

一种基于ATE的提高DSP测试覆盖率方法

陈培华¹, 魏领会^{2*}

¹北京冠中集创科技有限公司, 北京

²北方工业大学, 北京

Email: *weilinghui@126.com

收稿日期: 2021年4月28日; 录用日期: 2021年5月22日; 发布日期: 2021年5月31日

摘要

DSP功能高覆盖率的测试是一个比较难的课题。本文以自主研发CATT-400大规模集成电路测试系统为测试平台, 提出一种基于ATE (Automatic Test Equipment)的DSP的高覆盖率的测试方法; 通过基于美国TI公司TMS320F28xx系列DSP, 实现该测试方法开发验证, 并实现自动测试设备与测试向量的自动匹配。该方法对于有微处理器的芯片测试有很大的推广意义。

关键词

CATT-400, TMS320F28xx, ATE

A Method of Improving DSP Test Coverage Based on ATE

Peihua Chen¹, Linghui Wei^{2*}

¹GZICTEST Technology Co., Ltd., Beijing

²North China University of Technology, Beijing

Email: *weilinghui@126.com

Received: Apr. 28th, 2021; accepted: May 22nd, 2021; published: May 31st, 2021

Abstract

It is difficult to test the high coverage of DSP function. In this paper, the CATT-400 large-scale integrated circuit test system developed by ourselves is used as the test platform, and a high coverage test method based on ATE (automatic test equipment) DSP is proposed. Through the TMS320F28xx series

*通讯作者。

DSP of TI Company in the United States, the test method is developed and verified, and the automatic matching between the automatic test equipment and the test vector is realized. This method is of great significance for chip test with microprocessors.

Keywords

CATT-400, TMS320F28xx, ATE

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

目前, 进口的 DSP 处理器都是封装测试后的芯片, 对于一些可靠性要求比较高的像航天、汽车电子等领域, 需要对进口来的芯片进行二次测试; 但测试模式对国内用户保密, 导致芯片在二次测试时无法通过进入测试模式实现对芯片的测试。国内也有一些对该类型进行测试的方法研究。例如基于应用模式的 DSP 测试方法[1], 提出了使用测试机代替实装测试板测试的新方法; 针对 MCU 测试, 基于 ATE 的一种 MCU 测试方法[2], 提出一种创新型测试方法, 能够大大减少 MCU 重复烧写次数; DSP 结构与测试方法研究[3], 研究了由测试系统模拟外部程序存储器加载程序的并行测试算法。基于覆盖率驱动的高性能指令集验证方法[4], 利用面向对象特性、随机约束求解机制以及覆盖率统计机制, 提出一种快速搭建覆盖率驱动的随机测试平台的方法。目前看到测试方法基于理论性的研究比较多, 在实现方面应用性不强; 针对实用性和易实现性, 本文提出一种新的基于 ATE 的 DSP 测试解决方案, 该方案既可以实现 DSP 的高覆盖率测试要求, 也能较简单地实现。并且在 CATT-400 ATE 测试设备中验证了该方案的可行性和可靠性; DSP 与测试系统是两个不同的时钟源, 这个也是测试的难点, 也是本论文解决的问题。

2. 测试概述

2.1. CATT-400 测试系统介绍

CATT-400 是由北京冠中集创科技有限公司自主研发的大规模集成电路数字测试系统, 系统的最大优点就是数模混合测试系统, 可以进行 DC 测试、功能测试和 AC 测试, 具有测试精度高、速度快等特点。系统设计采用模块化结构设计, 每通道板可提供 64 路管脚电路和 8 路 DPS 电路, 系统最大配置为 8 块通道板 512 路测试通道, 最大测试速率为 400 MHz, 系统具有 Per Pin PMU 特性。本测试系统有较强的失效分析功能, 设计有每通道失效存储器, 用于失效分析, 失效向量深度可达到 128 M。系统具有动态定时功能, 每通道可以设置时间参数, 时间分辨率为 1.25 nS。适用于 IC 生产厂家进行芯片测试或成品检验。目前该系统已经用在高端 DSP、FPGA、SOC 等芯片的测试。该系统的结构框图如图 1 所示。系统采用光纤通讯方式可以解决通信干扰和误码率的问题, 增加系统控制的可靠性。

