文章编号:1671-4598(2023)04-0205-08 DOI:10.16526/j.cnki.11—4762/tp.2023.04.032 中图分类号:TP391.9 文献标识码:A

基于 Keil C51 的嵌入式软件外设虚拟化设计与实现

申 臻，宋雷军，魏冬冬，于清华，刘 涛
（上海航天电子技术研究所，上海 201109）

摘要：航空航天控制领域嵌入式软件测试主要存在软件运行物理环境受限和软件测试覆盖性不足等问题，为解决以上问题，对嵌入式软件的外部设备进行了研究，构建了数字化测试平台替代实物环境的方案，设计并完成了平台的总线 1553B、总线 RS422、AD 采集、I/O 等外部设备；并模拟了平台嵌入式软件真实的运行环境，使测试过程不受实物环境制约，测试激励的注入不受任何限制，保障了测试的充分性；最后，以某电源下位机测试过程为例，实现了故障注入、边界测试，验证了平台在嵌入式软件测试中的有效性和可靠性。

关键词：1553B 虚拟化；Keil；嵌入式软件；数字测试环境；目标码覆盖率

Design and Implementation of Embedded Software Peripheral Virtualization Based on Keil C51

SHEN Zhen, SONG Leijun, WEI Dongdong, YU Qinghua, LIU Tao
(Shanghai Aerospace Electronic Technology Institute, Shanghai 201109, China)

Abstract: Embedded software in the aerospace control field has the problems of limited physical environments and insufficient test coverage for software. In order to solve above problems, an external equipment of embedded software is researched, and a digital test platform is proposed to replace in the physical environment, and the peripheral components of the platform such as bus 1553B, bus RS422, AD acquisition, I/O, etc. are designed and completed. The real operating environment of the embedded software is simulated, the test process is not restricted by the physical environment, and the injection of test incentives is not restricted in any way, which ensures the adequacy of the test. Finally, taking the test process simulation of the lower computer of a certain power supply as an example, it can realize the execution of special test cases such as fault injection and boundary testing, which verifies the effectiveness and reliability of the platform in embedded software testing.

Keywords: 1553B virtualization; Keil; embedded software; digital test environment; target code coverage

0 引言

8051 单片机已在航空航天控制领域广泛应用，需要测评的嵌入式软件数量逐年增多。软件测试又是保障软件安全、正确、完整的关键环节[1]。测试人员使用被测件的硬件平台完成测试，通常会面临两个问题：1）重大科研或者重要型号项目的物理环境，难以为测试工作提供足够的使用时间[2]；2）不能执行可能引起物理环境破坏的异常测试，测试的覆盖性得不到保证[3]。面对只有软件源代码，实物测试环境不具备、故障测试用例无法执行的测评任务，如何提供有效测试环境成为软件测评单位的难题。

编译软件 Keil C51（美国 Keil Software 公司开发）为 51 系列单片机程序[4]提供开发调试环境，涵盖编辑、编译、连接、调试、仿真等整个开发流程。在测试中，通过 Keil 编译被测试的嵌入式软件（以下简称为“被测件”）代码，并提供被测件运行的仿真环境。基于 Keil 提供的高级仿真接口（AGSI，advanced generic simulator interface）[5]自研全数字测试平台（以下简称“测试平台”）模拟芯片指令[6]，实现被测件的仿真运行、外围激励注入、外设硬件接口模拟、覆盖率统计等。

基于以上背景，Keil C51 为被测件提供运行环境；测试平台提供虚拟化接口、外围激励环境[7]，实现测试用例执行、故障注入、边界测试以及测试结果的观察。测试平台与 Keil 之间通过用户数据报协议（UDP，user datagram protocol）完成数据交互[8]。

本文基于 Keil C51 及测试平台，完成被测件外设的虚拟化，生成可在测试平台上运行的可复用模块。虚拟化外设包括：1553B 总线、模数转换（AD，analog digital）采集、输入/输出（I/O，input/output）数据、RS422 总线。测评人员通过测试平台选择被测件运行所需的虚拟化外设模块，构建全数字测试环境，实现被测件运行过程的可控，内存、寄存器信息的可改。最后，以开发某电源下位机测

收稿日期:2022-08-20； **修回日期**:2022-09-09。

作者简介:申 臻(1990-),男,山西长治人,硕士,工程师,主要从事软件测试技术与仿真方向的研究。

引用格式:申 臻,宋雷军,魏冬冬,等.基于 Keil C51 的嵌入式软件外设虚拟化设计与实现[J].计算机测量与控制,2023,31(4):205-212.

试平台为例，完成故障注入、测试用例执行、测试结果观测，验证虚拟化外设在嵌入式软件测试过程中的有效性和可靠性。

1 系统结构及原理

嵌入式软件测试需要具备两个条件：1）嵌入式软件的运行环境[9]；2）测试激励注入及嵌入式软件输出数据的解析与显示[10]。虚拟化的外设、通讯中间件、Keil，三者关系结构如图 1 所示。测试平台的上位机部分，完成测试激励注入以及接收、解析、显示嵌入式软件输出的数据；测试平台的下位机部分，基于 AGSI 实现嵌入式软件运行时内存、寄存器读写监控、数据同步、触发中断、UDP 通信等；Keil C51 的 uVision 版本提供 Windows 下的集成开发环境（IDE，integrated development environment），实现嵌入式软件的编译、调试、运行[11-13]。本文重点介绍 1553B 虚拟化、AD 采集虚拟化、RS422 虚拟化、I/O 虚拟化。

