LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
-LIGO-

LIGO Laboratory /LIGO Scientific Collaboration

| | | |
|---|---|---|
| **Technical Document** | **LIGO-T1200372-v1** | **7/26/12** |

# FPGA Programming Guide

Stefan Countryman
*Columbia University*

*Distribution of this document:*

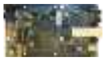
This is an internal working note
of the LIGO Project.

**LIGO Hanford Observatory**
**P.O.Box. 1970**
**Mail Stop S9-02**
**Richland,WA 99352**
Phone (509) 372-8106
Fax (509) 372-8137

**LIGO Livingston Observatory**
**P.O.Box. 940**
**Livingston, LA 70754**
Phone (225) 686-3100
Fax (225) 686-7189

**Columbia University**
**Columbia Astrophysics Laboratory**
**Pupin Hall - MS 5247**
**New York NY 10027**
Phone (212) 854-8209
Fax (212) 854-8121
E-mail: geco.cu@gmail.com

WWW: http://www.ligo.caltech.edu



# Programming FPGAs

This document provides step-by-step instructions for programming the Flash PROM found on Advanced LIGO timing devices. The aLIGO Timing System relies on modular devices, each of which contains a field programmable gate array (FPGA) and programmable read-only memory (Flash PROM). The FPGA's configuration is initialized on startup according to the data stored on the Flash PROM. If this data is nonexistent or out of date (e.g. the FPGA code has been updated, the device is blank from the factory, etc.), the Flash PROM must be programmed for the device to work. This is accomplished through the JTAG interface present on every Timing Master Rear Panel and Timing Slave Board. The most efficient programming method, described here, makes use of Xilinx iMPACT software and the Digilent XUP USB-JTAG Programming Adapter. Procurement and installation of these tools is described in Appendix I. Alternate JTAG cables are discussed in Appendix II. Xilinx iMPACT configuration files are discussed in Appendix III. An alternate programming method relying on Altium Designer and the Altium USB-JTAG Adapter is outlined in Appendix IV.

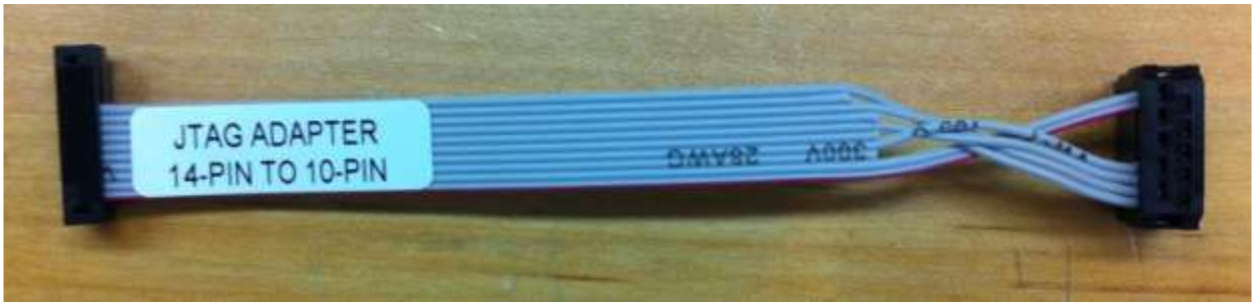
## 1. Download Configuration Files

1. Up-to-date FPGA code can be downloaded from the Timing FPGA page of the aLIGO Timing Wiki. At the top of the page, find the module type to be programmed and download the corresponding .iaf[1] file from the link provided. For example, this would be E1200032-v1.iaf for the Master Fanout version 1 code.

## 2. Power-up and connecting to the JTAG adapter

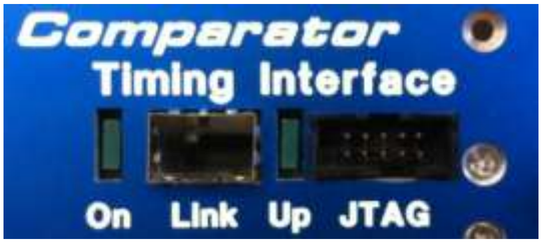
1. Connect device to a 12V power supply.

2. Connect the XUP USB-JTAG Adapter to the computer. Attach the 14-pin (2mm 2x7) end of the 14-pin to 10-pin JTAG Adapter (pictured below) to the XUP Adapter:



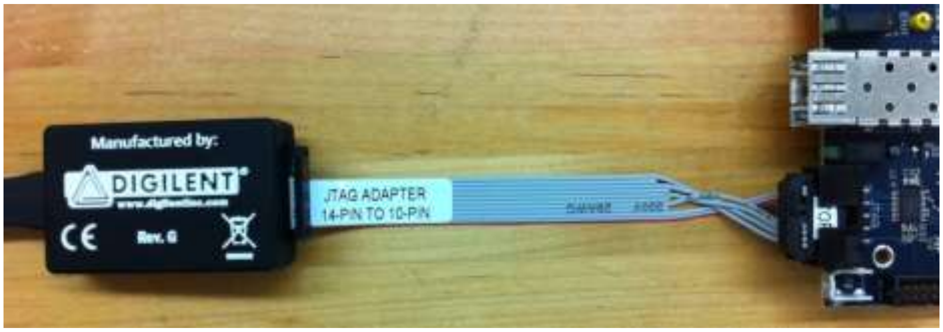
[1] This is a Xilinx iMPACT configuration archive, which is all that is needed for the programming technique here described. The true FPGA machine code comes in the form of .bit (FPGA) and .mcs (Flash PROM) files, which are compressed in the .iaf archive. They are also separately available on the Timing Wiki. For instructions on creating an iMPACT project starting from the .bit and .mcs files, see Appendix III. The files themselves are stored in the LIGO Document Control Center (DCC). All Timing System FPGA codes, as well as other Timing documents, are archived in DCC.

