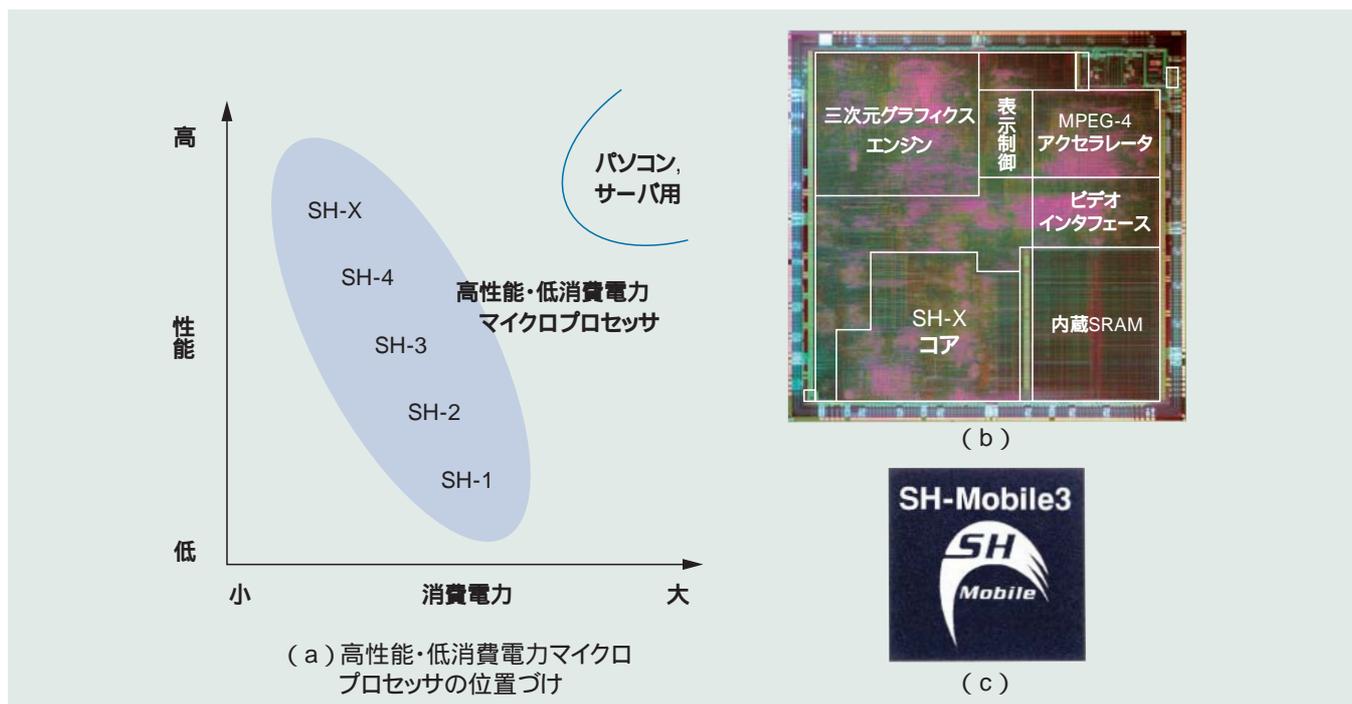


高性能・低消費電力マイクロプロセッサ

High-Performance and Low Power Consumption Microprocessors

内山 邦男 Kunio Uchiyama



注:略語説明 MPEG-4(Moving Picture Expert Group 4), SRAM(Static Random Access Memory)

高性能・低消費電力マイクロプロセッサの位置づけ (a)に示すように、マイクロプロセッサの高性能化と低消費電力化の研究開発により、1993年からSH-1, SH-2, SH-3, SH-4, SH-Xと開発を進めてきた。(b)は最近開発したSH-Xを内蔵した携帯電話用アプリケーションプロセッサ SH-Mobile3 のチップ、(c)は製品の外観である。

民生機器を中心に1990年代から始まったデジタル化の波は、現在では各分野に行き渡っている。この原動力となっているのが、高性能、低消費電力、低価格の三要素を兼ね備えたマイクロプロセッサである。

日立製作所は、このようなニーズに応えるために、1980年代の後半から高性能・低消費電力マイクロプロセッサの研究開発を進めている。その技術により、1993年に32ビットマイクロプロセッサ「SH-1」が最初に製品化された。この研究開発において、16ビット固定長命令の