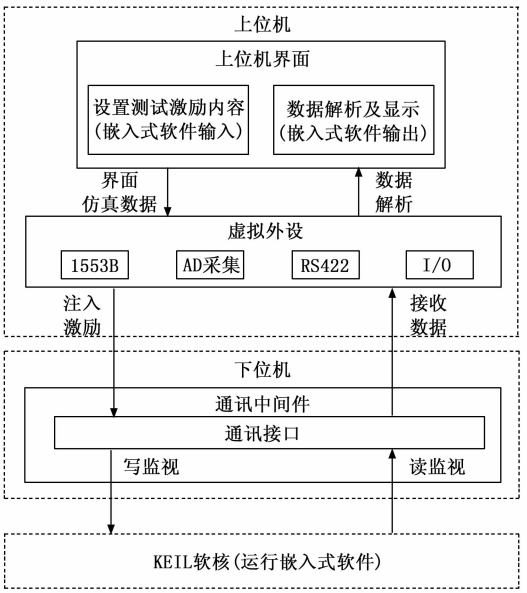

图 1 测试平台架构

嵌入式软件运行过程会响应不同中断源，CPU 检测到中断请求，执行中断服务程序做相应的处理。测试平台根据不同中断源，更改对应寄存器地址中的值，Keil 内核监测到特殊寄存器值更改时触发中断，嵌入式软件进入中断服务程序。其中特殊寄存器包括：串行口控制寄存器（SCON，serial control）、中断开发/禁止（IE，interrupt enable）、P0 口、P1 口、P2 口、P3 口（8bit 分别代表不同的中断源，如 P3.2 为外部中断 INT0，P3.3 为外部中断 INT1）等。对于中断方式为电平触发、边沿触发，测试平台可向 P3 口对应 bit 位写入高电平、低电平、上升沿、下降沿，实现中断的触发。

测试平台实现对被测件内存、寄存器的读写仿真，模拟总线 1553B、AD、RS422、I/O 接口等外设与被测件交互数据的过程。仿真过程如下：

1）上位机监控被测件需要向外设写数据的全部地址。当 Keil 软核识别到被测件运行至向外设输出数据处（即被测件写地址操作），虚拟软核触发下位机中的写回调函数，通知下位机此时被测件向哪个地址写入何值。下位机将收到的地址值及数据，通过 UDP 发送给上位机程序，上位机收到地址及数据完成数据解析与界面显示等后续处理。实现嵌入式软件内存、寄存器写地址动作的仿真。

2）上位机监控被测件需要从外设读数据的全部地址。当 Keil 软核识别到被测件运行至从外设读取数据处（即被测件读地址操作），虚拟软核触发下位机中的读回调函数，通知下位机当前被测件读取的地址值。下位机将收到的地址值，通过 UDP 发送给上位机程序。上位机将虚拟外设向该地址写入的模拟数据，通过 UDP 发送给下位机通讯接口，使得被测件采集到上位机写入的数据。实现嵌入式软件内存、寄存器读地址动作的仿真。

2 外设虚拟化设计与实现

测试平台的上位机部分主要包括上位机界面和虚拟外设。上位机界面分为输入数据、输出数据两大部分：1）输入数据部分，测试人员根据测试需求设置仿真数据，通过虚拟外设将仿真数据发送给被测件。测试人员可设置不同的测试数据，完成被测件的边界测试、故障测试，保障测试充分性；2）输出数据部分，实时显示被测件运行过程中输出数据的解析及状态显示，供测试人员分析测试结果。

虚拟外设将测试人员在上位机界面设置的仿真数据，转换为满足外设特定格式的数据帧，将转换后的数据帧发送给被测件，实现测试激励的注入。同时，虚拟外设接收被测件运行过程中输出的数据，转交给上位机界面，完成数据的解析、显示、存储等处理。

其中，虚拟外设的设计与实现主要包括：总线 1553B 虚拟化、AD 采集虚拟化、总线 RS422 虚拟化、被测件 I/O 接收虚拟化。

2.1 总线 1553B 虚拟化设计与实现

开展总线 1553B 虚拟化工作之前，需要研究 1553B 总线的物理实现逻辑，分析开发测试平台需要仿真 1553B 总线的方法及内容。本小节首先介绍测试平台所关注的 1553B 总线简介、1553B 物理实现逻辑。其次，根据总线 1553B 物理实现逻辑的研究，设计并实现 1553B 中总线控制器（BC，bus controller）、远程终端（RT，remote terminal）的仿真。

2.1.1 1553B 总线简介

1553B 出自美军标准 MIL-TD-1553B，原为美军航空电子综合通信的标准，全称为“飞机内部时分制指令响应式多路传输数据总线”[14]。1553B 为基于消息（Message）的通信协议，每条消息的最大信息量 32 字，分为命令字、数据字、状态字，每类字长 20 bit（有效数据 16 bit），每个字的前 3 位为单字的同步字头，最后 1 位是奇偶校验位[15]。命令字位于每条消息的起始部分，其内容决定消息的特征与标识，状态字只能由 RT 发出，其内容表征 RT 向 BC 发出的有效命令的反馈，BC 根据状态字内容决定下一步操作。数据字支持 RT－＞BC、BC－＞RT、RT－＞RT 传输数据[16]，具体字内容如图 2 所示。

