MicroBlaze Debug Module (MDM) v3.0

Product Guide for Vivado Design Suite

PG115 March 20, 2013

XILINX®

XILINX®

Table of Contents

Appendix A: Migrating

Appendix B: Debugging

Appendix C: Application Software Development

Appendix D: Additional Resources

Introduction

This document provides the design specification for the MicroBlaze™ Debug Module (MDM) which enables JTAG-based debugging of one or more MicroBlaze processors.

Features

- Support for JTAG-based software debug tools
- Support for debugging up to 32 MicroBlaze processors
- Support for synchronized control of multiple MicroBlaze processors
- Support for a JTAG-based UART with a configurable AXI4-Lite interface
- Based on Boundary Scan (BSCAN) logic in Xilinx FPGAs

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family(1)	Zynq™-7000, Virtex-7, Kintex™-7, Artix™-7
Supported User Interfaces	AXI4-Lite
Resources	See Table 2-1
Provided with Core	
Design Files	Vivado: RTL
Example Design	Not Provided
Test Bench	Not Provided
Constraints File	Not Provided
Simulation Model	VHDL Behavioral
Supported S/W Driver(2)	Standalone and Linux
Tested Design Tools	
Design Entry Tools	Vivado™ Design Suite
Simulation	Mentor Graphics Questa® SIM Vivado Simulator
Synthesis Tools	Vivado Synthesis
Support	
Provided by Xilinx @ www.xilinx.com/support	

Notes:

1. For a complete listing of supported devices, see the Vivado IP Catalog.
2. Standalone driver details can be found in the SDK directory (<*install_directory*>/doc/usenglish/xilinx_drivers.htm). Linux OS and driver support information is available from //wiki.xilinx.com.
3. For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.

Overview

The block diagram of the MicroBlaze™ Debug Module is shown in Figure 1-1.

