

Cortex-Mマイコン 低消費電力モードの研究

新連載

第1回 マイコンの低消費電力化の基本方針

中森 章

ARM Cortex-MシリーズのCPUの特徴の一つに低消費電力があげられます。本連載では、Kineticマイコン(フリースケール)を例に、Cortex-Mコアや半導体メーカーが個別に用意している低消費電力モードのしくみや使い方を紹介していきます。

半導体から！ マイコンの低消費電力化の基本方針

まずはどのようなしくみをマイクロコントローラに実装すれば低消費電力になるのかについて説明します。一般的にMOSトランジスタの電力は次の式で与えられるといわれています。

$$P = aCV^2f$$

a ：動作率， C ：負荷容量， V ：電源電圧，
 f ：動作周波数

経験的に動作周波数が高くなれば消費電力が増えることはわかんと思います。オームの法則などから、

$$P = IV = V^2/R$$

と、電力は電圧の2乗に比例しますから、この式は素直に受け入れられると思います。ということは、低消費電力を実現するためには

- (1) a (動作率)を減らす
- (2) C (負荷容量)を減らす
- (3) V (電源電圧)を減らす
- (4) f (動作周波数)を減らす

といった選択肢しかありません。 a はプログラム依存、 C は回路論理依存なので、これらを制御して低消費電力を実現するのは難しいでしょう。すると残りは電源電圧と動作周波数を減らすという方法です。実際に、低消費電力を実現するためには、電源電圧と動作周波数を制御するしか方法はありません。つまり、

- (1) 動作周波数をできるだけ下げる
- (2) 電源電圧をできるだけ低くする

というアイデアを、ハードウェア的にどう実装するかということです。この少ない持ち駒をどう活用するかでマイクロコントローラの低消費電力の達成度が決定します。

特に電源電圧は2乗で電力に効いてきますので、電

源電圧を下げることは効果的です。しかし、電源電圧を可変できる範囲は、製造する半導体プロセスで制限されてしまいますし、電源供給を行うレギュレータの出力を可変するにはコストがかかります。このため、比較的簡単な、動作周波数を下げるという手法が最初に採られます。

▶マイコンで採用されている電源電圧を下げる方法

動的に電源電圧や動作周波数を変化させて消費電力の最適化を図るDVFS(Dynamic Voltage and Frequency Scaling)という方法があります。これは処理内容に応じて電源電圧や動作周波数を動的に変化させる方法ですが、主たる操作は電源電圧の制御です。まっとうなDVFSでは数種類の電源電圧が可能なのですが、マイコンに採用する場合は「通常」と「低電力」の2種類のサポートが多いようです。

方針1：動作周波数を下げる

●CPUコアの動作周波数を下げる

これは文字どおり回路に供給するクロックの動作周波数を下げることを意味します。しかし、通信デバイスなどは規格で決められたクロックで動作しているため、動作周波数を変更することが難しいのが実情です。

一番簡単なのはCPUコアの動作周波数を下げることです。CPUコアは各周辺デバイスに指令を出すだけですから、動作周波数が遅くなくても「指令」の発行が遅くなるだけです。

●使っていない周辺モジュールのクロックを停止する

動作周波数を下げることと同等の効果があるのがクロックを停止することです。CPUを含む各周辺モジュールに対して、動作するときのみクロック供給を行う手法です。このクロック停止技術をクロック・ゲーティング(Clock Gating)といいます。

クロック・ゲーティングは周辺モジュールごとにソフトウェアでクロック供給を制御する方法もあります