

解説



5. 最近の CAD システムの話題

5.2 設計データベース管理手法†

須藤 常太^{††} 唐津 修^{††} 永谷 三義^{††}

1. はじめに

近年、論理装置の各種設計工程（方式・機能・論理・回路・レイアウト設計およびテストパターン生成・故障診断等）の計算機による設計自動化、あるいは設計支援ツールの研究実用化が盛んである。これは設計対象とする論理装置の複雑大規模化、および LSI 化論理の要求する高い設計品質が人手設計では保証できなくなってきたというニーズ面の高まりと、計算機リソースのコスト・パフォーマンス向上というニーズ面でのサポートとの両面が急速に展開してきたためと考えられる。

歴史的にふりかえると論理装置の設計に計算機を利用しようという試みは、論理シミュレーションや、回路シミュレーションの実行、プリント基板等のマスクパターン作成時における図形処理装置の利用、設計図面の機械描画、部品リストの印刷出力といった個別のアプリケーションから手がけられてきた¹⁾。当初は使用できる計算機の能力的制約も大きく設計過程全体から見るとごく部分的な処理が行えるだけであり、トータルなシステムと呼ばれるものではなかったが、速度と正確さの点で大きなインパクトを与えた。

その後、設計システムの高機能化と統合化をめざし積極的な開発が行われており、最近では、設計データベースを中心とする総合設計システムの開発・運用についていくつか報告が行われるようになってきている²⁾⁻⁴⁾。

本稿では論理装置設計用のトータル・システム構築時に中心的役割をはたす設計データベースのデータ管理手法、各種設計過程で生じる各設計データを統合的に管理するための統合データベースの考え方、および

その応用技術を中心に述べ、リレーショナルモデル、分散データベース、知識ベース等設計データベースの将来動向についても概説する。

2. 設計システムにおけるデータ管理手法

2.1 LSI 化技術と設計システム

1960年代後半に TTL による SSI ファミリが市販され始めて以来、論理装置の実現法は、TTL や ECL の SSI を搭載したプリント板をバックボード上に並べて組み上げる手法が主として採用されてきた。ここ数年、LSI 技術の進歩により、LSI ベースの論理装置実現法が台頭してきており、装置構成技術の大きな転換点をむかえていると考えられる。装置設計をサポートする設計システムに対する要求条件は SSI ベースと LSI ベースで自ずと違いが表面化しており、設計システム開発の力点も 1970 年代の前半から半ばが SSI 中心であったのに対し、1970 年代後半からは LSI 中心へと移行してきている。

SSI ベースと LSI ベースの設計を比較した場合、ごく大雑把に言えば、前者がプリント板の設計で済ませられるのに対し、後者は装置設計者が行うか、部品供給側が行うかの別はともかく、LSI の中身の設計に介入せざるを得ない点に大きな違いがある。

したがって設計システムの性格は、プリント板の設計から部品である LSI そのものの設計へと対象が変わったことによる変化を反映しながら発展していると考えられる。

論理 LSI の設計を SSI ベースのプリント板設計と比較して考えた時、主な特徴は次の 5 点である。

(1) 従来は、SSI、プリント板、バックボードといった明確な実装上の階層があったが、LSI の中ではすべてトランジスタに展開されて混然となっている。

(2) LSI 製造技術の進歩により、従来に比べ桁違いに多くの論理を搭載できるようになっており、設計規模が爆発的に増大している。

† Design Database Management by Tsuneta SUDOU, Osamu KARATSU and Mitsuyoshi NAGATANI (Integrated Electronics Development Division, Atsugi Electrical Communication Laboratory, N. T. T.).

