面向 100 Gbps 网络应用的 RISC-V CPU 设计与实现

李晓霖^{1,2)}, 韩萌³⁾, 郝凯⁴⁾, 薛海韵⁵⁾, 卢圣健^{1,2)}, 张昆明¹⁾, 祁楠⁴⁾, 牛星茂⁵⁾, 肖利民³⁾, 郝沁汾^{1,2)*}

¹⁾ (中国科学院计算技术研究所高通量计算机研究中心 北京 100190)

²⁾ (中国科学院大学计算机科学与技术学院 北京 100049)

³⁾ (北京航空航天大学计算机科学与工程学院 北京 100191)

⁴⁾ (中国科学院半导体研究所超晶格国家重点实验室 北京 100083)

⁵⁾ (中国科学院微电子研究所系统封装与集成研发中心 北京 100029)

(haoqinfen@ict.ac.cn)

摘 要: RISC-V 作为新一代开源精简指令集, 具有功耗低、面积小和性能高的优点, 因此基于 RISC-V 架构的技术和产品发展迅速。然而, 目前基于 RISC-V 架构的中高端 64 位 CPU 设计实例很少, 也缺乏相应的商用 IP, 尤其是在面向高速网络应用方面。因此, 首先改进了开源的 64 位 U500 RISC-V SoC, 包括增加了总线宽度和集成二级缓存等; 然后在 CPU 中实现了完整的 100 Gbps 以太网功能模块, 包括介质访问控制子层、物理编码子层和串行器/解串器以及用于该功能模块的发送缓冲区和接收缓冲区; 最后通过前端仿真、FPGA 验证以及启动 Linux 操作系统, 验证了所设计的 64 位 RISC-V CPU 以及 100 Gbps 以太网功能模块的正确性和有效性。所设计的 RISC-V CPU 和 100 Gbps 以太网功能模块可应用于智能网卡等数据中心应用场景。

关键词: RISC-V; 片上系统; 100 Gbps 以太网; 介质访问控制子层; 物理编码子层; 串行器/解串器; 智能网卡

中图法分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2021.18538

Design of RISC-V CPU for 100 Gbps Network Application

Li Xiaolin^{1,2)}, Han Meng³⁾, Hao Kai⁴⁾, Xue Haiyun⁵⁾, Lu Shengjian^{1,2)}, Zhang Kunming¹⁾, Qi Nan⁴⁾, Niu Xingmao⁵⁾, Xiao Limin³⁾, and Hao Qinfen^{1,2)*}

¹⁾ (High-Throughput Computer Research Center, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

²⁾ (School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049)

³⁾ (School of Computer Science and Engineering, Beihang University, Beijing 100191)

⁴⁾ (State Key Laboratory for Superlattices and Microstructures, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083)

⁵⁾ (SiP Center, Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029)

Abstract: As a new open-source reduced instruction set architecture, RISC-V has advantages of low power consumption, small area and high performance. Therefore, technology and products based on RISC-V are developing rapidly. However, currently there are few medium and high end 64 bit CPU design instances based on the RISC-V architecture, and it is also hard to find corresponding commercial IP, especially for high-speed network applications. In this paper, we firstly improved the open source 64 bit U500 RISC-V SoC by extending the Bus width and adding L2 Cache, and etc. Secondly we implemented a complete 100 Gbps Ethernet function, which includes MAC, PCS, and SerDes, and the TX buffer and RX buffer used for the function. Finally

收稿日期: 2020-08-05; 修回日期: 2021-04-30. 基金项目: 国家重点研发计划(2019YFB2203004); 北京市科技计划(Z191100004819006). 李晓霖(1997—), 男, 硕士研究生, 主要研究方向为计算机体系结构; 韩萌(1997—), 男, 博士研究生, 主要研究方向为计算机体系结构; 郝凯(1996—), 男, 硕士研究生, 主要研究方向为高速电路; 薛海韵(1984—), 女, 博士, 副研究员, 主要研究方向为高速集成电路; 卢圣健(1996—), 男, 硕士研究生, 主要研究方向为计算机体系结构; 张昆明(1995—), 男, 硕士, 助理工程师, 主要研究方向为计算机体系结构; 祁楠(1983—), 男, 博士, 研究员, 博士生导师, 主要研究方向为高速电路; 牛星茂(1997—), 男, 硕士研究生, 主要研究方向为高速集成电路; 肖利民(1970—), 男, 博士, 教授, 博士生导师, 主要研究方向为计算机体系结构; 郝沁汾(1969—), 男, 博士, 研究员, 博士生导师, CCF 高级会员, 论文通讯作者, 主要研究方向为计算机体系结构。

we proved correctness and effectiveness of entire 64 bit RISC-V CPU design and 100 Gbps Ethernet function by simulation, FPGA verification and boot of Linux operation system. The designed RISC-V CPU and 100 Gbps Ethernet function can be applied to data center application such as smart network interface cards.

