DOI:10.19651/j.cnki.emt.2107655

可重构高速数据加密系统设计和实现*

王 凯[1,2,3] 刘 凯[1,2,3] 李 拓[1,2,3] 符云越[2,3] 刘 唐[2,3] 王 骞[2,3]
(1.山东海量信息技术研究院 济南 250098;2.浪潮电子信息产业股份有限公司 高效能服务器和存储技术国家重点实验室 济南 250101;3.山东浪潮人工智能研究院有限公司 济南 250101)

摘 要:为解决 SM4 传统加解密方式存在的速度慢、效率低、占用 CPU 计算资源的问题,提出了一种可重构高速数据加密系统。该系统基于 Xilinx Virtex UltraScale VU9p FPGA,利用 PCIe 热插拔特性,可快速应用于办公主机或服务器,通过 PCIe 高速接口实现数据的快速传输,在 FPGA 内实现并行可调度 SM4 算法逻辑,设计有专用 DMA 模块,实现旁路主机 CPU 传输明文密文,减少主机端资源占用;采用 FPGA 实现的加解密系统具备可重构性,大大降低了算法迭代的硬件成本。系统分析测试和实验结果表明,该系统实现了数据的高速可靠传输与加密,总线速率达到 8 GT/s,能有效满足大容量数据快速加解密的需求;采用并行可调度流水线加解密,较 CPU 实现方式,加解密速率提升约 25.78 倍。

关键词:PCIe 高速总线;SM4 加解密;直接存储器访问;高速数据传输;现场可编程门阵列(FPGA)
中图分类号:TP303;TP309;TP274 **文献标识码**:A **国家标准学科分类代码**:510.4;520.6

Design and implementation of reconfigurable high-speed data encryption system

Wang Kai[1,2,3] Liu Kai[1,2,3] Li Tuo[1,2,3] Fu Yunyuc[2,3] Liu Tang[2,3] Wang Qian[2,3]
(1. Shandong Massive Information Technology Research Institute, Jinan 250098, China; 2. State Key Laboratory of High-end & Storage Technology, Inspur Electronic Information Industry Co., Ltd., Jinan 250101, China; 3. Shandong Inspur Artificial Intelligence Research Institute Co., Ltd., Jinan 250101, China)

Abstract: In order to solve the problems of slow speed, low efficiency, and CPU computing resources in the traditional SM4 encryption and decryption methods, a reconfigurable high-speed data encryption system is proposed. The system is based on Xilinx Virtex UltraScale VU9p FPGA, using PCIe hot-swappable features, can be quickly applied to office hosts or servers, fast data transmission through PCIe high-speed interface, parallel and schedulable SM4 algorithm logic in FPGA, and a dedicated DMA design the module realizes bypassing the host CPU to transmit plaintext ciphertext, reducing the resource occupation on the host side; the encryption and decryption system implemented by FPGA is reconfigurable, which greatly reduces the hardware cost of algorithm iteration. System analysis, testing and experimental results show that the system achieves high-speed and reliable data transmission and encryption, and the bus rate reaches 8 GT/s, which can effectively meet the needs of fast encryption and decryption of large-capacity data; it adopts parallel schedulable pipeline encryption and decryption, which is better than traditional software. In this way, the encryption and decryption rate is increased by approximately 25.78 times.

Keywords: PCIe high-speed bus; SM4 encryption and decryption; direct memory access; high-speed data transmission; field programmable gate array (FPGA)

0 引 言

数据加密是保证信息安全的重要手段之一。SM4 算法具有安全性强、效率高和易于硬件实现等优势,被广泛应用于数据加密领域,而利用硬件特性高效/高速实现 SM4 算法成为当前研究的热点[1]。当前 SM4 加解密的硬件密码卡存在性能不足的问题。为实现数据的快速传输,串行总线凭借其传输速度快,结构简单,逐渐在竞争中表现出其

收稿日期:2021-08-20
* 基金项目:山东省重大科技创新工程(2019JZZY010103)项目资助

优点，在许多领域已经开始替代传统的并行总线，例如 RapidIO 技术、PCIe 技术、InfiniBand、CCIX 等技术[2]。文献[1]中提出基于 FPGA 实现 SM4 算法，但其研究重点在 FPGA 资源消耗，成果并非一个成熟的系统；文献[3-4]中提出在 NVIDIA 的 GPU 中实现 SM4 轮转算法，具备高吞吐率的特点，但硬件成本过高，且 Tesla 型号的 GPU 并不适用于普通办公电脑，不具备通用性；文献[5]中实现的国产密码算法的高速 PCIe 密码卡，其国密算法为固化算法，采用软件实现，同时内置 CPU，利用卡内 CPU 加解密计算，与在主机端加解密计算，性能上无本质差异，且 CPU 在做并行计算时，并无优势，此设计性能远低于硬件并行实现国密算法；文献[6]中提出一种 USB 串行 SM1 加密卡设计方案，USB 在数据传输速率上远低于 PCIe，方案使用系统扩展指令完成加解密，存在系统差别导致指令不兼容隐患，采用协处理器软件流水方式实现加解密计算，效能低于硬件流水实现方式。

目前关于 SM4 算法的应用多基于软件实现，相比硬件实现，其运算成本高，需要频繁抢占 CPU 资源，且计算速度不快；而密码芯片又有不易迭代算法、投入成本高的问题。综合以上，本设计提出一种可重构高速数据加解密系统，基于 FPGA 实现，与主机端软件算法的对比实验，验证本设计在性能和传输效率方面的优越性，解决加解密系统关于高速数据传输存储、并行加解密加速、无需安装驱动、直接存储器访问、降低 CPU 中断和运算资源消耗、逻辑可重构的需求问题。