†† 日本電信電話公社厚木電気通信研究所集積回路研究部

(3) LSI 上では、論理構造とレイアウト上のトポロジとが密接な関係を持っており、設計者はこれらの条件を統合的に判断しながら作業を進める必要がある。

(4) 完成した LSI 上での修正作業が事実上不可能であるため、簡単に論理修正のできるプリント板設計に比べて極めて高い設計品質が要求される。

(5) LSI 技術が日進月歩で変化しているため、設計手法は柔軟に変化に追随する必要がある。

これらの技術上の違いは、以下に述べるような設計システム構成上の特徴となっておりあらわれている⁶⁾。

(1) 大規模化に対処するために、階層化・構造化設計を徹底させる。

(2) 各設計段階を一貫して表現できる共通設計記述言語、共通設計データベース等を採用する。

(3) 人為的な誤りを避けるため、自動設計処理を大幅に導入し、検証するためのフィードバックをきめ細かく用意する。

(4) 設計者が設計の進捗状態を的確に把握するために、使いやすいマンマシンインタフェースを採用する。

以下の節では、設計データベース構築上の手法について考察する。

2.2 論理装置設計システムにおけるデータベース構築手法

一般にデータベースを利用することの最大の利点はいわゆるデータ構造とプログラムとの独立性を確保することがあるが、設計システムの立場で整理すると次のようになる。

(1) LSI 設計時のデータの一貫性および修正時の時系列管理。

(2) ライブラリ提供による重複設計の回避、誤用の防止。

(3) 設計データの秘密保護と誤破壊の防止。

設計データベースをどのような形で構築し、アプリケーションプログラムとどのようにリンクさせるかという問題は、設計手法と設計記述言語、設計対象論理の規模等によって左右される。

1970年代の半ばに発表された設計データベースの例としてよく引用される Korenjak⁶⁾ のシステムは、その5年後に必ずしも成功しなかったとの報告が行われている⁷⁾。その理由は、データベースとプログラムとの関係を極めて密な形でシステムを構築したため、かえって身動きがとれず、その後の技術

進歩に十分追従しきれなかったためと総括されている。このことは設計の手法およびそれに従って改良されるアルゴリズムを実現するため、プログラムから見たデータベースが柔軟であるべき必要性を指摘したものである。

Korenjak とほぼ同時期に発表された設計データベースシステム^{8),9)} の多くは、電気回路等の接続関係をそのままネットワーク型データベースのリンク構造に反映させて設計されている。例えば Kawano らのシステムでは、論理データを分解して8個の個別ファイルの組合せに整理し、論理ゲートを表す「ELEM」や、接続線を表す「GNET」、「PNET」等を中心としてファイル間にネットワークリンクを定義してデータベースを構成している⁸⁾。

このような構成は、100 ゲート程度の設計データをデータベースへ入出力するために 0.3 MIPS 程度の計算機で 30~150 秒の CPU 時間を必要とすることが報告⁹⁾ されており、大規模データに対するパフォーマンス向上のための工夫が必要である。

Roberts の提案に代表される構成では図-1 のように汎用 DBMS とアプリケーションプログラムとの間にデータベースインタフェースプログラムをバッファとして介在させ、論理視野を提供しようとする考えである。この場合、データベースファイル上のデータは細かい論理構造をそのまま物理的構造として持つ必要がないので、ファイル構成は簡単化され、アクセスのパフォーマンス向上が期待される。

Roberts 流の構成が提案される背景としては、汎用の DBMS を設計データベースに流用した場合の限界によるところが大きい¹⁰⁾。汎用データベースでは、アクセスの主体は人間であり、一度にアクセスするデータ量は少なく、頻度もそう多くはない。このため、DBMS はいかに大量のデータの中から必要なレコー

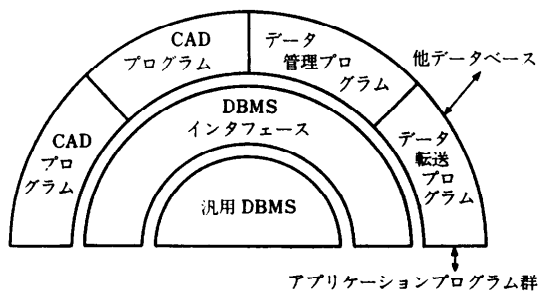


図-1 汎用 DBMS と DBMS インタフェースを組み合わせた設計データベースの構成例

ドを容易に検索してくるかという機能面に最大の重点が置かれて開発されている。設計データベースでは、アクセスの主体は設計プログラムであり、一度にアクセスするデータの量は数キロバイトから数十メガバイトに及ぶ場合もあり、大量の I/O が発生する。頻度もプログラムがフェッチするためトラヒック集中が起こるなど、汎用 DB と相当趣が違ふ。Roberts 流の考え方は、汎用の DBMS を工夫して使用することによりこの種の問題点のある程度回避できるという点で高い評価を得ている。

