

協調型論理設計エキスパートシステム co-LODEX

—概要—

箕田依子 澤田秀穂 滝沢ユカ 丸山文宏 川戸信明
富士通株式会社

1 はじめに

我々は、ハードウェアの機能レベルの仕様から、回路規模と遅延時間に関する制約条件を満たす回路を生成する論理設計支援システムを開発している。与えられた制約条件に対して設計結果を評価し、違反が検出された場合、制約条件を満たすよう設計のやり直し（再設計）を行なう方式の研究において、論理式で表現する制約条件違反情報（NJ: Nogo od Justification）を考案し、これを利用して再設計を実行するシステムを試作した。[1]

本稿では、この方式を並列処理システム上に拡張するための、並列協調方式を提案する。設計の流れを述べた後、並列協調方式について説明する。

2 システム概要

co-LODEXにおける設計の流れを図1に示す。エージェントと呼ぶ自律的に動作する実行単位を設けて、部分回路を並列に、全体に対する制約条件を満たすように詳細化することが特徴である。

入力は、機能レベルの動作仕様と、レジスタ、演算器などの機能ブロックと機能ブロック間のデータの流れを示すブロック図、生成した回路が満たすべき制約条件（ゲート数と遅延時間）の3種類である。動作仕様は、状態遷移とレジスタ転送などを表す仕様記述言語で記述したものであり、これだけでも仕様を入力することができるが、設計者が機能ブロックをイメージしている場合は、ブロック図を入力することにより、設計者の意図を反映させることができる。

動作仕様から、レジスタ転送とターミナル結合に関するオペレーション（データバスオペレーション）とステート遷移に関するオペレーション（制御オペレーション）を抽出し、これらのすべてがブロック図で表されたデータバス上で実行可能なことをチェックする。この過程において、機能ブロックとその接続関係、および制御回路の仕様を決定する。

制約条件は、デフォルトNJと呼ぶ、制約条件と等価な不等式（不等式が成立することが制約条件違反の必要十分条件になる）

に書き換える。設計の実行中に生成されるNJは、デフォルトNJを起源とし、その成立が制約条件違反の十分条件になる論理式である。NJは、属性値を示す変数と属性値、制約値を示す変数を用いて表す。

機能ブロック（制御回路も制御ブロックと呼ぶ機能ブロックのひとつ）とその接続情報に書き換えられた仕様を、部分仕様に分割する。分割は、並列協調アルゴリズムの特性を考慮して、「クリティカルパスの候補をなるべく少ない数のエージェントが担当するように分割する」という方針で行なう。回路分割に従って、デフォルトNJもエージェントの属性値を示す変数を使用したものに書き換える。

部分仕様と、デフォルトNJを各エージェントに通知する。エージェントは協調して、部分回路仕様の詳細化と評価を繰り返すことにより、制約条件を満たす回路を設計する。出力は、CMOSスタンダードセルの接続情報である。もし、制約条件を満たす回路を生成することができなかった場合は、違反した制約条件を提示し、ユーザが制約条件を変更することによって、再設計を開始することができる。制約条件の変更による再設計においては、蓄積されたNJを再利用する。

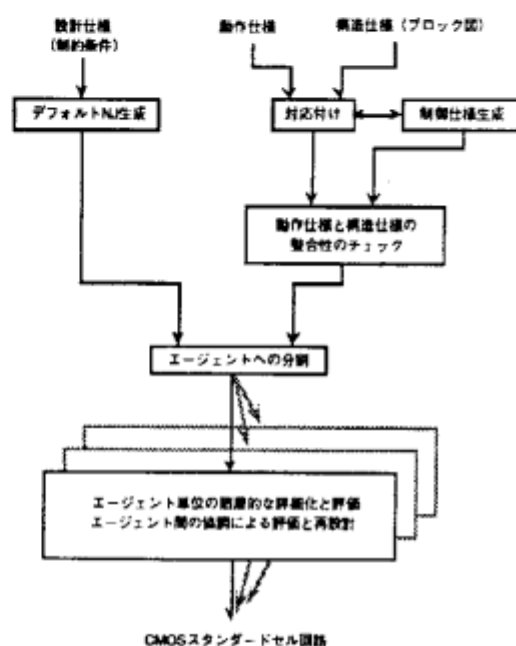


図1 設計の流れ

Overview of co-operative logic design expert system (co-LODEX)

