ICOT Technical Report: TR-446

TR-446

評価・再設計機構を備えた論理設計支援システム

丸山　文宏、角田　多喜子、
松永　裕介、箕田　依子、
澤田　秀穂、川戸　信明(富士通)

January, 1989

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03) 456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

論　文

評価・再設計機構を備えた論理設計支援システム

正員　丸山　文宏[†]　正員　角田多苗子[†]　正員　松永　裕介[†]
非会員　箕田　依子[†]　非会員　澤田　秀穂[†]　正員　川戸　信明[†]

Logic Design System with Evaluation-Redesign Mechanism

Fumihiro MARUYAMA[†], Taeko KAKUDA[†], Yusuke MATSUNAGA[†], *Members*, Yoriko MINODA[†], Shuho SAWADA[†], *Nonmembers and* Nobuaki KAWATO[†], *Member*

あらまし　VLSI技術の急速な進歩に伴い，高品質な製品を短期間で設計することのできる高度な設計支援システムの実現が望まれている．そこで，論理設計におけるデータパス系設計と制御系設計の二つの流れに着目した，協調型論理設計エキスパートシステムの研究を行っている．その特徴は，制約条件に対する評価/再設計の繰返しを自動化する，仮説推論を用いた評価・再設計機構にある．制約条件としては，回路規模およびスピードに関するものを考える．論理設計における選択肢である，回路の各部分の実現法をそれぞれ仮説とみなし，制約条件違反は矛盾と考える．再設計は矛盾解消の手続きとして実現する．本論文では，実際の制約値によらない形式をもつ，制約条件違反に対する理由付けを定義し，これに基づく評価・再設計機構を中心に述べる．また，実験システムを実装して評価・再設計機構の動作を確認した．これにより，具体的に数値で与えた制約条件をすべて満足する回路が設計できる．ユーザが制約条件を変更して実行を繰り返せば，同一動作仕様を実現する，特性の異なる複数の回路を効率的に設計できる．

1. まえがき

VLSI技術の急速な進歩に伴い，高品質な製品を短期間で設計することのできる高度な設計支援システムの実現が強く望まれている．そのためには，従来技術では取扱い困難な，熟練設計者のもつ優れた問題解決能力を対象とした研究を進める必要がある．そして，こうした能力を生み出す基本的枠組を備えたCADシステムを構築することが重要であると考えられる．本研究の目的は，このような枠組を探り，その上に論理設計支援システムを構築することである．

熟練設計者は，VLSI設計に関する各種の知識を利用/補完し，また，詳細化，制約条件に対する評価，再設計の繰返しを見通し良く行い，高品質な設計結果を得ていると考えられる．従って，各種の知識ベースあるいは設計ツールを統合し，設計過程全体を統一的に管理する設計環境を実現する必要がある．

米国カーネギーメロン大学のULYSSESシステムは，既存のCADツールをそれぞれ知識源とした，黒板モデルに基づく設計環境である[1]．知識源は，黒板と呼ばれるグローバルなデータベース内のファイルを介して通信し合う．これは，CADツールの蓄積を有効に利用した現実的なアプローチと考えられるが，評価・再設計の繰返しをシステムが直接サポートすることはできない．

イリノイ大学では，設計過程における詳細化を上位レベルの機能仕様から下位レベルの構造仕様への変換とみなし，構造仕様を次の機能仕様としてトップダウンに設計を進めるエキスパートシステムを提案している[2]．このシステムでは，仕様および制約条件が上位から下位へ一方向に流れるという，一つの流れのトップダウン設計に限定されている．また，下位レベルから上位レベルへの設計失敗の報告に対して，どのように再設計を行うかが課題である．

筆者らは，VLSI論理設計におけるデータパス系設計と制御系設計の二つの流れに着目した，協調型論理設計エキスパートシステムco-LODEX（cooperative logic design expert system）を構築している[3]．co-LODEXの特徴は，回路規模およびスピードに関する制約条件に対する評価と再設計の繰返しを自動化す

† 富士通㈱研究所，川崎市
FUJITSU LABORATORIES LTD., Kawasaki-shi, 211 Japan

る，仮説推論を用いた評価・再設計機構にある．仮説推論は，事実と共に，後で取り消される可能性のある「仮説」を用いて行う推論で，矛盾が生じると，それを解消するために仮説を取り替える．ここでは，論理設計における選択肢である，回路の各部分の実現法をそれぞれ仮説とみなし，制約条件違反は矛盾と考える．再設計は矛盾解消の手続きとして実現する．

