University of Toronto

Faculty of Applied Science and Engineering

ECE532 Digital Systems Design

Bluetooth PMOD Guide

Document Name	Bluetooth PMOD Guide with ILA Debugging
Authors	Shariq Khalil Ahmed Kazi Sudipto Arif
Revision	1.0
Revision Date	April 11, 2017

Table of Contents

1 Introduction

This document is intended as a quick guide to get your design to send and receive data with a BT2 Pmod Module in a few minutes. It also gives you a quick introduction to debugging with the ILA (Integrated Logic Analyzer) with bluetooth in context. The document assumes that the reader has completed the tutorials assigned at the start of ECE532S Digital Systems Design. A completed version of this tutorial can be found here:
https://github.com/shariqkhalilahmed/BT2PmodExample

2 Prerequisites

The following hardware is required to follow this guide:

- Nexys 4 DDR board
- BT2 Pmod

The following software is required to follow this guide:

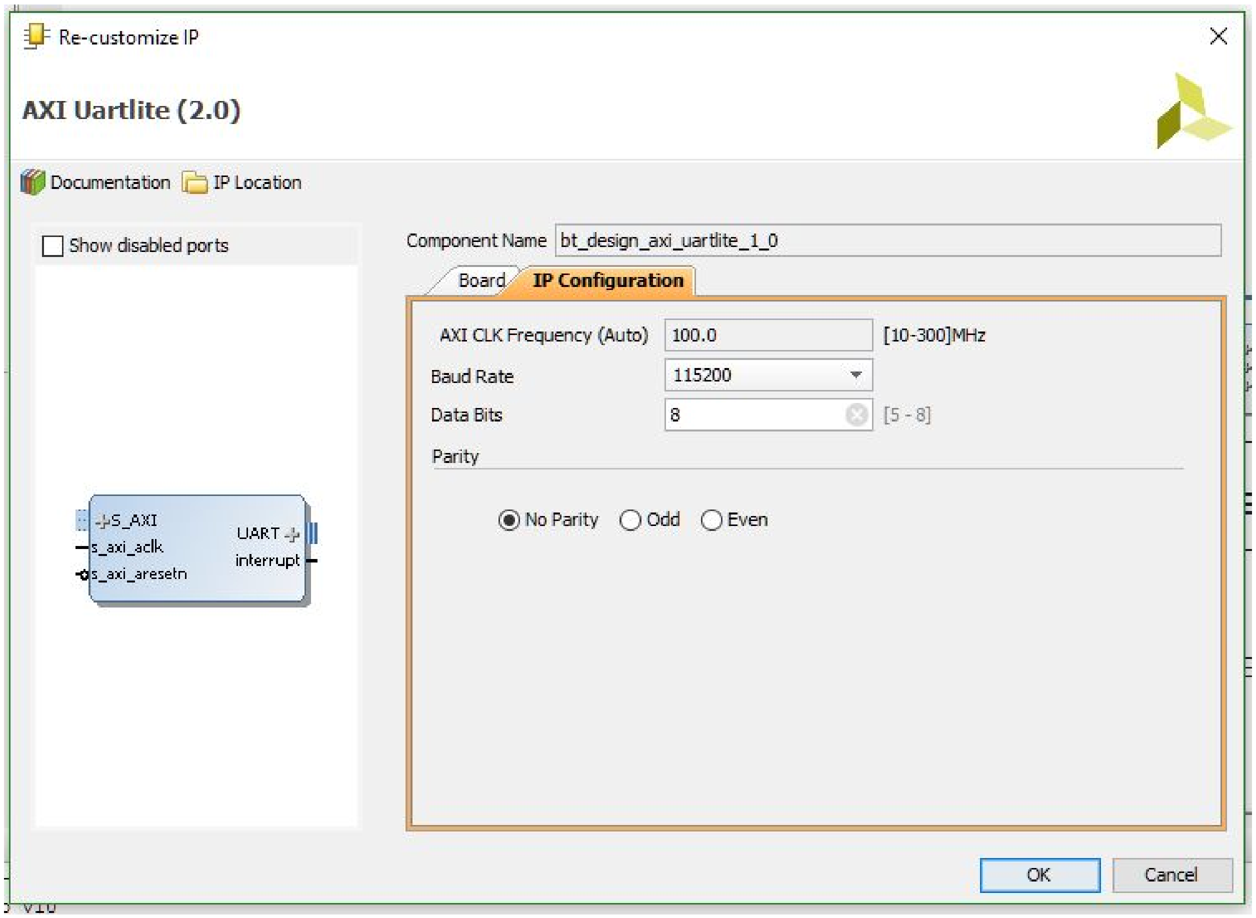
- Vivado Design Suite 2016.2 (Your mileage may vary with other versions of Vivado)
- Xilinx SDK 2016.2 (Your mileage may vary with other versions of Xilinx SDK)
- Base Microblaze project created in Tutorial 2 (microblaze) for ECE532

