

高位合成システムの試作

4E-5

高橋 奈穂美

開発 貴久  
(株) 東芝

黒澤 雄一

1 はじめに

LSIの機能設計過程からの支援を目的として、ハードウェアによる実現方法をあまり意識しない動作仕様記述を入力し、これを実現するのに必要なレジスタ、演算器等の構成要素とRTレベルの動作仕様を生成する高位合成システムを試作した。

機能設計の途中段階で、設計者が意図する時間制約と面積制約に合うような設計が行なえない時、どちらかの条件をわずかに緩和することによって意図する設計に近いものが得られることが多い。

そこで、本システムは所望の時間・面積制約にあう設計が得られない場合等において、設計者とのインタラクションを行なうことによって、設計者の要求に近い設計を得ることを可能とした。

2 システムの概要

2.1 特徴

設計者が設計の途中段階で介入していくつかの設計方式を試行錯誤し、時間と資源量に関して設計者の設計目標に合う結果を得ることを可能とするために、本システムは以下の特徴を持つものとした。

- 3種の介入が可能である。
  1. パラメータとしてサイクルタイム、使用する演算器数があり、これらを変更して、複数の設計案を作成できる。演算器数は各レベルで並列実行可能な演算数の最大値を与えるためこれを演算パラメータと呼ぶ。
  2. 設計の途中段階がパラメータに合わない(パラメータ違反)時、設計を続行するためのパラメータ変更、その他の設計選択を指示できる。
  3. スケジューリング、資源割り当ての途中段階でスケジュール変更、共有化を指示できる。
- 設計者が上の介入を行なうための指針となる評価情報を提示する。
  1. 無駄時間(各レベルで演算器資源が動作していない時間)、仕様全体の実行時間(while, forなどのループの繰り返しを考慮したもの)、資源量を提示する。
  2. 演算パラメータに合わせた設計の実行時間の増加量、実行時間を増加させない時の資源の増加量を提示する。

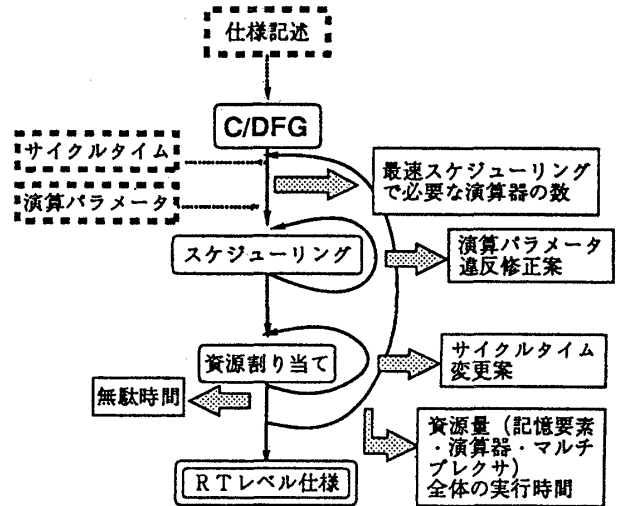


図 1: 高位合成のシステムの流れ

3. 各レベル毎に同時に実行される演算の数・種類(演算の並列度)や仕様全体の実行時間を増加させずにスケジューリング変更を行なえる最大レベル数(余裕度)などの評価情報を提示する。

- 各合成段階の設計情報(入力記述、コントロール・データフローグラフ(以下C/DFG)、設計結果であるRTレベル記述)間の対応表示機能がある為、仕様記述がどのようにスケジューリングされ、どのような資源に反映されているかを容易に知ることができるので、上記のような設計者の介入が簡単にできる。

2.2 処理フロー

システムの流れを図1に示す。

まず仕様記述をシステムの内部表現であるC/DFGに変換する。

次に設計者に入力されたサイクルタイムに基づき、C/DFG上に現れた代入、四則演算等の各演算ノードをどのレベルに実行するかを決定する(スケジューリング)。

そしてC/DFG上の四則、比較演算等の演算ノードに相当する演算を実行する演算器や、変数や演算の途中結果を保持する記憶要素を決定する(資源割り当て)。

最後に、以上の処理の結果得られたレベル毎の動作仕様をRTレベルの記述として出力する。

3 スケジューリング

最速スケジューリング(以下ASAP)を初期スケジューリングとし、演算パラメータに合うまで、余裕度の

Prototyping of High Level Synthesis System  
Naomi TAKAHASHI, Takahisa KAIHOTSU  
and Yuichi KUROSAWA.  
Toshiba Corporation.

