

User Guide
openPowerlink FPGA Slave
Reference Design

Author:	Zelenka Joerg
Version:	V1.0
Date:	27/10/2009
File:	User Guide.doc

INDEX

1	Document Overview	3
2	Design Features.....	3
3	Performance Restriction.....	3
4	Requirements	3
5	Quick Start	4
5.1	Unzip Package	4
5.2	Build Makefile	5
5.3	Program FPGA (Hardware)	6
5.4	Program Nios II CPU (Software)	7
5.5	Run a Nios II Terminal (immediate debugging)	8
6	Quartus II.....	8
7	Nios II IDE.....	9
7.1	Import Software Projects	9
7.2	Configure Target Connection.....	12
7.3	Run or Debug Target	12
8	Troubleshooting.....	13

1 Document Overview

This document gives you a "Quick Start" instruction (5 @ page 4) to successfully run the openPowerlink FPGA Slave Reference Design in a few minutes. In addition the performance restrictions (3 @ page 3) are given and should be considered at any time! In chapter 6 the steps are given to import the software projects into Nios II IDE to start development. In case of error messages please refer to section 8 starting at page 13.

2 Design Features

The openPowerlink FPGA Slave Reference Design uses one TXPDO and two RXPDOs (refer to Table 2-1).

Table 2-1: PDOs

Source/Sink on EvalBoard	I/O Size	PDO Index
4 Buttons (below 7-Segment Display)	Input 8 Bits (lower nibble used)	0x6000
6 LEDs (above Buttons)	Output 8 Bits (upper six bits used ²)	0x6200
16 Outputs (24V I/O Header) ¹	Output 16 Bits	0x6300

Table Note:

1. The 24V I/O stage needs an extra 24V supply!
Please refer to the DBC3C40's Datasheet (Version 1.10) at page 33!
2. LED0 and LED1 are used as CN Status- and Error-LEDs.

3 Performance Restriction

It is recommended considering the following restrictions (Table 3-1) to run a stable Powerlink Network with the provided openPowerlink FPGA Slave!

Table 3-1: performance restrictions

Powerlink Cycle Time ⁴	400	µs (min)
Input Process Data per Cycle ¹	3	(max)
Input Process Data volume per Cycle ^{1, 2}	36	Bytes (max)
Output Process Data volume per Cycle ²	36	Bytes (max)
Asynchronous Data MTU (ASnd ...) ^{3, 5}	1514	Bytes (max)

Table Note:

1. Per Powerlink Cycle it is allowed to receive the maximum data volume (EPL Payload) within one PDO (PReq) or three PDOs (PReq, PRes MN and PRes Cross Traffic).
2. EPL Payload
3. asynchronous phase (between SoA and SoC)
4. Compiler Optimization Level must be the highest (-O2)
5. Ethernet Frame excl. CRC

It is recommended to read the openMAC's documentation (docu_openMacv12.pdf, focus on chapter "Application Note") and openPowerlink specific documents (e.g. from SysTec ...) before performing any changes to the Reference Design (changes in hardware or software)!

4 Requirements

- Development Board EBV DBC3C40 (Mercury Board)
- Altera Quartus II v9.0 or newer (Web Edition is also possible)
- Altera Nios II Embedded Design Suite v9.0 or newer

- *Optional with "DBC3C40_Setup.exe for Quartus 9.0" installation to program the reference design non-volatile into the device (refer to the newest DBC3C40 data sheet – version 1.10 or newer)*
- Experiences with this development environment are required
- POWERLINK network as described in main readme.txt
- openPowerlink Stack V1.5.9.6

Usually the FPGA Slave Reference Design will be delivered separately in a Zip-file ("openPOWERLINK_openMAC_v1.5.9.6.zip" ... Reference Design with openMAC + Nios II).

The openPowerlink Stack is located in the file "openPOWERLINK_v1.5.9.6.zip".

5 Quick Start

This chapter helps you to run the openPowerlink FPGA Slave Reference Design. It is crucial to follow these instructions by order:

1. Unzip Package
2. Build Makefile
3. Program FPGA (Hardware)
4. Program Nios II CPU (Software)
5. Run a Nios II Terminal (immediate debugging)

In case of errors refer to section 8!

The package will be copied into a directory which can differ. So, in this guide it will be called "**MYROOT**".

5.1 Unzip Package

Please follow these instructions in the right order!

1. Unzip the openPowerlink Stack Source Files out of the file openPOWERLINK_v1.5.9.6.zip into your directory **MYROOT**.
2. Unzip the FPGA Slave specific Source Files (openMAC and System Hardware Design, Drivers ...) out of the file openPOWERLINK_openMAC_v1.5.9.6.zip into the same directory (**MYROOT**) as before.
3. Compare **MYROOT** with the in Figure 5-1 given structure.

