

II コンピュータアーキテクチャの動向と展望†

元 岡 達††

1. はじめに

計算機のユーザがプログラムを書く際に知っておく必要のある知識として、計算機の命令セットの定義がある。これがソフトウェアとハードウェアのインタフェースとして重視されるようになり、コンピュータアーキテクチャと呼ばれるようになった。これが狭義のアーキテクチャの定義である。一方、コンピュータ技術は急速な発展をとげしかも複雑化し高度化を続けたため、その全ぼうを一人で把握することが困難になり、ソフトウェアとハードウェアのトレードオフについても評価が困難になってきた。このためアーキテクチャをきめることが一つの独立した技術として重視されるようになってきた。

コンピュータに関連した技術の中で、ニーズ側とシーズ側の要求の間に立って妥協をはかり、適当なインタフェースをきめる必要のある局面は随所に現われる。最近ではアーキテクチャという用語がこのような場合に広く用いられるようになった。ネットワークアーキテクチャ（以後網アーキテクチャという）、ソフトウェアアーキテクチャなどがその代表例である。

ここではコンピュータ技術をやや長期的な観点から展望する立場をとる都合上、アーキテクチャを必ずしも狭義の意味に限定せずに、コンピュータ技術を考える上に必要な各種のインタフェースをとりあげ今後の問題点についてふれてみることにする。

第三世代のコンピュータ技術として、小・中形から大形のコンピュータに共通のアーキテクチャを用いるファミリー化の概念が提唱された。ソフトウェア開発の困難さ、ソフトウェア資産の蓄積といった問題を解決する手段として、アーキテクチャの固定化、標準化の必要性が叫ばれ、一般に広く受入れられてきた。しかしこれはメインフレームと呼ばれる汎用コンピュータ

の分野であり、その後に生れたミニコンピュータやマイクロプロセッサの分野では独自のアーキテクチャが採用されている。

少し長期的視野に立ったとき新しいアーキテクチャの採用が必要となりまた可能となってくると考えられる理由がいくつかある。これを列挙すると次のようになる。

(1) LSI 技術の発展によって、ハードウェア価格が大幅に低下して、大容量の高速記憶が手軽に使えるようになると共に、多くの機能を論理回路で実現し利用できる可能性が生れてきた。

(2) 大量生産が価格低下に大きく貢献すると言った LSI 技術の特性を、コンピュータアーキテクチャの面からも活用することを考える必要がある。

(3) 今後のコンピュータの応用分野としては事務計算、数値計算といった分野だけでなく、非数値データを含む多種類のデータの蓄積、処理、交信を考える必要があり、これらの用途に現在のコンピュータアーキテクチャは不向きである。

(4) ソフトウェアの記述が高級言語で記述される傾向にあり、これら高級言語レベルでの移植性が確保できれば、ソフトウェア資産の活用は可能であって、命令セットレベルでのアーキテクチャの統一は余り重視されなくなった。

(5) ファームウェア技術等の利用により、従来のアーキテクチャをエミュレータ等の形式で包含させることは容易であり、従来アーキテクチャとの両立性を保証することが可能である。

以上の立場から、本論文ではコンピュータの新しいアーキテクチャに焦点をあて、ニーズやシーズを生みだす周辺の環境や技術とアーキテクチャの関係について考察し、今後の動向を展望する。

2. コンピュータに対する期待

社会の情報化が今後急速に進むことを希望し、これを推進しようと考えている人々がコンピュータに期待

† Recent Trends and Future Views of Computer Architectures by Tohru MOTO-OKA (Faculty of Engineering, University of Tokyo).

