

# 单板级JTAG 测试技术

■ 王承 刘治国 编著



国防工业出版社  
National Defense Industry Press

# 单板级 JTAG 测试技术

王 承 刘治国 编著

國防工業出版社

· 北京 ·

## 内 容 简 介

本书是一本系统论述单板级 JTAG 测试技术的专著。内容包括：基于 IEEE1149.1 标准的边界扫描测试、可测性设计和测试功能及串行测试矢量；内建自测试和仿真测试；基于 IEEE1687 标准的集成电路测试技术发展趋势。

本书适合于从事集成电路开发的工程技术人员阅读，对于电子测量、通信工程和电路系统学科的学生，也有一定的参考作用。

### 图书在版编目 (CIP) 数据

单板级 JTAG 测试技术 / 王承, 刘治国编著. —北京:  
国防工业出版社, 2015. 6

ISBN 978 - 7 - 118 - 09986 - 7

I. ①单... II. ①王... ②刘... III. ①集成电路 - 测试技术 IV. ①TN407

中国版本图书馆 CIP 数据核字 (2015) 第 083722 号

※

国防工业出版社出版发行  
(北京市海淀区紫竹院南路 23 号 邮政编码 100048)  
国防工业出版社印刷厂印刷  
新华书店经售

\*

开本 880 × 1230 1/32 印张 6 $\frac{1}{4}$  字数 190 千字  
2015 年 6 月第 1 版第 1 次印刷 印数 1—2000 册 定价 58.00 元

(本书如有印装错误, 我社负责调换)

国防书店: (010) 88540777

发行邮购: (010) 88540776

发行传真: (010) 88540755

发行业务: (010) 88540717

# 前 言

测试是人们认识客观世界、取得定性或定量信息的基本方法,是信息工程的源头及重要组成部分。科技发展的日新月异,测试工作将处于各种装备系统设计和制造的首位,并成为生产率、制造能力即实用性水平的重要标志。当前,测试成本已达到所研发设备系统总成本的50%,甚至70%,而且编制测试程序所花的时间也远超系统设计所花的时间。因此,在未来的激烈竞争过程中,测试技术将把现代装备系统的设计和制造构成一个完整的整体,是保证现代装备系统实际性能指标的重要手段。

测试的作用贯穿产品的整个生命周期,即研发、生产、现场维护及故障返修等,为产品质量提升保驾护航,是客户满意度的有力保障。测试是生产周期中的最重要阶段之一,是产品质量的重要保证。在产品生产制造过程中,必须进行有效的测试,以降低产品的不良率,确保产品质量的控制。因此,在产品的研发阶段,就应当对测试策略进行论证和确定,以简化后续的生产测试流程,提升产品的测试覆盖率,还会使产品具有更低的测试成本和更高的可靠性。

电子技术的进步,如芯片封装技术、单板设计技术和高速电路设计,给传统的硬件接触测试方法及自动测试系统带来了大量难题。这主要表现为两个方面:对于芯片而言,管脚被隐藏于下方的硅芯片上,无法实现接触和探测,如球栅阵列(BGA)的芯片级封装;对于单板而言,高复杂度和高集成度芯片的大量应用,使得单板密度大大增加,没有空间增加测试焊盘,特别是对于高速I/O总线,测试焊盘容易引起电容耦合效应和反射,会对信号质量产生影响,目前在新的电路板设计

中,已经完全放弃了测试焊盘。上述种种发展和需求,给单板级 JTAG 测试技术的发展带来了契机:一方面,促使 JTAG 测试技术在工业领域得到了广泛应用,特别是基于 IEEE1149.1 标准的边界扫描测试技术的普及和推广;另一方面,推动了 JTAG 测试技术不断进步,新的国际标准不断推出,以满足电子技术迅猛发展的需要,解决工业界所面临的测试难题和困境。该书的内容主要分为以下四个部分。

第一部分讲述测试的基本概念和单板级 JTAG 测试技术,涉及第 1 章和第 2 章。第 1 章阐述数字电路测试的基本任务和基本方法;在电路敏化分析的基础上,介绍组合电路和时序电路的 D 算法及测试矢量生成;在可测性概念和定义的基础上,介绍基于边界扫描的设计方法。第 2 章对单板级 JTAG 测试的基本概念和相关技术进行了概述,以便于对单板级 JTAG 测试形成完整的概念。

