

ZYBOで始めるリアルタイム 信号処理システム設計入門

③ ロジック×ARMで実現! 堅牢で柔軟な デジタル・フィルタ

岩田 利王
Toshio Iwata

「複雑な浮動小数点演算」と「リアルタイムな固定 小数点演算」を兼ね備えたシステムを作るには

前回(本誌2015年6月号)はZYBOにLinux OSを載せて、C言語でデバイス・ドライバを操作してLEDを光らせたり、スイッチの状態を見たりしました。

今回は、図1のようにデジタル・フィルタの係数を計算してその特性を変えろといった、より複雑な操作を行います。

▶YouTubeでZYBO + filterなどで検索してみよう!
図1の様子は以下のサイトで確認できます。

<https://www.youtube.com/watch?v=D1apB9K4hR0>

動画のタイトルは「Zybo controls a digital filter」です。このようにLinux OSを載せるとZYBOがまるでPCのようになり、従来ガチガチのハードウェアであるFPGAがずいぶん柔軟になることが分かります。

▶SoC FPGAのメリットは硬軟合わせ持つシステムを実現できること

動画では以下の点をアピールしています。

① 平方根、三角関数、その他複雑な浮動小数点演算はLinux上のC言語で行う。

② 高速でリアルタイムな信号処理はHDLで開発し、FPGA内のロジックで行う。

①はロジックに不向きな仕事なのでOSに任せます。逆に②はLinuxのようなOSには不向きなのでロジックに任せます。

このように上手に役割分担すれば、結果的に柔軟で高速で安定的なシステムを実現することができます。特に②は、BeagleBone BlackやRaspberry Piなどには無理(プログラマブル・ロジックを持たない)なので、ZynqのようなSoC FPGAの大きな優位点だと言えます。

OSだけでは信号処理の リアルタイム性を保つのが難しい

図1のようなシステムをLinux OSだけ(ロジックなし)で行うとどのような問題が生じてくるか、ここで

C言語で書かれた
アプリケーション
で係数を導出する

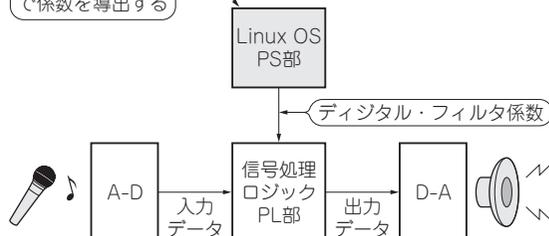


図1 デジタル・フィルタの係数をOSから制御する

実験してみます。また、逆にロジックだけ(OSなし)で行う際の問題点についても言及します。

● Linux OSだけでリアルタイム信号処理をやっ ていけない理由を体感しよう

ZYBO搭載のZynq デバイスはPL部(プログラマブル・ロジック)を持つので、高速でリアルタイムなデジタル・フィルタを安定的に実現できます。

しかし、ここではあえてPS部(プロセッサ・システム)にインストールしたLinux OSにその処理を任せます。それによってどのような不具合が起こるかを体感しましょう。

▶オーディオ・コーデック(注1)のデバイス・ドライバを介して入出力、間に信号処理

図2ではZYBOのMic入りにマイク、Headphone出力にヘッドホンを繋いで信号処理の実験をしています。OSはA-Dコンバータのデバイス・ドライバを介してデータを読み、フィルタリングの後、D-Aコンバータのデバイス・ドライバを介してデータを書いています。

注1: ZYBOにはSSM2603CPZ(アナログ・デバイス)というオーディオ・コーデックが搭載されており、その中にA-D/D-Aコンバータがある。