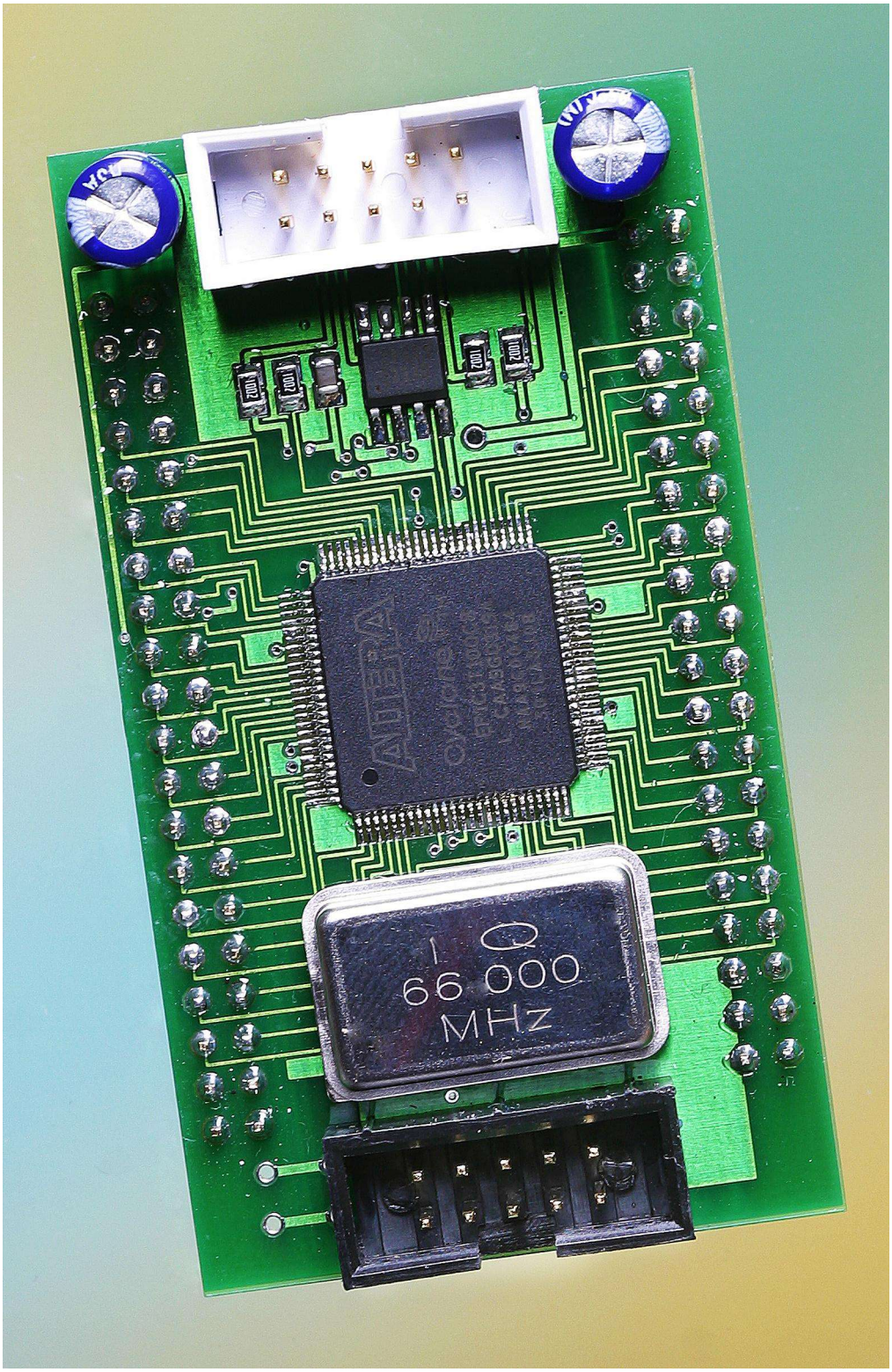

Sphinx-C100 User Documentation

Inventronik GmbH
Finkenstraße 48, D-70199 Stuttgart, http://www.inventronik.de

This page is intentionally left blank.

Table of Contents

Table of Figures

Tables

History

Issue	Date	Reason For Changes
01	20040119	Initial issue
02	20040413	Minor bugfixes

References

[1]

Introduction

Congratulations- with this product you have acquired a very modern FPGA module designed to shorten your hardware development time, and increase your efficiency.

Please read this user documentation completely and carefully before you use this product. For more valuable tips, as well as the latest documentation, see our homepage at http://www.inventronik.de .

Safety Restrictions

Inventronik is proud to supply you with a high-quality device, nevertheless we cannot guarantee that this module works under all possible conditions. Do not use this product in applications, where damage in the module could lead to direct danger for humans ,i.e. medical systems, protection devices etc. This product is conceived exclusively for use with a 5Volt power-supply. Higher voltages can lead to malfunction and/or the total loss of the module. **Please only use certified 5Volt power supplies.**

Shipping List

This package contains the following parts:

- 1 FPGA module Sphinx-C100
- 1 product CD with data sheets, documentation and software

Hardware User Documentation

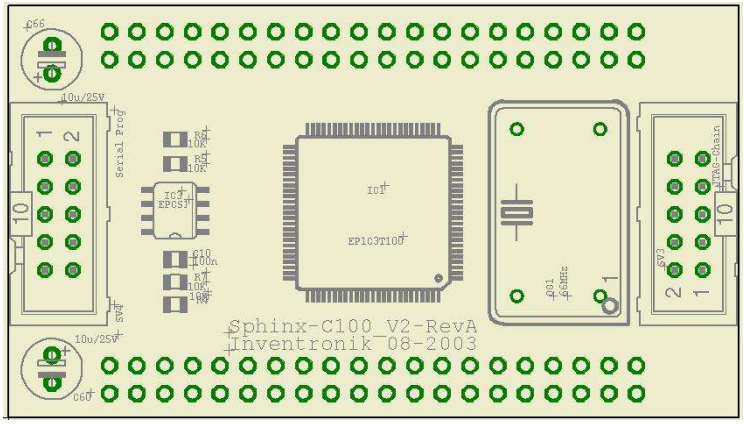

Figure 1: Top View of the Module Sphinx-C100

Description of the Module

The module Sphinx-C100 is conceptually designed under a „ready for use“ concept. This gives your system developers the possibility for a quick entry into the thematics of abstract modeling and digital logic. For operation you need the Sphinx-C100, a power supply, a download cable (Altera Byte Blaster-II) and the free web edition software QUARTUS II from Altera.

The modules are based on the **F**ield **P**rogrammable **G**ate **A**rray (FPGA) „Cyclone EP1C3T100“ from Altera. Although the size of the module is just 1.5“ x 2.6“, it is equipped with all the necessary components to operate the FPGA. The voltage supplies for the I/O units and for the FPGA core are „on board“, for that reason the Sphinx-C100 only requires a single 5V supply.

For Boundary Scan Tests a JTAG interface is provided. The configuration device on board is programmable via a second serial interface. A clock generator near the FPGA provides a stable (even high) frequency. The 65 digital inputs and/or outputs of the FPGA are connected to headers. The EP1C3T100 has 2910 logical elements, 59904 byte RAM, one Phase Locked Loop (PLL) and supports amongst others DDR.

Preparing Module Configuration

You don't need to change anything at the Sphinx-C100 before you start. Please refer for the appropriate pins in Table 3. The current consumption of the module depends mostly on the peripheral components connected to the port pins of the FPGA. The circuit designer must take care to keep the current consumption within the limits. The power supplies shipped by Inventronik deliver a maximum current of 1A at 5V. The modules are protected against wrong polarity using a protection diode in the supply line of the Sphinx-C100.