†† 東京大学工学部電気工学科

している機能や性能を実現するためには、どのような言語を使って実用システムを記述するのがよいか、その言語の持つ機能をソフトウェアとハードウェアでどう分担するのがよいか、このようなアーキテクチャの問題を検討する前段階として社会からのばく然とした期待をアーキテクチャに反映できる形の要求に書きかえる必要がある。

換言すれば、応用分野の多様化や拡大、利用形態の変化をどのようにとらえ、これを実現するためにシステムがそなえるべき機能や性能を明らかにすることである。以下にこのような立場からみた非常にグローバルな要求仕様の事項を列挙してみる。

(1) 偏微分方程式の計算など、高次になると今日の超大形計算機では実用上解を求めることのできない多くの数値計算が科学・技術分野にある。価格さえ安ければその需要は膨大なものになる。各種の実験に代るシミュレータとしての用途もこの範疇に含めて考えることが出来る。

(2) 各種図形や文章の処理など非数値データの処理に計算機を使おうとすると、その性能の低さに驚かされることが多い。人工知能に関連した研究が基礎研究の域を脱することの出来ぬ理由の一つとして多くの人々から指摘されている事実である。物体・画像・図形・音声・信号・文章といった各種のパターン処理は計算機の新しい応用分野としても、マン・マシンのインタフェースとしても重要である。

(3) ソフトウェアの記述に高級言語を用い、非手続き言語・関数形言語など既存のアーキテクチャとは独立した言語を採用することによってソフトウェアの移植性確保、書きやすさの追求、新しい機能の導入などが行われている。言語仕様とアーキテクチャの間の直接的なトレードオフを考える機運になっている。

(4) 情報システムを一般的な立場から考察したとき情報の蓄積管理が処理以上に重視されてきたことに気付く。これまでの計算機では情報処理に重点がおかれてきたが、情報システム全般の発展を考え、より知的なシステムの実現を考えると、蓄積・処理・交信についてバランスのとれたアーキテクチャの実現が望ましい。ここで交信といったのはマン・マシンインタフェースの問題を含めた広義の通信である。情報の蓄積といっても死蔵していたのでは意味がなく、必要に応じて高速にアクセスできる知的なデータベースとなっている必要がある。このような特性を持った大容量で安価な記憶システムが必要である。

(5) いつでもどこでも使えるためには、端末が手元にあり、これを介して希望の資源（データ、ハードウェア、ソフトウェア）に容易にアクセスでき、利用できる必要がある。この実現を目指したものが計算機網である。最新で正確なデータを大量に共用するために、このような情報システムは巨大化する傾向にあり、全世界的な情報システムの実現が指向されている。

(6) これまでの計算機の欠点の一つとして人間の感覚器官や運動機能に相当する入出力機能の不備が指摘できる。人間にとって親しみやすいシステム実現のためにも、画像・音声等各種のデータを自由に使いこなすためにもマン・マシンインタフェースの充実は非常に重要な課題である。

(7) パーソナルコンピュータの発展は、LSI技術の進展から予想される分野であり、個人で自由に使えるシステムの魅力は大きい。インテリジェント端末との差異は通信回線の使用頻度の差程度であるが、データベースなどの資源共存を重視するか、個人的な融通性を重視するか、考え方の上での差異は大きい。

(8) 使いやすいシステムにすることは情報化社会を推進し、ユーザにとって親しみやすい身近かなものにするために重要な要素である。このことには性能・価格・信頼性といった基本的な課題からマン・マシンインタフェース、使用言語さらには知識をそなえたシステムにすることにいたるまでの多くの研究課題が含まれている。

連想機能・推論機能・学習機能など人工知能の研究で得られた知見を情報システムに取入れる努力も必要である。

(9) 信頼性が高く、保守の容易なシステムにすることは、情報システムの大幅な普及のための大きな前提条件である。システムが巨大化すればするほどその中枢部にはより高い信頼性が要求される。端末機器など大量に分散配置されるシステム要素については定期保守を不要にする必要がある。情報システムの無人運転なども必要になってくる。

(10) 巨大な情報システムを安心して使いこなせるようにするためには理解しやすいシステムをいろいろなレベル、いろいろな側面からみて実現する必要がある。今日仮想化の考え方が広く受け入れられ、いろいろなところに導入されているのも見通しのよいシステム実現に対する要求からと考えることができる。ソフトウェアの作りやすいシステムを実現する上にも理解しやすいシステムである必要がある。

3. コンピュータ技術をささえる周辺技術

第三世代までのコンピュータ技術の進歩は主として素子技術の進歩に支えられて発展してきた。逆説的に言えば、ハードウェア技術はユーザ側のニーズを満たすにはほど遠く、シーズ技術を十分に活用して性能・価格比の優れたシステムを提供することに重点がおかれ、使いやすさの問題はほとんどすべてソフトウェア技術にまかされてきた。

