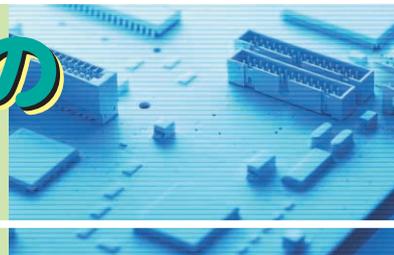


MIPS アーキテクチャの 変遷と現状



中上 一史

本章では、MIPS アーキテクチャの誕生から基本アーキテクチャの変遷、組み込み向けに拡張された機能、SoC 向けのシンセサイザブル・コアなど、MIPS アーキテクチャを理解するために必要な基礎知識を解説する。また、2種類のライセンス形態についても紹介する。

(編集部)

1 情報家電分野での MIPS

2000 年に入り、世界中の半導体メーカー各社が、システム・オン・チップ (SoC) を標榜し、プロセス技術のさらなる進歩、パッケージ技術の進化・改良、周辺 IP の充実、SoC の協調設計・検証環境の構築とさまざまな取り組みを進めています。しかしその後、未曾有の半導体不況が訪れ、世界の半導体メーカー各社は、生き残りをかけて、いろいろな対応策を模索しました。その結果、各社は自身の持ち味を生かした事業戦略を軸に、持てる技術や営業資産の集中をさまざまな形態で実行してきています。

そのような状況下、2003 年秋頃より、各種 STB/デジタル・テレビ、HDD/DVD ビデオ・レコーダ、デジタル・スチル・カメラ、デジタル・ビデオ・カメラなどで、従来の家電とは比較にならないほど、高性能・高機能な製品が市場に投入され、半導体メーカー各社の業績回復の一端を担ってきています。

これらの新たな組み込みアプリケーションでは、当たり前のように、PC との連携やネットワークへの接続など、これまでの組み込み向けマイクロプロセッサでは対応できない高い機能が要求されています。また、従来、最終製品のヒットを確定させる“Killer Application”も単一の機能ではなく、より複合的で、複雑なものになってきています。特に情報家電の分野では、消費者のニーズが多様化する中、実生活の一部に、違和感なく、それらのデジタル情報家電が取り込まれつつあります。この動きは消費者が自分のライフ・スタイルに合わせて、適宜に最先端のデジタル情報家電を実生活に取り入れることを示唆しています。これが“Killer Experience”です。

MIPS プロセッサは、消費者が実生活で望む“Killer Experience”を実現できるプロセッサ・コアであり、現在、これらの機器においておもに用いられている動作周波数帯の 200MHz ~ 400MHz の領域では、ほかのマイクロプロセ

サ・アーキテクチャより低消費電力です。さらに、下は数十 MHz から、上は 500MHz や 1GHz という高い動作周波数帯までを、単一のマイクロプロセッサ・アーキテクチャでカバーできる唯一の存在となっています。

2 組み込みマイクロプロセッサ・アーキテクチャの変遷

1975 年 10 月、IBM 社ワトソン研究所で、ハードウェア、コンパイラ、OS の開発が開始され、これが有名な IBM801 プロジェクトとなり、米国パークレー大学の SPARC アーキテクチャと米国スタンフォード大学の MIPS アーキテクチャに受け継がれました。まさに、このプロジェクトが RISC 型のマイクロプロセッサ・アーキテクチャの出発点であり、その起源といってもよいでしょう。

それ以降、複数の半導体メーカーが、さまざまな組み込み向けの RISC 型マイクロプロセッサを商品化しました。それぞれの RISC 型マイクロプロセッサに盛り込まれた機能ブロックや、その設計思想は、この 801 プロジェクトの成果を盛り込んでおり、かつ従来の組み込み向け CISC 型プロセッサの利点を取り込んで、さらなる進化を遂げてきました。

ここで、801 プロジェクトの設計思想について簡単にまとめてみます。

- 1) ロード/ストア・アーキテクチャをもつ多ポートの大きなレジスタ・バンク
- 2) 高度なコンパイラ技術への依存
- 3) 遅延ロード、遅延ストア、遅延分岐を含む単純な命令セットをパイプライン化して 1 サイクルで実行
- 4) キャッシュ・メモリの導入

