Actel UJTAG

(Test Access Port)

Author: Nathan Yawn
nathan.yawn@opencores.org

Rev. 1.0

Feb. 10, 2010

History

Rev.	Date	Author	Description
1.0	10/02/2010	Nathan Yawn	First Draft

Contents

1

Introduction

The actel_ujtag core is used for development purposes (hardware and software debugging). It uses the Actel "UJTAG" 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 Actel FPGAs which support the UJTAG 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 Actel documentation on the UJTAG macro.

The actel_ujtag core is designed to replace the "jtag" Test Access Port (TAP) core in an Actel-based system. It is designed to provide an interface between the FPGA's JTAG pins and the SoC debug core. In particular, the actel_ujtag 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 TAP. Note that while the actel_ujtag core provides only a single device enable output (for the debug core), it is a fully functional JTAG TAP, and other Data Register chains could easily be added.

2

Operation

This section describes the operation of the actel_ujtag core. The actel_ujtag core relies heavily on the Actel UJTAG macro for its operation. In fact, the core is primarily a thin wrapper and instruction decoder for the macro.

1 Actel UJTAG Macro

This macro is 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 UJTAG 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), and the value of the TAP Instruction Register (IR) is provided by the macro in order to allow the FPGA logic to decode one or more IR addresses.

The use of the UJTAG macro adds one constraint to a system. The TCK, TMS, TDI, and TDO ports of the UJTAG module must be connected to ports of the top-level entity with the same names. In the actel_ujtag core, these signals are connected to the tck_pad_i, tms_pad_i, tdi_pad_i, and tdo_pad_o ports respectively. These ports must be connected to ports of the top-level entity called TCK, TMS, TDI, and TDO. However, these top-level ports must not be connected to IO pins, they must be left unconnected – the Actel synthesis tools will detect these ports by name, and automatically connect them to the correct signals of the FPGA's hard TAP.

2 Instructions

The Actel UJTAG macro includes an IR which is 8 bits long. This is the same IR which is used to select the FPGA TAP's standard functions (Boundary Scan Register, BYPASS, IDCODE, etc.). However, instructions 0x10 through 0x7F are not used by the standard TAP functions, and may therefore be used by user logic via the UJTAG macro. By default, the actel_ujtag core uses address 0x44 for the debug unit (the debug_select_o line is high when the IR is 0x44).

3

IO Ports

This section describes the top-level ports of the actel_ujtag core. There are two groups of ports. The “JTAG ports” must be connected to ports of the top-level entity, which must be left unconnected. The “Debug ports” should be connected to the Advanced Debug Interface, or another JTAG client core.

1 JTAG Ports

Port	Width	Direction	Description
tck_pad_i	1	input	Test clock input
tms_pad_i	1	input	Test mode select input
tdi_pad_i	1	input	Test data input
tdo_pad_o	1	output	Test data output
trstn_pad_i	1	input	Test reset input

Table 1: JTAG Ports

2 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”
shift_dr_o	1	output	TAP controller state “Shift DR”
pause_dr_o	1	output	TAP controller state “Pause DR”
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 IR
tdi_o	1	output	TDI signal, connects to all TDI signals of sub-modules (i.e. debug module)
debug_tdo_i	1	input	TDO signal from debug module

Table 2: Debug Ports

4

Registers

This section specifies all registers in the actel_ujtag core.

1 Register List

Name	Width	Access	Description
IR	8	R/W	TAP Instruction Register

Table 3: Register List

2 IR (Instruction Register)

Bit #	Access	Description
7:0	R/W	Data Register to make active. 0100 0100b = DEBUG (Other values have other functions native to the Actel TAP; see Actel JTAG documentation for details.)

Table 4: IR Register

5

Integrated System

The actel_ujtag 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 1.

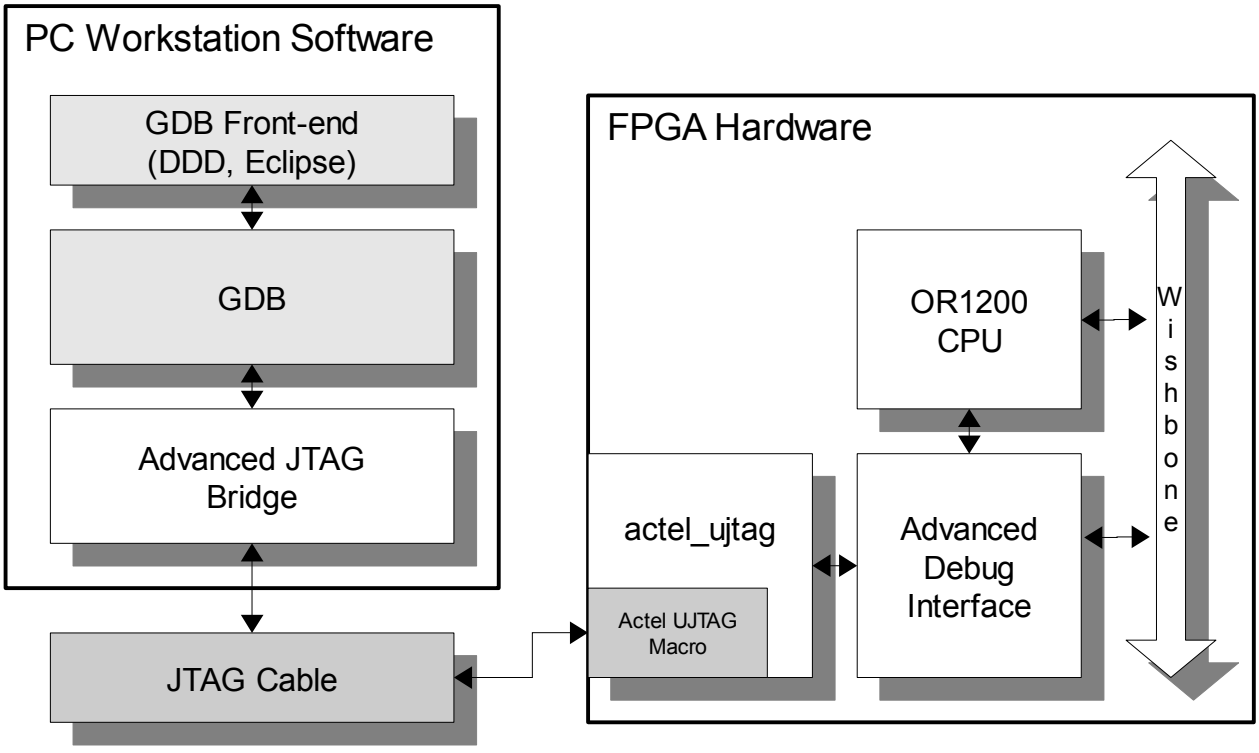

Figure 1: Complete Debug System Block Diagram