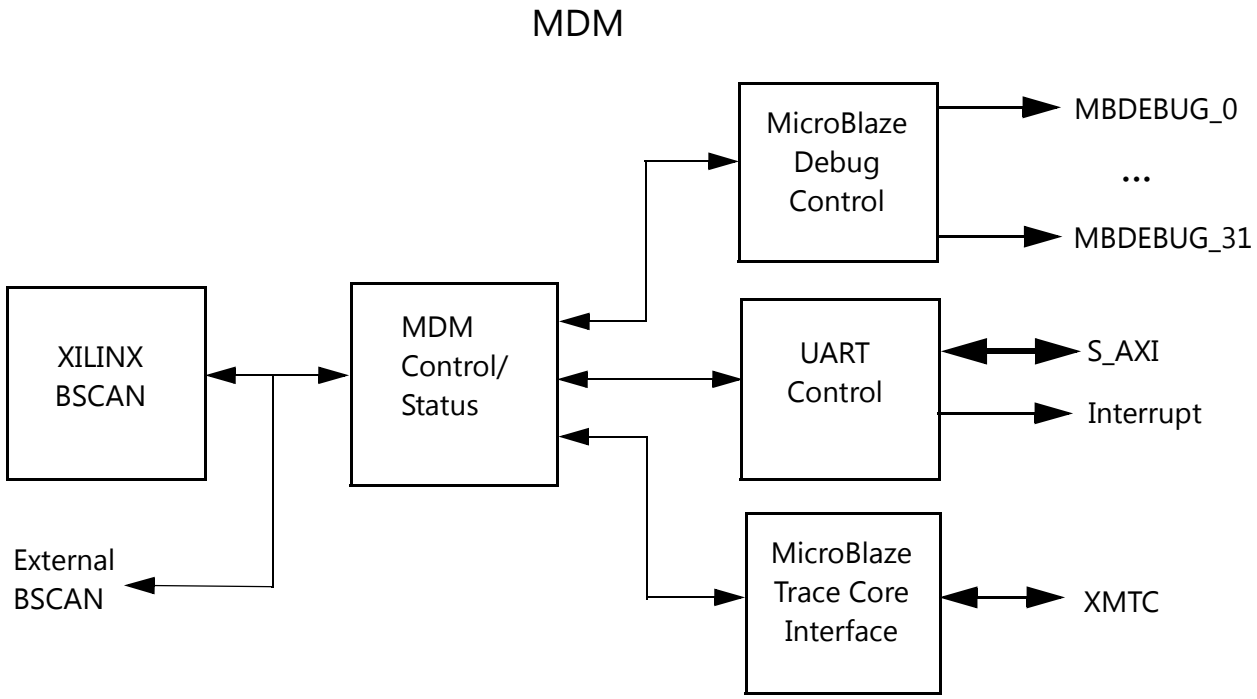

Figure 1-1: **Microblaze Debug Module (MDM) Block Diagram**

Feature Summary

The MicroBlaze Debug Module (MDM):

- Enables JTAG-based debugging of one or more MicroBlaze processors.
- Instantiates one BSCAN primitive, or allows an external BSCAN to be used. In devices that contain more than one BSCAN primitive, MDM uses the USER2 BSCAN by default.
- Includes a UART with a configurable slave bus interface which can be configured for an AXI4-Lite interconnect. The UART TX and RX signals are transmitted over the FPGA JTAG port to and from the Xilinx Microprocessor Debug (XMD) tool. The UART behaves in a manner similar to the LogiCORE™ IP AXI (UART) Lite core.

Licensing and Ordering Information

This Xilinx LogiCORE IP module is provided at no additional cost with the Xilinx Vivado™ Design Suite tools under the terms of the Xilinx End User License.

Information about this and other Xilinx LogiCORE IP modules is available at the Xilinx Intellectual Property page. For information on pricing and availability of other Xilinx LogiCORE IP modules and tools, contact your local Xilinx sales representative.

Product Specification

Standards Compliance

The MDM adheres to the AMBA® AXI4-Lite Interface standard (see ARM® *AMBA AXI and ACE Protocol Specification ARM IHI 0022D* [Ref 1]).

Performance

The frequency and latency of the MicroBlaze™ Debug Module are optimized for use together with MicroBlaze. This means that the frequency targets are aligned to MicroBlaze targets.

Resource Utilization

Because the MDM is a module that is used with other design elements in the FPGA, the utilization and timing numbers reported in this section are estimates, and the actual utilization of FPGA resources and timing of the MDM design can vary from the results for a Kintex-7 device reported in Table 2-1.

Table 2-1: **xc7k325t-2-ffg900 FPGA Resource Estimates**

Parameter Values (other parameters have default values)		Device Resources	
C_USE_UART	C_MB_DBG_PORTS	Flip-Flops	LUTs
0	1	69	49
0	2	71	60
0	4	73	78
0	32	125	336
1	1	131	140

Port Descriptions

The I/O signals for the MicroBlaze Debug Module (MDM) are listed and described in Table 2-2.

Table 2-2: **MDM I/O Signals**

Signal Name	Interface	I/O	Initial State	Description
System Signals				
Interrupt		O	0	Interrupt from UART
Debug_SYS_Rst		O	0	Debug system reset
Ext_BRK		O	0	External break
Ext_NM_BRK		O	0	External non-maskable break
AXI4-Lite Interface Signals				
S_AXI_ACLK	S_AXI	I	-	AXI Clock
S_AXI_ARESETN	S_AXI	I	-	AXI Reset, active-Low
S_AXI_AWADDR[C_S_AXI_ADDR_WIDTH-1:0]	S_AXI	I	-	Write Address
S_AXI_AWVALID	S_AXI	I	-	Write Address Valid
S_AXI_AWREADY	S_AXI	O	0	Write Address Ready
S_AXI_AWDATA[C_S_AXI_DATA_WIDTH-1:0]	S_AXI	I	-	Write Data
S_AXI_AWSTB[C_S_AXI_DATA_WIDTH/8-1:0]	S_AXI	I	-	Write Strobes
S_AXI_WVALID	S_AXI	I	-	Write Valid
S_AXI_WREADY	S_AXI	O	0	Write Ready
S_AXI_BRESP[1:0]	S_AXI	O	0	Write Response
S_AXI_BVALID	S_AXI	O	0	Write Response Valid
S_AXI_BREADY	S_AXI	I	-	Write Response Ready
S_AXI_ARADDR[C_S_AXI_ADDR_WIDTH-1:0]	S_AXI	I	-	Read Address
S_AXI_ARVALID	S_AXI	I	-	Read Address Valid
S_AXI_ARREADY	S_AXI	O	0	Read Address Ready
S_AXI_RRESP[1:0]	S_AXI	O	0	Read Response
S_AXI_RVALID	S_AXI	O	0	Read Valid
S_AXI_RREADY	S_AXI	I	-	Read Ready
MicroBlaze Debug Interface Signals				
Dbg_Clk_n	MBDEBUG_n	O	0	MicroBlaze Debug Clock
Dbg_TDI_n	MBDEBUG_n	O	0	MicroBlaze Debug TDI
Dbg_TDO_n	MBDEBUG_n	I	-	MicroBlaze Debug TDO