このような構成を採用した設計データベースでは、各設計データにいくつかの管理項目を付加して格納し、その後の管理運用は管理項目を通じておこなう方法が用いられている。管理項目としては、モジュール名、階層レベル名、使用目的、ライブラリ名、設計者名、プロジェクト名、設計年月日、修正年月日、版数等が考えられる¹⁴⁾。さらに抽象化した論理的機能(例えば NAND-4, SHIFTER-8, ALU-4)等を持たせられればなお良いと思われる。

一度定めたデータ形式は、LSI 技術の進展によって変更を受ける可能性も大きい。これに対処するために設計データベースでは、論理スキーマと物理的構造の間に一定の距離を置いてある程度の柔軟性を持たせておき、不測の変更に備える方法がとらえる。しかし、あまり距離を置き過ぎるとデータベースのアクセス・パフォーマンスはどんどん悪化し実用にならなくなる。なるべく単純な形で論理スキーマと物理構造を対応させておくのが望ましい。

2.3 各種設計工程とデータ管理手法

前節で引用した設計データベースは、回路・論理設計を主体としたものであり、論理装置の全設計工程に対して一律に適合するわけではない。

本節では、各種設計工程別の設計データ管理手法の現状について考察してみる。

現状における論理装置の各種設計工程とデータ管理手法の調査結果を図-2 に示す。ただし、一部結果の中には研究レベルのものも含まれていると思われる。

本結果は当然のことながら、各設計工程の CAD 化率を如実に反映している。論理設計・回路設計においては、設計の自動化はいまだ研究途上であるが、検証ツールとしての論理シミュレーション・回路シミュレーションの発達により、大多数が少なくとも計算機



サンプル数18(国内半導体メーカ, 研究機関, および大学)

図-2 現状における設計工程別データ管理手法の比較

上のデータとして管理しており、かつ半数以上がデータベースの管理手法を導入している。

レイアウト設計データに関しては、自動設計、人手設計を問わず、すべてが計算機データとして管理している。これは市販図形処理装置の普及、およびその後のマスクパターン生成過程を考えれば当然のことであろう。レイアウト設計工程は論理装置の全設計工程中、最も自動化が進んでいる分野であり、大型計算機上での処理の必要性から図形処理装置でのデータ管理のみでなく、自社独自のフォーマットあるいは市販図形処理装置のフォーマットに多少修正を加えた形のもの等、各社各様ではあるが、大型計算機上でのデータ管理を導入しているものと思われる。

3. データベースの統合化

論理装置の設計は、前述のようにシステム設計、機能設計、論理設計、回路設計、レイアウト設計、およびテストパターン設計等、各種設計工程を通じておこなわれる。しかし、各々の設計工程は独立したのではなく、たとえば高品質なレイアウト設計をおこなうためにレイアウト設計時に機能設計情報を参照する必要性も生じてくる。またプリント基板設計、LSI 設計を問わず、レイアウト後の配線遅延等を含めた遅延シミュレーションによる検証過程においては、レイアウト設計後のマスクパターンデータと論理設計データとの対応関係がデータベース上でとれている必要がある。ここでは、このような目的を達成するためのデータベースの統合化について考察する。

SSI ベースで論理装置を構成していた時代のデータベースでは、設計データの種類によって何種類かのデータベースを準備し、別々に管理運用している例が多い。前述した Kawano の例では、ネットワークを格納するイクイブメント・ファイル、SSI のデータを格

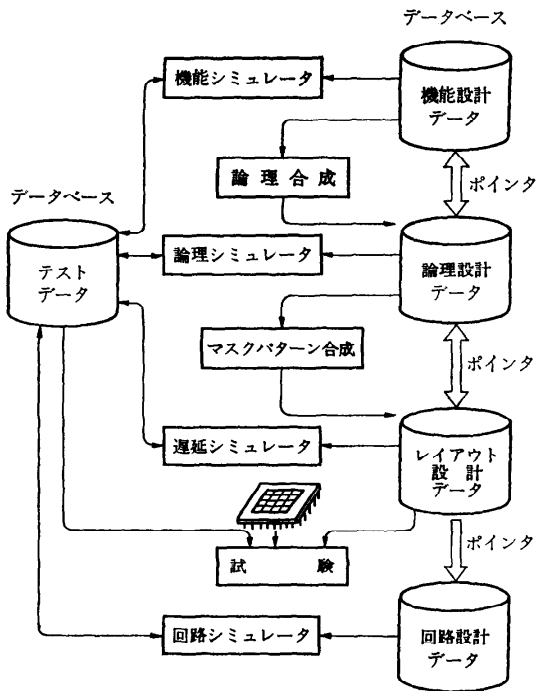


図-3 設計データベース統合化の概念図

納するモジュール・ライブラリ、設計基準等を格納するスタンダード・ライブラリ等を独立させている。論理装置の大規模化に対処するための階層的設計の導入、および LSI ベースの設計を考えると、これらは一元化しないと都合が悪い。