1　系统总体设计

系统主要由电源模块、FPGA 主控模块、GTYE4 接口、DDR 接口等部分组成，系统总体设计原理如图 1 所示。其中，电源模块为 FPGA、GTYE4 芯片、接口电路等提供电源电压，以保证整个系统的正常运行；GTYE4 接口电路负责接收 FPGA 发送的串行数据，并高速加密存储到 DDR 中；FPGA 主控模块是本设计的核心部分，接收外部发送的 PCIe 信号后，经过 SM4 加密的缓存数据写入到 DDR 中，完成数据的传输。

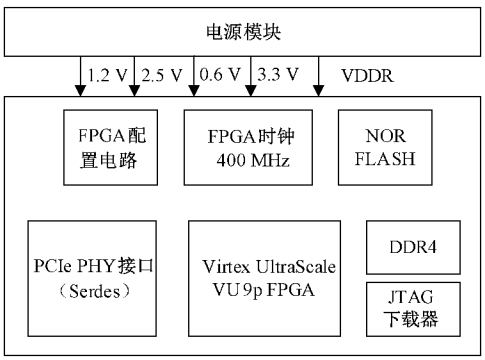

图 1　系统总体设计原理

2　硬件系统设计

2.1　电源模块设计

本设计的总电源由外部输入的 12 V 直流电源提供，通过 TPS54331、TPS51200 等电源降压芯片转换为各个模块所需要的电压。电源模块的供电拓扑结构如图 2 所示，外部输入的 12 V 直流电源经过 3 块 TPS54331 降压芯片，分别输出 3.3、1.5、1.2 V 电压，3.3 V 电压再经过下一级的降压电路输出 2.5、1.8 V 等电压。

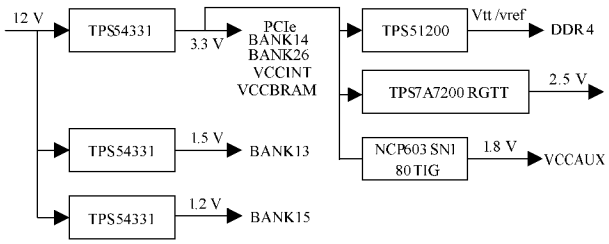

图 2　电源模块

其中，3.3 V 电压主要用于 FPGA 的 BANK14、BANK26、PCIe、VCINT、VCCBRAM 的 IO 供电电源，并为第二级降压电路提供电源；1.5 V 用于 FPGA 的 BANK13 的 IO 供电电源，1.2 V 用于 FPGA 的 BANK15 的 IO 供电电源，TPS51200 的电压用于连接 DDR4。3.3 V 为 FPGA 的内核、内部的 BLOCKRAM 资源和 PCIe PHY（physical，端口物理层）芯片提供电源；2.5 V 为 FPGA 配置电路和 BANK27 的 IO 提供电源；1.8 V 为 FPGA 辅助电压 VCCAUX 提供电源，用于给 FPGA 内各种功能模块的互联资源和输入缓存电路供电。

2.2　FPGA 主控模块设计

本设计选用了 Xilinx 公司生产的 Virtex UltraScale VU9p 系列的型号为 xcvu9p-flgb2014-1-e 的 FPGA 作为主控芯片，该芯片采用 16 nm 工艺实现，能够提供 702 个可用 IO，并具有 48 个 32.74 Gb/s 背板收发器以及 216 Mb BRAM，能够满足设计需求。FPGA 主控模块的设计包含了下载接口、配置电路、DDR4 缓存设计等。

配置电路：FPGA 内部硬件设计采用 SRAM 存储 Vivado 比特流，具有掉电丢失特性，采用以下方式避免出现掉电丢失数据，重启无法运行的情况[7]。在下载和配置 FPGA 时，采取 JTAG 接口烧录 NOR-FLASH 的方式，将 Xilinx 专用的 mcs 文件烧录至 FPGA 开发板的 NOR-FLASH 中，可实现掉电数据不丢失，因采用 FLASH 烧录，烧录时间短，易于调试。如图 3 所示，选用了型号为 MACRONIX-MX66U1G45G 的非挥发性存储器，存储 FPGA 比特流。采用此方式配置 FPGA，解决了掉电数据丢失问题，且加解密算法迭代时，可更新硬件逻辑，降低迭代成本，实现硬件结构的快速重构。

2.3　PHY 接口设计

较文献[8] Zynq 平台实现 PCIe 接口高速传输的 PHY

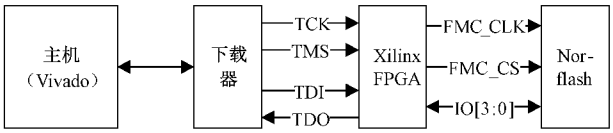

图 3　FPGA 下载与配置电路

特性，本设计采用了 Xilinx 公司的 GTY 系列高速收发器进行 PCIe PHY IP 的设计。选用的 PHY IP 为 GT wizard Sub IP，用以实现 FPGA 与主机的 PCIe 接口通信，该 IP 支持 PCIe 规范的链路层及传输层传输协议，并且能够兼容 PCIe 2.0、PCIe 3.0，最大链接速率可配(2.5、5.0、8.0 GT/s)，具有更高的兼容性。内置的高性能通用 G 比特收发器能够实现并行和串行数据的读写，支持 x2、x4、x8、x16、x32 lane 并行数据总线，可以直接连接主机端 PCIe 接口进行数据传输。PCIe Core 与 PHY IP 链接原理图如图 4 所示。

