

5.1 ■ 情報処理技術 — 過去十年そして今後の十年 —

計算機アーキテクチャの過去、 現在と未来

富田 眞治 京都大学情報学研究科

1KビットのDRAM, 最初のマイクロプロセッサIntel4004が1970年代初頭に世に出て約30年経過した。4004は4ビットプロセッサであり, 集積トランジスタ数2,300個, 動作周波数750KHz, 16ピン実装であった。この時点で誰が今日の高性能プロセッサの出現を予測し得たであろうか。4ビットではあるが, 1つのプロセッサが1チップに搭載され, 実用化され, オンチッププロセッサがスタートをきったのである。今日, たとえば, DEC Alpha21264では1,520万個のトランジスタを集積し, 600MHz, 587ピン実装となっている。この発展の原動力は半導体集積回路技術の飛躍的な発展であるのはもちろんであるが, 計算機アーキテクチャ, コンパイラ, OSなどの果たした役割もきわめて大きい。本稿では, 主として1990年代の計算機アーキテクチャの変遷を辿り, 次の10年を展望しようと思う。

■ 1980年代: RISCの時代

1970年代のVLSIの揺籃期を経て, 1980年代に入ると, チップ上の集積度はそれほど大きくはないが, 高機能プロセッサを1チップに集積できる時代になった。高級言語の普及を背景として, コンパイラとの協調設計が図られ, RISCの考えがワークステーションなど過去のプログラム資産の少ない分野で普及した。その基本的な考えは

- ①高級言語プログラムはコンパイラで機械命令プログラムに変換されるので, コンパイラで利用できる単純でしかも高速実行できる機械命令セットで十分である。
- ②CISCでは機械命令実行のための制御記憶が必要であり, 非常に大きな面積を必要とする。RISCではこれが必要ではないので, メモリ管理装置や浮動小数点演算装置にそのチップ面積を割くことができ, 高速化できる。
- ③RISCは単純であり, 生産サイクルが短いので, 最新の技術を常に新しい製品に反映できる。

上記②, ③はチップ上での集積度が飛躍的に向上した今日では, 当てはまらないが, RISCの洗練されたパイプライン構造とメモリ階層構造の設計法, コンパイラとの協調設計の重要性の指摘などは今日のプロセッサに非常に大きいインパクトを与えた。

■ 1990年代: 命令レベル並列処理の時代

最も単純なコンピュータでは, 機械命令は1つずつメモリから取り出され (IF), 解読され (D), オペランドアドレスが計算され (OA), オペランドが取り出され (OF), 演算実行され (EX), 結果が格納される (S)。次の命令アドレスの計算がこれらの実行過程の途中で計算され, 機械命令の実行が終了すると, 上記過程が次命令に対して繰り返される。

1990年代のコンピュータの高速化は, 一言でいえば機械命令レベル並列処理の原理に基づいている¹⁾。

これをアーキテクチャの側面から基本概念を明確にしつつ見てみると次のような原理を採用しつつ発展してきたといえる。

①時間並列性：これは工場の流れ作業に対応した考えであり、命令パイプライン方式と呼ばれている。すなわち、上記の命令実行過程を複数の命令でオーバーラップするものである。ステージ（上記のIFからSに対応）数をLとすると、理想的には、L倍高速化できる。しかし、先行する命令と後続する命令にはデータ依存関係、制御依存関係、資源の競合があり、理想の場合のようにパイプライン上を命令は流れない。そこで次のようないろいろな方法が提案され、実用化された。

②乱実行：先行命令が終了しなくても後続命令がそれと無関係でしかも実行時間が短ければ先に実行を終了させる方式が乱（Out-of-Order）実行である。このようにすると先行命令と後続命令が並列に動作でき高速化できる。

③分岐予測：分岐命令には偏りがあることが知られている。この偏りはエラー発生チェックのための分岐命令のように、静的に定まっている（エラーは通常起こらないので）ものもあるが、実行時に変動するものも多数存在する。したがって、分岐命令がTRUE（T）となるか、FALSE（F）となるか実行時に予測する方式が一般的にはとられるようになった。その分岐命令の過去数回の実行でのT/F回数やパターン系列情報、あるいはその分岐命令に隣接した他の分岐命令の履歴の加味など、多様な方式が提案され、実際のマイクロプロセッサにも用いられている。分岐予測のヒット率は応用や方式にもよるが、90～95%といわれている。

