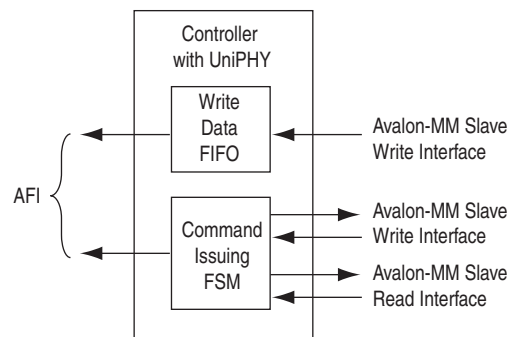


The RLDRAM II controller translates memory requests from the Avalon Memory-Mapped (Avalon-MM) interface to AFI, while satisfying timing requirements imposed by the memory configurations.

Block Description

This topic describes the blocks in the IP. Figure 6–1 shows a block diagram of the RLDRAM II controller architecture.

Figure 6–1. RLDRAM II Controller Architecture Block Diagram



Avalon-MM Slave Interface

This Avalon-MM slave interface accepts read and write requests. A simple state machine represents the state of the command and address registers, which stores the command and address when a request arrives.

The Avalon-MM slave interface decomposes the Avalon-MM address to the memory bank, column, and row addresses. The IP automatically maps the bank address to the LSB of the Avalon address vector.

The Avalon-MM slave interface includes a burst adaptor, which has the following two parts:

- The first part is a read and write request combiner that groups requests to sequential addresses into the native memory burst. Given that the second request arrives within the read and write latency window of the first request, the controller can combine and satisfy both requests with a single memory transaction.
- The second part is the burst divider in the front end of the Avalon-MM interface, which breaks long Avalon bursts into individual requests of sequential addresses, which then pass to the controller state machine.

© 2011 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.



Write Data FIFO Buffer

The write data FIFO buffer accepts write data from the Avalon-MM interface. The AFI controls the subsequent consumption of the FIFO buffer write data.

Command Issuing FSM

The command issuing finite-state machine (FSM) has three states. The controller is in the `INIT` state when the PHY initializes the memory. Upon receiving the `afi_cal_success` signal, the state transitions to `INIT_COMPLETE`. If the calibration fails, `afi_cal_fail` is asserted and the state transitions to `INIT_FAIL`. The PHY receives commands only in the `INIT_COMPLETE` state.

When a refresh request arrives at the state machine at the same time as a read or write request, the refresh request takes precedence. The read or write request waits until there are no more refresh requests, and is issued immediately if timing requirements are met.

Refresh Timer

With automatic refresh, the refresh timer periodically issues refresh requests to the command issuing FSM. The refresh interval can be set at generation.

Timer Module

The timer module contains one DQ timer and eight bank timers (one per bank). The DQ timer tracks how often read and write requests can be issued, to avoid bus contention. The bank timers track the cycle time (t_{RC}).

The 8-bit wide output bus of the bank timer indicates to the command issuing FSM whether each bank can be issued a read, write, or refresh command.

AFI

For information on the AFI, refer to [AFI 3.0 Specification](#).

User-Controlled Features

The following features are available on the **General Settings** tab of the parameter editor. These features are disabled by default.

Error Detection Parity

The error detection parity protection feature creates a simple parity encoder block which processes all read and write data. The error detection feature asserts an error signal if it detects any corrupted data during the read process. For every 8 bits of write data, a parity bit is generated and concatenated to the data before it is written to the memory. During the subsequent read operation, the parity bit is checked against the data bits to ensure data integrity.

Enabling the error detection parity protection feature reduces the local data width by one. For example, a nine-bit memory interface will present eight bits of data to the controller interface.

You can enable error detection parity protection in the **Controller Settings** section of the **General Settings** tab of the parameter editor.

User-Controlled Refresh

The user-controlled refresh feature allows you to take control of the refresh process that the controller normally performs automatically. You can control when refresh requests occur, and, if there are multiple memory devices, you control which bank receives the refresh signal. When you enable this feature, you disable auto-refresh, and assume responsibility for maintaining the necessary average periodic refresh rate.

You can enable user-controlled refresh in the **Controller Settings** section of the **General Settings** tab of the parameter editor.

Avalon-MM and Memory Data Width

Table 6–1 shows the data width ratio between the memory interface and the Avalon-MM interface. The half-rate controller does not support burst-of-2 devices because it under-uses the available memory bandwidth.

Table 6–1. Data Width Ratio

Memory Burst Length	Half-Rate Designs	Full-Rate Designs
2-word	No Support	2:1
4-word	4:1	
8-word		

Signal Description

This topic discusses the signals for each interface.

For information on the AFI signals, refer to [AFI 3.0 Specification](#).

Avalon-MM Slave Interface


Table 6–2 shows the list of signals of the controller’s Avalon-MM slave interface.

Table 6–2. Avalon-MM Slave Signals

Signal	Width	Direction	Avalon-MM Signal Type	Description
avl_size	1 to 11	In	burstcount	—
avl_ready	1	Out	waitrequest_n	—
avl_read_req	1	In	read	—
avl_write_req	1	In	write	—
avl_addr	≤5	In	address	—
avl_rdata_valid	1	Out	readdatavalid	—

Table 6-2. Avalon-MM Slave Signals

Signal	Width	Direction	Avalon-MM Signal Type	Description
avl_rdata	18, 36, 72, 144	Out	readdata	—
avl_wdata	18, 36, 72, 144	In	writedata	—

 The data width of the Avalon-MM interface is restricted to powers of two when using SOPC Builder or Qsys. Non-power-of-two data widths are supported when using the MegaWizard Plug-In Manager.

Document Revision History

Table 6-3 lists the revision history for this document.

Table 6-3. Document Revision History

Date	Version	Changes
November 2011	3.1	Harvested Controller chapter from 11.0 RLDRAM II Controller with UniPHY IP User Guide .