基于 QEMU 数字样机的多维度故障注入技术研究

侯龙伟[1] 陈 聪[1] 徐 海[1]
HOU Longwei CHEN Cong XU Hai

摘 要 高可靠领域嵌入式系统对容错能力与故障应对机制有着严苛的测试要求，传统故障注入测试方法存在硬件依赖性强、实施成本高、灵活性不足、故障覆盖维度单一、试验结果可信度低、难以工程化实施等诸多局限。为此，文章提出一种基于 QEMU 全数字仿真数字样机的多维度故障注入技术方案，依托 QEMU 数字样机实现硬件设备的数字化建模，构建覆盖寄存器、内存、总线协议、总线负载四个维度、贯穿物理层至应用层的多层次的故障注入测试系统，无需依托硬件设备即可完成嵌入式系统故障场景模拟与可靠性验证。该系统具备注入精度高、场景可定制、部署便捷、兼容性强等优势，能够有效提升嵌入式软件的容错能力与运行可靠性。

关键词 嵌入式软件测试；故障注入；数字仿真；QEMU；QMP

doi：10.3969/j.issn.1672-9528.2026.08.004

0 引言

故障注入技术是一种面向高可靠系统的主动式测试验证技术，其核心逻辑是通过人为可控的技术手段，向被测试系统注入特定故障或异常状态，加速被测系统故障触发或系统失效，进而观察和分析被测系统在故障条件下的行为反馈、容错机制有效性与故障恢复能力[1]。该技术是安全关键领域嵌入式软件测试的核心手段，尤其针对环境异常等造成的硬件整体或部分瞬时失效的极端工况，故障注入能够在实验室环境下复现嵌入式软件在异常条件下的运行场景，验证软件对故障的检测、隔离和恢复能力，从而提高软件的容错性与可靠性，降低软件在实际运行过程中的失效风险。

按照注入方式不同，故障注入技术分为基于硬件的故障注入、基于软件的故障注入和基于仿真的故障注入三类典型方向[2]。基于硬件的故障注入通过改造硬件设备，搭建专用故障注入线路实现故障模拟，该方法直接作用于物理硬件，真实性高，但存在成本高昂、硬件损伤风险大、故障注入过程不可控等缺点，难以大规模应用于工程化测试。基于软件的故障注入技术通过修改软件源码、注入钩子函数实现故障模拟，无需额外的硬件设备，但该方法对被测软件的侵入性强，可能改变软件原有的运行逻辑，甚至引入额外的故障风险，且难以模拟底层硬件故障[3]。基于仿真的故障注入以系统级仿真平台为基础，将故障注入目标从真实硬件迁移至仿真平台，兼具硬件级故障模拟能力与软件故障注入的灵活性，具有开发成本低、注入过程可控、无硬件损伤等优势，是实现多维度故障注入的理想载体[4][5]。

QEMU（quick emulator）是一款开源的跨平台全系统仿真工具，其核心基于动态二进制翻译（TCG）技术，可在宿主机上高效模拟 x86、ARM、PowerPC、RISC-V 等多种主流处理器指令架构[3]，并对完整计算机系统的 CPU、内存、总线、外设等硬件设备进行精细化建模。得益于其对目标硬件的高保真模拟，嵌入式软件可不经修改直接在 QEMU 数字样机环境中运行，无需额外适配或修改源码。同时，QEMU 支持行为的全链路可观测、可编程控制与可复现执行，为嵌入式软件的故障注入测试提供高保真、低侵入的数字样机环境。

因此，本文以 QEMU 数字样机平台为基础，开展多维度故障技术研究，构建覆盖寄存器、内存、总线协议、总线负载的全层级故障注入与验证系统，解决传统方法维度单一、可控性差、结果可信度不足的缺点，为航空航天嵌入式软件的可靠性验证提供一套高效、低风险、可工程化的解决方案[6]。

