

LSI/CAD用エキスパートシステム

Expert Systems for LSI/CAD

VLSI, ASICの時代を迎え、多種多様なユーザーニーズに応じて短期間で論理設計を行いたいとの要請が高まっている。本報告では論理設計を支援する知識処理方式の代表的な機能である設計、翻訳、検証を例に現在開発が進み、試用又は実用に供されているサブシステムを紹介し、この分野の可能性を明確にしている。

設計エキスパートでは、オブジェクト指向のフレームと論理形プログラムを用いて要求機能に適合して縮小可能な機能モジュールを用意することにより、人手並みの良質な論理回路を短期に生成できることを例示した。翻訳エキスパートではプロダクションルールと述語形ルールの組合せにより、熟練設計者以上の回路性能で翻訳の自動化ができ、実用されていることを示した。検証エキスパートでは証明の戦略を知識ベース化することにより、人手で変更を加えた部分回路などを自動検証できる可能性を示した。

浜田 亘曼* Nobuhiro Hamada
渡辺 俊典** Toshinori Watanabe
小林 康弘*** Yasuhiro Kobayashi

1 緒 言

VLSI, ASIC(Application Specific IC)の時代を迎え、多種多様なユーザーニーズに応じて短期間でLSIの論理設計をしたいとの要請が高まっている。しかし、論理設計分野での実用水準にある計算機支援技術は、論理シミュレータや回路図入力編集支援¹⁾などに限られており、大半を人間の設計作業に頼らざるを得ないのが実情である。電子回路のアーキテクチャからトランジスタ回路技術まで幅広い知識を持つ設計者が、絶対的に不足している。

一方、設計支援システムを構築する場合、FORTRANなどに代表される従来のプログラム手法では、論理回路の構造から動作、性能までを統一的に記述するのに無理があり、あえてプログラム化を行っても、プログラム構造が入り組んでいるため新しい要求の追加は非常に困難な作業となる。この問題は高速な演算能力と大容量記憶に支えられた知識処理技術の導入により解決する可能性がある。この技術は当初、医療やプラント²⁾などの診断問題に適用され実用化された。当然、診断と設計では基本的な差異があり、知識処理の設計への適用のためには設計問題中の適用領域の選択、知識処理プログラム技法の使い分けの検討、設計知識の表現形式の考案など新たな課題を解決しなければならない。

本報告では、論理設計を支援する知識処理方式の現状を概観し、LSI/CAD(Computer Aided Design)用エキスパートシステムとして代表的な機能である設計、翻訳、検証を例に各サブシステムを紹介し、この分野の可能性を明確にする。

2 特質と可能性

2.1 設計問題の分析

(1) 論理設計用エキスパートシステムの特徴

論理設計とは、ユーザーの意図する入出力データ間の関係を満たすような、電氣的制約を考慮した詳細な電子回路網を決定することである。通常、エキスパートシステムは必要な知識の広がり、その変化の速さを尺度として大別できる。論理設計の知識は非常に広範で、しかも時間とともに進歩し追加される性質がある。広範囲で多様な知識を扱うには、従来形の単純な数値計算や文字処理機能では不十分で、知識処理プログラムが好適である。一方、知識の膨大さや追加に柔軟に対応するためには、プログラムのモジュール性を格段に改善する必要がある。この意味でもオブジェクト指向のフレームや列挙形知識を扱えるプロダクションシステムが適している。更に、知識の変動に対応するには、長期的な知識獲得の支援が不可欠である。

(2) 論理設計用知識

LSIの論理回路の設計に必要な知識は次のように整理できる。

- (a) 設計支援の主要なねらいは、設計期間短縮か、品質重視(小形化や高速化)か。
- (b) 設計支援系の利用者(設計者)は、初心者かベテランか。
- (c) 設計対象の回路は、大規模か小規模か。
- (d) アナログ回路や非同期回路を含むか。
- (e) データパス部か制御部か。
- (f) 制御回路の構造はマイクロ制御方式かPLA方式かなど。
- (g) 回路実装テクノロジーはCMOS(Complementary Metal Oxide Semiconductor)かバイポーラか、など。

* 日立製作所日立研究所 工学博士 ** 日立製作所システム開発研究所 工学博士 *** 日立製作所エネルギー研究所 工学博士

(3) 設計方法

ALU(演算論理回路)や制御回路などを例として設計作業を分析すると、設計モデルは分割形、縮小形、整合形、変換形、繰返し形などに大別できる。

分割形では次数の高いブロックを適切なサブ ブロックに分割し、各部間の制約条件を考慮しながら各部の設計を逐次的に行う。縮小形でははん(汎)用的な回路構成に対し要求機能と構成要素の間の対応関係を利用して、要求機能で用いない構成要素を切り落として回路を構成する。整合形では非常に洗練された、はん用的で小さい回路に対し、要求仕様の入出力関係をうまく対応づけて、元の回路を利用する。これを別の視点で見ると、構造と入出力関係の制約を連立させて解を求めていることになる。

変換形では初期の入力情報から列举的な規則を局所的に適用し、徐々に新しい解を求める。変換用規則の前提部と結論部の関係で三つに分類できる。(1) 論理式→回路の結線情報、(2) 回路の結線情報→回路の結線情報、(3) 回路の結線情報→回路図。最後の繰返し形ではビット幅などのパラメータにより基本回路を繰返し配置する。

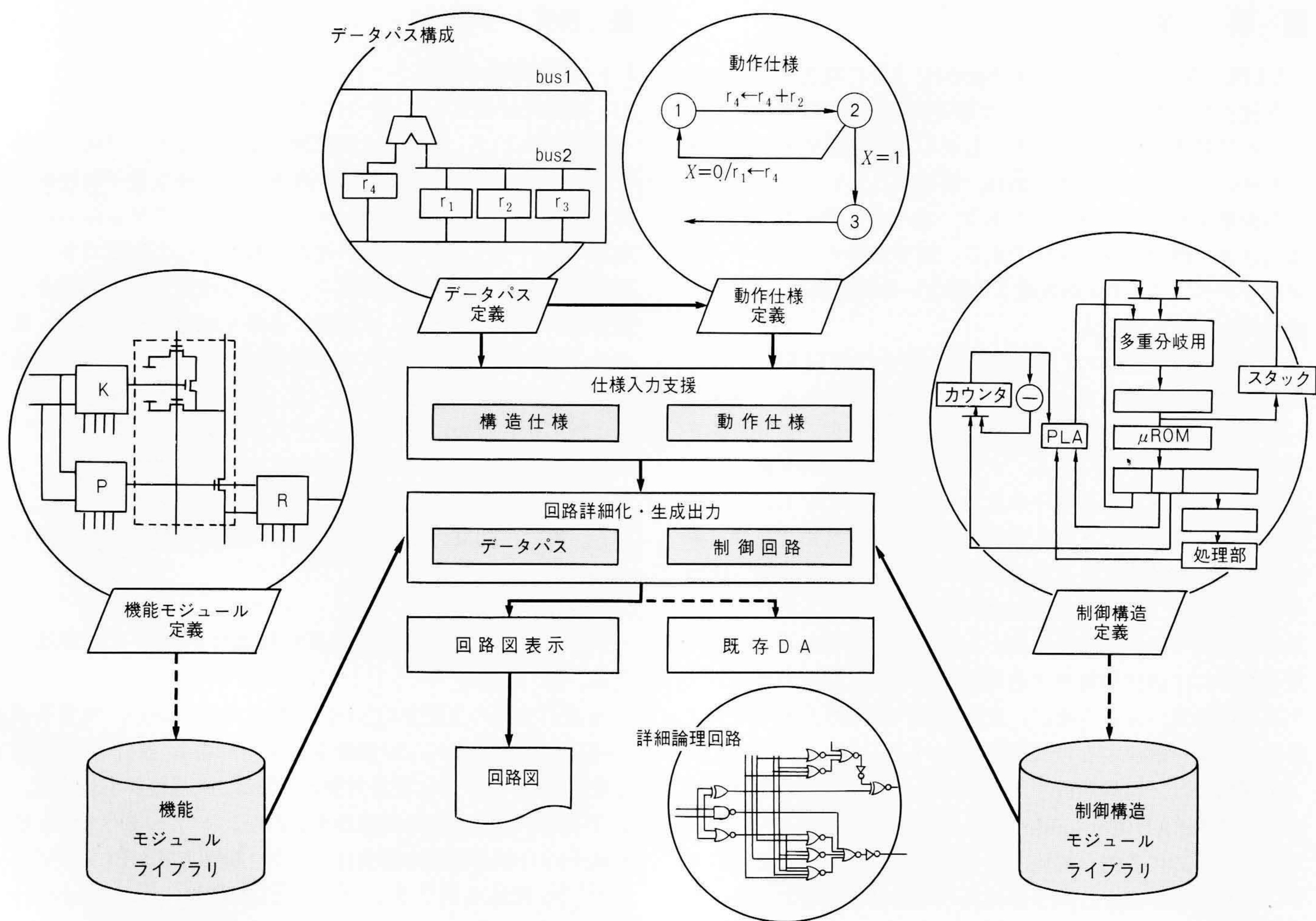
2.2 知識処理プログラム技法

論理設計エキスパートシステムを実現する知識処理プログラム技法を要約すると、次の三つになる。

プロダクションシステムは比較的等質な属性値の変換に対し、規則を列举的に用意し、その連立的ないしは三段論法的適用ができるのが特徴である。翻訳を含む上述の変換形設計法に適している。

