

具有硬件加速的SPI控制器IP核的设计与研究

蒋 宏, 张传武, 李宇欣

西南民族大学, 四川 成都

收稿日期: 2022年10月11日; 录用日期: 2022年11月11日; 发布日期: 2022年11月21日

摘 要

随着大规模集成电路的发展, 基于IP复用的SoC (System-on-Chip)技术已成为芯片设计的主要趋势, IP核的复用也减轻了集成电路芯片的负担。SPI (Serial Peripheral Interface, 串行外设接口)作为一个比较经典的IP核, 其相关研究和运用受到国内外的广泛关注, 因其通讯接口技术简单、传输速度快, 被广泛集成在微处理器的通信电路接口模块中。在SoC系统中, 为了进一步提高SPI通讯的速度和效能, 设计加入DMA (Direct Memory Access)控制器就显得意义重大。因此, 在这个设计中我们将把SPI与DMA组合在一起使用, 以实现提升数据交换的效率。本文首先介绍了相关的设计原理, 给出基于AMBA总线协议和AXI总线协议设计的系统框图, 描述验证环境, 并通过Windows SP7021 IDE集成开发环境进行软件验证, 来进行SPI接口IP核的设计和研究, 通过时序仿真测试, SPI接口能正确地对数据进行传输, 满足了SPI时序设计要求, 并且能满足实际工程应用。

关键词

SoC, IP核, DMA, SPI

Design and Research of SPI Controller IP Core with Hardware Acceleration

Hong Jiang, Chuanwu Zhang, Yuxin Li

Southwest Minzu University, Chengdu Sichuan

Received: Oct. 11th, 2022; accepted: Nov. 11th, 2022; published: Nov. 21st, 2022

Abstract

With the development of large-scale integrated circuits, SoC (System-on-Chip) technology based on IP reuse has become the main trend of chip design, and the reuse of IP cores has also reduced the burden of integrated circuit chips. As a classic IP core, SPI (serial peripheral interface) is widely

concerned at home and abroad for its research and application. Because of its simple communication interface technology and fast transmission speed, it is widely integrated into the communication circuit interface module of microprocessor. In SoC system, in order to further improve the speed and efficiency of SPI communication, it is of great significance to design and add DMA (Direct Memory Access) controller. Therefore, in this design, we will combine SPI and DMA to improve the efficiency of data exchange. This paper first introduces the relevant design principles, gives the system block diagram based on the AMBA bus protocol and AXI bus protocol design, describes the verification environment, and carries out software verification through the Windows SP7021 IDE integrated development environment to carry out the design and research of the SPI interface IP core. Through the time sequence simulation test, the SPI interface can correctly transmit data, meet the requirements of SPI timing design, and meet the practical engineering applications.

Keywords

SoC, IP CORE, DMA, SPI

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着集成电路和计算机技术的发展进步，SPI 接口的通信质量受到人们的关注。为进一步提高 SPI 接口通信的质量，人们对 SPI 数据通信的实现方法进行了不断的改进与完善。目前 SPI 数据通信最常见的实现方式可以分为三类：直接无缓冲区通信、带 FIFO 缓冲区通信、以及结合 DMA 控制器通信。首先是直接无缓冲区通信，每一次的通信只能实现最多 4 字节长度的数据传输，但由于部分需要发送的数据报文帧很长，不但增加了软件复杂度，而且同时面临抢占硬件资源的运作时间，从而大大降低了通信质量[1]；第二种采用了具有 FIFO (First in First Out)先进先出缓冲区的通信模式，有了 FIFO 缓冲，有效地解决了报文帧过长的难题，但由于 FIFO 缓冲通常与 SPI 系统模块绑定，而通信数据在被接受后，必须进行一次内存处理，对时间性能要求过高，这便让使用场合显得力不从心；第三种采用 SPI + DMA 的方法，数据的交互传输在传统方式中是通过 CPU 来控制完成的，其原理是 CPU 执行一系列外设发来的指令代码，先将原设备数据信息写入 CPU 内存中，再把内存中的数据信息写入目标设备，适用于数据量较少的情况。而数据量大的时候，其传输的效率会被 CPU 处理指令代码的速度和 CPU 与外设交互所影响，传统的数据交互传输方式不再能够适应这一情况。DMA 的出现解决了 CPU 在处理大量数据搬移过程中时间耗费成本高、速率慢的问题，利用 DMA 控制器的目的就是在释放较小的处理器的同时，也可以使 DMA 控制器以更高效的方式对存储器进行处理，既能解决对 CPU 持续的占用，又能解决内存的搬运问题。

