評価システムを用いたプログラム方式専用プロセッサの設計支援

2B－1

中岡 敏博　山口 雅之　山田 晃久　神戸 尚志
シャープ株式会社 生産技術開発推進本部 精密技術開発センター

1　はじめに

アーキテクチャ評価はアーキテクチャの最適化を目指したハードウェア／ソフトウェアの協調設計では重要な要素技術である [2, 3]。

我々は、カスタム DSP のデータパス構成と命令セットの用途に応じた最適化を目的としたアーキテクチャ評価システムを開発している [1]。本システムはプログラム方式 DSP のアーキテクチャ設計において、応用プログラムとデータパス構成を入力として性能、コスト、消費電力の評価を行なう。

データパス構成には演算器だけではなくバスなどの転送資源も含み、転送コストも考慮した評価を行なう。評価は静的解析と実データを用いた動的解析を組み合わせることで実動作を考慮した評価が可能である。さらに、データパス構成の並列制御性に対する制約として制御や命令セットをモデル化することで、処理能力を損なわない命令語長決定やコード割り当て最適化が支援可能である。本手法はコンパイラなどを用いる手法に比べ、設計初期の少ない情報から高速な概略評価が可能である。

本稿では、評価システムの性能評価について述べる。評価システムを実際の DSP 設計に適用してアーキテクチャの性能改善を行ない、その有効性を示す。

2　入力モデル

本システムでは、図1に示すハードウェアとソフトウエアのモデル化を行ない、ハードウェアモデル上でソフトウェアモデルを解析することにより評価を行なう。

2.1　ハードウェアモデル

ハードウェアは、データパス構成と並列制約でモデル化される。

データパス構成は資源とその接続からなる。各資源はライブラリ中の部品へのマッピングをもつ。接続にはクロックや制御信号は含まない。ここで資源は演算器、メモリの他に、トライステートバスやマルチプレクサ等の転送部品を含む。ライブラリは部品毎に実行ステップ数、入出力ポートなどの情報を保持する。並列制約は、制御回路や命令セットなどデータパス構成以外の理由から生じる資源の並列実行可能性に対する制約である。

2.2　ソフトウェアモデル

応用プログラムは C 言語のサブセットで記述される。通常のアルゴリズム記述と違い、データ転送を考慮するために内部 ROM、RAM などを配列で宣言し、初期データの配置や結果データの格納を陽に記述する必要がある。

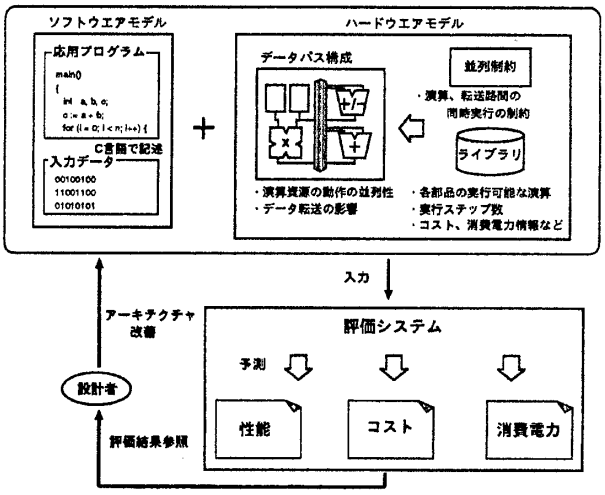

図 1: 評価システムの概略

応用プログラムは、基本ブロックと呼ばれる処理単位に分割される。基本ブロックは分岐等の制御構造を含まない (処理の流れがデータに依存しない) プログラムの連続した部分である。各基本ブロックはデータフローグラフでモデル化する。

3　性能評価手法

提案する性能評価の処理フローを図2に示す。性能評価はデータパス構成、並列制約、応用プログラムなどを入力とし、静的解析と動的解析を組み合わせて行なう。

- 静的解析
  ハードウェアモデルから抽出した情報を用いて基本ブロック単位で資源制約スケジューリングを実施し、実行ステップ数を得る。データパス構成には転送のための資源を含むため、データ転送にかかるステップ数や転送路におけるデータの衝突を考慮した解析が行なえる。

- 動的解析
  応用プログラムをもとに解析用コードを付加した解析プログラムを生成し、実際の入力データを用いて実行することにより各基本ブロックが何回実行されたかを求める。

静的解析による各基本ブロックの実行ステップ数に動的解析で得られたブロックの実行回数で重み付けして総和をとることで全体の処理ステップ数を得ると同時に、実行トレースによる資源利用情報も出力する。

4　評価システムによる改良実験

我々は提案手法にもとづく性能評価を組み込んだアーキテクチャ設計支援システムを開発し、音声処理 DSP

" A Design Method of Embedded System using Architecture Evaluation System ," by Toshihiro NAKAOKA, Masayuki YAMAGUCHI, Akihisa YAMADA and Takashi KAMBE, Precision Technology Development Center, Production Technology Development Group, SHARP Corporation

