

DOI:10.19651/j.cnki.emt.2108033

一种基于 DDR 的 PS 与 PL 数据交互方法的设计与实现

陈小宇 李常对 阳梦雪

(华中师范大学 物理科学与技术学院 武汉 430079)

摘要: 针对片上系统芯片中处理系统和可编程逻辑之间数据交互量大的应用,提出了一种基于双倍速率同步动态随机存储器的 PS 与 PL 数据交互方法。PS 与 PL 通过访问共同的 DDR,按照自定义的协议进行数据交互。将 DDR 中用于数据交互的空间划分为指令空间和数据空间,PS 和 PL 通过读写指令空间中的指令数据并按照协议分析其所传递的信息,以控制各自的读写进程。PL 通过高速片内总线访问 DDR,PS 利用内存读写工具实现对 DDR 的读写。测试结果表明该交互方法具有速度快、占用逻辑资源少、使用方便等优点,数据交互速度可达 88 MB/s,适用于 PS 和 PL 需要实时交互大量数据的应用场景,在基于三维激光雷达的车辆实时高精度定位系统中得到了成功应用。

关键词: 片上系统;数据交互;AXI 总线;处理系统;可编程逻辑;双倍速率同步动态随机存储器

中图分类号: TN409 **文献标识码:** A **国家标准学科分类代码:** 510.4030

Design and implementation of data interaction method between PS and PL based on DDR

Chen Xiaoyu Li Changdui Yang Mengxue

(College of Physical Science and Technology, Central China Normal University, Wuhan 430079, China)

Abstract: Aiming at the application of large amount of data interaction between processing system and programmable logic in system of chip, a data interaction method between PS and PL based on double data rate synchronous dynamic random access memory is proposed. PS and PL interact with each other according to the custom protocol by accessing the common DDR. The space for data interaction in DDR is divided into instruction space and data space. PS and PL control their reading and writing processes by reading and writing instruction data in instruction space and analyzing the information they transmit according to the protocol. PL accesses DDR through high-speed on-chip bus, and PS uses memory read-write tools to read and write DDR. The test results show that the interaction method has the advantages of high speed, less logic resources and convenient use. The data interaction speed can reach 88 MB/s. It is suitable for the application scenario where PS and PL need to interact a large amount of data in real time. It has been successfully applied in the vehicle real-time high-precision positioning system based on three-dimensional lidar.

Keywords: system of chip; data interaction; AXI bus; processing system; programmable logic; double data rate synchronous dynamic random access memory

0 引言

近年来,由三维激光雷达所生成的点云数据被广泛应用于导航定位、目标识别和模型构建等场景中^[1-4],点云数据量很大,且需要经过滤波、特征提取、配准和分割等多种处理^[5-7],对中央处理器(central processing unit, CPU)的要求很高。多核异构平台片上系统(system of chip, SoC)在单芯片结构上集成了处理系统(processing system, PS)和可编程逻辑(programmable logic, PL),可以充分发挥

PS 和 PL 各自的优势,利用 PS 对系统各个模块进行统一调度,将采集得到的大量点云数据传输给 PL,PL 对数据进行并行、实时和快速处理后将结果回传给 PS,此架构兼备 PL 的并行计算能力与 PS 的多任务调度能力^[8]。基于 PS+PL 双核心架构设计出的产品,往往具有动态响应快、可靠性高、实时性好、抗干扰能力强等特点^[9]。

当前 PS 与 PL 数据交互方法主要有 3 种。第 1 种是通过块随机存储器(block random access memory, BRAM)

实现 PS 与 PL 之间的交互,即处理器直接将数据写入块随机存储器,该方法缩短了数据的存储路径,极大地降低了工程结构的复杂度^[10],但其传输的数据量十分有限。第 2 种是利用 AXI(advanced eXtensible interface)总线,其拥有 3 种不同的接口标准,分别是 AXI-Memory Map、AXI-Stream 和 AXI-Lite^[11],可根据此设计一种互联结构,用于 PS 与 PL 之间的数据交互,此结构能够充分发挥出 AXI 协议的优点,在点对点传输时实现低延时、高效率和高吞吐率的特点^[12],此方法不需要利用外设,但程序复杂度较高。第 3 种是利用 GPIO 引脚,具体来说就是 EMIO(extendable multiuse I/O),通过拉高或者拉低 EMIO 引脚的电平,从而实现 PS 与 PL 之间的数据交互,此方法灵活性高,但传输的数据量很少,多用于控制信号或者握手信号的交互,以完善 PS 和 PL 双方的通信机制^[13],例如开关、复位等信号的传输。