一方、オブジェクト指向のフレームは、関連する各種の知識を構造化し、またデータと手続きを合体させることができ、従来の技法に比べて融通性の高い処置が可能となる。特にデータと手続きの合体により、既に値が決まっていればその値を、未定ならば必要な計算や入力要求により、値を確定させることができる。この方法により不要な再計算を省略し、支援システムの処理を高速化できる。

論理形プログラムは、論理的な関係をキーワードで列举し、そのキーワードの順序列として整合させながら条件を満足する解を探求する。これにより構造的制約と入出力の論理的关系を連立させて解を求めるなど、従来ではほとんど扱えなかった問題に対し、その条件を直接的かつ独立的に記述させ



注：略語説明など PLA(Programmable Logic Array), μ ROM(μ Read Only Memory), DA(Design Automation), $\longrightarrow$ (データの流れ), $\dashrightarrow$ (将来接続)

図1 論理設計支援エキスパートシステムの構成 左右のモジュールライブラリがシステムで用意されているとき、上部のデータバス構成と動作仕様をユーザーからの入力として、下部の詳細回路図を出力する。

ログラム化した上で解くことができる。この知識はユニフィケーションにより、入力と出力の関係が双方向に動作することも特徴である。

3 対話形論理設計支援エキスパートシステム

3.1 システムのねらい

本システムのねらいは次の2点である。

- (1) 知識処理と対話性を融合させ、人知の導入と介入により質の高い設計結果を早期に得る。
- (2) 知識獲得支援系を充実させ、応用範囲や生成回路の質をユーザー自体で拡張可能とする。

知識処理とは、人が判断したり探索したりする際に用いる条件を、多様な構造で記述し利用できることを応用し、人の判断や探索の一部を計算機に代行させるものである。一方、対話処理とは、仕様入力で不足な情報をユーザーに問い合わせたり、設計の中間結果をユーザーに示したりして、望ましい改善や明らかに避けるべき結果の回避を行うことである。また、知識獲得支援とは各種モジュールの追加などを容易とし、成長性のあるシステムとすることである。

3.2 システム構成

設計支援エキスパートは、図1の機能ブロックで構成し、各部を上述の知識処理プログラム技法の適宜組合せで実現する。

(1) 仕様記述入力部

データパスの概略構造、データパス上のデータ転送動作、及び仕様確定のためのパラメータが入力内容である。概略構造とは、ライブラリに登録された機能モジュールを構成要素として、接続関係を図形エディタなどで入力させるものである。データ転送動作とは、データパス上のデータの流を状態遷移の概念を考慮して記述したもので、通常は転送元レジスタ、転送先レジスタとその間の関係(代入、和などの演算内容)及びその転送を行う条件(状態と入力信号)を明記させる。パラメータ入力にはビット幅などがある。

(2) モジュールライブラリ

モジュールはデータパス用のALUなどの機能モジュールと制御信号を生成する制御構造モジュールの2種に大別できる。ユーザーは自分の希望する任意のデータパス構造を上記のデータパス概略構造入力系を用いて指定することができる。その構成要素となる機能モジュールが不足している場合、ユーザーはライブラリへ追加登録する必要がある。一方、制御構造モジュールは汎用性があるため、用意されたモジュールで機能的には十分であるが、より高速な制御を実現したいときや、よりコンパクトな構造で実現したいときには、専用の制御構造が必要となりユーザー登録が必要である。モジュール登録は知識獲得の一側面である。

(3) 回路詳細化・生成出力部

この部分も大きく二つの流れで構成される。一方はユーザーが直接関与するデータパス部、他方は背面でユーザーの要求する動作を達成させるための制御部である。いずれも、ユーザーが入力したデータパスの構造及び動作仕様情報からそこで必要なパラメータを抜き出し、所望の回路を生成する。

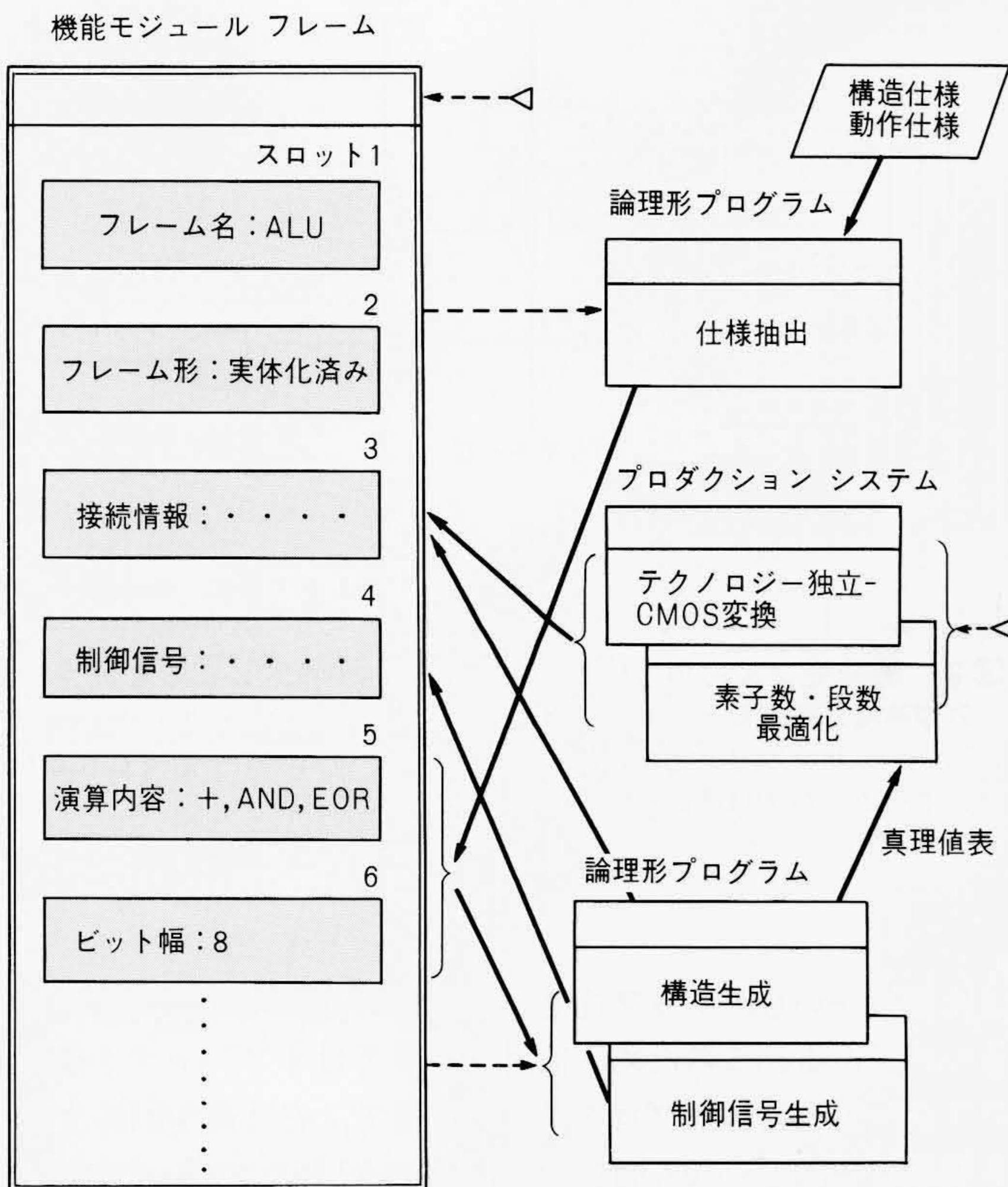
まず、概略データパスが入力された後、データパスが所望の転送を行うためにラッチ回路の挿入などの詳細化を行う。更に、各機能モジュールは仕様に合った部分を選択、整合した回路結線情報に分解する。次に、データパスの動作に必要な制御信号を発生する回路を、制御構造モジュールをベースに生成する。機能モジュールの生成と同様、その回路動作に不必要な部分は生成時除去されむだのない制御回路を生成する。最後にデータパスや制御回路をテクノロジー独立の回路から所望のテクノロジー(CMOS)にあった回路結線情報(ネットリスト)に変換する。

