**存储器自适应测试技术研究与实现**

目录

[故障类型 2](#_Toc155710383)

[故障注入 8](#_Toc155710384)

[自适应算法部分 9](#_Toc155710385)

[存储器测试算法 10](#_Toc155710386)

[目录 18](#_Toc155710387)

# 故障类型



引存储器测试算法研究及应用实现\_张显敞

**单个单元静态故障**

SF 状态故障 <0/1/->，<1/0/-> ，

SAF 固定性故障 <任意/1/->，<任意/0/-> ，

TF 转换故障 <0w1/0/->，<1w0/1/-> ，

WDF 写干扰故障 <0w0/↑/->，<1w1/↓/-> ，

RDF 读破坏故障 <r0/↑/1>，<r1/↓/0> ，

DRDF 伪读破坏故障 <r0/↑/0>，<r1/↓/1> ，

IRF 错误读故障 <r0/0/1>，<r1/1/0> ，同RDF



**对两个单元静态耦合故障**（a，v），四种故障分两种情况，8种类型

CFst 状态耦合故障 <0;0/1/->，<0;1/0/->，<1;0/1/->，<1;1/0/->

，，，

PS：<0;1/0/->中（1第二项为a<v情况，2第一项为a<v情况），<1;0/1/->中只给出a>v情况

PS：后面章节对耦合故障的分析，我们将只分析施主单元地址(a)大于故障单元地址(v)的情况 即a>v的情况，当a<v的情况下，只需将a>v情况下的地址的变化顺序取反即可。

CFds 干扰耦合故障

CFds rx ：<r0;0/↑/->、<r0;1/↓/->、<r1;0/↑/->、<r1;1/↓/->

，或者，，

CFds xwx ：<0w0;0/↑/->、<0w0;1/↓/->、<1w1;0/↑/->、<1w1;1/↓/->

，或者，，

CFds xw!x : <0w1;0/↑/->、<0w1;1/↓/->、<1w0;0/↑/->、<1w0;1/↓/->

、、或者 、

CFtr 转换耦合故障 <0;0w1/0/->、<0;1w0/1/->、<1;0w1/0/->、<1;1w0/1/->

、、、

CFwd 写破坏耦合故障 <0;0w0/↑/->、<1;0w0/↑/->、<0;1w1/↓/->、<1;1w1/↓/->

、、、

CFrd 读破坏耦合故障 <0;r0/↑/1>、<1;r0/↑/1>、<0;r1/↓/0>、<1;r1/↓/0>

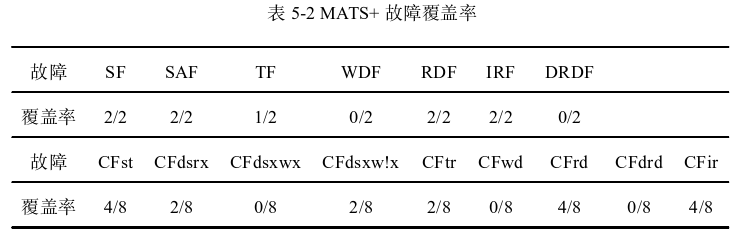
 、、或者 、

CFdrd 伪读破坏耦合故障 <0;r0/↑/0>、<1;r0/↑/0>、<0;r1/↓/1>、<1;r1/↓/1>

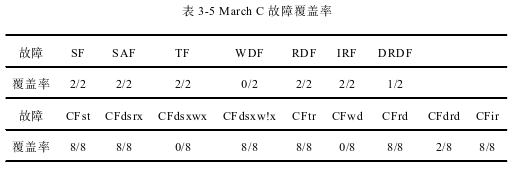
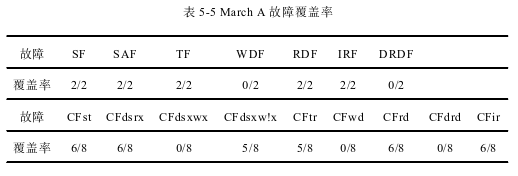
或者 、、、

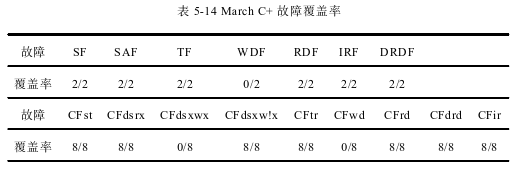
CFir 错误读耦合故障 <0;r0/0/1>、<1;r0/0/1>、<0;r1/1/0>、<1;r1/1/0>