これらの特徴を盛り込んだ商用プロセッサを開発すべく、前述の米国パークレー大学およびスタンフォード大学が精力的に研究を進めた結果、次のような結論を導き出しました。

- 1) コンパイラは複雑な命令をあまり使用しない
- 2) 複雑な命令は不合理な性能特性を示す
- 3) 複雑なマシンは設計に時間がかかる
- 4) 複雑な命令セットは設計上の誤りを含む確率が高い
- 5) 複雑な命令はマシン全体を遅くする
- 6) チップ面積を有効に使用するためにほかにすることがある

特に6番目の項目は、その後のマイクロプロセッサ設計およびSoC化に大きな影響を与えました。つまり、半導体プロセス技術の進歩にともなって、複雑な命令セットを1チップに集積することが可能になりましたが、それに反して単純なプロセッサを開発し、より速いトランジスタ、パイプライン、オンチップ・キャッシュにその領域を取っておくほうが有効であるという考えかたが主流になりました。

3 MIPS 基本アーキテクチャの変遷

前述の米国スタンフォード大学の研究成果は、MIPS Computer Systems社と、その半導体ライセンスによって、商用のCMOSプロセスを用いて製品化されました。最初の製品はR2000マイクロプロセッサで、32ビットの整数演算ユニット、例外処理機能、キャッシュ制御機能、メモリ管理機能を1チップ化したデバイスでした。それに外付けの汎用SRAM、浮動小数点演算ユニットR2010、ライト・バッファR2020を用いて高い処理能力をもったUNIXマシンが開発されました(当時のクロック周波数は20MHz)。その際に初めて用いられた命令セットをMIPS I命令セットと呼びます。

その後、さらなるアーキテクチャの改良、拡張が順次行われ、1988年にはR3000マイクロプロセッサ(25MHz)へ進化し、同時に命令セットもMIPS II命令へと進化しました。そして1991年には、従来複数チップだった32ビットの整数演算ユニット、例外処理機能、キャッシュ制御機能、メモリ管理機能、浮動小数点演算ユニット、キャッシュ・メモリ、ライト・バッファなどを1チップに集積するとともに、64ビット化とマルチプロセッサ構成のサポート機能追加が行われ、R4000およびR4400(MIPS II/命令セット)へと進化しました。1992年には、さらに浮動小数点性能を向上させたR5000が市場に投入され、命令セットもMIPS IIIへ進化しました。

しかし、これらのマイクロプロセッサ群をおもに用いたアプリケーションは、UNIXベースのEngineering Workstation(以下EWS)での応用が主で、MIPSアーキテクチャを組み込みアプリケーションに応用展開したのは、おもにMIPSアーキテクチャのライセンス(東芝、NEC、IDT、LSIロジックなどの半導体メーカー)でした。この第一線の有力な半導体ライセンスは、UNIX EWSを主体として発展してきたMIPSアーキテクチャを、組み込みアプリケーションに応用しやすいように改良を行いました。たとえば、例外処理方法の追加、外部割り込み応答性の強化、ビット/バイト操作を行いやすい命令の追加、

積和演算命令の追加などを前述のMIPS I、II、III、およびIVの命令セットに対して行い、レーザ・ビーム・プリンタ、複合複写機、デジタル・スチル・カメラ、PDA、ナビゲーション・システム、ゲーム機器など、初期の広範囲な組み込みアプリケーションにおいて、その応用領域を広げてきました。

MIPS Computer Systems社は、1992年にSGI社のハイエンド・マイクロプロセッサ開発部門として吸収されました。SGIはハイエンド・マイクロプロセッサとして、R10000、R12000などの開発を行い、2次キャッシュ・システムをもったシステムで必要となる命令セットのさらなる拡張、マルチプロセッサ環境で必要となる命令セットの拡張、グラフィックス処理などで必要となるSIMD系の命令セットの拡張をMIPSアーキテクチャに対して行い、MIPS V、VI、命令に反映させました。

このように、MIPSアーキテクチャは、広範囲な組み込みアプリケーションで応用され、進化してきたわけですが、1998年、現在のMIPS Technologiesが、SGI社からスピン・アウトする際に、1983年より面々と拡張・進化を遂げてきたMIPSアーキテクチャを“MIPS32”アーキテクチャおよび“MIPS64”アーキテクチャとして、再度体系化し、まとめ直しました(図1)。

