RAMA v1.0

LogiCORE IP Product Guide

Vivado Design Suite

PG310 (v1.0) April 4, 2018

XILINX

ALL PROGRAMMABLE™

Table of Contents

Chapter 1: IP Facts

Chapter 1

IP Facts

Features

- Performance improvement for random access memory, measured relative to the design without RAMA IP. The following table uses an improvement multiplier rather than the efficiency figure. For example, for 64 B read only transactions, the measured bandwidth without RAMA is 4225 MB/s, while with RAMA it is 40730 MB/s, thus an almost 10 times improvement in bandwidth.

Access Type	32 B	64 B	128 B	256 B	512 B
Read Only	10	10	5	3	2
Write Only	2	2	1.5	1	1
Read/Write	3	3	2	1	1

- AXI-4 interface on user's side, AXI3 interface on HBM side (256 bits). AXI-3 is used by the hardened AXI-3 Interconnect Switch to access HBM.

- Operating clock frequency (single clock domain): 450 MHz (420 MHz for -1 parts and 350 MHz for -2LV parts).

- Address width: Up to 33 bits.

- ID Width: Up to 6 bits.

IP Facts

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family[1]	Virtex® UltraScale+™ with HBM
Supported User Interfaces	AXI4 on Slave side AXI3 on Master side
Resources	Resource Use

PG310 (v1.0) April 4, 2018
Random Access Master Attachment IP

www.xilinx.com

Send Feedback

4

LogiCORE IP Facts Table	
Provided with Core	
Design Files	Encrypted RTL
Example Design	Not Provided
Test Bench	VHDL
Constraints File	Not Provided
Simulation Model	Not Provided
Supported S/W Driver	Not Applicable
Tested Design Flows[2]	
Design Entry	Vivado® Design Suite
Simulation	For supported simulators, see the Xilinx Design Tools: Release Notes Guide.
Synthesis	Vivado® Synthesis
Support	
Provided by Xilinx® at the Xilinx Support web page	

Notes:

1. For a complete list of supported devices, see the Vivado IP catalog.
2. For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.

Overview

The High Bandwidth Memory (HBM) subsystem in Virtex® UltraScale+™ devices performs well in applications where sequential data access is required. However, for applications requiring random data access, performance can vary significantly depending on the application requirements (for example, the ratio of read and write operations, minimum data word size, and memory size). The RAMA IP addresses such problems by significantly improving memory access efficiency in cases where the required memory exceeds 256 MB (one HBM pseudo-channel).

The following figure describes the connections between programmable logic masters and HBM.

Figure 1: **HBM Two Stack Configuration**
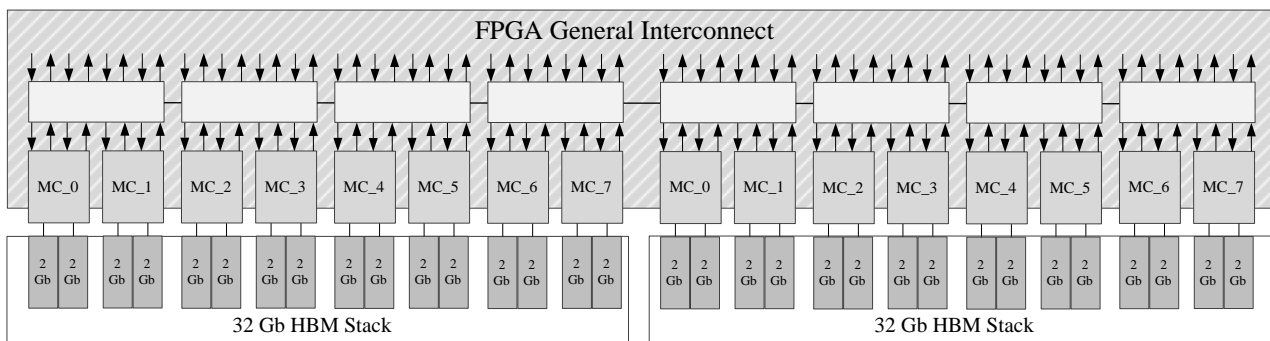
In the most common use case, where the master's transactions use a single AXI ID, the hardened AXI3 Interconnect switch, used to connect masters to the HBM, has the limitation that any master can have an outstanding transaction with only one slave. This limitation can significantly reduce data bandwidth when masters frequently switch between slaves. This effect is greater for read access, due to longer response times.

The RAMA IP uses AXI ID substitution and response reordering to provide memory access performance improvements for random data access in all cases where more than one pseudo-channel of the HBM stack is accessed by a single master. The greatest advantage is achieved for read intensive data accesses.

