

Dünner Kirchweg 77 32257 Bünde Germany www.trenz-electronic.de

2001-November-6

Introduction

This tutorial was created to give the interested engineer a quick hands on experience on the Xilinx' *WebPACK ISE* software suite.

The tutorial provides only a very brief overview of FPGA design methodology without going into too much detail. After completing this tutorial and the first own design steps, it is highly recommended to proceed with further readings, e.g. the various application notes and online manuals.

While concentrating on the basic steps and methods, it is also the goal of this tutorial to cover the complete development cycle. Beginning with design entry, through simulation, synthesis

Spartan-II Development System

Tutorial: First Steps with WebPACK ISE

and implementation up to the download to physical hardware all covered in a single, consistent project.

To achieve both of these goals, we decided to design a very simple hexadecimal counter which is implemented on our *TE-XC2S* Spartan-II Development System. All steps required to complete this design are covered by this tutorial. There are no hidden steps or imported and half-completed files. While this limits the possible size of the project (without exceeding a certain time frame) we feel that there is still no better way of learning something than from own experience.

Happy developing!

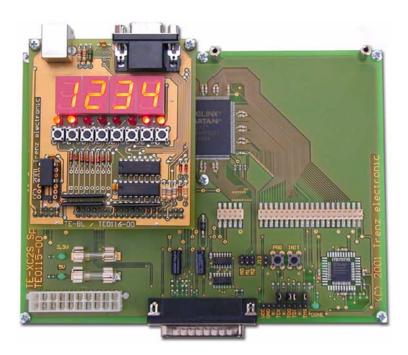


Figure 1: Spartan-II Development System

Download and Installation

In case you did not receive the Xilinx *WebPACK ISE* software on CD-ROM, you may download the latest revision from <u>www.xilinx.com</u>. From the home page follow the links to

Buy Online WebPACK

If you do not have a Xilinx account yet, you will need to register with Xilinx and create and account. To do so, follow the link to

Register for WebPACK ISE

In case you already registered with Xilinx, follow the link to

Single File Download

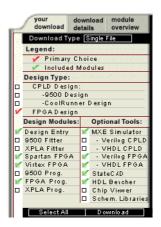
This will lead you to the following screen:



Click Design Configurations, FPGA Design, Select Configuration...

Getting Started	1	
Design Configurations	CPLD Design	
Individual Modules	FPGA Design	Select Configuration
Latest Updates		

... and the following box appears:



Clicking the Download button, pops up the Download Manager window. The software is divided into several modules, which may be downloaded and installed separately.

🖲 Download WebP	ACK - Micros 💶 🔲 🗙
S XILINX	Download Manager
C. AILINA	
Instructions Y	ou have chosen the
following module	s. Please click on the
links below to sa	ve / install the
WebPACK ISE	Software.
Modules Select	ed
Design Entry	
Spartan Fitter	
Virtex Fitter	
FPGA Programm	<u>her</u>
MXE Simulator	
MXE VHDL FP	GA Library
MXE Verilog FP	GA Library
StateCAD	
HDL Bencher	

The following tutorial assumes, that you download and install the following modules:

- Design Entry
- Spartan Fitter
- FPGA Programmer
- MXE Simulator
- MXE VHDL FPGA Library
- StateCAD

In Version 3.3WP8.x these files sum up to approximately 92MB of transfer volume and about 350MB of disk space after installing.

Optionally, you may download the following optional files, which sum up to another 40MB:

- Virtex Fitter
- MXE Verilog FPGA Library
- HDL Bencher

Creating a design

After launching the *WebPACK Project Navigator* for the first time, it comes up with an empty screen, divided into several panes. The *Source* window lists all sources being included in a project. The *Process* window lists the possible actions for a selected source. The *Edit* window is used to display and modify HDL sources. The *Transcript* window shows a log of the most recent shell operations.



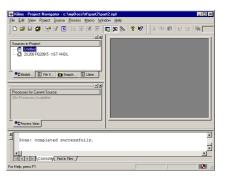
Before going further, we need to create a new project.

Xilinx - Project Navigator - c:\myDocs\tt\part2\part2.npl				
<u>File</u> <u>E</u> dit <u>V</u> iew Project	<u>Source Process Macro Window H</u> elp			
Ne <u>w</u> Project	 5 1 1 1 1 1 1 1 1 1			
Open P <u>r</u> oject				
Open Exa <u>m</u> ple				
Close Projec <u>t</u>				
Sa <u>v</u> e Project As				

A dialog pops up to specify the project's name and location. Furthermore the target family and device as well as the synthesis tool need to be chosen. To target the *TE-XC2S* board, we need to specify *Spartan2* as the device family and *2S200PQ208-5* as the device. Our synthesis tool is *XST VHDL*.

ew Project		
Project name:	Project Loc	ation:
part2	c:\myDocs	<pre>\tt\part2</pre>
Project Doujee Options:		
Project Device Options: Property	Name	Value
Property	Name	
	r Name	Value Spartan2 2S200 PQ208-5

After closing the dialog, an empty project will be created in the project location specified above:



Design entry

Adding existing Source Files

We start our project by adding an existing VHDL module to it. Copy the file from the CD-ROM into your project location, before proceeding. You may use the Windows Explorer to do so.

Xilinx - Project Navigator - c:\myDocs\tt\part2\part2.npl							
<u>F</u> ile <u>E</u> dit <u>V</u> iew	Project	<u>S</u> ource	Process	<u>M</u> acro	<u>₩</u> indow	<u>H</u> elp	
🗏 D 🚅 🗔 🕻	<u>N</u> ew	Source				5 8.	? №
	Add	Add Source In			rt	m 70	
	Add	Add Copy of Source Shift+Inse			t+Insert		
Sources in Project	<u>D</u> ele	Delete Implementation Data					

The file *TE-BL.vhd* comes with your *TE-XC2S* Development System and contains commonly used code to interface with the *Buttons & Lights* Expansion Board.

Add Existing	Sources			? ×
<u>S</u> uchen in:	🔄 part2	•	* 🖻	•
te-bl.vhd				
Datei <u>n</u> ame:	te-bl.vhd			Ö <u>f</u> fnen
Datei <u>t</u> yp:	Sources (*.txt;*.sch;*.vhd;*.vhd;*.vhd	;*.dia;)	J [Abbrechen

A *.vhd* suffix is ambiguous, as VHDL files may contain packages, modules, or test benches. Our file contains a VHDL entity or *module*.

Choose Source Type	×
te-bl.vhd is which source type? The suffix is ambiguous as to type.	
VHDL Module VHDL Package VHDL Test Bench	OK Cancel

The file is added to the source window. The name of the VHDL entity, *tebl*, is displayed, followed by the name of the source file, *te-bl.vhd*, in parentheses.

Sources in Project:
👘 🖻 Untitled
tebl (te-bl.vhd)
■L Module E File V La Snapsh Librar
📑 Module 📋 File V 💼 Snapsh 🗐 Librar

In the process window, we double click *Create Schematic Symbol*, as we want to use the entity tebl in a schematic diagram later on.

Processes for Current Source:				
🖃 🗝 💕 Design Entry Utilities				
😟 🤡 User Constraints				
View VHDL Test Bench Template				
Launch HDL Bencher Tool				
View VHDL Instantiation Template				
Launch ModelSim Simulator				
🗄 🗠 🖸 Synthesize				
Synthesize Synthesize Implement Design Create Programming File				
🗄 🖳 🛱 Create Programming File				
-				
Process View				

Successful generation of the schematic symbol is indicated by a green check mark.

Processes for Current Source:
🖃 🗝 💅 🛛 Design Entry Utilities
🕀 🚽 User Constraints
📄 🛛 View VHDL Test Bench Template
Launch HDL Bencher Tool
📄 🛛 📄 View VHDL Instantiation Template
Launch ModelSim Simulator
🗄 🗠 🖸 Synthesize
🗄 🕀 Implement Design
🗄 🗠 😋 Create Programming File
Process View

Using the HDL Editor

Next, we create a new VHDL file and add it to the project.

Nilinx - Project Navigator - c:\myDocs\tt\part2\part2.npl							
<u>F</u> ile <u>E</u> dit ⊻iew	P <u>r</u> oject ,	<u>S</u> ource	Process	<u>M</u> acro	$\underline{W} indow$	<u>H</u> elp	
	New Source Add Source Add Copy of Source			Inse Shifi	rt t+Insert	a 54	१ №
Sources in Proje	Delete	Delete Implementation Data					

A dialog box appears, to specify the type of source to create, the file name and location. As we want to create another VHDL entity, we chose VHDL Module.