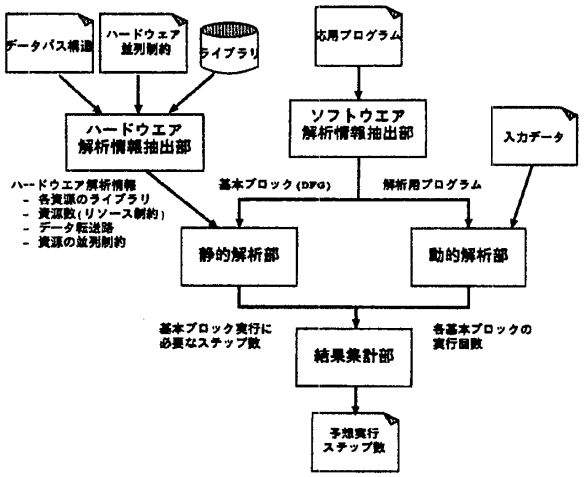

図 2: 性能評価の概略フロー

の評価値や並列制約の妥当性についての実験を文献 [1] で行なった。今回の実験では、既設計 LSI を初期アーキテクチャとし、可変長符号復号化を行なう画像処理 DSP の性能を評価した。

初期アーキテクチャのデータパス構成を図 3(破線部を除く) に示す。応用プログラムは ap1–ap5 の 5 つの処理で構成される。これらを初期アーキテクチャを用いて評価した結果を表 1に示す。表は画像 (サイズ 240×160) の復号処理ステップ数を示している。ここで、AP は各応用プログラムを表し、INS は人手アセンブラアルゴリズムの実行ステップ数であり、EVAL は評価システムによる評価値である。評価値は実際の値に比べ、48% から 293% までの値をとる。誤差の要因として、異なる基本ブロックに属する演算実行の並列性、レジスタの位置・数、人手プログラムの品質を考慮していないことが考えられる。特に ap2, ap5 では、演算実行の並列性を考慮していない影響が大きい。しかし、処理全体に占める各部のステップ数の割合が示す傾向は一致しているので、アーキテクチャ設計時における評価値としては十分である。

表 1: 初期アーキテクチャの評価結果

AP	EVAL	INS	比 *
ap1	634,361 (52.8%)	626,873 (61.6%)	101.2%
ap2	404,800 (33.7%)	138,000 (13.6%)	293.3%
ap3	70,860 (5.9%)	146,720 (14.4%)	48.3%
ap4	59,961 (5.0%)	88,575 (8.7%)	67.9%
ap5	30,915 (2.6%)	16,844 (1.7%)	183.5%
計	1,200,897 (100%)	1,017,012(100%)	118.1%

* INS に対する EVAL の比

この評価結果では処理結果が目標性能を満たさないことが判明した。そのため、次に、アーキテクチャの改良を実施した。処理割合が高い ap1 について、可変長符号の符号長が 4bit までで全体の符号データの 85%、8 bit までで 95% を占めることから、4 bit 復号化テーブル変換を行なう演算器を追加して一部処理をハードウエア化した (図 3, 破線部)。改良は応用プログラムのハードウエア化する部分をサブルーチン化して記述し、データパス構成にその演算を実行する演算器を追加することで行なった。評価の結果を表 2に示す。8 bit 復号化テーブルを持つ演算器を用いた場合、ハードウェアサイズは 4 bit に比べて若干増えるが、ap1 は改良前に比べ 59.2% 減少できることが確認できた。

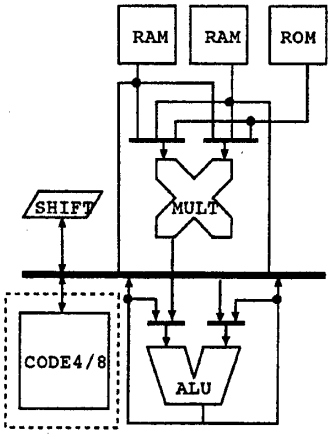

図 3: アーキテクチャ構成

表 2: 演算器追加による改良結果

アーキテクチャ	評価ステップ数	割合
改良前	634,361	100.0%
ハード (4bit) 追加	419,361	66.1%
ハード (8bit) 追加	375,385	59.2%

5 まとめ

データパス部のデータパス構成にもとづくアーキテクチャ評価システムを開発し、画像処理 DSP 設計に適用した。設計改良では、評価結果にもとづく効率の良い処理のハードウエア化を実現できた。

また、我々はコストと消費電力に関して同レベルのモデルからの予測手法の確立も目指している。コストは演算資源構成から予測できると考えられる。消費電力は評価過程で得られる各コンポーネントの実行頻度と入出力ポートのビット幅、電源電圧などの情報を元に評価対象のアーキテクチャの消費電力を見積もる機能を開発中である。

参考文献

[1] 山口雅之, 山田晃久, 中岡敏博, 神戸尚志, “演算資源構成をにもとづくアーキテクチャ評価の一手法” 情処研報, 95-DA-78-14, pp.85-90 (Dec. 1995).

[2] 冨山宏之, 赤星博輝, 安浦博人, “アーキテクチャ評価用コンパイラの自動生成,” 情処研報, 93-DA-69-19, pp.143-150 (Dec. 1993).

[3] 中田武治, 佐藤淳, 塩見彰睦, 今井正治, 引地信之 “ASIP 向きハードウエア / ソフトウエア・コデザインシステム PEAS-I におけるハードウエア生成手法,” 情処研報, 93-DA-69-20, pp.151-158, (Dec. 1993).