Functional Description

The main functions of the RAMA IP are as follows:

- Receive AXI4 Read/Write transactions on the AXI4 interface.

- Resize AXI transactions to a consistent burst size (burst fragment) of 64 or 128 Bytes to and from the HBM subsystem. Bursts are resized for the following reasons:

   - The maximum permissible burst size for the HBM Subsystem is 512 Bytes.

   - Reordering transactions is more efficient (buffer sizing) with smaller transactions.

   - Resizing bursts allows for transactions to be spread across memories, limiting congestion at a particular memory.

Note: Fragmentation will always occur (i.e. the maximum transaction size seen at the HBM Subsystem will be the same as the "burst fragment size").

- Interleave burst fragments across HBM memories to prevent short-term congestion on a particular memory. This is an optional feature.

- ID substitution is applied to AXI `AWID/ARID` fields for each burst fragment.

   - For read transactions: Implement read ID substitution to supply different read IDs on different transactions. This prevents bandwidth limitations as discussed in the Overview section. Reorder returning read data to obey AXI transaction rules.

   - For write transactions: Implement write ID substitution to supply different write IDs on different transactions. This prevents bandwidth limitations as discussed in the Overview section. Reorder write responses to obey AXI transaction rules.

- Generate read/write AXI3 burst fragment transactions to/from HBM Subsystem via AXI3 Master interface.

Related Information
Overview

Applications

The RAMA IP provides performance improvements in HBM random memory access efficiency in applications where memory space is greater than 256 MB. See AXI Thread IDs for the additional conditions required to maximize the RAMA IP's performance improvement.

The RAMA IP can additionally improve performance in applications where multiple masters share access to HBM pseudo-channels. The memory interleaving option can be used to stripe data across more pseudo-channels than required (in terms of memory capacity) increasing the number of pseudo-channels used. This increases the total memory bandwidth available and in turn increases the performance.

This feature can also be used to limit congestion on an individual pseudo-channel. Consider a case where long transactions occur from a particular AXI master. These long transactions can be split across multiple pseudo-channels balancing the load on the HBM Subsystem and preventing instantaneous congestion on a single pseudo-channel.

Related Information
AXI Thread IDs

Unsupported Features

The RAMA IP does not support *FIXED* AXI burst types on its master interface. The expected default transaction type is *INCR*. If *WRAP* transaction type is required, this can be enabled using an option on the GUI. For more details on transaction types please refer to ARM® AMBA AXI Protocol v2.0.

Licensing and Ordering

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

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

Product Specification

Standards

This core adheres to the following standard(s):

- Advanced Microcontroller Bus Architecture (AMBA®) AXI version 4 specification from Advanced RISC Machine (ARM®). See ARM® AMBA AXI Protocol v2.0.

Performance Improvement

The following table for an example system uses an improvement multiplier rather than the efficiency figure. For example, for 64 B read only transactions, the measured bandwidth without RAMA is 4225 MB/s, while with RAMA it is 40730 MB/s, thus an almost 10 times improvement in bandwidth.

Access Type	32 B	64 B	128 B	256 B	512 B
Read Only	10	10	5	3	2
Write Only	2	2	1.5	1	1
Read/Write	3	3	2	1	1

Note: Results shown are for a specific test scenario with four AXI masters, each randomly accessing four HBM pseudo-channels. The relative improvements quoted are the results with RAMA IP on each master compared to without RAMA IP on each master.

Latency

Latency figures should be considered carefully. The RAMA IP adds latency to an individual transaction due to data buffering and re-ordering. However, due to bandwidth improvements, using the RAMA IP means the time between transaction request and completion is, in general, much shorter. The following table below shows mean latency figures for 2000 Read Only and Write Only transactions.

Table 1: **RAMA IP Latency**

Transaction Size	Read Only (AXI Clock Cycles)		Write Only (AXI Clock Cycles)	
	Without RAMA	With RAMA	Without RAMA	With RAMA
32	225	597	44	591
64	247	532	46	134
128	240	497	55	49
256	263	512	77	78
512	304	564	119	137

To illustrate why latency figures can be misleading, consider the following: a given number of read transactions of 32 bytes in size may take 100 μs to complete without RAMA. This means that the last transaction would be delayed by almost 100 μs after it could have been issued by the master. Since bandwidth is 10 times better for 32 bytes using RAMA , the same number of transactions would be completed within 10 μs, plus latency added by buffering and reordering in the RAMA IP (in this case typically 1.3 μs).

Resource Use

The following table provides resource use information for the RAMA IP on a Virtex® UltraScale +™ HBM device (VU31P). These values were generated using the Vivado® IP catalog. They are derived from post-synthesis reports, and may change during implementation.

