

SDDR 存储器:新型存储架构设计

鲍丽娜, 张威, 张刚
(太原理工大学信息工程学院, 太原 030024)

摘要: 迄今为止, SDRAM 存储器的扩容完全依赖于半导体工艺水平的升级, 而提速取决于对时钟的利用方式。DDR3 SDRAM 已达 8 倍速率, 再提速已很困难。提出一种新型串行访问的 SDDR 存储器结构和片内串行只写总线, 将 DDR 存储器封装成消息连接的构件, 将访问存储器的命令、地址和数据等信息打成消息报包, 经片内串行只写总线与构件化的 DDR 存储器交换信息。SDDR 存储器减少了引脚, 连接简单并且抗干扰能力强、可靠性高, 易于扩容和进一步提升时钟速率, 具有明显的实用前景。

关键词: 存储器; DDR; SDDR; 串行只写总线; 消息包

中图分类号: V271.4; TP333 **文献标志码:** A **文章编号:** 1671-637X(2014)06-0099-04

SDDR Memory: A New Type of Memory Architecture

BAO Li-na, ZHANG Wei, ZHANG Gang
(Information Engineering College, Taiyuan University of Technology, Taiyuan 030024, China)

Abstract: So far SDRAM memory expansion is entirely dependent on the upgrade of semiconductor technology, and the speed-up is up to the clock utilization pattern. The speed of DDR3 SDRAM has been increased by a factor of 8, hence the further improvement is extremely difficult. A novel serial access SDDR memory structure and on-chip serial only write bus are presented in this paper, encapsulating the DDR memory into components connected by message, packaging such information of the accessed memory as command, address and data and so forth into message packet and exchanging information through on-chip serial only write bus and component-based DDR memory. SDDR memory reduces the number of pins, has simple connection, high anti-interference ability and reliability, and is easy for expansion and improving the clock rate further. Therefore, it has obvious practical prospects.

Key words: memory; DDR; SDDR; serial only write bus; package

0 引言

大数据时代对存储扩容和提速有日益增长的需求, 目前第三代同步动态随机存储器 DDR3 SDRAM (Double Data Rate 3 Synchronous Dynamic Random Access Memory) 的传输速率是第一代同步动态随机存储器 SDRAM 的 8 倍, 仍然不能满足使用要求。但进一步提速遇到并行总线固有的线间脉冲混扰的瓶颈。类似的例子是, 为提高传输速率, 硬盘接口标准 ATA (Advanced Technology Attachment) 发展到第七代 ATA-7, 额外增加了接口引脚数使干扰增大, 限制了其进一步发展^[1], 于是串行接口技术 SATA (Serial Advanced Technology

Attachment) 概念应运而生。SATA 采用点对点方式传输数据, 抗干扰性强且易于提升传输速率。目前 SATA3.0 已具有高达 6.0 Gb/s 的传输速率^[2]。本文研究并设计了一种新型串行访问的 SDDR (Serial Double Data Rate) 存储器结构和串行只写总线 BoW (Bus only Writing), 将 DDR 存储器封装成消息连接的构件, 将访问存储器的命令、地址和数据等信息打成消息报包, 经 BoW 与构件化的 DDR 存储器交换信息。SDDR 存储器减少了引脚, 连接简单并且抗干扰能力强、可靠性高, 打开了进一步提升时钟速率的空间, 具有明显的实用前景。

1 SDDR 存储器原理

传统 DDR 存储器的访问方式是, 主机通过并行总线将地址、命令、数据等信息发送到 DDR 控制器, 控制器解析命令得到访问需求, 并将其转换成 DDR 存储器可以识别的格式, 然后生成相应的控制时序, 发送到存

