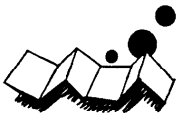
解　説



5. 最近の CAD システムの話題

5.5 論理装置の CAD における知識工学の応用†

川戸信明††　斎藤隆夫††

1. はじめに

半導体製造技術の進歩はとどまるところを知らず，年々その集積度を向上させている．この恩恵を享受すべく，論理装置は今後とも大規模・複雑化し，また多種多様な装置が設計されることになるであろう．したがって，論理装置の CAD に対する高性能化の要求が今後ますます厳しいものとなってくることは容易に予想される．

しかし，従来から研究されてきた CAD アルゴリズムの処理時間は装置規模の2乗，3乗に比例するものも多く，単なる改良の努力のみでは画期的な性能向上は期待できそうにもない．また支援レベルの高度化も困難である．そこで，新たな角度からさまざまな研究が開始されている．すなわち，従来の汎用マシン上の CAD の限界を越えるべく，並列処理による超高性能化をねらった専用処理装置やマンマシンインタフェースの高度化を目的とするワークステーションの研究開発が盛んに行われている．また，従来の CAD システムにおいては，各種ツールの扱う設計データの管理は必ずしも充分なものではなく，ユーザがその一貫性を保つ必要があった．これを解決すべく，これら設計データを統一的に管理し，矛盾のない設計を容易に行うことの可能なデータベース管理手法を追求する研究も始まっている．更に，設計対象システムのアーキテクチャをある程度固定することにより，動作記述から LSI のレイアウトを自動生成し，大幅に設計の生産性を向上させようとするシリコン・コンパイラの研究開発も活発化している．これらについては本特集号の他稿の解説を参照されたい．

本稿では，最近関心を集めている知識工学的アプローチを論理装置の CAD へ応用し，その高度化をはかろうとする研究に関して現状を紹介する．

知識工学へ大きな期待が寄せられる理由は，設計手法や設計の自由度を大幅に変更・制限することなく，従来熟練した技術者が行ってきた高度な設計活動を自動化できる可能性が高いからであると考えられる．

なお，知識工学自体今後大いに発展が期待される技術であり，またその設計問題への応用についても最近始まったばかりである．しかし，以下に示すように，すでに論理装置の設計に含まれるさまざまな問題について，熟練技術者の行う設計手法を計算機に行わせることに成功しつつあり，今後の発展が大いに期待される．

2. 設計の支援・自動化への応用

2.1 設計問題と知識工学[1)]

人間の知的な活動を解明し，その活動を人工的に実現することを目的とした人工知能に関する研究はすでに1950年代から開始されている．当初は，汎用問題解決システムのように高度な知的活動を一般的に実現する機構の追求が行われたが，その実現は非常に困難であることが明らかとなった．1970年代に入ると，一般的問題解決ではなく，対象分野を絞りその分野の問題解決に有効な諸性質，法則等の知識を明らかにし，これを積極的に利用しようとする研究が知識工学と名付けられて推進された．その結果，いくつかの実用システムも出現し，今後の情報処理において中心的役割を果す技術として発展しつつある．

知識工学におけるシステムは，分野固有の知識を知識ベースとして持ち，これを活用する推論機構を備えることを特徴とする（図-1）．特に，知識ベースに格納する知識を特定の分野における専門家から得た経験則や事実とし，専門家と同等の能力を持たせようとするシステムをエキスパート・システムと呼ぶ．従来，医療診断や機械の故障診断など解析型の問題を対象として多くのシステムが開発されている．

一方，設計とは，要求仕様を満たすべく既知の基本

† Application of Knowledge Engineering Techniques to Digital System CAD by Nobuaki KAWATO and Takao SAITO (Software Laboratory, Fujitsu Laboratories Ltd.).