Table 2: **Resource Use: UltraScale+ HBM Devices**

Parameter Values	Device Resources			
Reorder Memory Type	LUTs	FFs	BRAMs	URAMs
BRAM	2000	2900	8	0
URAM	2000	2900	0	4

Port Descriptions

The interface I/O signals for the Random Access Master Attachment LogiCORE IP are shown in the following tables.

AXI4 RAMA Slave Interface I/O Signals

Table 3: **RAMA Slave Interface I/O Signals**

Port Name	I/O	Description
S_AXI_AWID	I	Write Address Channel Transaction ID Default = 0
S_AXI_AWADDR[32:0]	I	Write Address Channel Address Input Required
S_AXI_AWLEN[7:0]	I	Write Address Channel Burst Length (0-255) Default = 0
S_AXI_AWSIZE[2:0]	I	Write Address Channel Transfer Size code (0–7) Input Required
S_AXI_AWBURST[1:0]	I	Write Address Channel Burst Type code (0-2) Input Required
S_AXI_AWVALID	I	Write Address Channel Valid Input Required
S_AXI_AWREADY	O	Write Address Channel Ready
S_AXI_WDATA[255:0]	I	Write Data Channel Data Input Required
S_AXI_WSTRB[31:0]	I	Write Data Channel Byte Strobes Default = All ones
S_AXI_WLAST	O	Write Data Channel Last Data Beat Default = 0
S_AXI_WVALID	I	Write Data Channel Valid Input Required
S_AXI_WREADY	O	Write Data Channel Ready
S_AXI_BID	O	Write Response Channel Transaction ID
S_AXI_BRESP[1:0]	O	Write Response Channel Response Code (0–3)
S_AXI_BVALID	O	Write Response Channel Valid
S_AXI_BREADY	I	Write Response Channel Ready Input Required
S_AXI_ARID	O	Read Address Channel Transaction ID Default = 0
S_AXI_ARADDR[32:0]	I	Read Address Channel Address Input Required
S_AXI_ARLEN[7:0]	I	Read Address Channel Burst Length code (0–255) Default = 0
S_AXI_ARSIZE[2:0]	I	Read Address Channel Transfer Size code (0–7) Input Required
S_AXI_ARBURST[1:0]	I	Read Address Channel Burst Type (0–2) Input Required
S_AXI_ARVALID	I	Read Address Channel Valid Input Required
S_AXI_ARREADY	O	Read Address Channel Ready
S_AXI_RID	O	Read Data Channel Transaction ID

Table 3: **RAMA Slave Interface I/O Signals** *(cont'd)*

Port Name	I/O	Description
S_AXI_RDATA[255:0]	O	Read Data Channel Data
S_AXI_RRESP[1:0]	O	Read Data Channel Response Code (0-3)
S_AXI_RLAST	O	Read Data Channel Last Data Beat
S_AXI_RVALID	O	Read Data Channel Valid
S_AXI_RREADY	I	Read Data Channel Ready Input Required

AXI3 RAMA Master Interface I/O Signals

Table 4: **RAMA Master Interface I/O Signals**

Port Name	I/O	Description
M_AXI_AWID[5:0]	O	Write Address Channel Transaction ID
M_AXI_AWADDR[32:0]	O	Write Address Channel Address
M_AXI_AWLEN[7:0]	O	Write Address Channel Burst Length code (0–255)
M_AXI_AWSIZE[2:0]	O	Write Address Channel Transfer Size code (0–7)
M_AXI_AWBURST[1:0]	O	Write Address Channel Burst Type (0–2)
M_AXI_AWVALID	O	Write Address Channel Valid
M_AXI_AWREADY	I	Write Address Channel Ready Input Required
M_AXI_WDATA[255:0]	O	Write Data Channel Data
M_AXI_WSTRB[31:0]	O	Write Data Channel Data Byte Strobes
M_AXI_WLAST	O	Write Data Channel Last Data Beat
M_AXI_WVALID	O	Write Data Channel Valid
M_AXI_WREADY	I	Write Data Channel Ready Input Required
M_AXI_BID[5:0]	I	Write Response Channel Transaction ID Input Required
M_AXI_BRESP[1:0]	I	Write Response Channel Response Code (0–3) Default = 0
M_AXI_BVALID	I	Write Response Channel Valid Input Required
M_AXI_BREADY	O	Write Response Channel Ready
M_AXI_ARID[5:0]	O	Read Address Channel Transaction ID
M_AXI_ARADDR[32:0]	O	Read Address Channel Address
M_AXI_ARLEN[7:0]	O	Read Address Channel Burst Length code (0–255)
M_AXI_ARSIZE[2:0]	O	Read Address Channel Transfer Size code (0–7)
M_AXI_ARBURST[1:0]	O	Read Address Channel Burst Type (0–2)
M_AXI_ARVALID	O	Read Address Channel Valid
M_AXI_ARREADY	I	Read Address Channel Ready Input Required

