

“魂芯”DSP 配套基础软件的自动化测试平台^①

林广栋, 耿锐, 王昊

(中国电子科技集团公司第三十八研究所 集成电路设计中心, 合肥 230088)

摘要: 介绍了一种基于调试系统的回放式自动化测试平台, 可以自动化测试“魂芯”DSP 配套软件单元. 该测试平台以主机调试器为测试对象, 通过主机调试器调试应用程序, 间接地测试编译器、软件模拟器、调试链接服务软件等软件单元. 该平台可以方便地添加测试用例, 并自动批量回放测试用例. 利用本测试平台, 可以大大减少“魂芯”芯片配套软件的测试工作量.

关键词: 软件测试; 调试; 自动化测试

引用格式: 林广栋, 耿锐, 王昊. “魂芯”DSP 配套基础软件的自动化测试平台. 计算机系统应用, 2018, 27(5): 91-94. <http://www.c-s-a.org.cn/1003-3254/6319.html>

Automatic Testing Platform for Basic Software of HXDSP

LIN Guang-Dong, GENG Rui, WANG Hao

(Integrated Circuit Development Center, The 38th Research Institute of China Electronics Technology Group Corporation, Hefei 230088, China)

Abstract: This paper introduces a replay style testing tool that could automatically test HXDSP related basic software kits. The testing tool mainly tests debugger host software, and can also test other related software such as compiler, simulator, and so on. It is convenient to add test cases and relay test cases by groups on the testing tool. The testing tool will significantly reduce workload of testing basic software kits on HXDSP.

Key words: software testing; debug; automatic testing

“魂芯”DSP 是由中国电子科技集团公司第 38 研究所自主研发的一款高性能浮点处理器芯片. 该芯片配套有若干个自主开发的基础软件单元, 如编译器、主机调试器、软件模拟器等等^[1,2]. “魂芯”芯片配套的各软件单元是一个紧密结合的整体, 其中任何一个软件单元产生错误都会使整体开发环境产生错误. 这些软件单元经常会修复一些自身的错误, 但常常带来另外一些新的问题. 或者单个软件单元的修改导致其与其他软件单元不再匹配, 进而使整个系统无法工作. 发现与解决这些问题非常耗费时间和人力. 虽然存在测试工具和方法对某个单独的软件单元进行测试, 但是缺少一种对整体软件环境进行测试的工具^[3-5].

为解决以上问题, 设计并实现了一个以主机调试器软件为测试核心的自动化测试平台, 称为: 回放式自动化测试平台. 回放式自动化测试平台是一个基于命令行接口对“魂芯”DSP 主机调试器软件进行自动化测试的平台. 该平台支持以命令行方式添加测试用例, 并自动化地将测试用例中的调试记录回放. 若回放结果与添加测试用例时的输出结果不一致, 则认为测试失败; 若全部一致, 认为测试成功. 由于通过主机调试器软件调试的应用程序与编译器、软件模拟器都有关系, 本平台也可以同时测试其他软件单元. 本平台支持批量式回放测试记录, 当主机调试器、编译器、模拟器等任何相关的软件单元发生变化时, 可批量回放测试

① 收稿时间: 2017-08-18; 修改时间: 2017-09-05; 采用时间: 2017-09-11; csa 在线出版时间: 2018-04-23

记录,以测试软件系统的整体正确性。

1 设计

1.1 被测软件单元介绍

被测软件单元是“魂芯”DSP 配套的一系列软件单元,包括:C编译器、汇编工具链、主机调试器、软件模拟器、调试链接服务软件等等。它们之间的关系如图1所示。

使用“魂芯”DSP 主机调试器软件的唯一接口就是命令行,而本测试平台以命令行方式进行测试,因此可以测试“魂芯”DSP 调试器的绝大部分功能。由于“魂芯”DSP 主机调试器的调试对象是使用“魂芯”DSP 编译器编译出的应用程序,因此,调试结果的正确性与编译器编译出的程序正确性、编译器生成的调试信息的正确性密切相关。因此,本平台还可以对编译器进行测试。自动化测试平台存储的测试用例不仅包括调试器的输入输出,还包括源代码程序。每次回放测试用例时,都要重新编译一次源代码程序,以实现编译器的测

