

報 告

パネル討論会

専用プロセッサの動向

昭和 51 年度第 17 回大会*報告

パネリスト

乾 篤男(富士通), 島田 俊夫(電総研), 都丸 敬介(横須賀通研)
 服部 光宏(日電), 前川 守(東芝), 司会 元岡 達(東大)

司会(元岡) 最初に専用プロセッサとしてここで取り上げますものがどういうものであるか、あるいはどういう趣旨でパネルをやるかということについて簡単に触れさせていただき、この後パネリストの方々からお話をいただきたいと思います。

汎用プロセッサに対して専用プロセッサというのが最近話題になっていますが何のために専用化するかという問題があります。大きく分けまして高性能をねらうことを目指した専用プロセッサと、コスト・パフォーマンスの向上を目的とした専用プロセッサがあります。その両方が今日の議論の対象になるわけで、専用化による性能の向上を両者に分けることは難しいんですけども図-1 のように思いきって分けてみました。

汎用の機械に対して専用の機械が考えられるようになったのにはそれなりの理由がなければいけないわけとして、次にその技術的な背景をまとめてみます。

先ず第一にアプリケーションの面から見ますと、応

- 専用化による性能向上を目指したもの
 - Array Processor
 - Data Base Machine
 - Communication Control Processor
 - Signal Processor
 - Image Processor
 - Associative Processor
- 専用化による性能/価格比の向上を目指したもの
 - Personal Computer
 - H. L. Language Processor
 - Terminal Processor
 - I/O Processor
 - OS の firmware 化

図-1 専用プロセッサのねらい

* 日時 昭和 51 年 11 月 21 日、15:30~17:30
 場所 慶應義塾大学日吉第 6 校舎〔全体会場〕

用分野の多様化、あるいは利用方法の多様化ということが挙げられます。

二番目には超大型システムに対する反省。

三番目にはオンライン利用の普及それにともなって分散制御システムへの指向ということが挙げられると思います。

ソフトウェアの面から見ますと、ソフトウェア技術にいろいろな問題点があることはわれわれよく体験していることでございますが、そのソフトウェアに対する負荷を専用機によって軽減しよう、あるいはソフトウェアの機能をファームウェア化することによって性能向上しようといった点が挙げられます。

それからハードウェアの面から見ますと、記憶であるとか論理回路の価格が低下したということが専用化を促す一つの原因であると思います。

それからミニコンであるとかマイクロプロセッサといったものの普及、あるいは今後の LSI 技術の導入にともなって専用プロセッサ化の問題が出てきます。

またデザイン・オートメーションに代表されるような設計技術の進歩が専用化を可能にしている一つの原因であります。

マイクロ・プログラムやファームウェア技術の進歩と普及、保守検査の自動化、こういうものが専用プロセッサの開発を促している要因であると思います。従来なぜ汎用機を使っていたか、汎用機が全盛の時代に専用プロセッサを作らない理由としてよく挙げられてきたものとしましては、ソフトウェア開発の困難さ、ハードウェア設計の困難さ、保守検査の困難さ、があります。このあと二つについてはかなり問題が解決されてきましたが、ソフトウェアの問題は依然大きな問

題が残っているわけですけれども、これを軽減するという効果も期待されています。しかし逆に専用機のソフトウェアという別の問題を引き起こす恐れがあります。そのへんの問題についてはあとでパネリストの方に議論していただきたいと思います。

今日、これからお話をいただく大筋を申し上げますと、次の五つのテーマについていろいろ討議してみたいと考えております。

一つは、専用プロセッサがいいのか汎用プロセッサがいいのかという問題で、これにつきましては高級言語プロセッサの開発を進めておられます電総研の島田さんにお話をいただきます。

二番目には、専用プロセッサにどういった利点があるかということで、これにつきましては通信制御プロセッサの開発に従事しておられます横須賀通研の都丸さんにお話をいただきます。

三番目としましては、専用プロセッサの設計技術といったテーマでアレープロセッサの開発をしておられます富士通の乾さんにお願いします。

四番目に、専用プロセッサのソフトウェアということで、データベース・マシーンの開発を担当しておられます東芝の前川さんにお話をいただきます。

それから五番目に、専用プロセッサと保守の問題ということで、端末システム関係の開発をなさっていらっしゃる日電の服部さんにお話をいただきたいと思います。

時間もございませんし、それぞれの立場からお話をいただくので、いろいろ抜ける点、議論のわかる点があると思います。みなさまの活発な御討論を期待している次第であります。ひとつよろしくお願ひします。

島田 私の話は専用プロセッサか汎用プロセッサかという題であります。本題に入る前に、専用プロセッサとしてどんなものを考えているかということを一応説明しておきます。専用プロセッサには特定のものに合わせてハードウェアから設計されたプロセッサとファームウェアによって実現されたものがあります。ファームウェアの場合には書き替え可能な制御記憶がありますからソフトウェアと同じように書き替えることができるわけですけれども、あまりそういうことは頻繁には行わないだろうということで一度ファーム・ウェアを変えてしまうと、別の専用プロセッサになったということにします。