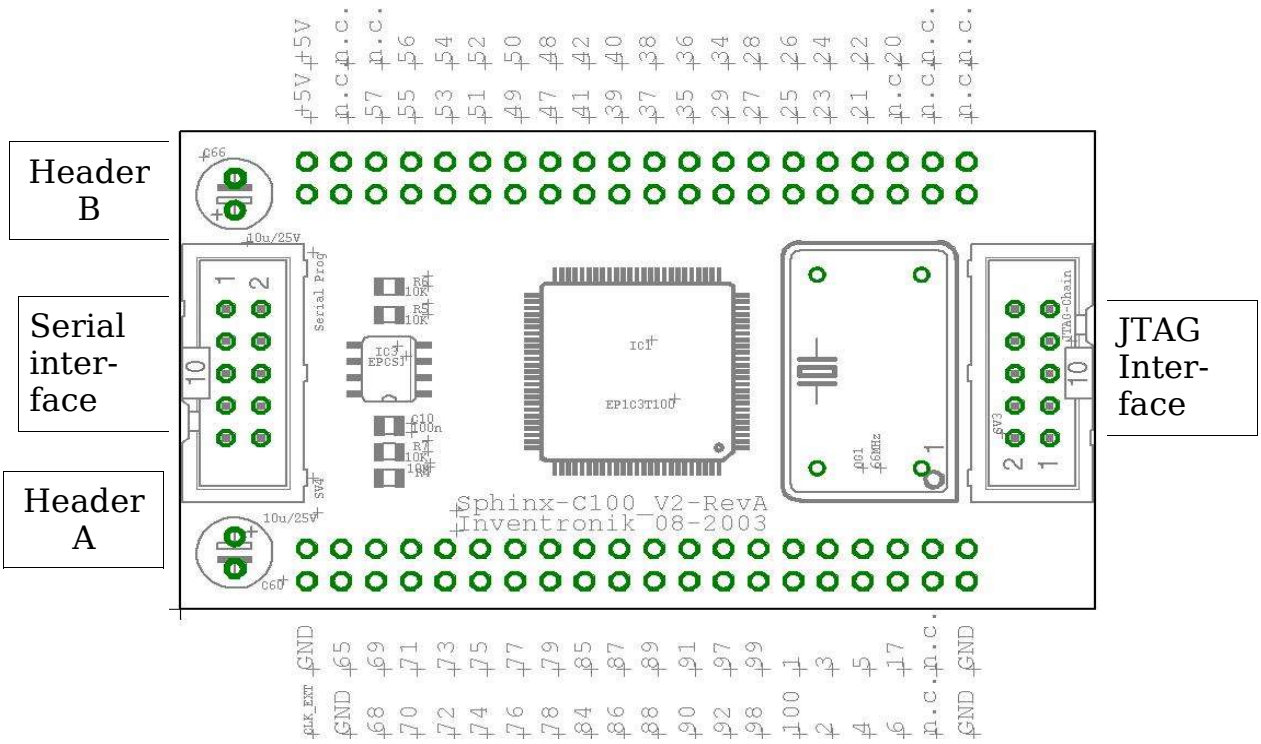

Figure 2: Sphinx-C100 connection diagram

The Sphinx-C100 module contains a crystal clock oscillator. It's frequency has a range between 8 MHz and 100 MHz, depending on the option you ordered (default is 66 MHz). Nevertheless, if you want to use another frequency (as the on-board oscillator), you can connect an external oscillator with the frequency of your choice to pin 2 (CLK_EXT). Switching from internal to external clock source is done by correcting the pin assignment of your digital design in the software development suite (QUARTUS II). CLK_EXT is pin 66 of the FPGA, whereas the on-board oscillator is connected to it's pin 10.

Sphinx-C100 Connections

After connecting the power supply to the Sphinx C-100, a connection between your PC and the Sphinx-C100 is also required. This connection must be established using the Byte Blaster II cable from Altera. You have two options for programming the target. On the one hand there is the JTAG- (**J**oint **T**est **A**ction **G**roup) interface or you can use a serial (ISP) interface. In the following text both variants will be compared to each other. For more information please refer to the Cyclone Device Handbook from Altera, which you can find on the product CD. You can also download it from the Altera web site.

The Cyclone has no way to store your logic design in a non-volatile fashion. It is equipped with a volatile SRAM and must be configured after every system start from a FLASH ROM based configuration device (non-volatile). This configuration is done automatically at system start. The configuration device is connected to the FPGA and has a serial programming interface connected to one header of the Sphinx-C100. Designing digital logic on your PC ends in a configuration file which can be downloaded to the configuration device using this serial interface.

Another way to execute the Cyclone without storing the digital logic information on the configuration device is to download the configuration file directly to the SRAM of the FPGA. The JTAG interface on the Sphinx-C100 can be used for this. Downloading the configuration file using the JTAG interface works fine until the voltage supply of the Sphinx-C100 is switched off. The advantage to using this programming method lies in the short turnaround time during debugging sessions or while trying new variants of your digital designs.

Both connectors are designed as 10-pin headers providing a direct connection to the Byte Blaster II download cable. Connect this to one or the other programming interface depending on your choice and using JTAG or serial programming method (see also Figure 2). The Byte Blaster II is on the PC-side connected to the line printer interface.

JTAG Interface

The JTAG interface was originally designed for testing purposes. You can do it with a so called Boundary Scan Test (see Altera application notes and concerning literature). The download of the configuration file using a Byte Blaster II cable is supported by the Quartus II software (see later). The pin out of the JTAG header on the Sphinx-C100 module is listed in the following table:

Connector-Pin	Function	Remark	Connector-Pin	Function	Remark
1	TCK	Clock	6	VCCIO	3,3V
2	GND	0V	7	n.c.	not used
3	TDO	Out	8	n.c.	not used
4	VCCIO	3,3V	9	TDI	in
5	TMS	Mode	10	GND	0V

Table 1: Pin out of the JTAG Programming Header

Serial Programming Interface

In comparison to the JTAG- the serial interface is used exclusively to program the configuration device. The pin out of the interface header is shown in Table 2.

Connector-Pin	Function	Remark	Connector-Pin	Function	Remark
1	DCLK	Clock	6	NCE	
2	GND	0V	7	DATA0	Data
3	CONF_DONE		8	/CSO	Chipsel.
4	VCCIO	3,3V	9	ASDI	
5	NCONFIG		10	GND	0V

Table 2: Pin out of the serial Programming Header

Pin Row Assignment

The Modul Sphinx-C100 is equipped with two 40-pin headers, one on each side of the module. All the I/O pins and dedicated inputs of the Cyclone are connected to these headers. The system designer, using a Sphinx-C100, has the possibility to use up to 65 free configurable inputs and/or outputs. Some pins of the Cyclone have alternative functions and may not be used depending on the module configuration. At this point we refer to the Cyclone Device Handbook. The orientation of the pin row headers is shown in Figure 2.

Pin-Number	Funktion / alternative Funktion	Alternative Funktion	Comment
1 (Header A)	GND		0V
2 (Header A)	CLK_EXT		External Clock
3 (Header A)	I/O: Cyclone-Pin 65		General Purpose I/O
4 (Header A)	GND		0V
5 (Header A)	I/O: Cyclone-Pin 69		General Purpose I/O
6 (Header A)	I/O: Cyclone-Pin 68	VREF1B3	I/O-Cell Reference
7 (Header A)	I/O: Cyclone-Pin 71		General Purpose I/O
8 (Header A)	I/O: Cyclone-Pin 70		General Purpose I/O
9 (Header A)	I/O: Cyclone-Pin 73	VREF0B3	I/O-Call Reference
10 (Header A)	I/O: Cyclone-Pin 72	DPCLK4	Dual Purpose CLK
11 (Header A)	I/O: Cyclone-Pin 75		General Purpose I/O
12 (Header A)	I/O: Cyclone-Pin 74		General Purpose I/O
13 (Header A)	I/O: Cyclone-Pin 77		General Purpose I/O
14 (Header A)	I/O: Cyclone-Pin 76		General Purpose I/O
15 (Header A)	I/O: Cyclone-Pin 79		General Purpose I/O
16 (Header A)	I/O: Cyclone-Pin 78		General Purpose I/O
17 (Header A)	I/O: Cyclone-Pin 85	VREF0B2	I/O-Cell Reference
18 (Header A)	I/O: Cyclone-Pin 84		General Purpose I/O
19 (Header A)	I/O: Cyclone-Pin 87		General Purpose I/O
20 (Header A)	I/O: Cyclone-Pin 86		General Purpose I/O
21 (Header A)	I/O: Cyclone-Pin 89		General Purpose I/O
22 (Header A)	I/O: Cyclone-Pin 88	VREF1B2	I/O-Cell Reference
23 (Header A)	I/O: Cyclone-Pin 91	VREF2B2	I/O-Cell Reference
24 (Header A)	I/O: Cyclone-Pin 90		General Purpose I/O
25 (Header A)	I/O: Cyclone-Pin 97		General Purpose I/O
26 (Header A)	I/O: Cyclone-Pin 92	DPCLK2	Dual Purpose CLK
27 (Header A)	I/O: Cyclone-Pin 99		General Purpose I/O
28 (Header A)	I/O: Cyclone-Pin 98		General Purpose I/O
29 (Header A)	I/O: Cyclone-Pin 1		General Purpose I/O
30 (Header A)	I/O: Cyclone-Pin 100		General Purpose I/O
31 (Header A)	I/O: Cyclone-Pin 3		General Purpose I/O
32 (Header A)	I/O: Cyclone-Pin 2		General Purpose I/O
33 (Header A)	I/O: Cyclone-Pin 5	VREF1B1	I/O-Cell Reference
34 (Header A)	I/O: Cyclone-Pin 4	VREF0B1	I/O-Cell Reference