2.2. DSP 测试介绍

DSP 是一款用于控制的高性能、多功能、高性价比数字处理芯片。芯片内部结构比较复杂, 外围单元比较多, 增加了测试的复杂性和测试覆盖率问题。对该 DSP 芯片既要进行直流参数测试, 功能测试, 也要对该芯片的交流参数测试。

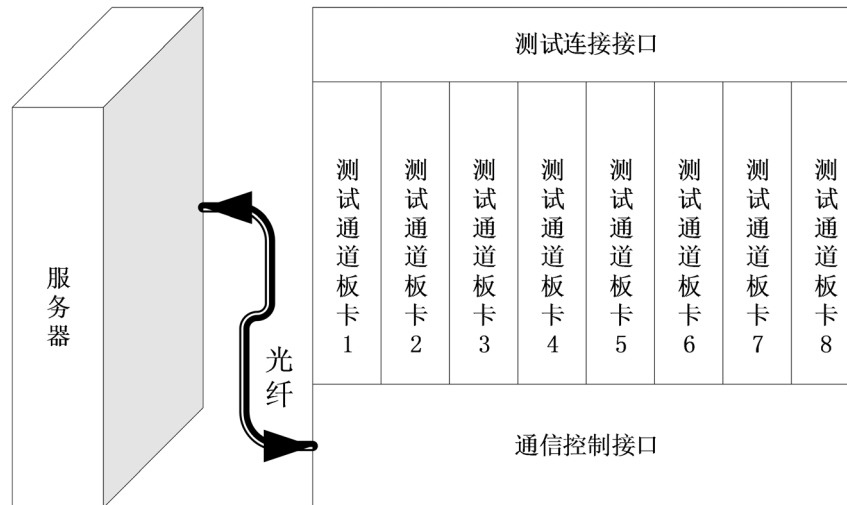


Figure 1. Structure diagram of test system
图 1. 测试系统结构图

直流参数包括: GPIO 管脚以及其它功能管脚的开短路测试、漏流测试、电平测试、动态功耗和静态功耗测试等;

功能测试: GPIO 管脚的输入输出功能测试、各个协议(SPI、SCI 等)功能测试、指令集、外部总线接口等测试;

交流参数测试: 主要是 datasheet 里面的各个管脚的上升时间, 管脚之间的时间延时测试、各种协议的时序测试等。

每一项测试都要编写测试程序, 并要考虑测试覆盖率的问题, 覆盖率越高, 对芯片测试的可靠性也越高。DSP 测试项目如表 1 所示。

Table 1. Project requirements DSP test
表 1. DSP 测试项目要求

| DSP 测试项 | 测试内容 |
|---------|------------------------------|
| 直流测试 | 开短路测试、漏流测试、电平测试、功耗测试等 |
| 功能测试 | GPIO 功能测试、外部接口功能、指令集测试、协议测试等 |
| 交流参数测试 | 上升时间、下降时间、延时等 |

3. DSP 测试平台

本测试以美国 TI 公司的 TMS320F28xx 芯片作为开发产品介绍。

3.1. TMS320F28xx 结构图[5] [6]

结构框图如图 2 所示, 它的基本功能如下:

- SCI (串行通信接口)
- SPI (串行外设接口)
- UART (通用串行接口)
- 12-bit, 16 通道 ADC
- Mc BSP (多通道带缓冲串行口)

