**APB Protocol with Dual port memory interface**

**Design & Verification Project Demo**

Client: XXXXX

Date: 2nd MAY 2024

This document is only for educational purpose. this document is **NOT CONFIDENTIAL** and use of SILICONOVA & XXXX **ONLY**.

TABLE OF CONTENT

[Overview 3](#_Toc165542623)

[Version 3](#_Toc165542624)

[GENERAL OVERVIEW 4](#_Toc165542625)

[FEATURE 4](#_Toc165542626)

[LIMITATIONS 4](#_Toc165542627)

[TOP LEVEL DIAGRAM 5](#_Toc165542628)

[FSM DIAGRAM: 6](#_Toc165542629)

[GENERAL DESCRIPTION 7](#_Toc165542630)

# 

# Overview

In this project, APB (Advanced peripheral Bus) protocol is designed with the dual port ram interface.

# Version

APB Slave with RAM Version 1.0

# GENERAL OVERVIEW

In this project APB (Advanced peripheral Bus) slave is designed with a dual port ram instantiation. So, the basic operation is writing and reading the data in the Dual Port Memory through APB slave interface.

# FEATURE

**1. Input Data Handling:**

* Accepts 8, 16, 24, or 32-bit input data via the PWRITE\_i port.
* Data is stored in the dual-port memory at a designated address provided through the PADDR\_i port.

**2. Byte-Addressable Memory:**

* When using the APB slave interface for writing, memory operates on a byte-addressable basis meaning individual byte writes are facilitated, controlled by the PSTRB\_i port.

**3. Data Retrieval:**

* Reading data from the dual-port memory retrieves 32 bits (4 bytes) at a time through PRDATA\_o port.

**4. Memory Depth:**

* The dual-port memory has a depth of 32.

# LIMITATIONS

In case of dual port memory design, the arbitration scheme has been skipped.

# TOP LEVEL DIAGRAM

APB SLAVE

DUAL PORT RAM

PCLK\_i

PRESETn\_i

PADDR\_i

PWRITE\_i

PWDATA\_i

PSTRB\_i

PSEL\_i

PENABLE\_i

PRDATA\_o

PREADY\_o

PSLVERR\_o

write\_a

addr\_a

byte\_sel

datain\_a

read\_b

addr\_b

dataout\_b

**TOP**

# FSM DIAGRAM:

PSEL \_i = 0

INITIAL

PSLVERR\_o = 0

PREADY\_o = 1’b1

PRDATA\_o = dataout\_b

PSEL \_i = 1

PSEL \_i = 0

PSEL \_i = 1

PRDATA\_o = dataout\_b

PREADY\_o = 1’b1

TRANSFER

PSLVERR\_o = PADDR\_i > DEPTH

The APB slave operates in two states: the initial state and the transfer state. When PSEL\_i is low, it remains in the initial state. However, when PSEL\_i is high, the system transitions to the transfer state. It remains in the transfer state until PSEL\_i returns low.

# GENERAL DESCRIPTION

In this system the APB slave interface is designed with a dual port ram instantiation. that means dual port ram is implemented in the APB slave interface. This configuration allows for seamless read and write operations through distinct ports.

During a write operation, data is transmitted through the designated `PWDATA\_i` port, depending on a high signal on the ‘PWRITE\_i’ line. Conversely, when the ‘PWRITE\_i’ signal is low, the system initiates a read operation, retrieving data from the RAM. The retrieved data is made available for access via the `PRDATA\_o` port. This dual-port RAM architecture optimizes data transfer processes, facilitating both input and output functionalities within the system. For any operation to take place, it's essential that the appropriate control signals (PSEL\_i) is activated. Additionally, the address for the operation must be provided via the `PADDR\_i` port.

A key feature of this system is error handling. If an invalid address is provided—meaning it exceeds the memory depth—the system raises an error signal (`PSLVERR\_o`).

Moreover, to ensure smooth operation, the slave signals its readiness to perform operations through the `PREADY\_o` signal. This allows the master device to coordinate communication effectively.