

コンピュータ・アーキテクチャについて*

山 本 卓 真**

1. はじめに

コンピュータ・アーキテクチャについてという題目で、ここでは、現在までのアーキテクチャの変遷から学ぶべきこと、LSI 技術の進歩等の要因による今後のアーキテクチャの方向を話題の中心とします。

アーキテクチャという言葉は、IBM 社のシステム 360 が生まれたときに、IBM 社で使用されはじめました。簡単に言えば、プログラム側から見たコンピュータの性格であるという定義です。しかし、このように定義しますと、例えば、現在大型計算機で広く用いられているキャッシュ・メモリ（バッファ・メモリ、ローカル・メモリとも呼ばれる）は、必ずしもインストラクションとか feature とか function という意味ではプログラムから見えないにもかかわらず計算機全体の動きには大きな影響を持っています。ある種のプログラムでは、キャッシュ・メモリをつけてもあまり能力が上がらないのに対し、例えば、科学技術計算の小さなループを描くものでは、キャッシュ・メモリの効果は絶大です。このように、狭義のプログラムからアーキテクチャをとらえるのは的確ではないようです。そこで、ここでは、より広い意味でアーキテクチャを考えて見たいと思います。前に、キャッシュの問題を上げましたが、マルチプロセッサについても、現在広く使用されている対称形のプロセッサ構成では、少なくとも処理プログラムあるいは使用者プログラムからはプロセッサの構成は見えないのに対して、ターンアラウンド・タイム、スループットの点ではプロセッサの構成には大きな関心が持たれております。

2. アーキテクチャの変遷

次に、アーキテクチャについてどのような変遷があったかを図-1 をもとにして考えて見ましょう。50 年代から 60 年代にかけて、キャラクタ・マシン、36

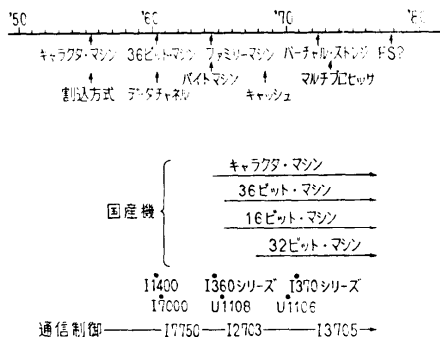


図-1 アーキテクチャの変遷

ビット・マシーンが現われ、キャラクタ・マシーン／ワード・マシンの議論が盛んでありました。これに続き 64 年には 360 シリーズで、シリーズとかファミリーといわれる一貫したシステムと、バイト（8ビット）を単位とする情報処理が固まってきました。70 年には IBM 社は 370 シリーズを発表して 360 シリーズに機能の追加を行ない、72 年に仮想記憶（Virtual Storage）の考え方をだし、73 年にはマルチプロセッサを発表しました。国産機に於いても同様な変遷が見られます。このようにアーキテクチャの変遷を見ているといくつかの経験的に学ぶべきことがあるようです。

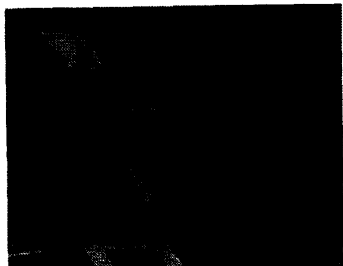
第一はアーキテクチャに絶対的なものはないのではないかということです。例えば、計算機の内部での処理の単位でも、32 ビット単位が絶対的であるならば全部 32 ビットになったでありません。しかし、現実には 16 ビット、36 ビットの計算機も広く世に受け入れられています。結論的には、単に細かい議論ではきまらず、販売力を含めた工業力が決定的な物を言うということが一般的でありまして、これだけでなくはないということはないようです。

次に互換性の要求が強くなっているということにお気づきになると思います。これは、例えば、IBM 社をとった場合、IBM 360 から 370 に 1970 年代の初

* 情報処理学会創立 15 周年記念大会招待講演（昭和 49 年 12 月 5 日）

** 本会理事、富士通（株）電子事業本部長

写真

で講演中の筆者
創立15周年記念大会

期に移っておりますが、この間で大きな変化が出来なかったことでも分かります。また、国産機についても新シリーズを出しておりますが、過去のソフトウェアの財産を尊重する意味から、新しいアーキテクチャを思い切って出すことは出来ないか、あるいはかなり思いきって変えるとしても、過去のソフトウェアをなんらかの意味で救済する手段を用意しなくてはならない状態にあります。

