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

Xillybus core generation for LimeSDR-PCIE board

- Guide -

Chip version:	-
Chip revision:	0
Document version:	1.0
Document revision:	00
Last modified:	03/06/2016 05:10:38 PM

Contents

1. Introduction.....3

2. Xillybus core generation.....4

2.1Signing Up.....4

2.2Creating new IP core.....5

2.3Setting core parameters.....5

3. Quartus project files.....9

3.1Quartus files.....9

4. References.....10

Revision History

Version 1.0r00
Started:02 June, 2016
Finished: 02 June, 2016
Initial version

1

Introduction

This document describes Xillybus core generation and instantiation for LimeSDR-PCIE board. Document consists of a chapters that describes steps to generate and download required files for LimeSDR-PCIE_lms7_trx project.

Xillybus documentation [1], [2], [3], can be found in links provided in this document reference.

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.

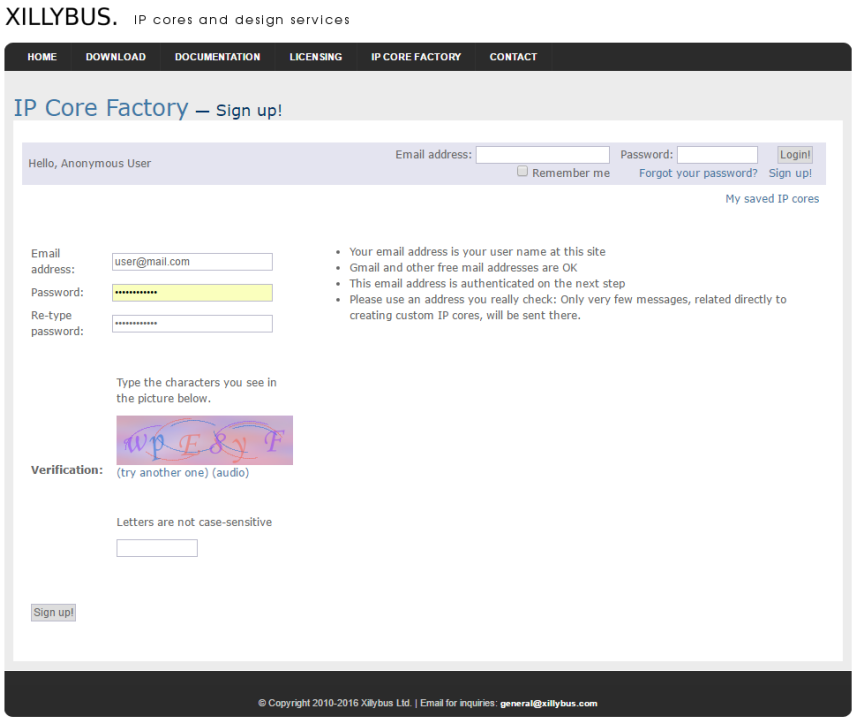

Figure 1 Registration form

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).

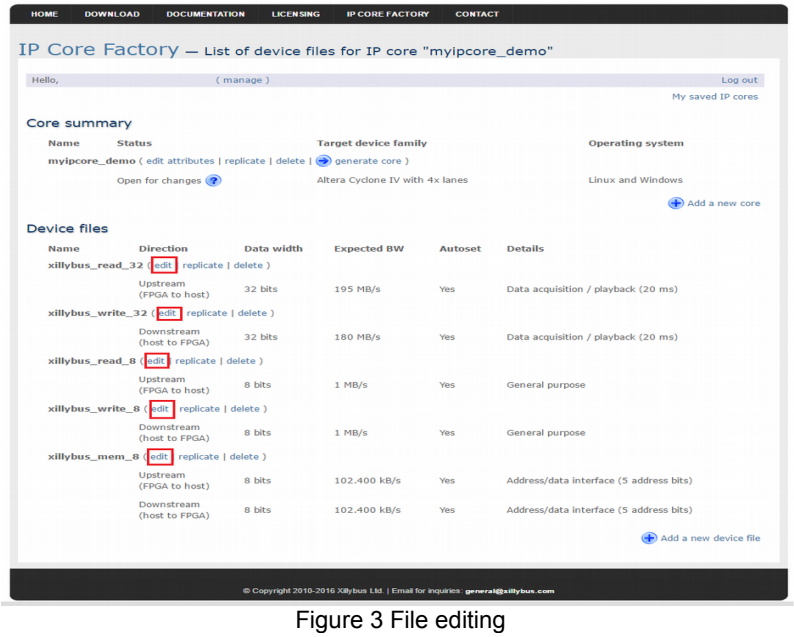

Figure 3 File editing

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 acquision / playback
- Data width - 32 bits
- Expected bandwidth - 195
- Autoset internals - unchecked
- Asynchronous/synchronous - Asynchronous
- Number of buffers - 256
- Size of each buffer - 8 kB

