AHB_SRAM
(Beta Release)

Version 1.0

August 4, 2024

Copyright

Trademarks

All Rapid Silicon trademarks are as listed at www.rapidsilicon.com. Synopsys and Synplify Pro are trademarks of Synopsys, Inc. Aldec and Active-HDL are trademarks of Aldec, Inc. Modelsim and Questa are trademarks or registered trademarks of Siemens Industry Software Inc. or its subsidiaries in the United States or other countries. All other trademarks are the property of their respective owners.

Disclaimers

NO WARRANTIES: THE INFORMATION PROVIDED IN THIS DOCUMENT IS “AS IS” WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF ACCURACY, COMPLETENESS, MERCHANTABILITY, NONINFRINGEMENT OF INTELLECTUAL PROPERTY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL RAPID SILICON OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (WHETHER DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL, INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE INFORMATION PROVIDED IN THIS DOCUMENT, EVEN IF RAPID SILICON HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISDICTIONS PROHIBIT THE EXCLUSION OR LIMITATION OF CERTAIN LIABILITY, SOME OF THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU.

Rapid Silicon may make changes to these materials, specifications, or information, or to the products described herein, at any time without notice. Rapid Silicon makes no commitment to update this documentation. Rapid Silicon reserves the right to discontinue any product or service without notice and assumes no obligation to correct any errors contained herein or to advise any user of this document of any correction if such be made. Rapid Silicon recommends its customers obtain the latest version of the relevant information to establish that the information being relied upon is current and before ordering any products.

Contents

IP Summary

Introduction

The AHB SRAM controller IP acts as a bridge between the Advanced High-performance Bus (AHB) and external SRAM memory. It translates AHB bus transactions into the appropriate control signals and data transfers required by the SRAM device. This IP core facilitates communication between various AHB bus masters and the external SRAM memory, enabling data storage and retrieval operations. This core can be configured via Raptor's IP Catalog GUI interface.

Features

- AHB interface is based on the AHB5 specification
- AHB slave 32 bit interface with Burst support

Overview

AHB_SRAM

In AHB_SRAM bridge the AHB Interface is slave interface on AHB bus side. It accepts the control signals when a transfer is initiated by AHB master and generates ahb_hreadyout based on the transfer progress on SRAM side. The Control logic is the main controlling unit which generates the respective signals depending upon the transfer type and progress of that transfer. It detects the properties of a transfer on AHB side (Read/Write, Burst, Single, transfer type) and upon this information generate the SRAM signals appropriately.

IP Specification

The working of an AHB_SRAM bridge involves several key operations and considerations:

Bus Protocol Translation: One of the primary functions of the bridge is to convert AHB transactions into AXI transactions and vice versa. This translation ensures that data and control signals are correctly interpreted and processed by devices on both sides of the bridge.

Address Mapping: The AXI memory map and the AHB memory map are one single complete 32-bit (4 GB) memory space. The AHB to AXI Bridge core does not modify the address for AXI; hence, the address that is presented on the AXI is exactly as received on the AHB interface.

Control Signals: The bridge also translates control signals between the AHB and AXI buses. This includes signals like read and write enables, burst types, and response codes. Proper translation and synchronization of control signals are essential for maintaining the integrity of data transfers.

Error Handling: Robust error handling mechanisms are essential in any bridge design. The AHB_SRAM bridge detects and manage errors, such as bus contention, address mapping faults, or data transfer errors, to ensure system reliability.

Configuration Options: Many AHB_SRAM bridges offer configuration options to adapt to various system requirements. Designers can often configure parameters like address mapping, data width, and burst sizes to match the specific needs of their system.

The AHB_SRAM module has two main parts: the write path and the read path. The write path takes data from the AXI4 slave interface and stores it in a write data FIFO. The read path takes data from a read data FIFO and provides it to the AXI4 master interface.

In addition, the AHB_SRAM is equipped with the flexibility to support various burst types and parametrizable data and address interface widths. Moreover, it offers the option to delay the address channel until either the write data is entirely shifted into the FIFO or the read data FIFO can accommodate the entire burst.

Standards

The AXI4-Full interface is compliant with the AMBA® AXI Protocol Specification and AHB interface is compliant with AMBA® AHB5 Protocol SPecification.

IP Support Details

The Table 1 gives the support details for AHB_SRAM.