New	×
New User Document Schematic VHDL Maclule VHDL Test Bench VHDL Test Bench VhdL Library State Diagram	File Name: counter Logation: c:\mydocs\tt\part2
< Zurück W	eiter > Abbrechen Hilfe

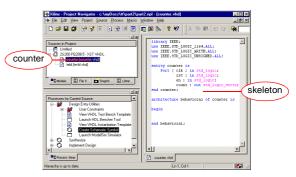
Next, an assistant pops up, querying information required, to create a skeleton for our VHDL entity declaration and architecture body. We need to add port name, and direction for every port. For buses, we need to specify the MSB and LSB as well.

Define VHDL Source				×
_				_
Entity Name cour	nter			
Architecture Name beha	avioral			
Port Name	Direction	MSB	LSB	
clk	in			
rst	in			
en	in			
count	out	15	0 🚖	1
	in			
	in			4
	in			
	in			4
	in			4
	in			-
	in in			-
< <u>Z</u>	urück <u>W</u> eiter >	Abbreck	nen H	lilfe

Before creating the skeleton source, a summary window is displayed.

Project Navigat following specif		te a new sk	eleton source	e with the	
Source Type: V		le			
Source Name: o					
Entity Name: co Architecture Nar		oral			
Port Definitions:					
-	clk	scalar		in	
	rst	scalar		in	
	en count	scalar vector:	15:0	in out	
	Count	vector.	10.0	out	
Source Direct	our c'imi	docs\tt\pa	+2		
Source Direct	ory: c:\my	docs\tt\pa	rt2		
Source Direct	ory: c:\my	docs\tt\pa	ıt2		
Source Direct	tory: c:\my < Zu		rt2 ertig stellen	Abbrechen	Hilfe

And finally, the source file is created and added to the project.



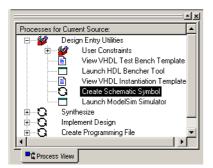
We complete the architecture body by adding the following lines to it:

```
architecture behavioral of counter is
    signal counter: STD_LOGIC_VECTOR(count'range);
signal div: STD_LOGIC_VECTOR(16 downto 0);
    signal div:
    signal divl, en2: STD_LOGIC;
begin
    process(clk, rst)
    begin
         if rst= 'l' then
             div <= (others=> '0');
             div1<= '0';
             en2 <= '0';
         elsif rising edge(clk) then
             div <= div + 1;
             divl<= div(div'left);
             en2 <= div(div'left) and not(div1);</pre>
         end if:
    end process;
    process(clk, rst)
    begin
         if rst= 'l' then
             counter<= x''0000'';
         elsif rising_edge(clk) then
    if en= 'l' and en2= 'l' then
                  counter<= counter + 1;</pre>
             end if:
         end if:
    end process;
    count<= counter;
end behavioral;
```

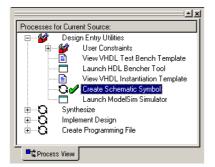
And like before, create a schematic symbol. We need to select our newly created source first...



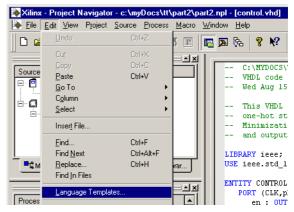
... double click Create Schematic Symbol...



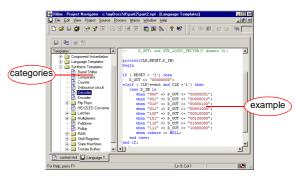
... and a green check mark indicates successful generation of the symbol.



In case we have questions about VHDL, or need some inspiration, there are language templates.



The left pane sorts the templates by category, while the right pane displays the actual sample code.



Check out the language templates and get familiar with the editor! It behaves much like any other programming editor, so you shouldn't have a hard time doing so.

Using the FSM Editor

Now we create a finite-state machine using the FSM Editor.

X	ilinx -	Proje	ct Navig	ator - c:	\myDocs	\tt\part	2\part2.r	npl	
<u>F</u> ile	<u>E</u> dit	⊻iew	Project	<u>S</u> ource	Process	<u>M</u> acro	$\underline{W} indow$	<u>H</u> elp	
Шn	É	E ć	<u>N</u> ew	Source				N R.	🤊 💦
	Add Source		Inse	rt					
			Add	Copy of S	ource	Shif	t+Insert		
So	urces i	in Proie							

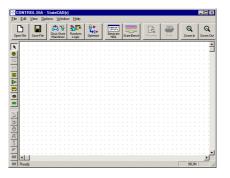
A dialog pops up, to query further information about the new source file. We chose *State Diagram* here.

New User Document Schematic VHDL Module VHDL Package VHDL Package	X
Vhdl Library State Diagram	control
	c:\mydocs\tt\part2
	Add to project
< Zuriick 🔛	eiter >AbbrechenHilfe

Before creating the new diagram, a summary window is displayed.

Project Navigator will create a new skeleton source with the following specifications: Source Type: State Diagram Source Name: control	
Source Name: control	
Course Directory, extendent/Ward2	
Course Directory extendent/Ward?	
Course Directory extendent/Ward2	
Course Directory estandee (Weed)	
Course Directory estander (West)	
Course Directory - claude - (Mar 42	
Source Directory: c:\mydocs\tt\part2	
<zurück abbrechen="" fertig="" hilfe<="" stellen="" td=""><td></td></zurück>	

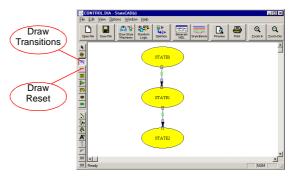
Next, the FSM Editor is launched, displaying an empty sheet.



We use the *Draw States* tool, to add a few states to our FSM.

	CONTROL.DIA - StateCAD(r)	
	Elle Edit View Options Window Help	
	Open File Save File <t< td=""><td>Com In Zoom Dut</td></t<>	Com In Zoom Dut
Draw States	STATE	1
	8 € 10 € 10 € 10 € 10 €	
	N R R R R R	
	Ready	NUM

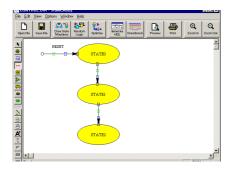
Next, we use the *Draw Transitions* tool, to connect our states.



We chose the *Draw Reset* tool, to add a reset to *State0*, and will be prompted for the type of reset. We select *asynchronous* here.

StateCAL)		×
?	Should this reset be asyncl	hronous (Yes) or synchronous (N	o)?
		<u>N</u> ein	

Now our machine looks like this:



In the following, we edit the state's properties. A double click on *State0* pops up the following dialog. We alter the state name and add an output.

Edit State 🗙
State Name: idle
Outputs: Unassigned outputs are made inactive (i.e. q=0)
en<= '0';
Output Wizard Create counters, muxes, etc. with the wizard.
Justify State Name Justify Output C Left C Center Bight
OK Cancel Help

We do the same for State1...

Edit State 🛛 🗙
State <u>N</u> ame: run
Outputs: Unassigned outputs are made inactive (i.e. q=0)
en<= '1';
_
Output Wizard Create counters, muxes, etc. with the wizard.
Justify State Name C Left © Center C Bight
OK Cancel Help

... and State2.

Edit State 🗙
State Name: stop
Outputs: Unassigned outputs are made inactive (i.e. q=0)
en<= '0';
×
Output Wizard Create counters, muxes, etc. with the wizard.
Justify State Name
C Left C Center C Bight
OK Cancel Help

Next, we need to set up the transitions. Double clicking on the transition from *idle* to *run* pops up the following dialog. We add a condition to this transition.

Edit Condition 🗙
Condition:
pbrun= '1'
<u>·</u>
Outputs: Unassigned outputs are made inactive (i.e. q=0)
T
Output Wizard To edit equations created with the wizard, place the cursor in the equation and click Output
Justify Condition Justify Output
C Left Center C Bight C Left C Center C Bight
Mutually Exclusive Border
Priority 0 + 0K Cancel Help C Asynchronous

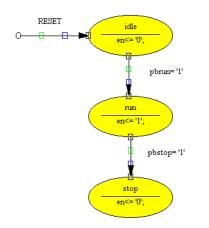
For the transition connecting *run* with *stop* we need to add a similar condition:

Edit Condition 🔀
Condition:
pbstop= '1
-
Outputs: Unassigned outputs are made inactive (i.e. q=0)
Output Wizard To edit equations created with the wizard, place the cursor in the equation and click Output
Justify Condition
C Left ⊙ Center C Bight C Left ⊙ Center C Bight
Mutually Exclusive Border
Priority 0 👗 OK Cancel Help C Asynchronous

Our reset vector is already properly set up, but to be sure, we may open the according dialog as well:

Edit Condition 🛛 🔀
Condition:
RESET
-
Outputs:
<u>^</u>
<u>_</u>
Output Wizard To edit equations created with the wizard, place the cursor in the equation and click Output
Justify Condition Justify Output
C Left Center C Bight C Left Center C Bight
Mutually Exclusive Border Mode C Synchronous
OK Cancel Help © Asynchronous

Now we finished editing our FSM, the final result should look like this:



Next, we compile our graphical representation of the state machine into synthesizable VHDL.

