文章编号:1671-4598(2018)04-0066-04 DOI:10.16526/j.cnki.11—4762/tp.2018.04.018 中图分类号:V443 文献标识码:A

基于动态二进制翻译技术的数管软件虚拟测试环境设计

韦涌泉，张红军，董振辉，朱剑冰
（北京空间飞行器总体设计部，北京 100094）

摘要：针对传统的数管软件测试环境硬件设备依赖性强、地检设备多、监视点分散和软件内部状态跟踪困难等问题，设计和构建了基于QEMU模拟器的数管软件虚拟测试环境；该环境基于动态二进制翻译技术模拟星载处理器，实现在异构平台下运行星载数管软件，同时增加指令跟踪记录功能，帮助软件异常问题定位；融合外围设备驱动和终端应用功能，对数据流进行仿真，并采用统一的格式集中管理数据，进行分层处理和实时差异比对，以日志形式记录数据变化，实现对系统状态的单窗口监视；在高分四号卫星等型号的应用表明，文章设计的虚拟测试环境能够模拟数管软件运行环境，提供更多的调试手段，使软件测试工作提前，促进了软硬件协同开发，提高了星载软件的开发测试效率。

关键词：软件测试；星载软件；模拟器；测试环境

Design of OBDH Software Test Platform Based on Dynamic Binary Translation

Wei Yongquan, Wang Xianghui, Zhang Hongjun, Dong Zhenhui, Zhu Jianbing
(Beijing Institute of Spacecraft System Engineering, Beijing 100094, China)

Abstract: A QEMU-based virtual test platform for OBDH software was proposed to deal with the problems of traditional test environment, including hardware dependence, too many types of equipment, monitoring point dispersion and lack of software internal watching. The virtual cpu based on dynamic binary translation was emulated to run onboard software in heterogeneous platform, and the machine codes were recorded to help software debug; combined with the chip driver and application, the device was simulated to offer the information flow transmission, and the system log was record in order to achieve a single window monitoring by hierarchical processing and real-time difference comparing in unified format. The virtual environment was used in GF-4 project, the result showed that it can emulate the running environment of onboard software, provide more debugging means and simulate the system data stream, software test can be carried out before the hardware is put into production, so that the software and hardware can be coordinated developed, and the efficiency can be improved.

Keywords: software test; onboard software; emulator; test environment

0 引言

数管分系统（OBDH）是航天器的信息处理中心，分系统软件承担数据流控制和自主管理等功能[1]，是关键的星载软件之一，需要进行充分的测试验证。传统的数管软件测试环境由星上设备和地面测试设备组成[2]，包括：数管计算机、供电设备、专用地面测试设备、专用测试软件、总线仿真设备及其软件。这种实物测试环境可以为数管软件运行提供真实的平台，但存在以下几方面问题：1）硬件依赖性强，在星载计算机完成调试之前，无法进行星载软件的测试工作；2）地检设备多、研制周期长，存在地检设备延期交付影响软件测试进度的风险；3）测试软件分散，难以有效监视系统所有的数据流变化；4）测试过程中不能记录软件内部行为，不利于测试问题的排查定位。

为摆脱实物测试环境的限制，国外航天领域将虚拟测试环境应用在星载软件开发中。NASA委托Triakis公司开发了通用虚拟系统模拟器开发工具IcoSim[3]，使用IcoSim为数十个航空航天项目创建了虚拟测试环境，在虚拟环境上进行软件的测试验证工作。ESA将虚拟测试环境技术应用于欧空局地面控制中心[4]，利用虚拟机模拟真实目标机并运行星载软件，用于地面飞行任务训练；同时将模拟器应用于软件开发、确认测试和在轨维护中，其开发的SPARC模拟器TSIM在Ariane-5，PROBA-2等项目的软件研制中发挥了重要作用。

我国航天领域近年来对虚拟测试环境也开展了研究，主要集中在第三方评测领域[5]，仍处于起步阶段，没有形成通用的、有效的虚拟环境解决方案和虚拟环境产品。虚拟测试环境对星载软件调试手段的支持以及对新型星载处理器的支持尚不够完善。

