

オーディオ/モータ/信号処理…タスク数個のゴリゴリリアルタイム用途にピタッ!

並列処理プロセッサ xCORE 徹底研究

MCUでも
FPGAでもない
ヘンテコッ!?

第3回 ソフトでリアルタイムI/Oを簡単に実現するメカニズム

圓山 宗智

前回は、英国XMOS社がリリースした、複数の32ビットCPUコアを内蔵してプログラムを並列処理する組み込み向けマルチコア・プロセッサxCOREのアーキテクチャと、専用プログラム言語XCによる並列プログラムの記述方法を解説しました。xCOREは1デバイスあたり4個から最大32個のCPUコアを内蔵して並列処理させるものであり、それらのコアを効率的にどのように使いこなすかに悩むことは大変楽しいものです。入手性のよい安価な評価ボード「startKIT」がリリースされており誰でも簡単にxCOREアーキテクチャに触れることができます。

今回は、xCOREの特徴の一つである、ハードウェア機能をソフトウェアで柔軟に実現する機能と、そのXC言語での記述方法について解説します。

周辺機能に対する基本思想

● シリアルやタイマなどの簡単な周辺機能はソフトで実現する

xCOREというデバイスは、複数の32ビットCPUコアを内蔵したマルチコアな組み込み向けMCU (Micro Controller Unit) です。各コアはDSP (Digital Signal Processor) 機能も含んでおり、そのトータル性能は、8コア内蔵品で最高500MIPS、16コア内蔵品で最高1000MIPSにのぼり、非常に高性能なマイコンです。

一方、周辺機能としては、USB 2.0 (Universal Serial Bus) やA-D変換器は内蔵していますが、一般的なマイコンが持つシリアル通信機能や波形入出力用タイマ機能といった、比較的シンプルな論理回路で実現できる機能は内蔵されていません。シリアルやタイマなどは、CPUコアとタイミング制御付きのI/O機能を使ってソフトウェアで実現するのが基本的な考え方になっています。

この考え方によりxCOREは、FPGA (Field Programmable Gate Array) のようなハードウェアに対するフレキシビリティも提供できるのです。

● 参考：一般のマイコンの場合…ソフトで実現すると正確なタイミングを実現しにくい

一般のマイコンでCPUによるソフトウェアを使ってタイミングに応じた処理をする場合、周辺機能のタイマなどから受ける割り込み要求に同期させたり、周辺機能のレジスタをポーリングするなどで同期させたりする方法があります。しかし、割り込みによる方法の場合、割り込み応答時間の長さにより高精度なタイミング同期ができず、また優先順位が高い割り込みサービスを処理中だと、さらに応答時間が長くなる問題が出て、あまり正確なタイミング処理ができません。レジスタをポーリングする方法では、待ち時間が長くなりマイコン内の唯一のCPUリソースを無駄にしてしまう問題があります。

● ハードウェア制御に不可欠なタイミングの概念

外部の入出力信号を相手にするシリアル通信機能やタイマ機能は、一般に論理回路というハードウェアで実現しており、その動作にはクロック信号が必要です。そのクロックを基準にして、外部からの入力信号をサンプリングしたり、外部への出力信号の変化タイミングを決めたりします。

xCOREで外部の入出力信号を相手にするハードウェア機能を実現する場合も、XC言語によるソフトウェアで記述しますが、クロックに対応するタイミング概念を簡単かつ効率的に実現する方法を提供されています。以下、その実例をいくつか説明します。

ソフトからタイミング制御するキモ! 10ns刻みハードウェア・タイマ

● 時計機能を実現するハードウェア・タイマ

xCOREアーキテクチャには、10ns刻みでカウント・アップするカウンタ (ハードウェア・タイマ) が内蔵されており、それを基準時計としてソフトウェアから簡単にアクセスできます。

ハードウェア・タイマをXC言語から読み取る方法をリスト1(a)に示します。timerという型の変数