

標準 IC/PLD 回路から LSI 回路を生成するシステム TL/C

TL/C : LSI Circuit Generator from Standard-IC/PLD

Circuit Descriptions

井川 智、岡崎 薫、喜納 久行、浦 正久、宮崎 守弘、五十嵐 祥晃

Satoshi IKAWA, Kaoru OKAZAKI, Hisayuki KINO, Masahisa URA

Morihiro MIYAZAKI, Yoshiaki IGARASHI

松下電器産業(株) 無線研究所

Wireless Research Laboratory, Matsushita Electric Industrial Co., Ltd.

あらまし 標準 IC/PLD 回路から CMOS-LSI 回路を生成するシステム TL/C を開発した。

本システムは、標準 IC に加え、動作記述により動作が規定された PLD を含む回路図を基に、素子位置及び配線経路類似でしかも最適化された LSI 回路図を生成する。システムのテクノロジーに依存する部分はルール・ベース及びライブラリといった形で実現されているため、他の回路系変換にも容易に対応できる。

本システムの利用により、LSI 設計の大幅な工数削減と、設計品質の向上を計ることができた。

Abstract This paper describes the design and implementation of a technology mapping system called TL/C. It transforms a standard-IC/PLD circuit into a CMOS LSI circuit. TL/C has the following features. First, it generates a schematic diagram of the LSI circuit that resembles the source schematic of the standard-IC/PLD circuit. Second, it can treat circuits that include Programmable Logic Devices (PLDs).

By using this system, the time taken to design LSIs is shortened and the quality is improved.

1. はじめに

近年、半導体技術の急速な進歩により LSI の高集積化が著しい。これに伴い、LSI 設計工数の増大が大きな問題となってきており、特に論理設計における工数の低減と設計品質の向上は、計算機設計支援における重要な課題となっている。

この課題に対処する一つの手段として、従来よりテクノロジー変換と呼ばれるシステムが発表されている [4-6]。これらは既存の論理回路を別のテクノロジーに基づく論理回路に変換するものである。しかし、従来のこれらのシステムでは、①変換前後の回路図において、素子位置、配線経路及び束線の対応付けが十分ではなく、変換後の回路のチェックや修正が容易でない、②プログラマブルな素子（例えば PLD）を扱うことができない、といった問題点を有していた。

本稿で述べるシステム TL/C (Standard-IC/PLD to LSI Converter) [1-2] の第 1 版は、標準 IC/PLD 回路を CMOS-LSI に変換するものであり、特に以下の点を特徴としている。

- ① 標準 IC/PLD 回路中の PLD に対して、ブール方程式、真理値表、ステートダイアグラムのいずれか、もしくはこれらの組合せにより記述された動作仕様を基に、多段最適化された LSI 回路を生成する。
- ② 原回路図である標準 IC/PLD 回路図と、素子位置及び配線経路（束線を含む）類似の LSI 回路図を生成する事により、原回路図と生成される LSI 回路図の対応付けを容易にしている。
- ③ ルール処理において、段層的なパターンマッチング手法を採る事により、多種の変換パターンを高速に検索する機能を有している。
- ④ 知識ベースを構成するルール及びルールの集合体であるモジュールの優先順位を規定する枠組みとして、スケジューリング・テーブルを備え、ルールもしくはモジュールの最適な実行順序決定のための試行錯誤を容易にしている。
- ⑤ 知識工学的手法の一つであるルール・ベース手法を用いており、各種のテクノロジー変換に対応が可能。

以下の章では、まずシステムの概要について述べ、その後システムの処理の流れに沿って、知識ベースの構成や回路図生成の手法等について述べた後、最後に実際の回路図に適用した例について述べる。

