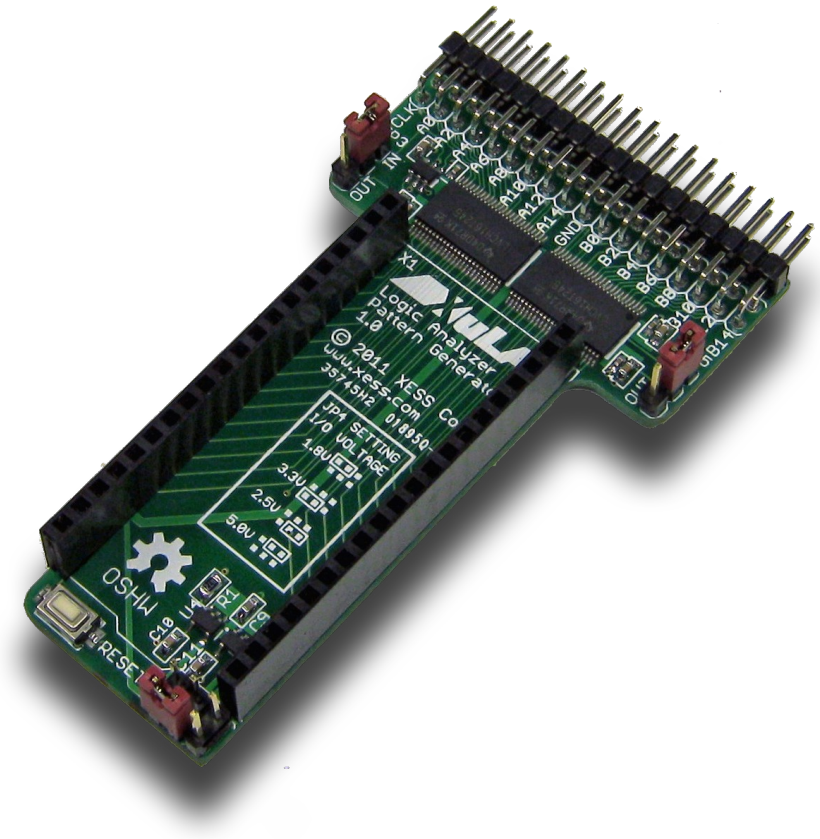
XESS Corporation

Logic Pod Manual

How to install and use your new Logic Pod Module



XESS is disclosing this Document and Intellectual Property (hereinafter “the Design”) to you for use in the development of designs to operate on, or interface with XESS hardware devices. XESS expressly disclaims any liability arising out of the application or use of the Design. XESS reserves the right to make changes, at any time, to the Design as deemed desirable in the sole discretion of XESS. XESS assumes no obligation to correct any errors contained herein or to advise you of any correction if such be made. XESS will not assume any liability for the accuracy or correctness of any engineering or technical support or assistance provided to you in connection with the Design.

THE DESIGN IS PROVIDED “AS IS” WITH ALL FAULTS, AND THE ENTIRE RISK AS TO ITS FUNCTION AND IMPLEMENTATION IS WITH YOU. YOU ACKNOWLEDGE AND AGREE THAT YOU HAVE NOT RELIED ON ANY ORAL OR WRITTEN INFORMATION OR ADVICE, WHETHER GIVEN BY XESS, OR ITS AGENTS OR EMPLOYEES. XESS MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE DESIGN, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NONINFRINGEMENT OF THIRD-PARTY RIGHTS.
IN NO EVENT WILL XESS BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, INCLUDING ANY LOST DATA AND LOST PROFITS, ARISING FROM OR RELATING TO YOUR USE OF THE DESIGN, EVEN IF YOU HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE TOTAL CUMULATIVE LIABILITY OF XESS IN CONNECTION WITH YOUR USE OF THE DESIGN, WHETHER IN CONTRACT OR TORT OR OTHERWISE, WILL IN NO EVENT EXCEED THE AMOUNT OF FEES PAID BY YOU TO XESS HEREUNDER FOR USE OF THE DESIGN. YOU ACKNOWLEDGE THAT THE FEES, IF ANY, REFLECT THE ALLOCATION OF RISK SET FORTH IN THIS AGREEMENT AND THAT XESS WOULD NOT MAKE AVAILABLE THE DESIGN TO YOU WITHOUT THESE LIMITATIONS OF LIABILITY.