	OL.D	A - StateCAD(r)				
<u>F</u> ile <u>E</u> dit	⊻iew	Options Window Help				
	_	Compile (Generate HDL) F7				
		3 StateBench (Create Test Bench) F5	e StateBench P			
Open File	Save	Optimization Wizard				
		Configuration				
LL		Machine <u>Attribute</u>				

There are several options, how our statemachine can be translated to VHDL. The most important option is, whether the outputs are registered or not. We chose not to register our outputs.

Optimize Outputs for Speed
Select outputs to be made reg
X en
Disable Optimize Speed
Optimize Cancel Help
Output signals which can be made registered are listed here. Select the signals to make into registered (X before the names), or disable this optimization. This feature may be enabled from the Configuration dialog.

The parser detected, that there is no transition leaving the stop state, except for a reset condi-

tion. For our machine this is intentional, so we may safely ignore this warning.

View Warnings	×
ID Number: 1 Warning [1] Problem:	Close
Warning[Z125]: Possible stuck at state 'stop'. ` exit the state is via reset.	The only way to 🔺
Solution:	
Add a transition (with a non-false condition) awa	ay from the state. 📥
]	7
Help: Click here for additional info	rmation.

Finally, the generated source code is displayed for review. You should have a look at the code and check, if this reflects, what you had in mind.

StateCAD HDL Browser - c:\mydocs\tt\part2\CONTROL.vhd
File <u>V</u> iew
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY CONTROL IS
PORT (CLK,pbrun,pbstop,RESET: IN std_logic;
en : OUT std_logic);
END;
ARCHITECTURE BEHAVIOR OF CONTROL IS
 State variables for machine sreg
SIGNAL idle, next_idle, run, next_run, stop, next_stop : std_logi
SIGNAL next_en : std_logic;

We exit the FSM Editor now and find ourselves back in the Project Navigator. Unfortunately our statemachine has not been added to project, so we need to do this manually.

Xilinx - Project Navigator - c:\myDocs\tt\part2\part2.npl											
<u>F</u> ile	<u>E</u> dit	⊻iew	Project	<u>S</u> ource	Process	<u>M</u> acro	<u>W</u> indow	<u>H</u> elp			
			<u>N</u> ew	Source				a R:	.	9	N?
	_		Add	Source		Inse	rt	- V4		•	
			Add	<u>C</u> opy of S	ource	Shif	t+Insert		_	_	_
So	urces i	n Proie									

For convenience, we add both, the VHDL file, and its graphical representation to our project. This allows us to open the diagram later on by simply double clicking the *.dia* file. Unlike the *.vhd* file, the *.dia* file is *not* required for synthesis.

Add Existing	Sources		? ×
<u>S</u> uchen in:	🔁 part2 💌	🗕 🖻 🖻	* III •
Compile	🔊 te-bl.vhd		
sic			
work			
CONTRO			
🖹 counter.vi	na		
		r	
Datei <u>n</u> ame:	"CONTROL.vhd" "CONTROL.DIA"		Ö <u>f</u> fnen
Dateityp:	Sources (*.txt;*.sch;*.vhd;*.vhd;*.vhd;*.vhd;*.dia	1 -	Abbrechen
<u>-</u>	Teeneer (and ready initial initial initial initial		/

Again, we are asked for the type of our VHDL source.

Choose Source Type	×
control.vhd is which source type? The suffix is ambiguous as to type.	
VHDL Module VHDL Package VHDL Test Bench	OK Cancel

To create a schematic symbol for our state machine, we select the source file.

Sources in Project:
🖃 🖻 Untitled
🛄 💼 control.dia
🖻 📲 2S200 PQ208-5 - XST VHDL
- V control (control.vhd)
counter (counter.vhd)
Tel: Module

Double clicking *Create Schematic Symbol* creates the symbol, which is indicated by the green check mark.

Processes for Current Source:
🕞 🗝 🌌 🛛 Design Entry Utilities
😟 🕀 🚽 User Constraints
📄 🛛 📄 View VHDL Test Bench Template
Launch HDL Bencher Tool
📄 🛛 View VHDL Instantiation Template
🖳 🔂 🖌 Create Schematic Symbol
Launch ModelSim Simulator
Emmin Synthesize Emmin Synthesize Implement Design
I⊞CI Synthesize I⊞CQ Implement Design I≣CQ Create Programming File
🗄 💮 Create Programming File
Process View

Creating state machines is a non-trivial task. It is highly recommended to get familiar with the various options of the state machine editor, their effect on the generated VHDL code and their effect on synthesis and implementation.

Using the Schematic Editor

The top layer of our design is created with the schematic editor, instantiating the symbols being defined during previous design entry steps. First, we need to create a new source.

Xilinx - Project Navigator - c:\myDocs\tt\part2\part2.npl							
<u>F</u> ile <u>E</u> dit ⊻iew	Project	<u>S</u> ource	Process	<u>M</u> acro	<u>W</u> indow	<u>H</u> elp	
□ 🚅 🗔 🕻	New Source					N 64	🤋 💦
	Add Source		Insert			• •	
	Add Copy of Source			Shif	t+Insert		
Sources in Proje	Dele	te Implem	entation D	ata			

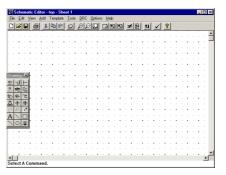
We specify *Schematic* as the source type and enter file name and location.

New	×
User Document Schematic VHDL Module VHDL Package VHDL Test Bench Vhdl Library State Diagram	File Name: top Logation: c:\mydocs\tt\part2
< Zuriick We	eiter >AbbrechenHilfe

A summary is displayed, before actually creating the new source.

	w skeleton sourc	e with the	
ematic			
	will create a ne ations: ematic	ations: ematic	ematic

Then, the schematic editor is launched, displaying an empty sheet.



First, we add our previously created schematic symbols to the sheet.

🔡 S	chem	atic E	ditor -	top - She	et 1				
<u>F</u> ile	<u>E</u> dit	⊻iew	Add	Template	<u>T</u> ools	<u>D</u> RC	Options	<u>H</u> elp	
DI	2 E] 8	<u>v</u>	<u>/</u> ire		F3	ДΙГ		æ∄ ⊉ 1
			<u>N</u>	et Name		F4			
			Expanded Bus Name						
			<u>B</u>	us Tap					
			12	<u>0</u> Marker		Alt+M			
			Symbol			F2			
			<u>l</u> r	nstance Nari	ne				
							_		

The library window pops up. We add the symbols *control, counter* and *tebl* to the sheet.

Schematic Editor - top - Sheet 1	
Elle Edit View Add Template Tools DRC Options	Help
DISH & MAR 2 PPB D	기미미 34월 11 / 9
	Symbol Libraries
L. CONTROL	Libraries/Directories
- COL	(All Symbols)
there a -	(Local Symbols)
2 -	
	Symbols
Drawing X	sonto
2 11-	counter
은 반 52	tebl
20 80 TE	
品中步	
C 4 0	
ALL	
	1
	Symbol Name Filter
<u>.</u>	
Symbol - Click to Place Symbol 'CONTROL'	

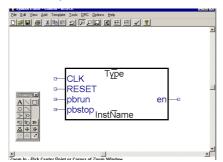
After adding the symbols, our sheet looks like this.

Schematic Editor - top - Sheet 1	
Elle Edit View Add Template Tools DRC Options Help	
	31월 11 V 🔋
Symbol - Click to Place Symbol 'TEBL'	Sumbole Control Counter 125 Symbol Liame Filter

Probably, we need to tidy up our symbols a bit.

22 S	chematic Edito	r - TOP - SI	neet 1
	<u>E</u> dit <u>V</u> iew <u>A</u> o <u>U</u> ndo <u>R</u> edo	ld Template F9 Shift+F9	Iools DRC Options Help 으 우ር 다음을 생활 1
	Cu <u>t</u> Copy Copy Image <u>P</u> aste	Ctrl+X Ctrl+C Ctrl+V	=-cik ^{CO} =-ak =-ak
	<u>D</u> elete Dup <u>l</u> icate	F5 F6	TEBL CLX48-0
	<u>M</u> ove Drag R <u>o</u> tate Mjrror	F7 F8 Ctrl+R Ctrl+E	- LECTON PETOD - DEST(55) SW(75) - CRL - CR
	<u>A</u> ttribute Tabl <u>e</u> Data	•	
	Sym <u>b</u> ol		100_02-40 100_03-44 100_044-40 100_044-40 100_044-40

Clicking on symbol will open the symbol editor window. We adjust *control* to look like this.



