平成 21 年度　学位論文

再構成可能テスタ・アーキテクチャおよび汎用テスタ言語とその応用に関する研究

Reconfigurable Tester Architecture and General Tester Language and Their Applications

佐藤　正幸
東京都立大学大学院 工学研究科
電気工学専攻 博士課程
平成 21 年 8 月

論文要旨

近年の半導体集積回路(VLSI)におけるプロセス加工技術の発展により，大規模且つ高速動作のVLSIが製造されるようになった．最近では，CPU(Central Processing Unit)やメモリなどのシステムが内蔵されたSoC(System on Chip)が開発されている．SoCの製造不良はテスト工程で検出しなければならない．SoCをテストするためには，テスタと呼ばれるテスト装置で測定する．テスタは計測リソース(テスタ・リソースとも呼ばれる)を多く持ち，それを制御してテストを実行している．その制御はテスト・プログラムとして記述されるが，高額なテスタ設備と膨大なデバック時間を必要とする．

テスタのアーキテクチャとしてさまざまな提案がなされており，シェアード・リソース・テスタやパーピン・テスタなどが使われている．また，テスト容易化設計(DFT: Design For Testability)に特化したDFTテスタも開発されている．これらのテスタを活用するには，そのアーキテクチャの十分な理解が必要で特殊な技術になっている．

テスタ用プログラムを記述するテスタ言語は，テスタがテスタ・リソースの制御装置であるために，テスタ機種ごとに記述仕様が異なる．テスト・プログラムはテスタごとに作成しなければならず，テスタに合わせたデバックが必要である．テスタは高価であるため，DFT技術でテストが容易になっても，テスト・コストが下がらないという問題が顕在化している．そこで本研究では，テスト・コストを低減する目的として，テスタの低価格化やテスト・プログラムの作成容易性の改善，および将来のテスト技術の可能性を探求した．

本論文は，6章から構成され，その要旨は以下の通りである．

第1章は序論であり，VLSIの設計および製造におけるテスト技術の位置付けと現状を示し，本研究の目的とその意義を述べる．

第2章では，本研究の背景であるVLSIのテスト技術，テスタのアーキテクチャおよび仮想テスタ技術について述べる．

第3章では，テスタの低価格化を目的として，仮想テスタ技術の知見から再構成可能テスタを提案した．テスタは構造を持ち，テスト・プログラムによってテスタ・リソースが記述されている．それ故，テスト・プログラムから各テストごとに必要なテスタ・リソースが把握でき，そのハードウェア記述が可能である．これを再構成デバイスであるFPGA(Field Programmable Gate Array)で実現することにより，小規模のハードウェアで実現可能である．テストごとに再構成すれば，すべてのテスタ・リソースを持つことなく，プリント基板型の安価なテスタが構成できた．このことにより，テスト・コスト低減に寄与した．

第4章では，テスト・プログラムの作成容易性の改善について述べた．テスタで使われるテスタ言語について調査した．テスタ言語として， FORTRAN型言語，BASIC型言語，およびC型言語が使われている．それらの記述はテスタ機種ごとに違っている．しかし，テスタ・リソースの形態自体はそれぞれのテスタ間で共通性がある．このテスタ・リソースをC言語関数で表現して記述するテスタ構造表現言語を提案した．これをGTL(General Tester Language)と呼び，種々のテスタ機種のテスト・プログラム記述に対応できることを確認した．また，GTLはテスタ構造を表現しているので，記述されたGTLのテスト・プログラムから必要なテスタ・リソースを抽出して，最適なテスタを照会するツールを開発した．これは，インターネットを使い公開することで，テスタを所有していなくてもSoCのテストがサービスできる環境を構築した．このことで，SoCテストの普及を試みた．

第5章では，将来のテスト技術の可能性の一つとして，テスタ・オン・チップの研究を述べた．再構成可能なFPGA自体も半導体であるので，SoCに内蔵すれば，SoC内でテスタを逐次構築してテストが実行できる．しかし，現時点でFPGAをSoCに内蔵することは現実的でないため，SoC内のメモリに注目した．特に，SRAMはSoCでは必ず使われるメモリである．このSRAMを用いてFPGAを構築する技術を開発した．これをMPLD (Memory Programmable Logic Device)と呼び，そのプロトタイプ開発を行った．これは，メモリ内のSRAMブロックの交互配置による論理要素と配線要素をプログラマブルに構築できるものである．プロトタイプは，0.18μmプロセスを使い試作した．現在はその動作確認と論理合成ツールの開発を推進している．最終的にSoCのメモリを使い，テスト時にSoCが必要としているテストの実行が可能になる．DFTテストだけでは不十分なテストに対して，ファンクション・テストなどの実行が可能となることを示すものである．

第 6 章は結論であり，本研究で得られた成果と今後の課題を総括した．

目 次

図 目 次

表 目 次

第 1 章

序論

本章の内容

1.1　VLSI のテスト技術

1.1.1　VLSI の設計とテスト

半導体プロセス加工技術の進展に伴い，トランジスタからなる回路が大規模に一つのチップ上に集積できるようになった．これを VLSI(Very Large Scale Integration)という．VLSI は，高度に集積されシステム全体がチップ上に実現されるシステムオンチップ (SoC: System on Chip) 技術へと発展している．設計手法としては，RTL (Register Transfer Level) 高位記述や IP(Intellectual Property) の再利用などが設計の効率化を図る手法として活用されている．

RTL 高位記述は，HDL (Hardware Description Language) を使用することにより，設計工程として論理合成 (Logic synthesis) や自動配線ができる．これら一連の作業は EDA (Electronic Design Automation) ツールを利用して進歩してきた．

このように，大規模化された回路に対しては，外部より高品質のテストを行うことは非常に困難である．

テストの品質を上げることは DFT(Design For Testability)技術の進歩で向上したが，テスト･コストの低減には問題が多い. これは, VLSI のテストに汎用テスタが使われるが，その装置コストや維持管理コストなどの問題を考慮しなければならないからである．

ますます複雑化し大規模化する VLSI をテストするためには，設計の後工程としてテスト設計したのでは手遅れである．設計の初期段階よりテストに対する配慮を行い，設計の中にテストを考慮させることが必要である．また，テストを行う汎用テスタへの考慮も必要である．VLSI テスト技術に関しては，設計から量産までを考慮し，そこで使われる装置の幅広い配慮が必要である．

1.1.2　VLSI のテスト内容

現在における VLSI のテストは，一般には外部の汎用テスタを使用して行われ，機能テストを行うのが従来テスト手法であった．しかし，テスト容易化設計を配慮したテス

トが主流になってきた．そのひとつとして縮退故障を主な故障の対象とし，構造テストが使われ始めた．しかし、機能テストも，顧客テスト要求として実施がされている．これは顧客での実動作速度テストが必要であるからである．更に，静特性試験である DC テストも依然重要なテストとなっている．これは多ピン化される VLSI において各デバイス・ピンの特性を保障しなければならず，そのテストはアナログ・テストであることから測定時間の掛かるテストになる．最近では，高速信号処理機能がシステム機器の性能向上に重要な役割となっている．その高速信号処理機能では，汎用テスタで測定できない周波数帯のテスト課題も大きい．VLSI のテストでは，テスト品質を上げるためにはテスト・コストが増加するなど，一般にテスト品質とテスト・コストとの関係は相反する場合が多く，最終的には両者のトレードオフによるテスト設計がなされる．テスト品質の 1 つの指標である故障検出率は，テスト容易化設計でその向上が実現できている．そして，その効率も上がり，テスト・コストの削減に寄与している．しかし，DC テストや機能テスト，高速信号処理機能のテスト時間は増加しているため，全体的なテスト時間は増加して，そのテスト・コスト削減には十分寄与していない．いずれにしても，高価な汎用テスタを使わなくてはならなく，その活用技術の研究開発が必要である．

ここで，テスト項目の内容とテスト・コストについて下記に示す．

(1) 構造テスト

テスト品質を上げることは，被テスト回路 (CUT: Circuit Under Test) に存在する故障をモデル化し，そのモデル化された故障を対象としてテストを考慮することが進められている．故障モデルは，従来の縮退故障モデル，短絡故障モデルや遅延故障モデルが研究開発されている．しかし，最近の微細化プロセスでは，クロストークや電源ノイズの影響を配慮することが必要となりつつある．抽象化が進む設計技術との乖離が進んでおり，その検出テスト・パターン生成が難しいなどの課題がある．

回路規模の増大に伴い，テストは一層困難となっている．そのため，設計段階よりテストを容易にするためのテスト容易化設計 (DFT) がなされている．VLSI は多くの順序

回路を含んでいて，入力と内部状態の膨大な組合せが必要となり，外部入出力端子だけでは満足なテストが出来ない．この対応としては，順序回路のテストに関して，ゲート・レベルでのスキャン設計がなされることが多い．スキャン設計では，論理回路を組合せ回路部分とフリップフロップ部分 (スキャン部分) に分離し，組合せ回路部分に対して外部よりスキャン回路を通し，テスト・パターンを印加することを可能とする．スキャン設計された回路は，組合せ回路と同様のテスト生成手法が適応できるためテストが容易となる．これらの技術を総称して構造テストと言われて良く活用されている．

(2) 静特性試験

上記の構造テストで大規模化する VLSI の内部半導体のテストは高い効率でテストできるようになった．しかし，市場に提供する VLSI としては，基本的な静特性試験を行わなければならない．

静特性試験の代表的な項目を下記に示す．

①消費電流テスト

VLSI が消費する電流を規定の VLSI 使用電圧で測定する試験．

②入出力リーク・テスト

VLSI の各デバイス・ピンが適切に処理されていて，システムを構成するとき，お互いに接続して問題ないか，そのリークをテストする試験．

③出力電圧テスト

VLSI の出力ピンが，適切な論理状態(Hi，Low)で所定の電圧値で出力されているかテストする試験．これには所定の負荷電流を印加する処置がいる．

これらのテストはデバイス・ピンごとに試験をしなければならず，テスト時間が掛かるテストである．この改善としては，各汎用テスタで種々なハードウェアの改善がされている．

(3) 機能テスト

VLSI が顧客使用上問題なく動作するかの機能をテストする試験．一般的に VLSI の論理回路のテストは構造テストでテストされるが，顧客要望で機能テストが実施されることもある．

回路の大規模化に伴い，高品質のテストを保証するために必要とされるテスト・パターンは増加する一方であり，そのためにテスト時間（テスト長）は長大化している．テスト時間の長大化は，高額な汎用テスタを中心とするテスト設備の利用時間を増加させるため，コスト増加の大きな要因となりつつある．

(4) テスト・コスト

半導体の微細化に伴う動作速度の高速化に対応するために，高速な汎用テスタが必要となり，テスタの使用コストを含めたテスト・コストの増大を招いている．これに対して，低速テスタによる高速デバイスのテストなど，低価格のテスタを用いたテスト手法の確立が課題となる．

(5)　チップ・コスト

多層配線がもたらすテストの困難化（特に信号観測性の悪化）などに対処するためにテスト容易化設計が行われるが，これに伴うエリアオーバヘッドによるチップ・コストの増大が課題となる．

(6) テスト設計コスト

被テスト回路の大規模化に伴い，テスト設計に要する設計工数やテスト生成の計算機コストの増大が課題となる．

また，そのテスト生成されたテスト・パターンを汎用テスタが解釈し動作させるパターン変換やその生成は自動化が進んでいる．しかし，そのパターン長大化に従い，汎用テスタでのデバックは，困難を深めている．これは，テスト設計者と汎用テスタを使うテスティング・エンジニアの間で，テスト・パターンの情報伝達に問題がある．それを合理化するツールも開発されているが十分ではなく，汎用テスタでのテスト・パターン・デバックが必要である．そのデバックのテスタ使用コストの増加も問題である．

最近は，テスト容易化設計のテスト・パターンから故障箇所を解析して，VLSI の歩留向上による半導体製造の収益性向上も課題で，その研究が進んでいる．その解析には汎用テスタが使われおり，そのテスタ使用コストも問題である．また，その不良テスト・パターンから不良箇所を確定するツールも高価であり，VLSI の歩留向上も大きな課題である．

1.2　開発～量産フロー

古典的な VLSI の開発～量産フローとその改善策を含めて図 1.1 に示した．図の左側が古典的なフローで，右側が後述する仮想テスタでの改善提案である[1].

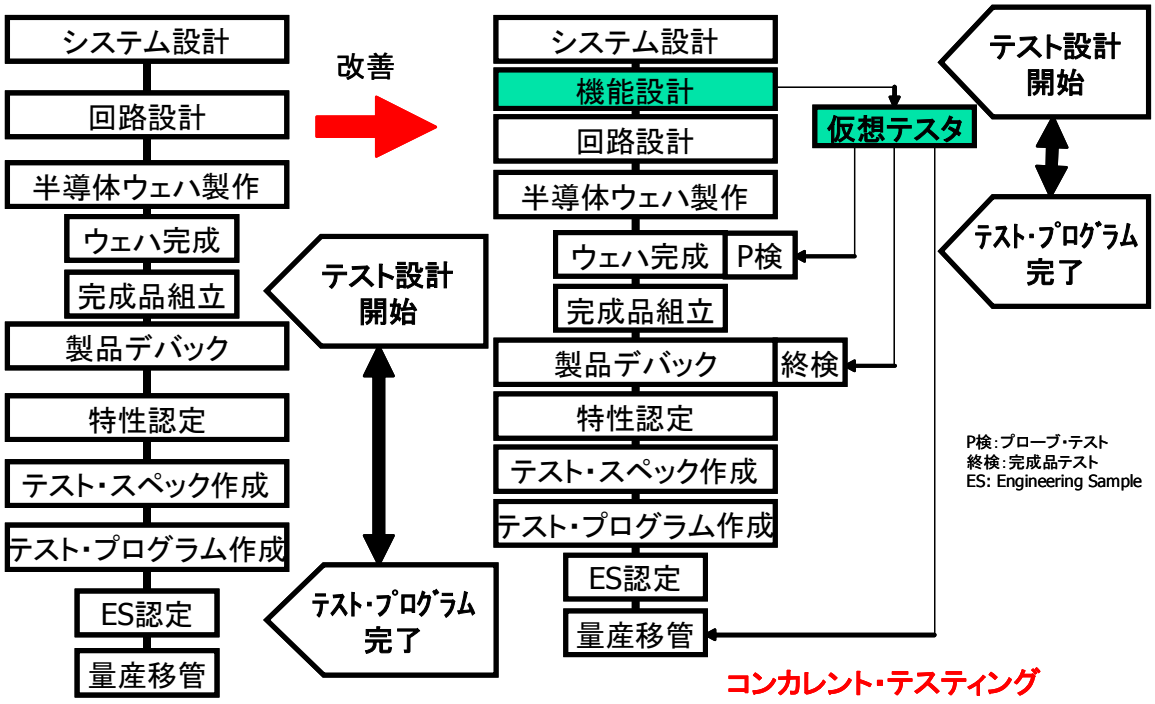


図 1.1　開発～量産フロー

Fig. 1.1　Developmennt　～　Mass production flow

古典的な開発フローは，システム設計から始まり種々の設計工程を経て回路設計および回路生成に至る．ここでは省いているが，その設計でプロセス・マスクを作成して半導体前工程に製品が投入される．その結果，試作ウェハが完成して製品評価に入る．ウェハでデバックすることも最近では多いが,詳細な特性評価を考えるとウェハから VLSI チップを切り出し，半導体パッケージに組み立てた後にテスト評価することが多い．
テスト・デバックはこの時点で行われるが，そのテスト環境を使い特性認定に入る．その特性認定データから最終的に量産でテストされる項目が決定され，テスト・スペックが作成される．そのテスト・スペックを基に，量産用にテスト・プログラムを作成して，ES(Engineering Sample)評価で顧客認定を取り，量産の立ち上げに入る．

これが古典的な開発～量産フローである．最近の技術の進展でさまざまな改善がされているが，テスト自体の実施は半導体工程の後処理として位置付けられている．

1.2.1　各工程とテストの関係

古典的な開発～量産フローの各工程でのテストとの関係を下記にまとめる．

（１）システム設計

システム設計では，顧客仕様が満たされるように，設計プランや機能仕様が求められる．この段階でテストが考慮されることが望ましいが，一般的に量産での詳細なテストが考慮されることが少ない．最近の VLSI はモジュール設計になっており，その際，DFT の考慮がされなければ，モジュール単位でのテストが不可能になり，テストの実行に大きな問題を残すことになる．このようにこの段階でのテスト検討は重要である．

製品スペックは顧客要求仕様で決まっているが，テストの詳細テスト・スペックは，明確ではなく，概略スペックとして提示されているのがこの段階である．

（２）回路設計

システム設計を経て種々の設計ツールを使い回路設計に入る．ロジックの回路では論理回路が確定しているので，自動設計が整備されている．自動設計でゲート・レベルの

生成がされるので，その故障仮定から故障検出率や故障テスト・パターン生成が可能である．このとき，スキャン・テストが考慮されていれば，そのテスト端子の形成とその明確化がテスタ上でのテスト実行としては重要になる．また，BIST(Built In Self Test)が使われていれば，その制御仕様の明確化がテスタ上での処理に重要となる．また，顧客の要望から機能テストも無視できなく，そのテスト・パターン生成も必要である．

なお，アナログ回路については基本的な DFT 手法がなく，アドホック的なテスト手法でテストを考慮しなければならない．

いずれにしても，この段階では使用されるテスタが決まっておらず，テスト手法のみの検討になってしまうのが現状である．

（3）　ウェハ完

回路設計からプロセス・マスク作成，ウェハ製作まで経てウェハの製作が完了する．この段階のテストの問題は，ウェハ・テストをするためのプローブ・カードの準備である．チップの面積最適化の設計変更から，チップでのパッド・レイアウトは多々変更されることがある．それに対応したプローブ・カードの事前準備が必要であるが，一般的に，ウェハ完までに間に合わず，以下に述べる組立後の完成品パッケージでのテストとなっている．

（4）　組立後の完成品

VLSI のパッケージは，標準化されており顧客要求から，開発初期段階で決まっていることが多い．この関係で実際のテスタの製品デバックは，この段階から始まるのが通常である．このとき，設計の評価としては評価項目に従い種々のテスタを使う．特に，デバック性を考慮して後述するパーピン・アーキテクチャ・テスタが使われる．さらに，アナログ内蔵デバイスでは，計測器を使って評価する場合もある．DFT テストについては DFT 専用テスタでデバックする．このように，この段階のテスタ・デバックは，製品の性能や品質状況を確認するために複雑になる．最終量産でのテスト・コストを考慮したテスト手法の検討は，この段階でなされない問題がある．

（５）　特性評価

上記の段階でのテスト・データを基に特性評価が行われ，特性認定がなされる．認定に問題の項目があれば追加テストや評価が実行される．

（６）　テスト・スペック作成

上記特性認定を踏まえ，顧客品質が満足されるテスト・スペックが作成される．このとき，テスト・コストが重要な課題となり，テスト項目の選定やテスト手法の見直しがされる．特に，テスタ機種選定は重要な課題であり，使用されるテスタに併せたスペックの決定がなされる．

（７）　テスト・プログラム作成

テスト・スペックに基づいたテスト・プログラムの作成が行われ，それが再度デバックされる．このとき，テスト・コストを配慮した多数個同時測定も行われるので，そのデバック実行は困難を極めている．

（８）　ＥＳ認定

顧客認定を取るために，多くのデバイスを上記のテスト・プログラムでテストする．それを ES デバイスとして顧客に納め認定を取る．

（９）　量産

これまで使ってきたテスト・プログラムで量産をする．このとき，テストとして重要なのは，量産後の品質管理や生産性向上のために，データ取得をしやすいようにするのが重要である．

以上述べたように，テスタでの VLSI のデバックはウェハ完か組立後の完成品から開始され，量産立ち上げに向かった対応となっている．また，使用テスタも各種あり，その立ち上げデバックには多くの工数を割いているのが現状である．

1.2.2　課題

VLSI は工業製品であることから, テストもその開発から量産までの工程を含めた総合的な観点が重要になる. 特に, テスト工程は, VLSI 開発の後工程として考えられているのが一般的であり, 不良品を除去する工程として付加価値のない技術と考えられ易い. しかし, VLSI が正常に動作し, 所定の品質を持つようにテストすることは, 製品企画段階で考慮されなければならい重要課題と考える. 要は, テスト工程は VLSI 開発の後工程として位置付けるのではなく, VLSI 企画段階における最初の検討課題として考慮されなければならないものである.

しかし, 現実は VLSI の複雑化と VLSI への顧客要求の変更があり, それが実現できていない. その状況の解決手段として仮想テスタの提案をした(図 1.1 右側). テスタ上でテスト・プログラムを作成して VLSI はデバックされる. そのデバック環境を図 1.2 に示した. デバックすべき被測定デバイス(DUT: Device Under Test)はソケットに挿入され, それとテスト・ヘッドに接続するテスト・ボードおよびテストを実行するテスト・プログラムがある.

図 1.2　デバック環境

Fig. 1.2　Enviroment of test debug

最近，設計技術として HDL 設計を基本とした機能設計が取り込まれている．Verilog や VDHL で VLSI の機能を記述して，その動作検証をするテスト・ベンチが活用されている．このテスト・ベンチの応用として，CPU にテスタの基本構造を表現して，設計段階での事前デバックを実現する．また，テスト・プログラム記述についても，そのデファクト化を進めて，テスタを決めずデバックできる手法を構築すべきである．そのことにより，実際の各工程での使用テスタの選択や展開が容易になる．機能設計段階での最適テスタの照会選択を可能にし，これにテスタのコストを加味することで，最適テスト・コストのテスト手法の実現が可能になるものと考える．VLSI への高機能化，高速化要求から動作が複雑になり，VLSI のテスタ上でのデバックの困難性が指摘されている．それに対応する仮想テスタ技術も，図 1.3 に示すようにテスト・デバックの困難性が半導体ロードマップ委員会(STRJ: Semiconductor Technology Roadmap Committee of Japan)で議論されている[2]．詳細説明は 2.2.1 節にて述べる．

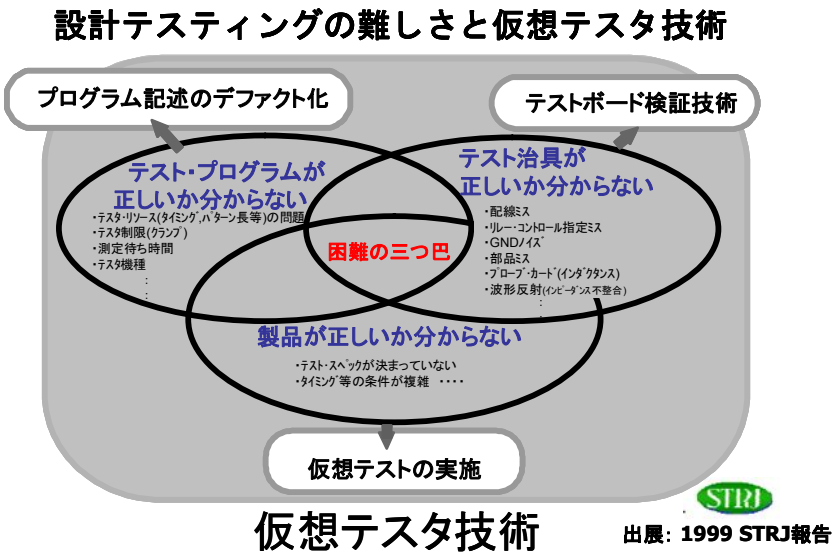


図 1.3　テスト・デバックの困難性

Fig. 1.3　Difficulty of test debug

1.3　本論文の目的と構成

1.3.1　本論文の目的

前節で述べた仮想テスタ技術の必要性とその実現の経緯から図 1.3 に示す内容の技術開発をしてきた．

まず，テスタの低価格化を目指し，仮想テスタ技術の内，仮想テストから構造可変テスタを構想した．これは，基板上の FPGA(Field Programable Gate Array)にテスタを構成していく手法であり，小型で消費電力が少ない低消費電力基板型構造可変テスタを開発した．これは，テスト・コストの大半がテスタ・ハードウェア投資コストからくる現状を解決できる提案である．

テスト・プログラムの作成容易性を目的として，プログラム記述のデファクト化を検討した．このために，テスタ・リソース（テスタ構造）を反映したテスト・プログラム記述できるテスタ構造表現言語を提案した．そして，それを応用してテスト・ソリューションを開発した．

ただし，テスト・ボード検証技術については，回路シミュレーション技術や 3 次元電磁解析技術を使うことから本研究では省略した．

なお，将来のテスト技術探索として構造可変テスタを発展させ，テスタ・オン・チップの研究を推進した．

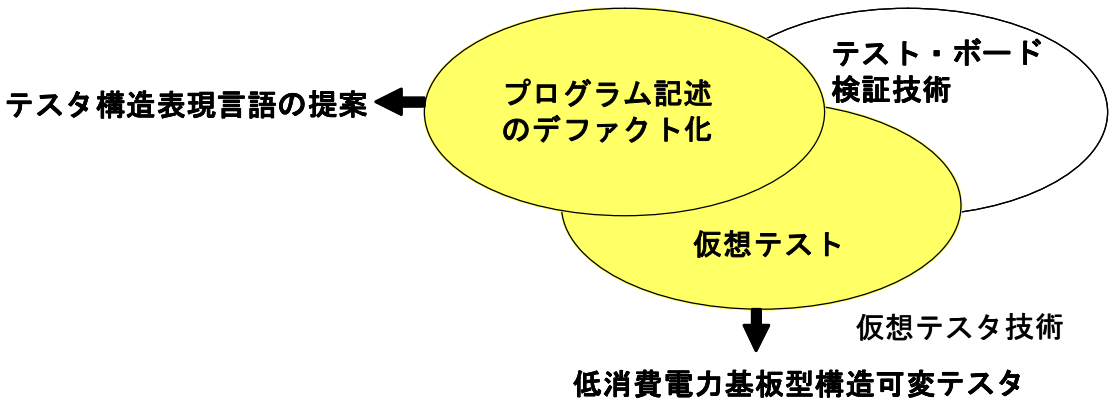


図1.4　研究の目的

Fig. 1.4　Purpose of the research

1.3.2　本論文の構成

第 2 章では，本研究の背景であるテスタ構造について各種テスタ・アーキテクチャを述べる.

第 3 章では，その仮想テスタ，特に，仮想テストの構成手法である HDL 記述の応用からテスト・コスト低減を目的とした低消費電力基板型再構成テスタについて述べる. そして，その応用例を具体的な製品応用として HDD モータ駆動コンボ IC のテスト適用について述べる.

第 4 章では，テスタのテスト・プログラム記述の調査を行い，そのデファクト化を検討した．その内容としては，テスタ構造表現言語を提案し，各テスタで動作可能なテスタ言語を提案する. このテスタ構造表現言語は, テスタ・リソースを表現しているので, テスト・プログラムが必要としているテスタ・リソースの把握が可能になる．その特徴を生かして，記述されたテスト・プログラムに適したテスタの照会機能を持たせることができた. この応用としてインターネットを使ったテスト・ソリューションを説明する.

第 5 章では, 仮想テストの HDL 記述から, VLSI に内蔵されているメモリ部分を使い, テスト回路の搭載, 再構成可能な手法として SRAM を使った一論理回路構成手法の基礎検討を述べる．これは VLSI に BIST を構成する手法を一歩進めるものとして研究した.

第 6 章は結論であり，本論文で得られた知見と残された課題をまとめる.

参考文献

[1] 佐藤正幸,“第 14 章　テスティング技術,”1999 半導体テクノロジー大全, 電子ジャーナル(株)，pp. 231-237，1998 年 10 月.

