

基于 eMMC 阵列的高速大容量数据的存储设计

叶聪相, 章增优

(浙江工贸职业技术学院 人工智能学院, 浙江 温州 325000)

摘要: 为实现对某飞行设备进行模拟飞行测试, 需要对大量的飞行仿真数据进行快速存储, 因此提出一种基于 eMMC (embedded MultiMedia Card) 阵列的高速大容量数据的存储设计; 设计选用 FPGA (field programmable gate array) 作为逻辑控制核心, 通过控制其内部 IP 核 GTX (Gigabit Transceiver) 实现与外部设备的通讯, 为匹配数据的传输速率, 采用缓存模块对 GTX 模块输出的并行数据进行缓存, 依据缓存大小来控制 eMMC 阵列控制器模块对 eMMC 阵列进行写入; 系统上电后, GTX 模块及 eMMC 阵列控制器模块自动进入初始化流程, 上位机通过 RS422 接口发送启动命令给 FPGA, 当 FPGA 检测到各模块初始化完成标志后, 通过 RS422 接口上传指令回应帧并启动数据的采集存储; 数据存储完成后, 为验证数据传输存储的正确性, 上位机发送读取命令对数据进行回读; 经验证, 回读数据正确, 无丢帧错帧的现象, 数据传输稳定可靠, 数据存储速率为 2.4 Gbps; 系统设计正确, 为模拟飞行测试提供了可靠的保障。

关键词: eMMC 阵列; GTX; 高速大容量; 存储; FPGA; RS422 接口

Design of High-speed and Large-capacity Data Storage Based on eMMC Array

YE Congxiang, ZHANG Zengyou

(Artificial Intelligence College, Zhejiang Industry & Trade Vocational College, Wenzhou 325000, China)

Abstract: In order to realize the simulation flight test of a flight equipment, it is necessary to store a large amount of flight simulation data quickly, so a design of high-speed and large-capacity data storage based on eMMC (embedded MultiMedia Card) array is proposed. The design uses FPGA (field programmable gate array) as the core of logic control. It communicates with the peripheral equipment by controlling its internal IP core, the GTX (Gigabit Transceiver). In order to match the data transfer rate, the parallel data output from GTX module is cached by cache module, and eMMC array controller module writes eMMC array according to the size of cache. When the system is powered up, the GTX module and eMMC array controller module enter the initialization process automatically, the host computer sends the Startup instruction to the FPGA through the RS422 interface. When the FPGA detects the completion of the initialization of each module, it uploads the instruction to respond to the frame through the RS422 interface and starts the data acquisition and storage. After the data storage is completed, in order to verify the correctness of the data transmission and storage, the upper computer sends a reading instruction to read back the data. It has been verified that the data read back is correct, there is no loss of frame error, the data transmission is stable and reliable, the transmission rate is 2.4Gbps. The design of the system is correct, which provides a reliable guarantee for the simulation flight test.

Keywords: eMMC array; GTX; high-speed and large-capacity; storage; FPGA; RS422 interface

0 引言

随着信息时代的蓬勃发展, 各行业对信息数据的需求更加迫切。产品研制阶段需要大量的数据支撑, 通过分析数据能够实现对产品性能的不断优化, 因此大容量数据的存储技术持续受到各行各业的关注。首先, 传统的存储芯片 Flash 随着芯片制造工艺的不断改进, 不同厂商生产的芯片也存在差异^[1], 因此虽然 Flash 存储空间越来越大, 但其可移植性降低, 进一步影响软件的开发周期^[2]。其次, 在对 FLASH 芯片进行读写控制时过程较复杂, 产品异常断电

时 Flash 芯片内部指针可能会随意跳转, 导致数据读写错误^[3], 且数据通过 FLASH 芯片接口传输时速率低^[4]。针对以上问题, 提出采用 eMMC 存储芯片来对数据进行存储。eMMC 芯片与其他存储芯片最大的不同是其内部包含了 FLASH 芯片以及 FLASH 芯片控制模块^[5], 解决了不同型号存储芯片兼容性的问题, 极大地提高了软件的可移植性, 节省了产品开发周期。该存储芯片存储容量大, 接口传输速率高, 可实现数据的高速传输^[6]。因此, 这种嵌入式芯片获得了越来越多的青睐。

收稿日期: 2021-03-08; 修回日期: 2021-04-01。

基金项目: 温州市科技基金项目(N20170010)。

作者简介: 叶聪相(1979-), 男, 浙江温州人, 硕士, 讲师, 主要从事计算机应用技术, 多媒体应用技术方向的研究。

章增优(1977-), 男, 浙江温州人, 硕士, 副教授, 主要从事计算机应用技术、人工智能技术应用方向的研究。

引用格式: 叶聪相, 章增优. 基于 eMMC 阵列的高速大容量数据的存储设计[J]. 计算机测量与控制, 2021, 29(10): 181-186.

