

BINALY — 階層構造モデルを扱う論理シミュレータ

(白 木 丹 米 倉 秀 光 平 川 和 之
 (沖 電 気 工 業 株 式 会 社))

概要

LSI化装置用論理シミュレーションシステム、BINALY (Block Integrator and analyzer) は、装置設計者とLSI設計者との間で共同利用されることを前提としたシステムであり、そのねらいは、LSI技術の進歩による両者の作業分担の変動を、吸収することにある。

その主たる機能は、①装置論理検証、②LSI論理検証用データ出力、③LSI論理検証、④LSIマスク設計用データ出力、⑤LSI試験用データ出力、である。

又、その運用方法には、大量データの入力・運用手順の変動、に対応するため、各種の工夫がされている。

1. ま え が き

最新のデバイス技術を利用したLSIの設計に際しては、その使用者である装置設計者と、提供者であるLSI設計者とのインタフェースは、非常に重要である。しかし、両者の作業分担は、最適設計を行う必要上、開発装置・開発品種ごとに異なる場合が多い。従って、現段階では、標準化等の方法によりインタフェースを確立する事は困難である。そのうえ、両者の利用するEDAシステムは、発生経過上から、互いに独立に作成されたものである場合が多い。

これらの問題点を解決するため、

- (1) 装置設計サポート用に開発されたEDAシステムと、LSI設計サポート用に開発されたものとを接続し、単一のシステムとして機能させる。
 - (2) 装置設計者とLSI設計者との作業分担の不明確な、LSI論理設計過程に関するサポートシステムを、両者から利用可能な形態で提供する。
- ことが計画され、その一部として本システムが作成された。(図-1)

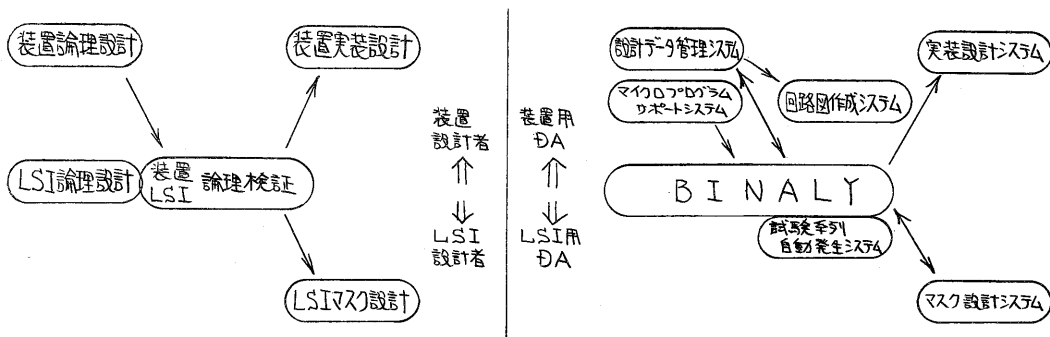


図-1 設計過程とEDAシステム

2. システム概要

システムは、単機能の多数のプログラム（プロセッサと呼ぶ）により構成される。又、システムの中央には、必要とする情報をすべて格納した、専用データベース（SMF）が位置する。プロセッサは通常、SMFと外界との間の情報の授受・SMF内部での情報の加工、のいづれかを行う。利用者は、これらのプロセッサを必要に応じて組合せて使用し、求める機能を実現する。

サポート機能 本システム作成の目的となった主要なサポート機能、および、それらの関係を、図-2に示す。もちろん、プロセッサの組合せ方法を変更することにより、ここに示した以外の使用方法も可能である。

