5P-6

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† 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た．[1]

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，プロセッサのスペック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．基本命令セットとして 最低限の演算や分岐を行う命令のみ用意する．基本命令セットは，DLX や MIPS などの既存のアーキテクチャの命令セットでオペレーティングシステムやいくつかのアプリケーションを実行したときの，命令毎の実行頻度の高いものを基本命令として採る．

基本命令以外の命令 (拡張命令) の中から，与えられたアプリケーションの解析によって，そのアプリケーションに有用な命令を追加する．また，命令の組み合わせ頻度から 命令や演算の合成を行い，命令実行の時間的・空間的最適化を行う．

この他のプロセッサ生成時の可変要素としては，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，アドレスバス等の幅，命令フォーマットを，アプリケーションに応じて変化させることにする．

3 システムの構成

アプリケーションプログラムから プロセッサを自動生成する流れとして，図１の手順で処理を行う．

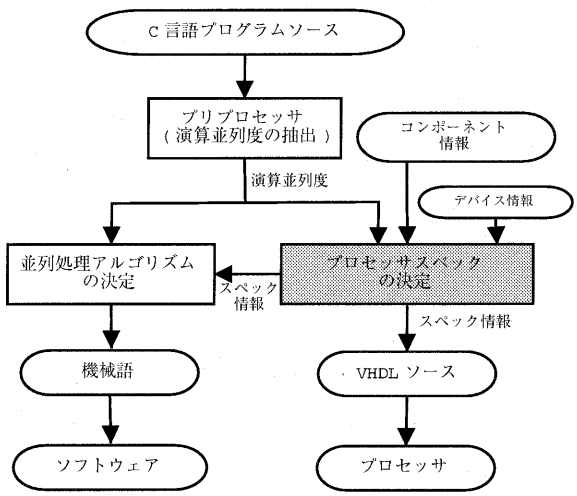

図 1: 処理の流れ

4 スペック決定の流れ

4.1 演算並列度の抽出 [2]

アプリケーションプログラムから，並列演算可能な部分のスケジューリングを行う．ループ処理で依存関

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

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

係がある部分を考慮しながら，データのロードタイミングを決める．

4.2 命令セットの定義

プログラム中の各演算の頻度の統計を取り，それが高いものについて，あらかじめ定義されている拡張命令の中から有用な命令のみを，プロセッサの命令セットに追加する．

次に，水平型・垂直型合成命令の定義を行う．

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を求め，それが高いものについては，1つのオペコードで複数の演算を並列に行う命令として定義する．例えば，図2の演算は，同時に実行される加算と乗算を1つの命令にまとめることで，オペコード1つ分の命令コード領域を減らす．

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．図3は，乗算と加算を連続して行う積和演算命令を定義している．

しかし，これらの合成はあらかじめ各演算器の遅延時間を知っておく必要がある．水平型では，並列に実行される演算の遅延時間を揃えておかなければならないし，垂直型では，クロックサイクル時間内で複数の演算が完了しなければならない．また，演算器はビット長が短くなる程，遅延時間が減少することが分かっている．そこで，コンポーネント情報として，各演算器のビット長を変えて計測した遅延時間をデータベースとして用意しておく．そして，プログラム中で使用されるデータのビット長に合わせた演算器を使って，合成命令を定義する．

図2では，ソースデータが乗算は8bit長，加算は20bit長で収まる場合，予備実験で8bit乗算器と20bit加算器の遅延時間がほぼ同じであると分かっているので，それらの演算器を用いて水平型合成命令の演算を実行することを意味する．図3では，乗算のソースデータが8bit，加算のソースデータが24bitで，8bit乗算器と24bit加算器の遅延時間の合計がクロックサイクル時間以下の場合，それらの演算器を用いて積和演算器を構成する事を意味する．

4.3 レジスタの変数割付

プログラムをライフタイム解析して，変数割付に必要なレジスタ数を決める．変数の最大サイズに合わせてレジスタのビット幅，データバス幅を決める．また，実行に必要なプログラムサイズとデータサイズからメモリサイズを決定し，アドレスバス幅を決定する．

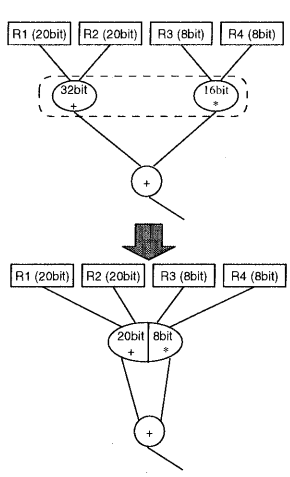

図2: 水平型合成命令

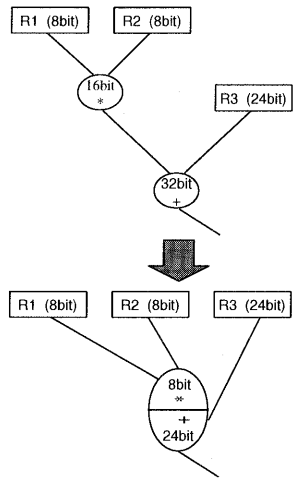

図3: 垂直型合成命令

4.4 スペック決定

演算命令，分岐命令等の命令の種類毎に規則性を保ちつつ，命令長ができるだけ短くなるようにオペコードサイズと各命令のオペコードを決定する．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，エンコード制御コードを付加して，VLIW型の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した．これまでに定義した命令が実装できるかどうか，デバイスのクロックサイクルの制限などの情報を考慮して判断する．また，実装する演算器，レジスタ数等から必要なゲート数を見積もり，それがターゲットとなるデバイスのゲート数より上回る場合は，命令数・レジスタ数の削減を行い，最終的なスペックを決定する．

5 おわりに

本稿においては，プログラムから得られた並列度からプロセッサのスペックを決定する方法について報告した．本稿では，プロセッサ内部のスペックを考えたが，プロセッサの外部にあるメモリへのアクセスについては考慮していないので，それを含めた形でのスペック決定が，今後の課題である．

参考文献

[1] 須藤，宮内，石川："画像処理プロセッサのアーキテクチャ決定法に関する検討 (1)～アプリケーションプログラムからの命令セットの選定法～"，情報処理学会，第59回全国大会，1999

[2] 門脇，松田，宮内，石川："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 －アプリケーションからの並列性の抽出－"，情報処理学会，第61回全国大会，2000