1 多维度故障注入模型构建

基于分层故障注入理念，并结合 QEMU 数字样机特点，构建多层级、多维度、全时序的故障注入技术模型，技术模型如图 1 所示。

1.1 故障注入时机模式

故障注入时机定义了故障生效在时间维度的控制方式，主要分为以下六类：

1. 中国航空工业集团公司西安航空计算技术研究所
陕西西安 710068

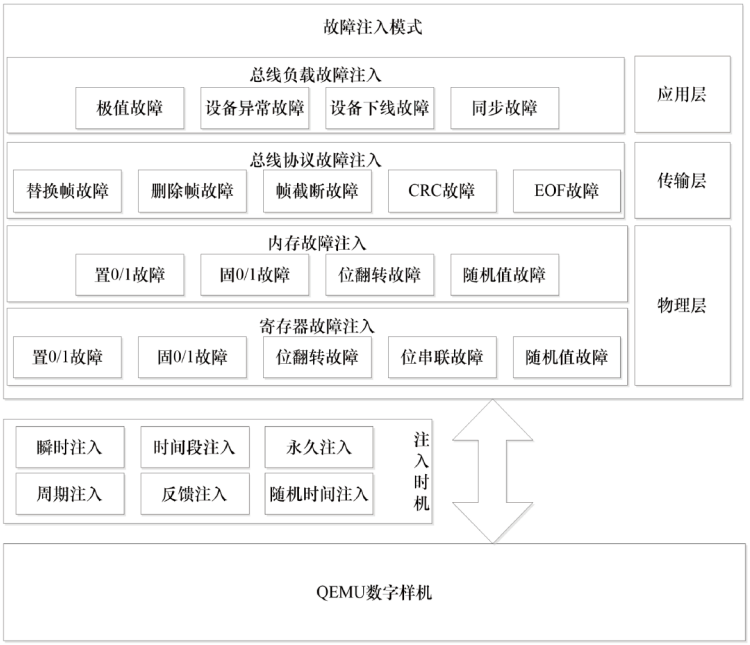


图 1 多维度故障注入模型

（1）瞬时注入：表示该故障在配置的特定时间点 T_0 生效，用于模拟瞬时扰动；

（2）时间段注入：表示该故障在配置的时间区间 [T_0, T_n] 内生效，用于模拟短时异常；

（3）永久注入：表示该故障在配置的 [T_0, ∞）时间段内持续生效，用于模拟永久性故障；

（4）周期注入：表示该故障在配置的 T_0、$T_{0+1}\times P$、$T_0+2\times P$、…、$T_0+n\times P$ 等周期性时间点重复生效，用以模拟周期性故障；