次に言えることは、超小型機から超大型機まで完全に一貫したアーキテクチャで貫くというのは理想ではありますが、これに成功したメーカーは一つもないと言えらると思います。例えば、360の時代に既に360-20のところの断絶があります。また、それ以前に開発した1800、1130という計算機が根強く残っております。また、私共でも230シリーズを出しましたが、270シリーズは今日でも広く使用されております。これらは、当然のことであります。建築の方でも、小さな家から大殿堂に至るまでを一つの論理で通すことに無理があります。現在予想されることは、例えば超大型計算機分野では、汎用計算機の他に、例えばアレー／ベクトルを計算するアレー・プロセッサ、更に次の将来には画像処理等の分野で新たなアーキテクチャを持った別のプロセッサが枝分かれして行く可能性があります。他の一つの傾向は通信制御装置の分野に於て見られます。IBM社では、360以前にプログラム制御の7750トランスミッション・コントロールを出しました。その後360を発表した時点では、通信制御を360で全て包含すると言っておりましたが、最近ではまた新しい通信制御プロセッサを発表しております。通信制御装置はやっかいな性格のものであります。CPU側からと、端末／線路側からのやっかいな問題をしわよせさせる傾向があります。また、多数の回線を制御するため論理自身が非常に多重動作のものであること、回線または端末の種類が非常に多く伝送

制御の手順も歴史的な重みがあり非常に多いということで、ストアード・プログラム計算機を専用に用いる方法がとられ、次に汎用計算機のプログラムで十分やれるのだということになり、更に最近では新たな形で通信制御プロセッサが現われてきました。この理由は、コンピュータ・ネットワークが非常に進歩してきたことにより新たなネットワーク・アーキテクチャが必要となることによります。簡単に言いますと「歴史は繰り返す」と言えます。

国産機も種々の端末、通信回線、制御手順、ネットワークの変遷を眺め、かつ過去の遺産を捨てずに次の進歩を求めるには、通信制御プロセッサは現時点では間違いない正解であろうと思います。

3. アーキテクチャの動向

3.1 使用者の要求

次に今後のアーキテクチャについて概観したいと思います。アーキテクチャにどのような要素が作用するかを考えると第1番目は、使用者の要求によるもので、コスト／パフォーマンス、互換性、使い易さ、信頼性等があります。

(1) コスト／パフォーマンス

どのような命令を用意すれば、速度が向上するかについては、過去にいろいろ検討され、現在でも部分的に検討されております。図-2は、U200計算機と米国のミニコンピュータとの比較ですが、命令の質の良さがプログラムのステップ数、所要メモリ量、処理速度に影響を与えることが分かります。

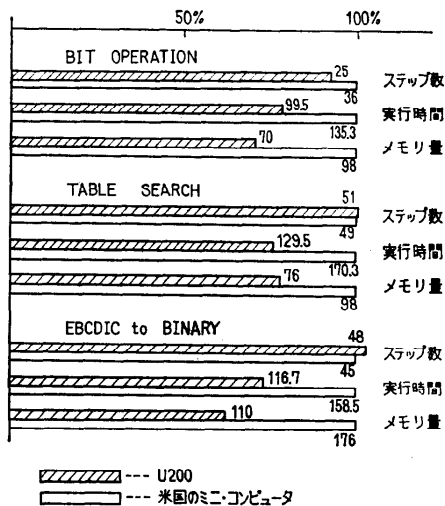


図-2 ミニコンピュータの性能比較

(2) 互換性

現在では、新しいアーキテクチャに急転換するよりも、新しい機能を拡大する方向が互換性の面から要請されていると思います。例えば、49年11月に発表しましたMシリーズ計算機も、新しい機能の追加に力を入れております。

一方、我々が注意しなければならないのは、単に過去のものとの互換性ばかりでなく、新しい要求、すなわち国際的な互換性が必要になってきています。これは問題を困難にしており、ある意味では、我々は我を捨てなければならないという局面にきていると思います。大学、研究所では(特に米国との間で)ソフトウェア財産を交換することは既に始まっておりますが、このような場合には、国産計算機は最小限、言語仕様では国際性を持たなくてはならない。これは高級言語の場合はよいのですが、更にアセンブラの段階になりますと非常に困難をとめない、アーキテクチャを大きく制約することになります。

