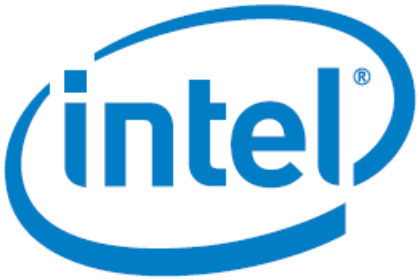


Chip ID Reading using AVST Mailbox IP in Agilex

Date: 10/1/2019

Revision: 1.0

1 Contents

1 Introduction/Overview

In Agilex, IP to perform chip ID reading is no longer available. Thus, custom logic has to be created and connected to Mailbox Avalon® ST Client Intel FPGA IP to perform chip ID reading.

This reference design indicates the usage of Mailbox Avalon® ST Client Intel FPGA IP to perform chip id reading. The tutorial is demonstrated using signal tap.

2 Requirements

2.1 Hardware Requirements

- You should create and download this example design to the Agilex™ F-Series Transceiver-SoC Development Kit. The example design can be downloaded from Intel Design Store. The title of the example design is Chip ID Reading using AVST Mailbox IP in Agilex.

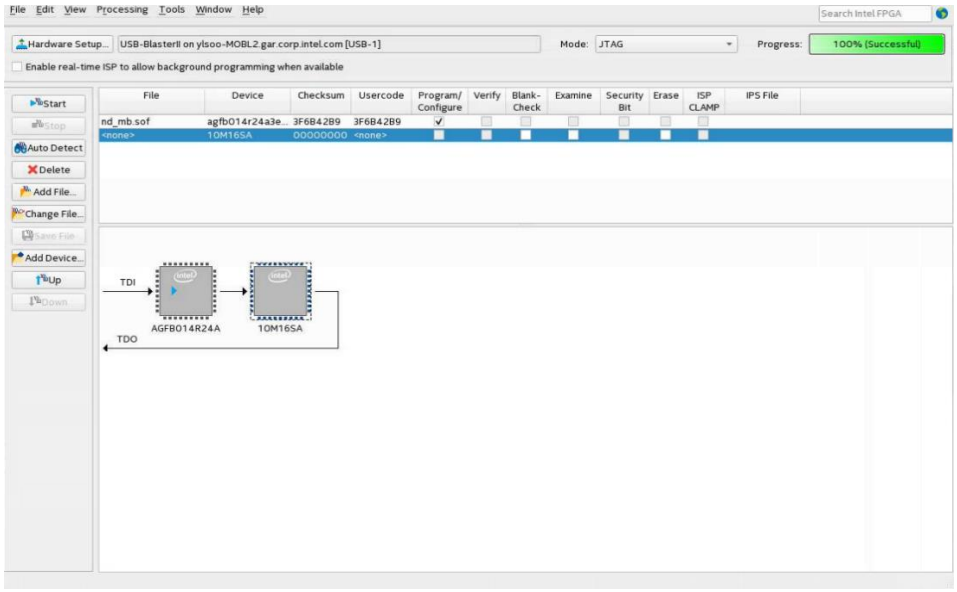
2.2 Software Requirements

- Intel Quartus Prime Pro Edition software version 19.3 or later.

3 Walkthrough

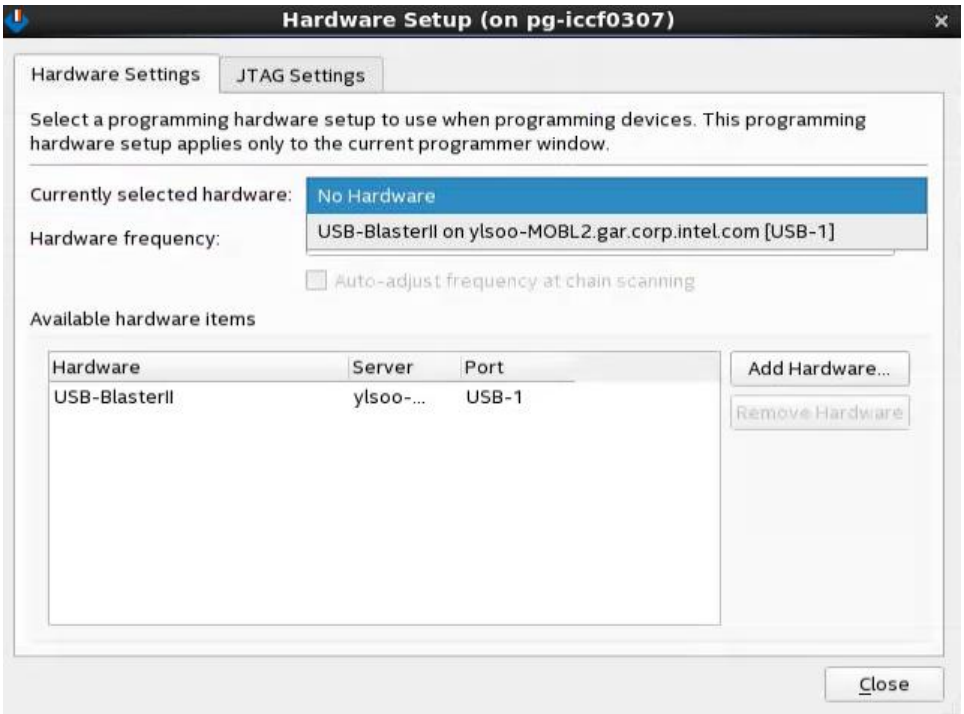
3.1 Programming flash devices with .sof programming file