本文针对传统数管软件实物测试环境的不足，设计和构建了基于动态二进制翻译技术的虚拟测试环境，基于广泛应用的QEMU模拟器，添加星载处理器的支持以及外围设备的模拟，补充目标码指令跟踪记录功能，对星载数据流集中仿真和实时监控，实现了在虚拟平台下完成数管软件的跟踪调试、测试工作，集中记录系统状态变化，解决了实物测试环境对软件调试、测试工作的制约，提高星载软件开发效率。

收稿日期:2018-01-15； **修回日期**:2018-02-22。

作者简介:韦涌泉(1986-),男,安徽临泉人,硕士研究生,工程师,主要从事航天器星载软件设计方向的研究。

1　动态二进制翻译技术及 QEMU 模拟器

动态二进制翻译技术是一种即时编译技术，它将源体系结构的二进制机器指令动态翻译为可在目的体系结构上运行的代码[6]。QEMU 是一款基于动态二进制翻译技术、开源的高性能模拟器[7-8]，能够支持包括 X86、ARM、MIPS 和 SPARC 等多种目标架构的模拟。

QEMU 作为一个用户进程运行于宿主操作系统之上，以基本块为单位进行二进制指令翻译执行。基本块是一组顺序执行的指令序列，以控制转移指令结尾，只有一个入口和一个出口。如图 1 所示，动态翻译器会把对基本块翻译和优化的结果缓存起来，再次执行该基本块时，就可以执行预存起来的翻译后的代码。模拟器开始执行时，首先从翻译缓存中查找对应的翻译块，为提高查找效率，QEMU 将最近使用的翻译块存放在一个哈希表中。如果基本块不在翻译缓冲中（该基本块尚未被翻译或已经从缓冲中移除），则启动基本块翻译过程。读取基本块目标指令，生成中间操作序列，翻译器将中间序列转换为宿主机器代码，并存放在翻译缓存中。物理处理器每执行完一条指令都会进行中断检查，对于 QEMU，每执行完一次基本块会进行一次中断响应检查。

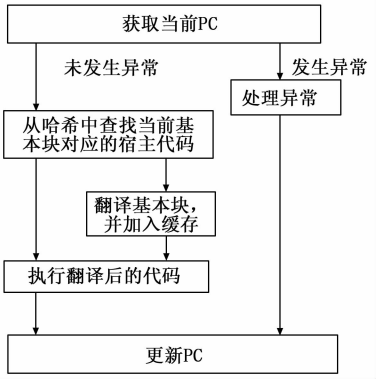

图 1　动态二进制翻译执行流程

2　虚拟测试环境设计

根据数管计算机硬件配置，在虚拟测试环境中实现星载数管软件运行所必需的环境支持，首先要解决在开发主机环境下运行星载处理器架构下可执行二进制代码的问题。本虚拟环境基于 QEMU 中的动态二进制翻译技术，将被测试系统的可执行二进制代码动态翻译为本地可执行的代码，在翻译过程中增加可执行代码的跟踪记录以增强调试功能；其次，实现数管计算机外部设备的虚拟仿真，为星载软件运行提供数据激励。对 1553B 总线等复杂的外部设备接口，采用数据解析的方法，融合终端应用功能，对数据流进行仿真；最后，采用统一存储、分层处理的方法对测试数据进行实时差异记录，提高星载软件测试状态的监视效率。