本文针对 PS+PL 的架构,以 Xilinx 公司的 Zynq UltraScale MPSoC 芯片为例,设计了一种基于 DDR 的 PS 与 PL 之间的数据交互方法。PS 与 PL 通过访问挂载于 PS 端的 DDR 进行指令传达和数据传输,并且根据自定义的交互协议,实现了指令更新和数据交互。

1 系统总体设计方案

Zynq UltraScale+EG 系列的 MPSoC 芯片拥有四核 ARM Cortex-A53 处理器,双核 ARM Cortex-R5 处理器,以及 16 nm 的 FinFET+可编程逻辑单元,支持图形、视频、波形与数据包处理^[14]。PS 与 PL 通过访问 PS 端挂载的双倍速率同步动态随机存储器(double data rate synchronous dynamic random access memory, DDR SDRAM)实现数据交互。图 1 为本方案的系统总体框图。

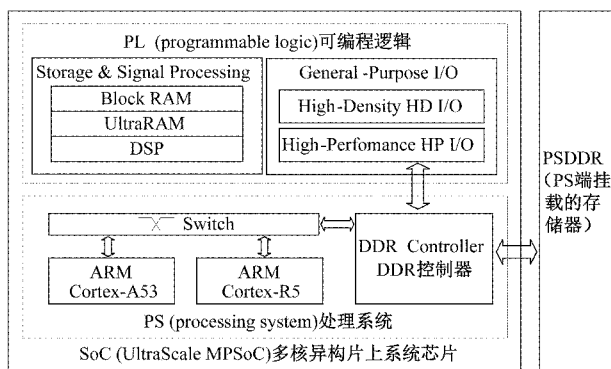


图 1 系统总体框图

PL 端有存储单元和信号处理单元,并且有资源丰富的通用 I/O 接口和高速通信接口。PL 可通过高性能 (high performance, HP) 接口与 PS 相连^[15]。PS 端 ARM 处理器可通过开关交互模块连接至 DDR 控制器,然后访问外设存储器 DDR,这大大扩展了芯片的内存空间。PL 与 PS 均可经过 PS 端的 DDR 控制器访问 DDR,实现指令的传达和数据的交互^[16]。

2 数据交互协议设计

2.1 DDR 空间划分设计

DDR4 中分别开辟指令空间和数据空间。指令空间中存放的指令数据决定了 PS 和 PL 何时读取数据、何时写入数据。数据空间则存储需要交互的数据。图 2 为 DDR 空间划分示意图。

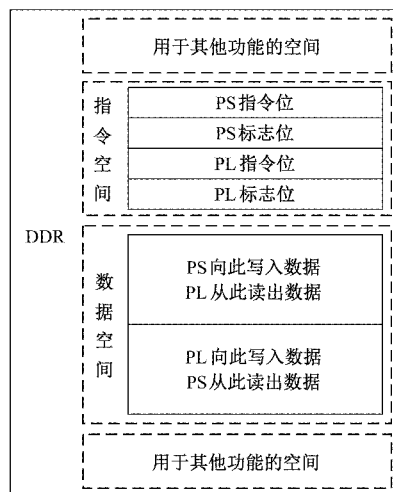


图 2 DDR 空间划分示意图

用于数据交互的空间应根据实际需求,事先在交互协议中约定好偏移地址、空间大小等参量。在本设计中,使用的是连续内存分配器(contiguous memory allocator, CMA)预留的空间,因为 CMA 可能会有其他外设参与使用,而高位地址被使用的概率几乎为 0,所以数据交互空间的首地址可以从比较靠后的地址开始。

访问 DDR 时,寻址的基本单位为字节。在本设计中,选择 128'h7000_0000 为起始地址,即从 DDR 的 0 地址开始,偏移了 1.75 GB 的空间。数据交互的空间大小可根据实际需求作灵活调整,只要不超出 CMA 预留的空间即可。