储器的并行总线^[3]。存储器内部控制部件根据命令进行数据读写操作。

1.1 总体结构

SDDR 存储系统各个模块设计见图 1, 主要包括 SDDR 主机接口、统一节点接口 UNI、串行总线 BoW 和 DDR 存储器构件 4 个部分。

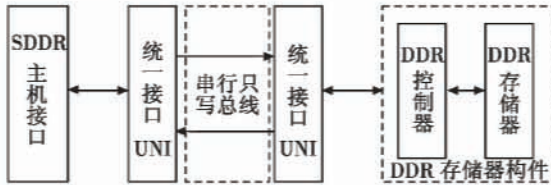


图1 SDDR 存储器模块设计

Fig. 1 SDDR memory module design

SDDR 存储器以 DDR 存储器构件为基础, DDR 存储器构件将 DDR 控制器与 DDR 存储器封装在一起; SDDR 主机接口依然采用 DDR 并行总线接口协议, 将主机访问 DDR 的命令、地址和数据提交给统一节点接口 UNI, UNI 内部的串/并转换电路将它们转换成串行消息, 通过串行只写总线 BoW 将消息传输到接收端; 接收端的统一节点接口 UNI 将消息恢复成并行方式, 生成访问 DDR 存储器构件的时序和命令, DDR 存储器构件内部的 DDR 控制器根据 UNI 提交的访问命令和时序进行 DDR 存储器的读写操作。

1.2 SDDR 串行数据帧格式

参照以太网帧格式定义 SDDR 串行数据帧格式, 如表 1^[4] 所示。

表1 串行数据帧格式

Table 1 The format of serial data frame

先导/B	帧分界/B	目的地址/B	源地址/B	命令/b
2	1	4	4	3
存取地址/b	数据长度/b	数据/填充/B	校验/B	
25	10	128	4	

先导: 表示为“1010101010101010”, 用于物理介质的稳定和同步。帧分界: 表示为“10101011”, 有效数据开始。目的地址: 表示发送到具体的某个 DDR 存储器, 用 4 B 长度数据表示。源地址: 表示发送数据的源地址, 用 4 B 长度数据表示。命令: 表示数据在 SDDR 中存取指令, 用 3 bit 长度数据表示。存取地址: 表示数据在 SDDR 中存取所需要的地址, 用 25 bit 长度数据表示。数据长度: 表示发送数据的数据长度, 用 10 bit 长度数据表示。数据/填充: 即有效数据, 传输的数据长度至少为 256 bit, 不够时通过填充凑足最小数据长度。1 个填充数据为“11001100110011001100110011001100”。如果是读命令, 当消息帧发送给 SDDR 存储器时, 该段只有前 32 位表示的是要读取的数据长度, 其余均为填充数据。校验: 表示帧校验。

2 串行只写总线 BoW

基于 BoW 总线的一个 SDDR 基本单元 IU (Infrastructure Unit) 见图 2, 它由一个主节点 MPN 和若干 SDDR 存储器从节点 SDDR1, SDDR2, ..., SDDR_n 各自通过统一节点接口 UNI 连接到只写总线 BoW 互联组成。

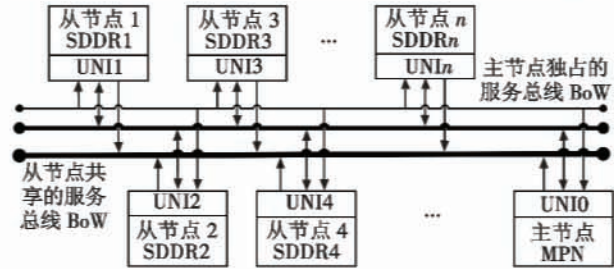


图2 基于只写总线的 SDDR 基本单元

Fig. 2 SDDR IU based on BoW

其中, 只写总线 BoW 包含 2 条消息传输通道和 1 条状态总线 FLAG; 主节点 MPN 是基本单元 IU 的主机访问接口; 从节点是本文研究提出的 SDDR 存储器模块。主处理节点 MPN 和从节点 SDDR 均通过各自配置的统一节点接口 UNI 连接到 BoW 总线, 其中, MPN 独享一条 BoW 通道, 可以向所有 SDDR 发送消息帧; 多个 SDDR 从节点共享另一条 BoW 通道, 可以分别独立地向 MPN 发送消息帧; 另外设置一条状态总线 FLAG, 用来解决多个 SDDR 同时发送消息引起的冲突^[5]。