2.1　虚拟测试环境的组织结构

数管软件虚拟测试环境以 QEMU 模拟器为基础，包括星载处理器模拟模块、外围设备模拟及数据流仿真模块，数据监控模块以及指令跟踪记录模块，组织结构如图 2 所示。

星载处理器模拟模块是整个虚拟测试环境的核心，负责运行星载软件可执行二进制代码，将执行记录传到指令跟踪记录

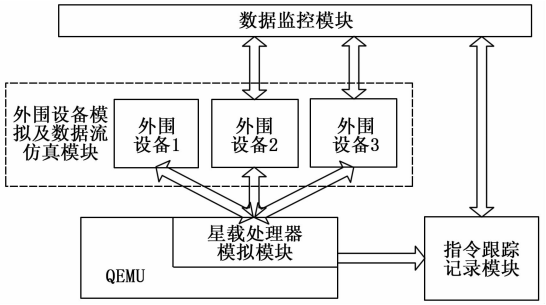

图 2　数管软件虚拟测试环境结构图

模块保存，同时通过中断和 IO 处理指令与外围设备进行数据交换；外围设备模拟及数据流仿真模块实现外设接口的模拟以及数据流的仿真，为数管软件运行提供数据激励，接收软件输出数据，并传到数据监控模块处理；数据监控模块负责对计算机所有输入输出数据进行存储分析，实时反映星载软件的状态变化，是星载软件测试的直接操作界面；指令跟踪记录模块提供处理器已执行指令的存储和查询功能，补充了 QEMU 的调试接口能力，为软件调试提供更多的手段。

2.2　星载处理器模拟及指令跟踪记录

本文所述在 QEMU 中增加星载处理器模拟的工作，主要包括对星载处理器特性参数的定义和系统寄存器的处理。根据动态二进制翻译技术的特点和星载数管软件运行的实际需要，星载处理器模拟模块只对物理处理器的部分功能进行仿真模拟。

QEMU 为每个支持的虚拟处理器设置了一组特性参数，包括处理器名称、IU 版本号、FPU 版本号、MMU 特性、寄存器窗口数量、以及指令集范围等，根据真实处理器情况定义星载处理器特征参数，并使用这些参数完成目标处理器动态翻译功能的初始化。

系统寄存器主要包括内存配置寄存器、IO 配置寄存器、中断配置和状态寄存器、RTC 寄存器等。其中，中断和 RTC 是星载软件运行的基础，模拟器必须实现；内存配置寄存器和 IO 配置寄存器控制处理器对内存和 IO 的访问时序，基于动态翻译技术的模拟器无法模拟精确的时序访问控制，因此在模拟器中不考虑此类寄存器配置数据对软件运行状态的影响。为方便星载软件开发人员确认星载软件向此类配置寄存器写入了正确的配置数据，所有对寄存器的访问都会被记录。

星载处理器模拟模块在系统环境中注册中断响应函数，用于响应外围设备提出的中断请求。星载处理器模拟模块综合分析中断未决寄存器、中断强制寄存器和中断屏蔽寄存器，判断虚拟星载处理器是否需要处理中断。当需要处理系统中断时，设置系统中相应的中断位，设置虚拟星载处理器下一步进入中断处理状态。RTC 是虚拟星载处理器运行的时间基准，为避免模拟器在不同环境下运行的速度差异对软件测试的影响，RTC 时间源以模拟器程序运行的时间为准，并为所有的外围设备模拟模块提供统一的时间基准。

如图 3 所示，虚拟星载计算机初始化包括处理器初始化、内存初始化、程序加载运行和外围设备初始化。处理器初始化完成处理器状态设置、寄存器初值设置以及中断申请、中断处理模块挂接。内存初始化完成内存的申请，设置内存大小和起