本論文では，実際の制約値によらない形式をもつ，制約条件違反に対する理由付けを定義し，これに基づく評価・再設計機構を中心に述べる．また，実験システムについても触れる．この評価・再設計機構により，与えられた制約条件をすべて満足する回路が設計できる．また，ユーザが制約条件を変更することで，同一動作仕様を実現する，特性の異なる複数の回路を効率的に設計することもできる．

2. システム概要

まず，co-LODEX の入力と出力を説明することにより co-LODEX が何をするシステムかを明らかにし，次に，内部の処理に注目したシステム構成を示す．

2.1 入力と出力

co-LODEX の入力は，仕様となる動作アルゴリズムおよび制約条件である．その出力は，動作アルゴリズムを実現し，かつ，制約条件を満たす，CMOS スタンダードセルの回路記述である．

動作アルゴリズムは，レジスタトランスファレベルの言語 DDL[4] を拡張したハードウェア記述言語 UHDL[5] で記述する．図 1 に，二つの自然数の最大公約数を計算するハードウェア（GCD と呼ぶ）の動作アルゴリズム[6] を示す．idle と loop は，インタバルと呼ばれ（:: の後ろが本体），DDL のステートに対応するが，その長さは 1 クロックサイクルに限定されない．STOP (rst＝0) は，rst が 0 になると idle が終了することを意味する．<- はレジスタへの転送，:= は左辺の信号線に右辺の値を載せることを示す．

```
FUNCTION: main: clk;
 idle::
   STOP(rst=0), x<-xi, y<-yi, GOTO loop;
 loop::
   IF(x=y) THEN(ou:=x, GOTO idle)
         ELSE(IF(x<y) THEN(y<-y-x)
                   ELSE(x<-x-y),
              GOTO loop),
FEND;
```

図 1　動作アルゴリズムの例
Fig. 1　Example of behavioral specification.

回路規模に関する制約条件は，「基本セルの総数が 300 以下」のように，基本セル数（ゲート数）についての条件として表現する．スピードに関する制約条件は，「最大遅延時間が 60 ns 以内」のように，遅延時間についての条件として表現する．これらはどちらも不等式の形で入力する．

ユーザは，データパス（全体の回路の中で，レジスタや演算器等の，データの保持・転送・演算を行う部分）のブロック図を入力することにより，データパス構造の概略を指定することもできる．この場合，システムは指定されたデータパスに基づいて設計を行う．図 2 に，図 1 の仕様に対する一つのデータパス例を示す．SUB，COMP，MUX は，それぞれ，減算器，比較器，マルチプレクサを意味する．

ブロック図は専用のグラフィックエディタを通して入力する．システム登録の機能ブロックには，図 2 に現れているもの以外にも，シフトレジスタ，カウンタ，ALU，加算器，補数回路，デコーダ等があり，これらを選択してその間を結線する．また，バスも登録されており，バス構成を指定することもできる．一方，データパスが指定されなかった場合，システムは，動作アルゴリズムの各オペレーションの実行に必要な機能ブロックを設け，オペレーションの同時実行可能性を解析して，機能ブロックの共用を図り，データ転送路を決定する．

co-LODEX が出力する回路記述は，CMOS スタンダードセルライブラリ中のセルから構成される回路の接続情報であり，配置・配線等の実装設計フェーズへの入力となる情報である．

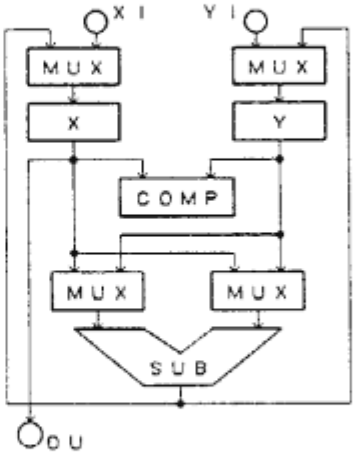


図 2　ブロック図
Fig. 2　Block diagram.

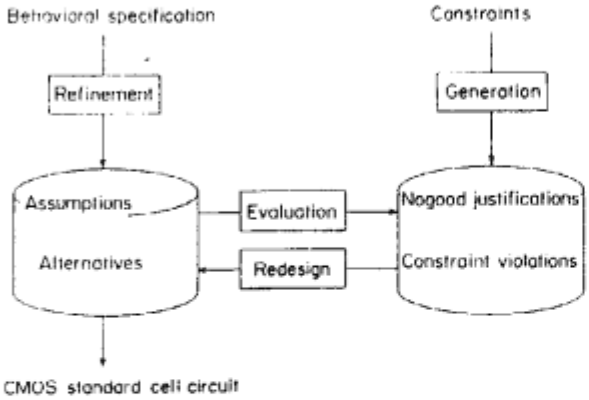


図 3 co-LODEX の概念的構成
Fig. 3 Conceptual configuration of co-LODEX.