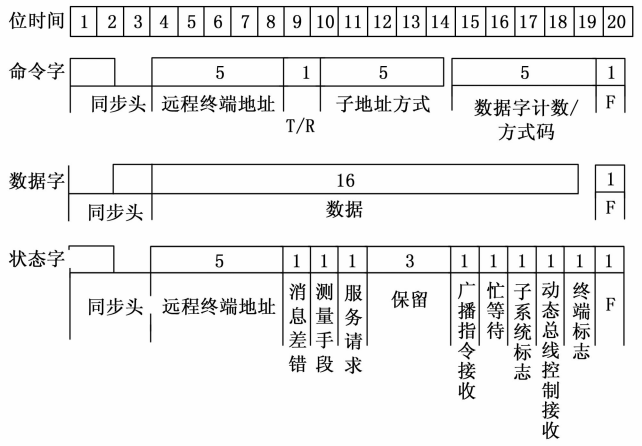

图 2　1553B 总线字格式

1553B 总线采用指令/响应型通信协议，包括 3 种终端：总线控制器 BC、远程终端 RT、总线监视器（MT，monitor terminal）。传输的信息格式有 BC－＞RT、RT－＞BC、RT－＞RT、广播方式和系统控制方式，且全部在 BC 的参与下完成。1553B 总线传输消息的标准过程为：BC 向某 RT 发送接收/发送指令，RT 在给定的响应时间范围内发回一个状态字，并执行消息的接收/发送[17]。消息传输格式如图 3 所示。其中 * 表示响应时间，范围是 4.0～12.0 μs，表示消息间隔时间，规定＞4 μs。

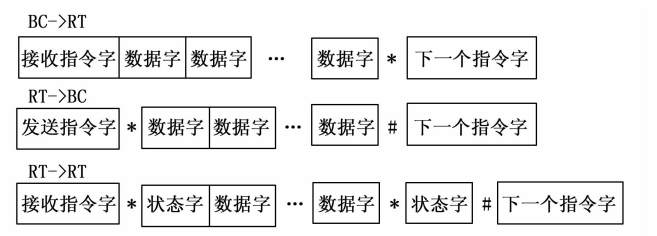

图 3　消息传输格式

2.1.2　1553B 物理实现逻辑研究

1553B 实现虚拟化，完成与被测件数据、指令交互，需要清晰的梳理出 1553B 物理实现逻辑。在 1553B 总线简介中指出 3 种终端：BC、RT、MT，其中 BC、RT 完成总线指令、总线数据的收发。本段着重介绍 BC、RT 的物理逻辑。

RT 存储器结构在非增强模式和增强模式下，有如表 1 所示几个区域均被设置为专用区。其他区域为数据块等。RT 查找表实现将 TX/RX/BCST 子地址对应的数据块映射到共享 RAM 区的机制。可对单独子地址所指向的存储空间进行读写操作，对广播、接收、发送数据分离处理，实现安全、独立的存取数据。RT 指令堆栈的长度可设为 125 字、512 字、1 024 字、2 048 字。

1553B 采用双缓存机制，对应查找表 A、B。每个查找表分 4 块：32 个发送子地址、32 个接收子地址、32 个广播子地址、32 个子地址控制字。RT 查找表具体内容如表 2 所示。

表 1　RT 存储器机构固定部分

地址	描述
0000-00FF	堆栈 A
0100	堆栈 A 指针(固定)
0104	堆栈 B 指针(固定)
0140-01BF	查找表 A(固定)
01C0-023F	查找表 B(固定)

表 2　RT 查找表

A 区	B 区	描述	说明
0140 … 015F	01C0 … 01DF	RX(/BCST) SA 0 … RX(/BCST) SA31	接收(广播)查找指针表
0160 … 017F	01E0 … 01FF	TX SA0 … TX SA 31	发送查找指针表
0180 … 019F	0200 … 021F	BCST SA0 … BCST SA31	广播查找指针表(可选)
01A0 … 019F	0220 … 021F	SACW SA0 … SACW SA31	子地址控制字表(可选)

子地址控制字中存储管理器 2、存储管理器 1、存储管理器 0（MM，memery management）3 bit 的取值，设置每次发送、接收或广播数据的长度，长度可设置为 128 字、256 字、512 字、1 024 字、4 096 字及 8 192 字。1553B 虚拟化设计中要支持根据子地址控制字设置收发缓存大小[18]。

在 1553B 虚拟化时，根据总线指令的收发子地址，对应查找表查询到该子地址对应的内存存储起始地址。接收到的数据被写入由查找表指针指向的数据块，需要发送的数据从查找表指针指向的数据块取出并发送。

2.1.3　BC 虚拟化实现

1553B 的 BC 虚拟化需要仿真上述“物理实现逻辑研究”中指出的 1553B 初始化、查询查找表、读写子地址对应存储区等。仿真 BC2RT 数据、BC2RT 指令、RT2BC 数据、RT2BC 指令等数据、指令的交互过程。使用共享内存模拟查找表及数据块的存储逻辑。定义两个枚举结构体 Enum1553B _ Channel _ Num、EnumCommand 分别表示 A \ B 区、BC 指令类型，对应代码如下所示：

```
public enum Enum1553B_Channel_Num{
    ChannelA = 0,
    ChannelB = 1
}
public enum EnumCommand{
    BC2RT,
    RT2BC,
    RT2RT,
```

```
    Broadcast,
    RT2RTs,
    ModeCode,
    BroadcastModeCode,
    NoUsed,
    Error
}
```