Table 2-2: **MDM I/O Signals *(Cont'd)***

Signal Name	Interface	I/O	Initial State	Description
Dbg_Reg_En_n	MBDEBUG_n	O	0	MicroBlaze Debug Register Enable
Dbg_Capture_n	MBDEBUG_n	O	0	MicroBlaze Debug Capture
Dbg_Shift_n	MBDEBUG_n	O	0	MicroBlaze Debug Shift
Dbg_Update_n	MBDEBUG_n	O	0	MicroBlaze Debug Update
Dbg_Rst_n	MBDEBUG_n	O	0	MicroBlaze Debug Reset
MicroBlaze Trace Core Interface Signals				
Ext_JTAG_DRCK	XMTC	O	0	Connection to MicroBlaze Trace Core
Ext_JTAG_RESET	XMTC	O	0	Connection to MicroBlaze Trace Core
Ext_JTAG_SEL	XMTC	O	0	Connection to MicroBlaze Trace Core
Ext_JTAG_CAPTURE	XMTC	O	0	Connection to MicroBlaze Trace Core
Ext_JTAG_SHIFT	XMTC	O	0	Connection to MicroBlaze Trace Core
Ext_JTAG_UPDATE	XMTC	O	0	Connection to MicroBlaze Trace Core
Ext_JTAG_TDI	XMTC	O	0	Connection to MicroBlaze Trace Core
Ext_JTAG_TDO	XMTC	I	-	Connection to MicroBlaze Trace Core
External BSCAN Interface Signals				
bscan_tdi	BSCAN	I	-	Connection to external BSCAN
bscan_reset	BSCAN	I	-	Connection to external BSCAN
bscan_shift	BSCAN	I	-	Connection to external BSCAN
bscan_update	BSCAN	I	-	Connection to external BSCAN
bscan_capture	BSCAN	I	-	Connection to external BSCAN
bscan_sel	BSCAN	I	-	Connection to external BSCAN
bscan_drck	BSCAN	I	-	Connection to external BSCAN
bscan_tdo	BSCAN	O	0	Connection to external BSCAN

Register Space

The MDM registers are listed and described in Table 2-3.

Table 2-3: **MDM Registers**

Register Name	Size (bits)	Address Offset	Initial State	Description
Rx_FIFO	8	0	0	JTAG UART receive data
Tx_FIFO	8	4	0	JTAG UART transmit data
Status_reg	8	8	0x04	Read only bit 7 rx_Data_Present bit 6 rx_Buffer_Full bit 5 tx_Buffer_Empty bit 4 tx_Buffer_Full bit 3 enable_interrupts
Ctrl_reg	8	C	0x03	Write only bit 3 enable_interrupts bit 5 Clear Ext BRK signal bit 6 Reset_RX_FIFO bit 7 Reset_TX_FIFO

MDM Interrupts

If the interrupt enable register bit in the control register is set, the UART raises the interrupt signal in the cycle when the TX FIFO goes empty, or in every cycle where the RX FIFO has data available.

Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the core.

General Design Guidelines

The MicroBlaze™ Debug Module supports multiple MicroBlaze cores, making it possible to use one MDM for multiprocessor systems with up to 32 processors.

The MDM outputs, `Ext_BRK` and `Ext_NM_BRK`, are not currently used, and need not be connected to MicroBlaze.