Key words: RISC-V; system on a chip (SoC); 100 Gbps Ethernet; medium access control (MAC); physical coding sublayer (PCS); serializer/deserializer (SerDes); smart network interface cards (NIC)

RISC-V 指令集架构是 2010 年由美国加州大学伯克利分校 David Patterson 教授等提出的第 5 代开源精简指令集, 以精简、高效、低能耗、模块化、可扩展、免费开放和无历史指令负担等为设计目标, 在功耗、面积和性能等方面的表现都不亚于甚至优于 ARM (advanced RISC machine) 指令集架构。由于 RISC-V 指令集架构巨大的优势, 基于 RISC-V 的产品和技术发展迅速^[1]。2019 年 3 月, SiFive 推出了开源 32 位 RISC-V 片上系统(system on a chip, SoC)处理器 FE310, 专为微控制器、物联网和可穿戴应用而设计^[2]。阿里巴巴旗下平头哥半导体有限公司也在同年 7 月发布了能够实现每周期 2 条内存访问指令的 64 位 RISC-V 处理器——玄铁 910, 可以用于设计制造高性能端侧芯片, 应用于人工智能以及自动驾驶等领域。尽管如此, 目前 RISC-V 的发展仍然处于初级阶段, 业界大部分的 RISC-V 中央处理器(central processing unit, CPU)都是面向微控制器和物联网等设计用途。同时, RISC-V 指令集架构的 IP 生态匮乏, 特别是缺少面向高速网络应用的 IP 设计参考, 这种情况严重影响了 RISC-V 在中高端 CPU 和数据中心领域的应用。

面向 100 Gbps 以太网应用, 本文设计并实现了介质访问控制层(media access control, MAC)、物理编码子层(physical coding sublayer, PCS)和串行器/解串器(serializer/deserializer, SerDes)等模块, 同时基于开源的 U500 RISC-V SoC 进行改进, 最终完成了一个完整的 CPU 设计。通过前端仿真、现场可编程逻辑门阵列(field programmable gate array, FPGA)验证和启动 Linux 操作系统验证了其有效性, 实现了一个面向高速网络应用的 RISC-V CPU 原型。根据调研, 本文工作应为第 1 个可以应用于高速网络场景的 64 位 RISC-V CPU 设计实现。

随着云计算的发展, 在数据中心中出现越来越多种类的网络协议, 如开源虚拟交换标准(open virtual switch, OVS)和远程直接内存访问(remote direct memory access, RDMA)协议等, 但由于传统数

据中心网卡处理芯片采用专用集成电路(application specific integrated circuit, ASIC)方式设计, 造成其无法适应网络协议的动态发展, 从而使宝贵的主机 CPU 资源被浪费用于处理这些网络协议。如 Mellanox 公司的 CX 系列网卡主要实现对以太网协议和 RDMA 协议等特定协议的支持。虽然有着较小的处理延时, 但对新的网络协议则很难扩展支持。因此, 网卡芯片 CPU 化以提升网卡处理网络协议的灵活性是当前一个技术趋势。网络设备公司已经意识到了这一点并做出了改变, 典型的例子有 Mellanox 公司刚发布的 Bluefield 智能网卡系列中加入了 ARM CPU 以实现对网络协议支持的灵活性。本文工作的最终目的是希望研究和开发原生基于 CPU 架构的智能网卡核心芯片, 并利用 RISC-V CPU 易于扩展的特点, 以指令扩展的方式支持各种网络协议, 在保证处理性能的前提下提升网卡芯片处理不同网络协议的灵活性。

1 工作介绍

由于 RISC-V 开源指令集具有清晰简洁的规范、灵活的扩展能力, 已成为当下芯片技术研究的热点。得益于谷歌、高通、BlueSpec 等科技公司加入 RISC-V 基金会, 基于 RISC-V 的产品和技术发展很快^[1]。目前国内外具有代表性的 RISC-V 芯片设计实例如表 1 所示。可以看到, RISC-V 目前的设计实现主要集中在微控制器和物联网、人工智能等应用场景。同时, 由于尚处于发展的初期, RISC-V