2.2 システムの概念的構成

図 3 に，内部の処理に注目した，co-LODEX の概念的な構成を示す．動作仕様(Behavioral specification)に基づく詳細化（Refinement）では，同一機能を実現する（一般には複数の）選択肢（Alternatives）の中から一つを選択するという決定が繰り返される．従って，設計は，選ばれた選択肢の蓄積とみることができる．我々は，選択肢を仮説(Assumptions)とみなすことにする．

制約条件違反（Constraint violations）に対する理由付け（Nogood justifications 略して NJ）が評価・再設計で中心的な働きをする．ユーザが入力した制約条件は，あらかじめ暗黙の NJ に変換される．元の制約条件に違反する設計はすべてこの NJ を成立させる．この意味で暗黙の NJ は制約条件と等価である．

NJ のうち成立するものがあるかどうかを検査することで，選択肢，すなわち仮説，の集合としての設計は，制約条件に対して評価（Evaluation）される．もしあれば，新しい NJ が生成・追加され，選択肢の取替えにより，再設計（Redesign）が実行される．最終的な選択肢の集合が設計結果としての CMOS スタンダードセル回路を表す．

以下の 3. と 4. では，それぞれ，二つのエージェントによる詳細化および評価・再設計機構について述べる．

3. データパス系設計と制御系設計

VLSI の論理設計においては，データパス系の設計と動作の流れやデータパスの各コンポーネントを制御する回路（制御系）の設計は，質が異なり，区別して行われる傾向がある．co-LODEX では，データパス系の設計を担当するエージェントと制御系の設計を担当するエージェントが協調して詳細化を行う．図 4 にその様子を模式的に示す．

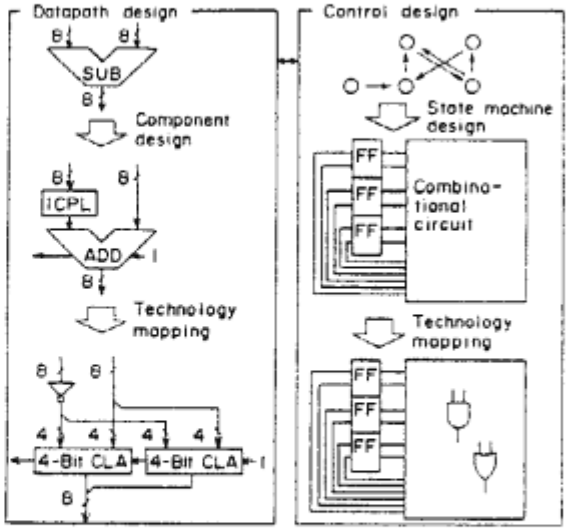


図 4 データパス系設計と制御系設計
Fig. 4 Datapath design and control design.

3.1 データパス系設計

データパス系設計とは，データパスの各コンポーネントを詳細化していき，最終的に，CMOS スタンダードセルで実現することである．データパス系設計エージェントは，各機能ブロックについて，それを他の機能ブロックで実現するコンポーネント設計（Component design）の知識をもち，これを用いて詳細化を行う．ライブラリ中のセルを用いて直接実現できるケースに対しては，セル割付け（Technology mapping）の知識をもつ．図 4 の例では，減算器を加算器(ADD)と 1 の補数回路(1 CPL)で実現するというルールが適用された．また，1 の補数回路にインバータセルを割り付けるルールと加算器に 4 ビット加算器セル（4-Bit CLA セル）を割り付けるルールが適用された．

一般に回路の実現法は 1 通りとは限らないから，上記の過程では，選択肢のうちの一つを選択していることになる．例えば，8 ビットの加算器を実現する他の選択肢には，2 ビットの加算器セルを用いるものや 1 ビットの加算器セルを用いるものがある．選択肢の違いにより，結果の回路の特性に違いが出る．

3.2 制御系設計

制御系設計エージェントは，まず，動作アルゴリズムに現れるインタバルをステートに分割することにより，動作アルゴリズムを実現するステートマシンを構

築する．次に，ステートマシンをフリップフロップ（FF）とその周りの組合せ回路（Combinational circuit）として具体化し，FFセル割付けの知識とゲート・セル割付けの知識を用いて，全体をCMOSスタンダードセルに落とす．一方，データパスのコンポーネントに対する制御信号を生成する回路も，論理式の形で抽出した仕様に基づいて組合せ回路を生成し，これに対してゲート・セル割付けを行う．