④制御投機実行：この分岐予測をさらに進めたのが、制御投機実行方式である。分岐予測した側の命令列を

どんどん実行してしまう方式であり、予測が成功すれば「めでたし」である。しかし、予測に失敗すると、レジスタやメモリの状態を元の状態に復帰させるという手痛い打撃（ペナルティ）を受けることになる。リオーダーバッファなどの復旧手法がとられているが、ハードウェア的には連想メモリを用いるので、複雑なものとなり、速度が低下する要因となる。

⑤局所空間並列性：以上に述べた方式を導入しても単一の命令パイプラインでは最大L倍の性能向上しか望めない。したがって、プログラムカウンタで指されるアドレスの近傍（局所）の命令をM個ずつ同時にフェッチし、パイプラインを太くして（空間並列）、多数の命令を並列処理する方式が採用された。時間並列と組み合わせるとM*L倍高速化される。スーパースカラ方式とVLIW（Very Long Instruction Word）方式がある。前者では通常の機械命令を同時にたとえば4命令同時フェッチして、命令間の依存性の解析を実行時に行う方式であり、現在の汎用プロセッサの多くに取り入れられている。これは目的プログラムのレベルでの互換性が保たれるためである。しかし、ハードウェアは多重度が大きくなるにしたがってきわめて複雑になる。一方VLIW方式はコンパイラが同時実行可能な命令をプログラム中から収集して、きわめて長い命令の各専用制御フィールドに埋め込む方式である。実行されるVLIW命令は同時実行可能性が完全に保証されているので、依存性の解析などを実行時に行う必要はなく、ハードウェアは単純になり、動作周波数を大きくできる特長がある。しかし、互換性やNOP操作の増大などの欠点があり、これまではメディアプロセッサやスーパーコンピュータのスカラプロセッサなど、プログラム自身に並列性があり、しかも専

用ハードウェア化するには標準化などの動向をみて柔構造化する必要のある分野に利用されてきた。

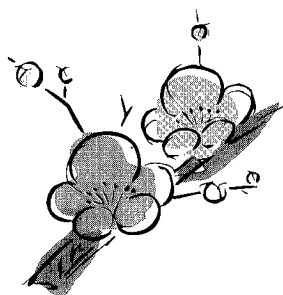
■2000年代：多様化の時代

2010年にはチップ上には10億個のトランジスタが集積され、周波数10GHzで動作可能な時代となるといわれている。これまで計算機設計者（アーキテクト）はいろいろな物理的な制約に苦しめられながら、システムの設計を行ってきたわけであるが、2010年頃には「何でもあり」の時代となるだろうといわれている²⁾。

では、2000年代のアーキテクチャ研究は薔薇色かといわれると、むしろ現在は集積回路・デバイス技術者と応用技術者の狭間にあって、ブレークスルーを模索しており、暗い閉塞感があるのが実状である。その閉塞感の諸要因としては、汎用高性能プロセッサ分野でのIntel社の絶対的支配、アーキテクチャ研究の飽和域への到達感、デバイス技術の突き上げとロードマップ指向、要素技術を軽視するメーカの製品開発ポリシーの変化などがある。

今後、ネットワークの発展、モバイル環境の整備などを背景として、高性能大規模サーバ（科学技術計算、データベース、マルチメディア（画像／CG）のためのスーパーコンピュータ）を中心としたハイエンドコンピュータ系、パソコンやワークステーションなどのデスクトップコンピュータ系、ゲーム機やパイル機器を中心としたローエンドコンピュータ系に、コンシューマシステムは多様に分化することは間違いない³⁾。





■ハイエンド/デスクトップ系

ハイエンド系は並列処理が普及し、その要素プロセッサは通常のマイクロプロセッサとなろう（スーパーコンピュータではベクトルプロセッサは衰退）。ハイエンド系の要素プロセッサとデスクトップ系では1990年代の技術のさらなる発展が次のようになされようが、主要技術はメモリとプロセッサ速度の乖離をいかに埋めるのかが焦点となろう。

