

文章编号: 1671-0576(2024)02-0025-04

# 基于 TestBed/TBrun 的高效测试执行方法

罗睿梅, 万东燕, 唐超, 辛增献, 陈 铖

(上海无线电设备研究所, 上海 201109)

**摘 要:** 在使用 TestBed/TBrun 工具执行软件单元测试用例过程中, 存在长数组无法自动赋值和结果比对导致测试效率低的问题。针对该问题, 提出了一种基于 TestBed/TBrun 的高效测试执行方法。通过在后台文件中建立可自动识别读取长数组的通用驱动函数和自动对比测试结果的比对函数, 对软件单元中的长数组输入/输出变量进行自动赋值及结果比对。经测试验证, 该方法降低了用例执行耗时, 提高了测试执行效率。

**关键词:** TestBed/TBrun; 单元测试; 通用驱动函数; 比对函数; 长数组

**中图分类号:** TP311

**文献标志码:** A

**DOI:** 10.3969/j.issn.1671-0576.2024.02.004

## Execution Method for Efficient Testing Based on TestBed/TBrun

LUO Ruimei, WAN Dongyan, TANG Chao, XIN Zengxian, CHEN Cheng

(Shanghai Radio Equipment Research Institute, Shanghai 201109, China)

**Abstract:** In the process of using Testbed/TBrun to execute test cases, the problem is found that long arrays cannot be assigned automatically which leads to low test efficiency. In order to solve this problem, an efficient test execution method based on Testbed/TBrun was proposed. Through building the universal driver functions for long array variables and the comparison functions for test results in the background files of the tool, the long array input and output variables in software units could be automatically assigned and the test results could be compared. The test results show that the proposed method can reduce executing test cases time and the test efficiency is greatly improved.

**Key words:** Testbed/TBrun; unit test; universal driver function; comparison function; long array

## 0 引言

软件测试是软件质量保证的重要手段。通过

软件测试可以及时发现软件缺陷, 保障软件的安全性和可靠性<sup>[1]</sup>。软件单元测试<sup>[2]</sup>是指对软件的基本可检测单元进行检查和验证。其测试流程是: 依托测试工具建立测试用例, 自动生成软件驱动模块和桩模块, 识别输入输出; 执行测试用例, 得到软件单元的实际运行结果; 通过对比预期结

收稿日期: 2024-02-02

作者简介: 罗睿梅(1976—), 女, 硕士, 高级工程师, 主要从事嵌入式软件测试技术研究。

果和实际结果,判别测试用例的有效性,从而验证软件单元的正确性。

TestBed/TBrun 是英国 LDRA 公司开发的一套主要用在软件测试与维护阶段、以提高软件产品质量为目标的测试工具。TestBed/TBrun 是目前被广泛采用的软件单元测试工具<sup>[3]</sup>。但在测试用例执行过程中,若软件单元的输入/输出变量为数组,且数组长度超过 50 时,TestBed/TBrun 将无法对此类长数组的输入变量自动赋值,只能采用人工方式逐一赋值。测试用例执行完成后,TestBed/TBrun 会输出长数组的实际测试结果,在进行预期结果与实际测试结果比对时,需加载数组的所有输出变量的实际测试数据,这会导致工具运行和显示缓慢,测试效率偏低。

为解决输入输出为长数组变量时测试用例执行效率低的问题,本文基于 TestBed/TBrun 测试工具,通过在后台文件中建立可自动识别读取长数组的通用驱动函数和结果比对函数,实现输入长数组变量的自动读取与输出长数组变量的自动比对,并测试验证软件单元长数组测试用例的执行效率。

## 1 TestBed/TBrun 测试工具

TestBed/TBrun 可同时支持编码规则检查、软件度量分析、数据流分析、覆盖率分析等<sup>[4]</sup>。TestBed/TBrun 的主要特性<sup>[5-7]</sup>如下:

- a) 主要用于在软件编程、软件测试与软件维护阶段提高软件产品质量,可明确定义所编软件达到的级别和所要进行的测试;
- b) 可自动生成软件测试驱动模块和桩模块,节省测试用例设计时间,提高软件测试效率;
- c) 兼容多种用 C 语言开发的嵌入式开发环境,如 CCS, VisualDSP, KerilC 等;
- d) 在运行测试用例的同时,可分析得出软件单元相应代码的测试覆盖情况,评估测试状态,从而为新的测试用例设计提供指导。

## 2 长数组测试用例高效执行方法

### 2.1 总体设计

软件单元测试主要分为测试用例设计、测试

用例执行及测试用例判别 3 个步骤<sup>[8]</sup>。为解决测试用例执行过程中长数组函数无法自动赋值与高效结果比对的问题,依据模块化和通用化的思想开展长数组测试用例高效执行方法研究。长数组测试用例执行共包括 3 个通用功能模块: Matlab 仿真数据产生模块、输入/输出通用驱动函数模块、测试结果自动比对模块,其执行流程框图如图 1 所示。