2. システムの概要

本システムは標準IC/PLD回路図を入力としてCMOS-LSI論理回路図を生成する。PLDについては交換の対象外とする事もできる（この場合、LSI回路図上にPLDの部品シンボルがそのまま現れる）し、また、ブール方程式、真理値表、ステートダイアグラムのいずれか、もしくはそれらの組み合わせにより記述した動作仕様からLSIゲート回路図を生成させる事もできる。

図2.1は、本システムのプログラム構成図を示している。システムは大きくアルゴリズム処理部とルール処理部に分けられる。

アルゴリズム処理部では主に大量のデータを定形的に扱う処理を行い、ルール処理部ではテクノロジーに依存する様な処理や、処理のアルゴリズムが明確でなく機能向上のための修正、拡張が随時必要となる様な処理を扱う。

また、図中、破線矢印(--->)は回路の図面データの流れを、二重線矢印(==>)は接続データの流れを示している。処理量の増加による処理時間の増大をさけるためにルール処理部においては図形データは扱わないようにしている。ルール処理部で起こる回路変更に伴う図形データの更新は、ルール処理が終了した時点でアルゴリズム処理により一括して行われる。

処理の流れに沿って述べる。

(1) 論理の展開

原回路内の標準IC及びPLDの内部論理を全て展開し、論理のみに着目した回路(GenericNET)に交換する。展開に際して領域が不足した場合には図面の拡張を行い、領域を確保した後展開を行なっている。

(2) 回路最適化

出力未使用回路の削除、不要インバータの削除や定値入力ゲートの簡約化等の冗長部の除去や、複合ゲート化処理によりLSIセル数の減少を図る。

(3) LSIセル割付

GenericNETを構成する素子をファンイン、ファンアウトを考慮してLSI回路素子に置き換える。

(4) 回路図生成

原回路図に素子位置及び配線経路類似のLSI回路を生成する。

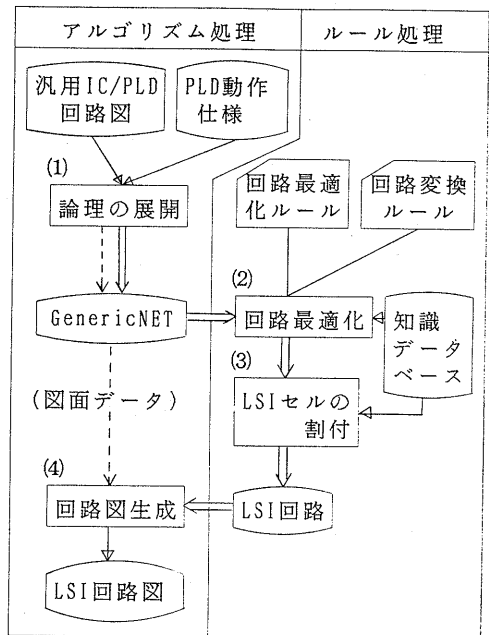


図2.1 プログラム構成図

このうち、回路最適化及びLSIセル割付の処理は、if-then型のルールの集合として定義され、知識データ・ベースに納められている。そしてエキスパート・ツールAID[3]の推論エンジンによって解釈・実行される。

3. 回路図データの読み出し

エンジニアリング・ワークステーションに入力された標準IC/PLD回路図データより回路の接続データと回路図の図面データを読み込み内部フォーマットに変換する。図面データは図面を仮想的な格子で区切り、この格子に割り付けた番号に基づいて、素子の位置や配線経路情報を量子化して作っている。これは、図形情報のデータ量を削減し、処理時間の増大を抑える事を狙いとしている。図面上に仮定する格子の間隔は部品の密度に応じて変化するようにしている(図3.1)。

4. 論理の展開

標準IC及びPLDで構成されている回路図中の標準IC及びPLDの内部論理を展開し、テクノロジーに依存しない論理のみに着目した回路であるGenericNETに変換する。後述の回路最適化の大部分はこのGenericNETを対象に実行される。これにより、他のテクノロジー交換に対応する場合のシステムの変更を最小限に

