1M−6

複数のアルゴリズムを実行する専用プロセッサの設計とその評価

池永 剛　　白井 克彦
早稲田大学

1. はじめに

高級言語で書かれたアルゴリズム記述を入力とする専用プロセッサ設計支援システムの開発を行なっている[1]。これは、ユーザが、実行したいアルゴリズムを通常のソフトウェアを記述するように入力することにより、そのアルゴリズムを効率よく実行する専用プロセッサを合成し、同時にそのプロセッサ向けの最適化コンパイラを自動生成することを目的としている。今までのシステムは単一のアルゴリズムのみ対象としていたが、ある程度の汎用性を持ったプロセッサ設計への要求は高い。そこで、複数のアルゴリズムを入力できるようにシステムを拡張し、実際に汎用性を持った専用プロセッサ設計を行ない、評価を試みる。

本報告では、複数の信号処理アルゴリズムをシステムに入力し、信号処理プロセッサの設計を行なう。また、設計されたプロセッサを既存の信号処理プロセッサと比較し評価を行なう。

2. 汎用性を持った専用プロセッサ設計

専用プロセッサ設計を考えた場合、対象をできるだけ狭い範囲に限定する方が、より高性能なプロセッサ設計が可能である。しかしながら、個々のアルゴリズムごとに、数多くのLSIを設計するよりも、速度性能を多少犠牲にしても、ある程度の汎用性を持ったプロセッサ設計への要求は高い。この場合も、幅広いアルゴリズムを対象とすると、専用プロセッサとしての意味が失われてしまうが、アルゴリズムの特徴が似ているものに限定すれば、それほど、無駄な要素が生成されることはない。ここでは、このようなアルゴリズム群として信号処理アルゴリズムを取り上げ、有効性を検討する。

3. 処理概要

本システムを用いて、信号処理プロセッサを設計する際の処理概要を図1に示す。システムは入力された複数の信号処理アルゴリズムそれぞれに対して、命令セット、記憶要素の最適設計を行う。そしてそれらのOR要素を取って全体を統合し、合成したプロセッサ(DSP)をRT情報として出力する。また、同時にそのプロセッサ向けの最適化コンパイラを生成する。

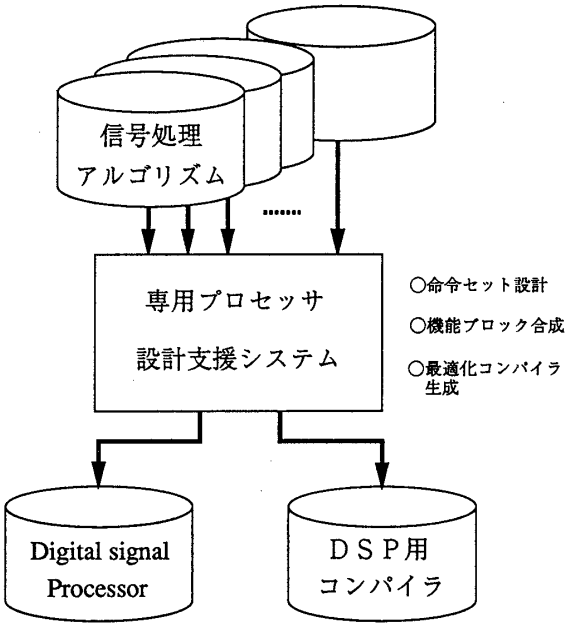

図1 処理概要

4. 信号処理プロセッサ

本システムに次に示す10個の代表的な信号処理アルゴリズムを入力し、プロセッサ設計を行なう。

- ○ FFT（高速フーリエ変換）
- ○ 自己相関関数
- ○ デルタ変調方式の符号化、復合化
- ○ ADPCM符号化、復合化
- ○ Durbin-Levinson-板倉法
- ○ Pitch抽出
- ○ PARCOR格子型フィルタ
- ○ FIRフィルタ（1次）
- ○ IIRフィルタ（3次）
- ○ Biquadフィルタ

これらのアルゴリズムを入力した結果、合成された命令セットを図2に示す。命令セットフォーマットは、演算命令、分岐命令、メモリアクセス命令ごとに規則性を保った上で、命令長ができるだけ短くなるように決定される。

図中のCALLは関数呼び出しである。関数については、関数器として布線論理を用いて合成される。ただし、標準関数については、あらかじめライブラリとして関数器が登録されていると仮定してある。ここでは、関数としてSIN、COSが用いられており、ライブラリを用いて実現されている。

Design and Evaluation of Special Purpose Processor Executing Muliple Algorithms.
Takeshi IKENAGA, Katsuhiko SHIRAI
Waseda University

図中のOP1、OP2は頻度解析、命令の組合せ解析の結果生成された複合命令であり、それぞれ積和命令、積差命令である。これは、全ての入力アルゴリズムに共通して、積和、積差命令が高頻度に参照されていることを示している。

