

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4回 マルチコア処理性能の簡易評価

圓山 宗智

リスト1 並列動作するマルチコア数による性能影響実験のプログラム

```
#include <xsl.h>

out port J7_01 = XS1_PORT_1F; // X0D13
out port J7_02 = XS1_PORT_1H; // X0D23
out port J7_03 = XS1_PORT_1G; // X0D22
out port J7_04 = XS1_PORT_1E; // X0D12
...

void Task_Toggle_Port(out port PORT)
{
    while(1)
    {
        PORT <: 1;
        PORT <: 0;
    }
}

int main(void)
{
    par
    {
        Task_Toggle_Port(J7_01);
        Task_Toggle_Port(J7_02);
        Task_Toggle_Port(J7_03);
        Task_Toggle_Port(J7_04);
        ...
    }
    return 0;
}
```

出力ポートの定義

出力ポートを
トグルし続ける
だけのタスク

メイン・ルーチン

上記タスクを必要個数だけ起動

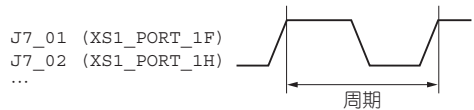


図1 並列動作するマルチコア数による性能影響実験

の製品の場合、1タイルの最大性能は500MIPSに制限されています。すなわち、単一コアあたりの性能は、タイル内で同時に動作するコア数によって制限を受け、同時動作するコア数が増えるとコアあたりの性能が低下します。1枚のタイル内で同時動作するコア数が4個までなら各コアの最低性能は125MIPSを維持しますが、それ以上であれば $(500 \div N)$ MIPS (N は同時動作コア数で4より大の場合) になります。これを実際に確かめてみましょう。

リスト1のように相互連携のない独立したタスクを複数同時動作させます。各タスクは個々のコアに自動的に配分されます。各タスク内の動作は、出力ポートのトグルを図1のように繰り返すだけです。並列動作するタスク数を変化させて、各出力ポートのトグル周期を測定した結果を表1に示します。

タスク(コア)数が4個までなら、コアの数に依存せず一定の性能でしたが、タスク(コア)数が4個を超えると、その数に応じて性能が低下しています。予想どおりの結果になりました。

今回は、評価ボード「startKIT」を使ってマルチコア・プロセッサxCOREアーキテクチャのマルチコア性能を簡単に評価した結果を紹介します。

xCORE アーキテクチャの性能評価実験

● 「startKIT」を使って性能評価

ここでは、評価ボード「startKIT」を使って、xCOREアーキテクチャの簡単な性能評価を行ってみたい。これにより、このデバイスの実力の一部をかいま見ることができるとおもいます。

● 並列動作するコア数による性能影響

xCOREの1枚のタイル内には最大8個のxCOREコア・ロジックが内蔵されますが、動作周波数が500MHz

● 処理をコアに分散させても性能が上がるわけではない

この評価結果を見ると、処理を並列分散させても、ハードウェアの制約によりそのまま性能が向上するわけではないことが実感できます。結局、8コアを内蔵する1枚のタイル内の性能はトータルで500MIPSと決まっています。xCOREのマルチコアのメリットは、通常のCPUのマルチ・タスク処理で生じるオーバヘッドの削減と、個々のタスクにハードウェア処理機能を持たせて独立動作させることができるシステム構築への柔軟性にあるようです。