3. Locate the device's 10-pin JTAG port. The port location varies by device, but all are externally accessible. On the Timing Comparator, the JTAG port is in the center of the front panel:

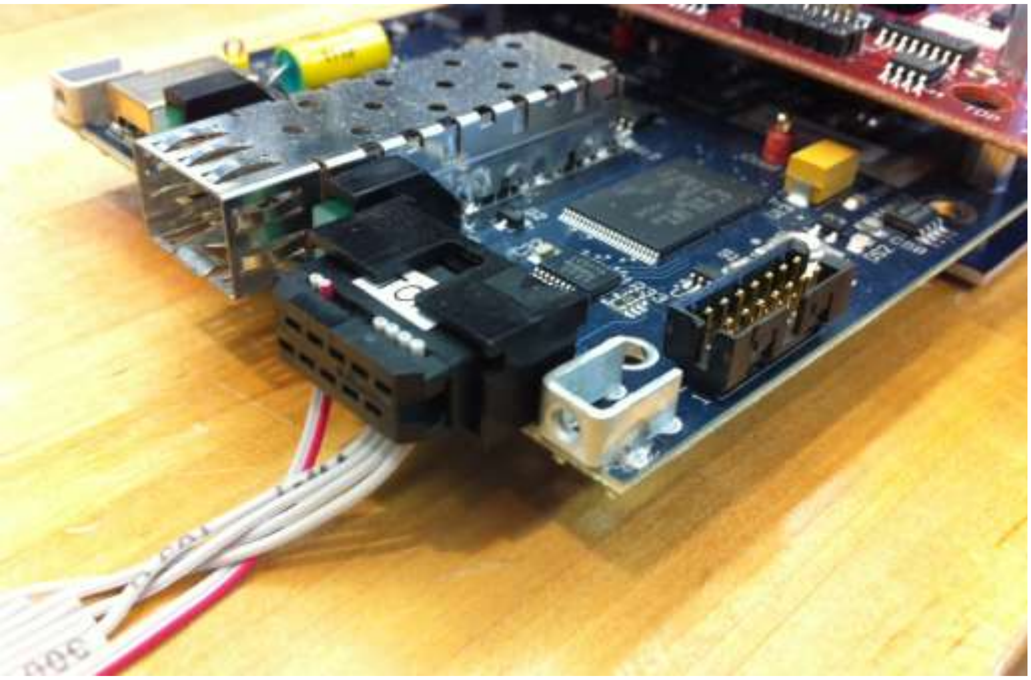


*10-pin (2.54mm 2x5) JTAG port*

Connect the 10-pin connector to the device's JTAG port. The top of the adapter is labeled for clarity. A properly connected Timing Slave-DuoTone assembly would appear as follows:



*Top view of adapter: 14-pin side of adapter, left, connected to XUP USB-JTAG Adapter*



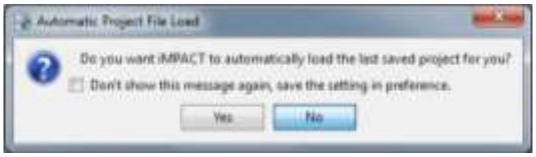
*Front view of Slave-DuoTone assembly: 10-pin side of adapter connected to device's JTAG port*

4. Open iMPACT. In the Start menu, go to:

*Programs > Xilinx ISE Design Suite 12.2 > ISE Design Tools > 64-bit Tools > iMPACT*

5. Dialogue Boxes

Press “No” in the *Automatic Project File Load* dialogue box:



Automatic Project File Load

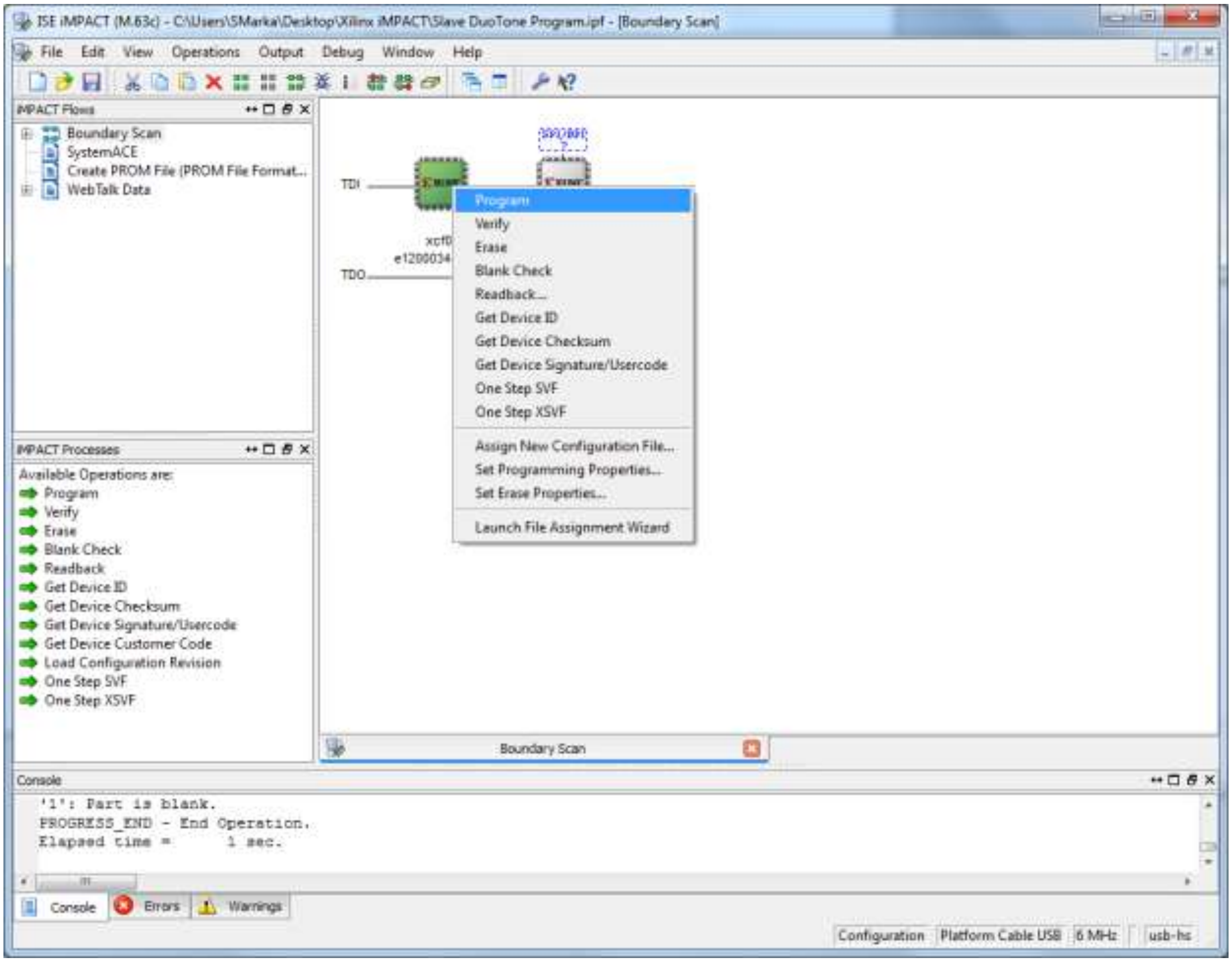
Do you want iMPACT to automatically load the last saved project for you?