1998年以降、MIPS Technologiesでは、これらのマイクロプロセッサ・アーキテクチャの技術供与(アーキテクチャ・ライセンス供与)および、それらのアーキテクチャを元に、MIPS Technologiesが開発したフル・シンセサイザブルな32ビット/64ビットのマイクロプロセッサ・コア(ソフト・コア)の技術供与(コア・ライセンス供与)を多数の企業に対して行っています。

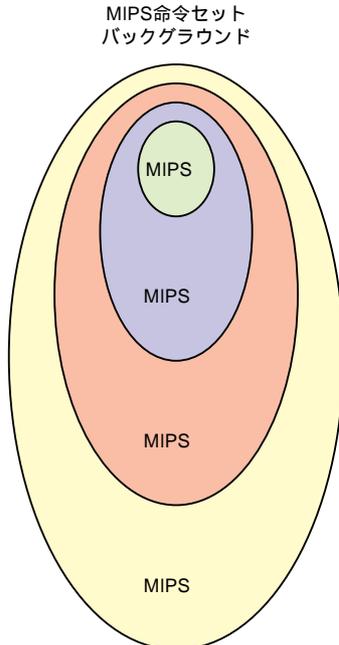
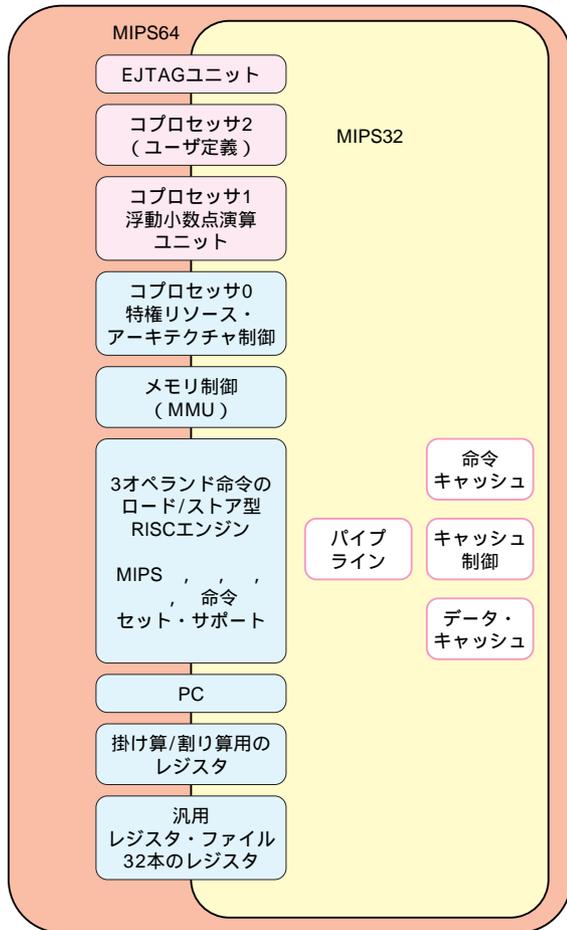
4 MIPS 基本アーキテクチャの拡張

SGI社からスピン・アウトした後、MIPS Technologiesは、MIPS32およびMIPS64アーキテクチャをより広範囲な組み込みアプリケーションで応用されることを想定して、2001年10月米国で開催されたMicroprocessor Forumにて、MIPS32 Release2アーキテクチャおよびMIPS64 Release2アーキテクチャを発表しました。

このRelease2アーキテクチャは、昨今の組み込みアプリケーション応用においてさらに重要視されている次の4点に着目して、拡張を行ったものです。

- 1) 割り込み応答性のさらなる向上
- 2) 効率的なビット操作系の命令の拡充
- 3) メモリ管理方法の拡充
- 4) コプロセッサ接続の柔軟性の向上

以降では、MIPS32 24Kにも実装した基本アーキテクチャ“MIPS32 Release2”で、それぞれ拡張された点について説明します。



- MIPS 命令セット
 - Load/Store
 - Computational
 - Jump and Branch
 - Co-Processor
 - Special
- MIPS 命令セット
 - Trapping
 - Load-linked and Store conditional
 - Sync, Branch Likely, Square root, ...
- MIPS 命令セット
 - 64 bit instruction set support (still 32 bit instructions)
- MIPS 命令セット
 - Conditional move operations
 - Prefetch Instructions, ...
- MIPS命令セット拡張
 - MDMX ASE
 - MIPS16 ASE
 - SmartMIPS ASE