①VLIW方式の高速化と汎用化：先に述べたようにスーパースカラではハードウェアが複雑であり、性能向上が限界に達しようとしている。一方、純粋のVLIW方式では、長命令内でのNOPが多く効率的ではなく、また互換性の面でも問題があった。Intel社が2000年に、Pentiumシリーズの後にそれらとまったく異なる命令セット（IA-64）を有するプロセッサMercedを発表し、販売を開始する。MercedではEPIC（Explicitly Parallel Instruction Computing）が標榜されており、VLIW方式をベースにしつつ、スーパースカラ方式の特長も取り入れ、命令長の圧縮、NOPの削除、互換性や拡張性の確保を狙っている。Mercedの大きな特長の1つである、プレディケート演算では、比較操作命令の結果に基づいて演算実行の正常終了や取消しの制御ができる。性能低下の要因となる分岐命令の削除や静的分岐予測に威力を発揮しよう。

②超スーパースカラ方式：現在のスーパースカラ方式の多重度は4程度

となっている。この多重度をたとえば16くらいに増やす方式が超スーパースカラ方式である。リオーダーバッファなどの構造がきわめて複雑になるとともに、TRUEの続く分岐命令では、分岐予測がヒットしても命令フェッチが複数のブロックに跨るため時間がかかり、性能向上ができない。このため、分岐パス上の命令をキャッシュメモリの連続アドレスに再配置するトレースキャッシュが提案されている。32命令多重までいけるという強気の研究グループもある。

③大域空間並列性の利用

- 空間並列マルチスレッド：この方式は比較的規模の小さなスーパースカラによるパイプラインを複数台用意し、プログラム内の制御依存のない一連の機械命令列（スレッド）を並列に実行させる方式である。これは命令レベルよりももう少し単位の大きいスレッドのレベルでの並列性の抽出で高速化が達成できる可能性があるからである。多重度の大きなスーパースカラより50%程度性能が向上するといわれている。DEC社の次期プロセッサで採用されるといわれている。

- オンチップマルチプロセッサ：スレッドをプロセスまで大きくした方式である。スヌープキャッシュ（キャッシュを分散配置して、コヒーレンス制御を行う方式）、共有キャッシュ（キャッシュメモリを集中管理し、共有する方式）、レジスタを共有する方式などが検討されている。

上記のいずれの方式も複数のスレッドやプロセスを実行できるためのキャッシュメモリの構成がカギを握っており、チップ面積がまだ（？）小さい間はきわめて困難である。

④データ投機実行と再利用：先行命令の結果を後続命令が利用するとき、後続命令は先行命令の終了を待たねばならない。しかし、先行命令の結果を予測できる場合がある。たとえば配列データの行方向へのアクセスの際にはアドレスは一定のストライドで加算される。この方式では先行命令と後続命令を同時に実行させ、先行命令の結果が予測通りであれば高速化できる。

データ再利用は先行命令の実行に際してその環境を調べ、その環境が前回と同様であれば先行命令自体の実行を省略する方法である。

これらの方式では多くのテーブル類や連想メモリが必要とされる。また、性能向上も10数%程度であろう。

⑤レジスタレスコンピュータ：今日のプロセッサではレジスタが32～128個も装備されている。JAVAはスタックコンピュータであるのでレジスタはない。また、スーパースカラ方式などではレジスタのリネームなどで連想メモリが必要であるので、性能が低下する要因となる。過去のデータフロー方式を踏まえたレジスタのないコンピュータも考えられる。

■ローエンド系

これまでアーキテクトは先に述べた超並列処理などきわめてハイエンドなシステムを追求してきた。そこにのみ知的な好奇心を満たす価値観があったといえてよい。しかし、携帯電話、カーナビ、モバイル機器などのカバーを開けてみると、実装技術を中心とした、学会の論文誌には投稿されない、信じられないような技術が存在する。このようなローエン

ド機器での中心技術は省電力化、超軽量化、高信頼性⁴⁾と超自動化とであろうと思われる。電卓並みの軽さと消費電力で、画面とキーボードがパソコン並みで、ポケットに丸めて入れられて、形状記憶合金のようにポケットから外に出すとしゃきっとして利用できるコンピュータが実現できれば理想である。このような「面白い」研究を推進するためには、これまでのアーキテクチャ研究に対する価値観を転換し、ローエンド機器研究に対する業績評価についての見直しが必要である。大学などでの論文至上主義はまず捨て去らねばならない。ローエンド系でのアーキテクチャ関連の技術には下記のようなものがある。

