

知識ベースシステムに基づいた LSI テスト不良原因解析について

武田 敏秀[†] 井上 俊介[†] 大野 国弘^{††} 温 暁青[‡]

半導体集積回路は、大規模化・微細化・低電圧化に伴い、テスト開発も複雑化して、その結果、テスト歩留りが低下している。テスト歩留り低下の原因は、設計や製造起因だけではなく、テスト環境での起因などと複合化している。テスト歩留り向上のためには、その不良原因の解析が必要不可欠であるが、この部分はテスト技術者の経験や知識とノウハウに依存して、効率的な解析が実施されていない。そこで本論文では、熟練テスト技術者が持つ豊富な知識やノウハウを知識ベース化した上、自動推論エンジンによってテスト不良原因の切り分けと歩留り向上のための助言を行う手法を提案する。

Knowledge-Based LSI Test Failure Analysis

Toshihide TAKEDA[†] Shunsuke INOUE[†]

Kunihiro OHNO^{††} Xiaoqing WEN[‡]

The problem of LSI test yield loss is getting worse due to ever-shrinking feature sizes and ever-growing integration densities. The cause for test yield loss is complex, and its analysis is time-consuming. The tool under development is intended to swiftly identify design, fabrication, and test related causes for test yield loss, by using a unique knowledge base technology. This paper shows the tool helps improve test yield by solving the shortage of test and analysis engineers and making it possible to analyze more yield-loss cases.

1. はじめに

半導体集積回路 (LSI) の製造における歩留りを低下させる欠陥には、大きく分類すると、主にウェハプロセス中に発生するランダム欠陥 (不良)、主にウェハプロセスの揺らぎによって発生するパラメトリック特性起因の不良や、特定の回路パターンで不良が発生するシステムティック欠陥 (不良) があげられる。近年の半導体微細化プロセスの一層の発展に伴って、これらによる製造不良の問題、特にパラメトリック不良やシステムティック不良がより深刻になってきている。LSI 設計においても、大規模化・微細化・低電圧化・高速化が進み、設計ミス、設計検証不足、設計マージン不足による設計起因での不良の問題が深刻化している。また、LSI を電氣的に正しく動作することを確認するためのテストでも、その測定環境であるシステムボードやプローブカード・テストソケットを含む DUT (Device Under Test) ボードについても、高速動作におけるシグナルインテグリティの問題や電源供給を十分考慮し作製することが必須となっている。

従って、LSI の開発・量産においては、実チップで動作不良や歩留りの低下が生じた場合、その原因が製造不良なのか、設計に起因する不良なのか、あるいは動作測定環境の不備なのかの切り分けが著しく困難になっている。このため、実チップのデバッグや不良原因特定には、多大な工数・期間を必要とし、事業の阻害要因のひとつになっており、その深刻さは益々増大する傾向にある。

また、不良原因特定は技術者の経験・知識とノウハウに依存して効率的な解析が実施されていない。経験の浅い技術者は適切な解析が出来ずに時間を浪費する場合あることや、知識やノウハウを技術者個人が所有して共有化が難しいという課題がある。そこで本研究では、技術者が実施しているテスト不良解析を技術者の思考と知識・ノウハウを体系化して、歩留り向上のための不良原因の効率的な推定と対策提示手法の確立、および知識ベースシステムによる技術者の知識・ノウハウの保守、継承を可能とする。

本論文では、テスト歩留り低下ロットの解析事例を知識ルールにて記述して、知識ベースシステムによって不良原因を推定できる手法を提案する。

2. テスト不良解析

2.1 概要

テスト工程での歩留り管理は、製造ラインで日常的に行われて、管理水準を超えたロットが発生した場合に、エンジニアがその原因を特定して異常か正常かを判断して、

[†] (財) 福岡県産業・科学技術振興財団 〒810-0001 福岡県福岡市中央区天神 1 丁目 1 番 1 号

^{††} (株) なうデータ研究所 〒820-0067 福岡県飯塚市川津 680 番地の 41

[‡] 九州工業大学情報工学研究院 〒820-8502 福岡県飯塚市川津 680-4

そのロットの処理を決定する。ここで異常が発見されれば、原因究明のために不良解析が行われる。まず、エンジニアは、当該製品の歩留りトレンドチャート、テストプログラム、ウェハーマップやテストデータログから不良内容を確認して、その不良原因解明のために不良現象の詳細解析や統計解析ツール YMS (Yield Management System) 等を用いて試行錯誤の上、原因を特定する。歩留り低下の原因が特定されたら、対策の必要な工程へのフィードバックを行って歩留り向上につなげる[1]。

