

## 「招待講演」 組込みプロセッサの現状と課題 —高性能と低電力の両立に向けて—

内山 邦男

(株)日立製作所 中央研究所 〒185-8601 東京都国分寺市東恋ヶ窪 1-280

E-mail: uchiyama@crl.hitachi.co.jp

あらし 近年、携帯電話、デジタルカメラ、カーナビなどに見られるように、デジタル民生機器におけるマルチメディア化の動きが盛んである。これらの機器では、画像、オーディオ、グラフィクスなどのメディアデータを高速にしかも低電力で処理することが必要となる。この実現のために、高性能で低電力、低価格を追求したマイクロプロセッサや SoC が開発されている。本報告では実際の組込みプロセッサ開発事例を参考にしながら、高性能と低電力の両立を狙って開発された技術と今後の課題について述べる。

キーワード 組込みプロセッサ、マイクロプロセッサ、SoC、デジタル民生機器、マルチメディア、低電力技術

## Technological Trend of Embedded Processors —Achieving Both High-performance and Low-power—

Kunio UCHIYAMA

Central Research Laboratory, Hitachi, Ltd. 1-280 Higashi-koigakubo, Kokubunji, Tokyo, 185-8601 Japan

E-mail: uchiyama@crl.hitachi.co.jp

**Abstract** Embedded processors which are used in recent digital consumer appliances must satisfy the requirements of high-performance, low-power, and low price. The technological trend of such kind of embedded processors is studied from the view points of architecture, high-performance, and low-power technologies.

**Keyword** Embedded Processor, SoC, Digital Consumer Appliance, Multimedia, Low-power Technology

### 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はじまり、80 年代になるとパソコンやワークステーション向けに高性能化が追求された。90 年代に入ると民生機器のデジタル化が始まり、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りはじめた。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、車やロボットなどに内蔵される制御装置には、高性能・低電力・低価格なマイクロプロセッサが必ず使われている。

この高性能と低電力の両立を狙ったマイクロプロセッサは、90 年代に入り研究開発が盛んになり各社で製品化が行われてきた。本報告では日立製作所とルネサス テクノロジで共同開発されてきた SuperH の開発経緯を参考にしながら、この組込みプロセッサ開発の狙い、アーキテクチャ設計のポイント、主要な高性能化と低電力化技術、製品への実装例について概観し、現状の技術と今後の課題について述べる。

### 2. 開発の背景と狙い

90 年代のはじめ、家電品に組込まれるマイコンは、テレビ、クーラーなどの制御に使われていた。価格重視のために、その性能は 1MIPS 程度であった。一方、当時のワークステーションやパソコンでは 50MIPS 程度の高い性能のマイクロプロセッサが使われていたが、消費電力は数 W で、価格も数万円と高いものであった。この状況の中で、電子手帳、プリンタ、FAX 付き電話機など、民生機器のデジタル化が始まり、90 年代の中頃からは、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電力・低価格で高速に処理する必要がある。従来の制御用マイコンやパソコン/サーバ用マイクロプロセッサではこれを実現することは困難であった。このため、高性能、低電力、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)。このニーズを満たす

べく 80 年代後半から現在に至るまで SuperH ように高性能と低電力の両立を狙ったマイクロプロセッサの研究開発が進められている。

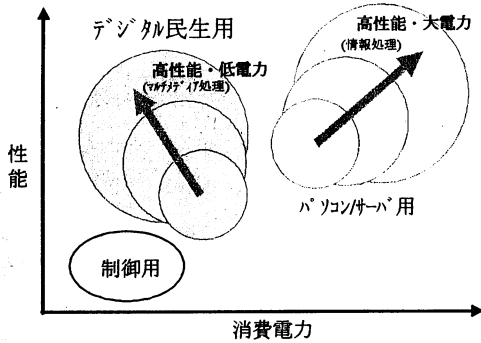


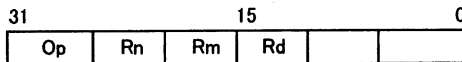
図 1 高性能・低電力マイクロプロセッサの位置づけ

### 3. アーキテクチャの設計

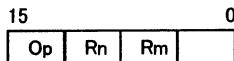
#### 3.1 16 ビット固定長命令セット

90 年代初頭のマイクロプロセッサのアーキテクチャは、可変長命令の CISC と 32 ビット固定長命令の RISC の 2 種類があった。前者は複雑な機能を持つ可変長命令を数多く持っているために高性能化のために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が、命令長が 32 ビット固定であるために、プログラムサイズが大きくなる欠点がある。