SoC 芯片系统里面最常用的硬件加速模块之一就是 DMA 控制器[2]，具有高带宽直接存储器存取的优势，因为他不需要 CPU 频繁干扰，他可实现系统间不同存储器的高速数据搬移，有效地提高了数据传输效率，从而提高系统整体性能，因此也被广泛地集成在各种 IP 模块里面。在此背景下，本文将基于 SPI + DMA 的方式来提高 SPI 接口通信效率，并且基于 Windows SP7021 IDE 集成开发环境，来进行 SPI 接口 IP 核的设计，从 IP 模块设计到 SoC 系统集成，再到系统软件的编程开发，实现功能验证。

2. 设计原理

2.1. DMA 控制器的设计原理

DMA (Direct Memory Access, 直接存储器访问)是用来实现外设和存储器之间以及存储器和存储器之间的高速数据传输, 允许不经过 CPU 的控制从存储器直接读取数据信息。该操作是基于地址空间标识的原理, 完成了单信道源地址与目的地址两者间数据的单向传输, 为不同系统模块之间的高速数据传输提供了一种可靠的方法。在 DMA 对数据进行搬移时, CPU 从接口单元对 DMA 控制器(Direct Memory Access Controller, DMAC)的配置寄存器进行初始化操作。接收数据时, 由外部设备在准备好数据以后, 直接向 DMAC 发送指令, 即数据信息的地址及大小, DMAC 则负责从外部设备直接将数据信息存储到存储器中。在数据信息发送时, 先定义发送存储器的区域, 由 CPU 直接把传输的数据保存至发送存储区域中, 把存储器的首地址和数据长度分别派发给对应的 DMA 信道。此时, DMAC 的主接口单元模块会根据 DMA 发出的中断请求信号向 CPU 发起总线控制权的申请。CPU 在处理完当前事务之后, 就对 DMA 的请求作出仲裁处理, 如果 CPU 将总线控制权交给了 DMA, 则由 DMA 来请求仲裁处理, 然后根据配置信息对数据进行传输, 在此过程中 CPU 不会介入。在数据传输完成后, DMA 控制器就会发出中断信号, 将总线控制权返还给 CPU, 至此就实现了一次 DMA 的数据传输。DMA 发送数据原理图如图 1 所示。

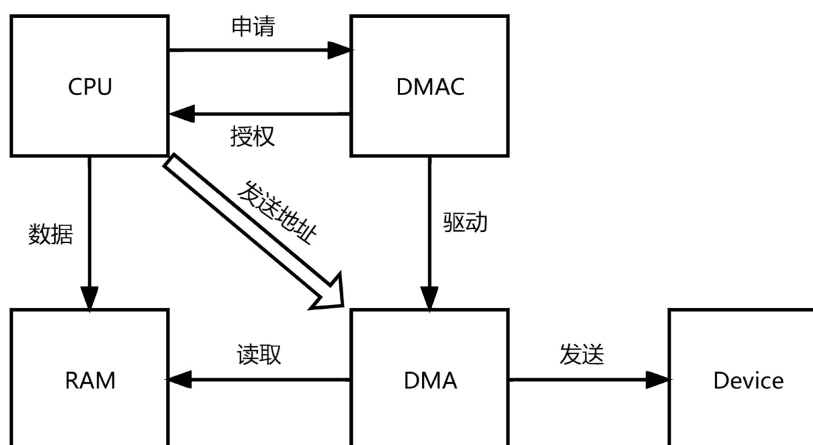


Figure 1. Principle of DMA sending data
图 1. DMA 发送数据原理

2.2. SPI 的设计原理

SPI (Serial Protocol Interface, 串行外围设备接口)是一种同步, 高速、全双工、同步通信的总线, 以主从方式工作, 其逻辑结构如图 2 所示。SPI 接口对数据的传输只需要四根线就能够实现, 节约了芯片的管脚, 同时为节省 PCB 布局空间提供了方便, 正是出于这种简单易用的特性, 现在越来越多的芯片集成了这种通信协议。其中 SCLK (Serial Clock): 串行通讯同步时钟信号, 他的作用是主设备向从设备发送时钟信号, 控制他们两者之间数据传输的时机和速率; MOSI (Master Out Slave In): 在 SPI Master 上也被称作发送通道 TX_channel, 控制 SPI 主机发送数据给 SPI 从机; MISO (Master In Slave Out): 在 SPI Master 上也被称作接收通道 RX_channel, 控制 SPI 主机接收 SPI 从机传送过来的数据; SS/CS (Chip Select/Slave Select): 串行通讯设备选择引脚, 由 SPI 主设备 Master 选择与哪一个 SPI 从设备 Slave 之间进行通信, 低电平表示从机被选中(低电平有效)。在实际运行中, 主设备 Mater 作为主控方, 是对数据发送和接收应答请求的发起方, 它可以通过时钟和数据线, 来保证数据在主设备、从设备两者之间的全双工通信[3]。

