

TM-0516他

89-1 電子情報通信学会
秋期全国大会

September, 1988

©1988, ICOT

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

- TM 0516 制約条件を考慮したデータパス系回路の再設計機構
松永祐介, 丸山文宏, 角田多苗子, 箕田依子, 川戸信明(富士通)
- TM 0517 協調問題解決システム実現の為に並列オブジェクト指向言語
POOL 市来宏基, 佐藤秀樹, 末永富美代(富士通)
- TM 0518 基準ポートフォリオ選択のための協調問題解決
末永富美代, 佐藤秀樹, 市来宏基(富士通)
- TM 0519 運用ポートフォリオ選択のための協調問題解決
佐藤秀樹, 市来宏基, 末永富美代(富士通)

制約条件を考慮したデータパス系回路の再設計機構

Redesign Mechanism for Datapath Circuit

松永 裕介

Yusuke MATSUNAGA

丸山 文宏

Fumihito MARIYAMA

角田 多苗子

Taeko KATUDA

笑田 依子

Yoriko MINODA

川戸 信明

Nobuaki KAWATO

富士通株式会社

FUJITSU LTD.

1. はじめに

ハードウェアの機能レベルの仕様記述をもとに論理設計を行うエキスパート・システムCO-LOGEXを開発中である。ここで報告するデータパス系回路の設計機構はそのサブシステムであり、考えられた制約条件(回路のゲート数、遅延時間)を満たすように回路の設計を行うものである。

加算器、カウンタといった回路要素(コンポーネントと呼ぶ)には、通常、異なるゲート数、遅延時間等のパラメータを持つ複数の構成法(オルタナティブと呼ぶ)がある。その中で最適なオルタナティブは一意的には決まらず、全体の回路の制約条件、および他のコンポーネントとの関係から決定される。このため、適当なオルタナティブを選択して、制約条件を満たしているかどうかを評価し、条件に違反している場合には他のオルタナティブを選択するといった設計のやりなおし、繰り返しが必要となる。そこで、この設計過程を効率的に行うための再設計機構を提案する。

2. システム構成

図1に構成を示す。設計モジュールは制約条件を考えずに、個々のオルタナティブを生成する。評価・再設計モジュールは設計されたオルタナティブを受け取り、適宜、選択を繰り返すことによって、全体として制約条件を満たすように回路を選択してゆく。その際、制約条件に違反した原因(これをN.J.: No good Justificationと呼ぶ)を保持することによって無駄な繰り返しを避けている。[1]

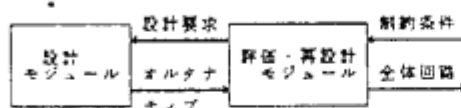


図1: データパス系回路設計サブシステムの構成

3. データ構造

設計データを表すために、コンポーネント・ノード、及び、オルタナティブ・ノードと呼ばれる2つの概念を用いる。コンポーネント・ノードは、現在、選択されているオルタナティブ(1Nオルタナティブ)と、選択されていないオルタナティブ(OUTオルタナティブ)を子として持つ。オルタナティブ・ノードは、それ自身を構成する部分コンポーネント(もしあれば)を子として持つ。この2種類の概念を用いて階層的に設計を行うことができる。

4. N.J. (No good Justification)

コンポーネントA, B, Cからなる回路の全体のゲート数制約が500で、コンポーネントAのオルタナティブa1のゲート数が150の時、a1を禁止するN.J.は、

$$150 + B \text{ のゲート数} + C \text{ のゲート数} > 500$$

となる。これは、同じ条件で(もしくは、さらに悪い条件で)このオルタナティブが再び選択される事を禁止するために用いられる。N.J.は、N.J.の展開、N.J.の合成という2つの操作によって生成される。N.J.の展開とは、設計変更のレベルを詳細化して行く時に、N.J.をそれに合わせて詳細化するものであり、N.J.の合成とは、逆にあるレベルでの設計変更に失敗し、1つ上のレベルに移る時に、詳細化されたN.J.を一般化するものである。

5. アルゴリズム

- 1: 全体の回路(階層構造の根のノード)の1NオルタナティブをA.L.T.としてステップ2へ。
- 2: A.L.T.および、その先祖のオルタナティブに書かれているN.J.をチェックする。真になるN.J.があれば、ステップ4へ、なければステップ3へ。
- 3: 全てのオルタナティブがOUTになっているコンポーネントを探す。もしなければ終わる(設計成功)、あればそのコンポーネントのオルタナティブを1つ選んで1Nにし、そのオルタナティブをA.L.T.としてステップ2へ。
- 4: N.J.の中から変更すべきコンポーネントを選び、そのコンポーネントの1NオルタナティブをA.L.T.とする。A.L.T.に対して、N.J.の展開を行い、ステップ5へ。
- 5: 展開、または合成されたN.J.中にA.L.T.の子孫が現れていなければ(つまりA.L.T.自身では解決できない)、ステップ6へ、そうでないならステップ4へ。
- 6: A.L.T.をOUTにして他のオルタナティブのどれか1つを1Nにする(もしくは新しいオルタナティブを設計モジュールに要求する)。そのオルタナティブをA.L.T.として、ステップ2へ。全てのオルタナティブがN.J.によって禁止されているならN.J.の合成を行い、親のオルタナティブをA.L.T.としてステップ5へ。親がなければ終わる(設計失敗)。

6. おわりに

上記のアルゴリズムの性能は、ステップ4のコンポーネントの選択の仕方に大きく影響される。現在、幾つかの方法を実装し、評価を行っている。

謝辞 本研究を進めるにあたって御支援頂いたICOT第五研究奨励基金に深く感謝します。

参考文献:

- [1] 協賛論理設計エキスパートシステムにおける仮説推論に基づく再設計機構、角田 他、第63A I 学会全国大会