C-017

資源制約を考慮したマルチプロセッサシステムの自動生成
Automatic Generation of Multiprocessor Systems Considered Resource Constraint

松村 将嘉† 中野 秀洋‡ 宮内 新‡
Masayoshi Matsumura Hidehiro Nakano Arata Miyauchi

1. はじめに

近年 LSI 設計において要求されている設計期間の短縮を実現するための手法として、高級言語からのプロセッサ自動生成システムが提案されている[1]。また、従来のシングルプロセッサでの命令レベル並列による性能向上は限界が見え始めており近年はマルチプロセッサを用いたスレッドレベル並列による性能向上が一般的になっている[2]。これに対して、目的のアプリケーションプログラムからスレッドレベル並列性を抽出し、それを効率的に利用できるプロセッサの個数を導出する自動生成手法が提案されている[3]。

本稿では、回路規模と実行時間を制約条件とし、さらにプロセッサの個数に加え実装する演算器の種類を可変要素としたマルチプロセッサの自動生成手法を提案する。

2. 従来手法

従来手法[3]は、並列化コンパイラと構成決定アプリケーションから成る。はじめに、高級言語で記述されたアプリケーションプログラムを並列化コンパイラにより並列化する。並列化コンパイラではループ分割による並列化が行われ、スレッドごとのアセンブラコードが出力される。続いてアセンブラコードは構成決定アプリケーションに入力される。構造決定アプリケーションでは、シミュレーションによりプロセッサ数を増加させた際の性能効率を求め、与えられた制約条件を満たすプロセッサの個数を導出する。そして、HDL で記述された単体のプロセッサを組み合わせ、導出された個数のプロセッサを持つマルチプロセッサを生成する。

3. 提案手法

本提案手法では、前節で示した従来手法の「構成決定アプリケーション」に対して以下の変更を行う。並列化コンパイラについては従来手法と同様とする。

3.1 制約条件の変更

従来手法では、プロセッサ構成決定のための制約条件として性能効率のみを用いていた。しかし、実際に LSI を設計する際は回路規模に制限がある。また、一般的にハードウェアコストと性能はトレードオフの関係にある。そのため、本手法では回路規模と実行時間を制約条件として用いる。実行時間Tは式(1)により定義される。

$$T = CC \times DELAY \quad (1)$$

式(1)において、CCはプログラムの実行に要するクロックサイクル数、DELAYは回路の最大遅延時間(動作クロック周波数の逆数)である。

† 東京都市大学大学院 工学研究科
Graduate School of Engineering, Tokyo City University
‡ 東京都市大学 知識工学部
Faculty of Knowledge Engineering, Tokyo City University

3.2 演算器の可変要素化

従来手法ではプロセッサの個数のみが可変要素であった。目的のアプリケーションプログラムに応じて実装する演算器の種類を選択することで、より効率的なプロセッサの生成が可能となる。よって、本手法ではプロセッサに実装する演算器も可変要素とする。ベースプロセッサには整数加減算や論理演算といった基本的な演算を行う演算器のみを実装しておき、それ以外の演算器は必要に応じて実装する。実装されなかった演算器を用いる演算命令は、基本的な演算の組み合わせで代用する。

3.3 プロセッサ構成導出手順

プロセッサ数や使用可能な演算器の組み合わせ全てに対して論理合成やシミュレーションによる評価を行うには多大な時間を要する。そのため、本手法ではあらかじめいくつかのシミュレーションと論理合成により求めた情報を用い、理論的な数値の計算を行うことにより回路規模および実行時間を見積もる。プロセッサ構成導出のフローチャートを図 1 に示す。

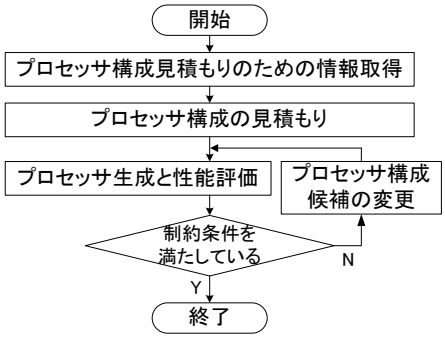

図 1: プロセッサ構成導出のフローチャート

3.3.1 プロセッサ構成見積もりのための情報取得

この手順では、プロセッサ構成の見積もりに必要な情報を取得する。表 1 に、項目と取得方法を示す。

表 1: プロセッサ構成見積もりのために必要な情報とその取得方法

項目	取得方法
基本命令以外の使用命令	アセンブリコードの解析
各構成要素の回路規模と遅延時間	論理合成
プロセッサ数の変化による実行クロックサイクル数	HDL シミュレーション
個々の追加実装演算器使用による実行クロックサイクル数	HDL シミュレーション

3.3.2 制約条件を満たすプロセッサ構成の見積もり

前項の手順により取得した情報を元に、数値の計算により制約条件を満たすプロセッサ構成の見積もりを行う。プロセッサの個数の最大値は、その回路規模が制約条件を超えない範囲とする。また、使用可能な演算器の組み合わせ全てについて見積もりを行う。

