Lime Microsystems Limited

Surrey Tech Centre
Occam Road
The Surrey Research Park
Guildford GU2 7YG
Surrey
United Kingdom

Tel: +44 (0) 1483 685 063
Fax: +44 (0) 1428 656 662
e-mail: enquiries@limemicro.com

LimeSDR-PCIE

- User Guide -

Chip version: -
Chip revision: 0
Document version: 1.0
Document revision: 00
Last modified: 14/02/2017 10:57:24 AM

Contents

Revision History

Version 1.0r00
Started:13 February, 2017
Finished: 14 February, 2017
Initial version

1

Introduction

Document consists of a chapters that describes steps to generate and download required files for LimeSDR-PCIE_lms7_trx project. Guidelines on how to compile, generate and program FPGA bitstream files are also provided.

In case when pre compiled bitstream file is used for board programming proceed straight to 4.2 Board programming procedure chapter.

Xillybus documentation [1], [2], [3], can be found in links provided in this document reference. PCIE Drivers can be downloaded from link [4]

Quartus prime version used with this guide: 15.1.2 Build 193 02/01/2016 SJ Lite Edition
Quartus Prime Lite Edition software can be downloaded from link [5]

2

Xillybus core generation

Xillybus core has to be generated before downloading. This chapter describes steps and parameters required to generate Xillybus PCIE core.

2.1 Signing Up

Xillybus requires to fill up free registration form in order to download generated core. Go to link http://xillybus.com/ipfactory/signup, fill required fields (Figure 1) and confirm registration via received eMail.

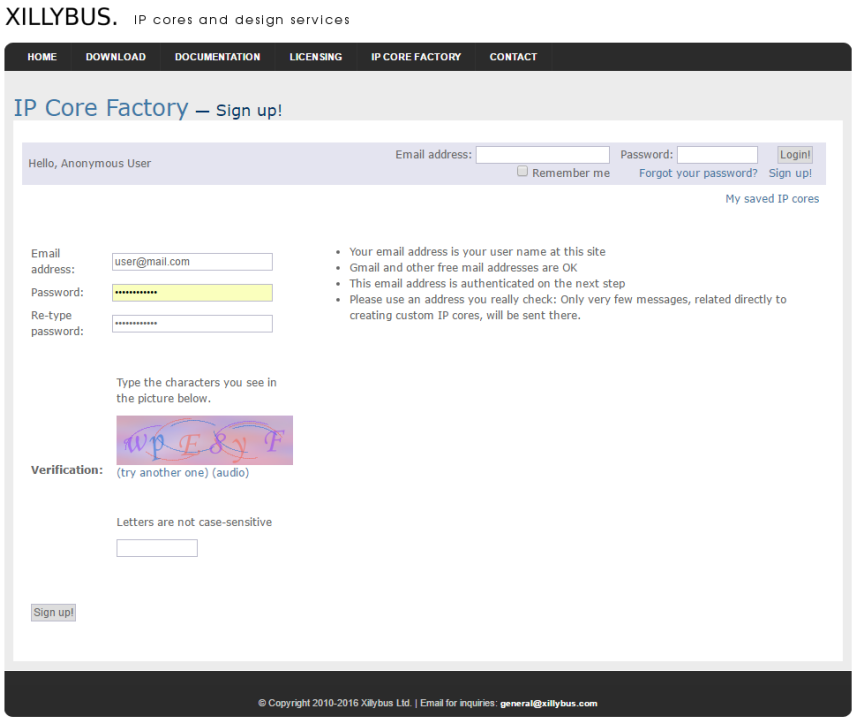

Figure 1 Registration form

2.2 Creating new IP core

After successful registration, go to IP core Factory page http://xillybus.com/ipfactory/, fill parameters as shown in Figure 2 and click *Create!*.

Figure 2 Create new IP core dialog

2.3 Setting core parameters

After new core creation in next dialog click *Edit* to change settings for each device files (Figure 3).

