

# **Quickstart Tutorial**



# Table of Contents



## Section 1

Introducti	on	1-1
1.1	Design Description	1-1
1.2	Design Flow	1-2
1.3	Expected Results	1-2

## Section 2

Step-by-S	Step	Procedure	2-1
2.1	Set	tting Up the Example Files	2-1
2.2	Ins	talling ImageCraft <sup>™</sup> Compiler	2-1
2.3	Set	tting Up the Project	2-2
2.4	Adv	vanced Flow	2-6
2.5	Co	mpiling the AVR Assembly File	2-7
2.6	Syr	nthesizing the FPGA file	2-7
2.7	AV	R-FPGA Interface	2-8
2.8	FP	GA Place & Route	2-11
2.9	Co	-verification	2-14
2.9	9.1	Changing the Compiler Tool Settings	2-14
2.9	9.2	Pre-layout Co-verification2-18	
2.9	9.3	Post-layout Co-verification2-26	
2.9	9.4	Introducing the Delay	2-31
2.10	Bit	Stream Generation	2-32
2.11	Ru	nning the Design	2-35

i

Table of Contents



ii



# 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.

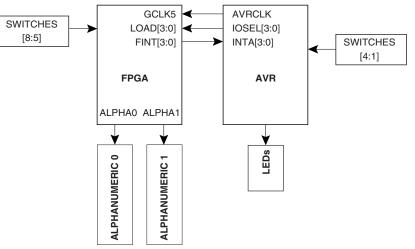
When running the FPGA and AVR stand-alone designs, the design must be combined at the end using System Designer. If this method is used, the interface must be clearly defined prior to the design stage, since the FPGA-AVR interface is fixed in silicon.

# **1.1Design**<br/>DescriptionThe design in this tutorial shows the user how to use the switches to invoke different<br/>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



Results

Design Flow	The flow below provides a step-by-step explanation of quickstart_lab:
Design now	<ol> <li>Compile Assembly/C Code: Wavrasm Assembler creates the qs_avr.hex file. Using ImageCraft Compiler creates the qs_avrim.hex (used for down- load) and qs_avrim.cof (used for debugging) files.</li> </ol>
	<ol> <li>AVR-FPGA Interface: Defines the connections between the embedded FPGA and AVR Core; qs_fpga.ict is created.</li> </ol>
	<ol> <li>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.</li> </ol>
	4. Synthesize VHDL Code: LeonardoSpectrum creates the <code>qs_fpga.edf</code> netlist file. This file is imported into the Figaro Place & Route tool.
	5. <b>FPGA Place &amp; Route Using Figaro:</b> qs_fpga.edf is imported into Figaro to place and route the design, qs_fpga.bst is created.
	<ol> <li>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.</li> </ol>
Expected	There are four patterns that are shown on each set of LEDs depending on the swi

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 <sup>®</sup> 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<sup>™</sup> 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

	tmei System Designer (no project)	
Proje	ect <u>E</u> dit <u>O</u> ptions <u>H</u> elp	
0 \$	Project - no name	Part - no part
1 1		
		Junna -
*		
<i>⊗</i> ⊠.		
D		ATMEL Corporation
		The second and the se
	▼ ▼	× ×
	Log	
	Log: Welcome to System Designer product of Log: Select Project->New from the menu to	
	no project	

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	Welcome to the New Project wizard
Sist	The New Project wizard will guide you in the creation of your new System Designer project.
₹~~	The steps that follow will include:
- <b>三</b>	1. Creating a project file
i Co	2. Selecting the parts for use with your project
DESI	3. Selecting the software tool sets for use with each part
ם ייסטיי	
	Cancel < Back Next > Fini



- 3. Press Next >. The Create Project File window opens, see Figure 2-3.
- Figure 2-3. New Project Wizard Step 2 of 6

New Project Wizard - Step	<sup>2 of 6</sup> × Create Project File
515	Select a name and directory path for your new project using the co below. When you are ready, click 'Next' to continue.
SYSTEM	c ∷training¥pslicVab1Vab1.apj
	Cancel < Back Next > Finish

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

New Project Wizard - Step 3	of 6 🛛 🛛 🛛
	What part do you want to use ?
515	Choose a part for your project from the Part Number list below. Architecture: Part Number:
SYSTEM DESIGNER	Any   Any   Product Family:   Any   Any   Any   Any   Package:   Any   Package:   Any   Speed Grade:   Any   Application:   Any
	Cancel < Back Next > Finish

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

New Project Wizard		oftware toolflow for this
SYSTEM DESIGNER	Please select a softwa from the list below. Toolflow: <u>Mentor-VHDL</u> Mentor-Verilog	are toolset for part U1 : AT94K40-25DQC  Description:  Mentor-VHDL  Tools:  AT94K Device Options  HDL Synthesis Software Compiler  AVR-FPGA Interface Pre-lavout Coverification
	Cancel	< Back Next > Finish

- 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

New Project Wizard	Step 5 of 6     Vould you like to add more parts ?     The list below shows the parts you have selected for your project.     If you would like to add more parts, click the add more parts button below.     When you are ready, click 'Next' to continue.
TEM IGNER	Project Parts: U1:AT94K40-25DQC - Mentor-VHDL
SYS DES	<ul> <li>Add more parts</li> <li>Done with parts</li> </ul>
	Cancel < Back Next > Finish

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



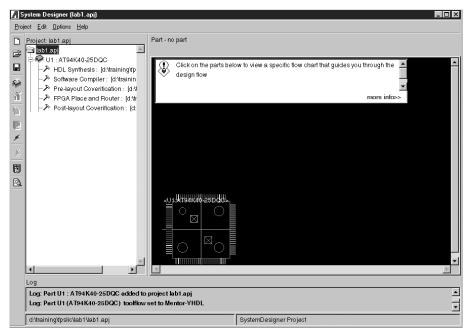
- 9. Press Next >. The last window of the wizard appears, see Figure 2-7.
- Figure 2-7. New Project Wizard Step 6 of 6

🔏 New Project Wizard -	Step 6 of 6
	Congratulations!
515	You have successfully completed the creation of your new System Designer project.
Ž~	The project tree shows the tools and design files included in your project. Using the vertical toolbar on the side of the tree you can add or delete parts, modify toolflows, or include new design files.
GNEF	To begin, click on a part in the project tree. This will display the design flow in the part window. Then, click on any button in the design flow to run the associated design tool.
SYS DESI	To close the New Project wizard and begin working with your project, click the 'Finish' button.
	Cancel < Back Next> Finish

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

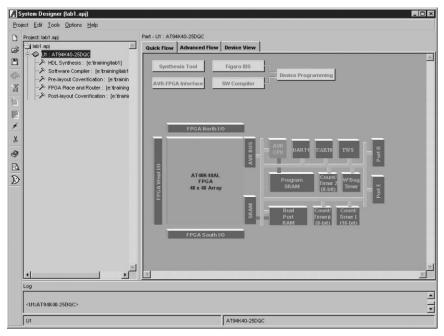


Clicking on *more info >>* opens the online help.



