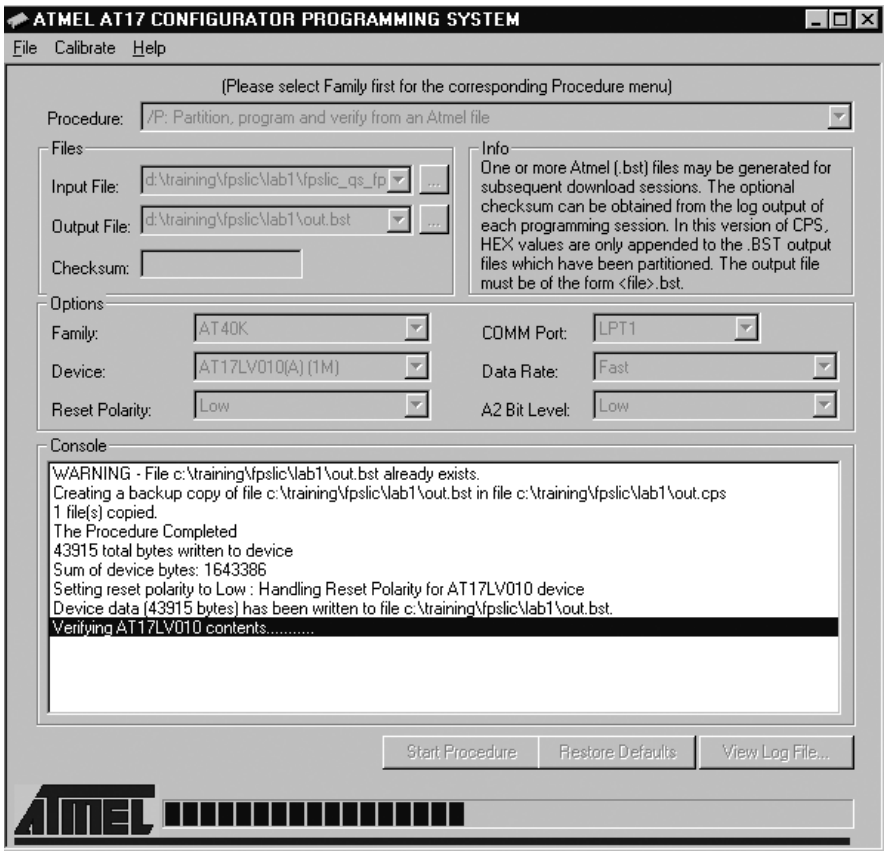
***Figure 2-56.*** Jumper Layout




11. Press *OK*. This will generate `fpslic_qs_fpga.bst`, which will be downloaded to the configuration memory on the starter kit board. Atmel's AT17 Configurator Programming System (CPS) window opens and automatically programs the device.

    If successful, the CPS console will show *Number of Fatal Errors: 0*, see Figure 2-57. If unsuccessful, go to the Help menu and select Contents > Trouble Shooting.

*Figure 2-57.* Atmel AT17 Configurator Programming System Window




## 2.11 Running the Design

1. Make sure to adjust SW10 to the RUN position.
2. Press the Reset switch (SW12) on the right edge of the board.
3. Press the switches one by one to see the changes in patterns on the Alphanumeric. The default pattern is the knightrider pattern.

Expected Results:

- When SW1 is pressed, the LEDs count up.
- When SW2 is pressed, the LEDs count down.
- When SW3 is pressed, the LEDs show Knightrider.

***Note:*** Some of the boards may not do this function since the AVR side of the switches SW2 and SW3 are shorted together.

- When SW4 is pressed, the LEDs bounce from the middle to the end and back.
- When SW5 is pressed, ∗ is displayed on the Alphanumeric LEDs.
- When SW6 is pressed, **+** is displayed on the Alphanumeric LEDs.
- When SW7 is pressed, **x** is displayed on the Alphanumeric LEDs.
- When SW8 is pressed, **0** is displayed on the Alphanumeric LEDs.

4. Close CPS.

## Atmel Corporation

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

## Regional Headquarters

***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, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

***Microcontrollers***
2325 Orchard Parkway
San Jose, CA 95131, USA
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, USA
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, USA
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


***Literature Requests***
www.atmel.com/literature




3478A–FPGA–04/2004 /xM