[2]　1999 年度 STRJ 報告書，第 2 章 2-2 テスト 2-2-4 課題 (8) ATE，1999.

第 2 章
テスタの構造

本章の内容

2.1　テスタ・アーキテクチャ

ここでは，VLSI を測定する汎用テスタに注目して，その構造：テスタ・アーキテクチャについて述べる．

VLSI は，半導体プロセス加工技術の進展に伴い，高速化，多ピン化，高機能化が急速に進んでいる．それに対して汎用テスタを使うテスティング技術も当然ながら，その対応が必要となる．

一般的に汎用テスタは，図 2.1 に示すようにパターン発生器(PG: Pattern Generator)，タイミング発生器(TG: Timing Generator)，デバイス電源(DPS: Digital Programmable Power Supply)，DC 計測系(DC: Direct Current measurement Unit)，ドライバ(DR: Driver)，コンパレータ(COMP: Comparator)で構成される．ドライバとコンパレータを併せ持った構造をピン・エレクトロニクス(P/E: Pin Electronics)と総称される．これらをテスタ・リソースと呼ぶ．テスタコントローラ(CPU)の記憶装置に，記憶されているテスト・パターンやテスト・プログラムがテスタ・コントローラの CPU 機能で解釈され各テスタ・リソースを制御して所定のテストを実行させる．

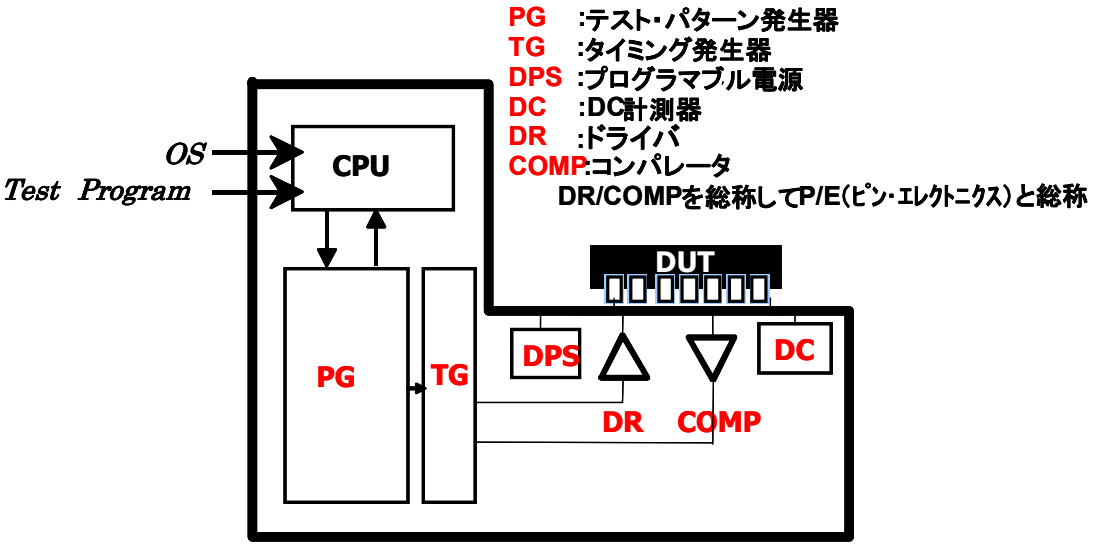


図 2.1　汎用テスタの基本構造

Fig. 2.1　Basic architecture of General-purpose tester

テスタ・リソースの構成から図 2.2 に示すように，大きくシェアード・リソース・テスタ，パーピン・アーキテクチャ・テスタ，フル・パーピン・アーキテクチャの 3 種類に大別される[1]．また，テスト容易化設計(DFT)に特化した DFT テスタがある．

それについて，下記に詳述する．

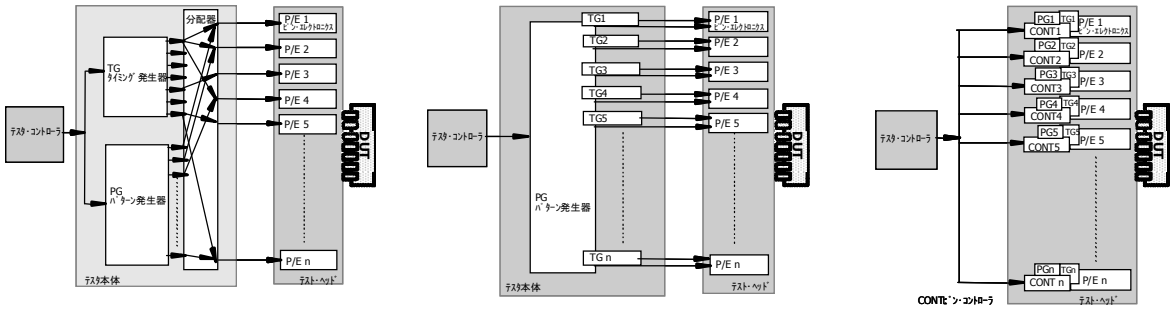


図 2.2　テスタ・アーキテクチャ

Fig. 2.2　Tester architectures

2.1.1　シェアード・リソース・テスタ

テスタは，過去，テスタ・ハードウェア，特に，タイミング発生器などの高速機能が

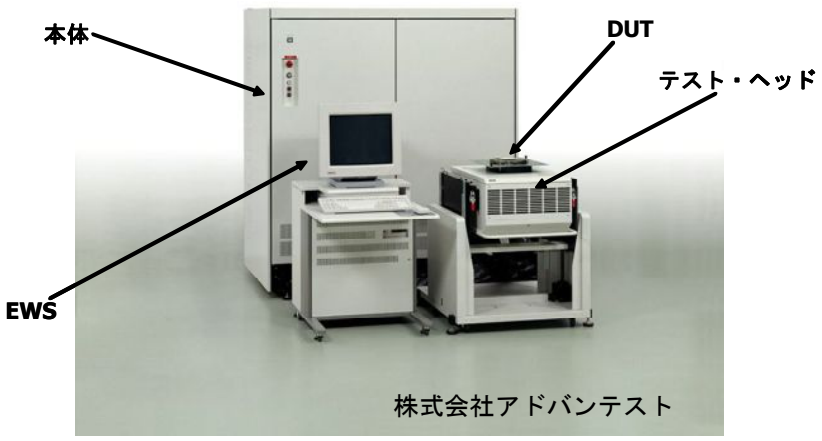


図 2.3　シェアード・リソース・テスタの写真

Fig. 2.3　Photograph of shared resource tester

高価だった．そのために，テスタ・リソース(タイミング発生器など)を分配器で分配し，各ピン･エレクトロニクスに供給するシェアード･リソース･テスタが主流であった[2,3]．テスト内容に従いテスタ・リソースの分配をしなければならない煩わしさがある．製品デバックを中心とする設計での使用ではデバック効率の悪いアーキテクチャである．その外観の例を図 2.3 に示す．

テスタ本体と被測定デバイス(DUT)を装着してテストを実施するテスト・ヘッドがあり，それらをコントロールする CPU であるエンジニアリング・ワーク・ステーション(EWS)がある．

2.1.2　パーピン・アーキテクチャ・テスタ

上記シェアード・リソース・テスタでの設計製品デバックの効率改善の要請から，各ピン・エレクトロニクスにタイミング発生器を有するパーピン・アーキテクチャ・テスタ（パーピン・テスタとも言う）が実用化されてきた[4]．各ピン・エレクトロニクスにタイミング発生器を持つことから，テスト内容によるタイミング発生器の分配を考慮せ

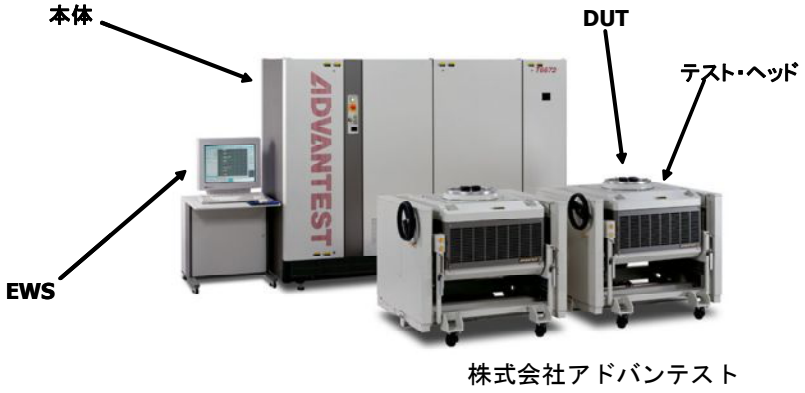


図 2.4　パーピン・アーキテクチャ・テスタの写真

Fig. 2.4　Photograph of per pin tester

ずにテストが実施できる．しかし，タイミング発生器を，各ピン・エレクトロニクスに持たせたためにその価格は高価になり，当初，設計段階の製品デバックのみに使われていた．しかし，最近の最先端半導体技術が使われ，その低価格化も進み量産テストでも使われ始めている．その外観を図 2.4 に示す．

2.1.3　フル・パーピン・アーキテクチャ

このパーピン・アーキテクチャ・テスタを進めたフル・パーピン・アーキテクチャがある．これは各ピン・エレクトロニクスにタイミング発生器だけでなく，パターン発生器もピンごとに個別に持たせたものである．そのテスタの例を図 2.5 に示す[5].

このテスタの例では，各ピン・エレクトロニクスにプロセッサを持つテスト・プロセッサ・パーピン技術が使っている．ピンごとにパターン発生器の制御ができることから VLSI テストの個別機能が測定可能になり，効率的なテストができるとされている．このテスト手法はコンカレント・テストと呼ばれている．

図 2.5　フル・パーピン・アーキテクチャの写真

Fig. 2.5　Photograph of full per pin tester

2.1.4　DFT テスタ

最近，DFT テストに特化した DFT テスタが使われ始めた．DFT テスタは，テスト・パターンの入力／出力が少ないことや，パターン制御が簡単なことから簡易なテスタにすることができる．DFT テスト・デバック専用テスタを図 2.6 に示す[6]．デバック専用テスタのため，デバイス電源(DPS)，DC 計測系(DC)などは省かれオプションとなっている．

図 2.6　デバック用 DFT テスタの写真

Fig. 2.6　Photograph of DFT tester fo debug

量産用 DFT テスタを図 2.7 に示す[7]．量産用となると通常テスタに近い構成となり DFT 専用テスタでも高価なテスタとなる．

図 2.7　量産用 DFT テスタの写真

Fig. 2.7　Photograph of DFT tester for mass production

2.2　仮想テスタ技術の必要性とこれまでの研究

2.2.1　仮想テスタ技術の必要性とその区分

第 1 章　序論で提案した図 1.3 仮想テスタ技術について詳述する.

（1）仮想テストの実施

VLSI の設計には複数のエンジニアが携わるために, 全体の回路動作について理解が不十分になりやすい. また, VLSI に対する顧客要求から設計変更が多くなされ, 最終的なテスト・スペックが決まらないことも多々ある. さらに, 動作タイミングも複雑になるなど問題があり, そのテスト実行に課題が多くなってきた.

テスタは制御機器の 1 つであり, その制御が適確でなければならない. 特に, 設計検証で使われた検証パターンは, 一般的にそのままテスタに搭載し実行できない. そのために, テスト・パターン変換が必要であり, それが適切に行われたかは, テスタ実機でのデバックとなっている. そのテスタ実機デバックは時間もコストもかかることから, CPU 上にテスタを表現して設計データと併せた事前デバック環境が開発されてきた. これを仮想テスト技術と呼ぶ.

（2）テスト・ボード検証技術

テスタでのデバック環境としては, DUT と接続を持たせるテスト・ボードがある. これは, 単純な接続の場合もあるが, 一般的には周辺部品を含めた複雑な接続となることが多い.

そのことから, 配線ミスや接続を切り替えるためのリレーが使われる. そのリレーをコントロールする指定にミスが起こりやすい. さらに, GND の配置による GND ノイズの問題も見逃せない. その他, 周辺部品のミスやインピーダンス不整合による波形反射の問題も最近では問題になっている. そのために, 作成しているテスト・ボード（治具）がデバック前に正しいか分からないのが現状である.

最近，システム基板のミュレーション技術も進歩しているので，その技術を活用したテスト・ボード検証技術の構築が期待される．

（3）プログラム記述のデファクト化

テスタは一種の計測装置であり制御装置である．このことから，計測リソース(テスタ・リソースとも呼ぶ)の保有状況はテスト実行の方法を考える上で大きな課題である．テスタ・リソースを多く持つテスタは高価であり，少ないテスタ・リソースのテスタでテスト実行するのが，コスト面で優位である．このテスタ・リソースの保有状況はテスタにより違い，保有状況を十分に把握したエンジニアが必要である．また，そのテスタ・リソースの制御命令であるテスト・プログラムに熟知しなければならない．このテスト・プログラムはテスタ機種ごとに違い，使用するテスタの専門エンジニアが必要である．また，テスト・プログラムはを専門のエンジニアがいつも作成するとは限らない．専門でないエンジニアが作成するテスト・プログラムが正しいかどうかは十分に事前把握できるわけではない．

テスタごとに違うテスト・プログラムを補完するプログラム記述のデファクト化が必要となっている．

以上，記載した問題点が重なり合い，VLSI のテスタ・デバックは困難を極める．特に，新製品をデバックする設計テスティングは困難である．この問題を解決するために，仮想テスト技術とテスト・ボード検証技術およびプログラム記述のデファクト化を整備することがテスティング技術の重要課題と考える．

2.2.2　仮想テスタ技術に関するこれまでの状況

仮想テスタ技術については,構想され検討してきたが,その実現には問題が多かった．現在では，デジタル信号のみを扱う仮想テストが実現され使用されてきた．その例を図2.8に示す．これは横河電機株式会社のPreTestStationの例である[8]．テスト・パターンと

タイミングに限定して論理シミュレーションを設計データと行うものである．これにより，テスタ実機評価のデバック時間が半減できたとの報告がされている．

このことは，製品検証で生成された検証パターンが，一般的にテスタに直接搭載できず，テスタ・リソース（特に，タイミング発生器）を配慮したテスト・パターン変換をしなければならないからである．そのテスト・パターン変換が適切かは，テスタを考慮した再検証が必要である．これを事前にCPU上で検証するのが仮想テストの意義である．

仮想テストの構成には，テスタのハードウェア情報を回路として搭載する手法もあるが，テスタのテスト動作やテスト・プログラムからHDL記述して搭載する手法もある．現状では，テスト・パターンとタイミングに限定したHDL記述による論理シミュレーションの仮想テスト技術が可能である．しかし，その他のテスタ・リソース（例えば，ピン・エレクトロニクス）などのアナログ値を扱うのは困難である．これは，回路シミュレーションに時間がかかることや，VLSIのアナログ・シミュレーション・モデルが確立していないからである．今後の回路シミュレーションの進歩が期待されるところである．

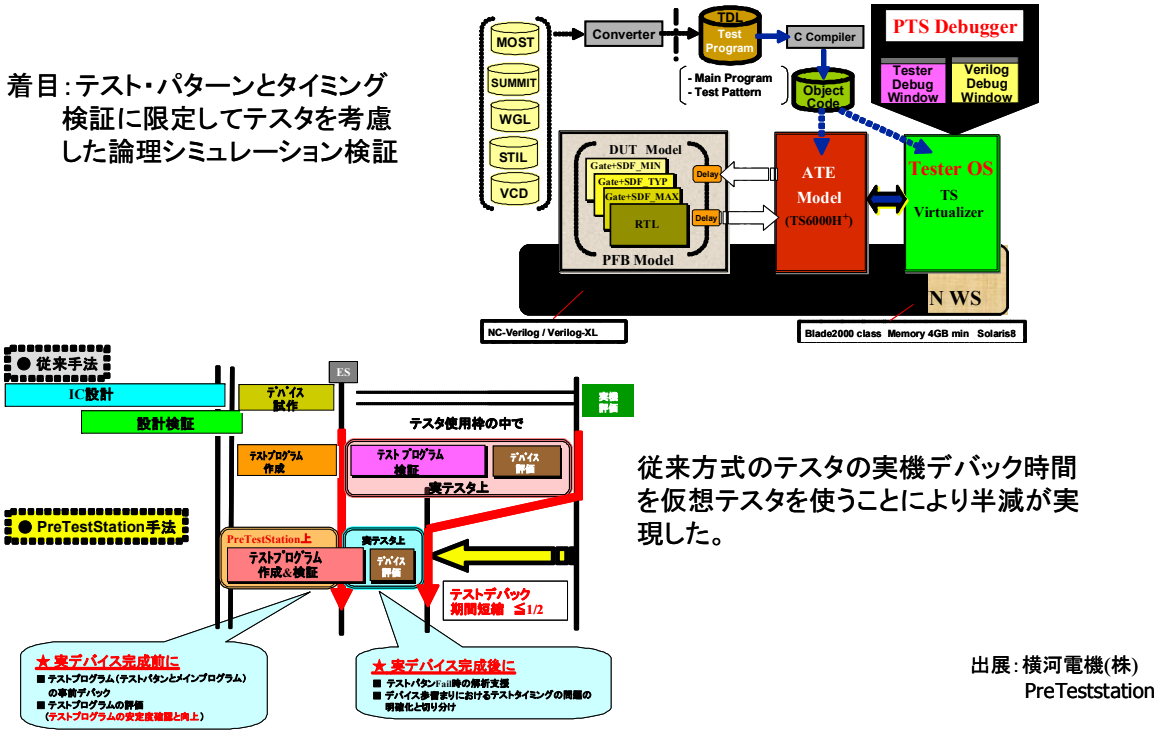


図2.8　仮想テスタの現状

Fig. 2.8　Current state of virtual tester

2.3　まとめ

テスタは上記のようにさまざまなアーキテクチャがある．基本構造は図 2.1 に示したように同じであるが，テスタ・リソースの保有の仕方やその制御方法が異なる．多くのテスタ・リソースを持つことにより，汎用性が高くテスト性も良いが，高価となる．低コスト化に限定したテスタはテスト性が制限されてしまう．最近では，低コスト化を意識して半導体製品の一分野に特化した形でテスタが開発されるが，テスタの多様化を招いている．また，テスタを制御するテスト・プログラムもテスタ機種ごと違い，その記述仕様を熟知していないと，多くのデバック時間を費やしてしまう．

各テスタの上での仮想テスタやテスト・プログラム作成容易さを考慮したツールは開発され提供されいているがテスタ間で一貫性がなく，使用するテスタに制限された環境が一般的である．テスト・コストを削減する技術についても，テスタ個別の特徴を生かした技術であり汎用性がない．

これらは，テスト・コストを低減するための技術課題であり，その研究開発が必要とされている．

参考文献

[1] 佐藤正幸，“第 12 章　テスティング技術，”1998 半導体テクノロジー大全，電子ジャーナル(株)，pp．210-214，1997 年 9 月 30 日．

[2] T. Kazamaki, H. Maruyama, and S. Sumida, “Trial Model of 100 MHz Test Station for High Speed LSI Test System,” Proc. International Test Conference, pp. 611-617, 1978.

[3] T. Kazamaki, “A 100MHz Tester – Challenge to New Horizon of Testing High Speed LSI,” Proc. International Test Conference, pp. 618-625, 1979.

[4] S. Bisset, “The Development of a Tester-per-Pin VLSI Test System Architecture,” Proc. International Test Conference, pp. 151-157, 1983.

[5] Verigy,http://www.verigy.com/ate/products/V93000/index.htm

[6] teseda,http://www.teseda.com/prod_v520.shtml

[7] Inovys,http://grouper.ieee.org/groups/1450/dot0/ITC2002/ITC-2002-Inovys.ppt

[8] 横河電機,http://www.yokogawa.co.jp/TR/2003-03.htm#01

第 3 章

低消費電力基板型再構成テスタの開発

本章の内容

3.1　はじめに

ディープ・サブミクロン半導体技術の出現により，VLSI のテストはますます困難になっている．従来から，大型汎用テスタが VLSI テストのために使われてきた[1]．メモリ・デバイスにはメモリ・テスタが使用され，ロジック・デバイスには汎用ロジック・テスタが使われている．また，ロジック機能とアナログ機能を内蔵したシステムオンチップ(SoC: System on Chip)はミックスト信号テスタでテストされるか，ロジック・テスタとアナログ・テスタの両方を用いてテストされている．

第 2 章　テスタの構造　2.1 節　テスタ・アーキテクチャの図 2.1 に汎用テスタの基本構造を示した．テスタは CPU 内の主メモリに記憶されたテスト・プログラムを，オペレーティング・システム(OS)の下でインタプリタが逐次解釈し，テスタ・リソースを制御しながらテストを実行する[2,3]．テスタ・リソースとして，被テスト・デバイス（DUT: Device Under Test）へ電源を供給するデジタル・プログラマブル電源(DPS: Digital programmable Power Supply)，DUT の直流特性：測定する DC 計測器(DC: Direct Current measurement unit)，DUT に信号を入力するドライバ(DR: Driver)，DUT からの出力信号を判定する比較器(COMP: Comparator)を有する．このドライバとコンパレータはピン・エレクトロニクスと称されている．ドライバとコンパレータを 1 ピン毎に併せ持ったピン・エレクトロニクスの場合は，実時間で入出力を切り替える機能を持つ．テスタは，テスト・データを生成するパターン発生器(PG: Pattern Generator)，テスト・データの印加タイミングや判定タイミングを発生するタイミング発生器(TG: Timing Generator)を有している．

大型汎用テスタは DUT の要求に従って，その構造を進化させてきた．1970 年代に開発されたシェアード・リソース・テスタは初期の方式であり，TG のような高価なテスタ・リソースを複数ピンで共有する構造である[4,5]．しかしながら，VLSI のタイミング・マージンを検証することが難しかった．この問題に対応するため，ピン毎に TG を持たせたパーピン・テスタが開発された[6]．また，近年，テスタ・リソースをピン毎に持たせたテスト・プロセッサ・パー・ピン技術[7]やタイム・ドリブン方式テスタ[8]などの構造が提案されている．このため，テスタ言語で記述可能なプログラムは機種毎に異なって

おり，そのテスタ言語を駆使して製品ごとのテスト・プログラムが作成される．このようにテスト・プログラムはテスタの構造を表現したものであり，テストされる製品に必要なテスタ・リソースを利用したプログラム記述によって作成されている．

大型汎用テスタは，高速化する VLSI に伴って，高速化されている．その結果，大型汎用テスタは非常に高価になり(8000 万円～5 億円)，テスト・コストが高騰する一つの要因になっている．テスト・コスト削減は，多数個同時測定やその他のテスト手法を用いて行われてきた[9,10]．最近，コンカレント・テスト技術も提案されている[11]．しかし，大型汎用テスタのコスト削減には限界がある．特に，大規模化している SoC では，デジタル機能とアナログ機能の両方をテストする必要があるので，それに合わせたコスト削減手法を確立する必要がある．

被テスト・デバイス DUT にテスト容易化設計(DFT: Design For Testability)を適用することはテスト・コスト削減の有効な手法であり，DFT を適用する試みが盛んである[12-16]．しかし，多くの場合，DFT 回路が入った DUT をテストするには高価なテスタを使わざるを得ない．このため，テスト・コスト削減を実現するには，DUT に対して DFT 技術を適用するだけでは必ずしも十分ではない．DFT が内蔵された DUT に対して，DFT に対応した低コスト・テスタが適用されればテスト・コスト全体の削減が可能と考えられる[17-19]．

最近，DFT に対応した DFT 専用テスタが提案されている[20,21]．これらのテスタは低価格で入手できるが，AC スキャンやその他の問題を抱えている[22]．また，DFT 専用テスタは DFT に特化したテスタであるため，DFT 専用テスタだけでは，特にミックスト信号デバイスのテストを実行できない．汎用テスタと組み合わせた 2 パス・テストとなることも多い．このように，DFT 専用テスタを用いても，テスト・コストをいつも削減できるわけではない．一つの解決策として，書き換え可能な FPGA ベースのテスタが，次世代のテスタとして期待されている[23]．これは，テスト実行中にテスタ・リソースを置換え可能だからである．テスト・プログラムには，その製品が必要としているテスタ・リソースのみが，テスト項目毎に記載されているので，テスト項目毎に再構成させることで 1 個の FPGA でテスタが構成できる．

そこで我々は，従来の大型汎用テスタに対し十分な低価格化を実現することを目的として，FPGA を使った再構成基板型テスタを開発した．開発したテスタはテスト・プログ

ラムから導かれた高位記述(HDL)によって構成される. それは必要なテスト機能を逐次的かつ動的に再構成する．DUT の設計実機検証も可能であり，設計の開発期間が短縮できる．

以下，この再構成可能テスタの基本的な考え方とアーキテクチャを述べ，2 種類の再構成可能テスタの開発について述べる．さらに，HDD モータ駆動コンボ IC への応用とグラフィカル・ユーザ・インタフェース(GUI)について報告する．

3.2　再構成可能テスタの基本概念

3.2.1　基本的な考え方

メモリに対するテスト･パターンは, アルゴリズミック･メモリ･パターン発生器(ALPG: Algorithmic memory Pattern Generator)によって発生される[24]．ALPG は，通常のメモリ・テスタではおよそ 500K ゲート規模で構成される．しかし，ALPG の機能をテスト・プログラムから抽出し，HDL で記述して論理合成すると，およそ 5K ゲートで実現できる[25,26]．すなわち，被テスト・デバイス DUT が実際に必要としている ALPG の機能だけを抽出し，それらをハードウェアで構成すれば，大幅なコスト削減につながる可能性がある．

メモリ・テストのためのテスタ回路 ALPG は 5K ゲート規模なので，1 個の FPGA に実現することが可能である．FPGA は短時間で再構成できるので，1 テスト単位毎に FPGA チップ上へ必要なテスト機能だけを動的に再構成すれば, 1 個の FPGA を用いてテスタの基本的なハードウェアが実現可能である．

ロジック・テストの場合は，テスト・パターンが長大なパターンとなり，パターン発生器メモリの大容量化を必要とする．しかし，それをコントロールする制御部分は HDL 化できる．このために，FPGA と外部メモリの構成でハードウェアが実現可能である．

3.2.2 再構成可能テスタ･アーキテクチャ

図 3.1 は低消費電力基板型再構成可能テスタの基本構造を示している. 設計した基板上には 1 個の FPGA と複数個のアナログ/デジタル変換器(ADC: Analog to Digital Converter), 複数個のデジタル/アナログ変換器(DAC: Digital to Analog Converter), 複数個のオペアンプと外部メモリから構成される. フォース・ライン(F), センス・ライン(S)とオペアンプは直流精度を保証するために使われている. DUT に対する電圧および電流はフォース・ラインを通して供給され, センス・ラインにより電圧および電流が帰還される. このため, 高精度な直流精度が維持できる[27].

デバイスへのテスト入力信号は FPGA のピンから伝えられ, 同じくそのテスト出力信号も FPGA のピンで受けるようにした. それらのデバイス入出力ピンの直流テストは, スイッチ･アレイを通して DAC, ADC およびのフォース・ライン(F), センス・ライン(S)に接続し, 切替えながらテストできるようにしている. さらに, 必要に応じてピン・エレクトロニクス(DR/COMP)を用いた.

