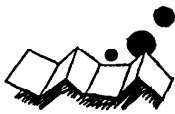


解説

論理装置 CAD におけるユーザ・インタフェース
の新動向†

杉 本 明††

1. はじめに

近年の半導体技術の進歩に伴い、論理装置は複雑・大規模化を続けている。このため、LSI 設計を中心とする論理装置 CAD 技術の研究・実用化が、各方面で盛んに行われている¹⁾。また実用化にあたっては、使いやすいユーザ・インタフェースの実現のため、種々の努力がなされてきた。そして、今日では新しい商用 CAD システムのほとんどが「ユーザ・フレンドリ」を特徴の一つとするにいたっている²⁾。

また、設計自動化技術の研究も進展してきた³⁾。たとえば、自動配置配線プログラムがレイアウト設計に使用されるようになった。さらに、機能レベルのハードウェア記述から LSI のレイアウト・マスク・パターンを自動生成するようなシリコン・コンパイラの研究も各所で行われている。

しかしながら、限定された問題に対しては、計算機が威力を発揮するものの、高品質の設計を得ようとするれば、人手に頼らざるをえない。これは、VLSI のように、高度に複雑なシステムを設計する場合、面積や速度の観点から最高の性能を追求するためには、熟練設計者の直感と創造的思考力を必要とするからである³⁾。

そこで、対話性の重視により設計者の知的作業を効率良く支援することを目的とした CAD システムの開発が、最近活発になっている。ここでは、このような論理装置 CAD における対話性向上の動きについて、開発者の立場から概観する。特に、CAD ツールの統合化とグラフィック装置の高度利用に焦点を当て、最近開発された CAD システムを題材に、対話性向上の試みを紹介する。

2. 論理装置 CAD におけるユーザ・インタフェースの課題^{4)~8)}

従来、論理装置 CAD において最も重視されていたのは、設計ミスの排除である。これは、LSI のように実装後の改修がきわめて困難なものに対する CAD だからである。そこで、ユーザ・インタフェースの向上に対しても、

- a) 設計データが簡単に誤りなく入力できること
- b) CAD ツールの操作が容易に学習できること
- c) 処理結果（たとえば検出した設計ミス）の表示

がわかりやすいこと

の3点に主眼が置かれていた。つまり、人手では困難な、設計エラーの検出などの仕事を計算機で処理させるための道具が CAD であり、効率的に誤りなく計算機処理できることが使いやすさとして求められていたのである。

一方、ユーザ・インタフェース向上の実際の手段は、CAD の目的とする機能により異なるものとなる。論理装置設計は図-1 のように階層分割されている⁹⁾。しかも、この階層の中でも設計手法の違いや細かい設計過程により、多種多様な機能をねらった CAD ツールが存在する。そして、個々のツールにおけるユーザ・インタフェースの実現では、機能に依存した工夫がはかられている。

その中で、最も重視されてきたのは対話性の向上である。たとえば、上記 a) の簡単で誤りの少ない入力方式として、グラフィック装置による対話型入力（データ入力時点での図形による入力の確認）がレイアウトや論理回路図の入力に多く用いられるようになった。

また、b) の CAD ツールの操作については、オンラインのヘルプ機能や、メニュー形式、アイコンなどによる対話型操作の導入により学習が容易になった。マルチウィンドウの採用による対話性の向上も重要な点である。複数の作業を端末の前に座ったまま同時に行

† A New Trend of User Interface for Digital System CAD by Akira SUGIMOTO(Central Research Lab., Mitsubishi Electric Corp.).

†† 三菱電機(株)中央研究所

* この点については、Frome による人間工学から見た批判がある⁹⁾。

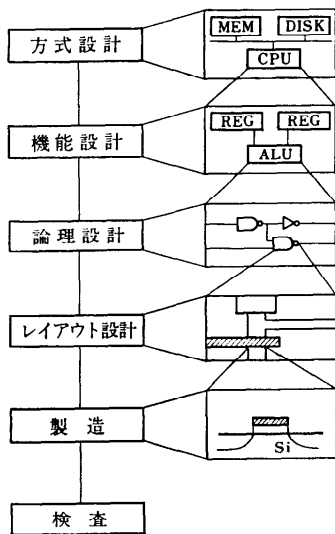


図-1 論理装置設計の階層

えるので、操作の効率が著しく向上するからである。

c) の処理結果の確認においても、問い合わせによる対話形式の出力が有効であった。これにより、従来のようにラインプリンタに打ち出した大量の出力結果を解釈する必要がなくなった。さらに、回路レベルや論理レベルのシミュレーション結果などは、波形出力をグラフィック装置にプロットすることによりわかりやすいものとなった。

一方、対話性の向上は、計算機環境の進歩によることも大きい。バッチ処理主体の処理方式から TSS によるオンライン化が始まり、やがて高機能のグラフィック装置が安価に利用できるようになった。さらに、EWS (Engineering Work Station)^{9), 10)} の登場やパーソナル・コンピュータの高機能化などにより、CAD システムを取りまく開発環境が急激に変化している。この環境の変化に合わせて、CAD システムの対話性が向上していったとも言える。

対話性向上の最初の試みは、従来のバッチ処理形式の CAD ツールに対話型インタフェースを付加していくことから始まった。次に、CAD ツール自体の対話型への改良（たとえばシミュレーション中に信号値を設定したり読んだりできるようにすること）が行われた。そして、ユーザ・インタフェースの重要性が認識され、またその実現手段が豊富になるとともに、最初から対話性の向上を目標とした CAD システムの開発が活発になってきている。すなわち、まず、対話性を重視したユーザ・インタフェースの仕様を設計し、そ

