3W−04

アプリケーションプログラムを基にしたプロセッサアーキテクチャの自動生成

−演算の並列性からのプロセッサの設計−*

松田 和幸† 門脇 一馬† 古川 剛史† 栗崎 正和† 宮内 新† 石川 知雄†
武蔵工業大学‡

1 はじめに

近年のプロセッサ開発サイクルでは，汎用プロセッサの中で処理時間の短縮が必要とされる部分を，専用のハードウェアを追加実装して処理することによって全体を高速化することがプロセッサ開発分野で行われている．しかし，その後，汎用プロセッサ自身の高速化や，ソフトウェアアルゴリズムの最適化が行われると，専用ハードウェアによる高速化が不要になることがある．従って，このような高速化を目的とした専用ハードウェアの開発は早いサイクルで行わなければならない．

我々は動画像処理を実時間で処理することを目標に，画像処理を高速に行うための専用プロセッサアーキテクチャの提案を行ってきた．本研究では，画像処理等の特定の分野に限定せず，与えられたある特定のアプリケーション・アルゴリズム(以下，ターゲットアルゴリズム)を高速化する為に，高級言語で記述されたプログラムから，演算が最適に並列化されたマイクロプロセッサのHDL記述を生成することを目的とする．本稿では，アプリケーションから抽出された並列度情報により，プロセッサに実装する命令を決定する手法についての提案を行う．

2 生成されるプロセッサの構成

本研究で生成されるプロセッサは，VLIW型で構成し明示的な並列演算を行うことにする．はじめに，ターゲットアルゴリズム向けのプロセッサ(以下，アルゴリズム向けプロセッサ)を作成する際の基準となるプロセッサ(以下，ベースプロセッサ)を設計した．ベースプロセッサは，実行並列度を2または3に設定でき，最低限の演算命令・メモリアクセス命令・分岐命令のみを実装してあり，全てのアルゴリズムを実行することが可能である．そして，ターゲットアルゴリズムが最適に実行できるように命令を加えることにより，アルゴリズム向けプロセッサを設計する．この他にベースプロセッサは次のような可変要素を持ち，これらをターゲットアルゴリズムに応じて変化させる．

- ALUユニット内の演算器の種類と数
- レジスタの本数と幅
- メモリの容量
- アドレスバス・データバスの幅
- 命令フォーマット
- 並列度

*Automatic Generation of Processor Architecture Based on Application Program -Design of Processor from Degree of Parallel Operation-

†Kazuyuki Matsuda, Kazuma Kadowaki, Takeshi Furukawa, Masakazu Kurisaki, Arata Miyauchi, Tomo Ishikawa
‡Musashi Institute of Technology

3 システムの構成

提案するシステムの構成は，図1である．

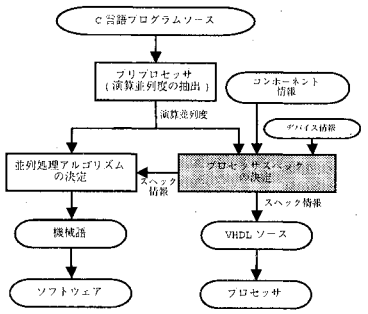

図1: システムの構成

1. プリプロセッサ：C言語プログラムソースから，プログラムの中で並列的に演算可能な箇所を抽出し，スケジューリングする．
2. プロセッサスペックの決定：演算の並列度やアプリケーションで使用する演算の種類から，コンポーネント情報・デバイス情報を考慮して，プロセッサのスペックを決定する．最後に，スペックに沿ったプロセッサのVHDLソースを出力する．
3. 並列処理アルゴリズムの決定：演算の並列度とプロセッサのスペック情報から，2.で生成されるプロセッサで動作する並列処理アルゴリズムを決定する．最後に，2.のプロセッサで動作するマシン語を出力する．

4 プロセッサスペックの決定

4.1 命令セットの定義

アルゴリズム向けプロセッサの命令セットは，基本命令，拡張命令，合成命令の3種がある．基本命令セットは，既存のRISCアーキテクチャ命令セットの中で多く使用される命令や，一般的な演算処理を行う上での最低限の命令セットを選定したものであり，前述のベースプロセッサに実装済みである．

拡張命令は，基本命令セットとは逆に既存のRISCアーキテクチャ命令セットの中であまり使用されない命令と，それに加えて特定のアルゴリズムに有用とされる命令を挙げておく．

事前に定義しておいた拡張命令の中に有用な命令が存在しない場合，複数の演算を1つの命令で実行する命令を自動的に作成する．これを命令の合成と呼ぶことにする．命令の合成は，水平型合成と垂直型合成がある．

水平型合成命令は，命令コードの空間的短縮を行う．水平型合成命令は，同一クロックサイクル中での並列演算の演算の組み合わせの頻度の高いものについて，1つのオペコードで複数の演算を並列に行うものである．これにより，命令コード領域を縮小することが可能である．

垂直型合成命令は，連続した演算の実行時間を短縮するものである．垂直型合成の定義は，連続して行われる演算の組み合わせの頻度が高いものについて，1つの命令発行で複数の演算を続けて行う命令を定義するものである．

合成命令は，コンポーネント情報を参照し，処理に必要なビット幅の演算器を追加実装したときのプロセッサの遅延を見積もり，ターゲットアルゴリズムの実行時間が短縮されるかどうか判断し，命令の実装を決定する．

