

基于 bit-map 算法的 DMA 数据流仲裁器设计^{*}

朱亚琦,侯晓娟

(中北大学 仪器科学与动态测试教育部重点实验室,太原 030051)

摘要: 针对集成电路产业快速发展所需要的复杂通信要求,设计了一种基于 bit-map 算法的 DMA 数据流仲裁器。DMA 数据流仲裁器由通道优先级 bit-map 映射模块、通道请求生成模块、通道请求仲裁模块、通道授权标志生成模块和外设请求应答模块组成。在完成逻辑设计后,通过 UVM 验证方法学对电路的各个功能进行验证。验证结果表明,本设计仅使用一个 32 位 bit-map 映射寄存器将 8 条通道的 32 种优先级配置映射到对应有效位,便可实现对 8 条通道 DMA 请求的仲裁,无需配置优先级比较电路,不会因逐级比较产生优先级配置相互干扰的问题,且在后期生成实际电路的过程中可以减小电路面积、降低电路功耗。

关键词: bit-map;DMA;数据流仲裁器;UVM

中图分类号: TN402

文献标识码: A

Design of DMA Data Stream Arbiter Based on bit-map Algorithm

Zhu Yaqi, Hou Xiaojuan

(Key Laboratory of Instrumentation Science and Dynamic Measurement, Ministry of Education,
North University of China, Taiyuan 030051, China)

Abstract: Aiming at the complex communication requirements of the rapid development of integrated circuit industry, a DMA data stream arbiter based on bit-map algorithm is designed. DMA data flow arbiter is composed of channel priority bit-map mapping module, channel request generation module, channel request arbitration module, channel authorization flag generation module and peripheral request response module. After completing the logic design, each function of the circuit is verified by the UVM verification methodology. The verification results show that the design only uses a 32-bit bit-map register to map 32 kinds of priority configurations of 8 channels to the corresponding significant bits, so that the arbitration of DMA requests of 8 channels can be realized. There is no need to configure the priority comparison circuit, and the problem of mutual interference of priority configurations will not be generated due to the step by step comparison. In addition, the circuit area and power consumption can be reduced in the later process of generating the actual circuit.

Keywords: bit-map;DMA;data flow arbiter;UVM

0 引言

近年来,随着 IC 产业的快速发展,SoC 设计与应用越来越受到人们的关注^[1],在 SoC 系统中,DMA 传输作为一种高效的数据传递方式被大量采用^[2]。DMA 传输依靠 DMA 控制器实现,大多数 DMA 控制器基本由寄存器组、仲裁器、总线接口、传输控制逻辑等部分组成,其中仲裁器用于对同时输入的多路通道 DMA 请求进行仲裁,根据优先级排序进行 DMA 请求授权,输出获得授权的通道编号^[3-4]。

设计 DMA 仲裁器已经成为当前业界的工作重点之一,但现有设计的功能还不够完善。薛巨峰等设计了一种

基于 PCIe 总线的 DMA 控制器^[5],该 DMA 控制器的仲裁器为 4 路请求通道各配置了一个请求队列,采用轮询方式处理队列中的 DMA 请求,这种方式虽然实现逻辑较为简单,但是仅能对 4 路通道的 DMA 请求进行处理且无法根据通道优先级对请求进行仲裁;王胜等设计了一款可应用于 SoC 系统中的 DMA 控制器^[3],实现对 7 条不同通道的 DMA 请求根据优先级高低进行仲裁,且每个通道配置了 4 种不同的优先级^[6-7],但是,该设计的每一条通道只能接收一种外设请求源,因此至多只能对 7 种外设请求进行处理;郭志强设计了一种 DMA 数据流仲裁器^[8],可实现 8 条通道 DMA 请求的仲裁处理,且每条通道可支持最多 8 种优先级,但是其采用传统的逐级比较仲裁方式会耗费大量电路面积,实现起来极为复杂且增加电路功耗。以上设计均存在处理能力差、仲裁方式单一等问题,以及由这些

^{*} 基金项目:山西省应用基础研究计划资助项目(20210302123059);山西省高等学校科技创新项目资助(2020L0326)。



问题带来的数据传输效率低、电路功耗过高、电路面积过大等问题,严重影响了电路性能。

本文创新性地提出了一种基于 bit-map 算法的 DMA 数据流仲裁器设计,解决了上述设计中普遍存在的问题,具有更强的可操作性。该仲裁器配有 8 条 DMA 请求通道,每条通道可对 8 个外设请求源进行接收,且每条通道支持 4 种优先级配置,同时设计了一个基于 bit-map 算法的 32 位 pri 寄存器,将各通道配置的优先级映射到寄存器的每一位,通过将 32 种优先级配置映射到 pri 寄存器便可实现对各通道请求的仲裁,无需配置优先级比较电路,不仅可以在后期生成实际电路的过程中减小电路面积、降低电路功耗,还可以解决因逐级比较而产生的优先级配置相互干扰问题。

