マイクロコンピュータと
ワークステーション 54－3
（１９８９．２．２１）

ＲＩＳＣアーキテクチャとＳＰＡＲＣの将来

村野　雄一
日本サンマイクロシステムズ（株）マーケティング本部

より優れたマイクロプロセッサを設計するには，２つのアプローチが考えられる。一つは既存のアーキテクチャのままで，ＣＰＵのクロックを高速化する方法であり，これは回路幅の縮小と高速で作動することのできる半導体技術を取り入れる事である。もう一つの方法は全く新しいアーキテクチャを開発する事により，ＣＰＵ自身の性能の向上を図る事である。　ＳＰＡＲＣは現在マイクロプロセッサの主流技術であるＣＩＳＣアーキテクチャに対し，ＲＩＳＣアーキテクチャを採用し，内部アーキテクチャを簡素化する事により９０年代の最新の半導体技術を取り入れるべく設計されたＣＰＵである。

RISC ARCHITECTURE AND FUTURE OF SPARC

YUICHI MURANO
NIHON SUN MICROSYSTEMS K.K.

Designing the better microprocessor can be considered two approaches, which is turn up the clock speed of existing CPU, that means employee the technology to condence transisters into smaller scale and adopt new semiconductor technology, such as ECL and GaAs. the other approach is to developee new CPU architecture to improve performance, such as RISC.
SPARC (Scalable Processor ARChitecture) employee RISC architecture versus existing CISC architecture, and be simplify internal design to be able to adopt new semiconderction technology which will be available in 1990's.

マイクロプロセッサーのもたらした最大の功績はコンピュータの高性能化と飛躍的なコストの低減である。現在主流となりつつある32bit CPUは10年前までは数億円クラスのコンピュータと同等の性能をもつ機器を数百万円のレベルまで低下させてしまった。現在90年代をめざした各種のコンピュータ技術の中でRISC（縮小命令セット型コンピュータ）技術がその高性能と拡張性のため一つのアーキテクチャとして確立されつつある。

SPARC (Scalable Processor ARChitecture)にはRISCアーキテクチャが採用されており、単純かつ効率的な命令セットを定義している。これらの命令のほとんどが１サイクルで実行されるため、平均サイクル/命令(CPI)が大きく低減されている。また１サイクルでは完了しない命令(LOADやSTORE)も、必要最小限のサイクルで実行されるようになっている。
現在のSPARCは約５万のトランジスターで構成されるCMOSゲートアレイ版が使用されており、16.6MHZのクロックで10MIPSの性能が発揮される。これは約35万トランジスタで構成される68020マイクロプロセッサが同じ16.6MHZのクロックで２MIPSである事と比較すると、７分の１の回路構成で５倍の性能が出る事になる。
この簡素化されたアーキテクチャと１サイクルでの命令実行がSPARCの最大の特徴であるスケーラビリティの基本となっている。
回路幅の縮小と高速動作素子の採用はこの簡素なアーキテクチャによりCISCでは技術的に採用が困難な半導体技術をいち早く取り入れることを可能とする。すなわち半導体技術をゲートアレイから、CMOSカスタム、ECLそしてGaAsと移行する事により、より高速での動作が可能となる。

現在のSPARCはSF9010IUインテジャーチップ、SF9010FPフローティングコントローラチップ、W1164マルチプライアチップおよびW1165ALUチップから構成される。（図１）