†† (株)富士通研究所ソフトウェア研究部

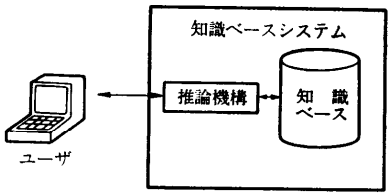


図-1 知識ベース・システム

要素からなる新しい対象を合成することが目標である．この際，一般には構成法に各種の制約があるのが普通であり，これらも同時に満足しなければならない．もちろん，この目標を達成したか否かを評価するために合成した対象を解析し，その機能や性能を評価することも必要である．評価結果が不満足ならば合成した設計対象の修正が行われることになる．このように，設計には解析問題が当然含まれるが設計が進むにつれて，設計対象には新たな情報が付加されたり，変更が加えられたりしてゆく．このように対象自体が操作され可変であることが設計問題の大きな特徴であると考えられる．

設計においても熟練した技術者は初心者より合成，解析の両面に関して優秀な能力を示すことは明らかである．経験から体得した知識の蓄積がこの差異を生ずると考えられる．したがって，このような経験的な設計知識も，従来より活発に研究されてきた解析型知識と同じく，これを獲得して計算機に格納し利用し得るものと見なしてよいであろう．

もちろん，現在の知識工学は演繹的推論技術が主であり，創造，発想，直観等を計算機に代行させることは困難であり，計算機が独創的で優秀な製品を設計するというようなことは当面望めない．

現在開始されている設計問題への知識工学の応用は，設計過程の中に効果的に演繹的推論技術を取り込み，できるだけ自動化をはかろうとしている段階と言ってよいであろう．このためには，設計過程を分析し，各過程においていかなる設計作業が行われているかを明確化する必要がある．

論理装置の設計の場合，この設計過程は図-2 のようになる[2]．半導体テクノロジ，実装技術等の選択を含めた基本仕様を決定した後，装置の基本的な機能ユニットと処理方式を定める方式設計が行われる．その後各機能ユニットの詳細な動作を定義する論理設計に進む．論理設計は機能設計と回路設計に分けられる．機能設計においては，演算回路やレジスタなどのブロックとこれらデータパス部をどのように制御するかが決定される．回路設計では，機能設計の結果に基づき，これを実現するため，AND/OR 等の基本回路の接続関係が定められる．更に，この論理設計結果から製造後の検査のためのテスト・プログラムやフィールドでの故障診断のための診断プログラムが作成される．論理設計が完了すると，プリント板への IC あるいは IC 内への基本回路の配置・配線を行う実装設計へと進む．実装設計終了後，製造，検査を経て，出荷されることになる．

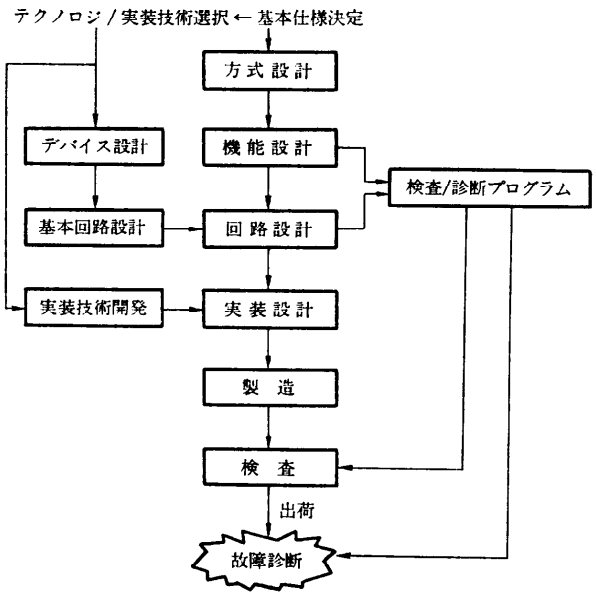


図-2 論理装置の設計過程

以下，まず本章の以下の節で論理装置の設計，特に合成の支援，自動化に関しての応用について述べ，3章では解析特に故障診断への応用を，4章では，その他周辺分野への応用について述べる．

2.2 方式設計

方式設計においては，装置の基本的な機能ユニットと処理方式を定める必要がある．この過程では，非常に高度な知識と熟練および創造性の発揮が要求され，通常最も経験豊富な高級技術者が方式設計を行う．このような方式設計を真の意味で自動化するためには，独創性等の高度な知的活動を計算機が代行せねばならず，すでに述べたように現在の知識工学技術では非常