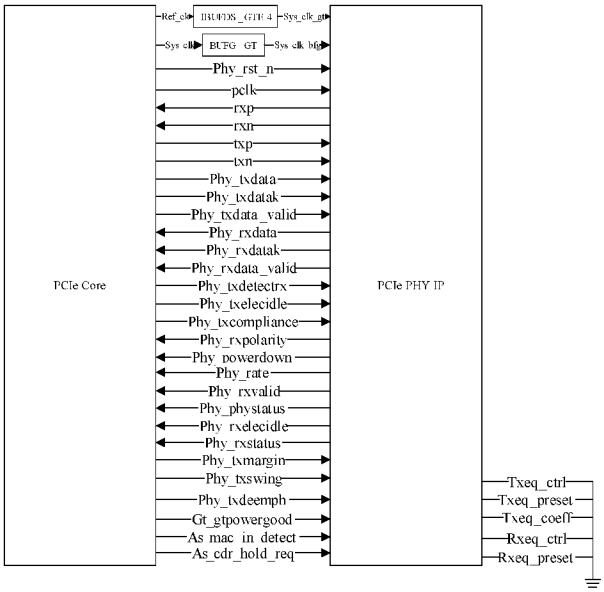

图 4　链路层与 PHY IP 链接原理

PHY IP 分别连接到 FPGA 和计算机，通过物理层接口与 FPGA 进行数据的传输，PHY IP 与 FPGA 的硬件连接如图 4 所示，由 FPGA 内部的 PCIe 核心逻辑模块作为 PHY IP 的控制器，通过对物理层接口进行数据读写来实现 FPGA 与计算机之间通信。本设计无需固件程序，上电后，由计算机端配置 PCIe 配置空间，建立计算机端至存储加密系统 DDR 的内存映射后，计算机端向映射空间直接发送待加密数据及解密数据，该加解密系统收到数据后会根据硬件逻辑自动执行加解密。

2.4　DDR 接口设计

设计内 DDR4 SRAM 内存大小 2 GB，为外置缓存，采用镁光-9UB45 D9XPG 颗粒。FPGA DDR4 IO 的管脚采用 SSTL12_DCI 标准，dqs_t[0]与 dqs_t[1]管脚采用 DIFF_POD12_DCI 标准，电压为 1.2 V。FPGA 与 DDR4 SRAM 硬件连接如图 5 所示。

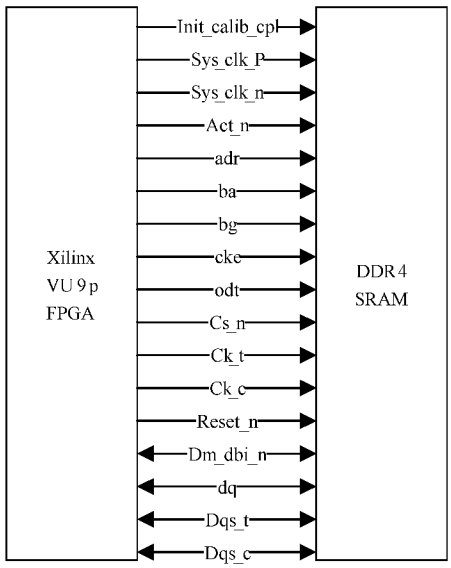

图 5　FPGA 与 DDR4 连接原理

3　系统软件设计

Xilinx 公司的 Virtex UltraScale 系列 FPGA 的程序使用 Vivado 开发环境，系统软件设计采用 verilog 语言编写，FPGA 板卡与计算机的硬件连接方式如图 6 所示。

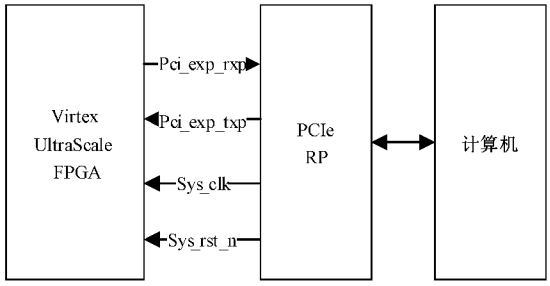

图 6　计算机与 FPGA 硬件连接

FPGA 内部的逻辑模块结构如图 7 所示，主要由 PCIe Core、DDR4 控制器、直接存储器访问(direct memory access，DMA)通信模块、SM4 加解密模块等组成。

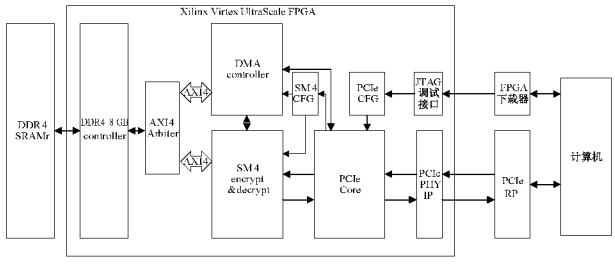

图 7　FPGA 内部模块结构

首先，PCIe RP 端识别端口 EP 的设备身份，并与 PCIe PHY IP 进行链路初始化和训练，完成初始化后，PCIe CFG 模块通过内部高速总线(DBI)访问并配置 PCI Configuration

Space，使能 ltssm_enable 信号，链路层状态机 LTSSM 进入 L0 状态，PCIe 进入正常工作状态。本设计存储管理方式采用固定分区方式，通过主机将 bar0（base address register）配置为输入明文数据地址（DDR 0x8_0000），bar1 为输出密文数据地址，bar2 为输入密文数据地址，bar3 为输出明文数据地址，bar0～bar3 映射的 DDR 内存大小为 2 GB，每个 bar 为 512 MB，bar4 为 SM4 加解密控制器及 DMA 控制器配置地址。