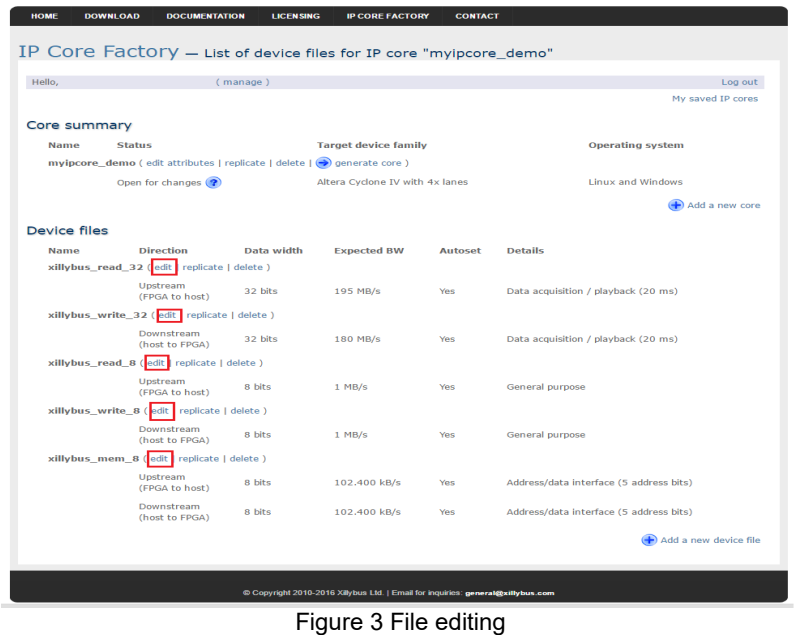

Figure 3 File editing

In *Edit* dialog (Figure 4) fill following parameters for corresponding file and click *Update!*. To enter all parameters *Autoset internals* has to be unchecked:

For xillybus_read_32:

- Device file's name - read_32
- Direction - Upstream (FPGA to host)
- Use - Data acquisition / playback
- Data width - 32 bits
- Expected bandwidth - 195
- Autoset internals - unchecked
- Asynchronous/synchronous - Asynchronous
- Number of buffers - 256
- Size of each buffer - 16 kB

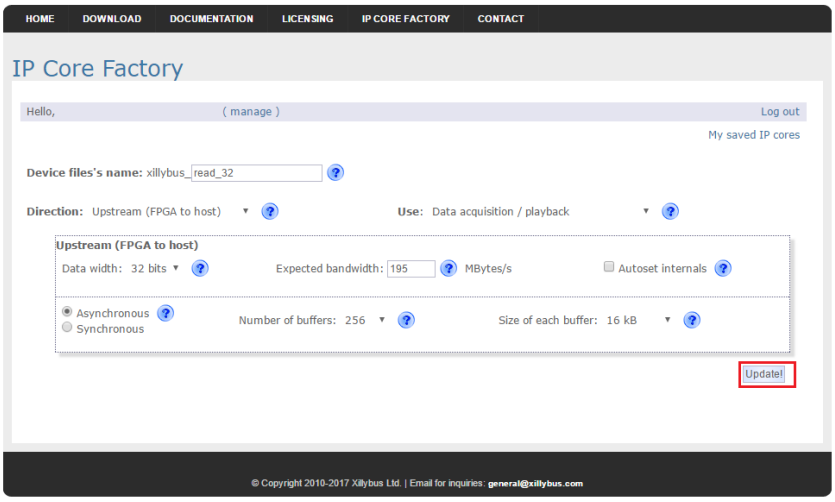

Figure 4 xillybus_read_32 file editing

Edit rest of the files with following parameters:

For xillybus_write_32:

- Device file's name - write_32
- Direction - Downstream (host to FPGA)
- Use - Data acquisition / playback
- Data width - 32 bits
- Expected bandwidth - 180
- Autoset internals - unchecked
- Asynchronous/synchronous - Asynchronous
- Number of buffers - 256
- Size of each buffer - 16 kB
- DMA acceleration - None