①DRAMとロジック混載：DRAMの構造は2次元格子状に配置されているので、同一行のデータへの同時アクセスが可能である。したがって、DRAM中にプロセッサなどの論理回路を混載すると、高いバンド幅、低レイテンシが実現でき、外付け回路が少なくなり、電力消費が小さくなる（三菱電機の32R/Dでは、電力は1/3程度になっている）。画像処理、グラフィックス、数値計算など定型的な処理（連続アドレスに対するデータアクセス処理）に適用できる。しかし、DRAMとロジックの製造プロセスが異なるので、コスト高になる欠点がある。

②進化するコンピュータ⁵⁾：FPGA(Field Programmable Gate Array)ではその論理構造が変更できるので、たとえば実行頻度の高い演算などについては並列化などで高速化したりできると期待されている。1970年代には制御記憶装置を問題ごとに書き換えできる可変構造型コンピュータが提案されたが、失敗に終わった。進化したコンピュータの保守やプログラム移植が困難であったことが大きな理由であろう。FPGAではどうなるか興味深いものがある。

③ソフト／ハード協調設計：マルチメディアの時代となって、画像処理やグラフィックスが身近なものとなった。画像処理では画像の符号化・復号化がMPEG2などの方式で標準化されている（画像圧縮率は1/20程度）。MPEG2はコサイン変換、動き予測などの処理の負荷が重く、符号化に100GOPS、復号化に10GOPS（OPS: Operations Per Second）程度の性能が必要とされる。家電製品として多く使用されるので、消費電力も小さく0.5～1.5W程度である必要がある。

このようなマルチメディア処理では標準化が目まぐるしく変わっていくという困った一面がある。VLIW方式のプロセッサやDSPなどの汎用性のある機能装置で構成するのか、かつて注目されたシストリックアレイなどの専用装置で構成するのか、またはそれらの機能分担をどのようにするのか、が問題となる。性能、消費電力、コストなどのトレードオフを解決できるソフト／ハード協調設計などの優れたCADツールの実用化が望まれる。また、スーパーコンピュータなどのハイエンド系で得られてきた各種技術のゲーム機などのローエンド系への技術移転も重要となろう。

■歴史に残るアイデアを

汎用的なコンピュータでの最も大きなブレークスルーはIBM360の開発、XeroxのAltoに代表されるワークステーションとEthernet開発、ソフトウェア危機を反映した高級言語計算機のアンチテーゼとしてVLSIの発展を背景にして提案されたRISCプロセッサの開発、スーパースカラ・VLIWなどの機械命令レベル並列処理などがある。いずれも非常に大きなインパクトをコンピュータ技術さらには社会変革に与えて

きた。

今日のプロセッサ技術の中身を見ても、多くの革新的なアイデアは1970年代（JAVAで利用されるスタックコンピュータ、RISCや乱実行などは1960年代）から1980年代に研究されている。数年の短いタイムスパンでの研究開発も重要ではあるが、10年、20年後のアーキテクチャ研究を地道に進める必要もある。

現在のシリコンVLSIには物理的限界がくるのは明確であり、量子コンピュータなどの新しい方式が検討されつつある。量子コンピュータは物質の持つ波の性質を利用したコンピュータであり、基本的なアルゴリズムや基本回路の提案などがなされている。現状では実用性はまったく未知数であるが、夢のある成果を期待したい⁶⁾。

コンピュータアーキテクトには、コンピュータ工学関連分野の広い知識のみならず、世界の政治経済構造や社会生活様式の変化への敏感な嗅覚を持ち、自らも含めたユーザのニーズを把握し、質的な技術的飛躍を可能とする独創的な発想力を持つことが強く求められよう。

参考文献

- 1) 富田眞治: コンピュータアーキテクチャI, 丸善 (1996).
- 2) Goodman, J. R. et al.: Billion Transistor Architectures, IEEE Computer, pp.46-49 (Sep. 1997).
- 3) 富田眞治: いま欲しいブレイクスルーアーキテクチャ, bit, 共立出版, Vol.31, No.3, pp.11-13 (1999).
- 4) Hennessy, J.: The Future of Systems Research, IEEE Computer, pp.27-33 (Aug. 1999).
- 5) 末吉, 稲吉編: やわらかいVLSI, 情報の情報処理, Vol.40, No.8, pp.775-801 (Aug. 1999).
- 6) 西野哲朗: 量子コンピュータ入門, 東京理科大学出版局 (1997).

(平成12年3月13日現在)