押さえている。

論理の展開は基本的には、原回路図中の部品をその内部論理に等価な回路図で置き換える事により行われる。展開に際して領域が不足する場合には図面を押し広げる様な形で展開を行っている。展開に際しての処理は標準ICとPLDの場合で異なる。

4-1 標準ICの展開

標準ICの内部論理に対応する回路図情報はあらかじめライブラリに登録されており、標準ICの品番を鍵にして検索できる様になっている。ライブラリに登録された回路図情報は、第3章において述べた図面データの内部フォーマットと同じフォーマットで格納されている。展開に際して領域が不足する場合には、図4.1で示す様にX軸方向、Y軸方向それぞれについて図面拡張を行った後、展開している。これにより、素子間の相対関係が大きく変化しないようにしている。

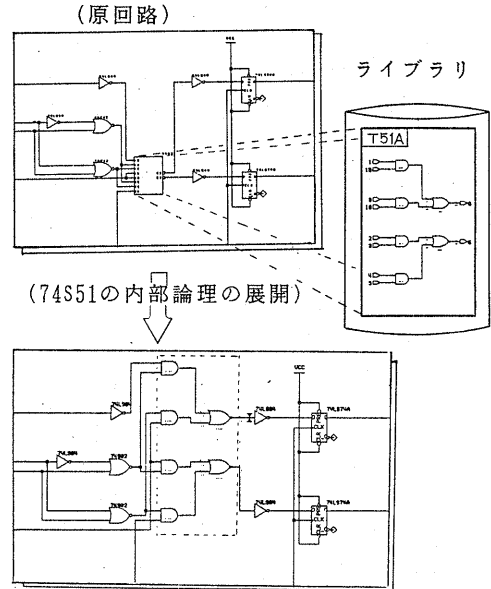


図4.1 論理の展開 (標準IC)

4-2 PLDの展開

PLDの展開におけるPLD内部論理に対応する論理回路図は図4.2に示す流れにより作られる。PLD部分の回路図生成後は先の標準ICの場合と同様の処理により展開を行う。

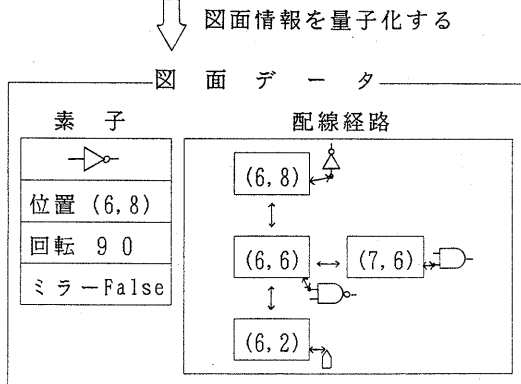
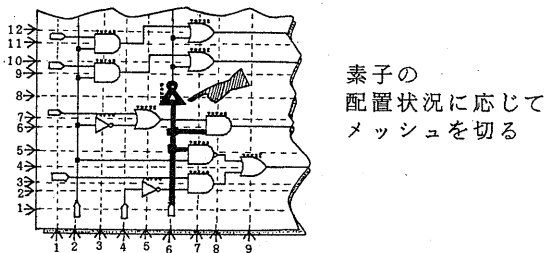