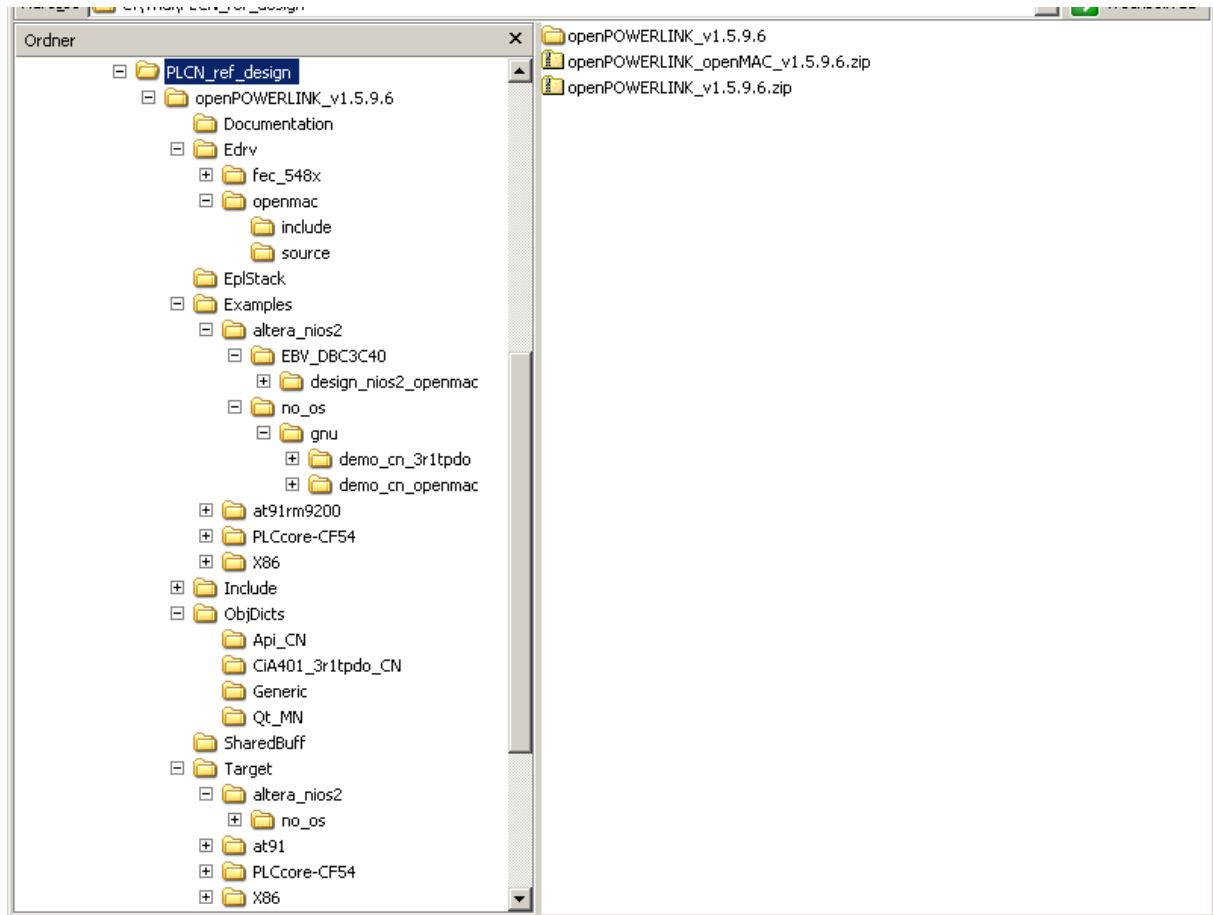


Figure 5-1: reference design directory

5.2 Build Makefile

If you have successfully unzipped the reference design project, you can build the Makefile for the software compilation. Please follow the following steps!

1. Open Nios II Command Shell out of the Start Menu (Start – Programs – Altera – Nios II EDS 9.0 – Nios II 9.0 Command Shell).
You can find an example command shell in Figure 5-2.

2. Change to your drive (usually C) by typing in

```
cd c:
```

3. Change to the directory

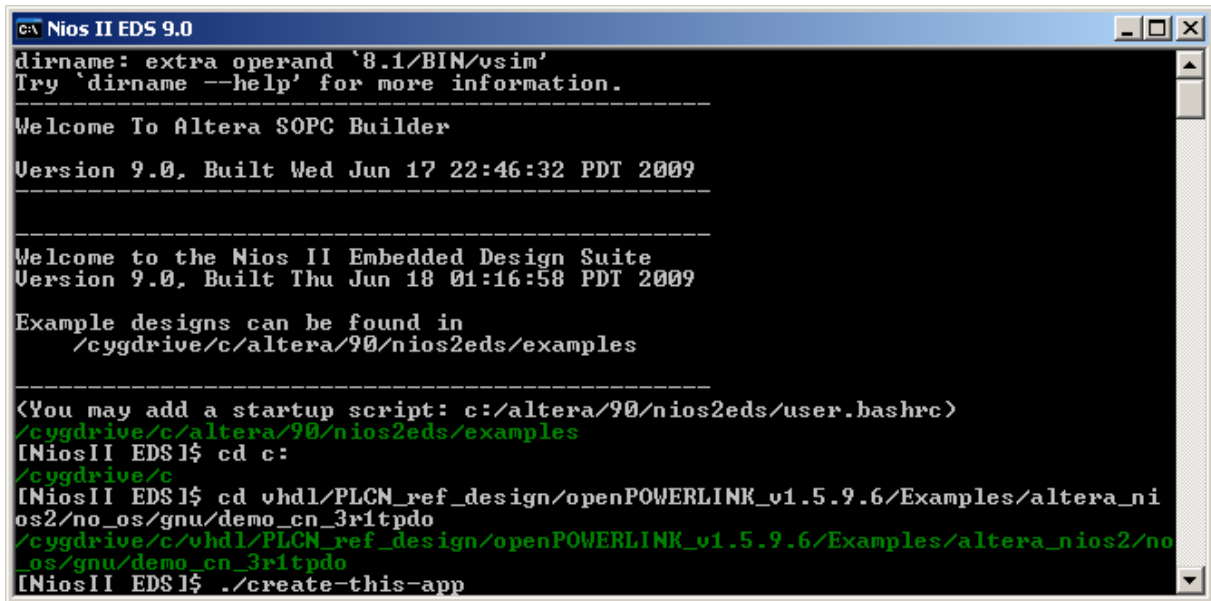
MYROOT/openPOWERLINK_v1.5.9.6/Examples/altera_nios2/no_os/gnu/demo_cn_3r1tpdo by typing in (no line break!)

```
cd myroot/openPOWERLINK_v1.5.9.6/Examples/altera_nios2/no_os/gnu/  
demo_cn_3r1tpdo
```

4. Run the script "create-this-app" with typing into the shell. The script execution will last some minutes (depends on your PC).

```
./create-this-app
```

5. Don't exit the command shell, because you will need it for downloading the application to the Nios II CPU!



```

Nios II EDS 9.0
dirname: extra operand '8.1/BIN/usim'
Try 'dirname --help' for more information.

-----
Welcome To Altera SOPC Builder
Version 9.0, Built Wed Jun 17 22:46:32 PDT 2009

-----
Welcome to the Nios II Embedded Design Suite
Version 9.0, Built Thu Jun 18 01:16:58 PDT 2009

Example designs can be found in
/cygdrive/c/altera/90/nios2eds/examples

-----
<You may add a startup script: c:/altera/90/nios2eds/user.bashrc>
/cygdrive/c/altera/90/nios2eds/examples
[NiosII EDS] $ cd c:
/cygdrive/c
[NiosII EDS] $ cd vhdl/PLCN_ref_design/openPOWERLINK_v1.5.9.6/Examples/altera_ni
os2/no_os/gnu/demo_cn_3r1tpdo
/cygdrive/c/vhdl/PLCN_ref_design/openPOWERLINK_v1.5.9.6/Examples/altera_nios2/no
_os/gnu/demo_cn_3r1tpdo
[NiosII EDS] $ ./create-this-app

```

Figure 5-2: example command shell

5.3 Program FPGA (Hardware)

The next step is to program the hardware design into the FPGA. This is done with the Programmer Tool in Quartus II. Follow the next steps carefully!