The Design is not designed or intended for use in the development of on-line control equipment in hazardous environments requiring fail-safe controls, such as in the operation of nuclear facilities, aircraft navigation or communications systems, air traffic control, life support, or weapons systems (“High-Risk Applications”). XESS specifically disclaims any express or implied warranties of fitness for such High-Risk Applications. You represent that use of the Design in such High-Risk Applications is fully at your risk.

Logic Pod Manual
MAN??? (V0.1) October 24, 2013

The following table shows the revision history for this document.

Date	Version	Revision
10/24/2013	0.1	Preliminary release for Logic Pod module V1.0.

Table of Contents

C.1 Preliminaries

Here's some helpful information before getting started.

Getting Help!

Here are some places to get help if you encounter problems:

- If you can't get the Logic Pod module to work, send an e-mail message describing your problem to help@xess.com or submit a problem report at http://www.xess.com/help.php.
- Our web site also has
  - answers to frequently-asked-questions,
  - example designs, application notes and tutorials,
  - a forum where you can post questions.

Take Notice!

It's pretty hard to get in trouble with this module. Just don't insert the XuLA or XuLA2 board backwards; the USB connector should be facing away from the Logic Pod I/O header.

Packing List

Here is what you should have received in your package:

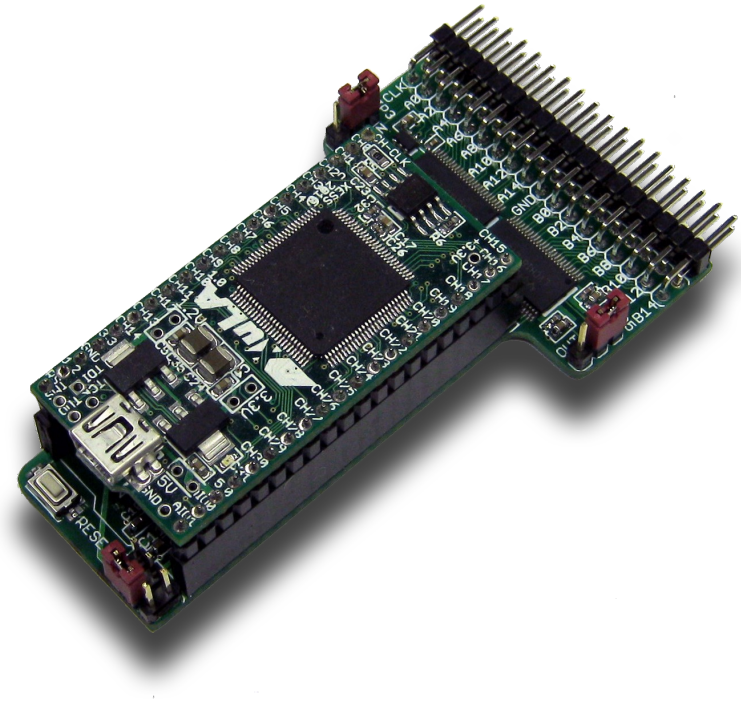
- a Logic Pod module.

C.2 Setup

The Logic Pod module provides an eight-position DIP switch that connects to an eight-bit PMOD or a Wing socket on your StickIt! board.

Inserting Your XuLA/XuLA2 into Your Logic Pod Module

To use the Logic Pod module, insert a XuLA or XuLA2 board as shown below. (**To insure a stable connection, only use headers on the XuLA/XuLA2 with 0.025" square pins.**)