试。若以软件模拟的方式运行被调试程序,则调试结果的正确性与“魂芯”芯片软件模拟器的正确性也密切相关。因此,本测试平台还可以用来测试软件模拟器。本平台支持添加基于硬件平台的测试用例,因此,也可以测试与硬件平台通信的调试链接服务软件。各软件单元的功能以及测试方式如表1所示。

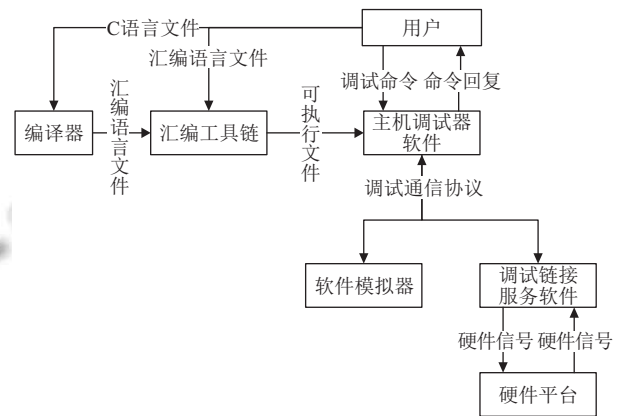


图1 被测软件单元之间的关系

表1 被测试软件单元的功能与测试方法

软件单元	功能	测试方法
C编译器	将C语言文件编译为汇编文件,并生成调试信息	通过运行程序、查看变量测试编译功能的正确性;通过查看变量、单步调试等调试操作测试调试信息的正确性
汇编工具链	将汇编文件汇编、链接为可执行程序	运行程序,测试程序的功能正确性,进而测试汇编工具链生成的机器码的正确性
主机调试器软件	执行调试操作、解析可执行文件中的调试信息	通过各种调试操作直接测试调试功能的正确性
软件模拟器	模拟芯片内核的运行	运行程序,测试程序的功能正确性
调试链接服务软件	解析调试通信协议、控制芯片执行调试操作、芯片状态上报	在真实芯片上进行调试,测试调试操作的正确性

1.2 编程语言与基础设施

回放式测试平台使用 TCL (Tool Command Language) 脚本语言编写。TCL 语言是一种解释型语言。在 TCL 语言中,所有的变量都可以看成是字符串。TCL 语言集成了大量对字符串进行处理的功能,且语言本身就支持正则表达式的解析。选择 TCL 语言是因为它进行字符串处理的功能很强,而本测试平台就是基于字符串同被测对象进行交互的。

回放式测试平台还使用了 Expect 软件组件。Expect 是一种基于 TCL 语言的支持与其他软件进行交互的软件组件。基于 Expect,可以使用常规的 TCL 语言,也可以使用 Expect 提供的额外功能。使用 Expect,可以方便地向其他软件发送字符串,然后对其他软件发回的

字符串进行解析。Expect 已经构建了一个软件框架,可以向第三方软件发送命令,然后期待 (expect) 第三方软件的某种格式的回复。由于回放式测试平台的测试对象是调试系统,它们之间通过字符串进行交互,因此使用 Expect 是比较合适的选择。

1.3 整体架构

使用自动化测试平台分为两个阶段:录制 (record) 阶段和回放 (replay) 阶段。录制阶段,回放式测试平台指导用户指定一个测试用例号、编译源文件、开始调试过程。在录制过程,用户不断输入调试命令,对调试系统、底层硬件、编译器、软件模拟器等进行测试。若用户在录制阶段发现主机调试器的输出与预期不符,用户应立即展开排查,解决问题。即用户录制到测试用

例库中的测试用例都应该是正确的、符合预期的。

自动化测试平台把发送给主机调试器的命令记录在一个文本文件中,把主机调试器反馈的回复记录在另外一个文本文件中.前者称为命令序列,后者称为回复序列.当测试平台检测到用户发送 exit 命令时,终止录制过程.每个测试用例还包括一个编译被调试应用程序的 makefile 文件.命令序列文件、回复序列文件、makefile 文件、应用程序源代码文件、测试用例号共同组成了一个测试用例.测试用例被存储在文件系统中,称为测试用例库.测试用例库的组织结构如图 2 所示. testcase 文件夹存储每个测试用例调试的应用程序源代码以及 makefile.每个测试用例号一个子文件夹,文件夹的名字就是测试用例号. testsuite 存储命令序列和回复回序文件.以 .cmdstr 为后缀的是命令序列文件;以 .resp 为后缀的是回复序列文件,文件名就是测试用例号.