これらの問題を解決するために SuperH では 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、あるいは、SH と呼ぶ）が設計された（図 2）。32 ビット固定長の従来型 RISC に比べて半分の 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りビット数を圧縮している。



(1) 従来型 RISC アーキテクチャ



(2) SH アーキテクチャ

図 2 SH アーキテクチャの命令フォーマット

従来型 RISC に比べて、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% 縮小することを可能にした（図 3）。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る。

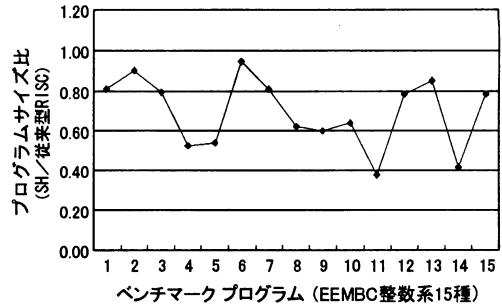


図 3 プログラムサイズの比較

また、マイクロプロセッサの CPU 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。95 年に開発された SH アーキテクチャのマイクロプロセッサ SH-3 では、当時のテクノロジーは 0.5  $\mu\text{m}$  技術であるが CPU の面積は 5mm<sup>2</sup> と小さい。最新の 130nm 技術ではこの部分は 3mm<sup>2</sup> 以下となっており、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向け拡張

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）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 MPEG4 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 や Dolby-AC3 など各種規格のオーディオ伸長処理を行っている。デジタル TV を受信する場合にも、H.264 や MPEG4 規格の映像データを伸張する必要がある。また、高品質なゲーム機能を実現するには、3 次元グラフィックス処理が必要となる。これらのマルチメディア処理では一般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 命令を導入した。

ゲームなど立体図形を扱う応用では、3 次元グラフィッ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や輝度計算がなされる。この計算をリアルタイムで行うために、図形が高精細になるにつれて膨大な計算量が必要となる。これに対応するために、図 4 に示すように、単精度浮動小数点演算を複数個実行できる 4 次元ベクトル命令を追加した。この FIPR 命令では 4 つの浮動小数点レジスタを一組の 4 次元ベクトル（FV<sub>m</sub>, FV<sub>n</sub>）として定義して、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 サイクル毎で実行することを可

能にした。

**命令記述**

FIPR FV<sub>m</sub>, FV<sub>n</sub>

**演算オペランド**

FV<sub>m</sub> = {FR[m], FR[m+1], FR[m+2], FR[m+3]}

FV<sub>n</sub> = {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:FIPR (Floating Point Inner Product)

図4 4次元ベクトル命令

画像やオーディオ処理では固定小数点形式のデータに対して積和演算を用いて各種の変換を行うことが多い。このため、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。図5はDSP命令を拡張する時の構成である。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メモリが追加される。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。

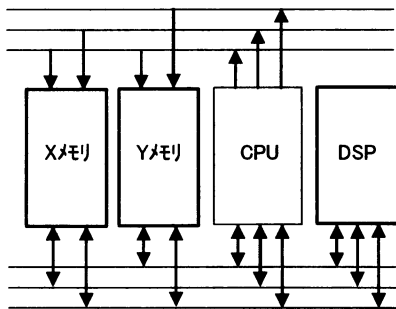


図5 DSPアーキテクチャ追加時の構成

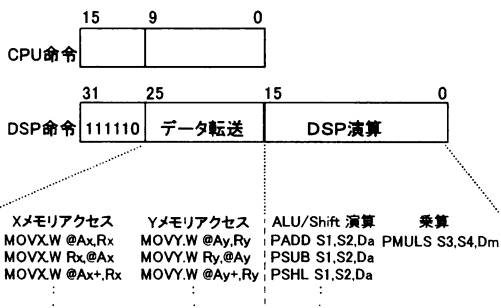


図6 DSP命令のフォーマット

図6にDSP命令のフォーマットを示す。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/シフタ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なる。この一つの命令で4つのオペレーション(2つのメモリアクセス動作、ALU/シフト演算、乗算)を実行することが可能であり、効率の良い信号処理やマルチメディア処理が可能となる。