テスト不良品は最終的にテストプログラムの判定基準で不合格となったデバイスであり、一般的にはテストプログラム内のテスト項目であるファンクションテスト、DCテスト（電圧、電流測定）、ACテスト（時間、周波数測定）等に層別することができる。また、テストでの不良原因を特定するために必要なエンジニアの知識とは、電子回路、統計解析、統計的管理技術、ウェハー製造工程、テスト仕様書、テストプログラム、LSI テスター、組立工程、パッケージ構造等の非常に幅広い知識が必要となる。

2.2 歩留り低下原因解析事例

テストでの歩留り低下の場合の不良解析事例を示す。ここでは、アナログ系のある製品のウェハーテストでの歩留り低下原因の解析事例で説明する。

テスト結果から対策提示までのフローは、一般的に以下の4ステップとなる。

A. テスト結果の状況把握、不良現象特定

図1にテスト歩留り不良率を示す。BIN データから Idd（電源電流）項目での歩留り低下であることがわかる。不良データの確認のために再現性を確認する。同時に LSI テスターの機能を使って全数（または時間の制約がある場合はサンプリング）のデータログを採取する。

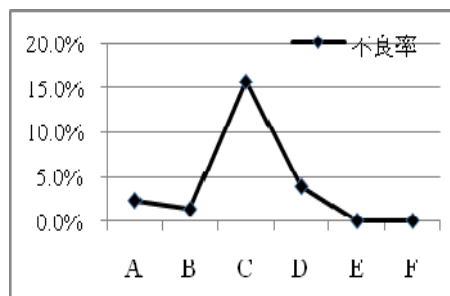


図 1 BIN (Idd) 不良率

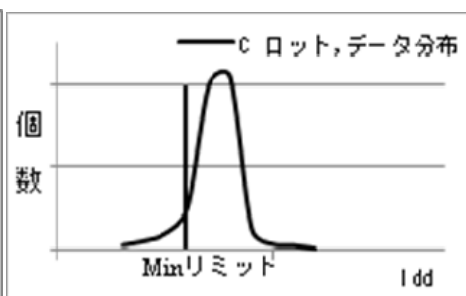


図 2 データ分布

データログの結果からデータの分布を見て歩留り低下項目の不良現象を明確にする。その結果を図2に示す。これから、ばらつきは正常のようであるが、平均値が Min(Low) リミット側へシフトしていることがわかる。歩留りの低下ロットと比較する為に過去のロットの測定データ分布を同様に確認する。

図3は、過去のロットとの測定データの分布の比較を示す。この結果、今回の歩留り低下ロット C は他のロットに比べて、Idd データの分布（平均値）が Min(Low) リミット側にシフト（測定値が減少）して、一部分の製品が不良判定されたためであることがわかる。

B. 不良現象から原因候補を推定

ここで Idd データが Min(Low) リミットに対してマージナルとなった原因については、様々な要因が推定されるが、経験上、電流値が減少していることから、まずはウェハープロセスでの抵抗成分のばらつきを要因として解析を進める。これを確認するためには TEG (Test Element Group) のパラメータであるシート抵抗との相関を見る。