指令空间中的数据以及数据空间中的数据均可按照所设计的交互流程重叠写入,即每一次新的数据来临时就将上一次的数据覆盖。交互流程可避免空间竞争。

2.2 交互流程设计

PS 指令位和 PS 标志位上的数据由 PL 读取,PL 通过分析判断读取到的指令和标志,来控制自己的读写进程,此流程在 SoC 芯片上电运行后循环执行,如图 3 所示。

PS 每写入一次新的数据,则将计数值加 1 并写入 PS 指令位中,PL 读取 PS 指令位,通过比较前后两次读出的计数值的大小,即可判断 PS 是否写完数据,只有 PS 写完数据后 PL 才能读取 DDR,同时也可避免 PL 重复读取相同的数据。

PS 在从 DDR 读取数据的过程中,将一个特殊标志如 128'd1 写入标志位中,PL 读取到此特殊标志,则处于等待

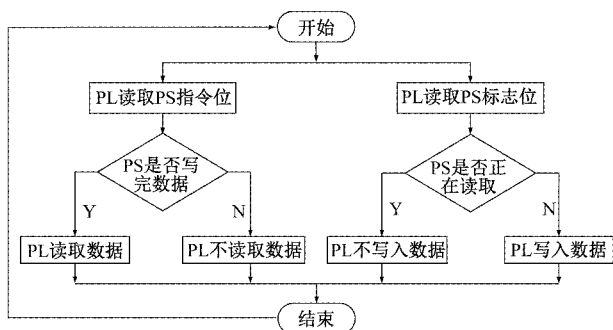


图3 PL读取指令和标志流程

状态,直到读取到PS标志位中的数据不是特殊标志后,PL再向数据空间写入新的数据。

PL向DDR指令空间中写入指令位和标志位的流程也是循环执行的,其写入的数据由PL端读写模块运行状态所决定。其流程如图4所示。

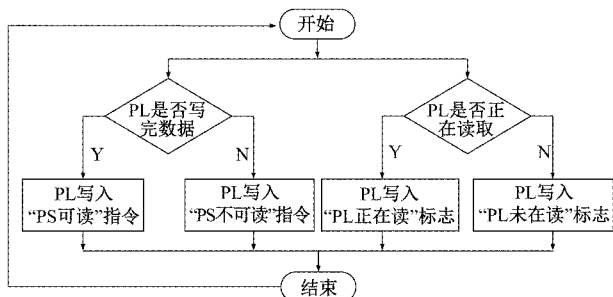


图4 PL写入指令和标志流程

当PL写完新的数据后,则将计数值加1写入PL指令位中,表示写入了新的一帧数据,PS可以读取,若不是,则保持上次的计数值不变,表示PS不可读取。

当PL正在读取数据时,则将一个特殊标志如128'd1写入PL标志位中,表示PL正在读取数据,否则,则写入非特殊值,表示PL未在读取数据,此时PS可以向数据空间写入新的数据。

3 PS端实现

3.1 Linux环境下建立内存映射

Linux操作系统因其代码开源、可移植性强、功能强大、定制方便等优势在众多领域中得到广泛应用^[17]。为了使本交互方法具有较高的推广价值,PS端的交互程序运行在Linux操作系统下。

Linux为每个进程维持了一个单独的虚拟地址空间,并且将地址空间划分为固定大小的页面^[18]。内存映射就是将进程虚拟空间和内核虚拟空间映射到同一个物理页面,实现用户态和内核态共享内存。映射成功后,用户对这段物理内存的操作就会直接反应到内核空间,相反,内核空间对这段物理内存的操作也同样会反应到用户空间^[19]。由此可知,PS端应先从Linux内核中分配一块内存空间出

来,然后通过映射,就可以确定PS和PL在交互时所访问的DDR的物理空间地址,这样便可防止数据交互空间与系统运行空间发生冲突,避免空间竞争。

3.2 PS端软件流程设计

PS端软件流程如图5所示。系统上电初始化后,PS端程序先定义需要映射的内存空间的大小,然后将内存空间映射到DDR上,为防止翻页处理,一次性将内存空间申请充足。因为DDR上电后其内部所存储的数据是不确定的,是无效的,尤其是指令位和标志位上的数据,若是随机数据,有可能引起PS和PL的误操作,所以要先对DDR进行初始化操作,调用写入函数Devmem_Write,将所申请映射的整块内存空间写入0,之后,根据所设计的数据交互协议,分别调用读取函数Devmem_Read和写入函数Devmem_Write^[20],根据指令位和标志位所传递的信号进行读取或写入的操作。