由于 FPGA 具有强大的并行处理数据的能力，且内部集成了很多硬核，对其进行编程时可直接调用，缩短了软件开发时间^[7]，系统选用 FPGA 作为控制芯片，利用其内部集成的 IP 核实现对外部数据的高速收发，然后对数据进行暂存及数据组帧处理，使用 MMC 接口实现对对 8 片 eMMC 芯片的读写操作，完成高速大容量数据的存储。本设计对于产品进行后续测试提供了保障。

1 系统总体设计

设计主要由 RS422 收发模块、主控模块、eMMC 存储控制模块、数据缓存模块、GTX 控制模块及电源模块等模块组成。系统总体设计如图 1 所示。

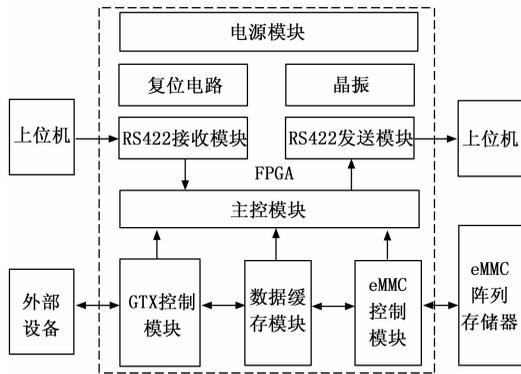


图 1 系统总体设计框图

系统上电复位完成后，eMMC 控制模块以及 GTX 控制器模块自动进入初始化流程。FPGA 通过 RS422 接口接收到上位机发送的启动采数命令时，首先对 eMMC 控制模块以及 GTX 控制模块的状态进行判断，若其初始化未完成，则将回应帧中的 Busy 标志位置 ‘1’；若初始化流程结束，则 Busy 标志位清 ‘0’，并准备数据的采集。GTX 控制模块启动对外部串行数据的高速采集，为匹配数据传输速率，使用数据缓存模块对高速收发控制器模块输出的数据进行暂存，FPGA 主控模块根据缓存模块输出的空满标志，一方面控制 eMMC 存储控制模块对 eMMC 芯片进行读、写、擦除等工作；另一方面对 GTX 控制模块的收发也进行控制。eMMC 控制模块对数据按照传输协议组帧后，通过 MMC 接口将数据存储至外部 eMMC 芯片中。回读是数据写入的逆过程。经系统实测，eMMC 接口传输速率为 2.4 Gbps，实现了设计需求。

2 系统硬件设计

2.1 GTX 收发模块设计

设计选用 Virtex-6 系列的 FPGA 作为逻辑控制器，采用其内部集成的 IP 硬核 GTX 实现对大容量数据的高速传输^[8]。GTX 内部由两部分构成，分别为接收模块和发送模块^[9]。其中，接收模块主要实现的功能依次对输入信号进行整形滤波，从输入串行数据流中将时钟与数据恢复出来，串并转换，数据字符对齐，解码，时钟纠正，位宽转换输出^[10]。发送模块实现数据发送的步骤大致与接收相反，依

次为位宽转换输入，编码，并串转换，信号预加重，信号转差分输出^[11]。发送模块与接收模块传输时互不影响，GTX 模块通过 SMA 接口与用户设备进行连接，信号输入为差分方式，使得数据传输更稳定^[12]。GTX 控制模块在进行数据传输时，其接口的传输速率理论值为 6.6 Gbps^[13]，完全符合设计对数据传输速率的要求。GTX 控制模块接口原理如图 2 所示。