3 Instructions

Note: While digilent offers their own custom IP for the BT2 Pmod we have found that unless you need the more complex functionality offered by the custom IP, using simple Uartlite IP is much easier when integrating in hardware as well as software.

3.1 Adding UART to Block Design

- Open the Base Microblaze project from Tutorial 2 (microblaze) in Vivado
- Select 'Open Block Design' under IP Integrator
- Right-click anywhere in the design and select 'Add IP'
- Search for and select 'AXI Uartlite'
- Right-click on the IP and select 'Customize Block'
- Under 'IP Configuration' tab, configure IP to match settings of BT2 Pmod. The default settings for BT2 Pmod are seen in Figure 1. Press OK.
- Right-click block and select 'Block Properties'
- In Block Properties > General, rename name to 'bt_uart'
- Change configuration of 'axi_uartlite_0' to match 'bt_uart' as well


Re-customize IP

AXI Uartlite (2.0)

Component Name: bt_design_axi_uartlite_1_0

IP Configuration

AXI CLK Frequency (Auto): 100.0 [10-300]MHz

Baud Rate: 115200

Data Bits: 8 [5 - 8]

Parity: [✓ No Parity] Odd Even

Figure 1. AXI Uartlite configuration for default BT2 Pmod settings

- Find 'microblaze_0_xlconcat' block in your design, right-click and select 'Customize Block'
- Change number of ports to 2
- Now connect interrupt output of 'bt_uart' to newly generated input port of 'microblaze_0_xlconcat'
- Right-click UART interface signal in bt_uart, and select 'Make External'
- Rename the external interface to uart_bt2
- Run Automatic Connection for S_AXI interface for bt_uart
- Ensure s_axi_aclk and a_axi_aresetn are connected (use 'axi_uartlite_0' connections for reference)

3.2 Adding Constraints for JA Header

- Open .xdc file found under 'Constraints' in 'Sources'
- Add the following three lines to the end of the constraints file
    - #JA PMOD RXD TXD for BT2PMOD

    - set_property -dict { PACKAGE_PIN D18   IOSTANDARD LVCMOS33 } [get_ports { uart_bt2_txd }]; #IO_L21N_T3_DQS_A18_15 Sch=ja[2]
    - set_property -dict { PACKAGE_PIN E18   IOSTANDARD LVCMOS33 } [get_ports { uart_bt2_rxd }]; #IO_L21P_T3_DQS_15 Sch=ja[3]
- Save file

3.3 Generate Bitstream File

We are now ready to generate the hardware:

- Go to 'Sources', right-click .bd design under wrapper and select 'Reset Output Products'
- Select 'Generate Block Design' under IP Integrator
- Select 'Generate Bitstream' under Program and Debug (This will ask to generate synthesis and Implementation as well)