図3.1 図形データの符号化

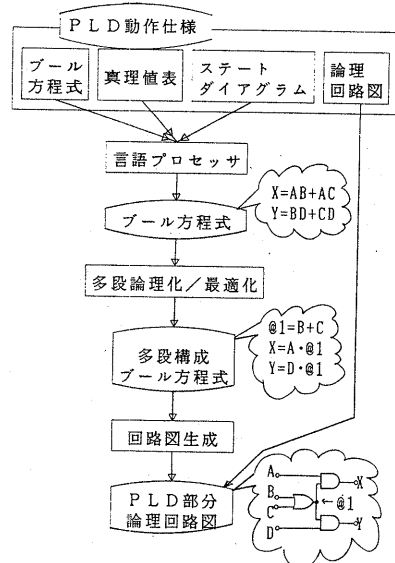


図4.2 PLD部分の回路図の生成

5. 回路最適化/LSIセル割付

回路最適化及びLSIセルの割付けは知識工学手法の一つであるルールベース処理形態を探っている。これは先にも述べた通り、テクノロジーに依存する処理や、処理のアルゴリズムが明確でなく、機能向上のための修正、拡張が随時必要となる様な処理については、このルールベース手法を用いる方が、テクノロジーの変更や機能の更新に容易に対応できるからである。

ルール処理部のシステム構成図を図5.1に示す。ルール処理部の入力となるのは、アルゴリズム処理部で作られたテクノロジーに依存しない回路である Generic NET であり、出力は CMOS-LSI 回路の接続情報である。

GenericNETはまず Black Boardと呼ばれるワーキングメモリに入力され、そして以後の回路最適化及びLSIセル割付は、この Black Board上のデータを対象にして実行される。

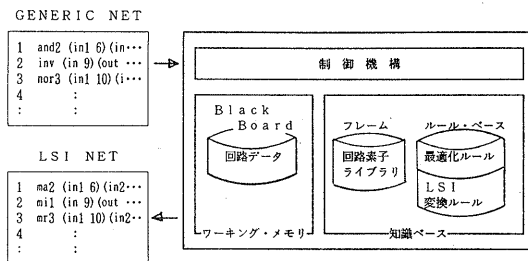


図 5.1 ルール処理部システム構成図

5-1 知識ベースの構成

(1) フレーム型知識ベース (フレーム)

フレームは、各々のLSI論理素子の入出力ピン情報やファンイン・ファンイウト数などの属性を階層的にもたせたデータベースである。上位レベル階層の知識はインヘリタンス機能を用いることで自由にアクセスできる。

(2) ルール型知識ベース

ルールベースには、LSIチップのアクティブエリアに相当するゲートのセル数を減少させる為の交換パターンを網羅している。

図5.2に示す様に、ルールベースは処理の種類によってモジュール化されている。各モジュールは更に構造的に分割された複数の知識源(KS)からなる。KSとは、ある状況において推論の対象となるひとつかたまりの知識で、IF-THEN文を呈するルールの集合により構成されている。この様にルール群をKSに分割することにより、推論の状況に応じて不要なルールの探索を行うことがなくなり、効率が高くなるというメリットがある。

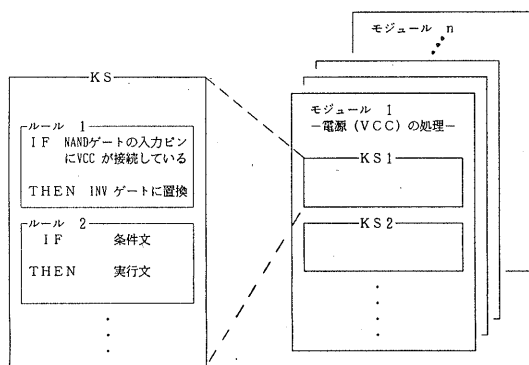


図 5.2 ルール・ベースの構造

図5.3に、知識ベースを構成するルールの例を示している。ルールは大きく3つに分類される。

① LSI用の回路規則適用ルール

- ・ワイヤードロジックの除去ルール (a)
- ・同一信号多入力ゲートの置換ルール (b)

② 回路最適化ルール

- ・定値入力ゲートの置換ルール (c)
- ・出力端末使用ゲートの除去ルール (d)
- ・出力信号線の共有化ルール (e)
- ・複合ゲート化ルール (f)
- ・不要インバータの除去ルール (g)