1. Open Quartus II (usually on Desktop or via Start Menu: Start – Programs – Altera – Quartus II 9.0 – Quartus II 9.0)
2. Open the openPowerlink FPGA Slave Reference Hardware Project
File – Open Project ...
3. Browse to the Reference Design at
MYROOT\openPOWERLINK_v1.5.9.6\Examples\altera_nios2\
EBV_DBC3C40\design_nios2_openmac
4. Select "nios_openMac.qpf" and open the Quartus II Project File
5. Click in Quartus II Tools – Programmer
6. Verify if your EBV Board is connected to a power source and to your PC (e.g. via USB Blaster)!
7. You can directly start the programming, because you have already opened the Quartus II Project File! So, click Start.
8. The binary download is done within some seconds and is indicated with the progress bar at top right.
9. You can exit the Programmer and Quartus II if you like.

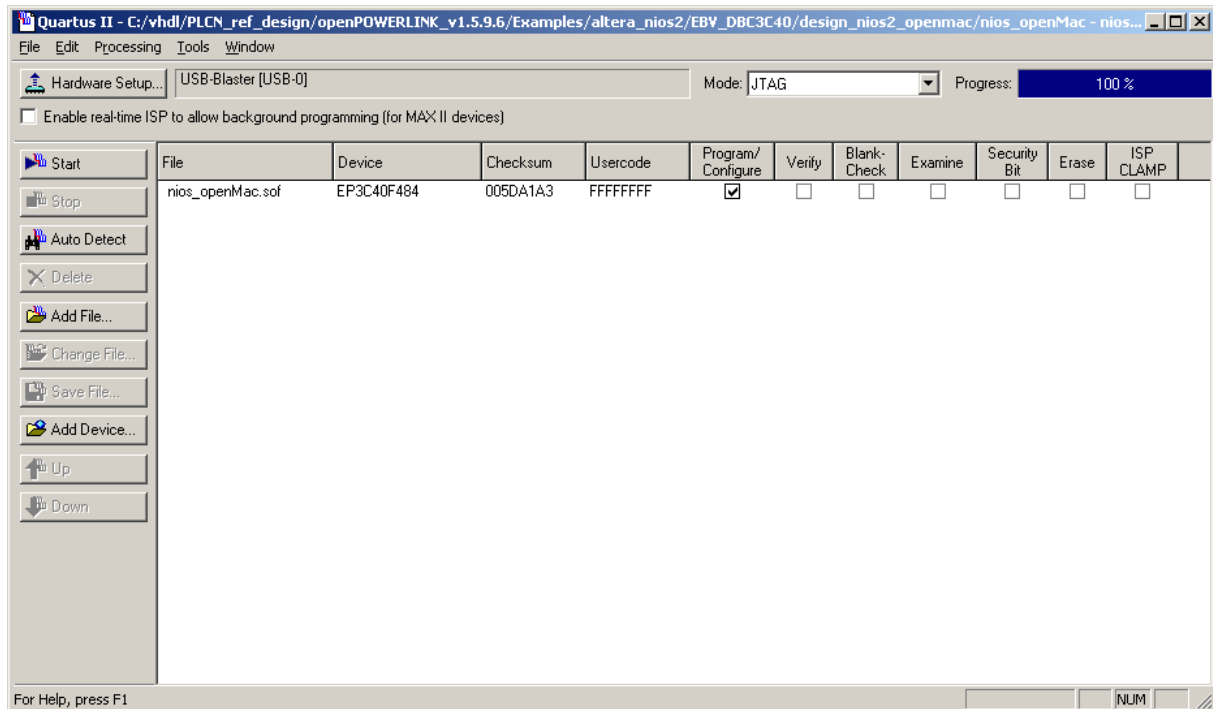


Figure 5-3: Quartus II Programmer example (progress done)

5.4 Program Nios II CPU (Software)

After downloaded the Hardware Design into the FPGA (done in section 5.3) you have the ability to access the Nios II CPU via JTAG. The software application can be downloaded in a command shell.

1. Open a Nios II Command Shell or use the one you have already opened in section 5.2.
2. Type the following command into the shell

```
make download-elf
```