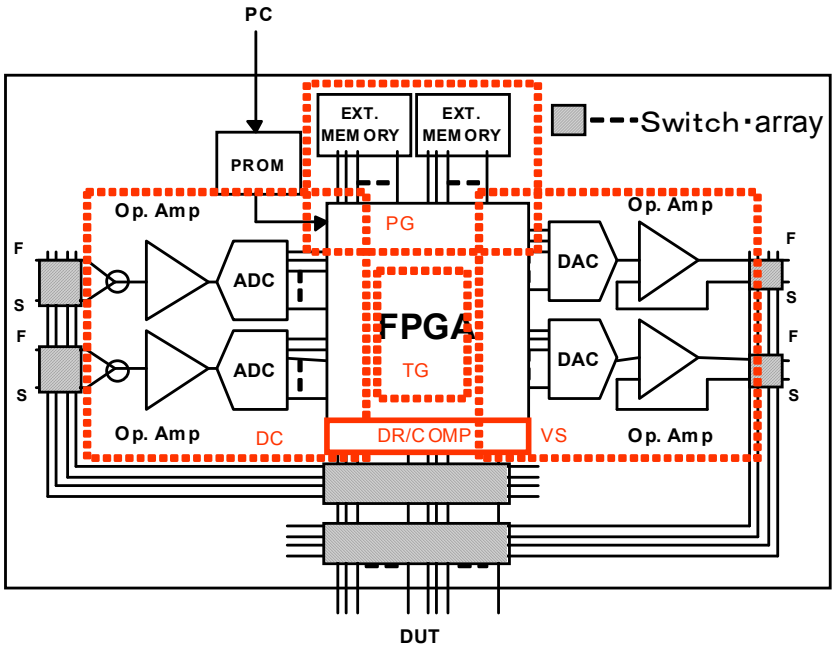


図 3.1　再構成可能テスタの基本構造

Fig. 3.1　Basic architecture for reconfigurable tester

図 2.1 の大型汎用テスタの基本構造に対応してパターン発生器(PG), タイミング発生器(TG)を FPGA 内に再構成できる．また，デジタル・プログラマブル電源(DPS)，直流計測器(DC)の制御系も FPGA 内に再構成できる．

外部のメモリ・モジュールで構成された外部メモリは，メモリ・テスト時にフェイル・ビットのマッピング・データの記憶に使われる．このメモリはロジック・テストのためのパターン・メモリとしても使うことができる．

様々なテスト機能に対応するための FPGA の構成データは，ボード上のプログラム可能な ROM(PROM)に格納される．このデータは外部パーソナル・コンピュータ(PC)から送信され，書き込むデータによって再構成され，製品毎の個別切替えを行う．

3.3　低消費電力基板型テスタの設計

以下に，我々が開発した 2 個の基板型テスタ TOB-I，および TOB-II の特徴を述べる．TOB-I はフラッシュ・メモリのテスト・コスト削減のために開発された．TOB-II はミックスト信号デバイスのテストに使われている．民生用半導体デバイスが 50MHz 以下であることから，これを目標テスト周波数とした．この値は，現在の FPGA の最大動作周波数(200MHz)で十分実現可能である．後述する TOB-II ではピン・エレクトロニクスとして安価なデバイスを用いることができた．

3.3.1　TOB-I の開発

TOB- I はフラッシュ・メモリのテストのために開発された．これは 15cm×15cm の大きさの基板に実装されている．図 3.1 の基本構造に対応して，1 個の FPGA と 2 個の DAC，2 個の ADC，4 個のオペアンプおよび 512K ビットの外部メモリ 2 個で構成されている．この基板には 96 ピンが搭載されており，このうち 2 ピンは高精度電源として使用できる．残りのピンは入力，出力及び入出力として使用可能である．使用した FPGA チップはザイリンクス社製 Vertex-II(1M ゲート規模)である．FPGA の構成を変えることによって，さまざまなデバイスがテストできる．最大テスト周波数は 50MHz である．図 3.2 は基板

型再構成可能な TOB-I の写真である．ピン・エレクトロニクスは搭載せず，FPGA の入出力をそのまま使った．

テスト実行には，PROM のデータをアクセスしながら，FPGA の構成データを FPGA に書き込ませて，テスト毎に実行させている．この書込み時間は 100ms 程度であった．著者らは，以上の構造を TOB(Tester On Board)と呼称してその開発を行った．

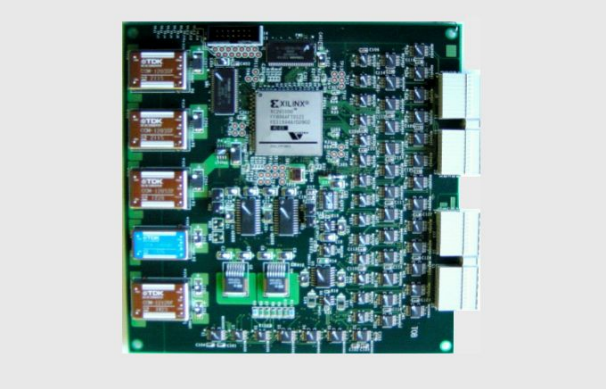


図 3.2　構造可変テスタ TOB- I の写真

Fig. 3.2　Photograph of reconfigurable TOB-I

例えば，TOB-I を 32Mb NOR フラッシュ・メモリのテストに使用すると，2 個のフラッシュ・メモリ・モジュールの同時測定が可能である．その時の TOB-I の FPGA 構成を図 3.3 に示す．この FPGA の構成は約 2000 行の Verilog-HDL から生成された．図中の ALPG はメモリに対するパターンを生成するブロックである．ピン・データ・セレクタ(PDS: Pin Data Selector)は各々のピンに対して ALPG から出力されるテスト・データを割り当てする．テスト・データは，アドレス情報や，書込み制御などである．波形フォーマット・

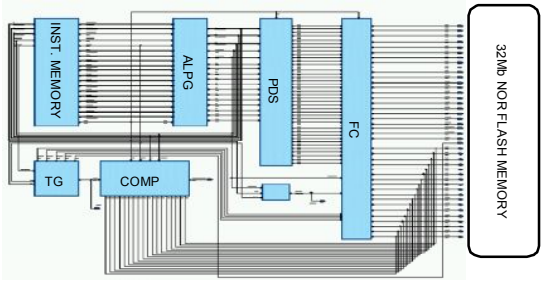


図 3.3　TOB- I におけるフラッシュ・メモリ・テストの FPGA 構成

Fig. 3.3　FPGA configuration for testing flash memory with TOB-I

コントロール(FC: Format Control)は波形モード，すなわち NRZ(Non Return to Zero)，RZ(Return to Zero)，RO(Return to One)等指定された波形を生成する．例えば，アドレスは NRZ 波形であり，書込み信号は RO 波形が出力されるように構成される．デジタル・コンパレータ(COMP)は，DUT の出力データを期待値と比較する．メモリ動作をモニタし良否判定して，フラッシュ・メモリの動作をテストしている．タイミング発生器(TG)はアドレスや書込み制御で必要としているタイミング，および期待値判定のタイミングを発生する．ALPG のプログラムはインストラクション・メモリ(INST.MEMORY)に格納される．ALPG は 5K ゲートであり，テスト回路全体ではおおよそ 10K ゲートであった．

図 3.4 にフラッシュ・メモリ・テストの時の書込み動作に対する波形を示す．入力電圧は FPGA の出力電圧と同じハイ・レベル 3.3V，ロー・レベル 0V である．図には TOB-I 上の DUT ピンで観測された 4 個の波形が示されている．最上段はライト・イネーブル信号であり，ロー・レベルで書込み状態を示しハイ・レベルで読出し状態を示す．2 番目は書込みデータの波形である． 3 番目は出力イネーブル信号であり，ロー・レベルで出力がイネーブルされる．最下段は，最下位アドレス・ピンの信号を観測したものである．STEP1 でステータス・モード・クリアと呼ばれる動作を実行している．STEP2 では，ワード書込みモードを指定しており，STEP3 では先頭番地からデータ 0, 1, 0, 1・・・の書込みを行っている．

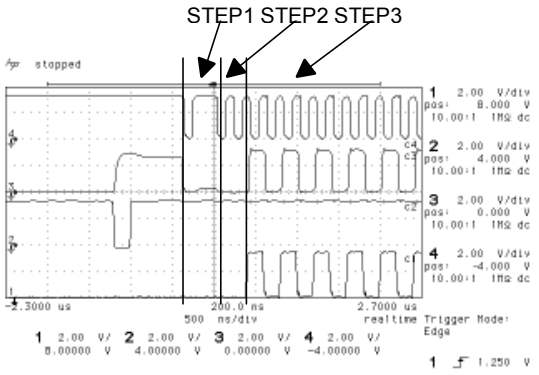


図 3.4　TOB－Ｉでの 32M フラュシュ・メモリ測定の書込み動作波形

Fig. 3.4　WRITE operation waveforms for testing 32-Mb NOR flash memory with TOB-I

3.3.2　TOB-II の開発

上記, TOB-I の開発の経緯から, 以下 5 項目を改良することにした．これらに従い, 我々は TOB-II を開発した．特に，ある 1 個のテスタ・リソースにおいて，テスト・プログラムから構成するテスタ回路が比較的小規模であることに注目した．つまり，FPGA の搭載可能ゲート規模の数パーセントしか使われない．現在の大規模 FPGA では，テスト・プログラムの全部またはかなりの部分に相当するテスタ・リソースを搭載して，条件設定で動作するようにできる．

(1)　ピン数の増加

大規模 VLSI が測定可能なようにピン数を増加させ，拡張を可能にした．

(2)　ピン・エレクトロニクス(P/E)の搭載

高精度な入力・出力レベル・テストができるように，汎用テスタと同じくピン・エレクトロニクスを搭載した．

(3)　パターン・メモリ容量の増加

大規模 VLSI の長大テスト・パターンを搭載にするために，パターン・メモリ容量の増加を図った．

(4)　直流計測精度の向上

DUT の消費電力の大電流やスタンバイ電流を測定するために，汎用テスタと同程度の精度を持つ高精度直流計測器 PMU(Precision Measurement Unit)を搭載した．また，DC テスト時間を削減するために複数の直流計測器を搭載した．

(5)　テスト条件を格納する RAM の追加

上述したように，現在の FPGA は大規模なので，テスタ・リソースをできる限り搭載した．そして，テスト条件を格納する RAM を搭載しテスタ・リソースの切替えを制御できるようにした．

TOB-II の構成は基本的に図 3.1 と同じ構造であるが上記の項目に従い再設計した．まず，項目(1)に対しては， TOB-II では 128 ピンを実装した．128 ピン以上が必要なときには，複数の TOB-II 基板を用いて同期させピン数の拡張ができる構成とした．これに対して，TOB-I では 96 ピン対応であった．

次に，項目(2)に関して， TOB-II では DUT の入力レベルをテストする必要があったので，高精度ピン・エレクトロニクスの搭載を図った．TOB-I では FPGA の出力の直接接続であった．

項目(3)に対しては，TOB-II は DIMM(Dual Inline Memory Module)タイプの DRAM モジュールを実装し，テスト毎に 128 ピン×1M ステップのテスト・パターンを格納できるようにした．6 通りのパターン・モード 0, 1, H, L, X, Z が割当て可能であり，1 ピン当り 3 ビットを割り当て 128M×3 ビットとした．

項目(4)に対しては，直流計測器の精度向上のために，高精度直流計測器 PMU を搭載した． 16 個の直流計測器を持った付加ボードを TOB-II 上に搭載して DC テストのスループット向上ができるように改良した. TOB-I では 1 個の直流計測器が実装されていた．項目(5)に対しては，再構成を行うことなく，テスタ仕様に基づきテスタ・リソースを搭載して動作できるようにした. そのテスト･パラメータを格納するための RAM を TOB-II 基板上に追加した．すなわち，再構成のための処理時間をなくし，テスト時間が削減できるようになった．TOB-I と比較して，TOB-II ではおのおののテスト機能を再構成するためのテスト毎に必要な時間が削減でき，測定時間短縮に寄与できた．テスト周波数は TOB-I,TOB-II ともに 50MHz である．

図 3.5 は TOB-II の写真である．28cm×30cm の基板に実装した．

図 3.5　再構成可能テスタ TOB－Ⅱの写真

Fig. 3.5　Photograph of reconfigurable TOB-II

テスト条件や FPGA 再構成データは外部 PC から USB インタフェースを通して転送される． USB インタフェースの制御にマイクロ・コントローラ SH-3 を搭載して，この転送のためのデータ処理が実行できるようにした．

図 3.6 に，標準ロジック・デバイス HD74LS74(エッジトリガ型フリップフロップ)を TOB-II で測定したときの波形を示す．観測点は DUT のピンである．FPGA につながれているピン･エレクトロニクスによりハイ・レベルとして 2.4V を，ロー・レベルとして 0.8V を制御している．

図中にはこのテストに対する 6 個の波形を示している．6 個の波形は最上段から反転出力 Q_，出力 Q，プリセット PRE，クロック CLK，入力 D とクリア CLR である．STEP1 では PRE と CLR がネゲートされている．STEP2 では CLR 信号がロー・レベルとなりクリア動作をさせている．STEP3 では，PRE 信号がロー・レベルとなりプリセットが実行され，STEP4 ではデータ 1 を保持データ保持する．STEP5 でデータ 0 を保持する．STEP6 で CLR 信号によりクリア動作を再度実行している．

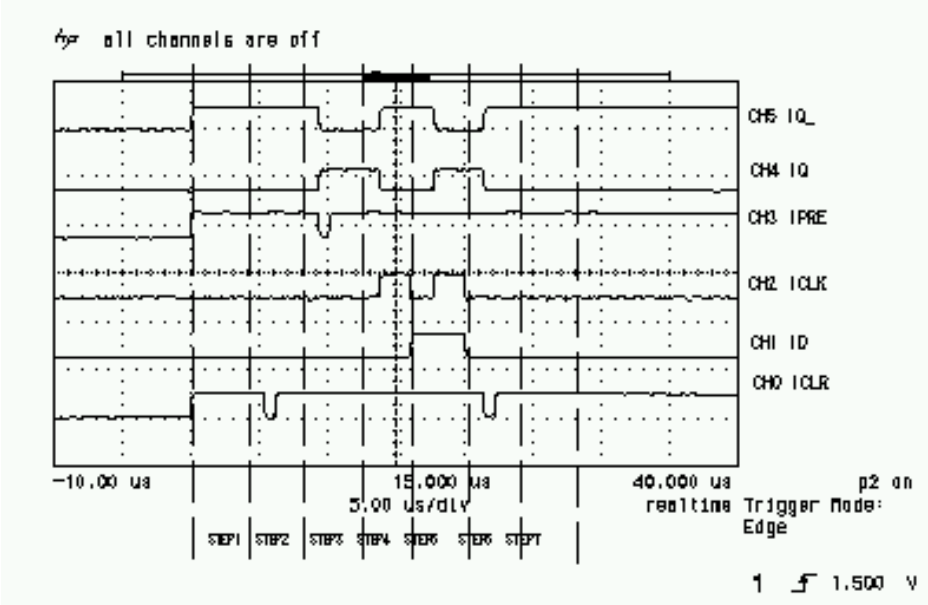


図 3.6　TOB—Ⅱを用いた HD74LS74 の P/E 観測波形

Fig. 3.6　Measured P/E waveforms for HD74LS74 with TOB-II

テスタの消費電力について述べる. 通常の大型汎用テスタは 15KW～30KW の電力を必要とするが，TOB-II の消費電力は 200W であり 1/100 以下であった．TOB-I の消費電力は 1W である．TOB-II が TOB-I と比較して 200 倍の消費電力となった理由は，ピン・エレクトロニクスを搭載し，1 ピン当りの消費電力が大きくなったからである.

3.3.3 TOB 開発のまとめとテスタ比較

開発した TOB-I，TOB-II と従来テスタ((株)アドバンテスト T3347)との比較を表 3.1 に示す．TOB-I，TOB-II は再構成可能テスタであり，96 ピン，128 ピンを有し 50MHz で動作する．それぞれ 15cm×15cm および 28cm×30cm の基板上に構成した．また，消費電力はそれぞれ 1W，200W である．T3347 は広く使用されている典型的な汎用テスタでありシェアード･リソース･テスタである．動作周波数は 40MHz であり，256 ピンを有し，大きさとしては，縦 95cm×横 164cm×高さ 180cm の装置である．消費電力は最小構成で 14700W である.

表 3.1 提案テスタと典型的従来テスタの比較

Table 3.1　Comparison of proposed tester and a typical tester

	本提案テスタ		従来テスタ
機種名	TOB－I	TOB－Ⅱ	T3347 注) ((株)アドバンテスト)
ﾃｽﾀ･ｱｰｷﾃｸﾁｬ	再構成可能テスタ		資源共有型テスタ (シェアード･リソース型テスタ
ﾃｽﾄ周波数	50MHz	50MHz	40MHz
ﾃｽﾄ･ﾋﾟﾝ数	96 ﾋﾟﾝ	128ﾋﾟﾝ	256ﾋﾟﾝ
サイズ	縦15 cm ×横 15cm	縦28cm×横30cm	縦95cm×横164cm 高さ180cm
消費電力	1W	200W	14700W

注)T3347A VLSIテスト･システム一般仕様書初版1995年2月15日

再構成可能テスタ TOB-I および TOB-II は，FPGA 1 個とその他のデバイスで構成されるので部品点数が少なく簡単な構造である．FPGA への論理構成技術を理解でき，アナロ

グ回路に関する技術を有し，テスタの基本構造を理解した技術者であれば，構成可能なものである．

しかし，各信号のタイミング発生については，FPGA の最大動作周波数に制限される．このため，FPGA への論理回路配置の調整や FPGA の I/O バッファでの取扱いによって調整した．この対策として，テスタ技術で活用されているタイミング・キャリブレーション技術の適用が必要である．

3.4　TOB-II の HDD モータ駆動コンボ IC への応用

TOB-II の応用として，HDD モータ駆動コンボ IC への設計機能評価に適用した．この例を以下に報告する．

3.4.1　HDD モータ駆動コンボ IC 概要と設計要求

対象とした被測定デバイス DUT は，ルネサステクノロジ社製の HDD モータ駆動コンボ IC である．このデバイスは 0.35μmBiCMOS プロセスで製造され，64 ピンのパッケージに実装されている．また，HDD のスピンドル・モータを駆動するための低ノイズドライバが集積化されている．また，このデバイスは HDD のアクチュエータを駆動するボイス・コイル・モータ(VCM: Voice Coil Motor)を制御するので，高精度の DAC を有している．

このようなデバイスの設計検証には，アナログ・テスタを用いることが望ましい．しかし，この設計検証のためのアナログ機能を含むテスタの立上げには，多大な労力を必要としている．このため，アナログ・テスタを用いず，試作チップを用いた設計検証することが一般的である．試作においては，試作チップとともにパルス発生器やデータ発生器を搭載した実機セットが用いられてきた．

TOB-II を使うことにより，開発段階で設計検証の期間を削減することは，開発期間の短縮に寄与する．TOB-II は再構成可能であるので，上述の設計要求に応えることができる．

3.4.2　実機評価環境

図 3.7 は TOB-II を用いたプロトタイプの筐体を示す写真である． HDD モータ駆動コンボ IC はソケットに差し込まれ，ユーザのボードを通して TOB-II に接続されている．

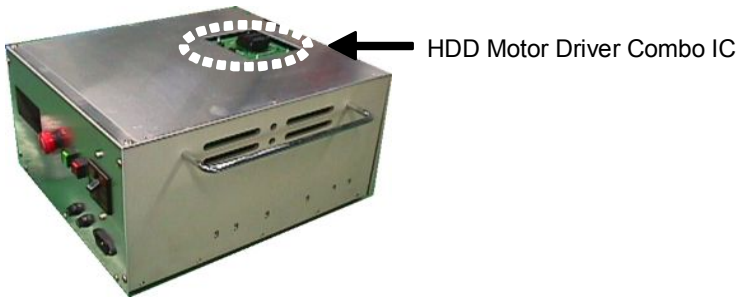


図 3.7 HDD 駆動コンボ IC プロトタイプ評価器
Fig. 3.7　HDD Motor Driver Combo IC prototype

図 3.8 に TOB-II 上の FPGA におけるテスタ回路の構成を示す．その基本機能は被測定デバイス DUT へのシリアル・コマンド出力，DUT からの出力波形の捕捉および時間を測定する機能の回路である．制御レジスタ(control register)はテスト条件のパラメータを格納して，それぞれのテスト回路を制御する．このデータはマイコン・コントローラ SuperH-3 から USB を通して接続された外部 PC に書き込まれる．図中のコントロール・コマンド(Control command to seiarl)は HDD モータ駆動コンボ IC をコントロールするシリアル・コマンドを生成する部分である．インターバル計測(Interval measurement)は各信号間の時間間隔測定をする部分である．A チャンネルと B チャンネルの時間を測定して判定するが，各々8 チャンネルをもってマトリックスで切り替えている．オーディオ・コントローラ(Audio controller)は，アナログ信号を生成し，スピンドル・モータやボイス・コイル・モータ(VCM)の周波数特性試験のアナログ信号を生成する．クロック発生器(CLK generator)はシリアル・コマンドの転送クロックを発生する．パターン発生器(Pattern generator)は機能テストをするためのパターンやタイミングを発生している．外部リレー・コントロール(External relay control)は，テスト基板上にあるリレーを制御して，必要

なテスト回路系を構成するリレーをコントロールする．電源コントロール(Power supply control)は，デバイスの電源発生を制御し，４電源を制御している．高精度 DC 計測コントロール(Precision Measurement Unit Control)は付加ボードの高精度直流計測器 PMU を制御して DC テストを実施している．負荷抵抗制御コントロール(Load Resistor control)はデバイスの低抵抗負荷の切り替えを行う.これらはテスト･プログラムにより制御され HDD モータ駆動コンボ IC のテストを実行する.

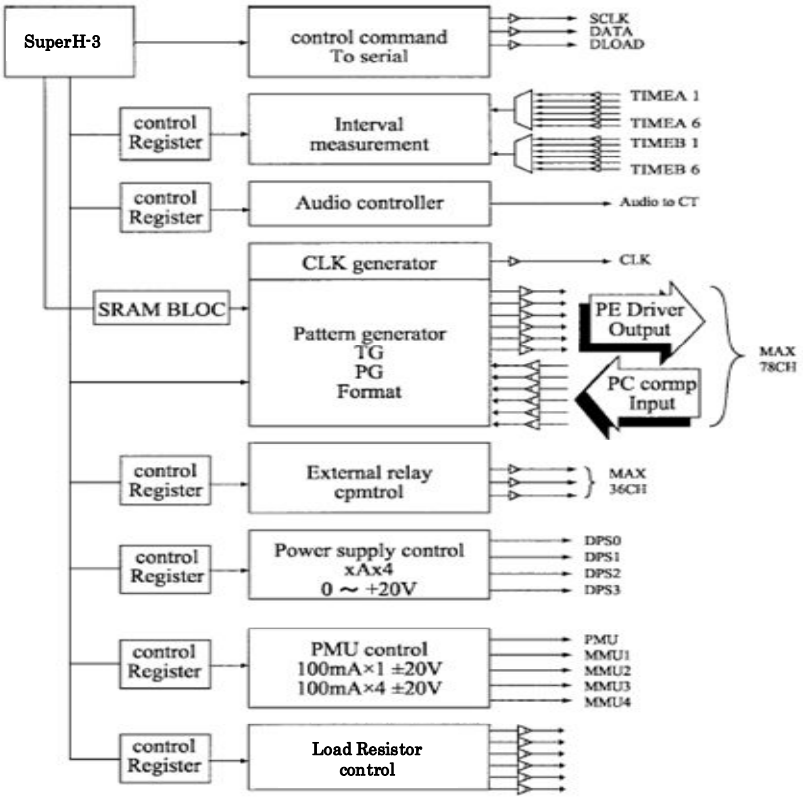


図 3.8 HDD 駆動コンボ IC テストの FPGA 構成

Fig. 3.8　FPGA configuration to test HDD Motor Driver Combo IC

TOB-II のテスト条件設定は，設計者が条件を設定出来るように GUI で操作されることが望ましい．GUI 環境を備えることにより，テスト・エンジニアだけでなく，システム設計者もテスト条件が簡単に設定できる．図 3.9 は GUI 操作環境の例である．デバイス電源(DPS)設定とそれに対応した Excel シートを用いた設定例を示す．GUI を使って，電

源電圧値，電流クランプ値，電流測定範囲を設定する．また，Excel シートを使ってテスト条件を確認し更新できる．GUI を使って電源条件や周波数条件を変えたテスト・データの記述ができる[28].

図 3.10 はデバッグ画面の例である．プログラムのロード(program load)やテスト実行(executed test)が指定できる．さらにデバッグ・モード(debug mode)，データ・ログ(logging)の選択を行う．測定データ(log viewer)と共に PASS/FAIL 情報(pass/fail indicator)も表示可能である．さらに，DUT の動作範囲を示すシュム図などの操作環境も準備した．これらのシステムは Visual C++で開発した．このシステムは TOB-II に接続された外部 PC 上に構築した.