さて、専用か汎用かというテーマに戻ります。ここでは計算機にとって重要な点を幾つか挙げて

説明してみたいと思います。

まずアーキテクチャに影響を与えるユーザの要求から考えていきますと、第一にコスト・パフォーマンスがあります。その場合にどちらかというと高速・高能率を目指すようなマシンと、どちらかというと低価格を目指すものがあります。例えばアレー処理用のプロセッサとか信号処理用のプロセッサは高性能の方に入るでしょうし、マイクロプロセッサを使ったものとか端末プロセッサ等は低価格化が主目標になるのではないかと思います。

高級言語の計算機について見てみると、研究用に作られているマシンが大変多い状況で、そのようなものは特に高性能を主目標にする場合が多いようです。しかし近年はハードウェアの価格が下がったため最近の商用のパーソナル計算機のようにマイクロプロセッサを使って低価格で使い易いものを目指したものも増えています。それでは専用機を作る境界はどのへんにあるかということを考えてみると、バランスのとれた常識的なコスト・パフォーマンスを目指す場合は、やはり汎用になるのではないかと思います。それに対し特定の応用において非常に高性能を目指すとか、あるいは逆に非常に低価格なものをねらう。そういう立場になると専用プロセッサを追究することになるのではないかと思います。

ですからコスト・パフォーマンスの問題というのは必ずしも専用化に拍車をかけるというわけではなくて、汎用プロセッサを選ぶ場合もあるし、立場によっては専用プロセッサを選ぶ場合もある。このへんに両者の共存を許す要因があると思われます。

次に互換性の問題を考えてみると、これは過去のソフトウェアの蓄積を生かすことが重要であります。機械語レベルでプログラムを書いている限りは、まず計算機間の互換性というものはありません。高級言語でシステム・プログラムを書くようになれば互換性は増すので本来、互換性は専用プロセッサに有利な要因と考えられます。しかし現状は各メーカーが古い機種の機械語をエミュレートする形で過去のソフトウェアの蓄積を生かす方向にあるので、互換性というのはどちらかというと専用化を妨げる要因として働くのではないかと思います。

次に使いやすさの問題を挙げてみると、これはハードウェアの問題とソフトウェアの問題に分けられると思います。ソフトウェアに関しては、これまでソフトウェアで行ってきたことをハードウェア化しないし

はファームウェア化するわけでして、ソフトウェアの負担が軽減します。これはたとえばOSの機能を組み込むことによってOSの作成が容易になるとか、ソース言語と機械語の距離が小さくなることによってコンパイラが簡単になり、ディバッグも容易になること等があり、かなり使い易くなるのではないかと考えられます。ハードウェアの面について述べますと、例えは小型化するとかパーソナル化するとか、つまり、電卓みたいな格好で手軽に使えるというようなこととか、スタートするまでの手続きが簡単になるとかそういう使い易さがあると思います。従って使い易いという要因は専用化を進める要因ではないかというふうに考えています。

それからもう一つユーザ側から見てみると、高信頼性という問題があるわけですけれども、これは専用プロセッサ、汎用プロセッサともLSI化が進んでいくでしょうから、ハードウェア的な差はないと思います。ソフトウェアに関しては専用プロセッサの方が簡単になるのでやや有利ではないかと思います。

元岡 今のお話につきまして御意見ございましたらお聞きしたいと思います。

専用プロセッサ化することが有利なことが一般に認められているというようなケースもかなりあると思います。Array Processorであるとか、Signal Processorなんかはその例だと思いますが、一方、汎用プロセッサを使うべきか専用プロセッサを使うべきかというこ

とが割合議論の対象になっている例としては DataBase Machineがあると思いますが、前川さん何か。

前川 汎用か専用かという件で DataBase Machineに限って申しますと、大体傾向は分散型に向っていると思います。幾つか理由が挙げられるわけですが、一つはデータの処理を分割して性能向上をはかること、第二に拡張がやさしくなるということが挙げられます。それからデータ・ベースのセキュリティというのが一つの問題なわけですけれども、情報をどこかでしまるということによってセキュリティの明確化がやさしくなる。更にはデータの共用ということもやりやすくなる。このような理由から専用化の方向に向くと思います。

元岡 どうも有難うございました。何かフロアーの方から御意見ございませんでしょうか。

伊藤(電電公社) 設計コストの問題を考えますと、今後プログラムのコストが非常に高くなる。それからもう一つはLSIの量産化が進んできますと、かなり大

処 理

きなフレキシビリティを考えないといけない。そういう二つの問題を考えますと、やはり汎用化がいいという要因がかなり出てくるように思います。島田さんは、そのへんを全然お話にならなかつたんですが、私はこのへんがかなり大きなエイトになるんじゃないかなと思います。そのへんについて御意見いただきたいと思います。