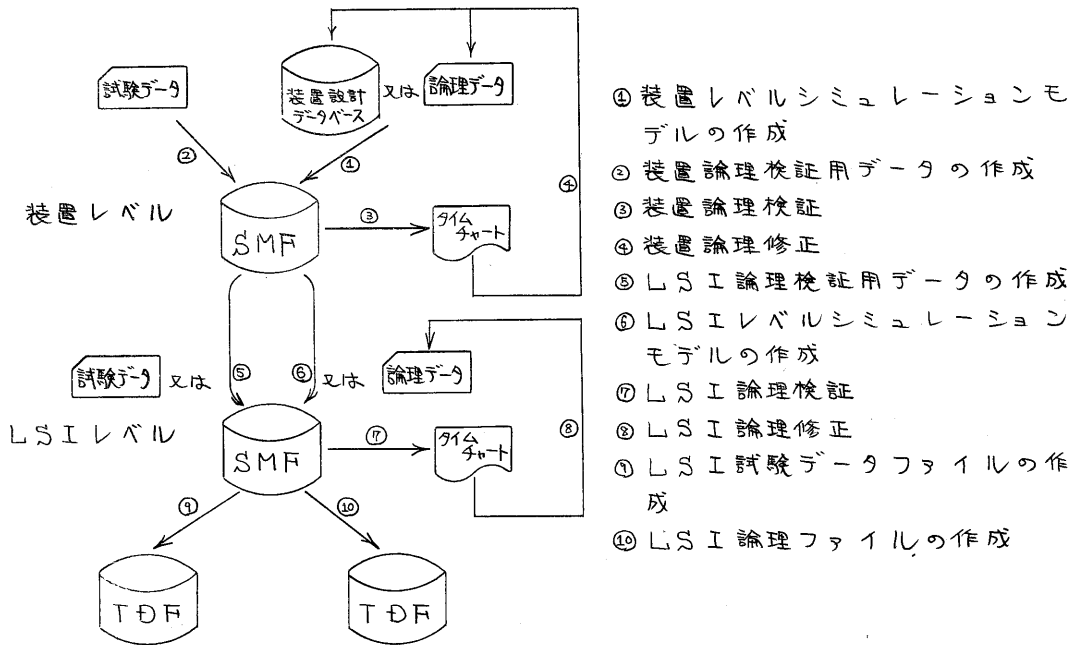


図-2 サポート機能

利用形態 システムは、現在大形計算機上に作成されており、通信回線を介して、BATCH・TSSの両形態で利用されている。関連他システムも、全て、同一計算機上に存在している。（図-3）

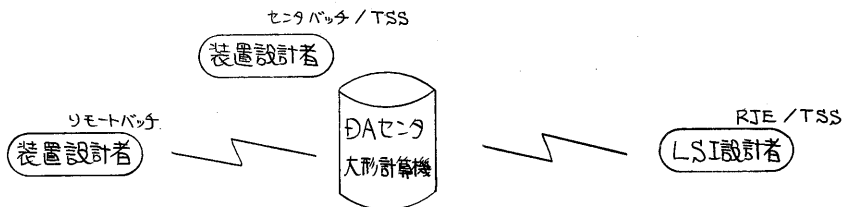


図-3 利用形態

3. 他システムとの関係

BINALYは、人手実装設計の論理装置等に、他システムとは全く独立なシミュレータとして使用出来る事も可能であるが、他システムと組合せて使用した場合に、最も効果を発揮する。すなわち、設計の初期の段階より本システムを使用し、順次詳細化を進めていく事により、設計の各レベルで検証が可能となり、最後には、LSIマスク設計・LSI試験用のデータが得られる。(図-4)