- Compile the Design and a .sof programming file (.sof) will be generated.
- Open **Programmer**, click **Add File**, select the generated .sof programming file (.sof) and click **Open**.
- Check the **Program/Configure** check box for the attached .sof file.
- Click **Start** to start programming the flash devices.
- Configuration is complete when the progress bar reaches 100%.

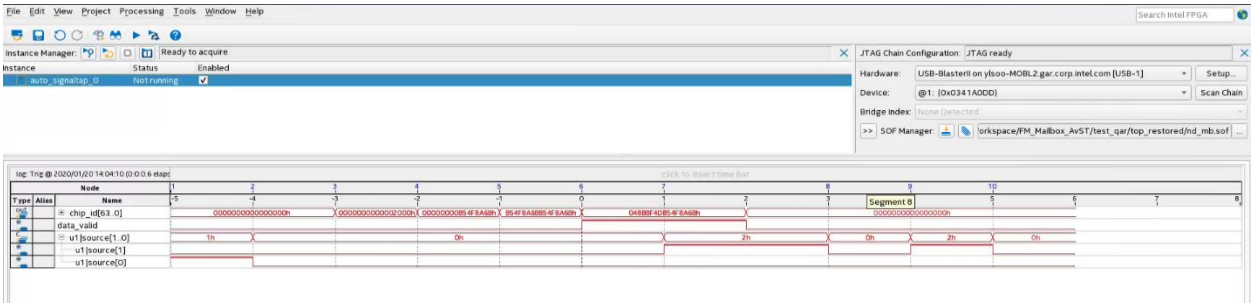


3.2 Running the signal tap

- Open Intel Quartus Prime Pro Edition software.
- On the **Tools** menu, select **Signal Tab Logic Analyzer**.
- Connect the Signal Tap Logic Analyzer with development kit by pressing **setup** and then select the USB blaster that we are using.

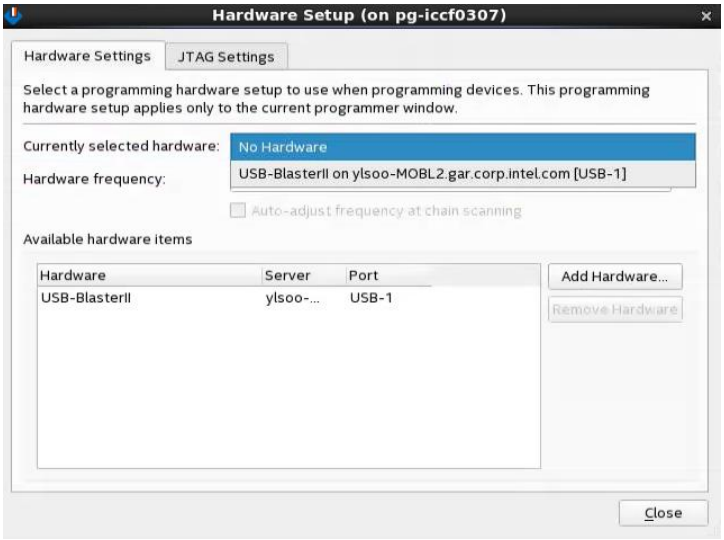


- When it states, **“Ready to acquire”**, that means the connection is ready and we can perform chip ID reading at any time.
- Click on **“Run Analaysis”** or **F5** to perform signal tab operation.