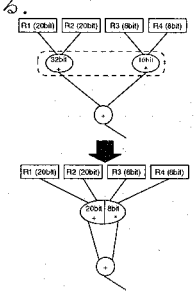

図 2: 水平型合成命令

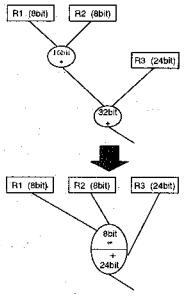

図 3: 垂直型合成命令

4.2 スペック決定

これまでに決定された実装する命令セット・演算器に基づいて，命令フォーマットを決める．また，レジスタ数に応じたオペランドサイズを用意し，1命令分のサイズを決める．そして，並列度に応じて命令を複数並べ，ユニット制御コードを付加して，命令フォーマットを決定する．

ここまでは，ハードウェアの制限無くスペックを決定を行ってきたが，デバイスのクロックサイクル，ゲート数の制限により実装が不可能と判断された場合は，追加された命令，レジスタ数の削減を行い再び見積もりを行う．このようにして，プロセッサの最終スペックを決定する．

4.3 アルゴリズム向けプロセッサの生成

アルゴリズム向けプロセッサは，ベースプロセッサを基に，決定したスペックに沿って命令・演算器のVHDL記述を追加する．拡張命令については，拡張命令毎にブロック記述を用意しておき，それをベースプロセッサに加える．合成命令については，各演算器コンポーネントを組み合わせて実装する．

5 アルゴリズム向けプロセッサの試作

本提案手法によるアルゴリズム向けプロセッサの性能について，評価実験を行った．バブルソート，2分探索，素朴法による文字列照合，画像フィルタ処理の各アルゴリズムに対して，ベースプロセッサによる並列処理と，合成命令を加えたアルゴリズム向けプロセッサによる実行時間の比較を行う．合成命令追加前後の実行ステップ数と，追加を行う合成命令を，表1に示す．論理合成[1]は，ターゲットデバイスとしてXilinx社のXCV2000E-FG680[2]を指定し，遅延時間を最適にする設定で合成した．

表 1: 合成命令による実行ステップ数の変化

アルゴリズム	並列度	基本命令のみ		合成命令追加		追加した合成命令	演算器数
バブルソート	2	702	100%	615	88%	加算→乗算の垂直合成	2
	3	656	100%	594	91%		
2分探索	2	71	100%	57	80%	加算→乗算の垂直合成	1
						1bit左シフト→加算の垂直合成	1
文字列照合	2	212	100%	209	99%	加算と加算の水平合成	1
画像フィルタ処理	2	1,863,421	100%	1,323,642	71%	加算と乗算の水平合成	2
	3	1,215,136	100%	1,021,588	84%		

そして，この結果を基に合成命令を追加したプロセッサを作成し，合成した結果を比較対照のベースプロセッサの結果と共に表2に示す．

表 2: アルゴリズム向けプロセッサの合成結果

	並列度	CLB数		動作周波数	
ベースプロセッサ	2	2,143	100.0%	28.2MHz	100.0%
バブルソートプロセッサ		2,178	101.6%	26.3MHz	93.3%
2分探索プロセッサ		2,177	101.6%	26.4MHz	93.6%
文字列照合プロセッサ		4,398	205.2%	26.3MHz	93.3%
画像フィルタ処理プロセッサ		2,178	101.6%	26.3MHz	93.3%
ベースプロセッサ	3	4,692	100.0%	26.8MHz	100.0%
バブルソートプロセッサ		4,720	100.6%	26.3MHz	98.1%
画像フィルタ処理プロセッサ		4,720	100.6%	26.3MHz	98.1%

使用デバイス：Xilinx XCV2000E-FG680

次に，各アルゴリズム向けプロセッサ上でアルゴリズムを実行した場合の実行時間の比較を表3に示す．

表 3: プロセッサによるアルゴリズム実行時間

アルゴリズム	並列度	ベースプロセッサ		各アルゴリズム向けプロセッサ	
		実行時間	高速化率	実行時間	高速化率
バブルソート	2	24.89 μs	0.00%	23.38 μs	7.40%
	3	23.26 μs	6.54%	22.59 μs	9.24%
2分探索	2	2.52 μs	0.00%	2.15 μs	14.68%
文字列照合	2	7.51 μs	0.00%	7.94 μs	-5.73%
画像フィルタ処理	2	66.08ms	0.00%	50.33ms	23.83%
	3	43.09ms	34.79%	38.84ms	41.22%

高速化率は並列度2のベースプロセッサを基準とする

その結果，バブルソート・2分探索・画像フィルタ処理については，合成命令の追加によって実行時間が短縮されていることが分かった．文字列照合については，合成命令による実行ステップ数の効率化ができず，プロセッサの動作周波数も低下しているため，ベースプロセッサの方が効率が良い結果となり，合成命令の実装は不適当であると判断する．

6 おわりに

本稿は，ターゲットアルゴリズムで得た並列度から，プロセッサの命令セットを決定する方法について報告した．4種類のアルゴリズムについて，それぞれに有用な合成命令を追加し，プロセッサの回路規模と実行時間の検証を行った．

今後は，今回改善が得られなかった水平合成命令をプロセッサに追加する手法について検討する．また，アルゴリズム向けプロセッサの命令セット決定からVHDL生成までの流れを確立していく．

謝辞 本研究の一部は，文部科学省科学研究費補助金 基盤研究 (C)(2) 課題番号：13680425 により行われたものである．

[1]論理合成ツール Exemplar Logic 社 Leonardo Spectrum Level3 v2001.1a32 を使用

[2]実装可能 CLB スライス数：19200