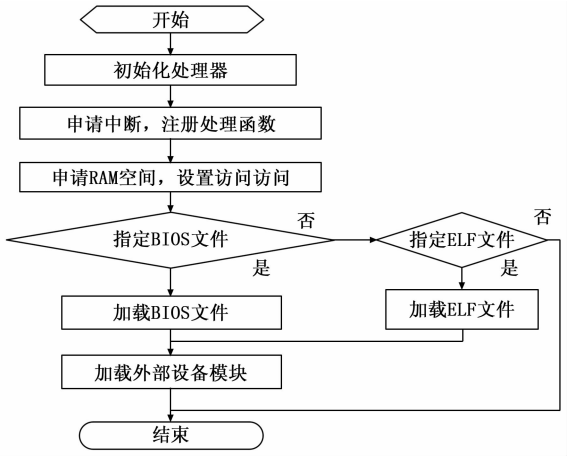

图 3 目标计算机初始化流程

始地址。程序加载有两种方式：一是加载 ELF 可执行文件，通过－kernel 参数指定，该方式包含调试信息，在利用 GDB 接口进行调试时是必需的；二是加载烧写在 PROM 中的程序二进制映像文件，通过－bios 参数指定。该方式与真实硬件环境下运行程序的状态完全一致。最后需要加载初始化各个外围设备驱动模拟模块，为了方便添加、修改外围设备模拟模块，实现具有不同外围设备的数管计算机模拟，本文所述设计改进了 QEMU 加载外围设备模拟模块的方式，采用动态共享库的方式加载外围设备模拟模块。

为全面准确记录星载软件的执行过程，提供更多的星载软件调试手段，本文所述设计在 QEMU 模拟器执行每个基本块时，记录该基本块的首地址，作为虚拟星载处理器目标码指令执行跟踪记录。每个基本块只使用 4 字节空间记录首地址，在内存中申请 10 MB 的存储空间进行循环缓存，实现记录多达数百万条历史指令的功能，远超过星载处理器调试单元的指令记录容量[9]。按顺序记录的历史指令为星载软件运行异常后，回查软件执行过程提供依据。在需要时，指令跟踪记录模块根据存储的基本块地址，在反汇编文件中获取汇编指令提供给软件调试人员。

2.3 外围设备模拟及数据流仿真

数管计算机的外围设备一般包括遥控遥测接口和总线接口。外围设备的模拟需要完成中断挂接和触发，根据设备状态对设备寄存器读写操作进行处理。

遥控遥测接口实现遥控数据的注入和遥测数据的下行功能。收到上行数据注入时，虚拟遥控设备模块向虚拟处理器提出中断请求，星载软件在中断处理过程中，读取相应的设备寄存器，触发模拟器调用外围设备模块对应地址寄存器的处理函数。虚拟设备模块在处理函数中向星载软件提供注入数据。

1553B 总线是国内航天器最常用的数据总线之一，通常承担平台数据传输任务[10]。根据数管软件访问总线接口的特点，虚拟总线模拟模块不仅实现了总线接口的模拟，还包括总线终端设备数据的仿真和处理。1553B 总线芯片寄存器多，设备存储区大，状态复杂。虚拟总线设备首先将芯片的寄存器区和内存区映射到内存，保存所有写入的配置数据和消息数据。虚拟模块监视配置寄存器值的变化，当发现配置寄存器的传输启动位被设置时，虚拟模块从映射的内存区中读出所保存的总线芯片栈寄存器地址、命令栈区内容和数据区内容，依次解析命令栈区的消息格式，根据设置的总线终端仿真数据内容重新设置数据区和命令栈区，实现 1553B 总线消息传输的仿真。

2.4 数据监控

作为整星的数据管理核心，数管分系统需要处理总线各终端设备的数据收发，遥控上注和遥测下传等多个数据流，对这些数据流的监控和分析是数管软件测试的主要工作。为解决传统数管测试环境中数据监视点分散的问题，防止数据漏判、错判，虚拟测试环境采用对所有数管软件输入输出数据进行分层处理、集中管理的方法，并实时记录数据的变化情况，如图 4 所示。

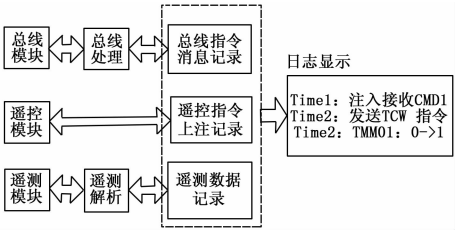

图 4 数据监控模块

总线传输数据，遥控注入数据和遥测下传数据由不同的虚拟外围设备产生或接收，经过对原始数据进行必要的解析处理后，添加时间信息，进行统一存储。各项数据记录在存储时与前一次的内容进行比对，形成差异日志，实时显示。本文所述设计在日志监视窗口中按时间顺序提示数管软件遥控数据接收情况、遥测数据变化情况以及总线数据变化情况，实现在单一窗口界面下监视数管软件状态变化过程，有效减少系统状态漏判、错判的风险。统计总线遥测采集周期、遥测源包下传周期等测试任务也可以通过对统一存储数据的查询获取。