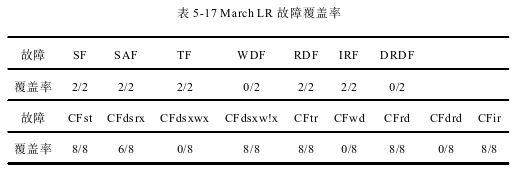
同CFrd： 、、或者 、

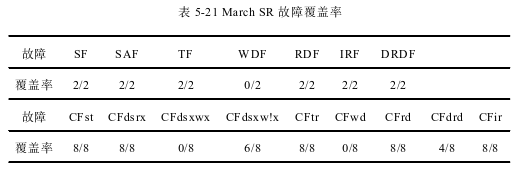


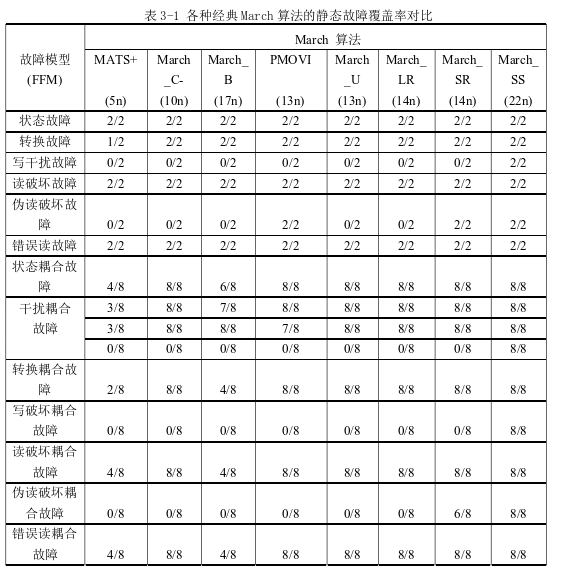
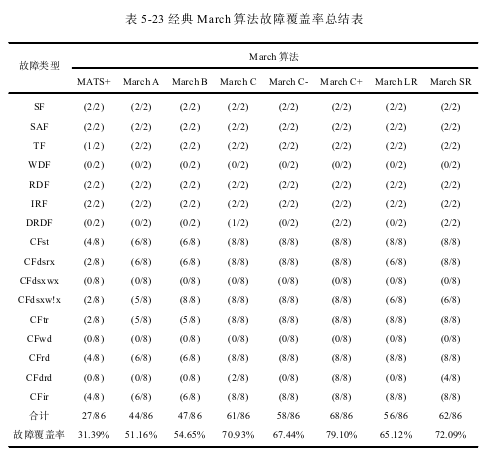
MATS+：