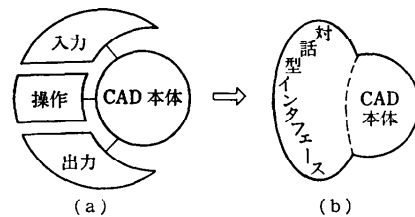


図-2 インタフェース設計と本体設計の融合化

れから、これを効率良く実現することを目指したシステム本体の開発が行われるようになった。

ここにおいて、図-2のような対話性向上の質的变化を見ることができる。すなわち、従来の論理装置CADでは、図-2(a)のようにCAD本体と、操作・データ入力・出力の別々のユーザ・インタフェースに分離され、それぞれについて対話性の向上が行われていた。ところが、ユーザ・インタフェースを中心としたCADシステムの開発では、図-2(b)のようにそれらを一体として対話性の向上が進められているのである。

これに応じて、ユーザ・インタフェースの課題も変化してきた。もちろん、上記 a), b), c) の課題が完全に解決されたわけではない。しかし、より積極的な課題も考慮されるようになった。すなわち、

d) 試設計や代案の評価が効率良く行えること

e) 人間により近いレベルで設計意図の入力ができること

f) 設計上の問題が総合的かつ直感的に把握できること

の3点である。

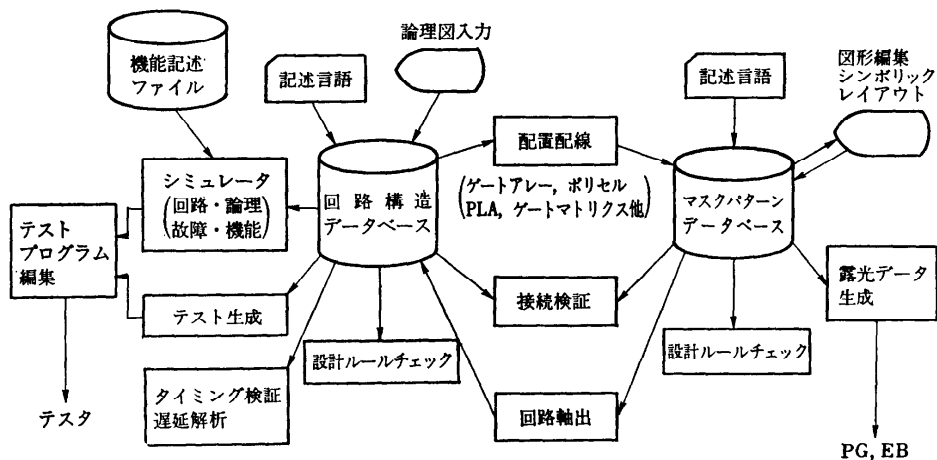
本来、CAD システムは設計者の創造的作業を効率良く支援し、より優れた設計を達成できるようにするのが目的である。従来の課題が人手では困難な仕事を計算機に処理させるための道具としての使いやすさを追求していたのに比べ、設計者の考える道具としての、いわば設計のしやすさが追求されるようになったのである。

以下では、このような対話性の向上の例として、論理装置 CAD ツールの統合化とグラフィック装置の高度利用について述べる。

3. 論理装置 CAD ツールの統合化

論理装置の設計がトータルな最適化を目指すものである以上、個々の機能を持つ CAD ツールを統合化することにより、全体として設計を支援することが重要である。

論理装置 CAD ツールの統合化において、ユーザ・

図-3 設計データベースを中心とした VLSI-CAD システムの例⁷⁾

インタフェースの観点からも重要な問題に、設計データの統合化がある。従来の CAD ツール統合化では、個別に開発された各種 CAD ツール間をデータ変換プログラムにより結合していた。しかし、この方法では基本的に同じデータが計算機内に散在し、データ管理が困難となる。そこで今日では、CAD ツール間の共通データベースを開発する方式が主流になっている。設計データの統合化については、総合的な解説（たとえば、参考文献 7), 11)) があるので、そちらを参考にされたい。

ここでは、最近の統合化における対話性向上の三つのアプローチを紹介する。まず、各ツールの独立性を比較的保った対話性向上手法として、共通ユーザ・インタフェース構築の例を述べる。次に、対話性を最優先にした、CAD ツールの対話型統合化の例を紹介する。最後に、ソフトウェア開発環境への CAD ツールの統合化の例を説明する。

3.1 共通ユーザ・インタフェースの構築

図-3 は、設計データベースを中心とした VLSI の CAD システムの例である⁷⁾。図に示すように、論理装置設計者は、多種多様な CAD ツールを使いこなす必要がある。その際、各ツールの使用方法に一貫性がないと、設計者がそれぞれの使用方法を学習する手間が増大する。

この問題に対処するには、図-4 のような共通ユーザ・インタフェースの構築が有効である。このような共通ユーザ・インタフェースを持った CAD システムの例として、Sherhart らによる EDeN¹²⁾ がある。EDeN では、一つのカラーグラフィック端末から階層