□ Don't show this message again, save the setting in preference.

Press “Cancel” in the *New iMPACT Project* dialogue box.

6. Open the configuration archive. Go to *File* > *Open Configuration Archive* (ctrl-H). Select the .iaf file downloaded in Step 1.1. Choose a location for iMPACT to unzip the contents of the archive; any will do. (A warning dialogue box might pop up; if it does, dismiss it and continue.)

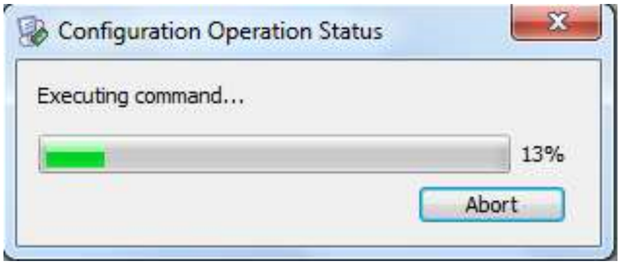
7. Program the Flash PROM. In the main window, right-click on the left-most device and click *Program*:



ISE iMPACT (M.63c) - C:\Users\SMarka\Desktop\Xilinx iMPACT\Slave DuoTone Program.ipf - [Boundary Scan]

```
'1': Part is blank.
PROGRESS_END - End Operation.
Elapsed time =      1 sec.
```

Programming should take a few minutes. Its status is displayed via progress bar:

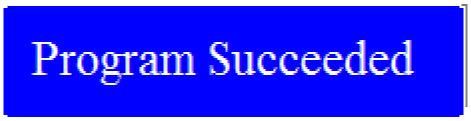


Configuration Operation Status

Executing command...

13%

Once complete, a blue box saying “Program Succeeded” should appear at the bottom of the main window:



Program Succeeded

8. After the Flash PROM is programmed, the code is generally loaded onto the FPGA automatically. If the device does not demonstrate proper operation within several moments, the FPGA can be reset by interrupting the device’s power supply (12V). When power is restored, the FPGA will be automatically reprogrammed from the Flash PROM.

The device is now ready to use.

## Appendix I: Setting up iMPACT and the USB-JTAG Adapter

1. Xilinx iMPACT software is used to program JTAG devices. It is part of Xilinx ISE Design Suite. The aLIGO Timing System uses ISE Design Suite version 12.2, available for free download on Xilinx’s website. Select the Full Installer for Windows (2.96 GB). Once it has finished downloading, decompress it, run setup, and do a full WebPACK installation. Though certain FPGA design components cost money, a free WebPACK license can be created and downloaded off of Xilinx’s website. This can be done by choosing to open the “Manage Xilinx Licenses” application at the end of the WebPACK installation, or at any time thereafter by going to *Start > Xilinx ISE Design Suite 12.2 > Accessories > Manage Xilinx Licenses*. The free WebPACK license is all that is required to use iMPACT. Once Xilinx ISE Design Suite 12.2 has been installed and a license has been downloaded, iMPACT is ready to use.

2. The Digilent XUP USB-JTAG Programming Adapter (Part number XUP-USB-JTAG on Digilent's website) allows iMPACT to connect to a JTAG device using a USB port. The XUP USB-JTAG Adapter is a slightly cheaper, cosmetically different OEM version of Xilinx’s own Platform Cable USB. Consequently, they share functionality, drivers, and documentation. Xilinx WebPACK version 12.2 has the appropriate drivers preinstalled, so installing the XUP Adapter consists merely of plugging it in. The adapter itself has a USB connection on one side (which should be connected to the computer via an included cable) and a 14-pin (2mm 2x7) JTAG connection on the other:

*Digilent XUP USB-JTAG Adapter*

*XUP Adapter, 14-pin (2mm 2x7) JTAG port*

Use the aLIGO Timing 14-pin to 10-pin JTAG Cable to connect the XUP Adapter to the 10-pin external JTAG ports, as in Step 2.2. If the 14-pin to 10-pin JTAG Cable is not available, see Appendix II for instructions on alternate methods for connecting the XUP Adapter to aLIGO Timing devices. If an XUP Adapter is not available, but an Altium USB-JTAG Adapter is, see Appendix IV for instructions on programming via Altium.

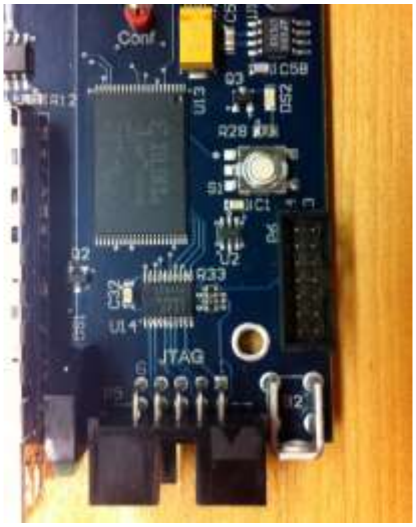
## Appendix II: Connecting to JTAG using 14-pin or Flying Wire Adapter

If a 14-pin to 10-pin JTAG Cable is not available, there are two alternate methods for programming the Flash PROM using the XUP Adapter. Each uses stock cables that are packaged with the XUP Adapter.