集積回路技術の発展と、ソフトウェア危機とによって、ハードウェアとソフトウェアとの間のトレードオフすなわちアーキテクチャ技術が現実の問題となってきたのは第三・五世代以後と言ってよからう。しかしこのことはシーズ技術の重要性が減少したことを意味するのではない。シーズ技術が単なる素子の速度向上と言った一面的な係わり合いではなく、アーキテクチャやシステム構成により直接的な影響を持つようになったと言うことができよう。

今後のコンピュータ技術、特にアーキテクチャに最大のインパクトを与える要素はやはり LSI あるいは VLSI と呼ばれるシーズ技術であろう。ニーズの検討が重視されるようになってきたのは LSI などのシーズ技術の発展によって適用分野、利用形態などの可能性が大幅にひろがり、選択の幅が広がったからにはほかならない。

LSI 技術のインパクトの最大のものは主記憶の LSI 化にみられるように、大幅な価格の低下と、小形化にあると言える。高密度化と大量生産による LSI の価格低下は3年で 1/2 という驚異的な速度であり、LSI 記憶の導入によってソフトウェアなどの評価基準にも大きな影響が生まれている。

論理装置に対する LSI のインパクトはマイクロプロセッサに見られるように、既に非常に大きな影響をコンピュータのみならず、多くの産業分野に与えていることは説明するまでもないことと思う。しかしプロセッサへの本格的な LSI の導入は今後の技術進歩にゆだねられており、これについては次章で述べる。

しかし、高速論理素子の追求の重要性が薄れたわけではない。困難さが増し、実用化研究に更に多くの年月を要するものとするが、化合物半導体素子であるとかジョセフソン素子のように実用化されれば現在の素子に比して1桁以上の高速化が期待されている素子がある。

周辺記憶は磁性面記憶がなお性能の向上を続けてお

り、当面実用上の大きな障害になる要素はない。磁気バブル技術も記憶の階層構成の中ですぐさま不動の地位を確保するのはむずかしいと思う。10¹³ ビットといったデータベースが当面必要としている記憶容量を提供する技術は既に確立しているが、画像データなど非コード化データの記憶には更に大きな記憶容量が必要であり、新しい技術の確立が必要である。

入出力装置の電子化は高速化や低騒音化に役立っただけでなく、無保守機器を実現して将来の情報処理技術の広範な普及に非常に大きなインパクトを与えた。日本語処理を例にとっても従来最大の隘路の一つであった漢字プリンタが、ROM と近年のプリンタ技術の組合せによって安価に提供される見通しになっている。

通信技術と計算機技術の融合が盛んになったことも情報システムの構成という広義のコンピュータアーキテクチャに大きなインパクトを与えつつある。計算機網による分散データ処理システムの普及をまねいており後章で述べる。衛星通信技術のインパクトも大きいがここでは取上げない。

光通信技術もデータの高速度伝送に役立ち、通信回線としてだけでなく、近距離のデータハイウェイや装置間の直接結合を容易にするのに役立っている。将来の機能分散システムにも役立つ技術とならう。

4. VLSI の導入

主記憶の VLSI 化は順調に進行しており、コンピュータアーキテクチャの面からみても本質的な困難はない。プロセッサへの VLSI 技術の導入には非常に大きな期待がかけられている一方、導入までに解決を要するいくつかの問題点がある。VLSI のプロセス技術から見れば数年後には 10⁶ ゲートすなわち今日の大形プロセッサを1チップに入れることが可能である。しかしこれに見合う設計技術、利用技術は確立していない。

VLSI でプロセッサを構成するための設計法としては次の三つが考えられる。(1)従来のゲートレベルの論理設計法に従って設計し、適当に VLSI に分割する設計法、(2)機能モジュールをあらかじめ設計しておき、これを組合せてプロセッサを設計する。VLSI チップには適当数の機能モジュールを搭載する設計法、(3)マイクロプロセッサの延長線上にある1チッププロセッサと記憶、スイッチを組合せる PMS 設計法、最後のものはコンピュータ複合体であり、VLSI の種類は少なく余り問題はない。大規模システム実現のた