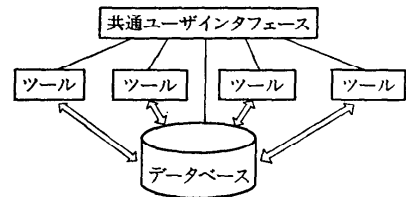


図-4 共通ユーザ・インタフェースの構築

的メニューにより多数の CAD ツールが（その端末の他の計算機で動くものも含めて）操作できる。各ツールに対応したメニューはできるだけ統一され、同じ機能のコマンドはメニューの同じ場所に同じ色で表示される。さらに、高水準のコマンドが用意され、ある機能を遂行するためのツール間の連結と起動を自動的に行う。また、同様な共通ユーザ・インタフェースを、ワークステーション上に実現した例として、Kao らによる ARIES¹³⁾ がある。

共通ユーザ・インタフェースの構築は、CAD ツール開発側にとっても各種インタフェース用ルーチンが共有できる利点がある。たとえば、上記システムの両方とも、各種シミュレーションの結果を波形表示するためのグラフィック・サブ・システムを持っている。また、最近のワークステーションでは、応用ソフトウェア開発者に対して対話型ユーザ・インタフェース構築用のユーティリティを提供している例が多い¹⁴⁾。この結果、各開発者間におけるユーザ・インタフェースの共通化が容易になりだしている。

ところで、共通ユーザ・インタフェースの構築にはフォーマルなアプローチもある。たとえば Janni¹⁵⁾ は、

各ツールに応じた階層的メニューの操作方法をオートマトン形式により記述する言語を開発し、共通ユーザ・インタフェース・プログラムがその記述を処理する方式を採用している。

3.2 CAD ツールの対話型統合化

共通ユーザ・インタフェースによる方法では、個々の CAD ツールが分離している。したがって、連続した作業を行うときも、ファイルやデータベースを介し、ツール間でデータを送信し合う必要がある。これは、連続した設計過程あるいは思考の流れを、個々のツールの起動・実行により分断することになる。この点で対話性を阻害する要因となる。

これに対して、ツール自体の対話型統合化を行った CAD システムの例を二つ紹介する。

3.2.1 Magic における例^{16), 17)}

Ousterhout らにより開発された Magic は、VLSI のカスタム・レイアウト・システムである。Magic の目標の一つは、実験的レイアウトを支援することである。レイアウト作成における試行錯誤は、初心者にとってチップの設計方法についてのより良い感覚を形成するのに有効である。また、設計者は、誤りの修正や、性能の向上のための他のレイアウトの評価・比較が容易にできる。

具体的には、Magic では次のことが目標にされている。

(1) 数日で大規模なレイアウトの初期配線ができかつ数時間で再配線が可能であること。Ousterhout らによる従来のシステム¹⁸⁾では、配線には数週間から数カ月間必要とした。

(2) 小さいバグの修正のターンアラウンドタイムが 15 分以内で済むこと。たとえば、レイアウトから抽出した回路の誤りがシミュレーション中に見つかったとする。誤りを修正した新しいレイアウトが幾何学的設計規則を満たしているかを確認し、回路情報を再抽出するのに少なくとも半日かかっていた。Magic では、これを 15 分以内で可能にすることを目標にしている。

(3) セルを再配置してレイアウトの違うトポロジを試すのに、数 10 分から数時間必要だったものを、30 秒から 1 分でできること。

このような目標は、従来のように a) レイアウト・エディタ、b) 設計規則検証プログラム、c) 自動配線プログラム、d) 回路抽出プログラムなどが別々に分散しては達成できない。そこで、Magic では、上記の機能を融合した高水準の対話型操作を設定し、それが

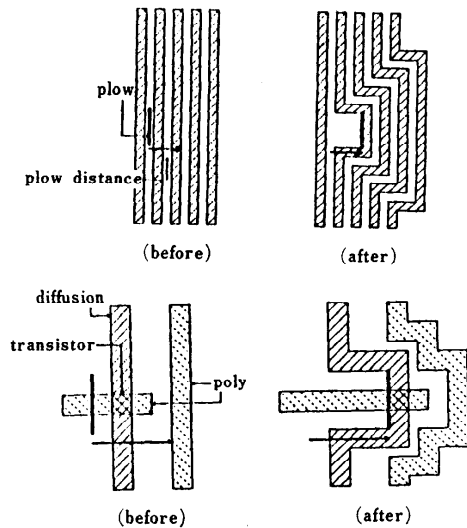


図-5 Magic の plowing の例¹⁷⁾

効率良く行えるようなアルゴリズムとデータ構造を開発している。すなわち、対話型グラフィック・レイアウト・エディタの中に、上記のすべての機能を統合化しているのである。

たとえば、Magic の対話型操作の一つに plowing がある。図-5 に plowing の例を示す¹⁷⁾。このような操作を行うには、設計規則やレイアウトの電気的接続、トランジスタに関する知識をシステムが持ち、しかもそれらの知識が効率良く使えるように統合化する必要がある。また、操作・レイアウト入力・操作結果が一体となっていることに注意されたい。

3.2.2 ICON における例¹⁹⁾

Hill により開発された ICON¹⁹⁾もレイアウト CAD システムの一つである。Magic がレイアウトレベルの設計に必要な機能を統合化した、いわば水平型統合化システムであるのに対し、ICON は論理レベルとレイアウトレベルの二つの設計レベルを統合化した、いわば垂直型統合化システムである。