に困難である．したがって，方式設計の自動化への応用例も現在のところないようである．

しかし，方式設計といえどもすべてについて新規設計を行うわけではなく，過去の経験等から得た設計例を豊富に持ち，これらをそのままあるいは適宜修正することによって設計を行う部分もかなりあると考えられる．したがって，このような設計例を知識ベースとして持ち，方式設計の支援を行うことは現在の技術でも充分可能と考えられる．今後この方向での研究が行われると予想される．

2.3 論理設計

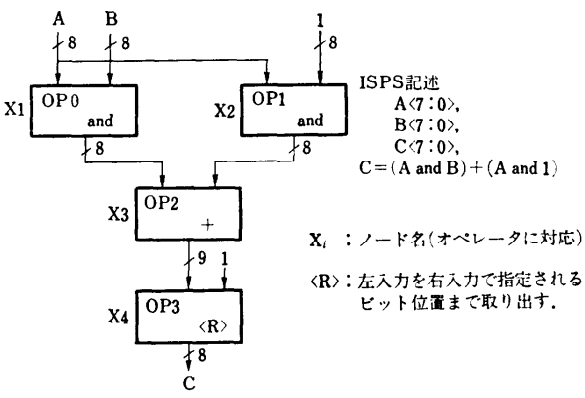
論理設計は機能設計と回路設計とに分けられる．機能設計においては，レジスタ，メモリ，演算部といった機能ブロックからなるデータパス部とこれらの間のデータ転送等を制御する部分についてその動作を具体的に設計する必要がある．回路設計においては，基本回路の接続により所望の動作を実現しなければならない．

機能設計から回路設計を得る段階で使用される設計知識は与えられた仕様を詳細化していく変換知識であるが，作業量も多くその結果誤りも犯しやすい．しかし，この段階での知識は，他の段階に比べて，比較的整理されており，また現在の知識表現に適していると考えられる．また，その自動化による生産性，信頼性の向上に対する効果も大きい．したがって，論理装置の設計に対する知識工学の応用はこの設計過程を対象とすることが現状では最も可能性が高いとも考えられる．

D. E. Thomas 等は，従来より CMU-DA と呼ぶ CAD システムの研究開発を進めている[3]．このシステムでは ISPS (Instruction Set Processor Specification) という方式設計記述言語によって動作仕様を与える．本システムは各種の支援ツールからなるが，知識ベースシステムのアプローチによって開発中のツールに DAA (Design Automation Assistant) がある．DAA の目的は ISPS 記述からレジスタ，マルチプレクサ，ALU 等の機能ブロックを自動生成することである．

ISPS 記述はまずデータフローを表現する有向非巡回グラフ (VT, Value Trace) に変換される (図-3)．DAA は汎用プロダクションシステム開発ツール OPS 5[4] に，以下のような合成ルール (OPS 5 のルール記述法に従って書かれたルールを英語に翻訳したものを示す) を組み込むことにより開発している (図-4)．

IF the most current active context is to create a link
　and the link should go from a source port to a destination port
　and the module of the source port is not a multiplexer or a bus
　and there is a link from another module to the same destination port
　and this other module is also not a multiplexer or a bus,
THEN create a multiplexer module
　and connect the multiplexer to the destination port
　and connect the source port and destination port link to the multiplexer
　and move the other link from the destination port to the multiplexer.



← 図-3 VT (Value Trace) の例

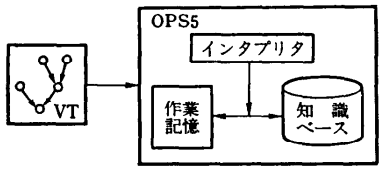


図-4 DAA システムの構成

このルールは，マルチプレクサやバス以外のモジュールの出力端子と他のモジュールの入力端子との接続法について述べている．すでに入力端子にマルチプレクサやバスではない別のモジュールの出力端子が接続されているならば，マルチプレクサを入力端子の手前に挿入し，各出力端子とマルチプレクサの入力端子とを接続すべきことを示している．

