AXI SDRAM V1.0

IP User Guide(Beta Release)

September 13, 2024

RapidSilicon

Contents

IP Summary

Introduction

An AXI SDRAM controller acts as a bridge between the AXI bus, and a simpler SDRAM (Synchronous Dynamic Random Access Memory) for off-chip data storage. This controller translates requests from the processor into the appropriate signals for the SDRAM. This core can be configured via Raptor's IP Catalog GUI interface.

Features

- 32-bit AXI4 slave interface
- Standard SDRAM interface

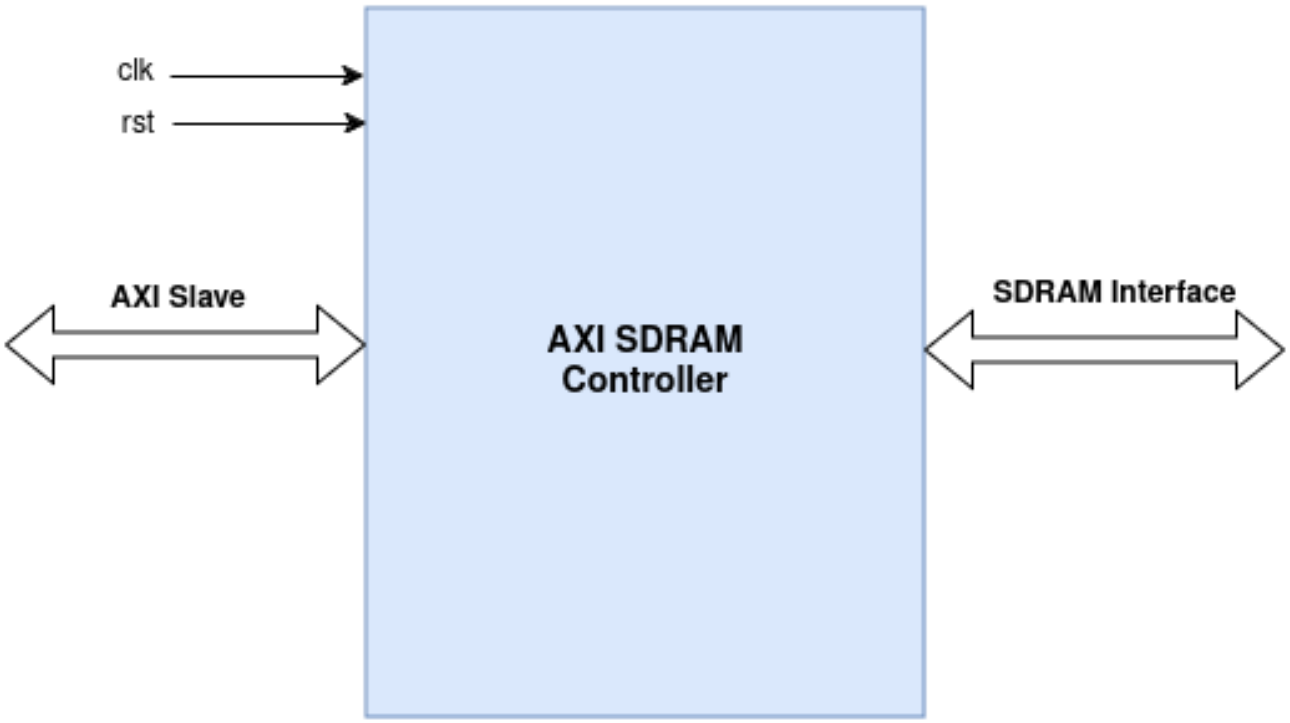


Figure 1. AXI_SDRAM block level diagram

Licensing

COPYRIGHT TEXT:

IP Specification

Overview

An AXI SDRAM controller acts as a crucial bridge within a digital system, facilitating communication between the high-performance AXI bus and the external Synchronous Dynamic Random Access Memory (SDRAM). This controller deciphers the processor's AXI read/write requests and translates them into the specific commands understood by the SDRAM.

Here's a breakdown of its key components:

AXI Interface: This acts as the entry point, receiving AXI requests and data packets from the processor. It understands the AXI protocol, which defines how data is formatted and transferred on the internal bus.