Pin-Number	Funktion / alternative Funktion	Alternative Funktion	Comment
35 (Header A)	I/O: Cyclone-Pin 17	ASDO	Serial data output
36 (Header A)	/CSO		Chip select output
37 (Header A)	not connected		not used
38 (Header A)	not connected		not used
39 (Header A)	GND		0V
40 (Header A)	GND		0V
1 (Header B)	+5V		Supply
2 (Header B)	+5V		Supply
3 (Header B)	not connected		not used
4 (Header B)	not connected		not used
5 (Header B)	not connected		not used
6 (Header B)	I/O: Cyclone-Pin 57	VREF2B3	I/O-Cell Reference
7 (Header B)	I/O: Cyclone-Pin 56		General Purpose I/O
8 (Header B)	I/O: Cyclone-Pin 55		General Purpose I/O
9 (Header B)	I/O: Cyclone-Pin 54		General Purpose I/O
10 (Header B)	I/O: Cyclone-Pin 53		General Purpose I/O
11 (Header B)	I/O: Cyclone-Pin 52		General Purpose I/O
12 (Header B)	I/O: Cyclone-Pin 51		General Purpose I/O
13 (Header B)	I/O: Cyclone-Pin 50		General Purpose I/O
14 (Header B)	I/O: Cyclone-Pin 49		General Purpose I/O
15 (Header B)	I/O: Cyclone-Pin 48		General Purpose I/O
16 (Header B)	I/O: Cyclone-Pin 47		General Purpose I/O
17 (Header B)	I/O: Cyclone-Pin 42	DPCLK6	Dual Purpose CLK
18 (Header B)	I/O: Cyclone-Pin 41	VREF0B4	I/O-Cell Reference
19 (Header B)	I/O: Cyclone-Pin 40		General Purpose I/O
20 (Header B)	I/O: Cyclone-Pin 39		General Purpose I/O
21 (Header B)	I/O: Cyclone-Pin 38	VREF1B4	I/O-Cell Reference
22 (Header B)	I/O: Cyclone-Pin 37		General Purpose I/O
23 (Header B)	I/O: Cyclone-Pin 36		General Purpose I/O
24 (Header B)	I/O: Cyclone-Pin 35	VREF2B4	I/O-Cell Reference
25 (Header B)	I/O: Cyclone-Pin 34	DPCLK7	Dual Purpose CLK
26 (Header B)	I/O: Cyclone-Pin 29		General Purpose I/O
27 (Header B)	I/O: Cyclone-Pin 28		General Purpose I/O
28 (Header B)	I/O: Cyclone-Pin 27		General Purpose I/O

Pin-Number	Funktion / alternative Funktion	Alternative Funktion	Comment
29 (Header B)	I/O: Cyclone-Pin 26		General Purpose I/O
30 (Header B)	I/O: Cyclone-Pin 25		General Purpose I/O
31 (Header B)	I/O: Cyclone-Pin 24		General Purpose I/O
32 (Header B)	I/O: Cyclone-Pin 23		General Purpose I/O
33 (Header B)	I/O: Cyclone-Pin 22		General Purpose I/O
34 (Header B)	I/O: Cyclone-Pin 21		General Purpose I/O
35 (Header B)	I/O: Cyclone-Pin 20	VREF2B1	I/O-Cell Reference
36 (Header B)	not connected		not used
37 (Header B)	not connected		not used
38 (Header B)	not connected		not used
39 (Header B)	not connected		not used
40 (Header B)	not connected		not used

Table 3: Sphinx-C100 Pin Row Assignment

Getting started with the Sphinx-C100

Please operate the Sphinx-C100 on an electrically insulated surface to avoid short circuits and damage to the module.

Note: the maximum current consumption depends primary on the circuit connected to the FPGA's output pins (i.e. the sum of the current of each output pin). Do not operate the module with too much current, as this will lead to high temperatures from lost power and may cause a malfunction or total loss of the Sphinx-C100.

Mechanical Dimensions of the Sphinx-C100

The following illustration shows the most important dimensions of the Sphinx-C100. **All dimensions are in mm.**

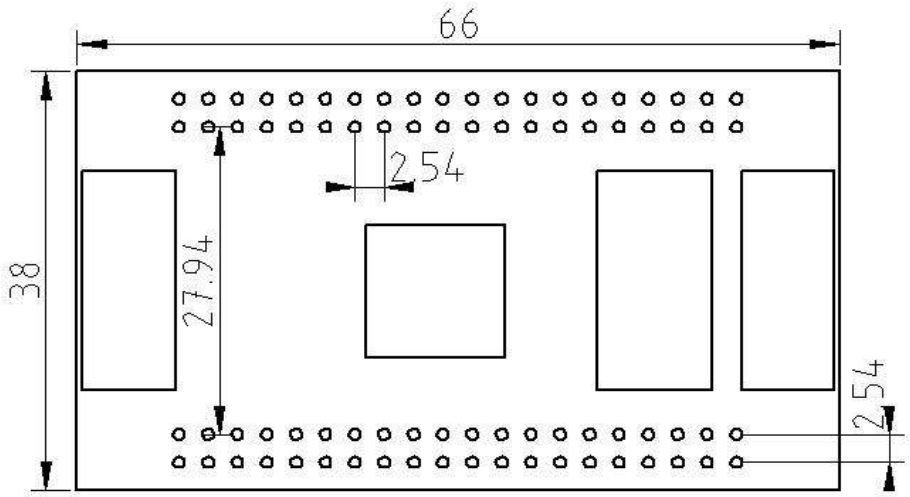

Figure 3: Dimensions of Sphinx-C100

Technical Specifications for the Sphinx-C100

FPGA:
- Type: Cyclone EP1C3T100
- I/O Pins: 65
- Logical elements: 2910
- RAM: 59904 Bytes
- Further features : PLL, DDR (see data-sheet)

Oscillator:
- Frequency 66MHz,
- Optional frequencies: 1MHz up to 100MHz (please ask)

Configuration device:
- Flash: about 100,000 programming cycles

Program interfaces:
- JTAG (Joint Test Action Group), also for boundary scan tests
- Serial (active serial programming)

Electrical Supply:
- Supply voltage: 5V ±10%, regulated DC voltage
- Current consumption: about 50mA, without current through port pins
- Maximum allowed current consumption: 1A

Others:
- Operating temperature range: 0°C to +70°C
- Board size: 66mm x 38mm
- Height: 20mm
- Weight: ca. 35g

PC system requirements using the Quartus II design software are a built in network card and the following (original excerpt from the Altera website):

> To download and run the Quartus II Web Edition software, your computer must meet the following requirements:
>
> - Pentium II PC running at 400 MHz or faster
> - Microsoft Windows NT version 4.0 (Service Pack 3 or higher), Windows 2000, or Windows XP
> - Microsoft Windows-compatible graphics card and SVGA monitor
> - Microsoft Windows-compatible 2- or 3-button mouse
> - One or more of the following ports:
>   - USB port for use with the USB Blaster or MasterBlaster communications cable (Windows 2000 & Windows XP only)
>   - Parallel port for use with the ByteBlaster II, ByteBlasterMV, or ByteBlaster parallel-port download cables
>   - Serial port for use with the MasterBlaster communications cable

Further Information

All data sheets for the Sphinx-C100 components can be found on the product CD. For a better understanding of the FPGAs, please read the data-sheets and application notes. You can also find more information on our Inventronik web site: http://www.inventronik.de and at Altera: http://www.altera.com. If you have specific questions about the Sphinx-C100 not covered in this documentation, please visit our forum at http://www.inventronik.de. We are happy to assist and help. You are welcome to use the forum to discuss Sphinx-C100 applications with other users too.