MIPS 命令セットは、MIPS 命令セットのスーパーセットであり、MIPS 命令セットは、MIPS 命令セットのスーパーセットとなっている

- アーキテクチャとして必須
- アーキテクチャ・オプション
- MIPS社のコアで一般的に実装

図1 MIPS32 および MIPS64 アーキテクチャ

MIPS64 アーキテクチャは MIPS32 アーキテクチャを完全に包含しており、完全なソフトウェア互換性を保っている

割り込み応答性のさらなる向上

前述したように、初期の MIPS アーキテクチャは UNIX 環境を基本に進化した関係で、組み込みアプリケーションにおいて要求される外部割り込みに対する処理が、ほかの組み込み型 RISC アーキテクチャと比べて良くないといわれていました。そのため、初期の MIPS アーキテクチャ・ライセンスがさまざまなくふうを凝らしてアーキテクチャを進化させ、それが MIPS32/MIPS64 アーキテクチャにまとめられています。

MIPS では、さらに将来を見据えて、より広範囲な組み込みアプリケーションに応用できるよう、さらなる応答性の向上を図りました。

- リアルタイム・システムにおいては、つねに割り込みの高速応答性が要求されるとともに、その処理時間を特定することが要求されている
- SoC 化が進むことにより、SoC 内部の割り込み処理は煩雑化・複雑化の一途をたどっている
- ブロードバンド環境が整うにつれてデータの転送能力は格段に向上し、それにともなって、膨大なバケットを高速に処理

するニーズが高まってきている

これらの要求に基づいて、MIPS32 Release2 では、次のことが強化されています。

- 1) アーキテクチャの規定の中で、標準的に外付けの割り込み制御方法を規定
- 2) 優先順位(プライオリティ)付きベクタ割り込みをサポート
- 3) 汎用シャドウ・レジスタを追加

外部割り込みコントローラは、発生した割り込み要因のプライオリティを決定し、また、チップ内のプライオリティ・エンコード、およびベクタ生成ブロックは、チップ内部の割り込み要因と外部からの割り込み要因を的確に解析して、特定のベクタを生成します。一つのコアで、最大 16 個の割り込みベクタをサポートできます。内部割り込み、および外部割り込みの組み合わせ、およびベクタ・アドレスなどは、すべてプログラムブルです。また、従来の MIPS32/MIPS64 アーキテクチャの割り込み制御方法と完全な互換性をもっています(図2)。

汎用シャドウ・レジスタ機構は、割り込みや例外が発生した際に、ハードウェアが自動的にその時点の汎用レジスタの値

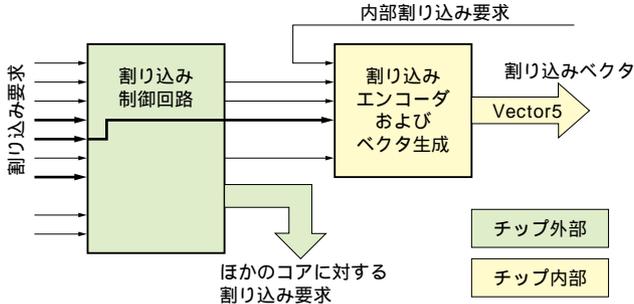


図2 ベクタ割り込み
組み込みシステムで要求される多数の割り込みに対応するために、アーキテクチャ・レベルで外部の割り込み制御方法を規定するとともに、高速な割り込み応答性を実現するために、ベクタ割り込みを規定した。もちろん、従来のMIPS32およびMIPS64の割り込みとも互換性を維持している