For xillybus_read_8:

- Device file's name - read_8
- Direction - Upstream (FPGA to host)
- Use - General purpose
- Data width - 8 bits
- Expected bandwidth - 1

- Autoset internals - unchecked
- Asynchronous/synchronous - Asynchronous
- Number of buffers - 4
- Size of each buffer - 4 kB

For xillybus_write_8:

- Device file's name - write_8
- Direction - Downstream (host to FPGA)
- Use - General purpose
- Data width - 8 bits
- Expected bandwidth - 1
- Autoset internals - unchecked
- Asynchronous/synchronous - Asynchronous
- Number of buffers - 4
- Size of each buffer - 4 kB
- DMA acceleration - None

For xillybus_mem_8:

- Device file's name - mem_8
- Direction - Bidirectional
- Use - Address/data interface (5 address bits)

  Upstream (FPGA to host)
  - Data width - 8 bits
  - Expected bandwidth - 0.1
  - Autoset internals - unchecked
  - Asynchronous/synchronous - Synchronous
  - Number of buffers - 4
  - Size of each buffer - 4kb

  Downstream (host to FPGA)
  - Data width - 8 bits
  - Expected bandwidth - 0.1
  - Autoset internals - unchecked
  - Asynchronous/synchronous - Synchronous
  - Number of buffers - 4
  - Size of each buffer - 4kb
  - DMA acceleration - None

After updating all files click *generate core* (Figure 5). Check core status and download it when available (Figure 6).