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

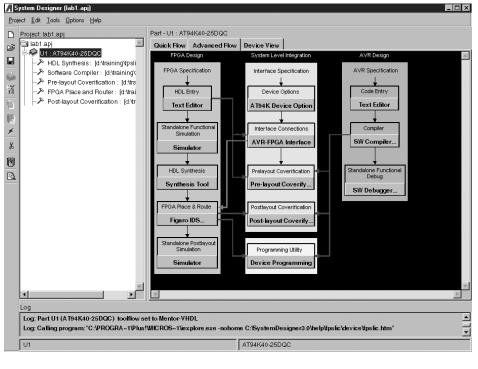
Figure 2-9. Quick Flow View



## 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 1. Press the *Synthesis Tool* button. A dialog box to add VHDL files appears, see FPGA file 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

🖉 Exemplar Logic - LeonardoSpectrum Level 1 Atmel - [Exemplar Lo	gic] 📃 🗖 🗶
<u>.</u> £5 <u>F</u> ile <u>E</u> dit ⊻iew <u>I</u> ools <u>W</u> indow <u>H</u> elp	_ <del>_</del> <del>_</del> <del>_</del> <del>_</del>
5 BA A S S Q D F M @	
Is     Is     Is     Is     Is       Quick Setup         Run the entire flow from this one condensed page. Specify your source files(s), technology and desired frequency, then press Run Flow.         Technology         Imput         Atmel         Atmel         Atmel         Imput         Rate Road         Atmel         Imput         Road Road         Imput         Imput	
Speed Grade: Vorking Directory:	<ul> <li>@rc:/training/fpslic/labl/gs_fpgs.vhd",line 49:</li> <li>@rc:/training/fpslic/labl/gs_fpgs.vhd",line 56:</li> <li>@rc:/training/fpslic/labl/gs_fpgs.vhd",line 61:</li> <li>@rc:/training/fpslic/labl/gs_fpgs.vhd",line 63:</li> <li>@ Info. Command 'read' tinished successfully</li> </ul>
Run Flow Help	Transcript Filtered Transcript
Ready	Working Directory: c:\training\fpslic Line 42 Col 1

- 4. Select *AT94K* as the *Technology*. Leonardo<sup>®</sup> automatically lists <code>qs\_fpga.vhd</code> under input files and lists the output file name <code>asqs\_fpga.edf.qs\_fpga.edf</code> 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.



Functive entre flow from this one condensed page. Specify your source field, technology and desired frequency, then press Run Flow.       Critical Path Report         Technology       Input       GATE       ARRIVAL         - AT40K       GATE       ARRIVAL         - AT40K       GATE       ARRIVAL         - AT6K02       - AT6K02       fd       0.00 (ideal)         - AT6K02       - AT6K02       fd       0.00 (ideal)         - AT6K02       - AT6K02       0.00 (ideal)       CATHODE dup0(0)/6       LUT2       2.90 7.29 up         CATHODE dup0(0)/6       LUT2       2.90 7.29 up       CATHODE (3)/0       0.00 9.29 up         CATHODE (3)/0       Outpl(3)       obuf /AD       obuf 2.00 9.29 up       CATHODE (3)/0         Constraints       Open files:       Working Directory:       working Directory:       unconstrained path         - Design summary in file 'c:/training/fpslic/labl/qs_fpga.xdb       -       Saving the design database in c:/training/fpslic/labl/qs_fpga.xdb         - Writing XDB version 1999.1       -       Saving the design database in c:/training/fpslic/labl/qs_fpga.xdb         - Virting XDB version 1999.1       -       Saving the design database in c:/training/fpslic/labl/qs_fpga.xdb         - Writing XDB version 1999.1       -       Saving the design database in c:/training/fpslic/labl/qs_fpga.xdb	Quick Setup		Exemplar Logic	: 100.1 MHz		
Interpretended and based indexed. Index of the present Higher (unconstrained path)         Image: provide index of the present Higher (unconstrained path)         Image: provide index of the present Higher (unconstrained path)         Image: provide index of the present Higher (unconstrained path)         Image: provide index of the present Higher (unconstrained path)         Image: provide index of the present Higher (unconstrained path)         Image: provide index of the present Higher (unconstrained path)         Image: provide index of the present Higher (unconstrained path)         Image: provide index of the present Higher (unconstrained path)         Image: provide index of the path index of the present Higher (unconstrained path)         Image: provide index of the path index of th	Run the entire flow from this one condensed	d page. Specify your sour				
• Atmel         • Atmel         • Attack         •				icical Fach Report		
AT40K         AT6K02         AT6K04         AT94K         Olock information not specified         clock information not specified         clock information not specified         Clock information not specified         CATBODE_dap0(0)/6         LUT2       2.90         CATBODE_dap0(0)/6         LUT2       2.90         CATBODE_dap0(0)/6       LUT2         CATBODE_dap0(0)/6       LUT2         CATBODE_dap0(0)/6       LUT2         CATBODE_dap0(0)/6       LUT2         CATBODE_dap0(0)/6       UT2         CATBODE_dap0(0)/7       0.00         CATBODE_dap0(0)/7       0.00         CATBODE_dap0(0)/7       0.00         CATBODE_dap0(0)/7       0.00         Catareourid time       not specified         data arrival time       9.29						
ATEXO2         ATEXO4         ATEXO5         ATEXO5         Openide         Openide         Openide         Openide         Openide         Constraints         Openide         Constraints         Openide         Device:         Mthz         Openide         Constraints         Output         Openide         Direc And Route         Contime tand Ro		Jga.vnu	NARE	GAIL	ARRIVAL	
AT34K         reg_cToggle/0       fd       0.00       4.39 up         CATHODE (3)/d       LUT2       2.90       7.29 up         CATHODE (3)/d       obsf       2.00       9.29 up         Device:       Image: Constraints       0.00       9.29 up         Constraints       Ming Directory       Ming       0.00       9.29 up         Constraints       0.00       9.29 up       0.00       9.29 up         Constraints       0.00       0.00       9.29 up       0.00       9.29 up         Constraints       0.00       0.00       0.00       0.00       9.29 up       0.00         Output <td></td> <td></td> <td></td> <td>ified</td> <td></td> <td></td>				ified		
Period:			delay thru clock network		0.00 (ideal)	
CATHODE (a) (D)/G       UT2       2.90 7.29 up         CATHODE (3) (but/PAD       obuf       2.00 9.29 up         CATHODE (3) (but/PAD       obuf       2.00 9.29 up         CATHODE (3) (A       0.00 9.29 up       0.00 9.29 up         data arrival time       9.29         data required time       not specified         data required time       not specified         data required time       9.29         working Directory       with         Optimize       unconstrained path         Design summary in file 'c:/training/fpslic/labl/qs_fpga.sum'       Saving the design database in c:/training/fpslic/labl/qs_fpga.sub'         Output       Writing file c:/training/fpslic/labl/qs_fpga.edf       Writing file c:/training/fpslic/labl/qs_fpga.edf         Place And Route       CRV time taken for this run was 2.15 sec       Run Successfully Ended On Thu Sep 20 15:05:29 Pacific Daylight Ti	AT94K					
CATHODE [3]_obut/PAD       obuf       2.00 9.29 up         CATHODE [3]/       0.00 9.29 up         data arrival time       9.29         constraints       unconstrained path         Optimize				fd	0.00 4.39 up	
CATHODE (3)/       0.00 9.29 up         Device:       0.00 9.29 up         Gata arrival time       9.29         data arrival time       not specified         Speed Grade:       working Directory:         Working Directory:       working Directory:         Constraints       unconstrained path         Optimize       Design summary in file 'c:/training/fpslic/labl/qs_fpga.sum'         Saving the design database in c:/training/fpslic/labl/qs_fpga.sum'         Saving the design database in c:/training/fpslic/labl/qs_fpga.sum'         Saving the design database in c:/training/fpslic/labl/qs_fpga.sum'         Start LUT decomposition for design.work.03.sEEHAV         CPU time taken for this run was 2.15 sec         CPU time taken for this run was 2.15 sec         Run Successfully Ended On Thu Sep 20 15:05:29 Pacific Deylight Ti         0						
Device:       Image: Constraints         Open Rise:       Image: Constraints         Working Directory:       Image: Constraints         Optimize       Seed Grade:         Optimize       Image: Constraints         Optimize       Image: Constraints         Optimize       Image: Constraints         Optimize       Image: Constraints         Output       Image: Constraints         Image: Constraints       Image: Constraints         Image: Constraints       Image: Constraints         Image: Constraints <td></td> <td></td> <td></td> <td>obuf</td> <td></td> <td></td>				obuf		
Device:       Image: Constraints         Speed Grade:       Image: Constraints         Image: Constraints       Image: Constraints         Optimize       Image: Constraints         Optimize       Image: Constraints         Optimize       Image: Constraints         Optimize       Image: Constraints         Output       Image: Constraints         Image: Constraints       Image: Constraints         Image: Constraints       Image: Constraints         Image: Constra						
Device:       Image: Constraints         Constraints       Image: Constraints         Optimize       Image: Constraints         Optimize       Image: Constraints         Optimize       Image: Constraints         Optimize       Image: Constraints         Output       File:         O			uata arrivar time		5.25	
Speed Grade:       Open File::	Device:		data required time		not specified	
Speed Grade:       Upen Nes:       9.29         Working Directory:       Withing Directory:       unconstrained path         Constraints	· · · · · · · · · · · · · · · · · · ·	-	data required time		not specified	
Working Directory:       Working Directory:       Unconstrained path         Constraints       Cock Frequency       Mhz         Optimize       Freserve Hierarchy	- Up	ien files:	data arrival time			
Constraints         Optimize         Extended Optimization Effort         Preserve Hierarchy         Output         Output         Output         Output File: [c:/training/fpsic/lab1/qs_fpga.edf         Place And Route         Run/integreted Rises and Route		orking Directory				
Clock Frequency       Mhz         Optimize       Design summary in file 'c:/training/fpslic/labl/qs_fpga.sum'         Saving the design database in c:/training/fpslic/labl/qs_fpga.sum'         Output       Writing file c:/training/fpslic/labl/qs_fpga.xdb         Writing file c:/training/fpslic/labl/qs_fpga.edf       Writing file c:/training/fpslic/labl/qs_fpga.edf         Place And Route       CPU time taken for this run was 2.15 sec         Run Integreted Reperand Route       0	- 116	Diking Directory.			unconstrained path	
Optimize       Design summary in file 'c:/training/fpslic/labl/qs_fpga.sum'         Extended Optimization Effort       Preserve Hierarchy         Output       Saving the design database in c:/training/fpslic/labl/qs_fpga.xdb         Output       Writing file c:/training/fpslic/labl/qs_fpga.edf         Place And Route       CPU this taken for this run was 2.15 sec         Run Integreted Flore and Route       Run Successfully Ended on Thu Sep 20 15:05:29 Pacific Daylight Ti	Vi Vi					
Extended Optimization Effort       Preserve Hierarchy         Output	Constraints					
Extended Optimization Effort       Preserve Hierarchy         Output	Constraints	Mhz				
Output	Constraints	Mhz		Las devaiusing devaluing	light for the second	
Output File: C./training/fpsiic/lab1/qs_fpga.edf         Place And Route         Place And Route         Run integrated Pisce and Route	Constraints Clock Frequency					
Place And Route     Writing file c:/training/fpslic/labl/qs_fpga.edf     CPU time taken for this rum was 2.15 sec     Rum Successfully Ended On Thu Sep 20 15:05:29 Pacific Daylight Ti     0	Constraints Clock Frequency Optimize Extended Optimization Effort		Saving the design datab Writing file c:/trainin	ase in c:/training/ ng/fpslic/labl/qs_fp	fpslic/labl/qs_fpga.xdl	ı
Place And Route       CPU time taken for this run was 2.15 sec         Image: Run Integrated Race and Route       Run Successfully Ended On Thu Sep 20 15:05:29 Pacific Daylight Ti	Constraints Clock Frequency Optimize Extended Optimization Effort Output	Preserve Hierarchy	Saving the design datab Writing file c:/trainin Writing XDB version 199	pase in c:/training/ ng/fpslic/labl/qs_fp 99.1	fpslic/labl/qs_fpga.xdb ga.xdb	I
Run integrated Place and Route	Constraints Clock Frequency Optimize Extended Optimization Effort Output	Preserve Hierarchy	Saving the design datab Writing file c:/trainir Writing XDB version 199 Start LUT decomposition	pase in c:/training/ ng/fpslic/labl/qs_fp 99.1 1 for design .work.Q	fpslic/labl/qs_fpga.xdl ga.xdb S.BEHAV	ı
	Constraints Clock Frequency Optimize Extended Optimization Effort Output Output Output File: [c:/training/ipsic/lab1/qs_fpg	Preserve Hierarchy	Saving the design datah Writing file c:/trainin Writing XDB version 199 Start LUT decomposition Writing file c:/trainin	pase in c:/training/ ng/fpslic/labl/qs_fp 99.1 n for design .work.Q ng/fpslic/labl/qs_fp	fpslic/labl/qs_fpga.xdl ga.xdb S.BEHAV	ı
	Constraints Clock Frequency Optimize Extended Optimization Effort Output Output Place And Route	Preserve Hierarchy	Saving the design datak Writing file c:/trainir Writing XDB version 199 Start LUT decomposition Writing file c:/trainir CPU time taken for this	wase in c:/training/ ug/fpslic/labl/qs_fp 19.1 a for design .work.Q ug/fpslic/labl/qs_fp s run was 2.15 sec	fpslic/labl/qs_fpga.xdd ga.xdb S.BEHAV ga.edf	
Pur Clau	Constraints Clock Frequency Optimize Extended Optimization Effort Output Output Place And Route	Preserve Hierarchy	Saving the design data Writing file c:/trainin Writing XOB version 195 Start LUT decomposition Writing file c:/trainin CPU time taken for this Run Successfully Ended 0	pase in c:/training/ gyfpslic/labl/qs_fp 19.1 A for design .work.Q ng/fpslic/labl/qs_fp s run was 2.15 sec On Thu Sep 20 15:0	fpslic/labl/qs_fpga.xdd ga.xdb S.BEHAV ga.edf	
	Constraints Clock Frequency Optimize Extended Optimization Effort Output Output Place And Route	Preserve Hierarchy	Saving the design datak Writing file c:/trainir Writing XDB version 199 Start LUT decomposition Writing file c:/trainir CPU time taken for this	pase in c:/training/ gyfpslic/labl/qs_fp 19.1 A for design .work.Q ng/fpslic/labl/qs_fp s run was 2.15 sec On Thu Sep 20 15:0	fpslic/labl/qs_fpga.xdd ga.xdb S.BEHAV ga.edf	