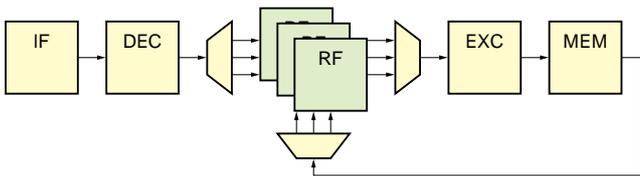


図3 シャドウ・レジスタ
規定したシャドウ・レジスタは、ソフトウェアから見た場合は、従来の単一の汎用レジスタ・セットと同様に見える、レジスタの切り替えをすべてハードウェアで実現している。また、各レジスタに対して割り込みや例外などの割り当てはすべてソフトウェアで明示的にプログラム可能

を、対応するシャドウ・レジスタに保存する機能です。各割り込みや例外をどのシャドウ・レジスタ・セットに対応付けるかは自由に設定できます。アーキテクチャ的には、最大16セットのシャドウ・レジスタを規定できますが、現在、MIPS社で供給しているコアでは、最大4セットのシャドウ・レジスタの搭載ができます(図3、図4)。

効率的なビット操作系の命令の拡充

組み込みアプリケーションにおいては、ビット操作、バイト操作など、マイクロプロセッサのデータ処理タイプとは違った操作が要求されます。このあたりの処理に関して、CISC系のマイクロプロセッサがきめ細かい対応を取ってきました。特にブロードバンド環境においては、

- データ転送能力の向上とともに、効率的なプロトコル処理が必要とされている
- より複雑でさまざまなプロトコル処理への対応が必要となってきた

これらの要因に基づいて、MIPS32 Release2 アーキテクチャでは、

- 1) ビット・フィールドに対するビットの挿入/取り出しをサポートする命令の追加
- 2) バイトのスワップ命令の追加
- 3) ローテート命令の追加

を行いました。32ビット/64ビット・レジスタに対して、ピッ

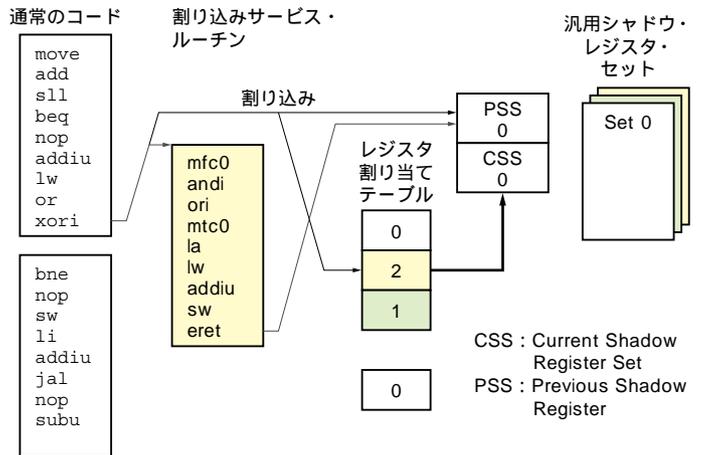


図4 シャドウ・レジスタの動作
割り込みが発生して、対応する割り込みルーチンがサービスされる際に、どのようにシャドウ・レジスタがハードウェア的に切り替わるのかを示している。CSSやPSSは、レジスタ割り当てテーブルの内容に基づいて、レジスタ切り替えを行う

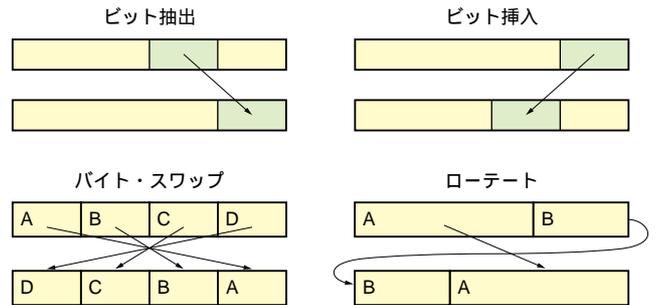


図5 ビット・バイト操作

トの挿入や抽出を可能とする命令、および同一レジスタ内でデータのエンディアンを切り替え可能な命令、さらに同一のレジスタ内でデータのローテート命令の追加を行い、オリジナルのMIPS32/MIPS64命令で同様のことを行う場合と比べて、大幅な命令数の削減、および処理に必要な一時レジスタの削減を実現しました(図5)。