第二部分讲述边界扫描测试技术标准及其应用,涉及第 3 章至第 6 章:第 3 章详细介绍了基于 IEEE1149.1 边界扫描测试的硬件结构和测试指令及边界扫描描述语言,在此基础上对目前工业界应用较为广泛的 IEEE1532 和 IEEE1149.6 进行了阐述;第 4 章介绍了单板级边界扫描测试的可测性设计,以便于在后期的单板大批量生产测试的实施和覆盖率提升;第 5 章介绍了边界扫描技术能够实现的测试功能项,以便于对该技术的整体把握;第 6 章介绍了在业界一种通用的测试矢量——串行测试矢量格式。该部分有助于在工程应用的实现,对实际单板的测试具有指导作用。

第三部分讲述两种高速 JTAG 测试方法,以解决传统边界扫描技术的测试速度低的不足,它包括第 7 章和第 8 章。第 7 章阐述了内建自测试的基本概念、原理和应用情况。第 8 章介绍了片上仿真测试的方法,通过 JTAG 端口配置处理器的内部寄存器,对芯片内部仿真模块的控制,从而实现处理器周边器件的实速测试。

第四部分篇幅不多,仅包括第 9 章内容,讲述单板级 JTAG 测试技术的发展趋势——嵌入仪器,主要针对 IEEE1687 国际标准,对相关的

概念、定义和实现方法进行阐述。

本书具有一定的理论性和实用性,反映了当前工业界单板级 JTAG 测试技术的最新进展,是一本较为浅显和易懂的入门书籍。它适用于从事数字系统测试的工程技术人员,对于仪表和测量、计算机、自动控制与电路学科的学生,也有一定的参考借鉴作用。

由于编者水平有限,书中难免存在不足乃至错误,殷切希望广大读者和同行进行交流沟通与批评斧正。有关本书的意见请发送至本人邮箱:heavenhost@ qq. com,不胜感激!

作者

# 目 录

|                                |    |
|--------------------------------|----|
| <b>第 1 章 测试的基本概念</b> .....     | 1  |
| 1.1 数字电路测试 .....               | 1  |
| 1.1.1 测试 .....                 | 1  |
| 1.1.2 测试分类 .....               | 2  |
| 1.1.3 数字电路分类 .....             | 2  |
| 1.2 故障及故障模型 .....              | 3  |
| 1.3 算法 .....                   | 3  |
| 1.4 测试覆盖率和故障检出率 .....          | 4  |
| 1.5 测试矢量 .....                 | 4  |
| 1.5.1 组合电路的测试矢量生成 .....        | 6  |
| 1.5.2 时序电路的测试矢量生成 .....        | 10 |
| 1.6 可测性 .....                  | 15 |
| 1.6.1 可控性 .....                | 15 |
| 1.6.2 可观性 .....                | 16 |
| 1.6.3 可测性设计方法 .....            | 17 |
| <b>第 2 章 单板级 JTAG 测试</b> ..... | 31 |
| 2.1 背景介绍 .....                 | 31 |
| 2.2 传统单板测试方法的困难 .....          | 32 |
| 2.2.1 在线测试 .....               | 33 |
| 2.2.2 光学测试 .....               | 33 |
| 2.2.3 功能测试 .....               | 34 |
| 2.3 生产制造应用 .....               | 35 |
| 2.4 JTAG 测试技术 .....            | 36 |
| 2.5 单板级 JTAG 测试 .....          | 38 |

|              |                                |            |
|--------------|--------------------------------|------------|
| 2.6          | JTAG 测试的优缺点 .....              | 39         |
| <b>第 3 章</b> | <b>IEEE1149. X 标准 .....</b>    | <b>40</b>  |
| 3.1          | IEEE1149.1(边界扫描测试) .....       | 40         |
| 3.1.1        | 边界扫描测试概述 .....                 | 40         |
| 3.1.2        | 边界扫描硬件结构 .....                 | 42         |
| 3.1.3        | 边界扫描描述语言 .....                 | 69         |
| 3.2          | IEEE1532(在系统编程) .....          | 84         |
| 3.2.1        | 标准测试与编程语言 .....                | 85         |
| 3.2.2        | 在系统配置硬件结构 .....                | 87         |
| 3.2.3        | 数据和地址寄存器的访问 .....              | 89         |
| 3.2.4        | 在系统配置指令 .....                  | 89         |
| 3.2.5        | 在系统编程实现 .....                  | 91         |
| 3.3          | IEEE1149.6(高级数字网络边界扫描测试) ..... | 93         |
| 3.3.1        | 连接模型 .....                     | 93         |
| 3.3.2        | IEEE1149.6 架构 .....            | 95         |
| 3.3.3        | IEEE1149.6 指令 .....            | 96         |
| 3.3.4        | IEEE1149.6 应用实例 .....          | 99         |
| <b>第 4 章</b> | <b>单板级边界扫描可测性设计 .....</b>      | <b>100</b> |
| 4.1          | 测试点和测试探针 .....                 | 100        |
| 4.2          | 应力分析 .....                     | 102        |
| 4.3          | 菊花链 .....                      | 103        |
| 4.4          | 复位管脚设置 .....                   | 104        |
| 4.5          | 菊花链 TAP 端口设置 .....             | 104        |
| 4.6          | 不同电压芯片的连接 .....                | 106        |
| 4.7          | 优选遵循 IEEE1149.1 芯片 .....       | 108        |
| 4.8          | 配置管脚设置 .....                   | 108        |
| 4.9          | 扫描链中芯片个数 .....                 | 108        |
| 4.10         | 逻辑芯片的设置 .....                  | 110        |
| 4.11         | 存储器的连接 .....                   | 110        |
| 4.12         | 器件族的连接 .....                   | 111        |
| 4.13         | 其他情况 .....                     | 111        |