設計過程における対象のモデルは，作業記憶に次のような要素の集合として表現される．

literalize module
id : adder. o
type : operator
atype : two's complement
bit-left : 17
bit-right : ϕ
attribute : +

これは，18ビットの2の補数加算器モジュール adder. o を表わしており，この他に接続関係や VT の対応するノードを示す情報も持つ．インタプリタはこの作業記憶を調べ，IF 部が成立するルールを見つけ，その THEN 部を実行して作業記憶を書き換えるという操作を繰り返すことによって，設計を進めて行く．現在約300のルールからなり，8ビット・マイクロプロセッサの設計に適用した結果，熟練技術者がほぼ満足する機能ブロックとその間の接続方法が得られている．しかし，バス競合を減らし性能向上をはかるために，若干の接続方法の変更を行うべきであるとの指摘があり，現在これを実現するためのルールの改良をはかっている．

筆者らは，レジスタ転送レベルの機能設計記述言語 DDL (Digital system Design Language) をベースとした設計支援システムを開発してきた[5]．本システムはシミュレータ，ベリファイアにより検証した DDL 記述からレジスタの転送条件等の回路設計情報を出力するトランスレータを持つ．しかし，この情報から半導体テクノロジに応じた回路設計を行う部分は完全に人手によっており，この過程を支援するならば，生産性，信頼性を大きく向上させ得ることが判明した[6]．そこで，この回路設計を熟練した技術者が行うのと同様なやり方で自動化することを目的として，DDL/SX (Synthesis eXpert) を開発している[7]．**図-5** は，DDL/SX システムの構成を示しており，グラフィック・エディタを介して，機能ブロック（マクロと呼ぶ一例えば入力数に制限のない AND 等）の接続関係を表現した機能図が仕様として入力される．機能図から，実在する基本素子の接続関係を表現した通常の論理回路図への変換は，システムに組み込まれた回路合成知識が自動的に行う．現在，標準 TTL IC を使用する回路設計を対象としている．

DDL/SX では，設計過程における対象のモデルを表現するために知識表現手法の一つであるフレームを採用している．フレームは，各事物（あるいは概念）に関する知識やこれらの関係を構造化された情報として表現するのに適したデータ構造である．マクロや基本素子に関する知識および機能図や論理回路図におけるこれらの接続関係はフレームにより容易に表現できる．

グラフィック・エディタは入力された機能図をフレームへ変換する．以降このフレームが操作され，最終的な論理回路図へと変換されていく．この際用いられる回路設計知識には，①マクロの実在素子への展開，②素子数削減のための最適化，③IC 数削減のための最適化，④ファンアウト調整等の設計基準の充足等，各種のものがありこれらを適切な順序で適用していくことが必要である．そこで，これら各種の知識を独立した知識源としてモジュール化でき，各知識源の起動の制御を容易に行える利点を持つ黒板モデルに基づくプロダクション・システム開発ツール AGE/F を開発し，これを利用して開発を進めている．なお，黒板モデルにおいては，知識ベースは複数の知識源から構成され，これらが黒板と呼ばれる共通の作業記憶を介して情報のやりとりをし（他の知識源を起動する情報も含む一図-5 では矢印で起動関係を表示），協調的に問題解決を行う．

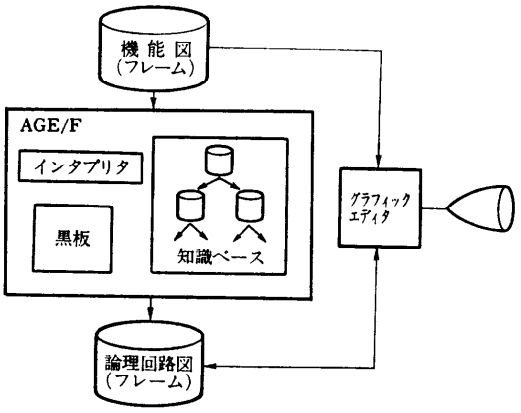


図-5 DDL/SX システムの構成

現在，15個の知識源と全体で約100個のルールからなるシステムをフィールド・テスト中であるが，熟練した回路設計者とほぼ同品質の回路を合成可能である．

なお，本システムでは，ユーザが合成過程に介入することを許しておりこれが容易に行えるよう，合成結果を描画して出力できるようにしている．このようなマンマシン・インタフェースに対する配慮も実用システムを考える際には，是非とも考慮しておかなければならない事項である．

