

# 信号処理プログラムを 最適化するための基礎知識

最近のデジタル信号処理のプログラミングは、開発環境により最適化が自動的に行われるようになってきたため効率が上がっている。しかし、ハードウェアに密接したデジタル信号処理の場合、開発者自身が最適化を図らなければ目的とする性能を出せない。本章では、どのようなときに最適化が必要になるのか、どのようにすれば最適化できるのかについて解説する。

中村 健真

## 1 最近のDSPとプログラムの最適化

近年、DSPの開発環境は日に日に充実してきており、Texas Instruments社やAnalog Devices(以降AD)社などのDSPメーカーが提供するコンパイラの最適化能力は、驚くほど向上しています。試しに、簡単なベクトル処理をさせてみると、並列実行やMAC演算を駆使した最適化が施され、ハードウェアの能力を使い切っていることに驚かされます。また、DSP自身の能力も向上しているため、DSPプログラムの最適化などは、もはや意味がないように思えます。

### ● なくなる最適化の必要性

しかし、こういった状況にもかかわらず、次の二つの理由から今後も信号処理プログラムの最適化は重要であり続けると考えられます。それは、

- (1) 民生用の画像処理アプリケーションの普及
  - (2) 低価格信号処理プロセッサの普及
- という理由です。

現在、ブルーレイ・ディスク・プレーヤやデジタル・テレビのビデオ・デコーダには、プログラマブルDSPが用いられています。これらはSoC LSIの中に埋め込まれているためなかなか表に出てきませんが、巨大な並列SIMDアーキテクチャ<sup>注1</sup>を使って多数のピクセルを一気に処理できるようになっています。

これらがソフトウェアで処理されているのは、複数の画像圧縮規格に対応しなければならないことと、画質向上のためのポスト・プロセッシング(後置処理)をカスタム化しやすいことが理由だと思われます。CEATEC 2008の会場では、Cellプロセッサを利用してさまざまな信号処理を行えるCell TVが展示されていました(写真1)。デモンストラーションのほか、開発・研究用のプラットフォームとして使っているようです。これなどは、DSPのプログラマブルな性質を活かした好例といえます。

デジタル・テレビやBD(Blu-ray Disk)-ROMやの信号処理は、大量のデータを高速で処理するため、どうしても処理効率を意識せざるを得ません。最適化によって処理効率を上げることができれば、与えられたアーキテクチャの能力に余裕が生じます。その分を使えば、付加機能を実装できるかもしれません。あるいは、1クラス下のハードウェアを使うことでコストダウンを図れるでしょう。

注1: SIMD (Single Instruction Multiple Data) アーキテクチャは、1回の命令で複数のデータを同時に処理する技術。音声や画像を高速で処理するのに有効。



写真1  
東芝のCell  
TV



一方、低価格のDSPも登場し、ローエンドの信号処理アプリケーションの裾野が拡大しています。DSP機能を組み込んだ16ビット・マイコンであるdsPICがその代表です。このような小規模DSPは、最近の組み込みマイコンと同様にフラッシュROMやRAMを内蔵しているため、1個で小規模システムを完結できるのが強みです。

dsPICはクロック周波数こそ120MHzですが、演算性能は30MMAC/s<sup>注2</sup>と、10年ちょっと前のDSP程度の速度しかありません。しかし、低価格なので、多分野に応用できます。そうすると、小さな処理能力でアプリケーションを実行することになり、最適化の需要が出てきます。

以上のことから、SoC LSIで使われるハイエンドDSPや、マイコン市場で使われるローエンドDSPではまだまだ最適化は必要になると考えられます。

## 2 最適化の方法には2種類ある

DSPプログラムの最適化には、目的別に2種類あります。それは、スループットの最適化と応答時間の最適化です。

### ● スループットの最適化

スループットの最適化は、メディア信号処理などに使われるものです〔図1(a)〕。この最適化は、単位時間に処理できる信号の量を最大化することが目的です。そのため、信号処理の遅延そのものはそれほど問題になりません。

もちろん、問題にならないといっても限度がありますが、そもそもデジタル・テレビやBD-ROMのストリームとデコードされたデータの間の遅延量を鋭敏に察知することはできません。そこで、多少の遅延を許容する代わりに、単位時間当たりの処理能力を高めるのがスループットの最適化です。

スループットの最適化は、工場の流れ作業に似ています。それは、部品が工場に入ってから自動車になって出てくるまでは時間がかかっても、出口からは次々と短い間隔で完成した車が出てくる場所です。

### ● 応答時間の最適化

応答時間の最適化は、スループットの最適化と異なり、ひたすらデータの入力から出力までの時間を縮めることに

注力します。この最適化戦略が効果を発揮するのは制御系です。モータ・コントロールのような機械系システムでは、遅延の大きさは制御されるシステムの特性に多大な影響を与えます〔図1(b)〕。制御装置の遅延が大きくなれば、系のモデルを立て直さなければなりませんし、何よりきびきびした動作が失われていくことになります。

応答時間の最適化は、反射神経をよくするトレーニングと似たところがあります。一発勝負のデータ処理を磨き上げることで、機敏な動作を実現するのが応答時間の最適化です。

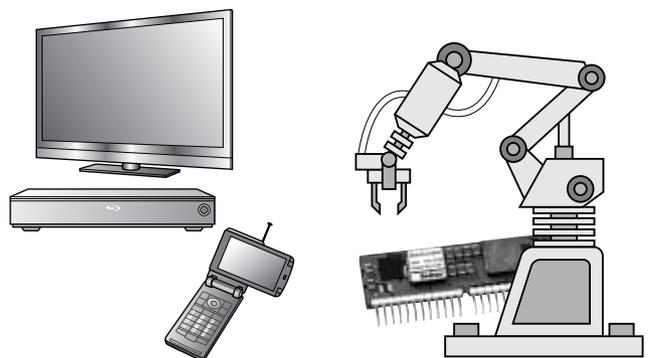
ところで、スループットの最適化と応答時間の最適化には、相反する面があります。したがって、両者を同時に実現しようとするときちょっと困ったことになりますが、片方だけなら定番がいくつかあります。そこで本章では、二つの最適化技法のうち、スループットの最適化について、よく使われる考え方や手法を紹介します。なお、DSPのアーキテクチャを使用した命令レベルで最適化する方法も面白いのですが、それについては別の機会に紹介します。

すでに述べたように、メディア信号処理ではスループットの最適化が重要です。最近の傾向は、これまで以上にノイマン・ボトルネックへの対処が重要になっています。

## 3 スループットの最適化

スループットの最適化には、DSPの命令の最適化、メモリ・アクセスの最適化、データ転送の最適化などいくつかの面があります。しかし、どの局面も最終的にはノイマン・ボトルネックとの戦いになります〔図2〕。

もともと、ノイマン型アーキテクチャのコンピュータは



(a) スループット性能が求められる分野

(b) 応答時間が求められる分野

図1 応用分野によって異なる最適化

注2：MMAC/s (Million Multiply Accumulates per second) は、1秒当たりの100万乗算アキュムレート数。