|            |                    |            |
|------------|--------------------|------------|
| 4.14       | 测试点的设计             | 111        |
| 4.15       | 小结                 | 113        |
| <b>第5章</b> | <b>边界扫描测试技术应用</b>  | <b>116</b> |
| 5.1        | 故障模型               | 116        |
| 5.2        | 器件模型               | 117        |
| 5.2.1      | 关键字符               | 118        |
| 5.2.2      | 器件模型实例             | 118        |
| 5.3        | 测试算法               | 130        |
| 5.3.1      | 计数/补偿算法            | 130        |
| 5.3.2      | 走步算法               | 131        |
| 5.3.3      | 边界扫描测试流程           | 132        |
| 5.4        | 芯片级测试              | 134        |
| 5.4.1      | JTAG 总线测试          | 134        |
| 5.4.2      | 芯片寄存器测试            | 135        |
| 5.5        | 单板级测试              | 139        |
| 5.5.1      | 基础测试               | 139        |
| 5.5.2      | 上/下拉测试             | 141        |
| 5.5.3      | 互连测试               | 142        |
| 5.5.4      | 器件族测试              | 143        |
| 5.5.5      | 存储器测试              | 144        |
| 5.6        | 系统级测试              | 145        |
| 5.6.1      | 系统级测试结构            | 146        |
| 5.6.2      | 多点网关芯片             | 147        |
| 5.6.3      | 基于 STA112 的系统级测试架构 | 149        |
| 5.7        | 应用方式               | 150        |
| 5.8        | 优缺点分析              | 150        |
| <b>第6章</b> | <b>串行矢量格式</b>      | <b>152</b> |
| 6.1        | 引言                 | 152        |
| 6.2        | SVF 命令集            | 152        |
| 6.2.1      | SVF 文件             | 152        |
| 6.2.2      | SVF 命令集            | 153        |

|              |                          |            |
|--------------|--------------------------|------------|
| 6.3          | SVF 命令详述 .....           | 154        |
| 6.3.1        | 寄存器测试结束 .....            | 155        |
| 6.3.2        | 测试频率设置 .....             | 155        |
| 6.3.3        | 寄存器头设置 .....             | 156        |
| 6.3.4        | 并行测试矢量设置 .....           | 158        |
| 6.3.5        | 并行管脚映射 .....             | 159        |
| 6.3.6        | 测试运行设置 .....             | 160        |
| 6.3.7        | 寄存器扫描 .....              | 163        |
| 6.3.8        | 状态路径设置 .....             | 165        |
| 6.3.9        | 寄存器尾设置 .....             | 166        |
| 6.3.10       | 复位设置 .....               | 168        |
| <b>第 7 章</b> | <b>内建自测试技术 .....</b>     | <b>170</b> |
| 7.1          | 芯片级内建自测试 .....           | 170        |
| 7.1.1        | 测试矢量生成和应用 .....          | 171        |
| 7.1.2        | 测试响应捕获和分析 .....          | 172        |
| 7.1.3        | 内建自测试控制器 .....           | 173        |
| 7.2          | 组合电路的内建自测试 .....         | 173        |
| 7.2.1        | 伪随机测试矢量生成 .....          | 174        |
| 7.2.2        | 特征分析的响应压缩 .....          | 177        |
| 7.3          | 时序电路的内建自测试 .....         | 180        |
| 7.4          | 宏单元的内建自测试 .....          | 180        |
| 7.4.1        | 低/中等复杂度的宏单元 .....        | 181        |
| 7.4.2        | 兆单元的内建自测试 .....          | 181        |
| 7.5          | 内建自测试与边界扫描测试 .....       | 181        |
| 7.5.1        | 具有内建自测试功能的边界扫描芯片结构 ..... | 182        |
| 7.5.2        | 内建自测试指令 (RUNBIST) .....  | 183        |
| 7.5.3        | 层次化内建自测试 .....           | 183        |
| 7.6          | 基于 FPGA 的单板级内建自测试 .....  | 184        |
| 7.6.1        | 基于 FPGA 的策略 .....        | 184        |
| 7.6.2        | 优缺点分析 .....              | 185        |

