

流用設計支援エキスパートシステム：TRIP

平林 良啓 鈴木 重信 尾藤 龍茂
柿本 正夫 高橋 万年 杉本 隆夫
(株) 日本電気

本稿では、コンピュータの設計用に使用されている流用設計支援エキスパートシステム・TRIPについて報告する。本システムは、旧テクノロジーで設計された装置を最新のテクノロジーに基づいた装置に変換するテクノロジー変換を支援するものであり、LSIの切り出しサポート、ルールベースに基づく変換や論理的・物理的最適化、変換の検証等の機能及びレイアウト設計の為に考慮等を含む実用的なツールシステムである。

本システムを利用することにより、集積回路技術の急速な進歩をタイムリーにコンピュータ設計の分野に取り入れることが可能になったばかりでなく、大幅な設計工数の削減、設計品質の向上を計ることができた。

TRIP : An Automated Technology Mapping System

Yoshihiro Hirabayashi Shigenobu Suzuki Tatsushige Bitoh

Masao Kakimoto Kazutoshi Takahashi Takao Sugimoto

NEC Corporation

1-10 Nisshin-cho, Fuchu-city, Tokyo 183, Japan

24TH ACM/IEEE DESIGN AUTOMATION CONFERENCE

Session Number 29.2

1. はじめに

集積回路技術の急速な進歩により、コンピュータ設計の分野に新しい手法が取り入れられ始めてきた。新規設計ばかりでなく、テクノロジー変換という新しい設計手法が利用可能になってきた。この手法は、新規設計を行うかわりに、既に設計済みの古いテクノロジーに基づく結果を流用して最新のテクノロジーに基づく装置を再設計するものである。テクノロジー変換には、プリント板の一部のLSI化、複数のプリント板又はLSIのLSI化、バイポーラLSIからCMOSLSI又はその逆等の様々なバリエーションがある。

テクノロジー変換はいわゆる論理合成の一つの応用と考えられる。過去10年余りの間に、個別テクノロジーへのマッピングを含む種々の論理合成システム[(7)-(12)]が発表されてきた。それらの多くはテクノロジーに依存しないハードウェア記述言語を入力とするが、いくつかは個別テクノロジーに依存した接続記述を入力することができ、テクノロジー変換に使用できる。しかし、これらの論理合成システムを応用したテクノロジー変換では主に論理の変換、最小化といった点を強調しており、実用上問題となる切り出し、或るいはレイアウト設計へのインタフェースといった点はあまり考慮していない。

本稿では、日本電気で既存のコンピュータ装置を最新のテクノロジーに適合するよう流用設計を行うのに使用されている流用設計支援エキスパートシステムTRIPについてその構成、手段、使用例等を報告する。本システムはLSIの切り出しから実装設計までを含む実用的なトータルシステムであり、過去2年余りの間に多くのLSI設計に使用されてきた。本システムの特徴は以下の通りである。

- (1) ルールベースに基づくテクノロジー変換
- (2) ルールベースに基づく論理的・物理的最適化
- (3) 装置レベルの切り出しのサポート
- (4) 装置レベルの変換
- (5) 変換の検証機能
- (6) 実装構造のマッピング
- (7) 実用的システムであること

以下の章では本システムの設計フロー、機能、実際の使用例等を大幅に設計作業の削減を実現するうえでのポイントに主眼を置いて紹介する。

2. システムの概要

本流用設計支援エキスパートシステムは日本電気におけるコンピュータの設計に用いられる既存のCADシステム[(1)-(6)]の一部とみなすことができる。既存のツールに加えて切り出しのサポート、回路変換等のツールが本システムの為に開発された。