（5）反馈注入：表示该故障在接收到数字样机特定反馈信号时触发生效，用于模拟事件驱动型故障；

（6）随机时间注入：表示该故障在配置的随机时间点 T_0=Random() 生效，用于模拟随机扰动场景。

1.2 故障注入模式

1.2.1 寄存器故障注入模式

针对处理器 / 外设寄存器的硬件级故障，定义以下寄存器故障注入模式：

（1）置 0/1 故障：将目标设备目标寄存器的指定位，强制设置为逻辑 0 或逻辑 1；

（2）固 0/1 故障：将目标设备目标寄存器的指定位，固定设置为逻辑 0 或逻辑 1；

（3）位翻转故障：将目标设备目标寄存器的指定位，强制翻转（0 变 1 或 1 变 0）；

（4）随机值故障：将目标设备目标寄存器的指定位，设置为随机生效的逻辑值；

（5）位串联故障：将目标设备目标寄存器的指定位与该设备其他寄存器的某些位进行关联，模拟寄存器间的跳线耦合。

1.2.2 内存故障注入模式

针对存储设备内存地址的硬件级故障，定义以下内存故障注入模式：

（1）置 0/1 故障：将目标存储设备指定地址的指定位，强制设置为逻辑 0 或逻辑 1；

（2）固 0/1 故障：将目标存储设备指定地址的指定位，固定设置为逻辑 0 或逻辑 1；

（3）位翻转故障：将目标存储设备指定地址的指定位，强制翻转（0 变 1 或 1 变 0）；

（4）随机值故障：将目标存储设备指定地址的指定位，设置为随机生效的逻辑值。

1.2.3 总线协议故障注入模式

针对通信总线的协议层故障，定义以下总线协议故障注入模式：

（1）替换帧故障：将原帧的指定部分内容替换为自定义数据，模拟协议数据被篡改的故障；

（2）删除帧故障：删除总线传输中的指定帧，模拟帧丢失故障；

（3）截断帧故障：将原帧从指定偏移位置截断，仅保留偏移前的帧内容，模拟帧传输不完整故障；

（4）CRC 故障：将原帧中的 CRC 校验字段修改为自定义值，模拟校验错误故障；

（5）EOF 故障：将原帧中的 EOF 结束字段修改为自定义值，模拟帧结束标识异常故障。

1.2.4 总线负载故障注入模式

针对数据总线的应用层负载故障，定义以下注入模式：

（1）极值故障：将数据负载字段中某些代表设备运行参数（如温度、压力等）的字段，设置为超出正常工作所能允许的极限值而引发的故障；

（2）设备异常故障：将数据负载字段中某些代表设备运行状态的字段，设置为与正常运行状态不符的情况进而引发的故障，如性能下降、功能失常、过压、过流等；

（3）设备下线故障：将数据负载字段中的心跳字段，设置为与预期不符合的情况进而引发的故障；

（4）同步故障：将数据负载字段中代表设备同步状态的字段，设置为异常数据进而引发的故障。

1.3 故障注入多维度建模

1.3.1 寄存器故障注入模型

基于寄存器属性、故障模式与注入时机，构建寄存器故障注入一元组模型 RFI：

RFI={id,dstDevice,dstReg,mode, triggerTime, value}

id：寄存器故障注入唯一标识，用于场景追溯与用例管理；

dstDevice：故障注入的目标设备，可为 QMUE 数字样机模型中的 CPU 或者外设元器件；

dstReg：故障注入的目标寄存器；

mode：寄存器故障注入模式，对应 1.2.1 节定义的寄存器故障模式；

triggerTime：故障注入触发时机，对应 1.1 节定义的注入时机类型；

value：寄存器故障注入的目标值，用于指定故障生效后的寄存器状态。