|                        |     |
|------------------------|-----|
| 第 8 章 片上仿真测试           | 187 |
| 8.1 片上仿真测试             | 187 |
| 8.2 实现步骤               | 188 |
| 8.3 ARM7TDMI JTAG 仿真测试 | 189 |
| 8.3.1 ARM7TDMI 测试结构    | 189 |
| 8.3.2 ARM7TDMI 处理器框图   | 190 |
| 8.3.3 ARM7TDMI 常用指令    | 191 |
| 8.3.4 ARM7TDMI 调试原理    | 191 |
| 8.4 测试实例               | 196 |
| 8.5 不足之处               | 197 |
| 第 9 章 嵌入测试             | 198 |
| 9.1 测试技术的发展            | 198 |
| 9.1.1 传统仪器             | 198 |
| 9.1.2 虚拟仪器             | 199 |
| 9.1.3 嵌入仪器             | 200 |
| 9.2 IEEE1687           | 201 |
| 9.2.1 背景概述             | 201 |
| 9.2.2 IEEE1687 架构      | 202 |
| 参考文献                   | 204 |

# 第 1 章 测试的基本概念

测试是人们认识客观世界、取得定性或定量信息的基本方法,是信息工程的源头及重要组成部分。随着科技发展的日新月异,测试工作将处于各种装备系统设计和制造的首位,并成为生产率、制造能力即实用性水平的重要标志。

测试的重要性贯穿产品的整个生命周期,即研发、生产、现场及返修等,为产品质量提升保驾护航,是客户满意度的有力保障。在产品生产制造过程中,必须进行有效的测试,以降低产品的不良率,确保产品质量的控制。因此,在产品的研发阶段,就应当对测试策略进行论证和确定,以简化后续的生产测试流程,提升产品的测试覆盖率,还会使产品具有更低的测试成本和更高的可靠性。

本章将介绍测试的基本概念,主要包括数字电路测试、故障及故障模型、可控性和可观性、测试矢量、可测性改进及单板级 JTAG 测试等,以便于对后续 IEEE1149.1 标准中测试理念和结构的理解。

## 1.1 数字电路测试

### 1.1.1 测试

测试,就是运用各种方法,对一个电路或系统进行检查,以确定是否偏离其预期的特性,涉及功能、时序、参数等。

数字电路或数字系统测试的目的,在于实现故障诊断(Fault Diagnosis),通常包括以下两个方面:

(1) 故障检测(Fault Detection)。确定被测器件或系统是“好”还是“坏”,即判断有无故障。

(2) 故障定位(Fault Location)。亦称为故障隔离(Fault Isola-

tion), 如有故障, 则应进一步查明故障的位置、产生原因及性质。

这里所谓的“好”或“坏”, 是指被测器件是否具备人们所期望的功能, 并不涉及该器件的设计是否正确、合理和完善等。通常, 对于所期望的功能, 可用真值表、布尔表达式、逻辑图或文字(如“与门”、“或门”、“全加器”等)来定义。

### 1. 1. 2 测试分类

在实践中, 对数字电路或数字系统的测试, 可分为参数测试和逻辑测试两大类。

#### 1. 参数测试

参数测试涉及两个类别: 静态测试和动态测试。静态测试, 是对被测器件的实际参数(电压、功率、负载能力等)的测量, 验证是否符合预期设计的指标; 动态测试, 是对被测器件在工作时的传播延时、脉冲宽度、前后沿、顶部和底部等参数进行验证。这类测试一般是在元器件制造完成后进行, 其方法属于传统的时域和频域测试。

#### 2. 逻辑测试

逻辑测试是在模拟类似于实际工作的环境下, 检查被测电路是否能实现预期逻辑功能的一大类测试, 贯穿整个产品的生命周期(研发、生产和维修等)。该测试亦称为数据域测试, 属于功能测试。