Figure 2-13. Synthesis Results

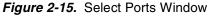
- 6. It is not required to save the project. Close Leonardo and return to System Designer.
- **2.7 AVR-FPGA** 1. Press the *AVR-FPGA Interface* button. A dialog box to select the top-level entity name appears, see Figure 2-14.

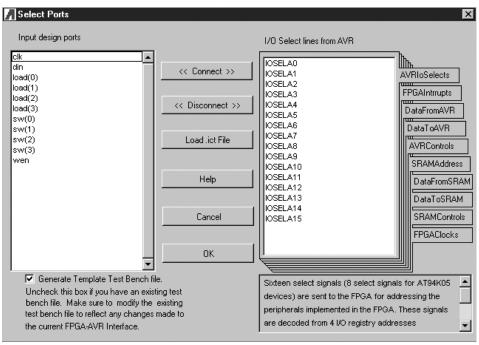
Figure 2-14. Top-level Entity Name Dialog Box



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







- 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.

Select Ports			×	
Input design ports		FPGA Clocks from AVR		
clk (GCLK5) din (ADINA0) load(0) (IOSELA0) load(4) (IOSELA0)	<< Connect >>	GCLK5 (clk) GCLK6		
load(1) ( IOSELA1) load(2) ( IOSELA2) load(3) ( IOSELA3) sw(0)	<< Disconnect >>		FPGAIntrrupts DataFromAVR	
sw(0) sw(1) sw(2) sw(3)	Load .ict File		DataToAVR AVRControls	
wen (FIOWEA)	Help		SRAMAddress DataFromSRA	
	Cancel		DataToSRAM	
	ОК	L	FPGAClocks	
Generate Template Test Bench file. Uncheck this box if you have an existing test bench file. Make sure to modify the existing test bench file to reflect any changes made to the current FPGA-AVR Interface.		Two global clock buses (5 and 6) are driven from clock signals generated within the AVR core. The AVR system clock connects to the GCK5 and GCK6 has a choice of the following:		

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

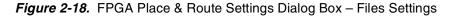
	×
Do you want to add the genera qs_fpga_pretb.vhd to the Pre-I	
Yes	No

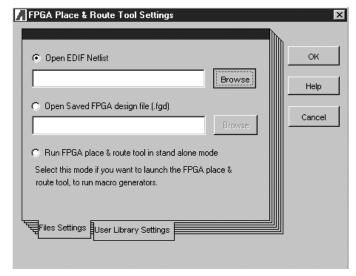
9. Select Yes.