図１　SPARC Implementation

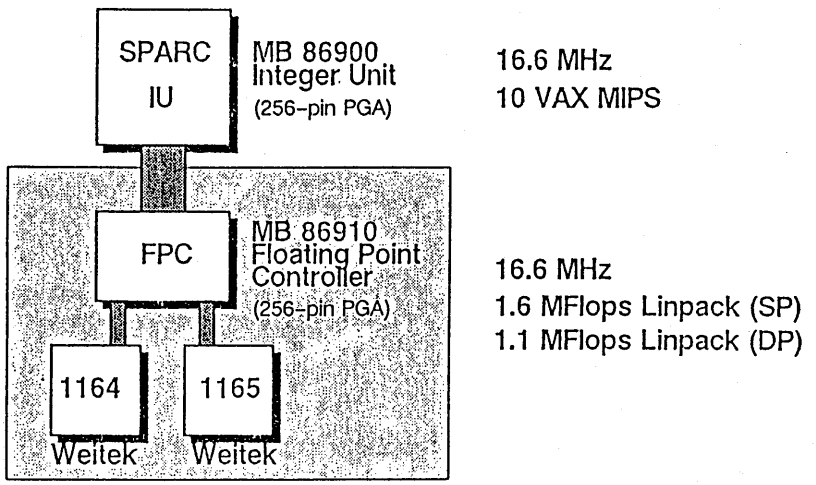

図 2

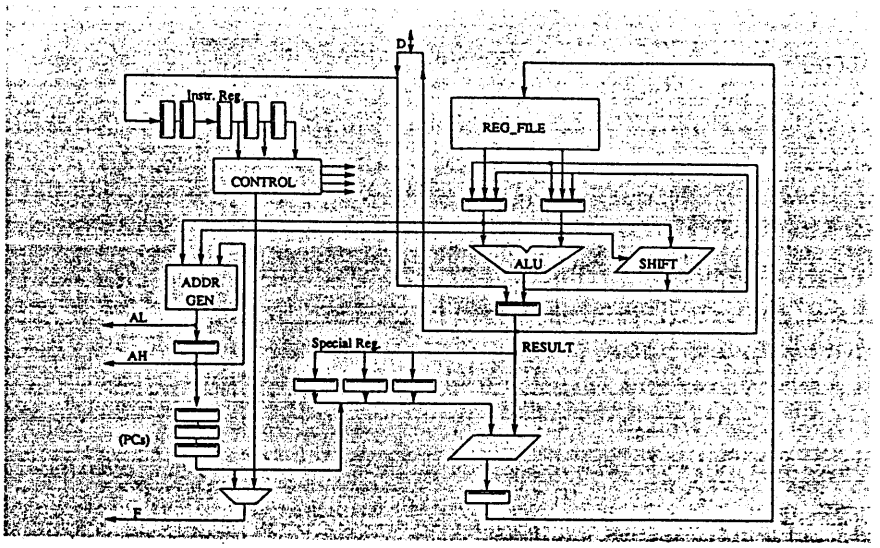

図２に示すようにインテジャーチップは４つの主要ユニットで構成されている。

レジスタファイルユニット: レジスタファイルユニットは32ビットの汎用レジスタを全部で120有している。これらのうち８個はグローバルレジスタであり残りはそれぞれ24個づつのレジスタからなる７つのオーバラップフレーム（ウインド）に分けられる。このチップでは７つのウインドしかインプリメントしていないが、実際のウインド数はインプリメントの方法に依存しており、将来のシステムではもっと多くなる事もある。プロセッサステータスレジスタのポインタ（カーレントウインドポインタ(CWP)）はこのレジスタファイルのカレントウインドを指す。