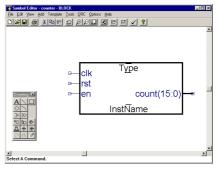
And exit the symbol editor.

📸 Symbol Editor	Symbol Editor - control - BLOCK																			
<u>F</u> ile <u>E</u> dit <u>V</u> iew	Add	Te	mpļ	ate]	ool	s	DF	С	Op	otior	ns	<u>H</u> e	lp						
<u>N</u> ew Open			1	B		2	Γ.	€	۶]	ľ	2	8 7	÷.	E	IJ.	~	1	9
	Ctrl+S		÷	Ì	:	Ì	:	Ì	:			:		:	Ì	:	Ì	:		:
Save <u>A</u> s										·										
Page Set <u>u</u> p Print			ľ	÷	:	÷					÷		•		÷	:	÷	:		
Print Image			ŀ	÷	•	÷	•	·		·		·	•	·	·		·		·	•
Exit			Ľ																	
			Ē		-						_								_	_

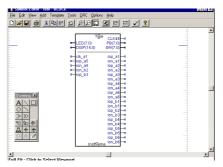
The schematic editor notifies us before updating the symbols. We click *OK TO ALL*.

Schematic Editor							
⚠	References to symbol control are out of date. When you close this schematic, you will be asked if you want to save it, even if you have made no changes.						
	(OK]	OK TO ALL					

Next, we adjust counter to look like this...

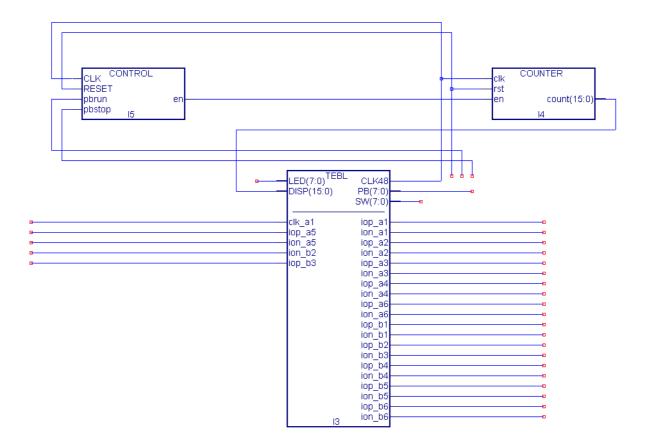


... and tebl to look like this.



Now we interconnect the symbol instances with a few wires.

22 S	chem	atic E	ditor -	TOP - Sh	eet 1				
<u>F</u> ile	<u>E</u> dit	⊻iew	Add	Template	<u>T</u> ools	<u>D</u> RC	<u>Options</u>	<u>H</u> elp	
ВI	ا 😭	. @	<u> </u>	<u>/</u> ire		F3	ם וג		承 题 1
				et Name		F4			
			E	Expanded Bus Name					
		_	B	us Tap					
		- C.U	1.1	0 Marker		Altaki			



ľ

The result should look like above.

Next, we assign names to some signals.

🗱 Schematic Ed	ditor -	TOP - Sh	eet 1		
<u>F</u> ile <u>E</u> dit ⊻iew	Add	Template	<u>T</u> ools	<u>D</u> RC	<u>Options</u> <u>H</u> elp
DIZIRI Z	<u>v</u>	/ire		F3	
	N	et Name		F4	
	E	xpanded Bu	ıs Name		
Clk a1	B	us Tan			

To do so, we edit the name in the status line ...

	13	ion_b6	
■ Net Name - Enter N	Net Na	me =iop_a1	

... and after hitting return, we can assign the name to a wire, by clicking on its red marker.

<u>x B</u>	₱₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽
iop_a1 iop_a1 iop_a2 ion_a2 ion_a2 iop_a3 ion_a3	
iop_a4 ion_a4 iop_a6 ion_a6 iop_b1	

We proceed to do so with some other lines...

-		
	iop_a1	iop_a1
	ion_a1	ion_a1
	iop_a2-	iop_a2
	ion a2	ion_a2
	iop_a3-	iop_a3
	ion_a3-	ion a3
	iop_a4	
	ion a4	ion a4
	iop a6-	
		ion a6
	ion_a6	
	iop_b1-	iop_b1
	ion_b1-	ion_b1
	iop_b2-	iop_b2
	ion_b3-	ion_b3
	iop_b4	iop_b4
	ion_b4	ion_b4
	iop_b5	iop_b5
	ion_b5-	ion b5
	iop_b6-	iop_b6
10	ion_b6	ion_b6
13		

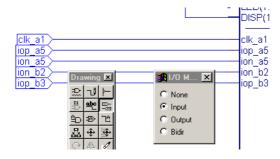
... and add I/O Markers to them.

🔡 S	chem	atic E	ditor -	TOP - Sh	eet 1				
<u>F</u> ile	<u>E</u> dit	⊻iew	Add	Template	<u>T</u> ools	<u>D</u> RC	<u>Options</u>	<u>H</u> elp	
	<u> 2</u> E] <u>8</u>	<u>N</u>	<u>(</u> ire et Name xpanded Bu us Tap	ıs Name	F3 F4			
			1/	<u>0</u> Marker umbol		Alt+M	_a1 _a1 _a2		

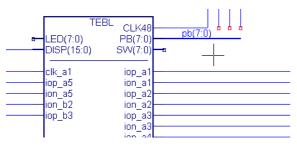
The I/O Marker selector pops up. We select output, and click all of our previously named wires.

-				
2.07				
	1			Law at
	iop_a1			$\frac{10p_a1}{2}$
	ion_a1			- <u>ion_a1</u> >
	iop_a2			-iop_a2>
	ion_a2			-lion_a2 >
	iop_a3			−iop a3>
	ion_a3			ion_a3 >
	iop_a4			iop a4 >
	ion a4			ion a4
	iop a6	 🚮 170 M 🗙		iop a6 >
	ion a6			-ion a6
	iop_b1	C None		liop_b1
	ion b1	C Input		$-\frac{10p}{10n}$ b1
	ion_b1	Output		
		C Bidir		
	ion_b3			- <u>ion_b3</u> >
	iop_b4			- <u>iop_b4</u> >
	ion_b4			- <u>ion_b4</u> >
	iop_b5			- <u>iop_b5</u> >
	ion_b5			- <u>ion_b5</u> >
	iop_b6	 		-liop_b6 >
13	ion_b6			ion b6 >
13	-			

The input pads are created accordingly.



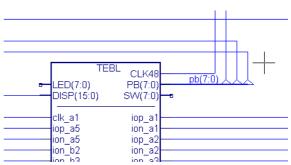
Next, we assign a name to a bus. The thickness of the wire is increased to reflect this.



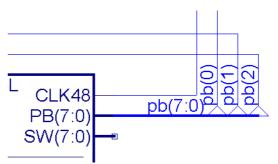
We need to split the bus and create some taps...

🚟 Schematic Ec	litor - TOP - Shee	t 1		
<u>F</u> ile <u>E</u> dit ⊻iew	Add Template <u>T</u>	ools <u>D</u> RC	Options <u>H</u> elp	
061	<u>W</u> ire	F3		1
	<u>N</u> et Name	F4		
	Expanded Bus N	lame		T L
	<u>B</u> us Tap			Ĭ
	I/ <u>0</u> Marker	Alt+M		
	Symbol	F2		

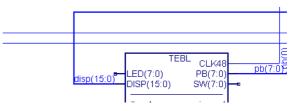
... like this:

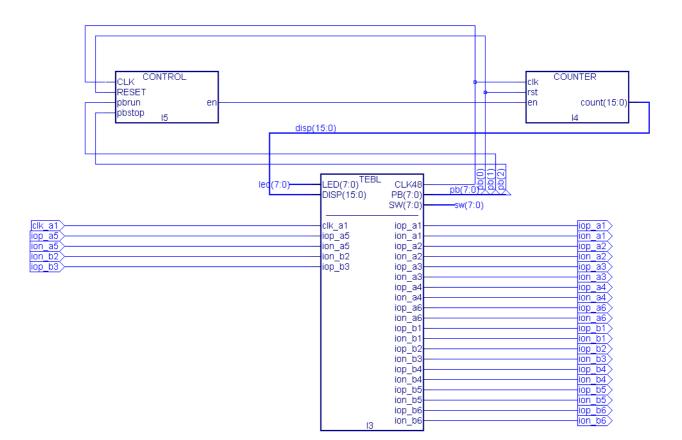


To assign a bus tap to a specific bus signal, we need to assign net names again.