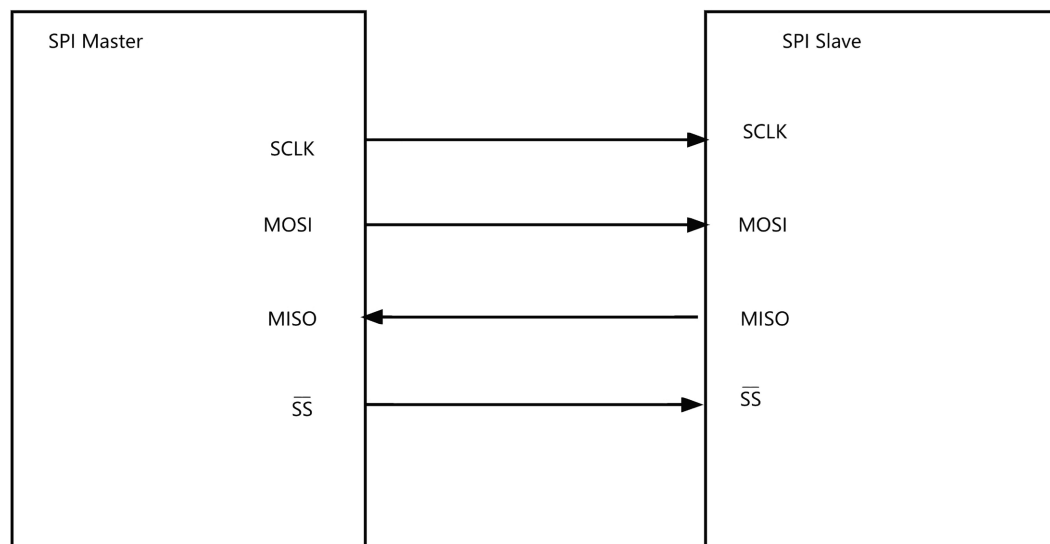


Figure 2. SPI interface logic structure diagram

图 2. SPI 接口逻辑结构图

在两个设备通信开始之前，SPI 从设备将 DMA 收发数据长度和缓冲区设置为最大报文帧长度，同时注册片选信号的外部中断变量，等待着 SPI 主设备进行数据的收发[4]。如图 3 SPI 的通信时序图，SPI 主设备在发送信号之前，会让使能片选信号在整个报文帧发送的过程当中拉低片选信号，在主设备发送报文帧完成后，再拉高使能片选信号，SPI 从设备其外部中断就会在该时该上跳沿信号被触发。在 SPI 从设备的外部中断处理程序中，会对报文帧进行接收和处理，并更新 DMA 数据收发长度及缓冲区最大报文帧长度，等待 SPI 主设备启动下一次的数据通信过程[5]。

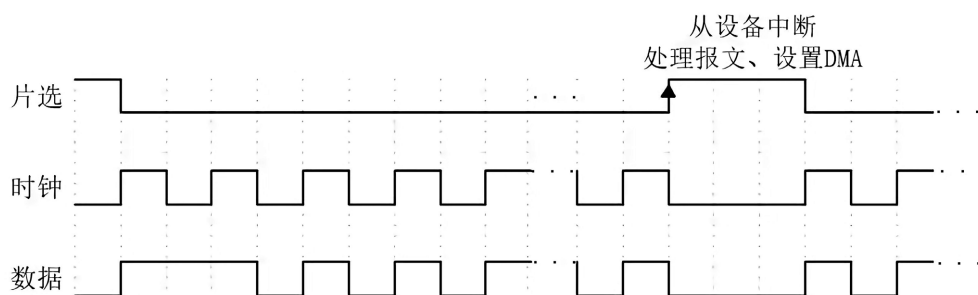


Figure 3. Communication sequence diagram

图 3. 通信时序图

3. 系统架构及设计方法

图 4 是设计的系统框架，整个设计主要被分为两个部分来实现：第一部分根据 SPI 的相关协议并结合项目需求，提取 SPI 功能特性设计出 SPI 功能框架，依照该功能框架对 SPI 接口设计进行模块划分，确定设计规格，利用 SPI_master 接口和 APB_slave 接口来实现数据的读写；第二部分是加入 DMA 控制器和 AXI 接口来实现数据的读写，加入 DMA 之后，一方面可以在芯片启动阶段，通过硬件加速器提供 SPI 高速接口，加速芯片系统启动，在芯片工作阶段，通过硬件加速器提供通用读写 SPI 高速接口，另一方面在数据搬移的时候，采用突发传输，增加数据吞吐量，加速系统对数据搬移的性能，同时降低 CPU 的负荷。