(4) 回路表示部

従来のDA(Design Automation)システム³⁾にデータを引き渡すには結線情報で十分であるが、人間がこの段階で設計結果の評価を行うには、回路図の形に図面化する必要がある。ここでは、ブロック図にある相対位置情報を利用し、人間が配置しそうなパターンを組み合わせで詳細回路の表示を行う。

3.3 具体例^{4),5)}

データパス上で中心的役割を果たし、速度上の律則部となるALUの生成を例にエキスパートシステムの内部構成と作用を説明する。設計中のALUの機能は、動作仕様中に現れる加算や論理的EOR(Exclusive OR)などの演算子に従って必要最小限に抑え回路生成したい。必要最少回路生成の効果は、特にLSIの場合肝要である。歩留まりのため価格がチップサイズ



注: 略語説明など ALU[演算論理回路(Arithmetic Logic Unit)]
CMOS(Complimentary Metal Oxide Semiconductor), EOR(Exclusive OR)
----- (起動制御), ▢--- (コマンド), ——— (データの流れ)

図2 機能モジュール生成用フレームの例 ALUなどの機能モジュールは、オブジェクト指向のフレームとして実現される。フレームの各スロットは論理形プログラムを必要に応じて起動する構成となっている。

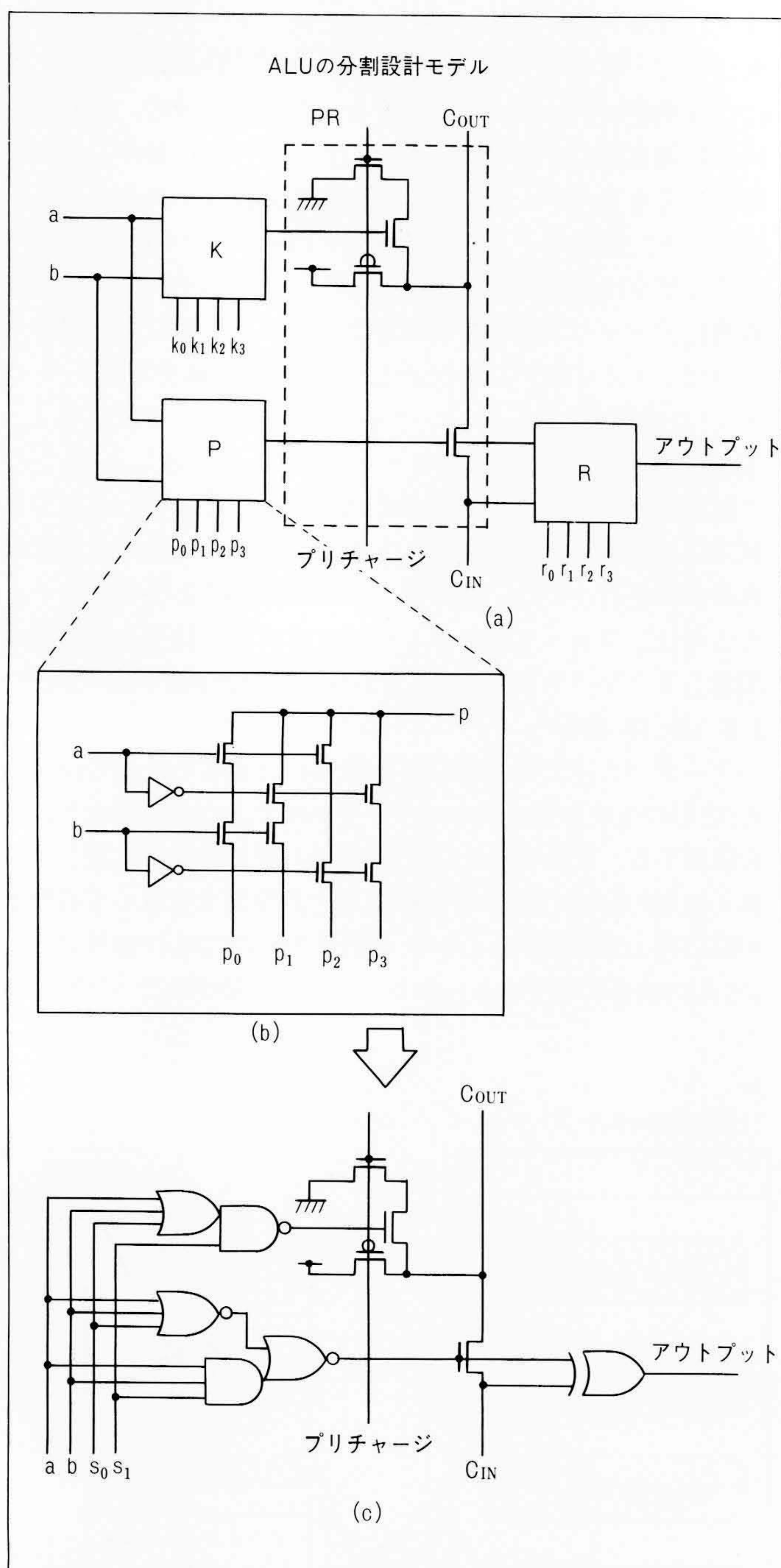


図3 機能モジュールの模範回路と演算内容指定による生成回路
処理速度の律則部となるALUは、分割形、整合形などを駆使して、模範回路(a)、(b)から縮小生成した回路(c)を得る。(b)は(a)の一部を拡大したものである。ALUは機能モジュールの代表例である。

に比例しないし、チップ性能が律則部の遅延時間で支配されるためである。このため、設計支援系ではオブジェクト指向のフレームと論理形プログラムの技術で、必要最少回路生成を実現している。模範回路とも言うべきALUのフレームには、図2のように仕様や構造や動作要求から成る一連のスロットと呼ばれる整理棚がある。各スロットは単純なデータ格納を行うばかりでなく、データが不足すればユーザーに直接問い合わせたり、関連フレームやスロットを参照したり、必要な値を計算したりするデータと手続きを合体させた構造となっ