Table 4: **RAMA Master Interface I/O Signals** *(cont'd)*

Port Name	I/O	Description
M_AXI_RID[5:0]	I	Read Data Channel Transaction ID
M_AXI_RDATA[255:0]	I	Read Data Channel Data Input Required
M_AXI_RRESP[1:0]	I	Read Data Channel Response Code (0–3) Default = 0
M_AXI_RLAST	I	Read Data Channel Last Data Beat Input Required
M_AXI_RVALID	I	Read Data Channel Valid Input Required
M_AXI_RREADY	O	Read Data Channel Ready

Global Port Signals

Table 5: **Clocks and Resets**

Port Name	I/O	Description
AXI_ACLK	I	Crossbar clock input
AXI_ARESETN	I	Crossbar Reset (active-low)

Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the Random Access Master Attachment LogiCORE IP core.

Clocking

A single clock domain, `AXI_ACLK`, is used. The design has been verified for clock frequencies up to 450 MHz , which is the nominal frequency for the HBM Subsystem AXI switch (see Features section for lower speed grades). Both interfaces, slave and master, must operate in the same clocking domain. For adapting clock domain and data width to RAMA requirements, see the *SmartConnect LogiCORE IP Product Guide* (PG247).

Note: The Smartconnect IP should not be used between the RAMA IP and the HBM Subsystem IP as the AXI ID multi-threading approach used in RAMA will cause the Smartconnect IP to consume large amounts of logic resources.

Any required soft switching should take place in front of the slave port of the RAMA IP.

Resets

Only the active-low reset, `AXI_ARESETN`, is used. This must be synchronous with `AXI_ACLK`.

The reset clears all internal buffer pointers, thus RAMA appears to the user as though it had all internal data cleared.

Addressing

The RAMA IP is transparent to all slave address segments. Slave address segments can be mapped from an attached slave through the RAMA IP to an attached master. No range checking is carried out in the RAMA IP on address segments mapped versus the address in a transaction.

Send Feedback

An address re-mapping is carried out by the *Memory Interleave* function of the RAMA IP. For this case multiple HBM peudo-channels are reorganized such that the pseudo-channels are interleaved per burst fragment.

It is the responsibility of the user to ensure that the correct memory segments are mapped to the AXI master in order to correctly use the memory interleaving feature. For example, if memory interleaving across four HBM pseudo-channels is selected but only two pseudo-channels are enabled on the HBM IP, the RAMA IP will not detect the error at build time.

Error Handling

The RAMA IP will generate AXI SLVERR on read/write response channels in the following scenarios:

- An AXI FIXED burst transaction is indicated on the read/write channels of the RAMA IP.
- An AXI WRAP burst transaction is indicated on the read/write channels of the RAMA IP when WRAP burst transaction handling is not enabled.
- An AXI SLVERR is indicated by the attached AXI slave.

The RAMA IP generates AXI DECERR on read/write response channels when AXI DECERR is indicated by the attached AXI slave.

AXI Thread IDs

To use the RAMA IP, an AXI master should meet one of the following criteria:

- Use a static, single ID on the AXI transaction ID ports (AxID).
- Use slow-changing (pseudo-static) transaction IDs.

If neither of these conditions are met, the thread creation used in the RAMA IP to improve performance will have little effect.

AXI - Unsupported Signaling

The AXI4 slave port on RAMA does not support the following features:

- AXI region identifiers (`AxREGION`)

- AXI cached/buffered transfers (`AxCACHE`)

- AXI protected support (`AxPROT`). The HBM Memory Subsystem does not support secure / non-secure access differentiation.

- AXI exclusive access (`AxLOCK`). The HBM Memory Subsystem does not support exclusive accesses.

- AXI quality of service (`AxQOS`). The HBM Memory Subsystem does not support quality of service identifiers.

- AXI user sideband signaling (`AxUSER`). The RAMA IP does not support optional user signaling.

Design Flow Steps

This section describes customizing and generating the core, constraining the core, and the simulation, synthesis and implementation steps that are specific to this IP core. More detailed information about the standard Vivado® design flows and the IP integrator can be found in the following Vivado Design Suite user guides:

- *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994)
- *Vivado Design Suite User Guide: Designing with IP* (UG896)
- *Vivado Design Suite User Guide: Getting Started* (UG910)
- *Vivado Design Suite User Guide: Logic Simulation* (UG900)

Customizing and Generating the Core

This section includes information about using Xilinx® tools to customize and generate the core in the Vivado® Design Suite.