③ LSIセル割付ルール

GenericNETを形成している素子にFANIN/FANOUTを考慮してLSIセルを割付ける。

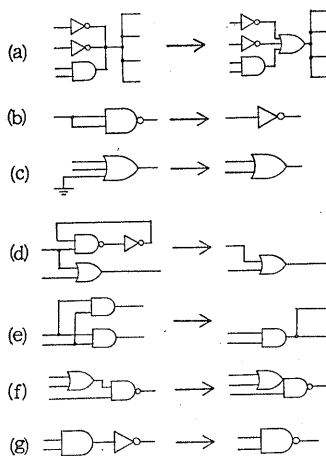


図 5.3 ルールの例

5-2 階層的パターンマッチング

ルールは基本的には、「もし回路中に、ある回路パターンAがあればそれを別の回路パターンBで置換する」という記述方法で記述される。すなわち、

IF 回路パターンA THEN 回路パターンB

ここで推論機構の動作として回路中の回路パターンAを探索するいわゆるパターンマッチングが行われるわけである。そしてこのパターンマッチングでは、対象となる回路部分を回路パターンAと照合するために、配線の分岐、接続素子や接続ピン等の複数の条件判定が必要となる。本システムではこの様な複数の判定すべき条件を階層化してパターンマッチングの高速化を図っている。

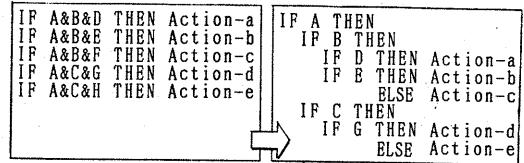
図5.4にa~eの複合ゲートに変換する場合を示す。例えば複合ゲートaに置き換える事ができる回路パターンを見出す場合の条件判定は、(i)先頭の部品がNOR2である(条件A) (ii)NOR2の一方の入力側ネットにAND3のみがつながる(条件B) (iii)NOR2の他方の入力側ネットにAND3のみがつながる(条件D)の3つであり、すなわちルールとしては、

IF A & B & D THEN 複合ゲートaとなる。

同様に他の複合ゲートb~eについてもルール化するとルールベースは図5.4(a)の様になる。このルール群を階層化するとルールベースは図5.4(b)の様になり、これを図式化したのが図5.4(c)である。

この例ではレベル1~3の階層をもたせている。破線で囲まれたルール群が1個のKSに相当し、その時々における推論の対象となる範囲を示している。そして、図中の大文字アルファベットA~Gが判定すべき条件である。

図中レベル1では、回路データ中のNOR2ゲートのみを取り出す。レベル2ではこのNOR2ゲートの入力側ネットの内の1つが分岐せず、AND3ゲートかAND2ゲートと接続しているパターンを有する回路データを取り出す。レベル3ではレベル2で取り出された回路データの中で、NOR2ゲートの残りの入力側ネットの接続関係によって、a~eの複合ゲートに置換する。ただし、条件F、Hは任意のゲートとのマッチングを示している。ある例題回路に対して階層化されているルールと、非階層化のルールをそれぞれ適用し処理時間を調べたところ、階層化により2倍の速度向上が見られた事を附記しておく。



(a) 非階層化 (b) 階層化

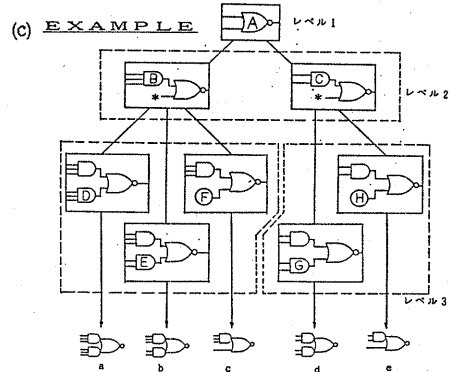


図5.4 階層的パターンマッチング

5-3 スケジューリングテーブル

知識処理システムにおいて多くの場合、知識の適用順序により処理結果や実行速度が変化する。このため最適な知識の適用順序を見出す事は重要である。

例えば図5.5においては、ルールAとルールBのいずれのルールが適用されるかによって結果が異なる。同図の場合、ルールBの適用がより効率的であるためルールBが先に適用されるべきである。しかしながら、この様な最適な適用順序を見出すためには、ある程度の試行錯誤が不可欠である。