When using the JTAG-based UART, the MDM Interrupt output can be connected to an interrupt controller to provide interrupt-driven serial output. If this signal is not connected, only polled-mode serial output is available.

Clocking

The `S_AXI_ACLK` input is only used when the JTAG-based UART is enabled, and AXI4-Lite interconnect is used. Then it should normally be set to the same clock as the interconnect.

Apart from the JTAG-based UART, the MDM is clocked from the BSCAN, with a clock frequency determined by the JTAG connection.

When programming a System ACE™ device, the MDM clock must be at least twice as fast as the System ACE tool controller clock for the ELF file to load correctly.

Resets

The `Debug_SYS_Rst` output can be used to reset the entire embedded system on the FPGA, including all processors and peripherals. Normally it is connected to a

proc_sys_reset IP core. The Xilinx Microprocessor Debugger (XMD) command rst can be used to activate the signal.

The Debug bus connecting each individual MicroBlaze processor handled by the MDM, has the Dbg_Rst reset signal. This signal can be used to just reset an individual processor. The XMD command rst -processor can be used to activate the signal for the selected target processor.

The S_AXI_ARESETN input is only used when the JTAG-based UART is enabled, and AXI4-Lite interconnect is used. Then it should normally be set to the same reset as the interconnect.

Protocol Description

The MDM Debug protocol is Xilinx internal, and not described in this document. All the details of the protocol are handled transparently to the user by the Xilinx Microprocessor Debugger.

Customizing and Generating the Core

This chapter includes information on using Xilinx tools to customize and generate the core in the Vivado™ Design Suite.

GUI

The MDM parameters are divided into three categories: Debug, UART and Advanced. When using the Vivado IP Integrator, the address for the JTAG-based UART is auto-generated.

The Customize IP dialog box is shown in Figure 4-1.

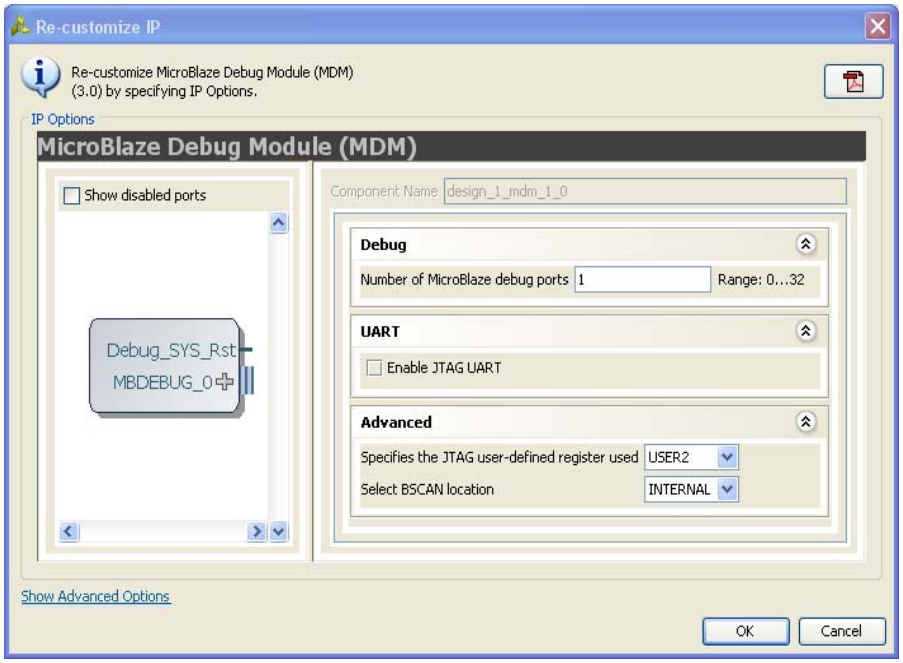

Figure 4-1: **Customize IP Dialog Box**

- **Number of MicroBlaze debug ports** - Sets the number of ports available to connect to MicroBlaze processors.
- **Enable JTAG UART** - Enables the JTAG UART and the AXI4-Lite interconnect to access the UART registers.