Sphinx C-100 Software

Overview

To accomplish your digital design projects you can use the software suite QUARTUS II from Altera, which provides support for graphical and text based design entries. On the product CD you'll find some „Getting Started“ examples which will help you to learn about Quartus II. In the directory „VHDL-QUARTUS-Projects“ you will find precompiled, loadable Quartus projects and in the directory „VHDL-Samples“ pure VHDL source codes for those who prefer another development software as Quartus II. The software QUARTUS II Web Edition (Version 3.0) is included on the product CD. Furthermore, you can download the Web Edition of Quartus II in the most **actual?** version from the Altera web site. The link to this site is:

http://www.altera.com/

The following description refers to the use of Quartus II. It shall give a painless entry into the usage of Quartus II and VHDL. The provided examples are all text based VHDL designs. The abstract modeling of electronic digital circuits is a standard method with the IEEE1076.1 standardized programming language VHDL.

Installation of Quartus and Licensing

The Quartus design software is easy to install on your PC. After you have done this you can find a program group „Altera“ and there the software „Quartus II 3 Web Edition Full“. Execute the program now. After a short while you will see a message window with the message „Evaluation Expired“ as shown in Figure 4. After selection of „Run the Quartus II software“ you can use the program anyway. But in the not licensed mode you cannot compile or create designs. To enable these features you have to license the software first. Go to the Altera web site and get one - it is available for free and valid for three months. After this period you can download another free license file for another three months. The link to the license page is:

http://www.altera.com/cgi-bin/authcode91.pl?where_am_i=product&product=q2_web_nic_id

Also read the installation guide on the Altera web site. You need a network card (NIC) and the associated MAC (**M**edia **A**ccess **C**ontrol) number to get the license file from Altera. This number is stored in your network card, identifies your PC in the network and is unique world-wide. Altera uses the MAC Number as reference for the creation of your license file. On the web site given in the link above there is advice on how to get your individual MAC number out of your PC.
After filling out the form you will receive the license file via e-mail. After this, you should copy the file only to the directory where Quartus is located (i.e. C:\program files\quartus). If the message shown in Figure 4 appears again please restart Quartus II and chose the option „Run the Quartus II software“. After program invocation select the license file in the menu „Tools-License

Setup“.

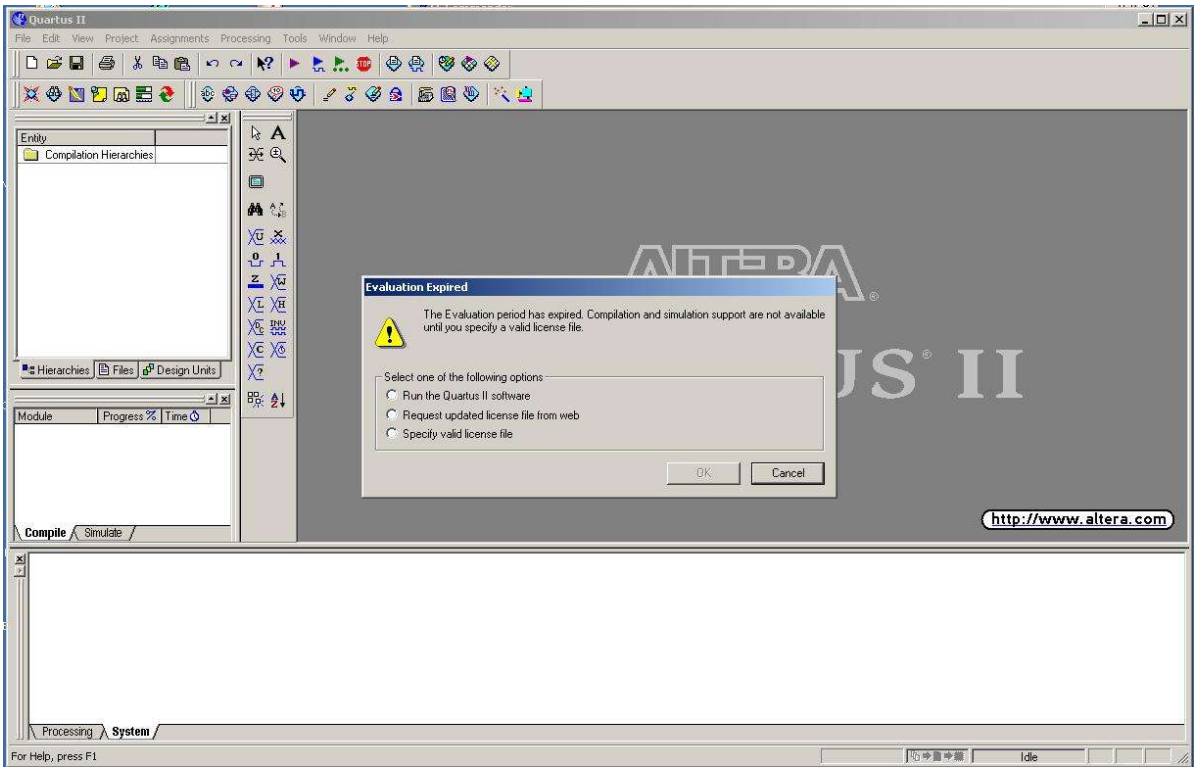

Figure 4: Quartus: first Start, not licensed

It is not mandatory to store the license file in the Quartus directory you are free to select any directory on your computer. Figure 5 shows an example of the license manager of Quartus II. If you did chose the correct license file, Quartus II should come up at the next program start without the message screen shown in Figure 4.