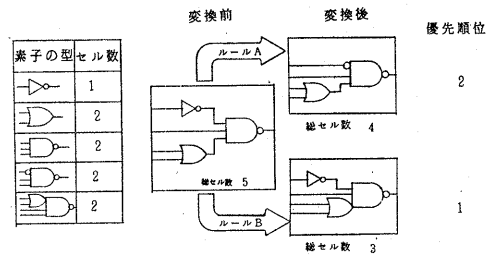


図5.5 優先順位による効率差の例

本システムではモジュール及びルールの優先順位を規定する枠組みとしてスケジューリング・テーブルを備える事により、モジュール及びルールの最適な実行順序決定のための試行錯誤を容易にしている。

スケジューリング・テーブルの構成を図5.6に示す。図に示す様に、スケジューリング・テーブルの各エントリには、各モジュールもしくはルールの優先順位、名前、コメント及び、作成更新日時情報が含まれ、

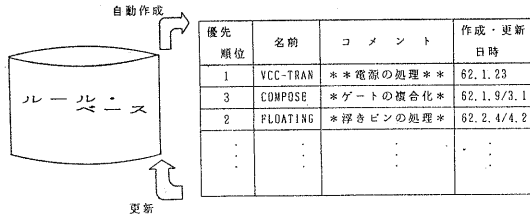


図 5.6 スケジュールリング・テーブル

知識ベース管理テーブルの役割をも果たす。スケジュールリング・テーブルを用いた知識の適用順序の変更方法を以下に示す。

- (1) 知識ベースより現時点のモジュールもしくはルールの情報を自動的に読み出し、スケジュールリング・テーブルを作成する。
- (2) 知識ベース作成者がスケジュールリング・テーブルの優先順位番号を変更する。
- (3) スケジュールリング・テーブルの内容に基づいて知識ベースを自動的に更新する。
- (4) 新しい優先順序による実行を行う。

知識ベース作成者は結果を評価しながら(2)~(4)を繰り返す事により、最適な知識の適用順序を見出す事ができる。

6. 回路図生成

本章で述べる処理では、前段の処理から引き渡された図面データに基づいた回路図生成を行う。第2章で述べた様に、原標準 IC / PLD 回路図から抽出された図面データは展開処理や回路最適化処理等の過程において更新されるが、この図面データの更新は出来るだけ元の標準 IC / PLD 回路図の図面データ、すなわち素子の相対的な位置関係、配線の経路、束線の情報等を尊重する形で行われる。これにより、生成される回路図は変換元の標準 IC / PLD 回路図に素子位置及び配線経路類似のものとなるわけである。

以下処理の流れに沿って述べる。

(1) 配置

上記図形情報に基づき、仮想的な格子で区切られた図面上に素子を配置する。素子の図面データとしては、各素子の位置と回転をして X 軸対称のミラーの有無といった原標準 IC / PLD 回路図から抽出された情報に基づくもの他に、素子の大きさを表わすデータが含まれる。素子の大きさの情報は上記格子を単位として、いくつの格子に又がるかという数で表わす。複数の素子が同じ位置に重なる場合には、信号の流れを考慮しながら近傍の空き領域に素

子を移動させ重なり合いを解消している。

(2) 概略配線

上記図面データには格子で区切られた領域を単位とする概略の配線経路情報が含まれているので、これに基づき概略の配線経路を決定する。図面データ内の配線経路情報は、論理の展開や、回路最適化といった途中の処理の過程で一部欠落することがある。この場合は、迷路法 [7] により最短経路を求め、欠落した情報を補っている。図 6.1 に図面データに基づく配置と概略配線の例を示す。