め第一および第二の設計法をとろうとすると VLSI を手軽に設計できる CAD システムを完成する必要が生じる。第二の機能モジュール方式を目指して階層形 CAD システムの開発が試みられているが、この技術の進展いかんによって将来のコンピュータアーキテクチャは大幅に変わる可能性がある。

5. 仮想化技術

システムをわかりやすいものにするには、使い易いシステムを実現するために大切である。理想的な機能をそなえたブラックボックスの集まりとしてシステムを理解でき、それで安心して使えることになれば、それらを用いて更に大規模かつ複雑なシステムを実現できることになる。

仮想記憶、仮想マシン、仮想端末など仮想記憶にはじまった各種の仮想化技術を理想化された単純な機能モジュールを実現するための技術としてとらえたい。

この技術はコンピュータ技術として独自のものであり、ソフトウェアにおける関数形言語の思想などとも類似性を持った技術であり、大切に育てる必要がある。

単一レベル記憶のように記憶とファイルを一体化する試みも仮想化技術の一種と考えることができるが、その是非は、簡単には結論できない。今後単一レベル記憶を実現したシステムの数が増し、ユーザ側からの評価が集積されて結論がでることになる。

6. 高級言語機能

従来機械語は主としてハードウェアの設計・製造上の制約から基本的なものに限定され、このことが、主記憶とプロセッサ間の情報交換の量を増し、いわゆる Neumann Bottle neck の一因となった。コンピュータアーキテクチャの本来の定義は機械語の定義を意味するから、ここで取上げる話題が本来のアーキテクチャの話題ということになる。

アドレス方式に関連しては、仮想記憶方式の採用と主記憶の容量増大にともなって記憶空間を拡大する要求が強まっている。また各種のデータ構造の取扱いを容易にするため、抽象データ構造の思想をハードウェアの面にも反映させる必要が生まれてきている。スタックポインタをアドレス方式に取入れたのもその一例である。タグマシンの提案やデスクリプタの提案はこれに対する一つの解答であろう。セキュリティの問題もコンピュータ利用がより高度に知的な分野に拡大するにつれて重要性を増し、ケーパビリティなどの保護

メカニズムが提案され、ハードウェア化が検討される時期に来ている。能率向上の観点から細かにケーパビリティを適用しようとする、アドレス方式と関連が出来てくるので、両者を含めて効率のよい方式を検討する必要がある。

機械語の操作コードに代表される操作機能についてはデータ構造との分離が進められよう。乗算命令は整数に対するもの、実数に対するもの、複素数に対するもの、ベクトルに対するもの、マトリックスに対するものすべて共通で、必要に応じてデータ形に適当な交換を加えながら演算が実施されるようになる。算術演算については浮動小数点表示の指数部、仮数部の長さを必要に応じて自動的に変えることにより、溢れやアンダフローの問題を実質的になくするという研究が盛んに行われている。標準化の努力も進められている様子なので近くハードウェア化が進むことになる²⁾。

コンピュータの名が示すとおり、四則演算関係の操作機能は一応備わっているが、今後の応用分野の拡大を考えると非数値データ処理のための操作機能をそなえたコンピュータの実現がまたれる。ビット演算、パターン照合、連想処理など、それぞれの応用分野からの具体的な提案もあるが、汎用機に必須の操作機能と呼べるほど実効のある操作は見つかっていない。その一因としてはデータ構造と操作機能の分離が十分に行われていないことがあげられよう。パターン処理プロセッサ、データベースマシン、リスト処理プロセッサなど各種の専用プロセッサの研究開発が進むにつれて次第に明らかになってくるものと思う。

当面、パターン情報処理や人工知能の研究を進めて必要な操作機能を明らかにすること、これらの研究を支援しつつ実装方式を研究するためにファームウェアによって操作機能を実装することなどを通して研究・開発が進められよう。

高級言語のより直接的な実装法として高級言語プロセッサがあげられる。ソフトウェア危機の解消、並列演算の直接的記述法、非手続き的記述法などの観点から関数型言語³⁾、データフロー言語などの提案があり、これらは新しいコンピュータアーキテクチャを生み出す萌芽となることを期待されている。