Milton R. Grinberg も，カウンタ，シフタ，メモリ，レジスタ等の機能ブロックからなる回路の合成を行う SADD (Semi-Automatic Digital Designer) システムを開発している[8]．このシステムの特徴は，機能ブロックの仕様を自然言語(英語)で与える点にある．なお，論理装置全体の動作仕様等を自然言語により表現することは必ずしも容易ではないと考えられ，一般に自然言語入力が適切か否かは議論の分かれるところであろう．

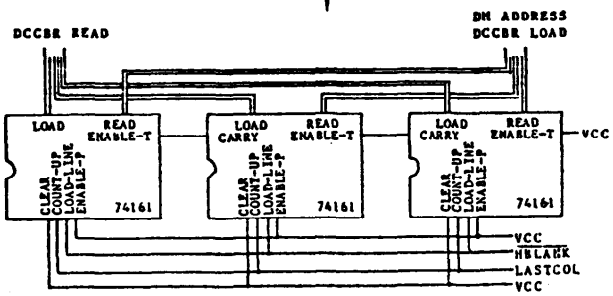
本システムにおける知識ベースは2種類ある．一つは，カウンタ等の機能の実現に必要な典型情報を格納したフレームである．入力文中で言及された機能に対応して，フレームのインスタンスが生成され，更に必要な情報が充足されて，対象のモデルが構築される．他の一つは，各機能の実現法であり，一般に同一機能を実現する複数の方法が用意される．各実現法は，その方法の適用可能条件を指定する部分と実際に機能を構成する方法を手続き的に表現したプログラムの名前とからなる．モデル中のインスタンスの情報と適用可能条件とのマッチングにより，最適な実現法が選択される．たとえば，図-6(a) のカウンタの仕様から，カウンタは2進出力で，リセット及びロードが可能な UP カウンタであるべきことがわかり (b) に示す74161カウンタを用いた実現法が選択され，プログラム

```
1. THE DISPLAY CHARACTER COUNTER (DCC) COUNTS FROM
   0 TO 3519.
2. WHEN THE COUNT OF THE SLC EQUALS 4, THE COUNT OF
   THE DCC CAPTURED-IN THE LOAD OF THE DCC-BASE
   REGISTER (DCCBR) WHEN THE HORIZONTAL BLANK
   (HBLANK) BEGINS.
3. THE DCC CAPTURED-FROM THE DCC-BASE REGISTER
   (DCCBR) WHEN HORIZONTAL BLANK (HBLANK) BEGINS.
4. EACH-TIME THE COUNT OF THE PIXEL COUNTER (PC)
   EQUALS 5 ENDS, THE DCC INCREMENTS.
5. THE COUNT OF THE DCC CAPTURED-IN THE ADDRESS OF
   THE DISPLAY MEMORY (DM).
```

(a) 英語による仕様の入力

```
(STRATEGY BIN74161 COUNTER
   (PREREQUISITES
      (COUNT-DIRECTION UP)
      (OUTPUT-CODING BIN))
   (ELIMINATORS )
   (TRAITS (RESETABLE YES)
           (LOADABLE YES)
           CARRY)
   (PROCEDURE $BIN74161)
   )
```

(b) 選択された実現法



(c) 合成されたカウンタ

図-6 SADD システムの動作例

$BIN 74161 により，必要な数だけ 74161 が接続されて（c）の回路が合成される．

2.4 実装設計

論理設計の結果得られる基本回路の接続関係に基づき，プリント板内あるいは LSI 内へこれら基本回路を配置配線（レイアウト）することが実装設計の目的である．最も古くから CAD が適用されている分野である．しかし，最も自動化の進んでいるマスタスライス方式 LSI においても完全な自動配線を行うことは困難であるのが現状である．