路张

**对单个单元动态故障的检测**

dWDF 动态写干扰故障

dRDF 动态读破坏故障

dDRDF 动态伪读破坏故障

dIRF 动态错误读故障

**对两个单元动态耦合故障的检测**

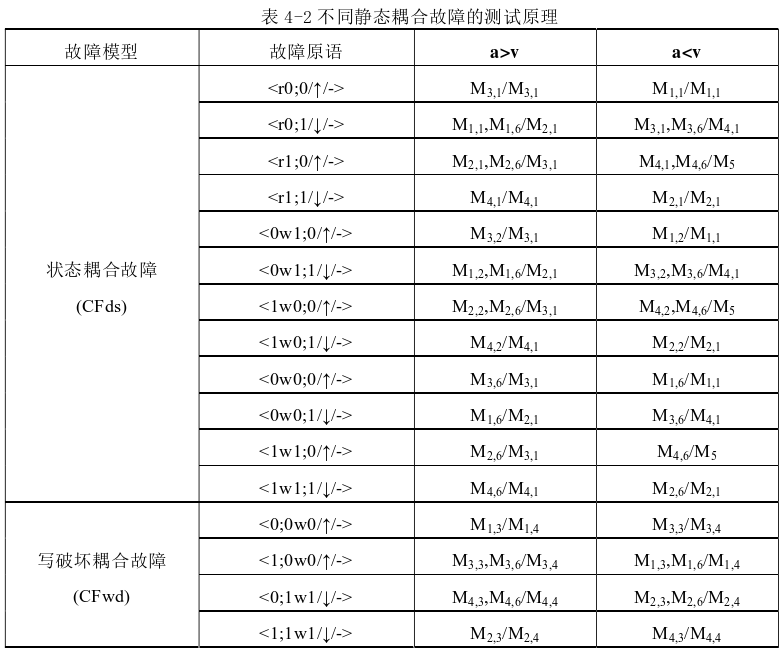
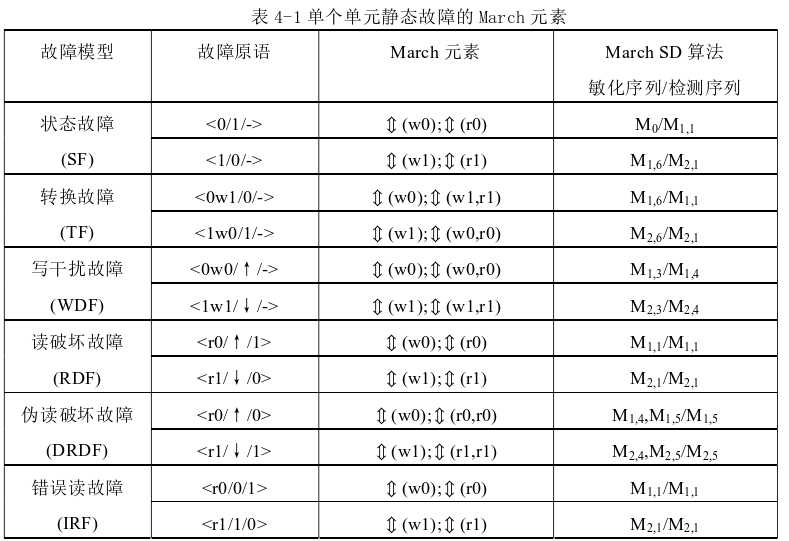
dCFwd 动态写破坏耦合故障

dCFds 动态干扰耦合故障

dCFrd 动态读破坏耦合故障

dCFdrd 动态伪读破坏耦合故障

dCFir 动态错误读耦合故障







# 故障注入

注入故障方法

1在写阶段，直接写入错误数据

2在读阶段，加到某一地址后，插入一段写入错误数据操作，然后再读取。

3在触发敏化操作后，写入错误数据

# 自适应算法部分

做可视化界面，错误标红注明故障类型，可以展开一页里的8乘32地址

1基础python

2高级算法处理：制造重叠复杂数据集，重点在于平衡覆盖率和测试逃逸率

3 LLM

4目前，最新的分类算法包括：

Transformer-based models：这种模型基于自注意力机制，能够捕捉句子的上下文信息，进而提高分类的准确率。例如BERT、GPT等。

Graph Neural Networks（GNNs）：这种模型适用于图像分类和推荐系统等领域，能够识别图像中不同部分之间的关系，进而提高分类的准确率。例如Graph Convolutional Networks（GCN）、Graph Attention Networks（GAT）等。

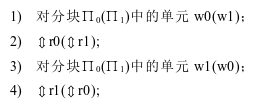
Deep Residual Networks（ResNets）：这种模型通过引入残差连接来解决深度神经网络训练时的梯度消失问题，从而能够训练更深的网络，提高分类的准确率。例如ResNet、DenseNet等。

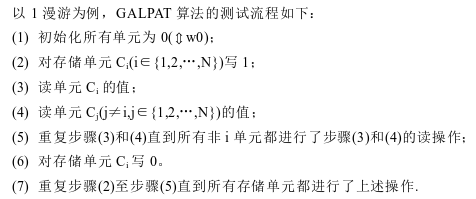
Capsule Networks（CapsNets）：这种模型基于胶囊网络理论，能够识别物体的姿态和相对位置，从而提高分类的准确率。例如Dynamic Routing Between Capsules（DRCaps）等。

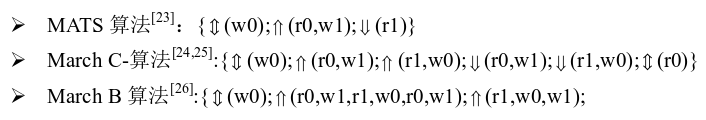
总的来说，这些最新的分类算法都具有很高的准确率和鲁棒性，已经在许多领域得到了广泛应用。