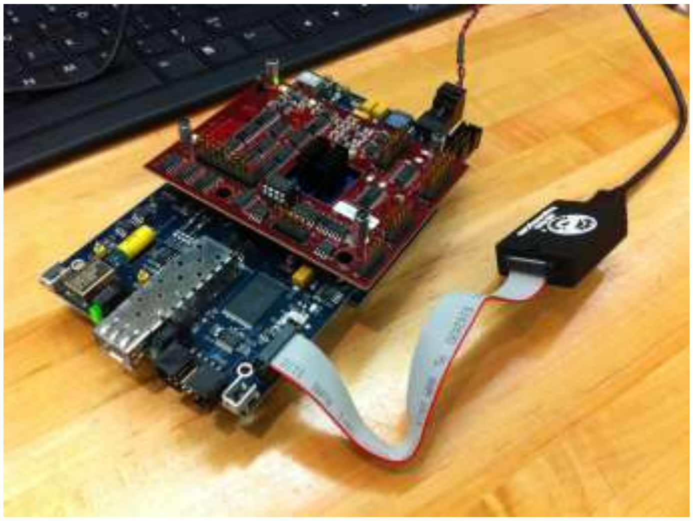
1. The first (and easier) of the two programming methods uses an included ribbon cable with 14-pin (2mm 2x7) IDC connectors at both ends:

One end of this cable connects to the 14-pin connector on the XUP adapter. The other end connects to a 14-pin connecter located on the device to be programmed. Unfortunately, the 14-pin connector is not externally accessible on aLIGO Timing devices, so this method requires disassembly of the device in question. If this is inconvenient, proceed to Appendix II, Part 2.

On the Timing Slave, the 14-pin connector is header P6. It is located directly behind the 10-pin external JTAG port:

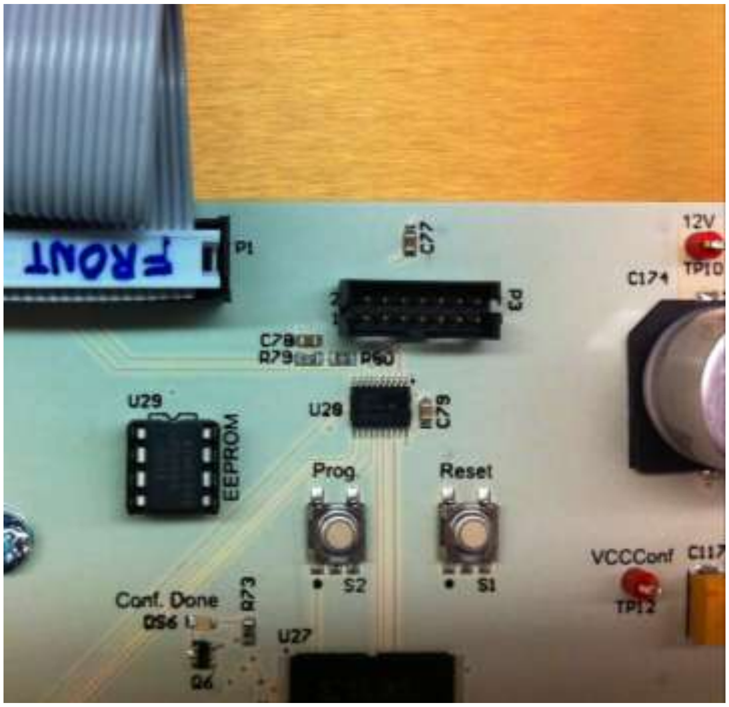


*14-pin JTAG, header P6*

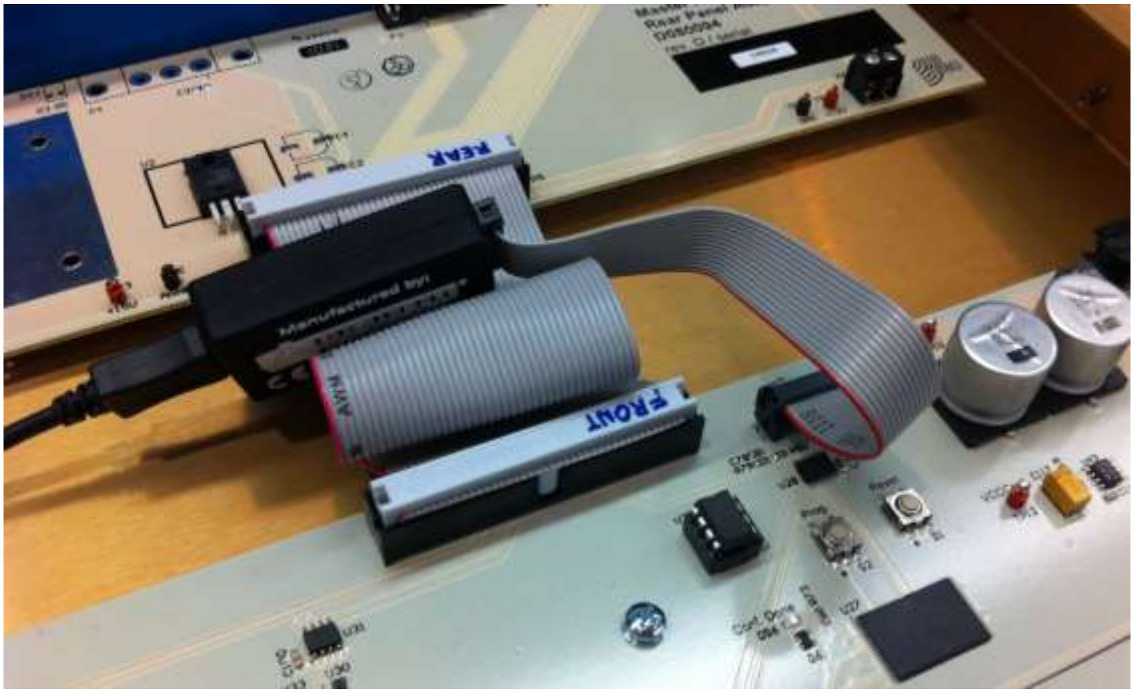


*Slave-DuoTone assembly connected via 14-pin to XUP*

On the Timing Master, the 14-pin connector is header P3. It is located at the rear of the front board, adjacent to the header for the ribbon connector:



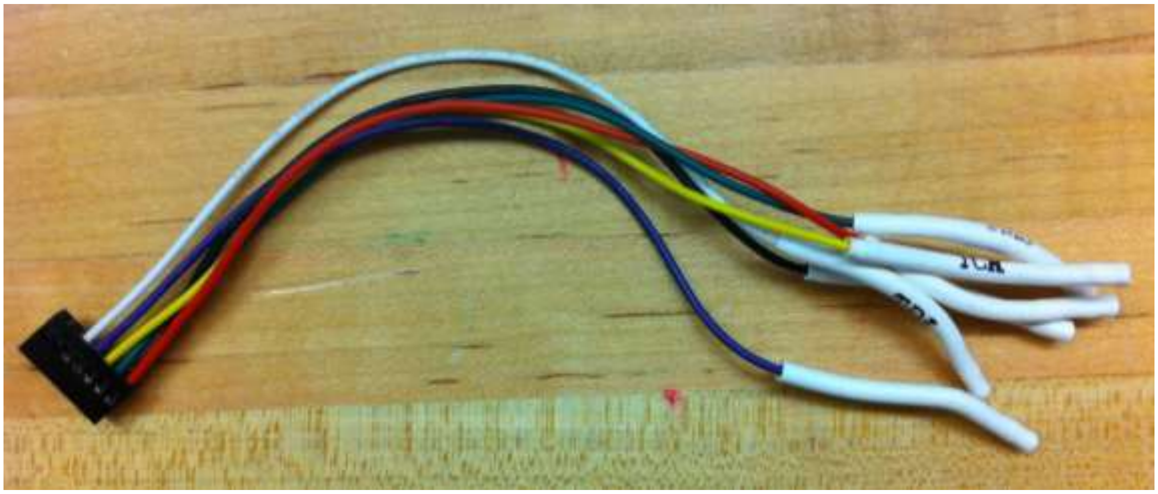
*14-pin JTAG, header P3, with surrounding features*



*Timing Master connected via 14-pin to XUP; view from front of chassis*

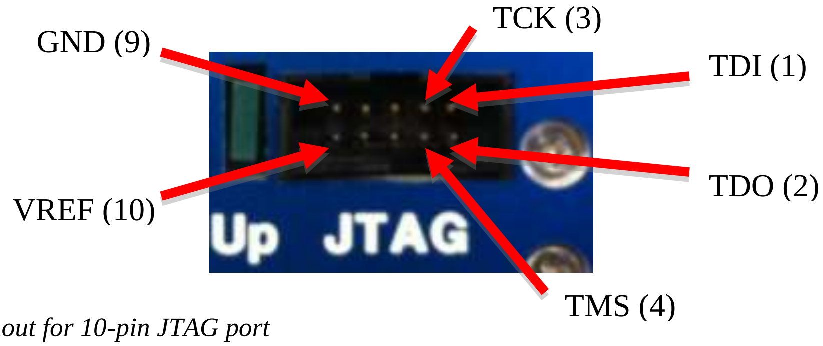
Once the connection has been made and the device has been connected to a 12V power source, programming can proceed as usual.

2. If a 14-pin to 10-pin adapter is not available, the XUP adapter can still be connected to the device's 10-pin JTAG port using the included flying wire adapter:



*JTAG flying wire adapter; XUP 14-pin connector on left, leads on right*

The flying wire adapter provides one wire for each of the five JTAG signals (TDI, DTO, TCK, TMS, and VREF/VCCAUX) as well as a ground wire. The signal carried by each wire is clearly labeled on the white lead. Connect the flying leads to the 10-pin port:



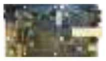
*Pinout for 10-pin JTAG port*

*10-pin JTAG port with flying wire adapter connected*

Align the leads with the square cross-section pins. Press firmly—the leads are thick and will get crowded. After connecting the leads, proceed with programming as usual.

3. A table of signal pinouts for the 10-pin Altium JTAG (used by aLIGO Timing devices) and the 14-pin Xilinx JTAG (used by the Digilent XUP Adapter) is provided for reference:

| **Signal Name** | **10-pin (Altium) pin number** | **14-pin (Xilinx) pin number** |
|---|---|---|
| TDI | 1 | 10 |
| TDO | 2 | 8 |
| TCK | 3 | 6 |
| TMS | 4 | 4 |
| Software TDI | 5 | — |
| Software TDO | 6 | — |
| Software TCK | 7 | — |
| Software TMS | 8 | — |
| GND/Ground | 9 | 1, 3, 5, 7, 9, 11, 13 |
| VREF/VCCAUX/Detect | 10 | 2 |

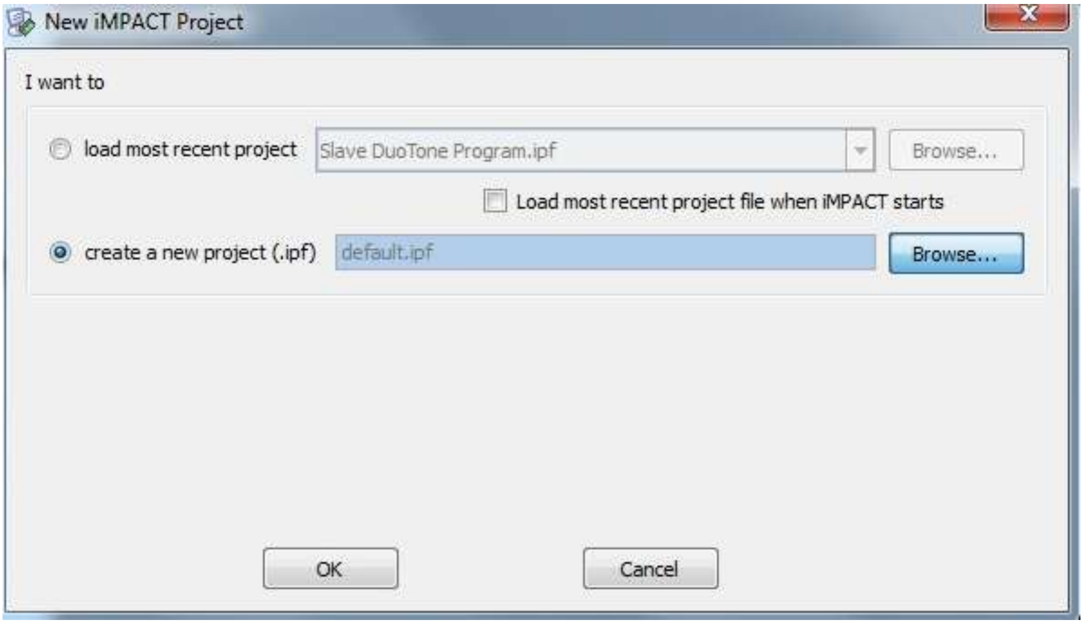


## Appendix III: Creating an iMPACT project using .mcs and .bit files

If a configuration file is not available, or if nonstandard code needs to be loaded onto the Flash PROM, it is possible to create a new iMPACT project file from existing binary code.