## 2.8 FPGA Place & Route

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





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

Please select a file 🛛
Drive: C:\
c:\training\fpslic\lab1
qs_fpga.edf
File Name: qs_fpga.edf OK
Files of Type: *.edf Cancel

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.

FPGA Place & Route Tool Settings		×
c:\training\fpslic\user94k.lib	•	OK Help Cancel
Add Remove		
User Library Settings		

*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

Import Constraints
ē. Auto
5 Auto
aints
ire FPGA design flow. To run individu e and route, click on OK and select sktop
ir e

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 <code>yourfile-name.rct</code> 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.

e Name:	Directories:	ок
gs_fpga.pin	c:\training\fpslic\lab1	Cancel
qs_fpga.pin	c:\	
	training fpslic	Help
	lab1	
ist Files of Type:		
	Drives:	

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.



		<ul> <li>14. Maximize the compile window. Choose one of these options to zoom in: <ul> <li>Go to the <i>View</i> menu and select <i>Zoom to Area</i>, or</li> <li>Press <i>F7</i></li> </ul> </li> <li>Choose one of these options to zoom out: <ul> <li>Go to the <i>View</i> menu and select <i>Zoom Out</i>, or</li> <li>Press <i>F8</i></li> </ul> </li> <li>15. Go to the <i>File</i> menu and select <i>Exit</i>.</li> </ul> <li>16. When asked, choose <i>yes</i> to save the design. Figaro will close.</li> <li>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.</li>	
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 Image-Craft compiler.	
2.9.1	Changing the Compiler Tool Settings	<ol> <li>Right-click on the SW Compiler button, a pop-up menu appears, see Figure 2- 23.</li> </ol>	
		Figure 2-23. SE Compiler Right-click Menu	
		Launch Tool ChangeTool Settings	

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

Help

Add Design Files...

## Figure 2-24. File Browser Window

Please select a file	×
Drive: C:\	• E *
c:\ice\bin	
admin.exe avrcalc.exe CI.EXE CO.EXE coffdump.exe diff.exe grep.exe iasavr.exe iccavr.exe iccavr.exe	
File Name: iccavride.exe	OK
Files of Type: .exe	Cancel
Executable file	Defaults



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

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

Do you want to upda	te Plug In:D:\SystemDesigner3.0\bin\at94k.plg with the updat
Tool settings? Updat	ting the plugin will make the new tool settings effective for all
subsequent session	e of Svetem Decimer
	o u oyatem dealquer
000000000	o o oyaram bearginar
o dissequent o coordin	

- 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.

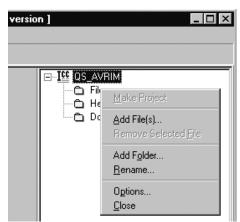
Figure 2-26. ImageCraft Window

5 I	nagel	Craft ID	)E for I	CCA	/R (9	itand	ard Vei	sion) [	WAR	NING:	30 D	ays E\	/ALUATI	ON versi	on]				_ 🗆 ×
<u>F</u> ile				<u>P</u> ro	ject	<u>R</u> CS	T <u>o</u> ols	<u>T</u> ermir	nal <u>H</u>	elp									
Ð	õ		1	d'	Ê	5	æ	- 24	STOP		X	11001							
															<u><u> </u></u>	NO PRO	DJECT O	PEN	
						1]	lo Open	File ]						[ N	o Open Pro	oject ]			

- 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.

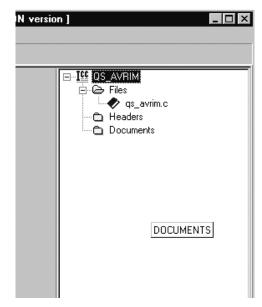


#### *Figure 2-27.* Add File(s)... Pop-up Menu



- 9. Select qs\_avrim.c and press Open.
- 10. Press imes 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.

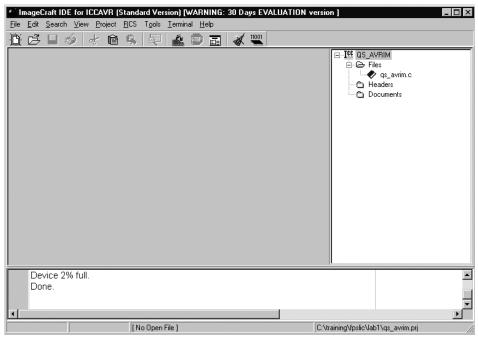


Compiler Options Dialog Box	×
Paths       Compiler       Target         Device Configuration       Image: Configuration         Custom       Image: Configuration         Memory Sizes (Bytes)       Program Memory         Program Memory       8192         Data Memory       512         EEPROM       Image: Configuration         Text Address (WORD)       Image: Configuration         Data Address (BYTE)       Image: Configuration         Image: Use long CALL/JMP       Image: Configuration         Enhanced Core       Image: Configuration         Program Type       Root Loader Options         Program Type       Image: Configuration         Internal SRAM       Image: Configuration	PRINTF Version         Image: small (int only, no modifier)         Image: long (+ long, and modifiers)         Image: float (+float, [needs > 8K])         AVR Studio Simulator IO         Additional Lib.         Strings in FLASH only         Advanced         Return Stack Size       16         Non-default Startup         Unused ROM Fill Pattern         Do NOT use R20R23
OK Cancel Set As Default	Load Default <u>H</u> elp

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.

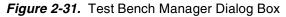
Figure 2-30. ImageCraft Status Window



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.



Please select the test bench file to be included with Pre-Layout Coverification To edit the file, click on the Edit File button
To edit the file, click on the Edit File button
c:\training\fpslic\lab1\qs_fpga_pretb.vhd Browse
Hardware Simulator Options
Execute Command File at startup
Browse
OK Edit Testbench Cancel

- 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

HDLPlanner: qs_fpga_pretb.vhd			×
<u>File E</u> dit V <u>H</u> DL <u>I</u> ools			
	: →	<u> 716 - 716</u> -	+∰ A+\$A /**/
Text: Category: Component: Medium Text: Component:		▼ De	fine Instance
SIGNAL sig_sw : std_logio_vector(3 downto 0); SIGNAL sig_wen : std_logio;			
SIGNAL sig_alpha0       : std_logio_vector(14 downto 0);         SIGNAL sig_alpha1       : std_logic_vector(14 downto 0);			
SIGNAL sig_cathode : std_logic_vector(3 downto 0); SIGNAL sig_fints : std_logic_vector(3 downto 0);			_
<pre>SIGNAL sig_avrolk_sel : std_logio_vector(1 downto 0) :="00"; SIGNAL dummy_vector : std_logio_vector(7 downto 0);</pre>			
<pre>SIGNAL dummy : std_logio; signal sig_ext : std_logio_vector(3 downto 0) := "1111";</pre>			
signal sig_pd : std_logic_vector(7 downto 0);			
BEGIN			
Instantiating top level design Component qs			<b>•</b>
File: c:\training\fpslic\lab1\qs_fpga_pretb.vhd	Ln 51	Col 46	EDITING



