ＤＳＰ の利用技術の研究

長谷川　直樹*　　五十嵐　晃*　　松本　好勝*

Study of technology for using DSP

by

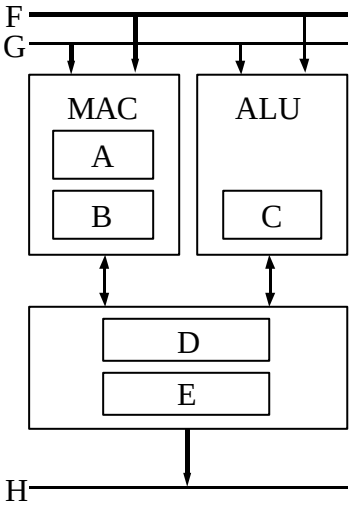
HASEGAWA Naoki*, IKARASHI Akira* and MATSUMOTO Yoshikatsu*

抄録

DSP の製品への応用に役立てる目的で、その特性について評価を行った。ディジタルフィルタのアルゴリズムを DSP にプログラミングし、動作の検証を行った。汎用の CPU とのパフォーマンスの比較を行った。

1.　緒言

近年のディジタル携帯電話に代表されるようにディジタル通信、マルチメディア関連製品の普及は著しい。それらはいずれも大量のディジタルデータを高速に処理する技術を必要とするのが特徴である。



A:乗算器　B, C:加算器
D, E:アキュムレータ
F, G:バス（読込）
H:バス（書込）

図 1　DSP の構造の例

DSP(Digital Signal Processor) は高速乗算器、ハーバード型アーキテクチャ、データ･アドレス計算ユニット等を基本アーキテクチャとし、CPU などの汎用プロセッサに比べ、ディジタルフィルタなどに多用される積和演算処理により適したものとなっている（図 1）。そのため、リアルタイムでのディジタル信号処理において高速化を図ることが可能となる。この特徴から前出のディジタル携帯電話等、その利用場面が急速に拡大している。

今回は基本的なディジタルフィルタの問題に応用し、そのパフォーマンスについて調べた結果について報告する。

2.　ディジタルフィルタについて

従来からノイズ対策に用いられてきたアナログフィルタは実際の素子を用いることによる値の経時変化などが問題となる場合がある。そのため半導体を用いたディジタルフィルタが注目されている。

フィードバックパスを持つ IIR フィルタとフィードバックパスを持たない FIR フィルタとに分けられる。

*下越技術支援センター

(1)FIR フィルタ

式

$$y_n = a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2} \qquad (2.1)$$

で表される処理は FIR フィルタの一種である。回路は図 2 のようになる。

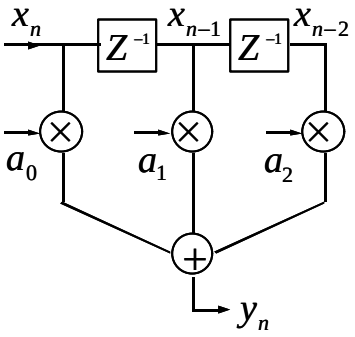


図 2　FIR フィルタの例

ここで a_0, a_1, a_2 は定数、x_n は現在の信号の入力値、x_{n-1}, x_{n-2} はそれぞれ 1 つ前、2 つ前の信号の入力値、y_n は出力値である。

(2) IIR フィルタ

式

$$y_n = a_1 y_{n-1} + a_2 y_{n-2} \qquad (2.2)$$

で表される処理は IIR フィルタの一種である。回路は図 3 のようになる。

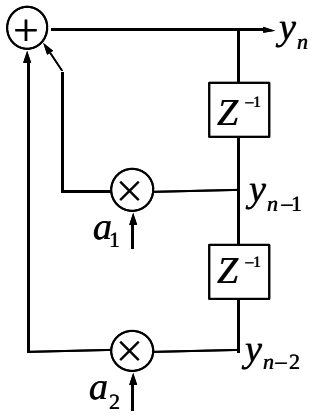


図 3　IIR フィルタの例

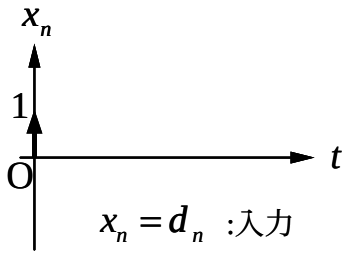
ここで a_1, a_2 は定数、y_n は出力値、y_{n-1}, y_{n-2} はそれぞれ 1 つ前、2 つ前の出力値である。

