Simulation Technology of Configurable Airborne Electromechanical Equipment Based on QEMU

Zhe-qi Zhu\(^1\), Li-hong Shang\(^1\) and Mi Zhou\(^2\)

\(^1\)College of Computer, Beihang University, Haidian District Beijing 100191, China
\(^2\)Reliable Avionics Technology Co. Ltd., Haidian District Beijing 100192, China

*zzq2014@buaa.edu.cn
*Corresponding author

Keywords: Airborne electromechanical equipment, QEMU, Configurable, Simulation technology

Abstract. Airborne electromechanical systems have the characteristics of high complexity, reliability and costs, therefore a lot of simulation is needed. In order to make software developers simulate a variety of devices and conduct semi-physical simulation or digital simulation under the same framework. In this paper, the virtual simulation technology is used to achieve the configurable simulation of airborne electromechanical devices, and simulation devices are used to build a simulation platform. In QEMU virtual machine the simulation of equipment front-end interface is realized in order to use real device driver, a thread is used to simulate the device function, finally, the data is transferred to data transmission module to simulate multiple data exchange methods of equipment. In the device function simulation module dynamic loading technology is used. Using C and Lua hybrid programming method to realize the equipment function of flexible and configurable. The experimental results show that only need to modify the configuration script the different simulation of airborne electromechanical equipment can be achieved.

Introduction

Airborne electromechanical system is an indispensable system of aircraft flight, with characteristics of high functional complexity, high reliability requirements and high cost. At present, the architecture of airborne electromechanical systems is moving towards centralized and distributed architecture, large-scale integration, standardization, generalization and reconfiguration. Therefore, the design of on-board electromechanical embedded control software emphasis on versatility and configuration more and more, this trend will lead to increasing complexity of software design and improving risk, so a large number of digital simulation, semi-physical simulation and physical verification are needed. Digital simulation is a pure software approach in system simulation. Semi-physical simulation is when conditions permit, physical equipment access system as much as possible to replace some of the mathematical model, in order to close to the actual situation and get more precise information. Physical verification is running the system on a real physical device for testing.

In order to enable software developers more convenient for semi-physical simulation or digital simulation to get rid of hardware dependence, products, location constraints during software debugging, an on-board electromechanical system simulation platform is needed. To construct such a simulation platform the system virtual machine will be used. At present, there are many types of virtual machines, such as Bochs, Virtual PC, VirtualBox\([1]\), VMware\([2]\), Xen\([3]\), QEMU\([4]\). Bochs can emulate Intel x86 CPUs, but it runs slowly. VMware is a commercial software that allows users to run multiple x86 virtual machines at the same time, but the guest operating system instruction code must be compatible with the host CPU. Xen can be paravirtualized and fully virtualized, paravirtualization performance loss less, but need to modify the system and can not support all operating systems; full virtualization needs through VT-X, AMD-V and other technologies, require special hardware support.
QEMU is a set of free software with high speed and cross-platform features to analog processor prepared by Fabrice Bellard. QEMU can simulate a variety of processor architectures, such as X86, POWERPC, ARM, MIPS. Support for user mode and full system mode which can simulate a complete system computer model, including CPU, memory, I/O devices. QEMU uses dynamic binary translation technology to simulate a variety of processor instructions, can meet the needs of a variety of embedded systems. QEMU is very suitable for the simulation of onboard electromechanical system hardware because it is open source, easy to expand and in high speed. QEMU provides a variety of devices, but does not provide GPIO, RS422, ARINC429, CAN and etc for the field of on-board electromechanical devices.

**System Overview**

As shown in Fig.1, Airborne electromechanical systems are usually composed of on-board electromechanical systems computer hardware and software. On-board electromechanical system software includes operating system, device driver, and application program. Airborne electromechanical systems Computer is usually made up of ordinary computer equipment and some airborne electromechanical system equipment, the majority of these devices are PCI devices, like RS422, GPIO, RS422, ARINC429, CAN and etc.

![Figure 1. The airborne electromechanical system structure.](image)

To achieve the on-board electromechanical system simulation platform need to simulate the airborne electromechanical system equipment firstly. In QEMU, the processor, memory, peripherals, bus are all implemented by simulation. In order to simulate a variety of devices under the same framework in simulation platform as well as semi-physical simulation or digital simulation. A configurable airborne electromechanical equipment simulation technology is proposed.