メモリ管理方法の拡充

昨今の組み込みアプリケーションのメモリ管理においては、次に示す点の重要性が増してきています。

- 複雑なマルチタスク・ソフトウェアは、独自のメモリ管理方法およびメモリ保護を必要としている
- システム・コストの低価格化は、より効率的なメモリ使用・管理方法を必要としている
- 効率的な仮想メモリ・システムの重要性が増している

これらの要因に基づいて、MIPS32 Release2では、次のことが追加されています。

- 1) きめ細かいページ・サイズをサポート
- 2) Contextレジスタに対するプログラマビリティの向上を行い、TLB例外処理時間を削減

Pr
1
2
Ap
Ap
3
4
5
6
Ap
Ap

最大 64M バイトまたは 256M バイトのページ・サイズをサポートすることにより、大容量のデータ領域を一つもしくは二つの TLB エントリでマップできるようになり、結果として、TLB ミスの発生回数を削減し、大容量のデータベースや、テーブルを用いるシステム・パフォーマンスの特定をしやすくしました。

また、1K バイトおよび 2K バイトのページ・サイズをサポートすることにより、より少ないメモリでも仮想メモリ・システムを構築できるようにしました。もちろん、従来の MIPS32/MIPS64 アーキテクチャでサポートしていた最小のページ・サイズである 4K バイトと互換性をもたせることで、過去のソフトウェアとの互換性を保っています。

また、昨今の OS は、それぞれ個別のページ・エントリ・テーブルのフォーマットを持っており、XContext および Context レジスタに対するプログラマビリティ性を向上させることにより、TLB 例外発生時の処理能力も向上させることが可能になります(図6)。

コプロセッサ接続柔軟性の向上

アプリケーションの複雑さが高くなるにしたがって、高いレベルでの処理の並列化の必要性が増すとともに、効率的なコプロセッサ処理は、複雑化の一途をたどるシステム構成の単純化、およびコストの削減に絶大な効果を発揮します。そのために、MIPS32 Release2 アーキテクチャでは、コプロセッサ接続の柔軟性を図7に示すように向上させ、アプリケーション・パフォーマンスの向上、ダイ・サイズの最適化や、過去の資産の有効活用をしやすくしました。

5 組み込み SoC 設計を取り巻く環境の変化

マイクロプロセッサの処理能力は年とともに向上し、また、プロセス技術の進歩と合わせて、有名な“ムーアの法則”にしたがって進化・向上を遂げています。同時に CISC(Complex Instruction Set Computer)と RISC(Reduced Instruction Set Computer)論争を経て、当初は組み込みアプリケーションに RISC プロセッサを応用することにはさまざまなリスクがあると議論されました。

しかし、今日では、RISC 型のマイクロプロセッサが非常に広範囲な組み込みアプリケーションに応用され、“Killer Experience”を実現しています。昨今のシステム LSI は、機器のデジタル化、高機能化、さらにはプロセス・テクノロジーの進化にともなって、大幅かつ急激に、その複雑さを増しています。

また、1990 年代初頭より始まったシステム LSI 設計の模索は、ASIC, ASSP, MCM, SIP など、さまざまな実装、集積化技術の開発を礎として、多数の周辺 IP ブロックや、32 ビット/64 ビットのマイクロプロセッサ・コアを用いた“システム・オン・チップ”(SoC)へと展開されています。さらに、半導体のプロセス技術、製造技術の発展にともない、今や、90nm プロセス、300mm ウェハの時代を迎えようとしています。

このような時代の流れの中、1998 年以降、MIPS は、“高い柔軟性・自由度”、“プログラマビリティ性の向上”、そして“スケラブルな高い処理能力”を持ったシンセサイザブル・コア、および 32 ビット/64 ビット・マイクロプロセッサ・コア・アーキテクチャの開発を進め、整理統合したものが最新のマイクロ

図6 Context および XContext レジスタ
Context レジスタ、ContextConfig レジスタ、BadVaddr の組み合わせにより、仮想メモリ・システムにおける Page Table Entry フォーマットと、実際のマイクロプロセッサ内部での仮想アドレス変換にともなう Table Format を分離できるので、さまざまな OS に対する移植性を高めることができる