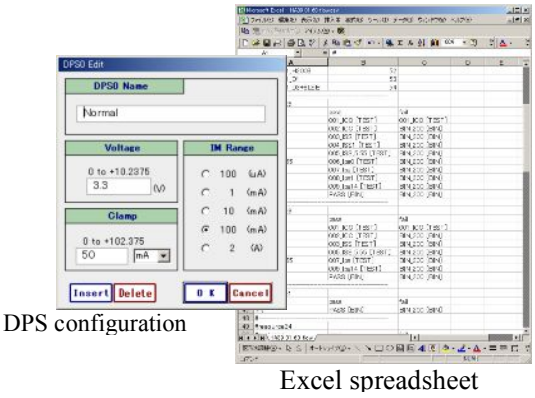


図 3.9　TOB-II の GUI (DPS 設定例)

Fig. 3.9　GUI for TOB-II. (Example for DPS setup)

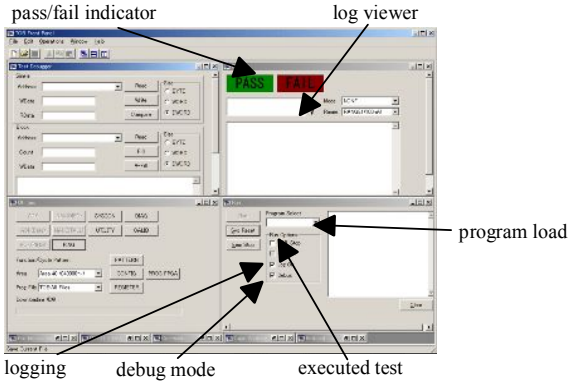


図 3.10　TOB-II のデバック画面

Fig. 3.10　Debug screen for TOB-II

3.4.3　実機評価結果

HDD モータ駆動コンボ IC は，HDD の制御 CPU からシリアル・コマンドを受けて，その動作を変更する機能を持つ．図 3.11 に HDD モータ駆動コンボ IC 実機評価で観測されたシリアル・コマンド転送時の波形を示す．4 個の波形を示している．最上段はモード・セレクトと呼ばれている信号 DLOAD の波形を示す．この信号がロー・レベルのときシリアル・コマンドが転送される．2 番目はシリアル・コマンドである．シリアル・コマンドは 1 ビットの ID コードと 5 ビットの HDD モータ駆動 IC 内のハードウェア･アドレス，1 ビットのモード・ビット(write/read)，16 ビットのデータ・コードで構成されている．このシリアル・コマンドで HDD モータ駆動コンボ IC の動作を決定している．3 番目はコマンド・データの転送クロック信号である．このクロック波形の立上りでシリアル・コマンドを受けている．この場合は被測定デバイスがコマンド 0100010・・・・を受け，内部レジスタに設定している．所定のデバイスのモードおよびデータが設定される．最下段は複数ボードを使用実装した時のボードの選択識別信号で，ハイ・レベルの時が選択された状態を示している．

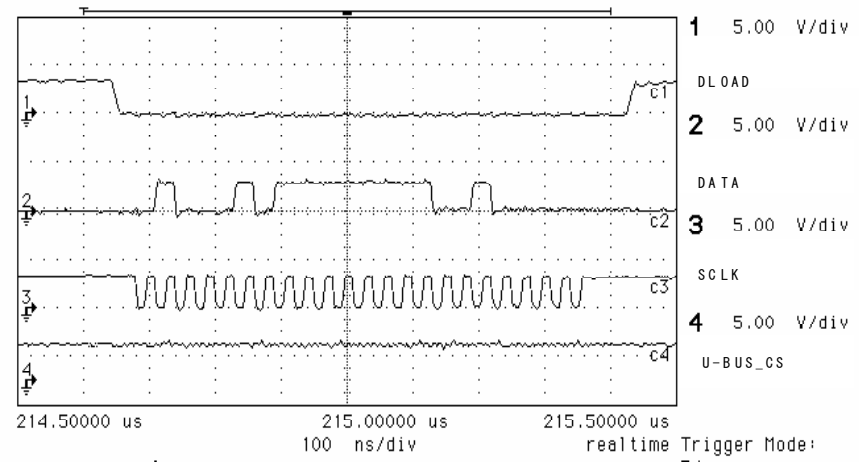
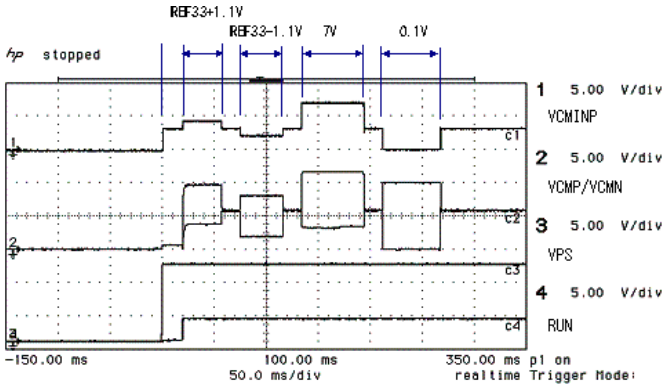


図 3.11　デバイス・モード設定のコマンド・データ波形

Fig. 3.11　Waveforms for serialized data to control device mode

図 3.12 は，図 3.11 の TOB-II から出力したコマンドを受け，HDD モータ駆動コンボ IC の VCM モード･セレクトにおいて DUT の波形をオシロスコープで観測した一例である．4 個の波形を示している．最上段の波形は VCM ドライバを制御する VCMINP への入力波形である．



STEP1 STEP2 STEP3 STEP4

図 3.12　VCM 制御信号測定波形

Fig. 3.12　Measured waveforms for VCM control signal

HDD モータ駆動コンボ IC は高精度 DAC の出力が DACOUT として出力される．この例では DACOUT が直接 VCMINP に接続され，VCM ドライバを制御している．２段目はその VCM ドライバからの出力であり，VCMP/VCMN はそれぞれの正相/逆相の信号である．VCMP と VCMN は HDD のボイス・コイルの端子につなげられ，この端子間に流れる電流値でモータを駆動し，HDD のアクチェータの移動を制御している．STEP1 では基準電圧(REF)3.3V に対して 1.1V 高い制御をしており，STEP2 では 1.1V 低い制御をしている．STEP3 と STEP4 では 7V と 0.1V の制御をしている．３段目はデバイスに供給する 12V 電源を示す．4 段目の信号におけるハイ・レベルは HDD モータ駆動コンボ IC がアクティブであることを示している．このように，設計者およびテスト・エンジニアが TOB-II を使ってコマンド転送とその動作波形を観測することができ，デバイスの評価に寄与できた．

また，HDD モータ駆動コンボ IC の DC テスト精度評価について，その一部を表 3.2 に示す．この表では，18 項目の DC テスト測定結果と，同じ項目を商用リニア・テスタで測定した結果を示す．この結果，TOB-II で十分なレベルの精度を実現して所定の DC テスト精度を得た．

表 3.2 HDD Motor Driver Combo IC DC 評価結果

Table 3.2　Evaluation results of DC characteristics for HDD Motor Driver Combo IC

Test #	NAME	DEVICE 1	
		リニア・テスタ	TOB－Ⅱ
1	ICC Ips+Iss	34.14mA	33.14mA
2	Iss0	3.54mA	3.47mA
3	Iss	5.27mA	5.03mA
4	Iss1	5.50mA	5.35mA
5	Iss(5.55V)	5.62mA	5.30mA
6	Ips0	9.07mA	9.12mA
7	Ips(10V)	27.27mA	26.68mA
8	Ips(12V)	28.64mA	27.79mA
9	Ips(14V)	29.22mA	29.30mA
230	U CUTOFF	4.70V	4.7022V
231	V CUTOFF	4.70V	4.7109V
232	W CUTOFF	4.79V	4.8016V
234	PHASE VOL	0.172V	0.1619V
235	PHASE VOH	3.15V	3.1225V
461	VBEMF AC	6.00V	5.9878V
461.1	VBEMF Vol	1.844V	1.8809V
462	VBEMF Vin 1KHz	51.9mV	52.5mV
465	VBEMF Vout 1KHz	388.9mV	355.9mV

ここでは，TOB-II を試作設計検証へ応用したが，設計検証は量産テストに引き継がれる．実機を接続しての実機評価は，特にアナログ・デバイスではしばしば使われる手法である．本稿では，再構成可能である利点を使って，TOB-II を設計検証に使う可能性について試作評価した．試作評価については上記説明したデバイス・モード設定のコマンド波形観測および VCM 制御信号測定波形をはじめとして，設計検証項目について継続検討中である．このアプローチはますます困難になっているミックスト信号デバイスの設計段階での設計期間短縮につながる．今後，再構成可能テスタはますます多機能化，高精度化されるミックスト信号デバイスの開発期間短縮に寄与すると考える．さらに，再構成可能テスタはメモリ・テスタにもロジック・テスタ，アナログ・テスタにも成りえる特徴を持ち量産でも使うことができる．量産でのテスト・コストを削減するものと期待できる．

3.5　まとめ

本章では，FPGA を使った再構成可能基板型テスタである TOB を提案し開発した．TOB-I はフラッシュ・メモリをテストするために使用され，TOB-II はミックスト信号デバイスをテストするために使用された．本章で述べたテスタは大型汎用テスタのテスタ・リソースを置き換えるべく，再構成可能な FPGA を使って開発したものである．TOB はテスト・プログラムから導出された高位記述 HDL 記述を使い構成されている．TOB-I は 15cm×15cm の基板に実装され，96 ピンのテスト・ピンを有し，50MHz でのメモリ・テストが実施できる．TOB-II は，28cm×30cm の基板に実装され，128 ピンのテスト・ピンを持つ．TOB-II では複数基板での同期を考慮して，ピン拡張ができるようにした．テスト周波数は TOB-I と同じく 50MHz である．汎用テスタと比べて，安価なもので，チップ・テスト全体のコスト削減につながる．また，このテスタは低消費電力であり，TOB-II の消費電力は 200W であり 1/100 以下であった．TOB-I の消費電力は 1W であった．さらに，ここで示したテスタは再構成可能であることから，メモリ・テスタにもロジック・テスタ，アナログ・テスタにも成りえる特徴を持つ．今回は，設計段階の製品評価に応用できた．その例を HDD モータ駆動コンボ IC への応用で示した．

参考文献

[1] Semiconductor Industry Association, *International Roadmap for Semiconductors*, 2002 Update, 2002.

[2] A. L. Crouch, *Design-for-Test for Digital IC's and Embedded Core Systems*, Prentice Hall, 1999.

[3] 佐藤正幸，“第 14 章　テスティング技術，”1999 半導体テクノロジー大全，電子ジャーナル(株)，pp. 231-237，1998 年 10 月．

[4] T. Kazamaki, H. Maruyama, and S. Sumida, “Trial Model of 100 MHz Test Station for High Speed LSI Test System,” Proc. International Test Conference, pp. 611-617, 1978.

[5] T. Kazamaki, “A 100MHz Tester – Challenge to New Horizon of Testing High Speed LSI,” Proc. International Test Conference, pp. 618-625, 1979.

[6] S. Bisset, “The Development of a Tester-per-Pin VLSI Test System Architecture,” Proc. International Test Conference, pp. 151-157, 1983.

[7] Verigy,http://www.verigy.com/ate/products/V93000/index.htm

[8] J. Katz and R. Rajsuman, “A New Paradigm in Test for The Next Millennium,” Proc. International Test Conference, pp. 468-476, Oct. 2000.

[9] H. Hashempour, F. J. Meyer, F. Lombardi, and F. Karimi, “Hybrid Multisite Testing at Manufacturing,” Proc. International Test Conference, pp. 927-936, Oct. 2002.

[10] A. C. Evans, “Application of Semiconductor Test Economics, and Multisite Testing to Lower Cost of Test,” Proc. International Test Conference, pp. 113-123, Sep. 1999.

[11] J. Rivoir, “Lowering Cost of Test: Parallel Test or Low-Cost ATE?,” Proc. Asian Test Symposium, pp. 360-363, Nov. 2003.

[12] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*, Kluwer Academic Publishers, 2000.

[13] B. Nadeau-Dostie, ed., *Design for At-Speed Test, Diagnosis and Measurement,* Kluwer Academic Publishers, 1999.

[14] B. Koenemann, C. Barrnhart, B. Keller, T. Snethen, O. Farnsworth, and D. Wheater, “A SmartBIST Variant With Guaranteed Encoding,” Proc. Asian Test Symposium, pp. 325-330, Nov. 2001.

[15] J. Rajski, J. Tyszer, M. Kassb, N. Mukherjee, R. Thompson, K. H. Tsai, A. Hertwig, N. Tamarapall, G. Mrugalski, G. Eidel, and J. Qian, “Embedded Deterministic Test for Low Cost Manufacturing Test,” Proc. International Test Conference, pp. 301-310, Sep. 2002.

[16] M. Tripp, T. M. Mak, and A. Meixner, “Elimination of Traditional Function Testing of Interface Timings at Intel,” Proc. International Test Conference, pp. 1014-1022, Sep. 2003.

[17] R. Raina, C. Pyron, R. Molyneaux, and N. Tendolkar, “At-Speed Testing of Delay Faults for Motorola's MPC7400, a PowerPC Microprocessor,” Proc. VLSI Test Symposium, pp. 3-8, Apr. 2000.

[18] J. Bedsole, R. Raina, A. Crouch, and M. S. Abadir, “Very Low Cost Testers: Opportunities and Challenges, ” IEEE Design & Test of Computers, Vol. 18, No. 5, pp. 60-69, Sep.-Oct.

2001.

[19] R. Kapur, R. Chandramoull, and T. W. Willams, “Strategies for Low-Cost Test,” IEEE Design & Test of Computers, Vol. 18, No. 6, pp. 47-54, Nov.-Dec. 2001.

[20] Inovys,http://grouper.ieee.org/groups/1450/dot0/ITC2002/ITC-2002-Inovys.ppt

[21] teseda,http://www.teseda.com/prod_v520.shtml

[22] K. Posse and G. Eide, “Key Impediments to DFT-Focused Test and How to Overcome Them,” Proc. International Test Conference, pp. 503-511, Sep. 2003.

[23] D. Edenfeld, A. B. Kahng, M. Rodgers, and Y. Zorian, “2003 Technology Roadmap for Semiconductors,” IEEE Computer, Vol. 37, No. 1, pp. 47-56, Jan. 2004.

[24] A. J. Van de Goor，*Testing Semiconductor Memories*，John Wiley & Sons，1991.

[25] 佐藤正幸，“メモリ仮想テスタ技術と今後の展開,” 信学技報，フォールトトレラントシステム研究会, FTS98-121，pp．33-39，Feb. 1999.

[26] M. Sato, “Memory Virtual Tester Technology and a Tester on Chip,” SEMI Technology Symposium, pp. 5:70-5:74, Dec. 2000.

[27] 半導体製造装置標準用語集, (社)日本半導体製造装置協会, 2002 年 12 月.

[28] 佐藤正幸，大塚信行，武藤治，新井雅之，福本聡，岩崎一彦，上原孝二，志水勲，“低消費電力基板型構造可変テスタの開発,” 信学技報，DC2003-94，pp．23- 28，Feb. 2004.

第 4 章

テスタ構造表現言語の提案とテスタ選択ツールの応用

本章の内容

4.1　はじめに

大規模化・複雑化される VLS および SoC(System on chip)では，テスト・コストの増大が問題視されている[1]．テスト手法やテスト容易化設計 DFT(Design For Testability)技術が提案され，テスタにも多くの新技術が適用されているが，テスト・コストの低減には課題が多い[2].

大規模 LSI のテストでは，従来汎用 LSI テスタが使用されている汎用テスタは，CPU の主メモリ上に記憶されたテスト・プログラムを逐次解釈して，テスタ・リソースを制御し，テストを実行する.

汎用テスタは，DUT(Device Under Test)のテスト要求に従って，その構造を進化させてきた[3]．初期のアーキテクチャは，シェアード・リソース・テスタ[4,5]，およびパーピン・テスタで構成されていた[6]．近年では，テスト・プロセッサ・パーピン・テスタ[7]やタイム・ドリブン方式テスタ[8]が提案されている．しかし,テスタ言語は機種ごとに異なっており，そのテスタ言語を駆使して製品(DUT)ごとのテスト・プログラムを作成する必要がある．また，テスタ・コストを大幅に下げる技術も開発されてコスト低減の努力がされている[9].

従って，テスタを動作させるソフトウェアであるテスト・プログラムとそのインタプリタは重要な構成要素となっている.テスタ言語形態は図 4.1 に示すようにプログラム言語形式の進化に併せて進展してきた．1970 年代は，アセンブリ型言語[10]が使われていた．その後，1980 年代には FORTRAN 型言語[11]や BASIC 型言語が使われ，現在でも多

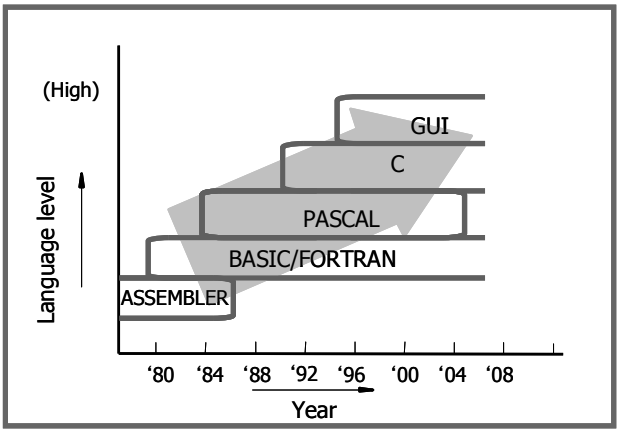


図 4.1 テスタ言語の推移

Fig. 4.1　Transition of tester language

くのテスタでこれらの言語形式が使われている．その後，PASCAL 型言語[10]が一部使われたが，主流にはならなかった．1990 年代からは，C 型言語の発展に伴い，テスタ言語も C 型言語が使われ始めた．現在における最新テスタでは C 型言語が普及している．また，テスト・プログラムの作成やデバックには，テスタごとのテスタ言語の習得や，そのデバックに特殊な技能を必要とする．これには，テスト・エンジニアのノウハウに頼っているのが現状である．

このため，テスト・プログラムのデバックは実機デバックを必要とし，コストが高いものとなっている．この改善の一手法としては仮想テスタ技術[12]も研究されて実用化されている．テスト・パターンの記述には，現在，VCD(Value Change Data)や WGL(Wave Generation Language)記述が使われている．STIL(Standard Test Interface Language)[13,14]での標準化も使われ始めた．しかし，これらのいずれもが個別のテスタ言語に変換して使われているのが現状である．このために，使用するテスタごとにテスト・プログラムを開発し LSI の試作品を待って実機デバックを行う．このように，現状のテスト・プログラミングには膨大な工数が掛かるため，LSI の短期間開発評価の要求に応えられない現状がある．

これらの事を背景に著者らは，テスト・プグラロムの作成効率向上や異種テスタへの搭載を可能にできる汎用性のあるテスタ言語の開発及びその応用を目的とし研究を行った．

4.2　テスト・プログラミングについて

4.2.1　テスタ構成

汎用テスタは，多くのテスタ・リソースを持ち CPU と各テスタ・リソースには CPU バスが構成されている．図 4.2 にテスタの構成を CPU との関係で論理的構造で示した．アドレスバスはテスタ・リソースを示すリソース・アドレスとそのテスタ・リソースが必要としているパラメータ設定のレジスタを示すサブ・アドレスからなる．データ・バ

スを通してデータは設定される．

テスタ・リソースには，以下がある．

●パターン発生器リソース PG(Pattern Generator)

テストの論理値を記憶してテスト・パターンを発生する．

●タイミング発生器リソース TG(Timing Generator)

被測定デバイス(DUT)に入力するテスト・パターンの入力タイミングや，DUT から出力される論理値を期待値と比較するタイミングを発生する．動特性試験(AC テスト)などのテストに使われる．

●デジタル・プログラマブル電源 DPS(Digital Programmable Power Supply)

被テスト・デバイス（DUT）のデバイス電源を供給する．

●DC 計測器リソース(Direct Current measurement)

DUT の静特性試験(DC テスト)を行う．

●ピン・エレクトロニクス P/E(Pin Electronics)

DUT に信号を入力するドライバと DUT から出力された信号を判定するコンパレータを有する．このドライバとコンパレータを 1 ピン毎に併せ持った構造もある．これを総称してピン・エレクトロニクスと呼ぶ．ピン・エレクトロニクスは，長距離布線で配線しても正常な信号特性が考慮されるように高周波伝送系が構成される．一般的には 50 オームの特性インピーダンスが使用されている．

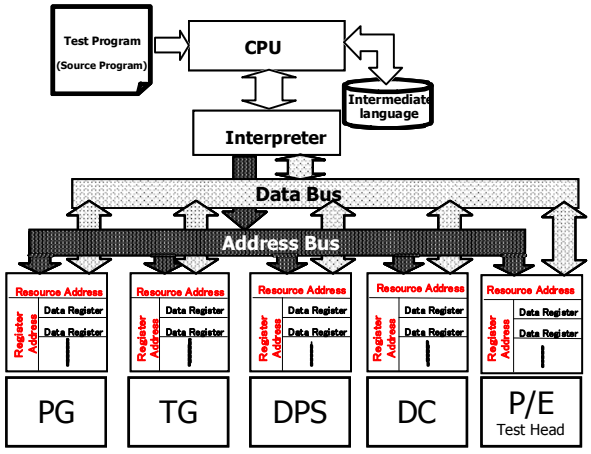


図 4.2 テスタ構成

Fig.4.2　Tester structure

テスト実行の際には各テスタ・リソースがリソース・アドレスで示され，それに必要なパラメータがサブ・アドレスで示されて，パラメータが設定される．

このため，テスタ・リソースの構造を配慮したテスト・プログラミングが必要である．

4.2.2　テスタ言語の種類

各社のテスタ言語の比較を表 4.1 に示す[15]．表 4.1 ではデバイス電源 DPS 設定とパターン記述について記載してある．

表 4.1　テスタ言語の調査

Table. 4.1　Survey of tester language

テスタ･メーカ		A社		B社		C社
テスタ機種		テスタ1	テスタ2	テスタ3	テスタ4	テスタ5
言語形態		Fortran Like	C言語 Like	BASIC LIKE	C言語 Like	C言語 Like
総プログラム行数(比率)		*2.0*	*3.8*	*1.0*	*1.8*	*1.2*

テスタ記述	デバイス電源	VS1=0.000V,R8V,M(0.8A),400MA,-400MA	DPSVSIM dpsvsim;
			dpsvsim.pin(VS1);
			dpsvsim.SRng(R8V);
			dpsvsim.SVal(0V);
			dpsvsim.MRng(M800MA);
			dpsvsim.CPVal(400mA);
			dpsvsim.CMVal(-400mA);
			dpsvsim.Load();
			tsvm.Load();
				DEFINE部: VCC=BS1(4R,4C)
				DATA部: VCC=0.0V
					uvi(VCC) {v=@vcc; vi=vfim; i=3mA
					; vrng=r8V;irng=r3mA; aiarm=off; }
						set vs1 v=0.000v,vr=10v,
						cur=400ma,ir=800ma;

	パターン記述	LPAT PFCT2 CHANNEL 1-8,10-13 CFPF NOP/T1 !1000XX1000XX NOP/T2 !0001LH0001LH ∫ STPS /T11 !1001XX1001XX END	LPAT PFCT2 CHANNEL 1-8,10-13 CFPF NOP/T1 !1000XX1000XX NOP/T2 !0001LH0001LH ∫ STPS /T11 !1001XX1001XX END	PATTERN1 XX0001XX0001 % SPL0 HL1000HL1000 % SPL0 ∫ XX1001XX1001 % HALT END	PPRO PATFUNCT MODE LCDPPCP PDSECT CHANNEL 1 2 3 4 5 6 13 12 11 10 9 8 PMSECT START:NOP /T1 ! 1001XX 1001XX NOP /T1 ! 0001LH 0001LH ∫ STOP: NOP /T1 ! 1001XX 1001XX END	PATTERN sk7474 MODE AT RDX 10 CHANNEL 1-6,8-13 CPFF MODULE PAT1 NOP ,T1 !1001XX XX1001; NOP ,T1 !0001LH HL1000; ∫ STOP,T1 !1001XX XX1001; MODULE END

標準ロジック Edge Trigger Type　F/F デバイスをサンプルとし,各テスタ言語を用いて，テスト・プログラムを作成した．総プログラム行数，プログラムの概略，テスト・パターンの記述を示す．BASIC 型言語を使うテスタ 3 は，プログラム行数（約 200 行）が一番少なかった．このテスタ 3 の総プログラム行数を *1.0* として，その他のテスタでのプログラム行数の比率を示す．FORTRAN 型言語を使ったテスタ 1 の比率は *2.0* であり，C 型言語を使ったテスタ 2 の比率は *3.8* であった．C 型言語のテスタ 4 とテスタ 5 は比率が *1.8* と *1.2* であった．これらのテスタ 4,5 では，C 型言語を使っていてもテスタ・メーカ個別のステートメント形式手法を使っている．

次に，図 4.3 の DPS のハード構成に従って FORTRAN 型言語のテスタ 1，C 型言語のテ

スタ２，C 型言語かつステートメント記述のテスタ 5 を取り上げて示す．

DPS は電圧を設定する電圧発生部 DAC(Digital to Analog Converter)，電圧発生の範囲を決める電圧レンジ設定機能(RNG)，ボルテージ・フォロア型の差動演算増幅器(AMP)，デバイスに流れる電流値を測定する電流測定機能と測定範囲を決める電流測定レンジ(A)，DPS を異常電流から保護するための電流クランプ機能(Upper Clamp ,Lower Clamp)を持つ．DAC は一般的に 12 ビットで，その発生範囲が+2.048～-2.047V である．データ・バスからレジスタにデータが送られ設定される．RNG はその電圧を増幅するためにある．一般的に，0.8V レンジ，8V レンジ，80V レンジの 3 種類を取ることが多い．AMP は DUT に必要な電流を供給する構造であり，センス・ライン(S)の電圧値を観測しながらフォース・ライン(F)の電圧値を制御して必要な電圧を供給する．(A)は電流測定器であり抵抗で構成されている．その抵抗間に発生する電圧を測定して間接的に電流を測定している．そのデータはデータ・バスで CPU に送られ判定される．抵抗値を変えることにより，測定電流のレンジが変えられる．一般的に，800nA，8μA，80μA，800μA，8mA，80mA，800mA のレンジが選べる．Upper Clamp，Lower Clamp は DUT の不具合による異常電流を検知して DPS 自体を OFF させて保護する．この値もデータ・バスから与えられる．

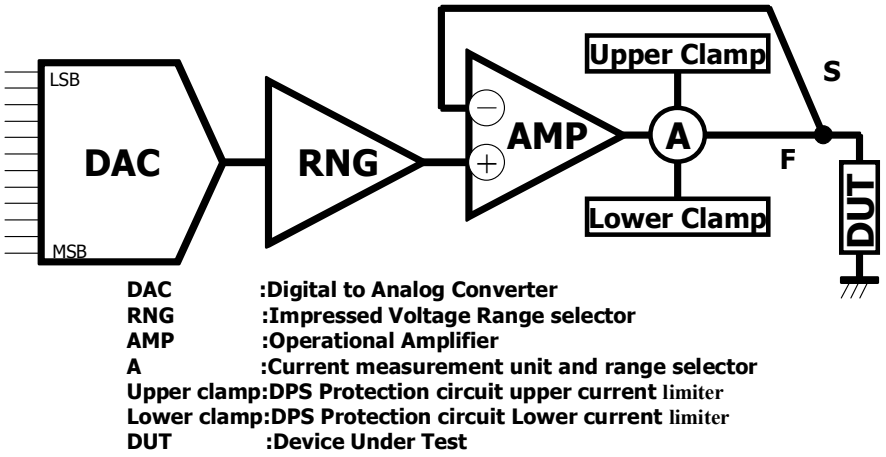


図 4.3 DPS の構造

Fig. 4.3　Structure of DPS

テスタ 1 は，1980 年に開発されたテスタであり，現在でも広く使われている．テスタ 1 の言語形式は FORTRAN 型言語の形式である．テスタ・リソースへの設定は代入式で行われる．以下に DPS の記述を示す．

VS1=0.000V,R8V,M(0.8A),400MA,-400MA

VS1 は DPS のリソース名称であり，設定値は＝の後に記載される．最初の因子 0.000V は印加電圧，第 2 因子 R8V は電圧レンジ，第 3 因子 M(0.8A)は電流レンジ，第 4 因子 400MA と第 5 因子-400MA は DPS を保護するための電流クランプ値を示す．上記の通りの順序で記述しなければならない．

テスタ 2 は 1999 年に開発された C 型言語の言語形式をもつテスタである．DPS 記述例を以下に示す．

```
DPSVSIM    dpsvsim;
    dpsvsim.pin(VS1);
    dpsvsim.SRng(R8V);
    dpsvsim.SVal(0V);
    dpsvsim.MRng(M800MA);
    dpsvsim.CPVal(400mA);
    dpsvsim.CMVal(-400mA);
    dpsvsim.Load();
```

その順序は最初の宣言文 DPSVSIM dpsvsim と最後のロード命令 dpsvsim.Load()以外は任意で良い．第 1 行と第 2 行がテスタ・リソース名，第 3 行は電圧レンジ，第 4 行は印加電圧値，第 5 行は電流測定レンジの設定である．第 6,7 行で電流クランプ値を設定し，第 8 行はテスタ・リソースへのデータ設定を行う．

テスタ 5 は 2000 年に開発された C 型言語の言語形態をもつテスタである．テスタ・リソースの設定には個別のステートメントの言語形態を取っている．その記述例を下記に示す．

```
set vs1 v=0.000v,vr=10v,cur=400ma,ir=800ma;
```

テスタ・リソースは set コマンドで記述され，その後にテスタ・リソース名を記述する．

設定される内容は＝の後に記述される. 最初の因子は印加電圧, 第 2 因子は電圧レンジ, 第 3 因子は電流クランプ値，第 4 因子 4 は電流測定レンジである．上記の順序で記述しなければならない.