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

HDLPlanner: qs_fpga_pretb.vhd		×
<u>File E</u> dit V <u>H</u> DL <u>I</u> ools		
♥ ┗☞■● ४ ☜ ⑱ ♡ ♡ ₩ 및 주 쏘 →	<u> </u>	A# <b>4</b> /*_*/
Text: Category: Component: Medium I Register	▼ Define	Instance
510.55		<b>_</b>
FIORE => open, FIOWE => sig_wen,		
avrolk_sel => sig_avrolk_sel,		
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		
PORTE => dummy_vector,		
TOSC1 => dummy, Input to the Timer/Counter oscillator amplifier	•	
UART0_RX0 => dummy, UART0 receive (input) pin		
UART1 RX1 => dummy, UART1 receive (input) pin		
SCL_IN => dummy, 2-wire serial input clock		
SDA_IN => dummy 2-wire serial input data		
		<b>•</b>
File: c:\training\fpslic\lab1\qs_fpga_pretb.vhd Ln 20	9 Col 26 E	DITING

# 6. Scroll down to "stimulus process" and add the following stimulus as shown in Figure 2-34:

stimulus\_process: PROCESS BEGIN

sig\_sw <= "0001"; --switch 5,you should see 0000001111111110 on alpha0 & alpha1
wait for 10 us;</pre>

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;</pre>

sig\_ext <= "1101"; --switch 2,you should see Counting Down pattern on
PORTD(LEDs)
wait for 10 us;</pre>