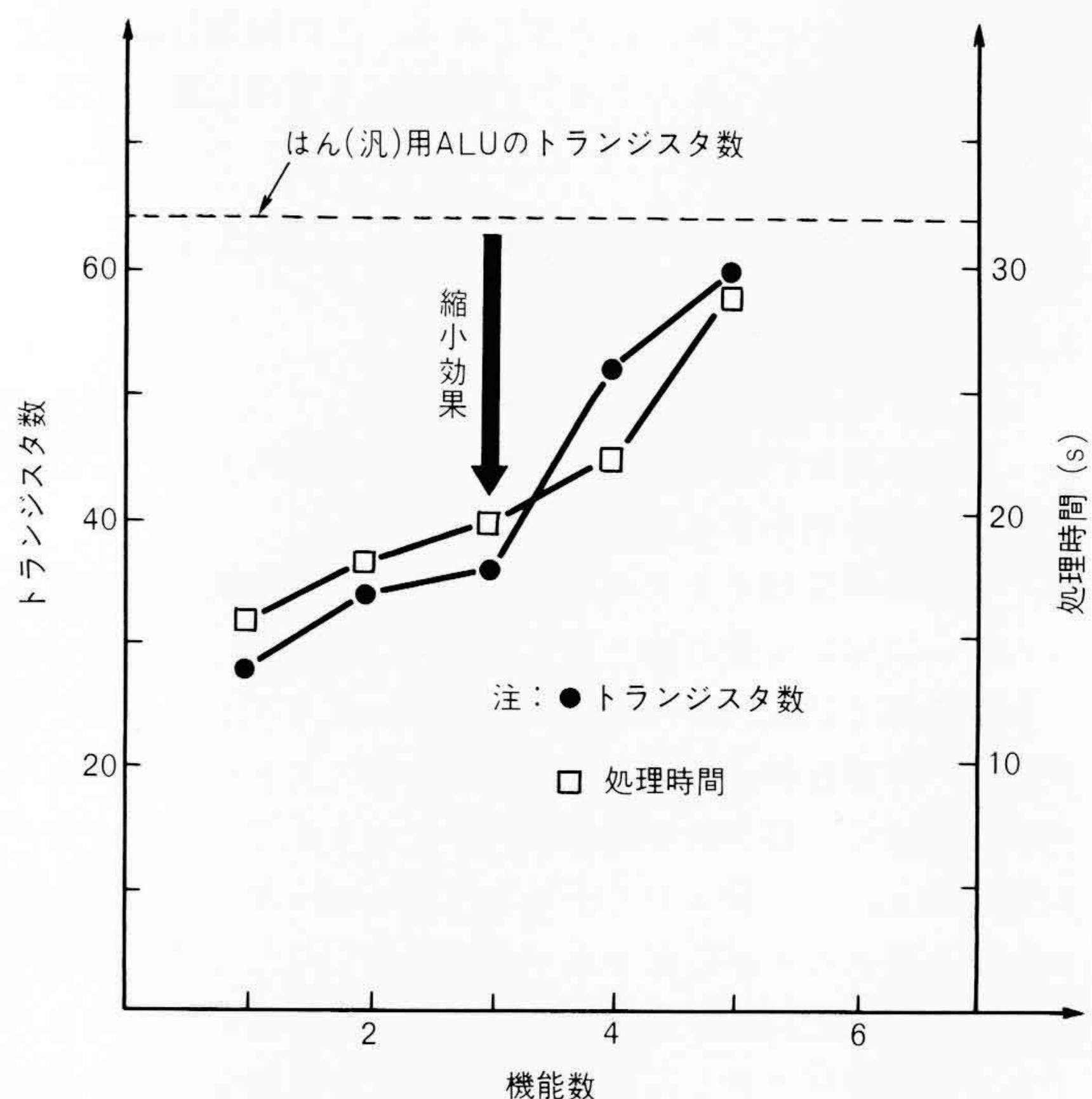


図4 はん用回路に対する縮小効果例(ALU) はん用ALUのトランジスタ数に比べて、必要な機能が少ないときは、数十パーセントの縮小効果が得られる。

ている。

手続きは論理形プログラムで実現しているので、結線情報などの個数が不定の系、再帰的な処理、構造と動作条件など異質な条件の連立下での解の探索が可能である。

さて、機能モジュールなどの設計対象回路がパラメトリックになる場合は、特に要求機能やビット幅などの値に従って、詳細構成や必要制御信号などを生成する手続きを用意する必要がある。これは当分は、システム開発側で提供することになる。

要求機能により必要最小限の回路を求めた効果例を図3、4に示す。本方式で縮小すると、ALUや制御構造の例で数十パーセント減が可能となり、人手に匹敵する品質の回路が得られた。

4 回路系自動翻訳エキスパートシステム^{6),7)}

顧客仕様に対し機能的に満足な回路でも、性能上などの理由で別のテクノロジーのLSI技術にマッピングする必要がある場合、現状では熟練者の多大な関与を要し、機械化が遅れており、設計者数がLSI開発可能量を制限してしまうという事態に直面している。本章では、最近顧客側で多用されているCMOS回路図をもとに、対ノイズ性、チップ面積、アナログデバイスとの相性の良さなどの点から、オーディオ機器用LSIなどに賞用されているバイポーラ系のIIL(Integrated Injection Logic)論理回路図を生成する回路系自動翻訳技術を紹介する。知識処理の枠組みのもとに設計者が作業時に使用する知識を直接計算機で利用することによりASIC設計上のあい路の一つを解消している。

4.1 知識処理による回路系自動翻訳の考え方

設計者の作業は、(1)原回路図面を、素子のファンアウト条

件などを考慮しながら目標回路図面に変換する。(2) 目標回路図面内の不要素子を、ファンアウト条件などを考慮しながら削除するというステップから成る。これらのステップでは様々な翻訳用知識が使用される。各知識は一般に対象回路内の素子の種別やその周辺素子との結線状況进行分析する部分と、分析結果に基づいて素子を変更する部分から成る。また、原回路図と素子位置類似の目標回路図を作成し、結果チェック時に両者の目視対比を容易化するという考慮も払われる。以上の設計作業を自動化するために以下の方針を採った。

(1) 翻訳ステップ

- S1. 原回路の回路構造認識：浮きピン除去、フリップフロップの定義変更などを行う。
- S2. 目標回路生成：原回路基本概念抽出、基本概念変換、目標回路生成の3ステップから成る。具体的には各々NOT及びNOR標準素子化、標準素子のIIL素子への変換、IIL素子のファンアウト整合処理を実施する。
- S3. 目標回路の最適化：IIL回路内の不要素子をファンアウト整合などの諸制約条件を守りながら削除する。

(2) 回路図の生成

原回路図に素子位置類似の目標回路図を自動生成するために、上記(1)の各ステップで回路結線情報(トポロジー)と回路図面情報とを同時並行に変換する。ステップS3の最後で目標回路図の見やすさ向上処理を行う。

4.2 システムの概要

(1) 機能構成

システム機能構成を図5に示す。知識ベースには前節(1)の処理を実施するためのルール群が蓄積されている。推論機構の一部であるワーキングメモリには、回路結線情報とともに

関連する図面情報が記憶される。ルール解釈実行機構は、結線情報と前提部が照合するルールを選出し、その結論部を実行する。結論部でワーキングメモリ内の結線情報と図面情報を同時変更させ、前節(2)の機能を実現している。ルールのうち述語論理形のものは、if-then形ルールのif部やthen部内で使用する基本述語の定義に当てている。

(2) ルールの具体例

回路系変換、回路最適化、述語形のルールの一例を図6、7に示す。前記(1)、(2)では結線情報に随伴した図形データ(長方形領域と配線)を結線情報変更と同時に更新する図面修正述語が含まれている。図6の例についてその意味を述べると、「各称NameのCMOSのnorゲートは、出力ノードXのファンアウト調整を行った後、親ブロックB₁をB₁₁、B₁₂、B₁₃の子ブロックに分割し、入力ノードの最初のものに対するIILゲートを生成し、更に残余入力ノードに対する変換を再帰的に実施せよ」である。当ルールによってCMOSのnorゲートが、その図面情報も含めてIILゲートに変換される。図7についても同様に解釈されるが詳細は略す。システムは約100個のif-then形ルールと約300個の述語形ルール、計約400個を保有している。

4.3 実施例

本システムの動作例を図8に示す。回路は模擬的なものを使用している。CMOS原回路からファンアウト整合を取ったIIL回路が得られ、最適化される。同図(b)に対し同図(c)は約30%の素子数となっており、チップ面積や消費電力の削減を可能とすることがうかがえる。また、原回路に素子位置類似のIIL回路が得られ、CMOS回路図を入力すれば、IIL回路図を自動生成するという所期の目標を達成していることが分かる。