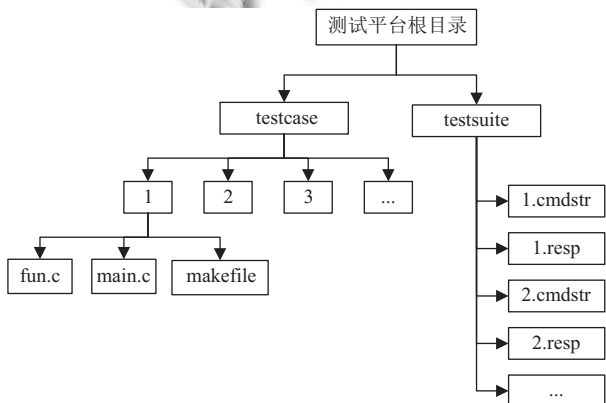


图 2 测试用例库组织结构

在回放阶段,用户只需要指定测试用例号,自动化测试平台会自动开始回放过程.测试平台首先会根据测试用例记录的 makefile 文件,运行该 makefile 文件,编译生成可执行文件.测试平台随后启动一个主机调试器软件,读取该测试用例的命令序列,发送给主机调试器软件.测试平台每发送一次命令,都等待主机调试器的回复,并将主机调试器的回复与测试用例记录的回复比较.若比较不同,则认为回放失败,立即报错,停止回放.若比较相同,则接着发送命令序列中的下一条命令,直到检测到 exit 命令为止.整个过程如图 3 所示.

1.4 常用命令缓存

每次进行调试时,需要进行与模拟器(或真实芯片)之间建立链接、选择被调试芯片、选择被调试文

件、加载文件等调试操作.这些操作需要一些固定序列的调试命令来驱动.添加基于同一份应用程序工程的测试用例时,前面若干条调试操作也经常是相同的.为了方便使用本调试平台,可在根目录下添加 common.cmdstr 文件.该文件存储每个测试用例的前面若干个调试命令,每行存储一个调试命令.添加测试用例时,首先执行 common.cmdstr 中存储的调试命令,然后再提示用户输入剩余调试命令.所有的调试命令,包括 common.cmdstr 中的调试命令和用户输入的调试命令,都会被记录到测试用例中.回放测试用例时,首先执行 common.cmdstr 中存储的调试命令,这些调试命令的回复不与测试用例回复序列进行比较.然后再执行测试用例中相应个调试命令之后的调试命令.例如,若 common.cmdstr 中有 3 条命令,则回放时,首先执行 common.cmdstr 中的 3 条命令,然后从测试用例中的第 4 条命令开始执行.通过这种方法,可以方便地进行测试用例之间的移植.例如:基于软件模拟器的内核,已经添加了一批测试用例.现在希望复用这批测试用例,将其在真实芯片上回放,只需要在 common.cmdstr 加入一行与真实芯片建立调试连接的命令即可.回放时,测试平台首先执行 common.cmdstr 中的这行命令,与真实芯片建立链接,并且不将主机调试器的回复与测试用例中的回复进行比较.之后,测试平台再从测试用例中的第二个命令开始进行执行,并将主机调试器软件的实际回复与测试用例记录的回复比较.只要被调试应用程序仅仅与处理器内核有关系,基于模拟器的调试回复与基于真实芯片的调试回复应该是相同的.若回放成功,说明本测试用例测试通过;若回放失败,说明本测试用例未能测试通过.