図1は、本システムの概略フローを示している。論理設計から実装設計までのすべての設計データはホストコンピュータ上のデータベースに格納されている。まず、装置レベルの切り出しツールが、LSIからバックボードまでの全ての実装階層のデータを分割して、新テクノロジーにおける仮想的なLSIの設計データを作成する。ある種のケースでは、回路図エディタを用いて仮想的なLSIの設計データを作成する。これらの仮想的なLSIチップは旧テクノロジーに基づいているが、新テクノロジーにおける入力ピン数、セル数、消費電力等の回路制限は満足している。次に各仮想LSIチップの接続記述データに対して回路変換プログラムによりルールベース中のルールが施される。回路変換はテクノロジー変換、論理的最適化、物理的最適化から成っている。このうち、

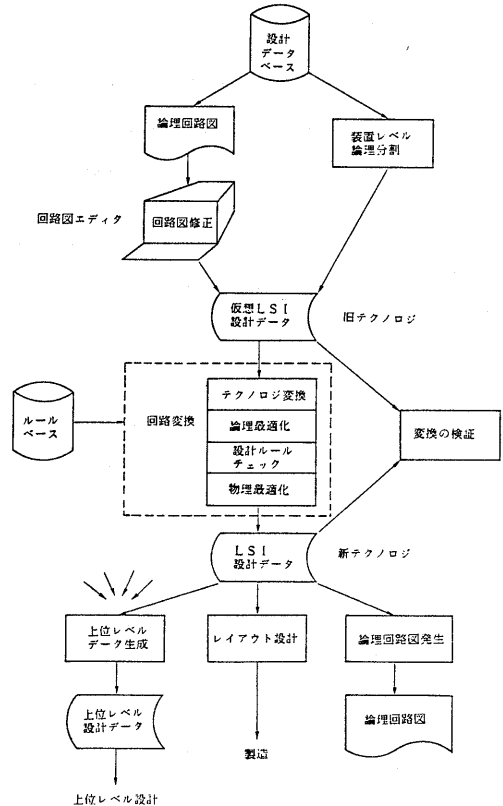


図1 システム概略フロー

物理的最適化は回路規則への適合を自動的に行うものであり、回路規則チェックプログラムの結果を利用している。次に回路変換前後のシミュレーション結果の比較により変換前後の接続記述データの等価性を検証する。最後に、レイアウト設計を実行する。レイアウト設計においてはクロック分配が自動化され、さらに、旧テクノロジーにおける実装構造が新テクノロジーのLSIのフロアプランとしてマッピングされる。さらに、変換後の回路を設計者が確認できるよう論理回路図が自動発生される。LSI以外の上位レベルの設計データは変換後のLSIの接続記述データから自動発生される。

この設計フローの中では、ほとんど全ての過程が自動化されており新規設計に比べほとんど人手を要しないようになっている。以下の章では、このフローの中のツールについて詳しく説明する。

3. 論理分割のサポート

テクノロジー変換においては、先づ始めに、旧テクノロジーにおける装置全体の論理接続データを新しいテクノロジーにおけるLSIチップ分割しなければならない。テクノロジーによってチップのピン数、ゲート数、消費電力等が異なるので分割の最適性は変換後のテクノロジーに依存する。最適な自動分割を行うのは難しいので、本システムでは2つのインタラクティブな手法を採っている。

1つの手法は既存の回路図エディタを利用する手法である。この手法は主に大規模CMOSLSIや高密度プリント基板をいくつかのLSIチップに分割する際に用いられる。旧テクノロジー回路図における必要な部分が、回路図エディタを用いて新しいテクノロジーにおけるLSIチップとして切りだされる。結線の切断・入出力端子の発生・領域の削除等特別なコマンドがこの手法の為に用意されている。入出力ピン数・消費電力等の情報が分割の最適性をチェックするために提供される。

もう1つの手法は装置レベルの切り出しである。これは旧テクノロジーにおける装置全体が、新テクノロジーにおける装置を構成する全LSIチップに分割されることを意味する。この手法は主に既存の装置全体を最新のテクノロジーに基づいて再設計する場合に用いられる。このような場合、回路図エディタを用いてLSIチップを切り出していくのでは多くの人工工数を必要とするので、新しい切り出しツールを開発した。図2は、このツールを使った分割手法の流れを示している。全ての実装階層における接続データが階層展開された後、サブモジュールが定義される。サブモジュールは論理的に結合関係の強いプリミティブブロックの集合であり、分割オペレーションを行う間、これが最小の単位として扱われる。サ

