

第2章

DSPエンジンを持つ信号処理用マイコンdsPICでサンプリング・レート変換してみる

C言語のままマイコンでデジタル信号処理する
メリットとデメリット

岩田 利王 Toshio Iwata

第1章でサンプリング・レート変換アルゴリズムである147/160倍処理を、固定小数演算のみで処理できるようになりました。このC言語プログラムであれば、組み込みマイコンでも動作させられそうです。ここではMicrochip社製dsPIC搭載DSPマイコン・ボードに作成したC言語プログラムを移植し、DSPマイコンでサンプリング周波数変換の一部の処理を実行してみます。

1. マイコンでデジタル信号処理する際の注意点と問題点

マイコンならC言語のままプログラムをハードウェア(ROM)化できます。もしそれで十分な性能が得られるのなら、わざわざHDL化する必要はありません。ここではマイコンによるリアルタイム信号処理の可能性を探ってみます。

● マイコンにおけるC言語はハードウェア独特の記述が必要

ご存知のように、最近のマイコン開発はほぼC言語で行います。ただ第1章のコラム1で述べたように、ハードウェア化を意識した記述にする必要があります。

● FPUを持たないマイコンなら全部整数で計算させる

また、今回使用するマイコンdsPICはFPU(Floating Point Unit)を持っていないので、事実上信号処理の演算は整数に限定されます^{注1}。したがって変数や係数を固定小数点化する必要があります。

● ビット数の制限による性能の劣化

FPGAの場合、デジタル・フィルタの係数などのビット数は任意に設定できます。それに対しマイコンの場合、ビット数はある決まった数に限定されます。

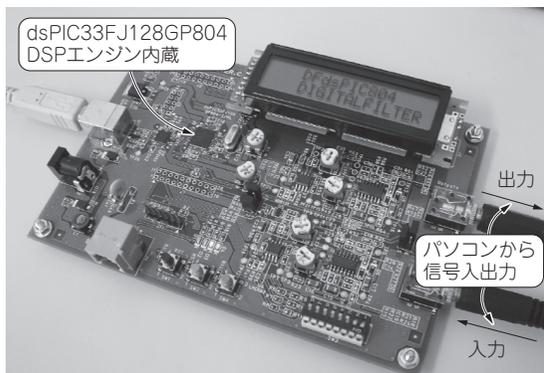


写真1 デジタル信号処理ボードdsPICマイコン版(DFdsPIC804)

例えばdsPICは16ビット・マイコンなので、その信号処理機能をフルに引き出すには係数のビット数は16である必要があります。第1章のWindowsアプリケーションでは20ビットだったので、それに比べると確実にSN比は劣化します。

● FPGAは並列処理、マイコンは逐次処理

サンプリング・レート変換器では、A-Dコンバータから常時入ってくるデータを信号処理し、D-Aコンバータに次から次へと渡します。したがって、一定時間に一連の処理を終えるリアルタイム性が必要になります。

FPGAは並列処理が容易です。同じ処理をもう1チャンネル増やしたいなら、同じ回路をもう一つ並列に動かせばよいだけだからです。

それに対しマイコンは逐次処理なので、1チャンネルで処理が間に合ったとしても、2チャンネルに増やしたときに間に合うとは限りません。タイマやDMAなど、独立して動く機能もありますが、基本的には一度に一つの仕事しかできないからです。

● ちょっとした信号処理システムならマイコンでもできそう

マイコンには、WindowsアプリケーションからC言語で書かれたプログラムをサクッと拝借して実装できるという魅力がありますが、FPGAと比較すると既に説明したような欠点が確実にあります。

それではマイコンならどれくらいの演算量のシステムがどれくらいの精度でできるのでしょうか? これから7/5倍レート変換をdsPICに実装してみます。

2. デジタル信号処理用マイコンdsPICでレート変換してみる

dsPICはMicrochip社が提供する16ビット・マイコンです(命令長は24ビット)。その名前から分かるよ

注1: 浮動小数点でも開発できるが、サイクル数が何倍もかかって演算が間に合わなくなる。