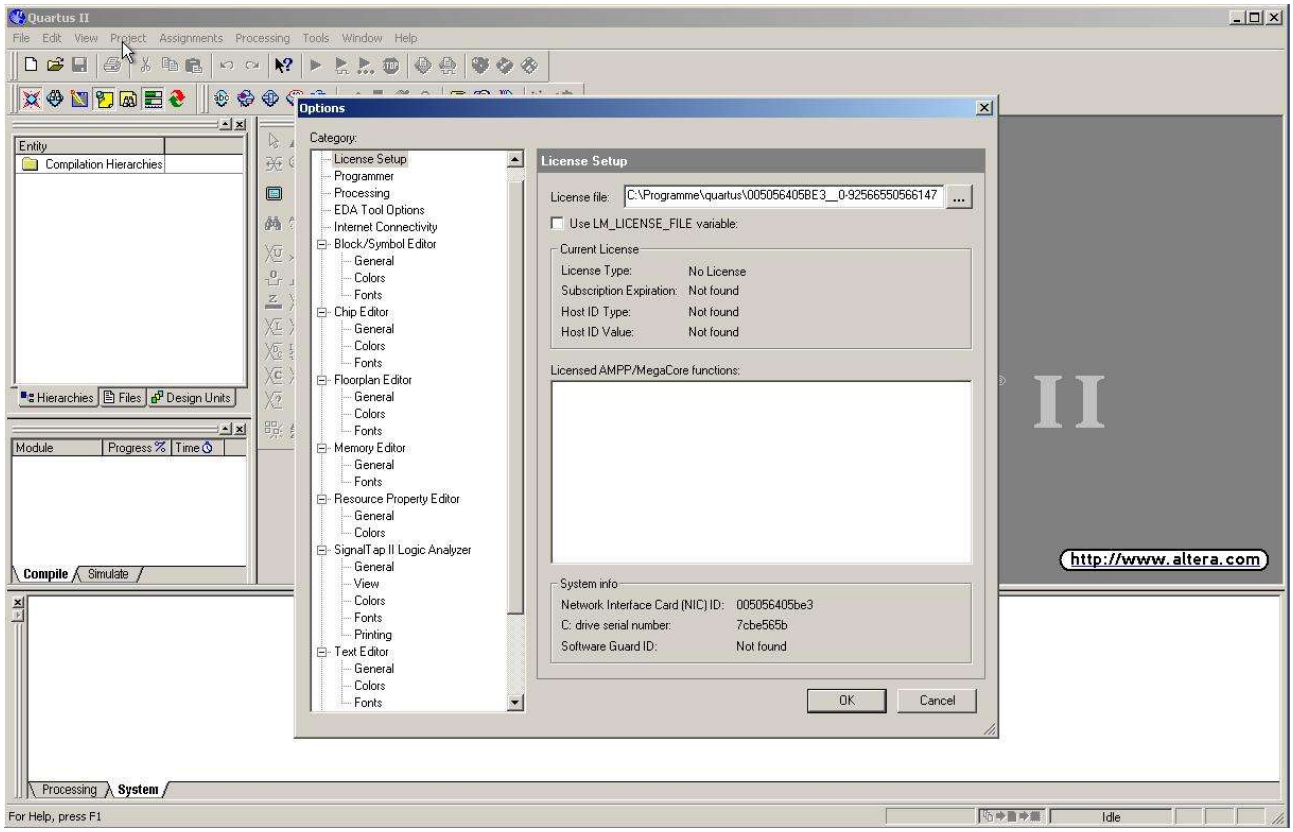

Figure 5: Quartus: Selection of the license file

Installation of a Driver for the Download Adapter Byte Blaster II, Installation of the Adapter.

To be able to download digital designs to the Sphinx-C100 there must be an established connection between Sphinx-C100 and your PC. For this purpose Altera offers several download cables. The following description refers to the use of a Byte Blaster II download cable, which is connected to the parallel line printer port of your PC. The other end of the adapter cable is connected to the JTAG interface or to the serial (AS) interface of the Sphinx. To operate the Sphinx please provide a 5V supply to the appropriate header pins (see above).

The driver for this adapter is stored in the Quartus II program directory i.e. „C:\program files\quartus\drivers“. The driver installation for your operating system works as follows (please note: instructions are for a Windows 2000 PC, unfortunately we don't own an english version of Windows 2000 and so we do not know the exact wording of the dialog windows):

1. Please go to the Control Panel (Start, Settings, Control Panel).
2. Double click on the hardware icon.

The following dialog appears (in english):

3. Chose the button „Next >“
4. The next dialog window shows several options. Choose the first one „Add device ...“ and afterwards „Next >“
5. The system will search for new hardware components, this may take some time. Afterwards you'll get a list of new hardware components found, you have to choose „Add new device“ and „Next >“.

Figure 6: Installation dialog 1

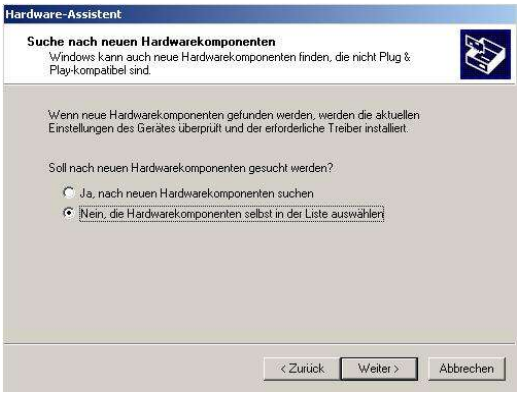

Figure 7: Installation dialog 2

6. The upcoming window offers two options. Please select the 2nd which says something like „No, select a hardware component from list“ and press the „Next >“ button.
7. Now, please select the option „Audio, Video and Game Controller“. The Byte Blaster II is handled as a audio device ;-). Again press „Next >“.
8. The next window contains two option lists. Please select „Unknown“ in the left list and press „From disc“. Next, a file select box will open and you have to select the driver from your Quartus installation directory (subdirectory „drivers“). (For example: Quartus has been installed in the directory C:\Programme\Quartus on Harddrive C; Then you have to select C:\Programme\Quartus\drivers) and confirm with „Open“ and „OK“. Now the following (Figure 8) window pops up:
9. The Byte Blaster device driver does not contain a valid digital signature. This is the reason for a warning. Just say „Yes“ here and the installation continues.
10. Now we are able to select the „Altera Byte Blaster“ from a list and to continue we have to press „Next >“.
11. The upcoming dialog window confirms the we have installed the new device with a default setting. Go on with „Next >“.

Figure 8: Installation dialog 3

12. And again some notice about the missing digital signature. Just ignore and say „Yes“ to go on.
13. The last, but not least dialog tells us that you have successfully installed the new device. Just press „Finish“ and reboot your system.

Finally all required installations are made and you own a powerful FPGA development environment to create complex logics for the Sphinx-C100. The programming environment Quartus II looks a bit confusing at the beginning.

First we want to use an example to show you how to create a logic design, how to program the Sphinx-C100 and how the simulation works, as well as how to start your own project.

An exact description of all the possibilities of Quartus is not intended. The Quartus help system is very good and there are a lot of good VHDL books available.

Creation of a digital Design with the Sphinx-C100

Introduction

The Sphinx-C100 needs a configuration file for executing a logic design. This is created by the synthesis of an abstract VHDL- or Verilog description or by a design, (created with the help of the graphic design entry feature of Quartus II). For successful compilation, the compiler has first to be informed about the name of the design entry file. After this the the fitter will place the design for a previously chosen FPGA type. The fitter has to be informed about the type of FPGA into which it should fit the compiled design.

There are several other adjustments and optimization variants from which to chose. All this information is summarized in a project. On the product CD you will find a predefined and well tested project. You can chose this with the dialog File -> Open Project -> ...\sphinx_counter_ex1. The file extension for Quartus II project files is „.quartus“. After the project is loaded into Quartus II, look on the left-hand side of the Quartus main program window, in a small project navigator window you will see the design entry file sphinx_counter_ex1. Open it (double click) in the middle of the Quartus II main window.

You will see the abstract VHDL design for a parametric, parallel loadable, 12bit forward- backward counter. The counter adds or subtracts the value of three on every positive edge of the clock signal and is asynchronously resettable. This clock requires only a few lines of VHDL and shows the power of this design language. The following figure shows a screenshot of the design software Quartus II with the loaded project and a open design entry file.