系统分为两种工作模式，由 SM4 加解密控制器（SM4 CFG）配置。

直接加解密模式：加密过程，主机端通过 PCIe 将待加密数据写入到输入明文数据地址 bar0，配置 SM4 加解密控制器密钥、系统参数（Fk）和固定参数（Ck），配置 SM4 使能寄存器。开始工作后，SM4 读取明文开始加密，加密预设长度的数据，密文输出 bar1 地址空间，主机端读取 SM4 CFG 状态寄存器判断加密是否结束；解密过程，主机端通过 PCIe 将待解密数据写入到输入密文数据地址，配置 SM4 加解密控制器密钥、系统参数（Fk）和固定参数（Ck），配置 SM4 使能寄存器，开始工作，SM4 读取密文开始解密，解密预设长度的数据，明文输出至 bar3 地址空间，主机端读取 SM4 CFG 状态寄存器判断解密是否结束。

DMA 方式加解密：配置 DMA 描述符（descriptor），包括：源地址 bar、目的地址 bar、传输长度、burst 长度，配置 SM4 控制器（加密/解密），使能 DMA 控制器，DMA 向 PCIe Core 发送 requester request 请求，PCIe 向源地址 bar＃发送数据，DMA 读取源地址 bar＃的数据并写入到 SM4 内部加解密逻辑，PCIe Core 处理完 lreq 后，发送给 DMA 控制器 requester completion，完成加密后，DMA 将密文/明文写入到目的地址 bar＃。

3.1　PCIe Core 逻辑设计

本设计较文献[9]实现了完整的 PCIe 3 层协议（物理层、链路层、传输层），具有更好的协议兼容性；较文献[10]增加了应用层逻辑，提供便利总线接口，易于实现加密协议种类扩展。

在 PCIe 数据传输过程当中，每一个 PCIe 报文，都需要经过物理层、链路层、传输层、应用层的处理，PCIe 传输的层次图如图 8 所示。物理层分为硬件部分和逻辑部分，硬件部分采用 PHY IP，即 2.3 章节介绍的 PHY 接口，逻辑部分采用 verilog 设计。PCIe EP 不仅需要 RP 端配置链路层和传输层，还要本地配置和使能 PCIe 状态机，但该加解密存储系统并未设计本地 CPU。因此通过在 FPGA 内部设计 PCIe 配置子模块（PCIe CFG），实现有效的 PCIe 状态机使能。

传输层、链路层、物理层实现了大部分事务逻辑、所有数据链路逻辑和物理层的逻辑部分，包括链路训练和状态机（LTSSM）。物理层模块通过 PIPE 连接到外部 PHY。PCIe Core 初始化过程如图 9 所示。PCIe Core 完成初始化

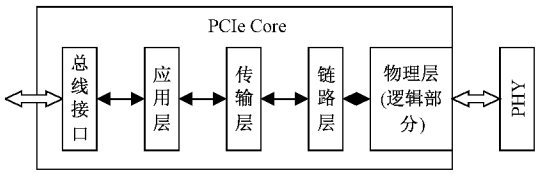

图 8　PCIe Core 层次划分

后，进入工作状态。如图 10 所示，计算机 PCIe RP 端将加密数据和配置数据组成连续的物理层报文（physical packet，PP），完成编码和扰频，在物理层差分驱动器作用下，通过物理接口 PHY 发送给 PCIe EP 端，EP 的物理层在接收到报文后，对接收 PP 报文进行去扰频处理，解码、解析报文得到链路层报文（data link layer packet，DLLP），DLL_R 模块解析链路层报文得到传输层报文（transaction layer packet，TLP），发送至传输层 bar 空间缓存，传输层解析 TLP 报文头文件，并将 TLP 发送给应用层接收模块（ADM_X）。相反方向为数据发送过程。

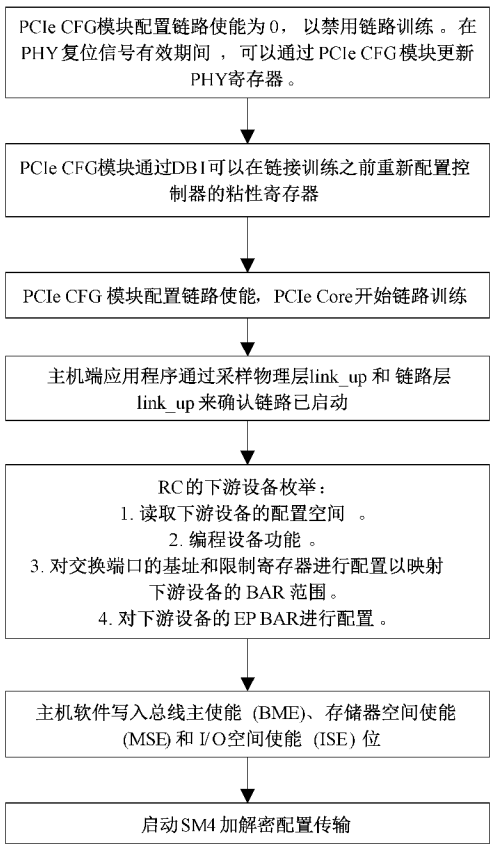

图 9　PCIe 初始化流程

应用层发送模块（ADM_X）为数据包传输实现 PCI Express 事务层的应用特定功能。传输路径使用直通架构。其功能包括：TLP 形成、TLP 仲裁、流量控制（FC）信用检查、返回 SM4 加解密数据及 DMA 请求。