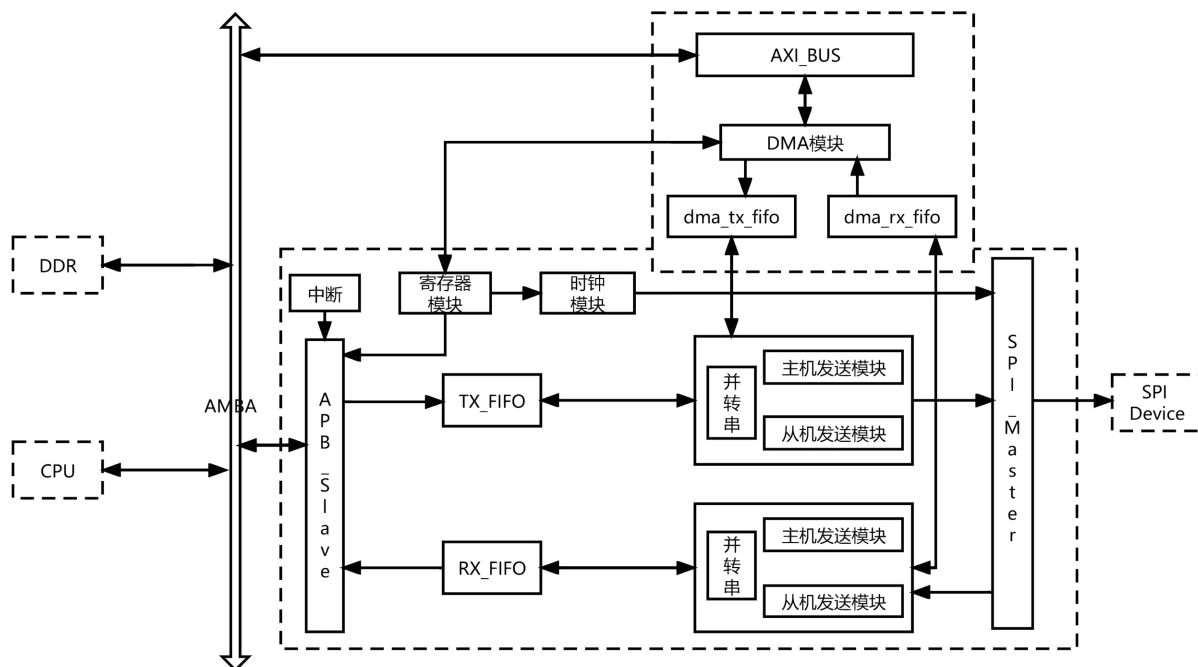


Figure 4. System block diagram
图 4. 系统框图

3.1. SPI 模块的设计

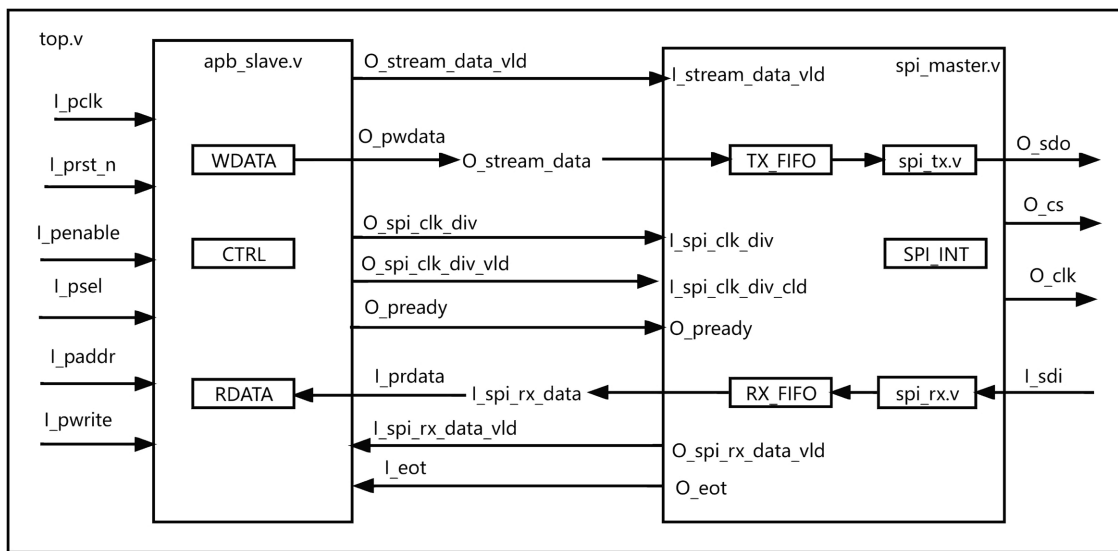


Figure 5. SPI_Master and APB_Connection of slave interface
图 5. SPI_MASTER 和 APB_SLAVE 接口的连接

图 5 是该设计的第一个模块，即基于 APB 总线的 SPI 接口的数据读写，设计的 SPI 协议的 IP 核实现了传统的 4 线结构。这一部分设计主要包括以下的模块：APB 控制寄存器 CTRL、写数据寄存器 WDATA、读数据寄存器 RDATA、SPI 中断处理 SPI_INT、SPI 发送数据模块 spi_tx.v、SPI 接收数据模块 spi_rx.v、发送/接收同步 FIFO。CPU 如果要发送一个数据给 SPI Device，CPU 发送一条指令表明数据

发送请求, 然后通过 AMBA 总线将数据传到 `apb_slave` 接口, `apb_slave` 通过寄存器模块 `W_DATA` 把他写进来, 然后作为 `O_pwdata` 输出端发送到 `TX_FIFO` 模块里面, 通过 `spi_tx` 模块把 `TX_FIFO` 里面的数据发送出去给 SPI Device, 至此就完成了一次数据的传输。在整个通信过程中, 由于 SPI 主从设备之间不需要其他的握手信号, 简化了通信双方的软件设计, 报文帧长可根据当下情况进行调整, 同时 SPI 从设备处理报文、重新设定 DMA 收发数据的时间窗口也从原来的单个字节传输时间的间隔扩展至两个发送报文帧之间的时间间隔, 一般性能的微处理器都能满足, 降低了报文帧接收过程中发生错误的概率。更进一步来说, 该方法只需要 SPI 从设备支持 DMA 传输模式, 此外对每一帧数据发送完毕的判断是以片选信号的电平高低为依据, 并不依赖事先的数据通信或额外的握手信号, 使其更加具有确定性, 所以在数据通信出错的时候可以自动恢复, 降低了发生连续错误累积效应的可能性[6]。