森等は，従来人間が行ってきた未結線配線の自動化を目的として配線設計のエキスパートシステムを開発中である[9]．会話型レイアウト CAD システムにおいて操作者が行う一連の手続きをルールとして，知識ベース化する手法を採用している．ただし，ルールの起動は操作者に委ねられており，今後この判断を自動化することが課題である．本システムは，FORTRAN との結合機能を有する PROLOG 型インタプリタを用いて実現しており，既存 CAD プログラムの有効利用と性能上の問題点の解決をはかっている（図-7）．

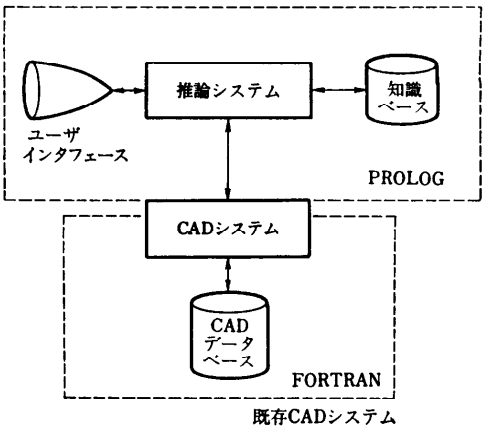


図-7　配線エキスパート・システムの構成

Harold Brown 等は方式設計から実装設計に到る全過程を統一的にサポートすることを目的として，Palladio システムを開発している[10]．本システムにおける支援ツール開発の基本方針は，半導体テクノロジの進歩に応じて支援ツール自体を容易に修正あるいは拡張可能なように構成することである．このため，オブジェクト指向，データ指向及びルール指向のプログラミングが可能な LOOPS[11] と論理型プログラミングが可能な MRS[12] を統合し，ツールごとに適切なパラダイムを選択できるようにして，システムの開発を進めている（図-8）．

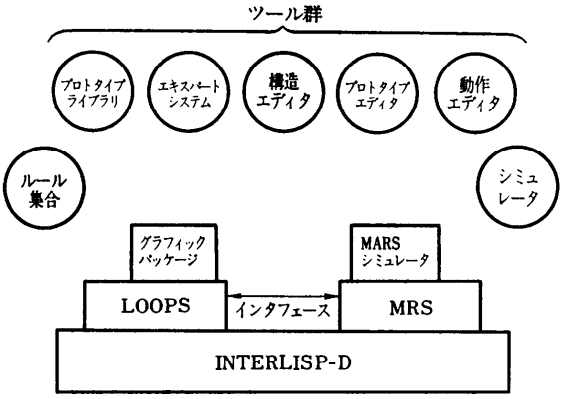


図-8　Palladio システムの構成

現在，知識ベースのアプローチを採用して開発しているツールに，コンポーネント間を接続する配線のマスク・レベルを決定するマスク・レベル・アサイナ（mask-level assigner）がある．本システムにおいても，対象のモデルはフレームにより表現される．このモデルを参照して，マスク・レベルの割当てを行う各種のルールが用意されている．図-9 にルールの例を示したが，これらのルールは交差する 2 つの配線 Wire 1, Wire 2 に対して適用される．最初のルールは Wire 1 のマスク・レベルがポリシリコン層で Wire 2 のマスク・レベルが拡散層ならば，プログラム Reassign Mask Levels を呼びマスク・レベルの再割当てをすべきことを示している．2 番目のルールは，Wire 1 の信号が電源で Wire 2 の信号が第 1 相クロックならば，Wire 1 のマスク・レベルを Wire 2 の前に割当てるべきであることを意味する．現在，人手設計とほぼ同等の割当てが可能であると報告されている．

```
Determine if mask levels need to be reassigned
    IF Wire 1: Mask Level=POLY
       AND
       Wire 2: Mask Level=DIFF
    THEN Reassign Mask Level
Order treatment of wires
    IF Wire 1: Signal=Power
       AND
       Wire 2: Signal=PHI 1
    THEN Assign Wire 1 before Wire 2
```