3.3 協調動作

データパス系設計エージェントは，各コンポーネントをCMOSスタンダードセルで実現したデータパスを部分解として出す．一方，制御系設計エージェントは，やはりCMOSスタンダードセルで実現した制御系回路を部分解として出す．この二つの部分解は，図5に示すように，コンポーネントの制御端子で結合され，完全な回路となる．

これら二つのエージェントは，与えられた制約条件を満たす回路を設計するために次のように協調する．

（1） データパス系設計エージェントのやり直し

制御系設計エージェントが構築したステートマシンのもとでは1サイクルで伝わるべきパスの中に，クロック周期に収まらないものがあると，データパス系設計エージェントはそのパス上のコンポーネントを設計し直す．

（2） 制御系設計エージェントのやり直し

データパス系設計エージェントがあるパスの上のコンポーネントをどのように設計しても，1サイクルに収まらない場合には，制御系設計エージェントは，ステートを増設することによりパスの分割を試み，このような遅延制約違反をなくす．

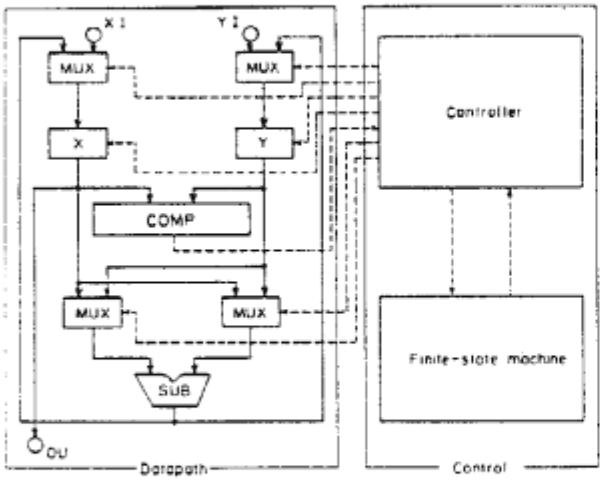


図5 データパス系と制御系の結合
Fig. 5 Linkage between datapath and control.

例えば，図1のインターバルloopを一つのステートで実現したときに生じる，図2におけるX → COMP → MUX → SUB → MUX → Xというパスを，XとYを比較した結果で新しいステートに遷移するよう変更することにより，二つのパスに分割する．

4. 仮説推論を用いた再設計

設計においては，最適な選択肢が一意的には決まらない．ある選択肢を選択しても，その後の評価で不適当なことがわかり，他のものに変更せざるを得ないこともある．しかし，選択をしていかない限り設計は進まないから，その時点で最も適当と思われるものを選んでいくのが普通である．選んだ選択肢を設計データに追加し，更に詳細化を進める．但し，制約違反が起こると，選択肢の変更によるやり直しが必要である．

我々は，このような選択肢を仮説と見立て，制約条件に対する評価を行い，違反が検出されると矛盾が起こったとしてやり直し（再設計）を行う方式を考案した．

代表的な仮説推論システムであるATMS[7]では，すべての仮説を数え上げておき，そのすべての組合せを対象としているが，ここでは，すべての組合せに意味があるわけではない．例えば，図2のデータパスの設計では，減算器を加算器で実現しない限り，加算器の実現法に関する仮説に意味はない．また，組合せだけでなく，その結果生じる回路の特性（規模やスピード）に興味がある．

以下では，制約条件違反に対する理由付け（NJ）に基づく評価・再設計機構について述べる．

4.1 階層的な設計記述

図3に示した概念的構成では，設計データに当たる選択肢の情報と評価・再設計のためのデータであるNJとは，分けられている．しかし，実際には，これらは図6に示すような階層構造に統合される．

ノードには，各コンポーネントに対応するコンポーネントノード（だ円）とその可能な実現法に対応する選択肢ノード（長方形）がある．現在採用されている実現法をinであると言い，その他の実現法をoutであると言う．コンポーネントノードは，たかだか一つしかinの選択肢ノードをもたないが，outの選択肢ノードも捨てずに保存し（例えば，図6のADD 2, ADD 3），後の再設計で必要になると引き出して使う．選択肢ノ