1.3.2 内存故障注入模型

基于内存属性、故障模式与注入时机，构建内存故障注入一元组模型 MFI：

MFI={id,dstDevice,injectAddress,mode,triggerTime, value}

id：内存故障注入唯一标识，用于场景追溯与用例管理；

dstDevice：故障注入的目标设备，可为 QMUE 数字样机模型中的任意存储单元；

injectAddress：故障注入的目标内存地址，需位于设备的有效地址区间内；

mode：内存故障注入模式，对应 1.2.2 节定义的内存故障模式；

triggerTime：故障注入触发时机，对应 1.1 节定义的注入时机类型；

value：内存故障注入的目标值，用于指定故障生效后的内存状态。

1.3.3 总线协议故障注入模型

根据总线协议属性、故障模式与注入时机，构建总线协议故障注入一元组模型 BPFI[7-10]：

BPFI={id,busType, mode, triggerTime, frameContent}

id：总线协议故障注入唯一标识，用于场景追溯与用例管理；

busType：总线协议故障注入的总线类型，需与 ICD 定义以及 QEMU 数字样机中的总线设备类型一致；

mode：总线协议故障注入模式，对应 1.2.3 节定义的总线协议故障类型；

triggerTime：故障注入触发时机，对应 1.1 节定义的注入时机类型；

frameContent：总线协议故障注入的目标帧内容，包含完整的帧头、数据负载与帧尾。

1.3.4 总线负载故障注入模型

根据总线负载故障注入的特点、故障模式与注入时机，构建总线协议故障注入一元组模型 BDFI：

BDFI={id, busType, mode, triggerTime, frameContent}

各字段含义同总线协议故障注入，重点针对数据负载中的应用数据异常。

2 故障注入系统模块设计

基于前文构建的多维度故障注入模型和 QEMU 数字样机，本文采用前后端分离架构完成故障注入系统整体设计，系统架构设计如图 2 所示，划分为故障注入激励软件与故障注入 QEMU 数字样机，前后端之间基于 QMP 命令通信协议实现实时数据交互与指令传输，最终搭建可覆盖寄存器、内存、总线协议、总线负载四大维度的嵌入式软件故障注入系统。

2.1 故障注入激励软件

故障注入激励软件的核心目标是实现故障注入流程的可视化配置与故障影响的量化分析，主要包含故障注入配置模块与故障反馈结果展示模块两部分，为故障注入过程提供全流程可视化管控能力。

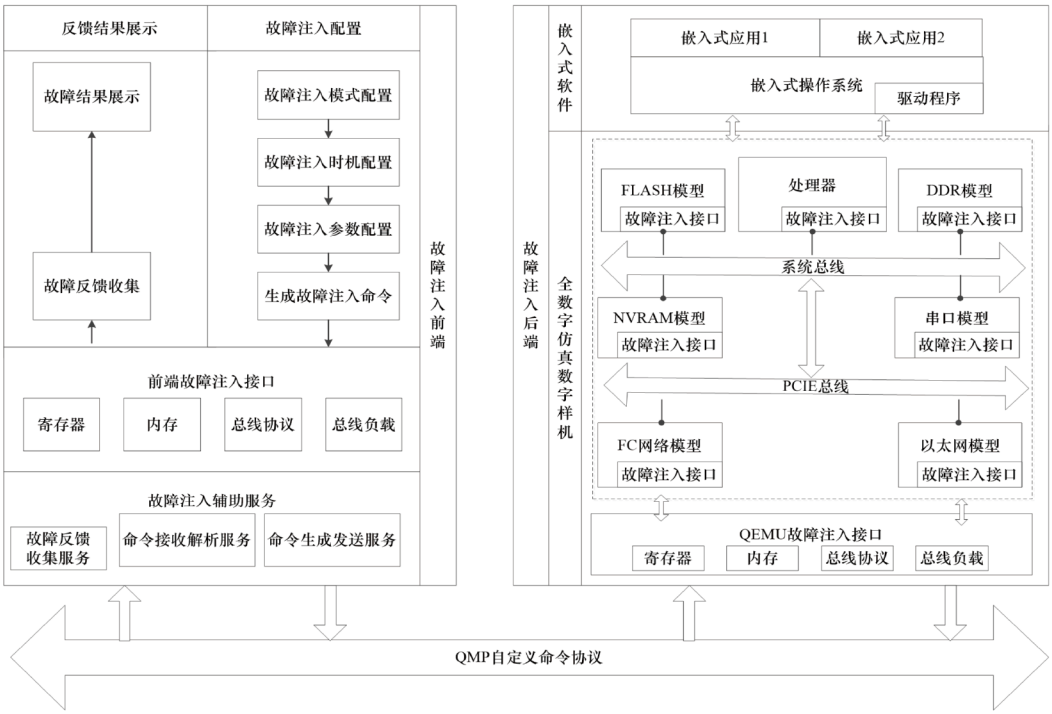


图 2 故障注入设计结构

故障注入配置模块包含故障注入模式配置单元、故障注入时机配置单元、故障注入参数配置单元以及故障注入命令生成与发送单元，实现从故障定义到指令生成的完整流程。故障注入模式配置界面支持用户选取寄存器、内存、总线协议、总线负载四类故障注入维度，并匹配对应维度下的具体故障模式；故障注入时机配置单元提供瞬时、随机、周期、反馈等多种注入时机配置能力，保障故障注入时序的可控性与可复现性；故障参数配置单元支持故障注入目标设备、作用位域、故障目标值等关键参数的配置，实现故障注入场景的精准定义；故障注入命令生成发送单元基于用户配置的故障参数，生成符合 QMP 命令协议规范的故障注入指令包，并按配置的注入时机发送至 QEMU 数字样机。