sig\_sw <= "0100"; --switch 7,you should see 000000010101010 on alpha0 & alpha1
wait for 10 us;</pre>

```
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;</pre>

sig\_ext <= "0111"; --switch 4,you should see Bounce pattern on PORTD(LEDs)
wait for 10 us;</pre>

END PROCESS stimulus\_process;

END arch\_test\_bench;

#### Figure 2-34. Adding Stimulus

HDLPlanner: qs_fpga_pretb.vhd 🛛
<u>F</u> ile <u>E</u> dit V <u>H</u> DL <u>I</u> ools
Text: Category: Component: Medium  Category: Component: Define Instance
SCL_OUT => open, 2-wire serial output clock SDA_OUT => open 2-wire serial output data
);
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;
File: c:\training\fpslic\lab1\qs_fpga_pretb.vhd Ln 1 Col 1 EDITING
The containing upsiculatings piga precound Entry Containing upsiculatings

- **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

Л Б	×
Select the top level entity of the VHDL testbench	1
qs	
qs_test_bench	
OK Cancel	

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 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 <code>qs\_fpga\_pretb.vhd</code> 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.

 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.

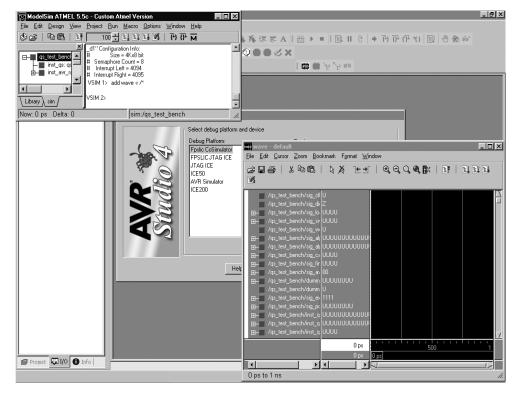
Debug Platform:
Poslic CoSimulator FPSLIC JTAG ICE JTAG ICE ICE50 AVR Simulator ICE200

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 II 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 
interview the details of External Interrupts and PORTD, see Figure 2-38.

Figure 2-38. External Interrupts and PORTD Details

AVRStudio - [qs_avrim.c]						_ B ×
Eile Project Edit View	<u>T</u> ools <u>D</u> ebug	Window Help				- 8 ×
0 2 8 <b>8 5</b> 3 4	68 22		A % %	% @ # A	₩ ▶ = 11 0 0 0 0	₩ 40 H 10 H 10 H
	displad	▼ % % ef 1				
	uisableu					
					1 🗰 Hr Fr Fr BUTO	
Workspace			- ×			
Name	Value	Bits	Address		rnal Interrupt 2 Service Rou interrupt_handler_EXT_INT2_	
			<b>_</b>		[_INT2_isr(void)	_
⊞ - 🗐 Register 16-31				{ cFla	ags = 0b00000100;	
Processor	0x00AE			}		
Stack Pointer	0x0FFC				Interrupt 3 Service Routine	
Sycle Counter	17504			#pragma	interrupt_handler FPGA_INT3	_isr:FPGA_INT3
···· 🕸 X-register	0x0062			Voia PP	GA_INT3_isr(void)	
Yregister	0x0FEE			1F1a	ags = 0b00000011;	
Z-register     Frequency	0x938A 4.00			3		
Stop Watch	4376.00 us			1		
E S I/O Fpslic_avr_core					rnal Interrupt 3 Service Rou interrupt handler EXT INT3	
Ē Ē CPU				void EX	[_INT3_isr(void)	
EXTERNAL_INTERR				CE1	ags = 0500001000:	
E FPGA_INTERRUPTS	0x00		13 (33)	}		_
E 🕈 FISUA	0x0F		14 (34)	woid in	itAT94K(void)	
🕀 🛷 FISUB	0x00		15 (35)	{		
E 🏘 FISUC	0x00			SREG	G δ <sub>c</sub> = ~(0x80);	// Disable Global Int
	0x00		17 [37]	POR		// Push-Pull Zero Out
	0x1C		12 (32)	DDRI	$0 = 0 \times FF;$	// Configure PORTD as
DDRD	0xFF		11 (31)	FISC		// Enable FPGA<>AVR
🗄 🔁 PIND	0x1C		10 (30)	FIST	$JA = 0 \times FF;$	// Clear & Enable FP(
				EIM	7 = 0xFF;	// Clear & Enable Ext
E TIMER_COUNTER_C	) 0x00		33 (53)	SREO	G  = 0x80:	// Enable Global Inte
E (S) TENTO	0x00		32 (52)	}		
	0x00		31 (51)			
B S TIMER_COUNTER_1				void ma:	in(void)	
				i une	igned char dElage = 0.	// Rit Flags for Dir
				ļu į		
Project 🐺 I/O 🕕 Info				gs_avrim.c		
		Modelsim R	unning		Fpslic_avr_o Fpslic CoSimula Auto Stopped	i 😑 Ln 73, Col 1 🛛 CAP NUM SCRL

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  $\mu$ s to see different patterns. The Time Elapsed field on the *Processor* window reports the time.

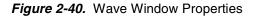
16. Press the *Debug Break* button **II** followed by the *Hardware Break* button **II** to take control over ModelSim.

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

/q_let/exc/vis_0         0           /q_let/	386   1 BB	k X 1± ≠	ା ପ୍ ପ୍	Q, 🔍 📴	IF I	i i 🕸						
	/qs_test_bench/sig_cl	0			-	1		-				_
g_         /(1)         C010010010010010010010010010010010000000	📕 /qs_test_bench/sig_dii	0										
a / q_1 bet, beeck/sig_al       0       Image: dispect/sig_al       00000111111110       Image: dispect/sig_al       000000111111110         b / q_1 bet, beeck/sig_al       000000111111110       Image: dispect/sig_al       000000111111110       Image: dispect/sig_al       000000111111110         b / q_1 bet, beeck/sig_al       000000111111110       Image: dispect/sig_al       000000111111110       Image: dispect/sig_al       000000111111110         b / q_1 bet, beeck/sig_al       000       00000011111110       Image: dispect/sig_al       000000111111110       Image: dispect/sig_al       0000000111111110       Image: dispect/sig_al       000000111111110       Image: dispect/sig_al       0000000111111110       Image: dispect/sig_al       0000000111111110       Image: dispect/sig_al       000000000000000000000000000000000000												
q       (q) Let, bench/sig, al       000000111111110       (q) Let, bench/sig, al       000000111111110         (q) Let, bench/sig, al       000000111111110       (q) Let, bench/sig, al       000000111111110         (q) Let, bench/sig, al       000000111111110       (q) Let, bench/sig, al       (q) Ditto (q) Dit		1000								10)(0100)(1000	(0001)(0010)(01	00)
generative         generative <thgenerative< th="" thgenerative<=""> <thgenerative< th=""></thgenerative<></thgenerative<>		0										
4/q2_test_bench/sig_s       1001       000         4/q2_test_bench/sig_s       000         4/q2_test_bench/sig_s       000         4/q2_test_bench/sig_s       000         4/q2_test_bench/sig_s       000         4/q2_test_bench/sig_s       001         4/q2_test_bench/sig_s       001         4/q2_test_bench/sig_s       001100         4/q2_test_bench/sig_s       001100         4/q2_test_bench/sig_s       001100         4/q2_test_bench/sig_s       001100         4/q2_test_bench/sig_s       001100         4/q2_test_bench/sig_s       0001100         4/q2_test_bench/sig_s       0001100         4/q2_test_bench/sig_s       0001100         4/q2_test_bench/sig_s       0001100         4/q2_test_bench/sig_s       00000011111110         4/q2_test_bench/sig_s       00000011111110         4/q2_test_bench/sig_s       00000011111110         4/q2_test_bench/sig_s       00000011111110         4/q2_test_bench/sig_s       00000011111110         4/q2_test_bench/sig_s       0000001111110         4/q2_test_bench/sig_s       00000011111110         4/q2_test_bench/sig_s       00000011111110         4/q2_test_bench/sig_s       00000011111100			<u>u i i</u>			<u> </u>	<u>t i i</u>					
4/2_1est_bench/stg.s       00         4/2_1est_bench/stg.s       000         4/2_1est_bench/stg.s       000         4/2_1est_bench/stg.s       000         4/2_1est_bench/stg.s       0000001111110         4/2_1est_bench/stg.s       0001100         4/2_1est_bench/stg.s       0001100         4/2_1est_bench/stg.s       0001100         4/2_1est_bench/stg.s       0001100         4/2_1est_bench/stg.s       0000         4/2_1est_bench/stg.s       0000         4/2_1est_bench/stg.s       0000         4/2_1est_bench/stg.s       0000         4/2_1est_bench/stg.s       0000				<u> </u>		χ_χ		<u>χ</u> χ				
											kaan okaa oa ka d	
g. / (a): Let. bench/max         ZZZZZZ         U         U         U           / (a): Let. bench/max         0111         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U         U				11,0111,1110	111011101110	11111111011101	1011,0111,11	10,1101,1011	<u>,0111,1110,11</u>	01,1011,0111		<u>ші</u>
4			00									
		0111	Y1110Y1101	1011/01111	110111011101	10111111110111	011101110111	11110/1101/10	11/0111/1110	1101/1011/0	11/1110/1101	1011
-       /q2_test_bench/mat_d       00000011111110       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -       -			Y Y Y	Y Y	Y Y Y	Y Y	Y Y Y	Y Y				
4/q_test_bench/mat_g       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001       1001 <td< td=""><td></td><td></td><td>hî î î</td><td>⊨<del>ĵ ĵ</del></td><td><del>î î î</del></td><td>÷ŶŶ</td><td>Î</td><td>F<del>î î</del></td><td></td><td></td><td></td><td></td></td<>			hî î î	⊨ <del>ĵ ĵ</del>	<del>î î î</del>	÷ŶŶ	Î	F <del>î î</del>				
4/q_test_bench/mat_q       0111       111011101101101110111011101101101101101		1001		າກກຳກາງກຳການ	ມໂນສາສມາໂນນາາກາ່າ	ກ່າວກ່ຽວແມ່ນການ	ກັນກາງກາງກາງກາງກາງກາງກາງກາງກາງກາງກາງກາງກາງກ	ພາກການການການການ				
4/22_text_bench/mat_g       0       Put/Put/Put/Put/Put/Put/Put/Put/Put/Put/		0111	1110/1101/10	11)0111)1110	0111011101110	11111111011101	101110111111	10/1101/1011	0111111110111	011101110111	1111011101101	i111