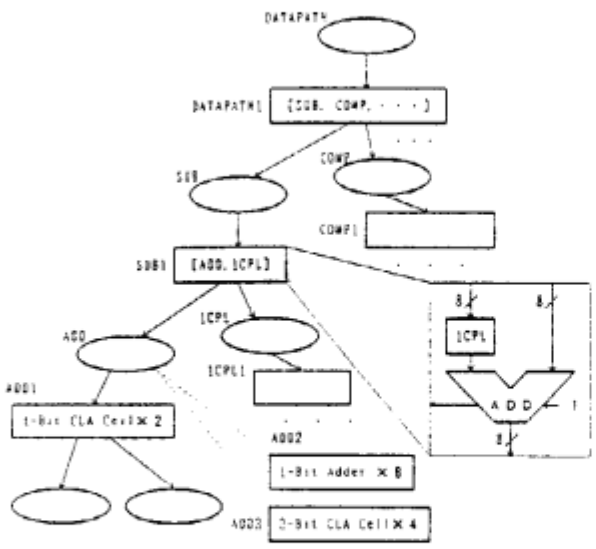


図 6 階層的な設計記述
Fig. 6 Hierarchical design description.

ードは，サブコンポーネント間の接続情報を保持し，サブコンポーネントを子ノードとしてもつ．また，NJは選択肢ノードに書き込まれる．

図6は次のことを示している．データパスは減算器，比較器，その他から構成し，減算器は加算器と1の補数回路で実現する．加算器は4ビットの加算器セル2個で実現する．

4.2 制約条件違反の理由付け（NJ）

制約条件違反の理由付け（NJ）は，設計中に成立してはならない条件式（線形不等式またはその論理式）である．その成立は制約条件違反を意味し，それが成立しなくなるよう再設計が起動される．

データパスの選択肢ノードに書き込まれた次の暗黙のNJは，制約条件と等価な条件式である．

SUB+COMP+…>DATAPATH　　（1）

これは，減算器，比較器，およびその他のコンポーネントの基本セル数の総和が変数DATAPATHの値を超えれば制約条件違反である，ことを示す．但し，DATAPATHは現在有効な制約値（例えば，300）を参照する変数である．

暗黙のNJから出発し，再設計が行われると，NJの展開あるいは合成という操作により新しいNJが生成され，選択肢ノードに書き込まれる．制約条件に対する評価は，成立しているNJがあるかどうかの検査に帰着される．

4.3 NJの展開

NJの展開は，再設計を行う際，階層構造を降下して設計変更のレベルを詳細化するのに伴い，NJを生成する手続きである．次のように定義される．

① 展開するNJに現れるサブコンポーネントのうち一つを選び，Cとする．

② NJ中のCをそのinの選択肢のサブコンポーネントで置き換える．もしCがスタンダードセルで実現されているなら，実際の基本セル数あるいは遅延時間の計算値で置き換える．

③ Cのinの選択肢ノードに降りてNJを書き込む．

今，暗黙のNJ(1)が成立したとする(制約条件違反が検出された)．(1)を展開すると，例えば，減算器を選び，加算器と1の補数回路で置き換え，減算器の選択肢ノードに書き込む．

ADD+1CPL+COMP+…>DATAPATH(2)

更に，加算器を選ぶと，スタンダードセルで実現されているから（例えば，4ビット加算器セル2個），実際の基本セル数（例えば，100）で置き換え，加算器の選択肢ノードに書き込む．

100+1CPL+COMP+…>DATAPATH　（3）

ここで，加算器の現在の選択肢(4ビット加算器セル2個）を他のものに取り換える．すなわち，加算器を作り直す．(3)は，これ以後，その条件のもとで加算器を4ビット加算器セル2個で作ることを禁止する．

①での選択に使われるヒューリスティックとしては，最も大きなコンポーネントまたは最も遅いコンポーネント，すなわち最も責任の重いコンポーネント，を選ぶというものが考えられる．

4.4 NJの合成

あるコンポーネントのすべての選択肢が制約条件違反を引き起こすとき，そのコンポーネントを含まない新しいNJが合成できる．すなわち，各選択肢を禁止するNJの論理積を作り，1レベル上の選択肢ノードに書き込む．この手続きは，導出原理(resolution principle)[8]に基づくものである．

今，4ビット加算器セルに加えて，2ビット加算器セルでも1ビット加算器セルでも制約条件に違反したとする．すると，(3)以外に，例えば，

64+1CPL+COMP+…>DATAPATH　（4）

64+1CPL+COMP+…>DATAPATH　（5）

が他の二つの選択肢ノードにそれぞれ書き込まれている(たまたま基本セル数が等しくなっている)．これ以外に選択肢はないとすると，(3)，(4)，(5)から合成されるNJは