Minoda Yoriko, Sawada Shuho, Takizawa Yuka, Maruyama Fumihiko, Kawato Nobuaki

FUJITSU LIMITED

3 並列協調方式

集中管理部を持たない、本並列協調方式の特徴は、以下の2点である。

- (1) 部分仕様を独立に（並列に）詳細化する。
- (2) 各エージェントは、全体の制約条件を満たすために、自律的に軌道修正する。

3.1 並列協調アルゴリズム

図2に沿って説明する。並列協調は、部分仕様とデフォルトNJが通知された時点で開始する。以下の処理は、各エージェントが独立に（並列に）行なう。

[1] 蓄積しているNJ（初めはデフォルトNJだけ）を成り立たせないように、部分回路を設計する。他のエージェントが担当している部分は0として扱う。

[2] 少なくともひとつのエージェントが部分回路の設計に失敗すると、与えられた制約を満たす回路は存在しない。すべてのエージェントが成功したことを確認する。

[3] 各エージェントの設計結果（部分回路の属性値）を通知し合う。

[4] ひとつでも前回と異なる結果があるかを調べる。

[5] 通知された設計結果を代入しても、デフォルトNJが成立しなければ成功である。

[6] 蓄積しているNJを成り立たせないように、部分回路を設計する。他のエージェントが担当している部分は[3]で通知された値を使用する。

[7] 少なくとも、ひとつのエージェントがNJを成立させない結果を得られたならば、成功した結果で置き換えて、[3]へ。この処理により、違反が生じている制約の数は減少する。失敗したエージェントには、エージェントが担当する部分回路に相当する変数を含まないNJが生成されている。

[8] 各エージェントが生成したNJを関係するエージェントに通知する。通知されたNJは、「NJの組み合わせ」を行なうことにより、新しいNJとして蓄積される。

[9] 違反している制約のうち、関係するエージェントが最少のものをひとつ選び、この制約を厳しくした一時的制約を与える。

[10] 部分回路を結合する。[11] 制約条件を緩和することにより、蓄積されたNJのいくつかを無視できるようになる。制約条件の緩和により、再設計を開始できる。

3.2 NJの組み合わせ

NJの組み合わせは、他のエージェントの設計不可能範囲から、自分自身の設計範囲を制限するNJを生成する処理である。例を用いて説明する。3つのエージェントに関係する制約があり、すべてのエージェントで再設計不可能であるとす。このとき、次のNJが生成されている。A1、A2、A3はそれぞれエージェントの属性値に対応する変数である。

エージェント1： $5 + A2 + A3 > C$ ， エージェント2： $A1 + 10 + A3 > C$ ， エージェント3： $A1 + A2 + 8 > C$

これらを互いに通知し合う。エージェント1においては、 $A1 + 18 > C$ という新しいNJが生成される。このNJにより、A1を除く部分に最低でも18必要なことがわかり、エージェントの条件が厳しくなる。部分回路の設計1においては、他のエージェントの担当部分を0として扱っているが、このときでも他のエージェントの設計不可能範囲を考慮することができるようになる。

4 おわりに

エージェントをプロセッサに割り当てたシミュレーションによる動作確認を終え、いくつかの例題を実行することにより、負荷分散方式の検討などを行なっている。

本研究は第5世代コンピュータプロジェクトの一環として行なわれているものであり、ご支援頂いたICOT 生駒研究部長代理、新田第七研究室長に感謝します。

〔参考文献〕 [1] 丸山他 「評価・再設計機構を備えた論理設計支援システム」 信学会論文誌 1989.8

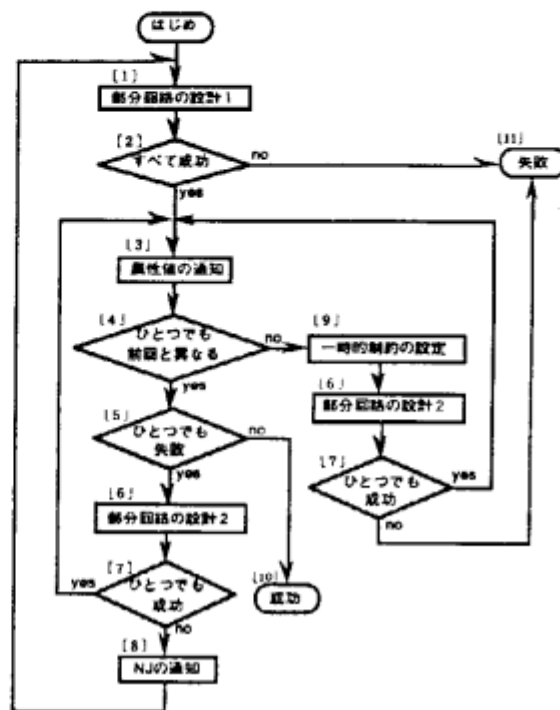


図2 並列協調アルゴリズム