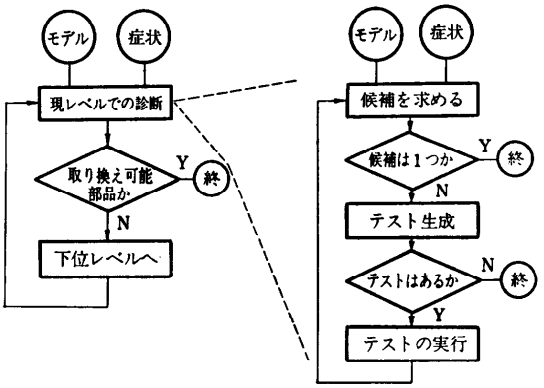
図-9　マスク・レベル・アサイナのルールの例

3. 故障診断への応用

論理装置の故障個所を診断するために通常採用されている方法は，D-アルゴリズム[13]等によりテスト・パターンを用意しておくものである．しかし，テスト・パターン生成の問題は NP-complete であり[14]，装置の大規模化に対処するのは困難になりつつある．そこで，ハードウェアの構造と各構成要素の正常動作を知識として持ち，症状からこの知識を利用した推論を行うことによって，故障個所を同定しようとする研究が始まっている．医療診断等のエキスパート・システムでは，症状と原因を結びつける経験則をルールとして持つのに対し，上記のアプローチはこのようなルールを持たず対象の厳密なモデルを持つ点が異なる．

Michael R. Genesereth の DART (Diagnostic Assistance Reference Tool) システムでは，SUBTLE という論理型言語により，階層的に設計されたハードウェアの構造と動作を論理式として記述し，モデル化する (**図-10**)[15]．診断は階層の上位レベルから下位レベルへと進められ，取り換え可能な部品が同定できた時点で終了する (**図-11**(a))．各レベルでは故障の可能性がある部品の決定，テスト・データの生成，テストの実行というループが繰り返されて故障部品が一つにしぼられる (**図-11**(b))．

故障部品の決定は，症状（期待値の否定）から出発し，モデルの論理式を用いた推論により ¬OK(c) でないコンポーネント c を導くことにより行われる．このような c が一つに決定しなければ，複数のコンポーネントのいずれが故障しているかを決定するためのテスト・データを同様に定理証明の方法により生成する．なお，本システムは前出の MRS を用いて開発されている．



(a) 全体の手順　　(b) 各レベルでのアルゴリズム

図-11 DART の診断方法

Randall Davis 等も同様のアプローチで，故障診断システムの研究を行っている[16]．

4. その他の応用

前章までは，論理装置の設計に直接的に関連した応用について述べた．ここでは，周辺分野における応用に触れる．

西田等は，質問応答やシミュレータに利用することを目的に，ハードウェア・マニュアルから情報を自動抽出する研究を行っている[17]．内部に持つハードウェア・モデルに基づき，マニュアルを理解し具体的なモデルを構成しようとするものである．

T. M. Mitchell 等はハードウェアの再設計問題(既設計の ASCII 端末用の設計データを利用して EBCDIC 用の端末を設計する等) の自動化を目的に，これに必要な推論方法，設計プランや回路動作の適切な表現法について研究している[18]．

```
MULT (M1) MULT (M2) MULT (M3)  } コンポーネントの
ADDER (A1) ADDER (A2)          } タイプ宣言
CONN (OUT (M1), IN1 (A1))      }
CONN (OUT (M2), IN2 (A1))      } コンポーネント間の
CONN (OUT (M2), IN1 (A2))      } 接続関係
CONN (OUT (M3), IN2 (A2))      }
        ⋮
ADDER(r)∧OK(r)                 }
        ∧IN1'(r)=x∧IN2'(r)=y→OUT1'(r)=x+y
ADDER(r)∧OK(r)                 } 動作記述
        ∧IN1'(r)=x∧OUT1'(r)=z→IN2'(r)=z−x   (OK(r) はコンポーネント r
ADDER(r)∧OK(r)                 }              が正常であることを示す)
        ∧OUT1'(r)=z∧IN2'(r)=y→IN1'(r)=z−y
```

図-10 SUBTLE の記述例

5. おわりに

論理装置の設計，診断等を支援する CAD システムへの知識工学の応用について概観した．すでに一部システムについては実際に適用が開始されているものの，全般に依然試作ないし研究段階にある．