64＋1CPL＋COMP＋…＞DATAPATH　（6）

で，減算器の選択肢ノードに書き込まれる．更に，1の補数回路にこれ以上選択肢がないなら，結局，1CPLをその基本セル数8で置き換えた，

72＋COMP＋…＞DATAPATH　（7）

が減算器の選択肢ノードに書き込まれる．ここで，減算器の構成法を変更することになる．

式(7)の72は，減算器のこの構成法には最低72基本セル必要であることを示す．式(7)は，これ以後，その条件のもとでこの構成法を禁止する．

4.5 評価・再設計アルゴリズム

NJの展開，合成を用いて，評価・再設計のアルゴリズムを定義する．

[評価・再設計アルゴリズム]

① 全体の回路（階層構造の根のノード）のinの選択肢ノードをALTとして②へ．

② ALTおよびその先祖の選択肢ノードに書かれているNJで成立しているものがあるかどうか検査する．あれば，そのNJが書き込まれている選択肢ノードをALTとして③へ．なければ⑦へ．

③ 注目しているNJの中にALTのサブコンポーネントがあれば④へ．なければ⑤へ．

④ NJの展開を行い，降りたノードをALTとして③へ．

⑤ ALTをoutにして，NJによって禁止されていない他の選択肢ノードを一つ選んでinにし，それをALTとして②へ．もしすべての選択肢ノードがNJによって禁止されているなら⑥へ．

⑥ NJの合成を行い，登ったノードをALTとして③へ．もし1レベル上の選択肢ノードがない（登れない）なら終了（失敗！）．

⑦ すべての選択肢ノードがoutになっているコンポーネントがなければ終了（成功！）．あれば，NJによって禁止されていない選択肢ノードを一つ選んでinにし，ALTとして②へ．　□

現在inの選択肢ノードをoutにして他の選択肢ノードをinにする場合，まず，現在outになっている選択肢ノードを探す．それらがすべてNJで禁止されていれば，詳細化のエージェントに新しい選択肢（構成法）を要求する．

1回の設計が終了する(つまり，上記アルゴリズムが成功/失敗で停止する)と，ユーザは制約条件を変更して再試行できる．例えば，遅延制約を強めてより速い回路を求めたり，失敗した場合に制約条件を緩めて実行させたりできる．システムは，制約値を更新し，上記アルゴリズムにより再評価を行う．再評価の過程では，それまでに書き込まれたすべてのNJが検査の対象になる．これにより，効率的に再試行できる．

5. 実験システム

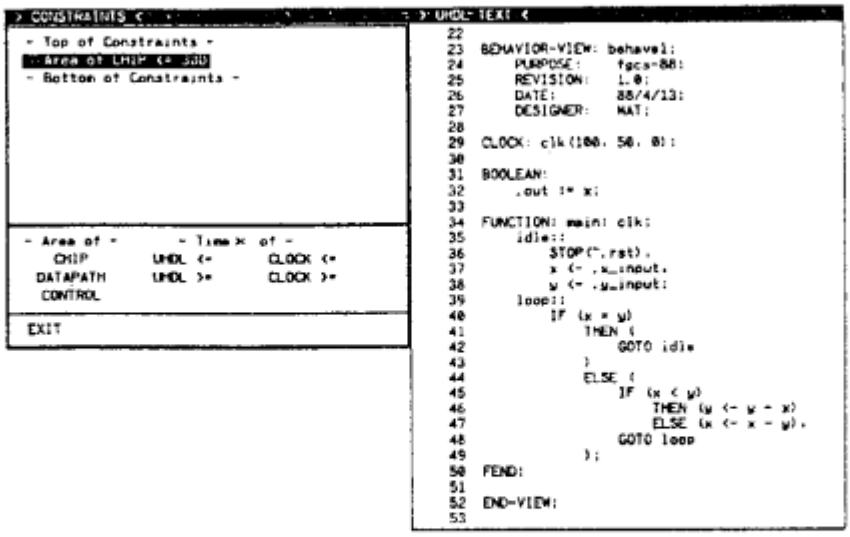
ICOTで開発された逐次型推論マシンPSI上に，言語ESP[9]を用いてco-LODEXの実験システムを実装した．データパス系設計エージェント，制御系設計エージェント，およびデータパス合成部は，いずれもルールベースシステムとして実現した．そのルールの中で，セル割付けルールは直接セルを指定する．つまり，その結論部には，ライブラリ中のセルの割付け方が書かれている．セルライブラリは，100種類余りのセルの特性データを格納するもので，各セルをESPのクラスで実現した．評価・再設計機構は4.5に示したアルゴリズムを実装した．階層的な設計記述にはESPのオブジェクト指向機能を利用した．ユーザインタフェースはカラービットマップディスプレイ上に構築した．