(3) 使い易さ

使い易さ、ソフトウェアのプロダクティビティの面でアーキテクチャを変える努力がなされています。この点でBurroughs社の努力は見るべきものがあります。中間言語プロセッサのアプローチをとりメモリ所要量を減少しかつ簡便な利用をねらっています。見るべき面はあるのですが、現在は最終的なコスト/パフォーマンスについてすべてのユーザに迎え入れられているかどうかは論議が分れる所であります。

(4) 信頼度

計算機が公企業体に於てなくてはならぬものになって来ておりますが、計算機も電話交換機と同等のタフなシステムになることが要求されてきます。このインパクトを受けてアーキテクチャに変革が予想されます。例えば、予備機を組み込む、エラー訂正コードを採用するのは当然であるというように変わって行くでありません。

3.2 L S I

更に大きなインパクトを与えるものはLSIです。以後、LSIについて述べて見たいと思います。

まず、LSIのはしりであるMシリーズの計算機のLSIの実例についてですが、写真-1にM190超大型計算機のLSIを示します。この内部の3mm平方の面積に約100ゲートが収容され、これは従来のプリント板1枚分に相当します。1ゲート当たりの遅延時間はICチップ内では0.7~0.8nsです。この3mm平方



写真-1 M190超大型計算機のLSI

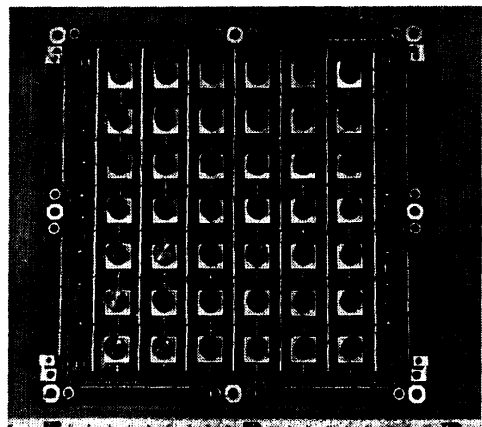


写真-2 M190超大型計算機用のLSIを42個実装したパッケージ

の内で実用状態で3ワット近辺の発熱があります。このようなLSIでは論理が高密度になり高速化につながります。しかし、LSIの内での遅延が0.7~0.8nsであるのに対して、システム全体としては1ゲート当たり2ns程度で動くことになります。残りの1.3~1.2nsがLSI外の相互配線等に於ける時間です。今後、計算機を高速にするにはパッケージング、実装の技術の進歩が必要となります。Mシリーズでは写真-2に示す190mm平方のパッケージに42個のLSIが実装されます。現在のM190では、このパッケージを41枚使用しています。

計算機を高速にする目的のLSIの方向について考えてみますと、ICメモリは図-3(次頁参照)に示すように、年に1.5~2倍の割合で集積度が向上することが予想されています。また、ランダム・ロジックについても図-4(次頁参照)に示すような予想がされています。しかし、最近分かって来まして重大だと思っていることは、必ずしもこの傾向のように行かないの

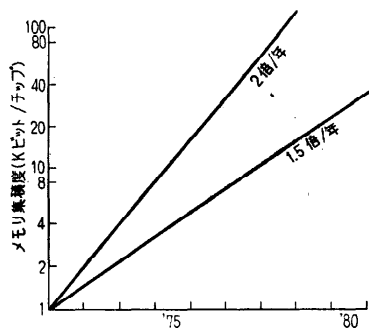


図-3 IC メモリ集積度の推移

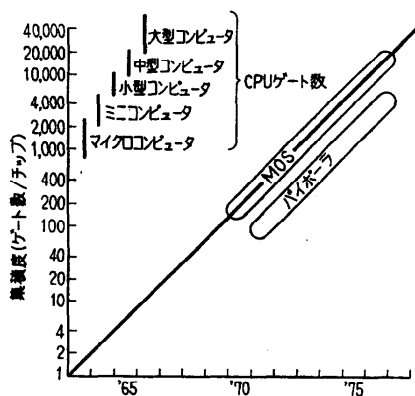


図-4 チップ当りゲート数の推移 ランダムロジック (開発時期)