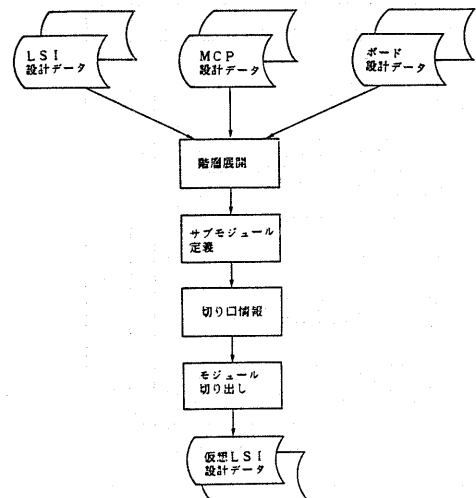


図2 装置レベル論理分割の流れ

ブモジュールを最小単位として扱うことで、分割オペレーションの回数の削減を図ることができた。サブモジュール定義の後、モジュールの切り出しが行われる。これはモジュールに含まれるサブモジュールを示すことによって行われる。ここでモジュールは新テクノロジーにおけるLSIチップに相当する。もし、このモジュールが新テクノロジーにおける回路制限条件を満足すれば、LSIの論理接続データが生成される。最適なモジュール分割を支援するため、サブモジュール間の結線数の情報等が提供される。さらに新テクノロジーにおける装置の合成を可能にするため、モジュール間のインタフェース信号名は装置全体を通じて保存される。これによりMCP、ボードなどの上位レベルのコンポントの論理データの合成は自動化することができた。

このツールは、切り残しの有無をチェックし、プリミティブブロック間の接続を乱すことが無いので、新テクノロジーの装置が切り出されたモジュールのみを使って構成されれば、それは旧テクノロジーにおける装置と等価である。

4. 回路変換

4. 1. ルールベース

回路変換は本システムの中核をなす。分割された旧テクノロジーに基づく論理接続データは、ルールベースに基づいて新テクノロジーに適合するように変換される。ルールベース中の回路変換ルールは新旧両テクノロジーの組み合わせに依存する。

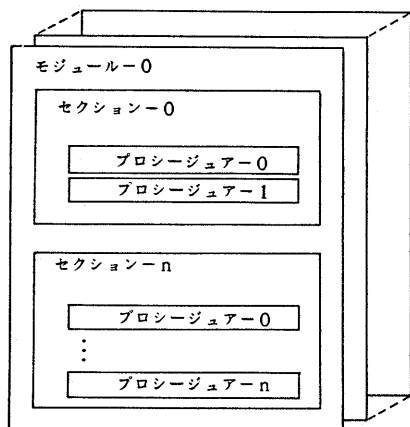


図3 ルールベースの構成

ルールベースは図3に示すように階層的に構成され、最上位のモジュールは実行単位であり、一つの新旧テクノロジーの組み合わせに対応する。次のレベルのセクションは最下位レベルのプロシージャの制御を行う。セクションは繰り返し属性を持つことができ、これがあると下位のプロシージャが変換を実行しなくなるまでセクションの実行を繰り返す。最下位のプロシージャは1つのルールに対応し、直前の変換結果に対して適用される。プロシージャもセクション同様、繰り返し属性を持つことができる。