g-         /qs_let_bench/ms_c         0000         D1001000000000000000000000000000000000		0										
a	/qs_test_bench/inst_q	0	ากมากกากกา	հուրուն	ปกางบานบาบป	Առուստոսո	ການນານ		hunnun			
/q:_let_bench/int_0       0       1000000000000000000000000000000000000	🖅 /qs_test_bench/inst_q	0000		,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		100000000000000000000000000000000000000		,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	000000000000000000000000000000000000000	00		
/ds_let_bench/mst_q       29       0       1       2       13       14       15       16       17       18       19       20       12       23       24       25       25       26       12       23       14       15       16       17       18       19       20       12       123       124       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       25       2	🖅 /qs_test_bench/inst_q	1000	0001/0010/01							10(0100)(1000	(0001)0010(01	00)
4/aset_bench/mat_a       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0	📃 /qs_test_bench/inst_q	0										
4/q:_let_bench/inst_0       0       10/10/10/10/10/10/10/10/10/10/10/10/10/1	📕 /qs_test_bench/inst_q	29	0)1)2)	3)(4)(5)	6 (7 )8 )	3 )(10 )(11 )(1	2 (13 (14 (19	<u>) (16 )(17 )(18</u>	)19 )20 )21	)22 )23 )24	(25)(26)(27)	28
a_ / qs_text_bench/inst_a 0001 0001000100010000 10000 000100100010000 0001001												
/qs_lest_bench/inst_a       U       Image: Lest_bench/inst_a       I											<u> </u>	
/qs_letl_bench/inst_a       U       Image: Section of the sect		0001	<u>X (0010)</u> 0	(00)(1000)	<u>,0010,0100,</u> 1	000) <u>(</u> 0010	<u>)(0100)(1000)</u>	(0010)(010	)1000)0001			
/qs_text_bench/inst_a       U       U       U         /qs_text_bench/inst_a       U       U       U       U         /qs_text_bench/inst_a       U       U       U       U       U         /qs_text_bench/inst_a       U       U       U       U       U       U         /qs_text_bench/inst_a       U       U       U       U       U       U       U       U       U       U       U       U       U       U       U       U       U       U       U       U       U       U       U       U       U       U       U       U       U       U       U       U       U       U       U       <		U										
/qs_lext_bench/inst_a       U       Image: Section of the sect		U										
/gs_test_bench/inst_e         U           /gs_test_bench/inst_e         U           /ds_test_bench/inst_e         U           474220000 ps         100 us         200 us         300 us         400 us		U										
/qs_test_bench/mat_a         U         Image: Constraint of the second se		U										
/qs_ber_bench/inst_e 474820000 ps100 us 200 us 300 us 400 us		u										
474820000 ps 100 us 200 us 300 us 400 us		0										
474620000 ps 100 us 200 us 300 us 400 us		<u> </u>										
		474820000 ps										
		0 ps	0 ps									

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



🕅 Wave Window Prope	rties	_ 🗆 ×					
Display Signal Path	Sn	ap Distance					
(# elements)	10	(pixels)					
Use 0 for full path	Re	w Margin					
	4	(pixels)					
Justify Value	Ch	ild Row Margin					
Eeft C Right	2	(pixels)					
Dataset Prefix Display							
O Always Show Dataset Prefixes							
Show Dataset Prefixes if 2 or more							
Never Show Da	taset Prefix	es					
	<u>o</u> k	<u>C</u> ancel					

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.

Beithe		
sig_clk	0	
📕 sig_din	0	
sig_load	0000	04003000000000000000000000000000000000
sig_sw	1000	0001/0010/0100/1000/0001/0010/100/0001/00010/010/000/0001/000/0001/0010/0100/1000/0001/0010/0100/100/0001/0010/0100/100/00001/0010/0100/00001/0010/0100/00001/0010/0100/00001/000/00001/000/00001/000/000000
sig_wen	0	
sig_alpha0	000000111111110	X X X X X X X X X X X X X X X X X X X
sig_alpha1	000000111111110	X X X X X X X X X X X X X X X X X X X
sig_cathode	1001	ארייגע אווארא אין אין אין אין אין אין אין אין אין אי
sig_fints	0111	1110(1101(1011)(0111)(1110(1101)(0111)(0111)(1101(1011)(0111)(1110)(1101)(011)(111)(111)(0111)(1110)(1101)(0111)(111)(110)(1101)(1011)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(11)(11)(11)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(111)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(11)(
sig_avrclk_sel	00	
dummy_vector	77777777	
dummy	U	
sig_ext	0111	<u> </u>
sig_pd	00011100	000000000000000000000000000000000000000
- alpha0	000000111111110	X X X X X X X X X X X X X X X X X X X
- alpha1	000000111111110	<u> </u>
- cathode	1001	2000 000000000000000000000000000000000
– fints	0111	1110(1101)10111(1110)1101(1011)0111(0111)0110(1101)1011(0111)110(1101)1011(01101)1011(0111)(1101)1011(0111)110(1101)1011)
📕 clk	0	
📕 din	0	
– load	0000	
- sw	1000	(0001/0010/0100/1000/0001/0010/0100/1000/0001/0010/0100/1000/0001/0010/0100/1000/0001/0010/0100/1000/0001/0010/0100/
wen	0	
cpulses	29	0 X1 X2 X3 X4 X5 X4 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20 X21 X22 X23 X24 X25 X26 X27 X28 X
cclock	0	
ctoggle	0	
– iload	0001	X X0010X0100 X1000X X0010X0100X1000 X X0010X1000X X0010X1000 X X0010X0100X1000 X0001
📕 fadin0	U	
📕 fadin1	U	
fadin2	U	
📕 fadin3	U	
fadin4	U	
fadin5	U	
fadin6	U	
	474820000 ps	100 us 200 us 300 us 400 us

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

0 ps to 498561 ns

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* ton 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

Test Bench Manager	×
Please select the test bench file to be included with Post-Layor To edit the file, click on the Edit File button	ut Coverification
-	Browse
c:\training\fpslic\lab1\figba\qs_fpga_posttb.vhd	Diowse
Hardware Simulator Options	
	Browse
-	
Disable Timing Checks	
Disable Glitch Warnings	
OK Edit Testbench Cano	el

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

HDLPlanner: qs_fpga_posttb.vhd		×
<u>File E</u> dit V <u>H</u> DL <u>I</u> ools		
♥ ┗┏┛┓ % ๒๒ ゃ ♀ ₩ ฿ ★ ★ →	<u></u>	A#4 /*.*/
Text: Category: Component: Medium I Register I	▼ Defi	ne Instance
<pre>signal sig_SCL_OUT: STD_LOGIC; signal sig_SDA_OUT: STD_LOGIC; signal sig_PORTD: STD_LOGIC_VECTOR (7 downto 0); signal sig_PORTE: STD_LOGIC_VECTOR (7 downto 0); signal one : STD_LOGIC := '1'; signal zero : STD_LOGIC := '0';</pre>		<u>.</u>
<pre>signal sig_ext : std_logio_vector(3 downto 0) := "1111"; signal sig_pd : std_logio_vector(7 downto 0); BEGIN</pre>		
inst_qs_fpga:QS <b>port map</b> ( clk => sig_CLK, din => sig_DIN, load => sig_load, File: c:\training\fpslic\lab1\figba\gs_fpga_posttb.vhd Ln 180	6 Col 1	



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

HDLPlanner: qs_fpga_posttb.vhd _File _Edit VHDL _Iools			×
฿ ๒๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛๛		<b>IA</b> 48	A#4 /**/
Text: Category: Component: Medium Register T	-	Define	Instance
FINT14 => one,			<u> </u>
FINT15 => one,			
UARTO_RXD => sig_UARTO_RXD,			
UART1_RX1 => sig_UART1_RX1,			
TOSC1 => sig_TOSC1,			
SCL_IN => sig_SCL_IN,			
SDA_IN => sig_SDA_IN,			
EXT_INTO => 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			
UART0_TXD => sig_UART0_TXD,			
UART1_TX1 => sig_UART1_TX1,			
XTAL2 => sig_XTAL2,			
TOSC2 => sig_TOSC2,			_
SCL_OUT => sig_SCL_OUT,			_
File: d:\SystemDesigner\designs\doc2331\figba\qs_fpga_posttb.vh( In 241	Col 49	E	DITING

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

PORTD => sig\_PD



◍▯◶▤◓	X 🖻 🛍	S CH &	<u></u> <u></u> <u></u> <u></u> <u></u> <u></u> <u></u> <u></u> <u></u> <u></u>	: →	<u>+16</u> +1	•₩	A⇒A	/**
ext: Categ Medium Regis		Component:			-	Define	Insta	nce
TOSC1 => sig TOSC1,		<u>.</u>			_			
SCL IN => sig_SCL IN,								
SDA IN => sig SDA IN,								
EXT INTO => sig ext(0),-	- External Inter	rupt 0						
EXT_INT1 => sig_ext(1),-								
EXT_INT2 => sig_ext(2),-								
EXT_INT3 => sig_ext(3),-	- External Inter	rupt 3						
UARTO_TXO => sig_UAR	TO_TXO,	·						
UART1_TX1 => sig_UAR	T1_TX1,							
XTAL2 => sig_XTAL2,								
TOSC2 => sig_TOSC2,								
SCL_OUT => sig_SCL_C								
SDA_OUT => sig_SDA_(								
PORTD => sig_pd, LEL	Ds							
PORTE => sig_PORTE								
);								

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

HDLPlanner: qs_fpga_posttb.vhd
<u>File Edit VH</u> DL <u>I</u> ools
Text: Category: Component: Medium  Category: Component: Define Instance
PORTE => sig_PORTE
);
stimulus_process: PROCESS BEGIN
sig_sw <= "0001"; switch 5, you should see 0000001111111110 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;
File: c:\training\fpslic\lab1\figba\qs_fpga_posttb.vhd Ln 1 Col 1 EDITING