定义 BC 仿真类 BcHelper，完成 1553B 总线的初始化。初始化包括：映射的内存地址 StartAddr、1553B 指令地址 BcCmdAddr，1553B 相关寄存器地址映射（RT 指令堆栈指针 A、RT 指令堆栈指针 B、传送矢量字、数据同步、查找表 A 起始地址、查找表 B 起始地址、发送查找指针表、接收查找指针表、广播查找指针表、子地址控制字表等），初始化内容如下所示：

```
public static EnumIntName IntName = EnumIntName.NoInt;
// BC 模拟器地址映射
public static UInt32 StartAddr = 0; //映射的 mem 地址
public static UInt16 BcCmdAddr = 0; //映射的 1553B 指令地址
//增强型 1553B 寄存器地址映射
private static UInt32 AddressStackA = 0; //堆栈
public static UInt32 AddressRtCommandStackPointerA = 0x100;
public static UInt32 AddressRtCommandStackPointerB = 0x104;
private static UInt32 AddressTransmitVetcorWord = 0x120;
private static UInt32 AddressSynchronizeWithData = 0x111;
private static UInt32 AddressReceiveLookupPointerTableA = 0x140;
private static UInt32 AddressReceiveLookupPointerTableB = 0x1c0;
private static UInt32 AddressTransmitLookupPointerTableA = 0x160;
private static UInt32 AddressTransmitLookupPointerTableB = 0x1e0;
private static UInt32 AddressBroadcastLookupPointerTableA = 0x180;
private static UInt32 AddressBroadcastLookupPointerTableB = 0x200;
private static UInt32 AddressSubaddressControlWordTableA = 0x1a0;
private static UInt32 AddressSubaddressControlWordTableB = 0x220;
private static UInt32 AddressDataBlock = 0x260;
}
```

BC 端发送控制命令的仿真函数如下所示：

```
/// <summary>
/// 发控制命令
/// </summary>
/// <param name="rtAddr">RT 地址</param>
/// <param name="rx">发送标志,0 标识 RT 端接收</param>
/// <param name="sa">子地址</param>
/// <param name="len">长度/方式字(指 1553B 发送的"字"的长度)</param>
/// <param name="Data">发送的数据</param>
/// <param name="bcst">是否是广播标志</param>
/// <param name="channel">通道标志</param>
public static void SendCmd(Enum1553B_Channel_Num channel, bool bcst, int rtAddr, bool rx, int sa, int len, byte[] data)
```

仿真控制命令后可实现如下过程模拟：BC 发送广播、BC2RT 数据、BC2RT 命令、方同步码等。控制命令流程如图 4 所示。其中查找表仿真方法为：

1）判断当前控制命令的类型：广播、发送。

2）判断当前命令由 BC 发到 RT 的哪个 SA。

3）查询查找表中该 SA 对应的存储起始地址，以及嵌入式软件设置的内存起始地址，计算出该 SA 数据块在共享内存中的存储区域。

4）将 SA 对应的存储区域数据取出后发送给 RT。

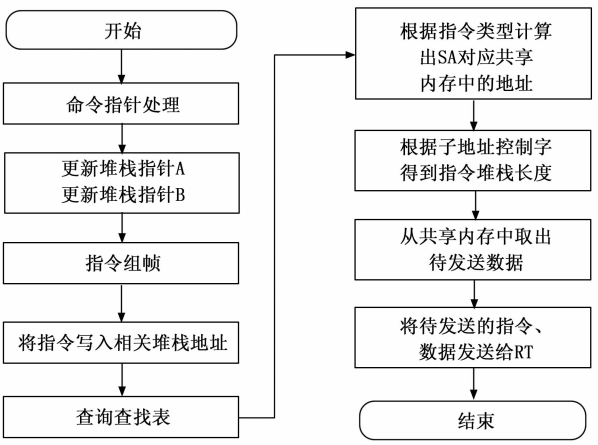

图 4 控制命令仿真流程

BC 端接收 RT 发来数据（RT2BC 数据）的仿真过程如图 5 所示。其中对查找表的使用与发送控制命令仿真部分相同。

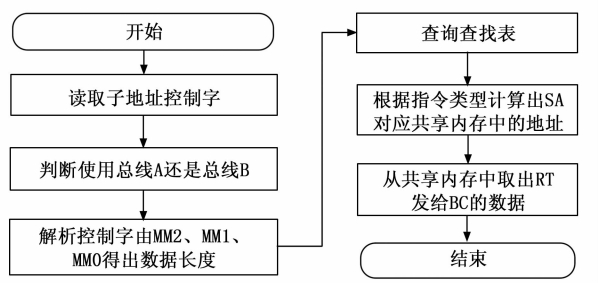

图 5 BC 接收 RT 数据流程

2.1.4 RT 虚拟化实现

实际应用中被测件有时作为 BC 端使用，在构建测试环境时需要对外设 RT 进行虚拟化仿真，即实现 RT2BC 数据、RT2BC 指令、RT2RT、RT2RTS、BC2RT 等数据、

指令的交互过程[19]。

模拟 RT 端接收 BC 发来的指令、数据，将收到的数据放入本地共享内存，共享内存的组织方式与“BC 端仿真模拟”一致。外设 RT 收到作为 BC 的被测件发送的。模拟 RT 端的处理流程如图 6 所示。

