4N−7　回路系自動変換のための高速ネット変換機能のPrologによる実現

杉田尚一†　渡辺俊典‡　林晋一‡　大沢秀行‡‡　太田之裕‡‡
†(株)日立京葉エンジニアリング　‡(株)日立製作所システム開発研究所
‡‡(株)日立製作所半導体設計開発センタ

1.　はじめに

異なる回路系(CMOS,TTL等)間の回路自動変換(テクノロジーマッピング)問題は、論理設計自動化の基本技術の一つである。しかし一般の論理生成技術では、取り扱われていない複雑な処理(回路内素子の相互接続関係の自動認識等)を必要とする。我々はCMOSからIIL論理を生成する問題を例として、本問題に対するルールベース形式のシステムを開発してきた。[1)2)3)] 今回はシステムの基本部であるルールによる回路接続認識と書き換え処理の方式について報告する。

2.　高速化アルゴリズム

素子数Nの回路において、r個の素子の接続条件を単純なルールベースによって、チェックする手間は$O(N^r)$となる。認識範囲を素子の近傍に限定することによって、高速化を実現した。

2．1　データ表現

(1)ゲートデータ

```
gate(Name,Type,Device,I,O,D,Shape,Prop).
```

ネット変換時に必要なノード情報(I,O,D)等と図面描画時に必要な座標情報(Shape,Prop)等を、gate(...)中にコンパクトに納めた。

(2)ノードデータ

```
node(Node,Gatelist).
```

初期処理で、ゲートの接続状況を調べ、ゲートデータの双対としてノードデータを生成する。(2.3.①)

ゲートとノードの双対データを用いて、近傍検索を高速化する。(2.3.④)

2．2　変換ルール表現

```
rule(type(From,To),if(IFparts),then(THENparts)).
```

IF-THEN形式で記述され、IFpartsにはゲートやゲートの接続状況判定条件が並べられている。THENpartsにはIFpartsの条件が満たされたときの変換結果を示すゲート表現や、変換に伴うノード情報の修正手続きが並べられている。すなわち、IFpartsとマッチングする回路パターンが認識できた場合、THENpartsに記述された回路パターンに書き換える。

2．3　回路ネット変換推論

原回路ネットデータを入力し、ネット変換推論則と変換ルールを用いて、ネットを逐次書き換え、目標回路ネットデータを出力する。

(1)変換ステップ

変換は以下の順番で実施する。

(Ⅰ)前処理（入力データのエラーチェック等）
(Ⅱ)ピボット系への変換（nor,or系を使用）
(Ⅲ)目標回路生成（ファンアウト整合等含む）
(Ⅳ)最適化（不要ゲート削除）

(2)推論方式

主推論（変換過程リスト，原データファイル，ノードファイル，出力データファイルを使用）
(ⅰ)ノードデータ生成（原データファイル入力，ノードファイル出力）%①
(ⅱ)ゲートデータ読み込み(原データファイル入力)
(ⅲ)ノードデータ読み込み(ノードファイル入力)
(ⅳ)全変換ステップを実行（変換過程リスト）
(ⅴ)データ出力（出力データファイルへ出力）

全変換ステップを実行（［第nステップ変換｜残ステップ］入力）%②
(ⅰ)特定変換ステップを実行（第nステップ変換を実施）
(ⅱ)全変換ステップを実行（残ステップ）%③

特定変換ステップを実行（第nステップ変換を実施）
(ⅰ)注目ゲートデータの選定%④
(ⅱ)近傍探索の実施（注目ゲートデータ入力，近傍ゲートデータ出力）%⑤
(ⅲ)変換ルール適用（第nステップ変換入力，近傍ゲートデータ入力，注目ゲートデータ入力）%⑥
(ⅳ)残ゲートデータ変換%⑦

近傍探索の実施（注目ゲートデータ入力，近傍ゲートデータ出力）
(ⅰ)注目ゲートの入力側を出力ノードとするゲートを探索
(ⅱ)注目ゲートの出力側を入力ノードとするゲートを検索

Implementing high speed local net-translator in Prolog

S.Sugita†, W.Watanabe‡, S.Hayashi‡, H.Oosawa‡‡, Y.Oota‡‡
†Hitachi Keiyo Engineering,Ltd.
‡Systems Development Laboratory,Hitachi,Ltd.
‡‡Semiconductor Design and Development Center,Hitachi,Ltd.