これら新しい機能の実装をソフトウェア/ファームウェア/ハードウェアが分担して行う訳であるが、これまでソフトウェアで負担していた部分をどしどしファームウェアさらにはハードウェアに転化するための

努力が今後盛んに行われよう。

7. 並列処理

アーキテクチャの立場からプロセッサの高性能化をはかる手段としては、前章で述べた新しい操作機能の実装と、並列処理とがあげられる。これまで Illiac IV に代表されるような SIMD 形マシンや多くの配列プロセッサにみられるパイプライン方式の MISD 形マシンが開発されてきたが、汎用の並列処理に対して十分な性能が発揮できなかった。大量生産によるコスト低下が期待できるという LSI の特質を十分に生かしたシステムアーキテクチャとして並列処理への期待は大きい。

データフローマシンはデータフローグラフで表現されたアルゴリズムをもとに、データの準備された演算を実施するというデータ駆動形制御方式をとっている。アルゴリズムの持つ演算レベルでの並列性をそのまま活用できることになる³⁾。ペトリネットをそのままシミュレートするような制御方式で、並列性の高いアルゴリズムに対しては高性能を発揮すると期待され、偏微分方程式の解などを主たる対象として各地で研究開発が盛んに行われるようになった。データフローマシンのアーキテクチャはごく基本的なレベルの提案にとどまっており、本格的なマシンが製作されて、数百倍、数千倍といった並列性がいろいろな応用分野で広くとり出されるまでには研究を要する多くの課題がある。

データフローマシンにはデータ駆動形制御方式のほか、関数形言語との対応のよい要求駆動形制御方式がある。高い並列度を確保するためには四則演算などの単純な演算レベルでのデータフローを考えるのがよいが、データフローにともなうオーバーヘッドを少なくするためには、プロシージャレベルといった複雑で大規模な演算レベルを考えた方がよい。

データフローマシンは MIMD 形マシンとみることができ、並列演算を直接指示する必要のないこと、比較的適用範囲が広いと考えられることなどの期待があるが、データ構造が単純な場合などでは SIMD 形やパイプライン制御のマシンの方が、制御のためのオーバーヘッドが少なく効率が良いかもしれない。

いずれにしても LSI の特色を生かした並列処理プロセッサの実現が期待されている。

8. 非数値データ処理

ハードウェアの価格の低下を利用してより知能レベルの高いコンピュータシステムを実現し、より知的な用途にコンピュータを用いたい、誰にでも使いやすいコンピュータを実現したいといった希望がある。このためにはデータベースマシンに大量の知識を貯え、人工知能の研究で得られた連想機構、推論機構、学習機構の知識をシステム内に取込んでコンピュータシステムの知識レベルを高めることや人間の持つ各種の感覚器官や運動機能をコンピュータに持たせてマン・マシンインタフェースをより自然で自由なものにすることが大切になってくる。

これらの機能をコンピュータに持たせるためには、従来の数値データを主たる対象としたアーキテクチャでは不十分であり、非数値データが自由に取扱え、これを処理するための演算機能をそなえたアーキテクチャが必要になる。

マン・マシンインタフェースとしては文章・音声・図形・画像などが多重メディアとして同時に使えることが望ましい。これら各種のパターン処理には各種のビット処理や連想処理が高速に行われることが必要であり、専用プロセッサの開発が試みられている。

パターン情報処理はリモートセンシング技術や医用画像処理に多くの例がみられるように、マン・マシンインタフェースとしてだけでなく、独自の多くの応用分野を開拓しつつある。しかしこれに適したアーキテクチャを持ったプロセッサの開発のおくれていることが、この方面も研究を実用化するための障害となっている。

文書処理は事務計算の次にくるコンピュータの大きな応用分野と考えられており、生産性向上の遅れている事務処理に革命をもたらすものと期待されている。文書処理をすすめていくと、ある程度の文章理解能力をコンピュータに持たせる必要が生まれてくるものと思われる。自然言語処理を始めとする人工知能の研究にはほとんどの場合 LISP が用いられ、このため LISP プロセッサの普及と性能向上の必要性が叫ばれており、リストを効率よく処理できるアーキテクチャの開発が、文書処理を高度化する上にも必要になってくる。LISP はまた代表的な関数形言語でもあり、この面からも独自のアーキテクチャの出現が期待されている。