As shown in Fig.2, through analyzing airborne electromechanical system commonly used equipment, found that most equipment is PCI devices, and they have the similar structure. They are usually consist of the interface chip, shared memory region, CPU/FPGA, and the data transmission module.
Design and Implementation

The equipment model used in QEMU is QOM (QEMU Object Model) [5], which supply a device frame for user-definition. QOM has the model including bus, interface, equipment, etc. In QEMU, to simulate a PCI device need to use the interface which QEMU provides, including equipment registration, equipment initialization, the I/O mapping, base address register configuration and interrupt pin assignment, etc. In order to make simulation equipment easy to realize and simulation equipment is flexible and configurable. Using C and Lua hybrid programming method to achieve this goal. Lua is a small open source scripting language [6] and composed of standard C. Its code is simple and beautiful, the Lua script is easy to interact with C language, so this experiment adopts the Lua as implementation configuration script.

During device initialization, adding Lua state machine initialization, simulation function to Lua registration, loading device configuration information, device processor simulation thread initialization.

As shown in Fig.3, C program main thread supply APIs for Lua scripts for simulation equipment, such as shmgetdata(), shmsetdata(), iobuffergetdata(), iobuffersetdata(). For example, shmgetdata() is when the virtual machine application through simulation device driver to send data, the data is sent to shared memory area and release the semaphore, shemgetdata() acquire semaphore and set data to the Lua table. iobuffersetdata() is when the data is ready, the data will be sent out. Through these APIs, configurable CPU/FPGA simulation can be realized.
Experiments

Testing the time of an application of the virtual machine start coding the packet to another application. This time includes package encoding time, sending time, transfer time, receiving time and package decoding time. Setting a timestamp on the sending package, when the receiving application receive the package, taking out the timestamp and do subtraction with local timestamp to obtain delay.

![SEND AND RECEIVE PACKAGE DELAY](image)

As shown in Fig.4, the delay between send and receive package is about 8ms, this is because the data in the process of communication conducted many buffer and store-and-forward, so impact the performance.

Literature References

In recent years, some use QEMU for device emulation, external bus simulation technology based on QEMU, the CAN controller SJA1000 simulation[7], electronic module system parallel bus (QBUS) simulation, virtual trusted platform vTPM, etc.

Mainly in the external bus simulation technology based on QEMU is integrated embedded system interface test problems, through the system in the virtual machine create external bus simulation module, realize the communication between the virtual machine, this paper puts forward a simulation based on Ethernet communication external bus structure, in order to simulate CAN bus as an example.

In the CAN controller SJA1000 simulation experiment, it is to shorten the CAN network development cycle and reduce the cost. In order to realize the CAN network simulation test, this paper use QEMU virtual machine and SocketCAN under Linux environment simulation system is established.

Electronic module system parallel bus (QBUS) simulation is to solve the parallel development of software and hardware, and improve the efficiency of the kernel mode driver development and testing. The simulation equipment between QBUS was designed and implemented in this paper, communicating between simulation QBUS equipment and model physical devices.

Virtual vTPM trusted platform for trusted computer system security flaws in the process of trust chain. In QEMU, add support for the trusted platform module, use the seal of the virtual machine and isolation for trust chain transmission to provide a safe environment to make the transfer process of trust chain performance improve greatly.

Conclusions

This paper presents a unified simulation framework, in this context can achieve the system's all-digital simulation, semi-physical simulation. Simulation equipment in the system can also be through the configurable script to achieve a variety of functions of the simulation. In this paper, a
configurable simulation device is taken as an example. The purpose of this paper is to explore a
general, configurable and flexible simulation test environment to make the testing of on-board
electromechanical system software more flexible and convenient to expose the errors in software
early and reduce the problems of software. This paper solves the problem that the simulation
equipment is not flexible, the simulation equipment is not easy to be extended and the simulation
equipment is not flexible enough.

Acknowledgements
This research was financially supported by Beijing New-star Plan of Science and Technology(NO.Z161100004916091).

References
/QOM.
[6] Ierusalimschy R., Celes W., de Figueiredo L.H. The programming language Lua [J]. online
[7] Pisa P., Yang J., Sojka M. QEMU CAN Controller Emulation with Connection to a Host System
CAN Bus [J].