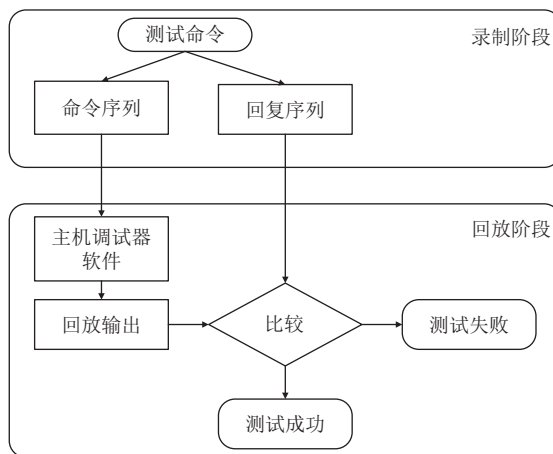


图 3 回放式自动化测试平台工作流程示意图

1.5 临时文件

为了避免编译应用程序生成的.out、.o等临时文件混杂在测试用例库中,本测试平台在添加测试用例和回放时,都会把被调试程序源代码复制到测试平台所在目录下workdir目录中,再进行编译、调试。被调试程序的makefile应位于被调试程序源代码所在目录的根目录下,makefile中的编译命令都应该是相对路径。被调试程序的整个工程源代码都应当位于该目录下。调试时,用户需要指定加载的.out可执行文件。用户必须指定位于workdir中刚刚被本测试平台编译出来的.out文件,不能指定其他文件夹中的.out文件。否则,无法进行下一步操作。

1.6 限制条件

由于回放式测试平台通过字符串与主机调试器之间进行交互,并通过字符串比较判断测试是否成功,因此本测试平台的使用具有一定的限制条件:

- 1) 不能运行scanf等接受用户输入功能的调试过程。
- 2) 调试过程中不能使用暂停命令;运行之后,再执行“暂停”调试命令,不同的平台上处理器暂停的时机不同,测试用例的回复和回放时的回复不一定相同。因此,不能运行含有“暂停”调试命令的测试用例。
- 3) 单核运行时,应在单核停止之后再行下一步调试操作。多核运行时,必须在所有的核停止之后再行下一步调试操作。
- 4) 多核运行时,若其中一个核带有printf打印操作,则运行结果有可能是不确定的。有可能是其中一个核先停下,也有可能是另一个核先停下来。因此,执行多核调试测试用例时,谨慎使用printf功能。只有当用户可以确定多核运行时每个核的输出(printf)顺序时,才添加多核运行测试用例。
- 5) 暂时不支持通过“上”、“下”方向键选择上一条调试命令。
- 6) 为了方便将测试用例在不同主机上迁移,用户输入的测试用例目录、makefile目录等等都应该使用相对目录。makefile内调用occ.exe(编译器)时应不带目录,通过环境变量找到occ.exe。

1.7 应用情况

该自动化测试平台已经提供给“魂芯”DSP配套基

础软件的测试人员使用。测试人员在该平台上添加了大量测试用例。这些测试用例包含软件模拟器调试和硬件芯片调试两类。测试用例加载调试的应用程序中包括对DSP的各项功能进行测试的应用程序。这些应用程序中也包括复杂的C语言程序,以测试编译器的正确性。这些测试用例可以一次性批量回放。测试人员反映该平台添加测试用例方便,回放测试用例更加快捷。每次发布一个“魂芯”DSP开发环境的正式版前,测试人员不必再逐项进行各项调试操作确认各功能的正确性,只需输入一条命令即可,大大减小了测试人员的工作量。

2 总结

本文介绍了一种“魂芯”DSP配套基础软件的自动化测试平台。该平台以主机调试器为测试对象,通过主机调试器与其他软件模块的交互,间接地测试其他软件单元。该平台可以方便地添加测试用例,也可以批量回放测试用例。是一个简单高效的自动化测试平台。下一步的工作方向是改进回放时的字符串匹配方式,不再简单地字符串比较,而是利用正则表达式,仅仅对关键的子串进行匹配,从而减少使用本测试平台时的限制条件。另一个工作方向是将该测试平台与集成开发环境(IDE)结合起来,将用户通过界面的调试动作记录下来,再进行批量回放,以实现更加方便地添加测试用例。

参考文献

- 1 林广栋,黄光红,耿锐.一种C语言级单步调试系统的功能实现方案.单片机与嵌入式系统应用,2015,15(2):48-51.
- 2 林广栋,朱艳,黄光红.一种调试系统中C语言复杂变量查看功能实现方案.中国集成电路,2015,24(8):41-49.
- 3 宋婷.浅谈软件测试自动化解决方案.中小企业管理与科技,2010,(7):217.
- 4 徐进.自动化软件测试的分析.信息技术,2010,(3):152-155.
- 5 郭俊.军用软件测试的影响因素及改进措施.现代计算机,2007,(11):54-55. [doi: 10.3969/j.issn.1007-1423-B.2007.11.018]