1 DMA 数据流仲裁器设计

1.1 总体设计

DMA 数据流仲裁器实现的基本功能是对各通道的 DMA 请求进行仲裁。本 DMA 数据流仲裁器配有 8 条 DMA 请求通道,每条通道均通过 DMA 配置寄存器内的 pri0~pri7 进行优先级配置,每条通道请求的优先级均有低、中、高、非常高(2'd0、2'd1、2'd2、2'd3)4 种选择,因此,8 条通道共有 32 种优先级排序。通过在仲裁器内部配置一个 32 位的 pri 寄存器,使用 bit-map 算法将 32 种优先级有效信号映射到该寄存器中。同时,每条请求通道均接入 8 个外设请求源,通过 DMA 配置寄存器内的 chsel0~chsel7 信号进行各通道外设请求源的选择,同一时刻只能选择一个外设占用总线进行 DMA 传输。因此,DMA 数据流仲裁器可对来自不同外设的 64 个外设请求源信号进行接收,并对它们进行仲裁处理。

本文所设计的 DMA 数据流仲裁器由通道优先级 bit-map 映射模块、通道请求生成模块、通道请求仲裁模块、通道授权标志生成模块、外设请求应答模块构成,具体的硬件架构框图如图 1 所示。

1.2 通道优先级 bit-map 映射模块

本文设计的仲裁器为每个通道配置了 4 种优先级,通过 pri0~pri7 位进行配置。在总线空闲时,若有多个通道发出 DMA 请求,则按照优先级进行授权,具有更高优先级的通道请求获得授权。若有相同优先级的通道请求,则编号较低的通道请求获得授权。因此,总共有 32 种可能的优先级排序。

根据仲裁规则,本模块内配置了一个 32 位的 bit-map 映射寄存器 pri,采用 bit-map 算法根据输入到模块的 pri0~pri7 通道优先级配置信号将各通道配置的优先级映射到 pri 寄存器上的对应位。pri 寄存器各有效位映射条

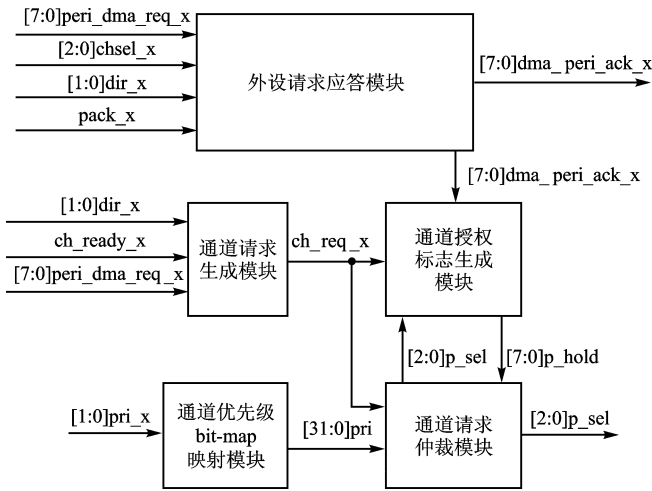


图 1 DMA 数据流仲裁器硬件架构框图

件表如表 1 所列,它反映了 pri 每一个有效位使能的条件。其中,pri 寄存器由高位到低位反映了总的优先级排序,位数越高,总优先级越高。pri 寄存器信号随后会被传输到通道请求仲裁模块参与仲裁。

表 1 pri 寄存器各有效位映射条件表

pri 寄存器的位	对应位的使能条件	pri 寄存器的位	对应位的使能条件
31	pri0=2'b11	15	pri0=2'b01
30	pri1=2'b11	14	pri1=2'b01
29	pri2=2'b11	13	pri2=2'b01
28	pri3=2'b11	12	pri3=2'b01
27	pri4=2'b11	11	pri4=2'b01
26	pri5=2'b11	10	pri5=2'b01
25	pri6=2'b11	9	pri6=2'b01
24	pri7=2'b11	8	pri7=2'b01
23	pri0=2'b10	7	pri0=2'b00
22	pri1=2'b10	6	pri1=2'b00
21	pri2=2'b10	5	pri2=2'b00
20	pri3=2'b10	4	pri3=2'b00
19	pri4=2'b10	3	pri4=2'b00
18	pri5=2'b10	2	pri5=2'b00
17	pri6=2'b10	1	pri6=2'b00
16	pri7=2'b10	0	pri7=2'b00