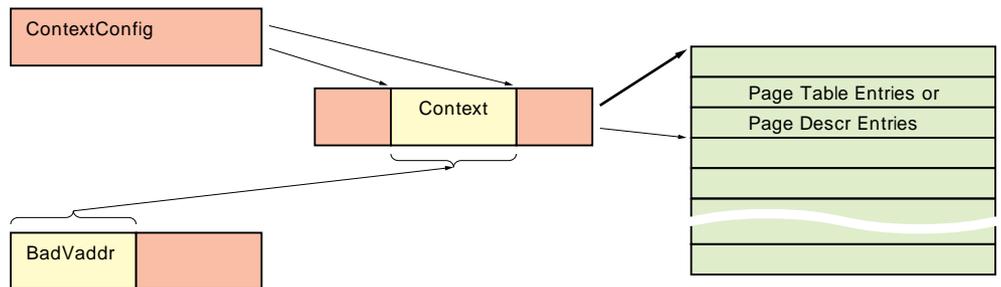
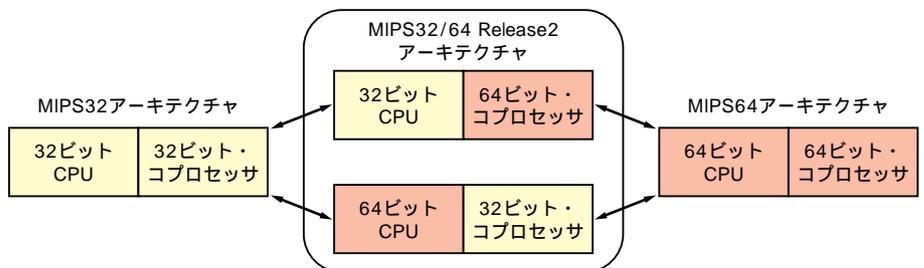


図7 コプロセッサ・サポート拡張
従来 32 ビットのプロセッサには、32 ビットのコプロセッサ、64 ビットのプロセッサには、64 ビットのコプロセッサ接続しか許されていなかったものを、MIPS32/MIPS64 Release2 アーキテクチャでは、32 ビットのプロセッサに 64 ビットのコプロセッサ、64 ビットのプロセッサに 32 ビットのコプロセッサを接続可能にした



プロセッサ・コア“ MIPS32 24K (以下 24K)です。MIPS では、この 32 ビット・コア・アーキテクチャを、デジタル家電、ブロードバンド・ネットワーク対応機器市場などへの展開を念頭に、シリーズ化を展開していきます(図 8, 図 9)。

6 MIPS のライセンス形態

2 種類のライセンス形態

1998 年以降、MIPS Technologies では、多数の企業に対してアーキテクチャ・ライセンスおよびコア・ライセンスの技術供与を展開しています。現在、アーキテクチャ・ライセンスは 14 社、コア・ライセンスは 97 社を数えるに至っています。

この後の説明の理解を容易にするために、まずアーキテクチャ・ライセンスとコア・ライセンスの違いを明確にしておきましょう。

▶ アーキテクチャ・ライセンス

MIPS32/MIPS64 アーキテクチャ準拠の 32 ビット、および 64 ビットのマイクロプロセッサ・コアを“ 独自に” 設計開発、製造、販売ができる企業のことをアーキテクチャ・ライセンスと称します。

▶ コア・ライセンス

ミッス・テクノロジーズが開発した MIPS32/MIPS64 アーキテクチャに基づくマイクロプロセッサ・コアを、自社の SoC 設計開発において利用することが可能な企業をコア・ライセンスと称します。

プロセッサ仕様の範囲

では、MIPS32/MIPS64 アーキテクチャでは、実際のプロセッサ仕様のどの範囲までを厳格に規定しているのでしょうか？ あまり厳格な規定だと、アーキテクチャ・ライセンスの自由度が損なわれ、また、あまりに大雑把な規定では、さまざまな派生アーキテクチャが誕生し、互換性の面で大きな問題を生じます。

MIPS Technologies では、1998 年に MIPS32/MIPS64 アーキテクチャを再統合・整理をする際に、アプリケーション、および OS などのソフトウェアからマイクロプロセッサを見た場合、最低限規定しなければならない要素を、仕様として次のように規定しました。