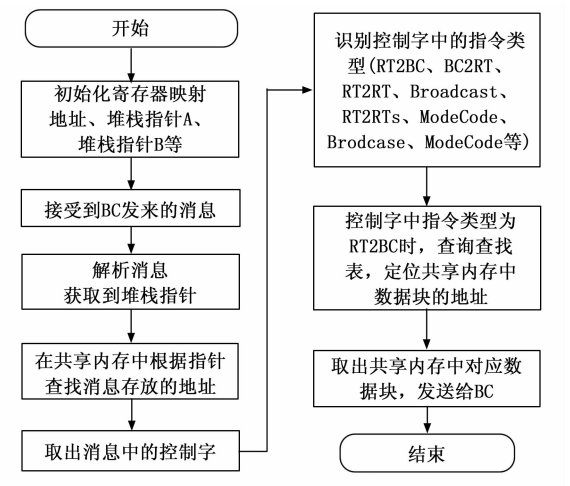

图 6　RT 端仿真处理流程

2.2　AD 采集虚拟化设计与实现

2.2.1　AD 采集虚拟化设计

实际应用中嵌入式软件通过 AD 采集模块采集模拟信号，经过模数转化，将转换后的数字量用于软件的后续使用[20]。

嵌入式软件采集信号后的处理过程，实际使用的是数字量而非模拟量。因此，在测试平台中无需仿真模数转换过程，可直接将数字量值赋给嵌入式软件。数字量可通过 AD 采集界面设置任意值，供嵌入式软件采集。既能满足嵌入式软件从 AD 采集模块获取数字量的要求，又能满足 AD 采集模块测试的充分性。

2.2.2　AD 采集虚拟化实现

不同被测件 AD 采集部分的处理过程有相似之处，将通用的处理过程抽象为 AD 采集模块，完成 AD 采集外设的虚拟化。AD 数据的采集与外设的交互部分涉及采集通道选择、采集高低字节、采集次数等。嵌入式软件与外设交互的流程如图 7 所示。

实现 AD 采集虚拟化，需要开辟一段共享内存用于存储测试人员设置的一组 AD 值，数值内容要求：

1）采集的路数；

2）每路数据采集次数；

3）每次采集值，支持设置不同值。

被测件第某次采集某路 AD 数据时，触发测试平台的读回调函数。在回调函数中判断当次采集为嵌入式软件第几次采集第几路 AD 数据，取出共享内存中对应的 AD 值，供嵌入式软件采集。AD 采集虚拟化处理流程与图 7 所示的嵌入式软件与 AD 外设交互流程相同。

不同嵌入式软件通过测试平台界面可配置采集 AD 数据通道、采集地址、CPU 时钟频率、采集方式等信息，如

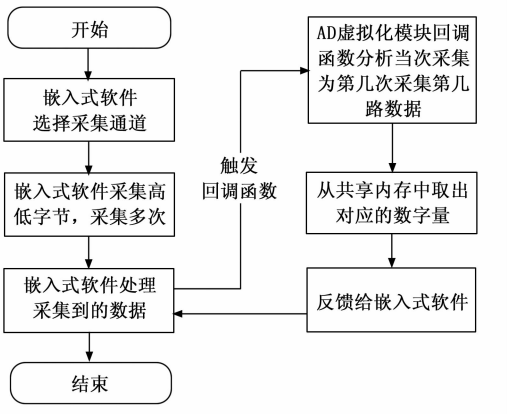

图 7　AD 采集外设交互流程

图 8 所示。

图 8　AD 采集配置界面

2.3　总线 RS422 虚拟化设计与实现

2.3.1　总线 RS422 虚拟化设计

串行总线 RS422 通信由于其在传输距离、抗干扰能力方面的优势，在航天嵌入式软件中普遍应用[21]。嵌入式软件测试时主要关心被测件从串口采集数据之后的处理流程正确与否，以及观察被测件输出的串口数据。被测件通过串口采集数据的方式为：MOV A SBUF；通过串口输出数据的方式为：MOV SBUF A。其中 SBUF 为寄存器 0x99。

因此，在测试平台开发中无需仿真串口全部物理功能，只需实现被测件从串口采集、输出数据的过程。测试人员根据测试需求，灵活设置正确值、边界值、异常值等供被测件采集，用白盒测试法分析被测件采集串口数据后不同处理分支的正确性。被测件通过串口输出的数据，由测试平台接收并依据通信协议完成解析，方便测评人员观察。

2.3.2　总线 RS422 虚拟化实现

实现对总线 RS422 的虚拟化，处理逻辑如图 9 所示。

具体方法如下：

1）测试人员设置输入激励并存入共享队列；

2）被测件运行到需要从 RS422 获取数据处，即从寄存器 0x99 读取数据，触发测试平台的读回调函数。测试平台将共享队列中的测试激励通过 0x99 注入被测件；

3）被测件通过串行总线 RS422 输出数据时，即通过寄

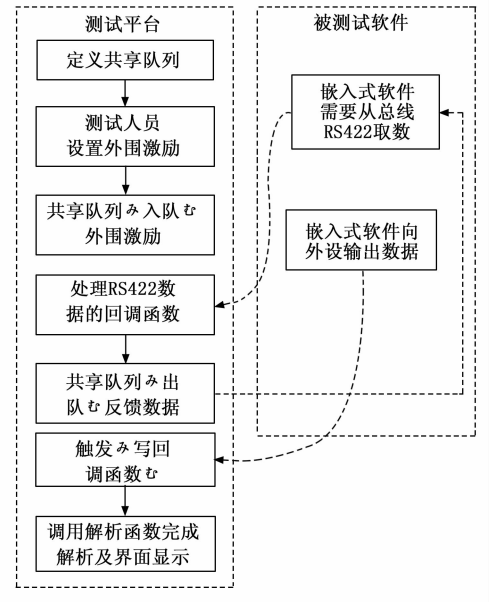

图 9 RS422 总线数据交互流程

存器 0x99 输出数据，触发测试平台写回调函数。测试平台依据被测件通信协议接收一整帧串口数据后，完成解析并在测试平台界面显示。测评人员通过界面观察被测件输出的串口数据，判断其运行是否合理。