表 1 国内外具有代表性的 RISC-V 芯片实例

芯片	数据位宽/bit	公司	应用领域
玄铁 910	64	平头哥	5G, AI
GAP-8 ^[3]	32	GreenWaves	IoT, AI
Pygmy	64	睿思芯科	IoT, AI
Falcon ^[4]	64	NVIDIA	MCU
FE310 ^[2]	32	SiFive	MCU, IoT
SweRV ^[5]	32	西部数据	MCU, IoT
GD32VF103 ^[6]	32	兆易创新	MCU

阵营 IP 生态匮乏, 几乎找不到面向高速网络应用的 IP 参考, 即便可以找到, 其昂贵的价格也令人难以接受. 这种现状阻碍了 RISC-V 在中高端 CPU 领域尤其是数据中心的应用和进一步发展.

2 64 位 SoC 改进设计

U500 RISC-V SoC 是一种支持标准 RV64GC 指令集的 64 位开源 RISC-V 实现^[7], 如图 1 所示, 灰色部分为本文设计实现或改进的部分. U500 有 4 个 64 位具有缓存一致性的 RISC-V CPU 核, 每个核都具有一条 64 位高性能有序单发射执行流水线,

独享一个 16 KB 的一级指令缓存(L1 ICache)和一级数据缓存(L1 DCache). U500 内部各个模块使用 TileLink 总线相连, 其中 4 个 RISC-V CPU 核模块、平台级中断控制器模块(platform-level interrupt controller, PLIC)、核部分中断控制器模块(core-local interrupt controller, CLINT)、引导 ROM 模块(Boot ROM)、发送缓冲区(TX Buffer)和接收缓冲区(RX Buffer)等连接在系统总线(Sbus)上. 此外, 系统总线上还连接着 100 Gbps 以太网功能模块、内存总线(Mbus)和外设总线(Pbus). 其中外设总线上连接着 SD 卡(SD Card)、串口(UART)、通用输入输出(GPIO)和闪存(FLASH)等外部设备.

图 1 RISC-V CPU 设计结构图

U500 RISC-V SoC 的系统总线宽度为 64 位, 在 1 GHz 的主频下, 即使系统总线的资源全部只用于发送或接收数据, 最高带宽也只能达到 64 Gbps, 并不能满足 100 Gbps 以太网的要求. 相比于提高系统总线的频率, 扩大系统总线的宽度显然更容易实现. 因此, 本文将系统总线到 RISC-V CPU 核、直接内存访问控制器(direct memory access, DMA)、发送缓冲区和接收缓冲区的数据宽度从原来的 64 位增加到 256 位. 同样在系统总线 1 GHz 的频率下, 即使只有一半的系统总线资源用于发送或接收数据, 最高带宽也能够达到 128 Gbps, 因此能够满足 100 Gbps 以太网的要求.

由于双倍速率同步动态随机存储器(double data rate dynamic random access memory, DDR)的 IP 昂贵, 在 SoC 设计中 CPU 使用 32 MB 同步动态随机存储器(synchronous DRAM, SDRAM)作为系统主存, SDRAM 通过 SDRAM 控制器(SDRAM Controller)连接在二级缓存(L2 Cache)之后.

SDRAM 控制器内部实现状态机以完成 SDRAM 的预充电、刷新和读写操作. 经过测试, 其随机写的带宽是 134 MB/s, 随机读的带宽是 126 MB/s. 同时, 在 SDRAM 控制器中实现了可调移相逻辑, 以适应印制电路板上不确定的延时, 保证了 SDRAM 总线上数据的可靠传输. 由于 SDRAM 读写数据较慢, SDRAM 控制器平均 971 个 CPU 时钟周期才能读出一个 64 B 的缓存块. 因此, 在 U500 RISC-V SoC 上集成了一个 4 核共享的 128 KB 二级缓存以减小访存延时. 由于流片费用基本与芯片面积成正比, 而构成高速缓存的静态随机存储器(static random-access memory, SRAM)又具有较大的面积, 因此集成较大的二级缓存会增加大量的流片费用. 出于成本的考虑, 本文 CPU 只集成了 128 KB 的二级缓存. 后续的商用版本将集成更大容量的二级缓存, 并且使用 DDR 作为芯片的系统内存, 以进一步提高访存性能. 一级缓存与二级缓存采用 non-inclusive^[8]机制进行管理, 并通过一致性管理

模块维护一致性. 二级缓存为8路组相连, 采用最近最少使用替换策略.