3.4 Export Hardware and Control with Xilinx SDK

- Select File > Export > Export Hardware and insure 'Include bitstream' is checked
- Select File > Launch SDK
- In the previous hello world.c, paste helloworld.c from github: https://github.com/shariqkhalilahmed/BT2PmodExample/blob/master/src/bt_test_proj/bt_test_proj.sdk/bt_simple/src/helloworld.c
- Plug in Nexys 4 DDR board with BT2 Pmod on JA header
- Flash the hardware from Xilinx Tools > Program FPGA (ensure latest hardware platform is being flashed)
- Under run configurations, select the current configuration and head to STDIO Connection
- Ensure that port is set to highest COM port and baud rate is set to whatever you set for Uartlite IPs in block design
- Click Run

3.5 Connect and Send Data over Bluetooth

You should now be able to send any byte data over bluetooth by typing in USB UART console and vice versa. Ensure that a device is connected over bluetooth to the BT2 Pmod. Any bluetooth device acting as serial terminal over bluetooth 2.0 should work. Here is a sample Android App that can connect to BT2 Pmod: https://github.com/shariqkhalilahmed/FPGASnapChatBluetooth. Here is an app on the Google Play Store that can send and received data with BT2 Pmod: https://play.google.com/store/apps/details?id=de.kai_morich.serial_bluetooth_terminal&hl=en

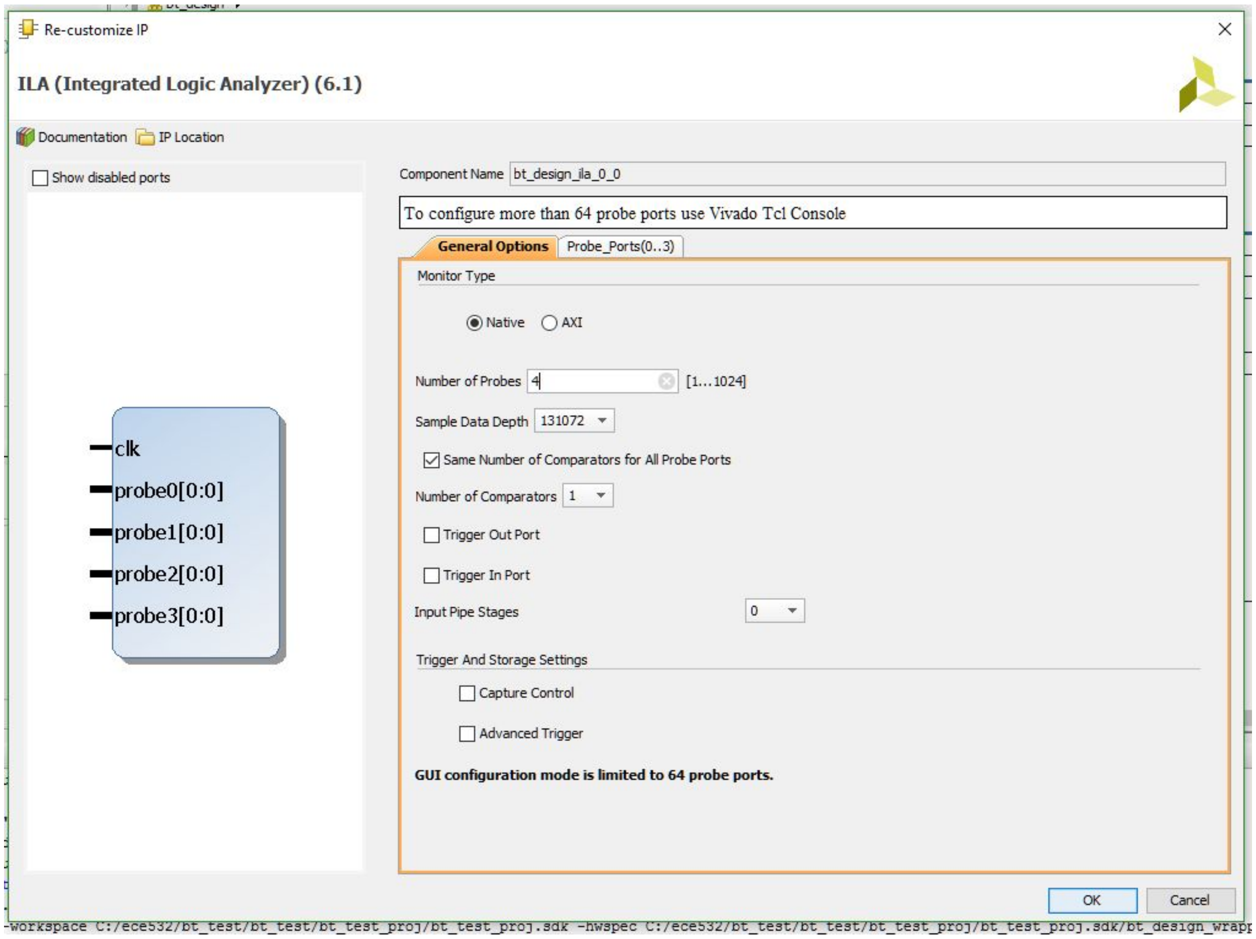
- Download and install either to an Android Phone
- Search for and connect to a name variant of 'RNBT-AAA3' (This is BT2 Pmod's name)
- Send and receive text over bluetooth from using USB UART console