本システムは、昭和61年3月段階からカメラ用ICの回路設

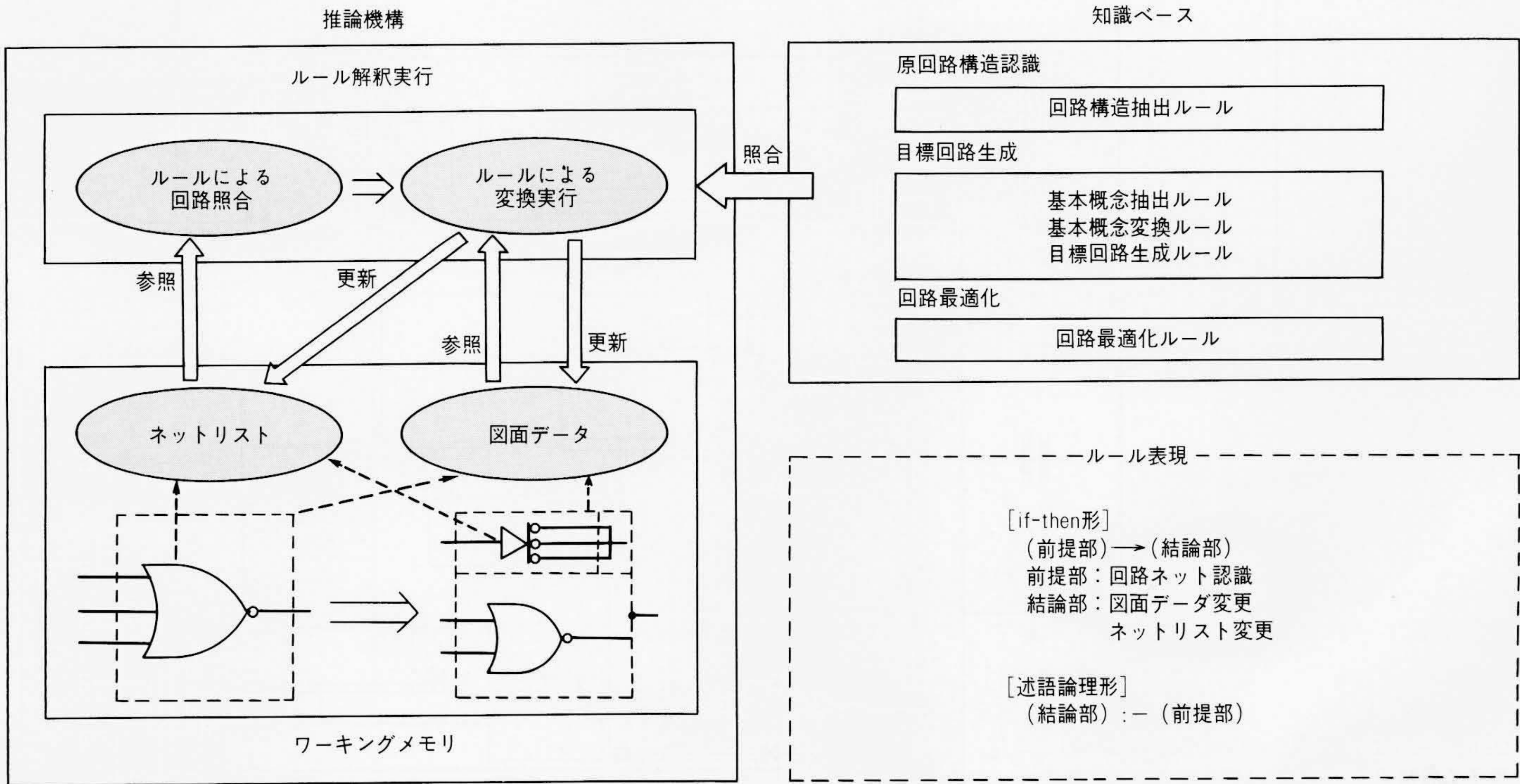
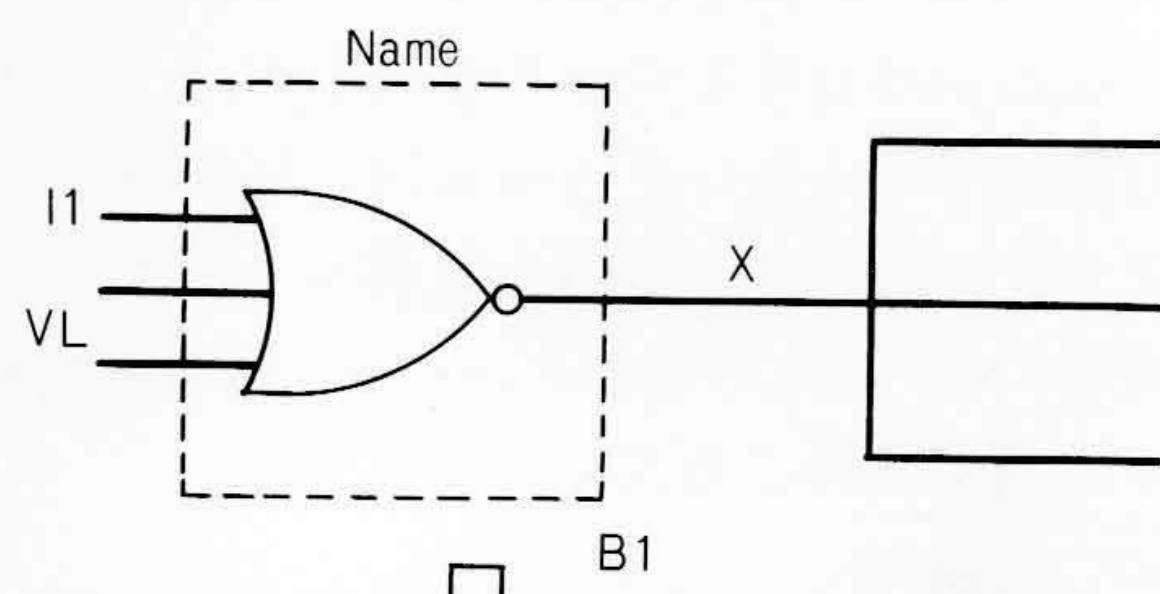


図5 回路翻訳システムの機能構成 if-then形ルール及び述語論理形ルール合計約400個の知識を保有し、ネットリストと図面データを同時に変換するという特徴を持つエキスパートシステムである。

ワーキングメモリ



目標回路生成ルール(例)

ゲート(nor, Name, cmos, 入力([I1 | VL]), 出力([X]))

→ ファンアウト調整(X, XL)

図面修正(親ブロック(B1), 分割ブロック(B11, B12, B13)),
ゲート(inv, Name, ii |, 入力([I1]), 出力([XL])),
ゲート(nor, Name, interm, 入力(VL), 出力([X])).

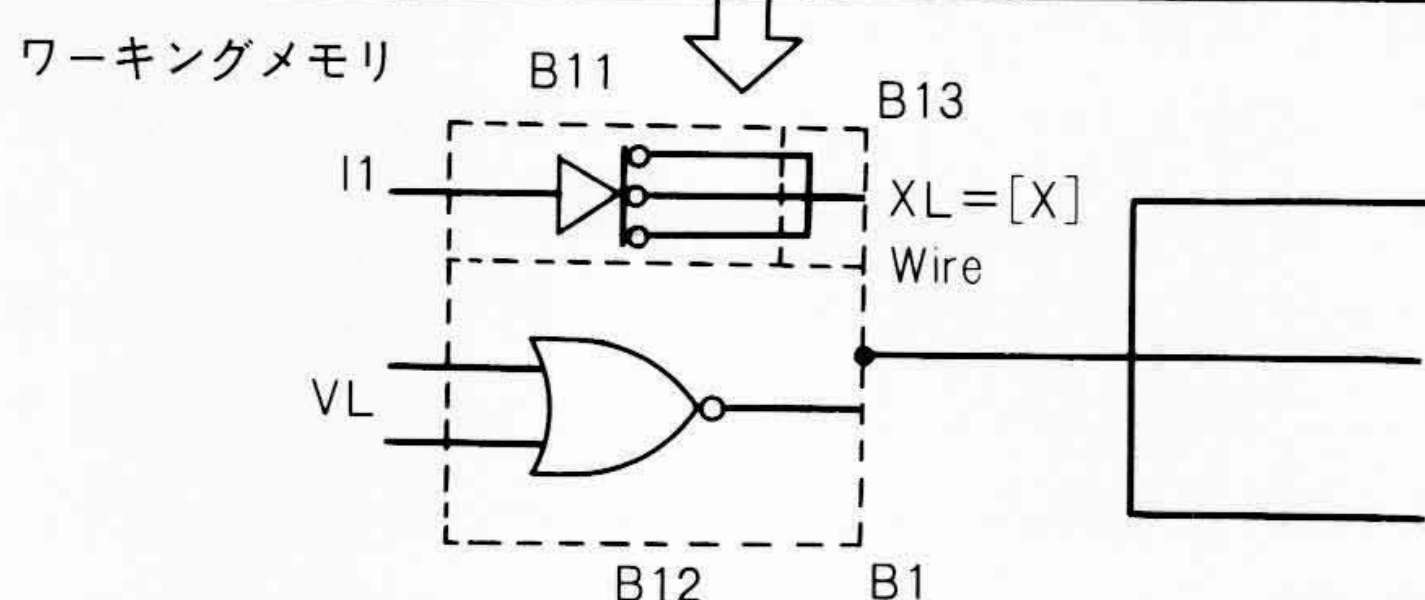
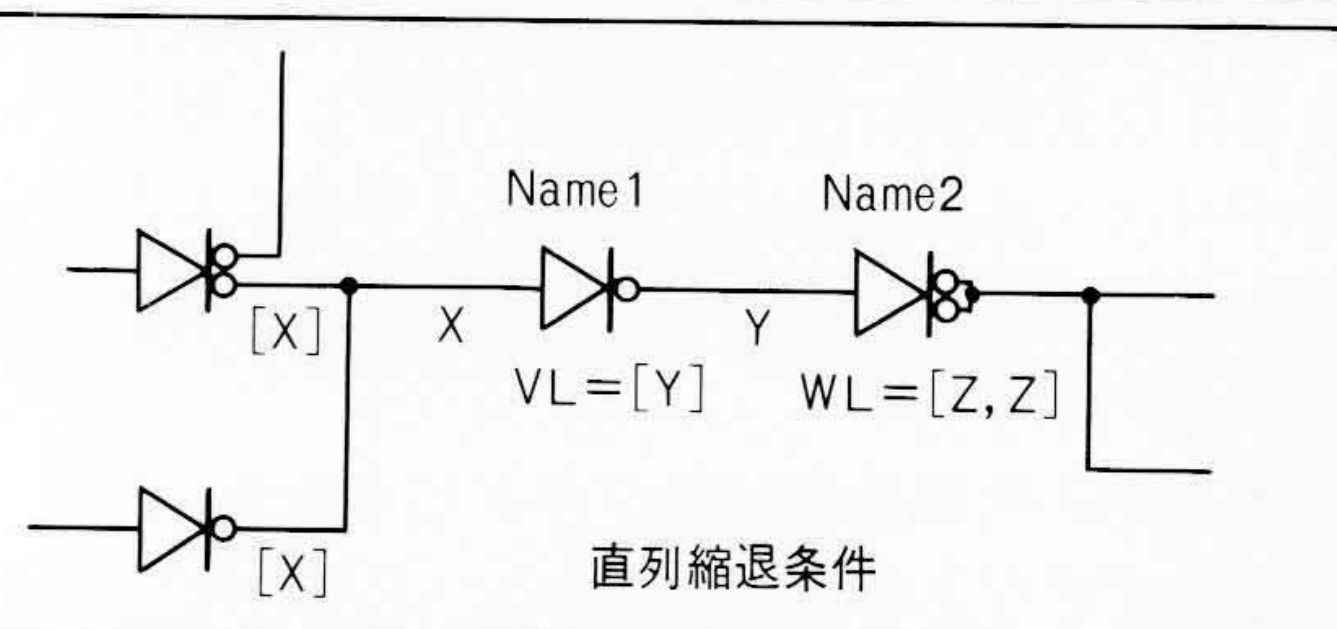