3. Verify if the elf-download was done without any error (refer to Figure 5-4).

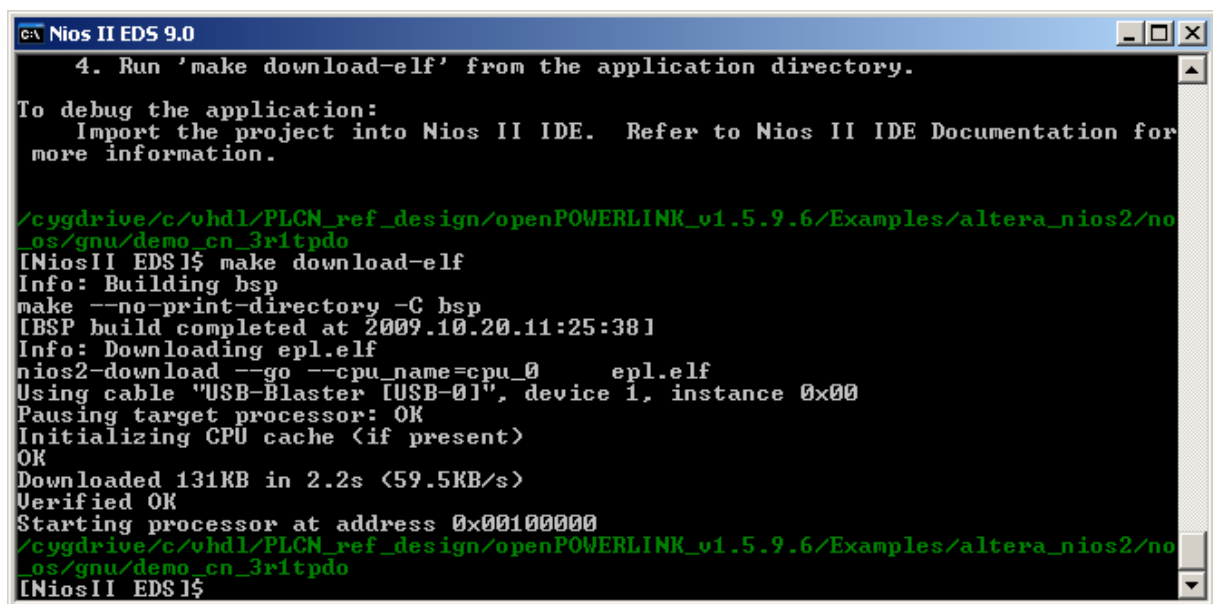


Figure 5-4: command shell download application

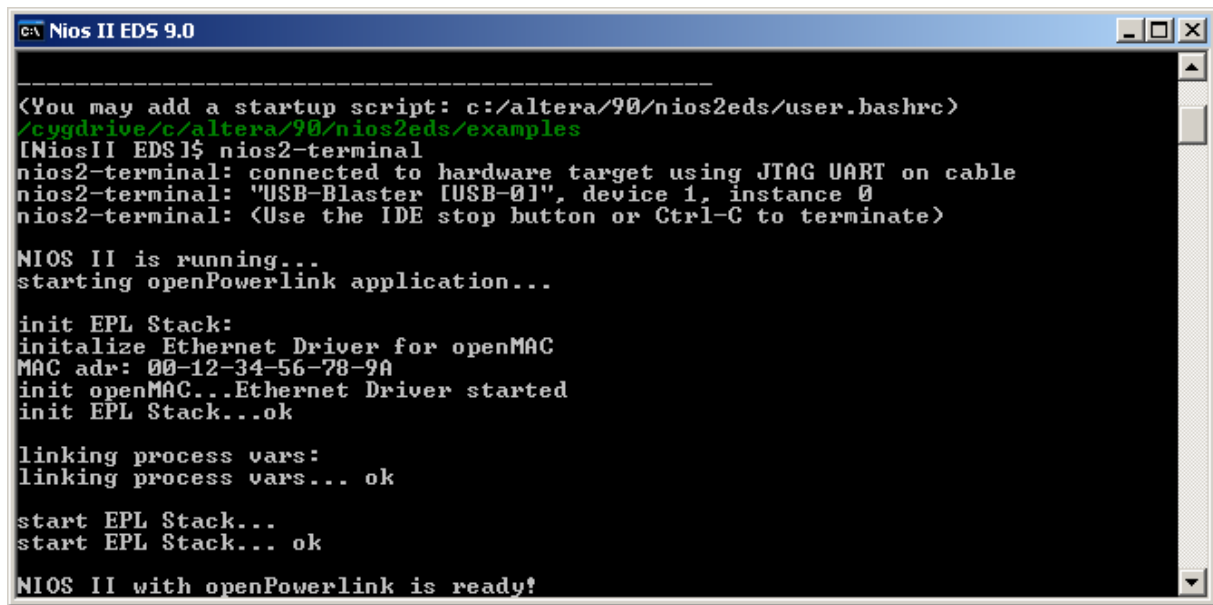
5.5 Run a Nios II Terminal (immediate debugging)

After downloading the Hardware Design into the FPGA (section 5.3) and programmed the Nios II CPU (section 5.4), the openPowerlink FPGA Slave is ready to operate. You can obtain further information via the JTAG interface. Follow these steps:

1. Open a new Nios II Command Shell out of the Start Menu (Start – Programs – Altera – Nios II EDS 9.0 - Nios II 9.0 Command Shell).
2. Type into the shell the following command

```
nios2-terminal
```

3. Now you can see STDOUT messages (refer to Figure 5-5).



```

CA Nios II EDS 9.0

<You may add a startup script: c:/altera/90/nios2eds/user.bashrc>
/cygdrive/c/altera/90/nios2eds/examples
[NiosII EDS] nios2-terminal
nios2-terminal: connected to hardware target using JTAG UART on cable
nios2-terminal: "USB-Blaster [USB-01]", device 1, instance 0
nios2-terminal: (Use the IDE stop button or Ctrl-C to terminate)

NIOS II is running...
starting openPowerlink application...

init EPL Stack:
initialize Ethernet Driver for openMAC
MAC adr: 00-12-34-56-78-9A
init openMAC...Ethernet Driver started
init EPL Stack...ok

linking process vars:
linking process vars... ok

start EPL Stack...
start EPL Stack... ok

NIOS II with openPowerlink is ready!

```

Figure 5-5: Nios II Terminal example

6 Quartus II

Quartus II includes a hardware development environment to build hardware designs for Altera FPGAs. The reference design's top level was built in a schematic file (refer to Figure 6-1).

You can use directly the reference design for building your specific openPowerlink Slave (CN) application. Please refer to the openMAC's documentation (docu_openMacv12.pdf, focus on chapter "Application Note")!

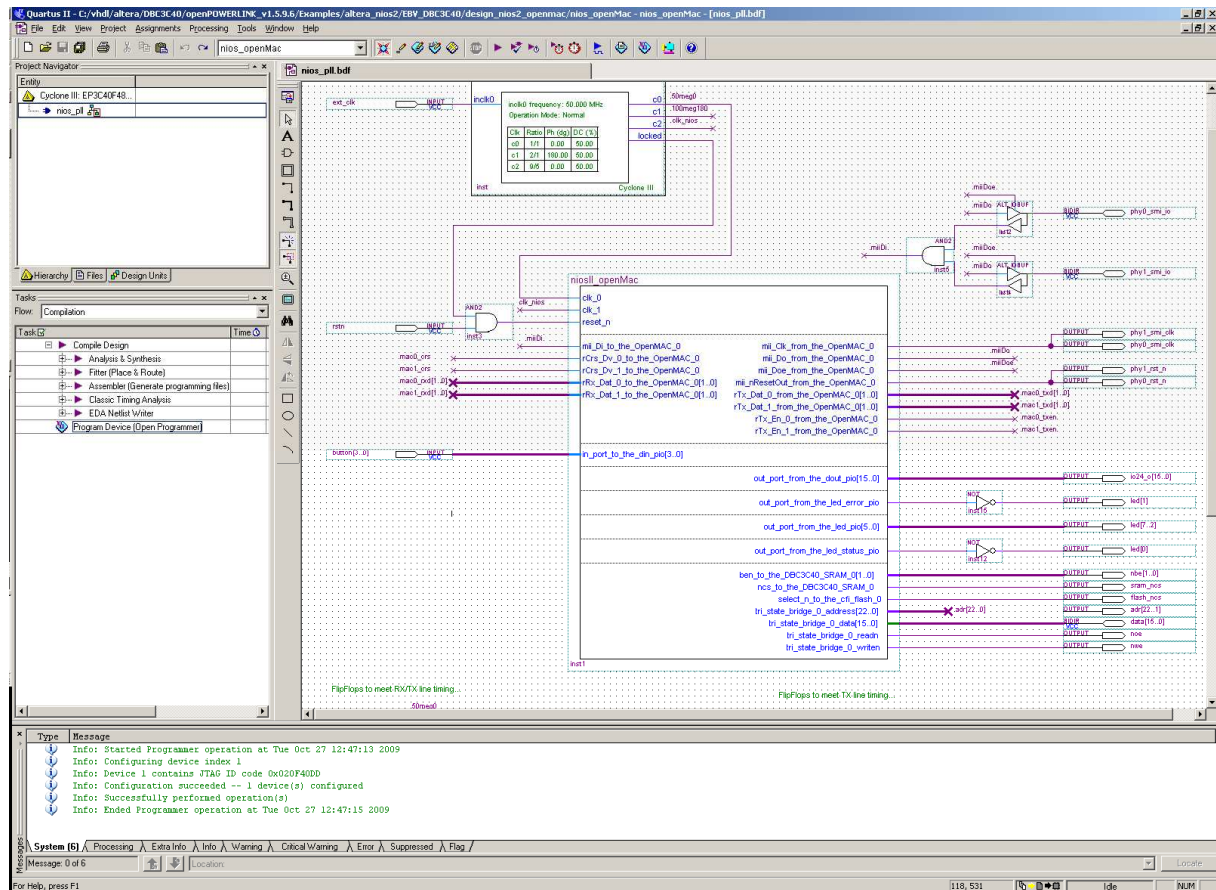


Figure 6-1: Quartus II with opened reference design

7 Nios II IDE

Altera provides a software development environment called Nios II IDE. The openPowerlink FPGA Slave Reference Design application can be imported into the IDE. Before doing that you need to build a Makefile considering the following arguments by calling the "create-this-app" script in the Nios II Command Shell.

```
./create-this-app ...
```