次にテスト・パターン記述を比較する．論理値の記述は同一であり，入力 Hi は 1，入力 Low は 0，出力期待値 Hi は H，Low は L，判定無視は X である．その例をテスタ 1 について以下に示す．LPAT はテスト・パターンであること，PFCT2 はパターン名である．CHANNEL はテスタの使用ピン番号である．CFPF 行から STPS 行の間にパターン情報を記述する．NOP はノー・オペレーションを示しており，/Tn はタイミング発生器のタイミング･セット番号を示している．！の後に個々のテスト･パターンが記述される． END はパターン・プログラムの終了を示す.

```
LPAT PFCT2
CHANNEL 1-8,10-13
CFPF        NOP/T1      !1000XX1000XX
            NOP/T2      !0001LH0001LH
            ∫
STPS        /T11        !1001XX1001XX
END
```

その他のテスタについても，類似の記述がされている．テスタ・リソースの記述に比べて相互の変換は容易である.

これらの比較によって得られた知見を以下に示す.

FORTRAN 型言語のテスタ用言語では，記述の順番を理解する必要があり,ソフト開発が困難でかつ理解しづらい.

C 型言語のテスタ言語ではプログラム行数が長くなるため可読性の問題がある.
プログラム行数の短縮のためにテスタ・リソースの必要なパラメータを 1 行のステートメント文で記述するステートメント形式が必要になることもあるが，テスタ・ベンダ個別となっている.

テスト・パターン記述はほとんど同じであり，パターン変換は容易である．

今回の比較では，簡単な標準ロジック Edge Trigger Type　F/F デバイスを対象に使ったが，SoC ではデバイス・ピンの増大，機能の複雑化，アナログ機能搭載のために，テスト・プログラム記述は複雑になり，行数の差は更に大きくと考えられる．

テスト・プログラムは作成者だけでなく，半導体検査工程の管理者や作業者に使われる．そのために，テスト・プログラムが読みやすく理解しやすいことが望ましい．これらの点に関して従来のテスタ用言語の配慮は十分ではなかった．

4.2.3　テスト･プログラミング手法の現状

現在のテスト・プログラムの変換手法を図 4.4 に示す．前節で説明したようにテスタごとにテスタ言語が違うために，各テスタ上でプログラムを開発するのが基本である（①）．しかし，テスタの有効活用をするために，別のテスタにテスト・プログラムを変換することがある（②）[16]．この手法の問題点は，対象にするテスタが増えると幾何級数的に変換に用いるツールが増えることである．この問題の対策として，最近ではテスト実行に必要なテスタ・リソースのパラメータを EXCEL などのデータ・ベースに一度変換した後に，記述変換する手法が開発されている（③）[17]．いずれにしても，使用するテスタを考慮したテスト・プログラミングであり，LSI の短期開発評価の要求に対応できるものではない．

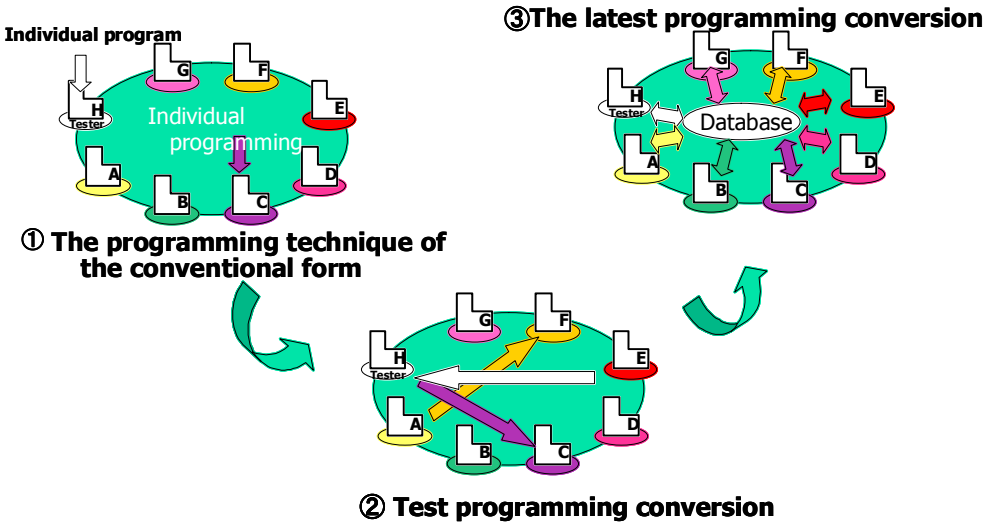


図 4.4 テスト・プログラミング手法

Fig. 4.4　Test programming method

4.3　テスタ構造表現言語

4.3.1　テスタ構造表現言語の提案

前述の課題を解決するために，以下の 3 つの方針に従ってテスタ構造表現言語 GTL(General Tester Language)を提案する．

①テスタ・リソースに併せた形で，C 言語の関数を定義する

②テスタ言語のデファクト化を目的とする

③テスト・パターンは現状のテスタ言語における一般的な記述を定義する

提案するテスタ構造表現言語方式はテスタ・リソースのデータ・ベースを C 言語の関数形として表現する．このために，各テスタのテスタ言語形式に関わらずテスト・プログラムが記述可能である．その関数形を逐次解釈し実行する関数定義を各テスタでの言語で記述し実行する．言語定義を表 4.2 に，その概略図を図 4.5 に示す．

表 4.2　テスタ構造表現言語

Table. 4.2 Tester structure expression language

テスタプログラミング言語記述		
C言語関数記述		
テスタ記述	デバイス電源	VS(ユニット番号,印加電圧,印加電圧レンジ,電流測定レンジ, 上限電流クランプ,下限電流クランプ)
	DC記述:電圧印加電流測定	VSIM(印加電圧,印加電圧レンジ,電流測定レンジ, 上限電流クランプ,下限電流クランプ)
	DC記述：電流印加電圧測定	ISVM(印加電流,印加電流レンジ, 電圧測定レンジ, 上限電圧クランプ ,下限電圧クランプ)
	ピン記述	PIN(ピン番号,入力Hi レベル,入力Low レベル,終端電圧,波形モード,クロック指定, Driver／Comparator／IO,出力比較Hi レベル,出力比較Low レベル,ストローブ指定)
	電圧印加シーケンス	TIME(シーケンス番号,待ち時間,ユニット名) SRON SROF
	テスト番号記述	TEST(テスト番号)
	測定命令記述	MEAS(測定ユニット名) REG(ユニット名,レジスタ名,データ) SEND(ユニット名,レジスタ名,data)
	待ち時間記述	WAIT(待ち時間)
	テスト･プログラム開始	MAIN(テスト･プログラム名) {
	テスト･プログラム終了	}
	テスト終了	STOP
	分岐制御命令	if(ARGn(n)<>0)
	ループ制御命令	for (I=n; I<m; I++)
	リミット記述	LIMIT(測定ユニット名,上限リミット,下限リミット)
	ピン番号のシンボル定義	PINLIST(ピン･リスト名,ピン番号)
	レート記述	RATE(レート時間)
	タイミング記述	CLK(クロック番号,ACLOK タイミング値,BCLK タイミング値,CCLK タイミング値) STRB(ストローブ番号,ストローブ･タイミング値)
	パターン記述	PATTER(パターン名) CHANNEL(ピン記述) NOP(タイミング･セット番号,パターン記述) ∫ STPS (タイミング･セット番号,パターン記述) END

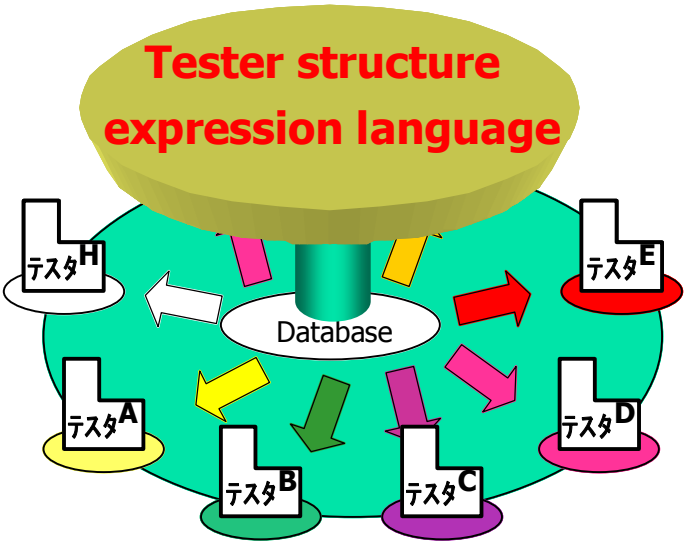


図 4.5 テスタ構造表現言語

Fig. 4.5　Tester structure expression language

DPS を例で，以下に説明する．

VS（Unit number,　Voltage,　Voltage　Range,

Measurement Current Range,　Upper Current Clamp ,

Lower Current Clamp)

VS はテスタ・リソース名であり，C 言語の関数名とし，リソースのパラメータを引数で示す．第 1 因子 Unit number はテスタ・リソースのユニット番号，第 2 因子 Voltage が印加電圧, 第 3 因子 Voltage　Range はその電圧レンジ, 第 4 因子 Measurement Current Range が測定電流レンジ, 第 5 因子 Upper Current Clamp と第 6 因子 Lower Current Clamp は保護電流クランプ値である．

提案する GTL では従来のテスト・プログラム資産を活用できる面や，テスタ構造に言語構造が対応している面より，テスタ・ハードウェアへの理解が容易になる．プログラム記述上ではハードウェアの制限を設けないため，テスタ機種に制限されずテスタを選択することができる．GTL ではハードウェアユニット毎のテスタ・リソースをテスタ・リソースとパラメータの組み合わせのステートメント形式で定義する事をを目的として

いる．　GTL のテスト・プログラムを構造木として図 4.6 に示す．図 4.2 に示したテスタ・バス構造で分かるように，データ設定はリソース・アドレスとサブ・アドレスの 2 段階で行う．GTL では，C 言語の関数を使って定義するため，現在の主流である C 型言語のテスタに関しては，関数定義が容易となる．

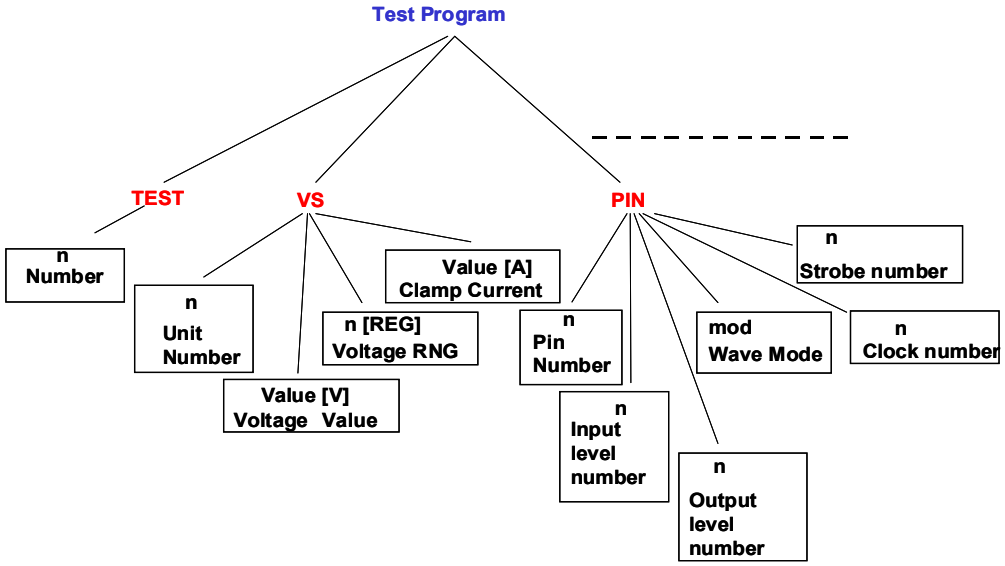


図 4.6 プログラム木

Fig. 4.6　Program structure tree

4.3.2　テスタ構造表現言語の応用

GTL の応用として，ツール群の開発を行なった．全体のシステム構成を図 4.7 に示す．ただし，⑤,⑥についてはサードベンダの物を使用した．ツール群が対象とするテスタ機種は 29 機種である．

①インタラクティブ・エディタ

インタラクティブ・エディタは GTL の言語ライブラリィを持つ．GTL で定義される関数名を記述することにより，効率よくパラメータが入力できる．このため，テスト・プログラムの形式である予約語やパラメータの順序等を知らなくても，テスト・プログラ

ムが作成できる．

②　テスタ・リソース評価ツール

GTL で作成されたテスト・プログラムを入力して，そのテスト・プログラムに必要となるテスタ・リソースを抽出する．このテスタ・リソースに従って，使用可能なテスタが確認できる．例えば，電源数，発生範囲，テスト・ピン数などのパラメータをテスタ仕様と対比させて使用可能かを判断する．

③　テスタ照会表示ツール

テスタ・リソース評価の結果として使用可能なテスタ種を表示する．使用に問題があるテスタはそのテスタ・リソースを表示し，問題点をユーザに指摘する．テスタは償却費や電力などの管理費から使用コストが決まる．そのコストについて計算して表示する機能を持たせることもできる．

④　テスタ・ナビゲータ

テスタ照会表示ツールで示された問題のあるテスタに対して，テスト・プログラム・リスト上の該当箇所を明示させる．そのプログラムをインタラクティブ・エディタで修正し，再び，テスタ・リソース評価を行う．その結果，使用可能なテスタの範囲を広げ，最良のテスタ・コストのテスタを選ぶことができる．

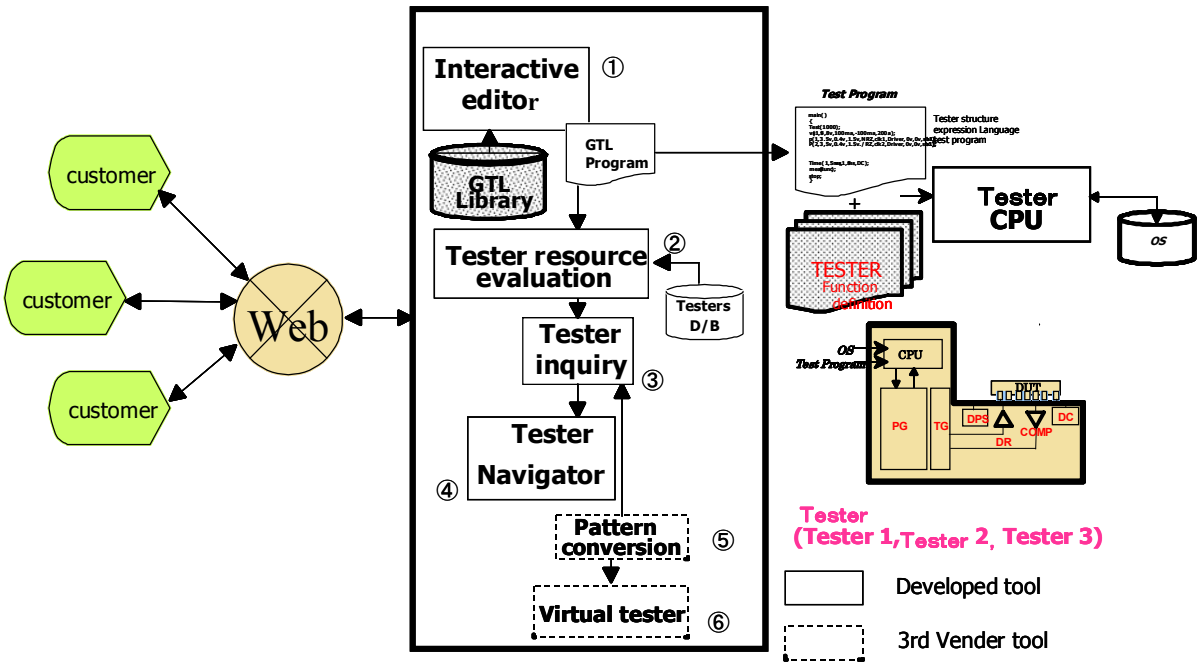


図 4.7 GTL を中心としたシステム構成

Fig. 4.7　System structure for GTL

⑤　パターンコンバータ

論理シミュレーションのパターンデータをテスト・パターンに変換する．今回は外部のベンダのツールを使った．VCD ファイル,WGL ファイル,そして STIL ファイルから変換できる．（外部ベンダツール）

⑥　仮想テスタ

エミュレータである仮想テスタは，テスト・パターンの動作検証を行い，実テスタでのデバック時間を短縮するために有効である．（外部ベンダツール）

4.4　実装結果

4.4.1　システム・スクリーン

開発した表示画面を図 4.8 に示す.

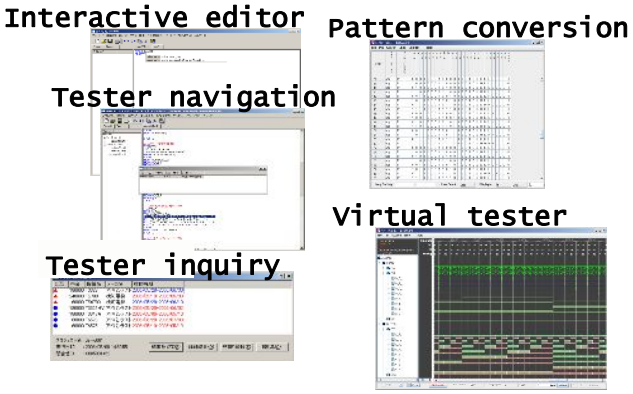


図 4.8 システム・ウィンドウ

Fig. 4.8　System windows

インタラクティブ・エディタを用いてテスト・プログラムの作成をする．次にテスト照会ツール及びテスタ・ナビゲーションを用いて使用可能なテスタの検索を行ない，その

テスター一覧を表示する．パターンコンバータと仮想テスタを用いてテスト・パターンのデバックを行う．

4.4.2　ツール群開発結果

ツール群の開発規模を表 4.3 に示す. 全体の開発規模は 15 人月で 27.8k step であった．この内，スペック・シート方式のテスト・プログラム生成を Excel VBA で作成し，テスタを知らないエンジニアにも使えるように配慮した．

表 4.3　ツール群の開発規模

Table. 4.3　Development scale of tools

Software		Tools	Man-month	Scale(Kstep)
InteractiveEditor		Visual C++	4	3.7
TesterNavi	TestCost Calculator	Visual C++	1	2.2
	Tester Resource evalution	Visual C++	2	3.2
	Tester inquiry	Visual C++	1	2.6
	GUI	Visual C++	4	11.6
Pattern Conversion		3rd Vender tool		-
Virtual Tester		3rd Vender tool		-
Test Spec Sheet		Excel VBA	3	4.5
		Total	15	27.8

4.4.3　GTL プログラムの実機評価

株式会社アドバンテスト製 T6575 及び T3343, 横河電機株式会社 TS6000 を用いてテスタ実機評価を行なった．DUT として FPGA を用い，テストには実行時間の評価をするために，8 ビット加算器を FPGA に実装して行なった．GTL で作成したテスト・プログラムは，デバイス・ピンのオープン／ショート・テスト，入力リーク・テスト，消費電力

テスト，出力電圧テスト，ファンクション・テストである．ファンクション・テストではデバイスの動作限界をテストするために，電源電圧補償範囲や入／出力レベル・テストも行った．GTL テスト・プログラムの行数は 440 行であった．

測定環境を図 4.9 に示す．デバイス/FPGA とテスタの接続は治工具の作成容易性を考えてケーブル接続で行った．このため，インピーダンスの不整合によるノイズが発生している．

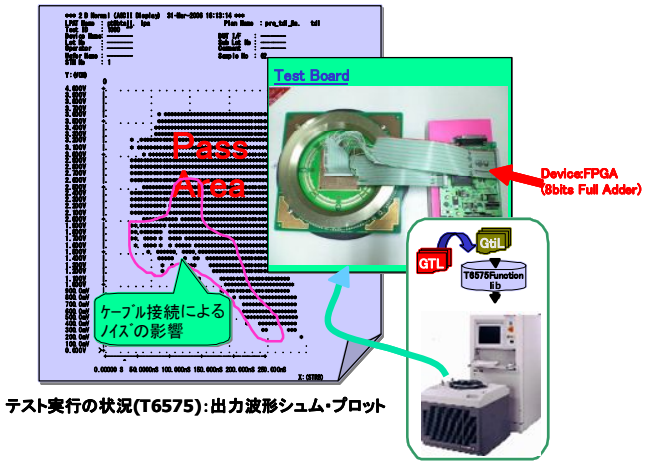
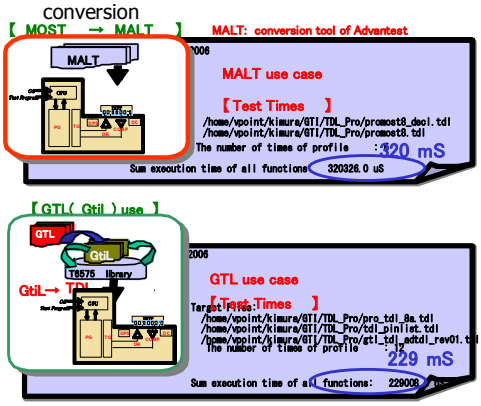


図 4.9 T6575 でのテスト環境と結果

Fig. 4.9　The environment and the result in T6575

提案手法と従来手法を比較するために，実行時間の面でテストプログラム変換手法(図 4.4②)の比較を行った．その結果をテスタのデータログも含めて図 4.10 に示す．



Test programming method	Execution Time
Test programming conversion	320ms
Proposal method (used GTL)	229ms

図 4.10 測定時間の比較

Fig. 4.10　Comparison of test time

従来のテスト・プログラム方式の測定時間は 320ms に対して，提案手法のでは 229ms であり関数定義方式でのテスト時間の増加はなかった．

4.4.4　テスタ言語の評価

図 4.11 に一般的なテスト言語で使われているコマンドを GTL がカバーする比率を示す．

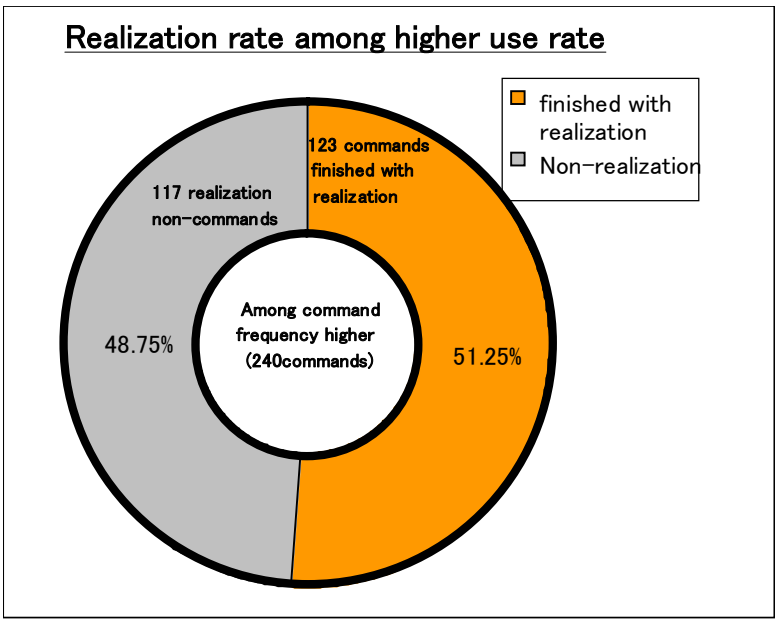


図 4.11 テスタ言語の評価

Fig. 4.11 Tester language evaluation

従来のテスタ言語では，テスタに接続する機器であるハンドラやプローバ等の制御言語やテスト・データ解析言語を含めて約 1000 コマンドを有する．このうち直接デバイス測定に使われるコマンドは約 200 コマンドであり，この半数以上を GTL はカバーする．ES(Engineering Sample)認定におけるテスト実行には十分適用できると判断する．

4.5　まとめ

本章では，GTL をテスタの構造を反映したテスタ構造表現言語として提案した．本稿で述べたように GTL 言語を C 言語の関数形記述として，各テスタでの関数定義を開発することで，テストが実行できる．また，テスタの構造を反映していることから，テスト・ハウスの所有しているテスタから，テストが実行できるテスタを選択させることができる．そのために，インターネットを使った最適テスタ照会，および，ナビゲーションができる．

今回，そのツール群の開発を行いその実動作を確認した．開発したソフトウェアの基本ソフトウェアは Visual C++を使った．その開発規模は 27.8k step であった．

参考文献

[1] The Semiconductor Industry Association , International Technology Roadmap For Semiconductor，2005 Edition.

[2] 2006 年度 STRJ 報告書，第 4 章 WG2 テスト 4-4　ATE-SWG の活動 4-4-6 テスト開発の経済性検討, 2006.

[3] 佐藤正幸，“第 14 章　テスティング技術，”1999 半導体テクノロジー大全, 電子ジャーナル(株)，pp．231-237，1998 年 10 月 28 日.

[4] T. Kazamaki, H. Maruyama, and S. Sumida, “Trial Model of 100 MHz Test Station for High Speed LSI Test System,” Proc. International Test Conference, pp. 611-617, 1978.

[5] T. Kazamaki, “A 100MHz Tester – Challenge to New Horizon of Testing High Speed LSI,” Proc. International Test Conference， pp. 618-625, 1979.

[6] S. Bisset, “The Development of a Tester-per-Pin VLSI Test System Architecture,” Proc. International Test Conference, pp. 151-157, 1983.

[7] Verigy,http://www.verigy.com/ate/products/V93000/index.htm

[8] J. Katz and R. Rajsuman, “A New Paradigm in Test for the Next Millennium,” Proc. International Test Conference, pp. 468-476，Oct. 2000.

[9] 佐藤正幸，大塚信行，武藤修，新井雅之，福本聡，岩崎一彦，上原孝二，志水勲，間明田治佳，“低消費電力型再構成テスタの開発，”信学論 D-Ⅰ Vol. J88-D-Ⅰ No.6, pp.1065-1075, jun. 2005.

[10] J386A/J386A-8 Customer Service Manual, April 1985, TERADYNE.

[11] マニアル番号 8311122，VLSI TEST SYSTEM (ATL-30) PROGRAMMING HANDBOOK，株式会社アドバンテスト, 1998 年 3 月 20 日.

[12] M. Sato, “Memory virtual tester technology and a Tester on Chip,” SEMI Technology Symposium (STS) 2000, section5 pp. 70-75, Dec. 6-8, 2000.

[13] IEEE Standard Test Interface Language (STIL) for Digital Test Vextor IEEE Std 1450-1999.

[14] IEEE Standard for Extensions to Standard Test Interface Language (STIL) (IEEE Std 1450-1999) for DC Level Specification IEEE Std 1450,2-2002.

[15] 2003 年度 STRJ 報告書，第 4 章 WG2 テスト 4-3 国内活動成果につい，4-3-2 ATE-SWG の活動 6) テスタ構造表現言語，2003.

[16] http://www.ate.co.jp/fmhp2/services/lsitest/testprogram.php

[17] http://www.aldete.com/jp/products/index.html

第 5 章

SRAM ブロックを用いた論理回路の一構成手法

本章の内容

5.1　はじめに

ここでは，テスタ・オン・チップの実現を目的として SoC に内装できる FPGA を研究した．

FPGA はリコンフィギャラブルシステムの一種であり，その研究は推進されている[1]. リコンフィギャラブルシステムでは, FPGA を細粒度演算回路の集合体としてのホモジェニアスなデバイスと位置づけられている. そのほか, 最近では粗粒度の演算回路を持ち, ヘトロジェニアスなデバイスの実用化もされている[2].