为了使100 Gbps以太网功能模块的发送和接收通道可以独立地访问内存, 同时也不会受其他的访存操作干扰, 本文CPU实现了2个单独的缓冲区: 发送缓冲区和接收缓冲区, 并将其连接在系统总线上. 为了进一步提高100 Gbps以太网功能模块的数据传输效率, 在介质访问控制层内部实现了直接内存访问控制器, 并且将其连接到系统总线上, 以在发送/接收缓冲区和介质访问控制层之间搬运数据.

3 MAC设计与实现

3.1 介绍

MAC实现了数据收发控制和媒体访问管理功能, 是开放式系统互联(open system interconnection, OSI)通信参考模型中数据链路层的主要组成部分. MAC采用全双工数据通路设计, 使用256位CGMII接口与物理层对接.

3.2 设计考虑

为充分利用100 Gbps物理链路提供的高速数据传输能力, MAC中引入直接内存访问控制机制以提高数据链路层的数据收发能力. 如图1所示, MAC划分为DMA和媒体控制器2部分. DMA访问数据缓冲区进行数据搬运; 媒体控制器进行MAC帧数据的封装, 校验并向物理层提供统一的数据接口.

由于U500 RISC-V SoC的系统总线使用的是TileLink协议, 因此DMA通过TileLink总线接口连接到系统总线, 采用独立的发送通道和接收通道, 它们分别处理数据包的发送与接收, 彼此之间互不干扰, 以实现全双工工作. 同时, RISC-V CPU核与DMA采用中断机制进行通信, 使用不同的发送中断和接收中断, 以减少处理中断的时间. DMA与媒体控制器之间的数据传递使用AXI Stream总线协议. 该总线协议去除了地址项, 允许无限制突发传输规模, 适合用于大量数据传输的场景.

3.3 DMA设计与实现

DMA描述符分为发送描述符和接收描述符. 描述符以循环链表的形式存放在内存中. 图2所示为DMA发送及接收数据包的流程, 该流程步骤如下.

Step1. RISC-V CPU核将发送/接收描述符、数据包写入内存.

Step2. RISC-V CPU核配置DMA.

Step3. DMA从内存中取出发送/接收描述符.

Step4. 若为发送流程, DMA解析描述符后从发送缓冲区中取出数据包, 发送给媒体控制器.

Step5. 若为接收流程, DMA接收从媒体控制器传递的数据包, 解析描述符后将数据包写入接收缓冲区.

Step6. DMA更新发送描述符状态.

Step7. 结束发送或接收数据包流程后, 产生对应的中断.

图2 DMA发送及接收数据包的流程

3.4 媒体控制器设计与实现

MAC帧是数据链路层的协议数据单元, 由前导码、定界符、目的地址、源地址、长度/类型、数据、填充数据和校验码组成. 媒体控制器的设计如图3所示. 媒体控制器对数据包进行计数、填充、CRC32校验码计算等工作, 最终将数据转换为CGMII接口格式并发送至物理层. 当物理层向媒体控制器传递数据包时, 媒体控制器接收端进行数据校验, 若该数据包无误, 在删除前导码和填充数据后, 将数据包传递至DMA. 在AXI Stream接口与媒体控制器发送端和接收端之间, 分别使用异步的发送队列和接收队列实现数据缓存与解决跨时钟域问题.

图3 媒体控制器

3.5 MAC回环测试

为测试MAC数据处理速度以及纠错能力, 在CGMII接口端对数据进行回环控制, 以屏蔽物理层对数据传输的影响. 回环测试中, RISC-V CPU核产生随机的数据包, 并在其中插入异常的数据

包,从数据帧写入发送缓冲区开始计算传输延迟,到该数据包被接收至接收缓冲区结束计时.经过测试,结果如图 4 所示.数据在 MAC 的平均停留时间为 127 ns,且可以正确地识别错误的数据包.

图 4 MAC 层回环测试结果

4 物理层设计与实现

许多通信协议都包括物理层.由于经济成本原因,工业界在许多协议(如 PCIe 和以太网)之间都共享物理层的实现.