- eCAN (增强型 CAN 模块)
- 独立的 32 位定时器
- 独立编程的 GPIO 管脚
- 外部扩展接口

通过结构框图可以看出这一类型的 DSP 芯片的功能比较强大、比较多; 每种功能又包括很多工作模式和参数设置, 在正常应用这些功能时, 这些工作模式都可能用到, 在测试时, 要把这些模式都覆盖掉。像 SCI 接口测试, 即包括可编程波特率又包括可编程数据格式; 在对这个功能测试时, 要有多种组合, 不同波特率和不同数据格式组合, 实现对该功能的全覆盖测试。

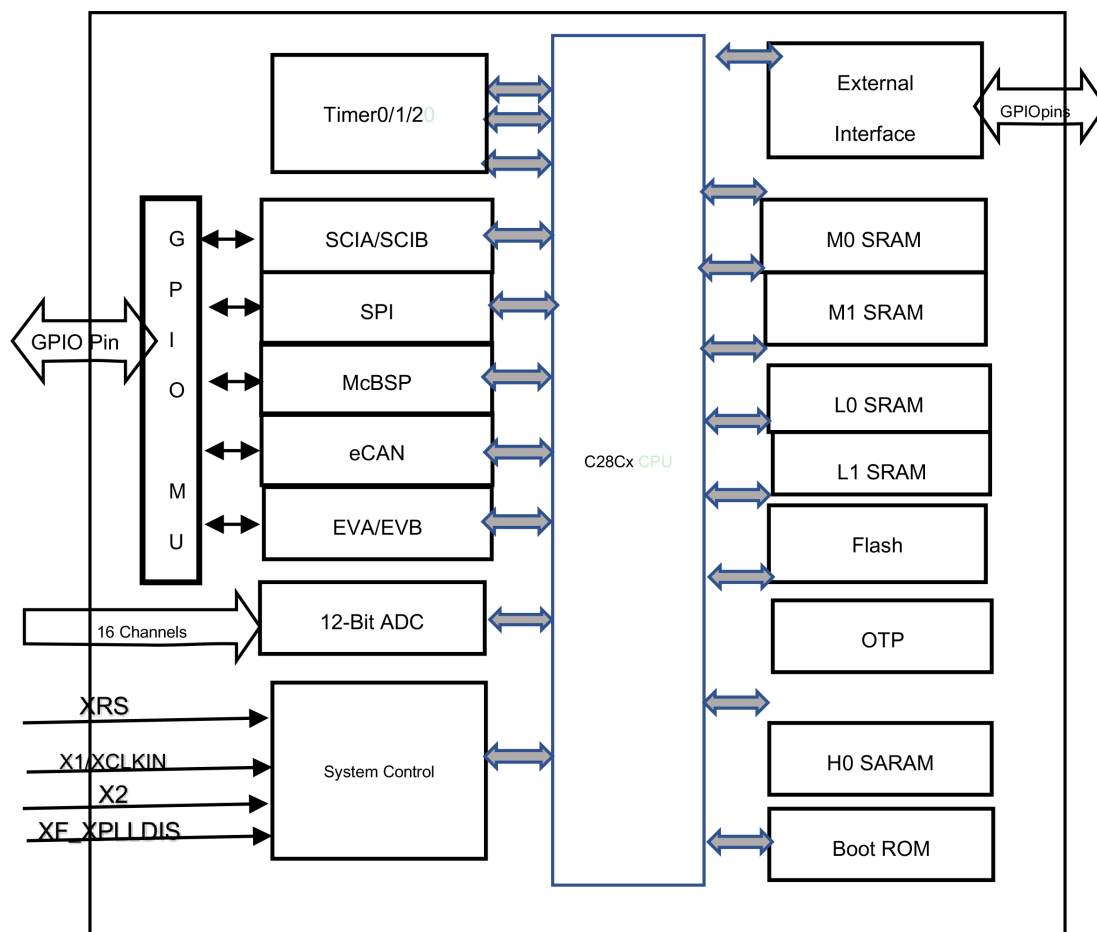


Figure 2. Structure diagram
图 2. 结构框图

3.2. 测试硬件组成结构

在测试过程中, 被测芯片 DSP 是放在测试夹具里面, 芯片的引脚通过测试夹具的进行连接。测试夹具是焊接在一个 PCB 板上, 夹具的每个管脚通过与 PCB 板上的插座相连, PCB 板上插座插在 CATT-400 测试系统上, 与测试系统的测试接口连接。TMS320F28xx 的测试资源主要用到测试系统的 IO 通道、DPS 通道和地。芯片的 GPIO 管脚及其它功能的管脚与测试系统的 IO 通道连接, 电源 VCC 与测试系统的 DPS 通道连接, 地与测试系统的地连接, 连接接口如图 3 所示。

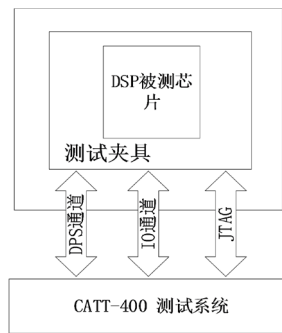


Figure 3. The connection of test hardware
图 3. 测试硬件连接

4. 测试方法介绍及实现

对于测试直流和 AC 测试实现起来比较容易, 覆盖率都是固定的, 该方法就不做介绍了。下面就对功能实现方法进行介绍。

4.1. 需要开发工具

在测试开发过程中, 主要用到以下几个工具:

CCS Studio 3.3: TI DSP 开发环境, 编译环境是 C 语言[5] [6] [7] [8] [9];

JTAG: TI DSP 仿真器, 可以把 DSP 生成的程序通过 JTAG 下载到 DSP 中去;

Catt-400 tool: CATT-400 开发环境; 包括测试程序开发、向量生成工具、校准工具等。