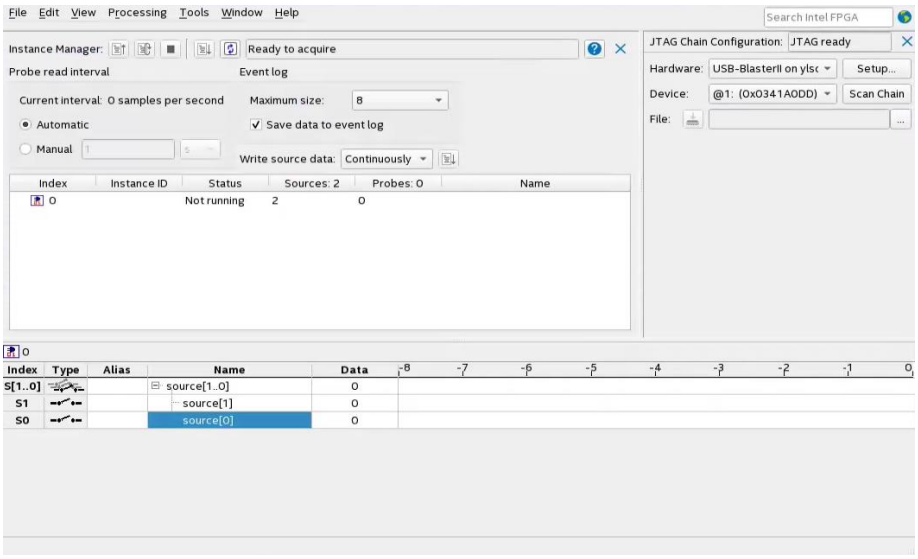


3.3 Running the In-System Sources and Probes Editor

- To perform chip ID reading, we need to assert reset input and then assert the chip ID reading input in the logic. To do so, we use in-system sources and probes editor.
- On the **Tools** menu, select **In-System Sources and Probes Editor**.
- Connect the **In-System Sources and Probes Editor** with development kit by pressing **setup** and then select the USB blaster that we are using.



- When it states, **“Ready to acquire”**, that means the connection is ready and we can use **In-System Sources and Probes Editor** at any time.



- Double click on “source[1]” first to assert the reset input and then de-assert it. Then, double click on “source[0]” to assert read id input. Finally, we can then go to signal tab, select **“Stop Analysis”** or **Esc**, and view the result.

4 Concept

4.1 Theory of Operation

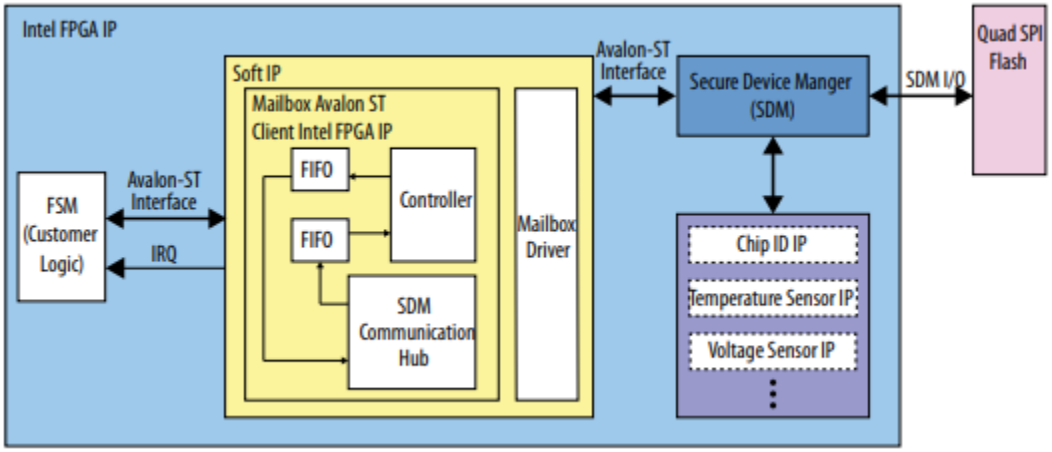
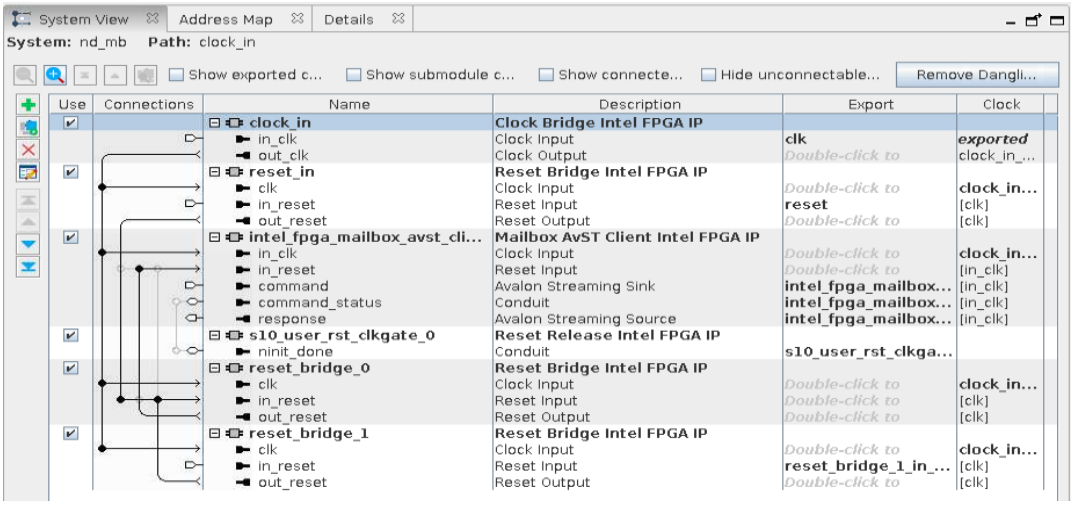