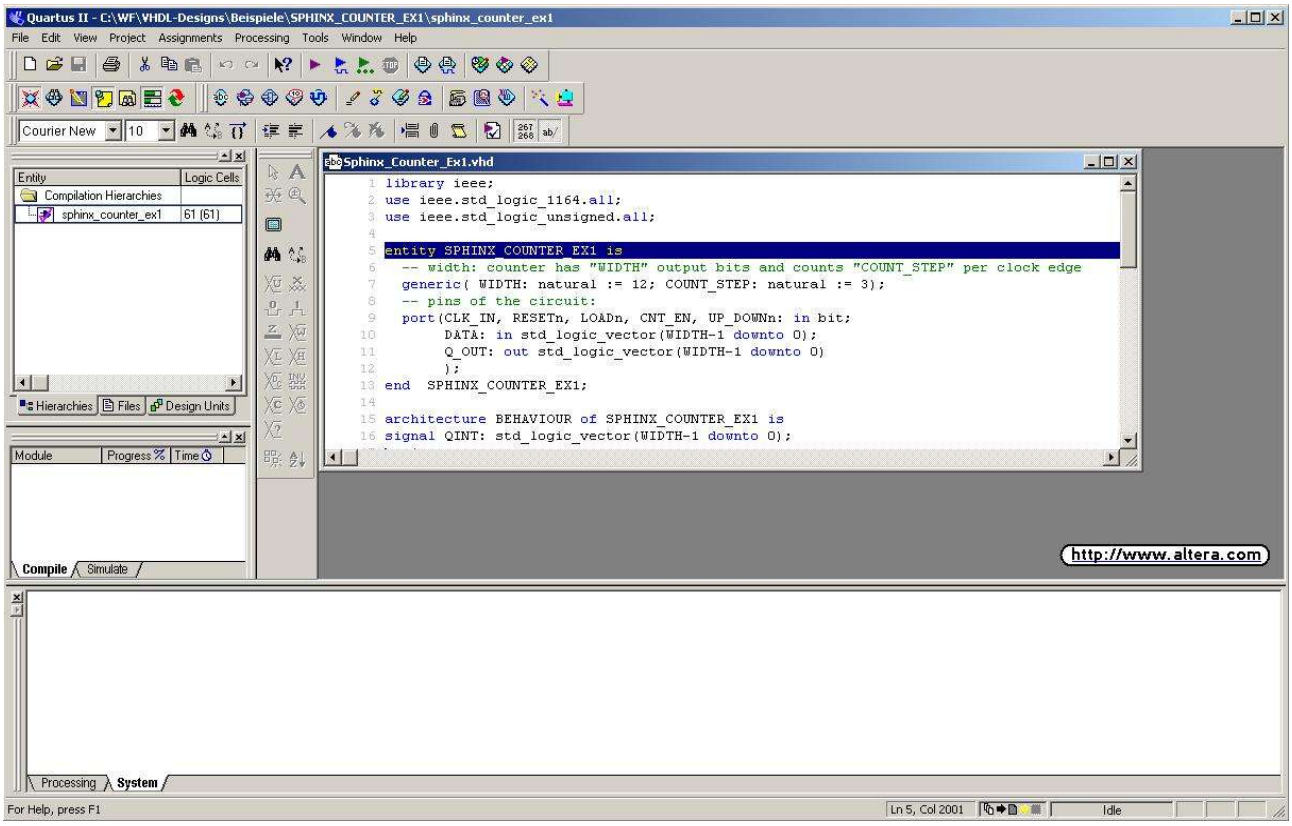

Figure 9: Loaded Quartus project with opened design entry file

Below the navigation window you can see a status window on the left-hand side where the progress of the compiler, fitter, simulator etc. is visualized. At the bottom there is a message window which displays compiler errors, compiler warnings and infos during the compilation of the project. Above these four window areas is a menu bar with several symbols and above the menu bar there is a pull down menu. Please take some time to play around with this design software to get familiar with the rich functionality. Quartus consists of several program parts, which are controllable in the menus „Tools“ and „Processing“. The most important components are:

1. Compiler: The compiler translates the design entry file in an object code which is the basis for the fitter. The fitter organizes the placement of the physical elements on the chosen FPGA type.
2. Simulator: with this component you have the opportunity to simulate every successfully compiled and fitted design without actually using the target hardware. The close conformity between simulation and hardware test results is surprising!
3. Programmer (in the menu „Tools“): This program is for downloading the configuration files to the target hardware; in our case to the Sphinx-C100.

Translation of the Design Entry File

The VHDL design entry file shown in the main window will be compiled and fitted for the Sphinx-C100 in the next step. All required Quartus II

adjustments are already stored in the project file. The compiler will be started using the menu „Processing“ by selecting „Start Compiler“ or by double clicking on the appropriate symbol. This may take awhile depending on your PC. During compiling and fitting you can track compiler, fitter and assembler messages in the message window and watch the status of the compilation in the status window. Figure 10 Shows a screenshot of the started compiler.

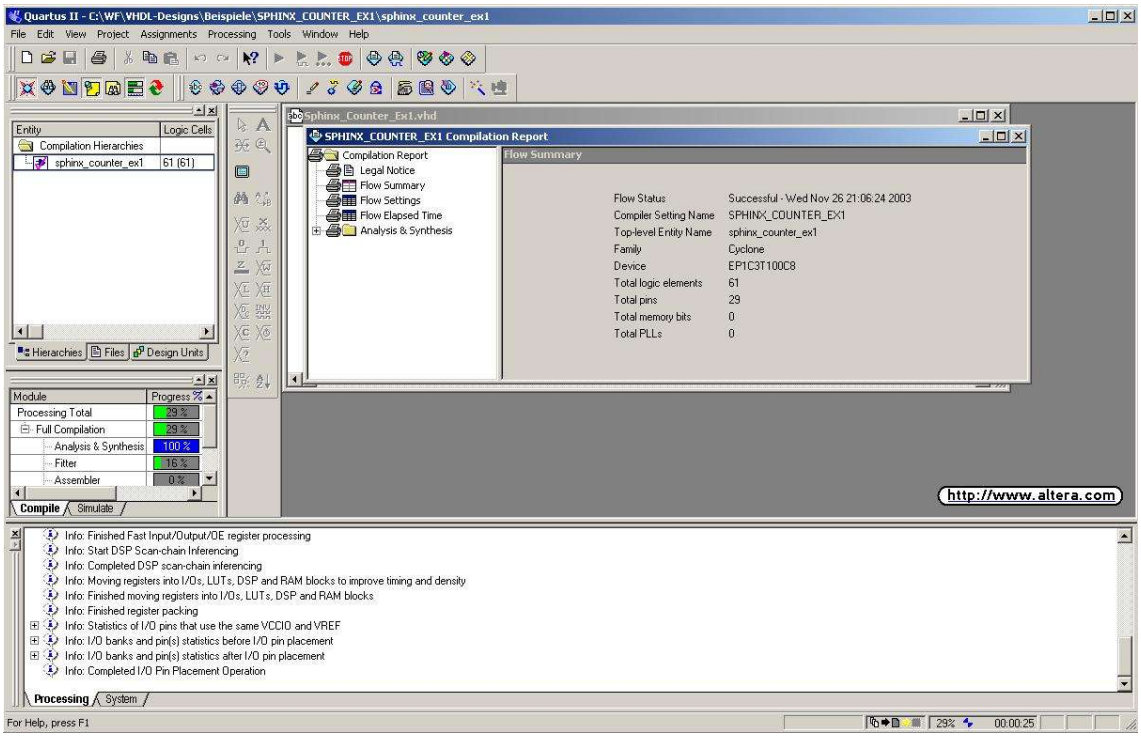

Figure 10: Quartus II compiler run

The compilation and fitting will lead, in the end, to the message „Full Compilation was successful“, if the design was without errors. Please confirm this message with „OK“. After compilation and fitting you can find several files which were created during the translation process in the design directory. There are two files with the file extensions .sof and .pof. These are the input files for the programmer, the configuration files, which are downloadable to the Sphinx-C100. Use the files (.sof or .pof), if you want to download to the JTAG interface and the file .pof, if you like to use the JTAG serial programming (AS) of the configuration device instead.

Programming of the Sphinx-C100

For programming, open the menu „Tools“ and select the entry „Programmer“. A dialog window shows where some adjustments are necessary:

1. Choice of the connected programming adapter:
   Select „Hardware“ and then „Add Hardware“. In the previously opened dialog window please select the programmer hardware (in this case Byte Blaster MV oder Byte Blaster II) and the interface, where the hardware is connected (in this case lpt1). Confirm with „OK“. After this the hardware should be confirmed with the push button „Select Hardware“ (chose the Byte-Blaster II in the list on the left window side before confirmation). The procedure ends with „Close“.

