「動かないコンピュータ」とアーキテクチャのスパイラル

マクロ的計算機構成論

コンピュータの誕生からいまに至るまでの、華々しい技術革新の裏側で、時のアーキテクトたちが総力を挙げて開発したにもかかわらず、ついに完成することのなかった、あるいは商売にならない性能ゆえに消えていった「動かないコンピュータ」が数知れず存在する。どうして「動かないコンピュータ」になってしまったのだろうか？　その事例を集めてみると、どうやら、コンピュータアーキテクチャのトレンド周期との関係が浮かび上がってくる。

かくて手法は繰り返す

経済やファッションにも似て、コンピュータのアーキテクチャにも技術トレンドの繰り返しがある。

基調として、コンピュータの回路を乗せるデバイスの技術は進化し、真空管からトランジスタ、IC、LSIへ、さらにLSIも微細化へと、回路は集積の度合を高めてきた。そして微細化と集積が十分に進むと、コンピュータは大きく姿を変えて次のステージに進む。

すなわち、最初はトランジスタや抵抗を配線して箱の中に詰め込んでいたが、次にはICやLSIを取り付けたボードを組み合わせる時代になり、そのうちに機能も性能も小さいマイクロプロセッサが登場し、それも性能と機能が成熟すると、プロセッサに多数の計算コアを詰め込むようになった。

さて、このようなアーキテクチャの変化のあいだにも、デバイスの微細化は着々と進む。微細化するということは、配線が短くなって動作が速くなり、一定面積にたくさんの回路を詰め込めるということである。プロセッサアーキテクチャは、豊かになったデバイスを使用して高速化のためのロジックを組み込むようになり、最初単純だったものが、だんだん複雑になっていく。

まずは、キャッシュ、メモリコントローラ、パイプラインといった機構が付け加えられる。これは、箱、ボード、ワンチッププロセッサのどの時代も同じだ。さらにデバイス技術が爛熟すると、高級言語やアプリケーションに特化したプロセッサまで作られるようになる。すると、そのうちに汎用なアーキテクチャでよいのではないかと考えられるようになり、そのころには十分に成熟したデバイス技術でコンパクトで単純なプロセッサを集積した、新しい形のコンピュータが現れるのである。

これで一巡だ。繰り返しとはいっても、デバイスの特性や利用形態に合わせてアーキテクチャも少しずつ変わり、次第に洗練されるので、技術はスパイラル状に進化している。

例を挙げよう。1990年代には、複雑な動作の命令を用意して、1回の命令でたくさんの仕事をさせるのがいいのか（CISC）、あるいは高速だが単純な動作の命令を用意して、それを組み合わせて実行するほうがいいのか（RISC）という論争があった。いまは、ワンチップに多数の計算コアを搭載するメニーコアプロセッサの流れがあり、計算コア単体を単純にするか、もっと複雑にするかという議論がある。それは、同じ種類の問題だ。なお、この問題の決着はついていない。

かつて、LISPやSYMBOLを高速化するコンピュータが作られたが、時代を下るとJavaを高速化するプロセッサが作られた。いまはまた、グラフィックス計算を高速化するGPUやFPGA内蔵のプロセッサが登場しているではないか!

アーキテクチャの失敗学

「動かないコンピュータ」は、このようなトレンドの変わり目で、新しいテクノロジーを編みだす挑戦的な開発、あるいは成熟期を過ぎたテクノロジーを工夫でなんとか持ちこたえさせようとしたときに、往々にして生じる。

たとえば箱の時代のコンピュータ、IBM Stretchは、パイプライン処理、命令プリフェッチなどの新しい仕組みの数々を実装し、計算機史上のマイルストーンとして位置付けられているが、性能が目標の半分にとどまり、それに比例して予価の半額で販売された。2台目以降は、販売されなかった。

初期の並列マシンETA10は、8プロセッサを搭載し、液体窒素で冷却しながらCMOS回路を高クロックで動作させた。しかし、ついに並列で動作することはなかった。ハードウェアとソフトウェアを擦り合わせて並列動作させる経験がなく、冷却の問題も最後まで尾を引いたのだろう。

時代が下ったIntelのItanium2は、ソフトウェアとの協業によって性能を出そうとしたが、開発が遅れているあいだにハードウェアの進歩が上回り、それを補うために設計が大規模化してさらに遅れ、結局、市場でポジションを得られないまま退出していった。