Finally, we create another bus...





... and yield the result above.

Now, we perform a consistency check...

🗱 Schematic Editor - TOP - Sheet 1		
<u>File E</u> dit <u>View A</u> dd Temp <u>l</u> ate <u>T</u> ools	DRC Options Help	
	<u>U</u> uery Utri+Q	HD-
	Consistency Check Electrical Check	
	 C <u>u</u> stomize	

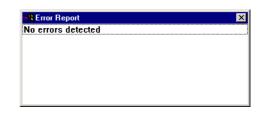
... which goes just fine...

🔒 Error Report	×
No errors detected	

and an electrical check...

🗱 Schematic Editor - TOP - Sheet 1		
<u>File E</u> dit ⊻iew <u>A</u> dd Temp <u>l</u> ate <u>T</u> ools	DRC Options Help	
	<u>H</u> ighlight <u>Q</u> uery Ctrl+Q	
	Consistency Check	
	Electrical Check	
	Curturily.	I

... which is also just fine.



Now we have completed design entry and may exit the schematic editor. The Project Navigator visualizes the hierarchy of our project.

<u>– – – – – – – – – – – – – – – – – – – </u>
Sources in Project:
🖃 🖻 Untitled
📄 📄 control.dia
🖮 🛄 2S200 PQ208-5 - XST VHDL
🗄 🗝 🚯 🚺 top (top.sch)
🔽 control (control.vhd)
🕅 counter (counter.vhd)
🛄 📝 tebl (te-bl.vhd)
📲 Module 🗐 File V 💼 Snapsh 🗐 Librar

Similar to the state machine editor, the schematic editor is only a graphical frontend to automatic VHDL code generation. This may be important in some rare cases, e.g. when signal naming conflicts with VHDL conventions.

Behavioral simulation

Creating a Testbench

Before simulating our design, we need to create a Testbench, which is applying stimuli to our circuit. Create a new source...

Xilin	e - Proje	ect Navig	jator - c:	\myDocs	\tt\par	2\part2.r	npl		
<u>File</u> <u>E</u> c	it <u>V</u> iew	Project	<u>S</u> ource	Process	<u>M</u> acro	<u>W</u> indow	<u>H</u> elp		
	2 🗆 1	e <u>N</u> ev	/ Source					9	N ?
			Source		Inse	ert		•	
		Add	Copy of 9	ource	Shif	t+Insert		_	_

... with the wizard.

New	×
User Document Schematic VHDL Module VHDL Package VHDL Test Bench Vhdl Library State Diagram	File Name: TBtop Logation: c:\mydocs\tt\part2
< Zurück. We	eiter > Abbrechen Hilfe

Review the information...

New Source Information	×
Project Navigator will create a new skeleton source with the following specifications:	
Source Type: VHDL Test Bench Source Name: TBtop	
Source Directory: c:\mydocs\tt\part2	
< <u>∠</u> urück Fertig stellen Abbrechen Hilfe	

... and assign the newly created file to the design's top level.

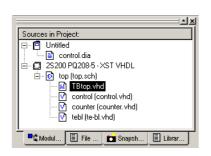
Associate TBtop.vhd with the sou affects.	ce that it
control counter tebl	OK
top	Cancel

Then type in your testbench code. The following code is the simplest form of a testbench, as it is only applying stimuli to the design. More sophisticated testbenches also perform checks on the design's responses.

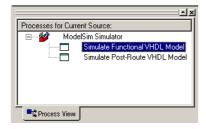
LIBRARY ie	ee;						
USE ieee.s	td logic .	1164. <mark>ALL</mark> ;					
USE ieee.n	umeric st	d.ALL;					
	-						
entity tes	tbench <mark>is</mark>						
end testbe	nch;						
architectu	re bhv of	testbench <mark>is</mark>					
compon	ent top p	ort (
_		In std logic:					
	iop_al:	Out std logic;	ion al:	Out	std logic	;	
	iop a2:	Out std logic;	ion_a2:	Out	std logic		
		Out std logic;					
		Out std logic;					
		In std logic;					
		Out std_logic;					
		Out std logic:					
		Out std logic:					
		In std logic;					
		Out std logic;					
		Out std logic;					
		Out std logic;					
end co	mponent;	ouc bod_rogre,	1011_00.	out	Jou_rogro		
chia co.	mponene,						
signal		seg_f, seg_a,	n nh · 🕅		тс.		
signal		beg_r, beg_a,			IC:= 'O';		
-	sim:				IC:= '1';		
	su2, s			D_106 D_106			
	nt zeroC:	wJ, 304.			IC:= 'O';		
	zero:			D_10G D_10G			
-	2610:		511	D_706	10;		
begin							
zero<=	zeroC;						
		after 15 ms;					
CIK<=	sim and n	ot(clk) after (.	1 us/48).	/2;			
		c					
		after 1.5 ms;					
sw3<=	'0', '1'	after 2 ms, 'O	after	3.5 m	s;		
sw4<=	'0', '1'	after 13 ms, 'O	after.	14.5	ms;		
c_pb<=	(seg_g a	nd sw2) or (seg	_f and s	თ3) <mark>ი</mark>	r (seg_a	and sw	(4);
	op port m						
		k, ion_a3=> seg		a4=>	seg_f,		
		g_g, iop_a5=> c					
io	n_a5=> ze:	ro, ion_b2=> ze:	ro, iop_1	b3=>	zero);		
end bhv;							

Using the Simulator

Select TBtop ...



... and double-click Simulate Functional VHDL Model...



... to launch the ModelSim simulator.

wave - default
Ele Edit Cursor Zoom Format Window
≝∎& X¤® <u>}</u> X 1±± q,q,Q,q, 3 1,1318}34
Andrez Alego and Andrez
0 ps 0 ps
0 ps to 1050 ns //

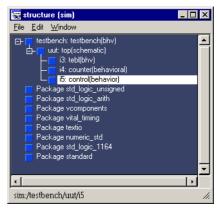
Select all signals...

= wave - default			
<u>File Edit Cursor Zoom For</u>	mat <u>W</u> indow		
🗃 🖬 🎒 🕴 👪 🗎	📐 🕺 🗶 쇠	ା ତ୍ର ପ୍ ପ୍ ପ୍	
/testbench/seg_e /testbench/seg_d /testbench/seg_c /testbench/c_pb /testbench/clk /testbench/sim /testbench/sw2 /testbench/sw3 /testbench/sw4	· · · · · · · · · · · · · · · · · · ·		
/testbench/zero	0		

... and delete them.

w.	ave - default			
<u>F</u> ile	<u>Edit</u> <u>C</u> ursor <u>Z</u> oom	⊨ F <u>o</u> rmat <u>W</u> ind	ow	
≥ [Cut Copy Paste Delete	Cntl-X Cntl-C Cntl-V		
	Select All			

Select the hierarchy level...



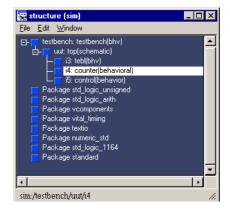
... and the signals to watch ...

📅 signals (sim)	- 🗆 ×
Eile Edit View Window	
en idle next_idle	
irun next_run stop next_stop	0 0 0
next_en	
sim:/testbench/uut/i5	

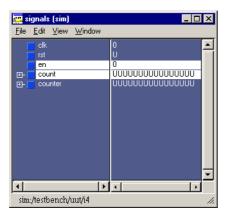
... and drag them into the wave window.

📻 wave - default	
<u>File Edit Cursor Zoom Format Winde</u>	WC
🗃 🖬 🚑 🕺 🛍 🛍 🗎 📐 💥 👘	┶– । �, q, q, q, IF I. [
/testbench/uut/15/pbrun -No Da /testbench/uut/15/pbstop -No Da /testbench/uut/15/reset -No Da	ta-

Repeating these steps...



... for entity counter as well...



... completes our wave window. Click the save button to store your setup for later sessions.

= wave - default			
<u>File Edit Cursor Zoom Format</u>	<u>W</u> indow		
🗃 🖬 🚑 🕴 👗 🖻 🛍 🗎 📐	🔉 🕒	ତ୍ତ୍ତ୍	
272	-No Data- -No Data- -No Data- -No Data- -No Data-		

Now restart the simulator...