ICON の統合化手法の特徴の一つは、論理回路図やレイアウトのシンボリック図、さらに実際のレイアウトパターンを、単一のグラフィック・エディタで混合して扱えることである。LSI 設計は図-1 のように階層的に分割されているとはいえ、論理設計を行う場合にもレイアウトを同時に考慮することが、LSI チップの面積を減らし、性能を向上させるために重要である。そこで、ICON では 3 種の図面データを同一の画面上で混在させて編集できるように、それらのデータ構造

を共通化した。

また、ICON ではシミュレータも統合化されている。そして、PLA や ROM などのようにゲート論理図を経由せず、レイアウト自動生成ツールにより直接作られたレイアウトパターンと、ゲート論理図を混在して編集した結果を、直接シミュレーションできるようにしている。また、後述するビジュアル・シミュレーションの機能も持つ。

3.3 ソフトウェア開発環境への統合化

共通ユーザ・インタフェース構築と対話型統合化のアプローチを比較すると、一般的にシステムの変更や機能の追加に対しては前者の方が柔軟性がある。しかし、対話型操作の実行速度の面では、後者の方が有利である。したがって、設計過程の全体を支援する CAD システムの開発には、一つの方法として設計作業の流れに応じた両方のアプローチの使い分けがある。

もう一つの方法は、LISP や PROLOG などのような対話性の良いプログラミング言語を中心とするソフトウェア開発環境¹⁰⁾の利用である。この例として、Cherry らによる NS²⁰⁾を紹介する。NS は LISP マシンのソフトウェア開発環境の中に、設計支援ツールを統合化したシステムである。このシステムでは、CAD 開発者も設計者も単一の言語（後述するオブジェクト指向言語の一つである Flavors²¹⁾）を使用する。そして、広いアドレス空間に、ツール群が LISP 関数の形で同時に存在し、即座に種々のレベルで組み合わせることができる。この方式の利点は、プログラムのデバッグに優れたソフトウェア開発環境が使えることに加えて、a) ツール群がデータを共有することによるツール間通信の効率の良さと、b) 機能拡張における柔軟性が良いことである。しかしながら、設計者が使いこなすには、LISP とシステム構成に対する深い知識が必要であろう。

また、同じ LISP 言語に基づいたオブジェクト指向言語 LOOPS²²⁾を使用した統合化環境として、Brown らの Palladio²³⁾がある。Palladio は、方式設計からレイアウト設計までの設計の全過程を、エキスパートシステムとして支援することを目標としている。

4. グラフィック装置の高度利用

論理装置 CAD システムの対話性の向上には、開発環境の進展によるところが大きい。特にグラフィック装置の高機能・低価格化は、大きな要因の一つであ

る。たとえば、アイコンやマルチ・ウィンドウなどによる操作性の向上は、高解像度のビット・マップ・ディスプレイの普及により始めて可能となった。また、3章で紹介した CAD システムにおいても、グラフィック装置が中心的役割を果たしていた。

さらに、論理図やレイアウト以外でも、ユーザ・インタフェースの向上のため、設計の記述に、テキストよりも図形を用いることが従来から盛んに行われてきた。たとえば、ハードウェアの機能レベルの記述に、フローチャートやオートマトン、ペトリネットなどの図的表現が採用されることは多い。これは、システムの機能把握の容易性、可読性を重視したものである。高機能グラフィック装置の普及により、今後、設計仕様の図的記述の役割がますます重要なものとなる。

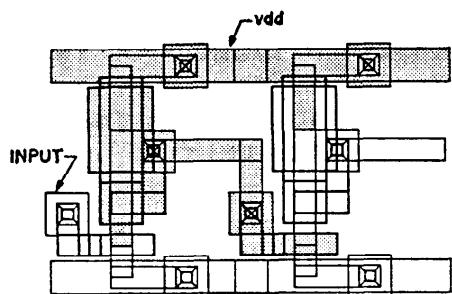
高機能グラフィック装置出現のインパクトはそれだけにとどまらない。ソフトウェア開発分野でもグラフィック装置の新しい利用方法について活発な研究が進められている²⁴⁾。論理装置 CAD の分野でも、操作・データ入力・出力が一体となった視覚イメージによる対話の試みが活発になってきた。以下では、その例を紹介する。

4.1 ビジュアル・シミュレーション

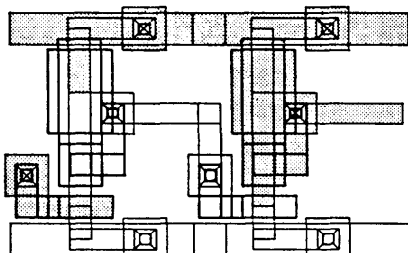
従来の論理装置 CAD のシミュレータの出力はラインプリンタへのリスト出力か、グラフィック画面上への波形出力が主であった。いずれにしても、ユーザの頭の中でシミュレーションの対象と出力形式の間の対応を取る必要があった。これに対して、ビジュアル・シミュレーションは、シミュレーションの対象自体をグラフィック画面に表示し、その対象の状態変化を画面上で動的に表現するアニメーションにより、シミュレーション過程を直感的にわかりやすく示す方式である。

たとえば、前述の ICON の開発者でもある Hill による Edisim²⁵⁾では、スイッチ・レベル・シミュレーションの過程を図-6 のようにレイアウト上の色の変化で示す。また、初期値設定も、画面上のレイアウトに対して直接行うようになっている。その後、ICON にもこの方式が受け継がれ、さらにゲート論理図を対象としたビジュアル・シミュレーションの機能も追加された。また、前述の Palladio においても同様な機能があることが報告されている²⁶⁾。