当某个 SDDR 发送消息时先检测状态 FLAG, 当 FLAG 为空闲状态“0”时, 先将状态 FLAG 置为忙状态“1”, 然后发送消息帧, 消息发送结束后再将状态 FLAG 置为空闲状态“0”。当检测状态 FLAG 为忙状态“1”时, 根据优先级和算法特性设置一个延迟时间 Δ , 延迟时间 Δ 结束后重新检测总线状态 FLAG, 直到状态 FLAG 空闲, 启动并完成消息发送任务。状态总线 FLAG 的默认设置为空闲状态“0”。

串行总线 BoW 是实现 SDDR 存储器系统的关键, 本文采用总线型低压差分信号 (LVDS) 技术构建高速串行总线, 在控制器与存储器控制接口之间实现高速可靠的数据传输^[6]。

3 统一节点接口 UNI

统一节点接口 UNI 的作用是将只写总线 BoW 封装成一个总线模块, SDDR 基本单元 IU 的主节点 MPN 或从节点 SDDR 均通过 UNI 挂接到这个总线模块上实现互联。

3.1 存储器端

在 SDDR 存储器端 UNI 主要实现存储器接口, 包括并/串和串/并转换电路、数据缓存、校验和转换单元

等,基本原理见图 3。

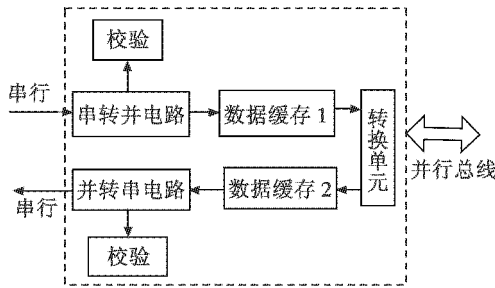


图 3 统一节点接口原理

Fig. 3 UNI principle

数据缓存采用 ISE 自带双端口 FIFO IP 核,其中数据缓存 1 输入端口采用高速串行时钟,位宽 1 bit;数据缓存输出端口采样时钟采用 DDR 控制器的单倍时钟,位宽 32 bit。串转并电路从 BoW 总线接收消息并解包,将有效数据送入缓存,将读写命令和地址发送给转换单元。转换单元生成读写命令的时序,将读写地址送入 DDR 存储器的并行总线,实现数据缓存与 DDR 存储器之间的信息交换。

并转串电路对 BoW 总线发送串行数据,将数据缓存 2 的 32 bit 数据转换为 1 bit 的串行数据流,封装成消息,按照帧格式发送到 BoW 总线。

串转并电路接收来自 BoW 总线的串行消息帧并对消息解包,分析出有效数据,将 1 bit 的串行数据流转换为 32 bit 的数据提交到数据缓存 1,将命令和存取的地址提交给转换单元,其工作流程见图 4。

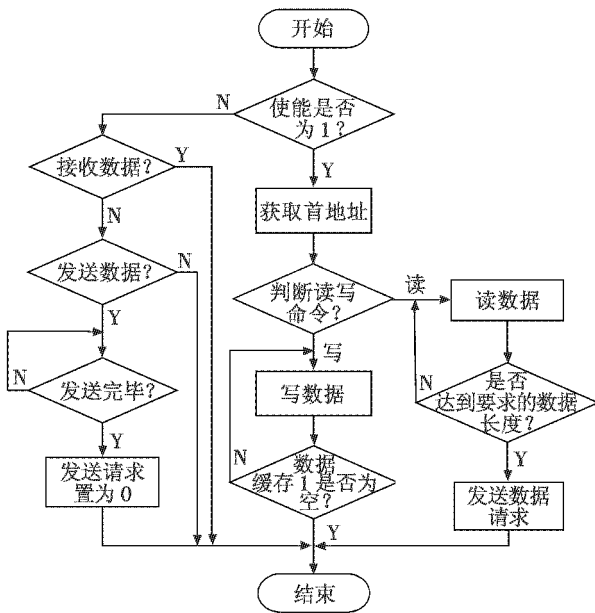


图 4 转换单元工作流程

Fig. 4 Diagram of converting unit

转换单元主要实现如下功能:当读写使能为 1 时写命令,转换单元将数据缓存 1 中的数据依次写入

DDR 存储器;读写使能为 0 时读命令,根据要求的数据长度,转换单元从 DDR 存储器读出数据并送入数据缓存 2。数据缓存 2 数据输入端口采样时钟采用 DDR 控制器的单倍时钟,位宽 32 bit;输出端口采用高速串行时钟,位宽 1 bit。在 DDR 存储器构件读数据的同时,转换单元向总线 BoW 发送请求,准备发送数据。