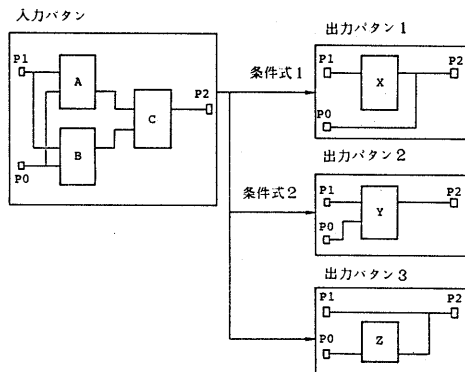
4. 2. ルールの記法

プロシージャは入力ボタン、出力ボタン、条件式から成る。入力ボタンは旧テクノロジーの回路中の変換すべき部分を示す。回路変換プログラムは、旧テクノロジーの回路中から入力ボタンに相当する部分を探し出し、これを条件式の指定に従って出力ボタンで置き換える。図4にルールの構成を示す。

入力ボタンはプリミティブブロック自体であってもよいし、複数のプリミティブブロックが相互に接続された回路であってもよい。入力ボタン中には3種類のシンボルを記述することができる。

1) Wholeタイプシンボル：旧テクノロジー回路中の対応するシンボルのすべての結合関係が、入力ボタン中のシンボルと一致しなければならない。

2) Partialタイプシンボル：入力ボタン中に記述された結合関係のみ対応する旧テクノロジー中の回路中のシンボルと一致しなければならない。あとで説明する図6(b)は、端子B以外の接続関係を考慮しなくてもよいのでこのタイプのシンボルを使用している。



```

//PROC 入力ボタン
IF 条件式1 THEN 出力ボタン1;
IF 条件式2 THEN 出力ボタン2;
                出力ボタン3;

```

図4 ルールの記法

3) Ambiguousタイプシンボル：Partialタイプシンボルの性質に加えて、プリミティブブロックの種類名が入力ボタンと旧テクノロジー回路とで一致することを要さないもの。図6(c)と(d)のルールは端子C以外の接続関係を考慮しなくてよく、入力ボタンと旧テクノロジー回路とでシンボルの種類名の一致を要さないのでこのタイプのシンボルを使用している。

WholeタイプとPartialタイプのシンボルは代替シンボル名を定義することができる。これらの定義は、入力ボタンと旧テクノロジー回路とで等価なプリミティブブロックの種類名を示し、ルール数の削減に役立っている。

条件式中には次のような条件を記述することができる

- 1) 空き端子であるかどうか
- 2) 0又は1クランプ端子であるかどうか
- 3) ワイアード接続であるかどうか
- 4) 空き端子、クランプ端子を除いた入出力端子の数
- 5) 出力端子のファンアウト数
- 6) 特定のシンボル、ネット、端子が過去のチェックプログラム実行の結果エラー属性をもっているかどうか。

最後の条件は重要である。回路変換プログラムがすべての回路規則をチェックするのは困難なので、本システムでは既存の回路規則チェックプログラムを活用している。もし、回路中のある部分が回路規則を満足しないと、回路規則チェックプログラムがその部分にエラー属性を与える。エラー属性を持っているかどうかは条件式中の条件として記述することができ、回路変換プログラ

ムはこのエラー属性を認識することができる。この方法によってファンアウト数違反、最小遅延時間違反などを自動的に修正することができる。

入出力パタンはPC上の回路図エディタを使用して論理回路図として記述する。これは、ルールをコーディングするよりもはるかに効率的である。

4. 3. 変換の過程

変換の作業は次のように行われる。旧テクノロジーの回路は入力パタンにマッチする部分があるかどうか調べられる。各々のマッチした部分は回路中から削除され、代わりに条件式によって対応する出力パタンが挿入される。入力パタン中のシンボルがPartialかAmbiguousタイプであると、入力パタン中で記述していない部分の結線関係は残される。出力パタンは空でもよく、この場合入力パタンに相当する部分の削除を意味する。

入出力パタン間の対応は各パタン中の入出力端子名の対応で決められる。入力パタン中のある入出力端子に相当する旧テクノロジー回路中のネットがオープンかクランプであれば、出力パタン中に対応する入出力端子がなくてもよい。等価入出力端子も1対1対応と同様に定義できる。旧テクノロジー回路中の等価入出力端子の一部がオープンかクランプならばn対1対応を有効に利用してルール数削減を図れる。1対n対応もファンアウト調整等に利用できる。