4 Debugging with ILA and Script

4.1 Adding the ILA core to Block Design

If the system about does not work or you want to verify incoming and outgoing data over UART RX and TX lines, this sections guides you through connecting an ILA (Integrated Logic Analyzer), using the ILA and parsing the collected dump using provided verify.rb Ruby script on github: https://github.com/shariqkhalilahmed/BT2PmodExample/blob/master/src/verify.rb

- Open Block Design
- Right-Click and Add IP
- Search for ILA
- Configure the ILA to match the settings seen in Figure 2


Re-customize IP

ILA (Integrated Logic Analyzer) (6.1)

Documentation IP Location

☐ Show disabled ports

Component Name: bt_design_ila_0_0

To configure more than 64 probe ports use Vivado Tcl Console

General Options | Probe_Ports(0..3)

Monitor Type

● Native ○ AXI

Number of Probes: 4 [1...1024]

Sample Data Depth: 131072

☑ Same Number of Comparators for All Probe Ports

Number of Comparators: 1

☐ Trigger Out Port

☐ Trigger In Port

Input Pipe Stages: 0

Trigger And Storage Settings

☐ Capture Control

☐ Advanced Trigger

GUI configuration mode is limited to 64 probe ports.

```
-workspace C:/ece532/bt_test/bt_test/bt_test_proj/bt_test_proj.sdk -hwspec C:/ece532/bt_test/bt_test/bt_test_proj/bt_test_proj.sdk/bt_design_wrapp
```

Figure 2. ILA configuration

- Ensure Probe Width is 1 under Probe Ports
- In Block Design, for both USB and Bluetooth Uartlite, expand UART interface by clicking + sign

- Right-click on rx and tx for both Uartlite select ‘Make External’
- Right-click the tx for both Uartlite and select ‘Mark Debug’
- Right-click the nets (wires) going into rx and select ‘Mark Debug’
- Connect the clk input of the ILA block to the clk s_axi_clk of the uartlite
- Connect the probes of the ILA in the following fashion
    - Probe[0] rx of bt_uart
    - Probe[1] tx of bt_uart
    - Probe[2] rx of axi_uartlite_0
    - Probe[3] tx of axi_uartlite_0
- Rename the external rx and tx ports to bt_tx, usb_tx, bt_rx, usb_rx
- Ensure connections are identical to Figure 3