3.2 主机端

UNI 在主机端主要实现与主机接口的互联,其基本原理、缓存、位宽和时钟的定义与存储器端的 UNI 基本相同,只是内部操作略有区别。

SDDR 基本单元 IU 的主节点 MPN 是 SDDR 主机接口,负责将读写命令、地址(和写操作时的数据)一起提交到统一节点接口 UNI 的转换单元,进行操作命令解析和数据缓存访问。读写使能为 1 时写命令,转换单元将数据缓存 2 中的数据依次提交到并转串电路,然后并转串电路将 32 bit 并行数据转换为串行数据流送入 BoW 总线。读写使能为 0 时读命令,串转并电路把从 BoW 总线接收的串行数据转换为 32 bit 并行数据送入数据缓存 1,转换单元从数据缓存 1 读出数据并提交到主机。

4 SDDR 存储器构件

构件是软件领域的一个概念,其通用性、可变性和可靠性均很高^[7]。本文把它扩展到集成电路设计中。凡是具有统一接口标准,基于消息的形式通过接口访问的功能实体均叫作构件。例如,通过电路联接的一个 ATA 硬盘作为存储数据的功能实体,具有统一接口标准,但不是基于消息访问的,不符合构件的条件。但将 ATA 电路接口封装成 SATA 消息接口之后,SATA 硬盘即成为一个构件,可以通过统一节点接口 UNI 挂接到 BoW 总线模块中。

SDDR 存储器构件将 DDR 控制器与 DDR 存储器封装在一起,设计相应的接口电路与统一节点接口 UNI 相连接,负责实现对 DDR 存储器的读写访问。

5 仿真验证

在 Xilinx ISE 验证环境下采用 Xilinx 公司提供的 Virtex2P 系列的 XC2VP30 开发板,串行总线时钟为 450 MHz,仿真见图 5。首先向 SDDR 存储器写入 10 个 32 bit 的数据“10101010”,然后随机从 SDDR 存储器中读取 4 个数据。图中,rasn、casn、wen 是控制命令,先完成 DDR 存储器的初始化,然后发送 10 个写命令“100”,经过一段时间发送 4 个读命令“101”,由图 5 可知,命令是正确的。add 是给出的 DDR 存储器的存取地址,dq_out 是主机端 UNI 发送的串行数据,st_o 表

示 SDDR 控制器端向 SDDR 存储器接口端发送帧的状态转移:当发送与写命令有关的信息时, $st_o = 0Dh$ 表示 dq_out 发送有效数据,即 10 个“10101010”;当发送与读命令有关的信息时,有效数据长度不够 256 bit,即 $st_o = 0Eh$ 表示填充数据。 din_in 是存储器端 UNI 接收到的串行数据,从图 5 中看出也是 10 个“10101010”,可见读取访问正确。 dq_in 是从 DDR 存储器中读出的数据“1010”。DATA_OUT 表示从 DDR 存储器中读出 16 bit 数据整合为 32 bit 数据,是“10101010”。 $data_out$ 表示将 DATA_OUT 的 32 bit 数据经过转换单元转换,再从存储

器端 UNI 经过串行总线发送给主机端 UNI 的串行数据,即 4 个 32 bit 的“10101010”,由于该有效数据的长度不够最短长度,所以由填充数据补足 256 bit。 dq_in 表示 SDDR 控制器接收到的数据,当 $st_rv_o = 15h$ 时表示接收有效数据,即 4 个“10101010”,见图 6,它是将图 5 的 dq_in 和 $data_out$ 部分放大后的显示。