ルール数を削減するために等価ゲート、等価端子等の定義も有効に利用される。又、変換後の回路に対応する論理回路図は自動発生される。変換後の回路をチェックしたり修正したりするためには論理回路図の見易さが必須なので、旧テクノロジーの回路図中のシンボル名や信号名を保存する手段を講じている。この名称対応を参考にして自動発生された回路図がもとの回路図のイメージを保存できるようになっている。

4. 4. ルールの実例

ルールベース中には2種類のルールがある。1つのグループはテクノロジー変換ルールである。これらは新旧両テクノロジーの組み合わせ毎に記述される。各々のルールは旧テクノロジーにおけるプリミティブブロックが新テクノロジーにおけるプリミティブブロックにどのように変換されるかを示している。これらは1シンボル対1シンボルの対応の他に1シンボル対nシンボル、あるいはnシンボル対1シンボルの変換も含んでいる。

他のグループのルールは最適化ルールである。これらは新テクノロジーに対応して記述される。これらはさらに論理最適化ルールと物理最適化ルールに分類される。論理最適化ルールはより少ないエリアを占有するような言い換えればゲート数の少ない等価な回路への変換を定義する。最も単純なケースは冗長削除であり図5(a) - (c)に例を示す。出力未使用のブロック、連続したインバータは削除される。N-1ビットがクランプされているゲートも単純化される。図5(d)はシンボル変換を示している。一部のビットがクランプされている多ビット入力ゲートはより少ないビット入力ゲートに置換される。図5(e) - (h)は回路の統合を示している。2ビット入力ゲートが連続していると、多ビット入力ゲートに変換される。同一入力を持つ等価なゲートはマージされる。又、テクノロジー固有の複合ゲートの有効利用も図られる。ワイアードロジックが使用できる場合これも利用される。