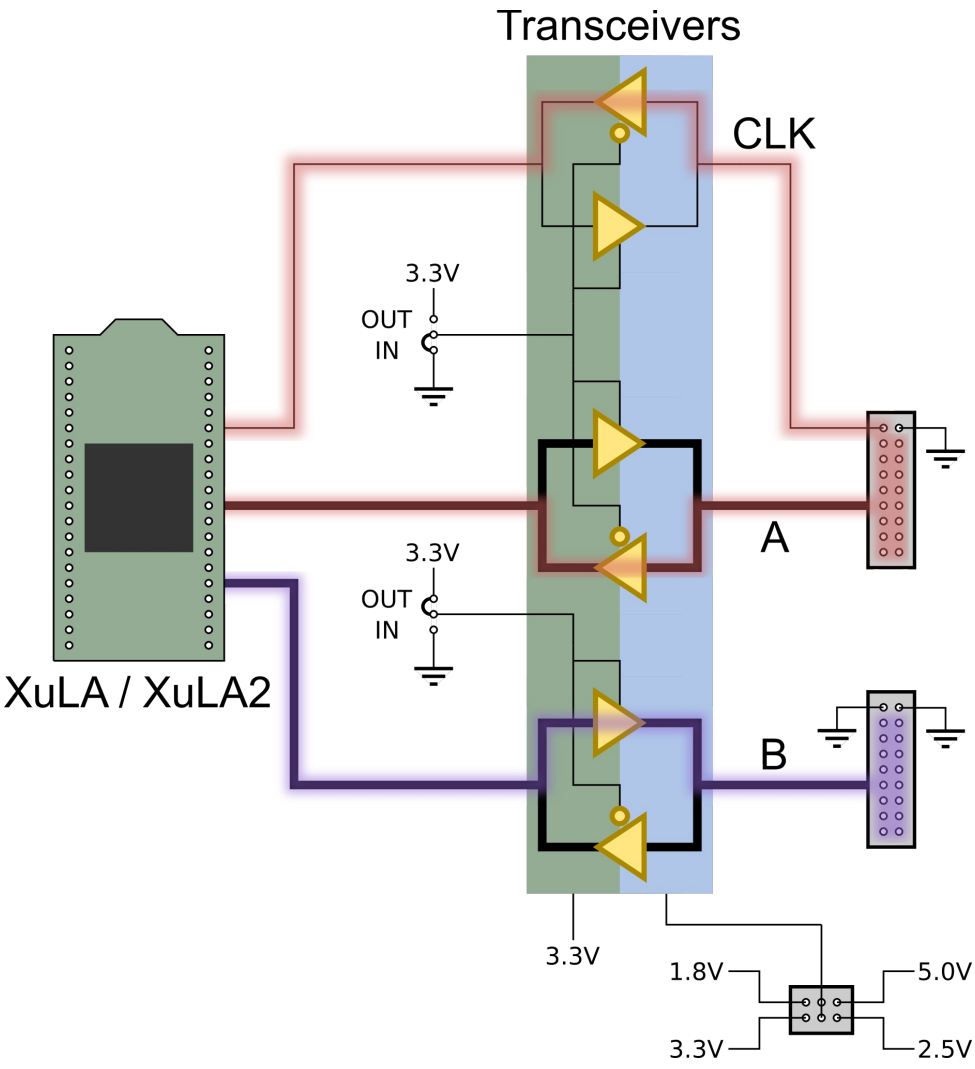


Setting the Jumpers

Jumper	Function
JP3	Setting this jumper to the OUT position lets the FPGA drive signals onto the 16-bit A and CLK header pins. Setting it to IN lets the FPGA receive signals driven through the A and CLK header pins by an external source.
JP5	Setting this jumper to the OUT position lets the FPGA drive signals onto the 16-bit B header pins. Setting it to IN lets the FPGA receive signals driven through the B header pins by an external source.
JP4	This jumper configures the voltage translators between the XuLA I/O pins and the CLK, A and B pins. The voltage can be set to 1.8V, 2.5V, 3.3V or 5.0V. The voltage setting only affects the logic levels on the CLK, A and B pins; because of the voltage translators, the XuLA board will always receive 3.3V logic signals.

C.3 Operation

This chapter describes the operation of the Logic Pod module using a simplified schematic. You can find a complete schematic at the end of this manual.



The Logic Pod module uses transceivers to translate between the 3.3V I/O of the XuLA board and one of four possible I/O voltages for the external system (1.8V, 2.5V, 3.3V, 5.0V). There are two banks of transceivers: one bank that handles the 16-bit A port and the CLK, and a second bank that handles the 16-bit B port. Each bank can be independently set by a jumper to input signals from the external system to the XuLA board or output signals from the XuLA board to the external system. This allows the Logic Pod to be used as a 32-bit logic analyzer plus clock, a 32-bit pattern generator plus clock, or a combination 16-bit logic analyzer / 16-bit pattern generator plus clock.