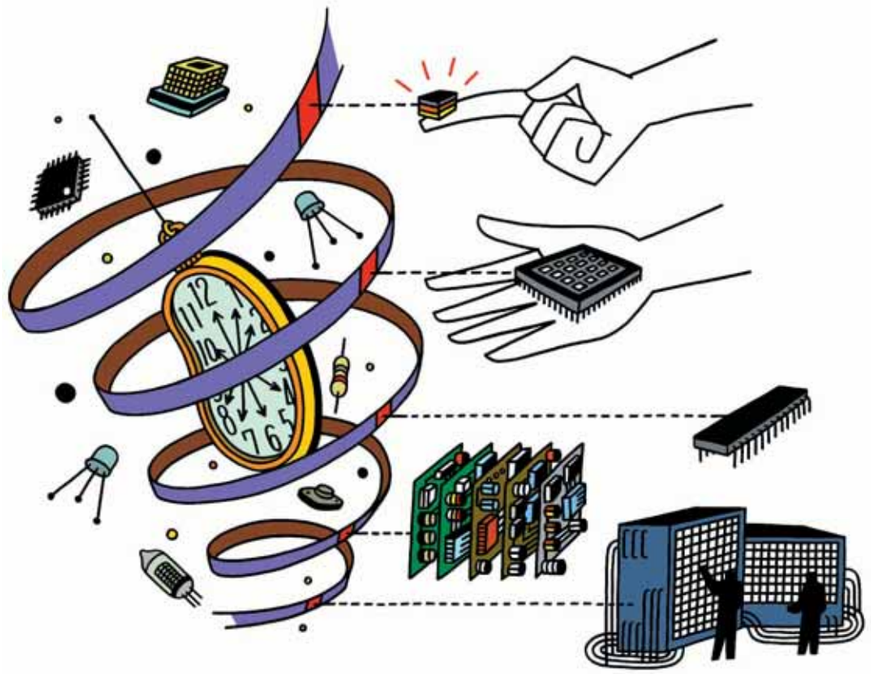
同様にして、共有メモリ、トランザクショナルメモリというように、新しい手法を導入するたびにコンピュータの墓が立ってきた。

逆に、よく知られた手法に改良を加えていけば、比較的手堅く「動くコンピュータ」を作れる。1970年代、商業的に大成功をおさめたCRAY-1は、従来の論理回路を踏襲し、冷却装置だけを刷新したものだ。

しかし、ずっと同じ路線を続ければ、大きな性能の飛躍を望めず、開発力も落ちていく。それは、動く設計と動かない設計の微妙な関係であり、設計力とは斬新な手法と手堅い手法のトレードオフを量り、選びとって組み合わせる力なのである。実際、前述のような失敗のあとには、その問題点を乗り越えたコンピュータが現れ、現在のコンピュータにつながっている。

そのようなわけで、動かないコンピュータもまったく無意味とはいえないが、やはり挑戦しつつも「動くコンピュータ」を作りたい。ここで2つのことがいえるだろう。

まず、理想を求めすぎて何もかもいっぺんに新しい技術を導入しないことだ。論文から新しい概念の粋を集めたような設計は、成功しない。もうひとつは、後退できる設計、すなわち構成要素の一部が当初の目標を達せられず、多少の犠牲をはらうことになったとしても、一定の成果を出せるようにしておくことだ。いくつかの不調がありながら帰還した惑星探査機「はやぶさ」と同様のことは、コンピュータの開発にもそのまま当てはまる。



1950年代から1960年代にかけて開発されたEDSAC、IBM Stretch、CDC 6600などのコンピュータは、真空管やトランジスタ、抵抗などの部品を配線して、箱の中に詰め込んでいた。1960年代なかばには、IBM 360のようなメインフレームコンピュータが、ICやLSIを取り付けたボードを組み合わせて製造されるようになった。1970年代にはいると、機能や性能は小さいながらシングルチップのプロセッサIntel 4004が登場し、以降は規模と性能が急進していく。1999年に発表されたIBM Power4では、プロセッサに2つの計算コアが搭載され、現在の商用プロセッサIntel Xeon Phiには、60個の計算コアが詰め込まれている。