4.2. 测试方法介绍

测试方法对于 dsp 部分来说[10] [11], 都是在 C 语言环境下进行的, 比较容易掌握。对于 TMS320F28xx 系列来说, 需要功能测试项如表 2 所示。

Table 2. Test function item

表 2. 测试功能项

| 功能 | 测试需求 |
|--------------------|------------------------|
| JTAG 功能 | JTAG 下载, 烧录功能 |
| SPI | SPI 接口协议测试 |
| SCI | SCI 接口协议测试 |
| UART | UART 接口协议测试 |
| eCAN | eCAN 协议测试 |
| GPIO | IO 输入输出功能测试, 驱动力测试 |
| 12-bit ADC | ADC 采集数据精度测试 |
| External interface | 外部数据接口, 包括数据访问, 存储单元访问 |
| 指令集 | CPU 运算单元的运算能力, 及准确性测试 |
| 存储单元 | 各个存储单元可靠性测试 |

芯片测试分两类[12], 一类是外部接口部分测试, 该部分测试是在 CCS 开发环境中把接口的功能用 C 语言编程, 包括每个功能的寄存器设置、协议发送和接受数据的格式、数据的模式等。在与测试机交互部分编写 pattern, 每段 pattern 对应 C 语言的每个功能, pattern 对应功能的协议和数据与 C 语言编写的协议要对应上。

一类是芯片内部存储单元和运算单元, 这类功能用 C 语言编程, 对指令部分, 编写复杂的运算包括一些算法操作数的组合和复杂的浮点运算, 使 DSP 的算法指令都给覆盖上; 内部存储单元采用往存储单元里写一些有规律的数据进行访问, 保证每个存储单元都被访问到; 每个功能测试的结果通过 GPIO 接口传到测试系统, 由测试系统的 pattern 来判断结果是否正确。

在这些测试方法中, 对于 C 编程测试部分是把所有的功能都和在一个程序里, 每个功能的执行通过相应的 GPIO 高低电平来控制的, 这样可以解决 DSP 与测试设备不同时钟源的问题; 外部通过测试机的 pattern 控制对应的 GPIO, 来执行不同的功能。