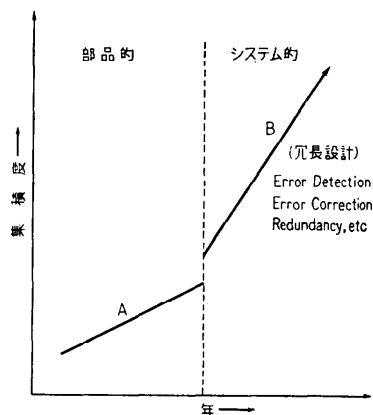


図-5 高集積度の要求動向

ではないかということです。図-5 は、想像図であり若干誇張して描いております。従来の部品のアプローチで IC デバイスが年間 1.5~2 倍集積度が增加するのは、図の A の部分の線上にあるのですが、1 チッ

プに 1k ビット、4k ビットのメモリが実現されていれば、ここはもはや部品と呼ぶよりシステムと呼んだ方がよくなります。そこで、このようなシステムの歩留まりを上げて安価にし、また、実用中のシステム信頼度を上げるためには、既にありますようにエラー検出コード、エラー訂正コードを利用する方法の他に、半導体の内に予備機を組み込む、現用と予備との切り換えのスイッチを組み込むようなアプローチがとられると予想されます。

このような意味で、図の B の部分の様に勾配が急になることと考えられます。そこで、従来のアプローチを私は部品のアプローチとしてとらえ、今後はシステムのアプローチとしてとらえなくてはならないのではないかと考えております。そうした意味で計算機は現在の半導体屋の仕事と計算機屋の仕事が接合する非常に重大なポイントに来ております。そのような意味で、図中の断層が現実にあるかどうか分からないのですが、あえて誇張して書いたわけです。そして、このような変革期にあることを我々は認識しなくてはならないと思っています。

例えば、メモリをとった場合、32 ビット中の 1 ビットのエラー訂正をする、2 ビットまではエラー検出をすることは当たり前のこととして行なっております。今後は、更に多くの誤りを訂正することが考えられます。これは単に実用になってからではなく、生産過程に於ても結晶の欠陥等をどこまで認め、運用上はその上にどこまでのエラーの訂正を認めるかといった限界の引き方、その予想、冗長性の持たせ方等の研究が必要となります。冗長度の持たせ方に例をとりますと、計算機のメモリではハミング・コードがよく使用されますが、冗長度を増加して行くときこのコードが最適であるのか、冗長度最小をねらうのではなく論理構成の面でエラー検出、エラー訂正のスピードの速いことをねらうのか等研究すべき課題が理論的と実際面でかなりあると思います。以上、LSI そのものの予想をしましたが、LSI が計算機のアーキテクチャにどのようなインパクトを与えるであろうかを考えて見ます。

3.3 ポリプロセッサ

LSI 化は高速に、安価にという方向でありました。しかし、集積度が急激に増加して来ますと、よほど多数個作らないと安くならないという矛盾が生じます。そこで、1つの計算機システム中のプロセッサの数を増さなければならない。これは、ある場合には同時に

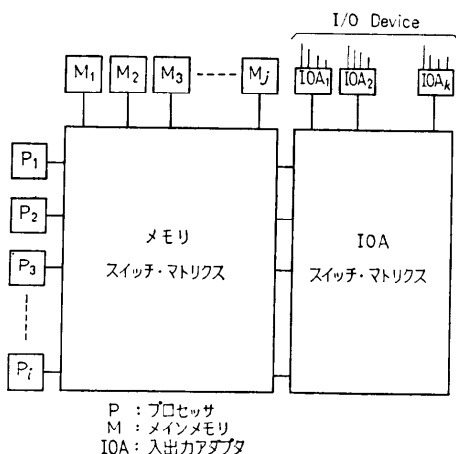


図-6 ポリプロセッサ・システムの例

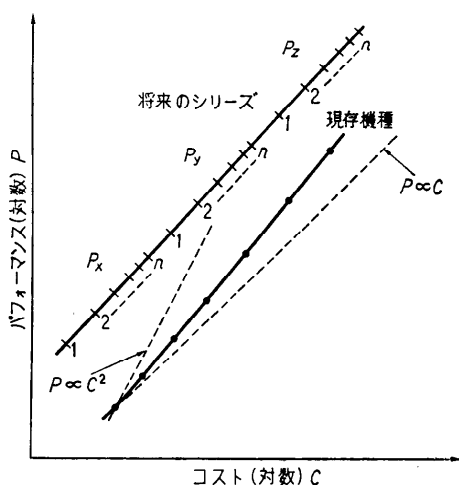


図-7 将来の計算機のシリーズ構成

処理するジョブ数、ジョブの分担実行という意味で使用者に受け入れられる可能性が十分にあると思います。図-6はプロセッサ $1 \sim i$ 、メモリ $1 \sim j$ をマトリクス・スイッチで切り換えて使用する方法を示しています。また、現在は行なわれておりませんが、インストラクション・プロセッサ、I/Oプロセッサ、サービス・プロセッサのアーキテクチャを共通にして、これを大量生産して安価にするアプローチがあります。