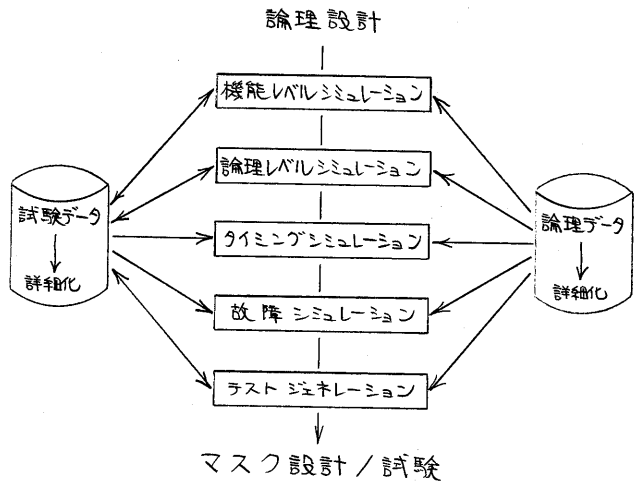


図-4 設計のレベルに応じた検証

装置設計用DAシステム BINALYは、その専用データベースであるSMFと、装置設計データベースとの間で、論理接続情報の授受を行う。その結果、設計中の装置に関する論理検証、本システムにより検証されたLSI情報の装置設計への利用、等が可能である。

装置設計データベース 装置設計データベース中では、装置は、その実装階層と対応する部品(ボード・LSI等)ごとの、論理実装情報の集まりにより表現されている。この階層構造の利用により、大規模回路の取扱いが可能である。

LSI設計用DAシステム BINALYは、そのSMFのサブセットである、TDF・LLBの2種のファイルにより、試験系列自動発生システム・LSIマスク設計システムの2者とインタフェースする。これら3者を総称して、特に、STARS-1と呼ぶ。(図-5)

試験系列自動発生システム このシステムは、BINALYと対になって働くシステムであり、故障シミュレーション、拡張アルゴリズムによる試験系列発生、及び、テスト対応の試験プログラムへの変換を行う。従って、最終的にテストで使用される系列中には、BINALYによって確認された人手発生の試験系列と、このシステムにより自動発生された試験系列とが混在する。

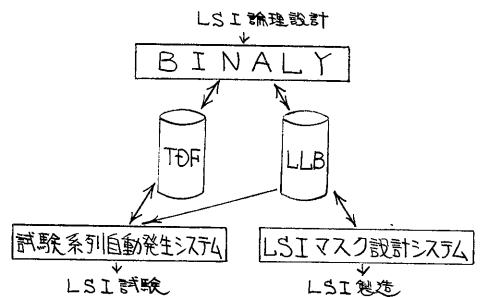


図-5 STARS-1

人手試験系列 人手により作成した試験系列は、パターン当りの検出故障数は低い場合が多いが、故障を検出した場合の個所の指摘が容易である。これは、新規開発コードの試作試験時には、大きな長所となる。

又、対象回路が複雑な順序回路の場合、拡張型アルゴリズムによる系列発生は、長時間の処理を必要とするため、論理検証時に作成したデータを流用する事は、コスト的にも有効である。

自動発生試験系列 上記の試験系列により、通常90%以上の故障検出率が期待できる。従って残りは数%となるため、数Kゲート程度のLSIでは、アルゴリズムでも十分有効である。

4. モデルの構造

BINARYでは、大規模回路の取扱いを容易(モデル作成時・シミュレーション実行時)にするため、各種の工夫がされている。それらの中で、最も本システムを特徴づけているものが、モデルのブロック化による階層構造表現である。

ブロック化 モデルを表現するためのデータの論理的なまとまりを、ブロックと呼び、モデルは通常、複数のブロックの集まりにより構成される。各々のブロックには、それらを識別するための記号(ブロックコード)が付与される。ブロックは、その内部の表現方法の相違により、3種に分類される。

組合せブロック その内部が、さらに他の複数のブロック(下位ブロック)と、その接続関係により表現されている場合をいう。全ての下位ブロックには、その内部を表現している他のブロックとの対応を示すため、必ずブロックコードが指定される。

基本ブロック その内部が、システムに登録済の論理演算ルーチンとの対応関係により表現されている場合をいう。システムには、現在、約60種類の論理演算ルーチンが登録されており、それをさらに追加することも容易である。

機能定義ブロック その内部が、メモリ・タイミン記述・IF-THEN-ELSE・DO-WHILE・代入文等の専用言語により表現されている場合をいう。これらの記述は、インタープリティブに実行される。