应用层接收模块（ADM_R）为数据包接收实现 PCI Express 事务层的应用特定功能。其功能包括：对接收到

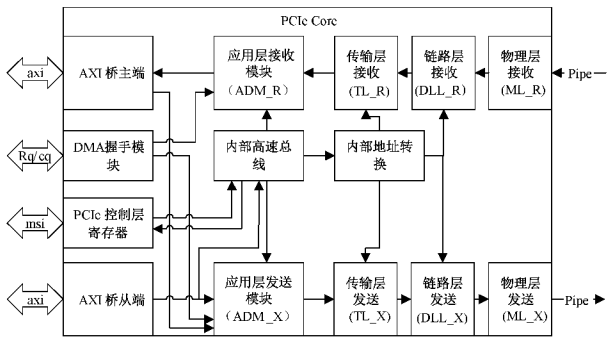

图 10　PCIe Core 架构

的 TLP 进行排序/过滤，过滤规则和路由是可配置的；将接收到的 TLP 缓冲和排队；将接收到的 TLP 路由到 AXI 桥的接收接口，同时将 bar 地址作为 AXI 接口读写基地址，报文解析地址为偏移地址，发送给 SM4 加解密模块。RADM 维护一个接收完成查找表（LUT），用于 ADM_X 非发布请求的完成跟踪和完成超时监控。当预期的 ADM_X 反馈完成未在超时时间内到达时，向 RP 端发送超时中断。Bar 地址映射表如表 1 所示。

表 1　Bar 地址映射表

BAR 标号	Size	Offset	Description
Bar0	512 MB	0x0	明文输入数据地址
Bar1	512 MB	0x0	密文输出数据地址
Bar2	512 MB	0x0	密文输入地址数据
Bar3	512 MB	0x0	明文输出地址数据
		0x0	SM4 加解密控制器地址
Bar4	512 B	0x160	DMA 控制器配置地址
		0x180	保留

图 7 中 SM4 CFG 模块为 SM4 加解密及 DMA 配置模块，其配置空间映射至 bar4 地址；占用 512 B 空间；已使用 128 B，保留 384 B；已使用的 0x00～0x160 用做 SM4 加解密控制、加解密长度、密钥寄存器、系统参数(Fk)寄存器和固定参数(Ck)寄存器；0x160～0x180 用做 DMA 描述符寄存器。

3.2　DMA 控制器设计

在 DMA 模式下，主机端软件首先需要在 PCIe bar4 中 0x160 位置初始化一系列的描述符，这些描述符描述了每一个 DMA 链节点所需的相关参数，包括：描述符控制字、传输字节数、源地址、目的地址等，且这些描述符之间互相链接成链[11]。然后通过指向下一个描述符的指针向驱动获取下一个描述符的信息[12]。SM4 控制器模块将 bar4 0x160 DMA 描述符转存至图 11 所示 DMA 寄存器，并将链式描述符转为描述符队列，仲裁器轮询仲裁队列中的描述符，队列指针指向优先级最高的描述符，SM4 加解密接口、AXI 接口自动从队列中读取描述符，通过与 PCIe Core 端的握手协议，进行 DMA 传输，DMA 的读写握手协议如图 12、13 所示。直到处理完队列中所有描述符。

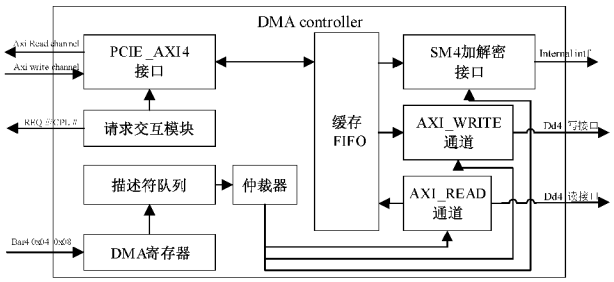

图 11　DMA 控制器逻辑

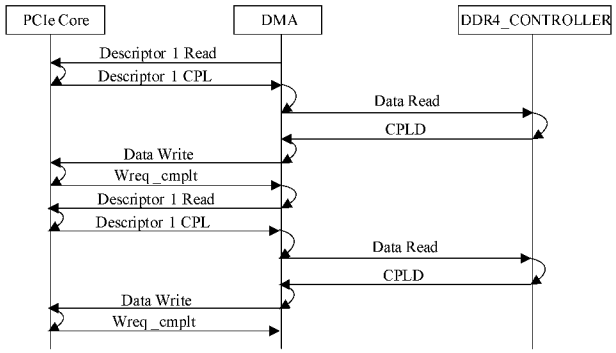

图 12　读握手传输

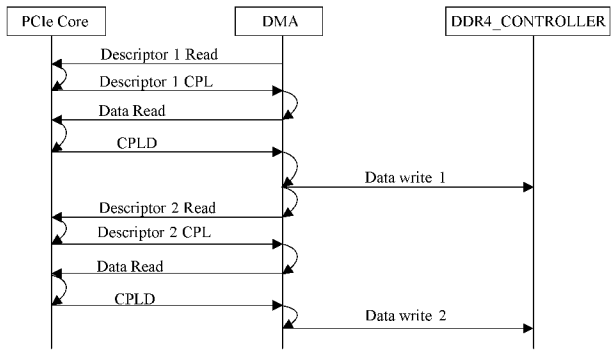

图 13　写握手传输

3.3　SM4 加解密设计

SM4 加解密模块采用流水线设计思想，在结构上进行优化，采用多模组并行逻辑，调整单模组流水线步骤，使用 FIFO 作中间缓存，增设流水线调度模块，支持模组全并行、分组并行和分时并行，最高支持频率为 800 MHz。经 Vivado 报告分析，LUT 资源占用为 25.82%，满足预留 20%FPGA 资源余量需求，实现 32 条并行加解密流水线。经过试验表明，满足 FPGA 资源上限及时序要求，Vivado 可完成逻辑综合及布局布线。