図7に実験システムの画面の例を示す（実際にはカラー）．(a)は入力画面で，右のウィンドウに動作アルゴリズムをテキスト形式で入力し，左側には制約条件を入力する．図は，「基本セルの総数が300以下」という制約条件を入力したところである．

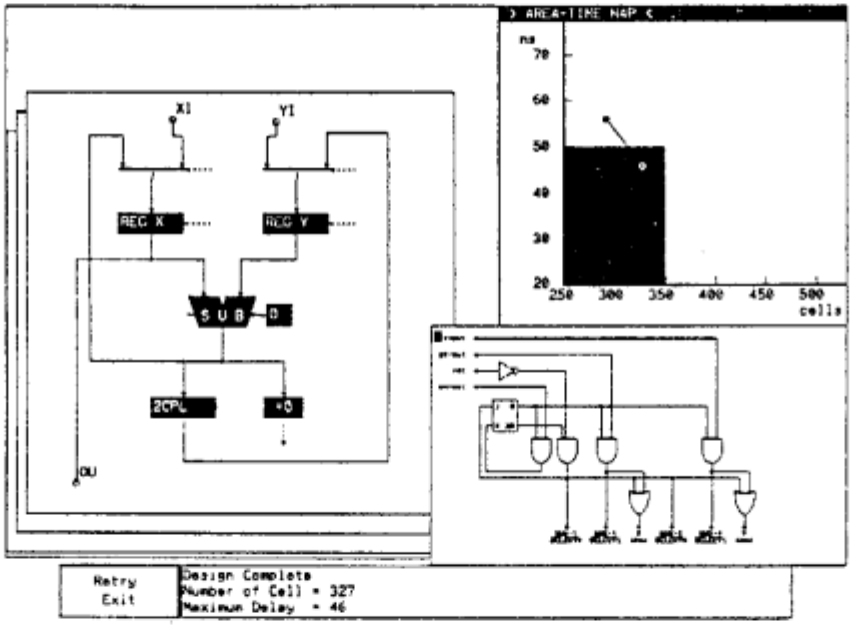
(b)は設計結果を表示している．左のウィンドウにはデータパスが示されている(このGCDの例では，動作アルゴリズムに従って六つのデータパス候補が自動的に生成された)．右下のウィンドウは，ステートマシンを含む制御回路を示す．右上の表示から，今回の制約条件（350セル以下，50 ns以内）により，はじめに設計した回路より少し大きく速い回路が設計できたことがわかる．下のウィンドウにはセル数と最大遅延時間の数値が与えられている．

処理時間は，8ビットのGCDについて30分程度であった．その内訳は，詳細化にほとんどの時間がかかり，また，その8割程度は遅延時間の計算である．再設計自体の合計時間は1回の設計で5秒以下であった．また，(制約条件を変更しての)4回の設計で，一つのデータパスにつき合計50程度のNJが生成された．

図8に，32ビットのGCDについて，制約条件を変えながら実行した結果を示す．最初は右端の回路を設計した．基本セル数の制約を強めていくと，矢印で示すように，異なる回路がいくつか設計された後最小の回



(a) Input



(b) Design result

図 7 ユーザインタフェース
Fig. 7 User interface.

路が得られた．ここから遅延制約を強め，基本セル数制約を緩めていくと，矢印のような回路系列が得られ，最後に最初の回路（最高速）に戻った．

6. むすび

データパス系設計と制御系設計の二つの流れによりCMOSスタンダードセルLSIを設計する，協調型論理設計エキスパートシステムco-LODEXについて述べた．特に，その特徴である，仮説推論を用いた評価・再設計機構について論じた．この機構により，