FPGA の基本単位は CLB(Configurable Logic Block)である．CLB 間はスイッチ・マトリックスで接続されている．その論理状態や接続スイッチの ON/OFF 状態は SRAM に記憶させている．この SRAM に構成データをロードして，論理動作を行なわせている[3]．基本的に論理回路をベースとしたデバイスのために, CLB 間を接続する構造が必要である. また, その FPGA 自体のテストも実施しなければならず, 多くの研究がされている[4-8]. FPGA はテストの合理化にも活用されている. テスタをより良く使う技術として仮想テスタが提唱されている[9]．特に，メモリ仮想テスタが開発され，大規模 FPGA を活用したハードウェア・エミュレータの活用が報告されている[10]．FPGA は,短時間で論理構成が可能である．1 テスト単位毎に必要なテスト機能を構成すれば，FPGA 1 個でテスタのハードウェアが実現できる．この技術は，第 3 章で述べた低消費電力基板型再構成テスタとして開発されている．半導体チップに，FPGA を構成することは原理的に可能である. このことにより，テストの論理回路を構成し，テストのアルゴリズムを実行することができる．テスト後，ユーザ論理回路として使える．しかし，チップに FPGA を搭載することはプロセス上の問題やコストの関係で適切ではない.

大規模 LSI のテストでは，テスト・コストの増大が問題視されており[11]，テスタの低価格化を目的として FPGA を用いた低消費電力基板型再構成テスタが開発されている[12]. また, 別な研究として, 大規模化するメモリ・デバイスのテスト手法において SRAM を用いた可変論理セルでテストするメモリ・テスト手法を研究してきた[13,14]．ここでは，我々は，チップ上の再構成可能な論理回路の構成を目標として， SRAM ブロックを用いた手法を提案する．最近の SoC では，内蔵メモリの増加が著しく[15]，SRAM を使

うことは一つの可能なデバイス選択と考える．SRAM は可制御性が良く，可観測性も良いデバイスである. そのテスト手法は良く研究されていて[16], そのテストは容易である．また，冗長ビットを持たせた不良修復技術（救済技術）も確立している．SRAM で論理回路を構成してテスト回路に使う研究は，上記の可変論理セルや，SRAM の論理回路としての研究がある[17]．今までの研究では論理回路の構成に通常の配線に頼っている．そのために，FPGA と同じく配線の規模に問題があった．このため，SRAM を用いて有効な論理回路を適切な容量で実現することが課題であり，SRAM ブロックを用いて論理回路を構成する基礎的検討をおこなった.

5.2　論理を構成する SRAM 構造

著者は，論理構成方法として，SRAM のアドレスの配置と入出力 I/O の配置を考慮した. その例を, 図 5.1 に示す. SRAM の m 本のアドレスとｍ本の I/O を考える, この SRAM で，上位アドレス m/2 本(Am-1～Am/2)を上方に配置して，下位アドレス m/2 本(Am/2-1～A0)を左方に配置する．さらに，I/O も上位 m/2 本(Dm-1～Dm/2)を下方に配置して，下位 m/2 本(Dm/2-1～D0)を右方に配置した．SRAM の真理値データの書込みは，各 SRAM

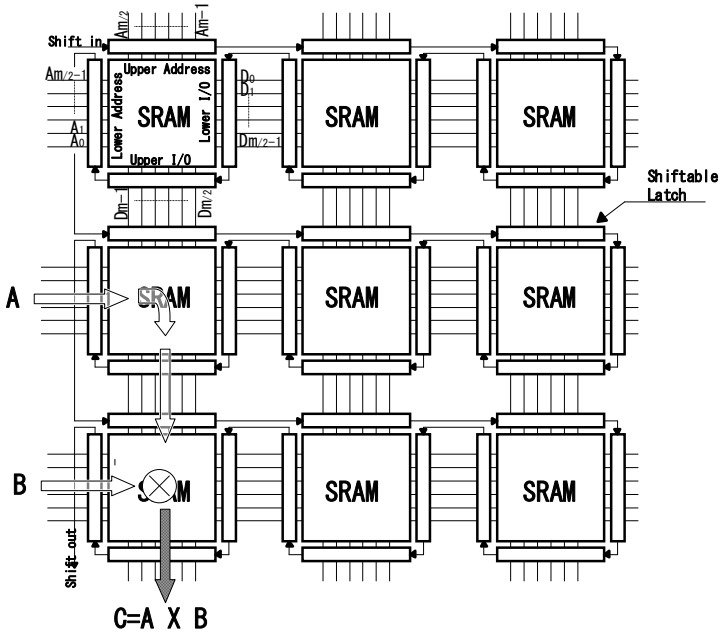


図 5.1　メモリ・モジュールの論理

Fig. 5.1　Logic of memory module

の周辺にあるシフタブル・ラッチで，アドレスを設定し I/O を入力として書込む．論理動作時は，I/O を出力にして読出しモードで動作させる．

このように配置し，各 SRAM に真理値データを設定することによって方向性と論理を持った構成が可能である．制限された形であるが，配線機能と論理機能を合わせ持つ構成ができる．例えば，乗算演算 C = A × B を，図 5.1 の左の列の SRAM2 段目，3 段目で行なった例で説明する．データ A は 2 段の SRAM の下位アドレス(Am/2-1～A0)に与えられ，その SRAM の真理値データに従い，上位 I/O(Dm-1～Dm/2)に出力される．それは 3 段目の SRAM の上位アドレス(Am-1～Am/2)に与えられる．データ B は 3 段目の SRAM の左方の下位アドレスに与えられる．3 段目の SRAM には乗算演算の真理値データを書込み，この SRAM の下方の上位 I/O に出力して実行される．

ここでの制限は，論理信号の方向性が上方または左方から，下方または右方へしかできないことである．これは，論理回路構成には不十分である．そこで，改善案としては，上下左右の 4 方向にアドレス Ai とデータ I/O Di を対にして，Ai,Di 対を持たせた配置が考えられる．これを 4 方向配置方式と呼ぶ．そのことにより，論理と配線を持たせる構造ができる．ただし，SRAM 容量規模の問題があり，それを現在の FPGA と比較して最適化の検討をした．

5.2.1　現在の FPGA の構成

ここで，現在の FPGA について分析をする． 特にアイランドスタイル FPGA の構造図を図 5.2 に示す．FPGA の基本単位は CLB である． それは配線につながれ図中では省略したが，下記に説明するスイッチ・マトリックで接続され，論理が構成される．そして，デバイス周辺の I/O 回路で外部のシステムと接続される．その CLB をその周辺のスイッチ・マトリックスも含めて図 5.3 に示す． 1 個の CLB は，基本的に 4 入力 1 出力の LUT を 2 段有する(必要に応じて論理切り替えの必要から 8 個持つ場合もある). LUT が SRAM で構成されているとすると 2**4=16b である。LUT を 8 個持つ場合は 128b のメモリ容量が必要である．

この CLB 間は，スイッチ・マトリックスで接続されている．それらは，図 5.3 に示した

ように，8×8 のマトリックスを 3 個と 2×8 のマトリックスが 1 個である．全てで 208 個のスイッチを必要としている．

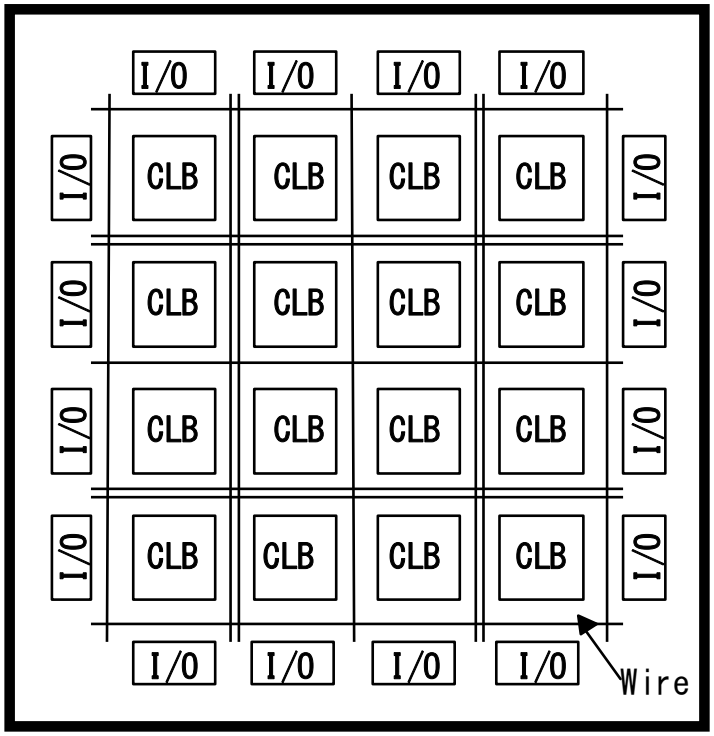


図 5.2 従来の FPGA 構造

Fig. 5.2 Conventional FPGA structure

それぞれのスイッチは，図 5.4 に示すように，接続の上下左右の完全な切り換えを行なわせるために，6 個の MOS スイッチで構成されている． その ON/OFF の制御のために SRAM を備えている．このために，スイッチ・マトリックスは 208×6=1248b の SRAM が必要である．例えば，Xilinx 社[18]の FPGA では 1 個の CLB におよそ 1.4Kb の SRAM を備えていると考えられる．XC2V3000(3M ゲート規模)の場合は，3584 個の CLB で構成されているので，4.9Mb 以上の SRAM が必要である．これは，XC2V3000 のコンフィギュレーション・メモリの必要容量が 10.6Mb であることからも過大な数値ではないと考える．

言い方を換えれば，4 入力 2 段の LUT で構成される CLB で必要 SRAM ビット数は 1.4Kb であると予想した．これに対して，適切な SRAM 容量で論理を構成する構造で実現できれば，チップ上の再構成可能な論理回路となる．以下，CLB 1 個のメモリ容量を 1.4Kb を評価数値として，SRAM ブロックを用いた論理構成の最適化を検討する．

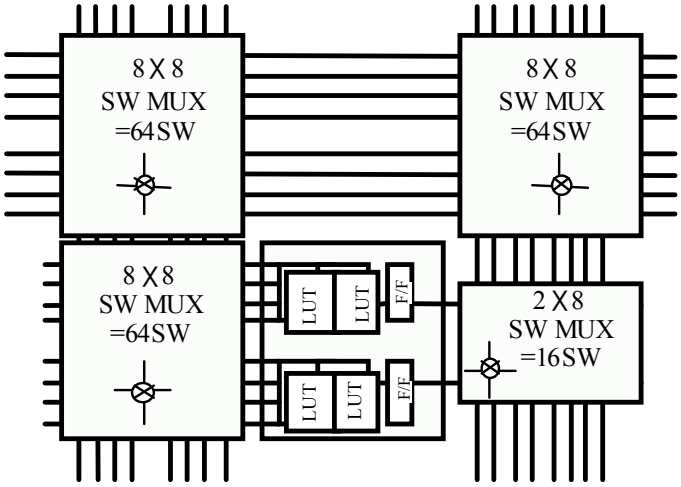


図 5.3　FPGA の CLB 構造

Fig. 5.3　CLB structure of FPGA

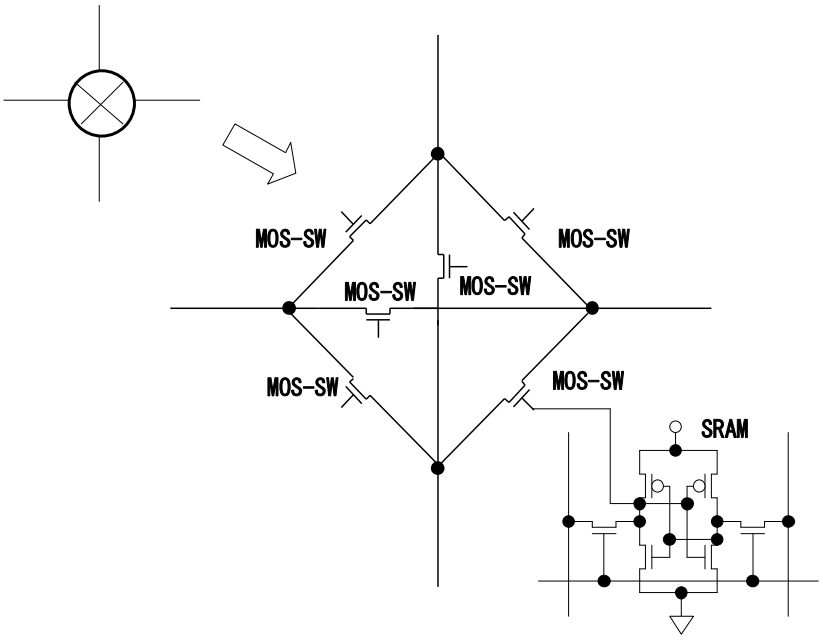


図 5.4　FPGA のスイッチ構造

Fig. 5.4　Switch structure of FPGA

5.2.2 SRAM で構成する論理回路の改善

まず，4 方向配置方式について説明する．使用される SRAM は m 本のアドレス A0～Am-1 と m 本の入出力 I/O D0～Dm-1 から構成されている．ここで，Ai と Di の対をアドレス・データ対と呼ぶ． 我々は，まず，FPGA の CLB とスイッチ・マトリックスの論理機能を同様に持たせるために，m 個のアドレス・データ対を 4 分割して，図 5.5 に示すように格子状に配置した．1 辺には 4 個のアドレス・データ対が配置される．このため，m は m = 4 + 4 + 4 + 4 = 16 となる．このために SRAM ブロックは 1Mb となり，合計容量は 1Mb×4 個=4Mb となる. これにより, 論理回路と配線接続を合わせ持った構成が取れる．ただし，これは大規模なメモリを使うことになりその活用には問題がある．なお，図を簡単にするために，シフタブル・ラッチは省いた．

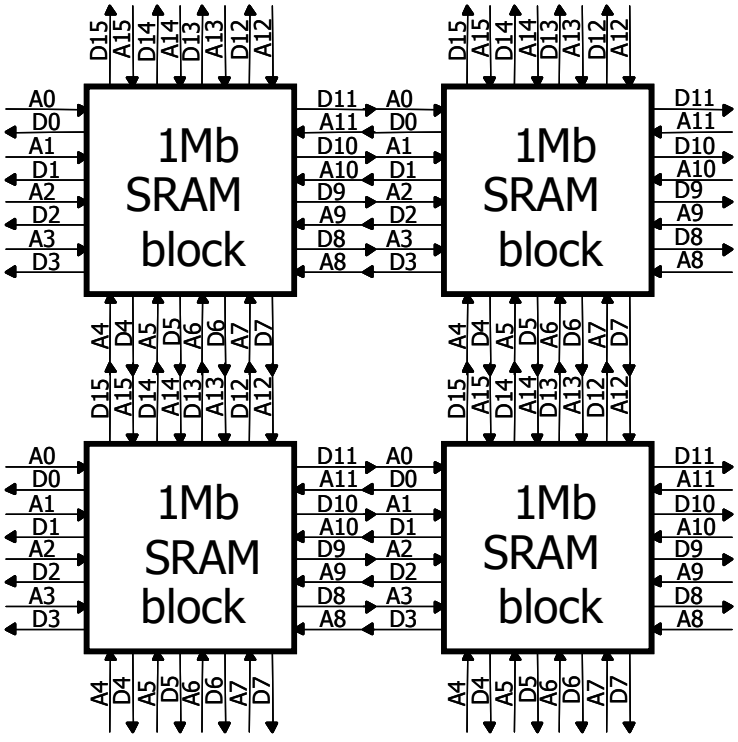


図 5.5　4 方向配置方式

Fig. 5.5　Four-direction arrangement method

この課題である SRAM の容量を抑える方法としては，下記の方法がある．

(1) 接続配置を最適化する方法

(2) 演算粒度を最適化する方法

ここでは，まず(1)の接続配置を最適化する方法を考察する．(2)については演算粒度の最適化を次節の実験で説明する．

接続配置を最適化する方法で，接続辺数を 4 辺から 3 辺に削減してアドレス本数を少なくする方法が考えられる．このように，アドレス・データ対 m を削減した 3 方向配置方式を考察した．各辺のアドレス・データ対の数を 4，2，2 とする構成を図 5.6 に示す．

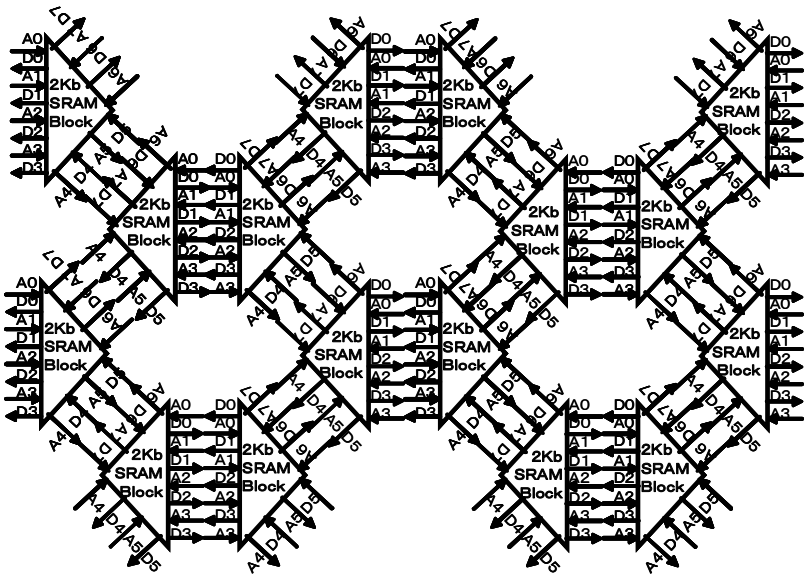


図 5.6　3 方向配置方式

Fig. 5.6　Three-direction arrangement method

m は m = 4 + 2 + 2 = 8 となる．このために，SRAM ブロックは 2Kb となる．CLB 相当にするためにその配置を考慮した，すなわち，上下左右各方向にアドレス・データ対をそれぞれ 8 個備えるために，SRAM ブロックを 16 個配列した．容量は，2Kb×16 個=32Kb に削減できる．このようにアドレス・データ対の配置を考慮すれば，SRAM ブロックの容量が削減できる．CLB(1.4Kb)の 15 倍程度にできた．

上記 3 方向配置方式について，図 5.7 のようにさらに改良する．2 列および 4 列のアドレス・データ対を 2 個ずらして配置する． この配置をさらに分析する．各辺のアドレス・データ対の 4，2，2 の配置で次のブロックへの接続は，右側の上のアドレス・データ対

は次のブロックの左側の 4 つのアドレス・データ対の下側に接続されている．右側の下のアドレス・データ対は次のブロックの左側の 4 つのアドレス・データ対の上側に接続されている．

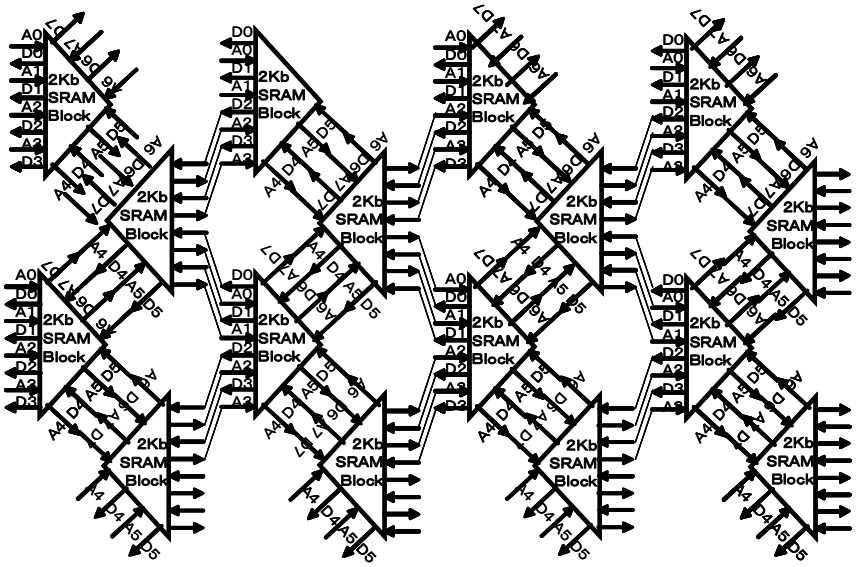


図 5.7　3 方向配置方式の改良

Fig. 5.7　Improvement of three-direction arrangement

このことに注目すると，3 方向配置方式は，図 5.8 のアドレス・データ対 4 の 2Kb SRAM ブロックの交互配置となる．

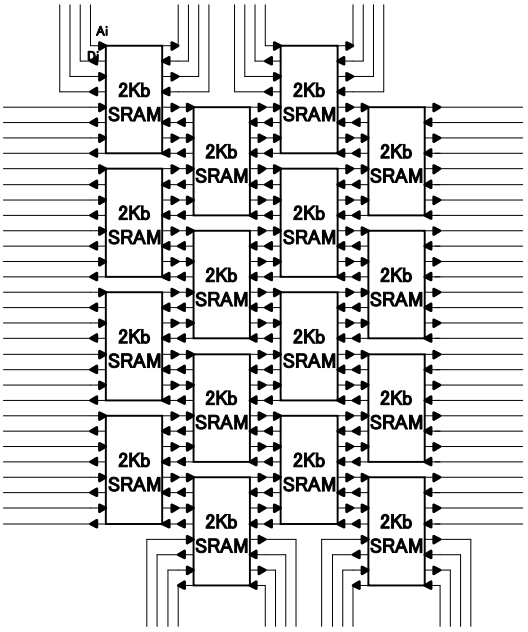


図 5.8　　2K ビット SRAM の交互配置

Fig. 5.8　Alternating　arrangement of 2Kb SRAM block

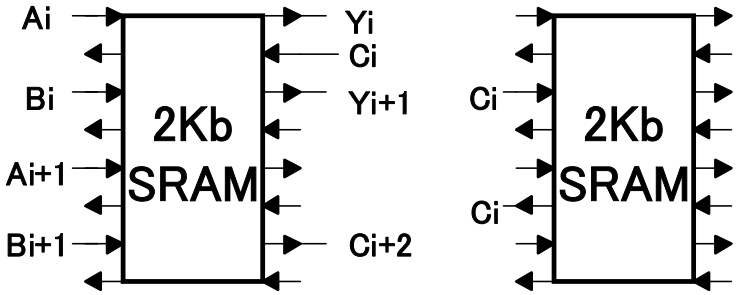
5.3　実験

図 5.8 の配置について，8 ビット加算器，32 ビット乗算器の実装およびその演算粒度を検討した．

5.3.1　8 ビット加算器の実装

8 ビット加算器の実現として，2Kb SRAM に真理値データを書込み，動作を検証した．1 個の 2Kb SRAM ブロックを用い図 5.9(a)に示した 4 進数加算器の真理値データを書込み，演算を実行している．これを図 5.8 の第 1 列 4 個の 2Kb SRAM に配置した．図 5.8 の第 2 列の 2Kb SRAM には図 5.9(b)に示したように真理値を格納しキャリア転送をしている．

この実験では，微少内蔵メモリを持つアルテラ社[19]の Stratix デバイスに実装した．使用したツールは同社の Quartus-Ⅱである．その実装状況を図 5.10 に示す．



(a)　　(b)

図 5.9　8 ビット加算器の要素

Fig. 5.9　Elements of 8-bit adder function

図 5.10 で示されるように 2Kb SRAM を 13 個実装した．その接続配線はツールの関係上表示されていないが，図 5.8 の 2Kb SRAM の交互配置となっている．各々の SRAM はクロック同期で動作する．左側の列 4 個の 2Kb SRAM は単に演算結果 Y0～Y7 を転送している．結果を図 5.11 に示す．上段が A データ入力で中段が B データ入力である．下段が演算結果の Y データ出力である．図 5.5 の 4 方向配置では，1Mb の SRAM を 2 個用いるので合計 2Mb の SRAM が必要であるが，図 5.8 の方式では，図 5.10 に示すように，8 ビット加算器は 9 個で構成される．このために，2Kb×9 個=18Kb で構成されたことになる．

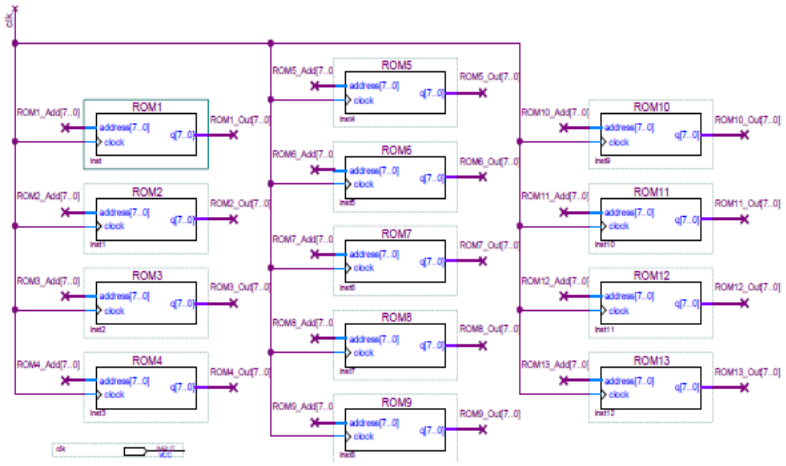


図 5.10　2Kb SRAM の実装(8 ビット加算器)

Fig. 5.10　Implementation of 2Kb SRAM (8-bit adder)

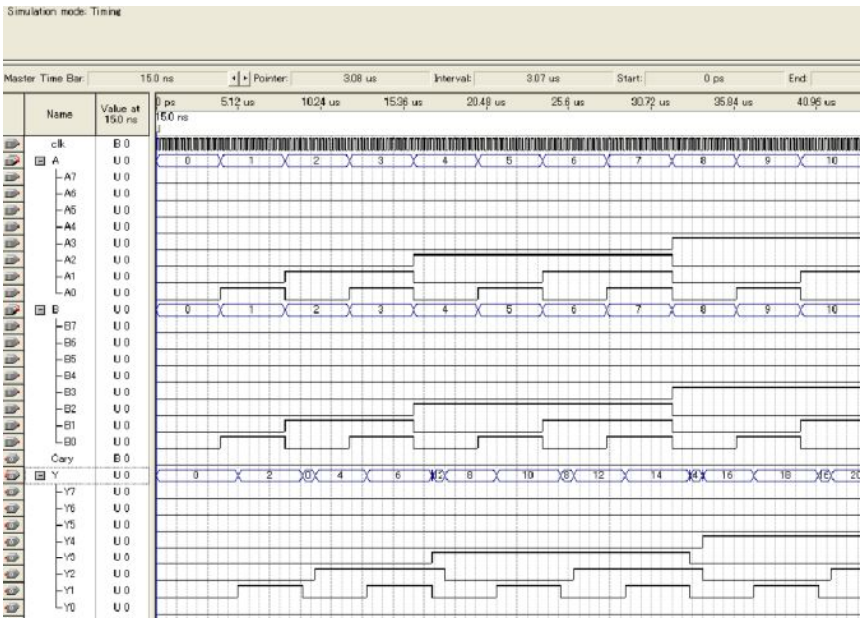


図 5.11　8 ビット加算器の動作

Fig. 5.11　Function of 8-bit adder

FPGA では，8 ビット加算器は 8 個の LUT を使う．このために，4 個の CLB を使うので 1.4Kb×4 個=5.6Kb の容量が必要になる．本手法を用いると，FPGA の 3 倍程度の SRAM ブロックで構成できた．

5.3.2　32 ビット乗算器の実装

32 ビット乗算の実装の前に，4 ビット乗算器を説明する．被乗数 4 ビット A データを A3,2 と A1,0 の 2 ビット形式とする．乗数 4 ビット B データは B0,B1,B2,B3 の 1 ビット形式である．演算 A×B は図 5.12 のように表され乗算結果 Y0,Y1,Y2,Y3 を求める．演算項は①から⑧の 8 項からなる．