# 存储器测试算法

MSCAN 

Checkerboard 

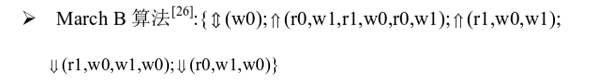
GALPAT 



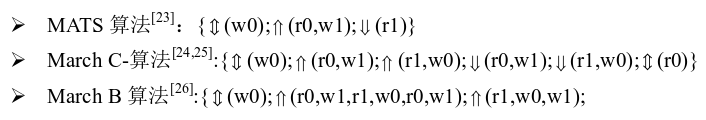
MATS+ 

MATS++ （March）

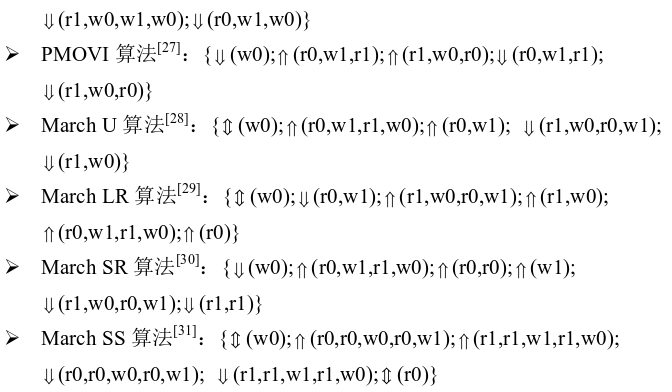
March A 

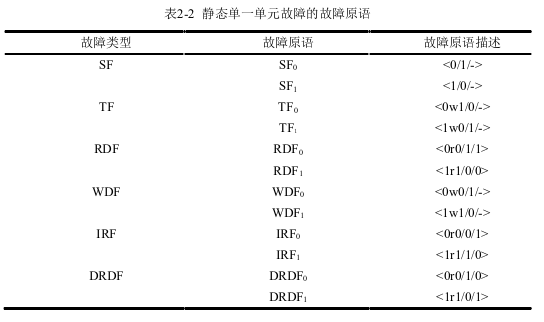


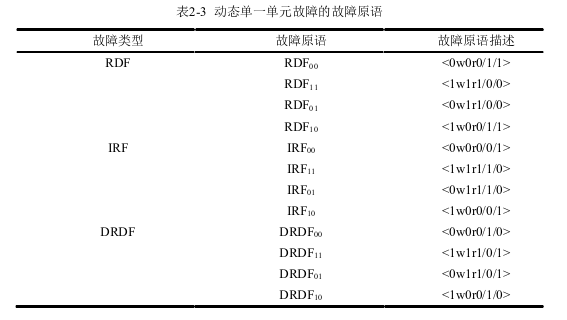
March C 



March C+ 







存储器故障模型

存储器的作用就是用于存储信息，可以实现数据存入和读取的功能。一个良好的存储器的每个存储单元，都可以存储数据1和数据0，并且可以实现数据从0到1或从1到0的变换。所以通常来说存储器的故障有：固定性故障、变迁故障、耦合故障和图形敏感故障等。下面简述以上四种故障类型。

（1）固定性故障

固定性故障指存储单元的存储数据固定为数据0或者数据1，不能对其进行写入数据，即数据0写入0或1仍是0，数据1写入0或1仍是1，如图1。对于测试固定性故障，只需要对存储器全部写1或0，然后读取数据进行对比，判断是否存在故障。

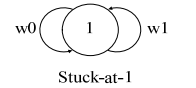
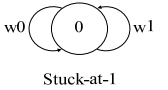


图1 固定性故障

（2）变迁故障

变迁故障指存储单元不能实现从数据0变换到数据1或者不能从数据1变换到数据0，如图2所示，变迁故障属于固定性故障的一个特例。对于测试变迁故障，需要对存储器实现一次数据从0到1的变换或者从1到0的变换，然后读取数据进行比对，可以判断出是否存在故障。

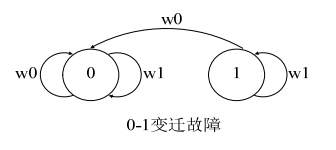


图2 变迁故障0-1

（3）耦合故障

耦合故障指某个存储单元的数据发生变化引起了另一个存储单元的数据变化。常见的耦合故障有反相和定值。反相耦合故障是当某一存储单元数据发生变化时，另一个存储单元的数据变为相反的值例如从1变0或从0变1。定值耦合故障是当某一存储单元数据发生变化时，另一个存储单元的数据变成一个定值0或1。如图3为反相和定值故障的表示。