反馈结果展示模块包含故障注入命令接收解析服务单元、故障反馈收集单元和故障结果可视化单元，实现对故障注入执行过程的监控与结果的可视化呈现。故障注入命令接收解析服务单元接收 QEMU 数字样机反馈的故障注入执行回执，解析指令执行状态码、错误码等信息，完成故障注入指令交互的闭环管控；故障反馈收集服务单元通过解析 QMP 回执、串口输出、仿真日志、系统快照等多源数据，提取故障注入后的系统行为特殊数据；故障结果可视化单元将故障反馈收集服务单元输出的系统数据进行结构化展示，为故障影响分析提供直观的数据支撑。

2.2 故障注入 QEMU 数字样机

由于原生的 QEMU 数字样机并不支持故障注入，因此 QEMU 数字样机需针对故障注入场景进行系统性改造，改造工作分为 QEMU 设备模型故障注入能力扩展与 QMP 命令协议故障注入命令支持两部分，实现故障注入指令的解析、执行与反馈闭环。

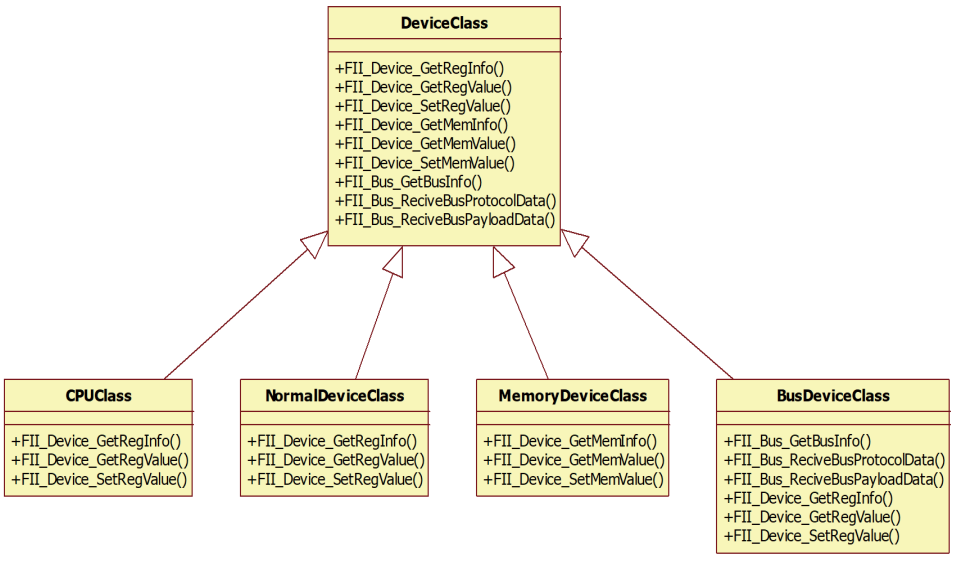


图 3 故障注入接口继承关系

2.2.1 设备模型故障注入能力扩展

QEMU 通过对象模型（qemu object model, QOM）实现 CPU、外设和总线模型的统一管理，所有的模型均继承自 DeviceClass 基类。基于该特性，本文在 DeviceClass 中扩展寄存器故障注入、内存故障注入、总线协议故障注入以及总线负载故障注入四类通用接口，并在对应设备模型中实现接口逻辑。各类设备的接口定义如图 3 所示，各接口具体功能如表 1 所示。