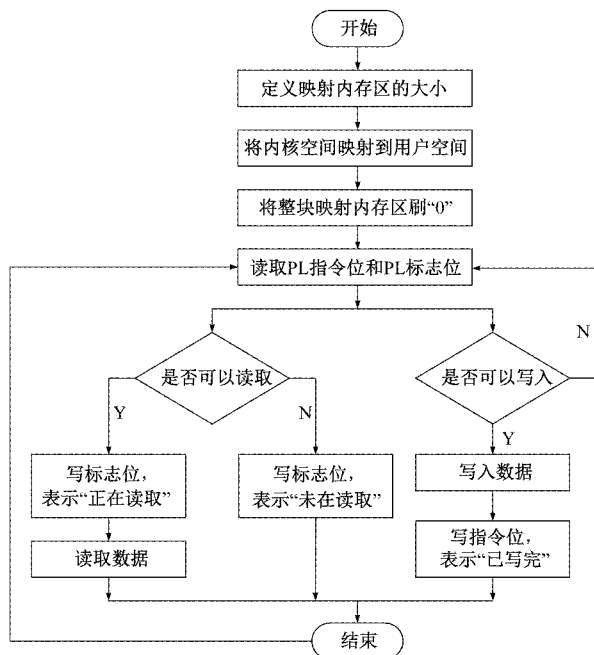


图5 PS端软件流程

4 PL端实现

4.1 PL端读写架构设计

PL端读写架构如图6所示。PL端交互系统有5个模块,分别是读取指令和标志模块Read_Flag、写入指令和标志模块Write_Flag、读取数据模块Read_Data、写入数据模块Write_Data以及FIFO数据缓冲模块。

根据交互协议,Read_Flag模块通过读取PS指令位,得到计数值PS_wr_cnt,该值传递到Read_Data模块,只有当前计数值大于上一次的计数值时Read_Data模块才会工作,从DDR中读取数据然后存入FIFO中缓冲,否则Read_Data模块不会工作。

Read_Flag模块通过读取PS标志位得到一个信号

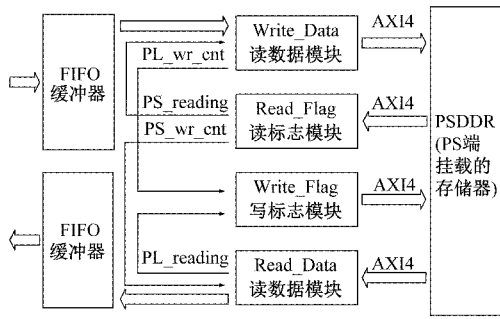


图 6 PL 端读写架构

PS_reading, 并传递到 Write_Data 模块, 当此信号值为 1 时, 表示 PS 正在读取数据, 则 Write_Data 模块停在 IDEL 空闲状态, 不向 DDR 写入新的数据, 只有此信号值为 0 时 Write_Data 模块才会工作, 将缓存在 FIFO 中的数据写入 DDR 中。

Write_Data 模块向 DDR 中写完新的一帧数据后, 会将计数值 PL_wr_cnt 加 1 并传递给 Write_Flag 模块, Write_Flag 模块则会将此值写入 PL 指令位上, 用以控制 PS 端的读取进程。

Read_Data 模块在读取 DDR 中的数据时, 会传递给 Write_Flag 模块一个高电平的 PL_reading 信号, 此时 Write_Flag 模块会将 128'd1 写入 PL 标志位上, 表示 PL 正在读取数据, PS 不可以写入新的数据。

4.2 PL 端读写逻辑设计

负责读取和写入的 4 个模块均通过 AXI 总线来实现对 PS 端 DDR 的操作。AXI 总线是一种高性能、高带宽、低延迟的片内总线, 它的地址、控制和数据通道是独立的, 并且是基于 burst 突发式的传输^[21]。突发式读写有 3 种类型: 固定式、增值式和包装式。包装式突发读写只需要知道包的首地址和包的大小, 即可以在传输过程中自动增加地址值, 直到到达包的边界。简化后的 AXI 总线包装式突发读写的时序图如图 7 所示。