3 虚拟测试环境的验证与应用

使用高分四号卫星数管软件对本文设计的虚拟测试环境进行了验证测试。表 1 列出了虚拟测试环境与实物测试环境的对比情况。首先在开发主机 Linux 操作系统中运行虚拟测试平台，加载高分四号数管软件映像；然后，通过虚拟遥控接口上注数管指令，由数据监视模块记录下数管软件的状态变化；最后，对比实物测试环境下数管软件运行结果与虚拟环境下软件运行结果是否一致。图 5 是在虚拟测试环境中进行高分四号卫星数管软件指令组功能测试的情况。验证结果表明，数管软件在虚拟测试环境下的运行结果与实物环境下的结果一致。

表 1 虚拟测试环境与实物测试环境对比

对比项目	虚拟测试环境	实物测试环境
星载设备	0 台	1 台
地面测试设备	1 台普通 PC	2 台普通 PC 和 2 台专用地检
软件加载时间	2s 以内	2～3 分钟
调试手段	串口打印、单步调试和指令记录等	串口打印
监视界面	1 个	4 个

本文设计的虚拟测试环境已经应用在高分四号卫星、火星着陆巡视器等型号的星载软件研发中。在型号中的应用表明，

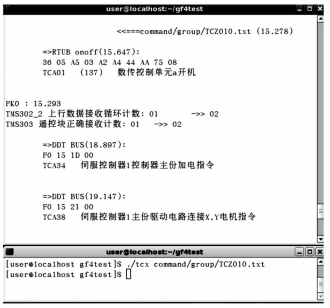

图 5　虚拟测试环境下数管软件测试界面

能够使用虚拟测试环境完成星载软件的运行、测试工作，实现对星载数据流的仿真和单窗口监视。同时，虚拟测试环境提供更多的调试手段，为星载软件问题定位分析提供帮助。虚拟测试环境的应用，可以避免物理设备不能及时到位对软件开发进度的影响，实现星载软件提前测试和软硬件协同开发，加快了软件研制进度，提高了软件开发效率。

4　结论

针对实物测试环境对数管软件开发测试的制约问题，设计并实现了虚拟集成测试环境。该环境基于动态二进制翻译技术，实现对星载计算机以及外围设备的模拟和数据流的仿真监视，具备目标码指令跟踪记录以及单窗口监视系统状态变化等功能。相对于传统的实物测试环境，该虚拟测试环境具有不依赖硬件设备、数据流仿真和监视便捷、调试手段丰富等优点，已经应用在高分四号、火星着陆巡视器等型号的数管软件开发中，提高了软件的开发效率。该设计方案可以推广到其他星载软件或嵌入式软件的虚拟测试仿真，也可应用于飞控协调演练等活动，具有一定的推广价值。

参考文献：

[1] 刘　鑫，韦涌泉，冯国平，等. 高分四号卫星数管分系统设计及在轨验证 [J]. 航天器工程，2016，25 (1)：93－98.
[2] 郭　坚，付连芳，翟君武. 一种星载软件系统测试环境的设计 [J]. 计算机测量与控制，2005，13 (5)：499－502.
[3] Bennett T L，Wennberg P W. The Use of a Virtual System Simulator and Executable Specifications to Enhance Software Validation，Verification，and Safety Assurance—Final Report [R]. Fairmont，West Virginia：NASA IV&V Facility，2004.
[4] Pidgeon A，Dawe G，Dartnell A. Software emulators：A virtual processor to support training simulations [R]. ESA Publications Division，2002.
[5] 郭向英，盛　庄，张西超. 基于 VTEST 的 TMS320C3x 指令集模拟器设计 [J]. 计算机工程与设计，2013，34 (6)：1973－1976.
[6] 张西超，郭向英，赵　雷. TCG 动态二进制翻译技术研究 [J]. 计算机应用与软件，2013，30 (11)：34－37.
[7] 李可生，杨　博，徐天伟，等. 基于 QEMU 的可重构专用处理器模拟器实现 [J]. 计算机工程与设计，2016，37 (5)：1335－1339.
[8] 鲍颖力. 基于虚拟机 QEMU 的嵌入式全系统仿真测试环境的研究与实现 [J]. 航空电子技术，2011，42 (4)：33－37.
[9] 张　鹏，樊晓桓，黄小平. 基于总线访问的片上调试方法研究 [J]. 计算机测量与控制，2014，22 (2)：510－512.
[10] 刘燕松，盛利元，冯旭哲. 卫星载荷 1553B 总线数据接口的设计与实现 [J]. 宇航计测技术，2013，33 (5)：52－56.