表 1 故障注入接口列表

故障注入接口名称	故障注入接口说明
FII_Device_GetRegInfo	查询设备所有寄存器信息
FII_Device_GetRegValue	查询设备指定寄存器值
FII_Device_SetRegValue	设置设备指定寄存器值
FII_Device_GetMemInfo	设置内存设备信息
FII_Device_GetMemValue	查询内存设备指定地址的值
FII_Device_SetMemValue	设置内存设备指定地址的值
FII_Bus_GetBusInfo	查询总线设备信息
FII_Bus_RecvBusProtocol	接收总线协议故障注入帧
FII_Bus_RecvBusPayload	接收总线负载故障注入帧

2.2.2 QMP 协议故障注入能力扩展

原生 QEMU 的 QMP 命令处理模块未定义故障注入相关指令的处理流程，需通过自定义扩展实现指令的解析与执行。具体实现流程如下：

（1）命令定义：在 qapi 文件夹下添加故障注入命令描述文件 fii_cmd.json，定义四类故障注入命令的格式与参数。

（2）框架生成：将 fii_cmd.json 引入 meson.build 与 qapi-schema.json 配置文件，QEMU 在构建后将自动生成故障注入命令的接收、转发框架代码以及 fii_cmd_parse() 命令处理函数。

（3）逻辑实现：在 fii_cmd_parse() 函数中实现故障注入命令的解析逻辑，根据目标设备、故障类型、目标位域等参数，调用对应设备模型的故障注入接口，完成故障值的写入操作，并在完成故障注入后，向故障注入激励软件返回执行码。

以寄存器故障注入为例，指令执行流程如图 4 所示。

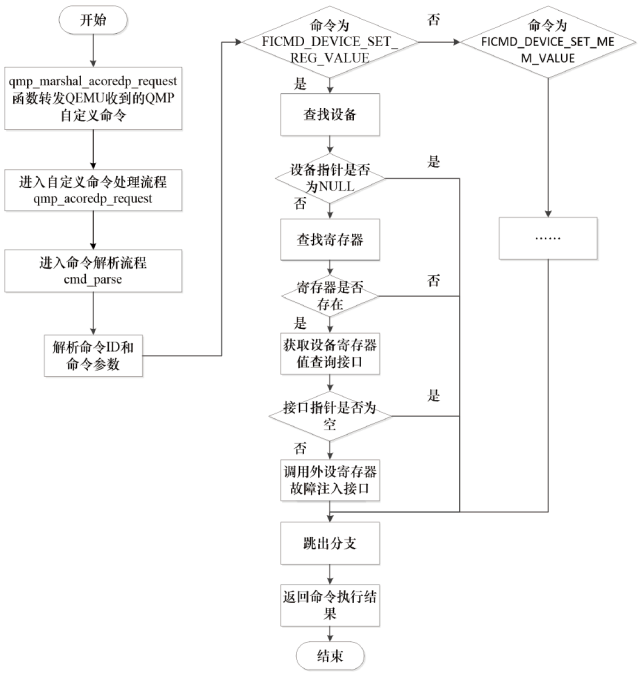


图 4 寄存器故障注入执行流程

2.3 故障注入命令扩展

故障注入激励软件与故障注入 QEMU 数字样机使用 QMP 命令通信协议实现实时数据交互与指令传输，QMP（QEMU Monitor-Protocol），即 QEMU 监视器协议，是 QEMU 所提供、基于 JSON 格式的标准化命令交互接口，支持外部工具对 QEMU 虚拟机实例的状态查询、运行控制与扩展指令执行，且具备良好的用户扩展能力，能够满足故障注入场景的定制化交互需求。

基于故障注入的场景需求，本文对 QMP 协议进行扩展，定义了五类故障注入命令，包括寄存器故障注入命令、内存故障注入命令、总线协议故障注入命令、总线负载故障注入命令以及系统查询命令。各命令定义如表 2 所示。