1. Connect the XUP adapter to the computer.

2. Open iMPACT. In the opening dialogue box, select "create a new project (.ipf)" and press "OK" (the project title and save location do not matter):



New iMPACT Project

I want to

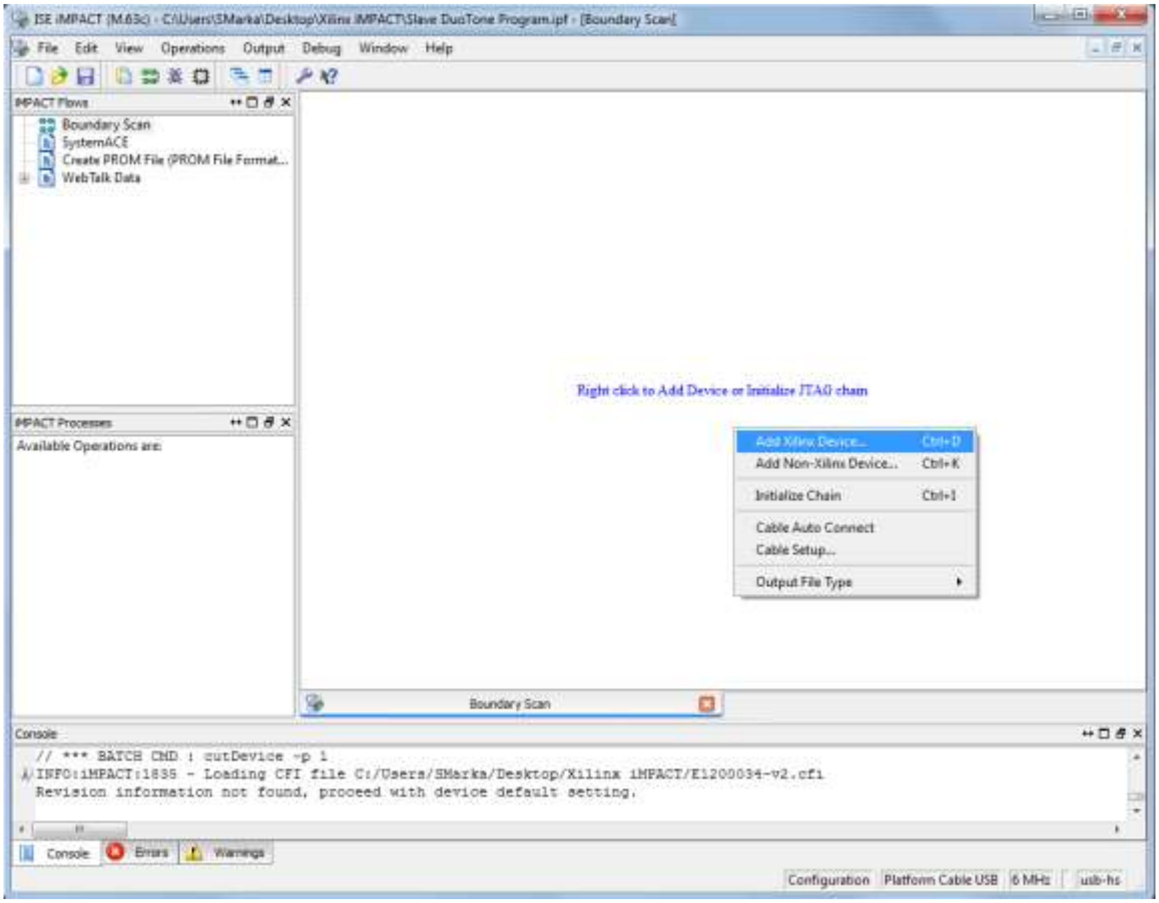
○ load most recent project: Slave DuoTone Program.ipf

□ Load most recent project file when iMPACT starts

● create a new project (.ipf): default.ipf

3. In the "Welcome to iMPACT" window, select "Configure devices using Boundary-Scan (JTAG)" and press "OK". A dialogue box will warn: "There are many unknown devices being detected." Press no in this dialogue box. An error will open and a red box will appear warning "Identify Failed". This is harmless.

4. Right-click anywhere in the main window and select "Add Xilinx Device..."

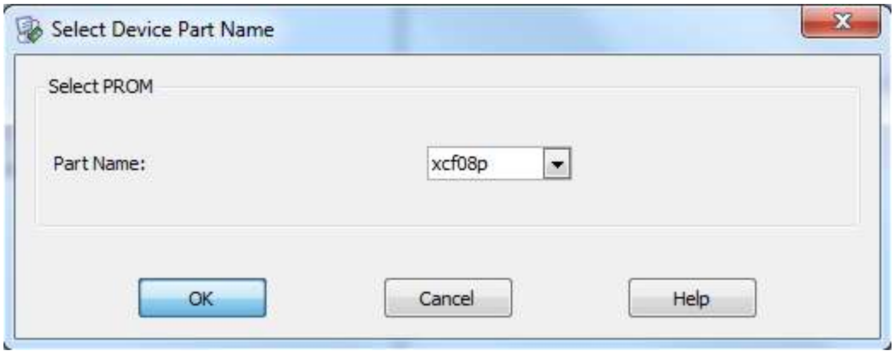


ISE iMPACT (M.63c) - C:\Users\SMarka\Desktop\Xilinx iMPACT\Slave DuoTone Program.ipf - [Boundary Scan]

```
// *** BATCH CMD : cutDevice -p 1
INFO:iMPACT:1835 - Loading CFI file C:/Users/SMarka/Desktop/Xilinx iMPACT/E1200034-v2.cfi
Revision information not found, proceed with device default setting.
```

Find and select the .mcs file you wish to load onto the Flash PROM.