3.2. DMA 模块的设计

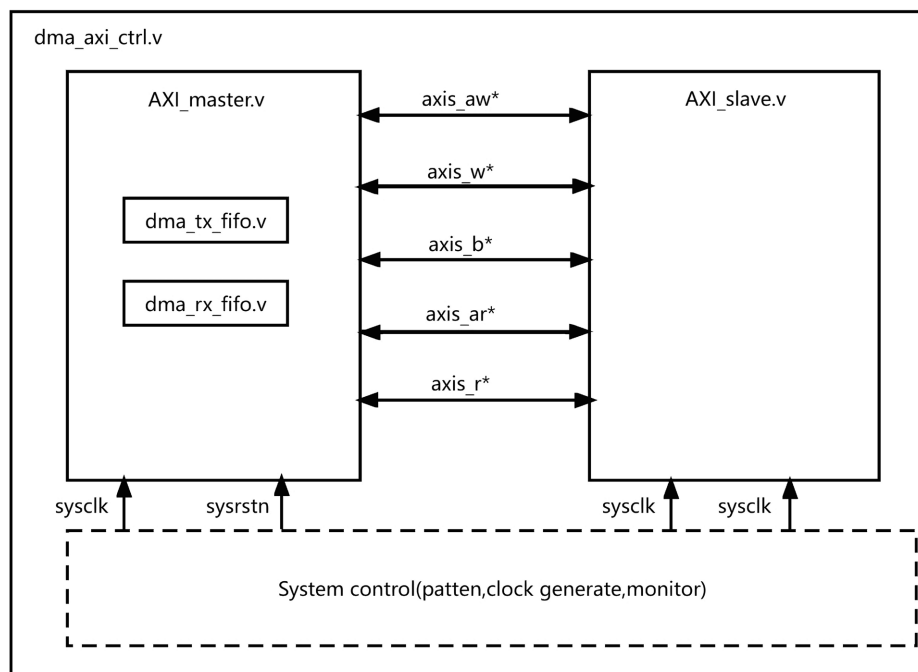


Figure 6. AXI and DMA connection module

图 6. AXI 和 DMA 连接模块

图 6 是该设计整体系统架构的第二个部分, 即把 DMA 和 AXI 相结合, 这一部分设计主要包括以下的模块: 控制寄存器 `DMA_CTRL`、状态寄存器 `STATE`、数据发送模块 `dma_tx`、数据接收模块 `dma_rx`、发送/接收同步 `dma_fifo`。该模块数据的传输采用 burst 突发传输模式, 不仅增加了数据的吞吐量, 还提高了整个系统的性能。DMA 的使用, 使得在芯片启动阶段, 通过硬件加速器提供 SPI 高速接口, 加速了芯片系统启动, 降低了 CPU 的负荷, 提高了系统的整体性能。利用 DMA 控制器来发送数据时, CPU 要发送一个数据给 SPI Device, 他首先通过 AMBA 总线发送一条指令给 AXI 总线, 表明数据通信的需求, 数据被传送过来之后需要被存储到 `dma_tx_fifo` 寄存器中, 之后进行串并转换, 然后由 `axi_master` 把数据传送给 SPI 接口, 至此完成一次 DMA 数据的传送。与 CPU 完成数据传输工作有所不同的是, DMA 数据传输是通过 CPU 来对数据搬移这一动作进行的初始化, 数据传输动作本身是由 DMA 控制器模块来实现的。在实现 DMA 对数据信息进行传输前, CPU 将对总线的控制权直接传给 DMA, 在 DMA 完成数据的

传输后,CPU 将重新获得对总线的控制权,不但降低了 CPU 的负荷,而且大大提高了数据搬移的效率[7]。通常情况下, DMA 都具备 burst 突发数据传输的性能,在这种突发数据传输状态下, DMA 就可以一次传送更多字节的数据,这样,整个系统的吞吐能力就得到大大的提升。鉴于上述优点,在本文设计的这个系统中使用 DMA 能够降低 CPU 的负荷,提高数据吞吐量,整体系统的效能将获得较大的提升。

4. 性能测试与分析

首先使用硬件描述语言 Verilog HDL 创建顶层 testbench,在顶层中为所编写的测试变量提供激励,从测试环境图 7 可以看到,将激励传输给 FAKE_CPU,monitor 会对比 golden data 和 test output data 是否相等。其测试功能包括:硬件复位、各个寄存器的读写及中断,CPU 模式下 SPI 各个指令的数据传输,DMA 模式下,数据的传输。