2.4 I/O 虚拟化设计与实现

2.4.1 I/O 虚拟化设计

嵌入式软件外设包括上述通用的 1553B、RS422、AD 采集，还包括与 FPGA、DSP 或其他外设交互的硬件信号、数据等。建立完整的嵌入式软件数字仿真测试环境，还需定制化的完成被测软件全部输入数据、全部输出数据解析的仿真[22]。这部分定制化的外设仿真统称为 I/O 虚拟化。

2.4.2 I/O 虚拟化实现

I/O 虚拟化实现过程与总线 RS422 虚拟化相似，与流程图 9 不同之处为："测试人员设置外围激励"，此处提供两种设置激励的方法，测试人员可通过界面和具有固定格式的 Excel 表格设置外围激励；其次，被测件从总线 RS422 取数，改为从通过 I/O 交互数据的外设采集取数。I/O 虚拟化实现的具体方法如下：

1）针对不同外设开辟一段独占的共享内存，用于存储测试人员设定的测试激励。

2）设计固定格式的 Excel 表格，如表 3 所示。测试人员将测试激励填入表格，测试平台运行后自动读取表格内容，将数据及地址成对的存入共享内存。

表 3 I/O 固定输入数据格式

序号	类型	名称	地址(H)	数据类型	位/bit	位值/bit	数据/byte
1	类型 1	内容 1	1 900	Bits	bit7	0	6A
2	类型 2	内容 1	1 901	Byte	\	\	AB
3	类型 2	内容 2	1 902	Byte	\	\	CD
…	…	…	…	…	…	…	…

3）测试过程中，通过界面实时设置测试激励。

4）被测件运行至采集外设数据地址处，触发测试平台的读回调函数。测试平台根据接收到的"读地址"，将共享内存中对应的数据发送给被测软件。

5）被测件向"写地址"输出 1 字节数据或者一帧遥测数据时，触发测试平台的写回调函数。测试平台记录接收到的"写地址"以及对应 1 字节数据或者一帧遥测数据，依照通信协议完成被测件输出数据的解析。测试人员可通过界面实时观察被测软件的输出信息。

3 实验结果与分析

为验证上述描述的嵌入式软件外设虚拟化的有效性和可靠性。以某电源下位机软件测试平台开发为例，模拟 1553B、RS422、I/O、AD 采集等外设，实现被测件外围环境的正常功能测试、异常故障用例注入，提高测试的充分性，完成对该电源下位机软件的测试工作。该电源下位机软件主要功能是太阳电池阵功率调节、蓄电池组充放电管理功能。需要完成工程参数的采集处理、遥控指令的执行、蓄电池充电管理、均衡器控制等功能，通过 1553B 总线实现与综合电子分系统的信息交换。

3.1 测试平台搭建

3.1.1 上位机部分搭建

基于自研测试平台选择 1553B 模块、AD 采集模块、针对该电源下位机开发的 I/O 模块。测试平台模块选择界面如图 10 所示。

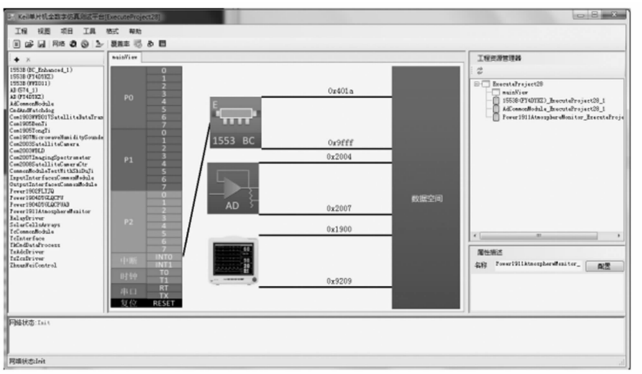

图 10 测试平台搭建界面

总线 1553B 虚拟化后需要通过界面配置接收发送子地址、广播子地址、寄存器、中断触发等信息，以完成被测试软件与外部设备通过 1553B 传输数据的仿真。配置界面如图 11 所示。

AD 采集外设模拟需要配置被测试软件采集地址、采集高低字节地址等信息，配置界面如图 8 所示。

3.1.2 被测件运行环境配置

嵌入式软件电源下位机程序的仿真运行环境由 Keil 提供，配置基于 AGSI 编写的动态链接库，使用 Keil 加载源程序完成编译。使用 Keil 软件运行电源下位机程序，可完成电源下位机程序动态运行下的白盒测试。其中动态链接

图 11　总线 1553B 配置界面

库实现测试平台与 Keil 软件 UDP 通信、被测试软件内存、寄存器的读写监视、定时器设置、时需同步、中断触发等。

3.2　测试平台运行与测试

测试平台运行界面如图 12 所示。测试人员可通过运行界面的左侧设置注入激励的内容，实时更改注入的总线 1553B、RS422 的指令帧或者数据，以及被测件需要从 I/O 外设采集的全部数据。运行界面右侧为被测件上传数据的实时显示。