島田 量の問題については、マイクロ命令にフレキシビリティを持たせておき、マイクロプログラムでのレベルで専用化をしていけばよいと考えています。つまり書き替え可能な制御記憶を持ったマイクロ・プロセッサを、いろんな問題に適用するということです。I/Oに対してはこのような解決法が既にとられていると思います。それから言語プロセッサに関しても商用機として出ているものは一種類の言語ではなくて、多種類の言語をインターフィットするようになっていますけれども、これもファームウェアのレベルでもってインターフェースを吸収するような格好になっています。これで量の問題はかなり解決できるんじゃないかなと考えています。ただファームウェアにあまりフレキシビリティを持たせますと効率の点で問題がでてきますから、どのようなファームウェアを用意するかということはまだ研究課題であると思います。ソフトの面はソフトウェアがハードウェア化されていますから、ソフトウェアの作成が簡単になりますので専用化のほうが有利だと思います。

元岡 この問題はある意味ではこのパネル全体のテーマみたいなものなので、それぞれの立場から御意見があると思いますけれども、最後にまたこの専用でやるか汎用でやるかという問題に戻らしていただくことにしまして、次に進めていきたいと思います。

次には専用プロセッサの利点という問題について通信制御プロセッサ関係をやっていらっしゃる都丸さんにお願いいたします。

都丸 専用プロセッサの利点は何かというテーマをいただいたわけでございますが、専用プロセッサの利点は、機能分散の利点とほぼ一致すると私は考えます。機能分散がなぜ最近注目されるようになってきたかというと、一つはコンピュータが大きくなりまして、そこにいろいろな機能を集中しすぎたためにいろいろな面でオーバーヘッドが大きくなりました。そのオーバーヘッドを減少させるために今度は機能分散が始まつたと言えるかと思います。また、あまりにも集中化したために、コンピュータが非常に固くなってしま

た。ソフトウェアそのものが非常に固くなってしまって、機能の追加であるとか変更に対する柔軟性がなくなってきました。それを解決する一つの手段として機能を分散しまして、なるべくとした範囲で機能の追加・変更をやれるようにしてきたということではないかと思います。それからもう一つはさきほどもお話をございましたが、信頼性の問題があります。あまり集中化いたしますと、その集中化したところがトラブルになったときにすべてが止まってしまいます。ところが処理の流れをみておりますと、すべての機能が働くなくても、ある一部の機能が働いていれば間に合うケースが多くあります。このような場合には特定の機能を専用化し、分散化することによってシステム全体としての信頼性を高めることができます。それでこれから私がただ今担当しております通信制御プロセッサを例に取りましてこのへんのことについて若干説明いたします。

通信制御プロセッサと申しましてもいろいろなものがありますので、まず通信制御プロセッサについて三つの点から御説明します。一番目は、通信制御プロセッサとは何かということ、その次になぜ通信制御プロセッサが必要になってきたかという点、それから三番目にその特徴はどういった点にあるかということについてお話をしたいと思います。

通信制御プロセッサとは何かということですが、代表的なものとして Front End Processor があります。これは端末と Host の間でやり取りする通信のための通信制御を主として担当します。これは一種の機能分散であります。複数の端末から入ってきます情報を収束いたしまして多重化することにより回線使用能率を高めることを目的とする Remote Concentrator も通信制御プロセッサの一つであります。それから更に少し高級なシステムになってきますと、いろいろな Host Processor 相互間を接続するために Message Switcher と呼ぶ交換機能を持った通信制御プロセッサが使われます。

次になぜこのような通信制御プロセッサが必要になってきたかということでございますが、代表的な要因をあげると次のようになります。

一番目は、機能の追加とか変更に対する柔軟性であります。通信制御に関して見ますと、端末の多様化が非常に早いスピードで進んでおります。多様化する端末装置を簡単に接続するために、その端末の属性がホストまで及ばないように Front End Processor や、

Remote Concentrator の中で吸収するという処理が行われます。

二番目に、通信制御に関するホスト計算機の負荷の軽減であります。これはシステムによって違うわけであります。例えば銀行などの全国的な規模のネットワークを持つシステムで、ホストに通信制御のためにどのくらいの負荷がかかっているかということを調べてみると、小さいもので数パーセントから 10%、大きなものになりますと 40% とか 50% にもなります。このようなシステムでは明らかに通信制御の機能を分散することによりホストの性能が上がるわけです。その結果システム全体としてのコスト・パフォーマンスがよくなります。

三番目に異種計算機間の接続があります。複数のコンピュータ相互間をつなぐコンピュータ・ネットワークは技術の趨勢としてどんどん広がりつつあるわけでございます。ところが現在コンピュータを使っておられる方がコンピュータ間の接続を行うために新しいコンピュータに置き替えるということはまずたえられないことでありまして、現在使っておりますコンピュータをそのまま他のシステムとつなぎたいという要望が当然起こってくるわけであります。そのためには、現在使っておりますコンピュータの機能の違いをどこかで吸収しなければならないわけで、通信制御プロセッサがそういうことを果たす役割を持っているわけであります。

