

もう一度重要になる気がする プロセッサ開発のセンス

第1回 今どきのプロセッサ開発に求められること
新連載

杉本 英樹

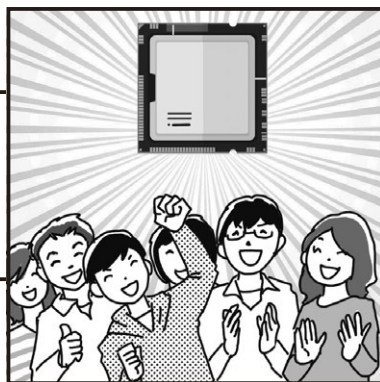


表1 プロセッサで使われる並列化方式

方式	粒度	動的処理対応	対称性	並列度	実装例	組み込み適正
データ並列	大	効率低下△	高のみ 低いと極端に性能効率劣化	中	SIMD/GPGPU	△/○
グラフ並列	中～小	細粒度は○ 粗粒度は△	自由 効率がほぼ変わらない	中～大	これから	○/◎
処理並列(粗)	大	○	中～高 低だと効率が低下	大	Many Core	×
処理並列(中)	中	○(ややオーバヘッド大)	自由	中	MIMD	○
処理並列(細)	小	効率低下△	自由	小	VLIW	○

スマートフォンやPCのようなコンピュータを直接意識する機器だけではなく、自動車やテレビ、さらにはエアコンなどの白物家電まで、現在ではさまざまな機器にプロセッサが使われています。

これだけ多く使われているプロセッサについて、ソフトウェアを開発している方はもちろん、機器を利用している方でもある程度その役割というのはイメージしていると思います。しかし、それがどのように作られているかとなると、知らない方も多いのではないのでしょうか。

これから複数回に渡って、プロセッサ開発経験者がリレー形式で、開発の現場とその強みを本音でお伝えしたいと思います。開発に興味がある方はもちろん、そうでない方にも開発の一面を知っていただくことで、日ごろの開発のヒントになることがあるのではないかと期待しています。

現在のプロセッサ技術

● 複雑で多種多様

プロセッサと一言で言っても、多種多様な種類が存在します。近年のプロセッサの開発競争は、いかに処理の並列度を上げるかの勝負となっており、表1のようにさまざまな技術を利用しています。さらに、これらが組み合わせられて使われています。

プロセッサは最初からこんなに複雑だったわけではありません。複雑化の背景には大きく2つの要因がありました。第一には、LSIに使われる半導体製造プロセス・テクノロジーの変化による集積度の向上、第二にはLSIの設計技術による複雑化です。

● 多くの素子と並列処理が必要

以前の半導体製造プロセスでは、一世代進むと平均ゲート遅延(LSI上の1つの論理素子がスイッチしてから次の素子にそれが伝搬するまでの時間)は20～30%短くなり、電力も40～50%削減されるのがトレンドでした。

ところが近年では、微細化によるさまざまな制約が発生しています。一世代進んでも、電力こそやや減少するものの、平均ゲート遅延はあまり短くなくなりました。一方で、同じ面積に搭載可能な素子(トランジスタ)数は一貫して加速度的に増加しています(図1)。これがいわゆる「ムーアの法則」と呼ばれるものです。

結果として、プロセッサの性能を向上させるには、

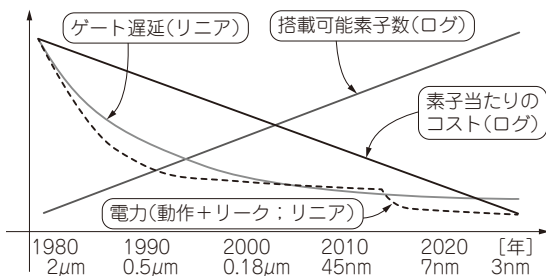


図1 搭載可能な素子数は加速度的に増加している