図-7 は、後述する VEGAMS²⁶⁾ によるスイッチ・レベルのシミュレーションの例である。MOS 回路トランジスタの双方向スイッチとしての役割が初心者に



入力=0 のときのレイアウトパターン



入力=1 のときのレイアウトパターン

図-6 Edisim のビジュアル・シミュレーション例²⁷⁾

もわかりやすいように、ワイヤの色だけではなく、トランジスタの形状も信号値により変化させている²⁷⁾。また、図-8 は VEGAMS によるマイクロ・プログラムのビジュアル・シミュレーションの例である²⁸⁾。この機能を中心に作成したマイクロプログラム開発支援システムは、実際の開発現場でも有効であった。

また、Molamed²⁹⁾らは、待ち行列システムの汎用ビジュアル・シミュレータを開発している。そして、それを複数の CPU を結ぶネットワークの評価などに用いている。

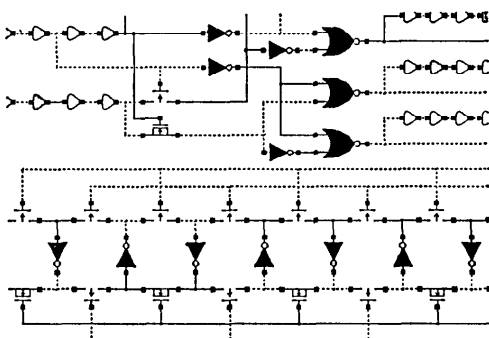


図-7 スイッチレベルのビジュアル・シミュレーション例

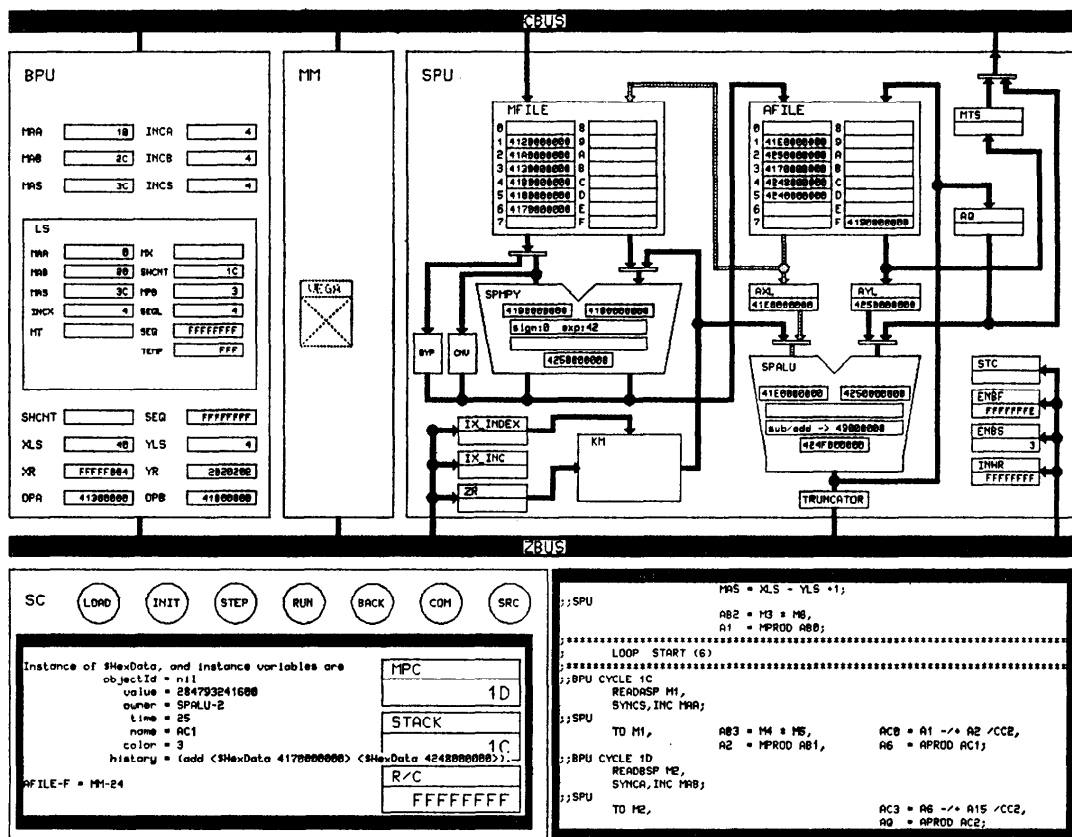


図-8 マイクロプログラムのビジュアル・シミュレーション例

4.2 視覚イメージによる対話とメタファの利用

ビジュアル・シミュレーションでは、シミュレーション結果の表示やユーザの操作インタフェースとして視覚イメージを使用している。しかし、ユーザがシステムを使用するときは、画面の背後にあるものを意識しているのが普通である。すなわち、シミュレーションの仕組みや実現方法に対してある程度の理解がないと使いこなすことができない。

もし、ユーザが画面上のイメージを実体として扱えるような、それ自体で閉じた一貫したモデル（概念モデル^{30)~32)}）を提供できれば、一層の対話性の向上が期待できる。このとき、ユーザがそのシステムを使いこなせるかどうかの鍵は、システム側の提供する概念モデルが明確でユーザの理解しやすいものであることである。