通信制御プロセッサの特徴について御紹介します。通信制御プロセッサといいましても、プロセッサそのものにつきましてはそれほど大きな特徴があるわけではありません。現にミニコンピュータを使用したもののが多く出まわっているわけであります。特徴の一一番目は回線とのインターフェースを持ってるということです。ミニコンをベースにしたものでも、それを通信制御プロセッサとして仕立てるために回線とのインターフェースの部分をつけております。それから専用に作られた通信制御プロセッサでもそれぞれのメーカーさんの特徴が出ている点、あるいは設計上特に注意しなければならない点は、回線とのインターフェースの部分であります。

二番目は、動作の実時間性に対する要求が非常に厳しいということであります。通信制御プロセッサではたくさんの回線から次々にデータが入ってくるわけでありますが、それがプロセッサの動作と非同期で入ってくるわけであります。入ってくるデータそのものが

回線のスピードで入ってくるのですから入ってきたあるデータをメモリに送るなり何なりの処理をしきらないうちに次のデータが入ってきますとデータが落ちてしまいます。そういったことから動作の実時間性に対する要求が厳しいわけでありまして、厳しい実時間性を満足するための方法として割込みの処理を多重にするとか、あるいは回線からメモリに対するダイレクト・メモリ・アクセス機能を持たせるとか、そういうことが必要になります。

三番目はコスト／性能比の考え方でございます。未だ通信制御プロセッサのコスト／性能比に対する考え方方が定着し確立していません。通常のプロセッサの場合には単位時間内にどれだけの命令が実行できるかということからコスト／性能比が評価されますが、通信制御プロセッサの場合には回線当たりどのぐらいの値段でできるかとか、単位時間にどれだけのデータを回線に対して有効な情報としてやり取りできるかというような尺度もあるかと思います。いずれにしてもまだ評価尺度が定着していませんので、はっきりさせる必要があります。細かいことになりますとまだいろいろとございますが、とりあえず問題点だけで終らせていただきます。

元岡 どうも有難うございました。何か御質問御意見ございませんでしょうか。

一つ伺いたいんですが、CPU を通信回線につなぐと、その負荷のために CPU の半分ぐらいが奪われてしまうような例があるということはよくわかるんですけれども、そうすればもう一台普通の CPU を置いてもいいわけです。そこで通信制御プロセッサという専用化されたシステムを使うのは、汎用のものを使うのに比べて何かが有利になるのだと思います。例えばゲート数というようなもので比較すると、汎用プロセッサに比べて同じ性能を出すのに 10 分の 1 に減るとか 5 分の 1 に減るとか、何か具体的な数値があればお話ししていただきたいと思います。

都丸 世の中でよく知られております IBM の 3705 という通信制御プロセッサについて見ますと、これが大体 Host の 370 の CPU と比較しまして一桁違います。分散したプロセッサ、特に通信制御プロセッサの場合に機能を分けて、しかもシステムとしてのコスト・パフォーマンスがよくなるというためには、先生が今御指摘になりましたように、同じ大きさではむずかしいと思います。一桁ぐらい小さいプロセッサで専用化することによってできなければ専用プロセッサと

いうのはトータルなシステムのコスト・パフォーマンスとしては成り立たないのではないかと思います。

元岡 他に何か御質問ございませんでしょうか。その専用プロセッサの利点というのはおそらくそれぞのプロセッサによってねらうところに差があるんだろうと思います。通信制御の場合考えられることは、さきほど言われたような柔軟性と信頼性と、それから今言いましたようなコスト・パフォーマンスの問題と、まあ三つあるとして、そのうちみんな大事なんだというのがお答えになるかも知れませんけれども、主としてねらっているところは何なのでしょうか。

都丸 一番大きなねらいは機能の柔軟性でございます。IBM を引き合いにして大変恐縮でございますけれども、3705 が出来てその後、最近 SNA を発表しておられます、なぜああいうものが出てきたかという背景を考えますと、端末の機種とか機能が多様化しすぎたことが大きな動機になっています。それなら端末の多様化を押えたらいいじゃないか、標準化したらいいじゃないかということが言えるかも知れませんが、端末というのはユーザと直接接触してる部分でありまして、ユーザの業務上の希望を取り入れなければならないこととマン・マシン・インターフェースも重要なことなどから多様化がどうしても避けられない。避けられない多様化を吸収するための柔軟性が一番大きな点だと思います。

元岡 それでは次にさきほどちょっと御質問にも出了しましたように、新しいプロセッサを設計するというのはこれは大変な問題なわけで、専用プロセッサを作る場合には専用プロセッサを作るのにふさわしいだけの設計技術の向上があったのだろうと思いますが、そのへんにつきましてアレイ・プロセッサを開発しておられます富士通の乾さんにお願いいたします。

乾 スーパー・コンピュータ、あるいはアレイ・プロセッサなどと呼ばれます科学技術計算を超高速に処理することを目的としました専用計算機は Illiac IV でありますとか、COCTSTAR 100 とかが大変有名なわけであります。最近この種の計算機が単なる研究の段階あるいは軍用などの特定の範囲での実用化にとどまることはなく、商用機として多く登場することになってきましたわけであります。代表的なアレイ・プロセッサについて高速化の手段を中心として、ハードウェアの構成を分類しますと、図-2 のようになります。計算機の処理の高速化は同時処理を高いレベルで行うことによって実現されるわけでありますが、その実現手段