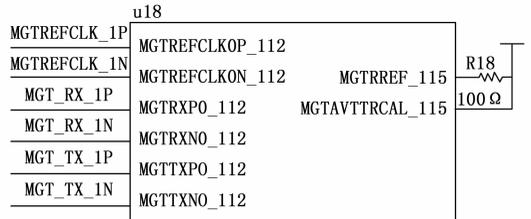


图 2 GTX 接口原理图

2.2 eMMC 阵列模块设计

设计采用美国美光公司的 eMMC 芯片 (MTFC64GJVDN-4MIT) 进行大容量数据的存储^[14]。通过该芯片进行数据的传输时，其接口的传输速率为 3.2 Gbps，能够实现数据的高速收发^[15]。由于 eMMC 芯片集成了 Flash 芯片及 Flash 控制器，FPGA 只需通过简单的接口就可对其进行操作^[16]。FPGA 与 eMMC 芯片通讯的接口原理如图 3 所示，设计采用 6 片 eMMC 芯片进行数据存储，每片可以存储的空间大小为 64 GB^[17]。FPGA 与 eMMC 阵列之间主要通过时钟线、数据线、命令线、电源线以及地线进行通讯。其中，时钟线、电源线以及地线是共用线，数据线和命令线则各自独立^[18]。硬件在设计时，为降低干扰，在电源接口处引入滤波网络，从而保证电源的可靠性^[19]。同时硬件上将数据线及命令线均进行上拉，提高了芯片管脚的驱动力^[20]。

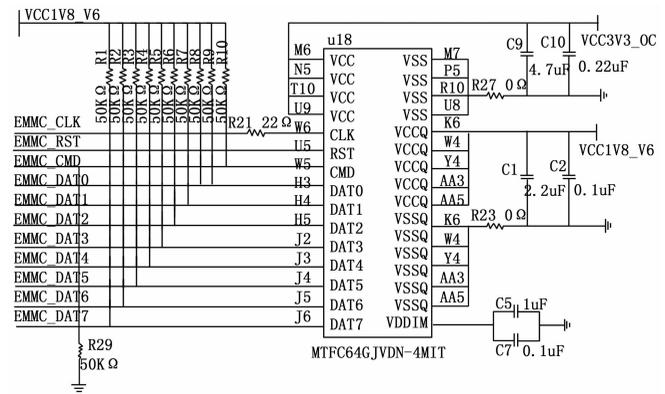


图 3 eMMC 阵列接口原理图

3 系统软件设计

系统采用 Verilog 语言在 ISE14.6 开发环境下实现软件代码的编写，软件整体架构如图 4 所示。系统上电后，上位机通过 RS422 接口发送命令给 FPGA，待 FPGA 各模块初始化完成后，启动对外部数据的接收存储。首先外部设备的数据经过 GTX 模块快速接收后，设计为匹配数据传输

速率, 保证数据传输无误, 将数据进行缓存后再发送至 eMMC 控制模块。eMMC 控制模块根据上位机命令对 eMMC 芯片进行相应的操作, 通过命令接口、数据接口实现对 eMMC 芯片的初始化、写入、读出等功能。

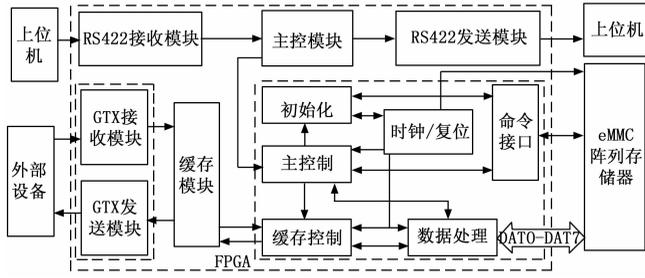


图 4 系统软件总体架构图

3.1 eMMC 协议帧格式

3.1.1 数据帧格式

FPGA 作为主机, 控制数据总线与 eMMC 芯片进行数据交互。在对 eMMC 芯片写入数据时, 需要按照正确的传输格式对数据进行重新组帧才可以实现数据的正常发送^[21]。其协议帧格式如图 5 所示, 主要由 4 部分构成。其中, 起始位为固定值 ‘0’, 结束位为固定值 ‘1’, CRC 为 2 选择字节, 数据包长度不定。数据在 SDR 模式下传输时, 可以选择 1 bit、4 bit 或者 8 bit 的传输位宽, 且数据均在时钟上升沿进行采样。