0——该功能不执行

1——该功能执行

4.3. 测试方法实现

实现主要有以下几个步骤:

1) 根据硬件连接图, 确定每个芯片管脚与软件对应关系; 软件对应关系是通过 excel 进行列出来的, 如图 4 所示。

2) 根据 DSP 功能, 编写 CCS 程序; 首先选择 GPIO 控制管脚, 选择的这个 GPIO 管脚为输入管脚。例如选在 GPIOA0; 在 CCS 里编程如下:

| A | B | C | D | E | F | G | H | I | J | K |
|----------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-------|
| channel_map_v1 | | | | | | | | | | |
| ChNo | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | |
| ChName | RST | CLK | IO | TST16 | SWPIO | GPIO3 | GPIO4 | GPIO5 | GPIO6 | GPIO7 |
| ChType | IO_INPUT | IO_INPUT | IO_INPUT | IO_INPUT | IO_INPUT | IO_INPUT | IO_INPUT | IO_INPUT | IO_INPUT | IO_IN |
| ChVDC | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 150049 | 150050 | 150052 | 150051 | 150054 | 150058 | 150059 | 150053 | 150055 | 1500 |
| 1 | 150065 | 150066 | 150068 | 150067 | 150072 | 150074 | 150075 | 150069 | 150070 | 1500 |
| 2 | 150193 | 150194 | 150196 | 150195 | 150198 | 150202 | 150203 | 150197 | 150199 | 1502 |
| 3 | 150209 | 150210 | 150212 | 150211 | 150216 | 150218 | 150219 | 150213 | 150214 | 1502 |
| 4 | 130241 | 130242 | 130244 | 130243 | 130246 | 130250 | 130251 | 130245 | 130247 | 1302 |
| 5 | 130257 | 130258 | 130260 | 130259 | 130264 | 130266 | 130267 | 130261 | 130262 | 1302 |
| 6 | 130193 | 130194 | 130196 | 130195 | 130198 | 130202 | 130203 | 130197 | 130199 | 1302 |
| 7 | 130209 | 130210 | 130212 | 130211 | 130216 | 130218 | 130219 | 130213 | 130214 | 1302 |
| 8 | 110001 | 110002 | 110004 | 110003 | 110006 | 110010 | 110011 | 110005 | 110007 | 1100 |
| 9 | 110017 | 110018 | 110020 | 110019 | 110024 | 110026 | 110027 | 110021 | 110022 | 1100 |
| 10 | 110049 | 110050 | 110052 | 110051 | 110054 | 110058 | 110059 | 110053 | 110055 | 1100 |
| 11 | 110065 | 110066 | 110068 | 110067 | 110072 | 110074 | 110075 | 110069 | 110070 | 1100 |
| 12 | 70049 | 70050 | 70052 | 70051 | 70054 | 70058 | 70059 | 70053 | 70055 | 700 |
| 13 | 70065 | 70066 | 70068 | 70067 | 70072 | 70074 | 70075 | 70069 | 70070 | 700 |
| 14 | 70193 | 70194 | 70196 | 70195 | 70198 | 70202 | 70203 | 70197 | 70199 | 702 |
| 15 | 70209 | 70210 | 70212 | 70211 | 70216 | 70218 | 70219 | 70213 | 70214 | 702 |
| 16 | 50001 | 50002 | 50004 | 50003 | 50006 | 50010 | 50011 | 50005 | 50007 | 500 |
| 17 | 50017 | 50018 | 50020 | 50019 | 50024 | 50026 | 50027 | 50021 | 50022 | 500 |

Figure 4. Channel correspondence

图 4. 通道对应关系

```
while(GpioDataRegs.GPADAT.bit.GPIOA0==0)
{
while(1)
{
If (GpioDataRegs.GPADAT.bit.GPIOA0==1)
{
{
加入测试功能
```

```

    }
}
else
{
    break;
}