### 1. 1. 3 数字电路分类

根据逻辑功能的不同特点, 可以分成组合逻辑电路(简称组合电路)和时序逻辑电路(简称时序电路)两大类。

#### 1. 组合逻辑电路

在逻辑功能上的特点是任意时刻的输出仅仅取决于该时刻的输入, 与电路原来的状态无关。

#### 2. 时序逻辑电路

在逻辑功能上的特点是任意时刻的输出不仅取决于当时的输入信号, 而且还取决于电路原来的状态, 或者说, 还与以前的输入有关。与组合电路最本质的区别在于时序电路具有记忆功能。

## 1.2 故障及故障模型

故障,就是所测的信号值与期望值不一致,这是由于缺陷所引起的电路异常操作。如果被测器件因物质方面的事件,改变了本来属于自身的构造特性,这就称为产生或存在一个缺陷(Defect)。缺陷是物质上的不完善性,例如:引线短路或断路、晶体管击穿等。由于故障,会导致电路输出不正常,称为出错或错误(Error)。

所谓模型,是对一个实物主要特点的抽象表示。对于数字电路(逻辑建模)而言,一个模型对应于真值表、文本或图形化的描述,使我们可以预期其行为,即对施加到电路输入端的任何逻辑值,均可获得对应的响应值。故障模型,是对导致电路故障(电路响应值与期望值不一致)的逻辑条件的抽象表示。

为了便于问题简化和系统分析,必须对被测对象的故障模型做出一定的限制。根据数字系统的特点,大致分为如下三类故障:

(1) 永久性(非可变、非瞬变)故障。即若不予修复排除,则将一直固定存在的故障。

(2) 呆滞型故障。在逻辑上,表现为一个网络或一条连线,呆滞于0(stuck-at-0, s-a-0, s@0)或呆滞于1(stuck-at-1, s-a-1, s@1)。

(3) 多故障。同时出现呆滞型故障和永久性故障,或者超过任意两个故障的组合。

对于一个好的故障模型而言,应具备两个主要特征:简单性,以简化测试矢量生成的过程,实现效率提升;通用性,以确保绝大部分的故障能够被检测,不能被覆盖的故障数量要非常少。

## 1.3 算 法

算法,是指在有限步骤内求解某一问题所使用的一组定义明确的规则,亦或是一个问题无二义性解的过程。

一个算法的优劣可以用时间复杂度与空间复杂度来衡量:时间复

杂度,是指执行算法时所需要的计算工作量;空间复杂度,是指算法需要消耗的内存空间(硬件成本)。

## 1.4 测试覆盖率和故障检出率

在工业界,还有测试覆盖率和故障检出率两个常用的相关术语,分别解释如下:

### 1. 测试覆盖率

是指在单板或系统中,将能够测试的管脚数目、网络数目或器件数目,与单板或系统中的总量进行相除,通常用百分比表示。如:单板上共有 3000 个网络,能够测试到的网络数为 900 个,此时的测试覆盖率为 30%。

### 2. 故障检出率

是指对同型号的批量产品测试中,检测出的故障数,通常用百分比表示。如:测试同型号的批量单板 10000 块,检出故障单板有 3 块,此时故障检出率为 0.03%。

## 1.5 测试矢量

测试矢量,亦称为测试激励,是一组数据集合,从待测电路的主输入端输入后,激活被测故障,并将故障的效应传播到电路的主输出端。测试生成的基本目标是确定测试矢量,当将其施加到被测电路时,测试者能区分正常电路和故障电路。

现在来看一个最简单的故障测试的例子。图 1-1 所示是一个二输入端与门电路。下面分类进行阐述。

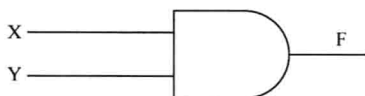


图 1-1 二输入端与门

(1) 正常真值表,如表 1-1(a)所列。

(2) 输入端 X(s@1)呆滞故障真值表,如表 1-1(b)所列。

如果该与门的输入激励为:  $X = 0, Y = 1$ ; 或者写成矢量形式:  $T1 = (0, 1)$ , 称为一个输入矢量, 亦称为测试矢量。那么该测试矢量就可检测出  $X(s@1)$  呆滞故障: 因为正常时, 输出  $F = 0$ ; 而当存在  $X(s@1)$  呆滞故障时, 输出  $F = 1$ 。