さらに、機能設計、論理設計、レイアウト設計等の各種設計工程に対応する各設計データも同一水準の管理状態下に格納し、かつ、各々の設計データ間に必要なポイントを張りめぐらしておく必要がある。このような手法により、例えば、レイアウト設計時に機能設計情報を参照したり、レイアウト設計後の配線遅延を含めた検証等がはじめて可能あるいは容易となる。

設計データベースの統合化の概念図を図-3 に示す。各種設計工程に対応する各設計データ間には必要に応じてポイントが張られる。例えば論理設計データとレイアウト設計データの間では、論理設計データ上の各ネットとそれを実現している図形要素との関係、論理設計データ上の各コンポーネントが配置されている座標の関係等を示すポイントである。この例ではこのポイントを張るのはマスクパターン合成プログラムである。このような統合設計データベースを運用するにあ

たっては、当然のことながら各設計工程が自動化までは至らなくても少なくともコンピュータによって支援されていることが必要である。

設計データベースの統合化における第一の問題点は一般的にデータ形式が異なる各設計データ間の効果的なポイントの張り方である。論理設計データ、回路設計データ、レイアウト設計データに関しては、電気的接続関係を表現したいわゆる構造記述データであり比較的データ構造が似通っており、それらの間の対応関係を持たせやすいが、機能設計データ、システム設計データ等抽象レベルが高くなるに従って明確な対応関係が取りにくくなる。

第二の問題点は、ある特定の設計データの修正にとまらう、他設計データとの一致管理である。例えば機能設計データを修正した時、対応する論理設計データは直ちには修正されず一定の時間遅れが生じることが予想される。この問題はデータベースの力だけで解決できる問題ではなく、適当な検証手段によって別途チェックする必要があるが、少なくとも対応関係が保証されなくなったという事実関係はデータベースが管理する必要がある。

これらの問題は、設計データベースに接続する各種設計自動化あるいは設計支援プログラムの構成の検討およびトータルな設計手法の検討を通じて実用的に解決していかざるをえないだろう。

4. 統合データベースの利用技術

ここでは、統合化された設計データベースの利用技術として、電電公社厚木電気通信研究所で開発・運用している2つの事例について紹介する。

【レイアウト設計結果に対する回路解析手法】

図-4 はレイアウト設計結果に対する配線遅延を含めた回路解析手法を示したものである¹²⁾。図の左側が処理の流れ、右側が各処理に対する結果を示している。LSI 全体を一度に回路解析することは回路解析プログラムの処理能力を超えるため、まず解析したい部分のみを切り出す処理を行う。このため(1)パス探索および(2)切り出しプログラムが使用される。パス探索は論理設計データを参照し、LSI 全体から設計者の指定した端子間の経路を見つけ、経路中のネット番号を切り出し情報として切り出しプログラムに渡す。切り出しプログラムは対応するレイアウト設計