- 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

Please select the test bench file to be included with Post-Layout Coverification To edit the file, click on the Edit File button c:\training\fpslic\lab1\figba\qs_fpga_posttb.vhd Browse
c:\training\fpslic\lab1\figba\qs_fpga_posttb.vhd Browse
Hardware Simulator Options
Execute Command File at startup
c:\training\fpslic\lab1\wave.do Browse
Disable Timing Checks
🗖 Disable Glitch Warnings
OK Edit Testbench Cancel

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

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

	×
Select the top level entity of the VHDL te	stbench
lpm_counter_f_t_f_t_9_0 lpm_counter_f_t_f_1_4_0 post_test_bench qs	
	<b>T</b>
OK Cancel	

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.



Л		х
	)F timing value <minimum, maximum<br="" typical,="">on the design instance</minimum,>	1>
minimum typical		]
maximum	OK Cancel	1

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.

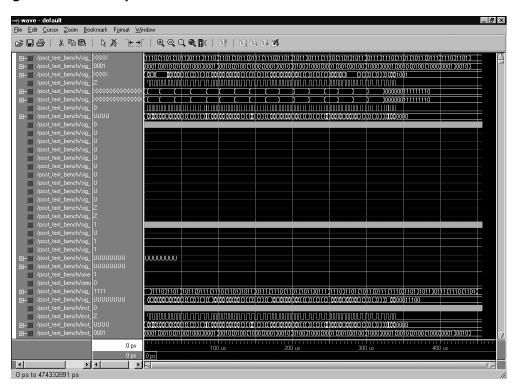
If ModelSim is not responding, AVR Studio is in control and you have to press the *Auto Step* button or *Run* for program execution. To view the waveforms, press the *Break* command II from the *Debug* menu to stop program execution, followed by a hardware break button to return control to ModelSim.

- 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  $\mu$ s to see different patterns. The Time Elapsed field on the *Processor* window reports the time.

- 15. Press the *Debug Break* button II tfollowed by the *Hardware Break* button II 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* ton 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 <code>qs\_avrim.c</code> 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

() Im	nageC	raft IDE	for IC	CAVR (	Standa	ard Ver	sion) [	2K D	emo V	ersion	. For EVALUATI
<u>F</u> ile	<u>E</u> dit	<u>S</u> earch	⊻iew	<u>P</u> roject	<u>R</u> CS	T <u>o</u> ols	<u>T</u> ermin	nal <u>H</u>	elp		
D	Ĩ		3   0	8 B	5,	S.		ŞTOP		X	11001
qs_a	avrim.(										
	77	/ ****	* * * * *	* * * * *	* * * *	* * * * *	* * * * *	* * * *	* * * *	* * * *	* * * * * * * •
	17	/ *	PR	OJECT	:	FPSI	LIC (	Quic	k St	cart	Tutori
	17	/ *	FI	LE:		QS_2	AVRIN	4.C	(Ima	ageC:	raft)
	17	/ *	VEI	RSION	:	1.0					
	17	/ *	DA!	re:		Sept	tembe	er 1	., 20	01	
			FUI	NCTIO	N:						on Exam <sub>l</sub>
							-				allowin
	111	/ *					-				design
		/ * / *					ıgn ⊲ for				loaded 1
		/ *				Kit	IOT	ver	1110	cati	on.
		/ *				тf	- Cot	Jori	fias	. + i o	n Simula
		/ *									M Flag (
		/ *									priate c
	- C C	/ *				-	ladei		1	- [ ]	
	1										
	1	/ ****	* * * * *	* * * * *	* * * *	* * * * *	* * * * *	* * * *	* * * *	• * * *	* * * * * * * *
	#3	includ	de <:	ioat9	4k.h	>					
	1)	def:	ine (	COSIM	1				17	/ Th:	is will
	1	/ Gloł	oal '	Varia	bles						
	ա	nsigne	ed cl	har c	Flag	s = (	);			-77	Bit Fla
	ա	nsigne	ed cl	har l	Flag	s = (	);			-77	Bit Fl 🔽



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.

De <sup>v</sup>	rice 42% full.		د
Doi	re.		ح
23: 4	e:\training\fpslic\lab1\qs_avrim.c	D:\training\fpslic\lab1\qs_avrim.prj	COFF/HEX

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

FPSLIC Control Register Settings	×
FPGA Bitstream  Include FPGA Bitstream  c:\training\fpslic\ab1\qs_fpga.bst Brow	/se FPSLIC CR Setting
AVR Hex       Include AVR Hex File       c:\training\pslic\ab1\qs_avr.hex   Brow	
Data RAM Load Data RAM during configuration	Help Cancel
Data Ram File Format: Atmel Text Format	Bitstream Download
Output Bitstream File: c:\training\pslic\ab1\pslic_qs_fpga.bs Brow	vse Density

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.

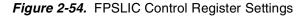
- To add the AVR file, enable Include AVR Hex File by checking the box and use the Browse button to add the file.
- To add the FPGA file, enable Include FPGA Bit Stream by checking the box and use the Browse button to add the file.

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.



FPSLIC Control Register Settings	×
FPSLIC Control Registers	File Settings FPSLIC CR Settin FPGA CR Settings
<ul> <li>✓ (B56) XTAL Pad Bias Resistor Enabled</li> <li>(B57) TOSC Pad Bias Resistor Enabled</li> <li>(B62) Enable Cache Writes to FPGA by AVR</li> <li>✓ (B63) Enable FPGA to Read/Write to Data SRAM</li> <li>External Interrupts (B48 - B51)</li> <li>Ext-INT0 driven by: ○ Port E&lt;4&gt; ○ INTP0 pad</li> <li>Ext-INT1 driven by: ○ Port E&lt;5&gt; ○ INTP1 pad</li> <li>Ext-INT2 driven by: ○ Port E&lt;6&gt; ○ INTP2 pad</li> <li>Ext-INT3 driven by: ○ Port E&lt;7&gt; ○ INTP3 pad</li> <li>UART Pins (B52 - B53)</li> <li>UART0 assigned to ○ Port E&lt;1:0&gt; ○ UART0 pads: UART1 assigned to ○ Port E&lt;3:2&gt; ○ UART1 pads</li> <li>AVR ports I/O drive (B54 - B55)</li> <li>AVR port E I/O ○ 6 mA ○ 20 mA</li> <li>AVR port E I/O ○ 6 mA ○ 20 mA</li> </ul>	OK Help Cancel Bitstream Download Program Configurato Density 1M

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



FPSLIC Control Register Settings	×
<ul> <li>FPGA Configuration Settings</li> <li>✓ (B2) Cascade Disabled</li> <li>✓ (B3) Check Function Disabled</li> </ul>	File Settings FPSLIC CR Settin
(B4) MemoryLockout Enable	FPGA CR Settings
<ul> <li>(B4) MemoryLockout Enable</li> <li>(B6) Global Tristate Enable</li> <li>(B13) Free running Oscillator</li> <li>(B16) GCLK0 disable during Configuration.</li> <li>(B17) GCLK1 disable during Configuration.</li> <li>(B18) GCLK2 disable during Configuration.</li> <li>(B19) GCLK3 disable during Configuration.</li> <li>(B20) GCLK4 disable during Configuration.</li> <li>(B21) GCLK5 disable during Configuration.</li> <li>(B24) FCLK1 disable during Configuration.</li> <li>(B25) FCLK2 disable during Configuration.</li> <li>(B26) FCLK1 disable during Configuration.</li> <li>(B26) FCLK2 disable during Configuration.</li> <li>(B21) Tristate during Configuration.</li> <li>Ocsillator Frequency</li> <li>1 Mhz ● 2Mhz ● 4Mhz ● 8Mhz</li> </ul>	PPGA CR Settings OK Help Cancel Bitstream Download ♥ Program Configurate Density 1M ♥
O 1Mhz O 2Mhz O 4Mhz O 8Mhz	

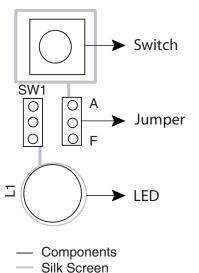
Note: Before you press OK, some hardware connections need to be performed.



- 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.
- 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

device.



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

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.



Calibra	te <u>H</u> elp						
		(Please :	elect Family	first for the (	corresponding Pro	ocedure menu)	
Procedu	ure: 7P:	Partition, progra	m and verify	from an Atm	nel file		
-Files	ile: d:\t	raining\fpslic\lat	o1\fpslic_qs_			<ul> <li>Info- One or more Atmel (.bst) files may be generated for subsequent download sessions. The optional</li> </ul>	
Output		raining\fpslic\lal	o1\out.bst	<b>_</b>	checksum can be obtained from the log output of each programming session. In this version of CPS, HEX values are only appended to the .BST output files which have been partitioned. The output file must be of the form <file.bst.< td=""></file.bst.<>		
Checks	sum:						
Options		<b></b>					
Family:		AT40K		7	COMM Port:	LPT1	7
Device	:	AT17LV010	(A) (1M)	7	Data Rate:	Fast	7
Reset F	Polarity:	Low		~	A2 Bit Level:	Low	7
Console							
Creatin 1 file(s) The Pri 43915 Sum of Setting Device	g a backu copied. ocedure C total bytes device by reset pola data (439	Completed s written to devic ytes: 1643386 arity to Low : Hai	∖training\fpsl e ndling Reset een written t	ic\lab1\out. Polarity for /	iists. bst in file c:\trainii \\T17LV010 devic \ing\fpslic\lab1\o	ce	ıt.cps
				Start F	Procedure R	estore Defaults	View Log File

#### 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.



Step-by-Step Procedure





#### **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

**Disclaimer:** Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time wihout notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

© Atmel Corporation 2003. All rights reserved. Atmel<sup>®</sup> and combinations thereof, aaa<sup>®</sup>, bbb<sup>®</sup> and ccc<sup>®</sup> are the registered trademarks, and aaa<sup>™</sup>, bbb<sup>™</sup> and ccc<sup>™</sup> are the trademarks of Atmel Corporation or its subsidiaries. aaa<sup>®</sup>, bbb<sup>®</sup> and ccc<sup>®</sup> are the registered trademarks, and aaa<sup>™</sup>, bbb<sup>™</sup> and ccc<sup>™</sup> are the trademarks of xxxx Company. Other terms and product names may be the trademarks of others.