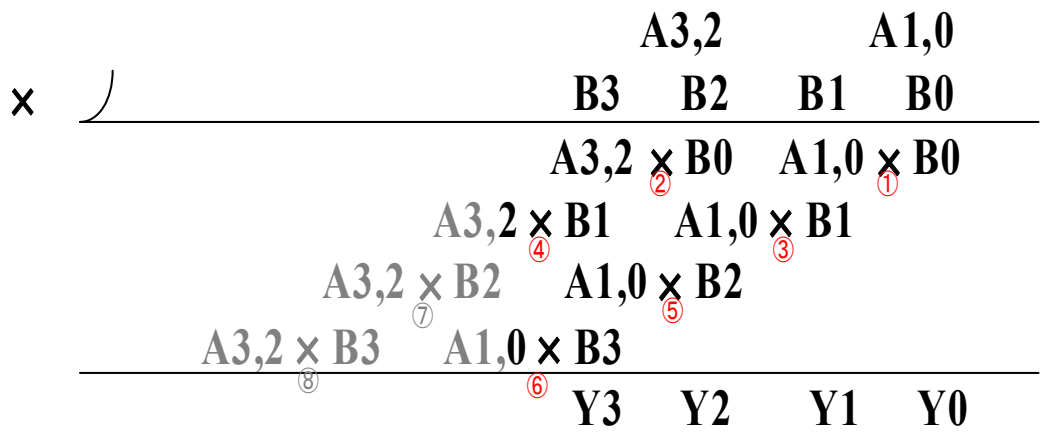


図 5.12　4 ビット乗算式

Fig. 5.12　4-bit multiplier expression

この演算項を 2Kb SRAM で求める論理と配置を図 5.13 に示す．今回は 4 ビット以内の演算とするので⑦，⑧項は不要である．それを 32 ビット乗算器にした配置を図 5.14 に示す．

以上の考察をアルテラ社 Stratix に展開する．そのために 2Kb SRAM を図 5.10 と同じように 17 段と 16 段の交互配置を 47 列配置した．その乗算のシュミュレーション結果の一部を図 5.15 に示す．

演算に使用された 2Kb SRAM ブロックは図 5.14 に示すように 392 個であり，使用メモリは 784Kb である．FPGA ではこの 32 ビット乗算器は 446 個の CLB で構成され 624Kb が必要なので，同程度で実装された．

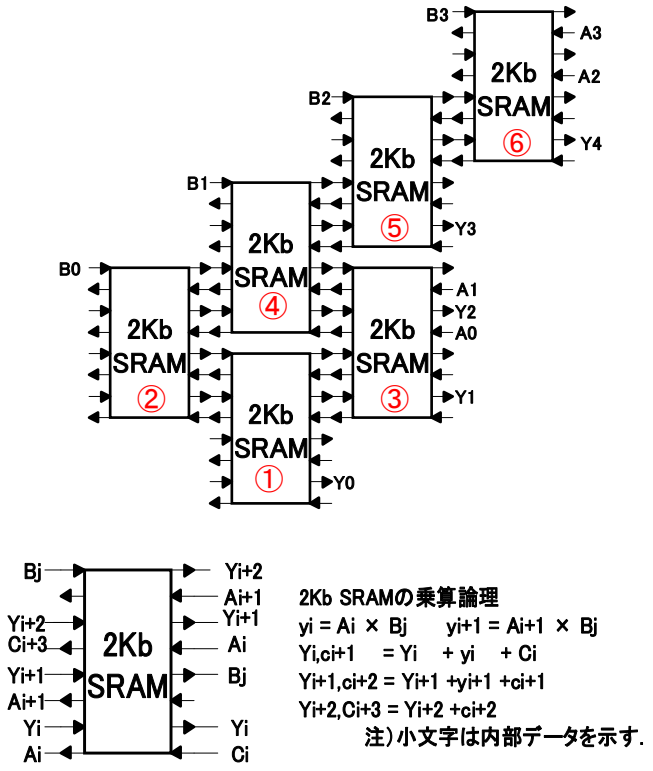


図 5.13　乗算演算項の 2Kb SRAM の論理

Fig. 5.13　Logice of 2Kb SRAM for multiplier sector

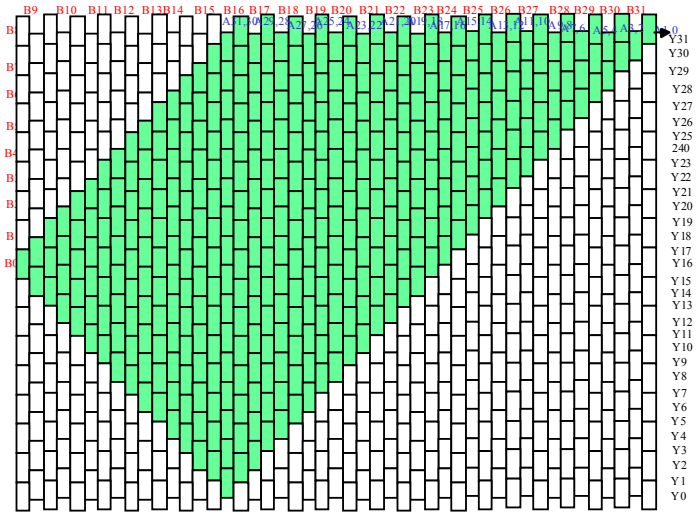


図 5.14　32 ビット乗算器の 2Kb SRAM 配置

Fig. 5.14　2Kb SRAM arrangement of 32-bit multiplier

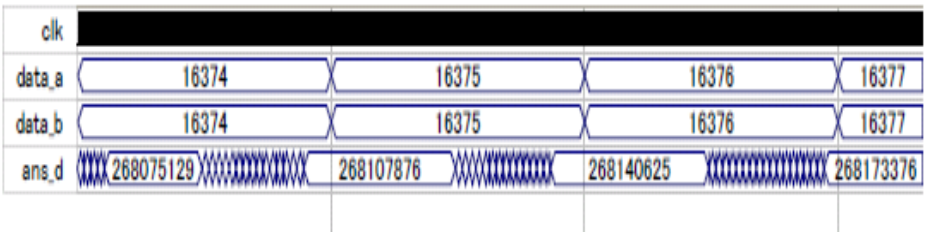


図 5.15　32 ビット乗算器の演算

Fig. 5.15　Function of 32-bit multiplier

5.3.3 演算粒度の検討

今までは，4 対のアドレス・データ対の SRAM ブロック(2Kb)の交互配置を実装してきた．そのさらなる改善を検討した．図 5.16 に示すように，アドレス･データ対すなわち演算粒度を 4 から 3 および 2 と削減することにより，CLB の 1.4Kb に近い配置が取れる．今回は演算粒度 2 の SRAM(64b)ブロックをアルテラ FPGA である Stratix1S40 の微小メモリ 512b で構成してみた．この構成についてはマルチ・ポート SRAM に構成し，且つ，SRAM ブロックを上位ブロックと下位ブロックに持たせて 128b で構成させた．

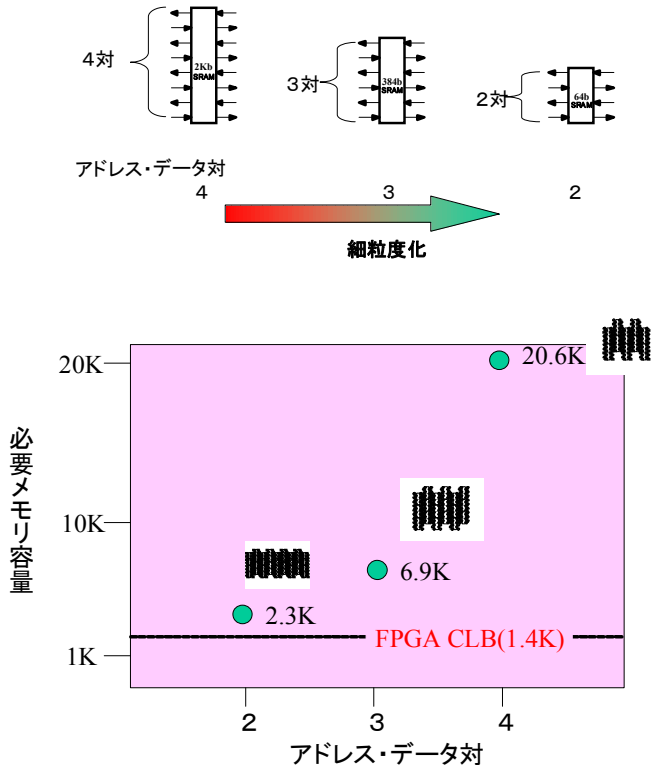


図 5.16　SRAM ブロックの細粒度化

Fig. 5.16　Fine-grained SRAM block

その構成を図 5.17 に示す．SRAM ブロックは半ブロック上下させて，書込みアドレス配線を一致させた．論理書込みは，書込みポートから行なわせ，論理動作は読出しポートで行なわせた．この構成の特徴は，書込みポートは通常のメモリとして使えることである．この構成を図 5.18 に示す．

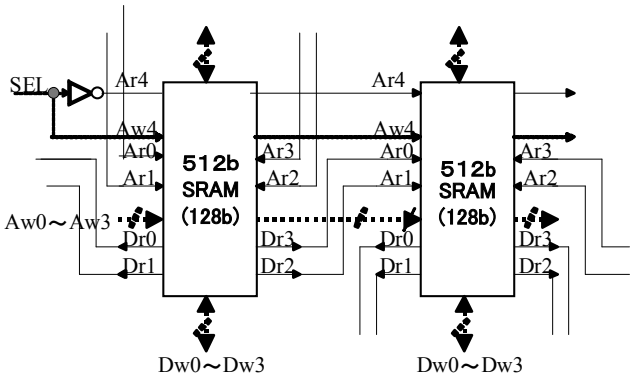


図 5.17　アドレス・データ対の接続

Fig. 5.17　Connection of address data pair

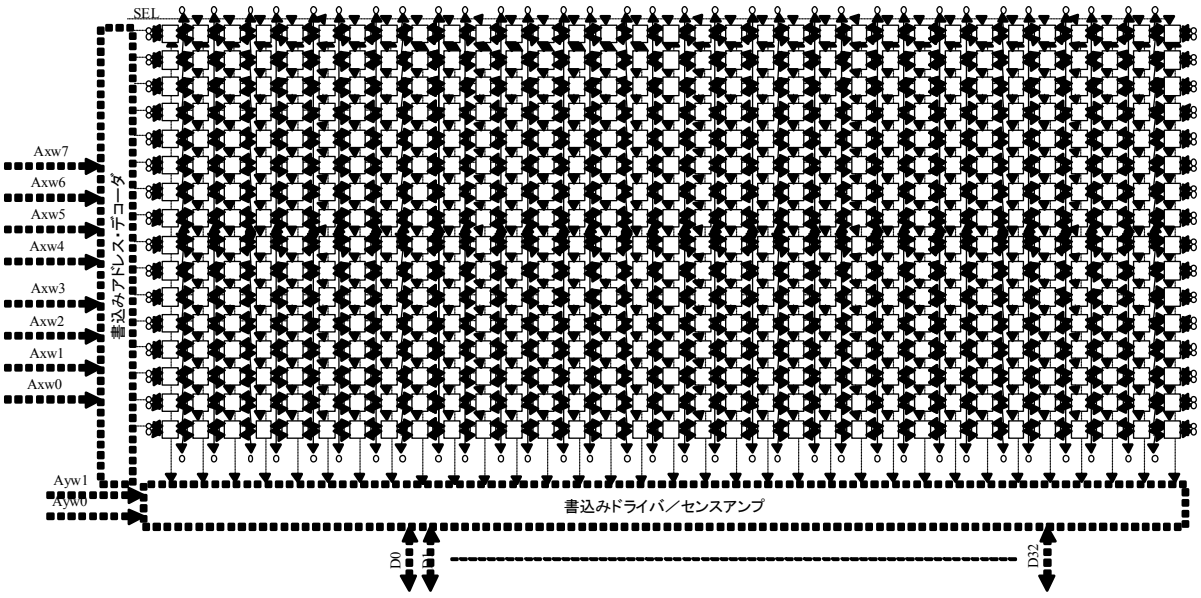


図 5.18　Stratix 1S40 での実装

Fig. 5.18　Construction on Stratix 1S40

この配置を使い階乗計算をさせた. それを図 5.19 に示した. 階乗計算をさせるために, 乗算 Y = A × B の演算結果 Y は入力 B に帰還させ, 入力 A に所定の数値を入力した. 128b の下位の 64b には掛け算の初期化(Y=1 の設定)を行なわせた. 上位 64b では乗算器を構成させた. 上位/下位アドレスを切り替えダイナミック・リコンフィギャラブル機能を実現した. Stratix 1S40 の 512b SRAM の個数の制限から 12 ビット演算となったが動作は確認できた. なお, この結果から 32 ビット乗算器で換算すると 140Kb で構成されることが分かる. このことから乗算器の改善が行なわれ, 演算粒度の細粒度化の効果が確認できた.

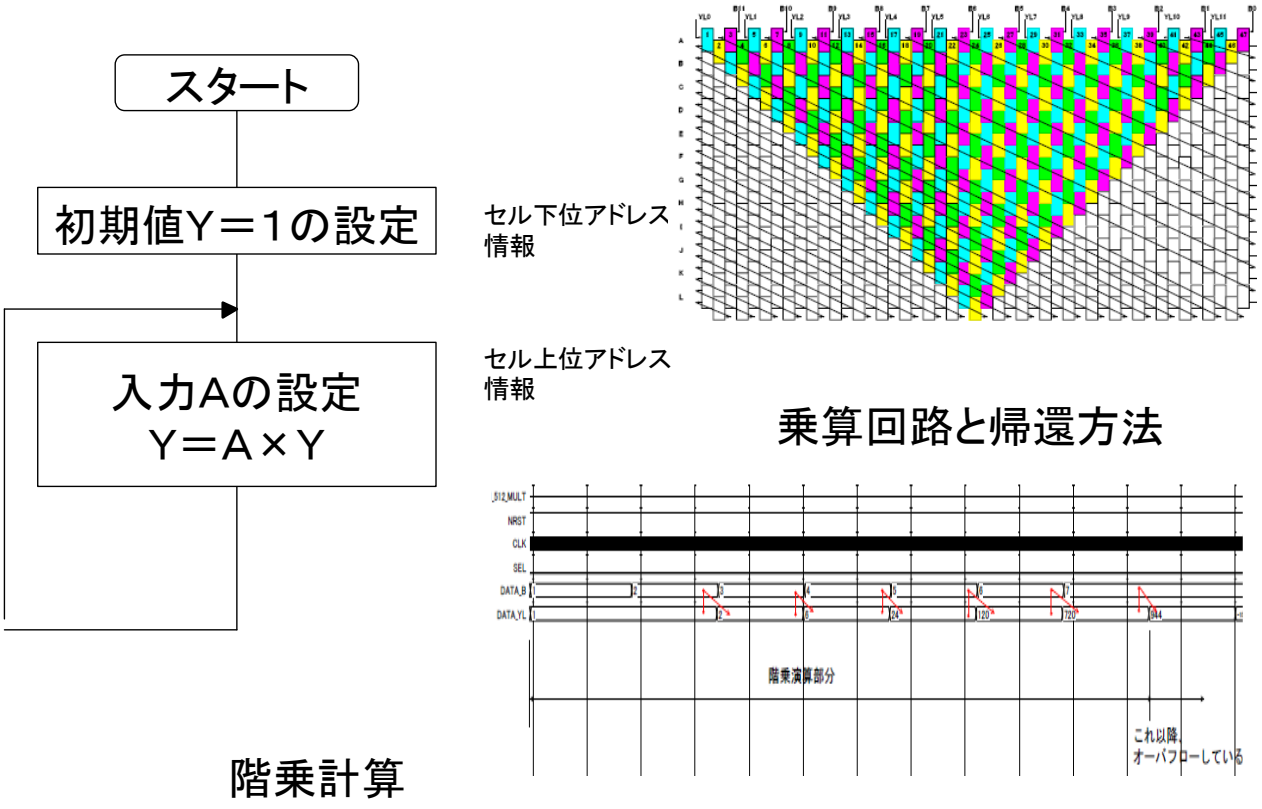


図 5.19　階乗計算の実験

Fig. 5.19　Experiment on factorial calculation

上記の実験のまとめとして, 下記に示す.

メモリを使った再構成論理回路として 2Kb SRAM の交互配置を提案した.

8 ビット加算器および, 32 ビット乗算器を実装して FPGA の 1～3 倍程度で実装できた. また，演算粒度を小さくすることにより乗算器規模削減の見通し得た.

アルテラ社 Stratix1S40 に実装してダイナミック・リコンフィギャラブル動作を階乗計算で確認した.

次の節でこの SRAM ブロックを用いた論理構成手法の具体的な試作結果を述べる.

5.4　MPLD の開発

5.4.1　MPLD の構成

上記SRAMブロックを用いた論理構成手法を発展させ, 論理要素LUTと配線要素の接続スイッチに使用することができるMLUT(Multiple Look Up Table)を導入した．この手法をメモリ・ベース・プログラマブル論理回路MPLD(Memory-based Programmable Logic Device)として提案した．そしてその試作設計を行った．以下，その詳細を示す.

図5.20にMPLDの構成を示す．MPLDは基本的にメモリデバイスであり，アドレス(Address)を持ち，列デコーダ(Column Decoder)でワードライン(Word Line)を選択させ，メモリセルを活性化させる．選択されたメモリセルのデータは，データライン(Data Line)に接続されており，制御信号(Control Signal)でデータの書込み読出しが行われる．この場合，データラインは行デコーダ(Row Decoder)で選択され外部のDataバスに接続されている.

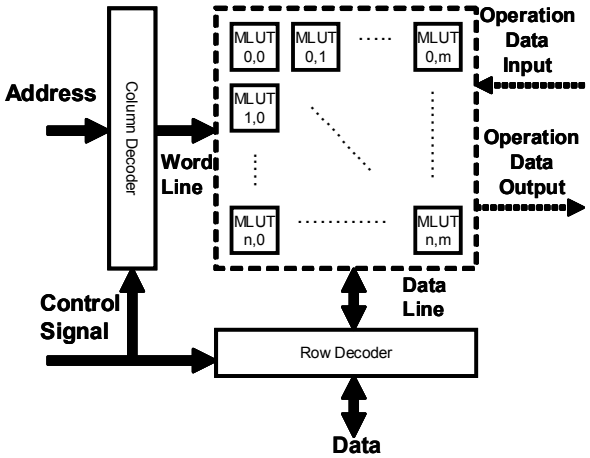


図 5.20　MPLD の構成

Fig. 5.20　Structure of MPLD

メモリセルは，適当なメモリ容量を構成単位としてメモリセル・ブロック(SRAMブロック)を構成している．このSRAMブロックをMLUTと呼び，アレイ状に配置(MLUT0,0～MLUTn,m)している．MPLDでは，このMLUTの配置と配線を工夫することでLUTの論理機能と接続スイッチ機能を持たせる．また，論理動作時の外部入出力のため，Operation Data Input/Output線を複数持つ．

MLUTには，2ポートメモリを用いて構成している．図5.21に示すように，ポートAは，メモリ動作用，ポートBは論理動作用である．メモリ機能を行わせる場合，ポートAのアドレス入力(MADR)と入出力データ(MDATA)でデータの読み書きを行う．これは，論理動作のコンフィギュレーション書込みにも使われる．ポートBのアドレス入力(LADR)とデータ入出力(LDATA)は，論理回路を構成するために利用する．ポートBのアドレス線Nビットの入力LADRと，データ線Nビットの出力LDATAを1ビット単位で対にする．このLADRとLDATAを対に使い，MLUTに対する入出力線路を構成する．具体的には$LADR_1$と$LDATA_1$，$LADR_2$と$LDATA_2$，･･･，$LADR_N$と$LDATA_N$の各対で，入出力線路を構成する．アドレスとデータの対で構成する入出力線をAD対と呼ぶ．このために，MLUT自体のメモリ容量は2^N×Nビットになる．

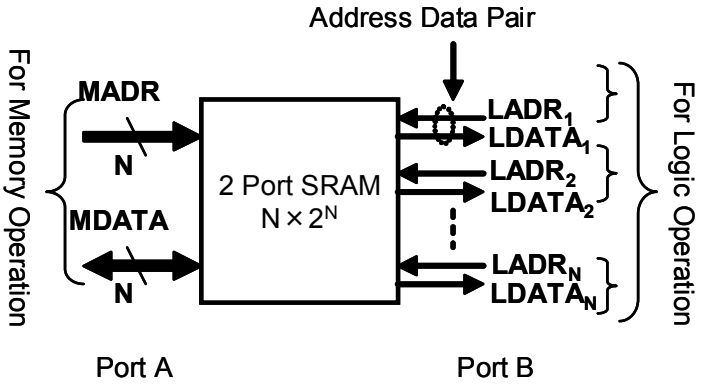


図 5.21　MLUT の構成

Fig. 5.21　Structure of MLUT(Multiple Look Up Table)

次に，MPLDのMLUT構成について述べる．図5.22にMLUTにおけるブロック図および配置図を示した．ここでは，AD対を4対用いて構成した場合を示した．図5.22(a)で示し

た図のように，AD対は左右に2対ずつ持たせる．このMLUTを図5.22(b)に示すように，MLUTを交互配置し，MLUTの論理値を考慮することで，接続スイッチ機能を持たせることができる．この図ではMPLDのMLUTアレイの左上の部分MLUT0,0～MLUT2,2を示している．MPLD周辺のMLUT(MLUT0,0 MLUT1,0 MLUT2,0 MLUT0,1 MLUT0,2)のLADR, LDATAは, 本デバイスの論理回路入出力として使用している(In00～In50, Out00～Out50). ここでは，説明の簡略からAD対数4の場合を説明したが，AD対数を選択することで，MPLDが必要とするメモリ容量を削減することができる．

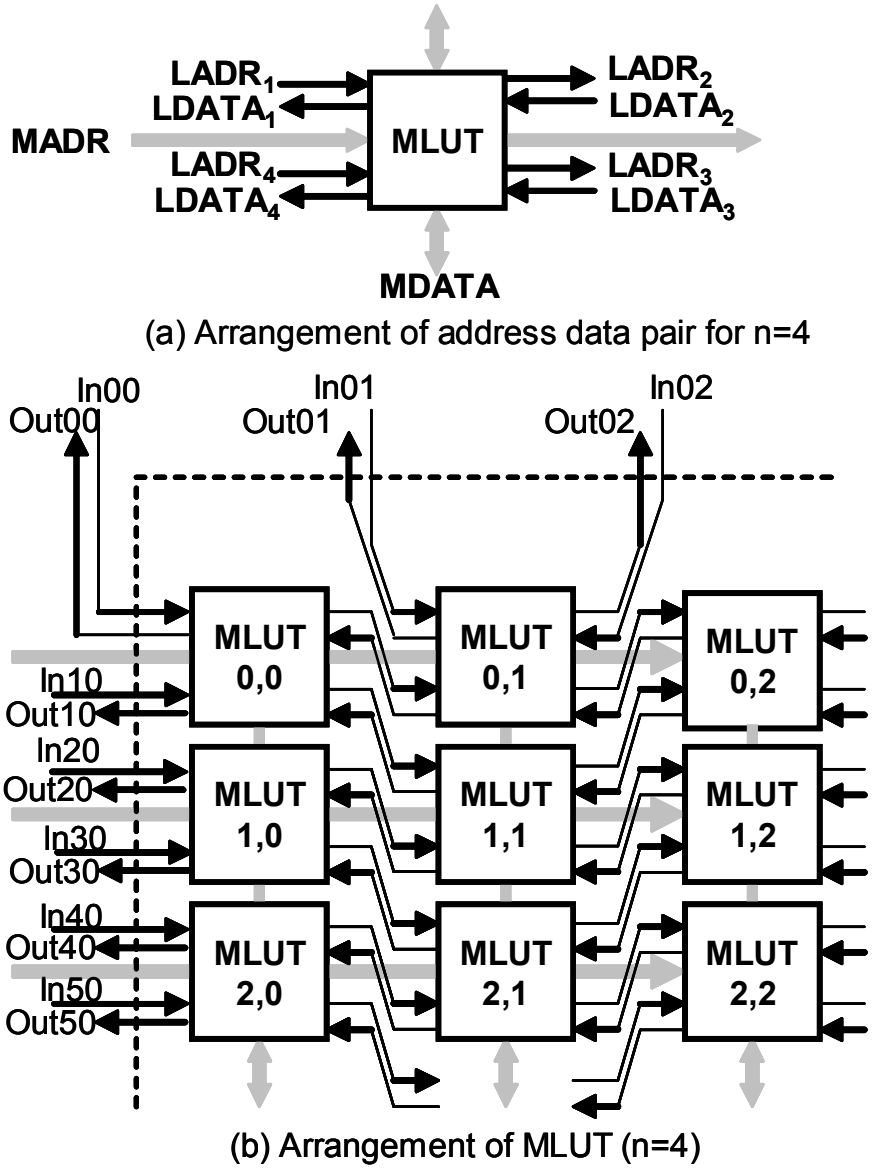


図 5.22　MPLD の MLUT 構成

Fig. 5.22　MLUT structure of MPLD

5.4.2　MPLD の設計

本MPLDの製品試作設計を行った．そのトップモジュールを図5.23に示す．MLUT間は，メモリ動作用と論理回路動作用の配線で接続している．MPLDは，SRAMブロックであるMLUTを交互配置でアレイ状に配置(MLUT0,0~MLUT15,3)している．MLUTのポートA(MADR，MDATA)は，この図では図示していないが，通常のメモリセル・ブロックと同じく構成されている．MPLDの最適化の研究[20]からAD対数を6とした．AD対数から決まるMLUTのメモリ容量は2^6×6=384ビットであるが，論理動作時にコンフィグレーションを同時に行うことができるように(マルチコンテキスト切替え)，2倍のメモリ容量768ビットとした．これを16×4ブロック持たせ，全体で48KビットのSRAMを構成した．

MLUT のポート B(LADR, LDATA)は，論理回路動作用として接続されている．また，順序回路実現のため D-FF(D type Flip Flop)を MLUT アレイ周辺に持たせた．

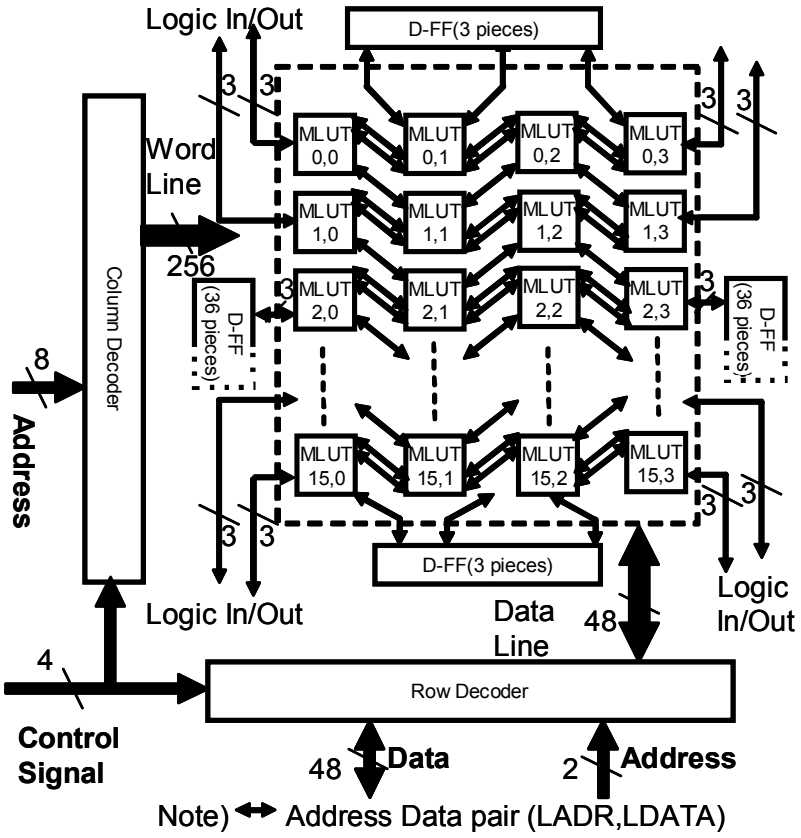


図 5.23　MPLD チップの構成

Fig. 5.23　Structure of MPLD chip

5.4.3　MPLD チップ試作

前節のMPLDの試作を行うのに，チップサイズ2.5mm角のROHM 0.18μm CMOSテクノロジを用いた．設計はフルカスタム設計で行った．この設計には，回路設計のために，Cadence社のComposer, レイアウト作成ツールとしてCadence社のVirtuosoを使った. また，レイアウトや回路図の動作シミュレーションは，Synopsys社のHSPICEやNanosimを用いた．作成したレイアウト図を図5.24に示す．作成したMPLDのコア部分の面積は1767.54×1690.06 μm^2であった．MLUTやMLUT間の配線は，1～3層で構成されており，4～5層で各MLUTの電源供給を行っている.