ModelSim XE/Starter	5.3d
<u>F</u> ile <u>E</u> dit <u>D</u> esign <u>V</u> iew	<u>Run Macro Options Window H</u> elp
🅸 🚅 🖻 🛍 📑	 Run 100 ps
# ** Warning: NUMERIC_ST # Time: 0 ps Iteration: 0 h	Bun All detected, returning 0!
# ** Warning: NUMERIC_ST	Continue detected, returning 0!
# Time: 0 ps Iteration: 0 In #**Warning: NUMERIC ST	detected returning O
# Time: 0 ps Iteration: 0 It	Just Vi3
# ** Warning: NUMERIC_ST	
# Time: 0 ps Iteration: 0 I # ** Warning: NUMERIC_ST	

... click restart ...

1	Restart 📃 🗆 🗙
	Кеер:
	🔽 List Format
	🔽 Wave Format
	🔽 Breakpoints
	Logged Signals
	Virtual Definitions
	<u>R</u> estart <u>C</u> ancel

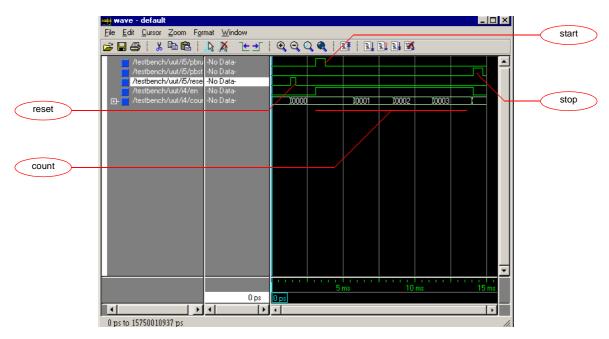
... and run the complete simulation. For our design, the total simulation range is 15ms, which results in a run time of approximately 5 minutes on a *Pentium-III* with 700MHz.

ModelSim XE/Starter 5.3d				
<u>F</u> ile <u>E</u> dit <u>D</u> esign ⊻iew	<u>Run</u> <u>Macro</u> ptions	<u>W</u> indow <u>H</u> elp		
🕸 🚘 🖻 🛍 📑	 Run 100 ps	Ef 🕱 🕁 🕁		
# ** Warning: NUMERIC_S1 # Time: 0 ps Iteration: 0 1	Pus All	detected, returning 0! /i3		
# ** Warning: NUMERIC_S1	<u>C</u> ontinue	detected, returning 0!		
# Time: 0 ps Iteration: 0 1	Run <u>-N</u> ext	/i3		

Zoom to full...

3, 9, 4, 27 21
00111001000

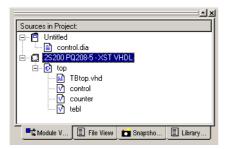
... to see our counter design working. You are invited to play around with the simulator. Browse the design hierarchy, watch signals, set breakpoints in the VHDL code and much more!



Compiling the design

Synthesis

Before synthesizing our design, we doublecheck that the correct target family is selected. When targeting the *TE-XC2S* board, the correct device is *2S200 PQ208-5*.



Next select the design's top level...

Sources in Project:
🖃 🖻 Untitled
📄 💼 control.dia
🗄 📲 2S200 PQ208-5 - XST VHDL
É- 🖸 top
📝 tebl
📕 📲 Module V 🔲 File View 💼 Snapsho 🔲 Library

... and double click on the Synthesize process.



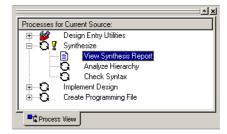
A green check mark indicates success...

Processes	for Current Source:
÷	Design Entry Utilities
⊕−• G ø	Synthesize
⊕ ` O	Implement Design
⊞…G	Create Programming File
<u> </u>	
Proces	s View

... while a yellow exclamation mark indicates warnings.



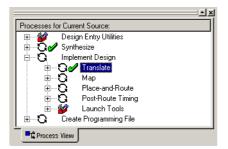
You should review the synthesis report to be sure.



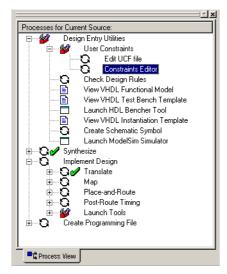
A number of options controls the way your behavioral description is translated into a structural netlist during synthesis. You may safely leave these options to their default values, unless you are working on a high-performance design.

Constraints

Before editing the constraints, you need to translate your design. To do so, double click on *Translate*.



Now launch the Constraints Editor.



Select the *clk_a1* line...

Xilinx Constraints Editor - [Global - top.ngd / top.ucf*]				
jile <u>E</u> dit <u>V</u> iew <u>W</u> indo	w <u>H</u> elp			
Dee X meno ??				
Clock Net Name	Period	Pad to Setup		
Clock Net Name		Pad to Setup		

... and edit the clock period...

Xi	%Xilinx Constraints Editor - [Global - top.ngd / top.ucf*]					
<u>F</u> ile	<u>E</u> dit ⊻iew	<u>W</u> indow	<u>H</u> elp			
D.	<u>D</u> elete	De		? №		
	<u>P</u> eriod					
Г	Pad to <u>S</u>		Pe	riod		Pad to Setup
7777	Clock to	Pad		. 0/		
cl_			: HIGH 50	76		

... to be 48MHz:

Clock Period	×
TIMESPEC Name:	ОК
TS_clk_a1	
Clock Net Name:	Cancel
clk_a1	<u>H</u> elp
Clock Signal Definition	
Specific Time	
Time: 48 Units: MHz 💌	
Start HIGH C Start LOW	
Time HIGH: 50 Units: 🕱 💌	
C Relative to other PERIOD TIMESPEC	
Reference TIMESPEC: TS_clk_a1	
Multiply by O Divide by	
Factor: 1	
Comment:	

Go to the Ports tab ...

Г	//O Configuration Options		- Pad Groups Group Name:	
	Prohibit I/O Locations]	Select Group:	
	Global Ports	Adv	vanced	
For He	lp, press F1			

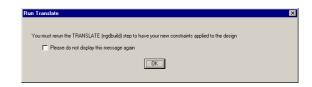
...and enter the following pin locations according to your *TE-XC2S* hardware:

Port Name	Port Direction	Location	Pad to Setup	Clock to Pad
clk_a1	INPUT	p185	N/A	N/A
ion_a1	OUTPUT	p192	N/A	
ion_a2	OUTPUT	p194	N/A	
ion_a3	OUTPUT	p199	N/A	
ion_a4	OUTPUT	p201	N/A	
ion_a6	OUTPUT	p205	N/A	
ion_b1	OUTPUT	p5	N/A	
ion_b3	OUTPUT	p9	N/A	
ion_b4	OUTPUT	p14	N/A	
ion_b5	OUTPUT	p16	N/A	
ion_b6	OUTPUT	p18	N/A	
iop_a1	OUTPUT	p191	N/A	
iop_a2	OUTPUT	p193	N/A	
iop_a3	OUTPUT	p195	N/A	
iop_a4	OUTPUT	p200	N/A	
iop_a5	INPUT	p202		N/A
iop_a6	OUTPUT	p204	N/A	
iop_b1	OUTPUT	p4	N/A	
iop_b2	OUTPUT	p6	N/A	
iop_b4	OUTPUT	p10	N/A	
iop_b5	OUTPUT	p15	N/A	
iop_b6	OUTPUT	p17	N/A	

Now save the result...

🍇 Xilinx Constra	🍇 Xilinx Constraints Editor - [Ports - top.ngd / top.ucf*]					
<u>File</u> <u>E</u> dit <u>V</u> iew	<u>W</u> indow	<u>H</u> elp				
New			Ctrl+N	2		
<u>O</u> pen			Ctrl+O	빋		
<u>C</u> lose				F		
<u>S</u> ave			Ctrl+S	h	Pad to Setup	Clock to Pad
Save <u>A</u> s					N/A	N/A
					NIA	

... click OK ...



... end exit the *Constraints Editor*. Click *Reset* to incorporate your new constraints.

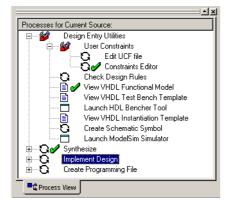
Notice	
Reset the Implement Design process so tha be read?	t your UCF changes will
The User Constraint File (UCF) has changed be possible to reproduce the same implement new UCF. To incorporate the new UCF at this to mark the Implement Design process. Otherwise, choo current implementation results intact and not UCF at this time.	ntation results using the s time, choose RESET of date. Then re-run the ose RETAIN to keep the
Reset	Retain

Most likely, you will need to set up your constraints only once. Later implementation steps will keep the settings you made here.