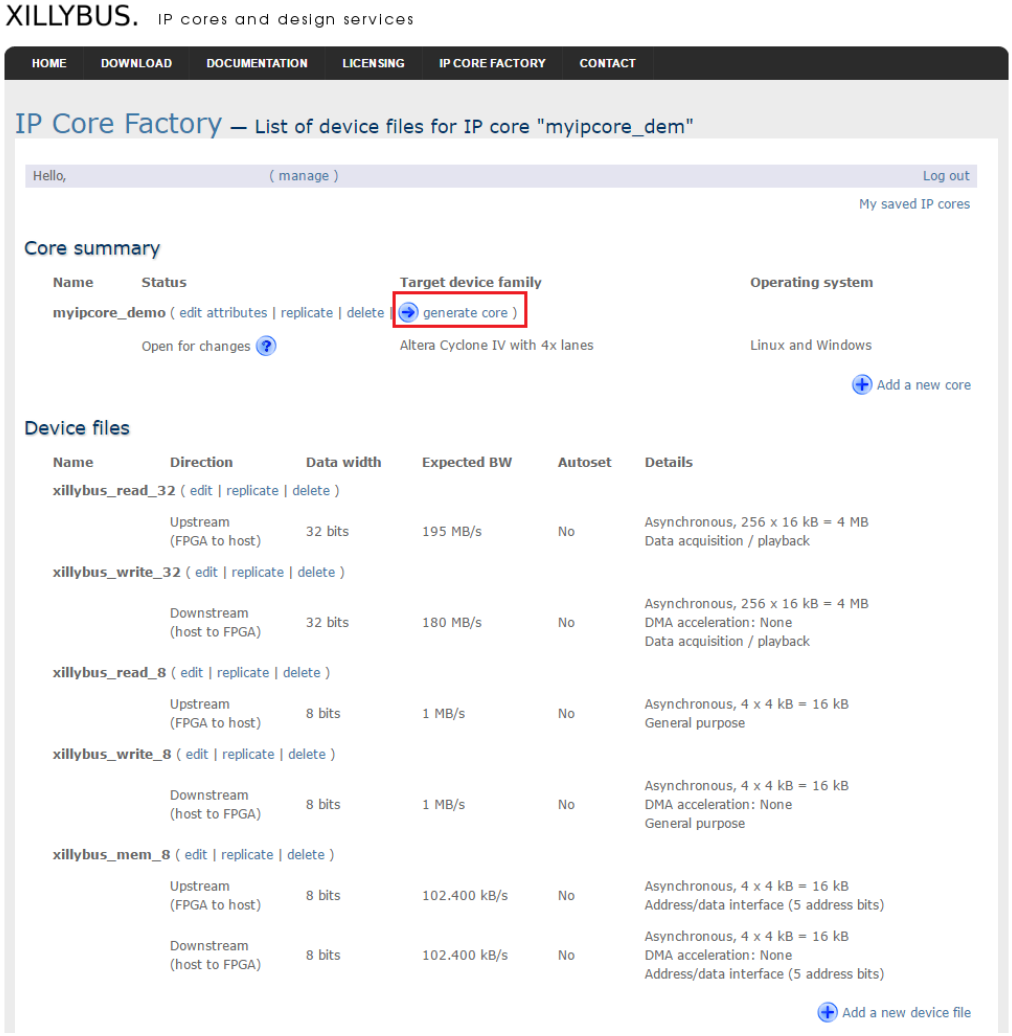

Figure 5 Core generation

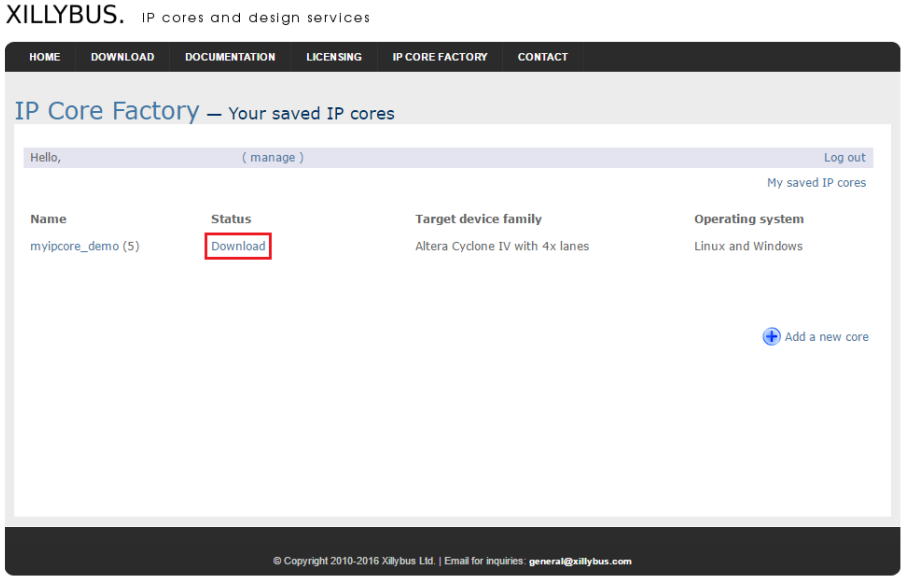

Figure 6 Download status

3

Quartus project files

This chapter describes steps to include Xillybus core to Quartus project.

3.1 Quartus files

Steps to include Xillybus core files into Quartus project.

- Extract downloaded .zip file "c*orebundle-myipcore_demo.zip*" (*myipcore_demo* – name that was entered during core generation).
- Place file *xillybus.v* to Quartus project directory limesdr-pcie_xillybus_core/
- Place file *xillybus_core.qxp* to Quartus project directory limesdr-pcie_xillybus_core/
- Open Quartus *LimeSDR-PCIE_lms7_trx* project and select *Project→ Add/Remove Files in Project..* and add files *xillybus.v* and *xillybus_core.qxp* to Quartus project (Figure 7).
- Recompile project *Processing → Start Compilation*.