図6 回路系変換ルールとその作用例 CMOS論理図内のnorゲートをNANDゲートに変換するルールである。ネットリストと図面データの同時変換を行う。

述語論理形ルール

直列縮退可能(入力([X]), 出力([VL]), 入力([Y]), 出力([WL]) : -
 ファンアウト増加数(VL, WL, N), ①
 コレクタ追加可能(X, N), ②
 not(結線論理(Y)). ③



②コレクタ追加可能

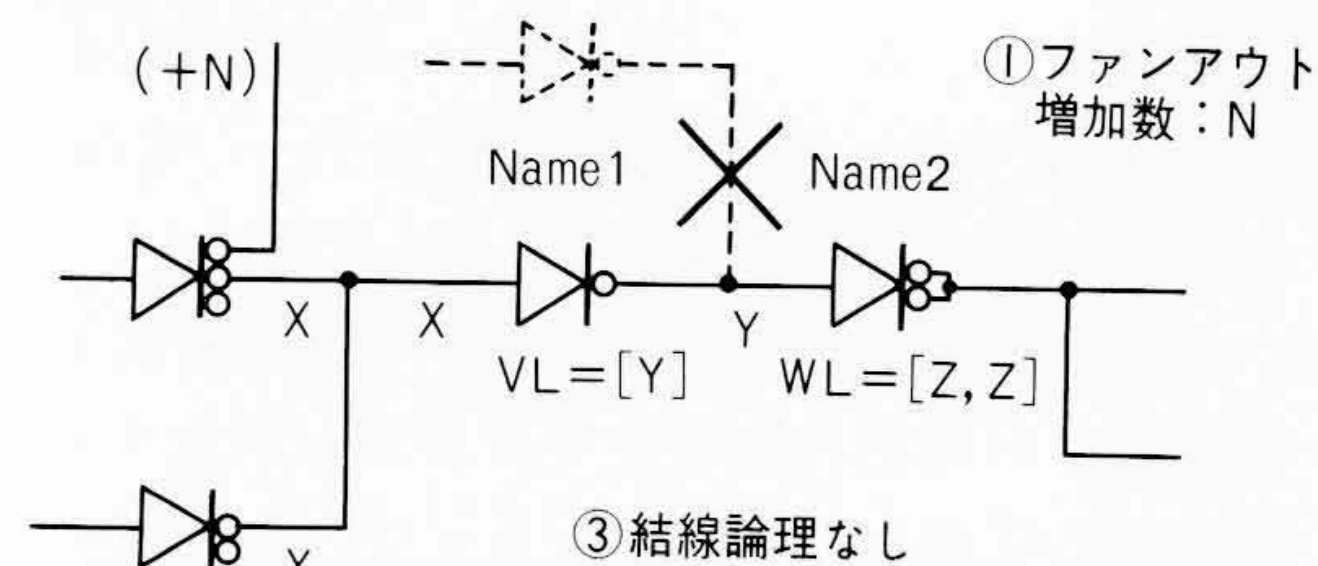


図 7 述語論理形ルールの例 直列 2 段の IIL(Integrated Injection Logic)ゲートにつき削除した場合の上流素子でのファンアウト増加可能性と 2 段のゲート間での結線論理の非存在性をチェックし、直列縮退可能性を判定するルールである。

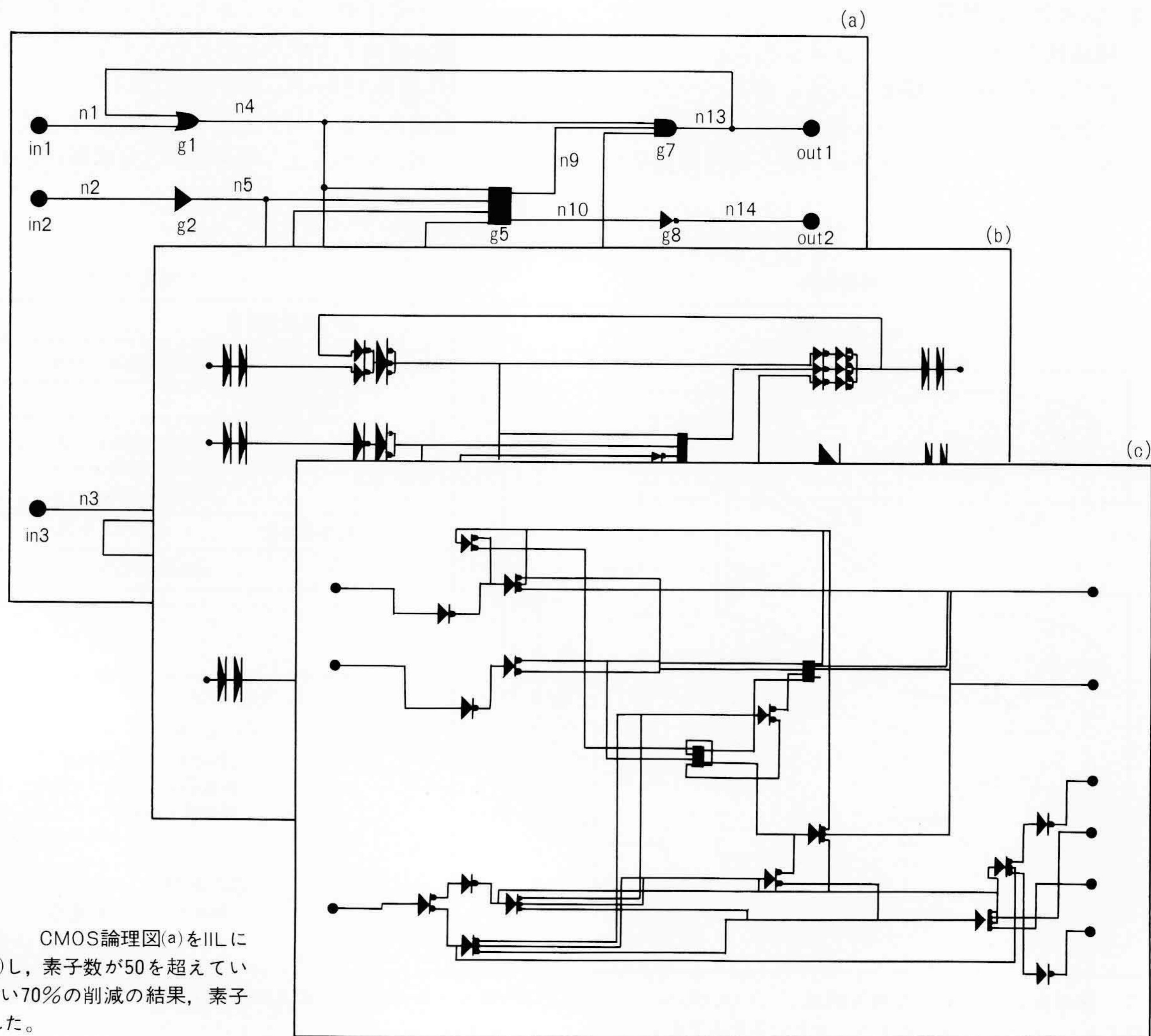


図8 回路翻訳の例 CMOS論理図(a)をIILに素子位置類似で変換(b)し、素子数が50を超えている。これに最適化を行い70%の削減の結果、素子数18の回路(c)が得られた。