RISCアーキテクチャやマルチメディア向け拡張命令、高性能と低消費電力を両立させる回路の活性化率低減方式やJava[®] アクセラレータ、LSIの微細化の進展に伴って大きな課題となっているスタンバイ電流の増大を抑える技術などを実現している。現在では、株式会社ルネサス テクノロジから製品化されているSuperHやSH-Mobileへと研究成果が反映されている。

) JavaおよびすべてのJava関連の商標およびロゴは、米国およびその他の国における米国Sun Microsystems, Inc.の商標または登録商標である。

1 はじめに

マイクロプロセッサは1970年代に制御用として開発が始まり、1980年代になるとパソコンやワークステーション用に高性能化が追求された。1990年代に入ると民生機器のデジタル化が進み、高性能だけでなく、消費電力が低く、価格も安いマイクロプロセッサが必要となってきた。現在、大きな市場を形成している携帯電話やデジタルカメラなどのデジタル民生機器、自動車やロボットなどに内蔵される制御装置には、高性能、低消費電力、低価格な

マイクロプロセッサが必ず使われている。

日立製作所は、1980年代の後半から、このニーズを満たすために高性能・低消費電力マイクロプロセッサの研究開発を進めている。この成果として、株式会社ルネサス テクノロジのSuperHやSH-Mobileが製品化され、幅広い分野で重要な役割を担っている。

ここでは、このマイクロプロセッサ開発のねらい、アーキテクチャ設計のポイント、主要な高性能化と低消費電力化技術、および製品への実装例について述べる。

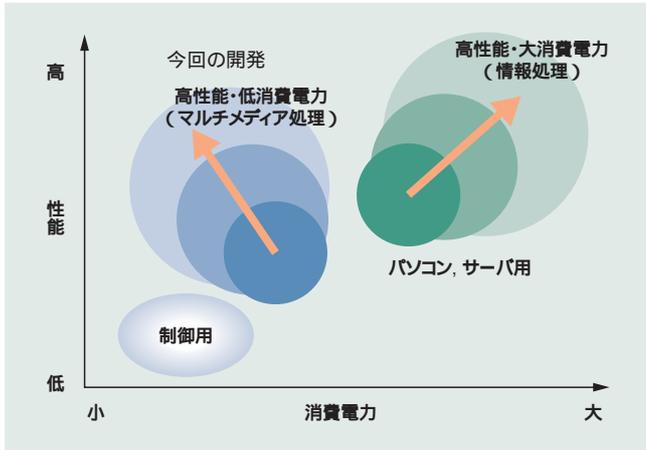


図1 高性能・低消費電力マイクロプロセッサの位置づけ
この開発では、消費電力を低く抑えながら、マルチメディア処理などの新しい
応用で高い性能を発揮するマイクロプロセッサの実現を目指す。

2 開発の背景とねらい

1990年代の初め、家電品に組み込まれるマイクロプロセッサは、テレビやクーラなどの制御に使われており、価格重視のために、その性能は1 MIPS(Million Instructions per Second)程度であった。一方、ワークステーションやパソコンでは50 MIPS程度の高い性能のマイクロプロセッサが使われていたものの、消費電力は数ワットで、価格も数万円と高価格であった。このような状況の中で、電子手帳、プリンタ、ファクシミリ付き電話機など、民生機器のデジタル化が始まり、同年代の中ごろからは、個人情報端末、デジタルカメラ、デジタルオーディオ・ビデオプレーヤ、デジタルテレビ、携帯電話、カーナビゲーションなど新しい情報家電が製品化された。これらの機器では、オーディオ、画像、グラフィクスなどのマルチメディアデータを低消費電力、低価格で高速に処理する必要がある。しかし、従来の制御用マイクロプロセッサやパソコン・サーバ用マイクロプロセッサでは実現することが困難であったため、高性能、低消費電力、低価格を同時に満たすマイクロプロセッサの開発が強く望まれていた(図1参照)。日立製作所は、このようなニーズに対応するため、高性能・低消費電力マイクロプロセッサの研究開発を進め、研究成果を製品化してきている。

3 アーキテクチャの設計

3.1 16ビット固定長命令セット

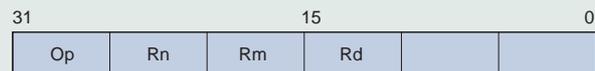
研究を開始した当時、マイクロプロセッサのアーキテクチャには、可変長命令のCISC(Complex Instruction Set Computer)と32ビット固定長命令のRISC(Re-

duced Instruction Set Computer)の2種類があった。前者は複雑な機能がある可変長命令を数多く持っているため、高性能化を図るためには複雑なハードウェアが必要となり、チップ面積も大きくなる。一方、後者は命令が簡素化されているためにハードウェア量は少なく済むものの、命令長が32ビット固定であることから、プログラムサイズが大きくなるという課題があった。