如图 14 所示，控制模块主要由计数器和 Moore 型状态机构成，负责根据输入密钥、待加解密数据和控制信号等，生成 round 和 valid 信号，控制轮密钥生成模块进行 32 轮迭代加密计算并存入寄存器，间接控制流水线加密模块启动数据加密[13]。

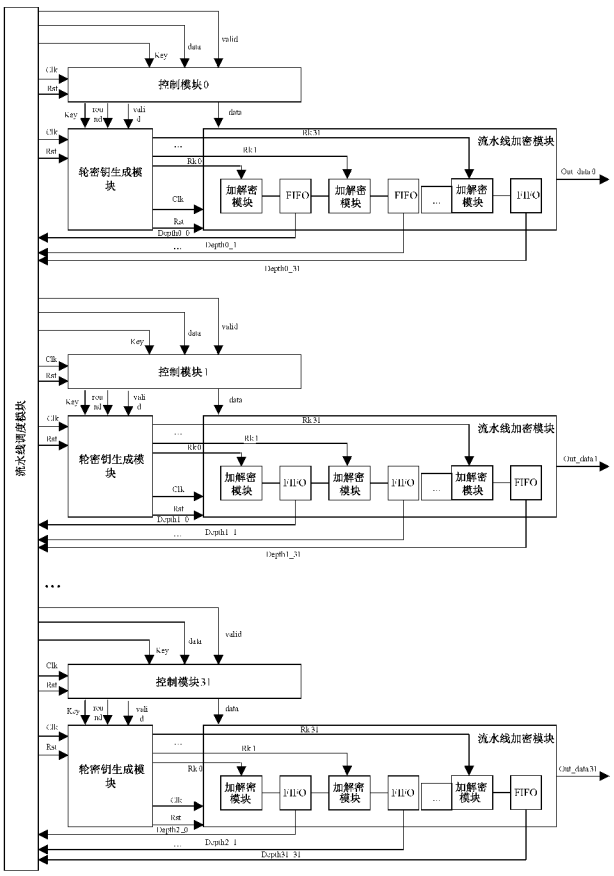

图 14　SM4 加解密模组框架

流水线调度模块主要是由计时器、状态寄存器、多条通道缓存组成。通过每条流水线的各级 FIFO 深度,判断每条流水线的使用程度,控制新的输入加解密数据密度,选择 FIFO 平均深度较低或空闲的流水线;数据加解密运算完成后,根据流水线编号,按照输入序列写回,完成加解密流程。

3.4　DDR4 控制器设计

DDR4 控制器采用 Xilinx DDR4 controller IP,DDR4 controller 采用 AXI4 interface,接口速率最大为 1 600 ps (625 MHz),最小为 833 ps,内存配置选项数据宽度为 16 bit,burst 长度为 8,内存存取潜伏期 (cas latency)为 22,列写潜伏期(cas write latency)为 16,匹配 DDR4 型号为 MT40A1G16RC-062EB,AXI 接口数据宽度为 128 bit,仲裁方式为 RD PRI REG,ID 宽度为 20 bit,地址宽度为 31 bit。

4　系统测试

4.1　测试平台搭建

系统测试的主要目的是测试数据传输的速度,并验证加密测试、解密测试、DMA 功能测试。为了进行系统的测试,首先,需要搭建测试平台,系统测试平台结构如图 15 所示。软件部分,PCIe 驱动设计参考文献[13]中 4.3WDF 驱动设计章节,计算机端加解密软件参考文献[14]中函数构造方法,快速软件实现 SM4 算法。硬件如图 16 所示,设计为外插板卡,插在主机 PCIe 卡槽内,通过 JTAG 接口链接主机 FPGA 下载器,主机端运行测试程序,进行高速数据传输加解密的测试。