我が国の場合、漢字かなまじり文の取扱いを無視し

て文書処理を考えることはできない。日本語用コードでは一文字あたり2バイトを用いるので、これが簡単に扱える操作機能の実装が望まれるが、技術的な困難はほとんどないものと思う。

9. データベースマシン

データベースマシンに要求される機能も関係代数などで代表される連想処理を大量のデータに対して高速にほどこすことである。情報システムが高速化し、知的な業務が増えるに伴ってデータベースの役割は重要になり、必要な情報を自由に高速に検索できることが情報処理全体の性能に対して大きな影響を持つようになる。非手続き形言語や QBE⁴⁾ のように図表など直観的な非手続き的手法で複雑な問合せを発することができ、しかも並列処理などを極力利用することにより、大量の情報から必要な情報を高速に検索できるアーキテクチャの研究は盛んに行われている。多数の個人が共同利用するデータベースで効率よく利用のできるデータベースマシンを実現するにはなお多くの研究が必要に思われる。

固定ヘッドディスクや磁気バブル記憶とデータベース管理用連想プロセッサを組合せたデータベースマシンの研究が盛んであるが、磁気バブル記憶の場合、論理を一部組込むなど、データベースマシン向き磁気バブル記憶装置を開発する研究も必要であろう。

10. OS 機能の組み込み

最近のシステムでは OS の機能の一部をファームウェアで組み込むことにより、システム制御のオーバーヘッドを減らす努力が盛んにはらわれている。この理由がファームウェア化による性能向上にあることは当然であるが、OS の機能に限る必要は必ずしもない。しかし、ユーザからみたアーキテクチャを変更することなくシステム性能を向上させるには良い方法と言えよう。他の理由としてはシステム制御のための処理には非数値データ処理が多く、従来のアーキテクチャの弱点であったこともあげられよう。

ファームウェアの役割としては、ソフトウェアの側からコンピュータに要求するアーキテクチャと、ハードウェアで直接実現されるアーキテクチャとの間にギャップがあるとき、これを埋めることがあげられる。

このことは(1)ソフトウェアとファームウェア間のインタフェースをソフトウェアからみて理想的なアーキテクチャにきめることができること、(2)またファ

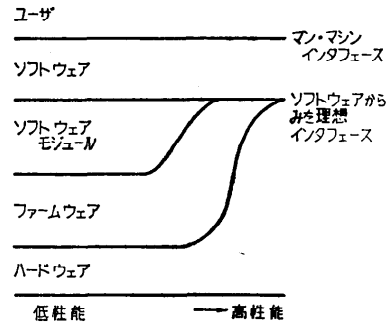


図-1 ソフトウェア/ファームウェア/ハードウェア間のトレードオフによる理想的設計法

ームウェアとハードウェアのアーキテクチャはその設計時点における技術レベルを考慮して最も経済的になるようにきめることができることを意味する。このようにしてソフトウェアからみてきめたアーキテクチャはハードウェアの技術変化に直接さらされないから長期にわたって理想的なアーキテクチャとして使えることが期待できる。

今後応用分野が拡大すると、応用分野ごとに必要とする機能およびその性能に対する要求が変わってくる。このことを考えると、図-1に示すようにソフトウェアモジュールとそれらを使うソフトウェアの間に一般的な理想アーキテクチャを設定し、それぞれの機能にどの程度の性能をあたえるかを応用システムごとに決定してその実装法をソフトウェアとファームウェア間のトレードオフと、ファームウェアとハードウェア間のトレードオフによって決定するというのが理想的な設計法となる。

各機能に要求される性能の最適値はシステムをとりまく環境の変化によっても変わるからこれに適合できるように柔軟性を持たせたシステムとすることが望ましい。たとえばAと言う機能の性能が重要になり、Bという機能の重要性が減ったとすると、Aに対するファームウェアを強化し、Bに対するファームウェアを減らしてソフトウェアでまかなうといった適合性をシステムに持たせたい。このためにはシステムの動特性を測定するモニターをつけて性能測定を行うなどの技術をより一層発展させる必要がある。