これらの課題を解決するために民生機器のデジタル処理を徹底的に分析し、将来のニーズも踏まえて、16ビット固定長命令セットのRISCアーキテクチャ(以下、SHアーキテクチャと言う)を設計した(図2参照)。従来型RISCは32ビット固定長であるが、SHアーキテクチャは16ビットですべての命令を定義した。例えば、演算命令は2オペランド方式として、レジスタ数も従来の32本から16本に減らすことによってビット数を圧縮している。

SHアーキテクチャは、従来型RISCに比べ、プログラム実行性能の低下を抑えながら、プログラムサイズを30~40%縮小することができる(図3参照)。これにより、機器に必要なメモリ数やオンチップのメモリ容量が削減可能となり、低価格化に大きく貢献できた。

また、マイクロプロセッサのCPU(Central Processing



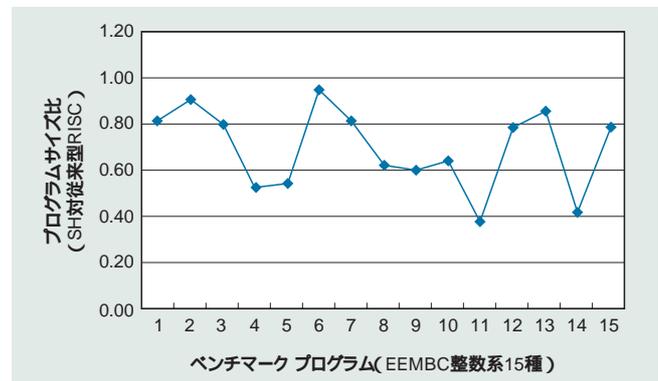
(a)従来型RISCアーキテクチャ



(b)SHアーキテクチャ

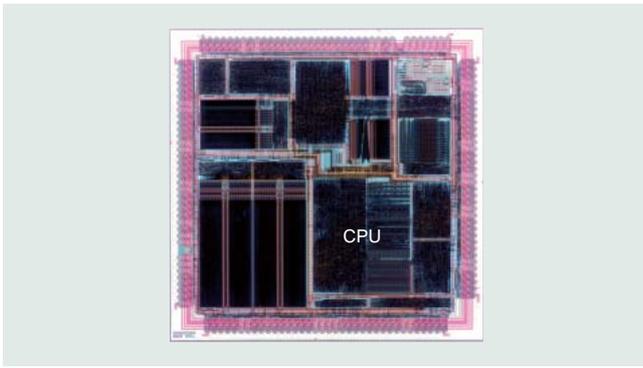
注:略語説明 RISC(Reduced Instruction Set Computer)

図2 SHアーキテクチャの命令フォーマット
従来型RISCでは32ビットで命令を定義していたが、SHアーキテクチャではその半分の16ビットで全命令を定義している。



注:略語説明 EEMBC(Embedded Microprocessor Benchmark Consortium)

図3 プログラムサイズの比較
SHアーキテクチャでは、従来型RISCに比べてプログラムサイズを30~40%縮小することを可能にする。



注:略語説明 CPU(Central Processing Unit)

図4 SHアーキテクチャの実装例
0.5 μm技術を用いたSH-3のチップの外観を示す。CPU面積は5 mm²と小さい。

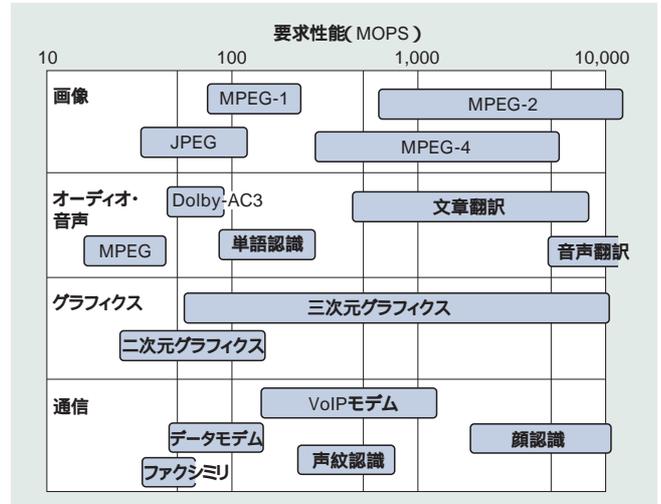
Unit)部分も従来型RISCに比べて30~40%のチップ面積で実現できるようになった。1995年にはSHアーキテクチャのマイクロプロセッサ“SH-3”を開発した(図4参照)。当時用いられたのは0.5 μm技術であるが、CPU部分の面積は5 mm²と小さい。最新の130 nm技術によるCPU部分は3 mm²以下となっており、きわめて高い性能面積比が実現されている。