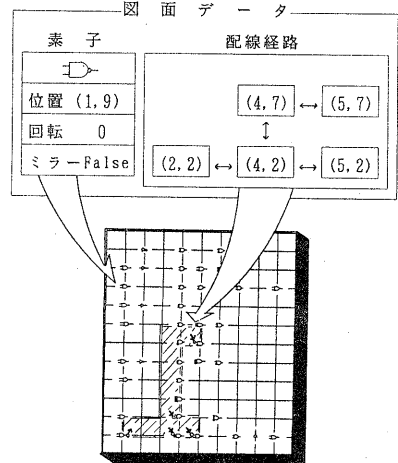


図 6.1 配置及び概略配線

(3) 本配線

概略配線経路内の領域を用いて実際の配線経路を決定する。格子間を通る事ができる配線の数はあらかじめ決まっており、上記概略配線経路内での配線が不可能な場合には概略配線経路の近傍の領域を用いて配線を行っている。

(4) 図面圧縮

(1)における格子の間隔は配線に必要な領域を考慮して大きめに設定されている。このため本配線終了時の回路図は少なからず無駄な領域を持っており、不必要に大きくなっている。そこで最後に素子及び配線経路の相対関係を保持しながら図面の圧縮を行っている。

図面の圧縮は、X 軸方向と Y 軸方向独立に行われる。まず始めに Y 軸方向 (縦方向) について圧縮を行った後、X 軸方向 (横方向) に圧縮を行っている (図 6.2)。これは、経験的に Y 軸方向に図面が大きくなる傾向にあるためである。

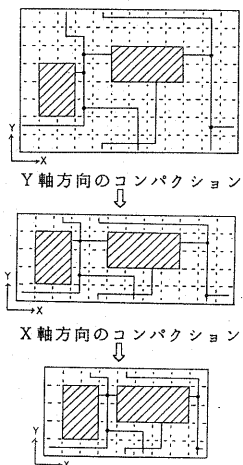


図 6.2 図面圧縮

7. 適用例

本システムは既に実際の L S I 開発への適用を開始しており、人手設計にほぼ匹敵する最適化と設計品質の向上が図られている。人手による設計と本システムによる設計の結果を比較できる機会というのは、本システムの性格上ほとんどないが、既に人手により CMOS-L S I 回路に変換されたことのある標準 IC/PLD 回路に対して、改めて本システムを適用してみた。結果を表 1 に示す。人手に比較し回路規模で 1.05 倍とほぼ互角の性能を發揮しているのが分かる。

次に、処理時間について測定を行った結果を表 2 に示す。なお、実行マシンの速度は 1 Mips である。

最後に、変換例として標準 IC/PLD 回路図と、それに対して本システムを適用し生成された CMOS-L S I 回路図を図 7.1 に示す。原回路図における素子位置、配線経路、束線の各情報が L S I 回路図に反映され、対応付けが容易となっているのが分かる。例えば、①は標準 IC : 74LS393 とそれに対する L S I 回路部分を示している。また、②は PLD に対して動作記述をもとに L S I 回路を生成している例である。

回路 X	本システムを用いた場合	人手設計
L S I 回路のゲート数 (NAND2 換算)	588	561
人手設計との比	1.05	1

表 1 人手設計との比較

8. おわりに

以上述べたように、標準 IC/PLD 回路図を基に CMOS-L S I テクノロジーに基づく回路図を生成するシステム T L / C を開発した。本システムは、標準 IC に加え、動作記述により動作が規定された PLD を含む回路図を基に、素子位置及び配線経路類似でしかも冗長部分が除去された CMOS-L S I 回路図を生成する。

本システムは既に実際の L S I 開発への適用を開始しており、大幅な設計工数の削減と設計品質の向上が計られている。

(参考文献)