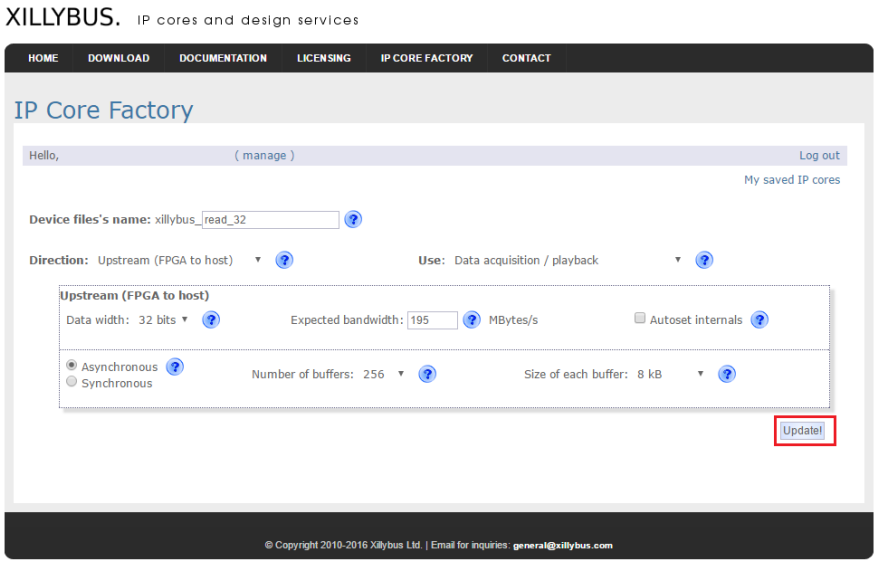

Figure 4 xillybus_read_32 file editing

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 acquision / playback
- Data width - 32 bits
- Expected bandwidth - 180
- Autoset internals - unchecked
- Asynchronous/synchronous - Asynchronous
- Number of buffers - 256
- Size of each buffer - 8 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).