3.2 マルチメディア処理用の拡張

携帯電話やデジタルカメラなどのデジタル機器では、静止画、動画、オーディオの記録、再生など、いろいろな種類のマルチメディア処理が必要となる。カメラ機能を実現するためには、静止画像(JPEG(Joint Photographic Expert Group)形式)の圧縮や伸長を短時間で実行する性能が要求され、ムービー機能では、Motion JPEGやMPEG-4(Moving Picture Expert Group 4)規格などによる動画の圧縮、伸張が必要となる。オーディオプレーヤでは、MP3(MPEG Audio Layer-3)やDolby-AC3(Audio Code Number 3)など各種規格のオーディオ伸長処理を行っている。また、デジタルテレビを受信する場合には、H.264やMPEG-4規格の映像データの伸張が、高品質なゲーム機能を実現するには、三次元グラフィクス処理がそれぞれ必要となる。

これらのマルチメディア処理では、一般的に、多量の演算を必要とする。この高い演算性能を達成できるように、SHアーキテクチャではベクトル命令とDSP(Digital Signal Processor)命令を導入した。マルチメディア処理の要求性能を、MOPS(Mega Operations per Second)値で図5に示す。

ゲームなど立体図形を扱う応用では、三次元グラフィクス処理が行われる。ここでは、各オブジェクトの頂点座標は、通常、単精度(32ビット長)の浮動小数点データとして表現され、この各頂点の座標データに対してアフィン変換



注:略語説明 MOPS(Mega Operations per Second), JPEG(Joint Photographic Expert Group), MPEG(Moving Picture Expert Group), VoIP(Voice over Internet Protocol)

図5 マルチメディア処理の要求性能
画像、オーディオ、グラフィクスなどのマルチメディア処理では、高い演算性能が要求される。

や輝度計算がなされる。この計算をリアルタイムで行う場合、図形が高精細になるにつれて計算量が膨大になっていく。これに対応するために、単精度浮動小数点演算を複数個実行できる四次元ベクトル命令であるFIPR(Floating Point Inner Product)命令を追加した(図6参照)。FIPR命令では、四つの浮動小数点レジスタを一組の四次元ベクトル(FV_m, FV_n)として定義し、そのベクトルに対する内積演算を行う。この命令により、7演算(加算×3, 乗算×4)を1サイクルごとに実行することを可能にした。

画像やオーディオ処理では、固定小数点形式のデータに積和演算を用いて各種の変換を行うことが多いため、SHアーキテクチャでは、積和演算を中心としたDSPアーキテクチャを追加できるようにした。DSP命令を拡張するときは、16ビット固定小数点演算用のレジスタファイル、加算器、乗算器などを内蔵するDSPと、データ格納用の16ビット幅Xメモリ、Yメモリを追加して構成する(図7参照)。XメモリとYメモリのアドレスはCPUによって計算される。

命令記述

FIPR FV_m, FV_n

演算オペランド

FV_m={FR(m), FR(m+1), FR(m+2), FR(m+3)}

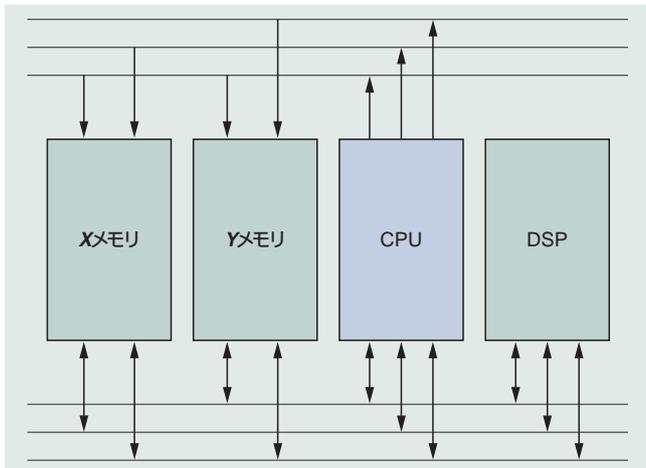
FV_n={FR(n), FR(n+1), FR(n+2), FR(n+3)}

命令動作

$$FR(n+3) = FR(m) \times FR(n) + FR(m+1) \times FR(n+1) + FR(m+2) \times FR(n+2) + FR(m+3) \times FR(n+3)$$