階層構造表現 モデルは、ある指定されたブロックより、組合せブロックを順にたどり、下位ブロックのブロックコードに対応する他のブロックを、モデルに結合させていく事により作成される。すなわち、并称回路に、論理的な階層構造が存在する場合、その階層ごとにブロックを作成すれば、モデルも同一構造に作成される。

リエントラント構造 モデル中では、同一ブロックに関する論理接続情報は、ただ1つしか存在しない。すなわち、ブロックをモデルに結合させる場合に、そのブロックが、既にモデル中の他の個所にあつたものであれば、シミュレーション実行時の状態領域の追加のみが行われる。(図-6)

従って、同一ブロックの使用頻度が大である程、取扱可能な回路規模は増大する。

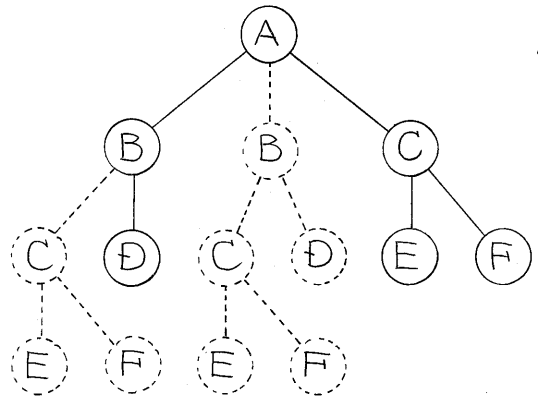


図-6 階層構造モデル

オーバーレイ構造 モデル作成時に、その階層構造を利用して、全モデルを複数のセグメントに分割(最小単位はブロック)し、それぞれの構造を定義しておけば、シミュレーション実行時には、個々のセグメントは磁気ディスク上に保存され、必要に応じて主記憶装置上にオーバーレイしながらロードされる。モデルの構造によっては、対象回路の論理構造を考慮しない単純なページング方式よりも、パフォーマンスが向上する。

パフォーマンス低下への配慮 ブロックは、通常列々に作成されるため、あるブロックは、その下位ブロック(のブロックコード)に対応するブロックの内部とは無関係である。しかし、ブロックの作成時に、参照すべき他のブロックを特に指定する事により、従来の階層関係を展開した場合とほぼ同様のデータ構造が作成される。シミュレーションのパフォーマンス上特に影響が大きいと思われる部分をこの方法で作成する事により、処理速度の著しい低下は避けられる。

5. モデルの作成

設計フローのバリエーションに応じた運用形態を可能にするため、各種の方法が提供されている。(図-7)

装置モデル 装置設計データベース中の論理情報のみを抽出し、1部品を1ブロック(組合せブロック)として作成する。従って、装置モデルは、通常、実装階層と同一の階層構造となり、シミュレーション結果の解析等も容易である。

又、特に指定すれば、任意の部品を、別途作成したブロックと置き換えることが可能であり、設計の詳細度に応じたモデルも作成できる。

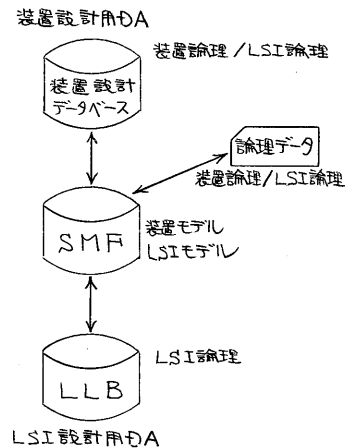


図-7 モデル作成

LSIモデル システムに用意されている、いずれの方法によっても作成できる。

- ④ 装置設計データベース中のLSIに対応するブロックをモデルとする。装置設計者が、装置とLSIの論理設計を同時に進める場合に利用される。
- ⑤ 装置設計言語のサブセットである、BINARY用論理記述言語(フリーフォーマット)を利用して、直接ブロックを作成する。マクロ・くり返し記法等が許され、大規模回路の場合でも比較的容易に記述できる。
- ⑥ LSI単位の論理情報であるLLBより作成する。LSIマスク設計の結果が追加されたものを入力して、タイミング解析等を実行する場合に利用される。

