通用 BRAM 控制器

2024.09.01

一 修订

版本	日期	编辑人	内容
1.00	2024.09.01	陈家耀	创建了第一个正式版本

二 简介和特性

通用 BRAM 控制器包含了 **AHB-BRAM 控制器**和 **AXI-BRAM 控制器**两种，请根据设计要求选择 AHB 或 AXI 接口。

AHB-BRAM 控制器结构简单、资源消耗较少，它直接将 AHB 传输转换成 BRAM 读写操作，因此无论对随机读写还是批量连续读写都具有良好的性能。**AXI-BRAM 控制器**则相对复杂，它需要在读写地址通道上进行仲裁，将突发传输转换为批量的 BRAM 读写操作。由于每启动一次突发传输都会消耗若干 cycle 来仲裁和锁存地址/控制信息，AXI-BRAM 控制器在稀疏随机读写方面性能较差，但它在批量连续读写方面性能仍较好。

AHB-BRAM 控制器具有如下特性：

- **支持 BRAM 读延迟为 1 或 2clk**
- **符合 AHB-FULL 协议，支持非对齐传输和窄带传输**
- **低传输延迟，写延迟为 1clk，读延迟为 2 或 3clk**

AXI-BRAM 控制器具有如下特性：

- **支持 BRAM 读延迟为 1 或 2clk**
- **符合 AXI-FULL 协议，但不支持非对齐传输和窄带传输**
- **可选读缓冲 fifo 以提高读传输的效率**

三 IP 功能

AHB-BRAM 控制器直接将 AHB 传输转换成 BRAM 读写操作，它支持 BRAM 读延迟为 1 或 2clk。当 **BRAM 读延迟为 1clk** 时，AHB-BRAM 控制器的读写延迟均为 1clk。当 **BRAM 读延迟为 2clk** 时，其写延迟为 1clk，读延迟一般为 2clk，仅当连续的写到读传输时才出现 3clk 的读延迟。

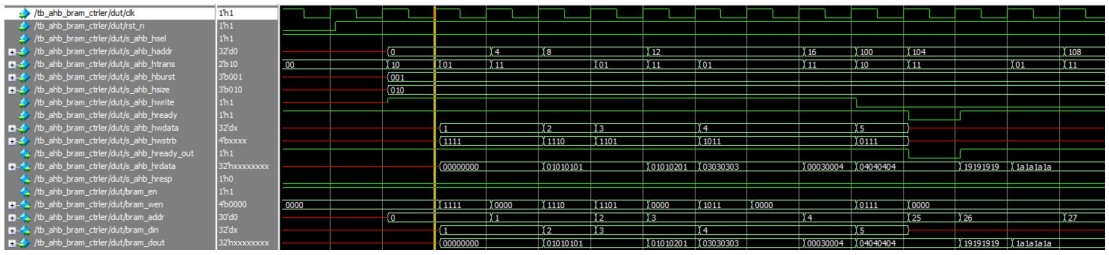

图 3-1 AHB-BRAM 控制器仿真波形（BRAM 读延迟为 1clk）

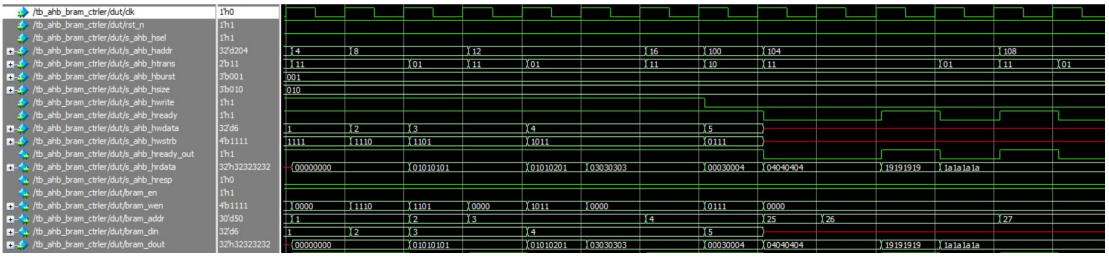

图 3-2 AHB-BRAM 控制器仿真波形（BRAM 读延迟为 2clk）

AXI-BRAM 控制器使用一个状态机控制读写地址通道的仲裁、读写突发的进行和写响应的发送，可启用读缓冲 fifo 来提高读效率。该控制器要求 AXI 接口必须先给出读/写地址(AR/AW)，再给出读/写数据(R/W)。由于每开启一次读写突发都需要若干 cycle，因此它可能在随机地址读写方面性能不佳。