5. A dialogue box will open asking for a part name. Select xcf08p and click "OK":



Select Device Part Name

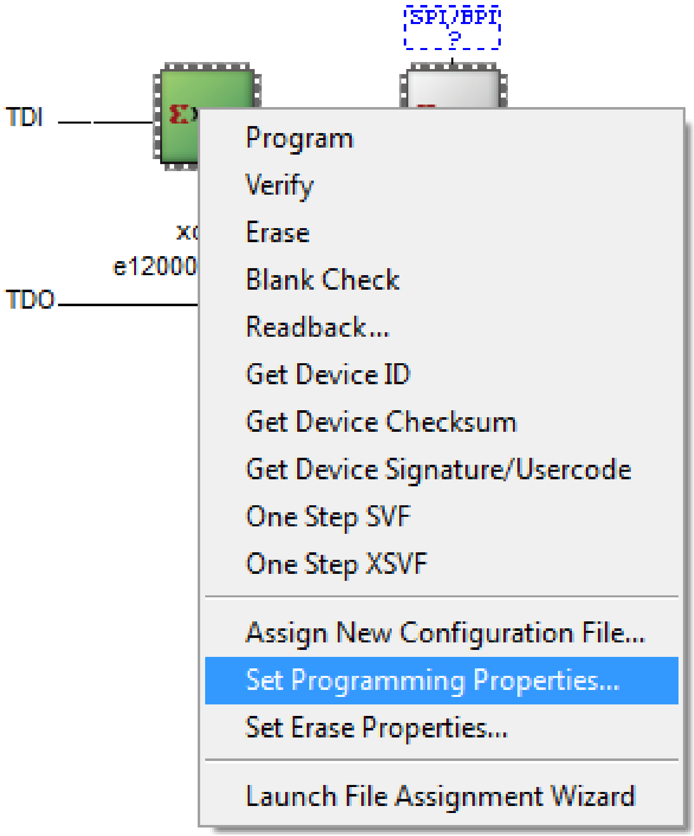
Select PROM

Part Name: xcf08p

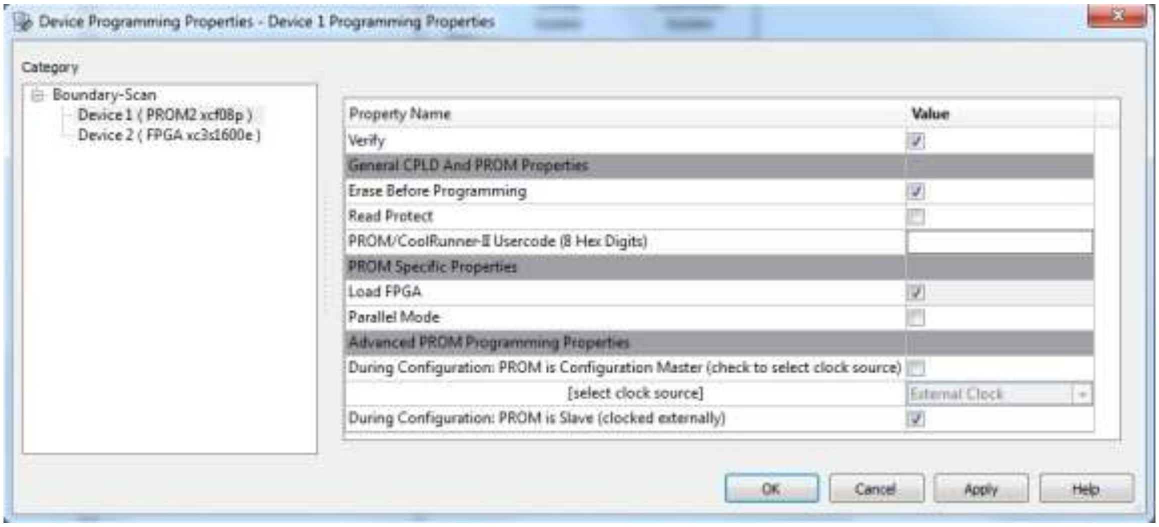
A square representing the newly added device will appear in the main window labeled "xcf08p" with the .mcs filename beneath it.

6. Repeat Part 4, selecting the .bit file you wish to load onto the FPGA. Again, the device will appear in the main window. If asked, skip SPI/BPI configuration.

7. Right-click on the Flash PROM device (the .mcs device, which should be on the left) and click "Set Programming Properties...":



In the popup window, make sure to check "Verify", "Erase Before Programming", and "Load FPGA", leaving the other fields unchanged, and press "OK":


Device Programming Properties - Device 1 Programming Properties
Category
Boundary-Scan
Device 1 ( PROM2 xcf08p )
Device 2 ( FPGA xc3s1600e )
OK
Cancel
Apply
Help

| Property Name | Value |
|---|---|
| Verify | ☑ |
| General CPLD And PROM Properties | |
| Erase Before Programming | ☑ |
| Read Protect | ☐ |
| PROM/CoolRunner-II Usercode (8 Hex Digits) | |
| PROM Specific Properties | |
| Load FPGA | ☑ |
| Parallel Mode | ☐ |
| Advanced PROM Programming Properties | |
| During Configuration: PROM is Configuration Master (check to select clock source) | ☐ |
| [select clock source] | External Clock |
| During Configuration: PROM is Slave (clocked externally) | ☑ |

8. Save the project file. Proceed from Step 2.6 (p. 4) to program the Flash PROM. To save a configuration archive, open *File > Save Configuration Archive...*

## Appendix IV: Programming the FPGA via Altium Designer

This section describes the Altium Designer method of FPGA programming. Though significantly slower than the iMPACT method, it can be useful in certain situations. The aLIGO Timing System FPGA code was written in Altium Designer, so modifications to the code (and subsequent compiling/programming/testing) must be performed in Altium. Whether this method is necessary is left to the discretion of the end user.

Straightforward programming using existing code can always be performed with iMPACT. Xilinx manufactures the Spartan3E FPGAs used in aLIGO Timing devices, and consequently, iMPACT is used to program them. Altium Designer relies on iMPACT as well as Xilinx drivers found in the ISE Design Suite in order to access the FPGA, so an Altium installation must be accompanied by a full installation of Xilinx ISE Design Suite 12.2. Unlike iMPACT, Altium requires a purchased license to run. It is also slower, more complicated, and more prone to programming failure.