・ 回路規模の見積もり

プロセッサの回路規模は各構成要素(ベースプロセッサ・バス調停回路・追加実装演算器)の回路規模の総和とする。

・実行時間の見積もり

構成Xのプロセッサにおけるプログラムの実行時間 T(X)は、式(2)により求める実行クロックサイクル数 CC(X)と、式(3)により求める回路の最大遅延時間DELAY(X)を用いて式(1)により求める。

$$\mathrm{CC(X)} = \mathrm{CC(N)} - \sum_{i \in I}\left\{P(i) \times \frac{R(i)}{N} + \left(1 - P(i)\right) \times R(i)\right\} \quad (2)$$

$$\mathrm{DELAY(X)} = \max(DELAY(BASE), DELAY(ARBITER), DELAY(I)) \quad (3)$$

ここで、Nはプロセッサの個数、CC(N)はベースプロセッサ N 個で実行した際のクロックサイクル数、Iは追加実装する演算器 iの集合、P(i)は個々の演算器iを用いる演算命令が並列化された部分で出現する割合、R(i)は個々の演算器iの使用により削減されたクロックサイクル数である。また、DELAY(I)は追加実装する演算器の最大の遅延時間である。

3.3.3 プロセッサ生成と性能評価

前項の手順により導出された構成の候補を一つ選び、実際に制約条件を満たしているかを評価する。回路規模は論理合成により求める。また、実行時間は論理合成により求まった回路の遅延時間と HDL シミュレーションによる実行クロックサイクル数を用いて求める。制約条件を満たしている場合はその構成を決定して終了する。満たしていない場合は他の候補を用い同様の手順を実行する。

4. 実験

画像フィルタ処理(8 近傍)を行うプログラムを対象アプリケーションプログラムとし、複数の制約条件を仮定して提案手法の検証実験を行った。全てのハードウェア構成要素は VerilogHDL を用いて記述した。回路規模評価の尺度には FPGA(Altera CycloneIII EP3C40F780C6)へプロセッサの回路を実装した際の使用 LE 数を用いた。ベースプロセッサは MIPS ベースの RISC プロセッサであり、乗算・減算・比較・論理演算を行うための演算器を備えている。表 2 に、本実験で使用可能な追加実装演算器を示す。

表 2: 使用可能な追加実装演算器

i	演算器の種類	対象演算
1	左バレルシフタ	論理左シフト
2	32CLK 逐次型乗算器	整数乗算
3	1CLK 配列型乗算器	
4	2CLK 配列型乗算器	
5	4CLK 配列型乗算器	

4.1 実験 1: 回路規模制約=8000[LEs], 時間制約=5[ms]

上記条件において提案手法により制約条件を満たすプロセッサ構成を見積もった結果、計 6 通りの構成が導出された。表 3 に、導出された構成のうち回路規模が最小であるものと実行時間が最小であるものを示す。表 3 に示した見積もり結果について、論理合成および HDL シミュレーションを行い求めた回路規模および実行時間を表 4 に示す。表 4 より、見積もられた双方の構成が実際に回路規模および実行時間の制約条件を満たしていることがわかる。

表 3: 実験 1 におけるプロセッサ構成見積もり結果

	プロセッサ数	追加実装演算器 i	回路規模 [LEs]	実行時間 [ms]
回路規模最小	1	1, 4	5044	4.33
実行時間最小	3	(none)	7957	2.21

表 4: 実験 1 におけるプロセッサの回路規模・実行時間

	回路規模［LEs］	実行時間［ms］
回路規模最小	5042	4.81
実行時間最小	7970	2.53

4.2 実験 2: 回路規模制約=24000[LEs],時間制約=1[ms]

上記条件において提案手法により制約条件を満たすプロセッサ構成を見積もった結果、表 5 に示す 1 通りの構成が導出された。表 5 に示した見積もり結果について、論理合成および HDL シミュレーションを行い求めた回路規模および実行時間を表 6 に示す。表 6 より、見積もられた構成が実際に回路規模および実行時間の制約条件を満たしていることがわかる。

表 5: 実験 2 におけるプロセッサ構成見積もり結果

プロセッサ数	追加実装演算器 i	回路規模 [LEs]	実行時間 [ms]
8	1	22483	0.94

表 6: 実験 2 におけるプロセッサの回路規模・実行時間

回路規模［LEs］	実行時間［ms］
22685	0.92

5. 結論

本稿では、目的のアプリケーションプログラムの時間制約とプロセッサの実装に要する回路規模制約を満たすマルチプロセッサシステムの構成を導出する手法を提案した。そして、検証実験により提案手法の有効性を示した。

今後の課題として、実行時間をより正確に見積もる必要がある。また、他のアプリケーションプログラムに対して本手法が汎用的に適用可能であるかを検証する必要もある。

参考文献

[1] 西沢 政則, 他, “C 言語からのフラクタル型マルチプロセッサ自動合成システム”, 情報処理学会研究報告. SLDM [システム LSI 設計技術], 2006(111) , (2006).

[2] 上村井 明夫, 小林 良太郎, 安藤 秀樹, 島田 俊夫, “単一チップ・マルチプロセッサ SKY における投機的スレッド実行の性能評価”, 電子情報通信学会技術研究報告 CPSY, コンピュータシステム, 104(592), (2005).

[3] 夏目 貴史, 中野 秀洋, 宮内 新, “自動生成に適したマルチプロセッサと並列化コンパイラの提案”, 電子情報通信学会 2007 年総合大会 講演論文集, D-6-3, (2007)