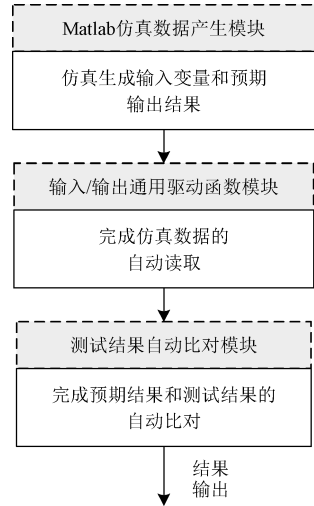


图 1 长数组测试执行流程框图

长数组测试用例执行工作流程是: Matlab 仿真数据产生模块生成长数组输入/输出数据,并按与 TestBed/TBrun 兼容的数据格式进行存储; 输入/输出通用驱动函数模块采用参数化的方式自动读取不同数据特征的 Matlab 仿真数据; 测试结果自动比对模块对长数组输出期望结果和实际测试结果进行自动比对,通过比对结果,判断测试用例是否执行通过。

### 2.2 Matlab 仿真数据产生模块

Matlab 仿真数据产生模块主要用于软件单元长数组输入/输出数据的仿真生成。其具体过程为: 分析软件单元,确定长数组输入变量及其类型特征; 采用等价类、边界值等测试用例设计方法,设计输入数据,并用 Matlab 仿真软件生成符合设计要求和变量特征的长数组输入数据,存储成与 TestBed/TBrun 兼容的 \*.dat 格式的数据文件; 根据函数的设计说明,用 Matlab 仿真得到被测函数的长数组预期输出数据,存储成 \*.dat 格式的数据文件。

### 2.3 输入/输出通用驱动函数模块

TestBed/TBrun 在运行测试用例时,会自动调用自带的头文件。利用 TestBed/TBrun 这一特征,在被调用的头文件程序中嵌入长数组数据的通用驱动函数。输入通用驱动函数的主要功能是:完成对不同存储路径、不同数据类型、不同数据长度、不同数据精度的仿真数据的读入。其功能框图如图 2 所示。

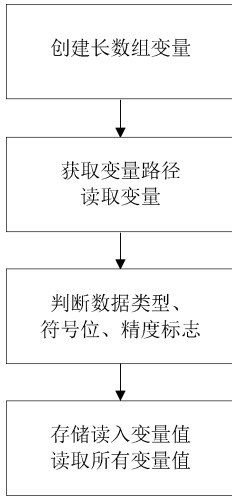


图 2 输入通用驱动函数功能框图

通用驱动函数采用参数化的设计思想,其参数包括与输入变量类型一致的变量名、数据精度位数标识、数据长度、输入数据的存储位置。在调用通用驱动函数时,TestBed/TBrun 首先按存储路径读取仿真数据,在变量缓存区对读入的数据进行逐位比较,当变量名、数据长度和存储位置发生变化时,只需对相应的参数进行调整。

在完成参数设置后,对数据类型进行判读。数据类型主要分为整数型和浮点型两种。若被读取的为整数型数据,则先判断其首位是否为符号位,再根据数据属性读取数据。若被读取的为浮点型数据,则先判断首位是否为符号位,并在读取过程中判断数据小数点的位置,依据精度要求确定数据结束的位数,完成浮点型数据的读取。

成功读取一个变量数据后,将该数据存储在输入变量数组对应的位置,并刷新缓存区,再开始读取下一个变量。输出通用驱动函数与输入通用驱动函数类似,只需在 TestBed/TBrun 的头文件中嵌入具有输出功能的模块即可。

### 2.4 测试结果自动比对模块

测试结果自动比对模块的主要功能是:采用脚本语言对测试用例执行中产生的实际测试结果和期望输出结果进行比对,判断实际测试结果是否符合要求。比对结果通过调用输出通用驱动函数直接产生,并以文件形式存储。测试结果自动比对流程如图 3 所示。根据长数组输出数据的长度,建立一个标志数组 flag[i] 用于存储比对结果, i 表示数组长度。将测试输出数据与期望输出数据逐项比对,若比对结果符合要求,则将对对应数组元素设置为 1,并将其存入标志数组对应位置;若不符合要求,则设置为 0,也将其存入标志数组对应位置。若标志数组中所有元素全为 1,则判定实际测试结果符合要求,该测试用例通过;若标志数组中元素不全为 1,则判定实际测试结果中有不符合要求的数据,该测试用例不通过。若该测试用例不通过,则需要分析原因,重新设计测试用例或者完善被测函数,再进行测试,直到标志数组中元素全为 1,完成测试。