Figure 2. Mailbox Avalon ST Client Intel FPGA IP System Design

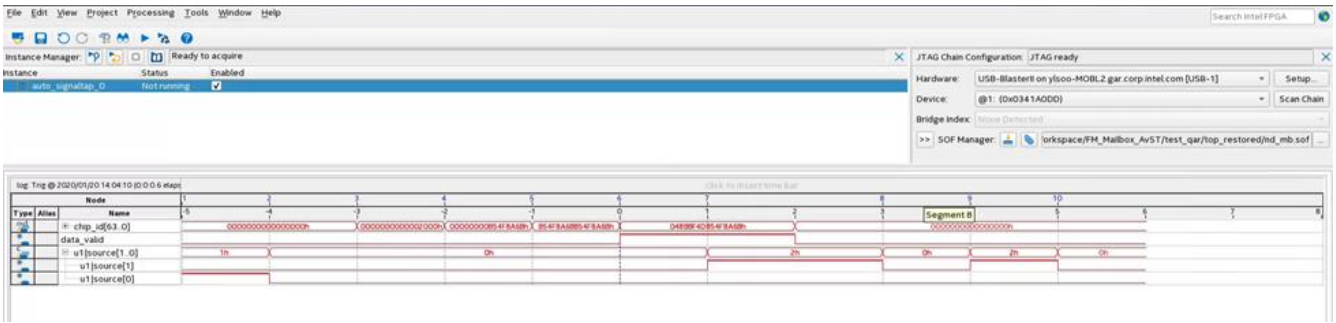
The Mailbox Avalon® ST Client Intel® FPGA IP provides a communication channel between your custom logic and the secure device manager (SDM). You can use the Mailbox Avalon ST Client IP to send command packets and receive response packets from SDM peripheral modules. The Mailbox Avalon ST Client IP defines functions that the SDM runs.

In this reference design, a custom logic will be created to perform chip ID reading. The logic will interact with the Mailbox Avalon® ST Client Intel® FPGA IP to perform chip ID reading. The component used in this reference design is as shown in the Platform Designer system. The JTAG to Avalon Master Bridge acts as the remote system update host controller for your factory and application images. Then, In-System Sources & Probes Intel® FPGA IP is included. This component allows you to easily control any internal signal and provides you with a completely dynamic debugging environment in user mode. Finally, connect the design with our custom chip ID reading logic and chip ID can be determined with SignalTap Embedded Logic Analyzer.



Design in Platform Designer

5 Results



Chip ID value can be read at **chip_id[63:0]** after signal tab operation is performed. As shown in the image above, the correct chip ID value is captured right after **data_valid** asserted.

6 Document Revision History

List the revision history for the application note.

Date	Version	Changes
June 2020	June 2020	V1.0