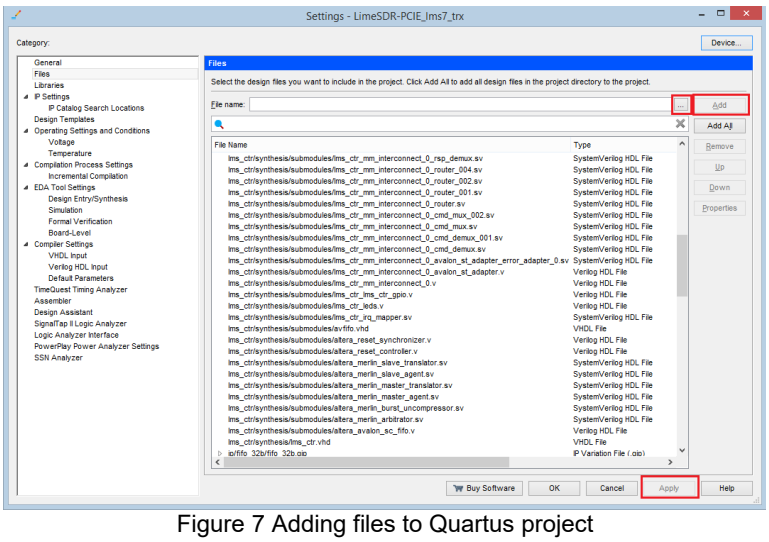

Figure 7 Adding files to Quartus project

4

Board programming

This chapter describes steps to program LimeSDR-PCIe board. If you already have valid programming ***.jic** file skip 4.1 Obtaining programming files and proceed to 4.2 Board programming procedure.

4.1 Obtaining programming files

After performing full project compilation *Processing → Start Compilation* in Messages window (see Figure 8) should appear messages stating that programming files are created:

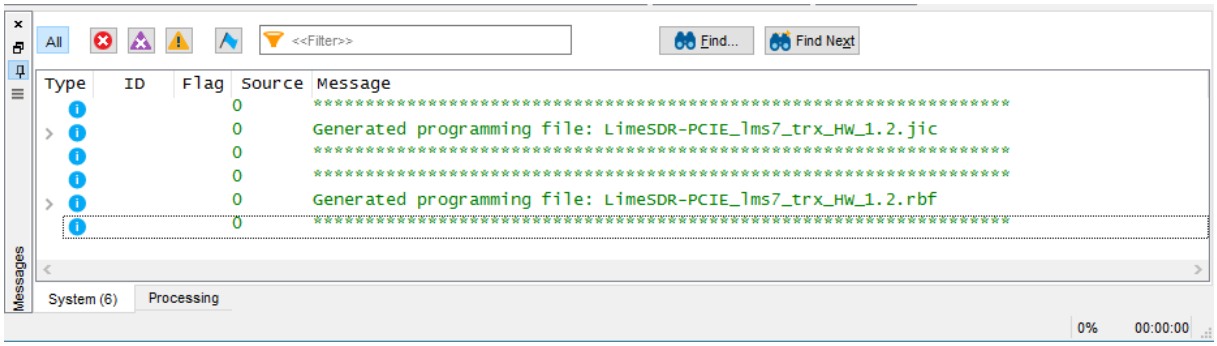

Figure 8 Project compilation message window

Programming files can be found in folder *output_files* from project directory:

***.jic** - JTAG Indirect Configuration File can be used to program FPGA gateware to FLASH memory (if valid file is loaded FPGA boots from FLASH when board power is applied automatically).

***.sof** - SRAM Object File can be used to program FPGA (has to be programmed every time after board power is applied)

***.rbf** - Raw Binary File can be used to program FPGA gateware into FLASH memory through LimeSuiteGUI (valid gateware has to be already running, see 4.2 Board programming)

4.2 Board programming procedure

For the first time use board has to be programmed using JTAG header J11. This procedure requires two computers (LimeSDR-PCIe board inserted into PCIe slot on computer #1 and Quartus Prime software running on computer #2).