注:略語説明 FIPR(Floating Point Inner Product)

図6 四次元ベクトル命令の例
単精度浮動小数点四次元ベクトルの内積演算をFIPR命令の実行で実現する。



注:略語説明 DSR(Digital Signal Processor)

図7 DSPアーキテクチャ追加時の構成
DSPアーキテクチャを追加する場合、Xメモリ、Yメモリ、DSPがCPUに追加される。

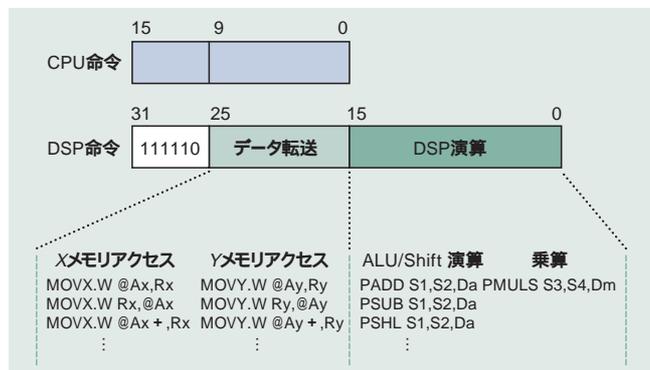


図8 DSP命令のフォーマット
32ビット長のDSP命令では、四つのオペレーションを同時に指定することが可能である。

DSP命令のフォーマットを図8に示す。CPU命令は、CPUで実行される16ビット長の命令である。一方、CPUとDSPの両方で実行されるDSP命令は32ビット長であり、Xメモリをアクセスするフィールド、Yメモリをアクセスするフィールド、ALU(Arithmetic and Logic Unit:演算装置)シフトを用いた演算フィールド、乗算器を用いた演算フィールドから成る。この命令一つで四つのオペレーション(二つのメモリアクセス動作、ALU・シフト演算、乗算)が実行でき、効率のよい信号処理やマルチメディア処理が可能となる。

4 高性能と低電力の両立

4.1 活性化率の低減技術

CMOS(Complementary Metal-Oxide Semiconductor)技術で作られているマイクロプロセッサの動作時の電力は、 $P \times F \times C \times V^2$ で主に決まる。ここで、 P は回路の活性化率、 F は動作周波数、 C は負荷容量、 V は電

源電圧である。性能を上げるために、単に動作周波数を上げると電力が増大してしまうので、性能を上げながら、活性化率を下げる技術が必要になる。この開発では、いろいろな手段を用いて P を下げるくふうをしている。そのうちの二つの手法について以下に述べる。

キャッシュメモリの選択起動方式は、1994年に製品化したSH-2で導入している。キャッシュメモリは、マイクロプロセッサの電力消費の多くを占めるために、低消費電力化が重要である。その構成と動作の仕組みを図9に示す。この方式では、キャッシュメモリを複数のバンクに分割して構成する。データへのアクセス時に、まず検索動作を行い、目的のデータが格納されているヒットしたバンクだけを活性化し、他のバンクは活性化しないために電力の消費が抑えられるので、キャッシュメモリ全体で60~70%の電力削減が可能となった。

ゲートドパイプライン制御は、1999年に製品化したSH-4で導入した。パイプライン制御では、各命令を整数演算、浮動小数点演算、ロード・ストア、分岐などにグループ化し、グループごとにパイプラインステージを定義して実行する回路を割り当てる(図10参照)。命令ごとに動