Controll Unit: This unit serves as the brain of the controller. It receives the AXI requests from the interface, decodes them, and generates the corresponding commands for the SDRAM. Additionally, it incorporate a scheduler to optimize the order of requests, maximizing memory access efficiency. This scheduler ensures that data packets are issued to the SDRAM in a way that minimizes wait times and leverages the memory's capabilities.

SDRAM Interface: This section acts as the translator on the other side. It takes the generated commands from the command unit and transforms them into the control signals and addresses that the SDRAM understands. It also handles the complex timing sequences specific to SDRAM technology, ensuring data integrity during transfers. SDRAM has specific requirements for when data can be read or written, and this interface ensures these timings are met meticulously.

Refresh Logic: DRAM is volatile, meaning it loses data when power is off. To maintain data integrity, the controller incorporates refresh logic. This component periodically activates refresh cycles, essentially reminding the SDRAM to recharge its storage capacitors and prevent data loss.

These components work coherently to manage the data flow between the processor and the SDRAM. The AXI interface receives requests, the command unit translates them, the SDRAM interface generates the appropriate signals, and the refresh logic ensures data remains valid. This intricate interplay allows the system to leverage the high-speed processing power of the processor while utilizing the vast storage capacity of the SDRAM.

IP Support Details

Compliance		IP Resources					Tool Flow		
Device	Interface	Source Files	Constraint File	Testbench	Simulation Model	Software Driver	Analyze and Elaboration	Simulation	Synthesis
GEMINI	AXI4	Systemverilog	SDC	Systemverilog	-	-	Raptor	Raptor	Raptor

Ports

Table 2 lists the top interface ports of the AXI SDRAM.

Signal Name	I/O	Description
AXI Clock and Reset		
i_s_axi_clk	I	AXI4 Clock
i_s_axi_resetn	I	AXI4 RESET
i_m_axi_clk	I	AXI4 Clock
AXI WRITE ADDRESS CHANNEL		
s_axil_awvalid	I	AXI4 Write address valid
s_axil_awready	O	AXI4 Write address ready
s_axil_awaddr	I	AXI4 Write address
s_axil_awprot	I	AXI4 Protection type
AXI WRITE DATA CHANNEL		
s_axil_wvalid	I	AXI4 Write valid
s_axil_wready	O	AXI4 Write ready.
s_axil_wdata	I	AXI4 Write data
s_axil_wstrb	I	AXI4 Write strobes
AXI WRITE RESPONSE CHANNEL		
s_axil_bvalid	O	AXI4 Write response valid
s_axil_bready	I	AXI4 Response ready
s_axil_bresp	O	AXI4 Write response
AXI READ ADDRESS CHANNEL		
s_axil_arvalid	I	AXI4 Read address valid
s_axil_arready	O	AXI4 Read address ready
s_axil_araddr	I	AXI4 Read address
s_axil_arprot	I	AXI4 Protection type
AXI READ DATA CHANNEL		
s_axil_rvalid	I	AXI4 Read valid
s_axil_rready	O	AXI4 Read ready
s_axil_rresp	I	AXI4 Read data
s_axil_rdata	O	AXI4 Read response
DDR Interface		
sdram_clk_o	I	SDRAM clock
sdram_cke_o	I	SDRAM Clock enable
sdram_cs_o	I	SDRAM chip select
sdram_ras_o	I	SDRAM Row address strobe
sdram_cas_o	O	SDRAM Column address strobe
sdram_we_o	I	SDRAM wriite enable
sdram_ba_o	I	SDRAM bank address

sdram_addr_o	O	SDRAM Address
sdram_dqm_o	I	SDRAM data mask
sdram_data_out_en_o	O	SDRAM data enable
sdram_data_output_o	I	SDRAM data
sdram_data_input_i	I	SRAM Input Data