① 具体的に数値で与えた制約条件（回路規模およびスピード）をすべて満足する回路が設計できる．

② 制約条件を変更して実行を繰り返せば，同一動作仕様を実現する，特性の異なる複数の回路を効率的に設計できる．

現在は，実験システムを実装し，評価・再設計機構

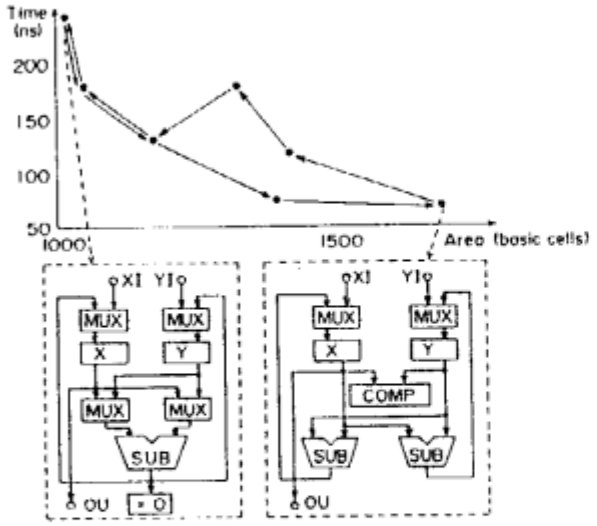


図 8　制約条件の変更による設計結果
Fig. 8　Alternative constraint designs.

の動作を確認した段階である．遅延時間の計算が処理時間の大部分を占めるといった問題点を改良すると共に，評価・再設計機構の評価を続行する予定である．また，作り直すコンポーネントの選択方式を検討することも重要なポイントである．

一方，再設計だけではなく，詳細化自身の強化も不可欠である．ルールベースの充実を図り，論理式からの自動合成等も検討していく必要がある．

また，協調型のシステムとしては，並列処理の上に拡張すること，その上で仮説推論の有効性を実証することが，今後の課題である．

謝辞　この研究は，第五世代コンピュータプロジェクトの一環として行われたものであり，御支援頂いたICOT 第五研究室藤井室長に感謝致します．

文　献

(1) M. L. Bushnell and S. W. Director: "VLSI CAD tool integration using the ULYSSES Environment", Proc. of 23rd Design Automation Conf., pp. 51-61 (1986).
(2) F. D. Brewer and D. D. Gajski: "An expert-system paradigm for design", Proc. of 23rd Design Automation Conf., pp. 62-68 (1986).
(3) F. Maruyama, et al.: "co-LODEX: A cooperative expert system for logic design", Proc. of FGCS'88, pp. 1299-1306 (1988).
(4) J. R. Duley and D. L. dietmeyer: "A digital system design language (DDL)", IEEE Trans. Comput., C-17, 9, pp. 850-861 (1968).
(5) H. Fujisawa, et al.: "UHDL (Unified Hardware Description Language) and its support tools", Int. J. Computer Aided VLSI Design (1989).
(6) P. Camposano: "Structural synthesis in the Yorktown silicon compiler", Proc. of VLSI'87, pp. 29-40 (1987).
(7) J. de Kleer: "An assumption-based truth maintenance system", Artificial Intelligence, 28, pp. 127-162 (1986).
(8) J. A. Robinson: "A machine oriented logic based on the resolution principle", J. ACM, 12, 1, pp. 23-41 (1965).
(9) T. Chikayama: "Unique features of ESP", Proc. of FGCS'84, pp. 292-298 (1984).

(昭和 64 年 1 月 7 日受付，平成元年 2 月 15 日再受付)

丸山　文宏

昭 53 東大・工・計数卒，同年(株)富士通研究所入社．昭 56～57 スタンフォード大学計算機科学科客員研究員，ハードウェア CAD，人工知能の研究開発に従事．昭 55 情報処理学会創立 20 周年記念論文賞，昭 57 本会学術奨励賞，昭 63 元岡賞各受賞．情報処理学会，日本ソフトウェア科学会，IEEE 各会員．

角田多苗子

昭 57 東大・教養・基礎科学卒，同年(株)富士通研究所入社．現在，システム研究部第二研究室所属，CAD システムの研究開発に従事，情報処理学会，人工知能学会会員．

松永　裕介

昭 60 早大・理工・電子通信卒，昭 62 同大大学院修士課程了，同年(株)富士通研究所入社．VLSI の論理設計 CAD の研究に従事，情報処理学会会員．

箕田　依子

昭 60 横国大・工・金属卒，同年富士通(株)入社．以来富士通研究所において論理設計エキスパートシステムの研究開発に従事，知識表現に興味をもつ．情報処理学会会員．

澤田　秀穂

昭61東京農工大・工・数理情報卒，同年富士通(株)入社，以来，富士通研究所において論理設計エキスパートシステムの研究開発に従事，知識表現に興味をもつ，情報処理学会会員．

川戸　信明

昭46東大・工・電子卒，昭48同大大学院修士課程了，昭51同大大学院博士課程了，同年(株)富士通研究所入社，現在，人工知能研究部長代理，この間，ディジタルシステムのCAD，人工知能などの研究開発に従事，工博，IEEE，情報処理学会，人工知能学会，日本ソフトウェア科学会各会員．