1. Installing Altium Designer. A typical installation of Altium Designer Summer 09 should be performed. Ideally, the version number should correspond to that listed on the aLIGO Timing Wiki where the FPGA code is located. The Summer 09 version of Altium used in designing the aLIGO Timing System is not officially compatible with ISE 12.2, but by modifying a text file called “XilinxVersions.txt” located in *C:\Program Files (x86)\Altium Designer Summer 09\System* (or whichever directory contains Altium’s program files), Altium can be made to operate smoothly with 12.2. Open it in a text editor and write “12.2” on a new line immediately beneath 12.1. Windows might prevent you from saving the file; in this event, save the file with the same title in a different directory. Delete the old file from the *System* folder and then move the new file into its place.

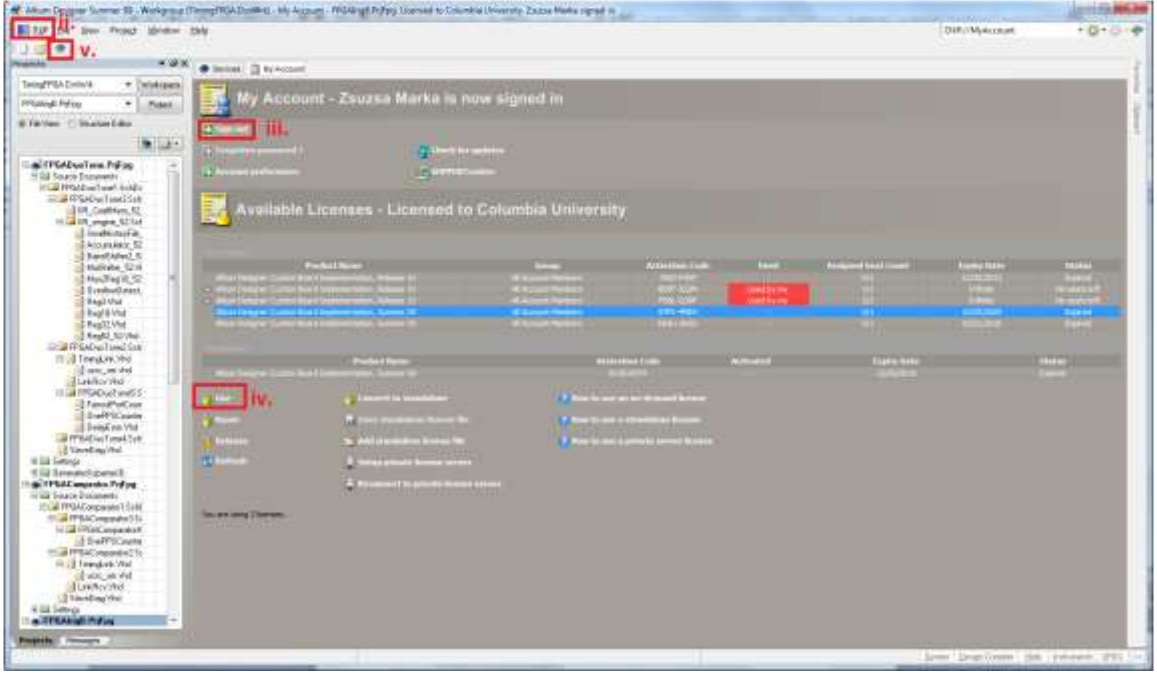
2. Download binary (.mcs and .bit) and/or Altium project files.

 i. Download the .mcs and .bit files for the FPGA device to be programmed. See Step 1.1 (p. 2) for instructions.
 ii. If modifying the FPGA code, also download the correct Altium project file for the device (this is unnecessary if existing .mcs and .bit files are being used). Altium project files are stored via subversion archive (SVN). See the Timing FPGA page for SVN configuration instructions. Use the SVN revision listed on the Timing FPGA page to obtain the correct file for the device in question.

3. Using Altium Designer. Altium requires an account to run. Once one has been purchased, the user must sign into his or her account and use Altium as follows:

 i. Open Altium Designer.
 ii. Open *DXP* > *My Acount*.
 iii. Press “Sign In,” Provide username and password, and press “Sign In.”

iv. Activate the product by clicking on the "use" button.


ii.
v.
My Account - Zsuzsa Marka is now signed in
iii.
Available Licenses - Licensed to Columbia University
iv.

v. Click on the "Device View" button.

vi. Load project. Go to *File > Open Project*. Open the Altium project file. It should be located in the topmost directory of the project folder. For example, Slave-Duotone Altium files reside in the DuoTone folder downloaded from the SVN: *C:\….\DuoTone\FPGADuoTone.PrjFpg*

vii. Connect the computer to the board using the Altium USB-JTAG adapter.

viii. Make sure that "Live" is checked on the upper right corner of the Devices View screen in Altium Designer.

ix. Make sure that the upper left corner of the screen reads "Connected."

x. If more than one Altium project is open, make sure that the correct one is selected for the FPGA. Once a board has been connected, two icons should appear in the Devices View. The left-hand device is the Flash PROM, and the right-hand device is the FPGA chip. From the menu below the FPGA icon, select the project you need. For example, if you are programming a Slave-DuoTone module, select the project labeled *FPGADuoTone / Spartan3*.

xi. Reset Flash PROM. Right-click on Flash PROM icon and choose "Reset Hard Device." Note that resetting can take up to a few minutes for the Flash PROM. Progress is indicated on the lower left corner of the screen.

xii. Upload FPGA program to Flash PROM. Right-click on the Flash PROM icon as in the previous step, and select "Choose File and Download." Locate the desired .mcs file. Note that uploading the code can take up to a few minutes. Progress is indicated on the lower left corner of the screen.

xiii. After the Flash PROM is programmed, the code is generally loaded onto the FPGA automatically. If the device does not demonstrate proper operation within several moments, the FPGA can be reset by interrupting the device's power supply (12V). When power is restored, the FPGA will be automatically reprogrammed from the Flash PROM. The device is now ready to use.