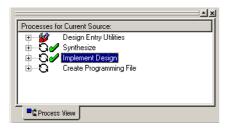
Be warned that wrong or incomplete timing constraints may lead to unexpected behavior, while wrong or missing pin lockings may lead to hardware damage!

Implementation

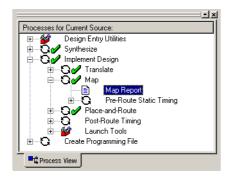
Now you are ready to run implementation. Double click *Implement Design* to do so.



A green check mark indicates successful operation.



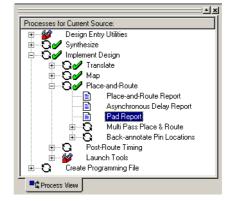
Check the Map Report...



... to learn about your design's resource usage.

Design Summary				
Number of errors:	0			
Number of warnings:	1			
Number of Slices:		88 OI	it of	2,352
Number of Slices cont	taining			
unrelated logic:	-	0 01	it of	88
Number of Slice Flip	Flops:	73 oi	it of	4,704
Number of 4 input LUT	ſs: Ī	90 oi	it of	4,704
Number of bonded IOBs	s:	21 ou	it of	140
Number of GCLKs:		1 01	it of	4
Number of GCLKIOBs:		1 01	it of	4
Total equivalent gate co	ount for d	esign: 1	1,376	
Additional JTAG gate cou	int for IO	Bs: 1,09	56	

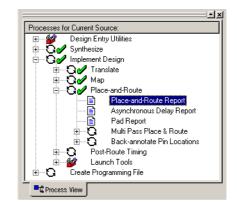
Review the Pad Report...



... to learn about your pin locations.

Pin ! Direction Pin Number clk_a1 INPUT P185 ion_a1 OUTPUT P192 ion_a2 OUTPUT P199 ion_a3 OUTPUT P199 ion_a4 OUTPUT P201 ion_b4 OUTPUT P5 ion_b5 OUTPUT P19 ion_b4 OUTPUT P16 ion_b5 OUTPUT P16 ion_b4 OUTPUT P18 iop_a1 OUTPUT P19 iop_a2 OUTPUT P193 iop_a5 INPUT P202 iop_a4 (VREF) OUTPUT P193 iop_a5 INPUT P202 P195 iop_b1 VREF) OUTPUT P204 P202 iop_b2 VREF) OUTPUT P4 P4 iop_b5 OUTPUT P4 P4 P4 iop_b5 OUTPUT P10 P17	Pinout by Pin Name:			
$\begin{array}{ c c c c c c c c c c c c c c c c c c c$		Pin 1	Direction	
	ion_a1 ion_a2 ion_a3 ion_a4 ion_a6 (VREF) ion_b1 ion_b3 (VREF) ion_b4 ion_b4 ion_b6 iop_a1 iop_a2 iop_a3 iop_a4 (VREF) iop_b5 ion_b1 (VREF) iop_b1 (VREF) iop_b5		OUTPUT OUTPUT OUTPUT OUTPUT OUTPUT OUTPUT OUTPUT OUTPUT OUTPUT OUTPUT OUTPUT OUTPUT OUTPUT INPUT OUTPUT OUTPUT OUTPUT OUTPUT OUTPUT OUTPUT OUTPUT	P192 P194 P199 P201 P205 P5 P9 P14 P193 P194 P195 P200 P202 P204 P4 P6 P10 P15

Check your Place-and-Route Report...



... to learn about the speed of your design

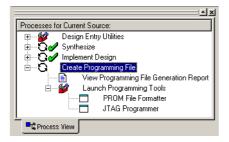
Constraint	Requested	Actual
TS_clk_a1 = PERIOD TIMEGRP 33 nS HIGH 50.000 %	20.833ns	10.532ns

Working with Hardware

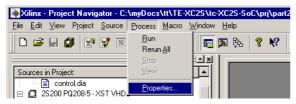
In the following paragraphs, we download our design to the *TE-XC2S* board. Whenever hardware is involved, double check the constraints, especially the pin lockings, to avoid damaging your hardware!

Configuring from PROM

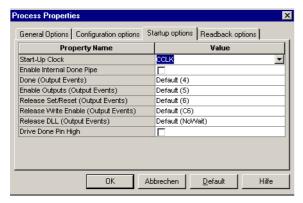
Before creating a programming file, check the options. Select *Create Programming File...*



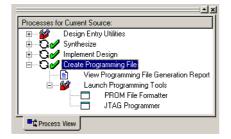
... and open the Properties.



When creating a PROM, make sure the *Start-Up Clock* is set to *CCLK*.



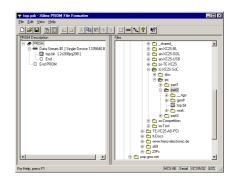
Now double click *Create Programming File* to create the bitstream.



Next, double click PROM File Formatter ...

Processes for Current Source:
🕀 🐨 💕 Design Entry Utilities
🗄 🖸 🖌 Synthesize
🗄 🗠 😋 💋 Implement Design
🗄 🕞 🥩 Create Programming File
View Programming File Generation Report
🖻 📲 🚧 🛛 Launch Programming Tools
PROM File Formatter
JTAG Programmer
Process View

... to open this tool:



Check the PROM Properties...

top.pdr - Xilinx PROM Fi	ile Formatter	
<u>File</u> <u>E</u> dit <u>V</u> iew <u>H</u> elp		
<u>N</u> ew Dpen Create PROM Create PROM <u>A</u> s Save <u>D</u> escription Save Dgscription As	Ctrl+N Ctrl+O Ctrl+S	Image: Second
Split PROM		= • • • •
PROM Properties	Alt+Enter	Ē Ē Ē

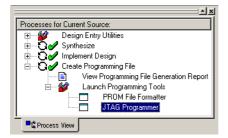
... and make sure the correct device is selected. Your *TE-XC2S* board uses an *XC18V02* Flash PROM.

PROM Properties	×
Format Data Streams Files	
PROM File Format:	Save As Defaults
	PROM Device [size in Bits]:
 Serial 	Automatic Selection
C Byte Wide	×C18V02 [2097152]
<u>P</u> ROM File Single PROM	Percent Used: 63%
C Split PROM	
Fill value to calculate Checksum	(2 digit Hex Value)
0x FF	
0	K Abbrechen <u>H</u> elp

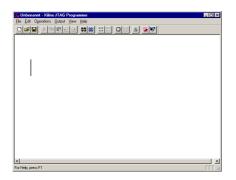
Now create the programming file,...

🔫 top.pdr - Xilinx PROM File	e Formatter	
<u>F</u> ile <u>E</u> dit ⊻iew <u>H</u> elp		
<u>N</u> ew	Ctrl+N	2 3 110 4 9 1
<u>O</u> pen	Ctrl+O	
Create PROM	Ctrl+S	Files
Create PROM <u>A</u> s		¯ͺͺͺͺͺͺͺͺͺͺͺͺͺͺͺͺͺͺͺͺͺͺͺͺͺ

... exit the *PROM File Formatter* and launch the *JTAG Programmer*.



The following screen appears:



Open the Cable Setup...

🎇 Unbenannt - Xilina	JTAG Programmer	
<u>File Edit Operations</u>	<u>O</u> utput ⊻iew <u>H</u> elp	
	Cable <u>A</u> uto Connect	ee ne s s s
	Cable <u>S</u> etup	
	Cable <u>R</u> eset	

... and setup for Parallel Cable III.

Cable Communication Setup	×
Communication Mode	
C Multilinx/USB	Parallel
C Multilinx/Serial	C XChecker
Baud <u>R</u> ate	Port
38400 💌	lpt1
OK Cancel	<u>H</u> elp

Successful cable detection is indicated like this:



Now scan your JTAG chain...

👺 Unbenannt - Xilinx JTAB	Programmer
<u>File</u> <u>E</u> dit Operations <u>O</u> utpu	t <u>V</u> iew <u>H</u> elp
<u>N</u> ew <u>O</u> pen	
Initialize Chain	Ctrl+I
<u>D</u> ebug Chain	Ctrl+B

... for the devices on your TE-XC2S board.

		TDI		x JTAG Programmer			_ 0
wi	TDI	TDI					
	TDI	TDI		162 오 📾 💥	11日日	🍰 🚾 🕅	
XC18U02 XCU200	TD0	TDD	TDI -	xcu200			
1		d					

Double click the *XC18V02* PROM and browse for the *MCS* file created by the *PROM File Formatter* a few steps ago.

×

Chose the XC18V02-VQ44 part.

Se	lect Part Name		×
	Part Name	xc18v02_vq44 🔽	
	OK	Cancel <u>H</u> elp	

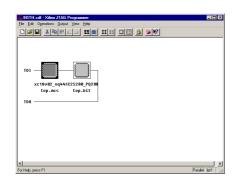
Double click the *XCV200* FPGA and browse for the *BIT* file created during Implementation.