高速化手段	パイプライン型			パラレル型
構成	汎用機接続 チャネルレベル	汎用機接続 プロセッサレベル	スタンドアロン	複合体
高 速 性	△	○	○	◎
生 産 性 (プログラム)	○ FORTRAN (CALL文) +a	○ FORTRAN +a	○ FORTRAN +a	× FORTRAN like
コ 斯 ト	◎	○	△	×
使 用 目 的	特定小規模 科学技術計算	大規模科学技術計算 (原子力、航空宇宙、画像処理、防衛シ ステム、気象計算、構造解析など)		
代 表 例	IBM 2938 UNIVAC -AP	F 230-75 APU.	CDCSTAR 100 CRAY-1 TI-ASC	ILLIAC IV PEPE STARAN

図-2 高速化処理実現の手段と特長

としてはよく知られていますように、パラレル型とそれからパイプライン型の二つがあります。パラレル型といいますのは、SOLOMON コンピュータによって初めて試みられたと言われていますが、現在実用化あるいは開発中というのも含めて、Illiac IV ありますとか、PEPE、STARAN などがございます。パラレル型は今のところ非常に大規模になるということ、あるいは経済性に難点があります。また一方ソフトウェアの面でもアレイ・プロセッサをフルに活用させるためには並列処理の高級言語の開発、即ち従来の汎用ソフトとは異なる新しい OS が必要になります。スピードの面ではパイプ・ライン型よりもはるかに可能性があるため、経済性よりも必要性のほうが優先する軍用などで実用化されるというのが現状であります。一方パイプ・ライン型は一つのマシンの中でのオーバーラップ処理を高度に取り入れて高速化を図るもので、技術的にはハードウェア、ソフトウェアとともに汎用機の延長上にあり、従いましてパラレル型と比べますと実現が比較的容易であると考えられます。商用機として開発されたアレイ・プロセッサは現在いずれもこのタイプのもので、比較的小規模のものとしては、IBM の 2938、UNIVAC-AP があり、大規模なものとしては CDC の STAR 100、cray-1、T1-ASC 等があります。スピードの点ではパイプ・ライン型はパラレル型に一歩譲るものであります、そのためにパイプライン型でもプロセッサを複数配して並列化を計っているものがあります。T1-ASC というコンピュータは最大四個のプロセッサを有しております。更にこれは 8 プロセッサにするという計画があるということとも聞いております。パイプライン型の計算システムの構成方法としては、既存の汎用システムに接続する

型と、それから独立の計算機システムを構成するスタンドアロン型があります。汎用システムへの接続型は汎用システムとシステム・リソースを共用できる点がスタンドアロン型と比べて有利な点であると考えられます。例えば一例としてソフトウェアについてみると、ベクトル命令を使用できるような FORTRAN の拡張をするということで既存の汎用 OS がそのまま使えると言ふことです。

汎用システムとの接続法にはこのチャネルレベルで接続するものと、プロセッサレベルで接続する二つがあります。チャネルレベルで接続する場合には、チャネル・インターフェースのほうでスピードが抑えられるために比較的小規模のものとならざるを得ません。この場合にはスピードの面で汎用超大型機が要求されるが、汎用機では少し高すぎるというような分野に対してのコスト・パフォーマンスの改善がねらいであると考えられます。プロセッサレベルで汎用システム接続型の F 230-75 のアレイプロセッサ、あるいはスタンダード型の STAR 100 など、いずれも汎用超大型機では実現できないスピードの壁を破ることが可能であり高速性を要求する分野に対しての広い範囲の適用性を指向していると言えます。

以上、代表的なアレイプロセッサについての高速化の手段と特徴を概略申し上げたわけですが、この中からパイプライン型であります F 230-75 アレイプロセッサについて、設計技術上特に留意した点を申し上げたいと思います。

F 230-75 のアレイプロセッサというのは、処理速度を 75 CPU の 10 倍以上ということに目標をおきまして、汎用性を重視したコスト・パフォーマンスの高い計算機システム、それから商用化ということを目標に開発しております。商用化を実現するという場合に開発コストを下げる事が非常に重要なわけであります。そこでアレイプロセッサにおきましては、このアレイプロセッサのためだけの新規開発というものを極力少なく、既にあるリソースの活用ということに重点をおいております。即ち第一に回路方式でありますとか、実装技術などのテクノロジーの面は、F 230-75 のものと同一のものを採用しております。

第二に、システムの構成方法としては、汎用システム接続型を採用しております。これはプロセッサレベルでの接続であり、このようなシステム形態をとることによって、ハードウェア、ソフトウェア両面においてリソースを有効に活用しております。即ちハードウ