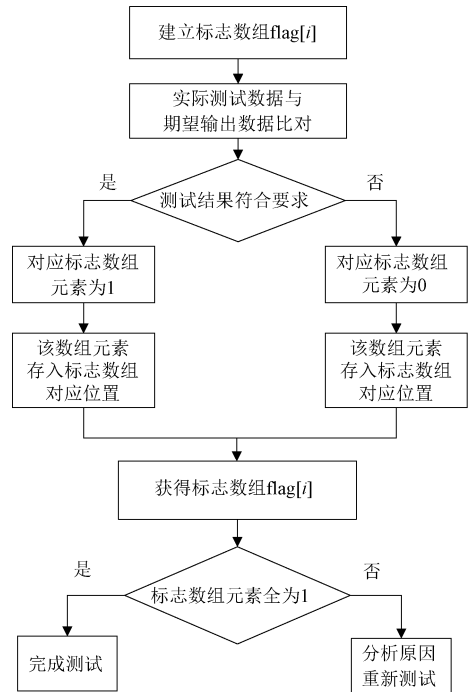


图 3 测试结果自动比对流程图

## 3 测试用例的设计与实现

### 3.1 被测函数

以恒虚警(CFAR)目标检测函数作为被测函

数设计测试用例。该函数的主要功能为根据标志数组寻找待检测区,在待检测区内完成时频域

CFAR 目标检测。时频域 CFAR 目标检测函数的输入/输出参数如表 1 所示。

表 1 时频域 CFAR 目标检测函数的输入/输出参数

序号	参数属性	参数名称	标识符	数据类型	数据长度/bit	变量属性	数据范围
1	输入	单边参考区和保护区单元总数	uUintS	无符号整数型	8	全局变量	[0,255]
2		积累脉冲数	uPNum	无符号整数型	16	全局变量	[512,8 192]
3		CFAR 检测向量	iVector	有符号整数型	8	全局变量	[0,1,-1]
4	输出	检测子区间及其参考单元起止数组	iRegion	有符号整数型	16	全局变量	[0,65 535]

该函数采用 C++ 语言编写,输入参数 iDetectVector 是一个长度为 512 的长数组,对应输出参数 iFinalRegion 也是长度为 512 的长数组,数据类型均为有符号整数型。

确定被测函数的输入/输出参数后,依据函数的设计说明,采用 Matlab 软件,按照 iDetectVector 的属性仿真得出输入长数组变量 R\_iDetectVector[512],存为 R\_iDetectVector.dat 文件;将预期的输出长数组变量 R\_iFinalRegion[512]存为 R\_iFinalRegion.dat 文件。

### 3.2 测试用例的执行

在 TestBed/TBrun 中配置测试环境,把通用驱动函数模块嵌入到 TestBed/TBrun 的自带头文件 Cpptbrunlib.h 中。

根据函数功能设计测试用例,将由 Matlab 仿真生成的长数组变量 R\_iDetectVector[512]加载为用例的输入变量,将长数组变量 R\_iFinalRegion[512]加载为预期输出变量,在 TestBed/TBrun 中运行测试用例,输出测试执行结果。若测试结果为“通过”,输出数组自动比对结果为 1,说明实际测试结果与预期相符,测试成功通过。

### 3.3 测试结果分析

对同一个长数组测试用例,分别采用本文方法与人工方法进行用例测试,用例执行过程耗时如表 2 所示。

由表 2 可知,采用本文所提方法,长数组测试的数据读入时间、结果生成及比对时间显著减少,整个用例执行时间可大幅缩短,测试用例执行效率显著提升。

表 2 测试用例执行耗时对比

测试方法	用例执行步骤耗时/min			用例执行总耗时/min
	仿真数据生成时间	数据读入时间	结果生成及比对时间	
本文方法	1.5	0.5	0.5	2.5
人工方法	≥4.0	≥12.0	≥6.0	≥23.0

## 4 结束语

本文基于 TestBed/TBrun 测试工具,提出了一种长数组输入变量自动赋值、输出结果快速比对的测试用例高效执行方法,并给出了该方法的具体实施步骤。该测试用例执行方法采用模块化、通用化设计,适合不同类型、不同长度、不同精度的数组变量。经测试验证,该方法缩短了测试用例执行耗时,提高了测试用例执行效率,这对提升软件测试质量具有积极意义。

### 参考文献

- [1] 单锦辉,姜瑛,孙萍.软件测试研究进展[J].北京大学学报(自然科学版),2005,41(1):134-145.
- [2] 傅瑞华,李凡,王俊峰.基于损失函数的单元测试用例自动化生成算法研究与实现[J].四川大学学报(自然科学版),2022,59(3):032002.
- [3] 胡丹瑞.基于 LDRA Testbed 的软件静态测试研究与实现[J].计算机安全,2012(6):69-71.
- [4] 王泉.基于 Tornado 的嵌入式软件单元测试[J].计算机工程与设计,2012,33(7):2683-2689.
- [5] 张新玉,李寅龙.TBrun 在航天嵌入式汇编软件测试中的应用[J].测控技术,2016,35(11):103-107.