1.3 通道请求生成模块

首先,通道请求生成模块会接收来自于配置寄存器的通道使能信号 ch_ready0~ch_ready7,只有在通道使能信号有效时才能生成通道请求信号。随后,会读取配置寄存器提供的传输方式控制信号 dir0~dir7。

当配置为外设到存储器和存储器到外设的传输方式时,需要通过外设向 DMA 仲裁器发出通道 DMA 请求后

才能生成通道请求信号 $ch_req0 \sim ch_req7$ 。因此,当配置为以上两种传输方式时,模块接收外设提供的通道 DMA 请求信号 $peri_dma_req0 \sim peri_dma_req7$ 。若通道 DMA 请求信号的各位不全为 0,则表示此时至少有一个外设在通过对应通道向 DMA 仲裁器发出 DMA 请求,对应的通道请求信号被使能。通道请求信号 $ch_req0 \sim ch_req7$ 会被传输到通道请求仲裁模块参与仲裁并被传输到通道授权标志生成模块用于生成通道授权标志信号 p_hold 。

1.4 通道请求仲裁模块

通道请求仲裁模块会对输入的通道请求进行仲裁以产生仲裁信号 p_sel 并传递到各外设、存储器中,获得仲裁通道上的外设可以占用总线,参与 DMA 数据传输。

首先,本模块接收来自于通道授权标志生成模块的 8 位通道授权标志信号 p_hold ,当 p_hold 信号的 8 位都为 0 时,表示此时未对任何通道的 DMA 请求授权,没有 DMA 数据传输,可进行通道请求仲裁操作。随后接收来自通道优先级 bit-map 映射模块的 32 位 pri 寄存器映射信号,对 pri 寄存器的 32 位映射信号按照从高位到低位的顺序进行扫描,当出现有效高电平时,立即对该位所对应的 $ch_req0 \sim ch_req7$ 通道请求信号进行检测,若所对应的通道请求信号有效,则授权该通道的 DMA 通信请求,生成通道编号对应的 p_sel 值,并将 p_sel 值传递到各外设、存储器中。

1.5 通道授权标志生成模块

通道授权标志生成模块的主要功能是生成 8 位通道授权标志信号 p_hold 以传递到通道请求仲裁模块中, p_hold 信号的 8 位与 8 条通道对应,用于表示获得授权的通道编号。

首先当未授权任何通道的 DMA 请求时,没有设备占用总线进行 DMA 数据传输,本模块会对来自于通道请求生成模块的 $ch_req0 \sim ch_req7$ 通道请求信号进行轮询。若检测到某一通道的请求信号有效,则立即检测通道请求仲裁模块提供的 p_sel 仲裁信号,若 p_sel 仲裁信号的值与请求的通道编号一致,则使能 p_hold 信号的有效对应位,表明此时该通道的 DMA 请求被授权,通道上对应的外设可以占用总线进行 DMA 数据传输。随后被授权通道上的外设会通过向 DMA 控制器发送 $pack0 \sim pack7$ 外设通信请求信号来请求占用总线进行数据通信,作为应答,外设请求应答模块会生成 $dma_peri_ack0 \sim dma_peri_ack7$ 应答信号并输出到发出请求的各个外设中。本模块会持续对生成的 $dma_peri_ack0 \sim dma_peri_ack7$ 应答信号进行轮询检测,若检测到应答信号出现低电平下降沿,则 p_hold 信号的有效对应位被清 0,此时表示 DMA 控制器应答了被授权通道上外设的通信请求,外设已经开始占用

总线进行 DMA 传输。

1.6 外设请求应答模块

当通道 DMA 请求被授权后,对应通道上的外设会向 DMA 控制器发送外设通信请求信号 $pack0 \sim pack7$,当检测到外设通信请求信号时,外设请求应答模块会读取外设请求源选择信号 $chsel0 \sim chsel7$ 的值,并根据 $chsel0 \sim chsel7$ 的值生成 $dma_peri_ack0 \sim dma_peri_ack7$ 应答信号,由仲裁器发送到连接各通道的外设中。该应答信号的 8 个数据位对应于通道的 8 个外设请求源,根据 $chsel0 \sim chsel7$ 的值使能对应的有效数据位,被使能数据位所对应的外设随后会占用总线进行 DMA 传输。