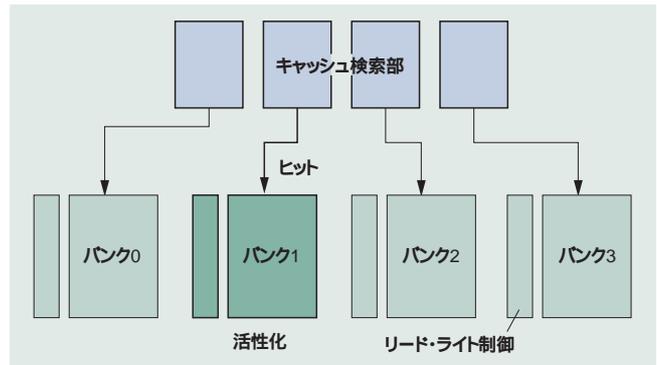


図9 選択起動型キャッシュメモリの構成と動作の仕組み
キャッシュを検索してヒットしたバンクだけ活性化することにより、消費電力を低減する。

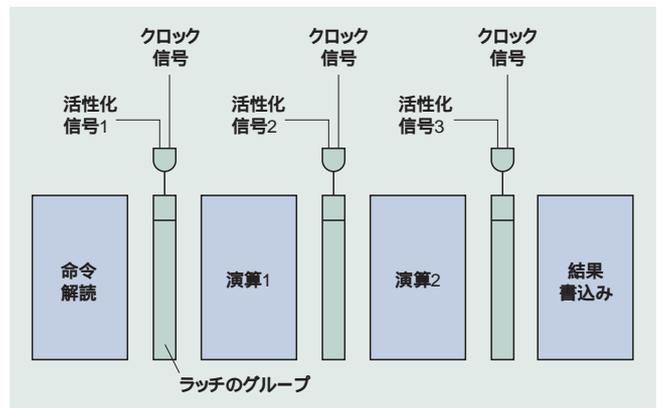
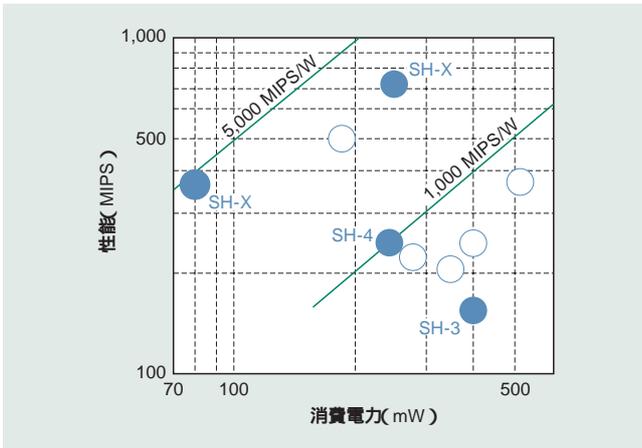


図10 ゲートドパイプライン制御の概要
各パイプラインステージの活性化信号を制御することにより、命令実行に必要な部分だけを活性化する。



注:略語説明 MIPS(Million Instructions per Second)

図1.1 消費電力とMIPS値の関係
最新のSH-Xでは、性能と消費電力比が4,500 MIPS/Wに達している。

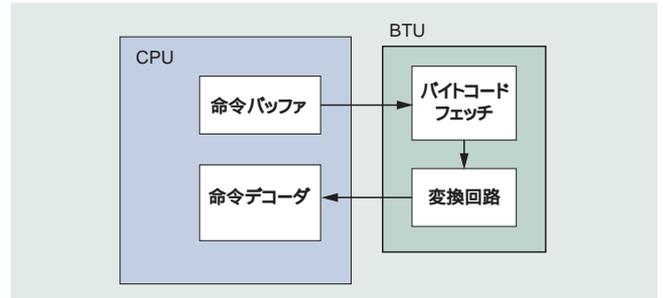
作すべき回路が異なるゲートドパイプライン制御では、命令解読ステージで命令が解読され、実行すべきパイプラインがわかった段階で、ステージ活性化信号が有効となり、1サイクルずつ後段に伝えられる。この信号はパイプラインステージ間のラッチのクロック信号とAND(論理積)演算され、パイプラインが選択されなかった場合にはクロック入力を抑える。これにより、命令実行に必要な部分だけが活性化され、不要な回路動作が抑止される。

以上のような低消費電力化方式を導入しながら高性能化が進められている。最近のSuperHの性能と消費電力を図1.1に示す。1999年に180 nm技術を用いて開発したSH-4では、240 MIPS、240 mWで1,000 MIPS/Wを達成した。さらに、2004年に130 nm技術を用いて製品化したSH-Xでは、360 MIPS、80 mWで4,500 MIPS/Wに到達した。このMIPS/W値は、パソコンやサーバ向けのマイクロプロセッサに比べて2けた程度高く、きわめて電力効率が良いことを示している。