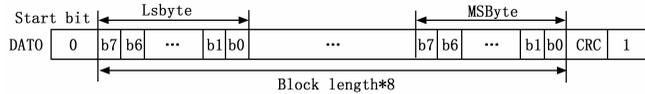


图 5 eMMC 协议数据帧格式

3.1.2 命令帧格式

FPGA 作为主机, 控制命令总线与 eMMC 芯片进行命令交互。为确保命令发送有效, 当 eMMC 芯片接收到命令时, 会发送命令响应至 FPGA。FPGA 通过发送不同的命令实现对芯片的操作, 但这些命令都具有相同的帧格式, 如表 1 所示。命令帧长度固定, 均为 6 个字节。

表 1 eMMC 命令帧格式

说明	比特位置	宽度	值
开始位	47	1	0
传递位	46	1	1
命令编号	[45 : 40]	6	X
参数	[39 : 8]	32	X
CRC7	[7 : 1]	7	X
停止位	0	1	1

主机发送不同的命令, 接收到的响应长度各不相同。且命令发出后, 将在各自要求的时间后收到响应信息。如图 6 所示, 当主机发送命令给单片 eMMC 芯片设置初始地址时, 等待 NCR 个周期后, 得到 eMMC 芯片发送的响应信息。

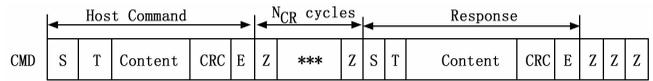


图 6 eMMC 协议命令和响应时序图

3.2 eMMC 初始化模块设计

系统上电后, 通过发送命令实现对 eMMC 芯片内的寄存器进行配置, 从而完成 eMMC 芯片的初始化工作。首先, 根据 eMMC 芯片特性, 芯片上电复位完成后, 配置 200 Hz 的时钟作为 eMMC 芯片的初始化主频。然后, 发送 CMD1 命令完成对工作条件寄存器的配置, 实现 eMMC 工作电压的检验与匹配。发送 CMD2 命令可获取产品出厂信息等。由于设计选用 6 片 eMMC 芯片用于大容量数据的存储, 因此当 FPGA 对其进行访问时, 需要唯一的芯片地址与之对应, 因此在配置芯片时, 可以通过写入 CMD3 命令对其内部相对地址寄存器进行每片芯片地址的配置, 使得数据在传输时具有一对一的效果。根据 eMMC 传输帧格式, 一帧数据中除去开始位、结束位以及校验位以外, 数据字节长度不定。通过发送命令 CMD9, 可以查询到 eMMC 芯片在数据传输过程中支持的数据字节长度等信息。然后发送 CMD7 命令选定 eMMC 芯片参与初始化流程结束后的数据读写工作。最后发送 CMD6 命令将输入时钟切换至 5 MHz, 并重复发送该命令确认数据传输位宽及采样时刻, 初始化流程结束。eMMC 控制模块初始化流程如图 7 所示。

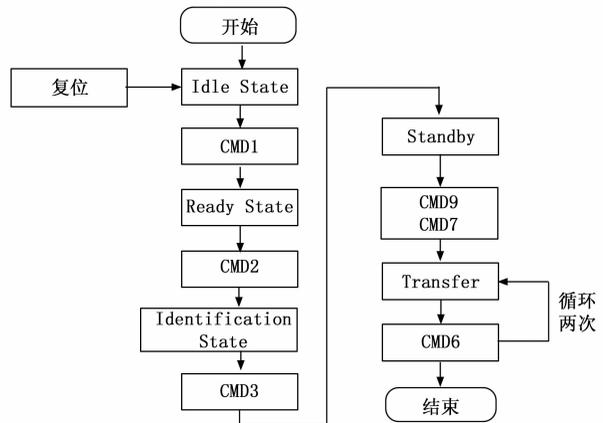


图 7 eMMC 控制模块初始化流程图

3.3 eMMC 写模块设计

将大容量数据写入 eMMC 存储芯片时, 首先由 FPGA 通过命令串行输出 48 bit 的 CMD7 至 eMMC 存储器, 用于指定 6 片 eMMC 存储器之一作为从机。然后发送命令 CMD16 配置有效的数据包长度。完成以上命令的发送后, FPGA 发送写指令 CMD24 至 eMMC 芯片, 并且在 FPGA 收到正确的命令响应之后开启数据的传输。eMMC 芯片开始接收时, 首先将数据接收到内部缓存器中, 同时对接收到的每 1 bit 数据进行校验, 当接收完一帧数据时, 会将 CRC 正确与否的标志通过数据线 0 发送至 FPGA, 该标志包括 3 位, 若该标志为 2 时表示数据写入正确, 此时缓存器