図 3

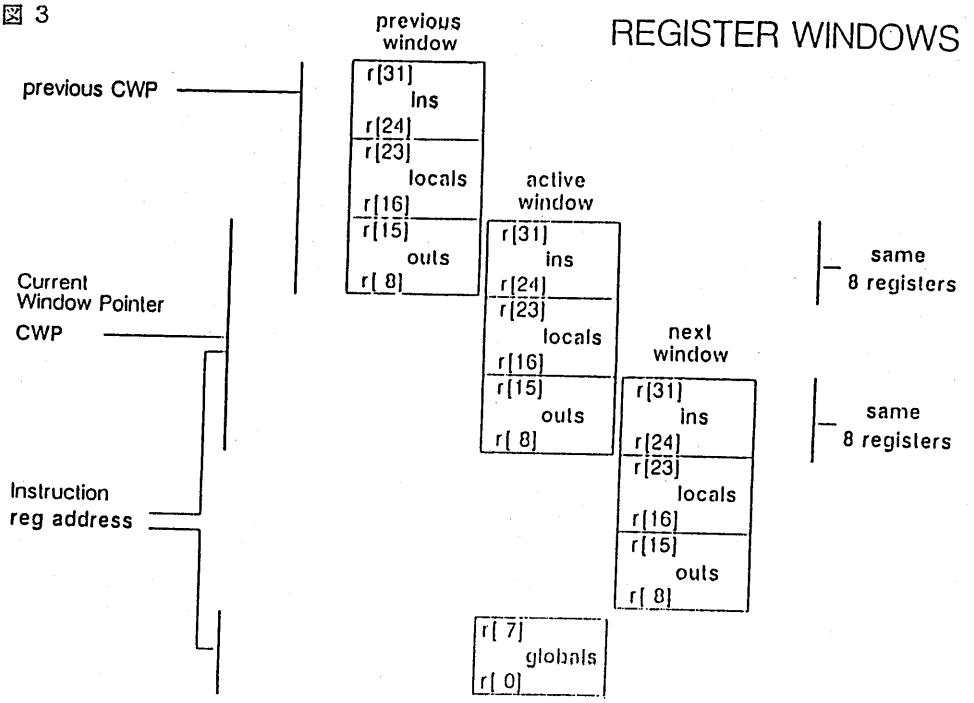

図３のように、隣接する２つのレジスタウインドでは、８つのレジスタがオーバラップしている。この様にウインドをオーバラップさせたのは、手続き呼び出しおよびリターンの最中にパラメータを効率的に引き渡せるようにするためである。通常、手続き呼び出し中には、呼び出し側は呼び出された側が使用しているウインド（次のウインド）とオーバラップしているレジスタにパラメータを入れる。そして、呼び出しの実行後、このウインドは変更され、呼び出された側が次のウインドのパラメータを直接アクセスするのである。レジスタファイルは２つの読み取りポートと１つの書き込みポートをもっている。命令の書式が均一であるため、どの命令についてもソース、オペランドを両方とも読み取り、書き込みポートを通して以前にフエッチした命令の結果をレジスタファイルに書き込むことができる。しかもこのプロセスは１サイクルで実行される。

実行ユニット：高速の32ビット、キャリールックアヘッドALU、32ビットバレルシフタ、条件コード生成ロジック、ロードアライメント/ストアアライメントロジック、およびオペランドと中間結果をセーブする関連パイプラインレジスタで構成されている。すべての計算/論理命令は１サイクルで実行される。
これはオペランドと連続した命令の実行結果の間に依存関係が存在する場合でも同じである。この処理を行わせるため、実行ユニットのデータバスには２つのバイパスバスが設けられている。最初のバイパスは、命令のソースオペランドがその前の命令の結果に依存している時に使われる。もう一つのバイパスは結果を格納したレジスタの出力をオペランドレジスタの入力として引き渡す。このバイパスはある命令のソースオペランドがパイプライン中の前の命令に先立つ命令の結果に依存している時に使用される。

命令フェッチユニット：命令フェッチユニットはプロセッサのプログラムカウンタと命令/データアドレス生成回路からなる。プログラムカウンタは４つあり、それぞれ命令パイプラインの４つのステージに対応している。これらのプログラムカウンタはプロセッサのパイプラインの最終ステージまでに例外が発生した時に使われる。

制御ユニット：制御ユニットは、プロセッサの制御の中心をなすユニットで主ステートマシン、命令パイプライン、命令デコーダ、プロセッサステータスレジスタ、例外/トラップ処理用回路、キャッシュ/フローティングユニットへのインターフェイスなどで構成される。

プロセッサパイプライン

SPARCプロセッサは４ステージからなるパイプラインを持っており、それぞれのステージで個々の命令の実行を完了させるために必要なオペレーションが実行される。そして１ステージで実行されるオペレーションはすべて１クロックサイクルで完了する。
フェッチステージ：命令のアドレスが読み出される。この命令はフェッチされた後、プロセッサのパイプラインに入る。

デコードステージ：命令がデコードされソースオペランドがレジスタファイルユニットから読み出される。このステージで読み出されたソースオペランドは実行ユニットと命令フェッチユニットの両方に送られ後に実行される。デコードステージでは次の命令のアドレスも生成される。つまり命令がデコードステージでデ