- **Specify the JTAG user-defined register used** - Select JTAG user-defined register. Can be set to *USER1*, *USER2*, *USER3* or *USER4*. Should never need to be changed from *USER2*, unless there is a conflict with another IP core in the system.
- **Select BSCAN location** - Selects whether internal or external BSCAN is used. Should never need to be changed from *INTERNAL* in an embedded system.

Parameter Values

To allow the user to obtain an MDM that is uniquely tailored a specific system, certain features can be parameterized in the MDM design. This allows the user to configure a design that only utilizes the resources required by the system. The features that can be parameterized in MDM designs are shown in Table 4-1.

Table 4-1: **MDM Design Parameters**

Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
System Parameters				
Target FPGA family	C_FAMILY	See LogiCORE IP Facts Table	virtex7	string
Debug Parameters				
Number of MicroBlaze debug ports	C_MB_DBG_PORTS	0-32	1	integer
Position in the FPGAJTAG chain	C_JTAG_CHAIN	1 = USER1 2 = USER2 3 = USER3 4 = USER4	2	integer
BSCAN selection	C_USE_BSCAN	0 = INTERNAL 1 = ICON 2 = EXTERNAL	0	integer
UART Parameters				
Enables the UART interface	C_USE_UART	0,1	1	integer

In addition to the parameters listed in this table, there are also parameters that are inferred for each AXI interface in the tools. Through the design, these inferred parameters control the behavior of the AXI Interconnect. For a complete list of the interconnect settings related to the AXI interface, see the *LogiCORE AXI Interconnect IP Data Sheet* [Ref 4].

Allowable Parameter Combinations

There are no restrictions on parameter combinations for this core.

Output Generation

The following files are generated by the IP in Vivado IP Integrator:

- Verilog/VHDL Template
- VHDL source files
- VHDL wrapper file in the library work

Constraining the Core

Required Constraints

There are no required constraints for this core.

Device, Package, and Speed Grade Selections

There are no Device, Package or Speed Grade requirements for this core.

Clock Frequencies

There are no specific clock frequency requirements for this core.

Clock Management

The MDM Debug logic is fully synchronous with the BSCAN module. Internally BUFG primitives are instantiated to buffer the DRCK clock from the BSCAN module.

The MDM JTAG-based UART is fully synchronous to the bus interface clock, and isolated from the BSCAN clock region with FIFOs.

Clock Placement

There are no specific Clock placement requirements for this core.

Banking

There are no specific Banking rules for this core.

Transceiver Placement

There are no Transceiver Placement requirements for this core.

I/O Standard and Placement

There are no specific I/O standards and placement requirements for this core.

XILINX®

Appendix A

Migrating

This appendix describes migrating from older versions of the IP to the current IP release.

For information on migrating to the Vivado™ Design Suite, see the *Vivado Design Suite Migration Methodology Guide* [Ref 3].

Debugging

This appendix includes details about resources available on the Xilinx Support website and debugging tools. In addition, this appendix provides a step-by-step debugging process to guide you through debugging the MDM core.

The following topics are included in this appendix:

Finding Help on Xilinx.com

To help in the design and debug process when using the MDM core, the Xilinx Support web page (www.xilinx.com/support) contains key resources such as product documentation, release notes, answer records, information about known issues, and links for opening a Technical Support WebCase.

Documentation

This product guide is the main document associated with the MDM core. This guide, along with documentation related to all products that aid in the design process, can be found on the Xilinx Support web page (www.xilinx.com/support) or by using the Xilinx Documentation Navigator.

Download the Xilinx Documentation Navigator from the Design Tools tab on the Downloads page (www.xilinx.com/download). For more information about this tool and the features available, open the online help after installation.

Master Answer Record for the MDM Core

AR 54413

Contacting Technical Support

Xilinx provides technical support at www.xilinx.com/support for this LogiCORE™ IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled DO NOT MODIFY.

To contact Xilinx Technical Support:

1. Navigate to www.xilinx.com/support.
2. Open a WebCase by selecting the WebCase link located under Support Quick Links.