令线上的回应为 0x0600000800cb, 表明 eMMC 芯片初始化已完成。

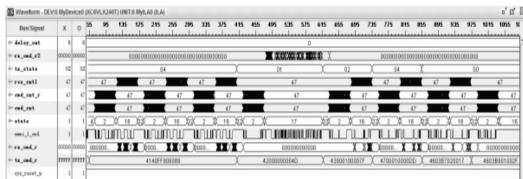


图 13 eMMC 模块初始化验证图

4.4 eMMC 模块数据写验证

为加快数据的写速率, 简化写入步骤。在 FPGA 对 eMMC 芯片进行写操作时, 采用多块写的方式进行。数据传输前, 设置好写入数据的块数据写入后, 将自行完成块数的递增, 并完成数据的写入。发送完预设的数据字节长度后, 不断查询数据线 0 上的校验标志, 来确认数据写入的正确性。如图 14 所示, 经系统验证, 数据线 0 返回的 3bit CRC 标志为 0x010, 表示多块写成功。

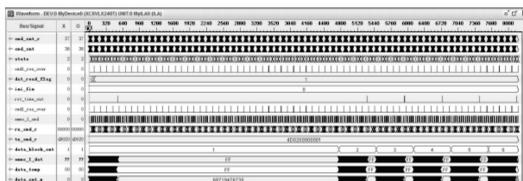


图 14 eMMC 模块多块写验证图

4.5 eMMC 模块数据读验证

eMMC 模块数据读操作同样采取预先设置指定读块数, 然后发送读命令后, 自动读取相应 eMMC 地址的数据。通过对块数预设值和读取值, 来判定读流程是否结束。经系统验证, 预设读取 6 块数据帧, 同时对读取的数据帧进行计数, 结果数值一致, 验证正确。完成的方式。eMMC 模块的数据读过程如图 15 所示。

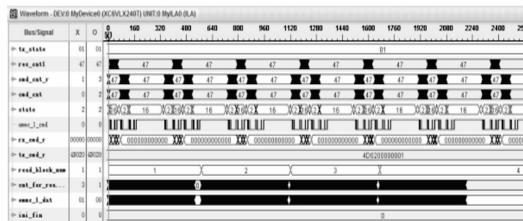


图 15 eMMC 模块多块读验证图

4.6 系统验证验证

对系统各模块功能进行验证后, 需针对全局时序进行仿真验证。FPGA 主控模块识别上位机输入的写命令, 控制 GTX 模块启动对外部数据的接收, 然后经过 eMMC 控制模块对数据重新组帧处理后, 将其串行输入至 eMMC 阵列中。对该过程采用 Chipscope 进行在线监测, 如图 16 所示。根据数据的采样点, 计算其传输速度为 2.4 Gbps, 从而验证了本设计可以实现数据的高速存储。

上位机输入读命令后, 启动对 eMMC 阵列的数据读取。

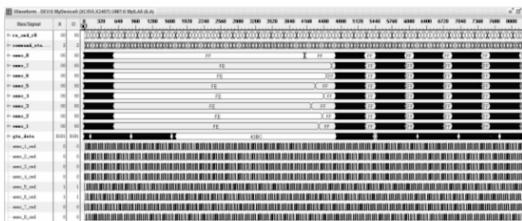


图 16 系统写验证图

然后将读取的数据进行相应的处理后, 发送给 GTX 控制模块发送至外部设备。对该过程采用 Chipscope 进行在线监测, 如图 17 所示。根据数据采样点, 计算其传输速度为 3.0 Gbps, 实现了数据的高速读取。