Device Properties	×
Checksum:	Not Available
Signature/Usercode:	Not Yet Read
Device ID:	Not Yet Read
<u>B</u> IT (.bit) or BSDL (.bsd) Fi top.bit	ïle <u>B</u> rowse
ОК	Cancel <u>H</u> elp

The *Spartan-II* family is a close derivative of the *Virtex* family, for this reason the following warning may be safely ignored.

Xilinx JT	AG Programmer 🔀
?	A bit file describing an XC2S200_PQ208 is about to be assigned to a device previously identified as an XCV200. Are you sure you want to do this?
	<u>Ja</u> <u>N</u> ein

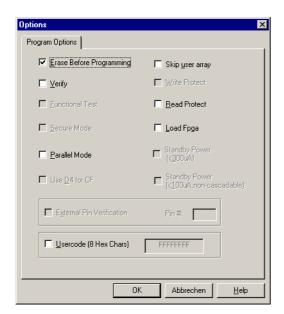
Select the Flash PROM again ...



... and chose to program the device.

👺 BOTH.c	df - Xilinx	JTAG P	rogran	nmer	
<u>F</u> ile <u>E</u> dit	Operations	<u>O</u> utput	⊻iew	<u>H</u> elp	
nieli	<u>P</u> rogram				
	<u>∨</u> erify				
	Ernon				

Setup the programming options...



... and program the device.

Operation Status X
top(Device1): Checking boundary-scan chain integritydone. top(Device1): Programming device.
"top(Device1)": Programming completed successfully.
All operations were completed successfully.
0K ⊻iew Log File

To configure your *TE-XC2S* board from the Flash PROM, setup the jumpers as follows...

Pin	Setting
MO	open
M1	open
M2	open

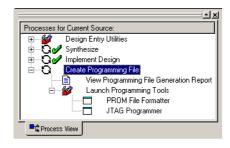
... and hit the *PRG* button to configure the FP-GA. When the *DONE* LED lights up, your board is configured successfully.

Don't forget to save your setup.

👺 BOTH.cdf - Xilinx JTAG	à Programmer
<u>File</u> <u>E</u> dit Operations <u>O</u> utp	ut ⊻iew <u>H</u> elp
<u>N</u> ew Open	Ctrl+N 🗱 📰 🖽 🖽 🔛 🕅
Initialize Chain Debug Chain	Ctrl+I Ctrl+B
Save Save As	Ctrl+S

Configuration via JTAG

Instead of programming the Flash PROM and configuring the FPGA from the PROM, the FPGA may be configured directly via JTAG. To do so, you will need to create a programming file with modified options. Select *Create Programming File*...



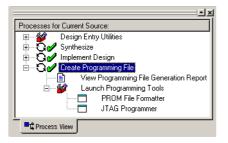
... and open the Properties.

Xilinx - Project Navigator - C:	\myDocs\tt\TE-	XC2S\tc-XC2S-SoC\prj\part2
<u>File Edit View Project Source</u>	Process Macro	<u>W</u> indow <u>H</u> elp
	<u>R</u> un Rerun <u>A</u> ll Stop	
Sources in Project:	⊻iew	
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	Properties	

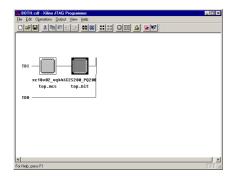
When configuring via JTAG, make sure the *Start-Up Clock* is set to *JTAG*.

General Options Configuration options Property Name	Startup options Readback options
Start-Up Clock	JTAG Clock
Enable Internal Done Pipe	
Done (Output Events)	Default (4)
Enable Outputs (Output Events)	Default (5)
Release Set/Reset (Output Events)	Default (6)
Release Write Enable (Output Events)	Default (C6)
Release DLL (Output Events)	Default (NoWait)
Drive Done Pin High	

Now double click *Create Programming File* to create the bitstream.



Launch the JTAG Programmer, open the previously created project and select the FPGA.



Make sure your *TE-XC2S* board is set up as follows...

Pin	Setting
MO	closed
M1	open
M2	open

Chose to program the device...

BOTH.cdf - Xilinx JTAG Programmer					
<u>F</u> ile <u>E</u> dit	Operations	<u>O</u> utput	⊻iew	<u>H</u> elp	٠
nieli	<u>P</u> rogram	1			
	<u>V</u> erify				
	Erase				

... setup the programming options...

Uptions	×
Program Options	
🔽 Erase Before Programming	🗖 Skip <u>u</u> ser array
C Verify	
Eunctional Test	<u>R</u> ead Protect
🗖 Secure Mode	🗖 Load Fpga
Earallel Mode	□ Standby Power (< <u>3</u> 00uA)
Use D4 for CF	Standby Power (<100uA,non-cascadable)
Egternal Pin Verification	Pin #:
Usercode (8 Hex. Chars)	FFFFFFF
ОК	Abbrechen <u>H</u> elp

... and program the device.

peration Status			×
Instance 'top(Dev Instance 'top(Dev Verification compl Boundary-scan ch 'top(Device2)': Ch 'top(Device2)': Re 'top(Device2)': Pro	ositions in boundary-scan ice 1) at position '1'verifie ice2) at position '2'verifie ted. uain validated successfully ecking boundary-scan cha ading bit-stream filedone ogramming device	ed. ed. in integritydone. done.	
	All operations were o	completed successfully.	
	ОК	⊻iew Log File	

Like before, the *DONE* LED lights up, once the FPGA configured successfully. Unlike configuring from PROM, the configuration is lost, when pressing the *PRG* button.

Wrap Up

Congratulations! Now that you completed the tutorial, you gained a basic understanding of FPGA development and the tools involved. Now it's up to you, to expand your knowledge using the various internet resources and further readings.

The *References* section lists those documents, which describe the soft- and hardware used in this tutorial in full detail. While these documents provide complete and accurate information, most of them are not too instructive.

An introduction to FPGA technology is given here:

• Spartan-II Development System Tutorial: Introduction to FPGA Technology Trenz Electronic, November 5, 2001

The following application notes provide complete sample projects and helpful hints:

- Spartan-II Development System Application Note: Buttons & Lights Trenz Electronic, September 20, 2001
- Spartan-II Development System Application Note: Game Of Life Trenz Electronic, September 20, 2001

To improve your knowledge on VHDL, we recommend the following resources:

- The VHDL Cookbook Peter J. Ashenden ftp://ftp.cs.adelaide.edu.au/pub/ VHDL-Cookbook/
- The Designer's Guide to VHDL, 2nd Edition Peter J. Ashenden ISBN: 1558606742 Morgan Kaufmann, January, 2001
- Schaltungsdesign mit VHDL Lehmann, Wunder, Selz ISBN 3-7723-6163-3 Franzis'-Verlag, 1994 http://www-itiv.etec.uni-karlsruhe.de/ FORSCHUNG/VEROEFFENTLICHUNGEN/ Iws94/lws94.html
- VHDL FAQ http://www.vhdl.org/vi/comp.lang.vhdl/

To find a helping hand to answer your questions, you should join the following usenet discussion groups:

- comp.arch.fpga
- comp.lang.vhdl

References

- Foundation Series ISE 3.1i User Guide Xilinx, Inc., October 17, 2000
- Xilinx Synthesis Technology (XST) User Guide Xilinx, Inc., October 17, 2000
- Development System Reference Guide Xilinx, Inc., October 11, 2000
- Constraints Editor Guide
 Xilinx, Inc., October 11, 2000
- PROM File Formatter Guide Xilinx, Inc., October 11, 2000
- JTAG Programmer Guide Xilinx, Inc., October 11, 2000
- Spartan-II Development System
 Product Specification
 Trenz Electronic, September 12, 2001
- Spartan-II 2.5V FPGA Family
 Product Specification
 Xilinx, Inc., October 31, 2000
- XC18V00 Series of In-System Programmable Configuration PROMs Product Specification Xilinx, Inc., April 4, 2000
- Spartan-II FPGA Family Configuration and Readback XAPP176 Xilinx, Inc., December 4, 1999
- Configuration and Readback of Virtex FPGAs Using (JTAG) Boundary-Scan XAPP139 Xilinx, Inc., February 18, 2000

Copyright

© 2001 Trenz Electronic.

All rights reserved. Reproduction in whole or in part is prohibited without the written consent of the copyright owner.

Revision History

Version	Date	Who	Description
1.0	2001sep25	FB	Created, WP3.3
1.1	2001nov06	FB	Minor additions

Table 1: Revision History