

1 K-5

大規模メモリテスト内での故障情報の格納について\*

内田 剛 井口 幸洋  
 明治大学 理工学部 情報科学科

1. はじめに

半導体メモリの大規模化に伴い、テストバーイン装置(以下ではこれを“テスト”と呼ぶ)も大規模メモリを取り扱うようになってきた。テストの際、各メモリチップの良・不良の判定だけでなく、後で行う故障診断の為に全ての故障情報を記憶しておく必要があるが、この故障情報をそのまま記憶すると大量の記憶装置がテスト内に必要となり現実的でない。そこで本稿では、この故障情報の圧縮法の提案をする。

2. 故障情報

テストから出力される故障情報は、図1のように2次元のマップ(フェイルマップ)で表現できる。図1で黒い四角は1つのセルが不良であることを示す。

1つのセルに不良が観測された場合には①、隣接するNセルに不良が観測された場合には②・③の様なパターンになり、アドレス線に障害が起こった場合には④・⑤・⑥・⑦の様なパターンになる。また、デバイス欠陥、等によってセルアレイのある一画に不良セルが観測された場合には⑧の様になる。今述べた8種類のパターンで全てのフェイルマップを表現することができる。

ここでは、セルの走査の順は左上から順次行われることとする。

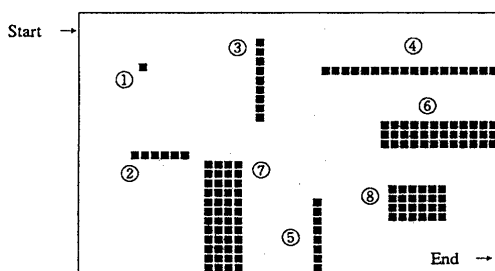


図1 フェイルマップと故障の種類

3. フェイルマップ圧縮アルゴリズム

3-1. 圧縮の条件

記憶装置の削減の為に、出力される故障情報を読み込んでリアルタイムに圧縮するアルゴリズムが必要となる。また、圧縮された故障情報は無雑音圧縮でなければならない。故障診断に用いるので故障の種類などを圧縮データからでも判定できるようにしたい。以上のことを満たすためには、従来の圧縮法では不向きであるので、以下に2つの圧縮法を提案する。

3-2. 圧縮のアルゴリズム1

圧縮データから故障の性質を判定するために故障データを

$$\text{故障データ} = \text{故障の種類} + \text{アドレス} + \text{付加情報}$$

の形で表す。故障によっては、故障範囲に対する情報が必要なので付加情報を追加した。表1に故障の種類と付加情報を示す。

表1 付加情報と種類コード

	付加情報	種類コード
①	なし	0 0
②	故障の長さ (横方向)	1 0 0
③	故障の長さ (縦方向)	0 1 0
④	なし	1 1 0
⑤	なし	0 1 1
⑥	故障の幅 (縦方向)	1 1 1
⑦	故障の幅 (横方向)	1 0 1 1
⑧	故障の幅 (横・縦方向)	1 0 1 0

これら8種類の故障の種類は図2に示す手順で簡単に判定できる。この図で || で囲まれた数字は故障種類を表し条件1~4は以下にします。

条件1: 故障セルが横に続く。

条件2: 故障セルが右端まで続く。

条件3: 故障セルが縦に続く。

条件4: 故障セルが下端まで続く。

以上の手順で1つの故障が起こっている場合には故障の特定ができる。しかし、故障の重なりが起こっている場合には検査中の故障同士の比較が必要となるので

\* Compression of Fail-Map in Memory Brun-in Tester

以下の2つの条件を加え、故障の重なりを特定する。  
 条件0：新たな故障セルが現れたら検査中の故障と比較を行う。  
 条件3'：故障セルが横にも続く場合はその長さを調べ条件2で得た長さと比較を行う。

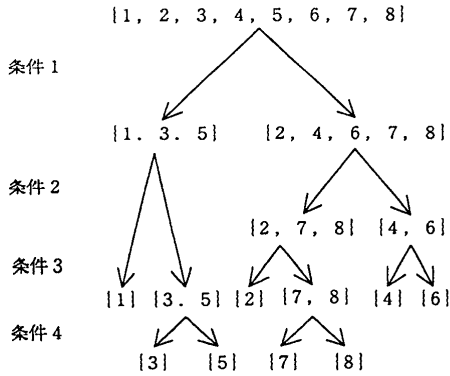


図2 故障種類の判定木

この手順は条件4まで検査しなくても特定できる故障があるので2つ以上の故障が重なっているフェイルマップに対して有効である場合がある。これは条件0の比較が検査中の故障に対して行われている為、既に特定されている故障との比較が無いからである。また、図2より故障の種類を表1のようにコード化でき、このコードはハフマン符号のように一意に決定できるコード化になっている。

3-3. 圧縮のアルゴリズム2

縦方向、横方向でセルに不良が観測された場合には、セルのアドレスを使い表現するとうまくいく場合がある。ここでは、二分木を用いてフェイルマップを表す方法を簡単に述べる。図3のフェイルマップを二分木で表現すると図4のようになる。<sup>[2]</sup>例えば図4でA3の左側につながるPは、A3が0のセルは全て正常に機能していることを表している。また、A2の下の方のFはA3A2が11のセルが不良であることを示している。

ここで、この木を作るには、アドレス順にセルが不良か正常かの情報が入ってくるので、その順にルートノードから枝を作成していけばよい。あるノードにつながる左の枝と右の枝の両方が揃ったら、図5に示すようにその下につながる2つの木が同形かを調べて、同形ならばそのノードを削除して単純化しながら作成していく。この方法は、ブロック故障などを表すのに有効であると考えられる。

4. まとめ

故障情報を圧縮するアルゴリズムを2種類提案した。最初の圧縮法は、圧縮されたデータで故障診断が容易に行えるように作られている。故障の数に応じて圧縮データが決定するので故障の少ない場合には特に有効であると思われる。2番目の圧縮法は、まとまった故障に対して有効であると思われる。今後の課題は、この2つの圧縮法を実際のフェイルマップに適用し評価検討することである。

謝辞 有益な御意見を頂きました日本エンジニアリング(株)の横山治男氏に感謝致します。

参考文献

- [1] 山田輝彦：“疑似ランダムパターンを用いた埋込み型RAMの組込み自己テスト”，信学論(D)，J72-D-1(1989, 1)
- [2] 石畑清：“アルゴリズムとデータ構造”，岩波出版，1989

	A0	0	1	0	1
A1	0	0	1	1	
A3A2	0	0	P	P	P
	0	1	P	P	P
	1	0	F	P	F
	1	1	F	F	F

P:正常 F:不良

図3 フェイルマップ

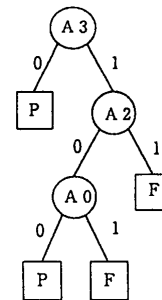


図4 二分木の例

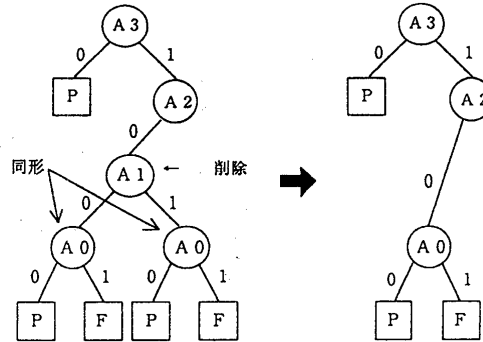


図5 ノードの削除