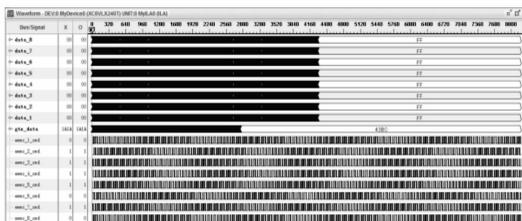


图 17 系统读验证图

测试设备采用对比工具, 将回读的数据与发送的数据进行比较后, 确认数据的一致性。验证了数据传输存储的正确性和可靠性。

5 结束语

系统为实现对某飞行仿真数据的快速存储, 设计了基于 eMMC 阵列的高速大容量数据的存储设计。采用 GTX 控制模块与 eMMC 控制模块相协作, 实现了对大容量数据的高速存储, 数据存储速率为 2.4 Gbps。通过对数据回读验证, 系统传输存储设计稳定可靠, 数据未出现错帧丢帧的现象。系统采用 eMMC 阵列进行数据的存储, 简化了 FPGA 对存储器的读写控制, 缩短了研发周期, 节约了成本, 为大容量数据的测试提供了良好的保障。

参考文献:

- [1] 秦建军. 基于 FPGA 的高速数据采集与存储系统设计 [D]. 哈尔滨: 哈尔滨工程大学, 2013.
- [2] 宋师军. 基于 eMMC 的 64 路无线数据存储系统设计 [J]. 中北大学学报 (自然科学版), 2019, 40 (5): 470-475.
- [3] 周珍龙. 基于 FPGA 的 eMMC 阵列存储系统设计与实现 [D]. 西安: 西安电子科技大学, 2015.
- [4] 张耀军. 基于 FPGA 的高速 eMMC 阵列控制器的设计与实现 [D]. 西安: 西安电子科技大学, 2015.
- [5] 胡文文. 基于 eMMC 存储系统性能的研究和优化 [D]. 北京: 北京邮电大学, 2012.
- [6] Xilinx. Virtex-6 FPGA clocking resource user guide [Z]. USA: Xilinx, 2012.
- [7] 戴荣, 阴陶. 基于 FPGA 的高速 eMMC 阵列控制器 [P]. 中国, 104409099A, 2014-12-15.

- [8] Micron. eMMCmemory data sheet [S]. USA: Micron, 2013.
- [9] 郑亮. SD/MMC 控制器的设计与实现 [D]. 济南: 山东大学, 2014.
- [10] 张守将. 基于 ROCKET I/O 的高速数据传输系统研究 [D]. 西安: 西安电子科技大学, 2013.
- [11] 王树峰, 刘纪文. 集成 NAND FLASH 闪存功能的 eMMC 存储器 [P]: 中国, 203038913U, 2012-12-25.
- [12] 李攀. 基于 eMMC 的星载大容量存储关键技术研究 [D]. 哈尔滨: 哈尔滨工业大学, 2017.
- [13] 丁朋程. 基于 SRAM 型 FPGA 的抗单粒子效应容错技术的研究 [D]. 兰州: 西北师范大学, 2013.
- [14] 王帅. 基于 FPGA 的 PCIE 多路高速数据采集系统的设计 [D]. 济南: 济南大学, 2017.
- [15] 方博. 基于 eMMC 的硬件电路设计及模型验证 [D]. 杭州: 杭州电子科技大学, 2017.
- [16] 侯天喜. 基于 eMMC 阵列的高速固态存储器的研究与设计 [D]. 太原: 中北大学, 2018.
- [17] 姜俊宏. 基于 eMMC 阵列的高速大容量存储与传输系统 [D]. 南京: 南京理工大学, 2018.
- [18] 高阳. 弹载数字图像采集系统研究 [D]. 太原: 中北大学, 2017.
- [19] 李远哲, 贺海文, 赵峰. 嵌入式系统大容量 NAND Flash 存储器分区管理设计 [J]. 计算机测量与控制, 2019, 27 (2): 212-215.
- [20] 张煜, 陈微, 吴利舟, 肖依. 基于 FPGA 的 HS400 模式 eMMC 控制器设计与实现 [J]. 计算机工程与科学, 2018, 40 (6): 969-976.
- [21] 陈锋. 基于 FPGA 的 8B/10B 编解码设计 [J]. 电子设计工程, 2010, 18 (5): 152-154.