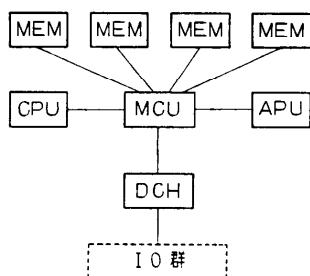


図-3 FACOM 230-75 APU システム

エアの面では、アレイプロセッサの本体だけの開発で、その他のデータ・チャンネル、メモリ等は既存のものをそのまま使って計算機システムが構成できるということです。また更にソフトウェアの面では、FACOM 230-75 のモニター・システムをそのまま使用しております。

第三に、高速化の手段としてパイプライン方式を探用したことあります。即ち、設計技術の面ではパイプライン方式のアレイプロセッサはハードウェア、ソフトウェアともに汎用プロセッサの延長上にあり、蓄積された汎用機の技術が活用できるからであります。

以上申し上げました3点が、このアレイプロセッサを開発するに当って特に留意した事柄です。

最後に、このアレイプロセッサの性能について付言しますと、パイプライン演算器は1クロック・タイム(90 ns)に1個の倍精度演算を実行する能力を持っています。この値は、同じテクノロジーを用いたFACOM 230-75 CPU と比べると10倍以上の性能に相当します。

元岡 どうも有難うございました。ただ今のお話につきまして何か御質問ございますでしょうか。

このアレイプロセッサによって性能が230-75の10倍以上になるということがわかったんですが、素子の数が10倍になったんでは意味がないんで素子の数はどのくらいですか。

乾 素子の数はCPUの1.5倍ぐらいです。

元岡 専用プロセッサの設計として実装技術、回路技術には汎用機と同じ技術を使ったという点は、實際には非常に重要な点なんだろうと思いますけれども、論理設計レベルではこういうアレイプロセッサの設計はあまり難しくないというか、あんまり問題は起らないんですか。

乾 さきほど申し上げましたように、ソフト、ハードウェアとも私のやっているものはパイプライン型であ

る程度汎用プロセッサの延長上にあるということありますので、汎用プロセッサの経験というものは大いに生かせられます。また非常に細かいレベルでも演算機の一部を汎用機と同じものを使うということも可能であります。そういうことでマクロに申し上げますと、汎用プロセッサの技術と同じであると考えられます。

元岡 専用プロセッサを実現可能にした設計技術の進歩といったようなもの、いったいどういう進歩があったから専用プロセッサが可能になったかというようなことが、幾つかあるんだろうと思いませんが、同じメーカーの立場から服部さん、何か御意見ございませんでしょうか。

服部 いわゆる計算機の部品技術の進歩のほかに設計支援手段、自動化技術の蓄積あるいは定着がただ今のようなシステムを実現する一つの土台にはなってるんじゃないかなと思います。ただ、実現された専用プロセッサのシステムへの結合あるいは拡張とか変更に耐え得るように設計する問題は今後に残されていると思います。そのへんは、むしろ元岡先生のお考えを伺いたいところなんです。

元岡 時間もございませんので、またあと時間がありましたらそのへん少しお話することにいたしますて次に行きたいと思います。

次は専用プロセッサとそれに関連するソフトウェアの問題について、データ・ベース・マシンを開発しているらっしゃいます東芝の前川さんにお話を願いいたします。

前川 いただきました題が、“専用プロセッサとソフトウェア”ということで広げればきりのない大きなテーマですが二つにしほってお話をしたいと思います。

一つは、データ・ベース・マシンの現在の動向について、次にデータ・ベース・マシンに特にしほりまして、ソフトウェアとの関連について述べたいと思います。

データ・ベース・マシンの専用化の一番初めとしましてバット・エンド・コンピュータが導入されました。普通のホスト・コンピュータにデータ・ベースの処理をいたします別のプロセッサを接続し従来1つの計算機で行っておりましたうちのデータの処理を別の計算機へ取り出したものです。ベル研究所の XDMs で広くその考えが知られるようになりました。これを更に発展いたしまして、ネットワークに持ち込んだ例ですと、ARPA ネットでのデータ・ベース専用機と

してのデータ・コンピュータがあります。

分散を更に進めていく場合に、分散されたデータ・ベースの処理プロセッサを更に専用化して高性能を求めるという動きがありまして、特に顕著なのは、連想処理機能を導入しようという考え方です。そのようなシステムとして特に知られておりるのは、トロント大学でのラップ (RAP) システムと、フロリダ大学の CASSM があります。トロント大学の RAP の特徴は、トラック単位にプロセッサをつけまして、トラック単位の連想処理を行っていることです。この連想処理方式によりかなりの性能向上が期待されます（現在建設中）。

フロリダ大学の CASSM では同様に連想処理を基本としておりますが、データ構造として階層構造を基本としていること、及びプログラムがデータと一緒に二次記憶装置に入っていることが異なります。