Instead of the "...", you can add arguments. The meaning is given in Table 7-1. These arguments can be used in combination. No argument is allowed if you build the Makefile the first time (no current Makefile is available).

Table 7-1: arguments' meaning

--rebuild	This argument causes the deletion of the current Makefile and rebuilds a new one. Should be used after changing the Hardware System in the SOPC.
--debug	This argument builds a Makefile with the highest compiler optimization level. Should be used when debugging the Nios II CPU's software application.

Example: rebuild Makefile and change optimization level to none (for debugging)

```
./create-this-app --rebuild --debug
```

7.1 Import Software Projects

Follow these steps to successfully import the openPowerlink Reference Software Project:

1. Run Nios II IDE (usually on desktop or via Start Menu Start – Programs – Altera – Nios II EDS 9.0 - Nios II 9.0 IDE)
2. Follow the instructions if you run IDE the first time!
3. Click File – Import

4. Expand Altera Nios II and select "Existing Nios II software build tools project or folder into workspace" (refer to Figure 7-1)
5. Next
6. Browse to **MYROOT**\openPOWERLINK_v1.5.9.6 (ignore warnings!) to get the openPowerlink Stack Source Files (refer to Figure 7-2)
7. Finish
8. Click File – Import again
9. Expand Altera Nios II and select "Existing Nios II software build tools project or folder into workspace" (refer to Figure 7-1)
10. Next
11. Browse to **MYROOT**\openPOWERLINK_v1.5.9.6\Examples\altera_nios2\no_os\gnu\demo_cn_3r1tpdo (ignore warnings!) to get the example application of a CN (refer to Figure 7-3)
12. Finish