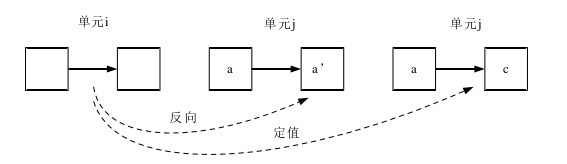


图3 耦合故障

(4)图形敏感故障

图形敏感故障指当一个存储单元受到其他单元的读写操作影响状态发生变化，其原因是存储器的存储密度过高，单元与单元之间相互之间干扰。如果一个存储单元作为基单元，因旁边的相邻单元的活动而导致其状态不确定，这种故障称之为邻居图形敏感故障，通常我们研究五相邻单元和九相邻单元，这种故障所用的测试图形和向量会比较复杂。

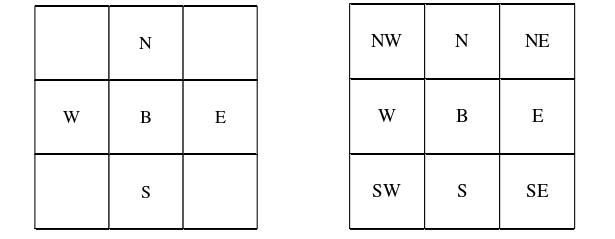


图4 五相邻单元与九相邻单元

存储器测试算法

1、MSCAN算法

MSCAN算也称为全0全1算法，顾名思义该算法实际上是形式固定的存储 器扫描序列，对所有单元写(读)0或者写(读)1：

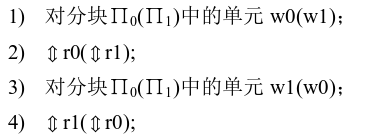


可以得出MSCAN算法的总的测试次数为4N，测试的复杂程度是跟存储单元 的容量N成正比，可见MSCAN的算法的复杂度较低，但是该算法除了能测试 SAF故障外，对其他类型的故障无能为力。

2、Checkerboard算法

Checkerboard算法也就是棋盘算法，因为其测试图形非常像国际象棋的棋盘 而得名。其测试流程大体如下：首先初始化使得每个单元与其相邻的四个单元的 值都不相同；然后整个存储阵列就被分成了两大块，所有单元状态为0的组成一 块，用符号∏0表示，所有状态为1的单元组成一块，类似的用符号∏ 1表示；接着分别对两块进行读写；最后换逻辑值相反的值重复上述的过程。

Checkerboard测试算法的测试流程如下：

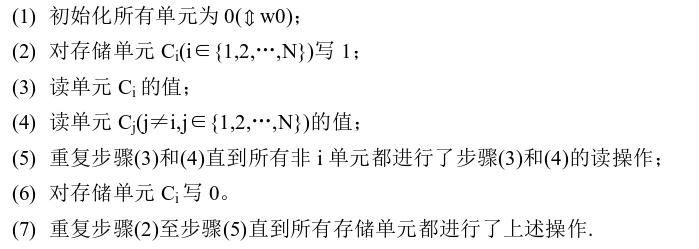


从测试流程可以得出Checkerboard算法的测试长度为4N，该算法可以检测和 定位SAF算法，相邻单元之间的短接故障。

3、GALPAT算法

该算法也被称为1(0)漫游算法或者乒乓测试算法。看过该算法的测试流程， 便可以得知该算法名称的由来。

以1漫游为例，GALPAT算法的测试流程如下：



4、March算法

这是当前最流行的研究最热的测试算法之一，在实际中也有较多的应用。算 法的基本原理就是建立一个有限的状态机，对所有单元逐一的进行读写。有资料 证明通过对每个存储单元进行不断的读写的方法，几乎可以检测出所有类型的故 障模型。当下March算法的衍生算法种类繁多，它们之间的不同之处或者本质上 的区别是采用不同的March元素。所谓March元素就是一个非常简单的MSCAN 操作，或者也可以由一系列的读写操作序列组成。

关于March算法，在下一章将会系统全面的介绍，这里仅介绍一种基本的 March算法，简称为MATS算法。该算法的测试流程如下：

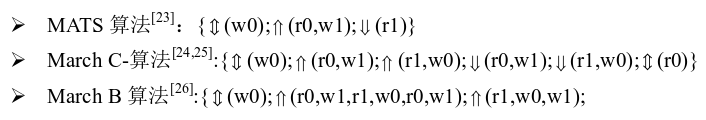
对所有单元初始化为0；然后按照地址升序的顺序依次对每个单元进行读0、 写1操作；最后再按照地址降序的顺序依次对每个存储单元进行读1、写0、读0 操作。