物理最適化ルールは、回路をテクノロジー固有の回路規則に適合させるためのルールである。回路変換プログラムは回路規則を意識しているわけではなく、一般的なルー

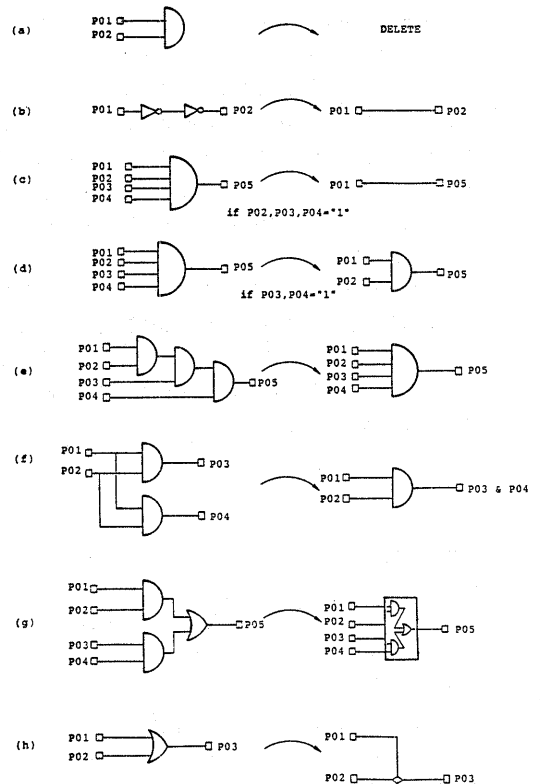


図5 論理最適化ルールの例

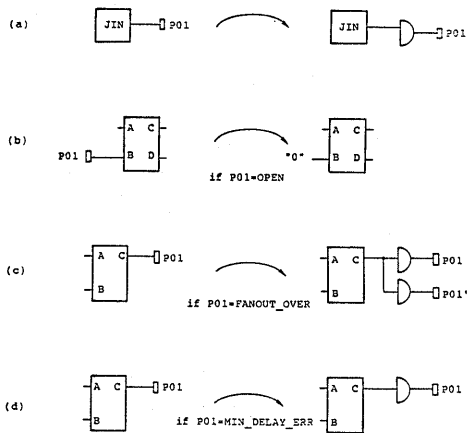


図6 物理最適化ルール例

ルの記法の中で回路規則への適合を記述する。図6 (a) - (d) にこれらの例を示す。図6 (a) は入力バッファ発生ルールの例である。図中の JIN は特殊なシンボルであり LSI の入力端子を示す。ある種のテクノロジーでは入力オープンが禁止されており図6 (b) はこれを自動的にクランプするルールを示す。残りの2つの例は回路規則チェックプログラムの結果を利用するものである。図6 (c) の場合、ファンアウトチェックプログラムが以前に実行されており、あるネットがファンアウトエラーの属性を持っていればファンアウトを分割することを示している。同様に図6 (d) の場合は、遅延チェックプログラムの結果、あるバスが最小遅延時間エラーの属性を持っていれば遅延素子を挿入することを示している。

これらの物理最適化ルールは非常に有効である。一般的に言って回路変換の結果は何らかの回路規則エラーを起こしているのが普通であり、何らかの人手介入が必須である。これらのルールは回路規則への適合作業を最小限度に抑えるのに役立つ。

5. 変換の検証

回路変換におけるルールベースのアプローチは実際的であり、柔軟性を増すのに役立っているが、ルールあるいは複数のルールのからみを検証するのが困難だという欠点がある。本システムをより実用的にするためには変換の検証作業が必須である。

当社においては、論理記述と構造記述の等価性を検証するシステムが既に開発されており、回路変換における変換前後の回路の等価性を検証することは、これのアナログと考えることができる。図7は変換の検証ツールの

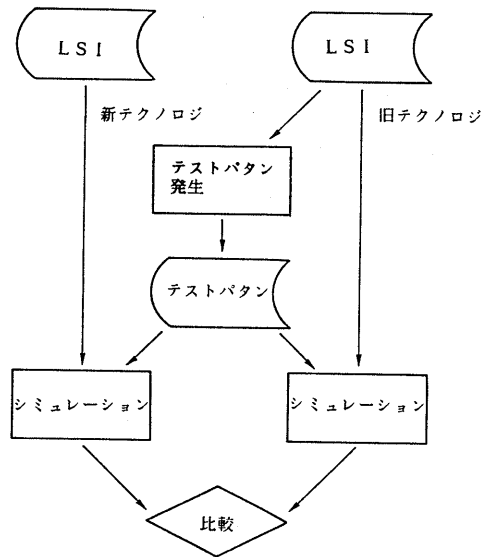


図7 変換の検証の流れ

様子を示している。新テクノロジーの回路に対して機能テスト自動発生システムによってテストパターンが自動発生される。同一のテストパターンが新旧両テクノロジーの回路に対して与えられシミュレーション結果が自動的に比較される。

回路の等価性がシミュレーション結果の照合で検証される場合はあるパターンでどの程度の等価性が検証されたかを量的に把握することが重要である。この場合単一編退故障モデルに基づく故障検出率が検証の尺度として利用されている。テストパターン発生にはDアルゴリズムが使用されており、発生されたテストパターンは製造時に機能テストパターンとしても利用される。

この検証ツールは非常に有効であり、多くの設計者によって作成される柔軟なルール記述を検証するのに役立っている。

6. レイアウト向上のための施策

レイアウト設計は基本的に新規設計と同様である。しかしながら、論理設計データ（テクノロジー変換の場合は変換後回路データ）は新規設計に比べ、レイアウト設計者にとっていくらか把握しにくい。そのような状況での人工工数を軽減するため2種類の施策を本システムでは採用している。