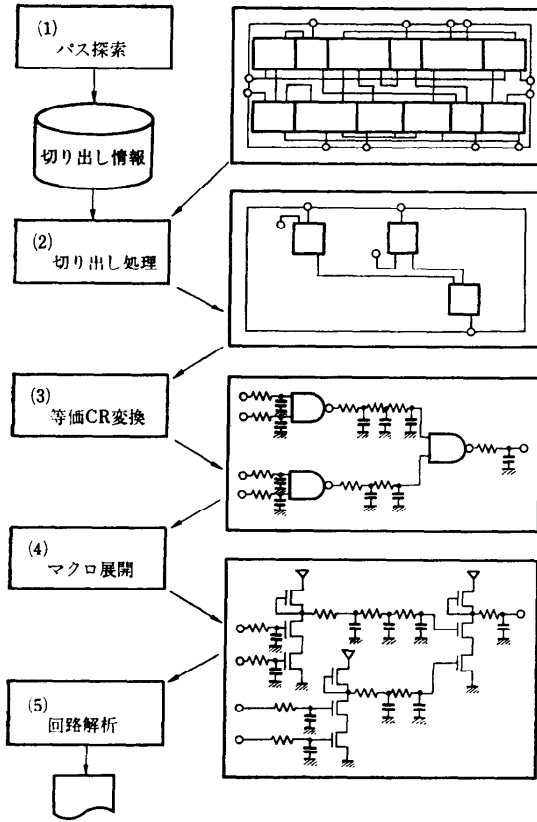


図-4 配線遅延を含めた回路解析手法

データを参照し必要な部分のみを切り出したデータ構造を作成する。次の(3)等価 CR 変換処理では、各ネットの配線を等価的な CR に変換し、等価 CR を含む論理接続データを作成する。さらに(4)マクロ展開処理で回路設計データを参照し各論理セルに対応する回路設計データをはめ込み、回路解析用のデータ構造を作成し、回路シミュレーションが実行される。

【電子ビームテスタを用いた故障診断手法】

統合化された設計データベースと電子ビームテスタを結合して LSI の故障解析を行うシステムを開発・運用している¹³⁾。

本システムは設計データベース内の論理設計データ、レイアウト設計データ、および論理シミュレーションから得られたテストデータと電子ビームテスタとを有機的に結合したものである。設計データベースからは LSI の時系列動作の各時刻における LSI チップ上の物理的位置に対応する電位の期待値が得られる。一方電子ビームテスタからはやはり LSI チップ

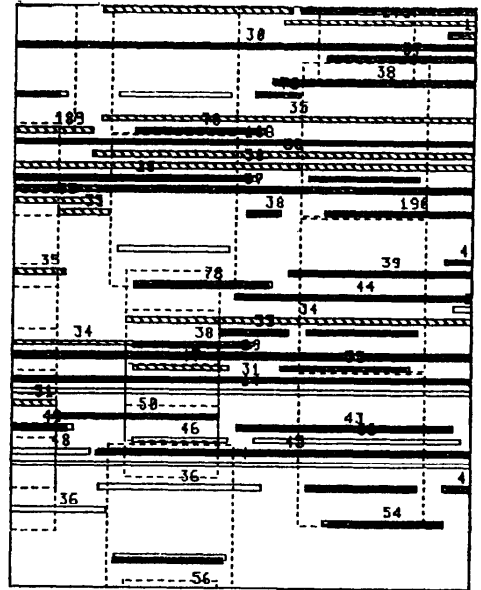


図-5 金属配線電位の期待値と実測値の比較結果

上の各金属配線の電位の実測値が得られる。この両者の結果を図形的に比較した結果を図-5 に示す。図中にぬりつぶされた線分は低電位で一致した個所であり、白ぬきの部分は高電位で一致した部分である。また斜線で示したところは期待値と実測値とが不一致であったことを示している。図中の数字は設計データベースから得られたその配線が属する論理的ネット番号を示しており、不一致を示しているネット番号を論理設計データ上で解析し故障箇所を発見する。

5. 将来動向と課題

設計データベースの構成手法、およびそれに要求される機能は、設計システムを構成するマシン環境および自動設計技術の発達そのものに大きく左右される。ここでは、今後のマシン環境、自動設計技術の発達の動向を踏まえた上で、設計データベース構成技術の将来動向と課題について考察する。

LSI 化技術の発達にささえられ、高機能・高性能なパソコン、いわゆるスーパーパソコンと呼ばれるものが普及してきている。これらの多くは高度なグラフィック処理機能および LAN (Local Area Network) に対する通信機能を備えており、設計システムの中で EWS (Engineer Work Station) として利用する試みが盛んである。これはとりもなおさず設計システムの

処理の分散化を意味するものであり、設計データベースに対する分散化の必要性も生じてくる。分散化された設計データベースの管理手法、および異なる OS (Operating System) からのアクセスを可能とする OS 依存度の低い設計データベースアクセス手法、同時並列に発生する設計データの更新に対する管理手法に対して、一層の工夫が要請されるものと思われる。