C.4 Using the Module

To use the Logic Pod module, you will need to do the following:

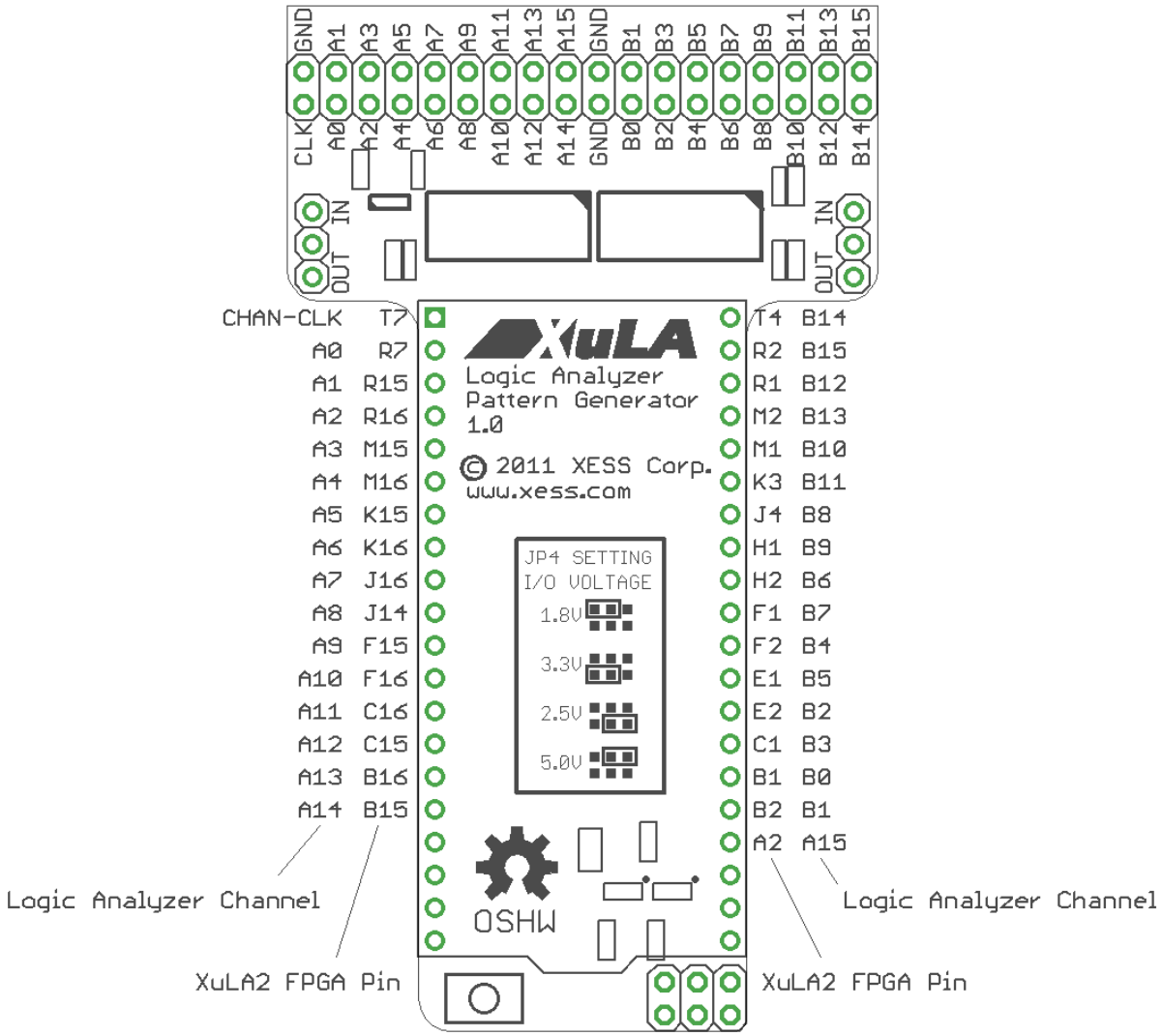
- Create a Xilinx ISE FPGA project and allocate up to 33 I/O ports to connect to the signals from the module.
- Attach the module to either a XuLA or XuLA2 board.
- Determine the FPGA pins that connect to each I/O pin of the module.
- Make a UCF file associating each FPGA pin with an I/O pin of the module.
- Include the UCF file in your ISE project.

That's a lot of work. I know that, otherwise I would have done it already. But I have done a simple project that tests the connectivity of the A and B ports. Just go to https://github.com/xesscorp/XuLA-Logic-Analyzer. There, you will find a subdirectory with a Xilinx ISE project that includes:

- a test design that uses the HostIoToDut module to drive the A port with a 16-bit value received from the PC over the USB link while also reading the 16-bit value on the B port,
- a UCF file containing the XuLA2 FPGA pin assignments for the Logic Pod module I/O,
- and a Python file that sets the A port to a random 16-bit value and then reads and compares the value on the B port.

A.1 I/O Locations

The connections from the XuLA2 to the Logic Pod module are shown below.



A.2 Schematic