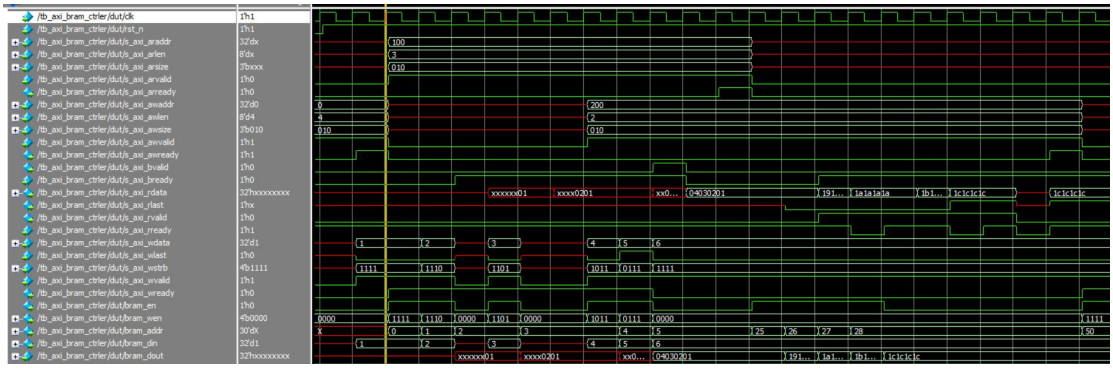

图 3-3 AXI-BRAM 控制器仿真波形（BRAM 读延迟为 1clk，启用读缓冲 fifo）

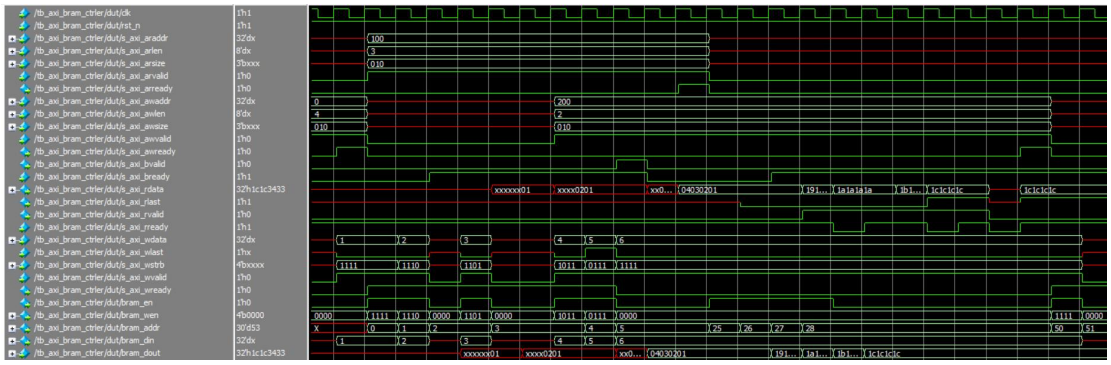

图 3-4 AXI-BRAM 控制器仿真波形（BRAM 读延迟为 2clk，启用读缓冲 fifo）

四 IO 描述

表 4-1 通用 BRAM 控制器 IO 表

端口名	方向	位宽	含义
时钟和复位			
clk	input	1	时钟
rst_n	input	1	复位，低有效
AHB 或 AXI 从接口			
……			
BRAM 主接口			
bram_clk	output	1	BRAM 时钟
bram_rst	output	1	BRAM 复位，高有效
bram_en	output	1	BRAM 使能
bram_wen	output	4	BRAM 字节写使能
bram_addr	output	30	BRAM 读写地址
bram_din	output	32	BRAM 写数据
bram_dout	input	32	BRAM 读数据
错误指示（仅 AXI-BRAM 控制器包含）			
axi_bram_ctrler_err	output	2	位 1：不支持的非对齐传输 位 0：不支持的窄带传输

五 可配置参数描述

表 5-1 AHB-BRAM 控制器 可配置参数表

配置参数名	含义	可取值
bram_read_la	BRAM 读延迟（以 clk 计）	1 | 2
simulation_delay	仿真延时，可用于仿真时模拟 D 到 Q 延迟	0.1f~100.0f

表 5-2 AXI-BRAM 控制器 可配置参数表

配置参数名	含义	可取值
bram_depth	BRAM 深度	32 | 64 | 128 | 256 | ...
bram_read_la	BRAM 读延迟（以 clk 计）	1 | 2
en_read_buf_fifo	是否使用读缓冲 fifo	"true" | "false"
simulation_delay	仿真延时，可用于仿真时模拟 D 到 Q 延迟	0.1f~100.0f

六 应用指南

通用 BRAM 控制器可以直接挂载在系统总线上进行使用，下面的 C 程序展示了如何通过这个控制器读写 BRAM：

```
/*****************************************************************************************
************************************
BRAM 控制器示例代码
@brief  读写 BRAM 示例
@date   2024/09/01
@author 陈家耀
@eidt   2024/09/01 1.00 创建了第一个正式版本
*****************************************************************************************
************************************/

#include <stdint.h>

//////////////////////////////////////////////////////////////////////////////////////////////////

typedef uint32_t BRAM_handler;

//////////////////////////////////////////////////////////////////////////////////////////////////

#define BRAM_CTRLER_BASEADDR 0x60000000 // BRAM 控制器基地址

//////////////////////////////////////////////////////////////////////////////////////////////////

static BRAM_handler* bram; // BRAM 控制器句柄

//////////////////////////////////////////////////////////////////////////////////////////////////

void bram_read_write_example(void){
    bram = (BRAM_handler*)BRAM_CTRLER_BASEADDR; // 初始化 BRAM 控制器句柄

    // 写 BRAM
    for(uint32_t i = 0;i < 100;i++){
        bram[i] = i + 1;
    }

    // 读 BRAM
    uint8_t verify_ok = 1;

    for(uint32_t i = 0;i < 100;i++){
        if(bram[i] != (i + 1)){
```

```
            verify_ok = 0;

            break;
        }
    }

    // 验证读写数据一致性
    if(verify_ok){
        // 读写一致
        // ...
    }else{
        // 读写不一致
        // ...
    }

    while(1);
}
```