Compliance		IP Resources					Tool Flow	
Device	Interface	Source Files	Constraint File	Testbench	Simulation Model	Analyze and Elaboration	Simulation	Synthesis
GEMINI	AHB & System Verilog	SDC	Python	System Verilog	Raptor	Raptor Icarus	Raptor	

Table 1: IP Details

Parameters

Table lists the parameters of the AHB_SRAM.

Parameter	Values	Default Value	Description
SRAM_DATA_WIDTH	1-8	8	SRAM Data width
SRAM_ADDR_WIDTH	2-13	13	SRAM address width.
DATA_WIDTH	32,64	32	AHB bus Data width.
ADDR_WIDTH	1-64	32	AHB bus address width.
HBURST_WIDTH	0-16	3	AHB HBURST WIDTH.
HTRANS_WIDTH	1-32	2	AHB HTRANS WIDTH.
HSIZE_WIDTH	1-32	3	AHB HSIZE WIDTH.
HRESP_WIDTH	1-32	2	AHB HRESP WIDTH.

AHB_SRAM Parameters

Port List

Table 2 lists the top interface ports of the AHB_SRAM.

Signal Name	I/O	Description
AHB Clock and Reset		
clk	I	Bridge clock Clock
rst_l	I	Bridge clock RESET
SLAVE INTERFACE		
haddr	I	AHB Write address valid
hburst	I	AHB Write address ready
hmastlock	I	AHB Write address
hprot	I	AHB Protection type
hsize	I	AHB Write valid
htrans	I	AHB Write ready.
hwrite	I	AHB Write data
hwdata	I	AHB Write strobes
hsel	I	AHB Write response valid
hready	I	AHB Response ready
hnonsec	I	AHB Write response

hrdata	O	AHB Read address valid
hready_resp	O	AHB Read address ready
hresp	O	AHB Read address
sram_clk	O	SRAM Clock

Table 2: AHB_SRAM Interface

Resource Utilization

The parameters for computing the maximum and minimum resource utilization are given in Table 3, remaining parameters have been kept at their default values.

Tool	Raptor Design Suite			
FPGA Device	GEMINI			
Configuration			**Resource Utilization**	
Minimum Resource	**Options**	**Configuration**	**Resource**	**Utilized**
	DATA WIDTH	32	BRAMs	1
	ADDR WIDTH	16	LUTs	209
	WRITE FIFO DEPTH	32	Registers	293
Maximum Resource	**Options**	**Configuration**	**Resource**	**Utilized**
	DATA WIDTH	1024	BRAMs	62
	ADDR WIDTH	32	LUTs	255
	WRITE FIFO DEPTH	512	Registers	2426

Table 3: Resource Utilization

Design Flow

IP Customization and Generation

AHB_SRAM IP core is a part of the Raptor Design Suite Software. A customized AHB_SRAM can be generated from the Raptor's IP configurator window as shown in Figure 1.