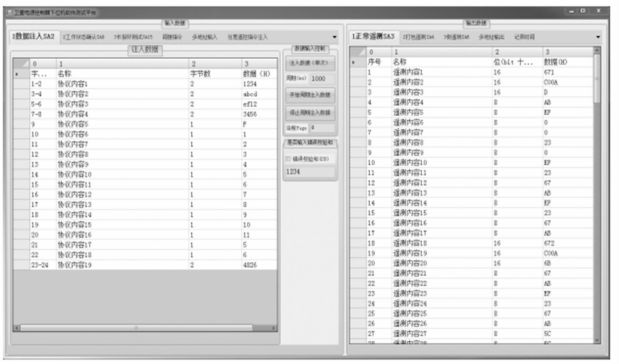

图 12　运行界面

AD 采集界面如图 13 所示，测试人员可实时更改供被测件采集的各路 AD 数据。支持直接在界面上更改 AD 值、导入 AD 数据 Excel 表格两种方式设置 AD 值。

图 13　AD 采集界面

通过测试平台完成被测件中断 INT0、T0、T1 执行时间、中断响应总线指令时间的记录，以及周期发送指令、覆盖率统计。程序地址计数器（PC，program counter）用于存放要执行指令的地址，16 位，能自动加 1。一个机器周期（M-Machine 周期）是 12 个时钟周期，当主频为 12 MHz 时，一个机器周期为：1 M = 1/12 MHz * 12=1 μs。被测件某段程序执行时间的计算方法是：记录该段程序起止 PC 之间执行的机器周期数，乘以一个机器周期的时间，得到该段程序的执行时间。

3.3　测试结果分析

该电源下位机软件共使用 3 种中断 INT0、T0、T1，中断服务程序执行时间的测试结果如表 4 所示。

表 4　中断服务程序执行时间

中断类型	中断服务程序起始 PC	中断服务程序终止 PC	中断服务程序最大执行时间/ms
INT0	0x6557	0x66FC	28.016 5
T0	0x3DA8	0x4360	0.755 5
T1	0x54C3	0x579C	0.467 0

覆盖率统计支持被测件目标码覆盖率情况保存、多个覆盖率文件合并，源码目标码对照显示。统计结果针对目标码执行情况进行分析，其中包括 6 种执行结果：

1）EX：executed，该条目标码语句执行，覆盖率统计结果中以绿色显示；

2）NE：Not Executed，语句没有执行，覆盖率统计结果中以红色显示；

3）JF：Jump Fully，分支完全执行，覆盖率统计结果中以黄色显示；

4）JN：Jump Never，分支顺序执行，但是没有跳转，即每次执行到这条语句，跳转条件都不满足，覆盖率统计结果中以黄色显示；

5）JO：Jump Only，执行了分支跳转，但是没有顺序执行，即每次执行到这条语句都是跳转，覆盖率统计结果中以黄色显示；

6）JNE（jump not executed），跳转分支、顺序分支都没有执行，覆盖率统计结果中以红色显示。

目标码覆盖率统计结果中先显示一句源码，紧接着显示对应的目标码，并在第一列显示每条目标码的执行情况。根据每条目标码的执行情况标示绿色、红色、黄色，方便测评人员观察，如图 14 所示。

被测件目标码覆盖率统计情况如表 5 所示，包括目标码模块名、总指令数、执行指令数、执行百分比、总跳转指令数、JF 指令数、JO 指令数、JN 指令书、JNE 指令数、JF 执行百分比。部分嵌入式软件目标码分支、语句覆盖率均要达到百分之百。表 5 展示覆盖率统计功能，并非电源下位机软件第三方测试的最终覆盖率统计结果。

实验结果证明，在自研测试平台基础上，基于 Keil C51 完成嵌入式软件外设虚拟化，实现嵌入式软件仿真运行、外围激励注入、外设硬件接口模拟的方法切实可行。通过数字测试平台不仅能够完成与实物测试平台相同的测试效果，还能完成故障注入、边界测试、目标码覆盖率等。

```
299:          if((uiTime>=LimitTime)&&((*flag2_t)==M_YES))
EX  C:0x5105  C3      CLR     C
EX  C:0x5106  ED      MOV     A,R5
EX  C:0x5107  9538    SUBB    A,0x38
EX  C:0x5109  EC      MOV     A,R4
EX  C:0x510A  9537    SUBB    A,0x37
JO  C:0x510C  400E    JC      C:511C
NE  C:0x510E  121562  LCALL   C?CLDPTR(C:1562)
NE  C:0x5111  B4A508  CJNE    A,#0xA5,C:511C
300:          {
301:              (*flag2_t)=M_NO;
NE  C:0x5114  745A    MOV     A,#flag2_EquiNBat(0x5A)
NE  C:0x5116  1215A8  LCALL   C?CSTPTR(C:15A8)
302:              return  M_YES;
NE  C:0x5119  7FA5    MOV     R7,#0xA5
NE  C:0x511B  22      RET
303:          }
304:          else  if(uiTime>=LimitTime)
EX  C:0x511C  C3      CLR     C
EX  C:0x511D  ED      MOV     A,R5
EX  C:0x511E  9538    SUBB    A,0x38
EX  C:0x5120  EC      MOV     A,R4
EX  C:0x5121  9537    SUBB    A,0x37
JO  C:0x5123  4008    JC      C:512D
305:          {
306:              (*flag2_t)=M_YES;
```