回路において⊕は加算器、⊗は乗算器、Z^{-1} は単位遅延回路である。

3.　IIR フィルタの応用

図 4 のような入出力特性をもつフィルタを考える。

入力



出力

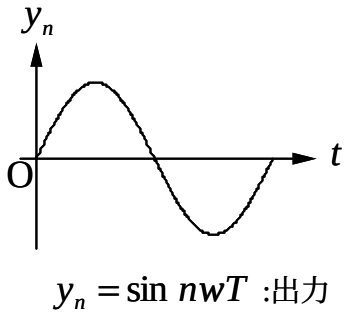


図 4　入出力特性

これは $0 < n$ において式

$$y_n = 2\cos wT \cdot y_{n-1} - y_{n-2} + \sin wT \cdot x_{n-1} \qquad (3.1)$$

で表される。ここで T はサンプリング周期である。初期条件を

$$y_0 = y_{-1} = 0 \qquad (3.2)$$

とする。入力は

$$x_n = 1\ (n = 0),\ x_n = 0\ (0 < n) \qquad (3.3)$$

である。これは図 3 の回路において

$$a_1 = 2\cos wT,\ a_2 = -1 \qquad (3.4)$$

および

$$y_1 = \sin wT \qquad (3.5)$$

と与えれば再現することができる。また z 変換においては

$$X(z) = \sum_{n=0}^{\infty} d_n z^{-n} = 1 \qquad (3.6)$$

$$\begin{aligned} Y(z) &= \sum_{n=0}^{\infty} \sin nwT \cdot z^{-n} \\ &= \frac{\sin wT \cdot z^{-1}}{1 - 2\cos wT \cdot z^{-1} + z^{-2}} \end{aligned} \qquad (3.7)$$

であり、これを用いて導出してもよい。

4. DSP 上での動作

3 の IIR フィルタのプログラムを DSP 上で動作させ処理時間の測定を行った。プログラミングは Code Composer Studio の C 言語を用いて行った。

動作環境

使用 DSP：Texas Instruments 社
TMS320C6711
クロック数 150MHz
浮動小数点型

開発環境：Texas Instruments 社
C6711DSK
Code Composer Studio ver.1.2

図 5　DSK(C5402,C6711DSK)

測定対象： 3 の処理 1 出力
測定結果：86.0ns(平均 12.9count)

また 3 の処理に
① 値の引渡し
② テキスト出力（printf12 文字）
③ テキスト出力（printf11 文字）
の 3 つの処理を追加して実行させた結果は 1.55μs(平均 232count)であった。

5. 他のプロセッサとの比較

同一のプログラムを他の汎用プロセッサ（パーソナルコンピュータ）で実行した結果は以下の通りである。C 言語は MS-DOS 上のものを用いた。

(1) Pentium150MHz (Windows98)で実行した場合 99.5ns
(2) PentiumⅢ1.13GHz (Windows2000) で実行した場合 18.5ns

同クロック数の汎用プロセッサより速く、最新の高クロック数のものに比べれば遅いという結果であった。DSP は汎用プロセッサに比べて安価であり、高速処理の目的で小型の機器に用途を限定した形で組みこまれていることが多い。今回の結果はそのような DSP の特徴を数値的に表したものといえる。

6. まとめ

DSP を用いたディジタルフィルタを製作し、そのパフォーマンスを測定した。実行時間を数値化し、汎用プロセッサの処理時間と比較することができた。

プログラミングについてはある程度習熟することができたが、さらに高度な処理を行うためには、より内部構造まで意識して行う必要があると感じられた。

画像等も取り扱うことができるが、汎用のフォーマットをサポートしていないため、やや扱いにくい印象である(図 6)。初心者は使いこなすまでにそれなりに時間をかける必要があると思われる。

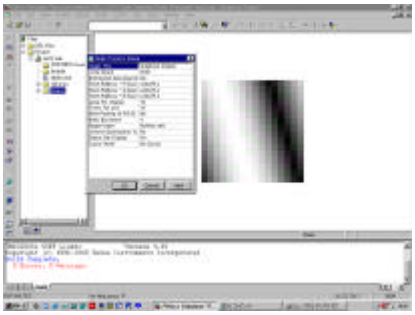


図 6　画像を扱った例

外部入力に対するリアルタイム処理とリアルタイム処理時間の把握が今後の主な課題としてあげられる。