If you are customizing and generating the core in the Vivado IP integrator, see the *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994) for detailed information. IP integrator might auto-compute certain configuration values when validating or generating the design. To check whether the values do change, see the description of the parameter in this chapter. To view the parameter value, run the `validate_bd_design` command in the Tcl console.

You can customize the IP for use in your design by specifying values for the various parameters associated with the IP core using the following steps:

1. Select the IP from the IP catalog.
2. Double-click the selected IP or select the Customize IP command from the toolbar or right-click menu.
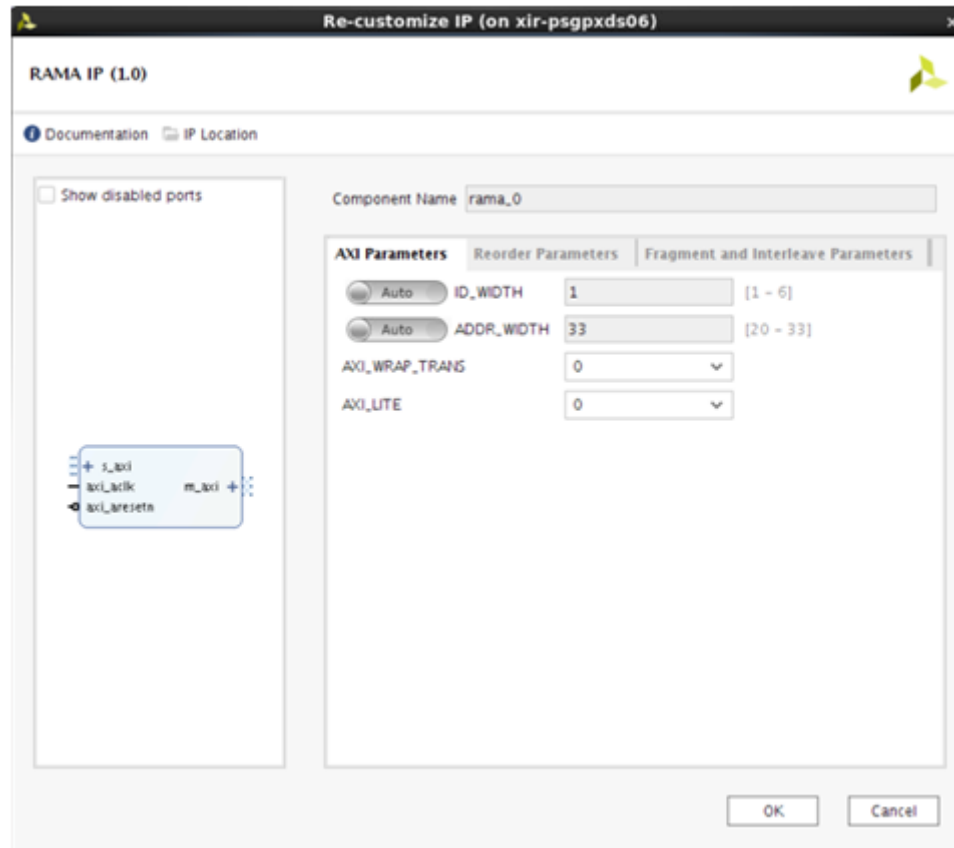
For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) and the *Vivado Design Suite User Guide: Getting Started* (UG910).

Figures in this chapter are illustrations of the Vivado IDE. The layout depicted here might vary from the current version.

AXI Parameters Tab

The AXI Parameters tab is shown below:

Figure 3: **AXI Parameters Tab**

- **ID_WIDTH:** The ID width is automatically propagated from the master by default. This can be overridden by the user by using the switch in the GUI.

- **ADDR_WIDTH:** The address width is automatically propagated from the master by default. This can be overridden by the user by using the switch in the GUI.

- **AXI_WRAP_TRANS:** This indicates whether AXI-4 wrap transactions are allowed:

  - 1 – Yes

  - 0 – No

  By default only AXI-4 INCR transactions are supported. AXI-4 FIXED transactions are never supported.