CAD システムの概念モデルを設計するとき、有効なテクニックとしてメタファの利用がある。Smith らの先駆的研究³³⁾では、オフィス情報システムの開発に、“Physical Office Metaphor”を用いている。すなわち、ユーザに親近感のある実際のオフィスにあるものを、画面上のアイコンを使用して電子的に提供することで、ユーザのシステムに対する理解を早めて

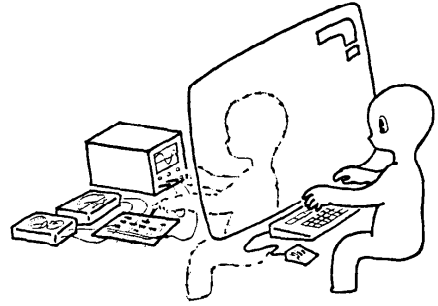


図-9 実験室のメタファ

いる。

そこで、論理装置 CAD 分野で視覚イメージによる概念モデルを設計する場合、ハードウェア技術者の実験室のメタファを使うことが考えられる。すなわち、図-9のように実際の実験室の中にあるトランジスタなどの回路部品やオシロスコープなどの道具を、グラフィック画面上のイメージの世界に移しかえることが有効であろう。

図-10 はこのような方針でハードウェア設計の実験環境の構築を目的として、筆者らが開発中の VEG-AMS²⁷⁾ の画面例である。ウィンドウ内には工具箱が

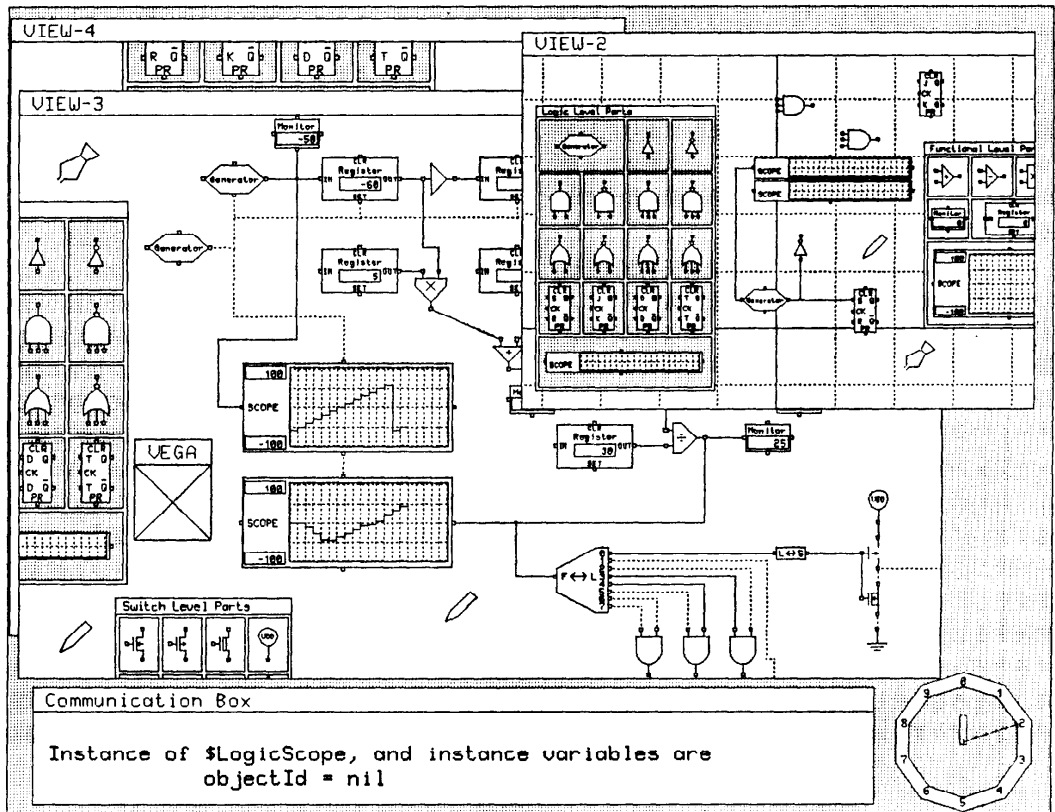


図-10 VEGAMS の画面例

置かれ、そこから回路部品やスコープをピンセットで取りだし、ワイヤで配線する。すると、回路部品や測定用道具がワイヤを通して情報の交換を行い動作する。もちろん、動作中に構造を変えると（すなわち、通電中の回路の接続をいじると）即座にそれに応じるようになっていく。

4.3 オブジェクト指向言語の採用

視覚的コミュニケーション技術は対話性向上の手段として有効である。しかし、このために CAD 開発の期間・コストがかかりすぎると、システムの硬直化を招き、LSI 技術の進展に追従することが困難となる。

この点で一つの解決策がオブジェクト指向言語の採用である。Smalltalk-80³⁴⁾を代表とするオブジェクト指向言語では、動作主体としての「もの (object)」が多数存在する。そして、それらの間のインタフェースはメッセージのみであり、メッセージを交換することで計算が進んでいく。すなわち、インタフェースという概念を言語の基本骨格に据えている³⁵⁾。たとえば、直接操作感をユーザに持たせる工夫として画面上のアイコンをマウスなどにより指示し、ボタンを押すという操作がある。このとき、同じボタンを押しても対象となるものに応じた動作をさせる必要があるが、このような機構がオブジェクト指向言語には始めから備わっている。