图 14 覆盖率统计结果

表 5 目标码覆盖率统计结果

模块名	总指令数	执行指令数	执行百分比/%	总跳转指令数	JF 指令数	JO 指令数	JN 指令书	JNE 指令数	JF 执行百分比/%
GLOBAL_INIT	445	422	94.8	2	2	0	0	0	100.0
BUS1553BINIT	307	303	98.7	12	8	4	0	0	66.7
INT0_1553B	218	218	100.0	14	13	0	1	0	92.9
SAMPLE_DATA7	262	236	90.1	6	4	2	0	0	66.7
TIME0_20MS	859	553	64.4	101	14	31	7	49	13.9
TIME1_20MS	399	274	68.7	39	9	7	14	9	23.1
MAIN	677	407	60.1	60	6	6	11	37	10.0

附注:表格中为第三方测试进行中,被测件目标码覆盖率统计结果的部分展示

4 结束语

本文提出的基于 Keil C51 的嵌入式软件外设虚拟化,实现了在真实环境中运行的嵌入式软件与全部外部设备交互数据的仿真,使得测试过程不受硬件实物平台的使用限制,拥有充分的测试时间。

1) 在测试过程中,外部设备的虚拟化提供了灵活的测试激励注入方式,能够完成实物测试环境难以模拟的故障注入、边界测试以及测试结果实时显示。

2) 目标码覆盖率统计能够帮助测评人员分析被测件代码执行情况,精准定位未执行的语句、分支。辅助测评人员分析未执行的原因,针对性的设计测试用例,进而完成未执行语句、分支的覆盖。

3) 嵌入式软件外设虚拟化后,主要解决了测试工作中的两个问题:测试过程不依赖嵌入式软件真实的运行环境;测试过程内存、寄存器信息可控可改,测试用例注入方式灵活,能够完成实物测试环境不能满足的故障、边界测试。

4) 外设的虚拟化设计,不仅能够缓解目前航天型号领域嵌入式软件测试过程中实物运行环境使用时间冲突的问题,而且对提高嵌入式软件的测试效率和测试充分性也有帮助。

参考文献:

[1] 颜运强,刘 志,高 峰,等. 嵌入式软件全数字仿真测试技术研究与应用 [J]. 西南科技大学学报,2013,28 (3):73-76.

[2] 杨丰玉,徐浩明,郑 巍,等. 嵌入式软件测试研究综述 [J]. 航空计算技术,2021,51 (1):112-115.

[3] 胡 杰,严 智. 软件错误注入测试技术研究 [J]. 计算机与数字工程,2015,43 (5):882-886.

[4] 杜伟略. 80C51 单片机及接口技术 [M]. 北京:化学工业出版社,2008.

[5] 王晓东. KEIL$_\mu$ V2 下 AGSI 接口结构及其仿真 DLL 的实现 [J]. 自动化技术与应用,2008,27 (11):33-35.

[6] CHEN H, PAN Y, YIN Y, et al. All-digital background calibration technique for timing mismatch of time-interleaved ADCs [J]. Integration the Vlsi Journal, 2017, 57: 45-51.

[7] 郭 华,武占峰,吴 瑾. 全数字仿真测试环境在星载嵌入式软件测试中的应用 [J]. 航天控制,2012,30 (6):70-72.

[8] 葛德明. 实时嵌入式软件的测试技术 [J]. 电子测试,2018 (10):88-89.

[9] 凌 杨,邵培南,佟 雷,等. 基于数字化仿真环境的嵌入式软件测试方法 [J]. 计算机工程,2011,37 (S1):49-51.

[10] 张 涛,李瑞军,范延芳. 基于 SPARC V8 的星载嵌入式软件全数字仿真平台设计与实现 [J]. 计算机测量与控制,2020 (1):11-15.

[11] 肖前远. 航空嵌入式软件全数字仿真测试技术研究 [D]. 南京:南京航空航天大学,2010:13-14.

[12] 佟 雷,邵培南,凌 杨,等. 嵌入式计算机数字化仿真测试环境设计 [J]. 计算机工程,2011,37 (S1):395-397.

[13] 黄 涛,孙 罡,同向楠. 嵌入式软件测试方法与技术 [J]. 电子技术与软件工程,2018 (12):212.

[14] 王新亮,陈 凯,薛琪琪,等. 基于 1553B 总线的飞控软件测试仿真平台设计 [J]. 计算机测量与控制,2020,28 (11):12-15.

[15] 淮治华,田 泽,杨 峰,等. 2M 1553B 总线仿真卡的设计与实现 [J]. 计算机技术与发展,2015,25 (4):229-232.

[16] 徐贵贤. 1553B 总线简介及其实现 [J]. 通信技术,2011 (5):172-174.

[17] 邰全亲,傅 岚,杨京松. 基于以太网的 1553B 仿真测试设备设计 [J]. 计算机测量与控制,2008,16 (11):1559-1560.

[18] 单忠伟. 1553B 总线远程终端的 FPGA 程序设计 [J]. 现代电子技术,2013 (9):28-30.

[19] 舒传华,唐海波,曹 赣. 1553B 总线消息解析方法研究和应用 [J]. 遥测遥控,2015,36 (6):57-63.

[20] 李田英,刘胜珍. 嵌入式实时软件在计算机软件设计中的应用 [J]. 电子设计工程,2017,25 (8):178-182.

[21] 李新贝,谭超,高 山,等. 星上 RS422 接口电路的建模与仿真 [J]. 航天器工程,2011,20 (1):109-113.

[22] 李 昌,邓矢斧,冯 雷,等. 基于全数字的航空机载软件验证平台研究 [J]. 计算机测量与控制,2018,26 (6):130-133.