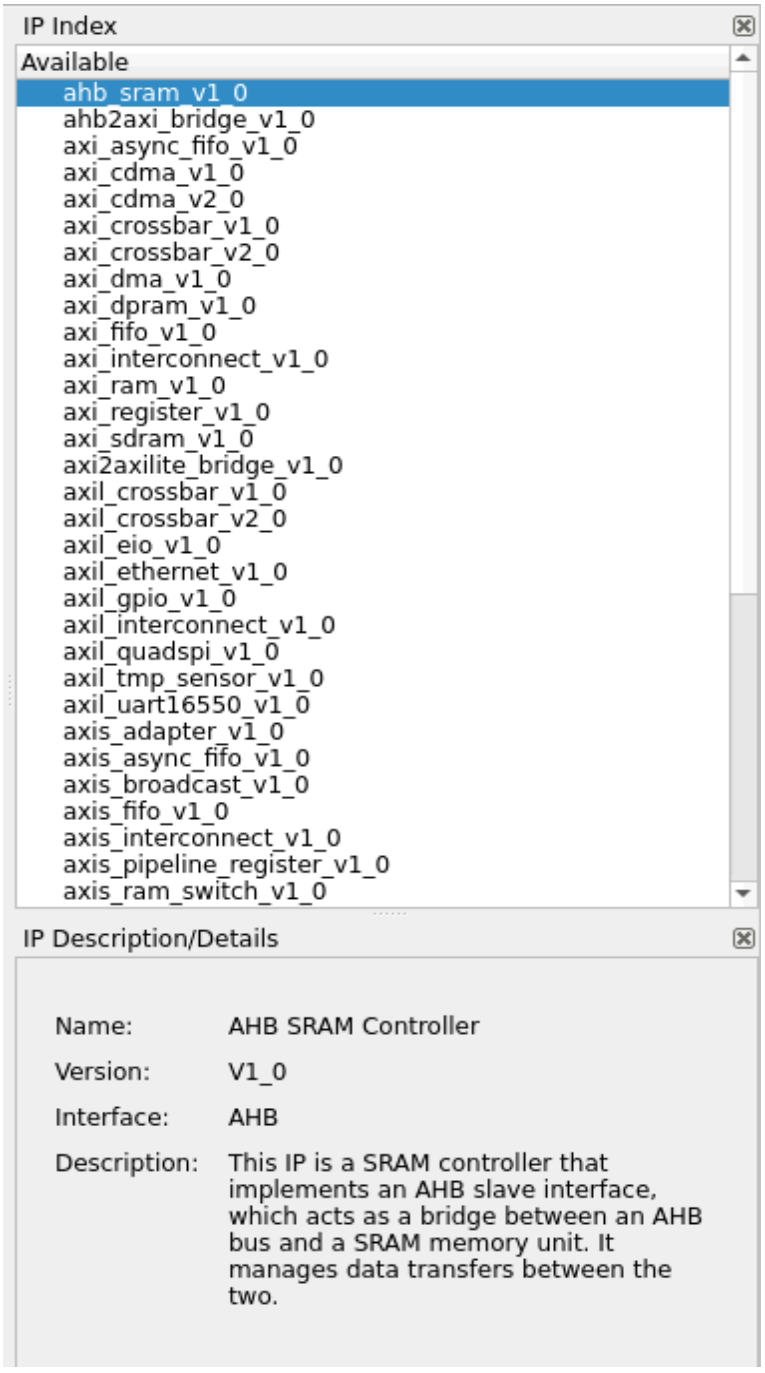

Figure 1: IP list

Parameters Customization

From the IP configuration window, the parameters of the AHB_SRAM can be configured and AHB_SRAM features can be enabled for generating a customized AHB_SRAM IP core that suits the user application requirement as shown in Figure 2. After IP Customization, all the source files are made available to the user with a top wrapper that instantiates a parameterized instance of the AHB_SRAM.

