30. Avalon-ST JTAG Interface Core

QII55008-9.1.0

Core Overview

The Avalon® Streaming (Avalon-ST) JTAG Interface core enables communication between SOPC Builder systems and JTAG hosts via Avalon-ST interface. Data is serially transferred on the JTAG interface, and presented on the Avalon-ST interface as bytes.

The SPI Slave/JTAG to Avalon Master Bridge is an example of how this core is used. For more information about the bridge, refer to the *SPI Slave/JTAG to Avalon Master Bridge Cores* chapter in volume 5 of the *Quartus II Handbook*.

The Avalon-ST JTAG Interface core is SOPC Builder-ready and integrates easily into any SOPC Builder-generated systems.

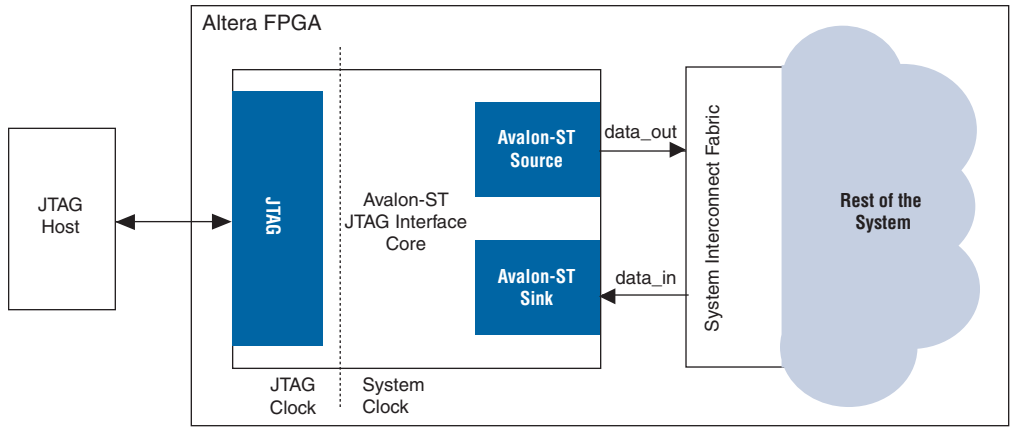
This chapter contains the following sections:

- "Functional Description"
- "Instantiating the Core in SOPC Builder" on page 30–3
- "Device Support" on page 30–3

Functional Description

Figure 30–1 shows a block diagram of the Avalon-ST JTAG Interface core in a typical system configuration.

Figure 30–1. SOPC Builder System with an Avalon-ST JTAG Interface Core



Interfaces

Table 30–1 shows the properties of the Avalon-ST interfaces.

Table 30–1. Properties of Avalon-ST Interfaces

Feature	Property
Backpressure	Only supported on the sink interface.
Data Width	Data width = 8 bits; Bits per symbol = 8.
Channel	Not supported.
Error	Not used.
Packet	Not supported.

For more information about Avalon-ST interfaces, refer to the *Avalon Interface Specifications*.

Special characters

Table 30–2 lists the special characters recognized by the core.

Table 30–2. Special Characters

Character	Description
`0x4a`	Idle. Idle characters are inserted into data streams when there is no data to send.
`0x4d`	Idle escape. An idle escape character is inserted into data stream when the data to send is a special character, followed by the data which is XORed with `0x20`.

Operation

The Avalon-ST JTAG Interface core accepts incoming data in bits on its JTAG interface and packs the bits into bytes. After each byte is formed, the core checks for the following special characters:

- `0x4a`—Idle character. The core drops the idle character.
- `0x4d`—Escape character. The core drops the escape character, and XORs the following byte with `0x20`.

Each valid byte is then transferred to the core's Avalon-ST source interface. As there are no means to backpressure this interface, you must ensure that sufficient storage is in place to avoid data loss.

In the opposite direction, the core serializes each byte received on its Avalon-ST sink interface and sends the bits to the JTAG interface. If there is no data on the sink interface, the core sends out idle characters. If the data is a special character, the core inserts an escape character and XORs the data with `0x20`.

The core supports four operation modes. From the system console, you can set the instruction register (IR) to enable the following supported modes:

- Normal mode—The core works as a bridge between a JTAG host and an SOPC Builder system. Set the IR to 0 to enable this mode.
- Loopback—Data received by the core is sent back to the host. Set the IR to 1 to enable this mode.

- Troubleshoot—The core retrieves the value of the system reset and clock signals, and return them to the JTAG host. Set the IR to 2 to enable this mode.

A TimeQuest SDC file (**.sdc**) is provided to cut any paths internal to the core.

Instantiating the Core in SOPC Builder

Use the MegaWizard™ interface for the Avalon-ST JTAG Interface core in SOPC Builder to add the core to a system. There are no user-configurable parameters for this core.

Device Support

The Avalon-ST JTAG Interface core supports all Altera® device families.

Referenced Documents

This chapter references the following documents:

- *Avalon Interface Specifications*
- *SPI Slave/JTAG to Avalon Master Bridge Cores* chapter in volume 5 of the *Quartus II Handbook*

Document Revision History

Table 30–3 shows the revision history for this chapter.

Table 30–3. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
November 2009 v9.1.0	No change from previous release.	—
March 2009 v9.0.0	No change from previous release.	—
November 2008 v8.1.0	Changed to 8-1/2 x 11 page size. No change to content.	—
May 2008 v8.0.0	Initial release.	—