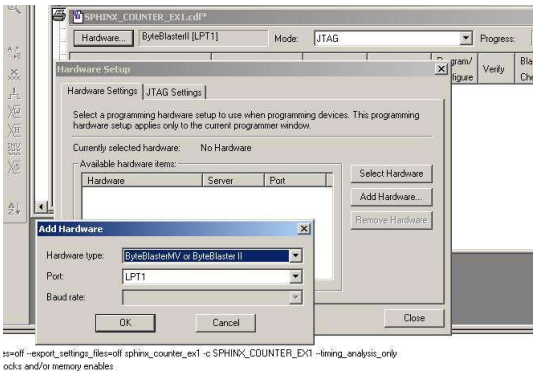

Figure 11: choice of the programming adapter

2. Selection of the programming mode: After „Mode:“ you will find a drop down menu; open this. Interesting for our purposes are the two entries JTAG and active serial (AS) programming. The JTAG mode is used, if you want to download the configuration file directly to the FPGA using the JTAG interface (method: volatile). The AS mode (last entry in the list) is used, if you want to program the configuration device using the AS interface. (method: non volatile).
3. Last but not least the programmer has to be informed about the configuration file. Right mouse click in the main area of the programmer window (Add File). The information about the selected FPGA is already stored in the selected file so that you have not to select the FPGA type by hand using the „ADD Device“ menu (right mouse click).
   For your information: on the Sphinx-C100 there is a configuration device of the type **EPCS1**. The successful configuration of the programmer looks like this:

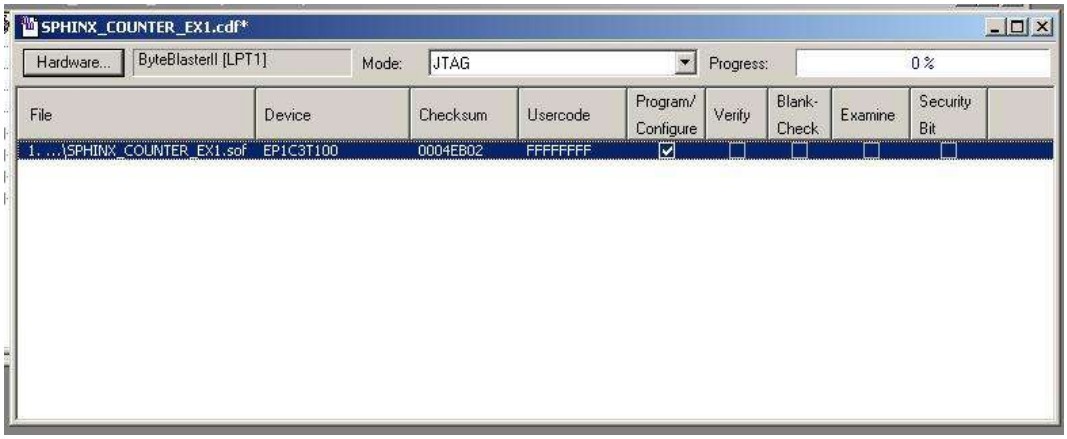

Figure 12: programmer configuration

The chosen configuration file (shown in Figure 11) is the selected one (marked blue). Right beside the filename there is information about the file's

check sum and the user code (FFFFFFFF means, that no user code has been programmed). Other than this there are options which are provided by the selected programming mode. In case of JTAG there is only one option possible: „Program/Verify“. If you use the AS mode you can chose „Program/ Configure“, „Verify“, „Blank-Check“ and „Examine“.

The download will be started by selecting „Start Programming“ in the main pull down menu „Processing“. Programming the Sphinx-C100 only takes a few seconds.

Now you have finished a complete development cycle starting with loading the design entry and ending with downloading to the target. You can find a lot of information about the compiled design in the info files of the project. For instance, there is the „Last Compilation Floor-plan“ in the menu „Assignments“. There you can find the actual pin-out of the FPGA. Please use this information, if you want to test your hardware.

What is missing? The design entry itself, which you loaded in the example from the product CD. How create your own new projects is described in the next chapter.

Opening a new Quartus II Project

Opening a new Quartus II project is easy. Use the menu „File“ and the entry „New Project Wizard“. The Wizard leads you through the process step by step. Please note, that you should select a new design directory for your new design and don't try to create a new design in an existing project directory, (this will fail). After you finish the Wizard all required adjustments are done and stored in the project file (.quartus).

While creating of a new project you have to choose the FPGA for your design. The Sphinx-C100 is equipped with a Cyclone EP1C3T100C8, please use this one.

Use the menu „File“ to create a new file (you might chose a graphic design entry, a text based VHDL or even a Verilog entry). Please name this file using the same name as you did for your project. This is the top level file of your design.

For example: a project is created with the project name „ChristmasTree“. All path assignments pointing to the empty project directory i.e. „ChristmasTreeDirectory“ and the project is named „ChristmasTree“ (all these settings are done by the „New Project Wizard“). In this directory you will later find (after successful compilation) all created files concerning this project (the database). If you open a new file (for example a graphical design entry file), then you should store this file with the name „ChristmasTree“ in the project directory. If you prefer to use VHDL text-based design entry please make certain that the entity of the top level design file is also named to match your project – in this case, „ChristmasTree“.

If you have done your set-up carefully using these connections it should now be easy to create designs, compile them and download the configuration files to the Sphinx-C100.

Simulation of a Project at the Example sphinx_counter_ex1

The simulation of a project can be done with the Quartus II design software suite built-in simulator. The following description refers to this simulator.

After successful compiling of a project, you will also find a timing file in your project directory. This file contains timing information for the simulator, defines the FPGA type and takes the pin placement of the circuit on the FPGA into consideration. For that reason it is not possible to simulate a circuit before the compiler and fitter have successfully been finished.

Before the simulation can be started, the simulator needs information which inputs and outputs of the design have to be taken into account. This information is provided by the vector waveform file. To create this file please use the button shown in the following figure.

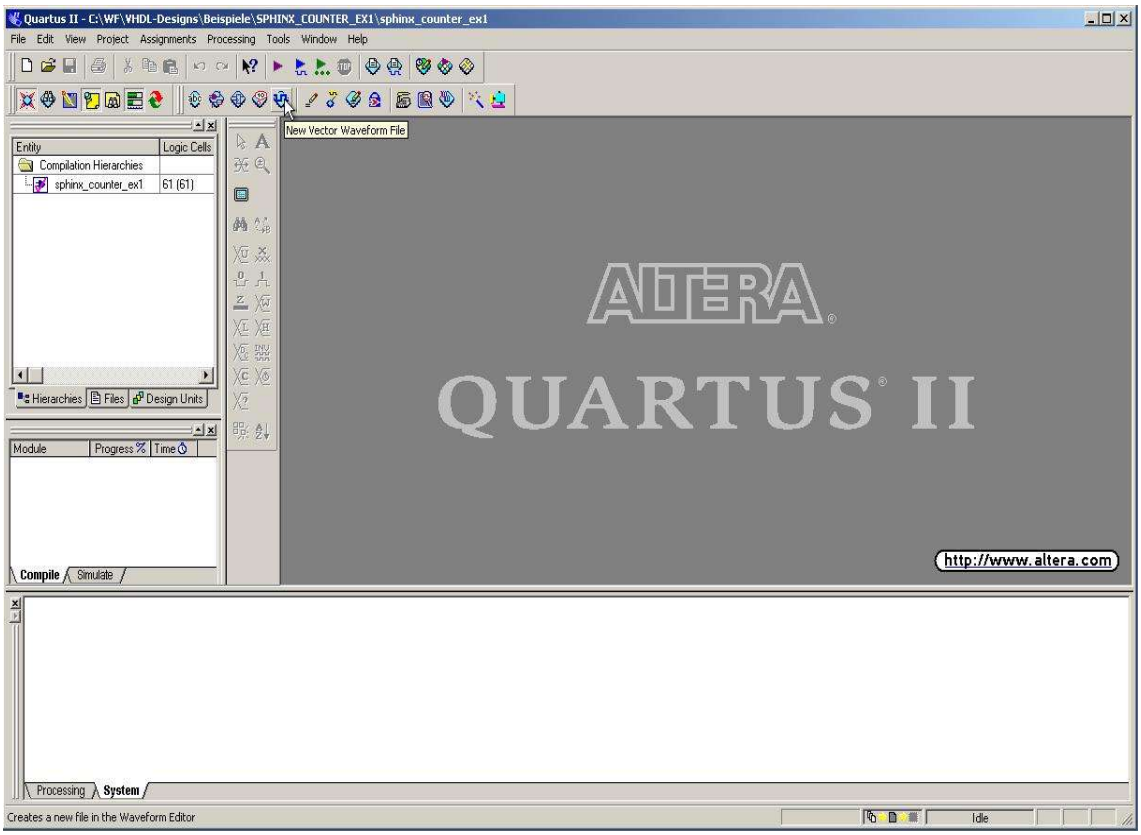

Figure 13: Simulator: Vector Waveform File

After opening a waveform file you see a window split in two parts. Right mouse click in the left-hand window area to open a dialog with the option „Insert Node or Bus“. Select this entry. In the next dialog chose the push button „Nodefinder“. After this you can assign inputs and outputs to the vector waveform file in the next window. When first open these window areas are empty. Selecting the push button „Start“, to see a list of all possible signals displayed in the left window area. Now you can select the signals of your interest and copy them to the right window area by using „>“ button. If you have finished your choice, confirm it with the „OK“ button. Figure 15 gives an example for the simulation of the counter.