Reorder Parameters Tab

The Reorder Parameters tab is shown below:

Figure 4: **Reorder Parameters Tab**
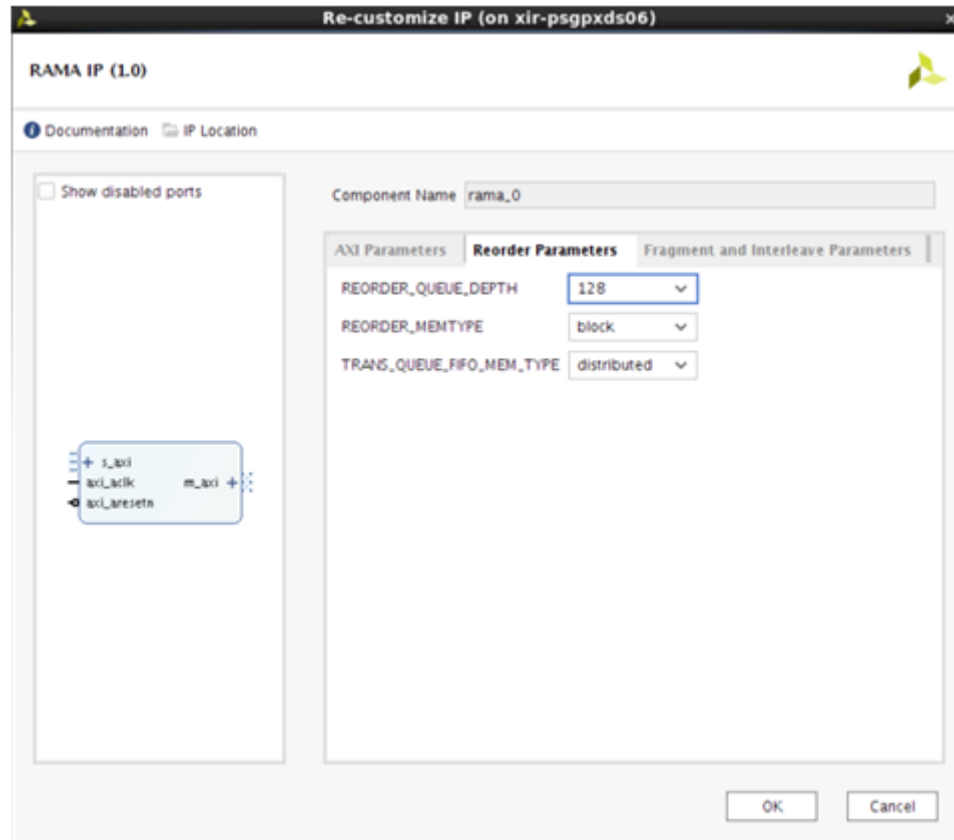
- **REORDER_QUEUE_DEPTH:** The depth of the RAMA reorder queue in "burst fragments". Increasing the depth of the queue may increase performance depending on the traffic profile. The burst fragment size is specified by the FRAGMENT_SIZE_BYTES parameter.

- **REORDER_MEMTYPE:** Reorder Queue Memory Type:
  - block – BRAM
  - ultra – URAM

- **TRANS_QUEUE_FIFO_MEM_TYPE:** Transaction Queue FIFO Memory Type:
  - distributed – distributed RAM
  - block – BRAM
  - ultra – URAM

www.xilinx.com

Send Feedback

Fragment and Interleave Parameters

The Fragment and Interleave Parameters tab is shown below:

Figure 5: **Fragment and Interleave Parameters**
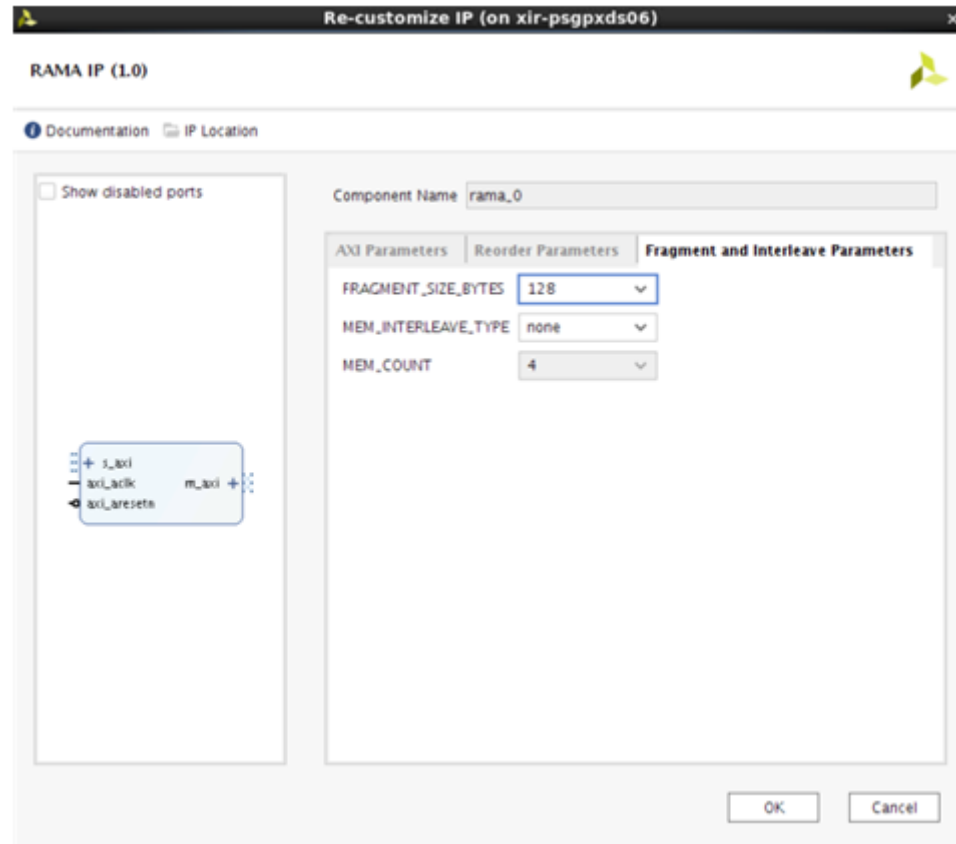
- **FRAGMENT_SIZE_BYTES:**