此外,考虑到 DMA 控制器到外设的应答信号和外设到 DMA 控制器的请求信号涉及到的跨时钟域传输问题,本模块设计了一种握手处理方式对外设发出的 DMA 请求 $peri_dma_req0 \sim peri_dma_req7$ 和 DMA 控制器的应答 $dma_peri_ack0 \sim dma_peri_ack7$ 进行握手处理。当外设接收到来自于 DMA 控制器的应答信号 $dma_peri_ack0 \sim dma_peri_ack7$ 时,会立即撤销 DMA 请求信号 $peri_dma_req0 \sim peri_dma_req7$ 。随后,本模块检测到 DMA 请求信号 $peri_dma_req0 \sim peri_dma_req7$ 被撤销后,也会随即撤销 DMA 控制器的应答信号 $dma_peri_ack0 \sim dma_peri_ack7$ 。若传输方式为存储器到存储器的传输,则不会产生外设请求与 DMA 应答信号,本模块不参与通信。

2 仲裁功能验证

基于上述设计,使用 System Verilog 语言展开面向 DMA 数据流仲裁器的 UVM 验证平台设计和搭建,通过 UVM 验证方法学对设计的功能进行验证。在 Linux 环境下,编写 Makefile 脚本辅助验证,使用 Synopsys 公司的 VCS 软件进行功能仿真,并使用 Verdi 软件调试波形。

当多路优先级相同的通道同时发出 DMA 请求时,编号较低的通道获得仲裁授权的功能验证。验证过程大致如下:首先向配置寄存器写入各通道的优先级配置信号 $pri0 \sim pri7$ 。其中,通道 0~通道 5 的优先级均为高(2'd2),通道 6、通道 7 的优先级为中(2'd1)。bit-map 映射寄存器 pri 即将各通道配置的优先级映射到 32 个标志位中,此时 pri 的值为 3'2hfc 0300。最后每隔一段时间同时向仲裁器随机发出通道 0~通道 4 的 DMA 请求信号 $peri_dma_req0 \sim peri_dma_req4$,通道请求仲裁模块会持续对 pri 的每一位和对应的通道请求信号进行扫描。因为当 5 个通道拥有相同的优先级时,编号低通道的请求会被授权,所以会先授权通道 0 的 DMA 通信请求,此时 p_sel 为 3'd0。授权通道 0 的 DMA 通信请求后,仲裁器会等待通道 0 上的外设发出 $pack0$ 外设通信请求信号,当接收到该信号后,

通道请求仲裁模块随即回送 DMA 应答信号 dma_peri_ack0 到连接通道 0 的各外设,根据电路设定的握手通信规范,此时通道 DMA 请求信号 peri_dma_req0 的各位在接收到 dma_peri_ack0 信号后拉低为 0,随后 dma_peri_ack0 信号的各位也拉低为 0。之后依次授权通道 1~通道 4 的 DMA 通信请求并进行相应的 DMA 数据传输。优先级相同的通道同时请求时的总体验证波形图如图 2 所示。优先级相同的通道同时请求时的局部验证波形图如图 3 所示。