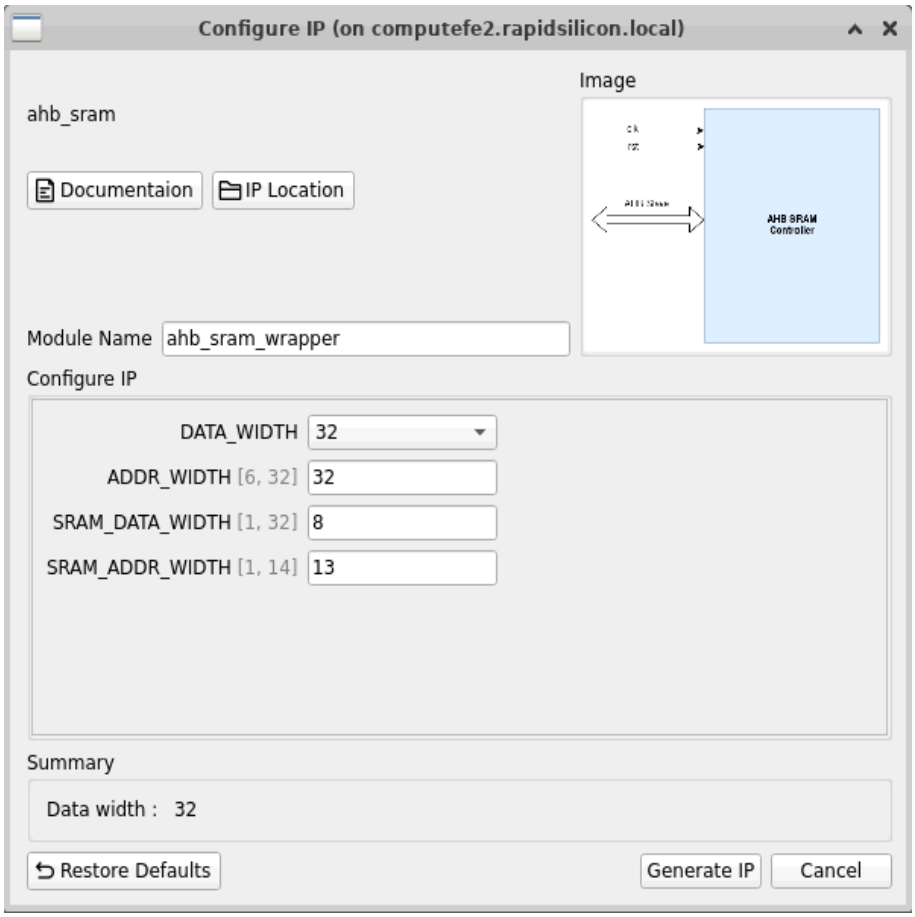

Figure 2: IP Configuration

Synthesis and PR

Raptor Suite is armed with tools for Synthesis along with Place and Route capabilities and the generated post-synthesis and post-route and place net-lists can be viewed and analyzed from within the Raptor. The generated bit-stream can then be uploaded on an FPGA device to be utilized in hardware applications.

Test Bench

The IP package hosts a simple Verilog based test bench that validates design functionality. It can be simulated using Iverilog, Verilator or other simulators. The simulation can be easily run by clicking the "Simulate IP" button as shown in figure 4. The waveforms are also dumped for in-depth analysis of the whole operation which can be seen by clicking the "View Waveform" button.

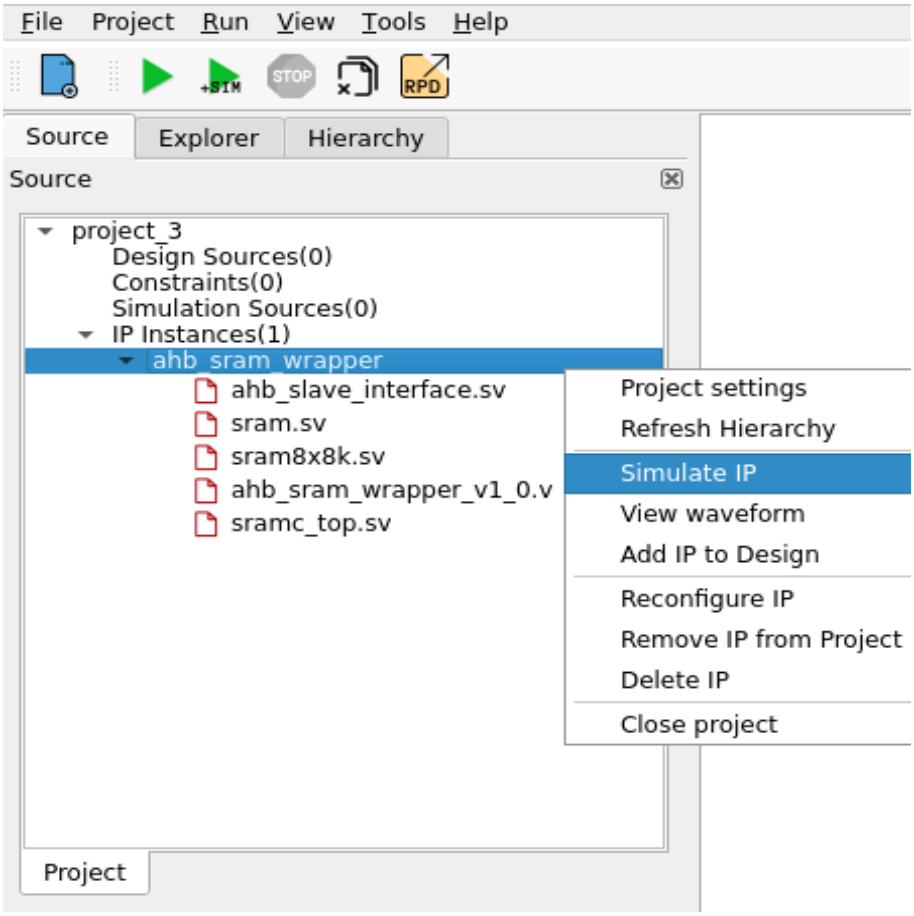

Figure 3: Simulate IP Window

The testbench is a basic system verilog testbench, that performs some write/read tests to verify the SRAM. Below shows a glimpse of the wavedump of the simulation.