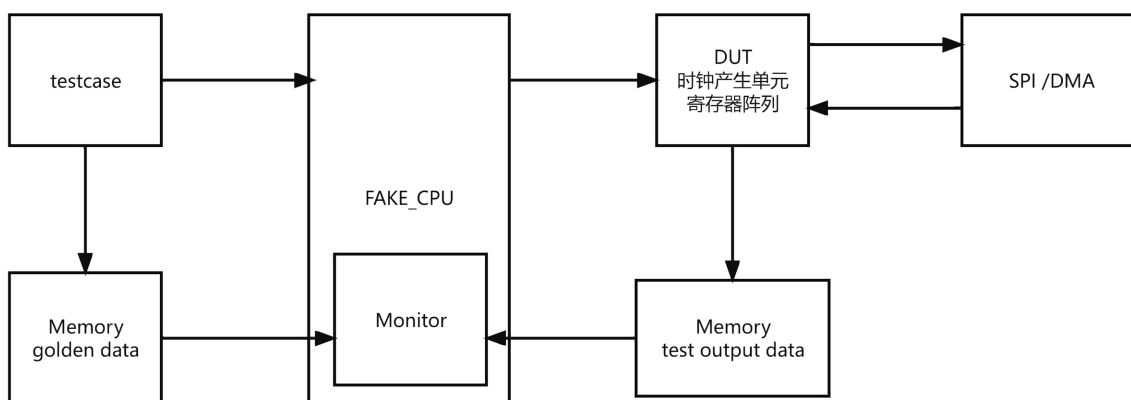


Figure 7. Testing environment

图 7. 测试环境

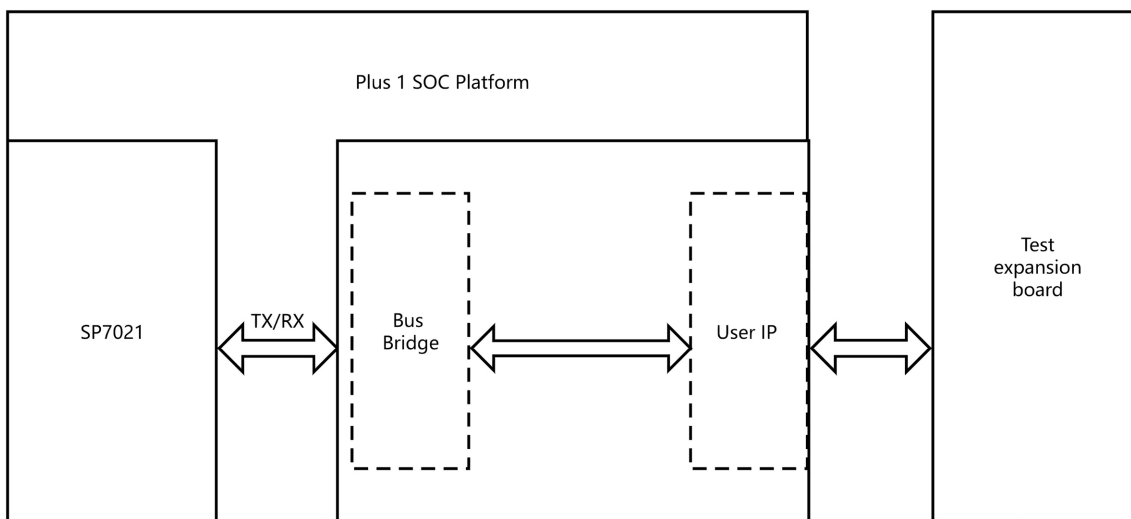


Figure 8. System hardware implementation platform

图 8. 硬件测试平台

仿真实验后,对该设计进行了 FPGA 验证。该项目硬件测试平台是 Plus1 SOC Platform,用 SP7021 SOC 平台测试只需将 Bus Bridge 模块的 axim 和 axis 32bit 总线接口连接在一起,并把相应的地址映射,即将 Bus Bridge 模块起始地址 0x70000000 的访问映射为 SP7021 芯片内 DRAM 起始地址 0x00000000 的访问

[8]。如图 8 所示, SP7021 提供运算所需的 CPU 及 4G DDR 内存, FPGA 是用户验证 IP 所需要的载体, 两者通过 Bus Bridge 模块连接在一起; 根据该项目的应用需求, 再外接测试扩展板(IP board)进行验证。

从图 9 SPI 软件测试图中可以看到数据发送成功的过程, 在 SPI 发送数据的时候, I_{tx_en} 为高电平, 表示允许数据的发送, 片选信号 O_{spi_cs} 为低电平, 表示低电平有效, 同时, 每当发送完一个数据后, O_{tx_done} 将会升为高电平。从图 10 DMA 发送数据软件测试图可以看到, 根据 AXI 协议, 在发送数据时, $TVALID$ 和 $TREADY$ 、 $TKEEP$ 为高电平, 在发送完最后一比特数据时, $TLAST$ 升为高电平。经过测试发现, SPI 每传输完一个字节就要等待一段时间, 且这段时间约为传输时间的十倍, 在加入 DMA 控制器后, 数据的传输采用突发传输, 比正常 SPI 接口数据传输速率高, 效果更好, 总体带宽利用率也提高了 26.9 个百分点, 结果符合设计要求。