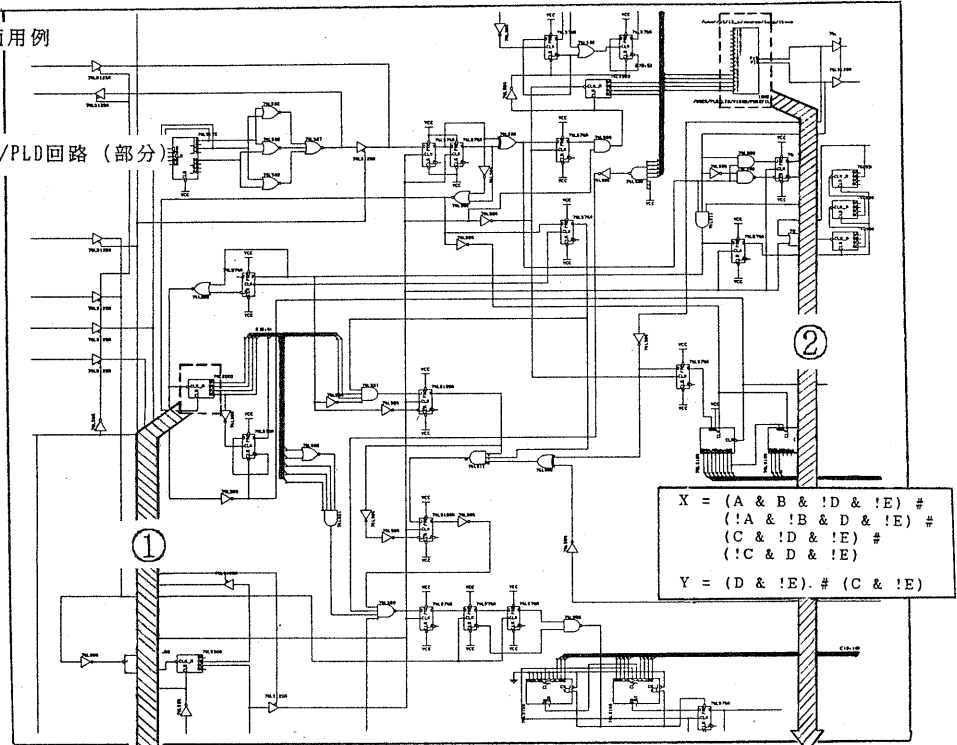
- [1] 井川, 他: 汎用 IC/PLD 回路から L S I 回路を生成するシステム T L / C (1) —— 概要及び回路図生成 —— ; 情報処理学会第 35 回全国大会講演論文集, PP.2239-2240, (1987)
- [2] 岡崎, 他: 汎用 IC/PLD 回路から L S I 回路を生成するシステム T L / C (2) —— 最適化処理 —— ; 同上, PP.2241-2242, (1987)
- [3] 喜納, 他: エキスパートシステム開発支援ツール A I D —— 基本思想とその応用 —— ; 情報処理学会第 34 回全国大会, PP.1575-1576 (1987)
- [4] 渡辺, 他: 知識処理による回路系自動変換システムの開発; 情報処理学会第 33 回全国大会講演論文集, PP.2225-2226, (1986)
- [5] 平峰, 他: 論理回路変換プログラム: LORBS/EX ; 情報処理学会設計自動化研究会 38-2 (1987.7.23)
- [6] 平林, 他: 流用設計支援エキスパートシステム: TRIP; 情報処理学会設計自動化研究会 39-2 (1987.10.15)
- [7] C.Y.Lee: An Algorithm for Path Connect and its Applications; IRE Trans. vol. EC-10, no.3, PP.346-365.

回路	L S I ゲート数 (NAND2 換算)	処理時間 (sec)
A	1032	1490
B	1074	1475
C	770	664
D	368	351

表 2 処理時間 (CPU 時間)

图 7.1 適用例

(a) 標準IC/PLD回路 (部分)



(b) LSI 回路 (部分)