計に適用を開始しており、熟練設計者と同等以上の性能を確認している。

5 論理回路検証エキスパートシステム

論理回路は、LSIとしてハードウェア化されてしまうと設計の変更が容易でないため、その設計が正しいかどうかを設計段階で検証することが重要である。仮に、論理設計が完全に正しいことが保証される自動化技術が登場したとしても、設計が部分的修正の繰返しを含む以上、設計検証の意義は変わらない。従来、設計検証は、テスト問題について論理回路の動作を計算機上で数値的にシミュレーションする方法が採られている。この論理シミュレーションは、現在のLSI設計検証に実績を持つ技術であるが、LSIの大規模化・複雑化に伴い、(1)検証するために数多くのテスト問題を用意する必要があり、(2)それらのテスト問題の組合せで設計検証に十分であるかどうかは必ずしも保証できない、という難点が顕著となる。本章では上述の課題を解消するために、最近注目されつつある定理証明法に基づく形式的検証方式を論理回路の検証に応用する技術について紹介する。

5.1 記号処理による設計検証

定理証明法による設計検証は、設計結果を仮説とみなし、その下で設計仕様を定理として証明することにより、設計仕様と設計結果に矛盾がないことを検証するものである。

定理証明法の基本的なアルゴリズムとしては、数理論理学に基づいて、命題の組合せにより矛盾を導く背理法的一种である導出法があり、記号処理技術を用いてはん用プログラムとして実現することができる。このため、既存の設計検証システムでは、はん用定理証明プログラムが使用されていた^{8),9)}。しかし、等号関係を取り扱う必要がある設計対象では、命題

の組合せの数が膨大となるため、規模の大きな推論を実施する必要がある、推論の効率化が重要な課題になっている。定理証明法による論理設計検証を効率的に実行するには、知識処理の考え方を導入し、証明過程に論理設計検証の問題特有の知識を組み込むことにより、有効な推論手順を優先的に選択する方法が有効と考えられる。

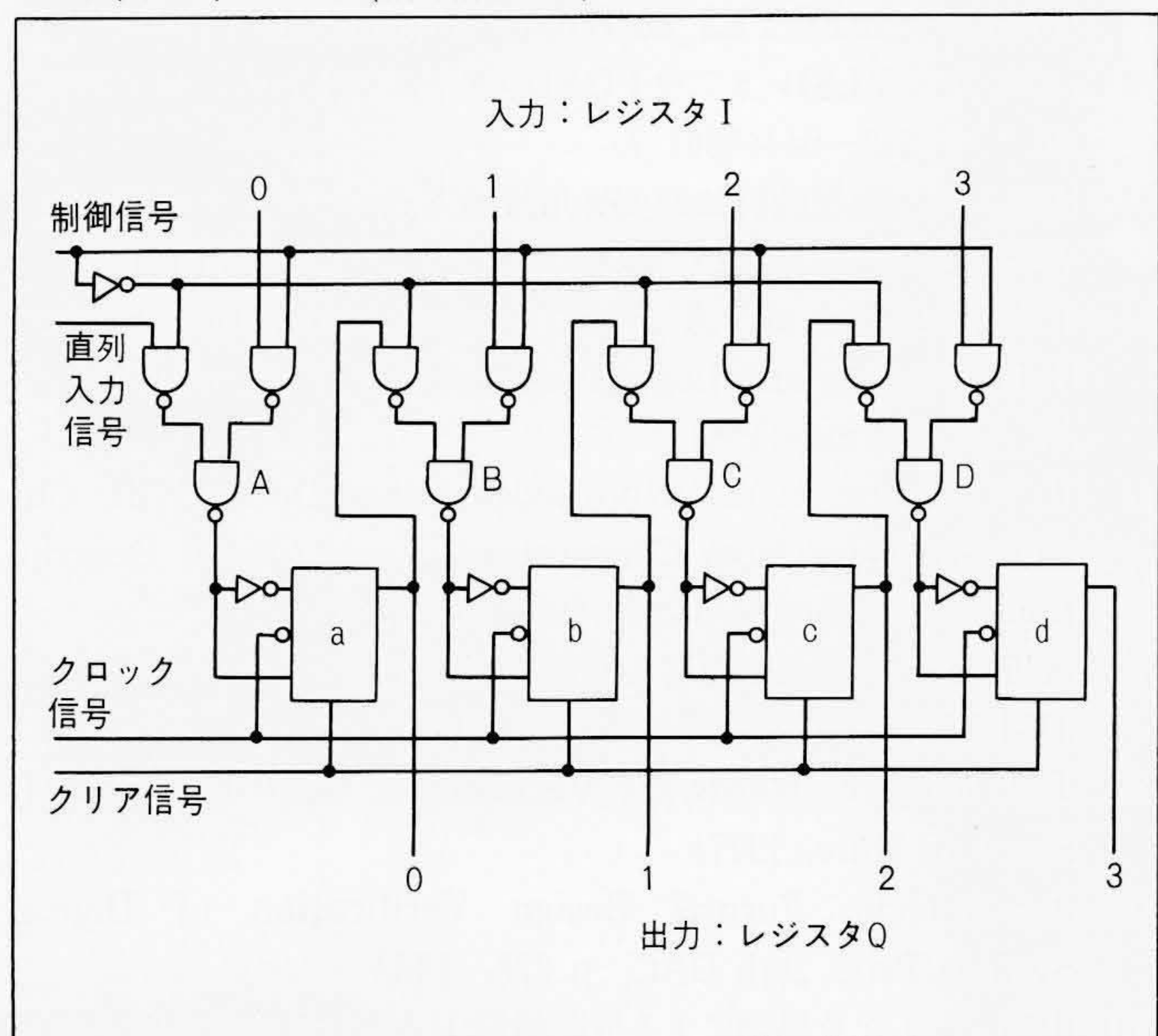
5.2 システムの構成

定理証明法による検証に、ヒューリスティクスと呼ばれる設計者の判断や経験、使用する定理の性質などの問題領域特有の知識を組み込むには、対話形定理証明プログラムが参考となる。対話形プログラムはユーザーが上位レベル、証明実行アルゴリズムが下位レベルという2階層の構造を持つ。ここでは、「対話形プログラムの自動化」という観点から、二つのレベルから成る基本構成を採用する。すなわち、定理証明プログラムを、証明過程を把握し証明手順を決定する上位レベルと、決定された手順に従って証明を実行する下位レベルの2階層で構成する。図9にシステムの構成を示す¹⁰⁾。

(1) 証明実行モジュール

下位レベルでは、命題を変形する操作である導出、あるいは命題を置換する操作である等号調整を行い、証明を実行する。効率的に証明を進めるためには、証明過程で推論可能な候補の検索を不要とし、次の推論ステップをユニークに決めていく必要がある。そこで推論法として、導出法と等号調整法を実行する結合グラフ法を採用する。結合グラフ法とは、証明過程で出現する命題の間の関係をグラフで表現して推論を進める手法である。この関係は、リンクによって表す。例えば、導出可能リンクは、ある命題とその否定を結ぶものであり、等号調整リンクは、ある命題とそれに同値なものを結ぶものである。

4ビットシフトレジスタ(設計結果の例)