コードされている間に命令フェッチユニットでN+2の命令のアドレスが計算される。

実行ステージ：命令が実行ステージに入ると、実行ユニットは計算演算と論理演算を実行する。このステージの実行結果はテンポラリレジスタに一時保管されてから、レジスタファイルに書き込まれる。演算結果をレジスタファイルに書き込むかどうかの判断はパイプラインの最終ステージでなされる。

書き込みステージ：このステージではプロセッサは演算結果をレジスタファイルに書き込むかどうかの判定を行う。このステージでは命令実行中に例外が発生すると打ち切られる。

図４

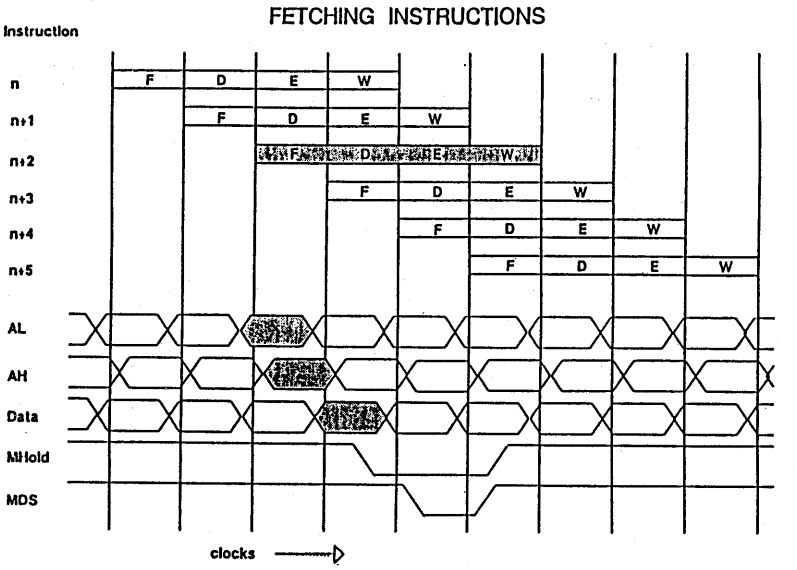

図４に示すように、プロセッサパイプラインでは４つの命令を同時に実行することができる。すなわち命令I(N)が実行されている間にデコードステージで命令I(N
-1)がデコードされる。実行ステージで命令I(N-2)が実行されている間に命令I(N-3)がその実行結果をレジスタファイルに書き込むと言う図式になる。実行中にいくつかの例外（トラップ）が発生した場合はトラップ処理ロジックにより最も高い優先順位のトラップを処理する。

パイプラインを常時効率よく利用するためデュアル命令バッファをもっている。このバッファは、マルチサイクル命令の実行中に命令をプリフェッチし、データバスを効率的に使用する事により高速化をはかっている。
プロセッサが一連のシングルサイクル命令を実行している時には、このバッファは空である。マルチサイクル命令がパイプラインに入ると、それ以降の命令は、そのマルチサイクル命令が完了するまでプリフェッチされて、このバッファに入る。

命令セット

SPARCではRISCアーキテクチャに基ずき単純な命令セットと定義している。ほとんどの命令は２種類の命令セットからなる。１つは、２つのレジスタがソースオペランドとなり、３つ目のレジスタがデスティネーションオペランドになる形式。最初のレジスタとイミディエート値がソースオペランドとなり、２番目のレジスタがデスティネーションオペランドになる形式である。
この様に簡略化された命令セットにより、ソースオペランドは命令のデコード処理により遅延なしに、レジスタファイルから即座に読み出すことができる。
この命令セットはサイクルタイムの低減の要素にもなっており、プロセッサの速度向上に貢献している。これらのオペランドすべてがレジスタファイル内に保持されていれば１サイクル／命令のピーク実行速度がほぼ実行できることになる。

命令のカテゴリは大きく５種類に分類される。最初のカテゴリは計算、論理、シフトからなる、２番目はステータスレジスタ、トラップレジスタに対して読み書きを行うもの、３番目はロード、ストア命令、４番目は各種のコンディショナルブランチ、そして５番目がフローティング命令とコプロセッサ命令からなる。命令の数は64種類（基本命令のみ）あり、すべて32bit固定長である。