- Insert LimeSDR-PCIe board into computer #1. Make sure that computer is turned off while inserting board.
- Connect one end of download cable (e.g Altera USB Blaster) to LimeSDR-PCIe board J11 connector and other end to USB port on the computer #2 running Quartus Prime software.
- Turn on computer #1 and interrupt the boot sequence to bring up the BIOS System Setup interface.
- Run Quartus Prime software in computer #2 and select *Tools → Programmer*
- Click *Hardware Setup..* button and select your download cable, click *Close* (see Figure 9).

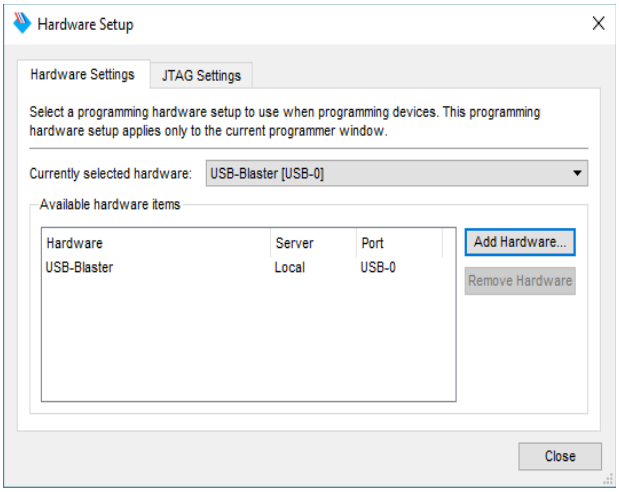

Figure 9 Selecting programming hardware

- Click *Add File..* and select **.jic* file (see options below):
  a) Pre compiled bitstream can be found in *gateware/LimeSDR-PCIE_lms7_trx_bs*
  b) If you have followed project compilation instructions and generated your own bitstream then your file is located in project directory */output_files.*
- Apply settings as in Figure 10 and click *Start*.

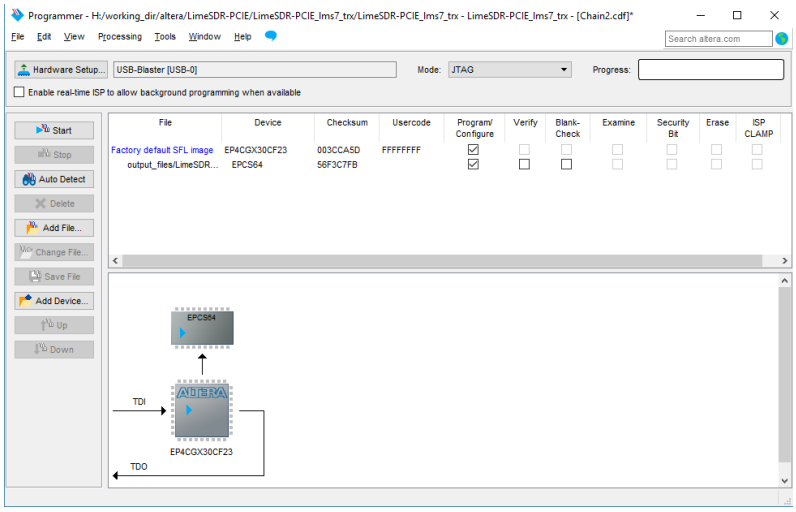

Figure 10 Adding programming file

- After successful programming turn off computer #1.
- FPGA boots from programmed FLASH memory automatically when computer #1 is turned on.

5

References

1. Xillybus documentation files: http://xillybus.com/doc
2. Getting started with Xillybus on a Windows host: http://xillybus.com/downloads/doc/xillybus_getting_started_windows.pdf
3. Getting started with Xillybus on a Linux host: http://xillybus.com/downloads/doc/xillybus_getting_started_linux.pdf
4. Xillybus drivers for Windows 7: http://xillybus.com/downloads/xillybus-windriver-1.2.0.0.zip
5. Quartus Prime Lite Edition download site: http://dl.altera.com/15.1/?edition=lite