(iii)(i), (ii)で探索したデータをリスト化（近傍ゲートデータ）

変換ルール適用（第ｎステップ変換入力，近傍ゲートデータ入力，注目ゲートデータ入力）

(ｉ)変換ルールの呼び出し（第ｎステップ変換入力，ＩＦ部取得，ＴＨＥＮ部取得）

(ii)ＩＦ部適用（ＩＦ部入力，近傍ゲートデータ入力，削除ゲートデータ出力）％⑧

(iii)削除（アサートデータ内の削除ゲートデータを削除）

(iv)ＴＨＥＮ部適用（ＴＨＥＮ部入力，変換後ゲートデータ出力）％⑨

(ｖ)ネイミング（変換後ゲートデータ入力，ネイミング済ゲートデータ出力）

(vi)追加（ネイミング済ゲートデータをアサート）

(vii)ノードデータ保守（削除ゲートデータ入力，追加ゲートデータ入力）％⑩

①ゲートデータからゲート間のノード情報を作成する。
②Steplist内に指定された変換ステップの一つ(例えば2.3(1)(i))を実行する。
④アサートされている順番に一個ずつgate(...)データを呼び出し、そのデータを注目ゲートとする。
⑤注目ゲートに隣接したゲートをノード情報を用いて高速に検索し、そのデータの集まりをNandmeとする。
⑥apply_ruleでNandmeを各変換ルールに飛ばし、⑧ルールの条件をみたすデータがあれば、⑨ルールに従って変換し、⑩ノードデータのメンテナンスもする。このとき、変換ルールの適合を注目ゲートと隣接したゲートに限定することにより、パターン認識範囲を局所化し、高速化が実現される。
⑦全てのゲートを次々に注目ゲートにして、局所的に回路の書き換えをすることにより、回路全体の変換をおこなう。
③一つの変換ステップを終了したのち、次のステップを同様にして実行する。

3．性能評価

変換結果の一例を図１に示す。

入力ゲート数に対するネット変換処理時間は$O(N^{1.4})$となった。（図２）

4．　まとめ

変換ルールを用いた回路変換システムにおいて、回路パターン認識を局所化することにより、$O(N^{1.4})$のネット変換速度が実現できた。

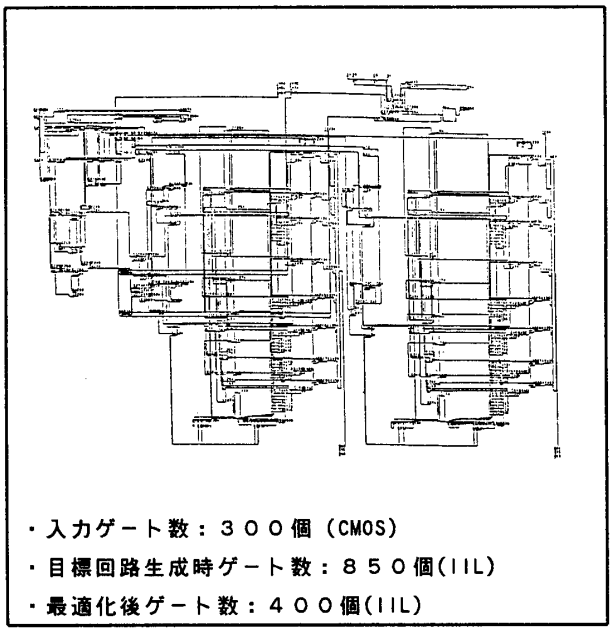

図１．　変換結果（最適化後）

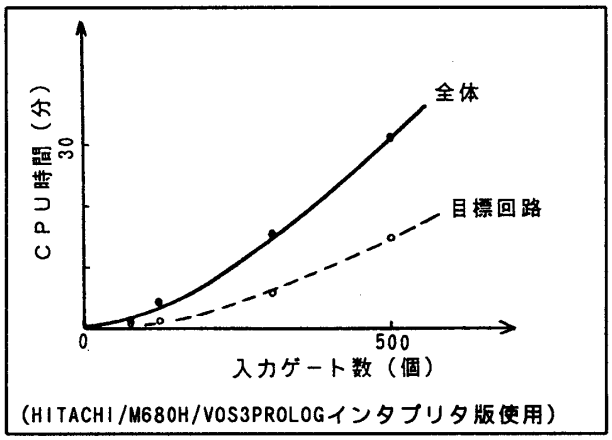

図２．　ゲート数－時間特性

参考文献

１）渡辺他：知識処理による回路系自動変換システムの開発，情報処理第３３回全国大会，講演論文集(Ⅱ)，pp2225-2226(1986)

２）杉田他：回路系自動変換システムにおけるループ除去方式の一検討，情報処理第４０回全国大会,講演論文集(Ⅱ),pp1365-1366(1990)

３）T.Watanabe.,et al.："Knowledge-Based Optimal IIL Circuit Generator From Conventional Logic Circuit Descriptions",IEEE Computer Society,23rd Design Automation Conference,Vol34,No3,pp608-614,1986