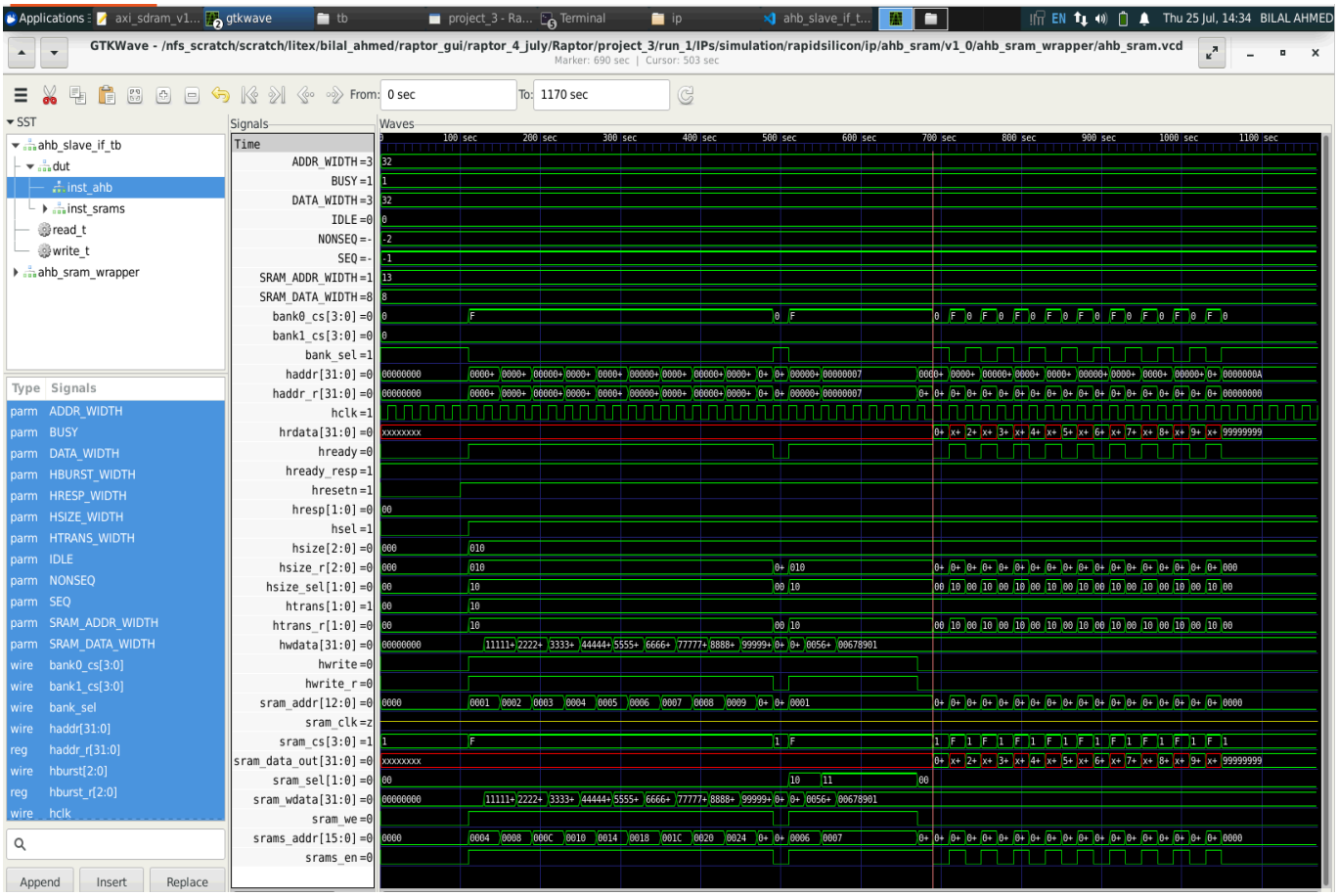

Figure 4: Wavedump of simulation

Release

Release History

Date	Version	Revisions
August 4, 2024	1.0	Initial version AHB_SRAM Bridge User Guide Document