AXI SDRAM Interface

Resource Utilization

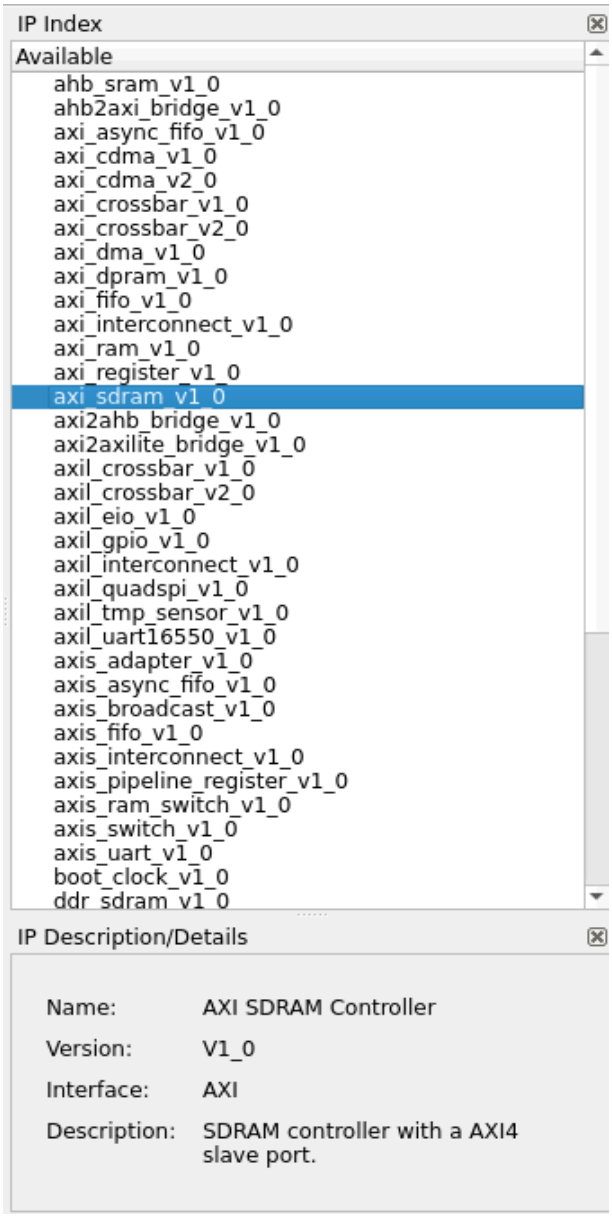
Please note that the utilization and timing figures provided in this section for the Processor System Reset IP core should be considered as estimates, as they are based on its usage in conjunction with other design modules in the FPGA. Once integrated with other designs in the system, the FPGA resource utilization and core timing may differ from the reported results.

Tool	Raptor Design Suite		
FPGA Device	GEMINI		
		Resource Utilization	
Minimum Resource	**Options**	**Resources**	
	LUT	514	
	Registers	397	
	BRAM	0	
	DSP	0	