- 1) マイクロプロセッサが実行可能な命令群の定義
- 2) マイクロプロセッサの特権、ユーザ・モードでの動作規定
- 3) 仮想メモリの規定
- 4) 例外・割り込み処理の規定
- 5) コプロセッサの規定
- 6) システム・リソース制御に関する規定(汎用レジスタ・ファイル、浮動小数点レジスタ・ファイル、浮動小数点演算制御レジスタ群、TLB 制御レジスタ、キャッシュ制御レジスタ、各種 Status および制御レジスタなど)

上記の規定を遵守することで、コンパイラ、デバッガ、OS、

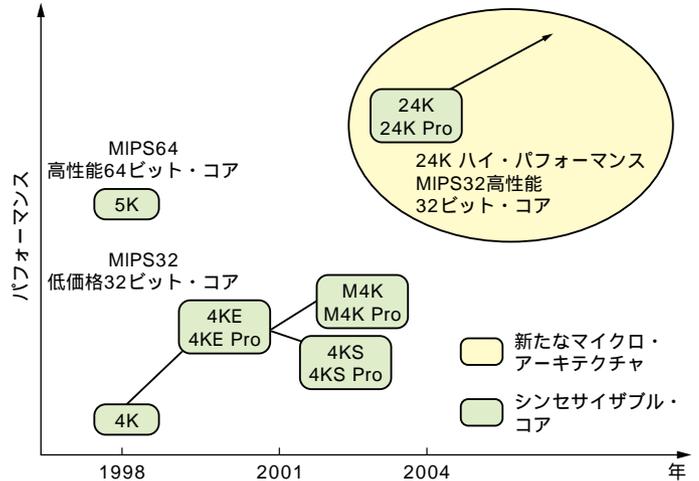


図 8 MIPS シンセサイザブル・コア・ロードマップ

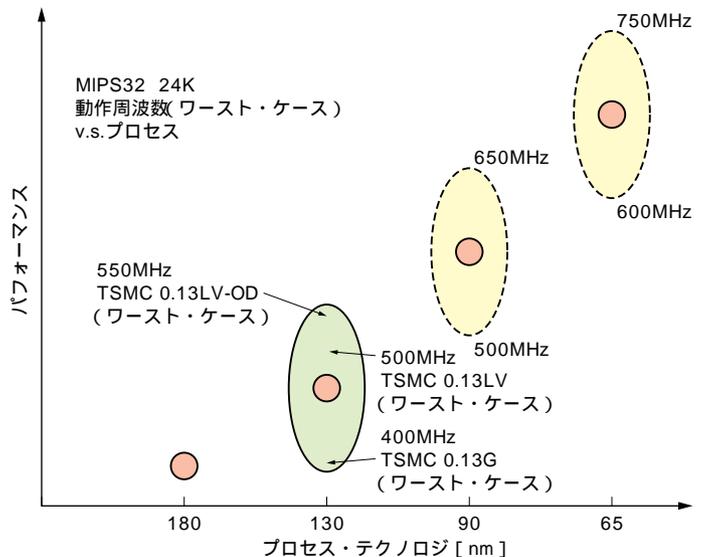


図 9 MIPS32 24K 動作周波数とプロセス・テクノロジー
LV-OD や LV, G は TSMC の 0.13 μm プロセスの名称

アプリケーション・ソフトウェアなどの互換性を維持しつつ、実際の内部回路の実現手法や独自機能の追加ができる高い自由度をアーキテクチャ・ライセンスに提供することが可能になります。この発想を元に、数百社におよぶサード・パーティ企業のさまざまな設計支援ツール、OS、ミドルウェア、デバッガ、評価ボードをシステム設計に利用できる環境“MIPS エコシステム”が構築されています。

参考文献

- (1) “比較研究 RISC アーキテクチャ VLSI RISC Architecture and Organization, 基礎から学ぶ、プロセッサ設計と VLSI チップの実例” Stephen B. Furber 著、豊橋技術科学大学 今井正治監訳、日経 BP 社

ながみ・かずふみ ミッス・テクノロジーズ