LLB LSIモデルを装置設計データベースのサブセットの形式に変換したものをLLBと呼ぶ。装置設計用DAシステム・本システム・LSI設計用DAシステムのいずれによってもアクセス可能である。LSIマスク設計の結果、ディレイ等の情報が付加される場合もある。

機能ブロック LLB中の最小要素を特に機能ブロックと呼ぶ。機能ブロックは、LSI論理設計者(装置設計者又はLSI設計者)と、LSIマスク設計者との間で、必要に応じてとりきめられ、LSIマスク設計システムにおける配置・配線の最小取扱い単位となる。

6. 試験データの作成

シミュレーション実行の制御は、入力端子に設定される信号の変化(セット文)等を記述したデータ(試験データ)と、実行時間・結果の表示方法等を指定した制御データにより行われる。試験データは一般に大量データであるため、その作成を容易にする各種の工夫が行われている。

テスト文 試験データ中には、出力端子に対する期待値(テスト文)を記述できる。シミュレーション実行時に不一致が生じた場合、タイムチャートとは別途メッセージが出力されるため、結果の確認が容易である。

観測端子群 似通った意味を持つ複数個の端子(観測端子群)に対し、あらかじめ名称を付与しておく事により、一括して、パターン・セット/テストを指定する事が可能である。この利用により、設計用の各種ドキュメント(動作概要・タイムチャート等)から試験データを作成する事が容易である。

タイミング記述 試験データ中には、文と文の間に、実行時間で時間差を表現できる。又、クロック信号(周期的な変化)があらかじめ定義されている場合には、それらを基準とした時刻表現も可能である。これらの工夫により、同期信号・非同期信号のいずれも簡単に記述できる。

マクロ展開 一連の文を、試験データとは別にあらかじめ登録しておき(マクロ)、試験データの記述時には、それを名称で参照しながら、部分的に変更して展開する事ができる。指定の方法によっては、変更方法をさらに規則的に変化させながら、複数個展開することも可能である。マクロのネスティングレベルに制限はない。マクロの記法は、モデル作成時に使用されるものと、全く同一である。

他に、四則演算式等を文中に記述する事も可能であるが、これらも、試験データのチェック時に演算が行われ、その結果に置き換えられる。

TDF 試験データは、必要に応じて、全入出力端子に対するセット文とテスト文のくり返しよりなる、中間形式(TDF)に変換される。TDFは、試験系列自動発生システムの入力となり、故障シミュレーションの後、拡張アルゴリズムによるパターンの追加が行われる。もちろん、BINARY自身に再入力する事も可能である。

TDF編集 TDFは、シミュレーション実行結果から編集する事も可能である。従って、セット文のみを作成してシミュレーションを実行した結果や、装置レベルで実行した結果より、LSI試験用のTDFを作成する事も可能である。

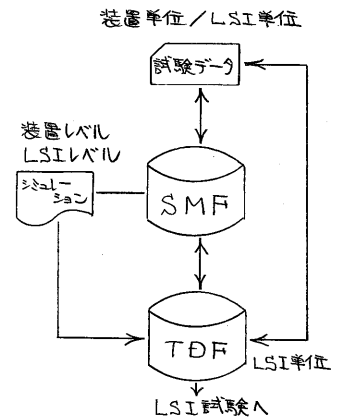


図-8 TDF作成

7. 論理検証

BINARYでは、設計の詳細度に応じたレベルでの検証が可能である。実行可能なレベルは、機能レベル・論理レベル(2値・多値)・タイミングレベルである。機能レベルと論理レベルは混在が可能である。