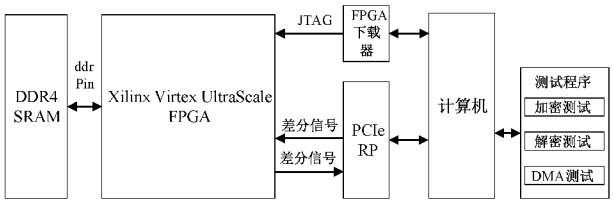

图 15　系统测试平台结构

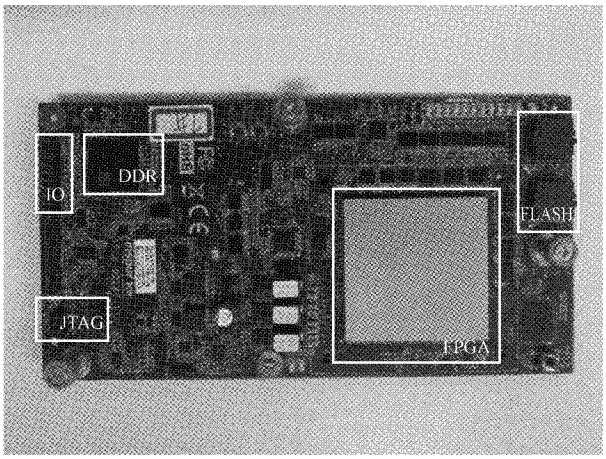

图 16　系统测试板卡

4.2　测试过程记录

系统测试平台搭建完成后,进行系统测试。测试分为 FPGA 烧录阶段、测试程序运行阶段、GOLDEN 数据对比阶段、加解密速度计算等阶段。本设计使用 Xilinx 公司提供的 Vivado 集成设计环境,进行 FPGA 代码综合、实现、下载、调试等工作。首先,对 FPGA 测试程序进行综合编译、实现编译,生成 bit 流文件,再将 bit 文件转换为 mcs 文件,并使用 FPGA 下载器连接电脑和 FPGA,将 mcs 文件烧录到 NOR-FLASH 中,使得整个系统断电后 FPGA 会自动从 NOR-FLASH 中加载程序,从而进入正常的工作状态,确保后续的系统测试能够进行。要使 FPGA 能够与计算机之间进行数据交互,需要先通过 JTAG 接口使能 PCIe 状态机,PCIe 完成初始化后,计算机端依次运行加密测试、解密测试、DMA 测试。通过 PCIe3.0 的接口发送明文、密文、配置数据给 FPGA 内的 SM4 加解密模块。加解密完成后,再将输出数据读回,与计算机本地的 GOLDEN 数据对比,用以验证加解密的正确性。加密测试数据对比如图 17 所示,通过 Notepad 比对软件算法模型加密结果与 FPGA 输出加密密文,确定加密结果正确,与预期结果一致;解密测试数据对比如图 18 所示,通过 Notepad 比对软件算法模型解密结果与 FPGA 输出解密数据,确定解密结果正确,与预期结果一致。

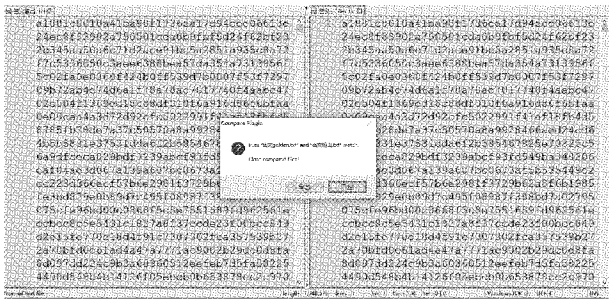

图 17　加密测试数据对比

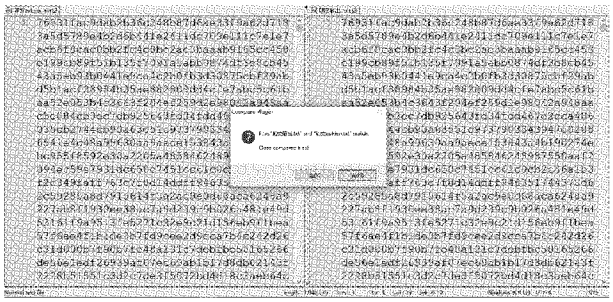

图 18　解密测试数据对比

4.3　测试结果分析

根据以上测试结果，可以看出在多次数据传输的过程中，数据传输稳定可靠，没有数据的丢失或错误，加解密功能符合设计需求。根据记录的加解密计算时间，整理出表 2、3，其中，E 为加速比，E_{s1} 为 CPU 与 FPGA 加解密时间比，E_{sp} 为 GPU 与 FPGA 加解密时间比，T_1 为 CPU 加密运算时间，T_s 为加密系统计算时间，T_{st} 为加密系统数据传输时间，T_p 为 GPU 加密时间。测试环境主机端为 win10/CPU intel i7，GPU 为 GTX1060，针对不同的明文数据块进行加解密速度分析，SM4 加解密时钟频率配置为 400 MHz(极限为 800 MHz)，在 32 条并行流水线工作时，设计完成 8 MB 数据块加密，最短用时 83.456 ms，加解密速度达到 98.159 KB/ms；但 CPU 具有硬件 cache，在少量数据加解密计算时，读取数据速度明显快于 PCIe3.0，使其花费的总时间较低；随着数据量的扩大，当数据量达到 8 KB 时，本文方案所采用的并行流水线与直接存储器访问架构优势逐步体现。如表 2，14 行所示，当数据量达到 8 MB 时，其加解密频率约为本测试环境下 CPU 实现的 25.78 倍，如表 3，14 行所示，其加解密频率约为本测试环境下 GPU 实现的 1.02 倍。相较于文献[15]，数据量达到 8 MB 时，该加密卡加解密所需时间远低于文献[15]的 115.81 ms。与文献[3]中基于高性能 GPU 实现的 SM4 算法加解密性能相差较小，但本文方案在硬件升级、迭代成本、功耗等方面具有明显优势。FPGA 综合报告资源利用部分如表 4 所示，LUT 利用率为 25.82%，BRAM 为 21%，满足预留 20% FPGA 资源的需求下，保留有至少 50% FPGA 资源可供算法升级，加解密速率提升空间约为 45.76%。

表 2　对比 CPU 加解密速度测试

数据块	流水线	T_s/ms	T_{st}/ms	T_1/ms	E_{s1}
2 KB	32	0.001 0	1.513	0.535	0.35
4 KB	32	0.001 1	1.521	1.031	0.67
8 KB	32	0.001 1	1.537	2.114	1.37
16 KB	32	0.001 2	1.549	4.416	2.84
32 KB	32	0.002 5	1.621	8.622	5.31
64 KB	32	0.005 2	1.776	16.452	8.99
128 KB	32	0.010 3	1.819	34.216	18.7
256 KB	32	0.021 0	3.415	68.348	19.9
512 KB	32	0.042 3	5.732	136.757	23.6
1 MB	32	0.085 9	11.654	273.393	23.2
2 MB	32	0.175 6	20.1	545.992	26.9
4 MB	32	0.374 1	41.79	1 100.391	26.0
8 MB	32	0.793 6	83.695	2 178.5	25.78