また、オブジェクト指向言語では、ある「もの」の機能や属性を継承する機構があり、これによりアプリケーションに応じたユーザ・インタフェースをより一般的なインタフェースのプログラムから容易に作成できるという利点も持つ^{20), 23)}。

そして、最も重要な利点は、画面上の表示が実質的に人間と計算機により共有されることである³⁶⁾。この点について筆者らの経験を紹介する。筆者らは以前、図-11(a)のような方式で VEGAMS と同様な機能を持つシステムを試作した³⁶⁾。ユーザに提供する視覚イメージによる概念モデルとは別に、回路構造やシミュレーション動作を実現するためのデータ構造とアルゴリズムが別に存在する。これを内部モデルと呼ぶ。概念モデルにはユーザがそれを実体として扱えるために一貫性が必要とされる。そこで、ユーザが回路の構造を変更したり信号を加えるたびに内部モデルの状態を変更する必要がある。また、シミュレーション進行による内部モデルの変化をすぐ視覚イメージに反映させる必要もある。そして、このモデル間の一致のための手続きがプログラムのいたるところに散在し、プログ

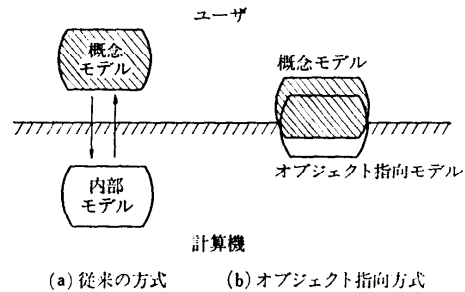


図-11 概念モデルの構成法

ラムの変更や機能の拡張が困難になった。

そこで、VEGAMS ではオブジェクト指向言語を採用したところ図-11(b)のように概念モデルと内部モデルをほとんど一致させることができ、プログラムのモジュール性が高まった。ただし、ユーザから見ると、B の概念モデルの方が親近感がより少なく、計算機の方に少し近づく。しかし、オブジェクト指向言語のメッセージにもとづく計算モデルは単純かつ強力である。そこで、ユーザは一度モデルを理解すると、使う上で自由度が高く柔軟性の高い操作が可能になる。

5. おわりに

本稿では、論理装置 CAD の分野で最近活発な、対話性の重視によるユーザ・インタフェース向上の試みを概観した。そして、CAD ツールの統合化とグラフィック装置の高度利用に焦点を当て、最近開発された CAD システムを題材に対話性向上の試みの具体例を紹介した。

設計のしやすい CAD システムの実現には、ユーザ・インタフェースの向上だけでは対処することができない。人間により近いレベルの高水準の対話を実現するには、CAD システムの高機能化が不可欠である。この点については、従来から設計自動化システムの開発において高水準の対話方式が研究されており^{6), 37)}、期待が大きい。また、AI 技術の応用も重要であろう³⁸⁾。たとえば、Kelly は CRITTER³⁹⁾という AI 技術を応用した、設計批評システムを開発している。このシステムは、デジタル回路の設計の、動作の正当性や速度、タイミングやデバイスのパラメータの変動に対する耐性を判定し、要約した結果を設計者に提示する。

このようなシステムに対しては対話手段の多様化も重要になろう。Samad⁴⁰⁾らは、将来の設計者の役割は、知的な設計環境と文字通り話しあって設計を進め

る知的な助言者になるだろうと考えている。そして、論理装置 CAD に適した自然言語インタフェースの研究を始めている。また、Yuschik⁴¹⁾ はグラフィックスと音声認識を統合した論理装置 CAD のユーザ・インタフェースについて実験を行っている。

また、最近、論理装置 CAD の分野でも人間工学からのアプローチが盛んである⁴²⁾。認知心理学³²⁾とあわせて、今後、この方面からのユーザ・インタフェースの向上も期待される。

なお、本稿で紹介した CAD システムの文献はシステム全体の機能や構成を説明したものが多く、かならずしも特にユーザ・インタフェースを中心として述べたものではないことをお断りする。