機能レベルの混在 モデルを構成する特定のブロックを、前述の機能定義ブロックと置き換える事により、部分的に機能レベルの混在した形式でシミュレーションが実行可能である。この機能は、詳細設計が終了して以降段階でシミュレーションを実行したり、論理検証の対象外の部分を、高速でシミュレートする場合に利用される。

詳細ディレイ 取扱可能なディレイは、次の3種である。

- (1) ブロックの種類(ブロックコード)に依存するディレイであり、出力端子対応に設定される。NOMINAL値を使用する場合と、立上り・立下りのMIN・MAXを使用する場合とがある。
- (2) ネットのファンアウト先(負荷)に依存するディレイであり、ファンアウト源に対し設定される。
- (3) ネットの配線に依存するディレイであり、ファンアウト先に設定される。

2値レベル 状態値は、2値で表現されるが、テストケースの平行度が高いため、処理が高速である。

多値レベル 状態値は、多値で表現されるため、平行度が低く、処理は低速であるが、ハガード等のチェックが容易である。

実行時レベル指定 上記のレベルの選択は、シミュレーション実行時の指定のみで可能であり、モデルの再作成等は不要である。

簡略化タイムチャート シミュレーション結果は、ラインプリンタ上に、横軸一観測端子群、縦軸一時間の形式で出力される。状態値は、端子対応に文字で表示されるが、1観測端子群の全ての状態が変化しなかった場合の空白化、全観測端子群が変化しなかった場合の行の削除、等がおこなわれるため、大量データの場合でも解析が容易である。

又、ミスマッチが発生した場合、タイムチャート中の該當時刻の個所にも、その結果が表示される。

ステートリスト 指定された部分のみの状態値を、時刻の順に表示する事も可能である。

シミュレーション再開 シミュレーション実行前後のモデルの全状態がファイル中に保存されるため、エラー終了時の修正ラン・長いテストケースの場合の分割ラン等が容易である。

観測端子群追加 論理ミスが検出された場合の原因追求のため、前回実行結果に対し観測端子群を追加してタイムチャートを出力する事が可能である。

8. 運用簡略化

本システムは、運用のバリエーションに対応するため、多数の機能を提供している。従って、従来の方法では十分な活用は困難である。そのため、運用手順を簡略化する工夫も行われている。

JCLの廃止 運用者の入力したデータは、一度プリアンプにより処理され、必要なJCL(ジョブ制御言語)が追加された後に実行される。従って、通常の運用方法では、その使用する機能の多少にかかわらず、必要なJCLの数は、3〜4スタートメントである。

運用データの保存 運用者がシステムに対して入力したデータは、全てファイル中に名称が付与されて保存される。2度目以降に実行する場合には、その名称と変更部分のみを指定すればよい。もちろん、変更結果も別途保存される。

専用データベースの導入 運用の結果作成されたデータは、全てSMF中に保存される。大部分のプログラムはSMFに直接アクセスするが、シミュレーション実行時は、処理速度を重視し、必要なデータ類を専用ファイル上に転写した後、使用する。

システムライブラリ SMF中の内容で共通性の高いデータは、システム管理者により、システムライブラリ中へ転写される。システムライブラリは、全ての運用者が参照可能である。

ファイル管理 上記ファイルは、運用の開始と同時に、システムにより確保される。データファイルは運用者ごとに、又、SMFはプロジェクト(同一システムに関連した運用者の乗り)ごとに1ファイル存在する。これにより、運用者の入力したデータは完全にプロテクトされ、運用の結果作成されたデータのみが、同一プロジェクト内で共用される。(図-9)