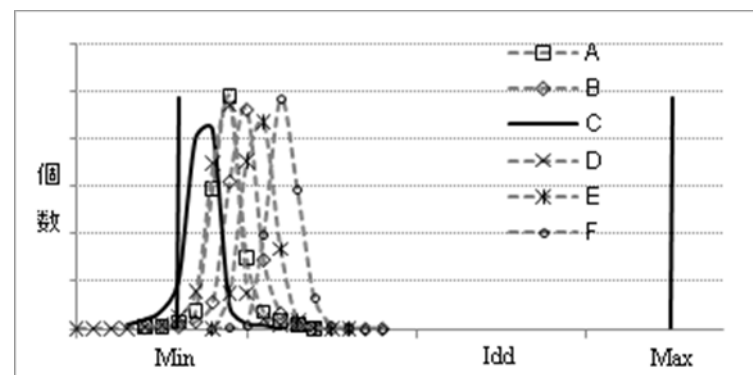


図 3 データ分布 (ロット A~F)

C. 条件検証と推測原因の導出

図4にシート抵抗と Idd との相関図を示す。これから、シート抵抗と Idd との間には、強い相関関係があることがわかる。この相関関係から、今回の歩留り低下ロットのデータの分布は、十分にシート抵抗のばらつき範囲内であり、製品自体の故障ではなく、Min(Low) リミットの設定が適切ではないという結論となった。

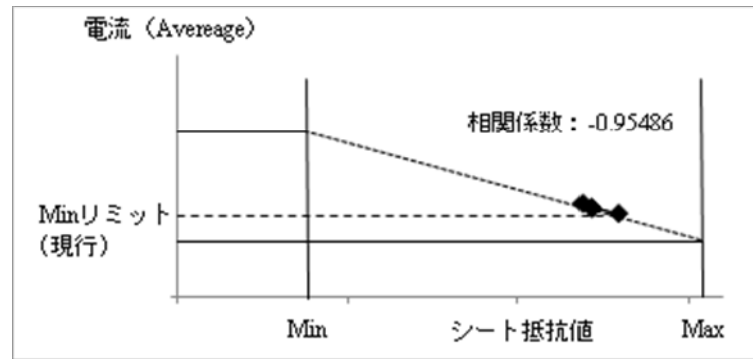


図 4 シート抵抗, Idd 相関図

D. 推測原因に対する改善対策の提示

対策としては、テストリミットをプロセスでのばらつきを考慮した値に変更すること、およびテストリミット決定時の方法を見直すことになる。

3. 知識表現

3.1 テスト不良解析データモデル

知識には大きく分けて二つの種類があり、それは、事実 (Fact) と判断手続き (Judgemental Knowledge) とから構成されている。

事実は宣言型知識と呼び、テスト不良解析での業務知識ではテスト結果のデータログ等にあたる。また、後者は手続き型知識と呼ばれ、特定の問題解決に利用される専門家のみが持っているもので、テスト不良解析での技術者の専門知識である。

テストデータの概要を図 5 に示す。これに示すように製品 (Product ID, Name) 毎にテストプログラムが定義され、それにはテスト項目 (Test ID, Pin Group, 等) や測定条件 (DC/AC condition, V_{ij}/V_{il}, DPS, 等), リミット (Hi/Lo Limit, 等) が記述されている。テストプログラムに準じたテスト結果が出力されて、テスト歩留りのデータ (Result+Log, 等) が含まれている。

上記のようにデータモデルの構造から、知識の表現形式としては、不良解析の知識は属性の従属関係によって表現することが可能である [2]。

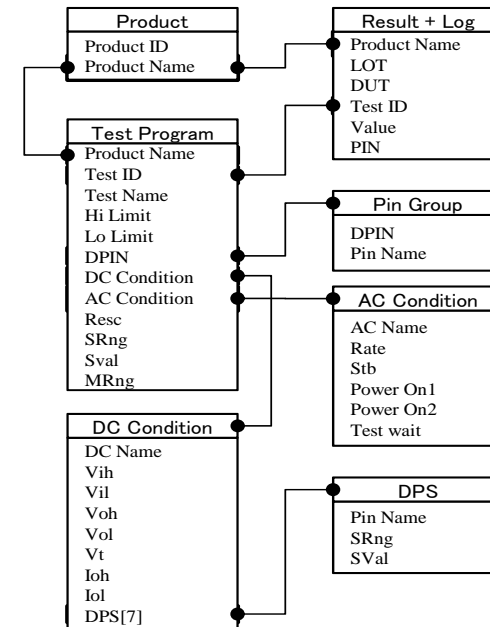


図 5 テストデータ概要 (一部)