After you have finished the assignment, the simulator window opens up. Now you can see all selected signals illustrated as traces. For the simulation it is

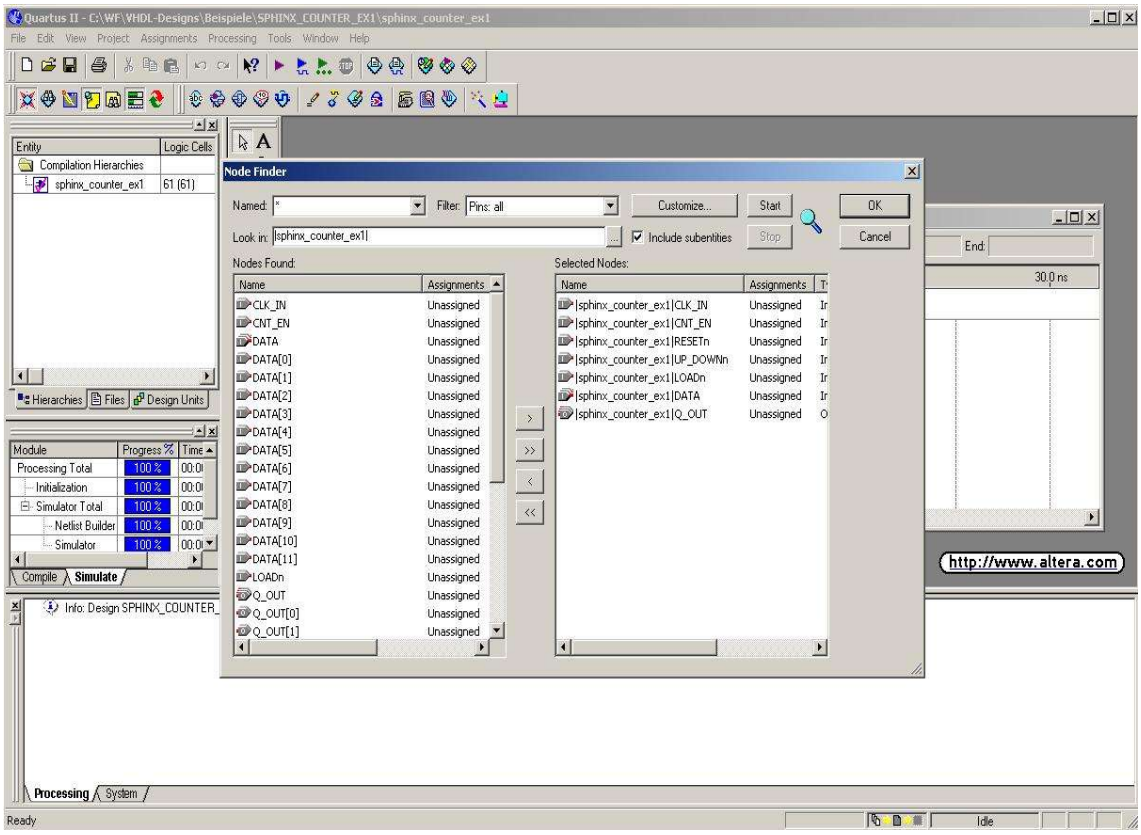

Figure 14: Simulator: Node-Finder

also necessary to stimulate the input signals. The simulator will use this information to calculate the behavior of the output signals. To do this, right mouse click on an input signal in the left-hand window area and assign a value to it. Please read the online documentation of Quartus II or just play a little bit to find out which other possibilities are available to you.

Note: The total simulation time is adjusted in the menu „Edit“ -> „End Time“. After all inputs are stimulated the simulator can be started in the pull-down menu „Processing“ -> „Start Simulation“.

In the following, last screenshot, the simulation result for the given counter is illustrated. The view area can be zoomed, and there are measurements possible (please refer to the online help). In this figure you can see the „Stimuli“ of the input signals. It needs practice to achieve good simulation results.

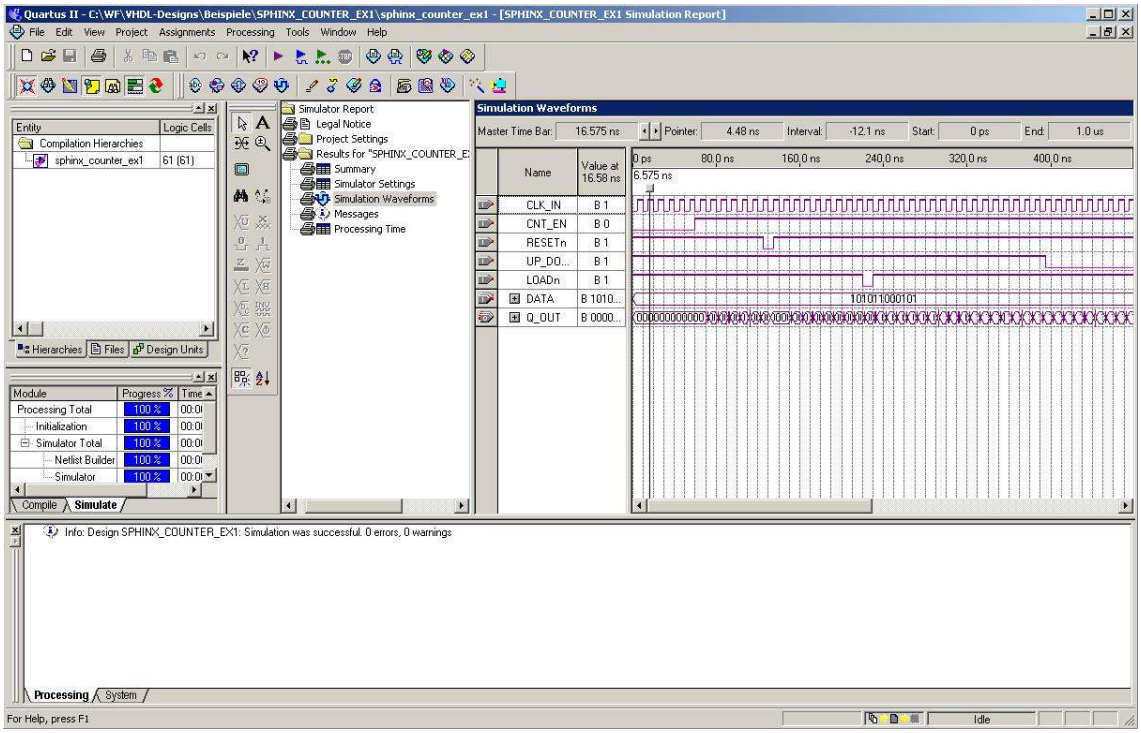

Figure 15: Simulation result for sphinx_counter_ex1

You have now become acquainted with the most important tools for creating your own designs and burn them in silicon. We hope that you will enjoy your Sphinx-C100 and it will help you to reduce your digital logic development time. Soon, you can find more information about Sphinx-C100 and examples at http://www.inventronik.de. Please feel free to give us feedback – we are always interested in improving our products.

Attachment

Copyright

Liability

Inventronik does not accept any liability for the use of the Sphinx-C100 or the provided software, or software downloaded from our Internet site whether they be used together or separately. The company Inventronik likewise does not accept any form of liability if, by the use of these products, patent or license rights of a third-party are injured. The company Inventronik reserves the right to make and instigate technical changes with an eye towards product improvements at will and without previous announcement, this includes changes and supplements to this documentation.

Trademark

The label names or registered trade marks mentioned in this document serve only as identification and are property of their respective owners.

Terms of the Warranty (guarantee)

Inventronik grants a warranty of two years, precluding the appropriate handling of the product. The warranty expires immediately if the product is mishandled or manipulated.