(3) 输出端  $F(s@1)$  呆滞故障真值表, 如表 1-1(c) 所列。

不难看出, 下列测试矢量均可检测出该故障:  $T1 = (0, 1), T2 = (0, 0), T3 = (1, 0)$ 。

表 1-1 与门真值表

(a) 正常真值表

| X | Y | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

(b) 输入端  $X(s@1)$  呆滞故障真值表

| X | Y | F |
|---|---|---|
| ① | 0 | 0 |
| ① | 1 | ① |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

(c) 输出端  $F(s@1)$  呆滞故障真值表

| X | Y | F |
|---|---|---|
| 0 | 0 | ① |
| 0 | 1 | ① |
| 1 | 0 | ① |
| 1 | 1 | 1 |

注: ①表示的呆滞故障值( $s@1$ )。

由上述分析可知: 一个测试矢量, 可检测出多种故障, 如  $T1 = (0, 1)$  可检测出  $X(s@1)$  或  $F(s@1)$  呆滞故障; 同一种故障, 也可由不同的输入测试矢量检测出, 如  $F(s@1)$  呆滞故障可以由  $T1 = (0, 1), T2 = (0, 0), T3 = (1, 0)$  检测出。

注意到, 如果只用  $T1 = (0, 1)$  这个矢量输入, 若得到输出  $F = 1$ , 表明被测与门电路有故障, 却无法定位到  $X(s@1)$  还是  $F(s@1)$  呆滞故障, 或者二者兼而有之。即只能进行故障检测, 无法做到故障定位。为了进行故障定位, 需要进一步的测试, 若增加一个测试矢量  $T2 = (0, 0)$ , 便可实现故障定位, 如表 1-2 所列。

表 1-2 与门故障定位

| 输入矢量 T | $T1 = (0, 1)$ |                                   | $T2 = (0, 0)$       |                     |
|--------|---------------|-----------------------------------|---------------------|---------------------|
|        | $Y1 = 0$      | $Y1 = 1$                          | $Y2 = 0$            | $Y2 = 1$            |
| 输出矢量 Y |               |                                   |                     |                     |
| 测试结果   | 正常            | 故障检测:<br>$X(s-a-1)$<br>$F(s-a-1)$ | 故障定位:<br>$X(s-a-1)$ | 故障定位:<br>$Y(s-a-1)$ |



将施加于被测器件的输入矢量序列  $T_1, T_2, \dots, T_n$  及其对应的输出矢量序列  $F_1, F_2, \dots, F_n$ , 称为一个测试 (Test), 记为

$$T = \{ T_1, T_2, \dots, T_n; F_1, F_2, \dots, F_n \}$$

其中:  $n$  为测试长度。例如表 1-2 所列是一个长度为 2 的测试:  $T = \{ (0, 1), (0, 0); (0), (0) \}$ 。

由此可见, 数字电路/数字系统测试, 一般包含以下三个方面的基本任务。

(1) 输入矢量序列的生成, 输入矢量序列亦称为测试样式 (Test Pattern) 或测试矢量 (Test Vector)。

(2) 输出矢量序列或输出样式的确定。

(3) 测试执行。将测试样式或测试矢量施加于输入端, 将实际输出结果与预期的输出样式比对, 进行故障检测和故障定位, 从而实现故障诊断。

### 1.5.1 组合电路的测试矢量生成<sup>[1]</sup>

组合电路的测试矢量生成算法较多, 如: 通路敏化法、D 算法、临界通路法、布尔差分法和故障字典法等。

#### 1. 通路敏化法

在组合逻辑电路中, 从一个主输入端 (或一条内部信号线) 通到一个主输出端的通路, 就称为一条传输通路 (Transmission Path), 简称为通路。其中: 主输入端 (Primary Input), 是指一条输入线, 不接受电路中其他任何一条线反馈的信号; 主输出端 (Primary Output), 是指一条通到电路外部去的信号线。

对一条通路中所有的门电路的所有输入端赋值, 以使通路上某一条信号线上的任何逻辑变化, 都能沿该通路传播到主输出端, 即: 主输出端的逻辑变化能够反映该信号线的逻辑变化, 这样的通路称为一条敏化通路 (Sensitized Path)。于是, 根据主输出端的逻辑变化, 就能侦查出敏化通路上的逻辑故障, 从而找出能侦查该故障的测试矢量。

以图 1-2 为例, 对电路敏化过程作具体的说明:

如果首先赋予输入端  $B = 1$ , 则  $Y = 0$ 。那么可以看到, 通路  $A - X - F$