3.2 不良解析技術の記述例

ここでは、梅田らによって提案された「設計計算のプログラム書法」[3]のプログラム例に基づいて記述例を示す。前章 2.2 の事例での A. テスト結果の状況把握、不良現象特定において、不良現象を把握する場合の例として、下記にテストプログラムのリミットとデータ分布 (平均と分散) の関係によって不良現象を判断する例を表す。

ここで、①②はテストプログラムから Hi リミット Low リミットを抽出して、テストデータから③平均値と④3σを計算して、⑤⑥⑦等の条件を満たした結果が判断として回答され、不良現象として把握される。

現象把握の記述例

```
evaluate_failer_mode(
    {TestDataObject, TestProgramObject},
    {Result},
```

```

method(
  HiLimit    := get_highLimit(TestProgramObject)  --- ①
  LoLimit    := get_lowLimit(TestProgramObject)  --- ②
  AveValue   := get_average(TestDataObject)      --- ③
  ThreeSigma := get_threeSigma(TestDataObject)   --- ④

  case(HiLimit =< AveValue + ThreeSigma          --- ⑤
    and AveValue + ThreeSigma =< HiLimit * 1.1
    -> Result := `Hi Marginal Failure` ;

    LoLimit * 0.9 =< AveValue - ThreeSigma       --- ⑥
    and AveValue - ThreeSigma =< LoLimit
    -> Result := `Lo Marginal Failure` ;

    AveValue + ThreeSigma =< LoLimit             --- ⑦
    -> Result := `Lo Failure`

  )
)

```

次に、2.2 章の、B.不良現象から原因を推定する例を以下に示す。
 ⑧⑨は、前記の⑤⑥⑦から判断された現象によって、推定原因の解を得られる。
 ここでは、Low マージナル不良の場合は、プロセスばらつきやテストリミット設定の要因が得られ、Malfunction (動作不良) の場合はコンタクト部分のプローブカード、またはトランジスタの特性等が要因として得られることになる。⑩⑪では、不良現象からこれらの推定原因の判断をしている。

不良原因候補推定

```

estimate_cause(
  { FailerMode },
  { Causes },

  method(
    case(FailerMode = `Lo Marginal Failure`  --- ⑧
      -> Causes := [ `Process Variation`, `Test Limits`, ... ] ;
  )
)

```

```

  FailerMode = `Malfunction`          --- ⑨
  -> Causes := [ `Contact ( Probe card , IC Socket )`, `Tr Parameter`, ... ]
)
)

judgement(
  {TestDataObject, TestProgramObject},
  {Result},

  method(

    call(evaluate_failer_mode, {TestDataObject, TestProgramObject}, {FailerMode}),
    call(estimate_cause, {FailerMode}, {Result})
  )
)
)

```

最後に、C. 条件検証と推測原因の導出の記述例を示す。
 ⑫では、解析該当ロットと過去のロットでの TEG パラメータのシート抵抗のばらつきを評価している。ここでは、該当ロットと過去のロットとのシート抵抗値が、±20% 以内であれば、⑬での結果は問題ない範囲であることが判断されて、Idd の変動はシート抵抗が要因ではないということを表している。

プロセスパラメーターの評価

```

evaluation_process_parameter (
  {TargetRsheets, Ref_Rsheet},          --- ⑫
  {Result},

  method(
    case(Ref_Rsheet * 0.8 =< TargetRsheets and TargetRsheets =< Ref_Rsheet * 1.2
      -> Result := `normal`               --- ⑬
    )
  )
)

```

```

true
-> Result := `process_abnormal`
)
)
)
    
```

4. 記述実験

4.1 知識の記述

不良解析の業務知識は、不良項目とそのテストデータや統計データとの関係のように属性で表わされる技術情報と、この技術情報を基に不良現象から要因を検証する知識ルールとに大別できる。