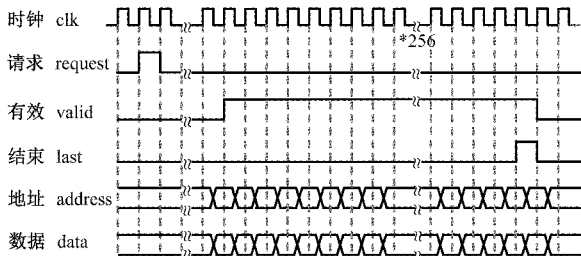


图 7 AXI 总线包装式突发读写时序图

若要读写数据, PL 端先将地址寄存器和数据寄存器中的数据准备好, 然后发送请求信号 request, 该信号持续一个时钟周期的高电平。等到 AXI 总线上的 valid 信号为高电平时, PL 在每个时钟的上升沿将寄存器中的数据赋给地址线和数据线。传输到最后一个数据时, 会有持续一个时

钟周期的高电平信号 last 产生, 代表此次突发式读写任务的结束。

根据 AXI 总线包装式突发读写时序, PL 读写模块状态转移图如图 8 所示。IDEL 为空闲状态, 主要对各个变量进行初始化, 初始化后转移到 JUDGE 状态; 在 JUDGE 状态时, 会根据其他模块传递过来的信号判断是否可以读写, 可读写则转移到 REQUEST 状态, 不可读写则返回 IDEL 状态; REQUEST 状态中首先判断是否读写完毕, 若读写完毕, 则转移到 IDEL 状态, 若未完毕, 则发送请求 request 并转移到 LAST 状态; LAST 状态中首先判断 last 是否为高电平, 若为高电平, 则代表本次包装式突发读写已完成并转移到 REQUEST 状态, 若为低电平则继续读写数据。

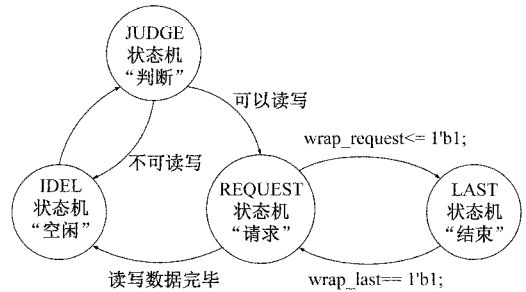


图 8 PL 读写模块状态转移图

5 测试与验证

测试采用的芯片型号为 xczu15cg-ffvb1156-2-i, DDR 由 4 片 Micro 公司生产的 mt40a512m16ly-062e-it-e 芯片组成, 单片容量 1 GB, 4 片容量共 4 GB, PL 端程序运行时钟为 100 MHz。测试硬件平台实物如图 9 所示。

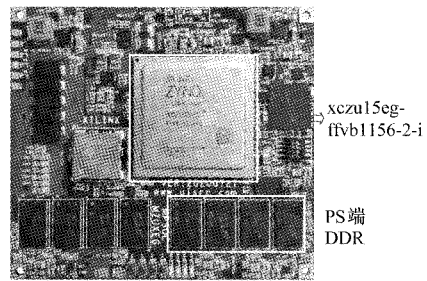


图 9 测试平台实物

PL 端的整个工程在 Vivado2019.2 的环境中开发, 表 1 是 PL 端工程由 vivado 实现 (Implementation) 之后统计得到的逻辑资源使用情况表, 由表可知, 该交互系统只使用了 PL 端较少的资源。LUT、FF 和 LUTRAM 资源都只使用了不到 0.5%, BRAM 资源使用了不到 1.5%, 这样就可以节省出更多的资源给 PL 端其他模块使用。

表 2 是基于 BRAM 的 PS 与 PL 数据交互的方法由 Vivado 实现后统计得到的逻辑资源使用情况表, 在该交互方法中, 对 BRAM 开辟了 2 MB 的存储空间, 由表可知, BRAM 的资源已经使用了 68.82%。