Design Flow

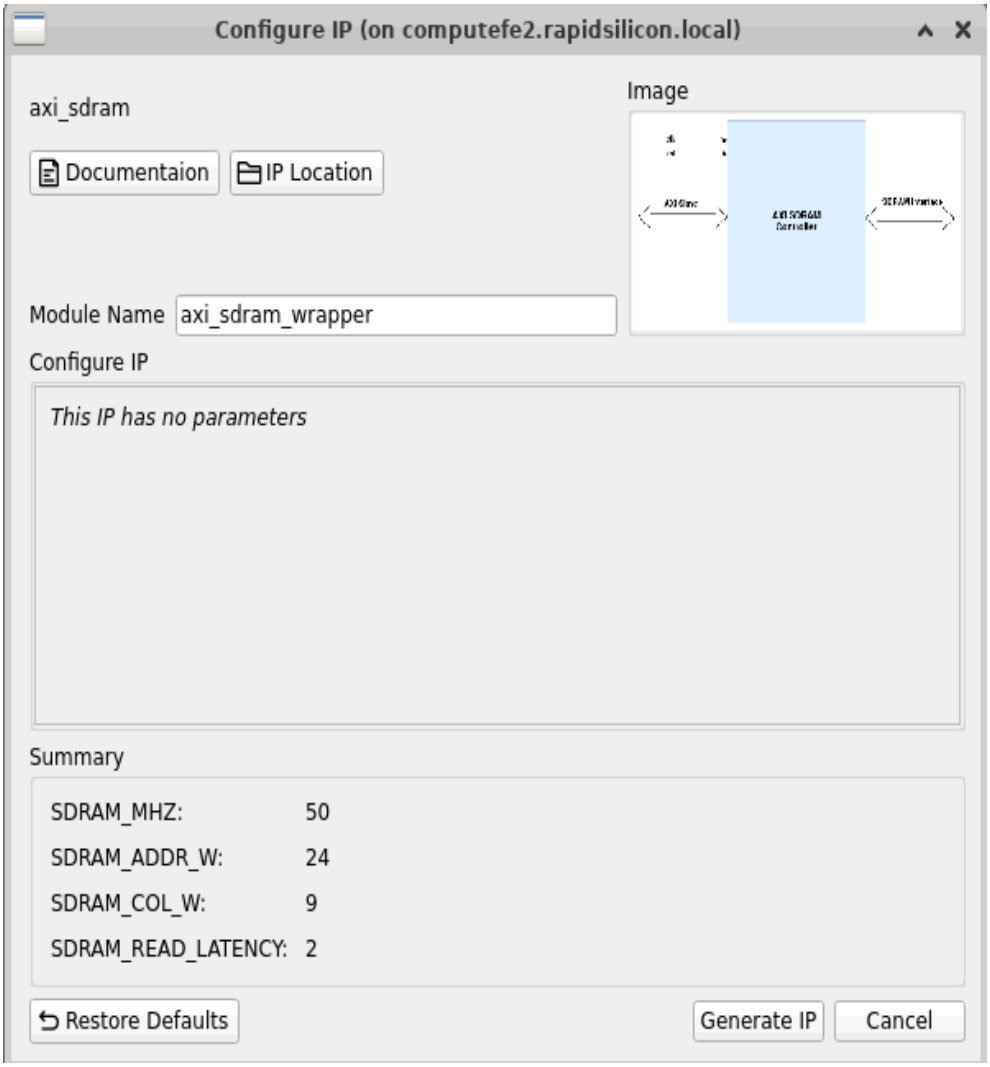
IP Customization and Generation

AXI DDR SDRAM IP core is a part of the Raptor Design Suite Software. A customized AXI DDR SDRAM can be generated from the Raptor's IP configurator window.



Selecting AXI DDR SDRAM from IP Catalog List

Parameters Customization: From the IP configuration window, the parameters of the AXI DDR SDRAM can be configured and AXI DDR SDRAM features can be enabled for generating a customized AXI DDR SDRAM IP core that suits the user application requirement.

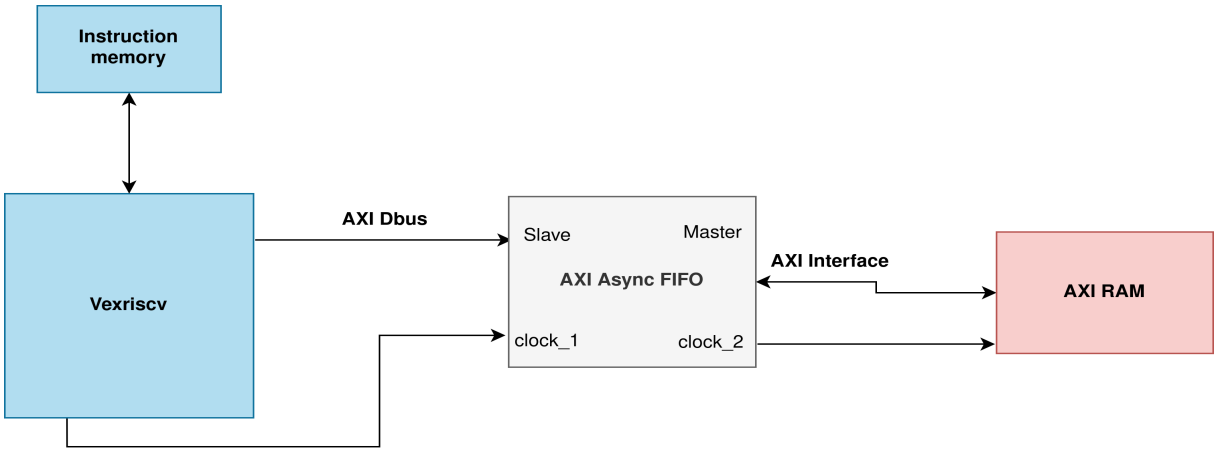


IP Configuration

Example Design

Simulating the Example Design

AXI SDRAM can be simulated using Raptor. The simulation collateral is available in the sim directory.



AXI Async FIFO example design simulation

Synthesis and PnR

Raptor Suite is armed with tools for Synthesis along with Post and Route capabilities and the generated post- synthesis and post-route and place netlists can be viewed and analyzed from within the Raptor. The generated bitstream can then be uploaded on an FPGA device to be utilized in hardware application

Revision History

Date	Version	Revisions
September 13, 2024	0.01	Initial version AXI/$_S$*DRAMUserGuideDocument*