11. 機能分散システム

多様化する応用分野に対応し、しかも LSI 化に必要な大量生産方式を取入れやすいシステム構成方式として、同一プロセッサを複数個用いて行う負荷分散シ

システムとそれぞれの目的に指向した専用プロセッサを組合せる機能分散システムとが考えられる。負荷分散だけでは台数を増したとき、オーバヘッドが増し、汎用機による 10 台以上の多重プロセッサシステムには無理があると言われている。そこでここでは負荷分散と機能分散の両方式を併用する広義の機能分散システムの構成法について述べる。

情報システムの持つべき機能を大別して情報処理、情報管理、情報交信の三つに分け、それぞれがサブシステムを構築するとする。中央処理プロセッサ、バックエンドプロセッサ、フロントエンドプロセッサの発展形と考えることができよう。それぞれのサブシステムを構成する専用プロセッサとして考えられるものを列挙すると次のようになる。

情報処理サブシステムとしては従来形演算プロセッサのほか、各種の高級言語プロセッサ、アレープロセッサ、データフローマシンなどが考えられる。

情報管理サブシステムとしてはデータベース管理用の連想処理プロセッサのほかファイルアクセスプロセッサ、記憶階層制御プロセッサ、大容量記憶制御プロセッサなどが考えられる。

情報交信サブシステムとしては従来形の入出力処理プロセッサのほか通信制御プロセッサ、パターン処理プロセッサ、画像処理プロセッサ、音声処理プロセッサ、信号処理プロセッサなどが考えられる。

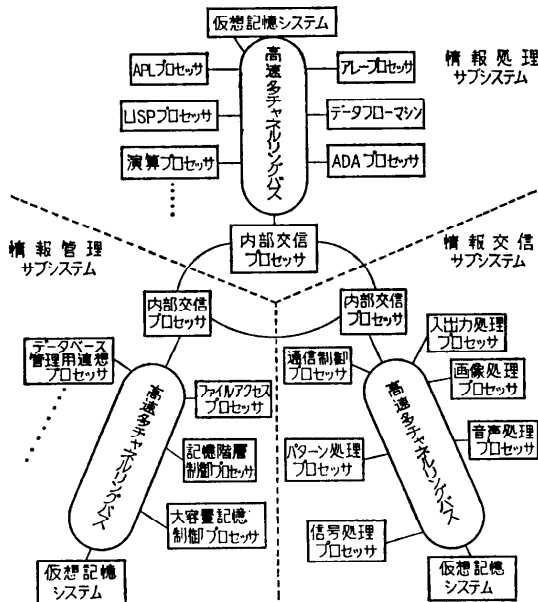


図-2 機能分散システムの構成イメージ

機能分散システムを実現するためには、これらの専用プロセッサをいかに結合してサブシステムを構築するか、さらにサブシステム相互間をいかに結合するかについて研究する必要がある。機能分散システムの構成方式の一例を図-2 に示す。

12. 網アーキテクチャ

計算機網は当初それぞれのシステムごとに独立して開発が進められ、各種の専用分散システムが作られたが、このような開発方式では、質の良いシステムを安価に提供できず、メーカ、ユーザのいずれの側からみても大きな不利をまねくことが明らかになった。このため、IBM が SNA と呼ばれる網アーキテクチャを発表したのを契機に、各社がそれぞれ自社用標準網アーキテクチャを設定している。一方情報システムとして考えたとき相互に結合され、どこでも自由に情報の交信ができることが理想的な体制と言える。このため計算機網を相互に接続する場合の網アーキテクチャは少なくとも標準化することが望ましいという考え方に立って 1978 年から ISO/TC 97/SC 16 で網アーキテクチャの国際標準を制定するための努力が開始されている。現在参照モデルと呼ばれる枠組についてレイヤ (図-3 参照) 構造を採用することで大筋の合意が得られた段階である⁵⁾。下位の 4 レイヤはデータ通信主体のレイヤであり、かなり討議が進んでいるが、上位の 3 レイヤは OS とも関係があり、利用法とも深い関わりを持っていくので今後多くの仕事が残されている。