本論文では、試行錯誤を含む計算手順を自然に表現出来るように考案された知識表現言語 DSP [4][5][6]を用いて知識ルールの記述実験を行った。

4.2 システムの概要

試作した知識ベースシステムの構成を図6に示す。外部にあるテストデータを取込むデータベース①と、前章で示した技術者の知識を知識ルールとして格納するルールベース②、これはユーザーインターフェース③を介して、知識ルールの追加、アップデート、メンテナンスを可能とする。また、テストデータベースと知識ルールから結果を推定する推論機構とその入出力の表示制御部④から構成されている。推論機構は、適用可能ルールを選択するルール制御部⑤と推論エンジン⑥とからなり、入出力表示制御部④を経由してユーザーインターフェース③とつながっている。

データベース①は、製品毎のテストプログラム、データログから、不良解析に必要な製品名、不良項目、被測定ピン、データ、ウェハーマップ等を外部のテストデータ(YMS等)から取込んで格納している。これらのデータは、ルールベース②に応じて必要な時にアクセスして参照できる。

次にシステムの動作を説明する。ユーザーから歩留り低下の解析実行が入力されると、ユーザーインターフェース③は、解析に必要なテスト結果や測定データをデータベース①に格納する。同時に解析に使われる知識ルールをルールベース②からルール制御部⑤で実行可能なルールが選択されて、推論エンジン⑥での推論結果を入出力表示制御部④を通じて不良推定原因と対策を提示する機構である。

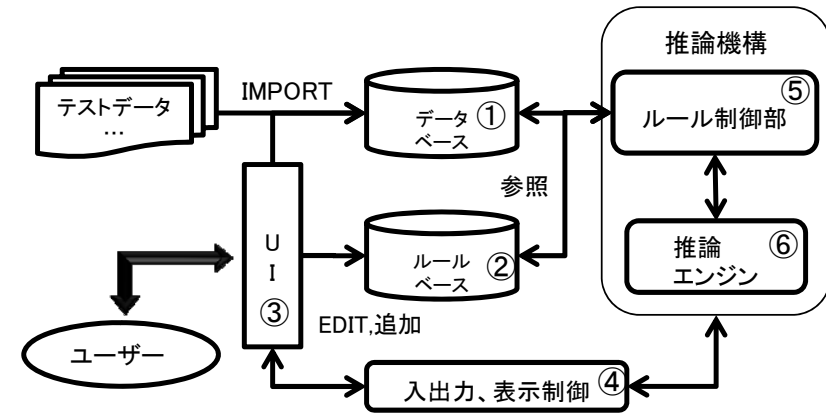


図6 知識ベースシステム概要

図7にユーザーインターフェースの表示例を示す。ユーザーは、(a)の部分で不良解析に必要なテストプログラムやテスト結果、データログ等のファイル名を入力することでテストデータがデータベースに取込まれる。(c)には、格納されたデータに基づいた不良項目と(d)にその項目のテスト条件が表示される。これらの不良項目の条件によって(b)の解析ルールが実行され、解析が進められる。その結果を(e)出力させる。これは、推論機構にて処理された推定原因である。推定原因を評価した後の推奨対策を(f)の部分に表示させている。

4.3 評価

前章の知識表現で述べた表現方法に基づいて知識ルールを記述して、2.2章で述べたテスト不良解析事例の歩留り低下原因を知識ベースシステムで導出できることができた。本事例での解析に要した処理時間は、平均25.4秒(DUT:2600/LOT, Total:6LOT)で、実行はノートPC(Core2Duo CPU @ 2.2GHz, 3GBRAM)を使用した。この結果から不良解析の知識ルールを充実することにより、不良解析の大きな効率化が可能となる。そのためにも、今後、ユーザーが知識・ノウハウを入力可能とするルールエディタを構築してシステムの有効性を高めること、解析時間や精度などについての定量的な効果評価を行う比較実験が必要である。また、さらに実用性を考慮して、解析プロセスを可視化できる表示機能を実装する予定である。