さらに記憶素子の低価格化、仮想記憶アーキテクチャの発達による大幅な記憶空間の増大にさえられ、Codd の提案したリレーショナルモデルを基本とする汎用リレーショナルデータベースの商品化が盛んである。この考えを設計データベースに応用し、設計データそれ自身をピンリスト、ネットリスト等のリレーションに分解して格納しようという試みもある¹⁴⁾。設計処理のアルゴリズム自体がリレーショナルモデルの概念と親和性が高く、将来の方向として有力な手法であると思われるが、現在の計算機のパフォーマンスを前提とすると実用レベルにするにはかなりの工夫が必要である。現状における設計データベースへのリレーショナルモデルの実用的かつ効果的な利用方法は、前述の統合化データベースにおける各設計データ間の関係を固定的なポインタではなくリレーショナル的に持たせる手法であろう。

一方、自動設計技術の発展の動向から鑑みると、機能設計、システム設計等、抽象レベルの高い設計データの格納方法、およびエキスパートシステム等、知的 CAD 向きの知識ベースの実現法が今後の課題となるであろう。特に知的 CAD を実現するには、従来の設計の FACT のみを格納した設計データベースのみでは不十分であり、設計のルール、設計の過程等設計知識を格納できる知識ベースの開発が必要であり、かつ知識ベースと従来の設計データベースとを有機的に結合したデータベースシステムの開発が要請されるものと思われる。

6. おわりに

論理・回路シミュレーション、プリント基板等のマスクパターン設計から始まった論理装置の CAD の研究・実用化は、今日では機能設計、あるいは論理の自動合成等の分野にまで広がり、論理装置設計の全工程をカバーする動きにある。さらに CAD を実現するためのファシリティについても汎用大型計算機のみでなくワークステーションや CAD 専用マシンの出現等により種々のものが利用可能となり、CAD シス

テムを実現するマシン構成自体も大規模・複雑になりつつある。

今後の論理装置の CAD は、利用可能な各種ファシリティを有機的に結合し、各々のファシリティに適した処理に分散するとともに、知識ベースを中心とする学習・推論機能等を取り入れたインテリジェントな設計システムを指向していくものと思われる。このような状況において各種設計データを統一的に管理するための設計データベースの役割はますます重要となっていくであろう。

最後に、図-2の現状における設計工程別データ管理手法の比較のデータは筆者らが新世代コンピュータ技術開発機構に属する委員会活動を通じて得たものであることを付言しておく。

参考文献

- 1) 樹下行三編：情報処理叢書 5, 論理装置の CAD, pp. 88-104, 情報処理学会 (1981).
- 2) Roberts, K. et al.: A Vertically Organized CAD Database, Proc. 18th DA Conf., pp. 595-602 (1981).
- 3) McWilliams, T. et al.: SCALD: Structured Computer Aided Logic Design, Proc. 15th DA Conf., pp. 271-277 (1981).
- 4) 唐津, 星野, 須藤: LSI 設計記述処理システム, 昭和 56 年度電子通信学会総合全国大会, S9-2 (1981).
- 5) 杉山, 唐津, 須藤: VLSI 設計システム, 情報処理学会電子装置設計技術研究会資料, 7-2 (1980).
- 6) Korenjak, A. et al.: An Integrated CAD Data Base System, Proc. 12th DA Workshop, pp. 399-406 (1975).
- 7) Rosenberg, L.: The Evolution of Design Automation to Meet the Challenges of VLSI, Proc. 17th DA Conf., pp. 3-11 (1980).
- 8) Kawano, I. et al.: The Design of a Data Base Organization for an Electronic Equipment DA System, Proc. 15th DA Conf., pp. 167-175 (1978).
- 9) Wong, S. et al.: A Computer-Aided Design Data Base, Proc. 16th DA Conf., pp. 398-402 (1979).
- 10) Sidle, T.: Weaknesses of Commercial Data Base Management Systems in Engineering Applications, Proc. 17th DA Conf., pp. 57-61 (1980).
- 11) 須藤, 杉山: カスタム VLSI 設計システム, 情報処理, Vol. 22, No. 8, pp. 791-796 (1981).
- 12) 永谷, 星野, 渡辺: VLSI 設計データベースと

- レイアウト情報処理プログラム, 昭和58年度電子通信学会総合全国大会, 440 (1983).
- 13) 久慈, 玉真, 永谷: 故障診断用オンライン電子ビームテスタ, 昭和59年度電子通信学会総合全国大会, 423 (1984).
- 14) Haynie, M. N.: The Relational Data Model for Design Automation, Proc. 20th DA Conf., pp. 599-607 (1983).
- (昭和59年8月28日受付)
-