```

3) 编程测试 pattern 和测试程序

根据 DSP 功能, 来编写测试机 pattern; 例如: SPI 协议, 可以利用 pattern 生成工具生成 TMS320F28xx 的测试机可以执行 pattern。利用对应的 GPIO 的通道进行控制是否执行 SPI 测试协议; 在 CATT-400 环境下编写测试程序; CATT-400 环境如图 5 所示。

4) 调试

前面生成的文件有两个, 一个是 ccs 生成的.out 文件, 这个是 DSP 的可执行文件, 一个是测试机的测试程序文件。在调试过程中是这两个程序相互配合来完成。调试过程中执行以下几个操作:

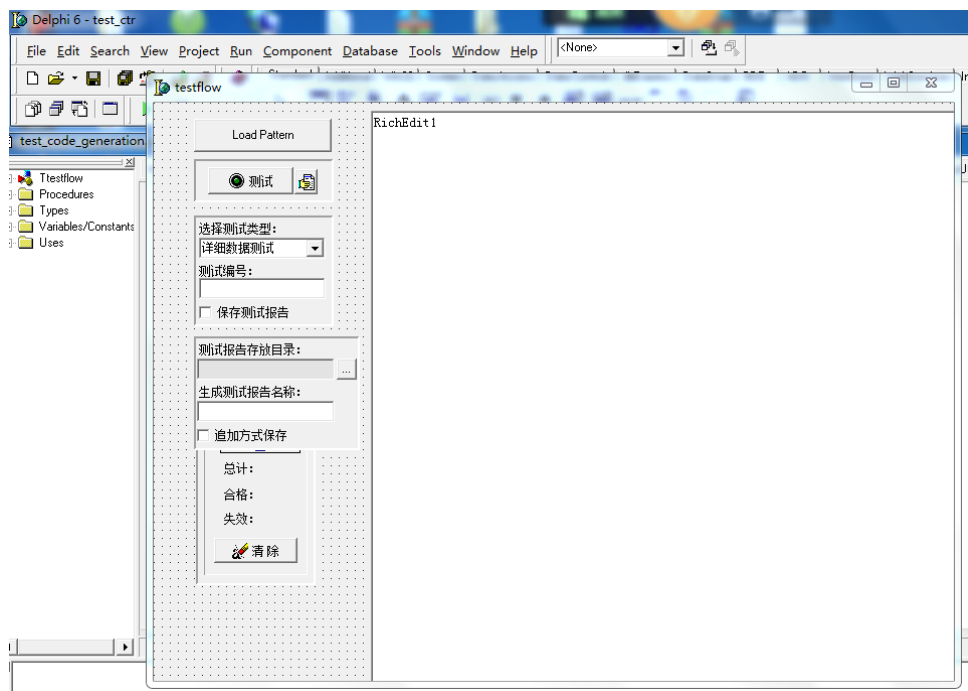


Figure 5. The test environment

图 5. 测试环境

- 把 CCS 生成的 DSP 程序通过 JTAG 下载到 DSP 中去, 运行程序;
- 在 CATT-400 debug 模式下进行 debug 某一项功能, 看测试结果, 是否 pass, 如果 fail 再修改, 直至 pass;
- 调试其它功能, 直至把所有的功能都调试完;
- 把所有的测试项放在一个测试流程里, 看测试是否全 pass, 如果 fail, 调试一直 pass 为止;
- 收集数据, 看稳定性, 如果稳定了, 就可以进行产品测试了。