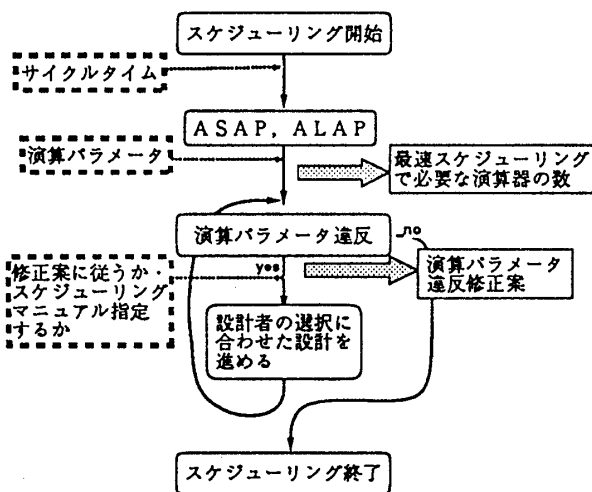


図 2: スケジューリング

大きい演算ノードからスケジューリングを遅らせていく。スケジューリングの流れを図 2 に示す。

### 3.1 設計者とのインタラクション

まず、ASAP に必要な演算器数を設計者に提示する。これが、設計者の資源量の制約より大きい場合は設計者は適当な演算パラメータを入力する。

次に入力された演算パラメータの違反が無くなるように、システムが演算ノードのレベルを遅らせていく時、全体の実行時間の増加量を設計者に提示する。この情報から、例えば演算パラメータの値を 1 つだけ増やすことにより全体の実行時間を大幅に少なくできるということがわかる場合がある。この時、演算パラメータの値を増やしてスケジューリングを再実行することができる。

また、演算パラメータを入力せずにシステムで途中でスケジューリングを行ない、各レベルの演算の並列度、余裕度等の現在のスケジュール状況を参照しながらマニュアルでスケジューリング変更するという使い方もできる。この時、システムはその変更を行なった時の全体の実行時間の増加などを表示し、その変更の可否を設計者に確認してから変更を行なう。

このように、資源と時間のトレードオフをとりながら設計案を作成することができる。

## 4 資源割り当て

本システムでは、演算器共有化によるマルチプレクサの増加量の少ないものから優先的に共有化していく。

### 4.1 設計者とのインタラクション

資源割り当て後には設計結果として、各レベルで演算器資源が動作していない無駄時間、全体の実行時間、資源量などを表示し、設計者の意図に合わない場合、サイクルタイムや演算パラメータを変更して資源割り当てや、スケジューリングをやり直すことができる。

資源割り当ての途中段階で、共有化によって挿入されるマルチプレクサによる遅延により、サイクルタイム内に処理が収まらない場合には、その共有化を行なうた

	(a)	(b)	(c)
乗算器数	8	3	2
加算器数	6	6	5
総資源量 (ゲート)	23834	10607	7412
総時間 (nsec)	550	561	572

(a) サイクルタイムを変更しない場合

(b) サイクルタイムを 51 nsec まで延長した場合

(c) サイクルタイムを 52 nsec まで延長した場合

図 3: 本システムの適用例

めに必要なサイクルタイムの変更時間と、それによる仕様全体の実行時間の増加量を提示し、その共有化の実行の可否を設計者に選択させる。5 にこの例を示す。

また、資源割り当ての途中で、割り当てられた資源の一覧表、共有可能な資源の一覧を見て、資源共有化のマニュアル指示を行なうことができる。

## 5 実行例

図 3 に 16bit データのフィルターの仕様記述を入力とし、演算パラメータ乗算器 2 つ、加算器 5 つ、サイクルタイム 50nsec で本システムを適用した設計例を示す。設計途中で、資源共有化で挿入されるマルチプレクサのレイを考慮すると演算処理がサイクルタイム内におさまらない。システムはサイクルタイムを 51ns まで延ばせば共有化できる資源と、52ns まで延ばせば共有化できる資源があることをレポートする。

(a) の例は、サイクルタイムを変更しない場合、(b)、(c) はそれぞれサイクルタイムの延長を 51ns、52ns まで許して資源共有化を行なった場合である。資源量は、演算器、記憶要素、マルチプレクサ等のを全て計算している。

このように、設計者はサイクルタイムを厳守する設計、演算パラメータを厳守する設計だけでなく、パラメータ違反の起こっているタイプ、どの程度パラメータに合わないか等のシステムが提示する情報を参考にしながら、時間・資源量のトレードオフをとるような設計案を得ることができる。

## 6 おわりに

システムが提示したレベル毎の演算の並列度や余裕度などの情報から、設計者がサイクルタイム、使用するタイプ毎の演算の並列度制約、あるいはスケジューリング、資源共有化などを指定して設計者の意図に近い設計案を作成することができることを目指した高位合成システムを試作した。

今後は本システムを設計者に提示し、どのようなシステムが望まれているかを検討していきたい。