如图 5 所示,典型的通信协议堆栈的物理层可以分为 PCS、物理介质附加子层(physical medium attachment, PMA)和物理介质相关子层(physical medium dependent, PMD)^[9].如果物理介质是光介质,则光收发器就实现了 PMD.为了同时兼顾多个应用场景(一种场景是 MAC 与物理层在主板上相距较远),

本文单独设计了 PMA,即 SerDes,并采用了 40 nm CMOS(complementary metal-oxide-semiconductor)工艺进行流片验证.PCS 由于要与数据链路层建立许多低速连接,因此选择将 PCS 与 MAC 集成在一起.

图 5 开放式系统互联通信参考模型

4.1 PCS 设计与实现

PCS 有发送端和接收端 2 部分.发送端对数据链路层中的数据进行编码和扰码,以方便 SerDes 传输数据.由于每个物理通道中的数据到达时间不同,因此还需要接收端同步数据. PCS 的内部结构如图 6 所示.

图 6 PCS 内部结构

为了提高性能,PCS 使用了几种设计.首先,接收端通过在每个通道中使用并行检测和同步队列来区分 2 位同步头的所有可能位置,以消除时间偏差.然后,通过实现一个 I/O 缓冲区动态调整空闲字符,以 16383 个块的间隔插入对齐标记后还能够保持数据速率.最后,接收端实现了检测模块,可以更早地检测到错误,因此接收端可以减少花在错误处理上的时间.

由于 6.25 Gbps 接近单端信号数据速率的限制,

同时考虑 SerDes 技术的经济性,在 PCS 和 SerDes 之间选择实现 16 路 6.25 Gbps 的数据传输速率.如果应用程序需要低延迟,则可以选择使用并行单端信号来实现 6.25 Gbps 的数据传输速率,否则使用数据传输速率为 6.25 Gbps 的 SerDes.

仿真结果表明,发送端和接收端分别可以在 9 个时钟周期(23.04 ns)和 23 个时钟周期(58.88 ns)内以 100 Gbps 的速度处理数据. PCS 的功耗为 42.6954 mW.

4.2 SerDes 设计与实现

SerDes 在发送端中将 16 路 6.25 Gbps 数据串行化为 4 路 25 Gbps 数据, 而在接收端中又将这 4 路 25 Gbps 数据转换成为 16 路 6.25 Gbps 数据. 图 7 显示了 SerDes 发送端和接收端通道中的所有模块. 考虑信道损耗和延迟不匹配, 发送端需要使用时钟与数据恢复, 以消除码间干扰引起的抖动并及时调整所有输入. 使用 3-tap 前向均衡驱动器的电压型驱动器用于全速数据传输. 除了相反的数据流外, 接收端的模拟前端电路的实现也与发送端的不同. 为了补偿信道损耗和固有带宽, 接收端的模拟前端电路采用串联电感峰值作为输入端, 并采用连续时间线性均衡器进行可调的均衡.

图 7 SerDes 原理图

仿真结果表明, 以每通道 25 Gbps 全速率输入时, 接收端的模拟前端电路在奈奎斯特频率下能提供 10 dB 的均衡. 每个传输通道的发送端和接收端的功耗之和为 320 mW.

5 CPU 的仿真与验证

图 8 所示为 CPU 的后端设计布局布线图, 绿

色框内为系统总线. 在未集成 SerDes 时, 总面积为 6.70 mm^2 , 总动态功耗为 1.3 W, 总静态功耗为 300 mW. 同时, 综合后 CPU 的主频能达到 1 GHz, 通过 FPGA 启动 Linux 操作系统, 并运行 Dhrystone 测试程序和 CoreMark 多核测试程序, 验证设计的正确性. Linux 操作系统完成启动界面如图 9 所示.

图 8 后端设计布局布线图

图 9 Linux 操作系统启动界面

性能方面, 通过运行单核 Dhrystone 和 4 核 CoreMark 程序进行性能测试. 所设计的 CPU 在运行单核 Dhrystone 测试程序时, 分数达到了 2.82 DMIPS/MHz, 超过了 ARM 的 Cortex A9 处理器, 介于 Cortex A9 处理器和 Cortex A12 处理器之间. 同时, 4 核 CoreMark 测试程序的分数为 21.6 CoreMark/MHz, 超过了 ARM 的 Cortex A9 处理器和 Cortex A15 处理器^[10]. 表 2 和表 3 分别为设计的 CPU 与其他 CPU 在单核 Dhrystone 和 4 核 CoreMark 测试程序上的性能比较.