SPARCチップの性能

現在のSPARCチップは富士通製のゲートアレイを使用したSF9010IUが実際に使用されている。プロセッサの性能は、サイクルタイム、平均サイクル数／命令、コード密度に大きく左右される。現在のSPARCチップである、SF9010IUにおいては最長の遅延パスを判断すべく十分な分析が行われた。プロセッサ全体は1.5ミクロンの２万ゲートのチップ上に実装されており、プロセッサの４分の１は大規模レジスタファイルで占められており、残りの４分の３がデータパスと制御回路に使用されている。
全体の設計は階層化が図られており、ブロック、サブブロック、そしてサブサブブロックというように区分されているため、チップ全体のルーティングが容易にできるとともに、レイアウト後のワイヤ遅延も容易に予測できる。
設計の目標である50NS以下の一般的なサイクルタイムを実現すべく、50NSを越える遅延を示すすべての内部あるいは外部パスを改善または再設計する方法を行い、最悪のケースでサイクルタイムは約60NSとなり、16.6MHZのクロックの周波数が可能となった。

SF9010IUはほとんどの命令は１サイクルで実行される。２サイクル以上必要とする命令はごくわずかしかない。この例外命令は下記に示すものとなる。

命令タイプ	サイクル数
ロード	2
ロード（ダブル）	3
ストア	3
ストア（ダブル）	4
ロード／ストア（アトミック）	4
フローティング	2 +
ジャンプリターン	2

ブランチ（NO）	2
ブランチ（YES）	1

命令の使用率を（ロード：15％、ストア：５％、ブランチ(YES)：15％、ブランチ(NO)：５％）と仮定して計算すると平均サイクルタイム／命令(CPI)は約1.3となる。さらにキャッシュのミス率を１％でペナルティを10サイクルとするとCPIは；

$$CPI = 1.3 + (0.01 \times (1 + 0.15 + 0.05) \times 10) = 1.42$$

サイクルタイムを60NSとすると、ネイティブのMIPS値は11.74MIPSとなる。
RISCアーキテクチャにおいては与えられたプログラムに対して生成されるコードは、CISCにおける同じプログラムに対して生成されるコードも簡素である。したがってRISCアーキテクチャに対してコンパイラが生成するコードは長くなる可能性がある。
通常このコード生成によるオーバヘッドは最適化コンパイラによって大きく低減できる。
SPARCに使用される最適化コンパイラは４段階の最適化を行うことができる。

SPARCの将来

RISCが90年代に要求されるCPUの性能に対する一つの答えであることは明白な事実であり、マイクロプロセッサーの高速化はRISCなしでは実現できない。現在あるCISCアーキテクチャはそのソフトウェア、ユーザベース等の資産から、既存の市場の中で成長をすると考えられるが、高速化ではまったくRISCに及ばなくなり、数10MIPSのCPUはすべてRISCになると考えられる。
今後の高速化、高性能化の一つのコンピュータアーキテクチャとしてパラレルプロセッシングの技術がある。これも現在ではソフトウェア技術が未完成であり、現実に使用できるにはまだ多少の時間を要する。しかしこの技術も核となるCPUにRISCを使うことによりさらに高性能化が可能となる。
SPARCはアトミック命令の様に将来のパラレル化を考えた機能も揃えておりこのアーキテクチャでもRISCが核になるものと考える。

また、CPUの64bit化（データバス）も計画されており、近い将来出現することになる。

SPARCはRISCアーキテクチャを採用したCPUであるが、UNIXオペレーティングシステムのもつ拡張性に100％ミートするハードウェアとして設計されている点に大きな特徴がある。高性能なアーキテクチャであるがその簡素化された構造により実際のハードウェアコストの低下を可能としている。これはSPARCベースのパーソナルコンピュータ、そしてECL、GaAsを使用した数10－数100MIPSのSPARCベースのスーパーコンピュータをも現実のものとする。そしてこれらすべてのコンピュータが同一アーキテクチャで実現することは、同じソフトウェアがパソコンからスーパーコンピュータまで使えることを意味し、コンピュータ世界に一つの革命を起こすものとなる。それはユーザにおけるソフトウェア資産の飛躍的な拡大につながることになる。