Xilinx Internal JTAG

(Test Access Port)

Author: Nathan Yawn
nathan.yawn@opencores.org

Rev. 1.0

May 16, 2009

History

Rev.	Date	Author	Description
1.0	18/07/2008	Nathan Yawn	First Draft
1.1	16/05/2009	NY	Edit for initial release

Contents

1 INTRODUCTION.....5

2 OPERATION.....6

2.1 Xilinx BSCAN_* Macro Blocks.....6
2.2 Instructions.....7
2.3 Non-standard Behavior.....7

3 IO PORTS.....8

3.1 Debug Ports.....8

4 REGISTERS.....9

5 INTEGRATED SYSTEM.....10

1 Introduction

The Xilinx Internal JTAG core is used for development purposes (hardware and software debugging). It uses a Xilinx BSCAN_SPARTAN or BSCAN_VIRTEX IP library function macro to allow debugging of an OpenRisc-based System-on-Chip (SoC) using the same JTAG interface IO pins used to upload the bitstream to the FPGA. This core can only be used in Xilinx FPGAs which support a BSCAN_* macro. Before continuing with this document, it is recommended that you familiarize yourself with the IEEE 1149.1 specification (JTAG and Boundary Scan), and also that you read the section of the Xilinx Unified Libraries manual which deals with the BSCAN_* macros.

The xilinx_internal_jtag core is designed to replace the "jtag" TAP core in a Xilinx-based system. It is designed to provide an interface between the FPGA's JTAG pins and the SoC debug core. In particular, the xilinx_internal_jtag core is designed to interface with the Advanced Debug Interface (adv_dbg_if core). Other debug cores may require modification in order to work with this one. Note that this core provides only a single device enable output (for the debug core); adding other JTAG scan chains may be done by hand, if multiple USER instructions are supported by the BSCAN_* block in your particular FPGA.

2 Operation

This section describes the operation of the xilinx_internal_jtag core. The core relies heavily on the BSCAN_* macro blocks for its operation. In fact, the core is primarily a wrapper for the BSCAN macro, with some additional logic to make the signals more compatible with standard JTAG signaling.

2.1 Xilinx BSCAN_* Macro Blocks

These macro blocks are designed to allow a user-defined JTAG scan chain to be accessed through the same pins used to upload a configuration bitstream to the FPGA. In particular, the BSCAN macro allows a user to attach programmable FPGA logic to the hard-wired JTAG TAP. The FPGA logic is connected to the TAP as a Data Register (DR), which is selected when one of the USERn instructions is active in the TAP IR.

The BSCAN_* blocks provide different signals, depending on what type of Xilinx device is used. All provide SHIFT_DR and UPDATE_DR outputs, as well as a SELECT output. Many provide a CAPTURE_DR output, and this signal can be implied from other signals in devices that do not provide the signal. Clocking is of particular interest; with the exception of one device (BSCAN_SPARTAN3), the BSCAN_* macros do not provide a regular TCK JTAG clock. Instead, they provide a clock named DRCK. The DRCK clock first goes high when USERn is made active in the TAP's IR. The clock does not change state until the TAP reaches the CAPTURE_DR state. While the TAP is in the CAPTURE_DR or SHIFT_DR states, DRCK follows the TCK input clock. Once the TAP leaves the SHIFT_DR state, DRCK returns to a constant high level. This means that no transitions on the DRCK line occur while the UPDATE_DR output signal is active, which is incompatible with many JTAG devices. See Figure 1 for an example of the output waveform of a BSCAN_* device. (Where possible, the xilinx_internal_jtag core compensates for this, see section 2.3.)