表 2 Dhrystone 测试程序性能比较

CPU	频率/MHz	性能分数/(DMIPS·MHz ⁻¹)
A8 ^①	1 000	2.00
A9 ^①	1 200	2.50
本文	1 000	2.82
A12 ^①	1 200	3.50

表 3 CoreMark 4 核测试程序性能比较

CPU	频率/MHz	性能分数/(CoreMark·MHz ⁻¹)
A7 ^[10]	1 500	10.27
A9 ^[10]	1 400	15.89
A15 ^[10]	2 000	15.08
本文	1 000	21.60

6 结 语

本文面向 100 Gbps 网络应用的 RISC-V CPU, 实现了包括 4 个 RISC-V CPU 核、总线、二级缓存、SDRAM 控制器, 以及由 MAC, PCS 和 SerDes 组成的 100 Gbps 以太网功能模块. 所设计的 CPU 单核 Dhrystone 测试程序的分数为 2.82 DMIPS/MHz, 超过了 ARM 的 Cortex A9 处理器. 同时 4 核 CoreMark 测试程序的分数为 21.6 CoreMark/MHz, 超过了 ARM 的 Cortex A15 处理器. PCS 的发送端和接收端可以分别在 23.04 ns 和 58.88 ns 内以 100 Gbps 的速度处理数据, 并且以每通道 25 Gbps 全速率输入时, SerDes 接收端的模拟前端电路在奈奎斯特频率下能提供 10 dB 的均衡.

本文设计的 CPU 可以应用于既需要 100 Gbps 以太网支持同时又需要 CPU 的灵活性的应用场合, 如智能网卡. 未来的工作包括将继续提升 RISC-V CPU 核的性能、增加核的个数, 将 RISC-V CPU 与串行器/解串器集成等, 以推动 RISC-V 在中高端场景应用.

参考文献(References):

- [1] China RISC-V Alliance. Report of open ISA and open source chip[R]. Beijing: China RISC-V Alliance, 2019(in Chinese) (中国开放指令生态 (RISC-V) 联盟. 开放指令集与开源芯片发展报告[R]. 北京: 中国开放指令生态(RISC-V)联盟, 2019)
- [2] SiFive Inc. SiFive FE310-G000 Manual[OL]. [2020-08-05]. <https://static.dev.sifive.com/SiFive-E310-G000-manual-v1.0.1.pdf>
- [3] Flamand E, Rossi D, Conti F, *et al.* GAP-8: A RISC-V SoC for AI at the edge of the IoT[C] //Proceedings of the 29th International Conference on Application-specific Systems, Architectures and Processors. Los Alamitos: IEEE Computer Society Press, 2018: 1-4
- [4] NVIDIA Inc. RISC-V in NVIDIA[OL]. [2020-08-05]. <https://riscv.org/wp-content/uploads/2017/05/Tue1345pm-NVIDIA-Sijstermans.pdf>
- [5] Western Digital Corporation. RISC-V SweRV EH1 programmer's reference manual[OL]. [2020-08-05]. https://github.com/chipsalliance/Cores-SweRV/blob/master/docs/RISC-V_SweRV_EH1_PRM.pdf
- [6] GigaDevice Semiconductor Inc. GD32VF103 RISC-V 32-bit MCU[OL]. [2020-08-05]. http://www.gd32mcu.com/data/documents/shujushouce/GD32VF103_Datasheet_Rev%201.1.pdf
- [7] SiFive Inc. SiFive freedom U500 platform[OL]. [2020-08-05]. <https://static.dev.sifive.com/SiFive-Freedom-U500-datasheet-v1.0.pdf>
- [8] Backes L, Jiménez D A. The impact of cache inclusion policies on cache management techniques[C] //Proceedings of the International Symposium on Memory Systems. New York: ACM Press, 2019: 428-438
- [9] IEEE Computer Society. IEEE standard for information technology - local and metropolitan area networks - specific requirements - part 3: CSMA/CD access method and physical layer specifications amendment 4: media access control parameters, physical layers, and management parameters for 40 Gb/s and 100 Gb/s operation[OL]. [2020-08-05]. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4299496>
- [10] EEMBC. CoreMark score[OL]. [2020-08-05]. <https://www.eembc.org/coremark/scores.php>

① https://blog.csdn.net/qq_21475601/article/details/106564903