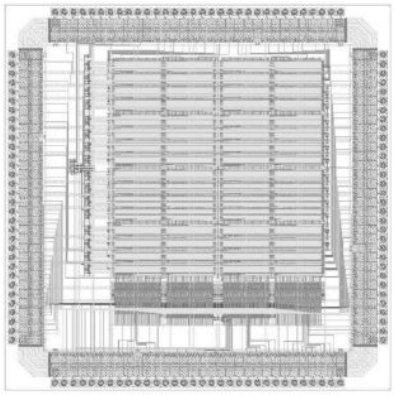


図 5.24　MPLD チップ

Fig. 5.24　MPLD Chip

設計したMPLDチップ諸元を表5.1に示す．メモリ動作時のアドレスは，10ビットでワード幅は48ビットである．制御信号としては，書込み信号WE，読出し信号RE，プリチャージ信号PREとマルチコンテキスト切替え信号Contextがある．MLUTは16×4ブロックで

表 5.1　MPLD チップ諸元

Table. 5.1　MPLD chip parameters

Function	Specifications	Parameter	Note
Memory	Address line	10	48Kbits
	Data line	48	
	Control signal	4	WE,RE,PRE Context
Logic	MLUT Pieces	16×4	AD pairs:6
	Input line	102	D-FF:78pices PAD connect 24/24
	Output line	102	

ある．その結果，MLUTアレイ外周に出てくる論理回路動作用入出力線数は各々102本であるが，この内，各々78本は内部のD-FF接続に使い，残りの各々24本をチップの外部I/0パッド接続に使用した．

HSPICEでの動作速度の評価結果を表5.2に示す．MPLDのMLUTポートAによるメモリ動作読出しの遅延時間は16.4nsec，書込みの遅延時間12.8nsecであった．また，MPLDのMLUTポートBの論理動作評価としては，MLUTとMLUT外周のD-FFで32ビット･リップルキャリーカウンタを構成させ評価した．その論理動作の遅延時間は，9.35nsecであった．

表 5.2　評価結果

Table. 5.2　Evaluation Results

Behavior	Latency
Read	16.4nsec
Write	12.8nsec
32-bit Counter	9.35nsec

5.5　まとめ

本章では，SRAMブロックを用いた論理構成手法を発展させ，FPGAの論理要素と配線要素の双方を，SRAMブロックであるMLUTで実現するMPLDを提案した．その実現性確認のため，プロトタイプデバイスを設計し，評価結果を示した．評価結果から，MPLDはMLUTという均質構造で実現でき,再構成デバイスとしてもメモリとしても使用できることを示した．これは，アイランドスタイルFPGAに近い規模で構成できるものであり，SoCに内装できる再構成テスト回路として活用できる見通しを得た．図5.25にアイランド

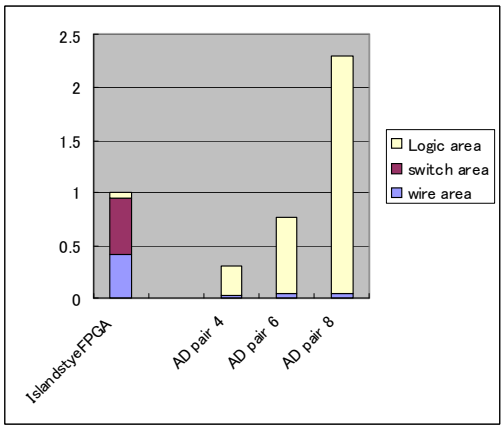


図 5.25　規模比較

Fig. 5.25　Scale comparison

スタイルFPGAとの規模比較をAD対4，6，8の比較で示した．このように，アイランドスタイルFPGA規模相当で，論理構成ができる．今後の課題は，このMPLDの改良とSoC内装技術の検討，および，そのテスト応用である．

参考文献

[1] 末吉敏則，天野英晴，リコンフィギャラブルシステム，オーム社，2005.

[2] 榊原泰徳，佐藤友美， 動的再構成可能デバイス,その素性と実力, 第 2 章 DAPDNA のデバイスアーキテクチャ, Design Wave Magazine 2004 年 8 月号，CQ 出版，2004.

[3] The Programmable Logic Data Book， Xilinx， Inc. 1998.

[4] M.Renovell，P.Faure，J.M.Portal，J.Figuers, and Y.Zorian，“ IS-FPGA:A New Symmetric FPGA Architecture with Implicit SCAN,” Proceedings IEEE International Test Conference，pp.924-931，2001.

[5] S.Toutounch，and A.Lai，“FPGA Test and Coverage，” Proceedings IEEE International Test Conference，pp.599-607，2002.

[6] D.Fernandes，and I.Harris，“Application of Built in Self-Test for Interconnect Testing of FPGAs，” Proceedings IEEE International Test Conferencepp.1248-1257，2003

[7] C.Stroud，K.Leach，and T.Slaughter，“BIST for Xilinx 4000 and Spartan Series FPGAs:A Case Study，” Proceedings IEEE International Test　Conference，pp.1258-1267，2003

[8] D.Mark，and J.Fan，“Localizing　Open　Interconnect Defects using Targeted Routing in FPGA’s，” Proceedings IEEE International Test Conference，pp.627-634，2004

[9] 半導体技術ロードマップ専門委員会(STRJ)2002 年度報告, 第 3 章 2002 年度国際・国内活動報告, 3-2 WG2 テスト, 3-2-3-2 ATE-SWG,(2)活動報告, 2)新テスタ技術, 2002.

[10] 佐藤正幸, “メモリ仮想テスタ技術と今後の展開,” 信学技報, フォールトトレラントシステム研究会, FTS98-121 , pp. 33-39, Feb. 1999.

[11] The Semiconductor Industry Association,International Technology Roadmap For Semiconductor1999 Edition, 2005.

[12]佐藤正幸,，大塚信行，武藤修，新井雅之，福本聡，岩崎一彦，上原孝二，志水勲，間明田治佳，“低消費電力型再構成テスタの開発,” 信学論 D-Ｉ, Vol. J88-D-Ｉ, No. 6,

pp. 1065-1075, Jun. 2005.

[13] 山形優輝, 市野憲一, 新井雅之, 福本聡, 岩崎一彦, 佐藤正幸, 板橋裕行, 村井崇, 大塚信行, “SRAM を用いた可変論理セルで構成したメモリテスタの実装, ” 2003 年 電子情報通信学会総合大会 D-10-7, p.162, Sep. 2003.

[14] Y.Yamagata, K.Ichino, M.Arai, S.Fukumoto, K.Iwasaki, M.Sato, H.Itabashi, T.Murai, and N.Otsuka , “ Implementation of Memory Tester Consisting of SRAM-Based Reconfigurable Cells, ” Proceedings Asian Test Symposium 2003, pp.137-142 November 11-12, 2003

[15] 半導体技術ロードマップ専門委員会 平成 15 年度報告,第 2 章,設計タスクフォース, pp.12-56, Mar. 2004.

[16] A. J. Van de Goor, *Testing Semiconductor Memories,* John Wiley & Sons, 1991.

[17] 井口幸洋, 笹尾勤, “LUT カスケード・アーキテクチャについて, ” 電子情報通信学会技術研究報告, 第 1 回リコンフィギャラブルシステム研究会, pp.193-198, Sep. 2003

[18] http://www.xilinx.co.jp/

[19] http://www.altera.co.jp/

[20] 吉原理記, 平川直樹, 谷川一哉,弘中哲夫, 佐藤正幸, “再構成デバイスとしても動作するメモリ(MPLD)の一実装例,” 信学技報, RECONF2007-16, pp.7-12, Sep. 2007.

第 6 章

結論

本論文では，汎用半導体テスタの有効活用やその合理化目的として，汎用半導体テスタのアーキテクチャについて検討し，その分類や特徴を分析した．また，テストのテスタ・デバックの事前検討のために仮想テスタを提案した．

汎用半導体テスタは，構造をもっており，一般的にシェアード・リソース・テスタやパーピン・テスタ，フル・パーピン・アーキテクチャに区分される．各アーキテクチャは各々の特徴がある．例えば，シェアード・リソース・テスタは過去から使われており，安価であるのが特徴である．パーピン・テスタは設計でのデバックを容易にするために，各ピン・エレクトロニクスにタイミング発生器を搭載させた．それを進めたのが，フル・パーピン・アーキテクチャである．テスタの活用においては，その構造を理解してテストの実行を考慮しなければいけない．最近，テスト容易化設計(DFT)も活用され構造テストとして活用されている．その DFT 用の専用テスタも開発されており，デバック用では安価な装置も発表されている．しかし，量産では汎用半導体テスタと同じ価格帯のテスタとなり，テスト・コストが問題である．テスト・コストについては，VLSI の製品開発～量産フローまで見た対応が必要である．

そこでまず，**第 2 章**では，このフローを概観した．テストの工程は半導体設計の後工程と位置づけられ，その検討はテスタ上での実機デバックとなる．また，そこに使われるテスタの機種もさまざまなものが使われ，最終的に量産が可能になるには多くの工数がかかる．最近の設計技術として HDL 設計が使われ機能設計が取り込まれている．その検証はテスト・ベンチが使われている．テスタの機能を HDL 記述することによって，その機能設計での仮想テスタが実現され，テスト上の問題が解決されると考えられる．

第 3 章で，仮想テスタの応用で，テスタ構造を HDL 化することにより，低消費電力基板型再構成テスタの開発をした．これは基板上の FPGA にテスタ構造を HDL 化して論理構成を行い，テスタを構成した．その構成方式には，必要なテスタ機能構成のみを搭載したので，FPGA 1 個に搭載でき，構造可変テスタ手法が開発できた．また，消費電力も大幅に削減される低コストのテスタが構成でき，TOB-Ⅰとして開発した．その応用として，フラッシュ・メモリのテスト装置に使った．

次に，TOB-Ⅰの改良として，機能追加をして TOB-Ⅱを開発した．これにはパターン・メモリの拡張やピン・エレクトロニクスの追加，高精度 DC 計測器の追加で，VLSI テス

トに使われる可能性の高いものにした．その応用して HDD モータ駆動コンボ IC に活用した．その基本的テストは活用できる見通しを得た．

第 4 章では，テスタ･メーカ各社のテスタのテスタ言語を調査して，テスタ構造表現言語を提案した．

テスタでのテスタ言語は，そのテスタの構造を示すために，ステートメント方式が一般的である．しかし，その記述は各社により異なっている．各テスタでの個別テスト・プログラム作成が一般的である．テスタの有効活用をするために，テスタ間でのテスト・プログラム変換も使われるが，対象機種が多くなるとそのテスト・プログラム変換の開発が煩雑になる．最近では，テストの各種パラメータをデータ・ベース化して，そのデータ・ベースを通じてデータ・ベース変換も行われてきた．しかし，テスト・プログラム記述自体は各テスタでの記述によるために，一貫したテストの合理化ができない．それに対して，C 言語の関数記述によるテスト・プログラム記述を提案した．これを GTL と呼ぶ．GTL ではテスタの基本的なハードウェア（テスタ・リソース）を記述するために，テスタ構造表現言語とも呼ぶ．各ハードウェアの関数記述に対して関数定義を持つことで，各種テスタ上でのテスト・プログラム実行を可能にした．また，その形式でのテスト時間の長大化はなく，通常のテスト・プログラム記述と同じように動作を確認した．

テスタ構造表現言語 GTL は，汎用半導体テスタが持つ基本的なテスタ・リソースを表現している．このことの応用として，GTL で記述されたテスト・プログラムから，それが必要としているテスタ・リソースを抽出して，適用可能なテスタのリスト・アップをするテスタ照会選択機能を持たせることができる．このことは，インターネットで GTL を公開しテスト・プログラムを作成させ，その上でのテスタ選択を行わせ，最適テスタの照会ができるビジネス・モデルが提案できた．

第 5 章では，**第 3 章**で述べた FPGA によるテスタ構造可変テスタの VLSI チップ上での実現として，SRAM ブロックを使った論理構成の考察をした．FPGA は半導体であり，VLSI チップに搭載可能であるが，高度なプロセスを使うことや，FPGA メーカの知財に関わることから，その SoC 搭載が困難である．そこで最近の VLSI に多く搭載されている SRAM に着目して，その SRAM 上での論理動作の構築を考察した．

その結果，アドレス・データ対を持つ SRAM ブロックの交互配置で，論理機能も配線機能も持たせられる構造の提案をした．これは，SRAM のメモリ・セル構造を大きく変えることなくチップ上に搭載できる可能性を持つので,その手法について調査研究した．特に，アドレス・データ対の数により，現在の FPGA に近いメモリ容量で実現できることが分かった．そして，この手法で各種演算器の実現を確認した．半導体の製品状態に併せてテストしなければならない時代でのテストの応用の機会になればと考える．

以上，汎用半導体テスタの研究に基づき，低価格な構造可変テスタの開発や，テスタ構造表現言語での最適テスタ照会の応用，VLSI にテスト回路構成可能な SRAM ブロックによる一論理構成手法の探索を行ってきた．

今後の課題として,これら技術の産業界の実用化として研究を進めていく必要がある．

謝辞

本研究を進めるにあたり，終始御指導，御教鞭いただきました東京都立大学大学院工学研究科岩崎 一彦教授, 福本 聡准教授, 三浦 幸也准教授に厚く御礼申し上げます.

私は，平成 14 年度より岩崎 一彦先生のもとで LSI のテストに関する基礎教育を受けてきました．社会人学生として勤務しながらの研究でしたが，電子メール，FAX などを活用して頂き，研究室での御指導はもとより非常に多くの御指導をして頂きました．特に，私の専門分野である汎用半導体テスタの技術的見解や考え方については，学術的な位置づけをご指導いただきました．また，研究会の参加を通して，研究者としての見聞を広めて下さいました．ここに深く感謝申し上げます.

また，本研究を遂行するにあたり，各テスタ・メーカのご協力に感謝いたします．特に，在籍していたイノテック株式会社およびジェネシス・テクノロジー株式会社の皆様には感謝申し上げます．本研究に関して有益な御討論をして頂きました，東京都立大学大学院工学研究科情報工学講座および卒業生の皆様に感謝致します.

SRAM ブロックによる論理回路の構成の研究には，その試作をいただいた広島市立大学情報科学部情報工学科 弘中 哲夫教授に感謝いたします.

企業在職中には，当時の上司である日立製作所半導体事業部技術開発本部本部長下東勝博様(現在，半導体理工学センタ社長)には，一貫して研究に取り組む姿勢や考え方学ばせて頂きました．ここに記して謝意を示します.

そして，現職を続けながらの研究活動に対して，深い理解と協力を頂いた東海大学浅川 毅准教授に感謝致します. 現在在職している太陽誘電株式会社では，関係各位のご支援をいただき，ここに感謝いたします.

最後に, 影ながら支えてくれた両親, 妻, 子供達に感謝の意を表して謝辞を終えます.

研究業績一覧

1. 学術論文（査読あり）

[1] 佐藤正幸, 大塚信行, 武藤治, 新井雅之, 福本聡, 岩崎一彦, 上原孝二, 志水勲, 間明田治佳, “低消費電力型再構成テスタの開発,” 信学論D-Ⅰ,Vol. J88-D-Ⅰ, No. 6, pp. 1065-1075 , Jun. 2005.

[2] M. Sato, H. Wakamatsu, M. Arai, K. Ichino, K. Iwasaki, and T. Asakawa, “Tester Structure Expression Language and its Application to the Environment for VLSI Tester Program Development,” Journal of Information Processing Systems, Vol.4, No.4, pp. 121-132, Dec. 2008.

2. 国際会議学術論文（審査あり）

[1] R. Yamada, Y. Ichinoseki, K. Ichino, S .Fukumoto, K. Iwasaki, and M.Sato, “Memory-Based Reconfigurable Chip Architecture and Its RT-Level BIST,” Workshop on RTL ATPG & DFT , pp. 38-43, Nov. 2001.

[2] Y. Yamagata, K. Ichino, M. Arai, S. Fukumoto, K. Iwasaki, M. Sato, H. Itabashi, T. Murai, N. Otsuka, “Implementation of Memory Tester Consisting of SRAM-Based Reconfigurable Cells,” Asian Test Symposium, pp. 28-31, Nov. 2003.

[3] M. Sato, N. Otsuka, O. Muto, M. Arai, S. Fukumoto, K. Iwasaki, K. Uehara, I. Shimizu, H. Mamyouda, “Development of Low-power Board-mounted Reconfigurable Tester,” Workshop on RTL and High Level Testing, pp. 137-142, Nov. 2004.

[4] M. Sato, H. Wakamatsu, M. Arai, K. Ichino, K. Iwasaki, T. Asakawa, “Tester Structure Expression Language and Its Application to Tester Selection,” Workshop on RTL and High Level Testing, pp. 145-150, Oct. 2007.

[5] N .Hirakawa, M.Yoshihara, M.Sato, K. Tanigawa, T.Hironaka, “Low Cost PLD with High Speed Partial Reconfiguration,” International Technical Conference on Circuit/Systems, Computers and Communications, pp. 557-560, Jul. 2008.

3. 国内発表

3.1 研究会

[1] 佐藤正幸，“メモリ仮想テスタ技術と今後の課題,” 信学技報,フォールトトレラントシステム研究会, FTS98-121,pp. 33-39, Feb. 1999.

[2] 佐藤正幸， 大塚信行，武藤治，新井雅之，福本聡，岩崎一彦，上原孝二，志水勲，“低消費電力基板型構造可変テスタの開発,” 信学技報，ディペンダルコンピューティング研究会, DC2003-94,pp. 23-28, Feb. 2004.

[3] 佐藤正幸，若松弘樹，“SRAM ブロックを用いた論理回路の一構成手法,” 信学技報，リコフィギャラブルシステム研究会, RECONF2006-23,pp. 17-22, Sep. 2006.

[4] 吉原理記，弘中哲夫，佐藤正幸，“再構成デバイスとしても動作するメモリ LSI の検討,” 信学技報，リコフィギャラブルシステム研究会, RECONF2006-24,pp. 23-28, Sep. 2006.

[5] 土屋秀和，浅川毅，佐藤正幸，“LSI テスタとデバイス間の伝送経路の評価とシミュレーションモデル化について,” 信学技報，信頼性研究会, R2007-34,pp. 29-34, Sep. 2007.

[6] 吉原理記，平川真樹，谷川一哉，弘中哲夫，佐藤正幸，“再構成デバイスとしても動作するメモリ(MPLD)の一実装例,” 信学技報，リコフィギャラブルシステム研究会，RECONF2007-16, pp.7-12 , Sep. 2007.

[7] 小田祐太郎，谷川一哉，弘中哲夫，平川直樹，戸口博昭，佐藤正幸，“再構成デバイス MPLD への組み合わせ回路マッピング手法の検討,” 信学技報，リコフィギャラブルシステム研究会, RECONF2008-37,pp. 87-92, Sep. 2008.

[8] 弘中哲夫，平川直樹，吉原理記，谷川一哉，“ハイパフォーマンスコンピューテイングを目指した MPLD アーキテクチャの検討,” 信学技報，コンピュータシステム研究会, CPSY2008-31,pp. 13-18, Oct. 2008.

3.2 大会発表

[1] 山形優輝，市野憲一，新井雅之，福本聡，岩崎一彦，佐藤正幸，板橋裕行，村井崇，大塚信行，“SRAM を用いた可変論理セルで構成したメモリテスタの実装,” 電子情報通信学会総合大会,

D-10-7, pp.162, Sep. 2003.

[2] 佐藤正幸，若松弘樹，岩崎一彦，“SRAM ブロックを用いた論理回路の一構成手法,” 電子情報通信学会総合大会, D-10-9,pp.118, Mar. 2006.

3.3 その他

[1] 浅川毅，山田力大，市野憲一，福本聡，岩崎一彦，佐藤正幸，“メモリチップのみを用いた論理回路の設計,” 第 43 回 FTC 研究会,セッション 9, Jul. 2000.

[2] 佐藤正幸，栗田浩，“メモリ仮想テスタ技術と今後の展開,” 第 44 回 FTC 研究会, セッション 2, Jan. 2001.

[3] 山田力大，浅川毅，市野憲一，福本聡，岩崎一彦，佐藤正幸，“メモリチップのみを用いたメモリテストを目指して,” 第 44 回 FTC 研究会, セッション 2, Jan. 2001.

[4] Y.Yamagata, K.Ichino, M. Arai, S. Fukumoto, K. Iwasaki, M. Sato, H. Itabashi, T. Murai, and N. Otsuka, “Implementation of Memory Tester Cosisting of SRAM-Based Reconfigurable Cells,” Internatinal Symposium on Low-Power and High-Speed Chips, p. 80, Apr. 2003.

[5] 佐藤正幸，大塚信行，小林康郎，武藤治，“低消費電力基板型構造可変テスタ,” 第49回FTC研究会, セッション7, Jul. 2003.

[6] 佐藤正幸，若松弘樹，岩崎一彦，“テスタ言語の調査とテスタ構造表現言語,” SEMI テクノロジーシンポシウム(STS) , セッション 4, pp. 25-31, Dec. 2006.

[7] 佐藤正幸，若松弘樹，新井雅之，市野憲一，岩崎一彦，浅川毅，“テスタ構造表現言語の提案とテスタ選択ツールへの応用,” 第 57 回 FTC 研究会, セッション 4, Jul. 2007.

4. 商業雑誌

[1] 佐藤正幸，“Virtual Tester 時代の到来,” 電子ジャーナル，p. 13, 1996. 7 月号.

[2] 佐藤正幸，“第 12 章　テスティング技術,” 1998 半導体テクノロジー大全, pp. 210-214, 1997.

[3] 佐藤正幸，“第 14 章　テスティング技術,” 1999 半導体テクノロジー大全, pp. 231-237, 1998.

[4] 佐藤正幸，“21 世紀型テスティングの胎動,” 電子ジャーナル, p. 40，2001. 3 月号.

5. 特許

5.1 登録特許

[1] 佐藤正幸，内山邦男，“半導体集積回路装置および製造方法,” 登録番号特許 3980827, 登録日 2007 年 7 月 6 日，特願 2000-364005，出願日　2000 年 11 月 30 日.

[2] 志水勲，佐藤正幸，“半導体集積回路デバイスの開発方法,” 登録番号特許 3971104, 登録日 2007 年 6 月 15 日，特願 2000-539433，出願日 1997 年 12 月 16 日.

[3] 佐藤正幸，志水勲，吹上寛，“半導体集積回路およびメモリの検査方法,” 登録番号特許 3867882, 登録日 2006 年 10 月 20 日，特願平 10-543742，出願日　1998 年 4 月 16 日.

[4] 石原和子，石川誠二，下社貞夫，中里純，松岡一彦，宮本佳幸，鳴島正親，宮崎功，執行義春，佐藤正幸，大嶋孝幸，“検査システム、解析ユニット及び電子デバイスの製造方法,” 登録番号特許 3572626, 登録日　2004 年 7 月 9 日, 特願 2000-10131, 出願日 1993 年 3 月 4 日.

[5] 佐藤正幸，大嶋孝幸，“半導体メモリ不良解析システムと不良セル表示出力方法,” 登録番号特許 3256555，登録日 2001 年 11 月 30 日，特願平 4-289477，出願日 1991 年 3 月 18 日.

[6] 佐藤正幸，志水勲，吹上寛，“テストシステムおよび半導体集積回路装置の製造,” 米国特許登録番号 USP6,400,1736，登録日 June 4, 2002，特願 2000-264193，出願日 2000 年 8 月 31 日.

5.2 特許出願

[1] 佐藤正幸，“半導体集積回路,” 特願 2005-504364, 2003 年 7 月 16 日.

[2] 佐藤正幸，大塚信行，板橋裕行, “半導体装置の検査方法,” 特願 2003-63795, 2003 年 3 月 10 日.

[3] 佐藤正幸，大塚信行，板橋裕行，“半導体集積回路のテスト装置,” 特願 2003-170314, 2003 年 6 月 16 日.

[4] 佐藤正幸，志水勲，山田力大，市野憲一，淺川毅，福本聡，岩崎一彦，“テスト方法および半導体装置,” 特願 2001-351497, 2001 年 11 月 19 日.

[5] 佐藤正幸，“テスタ構築データの生成方法およびテスタの構築方法並びにテスト回路,”特願 2001-203776，2001 年 7 月 4 日.

[6] 高橋秀明，佐藤正幸，奈良孝，“半導体集積回路,” 特願 2000-302518，2000 年 10 月 2 日.

[7] 佐藤正幸，内山邦男，“半導体集積回路および設計方法並びに製造方法,” 特願 2000-117001, 2000 年 4 月 18 日.

[8] 佐藤正幸，志水勲，奈良孝，“半導体集積回路およびその検査方法並びに製造方法,” 特願平 11-258554，1999 年 9 月 13 日.

[9] 佐藤正幸, 志水勲, 高橋秀明, 工藤正明, “半導体集積回路および製造方法,” 特願平 11-122229, 1999 年 4 月 28 日.

[10] 佐藤正幸，志水勲，高橋秀明，“半導体集積回路およびそのテスト方法並びに製造方法,” 特願 2000-611314，1999 年 4 月 14 日.

[11] 佐藤正幸，大嶋孝幸，志水勲，高橋秀明，“半導体集積回路およびそれを用いた論理集積回路の設計方法,” 特願 2000-603090，1999 年 3 月 4 日.

[12] 佐藤正幸, 志水勲, 吹上寛, “半導体集積回路および論理回路の診断方法,” 特願 2000-524682, 1998 年 12 月 9 日.

[13] 佐藤正幸，志水勲，吹上寛, “検証用半導体集積回路および回路エミュレータ,” 特願平 10-194436，1998 年 7 月 9 日.

[14] 佐藤正幸，志水勲，吹上寛，“検証用半導体集積回路および回路シミュレータ並びに回路シミュレーション方法,” 特願平 11-502026，1998 年 1 月 14 日.

[15] 志水勲，佐藤正幸，“システムボードの開発方法,” 特願 2000-539434，1997 年 12 月 16 日.

[16] 佐藤正幸，志水勲，吹上寛，“検証用半導体集積回路および回路シミュレータ並びに回路シミュレーション方法,” 特願平 11-502022，1997 年 6 月 13 日.

[17] 佐藤正幸，山田靖，“テストシステム,” 特願平 6-116790，1994 年 5 月 30 日.

[18] 佐藤正幸，山田靖，“テストプログラム作成支援システム,” 特願平 5-104661, 1993 年 4 月 30 日.