注: ---> 入力, —> 実行, - - -> 出力

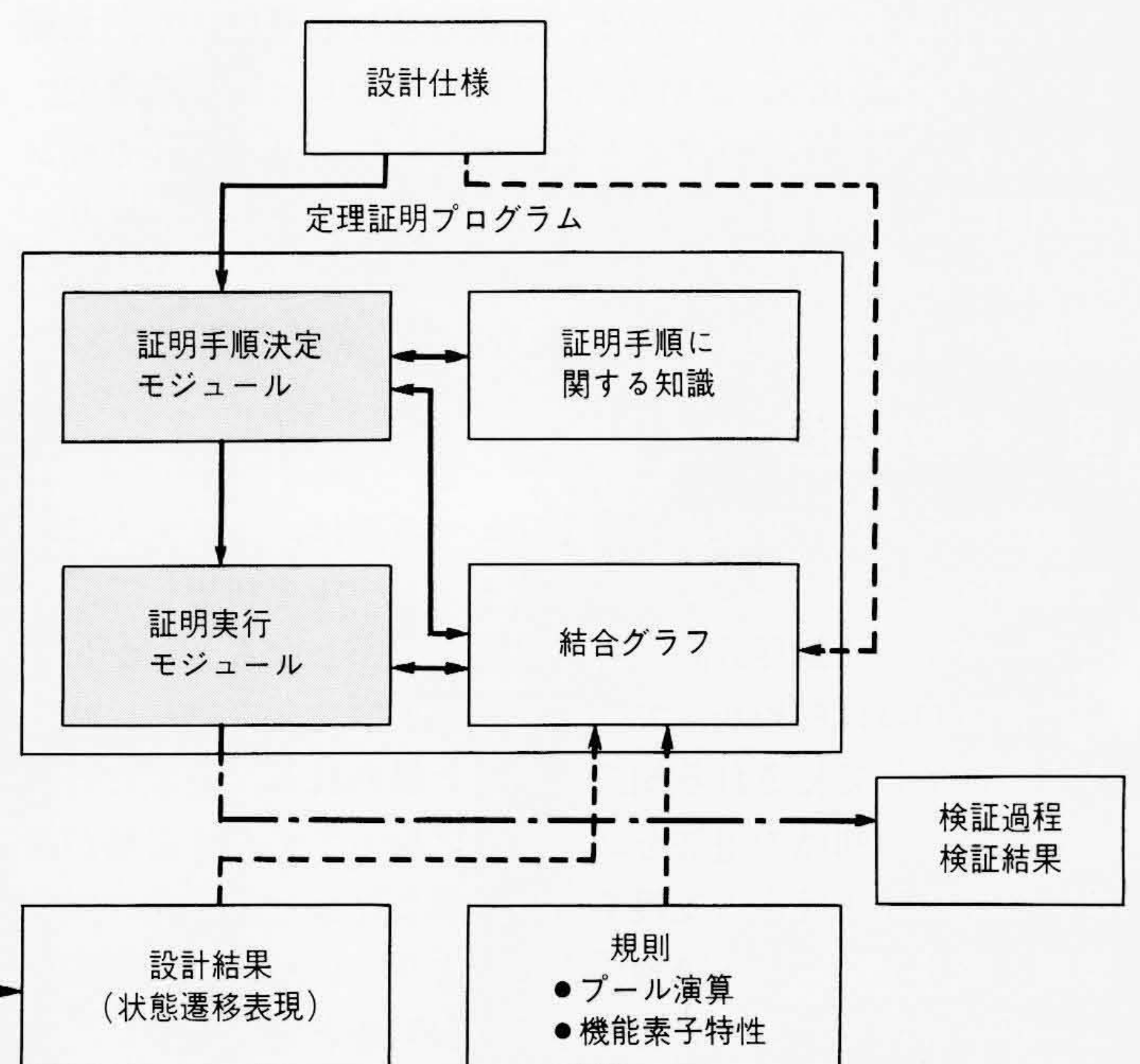


図9 論理回路検証エキスパートシステムの構成 定理証明プログラムは、上位レベルの証明手順決定モジュールと下位レベルの証明実行モジュールの2階層の構造を持つ。前者は、対話形プログラムでユーザーが果たしていた役割を担当する。

(2) 証明手順決定モジュール

上位レベルは、証明すべき定理の解釈、その時点での結合グラフの状況を把握し、これらと証明手順を結び付けるヒューリスティクスを利用して証明手順を決定する。なお、証明手順決定モジュールは、このような推論に基づいて決定された手順を、下位レベルでの処理の流れを直接ガイドする情報として、結合グラフに含まれるリンクのインデックスに変換し、証明実行モジュールで利用可能とする¹¹⁾。

5.3 システムの特徴

(1) 処理の効率化

(a) 知識を大局的な方針に関する知識(以下、ストラテジと呼ぶ。)と、その下で具体的な手順を与える知識(以下、タクティクスと呼ぶ。)に分け階層的に使用する。このため、証明手順決定の際に探索すべき範囲を限定できる。

(b) 知識処理と結合グラフ法を組み合わせることにより、知識の一部という形で、証明すべき定理、証明に使用する定理、公理の特徴をも利用している。このため、推論を知的かつ効率的に進めるように制御できる。

(c) 知識を事前に定義された関数を使用したルール形式で表現する。このため、各知識を実行可能なプログラムに変換して、推論を高速に行うことができる。

(2) 知識の表現しやすさ

知識は、手続き的に記述する必要がなく、表現しやすいものとなっている。

(a) 知識は述語論理に基づいて表現する。ストラテジ、タクティクスともに、if-then形ルールで記述している。

(b) 設計仕様、設計結果の表現には、設計記述言語として、状態遷移表現言語であるRTSを使用している⁸⁾。

5.4 適用例

本システムを数種の基本回路の検証に適用し知識処理による性能向上、自動化の効果を確認した。4ビットシフトレジスタの例では、本システムは、知識処理の効果により、文献9)のシステムに比べ、証明ステップ数で $\frac{1}{2}$ という処理効率で、与えられた仕様を満たすことを証明できた。リップルカウンタの例でも、問題領域特有の知識を直接利用することにより、ユーザーが戦略を教える文献8)の対話形システムに比較して証明ステップ数を増加させることなく、自動的に証明できた。

6 今後の展望

設計エキスパート開発の第1フェーズでは、システムに既登録の機能モジュールと制御構造モジュールを利用して、ユーザーが所望の論理設計を行う対話支援系となる。しかし、設計者の知識は初めからすべて計算機入力できないし、時とともに進歩し発見される知識もあると思われる。このため、必要に応じて知識の追加をユーザーに行ってもらい必要がある。これは知識獲得の支援であり、第2フェーズ開発の中心課題と考える。この知識獲得支援により、いっそう本格的にユーザーが利用できる高品質な設計が可能となる。

翻訳エキスパートで紹介した推論方式は実用性が高く、変換法の設計モデルに応用可能であり、設計系でも広く用いている。課題は多数の規則相互の矛盾検査の自動化である。一

方、定理証明法による論理回路検証は、現状では推論処理性能上、回路規模に制約があるが、知識処理応用による証明過程の自動化・効率化と、一部分に焦点を絞ったり、階層的に分割するなどの方式的工夫により、近い将来本格的実用に入ると期待される。

7 結 言

VLSI/CAD用の三つのエキスパートサブシステムのねらいと効果を概観し、その可能性と今後の展開を述べた。

設計エキスパートでは、オブジェクト指向のフレームと論理形プログラムを用いて、異種条件の連立や要求機能に適合して縮小可能な各種モジュールを用意することにより、人手に匹敵する良質な論理回路を短期に生成できることを例示した。

翻訳エキスパートでは、プロダクションルールと述語形ルールの組合せにより、熟練設計者以上の性能で自動化でき、実用されていることを示した。

検証エキスパートでは、証明の戦略を知識ベース化することにより、人手介入で変更のあった部分回路などを自動検証できる可能性を示した。

今後の課題として、上記機能のより広範囲な意味での早期実用化、及び設計用知的支援技術の他分野CAD/CAEへの応用展開を図りたい。

参考文献

- 1) G.Suzuki, et al.: Advanced Schematic Capture for VLSIs Using a New Graphic Data Management Method, Proc. MICAD '85 Paris(1985)
- 2) C.Fukui, et al.: Expert System for Fault Section Estimation Using Information from Protection Relays & Circuit Breakers, IEEE PES '86 Winter Meeting 112-7(1986)
- 3) 山城, 外: VLSIレイアウトCADシステム“MCDA”, 日立評論, 68, 7, 539~544(昭61-7)
- 4) 横田, 外: 論理設計知的支援用推論方式のプロトタイピング, VLSI CADへの知識工学の応用シンポジウム, (昭61-1)
- 5) T.Yokota, et al.: A VLSI Design Automation System Using Frames and Logic Programming, Proc. of Conf. of AI Applications(1987)
- 6) T.Watanabe, et al.: Knowledge-Based Optimal IIL Circuit Generator from Conventional Logic Circuit Descriptions, Proc. 23rd DAC, pp.608~614(1986)
- 7) 渡辺, 外: 知識処理による回路系自動変換システムの開発, 第33回情報処理学会全国大会, (昭61-10)
- 8) T.J.Wagner: Hardware Verification, PhD Thesis, CSD Stanford Univ.(1977)
- 9) A.S.Wojcik: Formal Design Verification of Digital System, Proc. 20th DAC, p.228, (1983)
- 10) 山田, 外: 記号処理による設計検証(1), 第32回情報処理学会全国大会 4C-1(昭61-3)
- 11) 山田, 外: 記号処理による設計検証(2), 第33回情報処理学会全国大会 7M-6(昭61-10)