一つはクロック分配の自動化である。新規設計においては、論理設計者はクロックキーをできるだけ小さくするように設計を行う。論理的に近いフリップフロップに

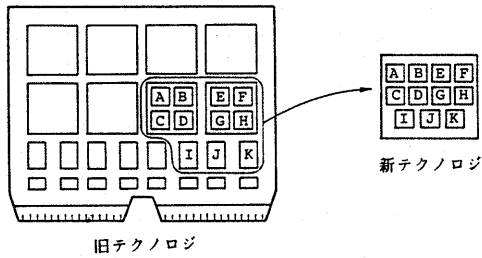


図8 実装構造のマッピング

同一のクロックを分配し、レイアウトプログラムはこれらの論理的に近いフリップフロップを物理的に近くに配置するので結果的にクロックスキューは小さくなる。テクノロジー変換においては次の手法でクロック信号が取り扱われる。最初に全てのクロック分配器が削除されすべてのフリップフロップに同一のクロック信号が分配される。その後ファンアウト調整ルールによって回路規則が適用される。結果的に適当な数のクロック分配器が付加され、一つのクロック信号が適当な数のフリップフロップに分配される。この過程においては論理的に近いフリップフロップに対して必ずしも同一のクロック信号が分配されるわけではない。この欠点を補うため、レイアウトプログラムは配置完了後近接して配置されたフリップフロップには同一のクロック信号を分配するようクロックのつなぎ換えを行うという機能を持っている。この機能は遅延解析後の論理修正に要する人工工数を大幅に削減するのに役立っている。

もう1つの施策はフロアプランサポートである。レイアウト設計が大部分自動化されているとはいえ、フロアプラン指定は大規模LSI設計においては未だ重要な作業である。テクノロジー変換においては、旧テクノロジーにおける実装構造は過去の実装設計の結果であり、これを新テクノロジーにおけるフロアプランとして自動利用できる。本システムでは実装構造のマッピングは次のようにして為される。旧テクノロジーにおける各ブロックの実装位置情報は論理分割及び回路変換の間保存となる。レイアウト設計時にフロアプランの指定をもとの実装構造上の言葉で行うことができる。実装構造マッピングの基本的な概念は図8に示してある。この機能はいかなるテクノロジーでもチップの配線性向上に有効である。

7. 使用例

この流用設計支援エキスパートシステムは1984年以来使われており、10kから20kゲートのLSIが20個以上このシステムを使って設計されてきた。旧テ

クノロジーはCMOS、バイポーラLSIやプリント基板上のTTL、ICであり、新テクノロジーは最新のCMOS、バイポーラLSIである。130kゲートの設計量のある専用コンピュータがこのシステムを用いて最新のテクノロジーに基づく装置に再設計された。テクノロジー変換は装置全体にわたって適用された。その時の経験によると詳細設計以降に要した工数は新規設計の場合に比べ10分の1に減った。又、詳細設計以降のバグ数も約40分の1に減少し、設計品質の向上を計ることができた。

実際に同じ装置については新規設計が行われていないのでテクノロジー変換の性能を計ることは難しいが、変換前後のゲート数を比べると1.1から1.3倍増加している。必ずしも新旧両テクノロジーでゲート数の概念が同じわけでないが、このゲート数の増加は初期設計が旧テクノロジーに基づいて行われているという事実に起因していると考えられる。この結果は設計工数の大巾な削減に比して設計者を十分に満足させるものであった。表1はCMOSLSIからバイポーラLSIへのテクノロジー変換の例を示している。冗長度は新旧両テクノロジーのゲート数を比を意味している。使用計算機はACOS1000である。

表2は上例での種々の最適化のルール数を示している。これらのルールによって平均30%のゲート数が減少した。

表1 テクノロジー変換の例

LSI 種類	ゲート数		冗長度	処理時間
	変換前	変換後		
A	3059	3957	1.29	7.4分
B	3230	4092	1.26	6.9分
C	4226	5510	1.30	4.5分
D	4224	4665	1.10	4.4分