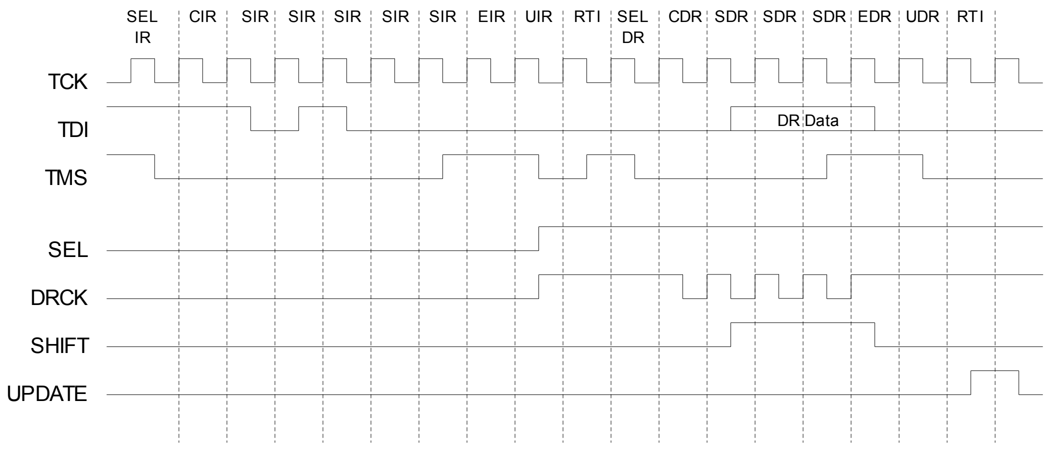

Figure 1: JTAG signals and their corresponding BSCAN_* outputs

2.2 Instructions

The xilinx_internal_jtag core instantiates a BSCAN_* macro associated with the USER1 instruction. To make the debug core active, the USER1 instruction must be shifted into the FPGA's IR. The value (and IR length) may change depending on which FPGA is used, check the chip's documentation or JTAG BSDL file for details.

The debug interface core is connected as the Data Register (DR) of the JTAG TAP when the USER1 instruction is active in the IR. Data from the xilinx_internal_jtag core to the debug core is assumed to be latched on the rising edge of the TCK ouput clock. Data from the debug core to the internal jtag core is latched on the falling edge of TCK.

2.3 Non-standard Behavior

Because of the behavior of the DRCK line, logic has been added to the xilinx_internal_jtag core in order to make it more compatible with standard behavior. The tck_o clock output from the xilinx_internal_jtag core is AND'ed with the inverse of the UPDATE_DR output line of the BSCAN_* macro. This forces the clock low during the UPDATE_DR TAP state. By latching the UPDATE_DR line from the BSCAN_* macro, the update_dr_o output is held high when the UPDATE_DR line is de-asserted, causing a low-to-high transition on the tck_o output clock, which should allow connected devices to register the UPDATE_DR state.

The update_dr_o output remains high until the TAP's CAPTURE_DR state is entered, or the SELECT output of the BSCAN_* macro is de-asserted (there are no DRCK events during this period). This means that zero clock events will occur on tck_o between the UPDATE_DR and CAPTURE_DR states. Some devices, such as the adv_dbg_if core, require at least one clock during this period in order to maintain their internal state machine. In this case, a tck_o event must be generated by shifting a different instruction into the TAP IR (BYPASS is recommended), then re-activating USER1. The tck_o line will be forced low while USER1 is not selected. Driver software on the PC must be modified to perform this operation if necessary.

None of the BSCAN_* macros provide PAUSE_DR or RUN_TEST_IDLE outputs. While these ports are present in the xilinx_intenal_jtag interface (as pause_dr_o and run_test_idle_o, respectively), they are only for port compatibility with other JTAG TAP cores. The value of these signals is a constant '0'.

3

IO Ports

This section describes the top-level ports of the xilinx_internal_jtag core. Because all of the JTAG signals come through the BSCAN_* macro, the only ports from the core are used to interface to the debug interface.

3.1 Debug Ports

Port	Width	Direction	Description
tck_o	1	output	JTAG clock signal
test_logic_reset_o	1	output	TAP controller state “Test Logic Reset”, acts as reset signal to sub-modules (debug unit etc.)
run_test_idle_o	1	output	TAP controller state “Run Test / Idle” Constant '0' value.
shift_dr_o	1	output	TAP controller state “Shift DR”
pause_dr_o	1	output	TAP controller state “Pause DR” Constant '0' value.
capture_dr_o	1	output	TAP controller state “Capture DR”
update_dr_o	1	output	TAP controller state “Update DR”
debug_select_o	1	output	Debug select, true when DEBUG instruction active in the virtual IR
tdo_o	1	output	TDO signal, connects to all TDI signals of sub-modules (i.e. debug module)
debug_tdi_i	1	input	TDI signal from debug module

Table 1: Debug Ports

4 Registers

There are no registers in the Xilinx Internal JTAG core. While the instantiated BSCAN_* macro does include a JTAG Instruction Register, which must be loaded with the USER1 instruction to enable this core, the length of the register and the value of the USER1 instruction vary depending on which Xilinx FPGA is used. Consult the documentation for your particular FPGA (or its JTAG BSDL file) to determine these value.