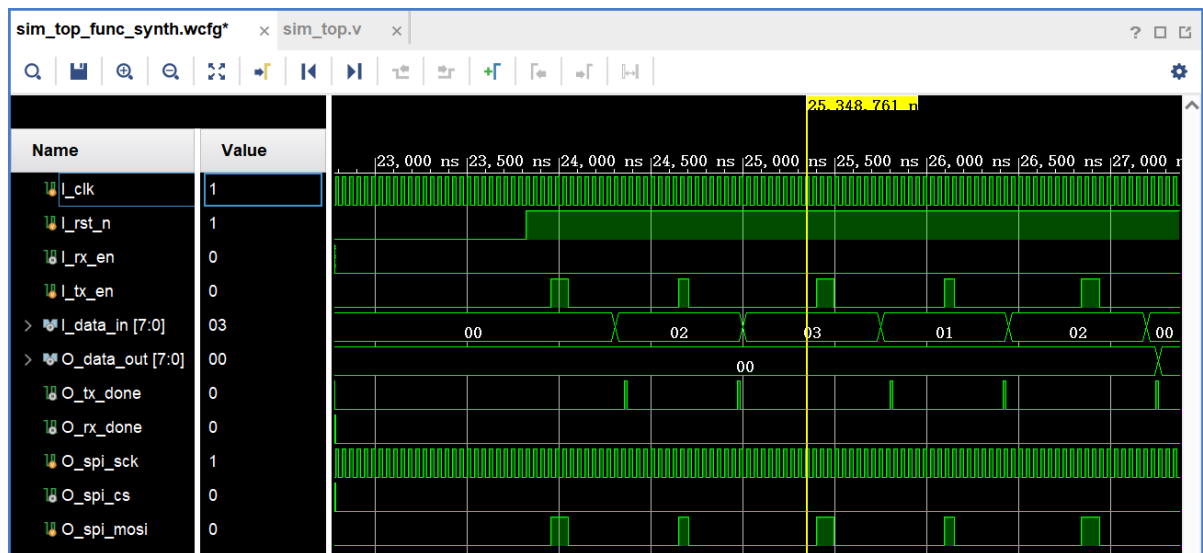


Figure 9. SPI sending data test part

图 9. SPI 发送数据测试部分

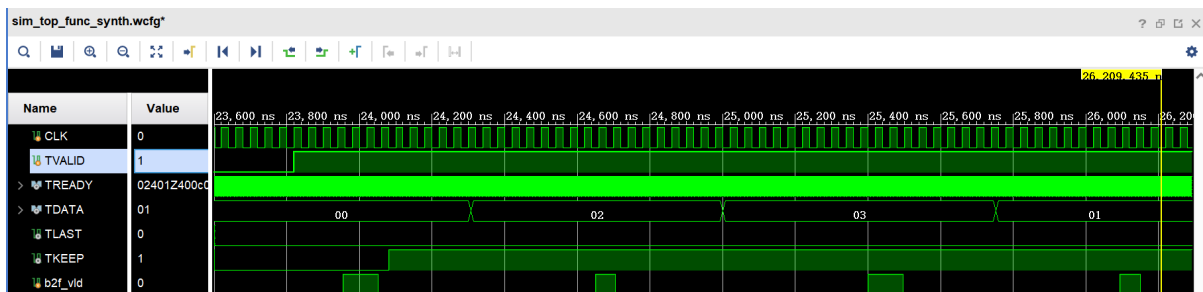


Figure 10. DMA sending data test part

图 10. DMA 发送数据测试部分

5. 结束语

本文介绍了引入硬件加速这一数据发送机制之后, CPU 仅需在 DMA 数据发送完毕或产生中断时, 对缓存区读写指针作出确认, 随后根据需求再一次触发 DMA 控制器的传输即可。该技术可使 CPU 在 DMA 传输数据的进程中无须实时等待, 降低了数据传输所带来的等待时间, 也保证了接口间的数据传输时间周期与 CPU 处理速度二者之间的平衡, 从而大大提高了串行接口信息传输的准确性与有效性。实验

结果表明, 其设计方法简单, 功能有效, 满足设想。综上所述, 本文通过把 SPI 接口与 DMA 控制器组合起来, 做到了大量数据灵活快捷的传输, 并大大地提高了串行接口的数据处理效果与准确度。

参考文献

- [1] 徐阳. 高速 SPI 接口电路的设计与验证[D]: [硕士学位论文]. 西安: 西安电子科技大学, 2020.
- [2] 吴剑箫, 王鹏. 面向超高速以太网 ADB-DMA 的设计与实现[J]. 计算机应用与软件, 2022, 39(6): 116-120.
- [3] 许云龙. 基于 APB 总线的 SPI 接口的设计与实现[J]. 电子质量, 2020(7): 128-132.
- [4] 吴飞. 基于 AHB 协议的 Quad-SPI 控制器设计及 UVM 验证[D]: [硕士学位论文]. 西安: 西安电子科技大学, 2021.
- [5] 郭艾华. 基于 APB 总线接口的 SPI 协议 IP 核的设计与验证[J]. 无线互联科技, 2013(11): 132-134.
- [6] 芮正新, 辛克廷. 一种基于 DMA 方式 SPI 接口的通信方法[J]. 仪器仪表与分析监测, 2020(3): 9-12.
- [7] 何志宏, 丁楚平, 吴峻, 基于 PicoBlaze 的 SPI 核设计与实现[J]. 微处理机, 2010, 31(1): 86-89.
- [8] 凌杰, 刘天奕, 冯艺波. 基于多层 AHB 总线架构的 DMA 控制器设计[J]. 集成电路应用, 2021, 38(8): 12-13.