  Specifies the burst fragment size in bytes. If the number of bytes to be transferred in a transaction is greater than the burst fragment size, the AXI transaction is split into multiple burst fragments of a consistent size to allow easier reordering.

- **MEM_INTERLEAVE_TYPE:** Memory interleaving allows transactions from a single master to be spread across multiple HBM pseudo-channels to distribute the memory load:
  - none – no pseudo-channel interleaving.
  - per memory – consecutive burst fragments are written to consecutive HBM pseudo channels.
- **MEM_COUNT:** This is the number of pseudo-channels to interleave across.

Send Feedback

Memory Interleave Example

Take a case where four AXI Masters need to access 512 MB. 512 MB corresponds to 2 HBM pseudo-channels. It may be more advantageous to use four HBM pseudo-channels in this case in order to have twice the memory bandwidth for the four masters.

In order to do this, MEM_INTERLEAVE_TYPE = per_memory would be selected with MEM_COUNT = 4.

The 512 MB would be composed of 128 MB from each HBM pseudo-channel with access alternating/"striped" across the pseudo-channels on a "burst fragment" basis.

Output Generation

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896).

User Parameters

The user parameters for the RAMA IP are shown in the following table:

Table 6: **RAMA IP User Parameters**

Vivado IDE/User Parameter	Value	Default Value
FRAGMENT_SIZE_BYTES	Integer (64, 128)	128
MEM_INTERLEAVE_TYPE	String ("per_memory", "none")	"none"
MEM_COUNT	Integer (2, 4, 8, 16, 32)	4
REORDER_QUEUE_DEPTH	Integer (128, 256)	128
TRANS_QUEUE_FIFO_MEM_TYPE	String ("distributed", "block", "ultra")	"distributed"
REORDER_MEMTYPE	String ("block", "ultra")	"block"
AXI_WRAP_TRANS	Integer (0,1)	0

Constraining the Core

Required Constraints

This section is not applicable for this IP core.

Device, Package, and Speed Grade Selections

All Xilinx® UltraScale+™ devices with High Bandwidth Memory (HBM) can use RAMA IP.

Clock Frequencies

`AXI_ACLK` has a nominal frequency of 450 MHz. A lower clock frequency may be used, with the consequence that HBM access efficiency would be reduced.

Clock Management

This section is not applicable for this IP core.

Clock Placement

This section is not applicable for this IP core.

Banking

This section is not applicable for this IP core.

Transceiver Placement

This section is not applicable for this IP core.

I/O Standard and Placement

This section is not applicable for this IP core.

Simulation

For comprehensive information about Vivado® simulation components, as well as information about using supported third-party tools, see the *Vivado Design Suite User Guide: Logic Simulation* (UG900).

Synthesis and Implementation

For details about synthesis and implementation, see the *Vivado Design Suite User Guide: Designing with IP* (UG896).

Debugging

This appendix includes details about resources available on the Xilinx® Support website and debugging tools.

Finding Help on Xilinx.com

To help in the design and debug process when using the core, the Xilinx Support web page contains key resources such as product documentation, release notes, answer records, information about known issues, and links for obtaining further product support.

Documentation

This product guide is the main document associated with the 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 or by using the Xilinx® Documentation Navigator. Download the Xilinx Documentation Navigator from the Downloads page. For more information about this tool and the features available, open the online help after installation.

Answer Records

Answer Records include information about commonly encountered problems, helpful information on how to resolve these problems, and any known issues with a Xilinx product. Answer Records are created and maintained daily ensuring that users have access to the most accurate information available.