一つの計算機として、ポリプロセッサ・システム構成がとられるであろうと述べましたが、更に、シリーズとしてはどのようなになるであろうかを考えて見ます。グロッシュの法則で性能は、コストの2乗に比例するとありますが、現存の計算機は図-7に示すよう

に2乗と1乗との中間に位置しています。そしてモデル 10, 20, 30, 40, 50, 60 のようにシリーズを構成しています。モデル間の性能比は通常2~3倍に設定しています。例えば IBM 社のシリーズは 115 から 168 まで6種類あります。但し、158 と 168 とはマルチプロセッサ構成があります。

次の LSI 化の時代には図-7 に示すように P_x , P_y , P_z のようなプロセッサが出来、 P_x は1から n 台のプロセッサまで用意することで性能を上げる。その上は、 P_y プロセッサを1から n 台使用する等の形式で使います。 P_x , P_y , P_z について n を等しくする必要はありませんが、主としてソフトウェアの互換性から、大体同じ値に選ぶでまいしょう。このようなポリプロセッサ構成は LSI の使用数を増加したいという工業側からの必然の方向であります。使用側から見ても十分利点があるものであります。計算機の性能をレベル・アップするときに2~3倍の性能アップにかかわらず、CPU の入れ換えの大仕事を行なうことはせずに、LSI 化された小さな CPU を明日にでも簡単に増設できるなどの利点があります。

性能は2台にしたから2倍になるものではなく、 n 台にしたから n 倍になるのではないのですが、ほぼそれに近い値になります。私共の経験でも、大学のセンタ等で2台で1.7という値を得ております。特殊な場合には1.96という値もあります。通常は1.7~1.8近辺の値が予想されています。これを3台以上にしたらどうなるについては、ハードウェアの論理の組み方、ソフトウェアの共通領域のロックの方法などによるので一概には言えませんが、これ以上台数を増加して意味がないという限界は $n=5 \sim 6$ より上になるものと思われます。これによって機種が減りますと、予備品の問題、フィールド・エンジニアのサービスの問題が非常に容易になります。今、 $n=5$ と仮定しますと、5台のうち1台が故障して4台となっても全体の性能は簡単にいって2割程度しか落ちないことになります。この利点は、現場で働くフィールド・エンジニアだけでなく、使用者側にも十分認められるものでまいしょう。ここで注意しなければならない技術的な問題が多くあります。例えば、ポリプロセッサ・システムでプロセッサ P_1 , P_2 を使用している時、 P_3 以降を増設するには、 P_1 , P_2 を生かしたまま P_3 以降を増設することが要望されます。これは、現在の通信システムに於て新局の開設のときに採られる手法と同じですが、きゃしゃな半導体を相手に切り換えを行なうという技

術的な問題を持っています。

3.4 メモリ

次にメモリですが、メモリは現在 4k ビット／チップの時代が始まりつつあります。ポリプロセッサ構成でプロセッサ数が増加すれば、当然タスクの多重度が上がり、これにともなってメモリ量の要求が増加します。このメモリ量の増加に対応するアーキテクチャを考える必要があると思います。現在の計算機は、世界的に見て 16 M バイト近辺がメモリ量の限界ですが、次のステップではこの限界を越さなくてはならないと思います。

3.5 ファームウェア

工業として見た場合、インストラクション・プロセッサ、I/O プロセッサ、通信制御プロセッサ等を同一のアーキテクチャのプロセッサを用い、ファームウェアによって性格を使いわけるとことは一つの理想であります。現実にはなかなか困難であります。インストラクション・プロセッサと I/O プロセッサをとっても主要な機能がかかなりずれています。次のステップではそれを更に細分化してインストラクション・プロセッサと I/O プロセッサの共通部分を見つけるような研究が必要となります。通信制御プロセッサについてはこれをインストラクション・プロセッサと同一アーキテクチャにすることはコスト／パフォーマンスの点から I/O プロセッサ以上の困難が予想されます。