5 Integrated System

The Xilinx Internal JTAG core is just one part of the complete debugging system. To be useful, the system-on-chip must also include a compatible debug core (i.e. the adv_dbg_if core), a WishBone bus, and an OR1200 CPU (or a CPU with a compatible debug interface). Externally, the debugging system must include a JTAG cable, GDB (the GNU Debugger program), a GDB-to-JTAG bridge program (i.e. adv_jtag_bridge), and an optional graphical front-end to GDB, such as DDD or Eclipse. A block diagram of this system is shown in Figure 2.

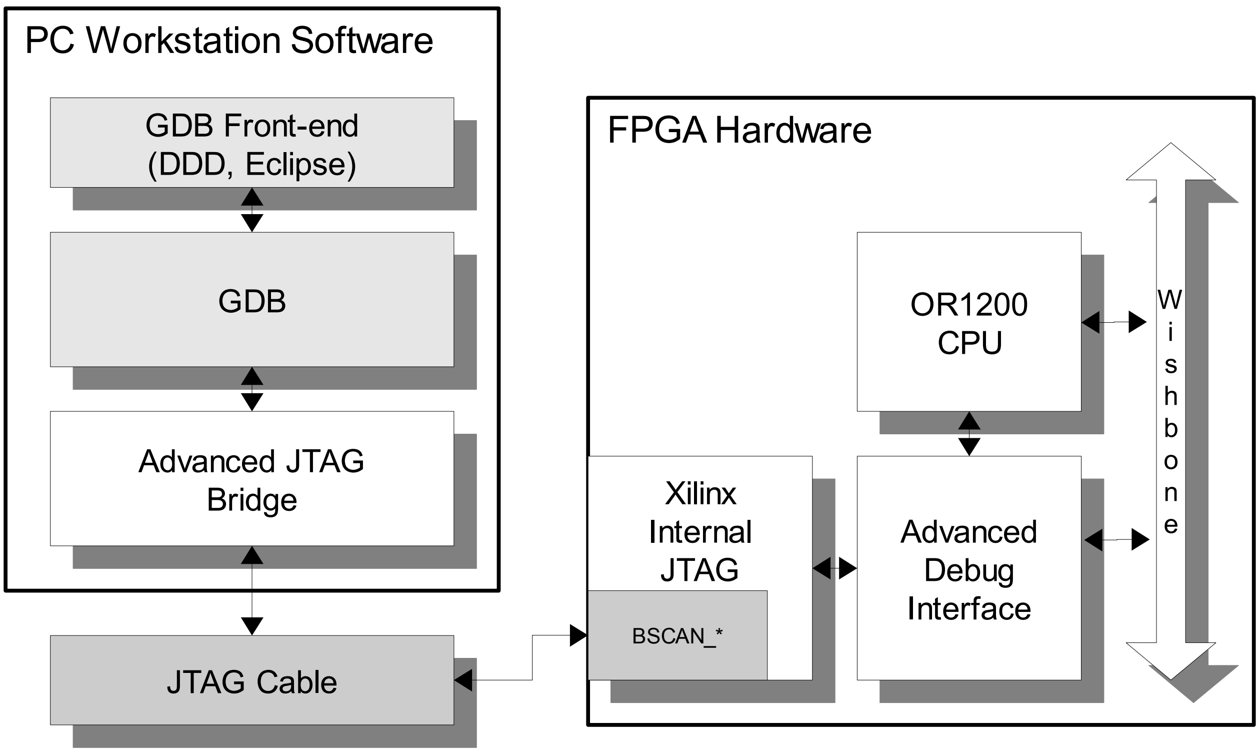

Figure 2: Complete Debug System Block Diagram