Answer Records for this core can be located by using the Search Support box on the main Xilinx support web page. To maximize your search results, use keywords such as:

• Product name

• Tool message(s)

• Summary of the issue encountered

A filter search is available after results are returned to further target the results.

Master Answer Record for the RAMA IP

AR 69267

Technical Support

Xilinx provides technical support in the Xilinx Support web page for this LogiCORE™ IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support if you do any of the following:

- Implement the solution in devices that are not defined in the documentation.
- Customize the solution beyond that allowed in the product documentation.
- Change any section of the design labeled DO NOT MODIFY.

To contact Xilinx Technical Support, navigate to the Xilinx Support web page.

Debug Tools

There are many tools available to address RAMA IP design issues. It is important to know which tools are useful for debugging various situations.

Vivado Design Suite Debug Feature

The Vivado® Design Suite debug feature inserts logic analyzer and virtual I/O cores directly into your design. The debug feature also allows you to set trigger conditions to capture application and integrated block port signals in hardware. Captured signals can then be analyzed. This feature in the Vivado IDE is used for logic debugging and validation of a design running in Xilinx® devices.

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)

See the *Vivado Design Suite User Guide: Programming and Debugging* (UG908).

Reference Boards

Various Xilinx® development boards support the RAMA IP core. These boards can be used to prototype designs and establish that the core can communicate with the system.

- Virtex® UltraScale+™ FPGA evaluation boards:
  - ◦ VCU1551
  - ◦ VCU128

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® debug feature is a valuable resource to use in hardware debug. The signal names mentioned in the following individual sections can be probed using the debug feature for debugging the specific problems.

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.
- If your outputs go to 0, check your licensing.

AXI Protocol Violations

When designing with *custom* or *non-production* IP, it is common to encounter system malfunctions caused by AXI protocol violations. Xilinx® AXI IP cores, including RAMA, do not contain any logic to guard against AXI protocol violations incurred by IP cores to which they are connected.

One of the most common symptoms of an AXI protocol violation in a system is an apparent lock-up of a connected core. When such a lock-up condition occurs, it often appears that an AXI channel transfer (valid/ready handshake) completes on one interface of the RAMA, but the resultant transfer is never issued on the expected output interface. Other possible symptoms include output transfers that appear to violate AXI transaction ordering rules.

RECOMMENDED: *Xilinx strongly recommends that you use the available AXI Protocol Checker IP core to test for AXI protocol compliance before deploying any custom IP or IP with custom modifications.*

Additional Resources and Legal Notices

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see Xilinx Support.

Documentation Navigator and Design Hubs

Xilinx® Documentation Navigator provides access to Xilinx documents, videos, and support resources, which you can filter and search to find information. To open the Xilinx Documentation Navigator (DocNav):

- From the Vivado® IDE, select **Help→Documentation and Tutorials**.
- On Windows, select **Start→All Programs→Xilinx Design Tools→DocNav**.
- At the Linux command prompt, enter `docnav`.

Xilinx Design Hubs provide links to documentation organized by design tasks and other topics, which you can use to learn key concepts and address frequently asked questions. To access the Design Hubs:

- In the Xilinx Documentation Navigator, click the **Design Hubs View** tab.
- On the Xilinx website, see the Design Hubs page.

Note: For more information on Documentation Navigator, see the Documentation Navigator page on the Xilinx website.

References

These documents provide supplemental material useful with this product guide:

1. *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994)
2. *Vivado Design Suite User Guide: Designing with IP* (UG896)
3. *Vivado Design Suite User Guide: Getting Started* (UG910)
4. *Vivado Design Suite User Guide: Logic Simulation* (UG900)
5. *ISE to Vivado Design Suite Migration Guide* (UG911)
6. *Vivado Design Suite User Guide: Programming and Debugging* (UG908)
7. *Vivado Design Suite User Guide: Implementation* (UG904)
8. *AXI Interconnect LogiCORE IP Product Guide* (PG059)
9. *AMBA AXI Protocol v2.0*
10. *SmartConnect LogiCORE IP Product Guide* (PG247)
11. *AXI High Bandwidth Memory Controller* (PG276)

Revision History

The following table shows the revision history for this document.

Section	Revision Summary
04/04/2018 v1.0	
Initial Xilinx release.	

Please Read: Important Legal Notices

related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at https://www.xilinx.com/legal.htm#tos; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at https://www.xilinx.com/legal.htm#tos.

AUTOMOTIVE APPLICATIONS DISCLAIMER

AUTOMOTIVE PRODUCTS (IDENTIFIED AS "XA" IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE ("SAFETY APPLICATION") UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD ("SAFETY DESIGN"). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.

Copyright

© Copyright 2018 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries.All other trademarks are the property of their respective owners.