次にデータ・ベース・マシンとソフトウェアとの関係を考えてみます。ここでは二つの点にしぼって考えたいと思います。一つはソフトウェアとハードウェアとのトレードオフ、もう一つは専用化した際のソフトウェアの問題、特にプロセッサ間のインターフェースとしての言語です。ソフトウェアとハードウェアのトレードオフとしては、データベースシステム実現法としまして通常の計算機にソフトウェアで実現する方式、それからファームウェアにより効率を上げる方式、更には完全な連想処理を利用したハードウェアを利用する方式、の三つの代表的なものを考えることができます。これらの方程式の利点と欠点を考えてみると、ソフトウェアの場合には通常汎用であるのに對し連想処理方式の場合には特殊なものになる傾向があります。まずデータ・ベース・マシンということによる特徴化、更にはその中のデータ・モデルによる特徴化が行われます。そういうことで連想処理方式の場合には特徴化が決定的になります。次にソフトウェア方式の場合には今のところコスト・パフォーマンスが一般に良いということがあります。価格もより低い。そういう点がソフトウェア方式の利点ではないかと思います。

一方、連想処理方式の場合だと、大きな利点としてはまず高性能があります。これは先程のアレイプロセッサと似たような方向だと思われます。価格は高くなるけれどもとにかく高性能の機械が必要だという場合には有効な方法です。しかし、連想処理は、検索において条件にマッチするレコードの数がかなり多い場合に有効で、そうでない場合は今のところあまり高性能を望めないと注意する必要があります。これは検索時間が連想処理では加算的であるのに対し、そうでない場合には乗算的であることによります。連想処理方式の次の大きな利点はソフトウェアの簡単化ということです。ソフトウェアで処理をする場合には効率を上げるために非常に工夫をしなくてはいけない。そのためデータ・ベース・システムのソフトウェアが極めて複雑になり、その開発が非常に大きな問題になります。ところが連想処理方式だと、データ構造を非常に簡単にとることができますので、処理が非常に簡単なところに利点があります。

一方、連想処理方式の欠点としては、現在の技術では容量的に十分な大きなものを作ることが非常に困難であるという問題があります。このため、ディスクに対するキャッシュを使うという考え方があります。そのような考え方で開発が行われている例も多いようです。以上がソフトウェアとハードウェア方式の比較です。ファームウェア方式の利点・欠点は両者の中間でソフトウェア寄りと考えられます。

有澤(横浜国大) 2つの点についておたずねします。

1つはデータ・ベース・マシンというものを作るときやはりデータの並列処理が大きなメリットになると思いますがその点について、次に並列処理を行うとした場合に、ソフトウェアというものは必ずしも簡単なものにならないのではないかという点について、おたずねします。

前川 原理的に申しましてデータ処理のためには並列処理が重要で、できるだけ並列処理を行うべきだと思います。しかしその方法が問題で現在又近い将来においてはその原理を直接ハードウェアで生かすのには問題が多いと思います。

次にソフトウェアが簡単になるかという問題については、既に述べましたように性能向上のための工夫が必要となる部分が多くそれだけ簡単になるといえます。

元岡 まだ御意見もあると思いますが時間が大分すぎましたので、次に進ましていただきたいと思います。最後に専用プロセッサと保守の問題について端末関係のプロセッサをやっていらっしゃいます服部さんにお話ねがいます。

服部 専用プロセッサの保守の考え方としては、二つの場合があると思います。中・大型のホスト系でその専用プロセッサを考える場合、端末系での場合ですが、ここでは端末系を中心に考えてみたいと思いま

す。端末系の保守の場合、種類、台数が多いこと、地理的に分散して設置される、などが大きな特徴となります。そのほかにいくつか考えられます。システムの変更拡張時には、端末系では比較的モデルのグレードアップが少ないと、ハードウェア構成の変更拡張というのは割合多いということです。また端末単位の増設が多いこと、さらに、ホスト系の場合と同じようにソフトウェアに関する保守コストの増加があります。またホストシステムとの結合性も端末系の保守に影響する大きな問題です。

一方、障害時に関する特徴では、1つは装置の重量が比較的軽いため装置交換の可能性もあることです。それから自己診断機能の充実、これは最近のLSI化による(論理)部品の減少や相対的な分解能の向上によります。この真にはマイクロ診断の進歩や入出力デバイスのオンライン、オフラインのテスト(携帯用)シミュレータの利用などがあるわけです。

次に端末系の保守に影響を与えると思われるリモート保守についてふれてみたいと思います。

一つは遠隔地からの保守に関するシステム管理があり、これにはリモートでシステムを診断中段・テストする問題、システムの導入・変更・拡張・移行時のシステム管理の問題が含まれます。それから遠隔地からの保守情報を管理すること、これには遠隔地からの情報提供によって保守のためのエンジニアを支援する。あるいは実際の現場からの問い合わせに対して遠隔地に控えている専門家が正確な情報を与えるとかにより保守支援していくことが考えられます。これらのリモート保守につきましては両方とも既に実用化段階にあるわけですけれども、本格的な利用は今後のことではないかと思います。それで端末系のリモート保守の特徴ですけれども、システムがあまり複雑でないということ、これはリモート保守を、リモート診断に限れば、ある意味で実現しやすくしています。またシステム価格が低いこと、例えばホスト系の保守に関しては、現在メンテナンス・プロセッサあるいはサービス・プロセッサというようなものをホスト・プロセッサ系に結合して、保守センターから保守することが考えられるわけですが、端末系ではシステム価格が比較的低いために、そのような方式を一般に採用しにくいと言えると思います。