表 2 故障注入相关命令定义

故障注入命令名称	故障注入命令说明
FICMD_DEVICE_GET_REG_INFO	查询设备所有寄存器信息命令
FICMD_DEVICE_GET_REG_VALUE	查询设备指定寄存器值命令
FICMD_DEVICE_SET_REG_VALUE	设置设备指定寄存器值命令
FICMD_DEVICE_GET_MEM_INFO	查询内存设备信息命令
FICMD_DEVICE_GET_MEM_VALUE	查询内存设备指定地址的值命令
FICMD_DEVICE_SET_MEM_VALUE	设置内存设备指定地址的值命令
FICMD_BUS_GET_BUS_INFO	查询总线设备信息命令
FICMD_BUS_RECV_BUS_PRT_DATA	接收总线协议故障注入帧命令
FICMD_BUS_RECIV_BUS_PLD_DATA	接收总线负载故障注入帧命令
FICMD_DEVICE_GET_DEVICE_LIST	查询设备列表信息命令
FICMD_DEVICE_GET_MEM_LIST	查询存储设备列表信息命令
FICMD_DEVICE_GET_BUS_LIST	查询总线设备列表信息命令

以寄存器故障注入命令为例，故障注入命令格式如图 5 所示。

```
{
    "execute":"ACoreDP_Request",
    "argument":{
        "command":{
            "commandID":"4",
            "commandName":"setRegValue",
            "commandTime":"2024-04-02:08-44",
            "commandVersion":"V 1.0.0"
        },
        "requestParam":{
            "devName":"device_ft2000_uart1",
            "devPath":"device_systemt_bug/device_ft2000_uart1",
            "regName":"RD",
            "regValue":"0x000000F0",
        }
    }
}
```

图 5 寄存器故障注入 QMP 命令

3 实验

为验证本文提出的多维度故障注入技术的有效性，选取寄存器、内存、总线负载三类典型故障场景进行测试，试验基于 QEMU 构建的全数字仿真平台，实验的软硬件环境如表 3 所示。

表 3 实验环境配置

目标项	配置
宿主机处理器	Intel Core5 2.2 GHz
宿主机操作系统	Windows 11
QEMU 模拟的目标架构	FT2000
QEMU 运行的操作系统	天脉 ACoreOS1 操作系统
QEMU 运行的嵌入式应用	某型飞机飞行控制软件

寄存器故障注入：选择 FT2000 目标板的串口设备“ft2000_uart1”，目标寄存器选择“UARTCR 控制寄存器”，注入时机选择瞬时注入，寄存器原值为 0X000000081，修改第 8 位从 1 设置为 0（串口输出使能标志，1 表示使能，0 表示禁用），即故障值为 0X000000001，从图 6 可以看出，寄存器值已变化，且 QEMU 的串口输出暂停。



图 6 寄存器故障注入试验结果

内存故障注入：选择 FT2000 目标板的 flash 设备“mach-virt.flash3”，将地址 0X00000000~0X0000000F 中每个字节的值从 0X0000 改为全 0XCFCF，注入时机选择瞬时注入，从图 7 可以看出，内存值已变化。

总线负载故障注入：选择 AFDX 数据总线以及某型飞机飞行控制软件 ICD 的文件，注入目标为“燃油温度”，故障模式选择“极值故障”，注入值为“105 ℃”（正常范围为

0~80 ℃），注入时机选择永久注入，从图 8 可以看出，“燃油温度”已变化，且系统已发出告警信息。



图 7 内存故障注入试验结果



图 8 总线负载故障注入试验结果

根据上述三类故障注入类似的操作流程，测试其余寄存器、内存和总线负载故障模式，实验结果如表 4 所示。

4 结语

本文针对嵌入式软件测试中传统故障注入方法硬件依赖性强、实施成本高、灵活性不足、故障覆盖维度单一、试验结果可信度低、难以工程化实施等缺陷，提出并实现了一种基于 QEMU 数字样机的多维度故障注入系统。该方案通过二次扩展 QEMU 仿真内核，实现寄存器、内存、总线协议、总线负载四层故障注入接口，统一时序调度模型支撑瞬时、周期、永久等多类故障触发逻辑，完整覆盖硬件底层至应用层典型失效场景。