(a) Input

(b) Create Test Item List
 Data Processing : Target Test
 Evaluation : Phenomenon
 Create Failure Cause List
 Data Processing : Other Lots Test
 Create Process Parameter List
 Process Parameter Selection
 Evaluation : Process Parameter
 Evaluation : Process Variation
 Evaluation : Test Condition(Limits)

(c) Test Condition

P.Name	BIN	TestID	TestName	Freq
Sample	1	100	Idd	417

(d) Test Name: Idd, Resc: DPS, Hi_Limit: 1.0 mA, Lo_Limit: 0.58 mA, PIN: Vcc, DC: None, AC: None, SRng: R8V, SVal: 1 V, MRng: M8MA, Test Term: Idd

(e) Presumed cause list
 プロセス(バラつき起因)
 テストリミット起因
 不明の原因

(f) Output (Evaluation Result)
 測定値とプロセスパラメータに相関があるが、プロセス評価は問題ないので原因ではない。
 バラつきを考慮したテストリミットとなっていない可能性があります

図 7 テスト条件，解析結果ページ例

5. まとめ

本論文では、歩留り低下のテスト不良解析を知識ベースシステムにおいて実施することにより、不良原因の推定と対策の提示を行う手法を提案した。この手法は、根本原因へのフィードバックを迅速に行うことが出来て、歩留り向上につながれると考える。また、知識ルールを知ることにより新人技術者への教育や技術者個人が持っている知識やノウハウの保守・継承にも有効であると考えられる。

本事例以外にも、不良原因は設計起因、製造起因と多岐にわたるため、知識獲得を

進めていくと知識ルールが膨大になることが予想される。その対策として、適用可能ルールを効率的に運用するために図 6 のルール制御部⑤を設けて、知識ルールの優先順位から推論結果（歩留り低下原因）を導出する手法を導入する予定である。

また、知識獲得には、歩留りやコストが絡むため社外秘の情報が多く、本システムを有効に活用できるようにユーザー（技術者）自身での知識ルールの入力を支援するシステムの検討も必要である。

謝辞 九州工業大学情報工学府および福岡県産業・科学技術振興財団の関係者の皆様に、謹んで感謝の意を表する。

参考文献

- 1) LSI テスティング学会[編], LSI テスティングハンドブック, 2 編 1.4 pp.117-123, オーム社
- 2) 山口秀行, 長澤勲, 梅田政信, 望月雅光, 章志華: プラント機器の基本設計支援のための知識表現モデル, 情報処理学会論文誌, Vol.41, No.11, pp.3180-3191 (2000).
- 3) 梅田政信, 長澤勲, 樋口達治, 永田良人, 設計計算のプログラム書法, 信学技報, AI91-60, pp.25-32 (1991)
- 4) 手越義明, 長澤勲, 前田潤滋, 牧野稔, 建築物設計における小規模な組み合わせ選択問題の一解法, 日本建築学会計画系論文集, No.405, pp. 157-165(1989).
- 5) 長澤勲, 前田潤滋, 手越義明, 牧野稔, 建築物設計支援システムにおける小規模な組合せ選択問題のためのプログラミング手法, 日本建築学会計画系論文集, No.417, pp. 157-166(1990).
- 6) Umeda, M., Nagasawa, I. and Higuchi, T., The Element of Programming Style in Design Calculations, *Proceedings of the Ninth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pp. 77-86 (1996)
- 7) 片山卓也, 属性文法型計算モデル, 情報処理, Vol.24, No.2, pp.147-155 (1983).
- 8) 梅田政信, 長澤勲, 伊藤公俊, 標準部品にかかわる技術情報流通のための知識表現モデル, 情報処理学会論文誌, Vol.38, No.10, pp.1905-1918 (1997).
- 9) 上野春樹, 小山照夫, エキスパートシステム, オーム社, 1988.
- 10) 大原育夫, 人工知能の基礎知識, 近代科学社, (2008).
- 11) 木下哲男, 人工知能と知識処理, 昭晃堂, (2009).