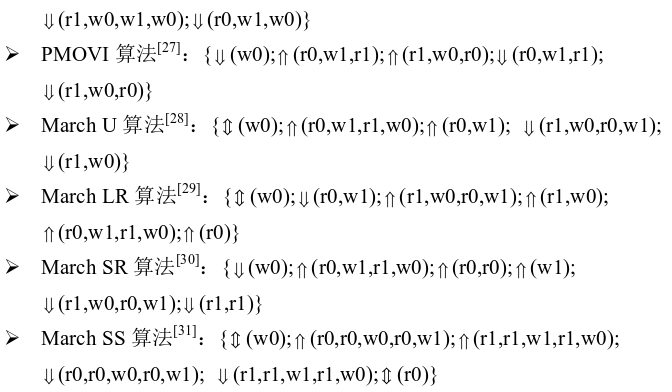
该算法用符号表示如下：

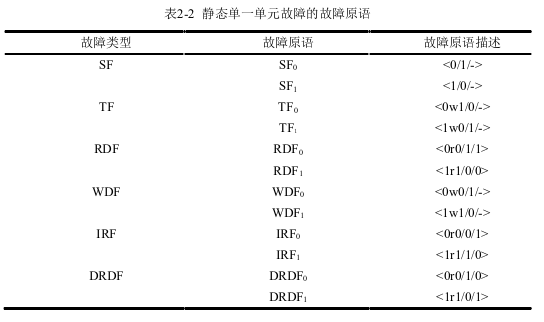


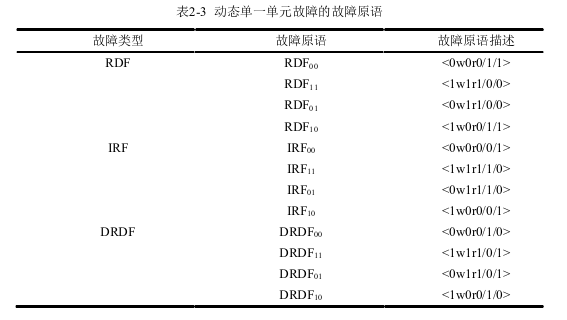
由上式可以得出该算法的总的测试次数为6N。该算法可以检测出固定性故障、变迁故障。

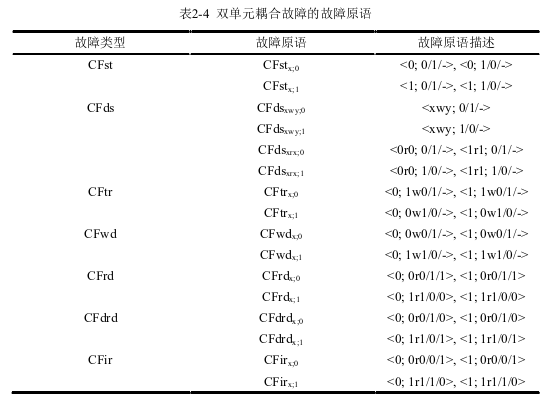
一些常见算法：











# 目录

存储器自适应测试技术研究与实现

第一章绪论

1.1研究工作的背景与意义

1.2国内外研究现状

1.3课题的研究内容及创新点

1.4本文的结构安排

第二章存储器测试技术基本原理

2.1存储器的分类

2.2存储器测试原理

2.3常见存储器测试方法

2.4存储器故障原语

2.5存储器故障类型

2.5.1单个单元静态故障

2.5.2两个单元静态耦合故障

2.5.3单个单元动态故障

2.5.4两个单元动态耦合故障

第三章存储器测试图形

3.1MSCAN算法

3.2Checkerboard算法

3.3Diagonal算法

3.4GALPAT算法

3.5March类算法

3.5.1MATS

3.5.2MATS+

3.5.3MATS++

3.5.4March A

3.5.5March B

3.5.6March C

3.5.7March U

3.5.8March LR

3.5.9March SR

3.5.10 March SS

3.6自适应图形算法

第四章存储器测试数据获取与分析

4.1硬件测试平台设计

4,1,1眼图测试，评估平台信号质量

4.2故障注入技术研究

4.3存储器测试图形实现

4.4测试与结果分析

第五章存储器自适应测试的实现

5.1自适应测试技术

5.2测试数据预处理

5.3存储器测试图形优化

5.4存储器自适应测试实现

第六章总结与展望

参考文献