今後，本格的な適用をはかるためには，設計ノウハウの護得，処理速度，取扱い規模など解決すべき課題は多い．また，現在の知識工学技術では扱えない一層高度な知的活動の解明には相当の長期間を要するであろう．

しかし，知識工学技術は，装置の大規模・複雑化及び設計需要の増大に対処し得る可能性の高い技術である．したがって，人間と機械の役割を充分に考慮しつつ，積極的にその成果を取り入れた CAD システムの研究開発を推進していくことが重要であると考える．

参 考 文 献

1) 大須賀節雄：CAD/CAM と人工知能，PIXEL, No. 12-16 (1984).
2) 樹下行三編：論理装置の CAD, 情報処理叢書 5, 情報処理学会 (1981).
3) Thomas, D. E., Hitchcock Ⅲ, C. Y., Kowalski, T. J., Rajan, J. V. and Walker, R. A.: Automatic Data Path Synthesis, IEEE Computer, Vol. 16, No. 12, pp. 59-70 (1983).
4) Forgy, C. L.: OPS 5 User's Manual, Dept. of Computer Science, Carnegie-Mellon University (1981).
5) 上原貴夫，川戸信明，斎藤隆夫，丸山文宏：大型コンピュータなどの論理設計に適用できる CAD システム，日経エレクトロニクス，No. 225, pp. 104-130 (1979).
6) 広瀬貞樹，川戸信明，丸山文宏，斎藤隆夫：DDL システムによるハード設計とその評価，57 年度電子通信学会全国大会，pp. 6, 83 (1982).
7) 川戸信明，斎藤隆夫，杉本裕之：機能図展開エキスパートシステムにおける知識とその獲得，情報処理学会第 27 回全国大会論文集，pp. 1189-1190 (1983).
8) Grinberg, M. R.: A Knowledge Based Design System for Digital Electronics, Proc. of the First Annual National Conf. on Artificial Intelligence, pp. 283-285 (1980).
9) 森 啓，光本圭子，藤田友之，後藤 敏：配線設計エキスパートシステム，情報処理学会第 28 回全国大会論文集，pp. 1073-1074 (1984).
10) Brown, H., Tong, C. and Foyster, G.: Palladio: An Exploratory Environment for Circuit Design, IEEE Computer, Vol. 16, No. 12, pp. 41-56 (1983).
11) Bobrow, D. G. and Stefik, M.: The LOOPS Manual, Memo KB-VLSI-81-13, Xerox Palo Alto Research Center (1981).
12) Genesereth, M.: MRS: A Metalevel Representation System, Working Paper HPP-83-28, Stanford University (1983).
13) Roth, J. P., Bouricius, W. G. and Schneider, P. R.: Programmed Algorithm to Compute Tests to Detect and Distinguish Faults in Logic Circuits, IEEE Trans. Comput. Vol. C-16, No. 5 (1967).
14) Ibarra, H. and Sahmi, S.: Polynomially Complete Fault Detection Problems, IEEE Trans. Comput, Vol. C-24, No. 3, pp. 242-250(1976).
15) Genesereth, M. R.: Diagnosis Using Hierarchical Design Models, Proc. of the Third Annual National Conf. on Artificial Intelligence, pp. 278-283 (1982).
16) Davis, R., Schrobe, H., Hamschen, W., Wieckert, K., Shirley, M. and Polit, S.: Diagnosis Based on Description of Structure and Function, Proc. of the Third Annual National Conf. on Artificial Intelligence, pp. 137-142 (1982).
17) 西田豊明，小坂 晃，堂下修司：ハードウェアマニュアルからの情報の自動抽出に関する考察，信学技報，Vol. 82, No. 204, PRL 82-56 (1982).
18) Mitchell, T. M., Steinberg, L., Smith, R. G., Schooley, P., Jacobs, H. and Kelly, V.: Representations for Reasoning about Digital Circuits, Proc. of the Seventh International Joint Conference on Artificial Intelligence, pp. 343-344 (1981).

(昭和 59 年 6 月 6 日受付)