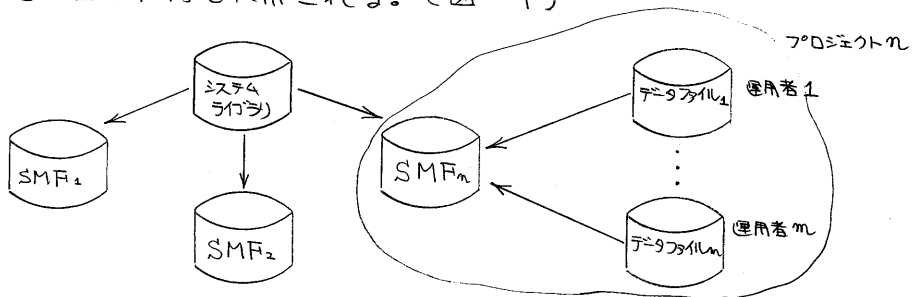


図-9 ファイル構成

9. 結果

論理レベルシミュレーションが可能となった段階で、LSI単体レベルのシミュレーションが運用された。その際の運用フローを図-10に、又、運用結果を表-1に示す。尚、表-1中の故障検出率は、試験系列自動発生システム中の故障シミュレータによるものであるが、人手入力パターンのみに対する値であり、ジェネレーションによるものは含んでいない。

表-1 運用結果

項目	LSI1*	LSI2	LSI3	LSI4	LSI5
機能ブロック数	532	159	387	434	275
機能ブロック種類	9	4	10	8	8
試験系列数	5000	2000	1500	2000	1000
エラー検出数	0	1	4	4	0
故障検出率	-	-	97.7	98.1	94.6

*他システムにより論理シミュレーション済

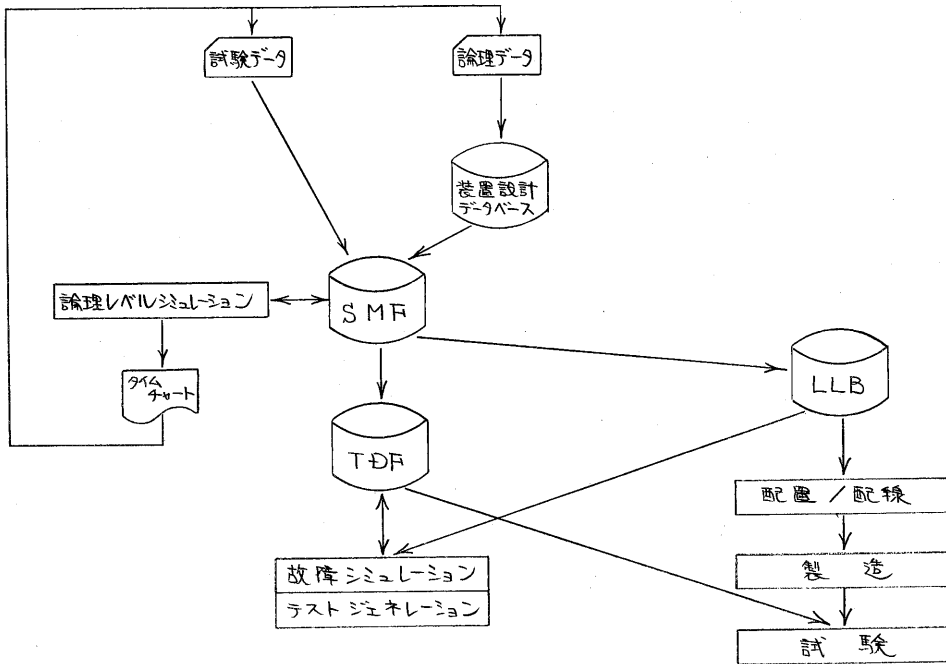


図-10 運用フロー

10. あとがき

今後は、大規模回路に関する運用データを収集し、より一層の、パフォーマンスの向上、設計期間短縮のための運用法の改善、を行う予定である。

謝辞 本システムの検討にあたって御指導頂いた、プロセッサ研究部上原部長、LSI開発部山川部長、なすびに関係各位に深謝いたします。

参考文献

- (1) 浜崎他、大規模階層構造モデルに関する論理シミュレーションの一手法
昭和54年度情報処理学会第20回全国大会 4B-7。
- (2) 菊地原他、LSI故障シミュレータ高速化の一手法
昭和54年度情報処理学会第20回全国大会 4B-8。