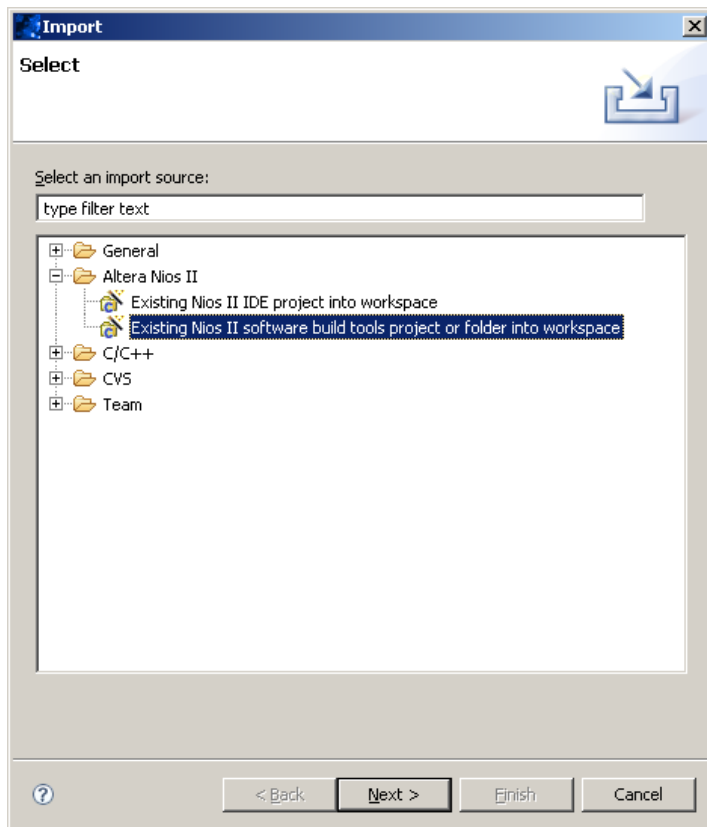


Figure 7-1: import existing software project

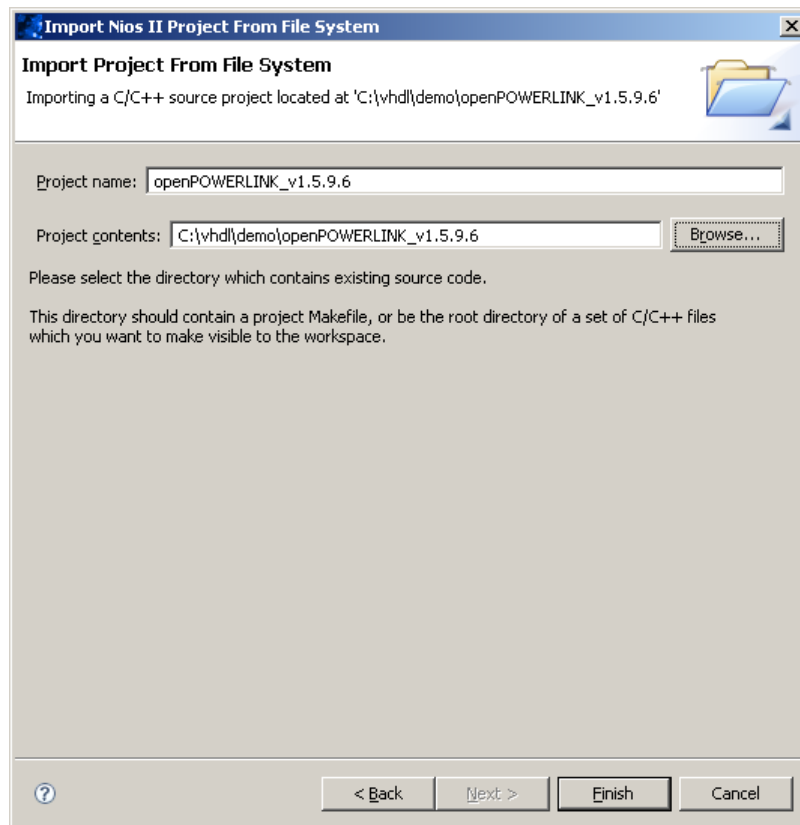


Figure 7-2: import openPowerlink Stack sources

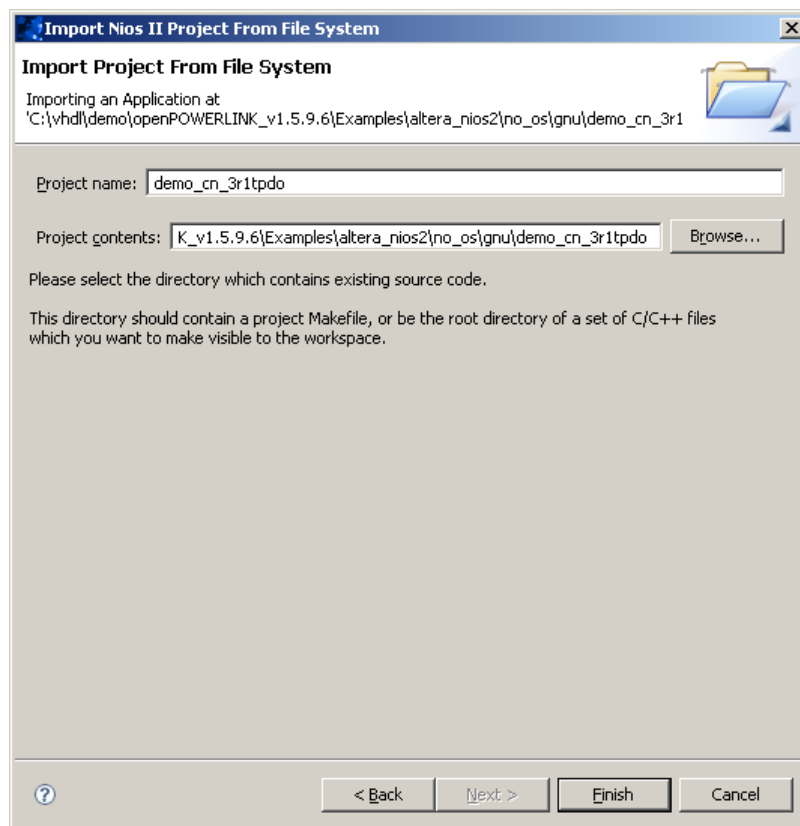


Figure 7-3: import the example application

7.2 Configure Target Connection

For starting debugging or running sessions you need to configure a Nios II Hardware configuration. Follow these steps (refer to Figure 7-4):

1. Click Run – Run...
2. Select Nios II Hardware and click on the new button
3. An error will occur, because the *.ptf file (create by SOPC) was not found.
4. Click brows at the Target Hardware Section
5. Browse to
MYROOT\openPOWERLINK_v1.5.9.6\Examples\altera_nios2\EBV_DBC3C40
 \design_nios2_openmac
6. Select the file niosII_openMac.ptf
7. Click Close and Save Changes!

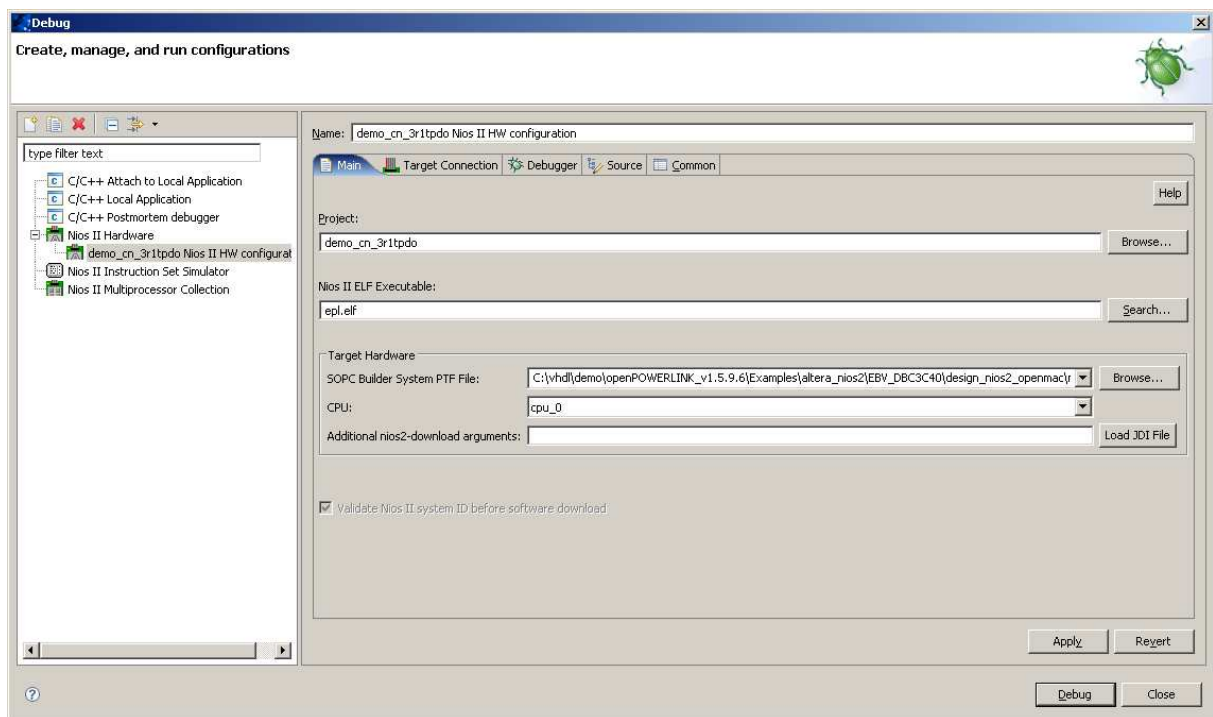


Figure 7-4: configure target

7.3 Run or Debug Target

You can run or debug the target via the buttons in Nios II IDE or via the menu Run – Run... or Run – Debug...

Always verify if the EBV Board is connected to a power source and to your computer via the USB Blaster box.