系统已落地应用于某型飞机飞控软件可靠性验证工程，在验证工程中完成四大类故障模式与实际飞控软件故障映射后，可在硬件设计图纸阶段和软件研发阶段启动故障仿真，无需硬件实物支撑实现“软件测试前置”，提前排查飞控软件的 BIT 检测、故障隔离、失效恢复等逻辑缺陷，有效减少型号项目后期硬件修改迭代次数。并在完成自动化部署改造后，支持自动化测试调度，持续提升嵌入式软件可靠性和稳定性。

表 4 寄存器故障注入实验结果

注入部位	注入类型	注入时机	注入值	注入现象
FT2000 处理器 Timer2 定时控制寄存器	寄存器故障注入	周期注入，周期为 100 ms	0X00000000	定时器中断周期失效，飞控 100 ms 周期任务执行报错
FT2000 处理器 Timer2 定时控制寄存器	寄存器故障注入	周期注入，周期为 100 ms	0X00000000	定时器中断周期失效，飞控 100 ms 周期任务执行报错
AFDX 总线控制器 INT 中断屏蔽寄存器	寄存器故障注入	瞬时注入	0XFFFF0000	总线中断程序停止执行，飞管软件无法接收命令或数据
ADC 模数转换外设数据寄存器	寄存器故障注入	永久注入	0X7FFF0000	飞管软件采集的高度值持续报数据异常错误
FLASH 代码地址段 0X00104000	内存故障注入	永久注入	0X00000000	空地状态异常，飞机无法进入空中状态
FLASH 代码地址段 0X00200080	内存故障注入	永久注入	0XFFFFFFFF	飞管程序偶现运行崩溃
起落架状态负载字段	总线负载故障注入	永久注入	0X00000001	空地状态异常，飞机始终处于地面状态

参考文献：

[1] 刘宪忠，赵昶宇．嵌入式系统故障注入技术研究 [J]. 科技与创新，2022(7):25-27.

[2] 苏银科，李艳雷，周平，等．基于故障注入的实时嵌入式软件仿真测试技术研究 [J]. 导航定位与授时，2014,1(3):69-73.

[3] 方加娟，李婉晴．基于软件故障注入的系统运行鲁棒性测试方法 [J]. 工业控制计算机，2025,38(8):121-123.

[4] 徐军，徐萍．面向 BIT 验证的存储器故障模拟及注入方法研究 [J]. 计算机测量与控制，2013,21(8):2027-2029.

[5] 李毅，徐萍，万寒．基于 QEMU 实现的处理器类故障模拟与注入方法研究 [J]. 计算机工程与科学，2014,36(1):19-27.

[6] 中国航空工业集团公司西安航空计算技术研究所．一种针对基于 QEMU 的全数字数字样机的故障注入方法：202411957015.7[P].2025-10-21.

[7] 范楠欣，王宁，王佳明．基于 QEMU 的 1394 总线通信仿真 [J]. 民用飞机设计与研究，2025(2):81-86.

[8] 周丽华．基于故障注入的 AFDX 网络监测系统软件设计与实现 [J]. 网络安全和信息化，2025(10):102-104.

[9] 姚雪美，洪应平，党立志，等．基于软件实现的 RS485 总线电气层故障注入仪设计 [J]. 电子器件，2025,48(1):55-59.

[10] 刘光星，焦方新．基于 ARINC664 帧的故障注入方法研究 [J]. 信息技术与信息化，2025(12):16-19.

【作者简介】

侯龙伟（1988—），男，山西忻州人，硕士，工程师，研究方向：数字样机。

（收稿日期：2026-07-16 修回日期：2026-08-11）