それからシステム機能が相対的に低いこと、通信機能がない場合は別として、例えばホスト系のOSに相当するレベルのシステム機能がない場合や、端末系に

エラー・ログを持つことは少ないなどむしろリモート保守を促進する要因もあります。一方システム構造は必ずしもリモート診断に適しているとは限らず、例えば二重構造、あるいは階層型のシステム構造を持っていることは少ないと言えます。それから一つのホスト・システムに対して、複数メーカの複数種、複数台の端末系が結合することは、一つのホスト・システムから端末系を保守しにくくし、逆に保守センター側から端末系のリモート保守を行うのを促進させる要因になると思われます。さらに保守用の個々のユーザ側組織が非常に小さいということもむしろリモート保守が歓迎される理由になります。この他に、端末系の特徴としましてそれらが分散しているということは当然のことながら、リモート保守に有利な点になります。

以上、端末系の専用プロセッサの保守に関する課題といたしましては、保守以前の段階、即ちシステム設計段階で保守機能・性能／コストのよい(1つの専用プロセッサではなくて)専用プロセッサ系を設計する問題と、純粋な保守技術上の問題で、自己保守あるいはローカルな保守(機能)とリモート保守(機能)とのトレードオフ、などがあげられると思います。

元岡 どうも有難うございました。何か御質問ございませんか。

都丸 一番最初の元岡先生のお話の中で、専用プロセッサが出てきた技術的な背景として保守や設計技術の進歩があるというご指摘がありました。ただ今の服部さんのお話でも具体的な幾つかの例があったんですが、専用プロセッサというものは、単独で使用する専用プロセッサよりも、むしろ機能分散のほうが、効果がより大きいと思います。その場合全てのプロセッサを全結合した形の試験というのは決してやさしいものではないと思います。それからもう一点ですが、専用プロセッサは機能分散が大きなねらいであるとするならば、当然のことありますが、それぞれの専用プロセッサはシステムの構成要素としての部品化が行われるべきであろうと思います。部品化というのは一つの全体のシステムの機能を統一的な考え方で分けて設計するんではなくて、あちこちで作られた専用プロセッサを持ってきて適当に組み合せればそれでいいものができますということになります。そういうふうに作るためにまだ技術が進まなければならぬのですが、今の服部さんのお話にもございました、インターフェースを標準化するということが大事なことであって、そういうことができないと専用プロセッサのはん

とのメリットというのは生かせないのではないかという気がいたします。ここらへんのところを、先生の御意見をお伺いいたしたいと思います。

元岡 それはまさに電電公社にとっては一番大事な問題であります。ただシステム全体がうまく動いているかどうかというのは確かに難しい点があると思います。

それから正しく動くシステムになってるかどうかということをどうやって保証するのかということもこれまた難しい問題だと思うんですが、ただその前提として個々のプロセッサが正しく動いているかどうかということについてはある程度切り離して試験ができるんじゃないかと思います。今日のところはシステムの話とそれから単独のプロセッサの話と両方ませちゃいますとますます話がダイバージするので今日のところはプロセッサに限っております。その点御容赦いただきたいと思います。

それでは最後に締めくくらせていただきますが、いろいろな種類の専用プロセッサについてお話をいただきまして、私が得た感想は、実際に商用ベースで本格的に実用化されているというのは端末関係のインテリジェント・ターミナルに見られるようなコスト・パフォーマンスの問題を克服したというか、コスト・パフォーマンスの面で優れているからという格好で専用プロセッサが使われている場合ではないかと思います。

一方、アレイプロセッサ、データ・ベース・マシンと

か、高級言語プロセッサもそうですが、いわゆる高性能を指向した専用プロセッサがあり、この場合はコストの問題は二つの次になります。それからもう一つ実際によく使われているものとして、通信制御プロセッサがあるわけで、この場合はコスト・パフォーマンスの点でもそのほうが有利である。それと同時に柔軟性を専用プロセッサに求めしていくというお話をありました。専用プロセッサと柔軟性というのは何か相矛盾する概念のような感じがしてるんですが、これはいろいろな多様化する用途に対して、汎用プロセッサでは対処できないんだと、そういう意味で専用化が必要なんだということであろうと思います。そういう意味で今後計算機の分野の用途が広がるに従いまして、従来の汎用計算機ではカバーできないような新しい分野というのがどんどん多様化して出てくると思います。現在のところ専用プロセッサという言葉を使っていますが、例えばオフィス・マシンみたいに、非常に台数が多くなると、あれは専用プロセッサというべきかどうかというのは疑問なわけです。すべてのコンピュータが同一のアーキテクチャということではない、いろいろな種類のアーキテクチャのプロセッサが今後伸びていくんだろうということについてはみなさんの御意見が一致しているのではないかと思います。

まとまりのない結論ですが、これで締めくくらせていただきたいと思います。どうも長い間御清聴有難うございました。（拍手）

（おわり）