当面、仮想端末やファイル転送、システム管理といった用途に必要なプロトコルの標準化から手がけられ

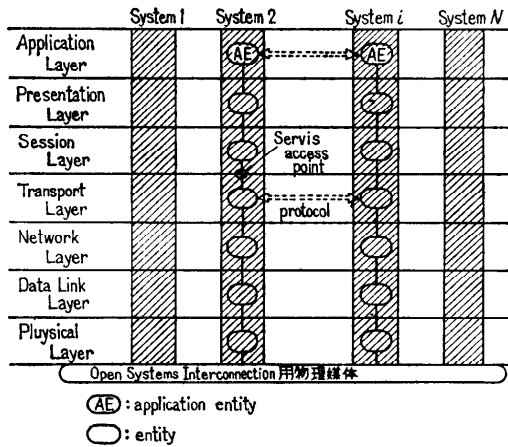


図-3 網アーキテクチャのレイヤリング構造

ることになる。

分散データベースの実現は広域情報システムにとってもっとも重要な課題であるが、基礎的な研究と、研究を必要とする課題の整理が進められている段階であり、標準的な手法が確立するまでには多くの研究が必要である。

網アーキテクチャの標準化などでは、いろいろな通信方式や利用法を考慮して、それらを包含できる汎用性のあるものにする努力がはらわれる。しかし汎用性を持たせることは個々のシステムの立場からみると効率の悪い方式となることが多い。この間の妥協をはかり、汎用性と効率が両立するような方式を見出す努力が必要になってくる。

13. おわりに

コンピュータ技術もらん熟期に入り、多方面の応用分野と多様な利用形態にきこまかく対応できる環境が整備されつつある。このことはアーキテクチャの面からみると一世代前の単一アーキテクチャ時代を脱し多様なアーキテクチャの時代を迎えることを意味する。

このため簡単ですっきりした展望を行うことは困難と考え、アーキテクチャにまつわる課題を出来るだけ列挙することを試みた。しかし、システムの信頼性を高め、保守、運用を容易にするといった重要な立場からのアーキテクチャの考察など欠けた点も多い。紙数の制約もあり、課題の列挙にとどまって、その内容の説明や解法の私見などを述べることはできず、読みにくい論文となったことをお詫びする。

以上述べてきたことを要約すると今後のアーキテクチャの研究課題として次のようなことがらを考慮に入れたアーキテクチャの実現があげられる。

(1) 人工知能の研究などで得られた知見のうち、コンピュータの知識レベル向上に役立て得るものを取

入れ活用することのできるアーキテクチャ。

(2) ソフトウェア学の研究で得られた知見のうちソフトウェアの危機の解消に役立て得るものを取り入れ活用することのできるアーキテクチャ。

(3) 高度の並列処理の実現、非数値データ処理の高効率化を目指した非ノイマン形アーキテクチャ。

(4) LSIあるいはVLSIの特徴をいかして、安価で高性能であり、ユーザにとって使い易い機能をそなえたアーキテクチャ。

いずれにしろ、(1)人間社会を豊かにすることに役立つ情報を提供できるシステム、(2)人類にとって未知のことがらの発見に役立つ実験システム、(3)人間が生活し仕事をすすめる上に役立つ奴れいロボットといった夢を実現することを目指して、アーキテクチャの研究が着実に進められることを希望して筆を措く。

参考文献

- 1) Coonen, J. T.: An Implementation Guide to a Proposed Standard for Floating-point Arithmetic, Computer Vol. 13. No. 1, pp. 68-79 (Jan. 1980).
- 2) John Backus: Can Programming Be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs, Comm. of ACM Vol. 21, No. 8, pp. 613-641 (Aug. 1978).
- 3) Dennis, J. B. and Misunas, D. P.: A Preliminary Architecture for a Basic Data-Flow Processor Proc. of 2nd Annual Symposium on Computer Architecture IEEE pp. 126-132 (1975).
- 4) Zloof, M.: QUERY BY EXAMPLE: a Data Base Language IBM System J. Vol. 16. No. 4, pp. 324-343 (1977).
- 5) 元岡, 苗村: 開放型システム間接続 (OSI) の標準化, 情報処理, Vol. 20, No. 12, pp. 1096-1104 (Dec. 1979).

(昭和55年4月3日受付)