次にインストラクション・プロセッサに於けるファームウェア化の可能性について考えて見ます。図-7 の P_2 に於けるプロセッサが最高のテクノロジーで出来ていると仮定します。 n 個のプロセッサを使用したポリプロセッサが実現したと仮定しますと、性能の面で社会の需要を上回ることにようになりますが、この時点がファームウェア化する絶好の機会であります。従来のアーキテクチャに制約されて新しいアーキテクチャをなかなか出しにくいということを前に述べましたが、使用し易さ、メモリ量の拡大、各種の処理効率の改善など多くの課題を解決するとすればやはりこの時点であると思います。結論的には、ファームウェア化は技術的にも社会的にも十分その可能性があり、従来 Burroughs 社が開発して来た中間言語プロセッサのアプローチ等が生かされる余力が出ると思います。ただし、現時点では特に大型機の分野ではマイクロプログラムの手順を踏めば踏むほど、コスト／パフォーマンスは悪くなるということから商業的には成功したもの

表-1 ファイルの性能

磁気ディスク	容 量 (MB)	平均アクセス (ms)	容量×アクセス時間 (MB・ms)
I 2311/F 462	7.25	87.5	635
I 2314/F 472 L	29	72.5	2103
I 3330/F 477 L	100	38.3	3830
I 3340/NP-20	70	35.1	2457

はないようです。

3.6 ファイル・システム

ファイル・システムのアーキテクチャについて考えてみましょう。ファイルは元来その容量が2倍になれば、アクセス速度は2倍にならなければならない。ディスクを例にとっても、5枚の面を持つディスクの容量を2倍にして10枚の面にしたとしますと、従来 100 ms のアクセス時間であれば、2倍にしたときは 50 ms にしなければならなくなります。これは当然の原理であります。例えば、銀行の預金システムで1バックに1支店の口座（例えば1万口座）が納められているとします。これがファイルの性能が上って容量が2倍になった場合、2支店分のデータが納められるので、2支店分のアクセスがこのバックに集中する。即ち、アクセス速度が倍になる必要があります。そこで、現在までに開発されたファイルは、容量とアクセス速度とが比例しているのでしょうか。これを表-1に示しました。アクセス速度をアクセス時間に置き換えますと、容量×アクセス時間は定数になるはずですが、表のように定数にはならず年々増大の傾向になっていました。これは容量が増加する分だけアクセス速度が向上していないことによります。今後、オンライン・システム、コンピュータ・ネットワークが増加して行くのにもなってファイル・システムが重要になって来ます。

2311, F 462 の時代にはファイルが非常に高価であったため、非常に濃縮した情報をファイルに格納していました。現在では、ファイルのコスト／パフォーマンスが向上し使用者としての余裕が生まれて来ました。

3330, F 477 ではアクセス時間の向上の目的でブロック・マルチプレクサ・チャンネルと Rotational Position Sensing 等の方法が採用されております。ところで、ファイル・システムの原則を認識する一方、ディスク装置としては大変な矛盾（大きなものを速く動かすという矛盾）を認識しておれば、当然のこととしてパラレル・アクセス、バスの時分割多重使用等を思い付き、その開発を我々日本人の手で出来たのではな

いかと思いますが、残念ながら実際はそうはなっていません。ファイルは大きなシステムの中で重要な役割をしており、今後の技術の方向として当然考えつくべき解法があるのではないかと考えております。

ファイルの信頼性については、現在でも重要なファイルについて2重化などの対策がとられています。また、大学計算システムでもファイルには磁気テープとディスクとの階層構造をなす情報の記録を持つことが当然になりましょう。

4. おわりに

以上、コンピュータ・アーキテクチャについて近い将来に何が起るかについて非常に簡単に述べました。

LSI を例にとりましても、大口径のウェハーを作る

こと、極力欠陥の少ないシリコン結晶を作ること、電子露光、精密なプロセスの制御、Computer Aided Design 等が必要になります。またランダム・ロジックは、ロジックそのものがシリコン上に展開されるといってもよいと思います。前に述べましたように、現在では LSI 自身のトランジスタ等の遅延は 40% 程度を占め、あとの 60% はパッケージと布線の遅延であることから、これらを総合的に考える必要があります。従来の計算機の開発は、半導体、回路、実装、生産、テストのようにどちらかというと直列的な流れ作業であったが、今後は計算機の開発をシステム設計としてとらえ、半導体開発の担当者をも含めてシステム開発としてとらえなくてはならないでしょう。まず私共自身にこのような変革が来ているように思います。