的三维激光点云数据采集和算法处理中得到了成功应用。

参考文献

- [1] MOLEBNY V, MCMANAMON P, STEINVALLO, et al. Laser radar: Historical prospective—from the East to the West [J]. *Optical Engineering*, 2016, 56(3):031220.
- [2] WU J, TAMURA Y, WANG Y, et al. Smartphone zombie detection from LiDAR point cloud for mobile robot safety [J]. *IEEE Robotics and Automation Letters*, 2020, DOI:10.1109/LRA.2020.2970570.
- [3] WEON I S, LEE S G, RYU J K. Object recognition based interpolation with 3D LIDAR and vision for autonomous driving of an intelligent vehicle[J]. *IEEE Access*, 2020, DOI:10.1109/access.2020.2982681.
- [4] LIU Q, LI H. Research on 3D point cloud model reconstruction method based on multi-kinects[C]. 2018 Eighth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC), IEEE, 2018:1805-1809.
- [5] 于方磊,赵硕,程坤,等.基于单光子探测的LiDAR三维点云数据处理方法设计[J]. *国外电子测量技术*, 2019, 38(2):10-14.
- [6] 李茂月,马康盛,王飞,等.基于结构光在机测量的叶片点云预处理方法研究[J]. *仪器仪表学报*, 2020, 41(8):55-66.
- [7] 谢小芳,万勇,严可馨,等.地下洞库三维激光扫点云区域分割和容积计算[J]. *电子测量与仪器学报*, 2019, 33(9):80-86.
- [8] 姚洪奎,颜雁军,陈凯,等.基于Zynq平台的便携式示波器Linux启动[J]. *电子测量技术*, 2020, 43(9):182-188.
- [9] 支萌辉,尹泉,吕松奎,等.基于ARM+FPGA的数字交流伺服驱动器设计[J]. *电气传动*, 2020, 50(9):35-41.
- [10] 冯志华,王华卓,安东博,等.基于BRAM的NVMe控制器原型仿真平台设计[J]. *计算机工程与设计*, 2021, 42(4):1181-1187.
- [11] 李振宇.基于PL-PS架构的图像处理系统的实现与算法应用[D]. 济南:山东大学,2016.
- [12] 肖潇.基于AXI的SoC互联结构的设计与验证[D]. 长沙:国防科学技术大学,2015.
- [13] 赵树新.基于Zynq-7000的星图识别系统的设计与实现[D]. 北京:北京工业大学,2016.
- [14] 王树青.基于Zynq平台的无人机信号处理嵌入式系统设计及实现[D]. 成都:电子科技大学,2019.
- [15] RAMAGOND S, YELLAMPALLI S, KANAGASABAPATHI C. A review and analysis of communication logic between PL and PS in ZYNQ AP SoC [C]. *International Conference on Smart Technologies for Smart Nation*, 2017:946-951.
- [16] 施赛烽,叶润川,林雪,等.基于FPGA和DDR3 SDRAM的高精度脉冲发生器设计与实现[J]. *合肥工业大学学报(自然科学版)*, 2021, 44(2):206-209,283.
- [17] 董艳雪,韩卫光.基于多核ARM的Linux操作系统的实时性研究[J]. *小型微型计算机系统*, 2017, 38(6):1262-1266.
- [18] 郭润.3 GHz 谱仪Linux系统的软件设计与实现[D]. 成都:电子科技大学,2016.
- [19] 田泉,艾丽蓉,陈杰. Linux实时内存的研究与实现[J]. *微电子学与计算机*, 2014, 31(8):45-48.
- [20] 黄婧.基于Linux无线视频采集传输系统的研究[D]. 北京:北京林业大学,2017.
- [21] 朱江波,赵志衡,刘洋,等.基于ZYNQ SoC的黑白色选系统设计[J]. *电子测量技术*, 2019, 42(10):52-57.
- [22] 刘汝卿,吴宇佳,李锋,等.激光雷达实时数据传输存储系统设计与实现[J]. *电子测量技术*, 2020, 43(20):16-20.

作者简介

陈小宇,工学博士,副教授,主要研究方向为嵌入式系统与应用、信号与信息处理等。

E-mail:chenxy@mail.ccnu.edu.cn

李常对,硕士研究生,主要研究方向为嵌入式系统与应用。

E-mail:3327306495@qq.com