When opening a WebCase, include:

- Target FPGA including package and speed grade.
- All applicable Xilinx Design Tools and simulator software versions.
- Additional files based on the specific issue might also be required. See the relevant sections in this debug guide for guidelines about which file(s) to include with the WebCase.

Debug Tools

The main tools available to address MDM design issues are the Vivado Lab tools.

Vivado Lab Tools

Vivado inserts logic analyzer and virtual I/O cores directly into your design. Vivado Lab Tools allows you to set trigger conditions to capture application and integrated block port signals in hardware. Captured signals can then be analyzed. This feature represents the functionality in the Vivado IDE that is used for logic debugging and validation of a design running in Xilinx FPGAs in hardware.

The Vivado logic analyzer is used to interact with the logic debug LogiCORE IP cores, including:

- ILA 2.0 (and later versions)
- VIO 2.0 (and later versions)

Reference Boards

All 7-Series Xilinx development boards support MDM. These boards can be used to prototype designs and establish that the core can communicate with the system.

Simulation Debug

The simulation debug flow for Questa SIM is described below. A similar approach can be used with other simulators.

- Check for the latest supported versions of Questa SIM in the Xilinx Design Tools: Release Notes Guide. Is this version being used? If not, update to this version.
- If using Verilog, do you have a mixed mode simulation license? If not, obtain a mixed-mode license.
- Ensure that the proper libraries are compiled and mapped. In the Vivado Design Suite **Flow > Simulation Settings** can be used to define the libraries.
- Have you associated the intended software program for the MicroBlaze™ processor with the simulation? Use the command **Tools > Associate ELF Files** in Vivado Design Suite.
- When observing the traffic on the interfaces connected to the MDM, see the timing in the relevant specification:
  - For AXI4-Lite, see the *AMBA® AXI and ACE Protocol Specification* [Ref 1].

Hardware Debug

Hardware issues can range from link bring-up to problems seen after hours of testing. This section provides debug steps for common issues. The Vivado Lab Tools are valuable resources to use in hardware debug. The signal names mentioned in the following individual sections can be probed using the Vivado Lab Tools for debugging the specific problems.

Many of these common issues can also be applied to debugging design simulations.

General Checks

Ensure that all the timing constraints for the core were properly incorporated from the example design and that all constraints were met during implementation.

- Does it work in post-place and route timing simulation? If problems are seen in hardware but not in timing simulation, this could indicate a PCB issue. Ensure that all clock sources are active and clean.
- If using MMCMs in the design, ensure that all MMCMs have obtained lock by monitoring the `LOCKED` port.

Interface Debug

AXI4-Lite Interfaces

Read from a register that does not have all 0s as a default to verify that the interface is functional. Output `Sn_AXI_ARREADY` asserts when the read address is valid, and output `Sn_AXI_RVALID` asserts when the read data/response is valid, where *n* is the interface number (0 or 1). If the interface is unresponsive, ensure that the following conditions are met:

- The `Sn_AXI_ACLK` input is connected and toggling.
- The interface is not being held in reset, and `Sn_AXI_ARESETN` is an active-Low reset.
- The common core reset is not active, and `SYS_Rst` is an active-High reset.
- If the simulation has been run, verify in simulation and/or a Vivado Lab Tools debugging tool capture that the waveform is correct for accessing the AXI4-Lite interface.

Application Software Development

Device Drivers

The MDM JTAG-based UART interface is supported by the UART Lite driver, included with Xilinx Software Development Kit.

Appendix D

Additional Resources

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see the Xilinx Support website at:

www.xilinx.com/support.

For a glossary of technical terms used in Xilinx documentation, see:

www.xilinx.com/company/terms.htm.

References

These documents provide supplemental material useful with this user guide:

1. AMBA AXI and ACE Protocol Specification (ARM IHI 0022D)
2. Vivado™ Design Suite user documentation
3. Vivado Design Suite Migration Methodology Guide (UG911)
4. LogiCORE AXI Interconnect IP Data Sheet (UG768)
5. Vivado Design Suite User Guide, Designing with IP (UG896)
6. Xilinx AXI Reference Guide (UG761)

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
03/20/2013	1.0	This Product Guide replaces PG062. There are no documentation changes for this release.

Notice of Disclaimer