通过以上几个流程的开发就可以实现对 DSP 的高覆盖率测试, 流程图如图 6 所示。

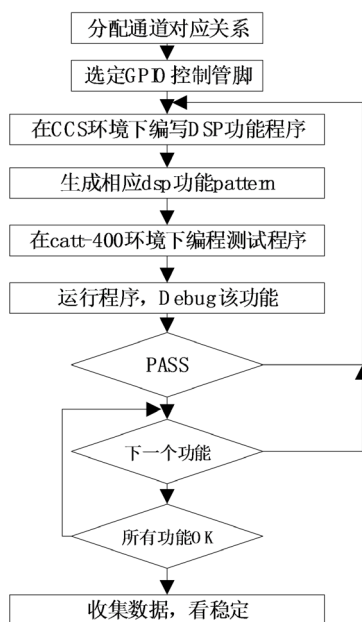


Figure 6. Test development process
图 6. 测试开发流程

4.4. 测试结果分析

通过测试流程编程的方法, 编写测试程序, 在测试机台上进行测试验证, 测试结果见图 7 显示部分测试结果; 通过测试结果验证了该测试方法的可行性, 并且把 DSP 的功能测试都能实现, 提高了测试的覆盖率。

```

*****SPI接口协议测试*****
SPI 接口测试          PASS
time                  24sec
*****SCI接口协议测试*****
SCI接口协议测试      PASS
time                  300sec
*****UART接口协议测试*****
UART接口协议测试    PASS
time                  200sec
*****eCAN协议测试*****
eCAN协议测试        PASS
time                  128sec
*****IO输入输出功能测试, 驱动力测试*****
IO输入输出功能测试, 驱动力测试    PASS
time                  70sec
*****ADC采集数据精度测试*****
ADC采集数据精度测试    PASS
time                  95sec
*****外部数据接口数据访问测试*****
外部数据接口数据访问测试    PASS
time                  88sec
*****外部数据接口存储单元访问测试*****
外部数据接口存储单元访问测试    PASS
time                  76sec
  
```

Figure 7. The test results
图 7. 测试结果

5. 结论

该文提出了一种解决 DSP 测试的方法, 该方法既解决了 DSP 原来测试覆盖率低的问题, 可以通过与测试机的配合, 完成 DSP 各种功能的程序编写并在机台上验证, 也为不同时钟源的测试提供了一种新的解决方法; 利用 C 语言对 DSP 进行测试编程比较方便, 更接近于芯片的应用状态; 提高了对芯片测试可靠性的测试要求。

参考文献

- [1] 武乾文, 奚留华, 郭晓宇. 一种基于应用模式的 DSP 测试技术[J]. 电子与封装, 2018(5): 12-14+22.
- [2] 何立, 潘宇. 基于 ATE 的一种 MCU 测试方法[J]. 电子质量, 2021(1): 30-33.
- [3] 武乾文, 冯妍. DSP 结构与测试方法研究[J]. 计算机与数字工程, 2010(9): 84-87.
- [4] 刘畅, 郭阳. 基于覆盖率驱动的高性能指令集验证方法[J]. 计算机工程, 2014, 40(6): 317-321.
- [5] 于明. TMS320F281x Digital Signal Processors [J]. 电子测试, 2016(13): 9-12.
- [6] 孙丽明. TMS320F2812 原理及其 C 语言程序开发[M]. 北京: 清华大学出版社, 2008.
- [7] 杨珂瑶, 张小芳, 曾雷杰. 基于 DSP 的嵌入式软件测试方法[J]. 计算机与现代化, 2014(10): 61-65.
- [8] 边倩, 林智慧. 用 C 语言实现 DSP 程序设计的研究[J]. 电子技术与软件工程, 2016(23): 251.
- [9] 朱业腾. 基于 DSP 环境下 C 语言的编程优化[J]. 电子测试, 2017(20): 41+27.
- [10] 高剑, 冯建科, 郭士瑞. DSP 功能测试方法研究[J]. 计算机与数字工程, 38(9): 28-40.
- [11] 武乾文, 冯妍. DSP 运算方法研究[J]. 电子与封装, 2011(7): 24-28.
- [12] 王芳. 集成电路芯片测试[M]. 杭州: 浙江大学出版社, 2014.