参 考 文 献

- 1) 村井, 南谷編: 大特集: 論理装置 CAD の最近の動向, 情報処理, Vol. 25, No. 10 (Oct. 1984).
- 2) Frome, F. J.: It Takes User Studies to Make CAD Systems "USER-FRIENDLY", Proc. IC-CD-84, pp. 4-6 (1984).
- 3) Sequin, C. H.: Managing VLSI Complexity: An Outlook, Proc. IEEE, Vol. 71, No. 1, pp. 149-166 (1983).
- 4) 渡辺 誠: LSI-CAD の現状と問題点, 情報処理, Vol. 23, No. 7, pp. 623-627 (July. 1982).
- 5) 元岡 達: 総論: 論理装置の設計支援システム, 情報処理, Vol. 25, No. 10, pp. 1026-1032 (Oct. 1984).
- 6) 永井, 牧野, 後藤: マンマシンインタフェース (III), 電子通信学会誌, Vol. 65, No. 7, pp. 754-759 (1982).
- 7) 蒲生, 佐藤, 岡崎: VLSI 設計におけるヒューマン・インタフェース, 第 1 回ヒューマン・インタフェースシンポジウム予稿集, pp. 53-56 (1985).
- 8) 山田他: パネル 論理設計自動化の現状と問題点, 情報処理, Vol. 26, No. 9, pp. 1043-1054 (Sep. 1985).
- 9) 志方, 山口: 設計用ワークステーション, 情報処理, Vol. 25, No. 10, pp. 1131-1136 (Oct. 1984).
- 10) 長谷川, 高木編: 特集: 高機能ワークステーション, 情報処理, Vol. 25, No. 2 (Feb. 1984).
- 11) 須藤, 唐津, 永谷: 設計データベース管理手法, 情報処理, Vol. 25, No. 10, pp. 1137-1143 (Oct. 1984).
- 12) Sherhart, K. et al.: The Engineering Design Environment, Proc. 21st DA Conf., pp. 466-472 (1984).
- 13) Kao, W. H. et al.: ARIES: A Workstation based, Schematic Driven System for Circuit Design, Proc. 21st DA Conf., pp. 301-307 (1984).
- 14) APOLLO: DOMAIN/DIALOG MANUAL, Apollo Computer Inc. (1985).
- 15) Janni, A. D.: Unified User Interface for a CAD System, Proc. 22nd DA Conf., pp. 9-15 (1985).
- 16) Ousterhout, J. K. et al.: Magic: A VLSI Layout System, Proc. 21st DA Conf., pp. 152-159 (1984).
- 17) Scott, W. S. and Ousterhout, J. K.: Plowing: Interactive Stretching and Compaction in Magic, Proc. 21st DA Conf., pp. 166-172 (1984).
- 18) Ousterhout, J. K.: The User Interface and Implementation of an IC Layout Editor, IEEE Trans. CAD Vol. CAD-3, No. 3, pp. 242-249 (1984).
- 19) Hill, D. D.: Icon: A Tool for Design at Schematic, Virtual Grid, and Layout Levels, IEEE DESIGN & TEST Vol. 1, No. 4, pp. 53-60 (1984).
- 20) Cherry, J. et al.: NS: An Integrated Symbolic Design System, Proc. VLSI-85, pp. 319-328 (1985).
- 21) Weinreb, D. and Moon, D.: Objects, Message Passing, and Flavors, in "Lisp Machine Manual" Fourth Edition (1981).
- 22) Bobrow, D. G. and Stefic, M.: The LOOPS Manual, Memo KB-VLSI-81-13, Xerox Palo Alto Research Center (1981).
- 23) Brown, H. et al.: Palladio: A Exploratory Environment for Circuit Design, Computer, Vol. 12, No. 20, pp. 41-56 (1983).
- 24) Rader, G.: A Survey of Current Graphical Programming Technique, COMPUTER, Vol. 18, No. 8, pp. 11-16 (1985).
- 25) Hill, D. D.: Edisim: A Graphical Simulator Interface for LSI Design, IEEE Trans. CAD, Vol. CAD-2, No. 2, pp. 57-61 (1983).
- 26) Sugimoto, A. and Fukushima, M.: VEGA: A Visual Modelling Language for Digital Circuit Design, Proc. ICCD-84, pp. 807-812 (1984).
- 27) 杉本 明: デジタル・システム設計のための図的モデル作成言語, 信学会研究会報告, EC84-28 (1984).
- 28) 杉本, 阿部, 黒田, 加藤: オブジェクト指向言語によるマイクロプログラム開発支援システム, 信学会計算機研究会報告, EC85-29 (1985).
- 29) Molamed, B. and Morris, R. J. T.: Visual Simulation: The Performance Analysis Workstation COMPUTER, Vol. 18, No. 8, pp. 87-94 (1985).
- 30) 溝口文雄: 応用認知: インタフェースの設計, 数理科学, No. 266, pp. 5-12 (Aug. 1985).
- 31) 田村浩一郎: マンマシンインタフェースの課題, 数理科学, No. 266, pp. 13-17 (Aug. 1985).
- 32) 三宅芳雄: 認知心理学からみたヒューマン・インタフェース, 昭和 60 年電気・情報関連学会連

- 合大会予稿集, Vol. 5, pp. 85-88 (1985).
- 33) Smith, D. C. et al. : Designing The Star User Interface, Byte Magazine, Vol. 7, No. 4 (1982).
- 34) Goldberg, A. and Robson, D. : Smalltalk-80 : The Language and Its Implementation, Addison-Wesley, Reading (1983).
- 35) 竹内郁雄 : オブジェクト指向言語とマン・マシン・インターフェース, 数理科学, No. 266, pp. 50-53 (Aug. 1985).
- 36) 杉本, 福島 : MOS 回路スイッチ・レベル・シミュレーションにおけるビジュアル・インタフェース, 第 28 回情報処理全国大会論文集, pp. 1459-1460 (1984).
- 37) 大附辰夫 : 配線処理手法, 情報処理, Vol. 25, No. 10, pp. 1090-1099 (Oct. 1984).
- 38) 川戸, 斎藤 : 論理装置 CAD における知識工学の利用, 情報処理, Vol. 25, No. 10, pp. 1161-1168 (Oct. 1984).
- 39) Kerry, V. E. : - The CRITTER System-Automated Critiquing of Digital Circuit Designs, Proc. 21st DA Conf. pp. 419-425 (1984).
- 40) Samad, T. and Director, S. W. : Towards a Natural Language Interface for CAD, Proc. 22nd DA Conf., pp. 2-8 (1985).
- 41) Yoschik, M. : Speech for Control of VLSI Graphics, IEEE ICCAD-83, pp. 16-18 (1983).
- 42) Frome, F. S. : Human Factors Update, IEEE DESIGN & TEST, Vol. 2, No. 1, pp. 8-10 (1985).

(昭和 60 年 11 月 7 日受付)