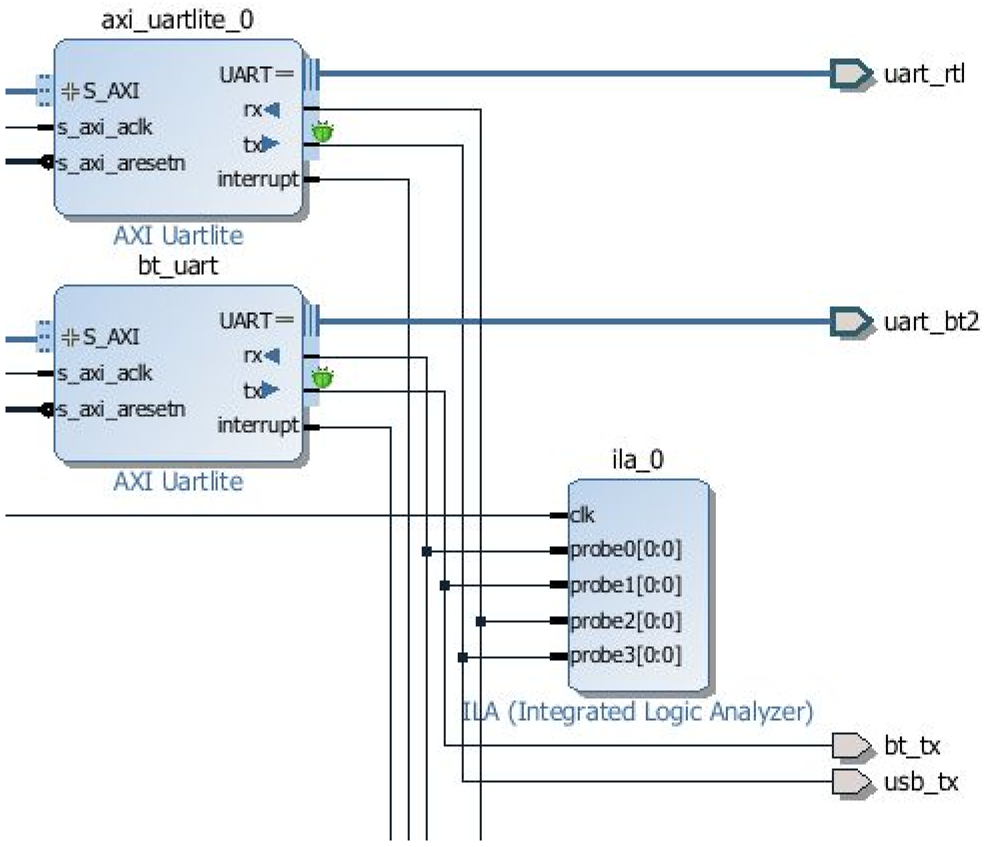


Figure 3. ILA connections to Uartlite Blocks

- Modify the following lines in the .xdc constraints files to:
    - set_property -dict { PACKAGE_PIN C4   IOSTANDARD LVCMOS33 } [get_ports { usb_rx }];
    - set_property -dict { PACKAGE_PIN D4   IOSTANDARD LVCMOS33 } [get_ports { usb_tx }];

    - set_property -dict { PACKAGE_PIN D18   IOSTANDARD LVCMOS33 } [get_ports { bt_tx }]; #IO_L21N_T3_DQS_A18_15 Sch=ja[2]
    - set_property -dict { PACKAGE_PIN E18   IOSTANDARD LVCMOS33 } [get_ports { bt_rx }]; #IO_L21P_T3_DQS_15 Sch=ja[3]

4.2 Getting Data from ILA core

- Perform Steps 3.3 & 3.4
- Open Hardware Manager under Program and Debug in Vivado
- Open Target >Auto Connect to Target
- Select ILA Core in Hardware Window
- In Dashboard, under Trigger Settings, set a trigger on bt_rx equivalent to sense a 1-to-0 transition
- Right-click and Run trigger to arm the ILA for capture
- Send text using bluetooth device
- ILA should have triggered and would be showing waveforms
- In the waveform window, select the save file button (White file with green +)
- Save the ILA captured data in .csv format
- Edit the ruby script 'verify.rb' from the github repo and change the filename parameter on line 6 to the file you saved (https://github.com/shariqkhalilahmed/BT2PmodExample/blob/master/src/verify.rb)
- Make sure Ruby is installed (Available for both Windows and Linux) and run the script (ruby ./verify.rb)
- Script should print baud rate, messages received and sent along with data integrity check