表 3　对比 GPU 加解密速度测试

数据块	流水线	T_s/ms	T_{st}/ms	T_p/ms	E_{sp}
2 KB	32	0.001 0	1.513	2.058	1.35
4 KB	32	0.001 1	1.521	2.098	1.37
8 KB	32	0.001 1	1.537	2.174	1.41
16 KB	32	0.001 2	1.549	2.017	1.30
32 KB	32	0.002 5	1.621	2.026	1.24
64 KB	32	0.005 2	1.776	2.220	1.24
128 KB	32	0.010 3	1.819	2.456	1.34
256 KB	32	0.021 0	3.415	3.864	1.12
512 KB	32	0.042 3	5.732	6.276	1.08
1 MB	32	0.085 9	11.654	11.671	0.99
2 MB	32	0.175 6	20.1	22.854	1.12
4 MB	32	0.374 1	41.79	43.907	1.04
8 MB	32	0.793 6	83.695	85.768	1.02

表 4　FPGA 综合报告资源部分

资源	已用	可用	利用率/%
LUT	30 526	1 182 240	25.82
LUTRAM	2 508	591 840	0.42
FF	267 825	2 364 480	11.33
BRAM	453	2 160	21
URAM	145	960	15.1
DSP	17	6 840	0.25
IO	75	702	10.6
BUFG	35	1 800	1.94
MMCM	1	30	3.33
PLL	1	60	1.67

5　结　　论

本文设计了一种基于 PCIe 的可重构高速数据加密系

统。相较于传统 SM4 加解密卡，它实现了高速数据传输存储，直接存储器访问，并行加解密加速，逻辑可重构的硬件设计。本文设计采用 verilog 语言设计实现，设计有 DMA 模块，降低了 CPU 中断和运算资源消耗；设计有 PCIe3.0 传输逻辑，接口速率达到 8 GT/s；采用并行流水线加密，加解密计算需求可灵活搭配流水线资源，数据加解密计算速率能够达到 98.159 KB/ms。实际检验表明，数据块为 8 MB，加解密频率约为 CPU 实现方式的 25.78 倍，为一般 GPU 实现方式的 1.02 倍，且具有较高的正确率和稳定性。考虑到数字逻辑综合时序检查，实验采用 SM4 计算逻辑时钟频率为 400 MHz，Xilinx VU9p FPGA 理论极限为 800 MHz，该频率对设计综合后数字电路的建立时间和保持时间要求较高，FPGA 资源整体利用率不足 26%，方案在电路时序和 FPGA 资源利用方向具备巨大的升级空间。较先前学者提出的 SM4 加解密设计方案，在系统传输速率、加解密性能、节省主机资源、迭代成本、可扩展性上都有明显优势。

参考文献

[1] 何诗洋，李晖，李风华. SM4 算法的 FPGA 优化实现方法[J]. 西安电子科技大学学报，2021，48(3)：155-162.

[2] 刘健，李会方. 基于 PCIe 的 FC 数据采集存储方案的设计与实现[J]. 国外电子测量技术，2013，32(10)：42-44.

[3] 李秀滢，吉晨昊，段晓毅，等. GPU 上 SM4 算法并行实现[J]. 信息网络安全，2020，20(6)：36-43.

[4] 张才贤. 基于 CUDA 的并行 SM4-GCM 设计与实现[D]. 西安：西安电子科技大学，2019.

[5] 赵军，曾学文，郭志川. 支持国产密码算法的高速 PCIe 密码卡的设计与实现[J]. 电子与信息学报，2019，41(10)：2402-2408.

[6] 张锋，朱振荣，史胜伟. 一种高速免驱 USB 加密卡的设计与实现[J]. 计算机工程，2017，43(11)：292-296，302.

[7] 李锦明，郑志旺. 基于 LVDS 和 USB3.0 的高速数据传输接口的设计[J]. 电子测量技术，2021，44(7)：1-6.

[8] 杨亚涛，张松涛，李子臣，等. 基于 Zynq 平台 PCIE 高速数据接口的设计与实现[J]. 电子科技大学学报，2017，46(3)：522-528.

[9] 廖寅龙，田泽. FC 网络通信中 PCIe 的接口的设计与实现[J]. 航空计算技术，2011，41(4)：127-130.

[10] 李明. 基于 PCI-E 高性能密码卡的关键技术的研究[D]. 西安：西安电子科技大学，2018.

[11] 王亮，杨玻，王璇. 基于 PCIe 总线的多处理器通信算法设计[J]. 信息技术与信息化，2021(2)：156-158.

[12] 孙欣欣，李娟，田粉仙，等. 一种基于 PCIE 总线的 DMA 引擎研究[J]. 云南大学学报(自然科学版)，2021，43(3)：444-450.

[13] 程鹏. 基于 WDF 驱动模型的 PCIE 压缩板卡的驱动设计[D]. 南京：东南大学，2016.

[14] 张笑从，郭华，张习勇，等. SM4 算法快速软件实现[J]. 密码学报，2020，7(6)：799-811.

[15] 王德民，陈达. 基于 CUDA 的 SM4 加密算法高速实现[J]. 石家庄铁路职业技术学院学报，2017，16(1)：59-63.

作者简介

王凯，本科，主要研究方向为集成电路设计、IC 验证技术。

E-mail：745579593@qq.com

刘凯，硕士，主要研究方向为集成电路设计、IC 验证技术。

E-mail：liukaibj@inspur.com

李拓，硕士，主要研究方向为集成电路设计、IC 验证技术。

E-mail：lituo@inspur.com

符云越，本科，主要研究方向为集成电路设计、IC 验证技术。

E-mail：fuyunyue@inspur.com

刘唐，博士，主要研究方向为集成电路设计、IC 验证技术。

E-mail：liutang01@inspur.com

王赛，硕士，主要研究方向为集成电路设计、IC 验证技术。

E-mail：wang.qianlc@inspur.com