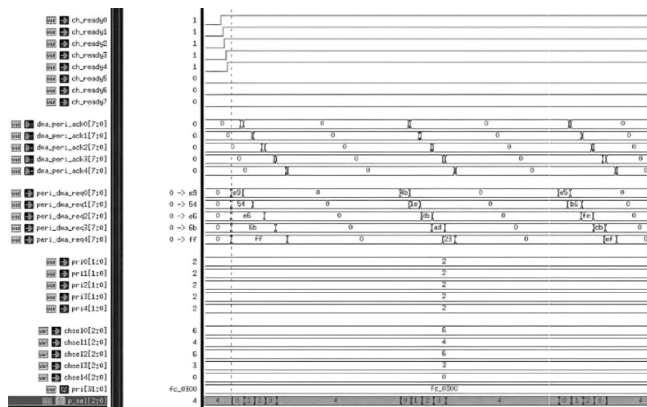


图 2 优先级相同的通道同时请求时的总体验证波形图

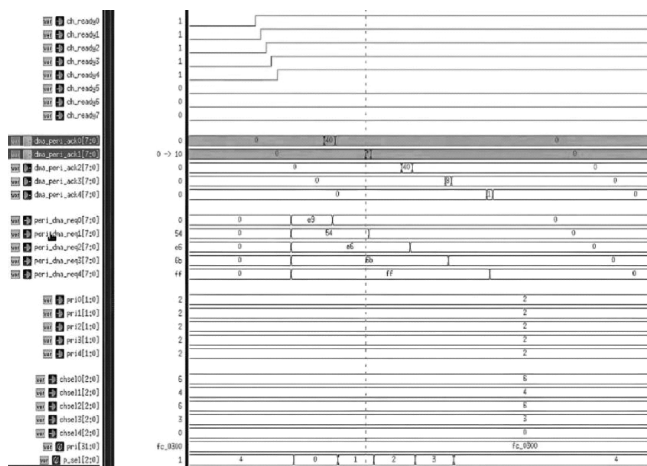


图 3 优先级相同的通道同时请求时的局部验证波形图

接着进行当多路通道同时发出 DMA 请求时高优先级通道获得仲裁授权的功能验证,与优先级相同的通道同时请求的验证过程类似。优先级不同的通道同时请求时的总体验证波形图如图 4 所示,优先级不同的通道同时请求时的局部验证波形图如图 5 所示。

3 结 语

本文详细介绍了一种基于 bit-map 算法的 DMA 数据流仲裁器设计,并通过 UVM 验证方法学在 Linux 环境下使用 VCS 和 Verdi 软件对设计的功能进行验证。该仲裁器由通道优先级 bit-map 映射模块、通道请求生成模块、

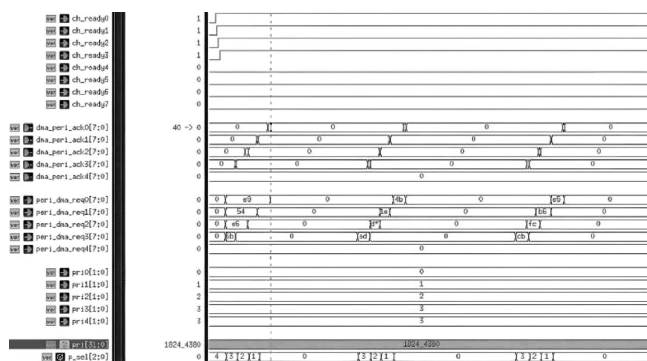


图 4 优先级不同的通道同时请求时的总体验证波形图

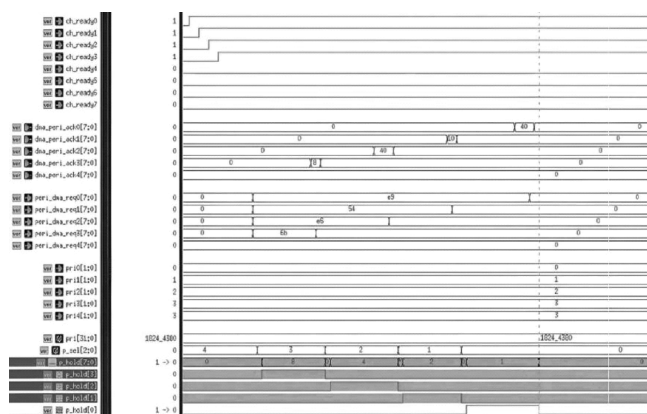


图 5 优先级不同的通道同时请求时的局部验证波形图

通道请求仲裁模块、通道授权标志生成模块、外设请求应答模块组成,配有 8 条 DMA 请求通道,每条通道可接收 8 个外设请求源的 DMA 通信请求,对来自不同外设的 64 个 DMA 通信请求信号进行仲裁处理。同时,使用了一个基于 bit-map 算法的 32 位 pri 寄存器将各通道配置的优先级映射到寄存器的每一位。相较于传统的 DMA 仲裁器,本 DMA 数据流仲裁器仅使用一个 32 位的 pri 寄存器,将 32 种优先级配置映射到寄存器内就可以完成通道请求的仲裁,无需按照传统仲裁方式配置优先级比较电路对各通道请求的优先级逐级比较。这样就不会在时序逻辑下因逐级比较而产生优先级配置相互干扰的问题,且因为无需配置专门的优先级比较电路,所以在后期生成实际电路的过程中可以显著减小电路面积、降低功耗。

参考文献

- [1] 张军,马琪.基于 AMBA 总线的 DMA 控制器设计[J].科技通报,2011,27(2):268-271.
- [2] 高旭栋.基于新型 AHB 总线架构的多通道 DMA 控制器设计[D].西安:西安电子科技大学,2021.
- [3] 王胜,史兴强,杨晓刚. SoC 系统中 DMA 控制器的设计与实现[J].电子与封装,2017,17(8):25-28.