(上接第 65 页)

用户通过使用身份工作卡，进行身份识别，选择想要借出的设备，系统会打卡设备所在的位置，并对借出人员的信息和借出时间进行登记，并对其内容进行保存，可被服务器进行查询访问。同时归还时，系统会记录归还人员的信息和时间，对校准时间进行登记和校准时间到期提醒，以及识别归还的设备是否正确，并将所有的过程记录到数据中。

3.4　软件工作流程

上述的各软件系统均运行在各自的计算机上，在测试系统工作后，要保证整体的运行，无线电设备控制软件和电子工艺卡软件需相互配合，共同完成整个测试流程，流程图如图 4 所示。

软件工作流程中，无线电设备控制软件和电子工艺卡软件打开后，首先都是用户登录，验证用户身份，确认当前用户是否具有对系统的操作权限，如果用户验证失败，系统仍处于登陆状态，也可以选择取消退出登录界面；登录成功后，无线电设备控制软件则向子工艺卡软件发联机请求命令，如子工艺卡软件准备好，给无线电设备控制软件回联机确认信息；随后，工艺卡软件向无线电设备控制软件发送自检命令，控制各仪器设备调用自检功能模块分别完成本身自检，并接收各测试资源的自检结果，把自检结果信息上传给工艺卡软件；接下来无线电设备控制软件可以进行系统配置，配置完成，进入相关测试任务的执行，这里由电子工艺软件下发测试的 TPS，具体的 TPS 执行将下发到相应的测试仪器设备中完成，并由仪器设备上传测试数据，由电子工艺软件完成数据分析与结果记录，并生成报表打印。

5　结论

飞机无线电设备总装测试系统，采用了先进的测试技术及模块化和标准化的仪器设备，该设备已经在某飞机总装车间使用，用户反映系统操作方面、测试准确、结构布局合理，功能完善，大幅度提供了飞机生产效率；该飞机无线电设备总装测试系统是一套具可靠性高、性价比优良，实用性强、操作简单、维护方便并具备扩展性和升级能力的测控系统。

参考文献：

[1] 欧阳寰，王超勇，陈遵银，等. 某型飞机机载计算机检测仪测试软件设计 [J] 计算机测量与控制，2017，25 (8)：116－119.
[2] 欧阳寰，王超勇，韩兆福，等. 某型飞机武器控制系统计算机检测仪的设计与实现 [J]. 计算机测量与控制，2016 (12)：56－62.
[3] 张　超，孙元亮. 飞机移动装配线生产管理系统研究 [J]. 航空制造技术，2014 (1)：71－74.
[4] 刘　洋，陈雪峰. 飞机无线电导航设备自动测试系统设计 [J]. 现代电子技术，2014 (19)：99－102.
[5] 马明建. 数据采集与处理技术 [M]. 西安：西安交通大学出版社，1998.
[6] 张红梅，徐启功，徐贵水，等. 虚拟仪器在航空仪器检测中的应用 [J]. 空军工程大学学报：自然科学版，2003，4 (2)：70－73.
[7] 霍朝晖，谭杨森，祁　春. 飞行试验机载关键参数快速处理系统设计 [J]. 现代电子技术，2013，36 (5)：121－124.