以上验证过程表明该存储器设计正确,目前正在研究将其扩展为 SDDR 存储器阵列,在存储器扩容的同时实现访问提速,为进一步将其实现为 SDDR 存储器集成电路芯片进行设计验证。

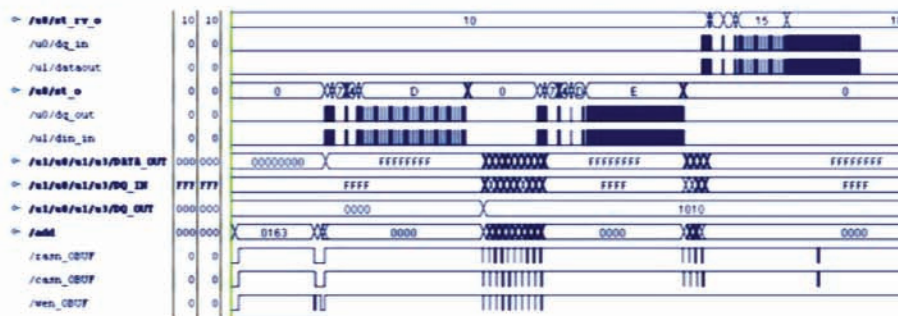


图5 Chipscope 波形

Fig. 5 Chipscope waveform

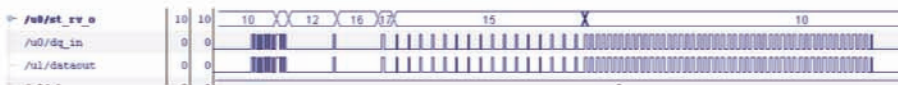


图6 放大后波形

Fig. 6 Amplified waveform

6 总结与讨论

本文提出一种新的 SDDR 存储器系统,详细介绍了其工作原理。以传统的 DDR 存储器为基础,引入高速串行总线实现快速存取数据,是存储器领域的一个创新性产品。SDDR 存储器传输速率高、抗干扰性能强,可以进一步扩展为存储器阵列,具有明显的应用前景。

参考文献

- [1] 寇科男. SATA 接口技术研究及设计[D]. 哈尔滨:哈尔滨工业大学,2010.
KOU K N. SATA interface technology research and design [D]. Harbin:Harbin Institute of Technology, 2010.
- [2] 范俊. 基于 FPGA 的 SATA 控制器的研究与实现[D]. 武汉:华中科技大学,2006.
FAN J. Design and implementation of SATA adapter using FPGA [D]. Wuhan:Huazhong University of Science and Technology, 2006.
- [3] 雷海军,刘鹏,陈战夫,等. 视频格式转换系统中 DDR 控制器设计[J]. 电视技术,2011(14):22-24.
LEI H J, LIU P, CHEN Z F, et al. Design of DDR controller in video format conversion system[J]. Video Engineering,

2011(14):22-24.

- [4] 郭晓宇. 基于 IEEE802.3 标准的以太网数据帧格式的封装实现[D]. 北京:北京交通大学,2008.
GUO X Y. The encapsulation implementation of Ethernet data frames format based on IEEE802.3 standard [D]. Beijing:Beijing Jiaotong University, 2008.
- [5] 王治法. 新型 DDR SDRAM 存储器架构(SDDR)的设计[D]. 太原:太原理工大学,2012.
WANG Z F. Design of new DDR SDRAM memory architecture (SDDR) [D]. Taiyuan:Taiyuan University of Technology, 2012.
- [6] 蔡月明,仇新宏,李惠宇,等. 基于低压差分信号高速串行总线的智能变电站硬件平台设计[J]. 电力系统自动化,2012(21):73-76.
CAI Y M, QIU X H, LI H Y, et al. Smart substation hardware platform design based on low voltage differential signal high-speed serial bus[J]. Automation of Electric Power Systems, 2012(21):73-76.
- [7] 马鸣. 基于构件的软件工程理论与方法探讨[J]. 电子测试,2013(16):5-6.
MA M. Discussion of software engineering theory and method based on component [J]. Electronic Test, 2013(16):5-6.