表2 最適化ルール数

最適化	ルール数	
論理最適化	冗長削除	41
	シンボル置換	14
	回路統合	59
物理最適化	入出力バッファ発生	3
	ファンアウト調整	2
	最小遅延時間補償	1

このシステムは、他の設計の側面にも影響を与えた。最適化ルールは論理マクロを使用しているような新規設計でも使用された。一般に論理マクロ中の一部の回路は実際には使われないことがあるので、これらを論理最適化ルールで削除することができる。又、回路規則エラーの自動修正は新規設計でも有効に利用することができる。他の側面は教育的効果である。最適化ルールを記述することによって設計者は自分の知識を定式化することができる。

8. まとめ

以上述べたように、既存のコンピュータ装置を最新のテクノロジーを用いた装置にタイムリに再設計する流用設計支援エキスパートシステムを開発した。このシステムでサポートされている機能は論理分割サポート、ルールベースに基づく変換・最適化、変換の検証、レイアウト設計のための施策である。変換結果は大巾な設計工数の削減に比して充分満足できるものであった。

このシステムの結果として、テクノロジーの進歩を早期にコンピュータ設計の分野に取り込むことができた。今後は最適化ルールの拡充を図っていく予定である。

参考文献

1. W.H.Joyner Jr., H.Trevillyan, D.Brand, T.A.Nix and S.C.Gundersen, "Thechnology Adaptation in Logic Synthesis", Proc. of 22nd DAC, 1986, pp.94-100.
2. K.Enomoto, S.Nakamura, T.Ogihara and S.Murai, "LORES-2: A Logic Reorganization System", IEEE Design & Test of Computers, Vol.2, No.5, Oct. 1985, pp.35-42.
3. T.Shihsha, T.Kubo, M.Hikosaka, A.Akiyama and K.Ishihara, "POLARIS: Polarity Propagation Algorithm for Combinational Logic Synthesis", Proc. of 21st DAC, 1984, pp.322-328.
4. J.L.Gilkinson, S.D.Lewis, B.B.Winter and A.Hekmatpour, "An Automated Technology Mapping", IBM J. Research & Development, Vol.28, No.5, Sept. 1984, pp.546-556.
5. J.A.Darringer, D.Brand, J.V.Gerbi, W.H.Joyner Jr. and L. Trevillyan, "LSS: A System for Production Logic Synthesis", IBM J. Reseach & Developmant, Vol.28, No.5, Sept. 1984, pp.537-545,
6. J.B. Bender, "Design Through Transformation", Proc. of 20th DAC, 1983, pp.253-256.

7. S.Suzuki, K.Takahashi, T.Sugimoto and M.Kuwata, "Integrated Design System for Supercomputer SX-1/SX-2", Proc. of 22nd DAC, 1985, pp.536-542.
8. T.Sasaki, S.Kato, N.Nomizu and H.Tanaka, "Logic Design Verification Using Automated Test Generation", Proc. 1984 International Test Conference, 1984, pp.88-4.
9. M.Nomura, S.Sato and N.Takano, "Timing Verification System Based on Delay Time Hierarchical Nature", Proc. of 19th DAC, 1982, pp.622-628.
10. S.Yabe, M.Nomura, N.Takano, H.Kawanishi, H.Yoshizawa, H.kato and S.Noda, "MASTER2: A Hierarchical Layout System for Gate Arrays", Proc. of ICCAD '83, Sept. 1983, pp.46-48.
11. M.Kawai, H.Shibano, S.Funatsu, S.Kato, T.Kurobe, K.Ookawa and T.Sasaki, "A High Level Test Pattern Generation Algorithm", Proc. 1983 International test conference, 1983, pp.346-352.
12. S.Kato and T.Sasaki, "FDL: A Structural Behavior Description Language", 6-th International Symposium on Computer Hardware Description Language and Their Application, 1983, pp.137-152.
13. S.Suzuki, T.Bitoh, M.Kakimoto, K.Takahashi and T.Sugimoto, "TRIP: An Automated Technology Mapping System", 24th DAC, 1987, pp.523-529.