4.2 Javaアクセラレータ

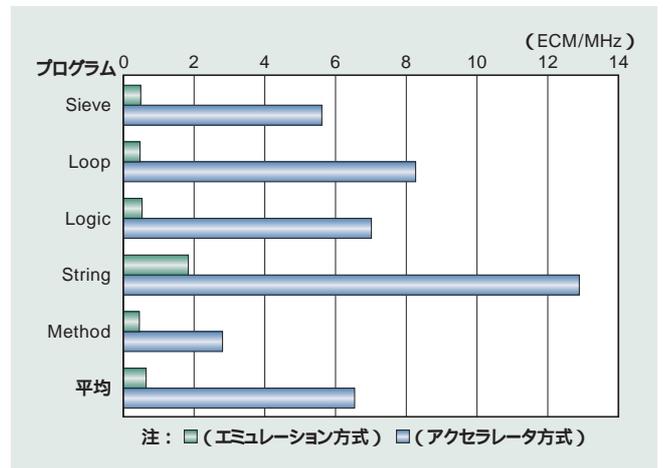
プログラミング言語にはさまざまな種類があるが、最近のデジタル民生機器ではポータビリティが良いことから、Javaを使用することが多い。このため、Javaで記述されたプログラムを低消費電力で高速に実行できれば大きなメリットが得られる。

2004年に製品化されたSH-Xでは、Javaのバイトコードを直接実行することができるアクセラレータ、BTU(Byte-Code Translation Unit)を開発した。BTUをCPUに接続することで、Javaの159個の基本バイトコードと6個の拡張バイトコードが直接実行される。BTUはCPUの命令バッファと命令デコーダの間に置かれ(図1.2参照)、BTUは命令バッファからバイトコードをフェッチ(読み出し)して、



注:略語説明 BTU(Byte-Code Translation Unit)

図1.2 Javaアクセラレータの構成
Javaのバイトコードは、BTUによってSHアーキテクチャの命令列に直接変換される。



注:略語説明 ECM(Embedded Caffeine Mark)

図1.3 性能比較
BTUを用いることにより、従来のエミュレーション方式に比べJavaプログラムが約10倍高速化される。

変換回路がそのバイトコードをSHアーキテクチャの命令列に変換する。変換後の命令列はCPUで通常と同じように実行される。バイトコードを直接ハードウェアで行うために、ソフトウェアでバイトコードを解釈してから進める従来のエミュレーション方式に比べて、Javaプログラムの実行が高速化される。また、命令フェッチのメモリアクセスが減ることにより、消費電力も削減される。

また、JavaのベンチマークプログラムであるECM(Embedded Caffeine Mark)の実行において、このアクセラレータ方式を用いた場合には、従来のエミュレーション方式に比べ約10倍の性能向上が確認されている(図1.3参照)。

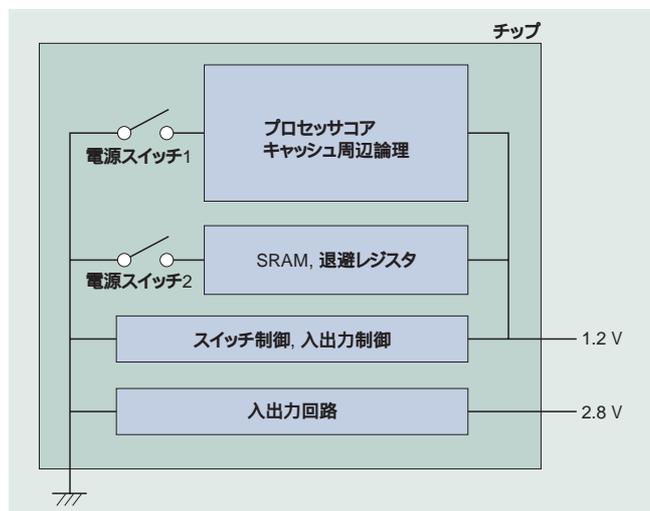
5 漏れ電流の削減

電池駆動のデジタル機器では、マイクロプロセッサの低消費電力化は、動作時だけでなく、スタンバイ時にも必要となる。例えば、携帯電話では数百時間という待機時間が要求されるが、このときにもマイクロプロセッサには電池から電源が供給され、必要なときにすぐ立ち上げられるよう

にしている。このため、スタンバイ時にマイクロプロセッサに流れる電流は100 μ Aといった、きわめて小さな値が求められる。通常、デジタルLSIはCMOS回路で構成されている。従来、この回路方式では動作停止状態で漏れ電流がほとんど流れなかったが、LSIの微細化が進むにつれ、漏れ電流が大きくなっている。最近の130~90 nm技術ではチップ全体で数ミリアンペアの漏れ電流が流れてしまうので、この値を2けた程度下げることが必要である。