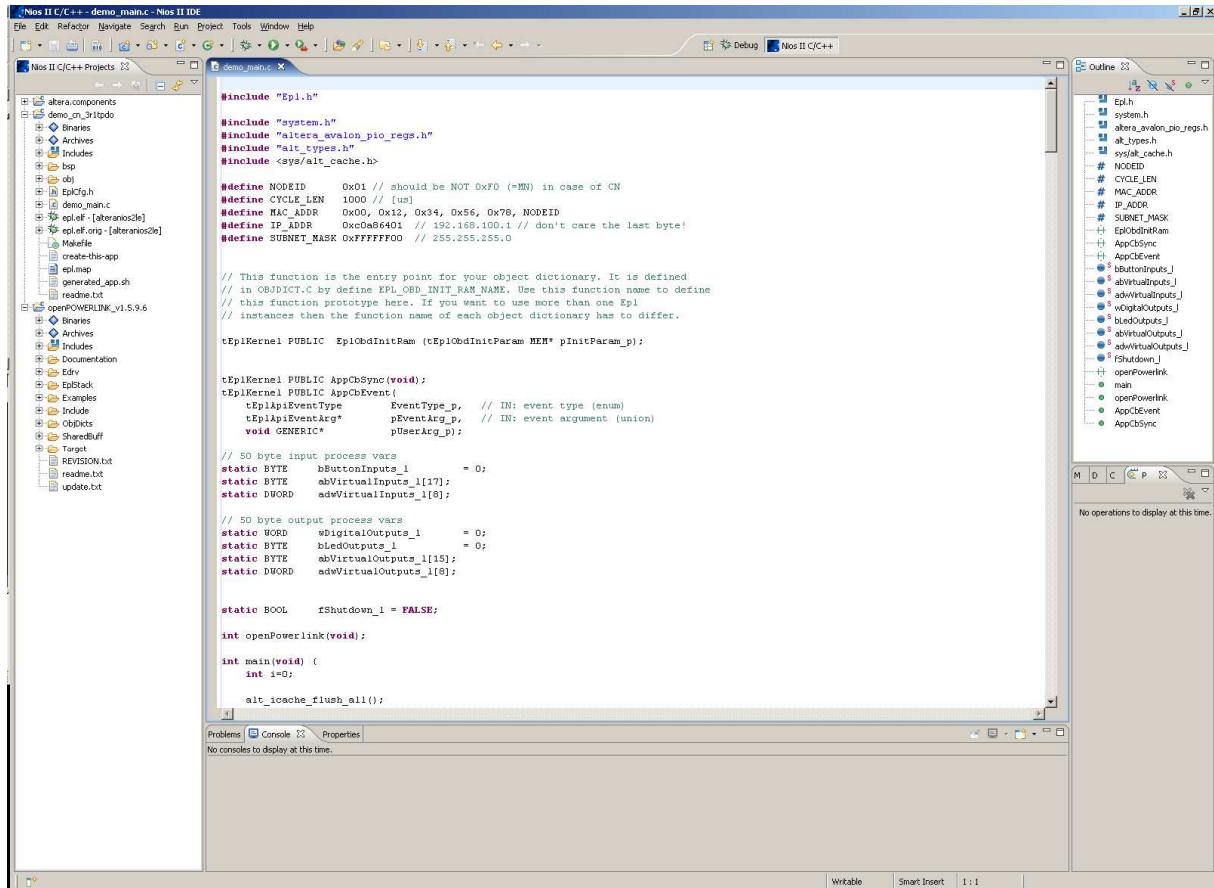


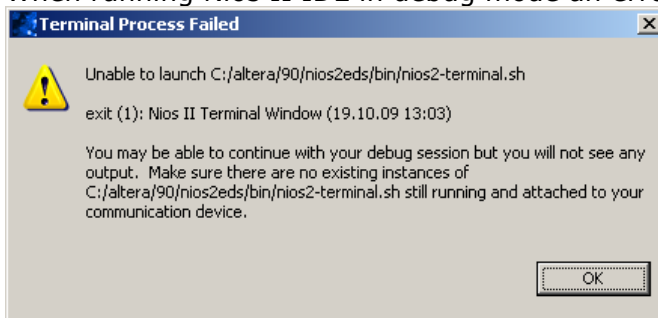
Figure 7-5: Nios II IDE with application example

8 Troubleshooting

This chapter comprises a collection of possible errors that may occur when running through this user guide.

"!" Trouble
 "→" Solution

! When running Nios II IDE in debug mode an error message appears.



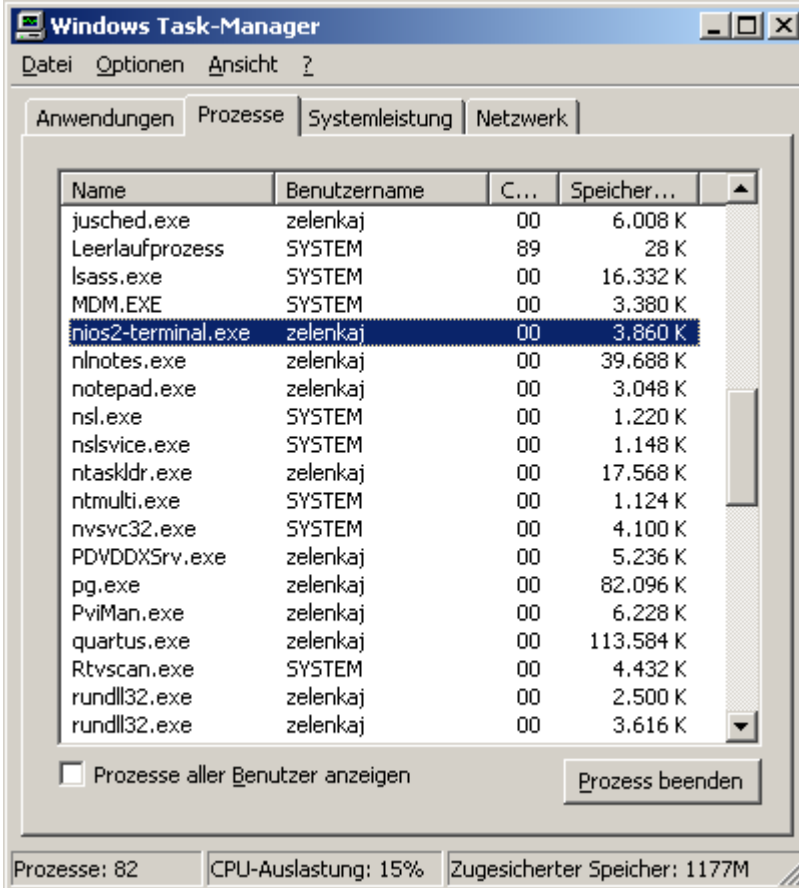
Problems Console Properties

```
<terminated> demo_cn_3r1tpdo Nios II HW configuration [Nios II Hardware] Nios II Terminal Window (19.10.09 13:03)
nios2-terminal: Unable to connect to JTAG UART because another application
nios2-terminal: (nios2-terminal on 127.0.0.1)
nios2-terminal: is using it (or is using something else which clashes). Please c
lose
nios2-terminal: the other application and try again.
```

→ An other Nios II Terminals (e.g. in a Command Shell or IDE) is running and connected to your target. Close the other terminal session!

! A Nios II Terminal is running (e.g. in IDE) but it is not possible to stop this session.

→ Open the Task Manager and stop the task "nios2-terminal.exe".



! When stepping through the application source (debug mode in IDE), the instructions won't be executed line by line.