(上接第 138 页)

5 结束语

为了适应智慧校园安全应用的需求, 提出了一种基于大数据技术的智慧校园安全管控平台设计方案, 针对智慧校园体系各层的特征设计不同安全防护机制, 通过在智慧校园各层部署感知器和在关键网络口部署流探针采集器设备、系统、攻击者、网络、用户等日志和行为数据, 利用大数据技术进行数据融合分析, 及时分析处置安全威胁, 预测智慧校园安全风险, 全面感知智慧校园安全态势, 通过近 1 年来智能运营管理平台实际应用数据分析, 安全风险告警及时性以及自我处置能力明显增强, 有效推动智慧校园建设与应用进一步深入。后期将进一步研究智慧校园安全数据融合技术、数据关联处理技术, 并结合态势感知技术, 使之更加有效处理分析各类复杂的安全数据, 真正实现全面实时感知智慧校园安全态势。

参考文献:

- [1] 陈跃辉, 王以伍. 高校网络安全现状分析及风险应对策略 [J]. 中国医学教育技术, 2019, 33 (1): 68-71.
- [2] 童瀛, 姚焕章, 周宇. 大数据背景下网络信息安全技术体系分析 [J]. 网络安全技术与应用, 2021 (5): 67-68.
- [3] 黄丽芬. 大数据环境下高校网络空间安全治理策略研究 [J]. 职业教育研究, 2020 (5): 37-41.
- [4] 杨婕. 基于顶层设计思路的企业安全架构总体设计 [J]. 信息通信, 2017 (7): 249-251.
- [5] 田由辉. 教育信息化 2.0 背景下智慧校园的网络信息安全治理研究 [J]. 信息技术与信息化, 2020 (10): 184-187.
- [6] SHELAR D, AMIN S. Security assessment of electricity distribution networks under DER node compromises [J]. IEEE Transactions on Control of Network Systems, 2017, 4 (1): 23-36.
- [7] 龚俭, 臧小东, 苏琪, 等. 网络安全态势感知综述 [J]. 软件学报, 2017, 28 (4): 1010-1026.
- [8] 刘冬兰, 刘新, 等. 基于大数据的网络安全态势感知及主动防御技术研究与应用 [J]. 计算机测量与控制, 2019, 27 (10): 229-233.
- [9] 朱益中. 广电网网络安全日志大数据分析的探索与实践 [J]. 中国传媒科技, 2019 (6): 115-117.
- [10] 奇安信科技. 网络边界智慧安全管控一体化解决方案 [EB/OL]. <https://www.qianxin.com/trade/detail/tid/26>.
- [11] Franke U, Brynielsson J. Cyber situational awareness a systematic review of the literature [J]. Computers & Security, 2014, 46: 18-31.
- [12] 张布. 开放大学教务管理系统的设计与实现 [D]. 长沙: 湖南大学, 2019.
- [13] 芦效峰, 付淞兵. 属性基加密和区块链结合的可信数据访问控制方案 [J]. 信息安全, 2021, 3 (3): 7-14.
- [14] BASS T. Intrusion detection systems and multisensor data fusion: creating cyberspace situational awareness [J]. Communications of the ACM, 2000, 43 (4): 99-105.
- [15] 谢凯, 代康. 基于负载预测的通信网络入侵检测系统设计 [J]. 计算机测量与控制, 2021, 29 (8): 62-66.
- [16] WANG Z G, LU Y, LI X. Active defense strategy selection of military information network based on incomplete information game [J]. ACTA Armamentarii, 2020, 41 (3): 608-617.
- [17] WANG J X, FENG Y, YOU R. Measuring network security by dependency relationship graph and common vulnerability scoring system [J]. Journal of Computer Applications, 2019, 39 (6): 1719-1727.
- [18] National Internet Emergency Center. Report on China internet network security in 2019 [M]. Beijing: Posts and Telecommunications Press, 2020.
- [19] 金志刚, 王新建, 李根, 等. 融合攻击图和博弈模型的网络防御策略生成方法 [J]. 信息安全, 2021, 21 (1): 1-9.
- [20] 王鹏, 胡宏彬, 李勇. 大数据融合模型的智能化网络安全检测方法 [J]. 计算机测量与控制, 2021, 29 (5): 40-44.
- [21] 杨润佳. 大数据驱动下主动防御网络安全性评估技术 [J]. 计算机测量与控制, 2018, 26 (10): 304-308.