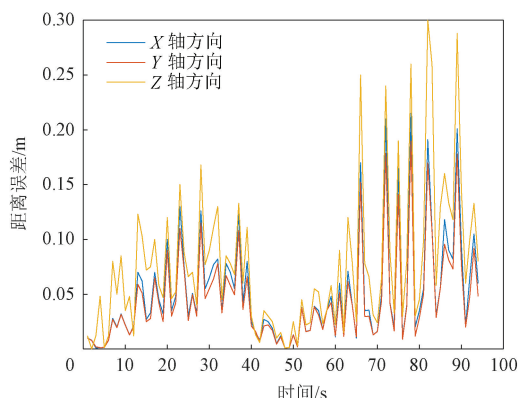


图 3 通用算法与简化算法的距离误差计算结果比较图

表 1 通用算法与简化算法计算量对比

算法	运算次数		
	乘法	除法	加法
通用算法	93	6	87
简化算法	14	6	9

综上所述,简化算法在计算误差与计算量上都能满足星上计算的需求。

### 3 结论

本文介绍了星上天线波束指向角的简化算法,该算法与星上天线波束指向角通用算法最大的区别在于转换坐标系的选择。通用算法采用 J2000 坐标系,该坐标系与时间密切相关,且坐标

转换过程中涉及地球极移模型和章动岁差模型的选取问题,计算过程复杂;简化算法通过建立当前时刻与 CGCS2000 坐标系重合的惯性坐标系 I,避免了通用算法中的复杂模型的选取与计算消耗,实现了从 CGCS2000 坐标系到轨道坐标系 O 的快速转换。

### 参考文献

[1] 汪春霆,翟立君,徐晓帆. 天地一体化信息网络发展与展望[J]. 无线电通信技术, 2020, 36(2): 493-504.

[2] 阎鲁滨. 星载相控阵天线的技术现状及发展趋势[J]. 航天器工程, 2012, 21(3): 11-16.

[3] 李广宇. 天球参考系变换及其应用[M]. 北京: 科学出版社, 2010.

[4] 张挥卫,郑勇,马高峰. GCRS 与 ITRS 之间的坐标转换研究[J]. 大地测量与地球动力学, 2011, 31(1): 63-67.

[5] 高龙超. 定向通信系统天线波束指向误差分析[J]. 航空电子技术, 2021, 52(4): 56-61.

[6] 张毅,肖龙旭,王顺宏. 弹道导弹弹道学[M]. 长沙: 国防科技大学出版社, 2005: 5-30.

[7] 郝辉,李雪瑞,舒健生,等. 导弹常用空间直角坐标系间转换方法[J]. 四川兵工学报, 2013, 34(2): 18-20.

[8] 严恭敏,戴晨杰,陈若彤. 地球自转模型误差对高精度惯导系统定位精度的影响分析[J]. 中国惯性技术学报, 2022, 30(2): 153-158.

(上接第 28 页)

[6] 刘荔娜. 基于 LDRA TBrun 软件集成测试的研究[J]. 电脑与电信, 2020(4): 64-67.

[7] 黄晨,董燕,于倩,等. 基于目标码的测试覆盖不可达分析方法[J]. 测控技术, 2017, 36(1): 100-

103, 107.

[8] 刘高军,钱程. 基于边界值分析法的 CNONIX 标准测试用例生成方法研究[J]. 工业技术创新, 2015, 2(2): 228-233.

(上接第 44 页)

[6] JOHNSON D, MALTZ D. Dynamic source routing in ad hoc wireless networks[J]. Mobile Computing, 1996, 335(1): 153-181.

[7] PERKINS C E, BELDING-ROYER E M, DAS S R. Ad hoc on-demand distance vector (AODV

routing[Z]. Reston: The Internet Society, 2003.

[8] 陈勇. 有限状态机的建模与优化设计[J]. 重庆工学院学报(自然科学版), 2007, 21(5): 55-58.

[9] 李森. 面向 FPGA 的 DSR 路由表项设计与实现方法[J]. 电子技术应用, 2018, 44(12): 89-92.