2004年に携帯電話用アプリケーションプロセッサとして開発したSH-Mobile3では、新しいスタンバイ方式を導入した。新しい方式には、内部状態を保持するRスタンバイ状態と保持しないUスタンバイ状態を持ち、チップ内部には2種類の電源スイッチを設けている(図14参照)。Rスタンバイに入るとチップ内部の主要なレジスタの内容がSRAM (Static Random Access Memory) に退避され、電源スイッチ1がオフとなる。さらに、Rスタンバイ時にはSRAM部の電源電圧を自動的に下げて、この部分の漏れ電流を極力小さくすることにより、スタンバイ時の電流を100 μ A以下に下げることが可能にした。退避情報を内蔵SRAMに保持しているために動作状態への回復時間も3 msと速いので、キー入力待ちでRスタンバイに入り、キーが押された瞬間に動作状態に戻るといった切替も可能となる。一方、Uスタンバイでは二つの電源スイッチをオフにして待受け時の電流を10 μ A程度まで下げることができる。

SH-Mobile3のチップ写真を図15に示す。SH-Mobile3ではチップ内部にゲート幅が約20~80 μ mのきわめて長い電源スイッチ用トランジスタを設けている。電源スイッチを動作状態に応じてオン・オフすることにより、漏れ電流を削減し、電池使用時間を大きく延ばすことができる。



注:略語説明 SRAM(Static Random Access Memory)

図14 チップ内部の電源構成
チップ内部の二つの電源スイッチを制御することにより、スタンバイ時の漏れ電流を削減する。

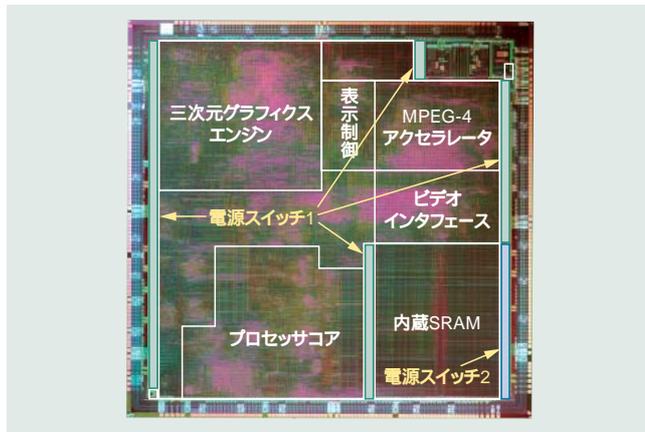


図15 SH-Mobile3のチップ
チップ内部には、二つの電源スイッチ用トランジスタが配置されている。

6 おわりに

ここでは、日立製作所のマイクロプロセッサの開発、アーキテクチャ設計、高性能化と低消費電力化技術、および実装例について述べた。

1990年代以降、多方面の分野で高度なデジタル化が進み、そこで使われるマイクロプロセッサへの高性能化と低消費電力化への要求が高まっている。日立製作所は、このニーズに応えるため、アーキテクチャ、方式、回路、プロセス面から総合的に技術開発を進めている。今までは、微細加工が進むにつれて電源電圧が下がり、それなりの低消費電力化も達成できた。しかし、これからは、電源電圧をさらに下げることや、漏れ電流の増加という弊害なしにトランジスタ特性を向上することは期待できない。一方、このような状況でも、マイクロプロセッサやSoC (System-on-Chip)への性能や多機能化についての要求は上がっている。今後は、このような課題を解決していくために、応用システムをよく把握しながら、ソフトウェア、アーキテクチャ、方式、回路面からの多面的な検討を進めていく考えである。

参考文献

- 1) 荒川, 外: マルチメディアを支えるSuperH RISC engineとメモリ, 日立評論, 81, 10, 619~622(1999.10)
- 2) 内山, 外: ネットワークとマルチメディアに対応するCPU技術, 日立評論, 82, 10, 633~636(2000.10)
- 3) 金井, 外: 携帯電話用アプリケーションプロセッサ "SH-Mobile", 日立評論, 84, 10, 647~650(2002.10)



内山 邦男

1978年日立製作所入社, 中央研究所 主管研究員
現在, マイクロプロセッサやSoCの研究開発に従事
工学博士
IEEE会員, 電子情報通信学会会員
E-mail: uchiyama@crl.hitachi.co.jp