5. 評価

表1に本システムで得られたプロセッサを代表的な信号処理プロセッサTMS320C25（以下C25）と比較した結果を示す。

5. 1. ハードウェアの比較

命令数はC25と比較して少ないが、これは本システムの場合、汎用性を持たせたといっても、完全な汎用DSPではないためである。本システムでは、設計途中で命令の追加等のユーザの介入を許しており、これによりさらに汎用性を追求したプロセッサ設計も可能である。また、最近のDSPは浮動小数点命令を持つものが増えてきているが、これについては検討中である。

特殊命令は、両者とも信号処理アルゴリズム特有な積和、積差といった演算を行なうものを備えている。

内部データRAMは多いが、これはピッチ抽出などの多くのデータ領域を必要とするアルゴリズムが含まれるためである。このアルゴリズムを実行する場合、

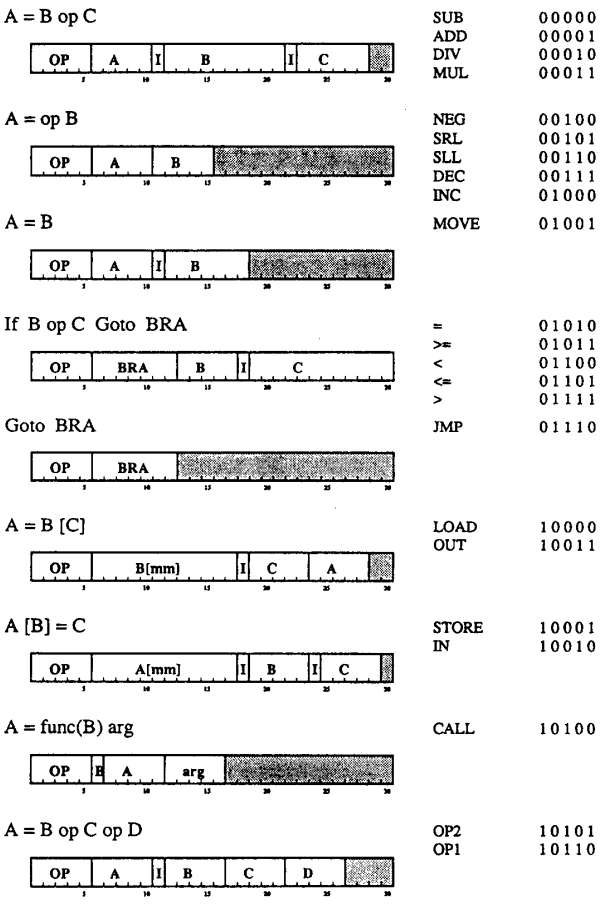

図2 命令セットフォーマット

表1 DSP(TMS320C25)との比較

	本システム	TMS320C25
命令数	23個	133個
特殊命令	積和、積差命令	積和、積差命令
内部データRAM	4096W	544W
プログラムROM	128W	4096W
レジスタ	32W	8W
ソフトウェア開発	Pascal	アセンブリ言語(C言語)

C25では、内部RAMと外部RAMを使い分けなければならず、効率が悪い。

プログラムROMは少ないが、これは入力アルゴリズムの規模がそれほど大きくないのと、コンパイラが効率のよいコード生成が可能であるためである。

レジスタは多いが、これは本システムがRISCプロセッサのようにレジスタ演算を基本としているためである。C25はレジスタが少ないためメモリ演算が多くなり効率が悪い。

5. 2. ソフトウェアの比較

ソフトウェア開発は、本システムではPascal、C25ではアセンブリ言語を用いて行う。C25はC言語によるソフト開発も可能であるが、コンパイラは、現状ではプロセッサの性能を必ずしも十分引き出していない。一般に、DSPのソフト開発は低次元のアセンブリ言語を用いて行なわねばならず、効率が悪い[2]。本システムは、高級言語によるソフト開発を前提としており、命令セット等もコンパイラが最適なコードを生成できるように決められるため、効率の良いソフト開発が可能である。このように、ソフト開発環境は本システムの方が優れている。

6. むすび

専用プロセッサ設計支援システムを用いて、複数のアルゴリズムを実行できる専用プロセッサの設計を行なった。本システムを用いることにより、今までの専用プロセッサの弱点であったソフトウェア開発環境も含めたプロセッサ設計が可能である。今後は、既存の論理合成システムと接続し、合成結果をフィードバックさせて、さらにシステムの性能向上をはかりたい。

参考文献

[1] 池永、竹沢、白井：高位の仕様記述に基づく専用プロセッサ設計支援システム、情報処理学会第39回全国大会、4X-7、(1989-10)

[2] 小野定康：DSPのプログラム開発環境、情報処理、Vol.30、No.11 (1989)