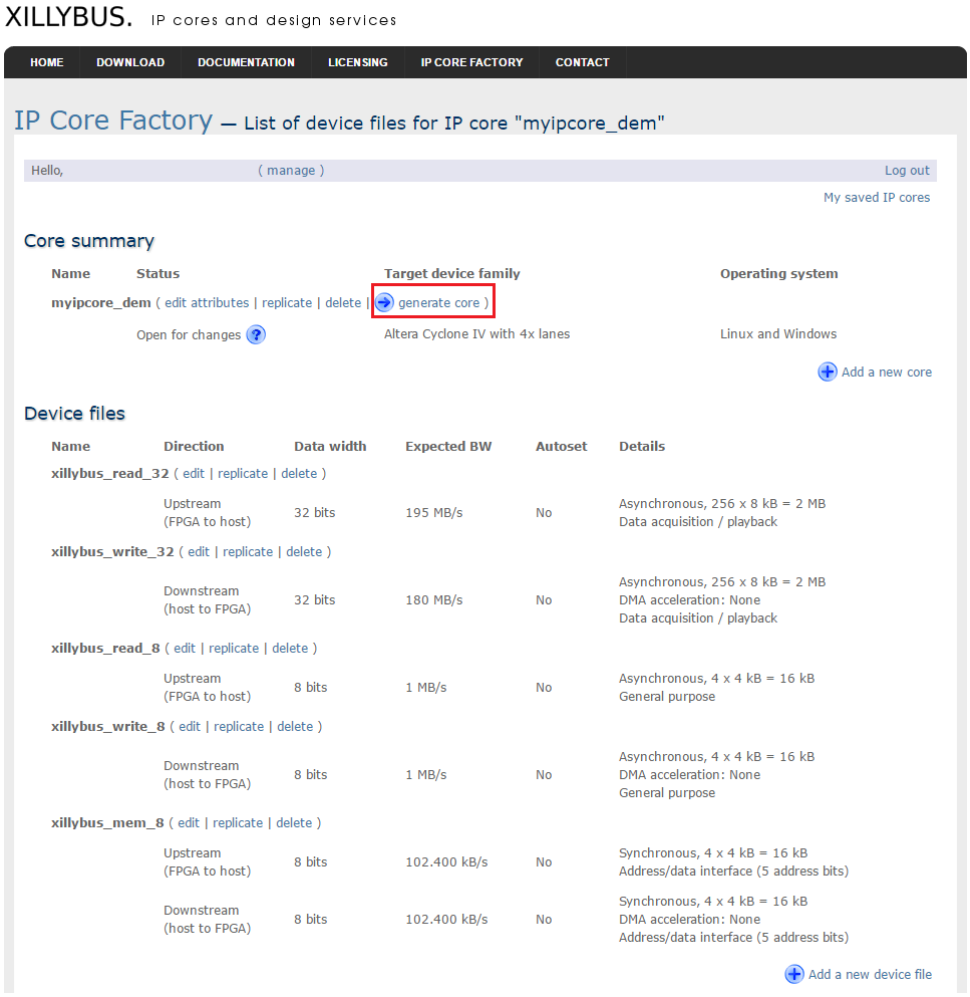

Figure 5 Core generation

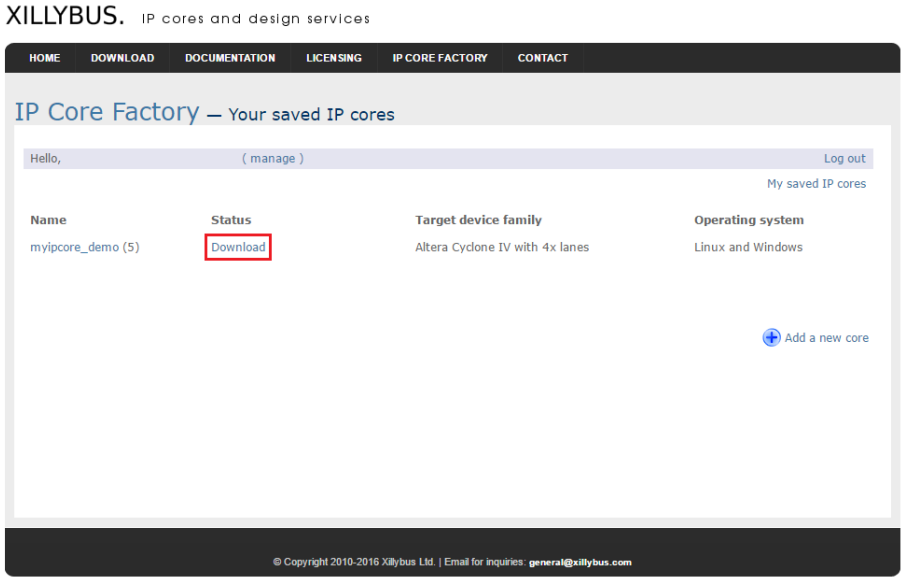

Figure 6 Download status

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.

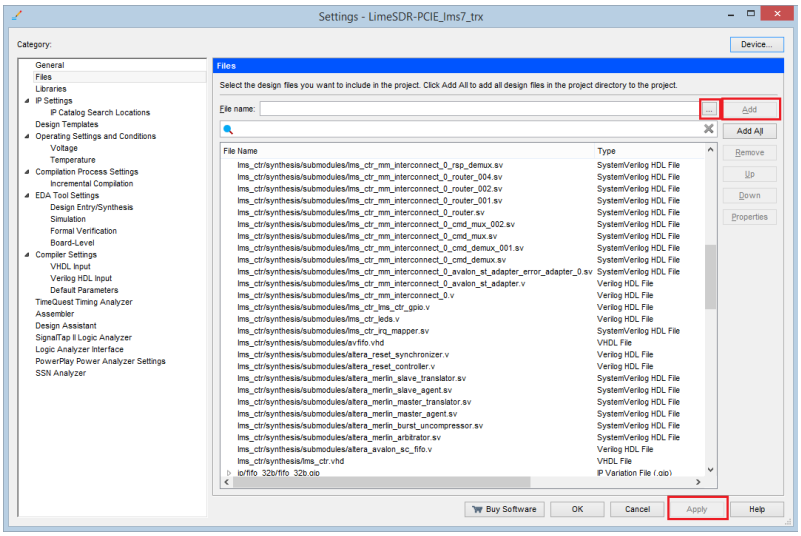

Figure 7 Adding files to Quartus project

4

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