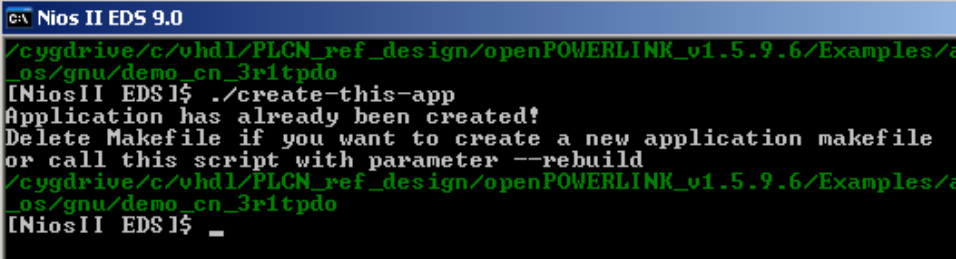
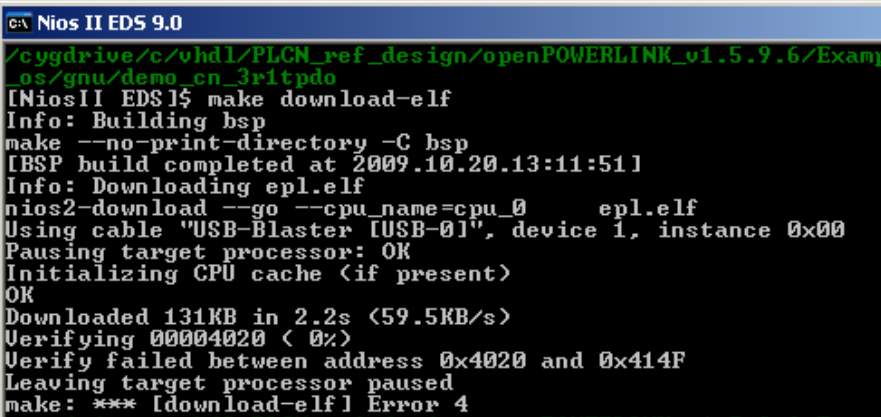
→ The compiler optimization level is unequal none. You need to create the Makefile again with the argument --debug (refer to Table 7-1).

! When compiling the reference design with Quartus II, the following error is reported.

Type	Message
Info	Elaborated megafunction instantiation "lpm_ff0:inst6\lpm_ff:lpm_ff_component"
Info	Instantiated megafunction "lpm_ff0:inst6\lpm_ff:lpm_ff_component" with the following parameter:
Error	Node instance "the_clock_crossing_0" instantiates undefined entity "clock_crossing_0"
Error	Node instance "the_niosII_openMac_clock_0" instantiates undefined entity "niosII_openMac_clock_0"
Error	Node instance "the_niosII_openMac_clock_1" instantiates undefined entity "niosII_openMac_clock_1"
Error	Node instance "the_niosII_openMac_clock_2" instantiates undefined entity "niosII_openMac_clock_2"
Error	Node instance "the_niosII_openMac_clock_3" instantiates undefined entity "niosII_openMac_clock_3"

→ The delivered Quartus II Project is incomplete to minimize the package's size. So, temporary files had been deleted!
Open SOPC and generate the Nios II system, to compile the Quartus II Project successfully.

! When trying to program the FPGA with the Quartus II Programmer the following error (for further information refer to Altera Support sources) is shown:

	<table border="1"> <thead> <tr> <th>Type</th><th>Message</th></tr> </thead> <tbody> <tr> <td>✖</td><td>Error: File "stpl.stp" is not a valid SignalTap II File -- ignoring</td></tr> <tr> <td>ℹ</td><td>Info: Enable SignalTap II - Command: quartus_stp nios_openMac --enable --signaltap --stp_file=stpl.stp</td></tr> <tr> <td>ℹ</td><td>Info: Started Full Compilation at Mon Oct 19 13:48:23 2009 Westeuropäische Normalzeit</td></tr> <tr> <td>ℹ</td><td>Info: Ended Full Compilation at Mon Oct 19 13:52:11 2009 Westeuropäische Normalzeit</td></tr> <tr> <td>ℹ</td><td>Info: Started Programmer operation at Mon Oct 19 13:52:17 2009</td></tr> <tr> <td>✖</td><td>Error: Application nios2-terminal on 127.0.0.1 is using the target device</td></tr> <tr> <td>✖</td><td>Error: Operation failed</td></tr> <tr> <td>ℹ</td><td>Info: Ended Programmer operation at Mon Oct 19 13:52:17 2009</td></tr> </tbody> </table>	Type	Message	✖	Error: File "stpl.stp" is not a valid SignalTap II File -- ignoring	ℹ	Info: Enable SignalTap II - Command: quartus_stp nios_openMac --enable --signaltap --stp_file=stpl.stp	ℹ	Info: Started Full Compilation at Mon Oct 19 13:48:23 2009 Westeuropäische Normalzeit	ℹ	Info: Ended Full Compilation at Mon Oct 19 13:52:11 2009 Westeuropäische Normalzeit	ℹ	Info: Started Programmer operation at Mon Oct 19 13:52:17 2009	✖	Error: Application nios2-terminal on 127.0.0.1 is using the target device	✖	Error: Operation failed	ℹ	Info: Ended Programmer operation at Mon Oct 19 13:52:17 2009
Type	Message																		
✖	Error: File "stpl.stp" is not a valid SignalTap II File -- ignoring																		
ℹ	Info: Enable SignalTap II - Command: quartus_stp nios_openMac --enable --signaltap --stp_file=stpl.stp																		
ℹ	Info: Started Full Compilation at Mon Oct 19 13:48:23 2009 Westeuropäische Normalzeit																		
ℹ	Info: Ended Full Compilation at Mon Oct 19 13:52:11 2009 Westeuropäische Normalzeit																		
ℹ	Info: Started Programmer operation at Mon Oct 19 13:52:17 2009																		
✖	Error: Application nios2-terminal on 127.0.0.1 is using the target device																		
✖	Error: Operation failed																		
ℹ	Info: Ended Programmer operation at Mon Oct 19 13:52:17 2009																		
→	A Nios II Terminal is running and blocking the access to the FPGA. Close this connection or terminate the "nios2-terminal.exe" task in the Task Manager.																		
!	When trying to build the Makefile an error is shown in the command shell. 																		
→	The Makefile was already built, so you can only rebuild the Makefile with the appropriate argument (refer to Table 7-1).																		
!	When downloading the application the Nios II CPU the following error appears (for further information refer to Altera Support sources). 																		
→	The Nios II CPU on the FPGA is not the CPU from the reference design. Reprogram your FPGA (refer to 5.3).																		