双方均可正常接收到对方的数据,且与发送数据保持一致,由此验证了透传设备在调试模式下可正常工作。



图8 网口与RS-422和LVDS数据透传调试图

4.2 性能测试

在实际情况下,PC机与某地检设备的通信机制是PC机给地检设备发送一包数据后,当地检设备接收到PC机所发送的数据时,需要返回一个响应信号,PC机才会发送下一包网络数据。为了测试方便,在地检设备中做循环处理,利用网络助手发送数据到数据透传设备,再经过RS-422/LVDS串口发送给地检设备,然后地检设备将接收的数据原封不动地发送给数据透传设备,最后再返回到网络助手。通过比较网口发送字节和网口接收字节来判断设备的数据透传性能,图9为网口与RS-422和LVDS数据透传测试图,测试环境均在串口波特率为2 Mb/s、每包数据220字节的情况下,分别透传10 780包和11 000包数据,对比可知,发送和接收数据包没有出现丢包和误码的情况。



图9 网口与RS-422和LVDS数据透传测试图

- 77
- [4] 陈卓,杨爱良,王骥.基于PLB总线的多通道SGDMA设计[J].航空电子技术,2009,40(1):4.
- [5] 薛巨峰,黄爱娟.基于PCIe总线的多路复用DMA高速传输系统的设计[J].电子技术,2015,44(5):87-91.
- [6] 金志成.高带宽存储器的模型建立与优先级调度研究[D].成都:电子科技大学,2020.
- [7] 王佳萌.基于UVM的直接存储器读写控制器的验证平台设计[D].西安:西安电子科技大学,2020.

5 结 语

该数据透传设备已经实际应用于上位机对某地检设备的数据加载,经实际应用证明,该数据透传设备的RS-422串口和LVDS串口在2 Mb/s以内任意速率下的数据透传功能具有较高的稳定性,并且该透传设备不仅适用于某地检设备,还适用于所有支持RS-422和LVDS接口的串口设备,为网口数据与串口数据的透传提供了一种解决方案。

参考文献

- [1] 邹传智.基于多串口转以太网的数据采集及分析系统设计[D].上海:东华大学,2017.
- [2] 王庆贺,吴超,唐顺晨,等.基于DSP的以太网-RS422通信转换器的设计与实现[J].电子质量,2020(1):26-29,35.
- [3] 胥泽飞.基于Zynq的智能相机硬件设计[D].南京:南京邮电大学,2018.
- [4] 李金科,秦涛,吴爱平.基于ZYNQ的可配置高速数据传输系统设计[J].信息技术,2023,47(1):1-8.
- [5] 陈晓锦,刘跃成,刘慧芳,等.高波特率网口与串口的转换模块设计[J].单片机与嵌入式系统应用,2022,22(8):75-78,82.
- [6] 赖川,邓蕾,张显.基于STM32F407的FPGA动态加载设计[J].通信技术,2022,55(3):386-393.
- [7] Xilinx Corporation. Zynq-7000 SoC Technical Reference Manual (UG585),2018.
- [8] 李建翔,王洪凯,杨玉华,等.基于RS422+LVDS高速长线通信的设计与实现[J].仪表技术与传感器,2021(3):40-44,96.
- [9] 江耘宇.一种组合导航计算硬件平台的设计与实现[D].重庆:重庆大学,2019.
- [10] 李健一,闫强,李晓蕊.支持数据缓存的LVDS-以太网转接板设计[J].电脑编程技巧与维护,2023(1):82-84.

许向阳(副教授),主要研究方向为电子信息;张宇(硕士研究生),主要研究方向为信息处理;毕胜(硕士研究生),主要研究方向为强化学习;王晓君(教授),主要研究方向为智能信息处理。通信作者:张宇,1123339765@qq.com。

(责任编辑:薛士然 收稿日期:2023-04-27)

- [8] 郭志强.基于AMBA总线协议的可编程DMA控制器的设计与实现[D].长沙:湖南大学,2020.

朱亚琦,主要研究方向为数字电路设计与片上系统、硬件电路设计;侯晓娟(副教授),主要研究方向为能量收集和功能材料。通信作者:侯晓娟,houxiaojuan@nuc.edu.cn。

(责任编辑:薛士然 收稿日期:2023-04-21)