4. 高性能化と低消費電力の両立

4.1 活性化率の低減技術

CMOS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、充放電の電力  $P \times F \times C \times V^2$  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する。このため、性能を上げながら、活性化率を下げる技術が必要になる。SuperHの開発では、いろいろな手段を用いてPを下げる工夫をしている。その内の2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年に製品化されたSH-2で導入された。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電力化が重要である。図7にその構成と動作を示す。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する。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。この方式により、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。

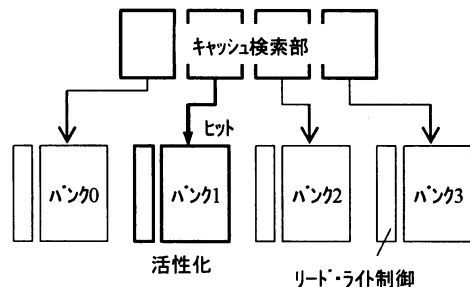


図7 選択起動型キャッシュメモリ

ゲートッド・パイプライン制御は、1999年に製品化された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、それを実行する回路を割当てる。命令ごとに動作すべき回路がことなる。図8に示すよ

うにゲーテッド・パイプライン制御では、D ステージで命令が解読され、実行すべきパイプラインが判った段階で、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 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が抑止される。これにより、命令実行に必要な部分だけが活性化され、不必要な回路動作が抑止される。

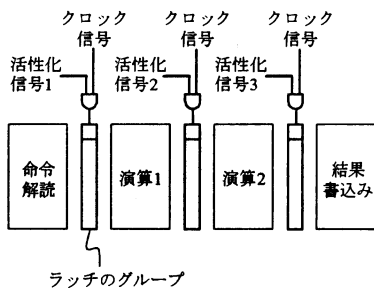


図 8 ゲーテッド・パイプライン制御

以上のような低電力化方式を導入しながら高性能化が進められている。図 9 は、最近の SuperH の性能と消費電力を示している。99 年に 180nm 技術を用いて開発された SH-4 では、240MIPS、240mW で 1000MIPS/W が達成された。2004 年に 130nm 技術を用いて製品化された SH-X では、360MIPS、80mW で 4500MIPS/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 桁程度高く、電力効率の良いマイクロプロセッサが実現されている。

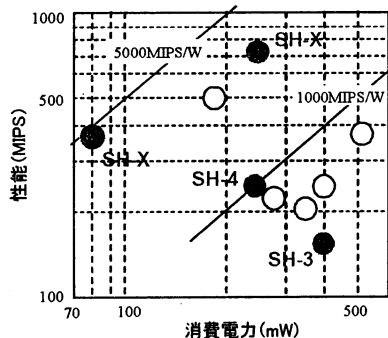


図 9 消費電力と MIPS 値

#### 4.2 Java アクセラレータ

最近のデジタル民生機器ではポータビリティの良さからプログラミング言語 Java でプログラムが記述されることが多い。このため、Java で記述されたプログラムを低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。図 10 に示すように、BTU は CPU の命令バッファと命令デコーダの間に置かれる。BTU は命令バッファからバイトコードをフェッチして、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。ソフトウェアでバイトコードを解釈して実行する従来のエミュレーション方式に比べて、バイトコードを直接ハードウェアで実行するために 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ため、電力も削減される。

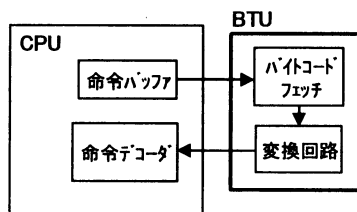


図 10 Java アクセラレータの構成

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、従来のエミュレーション方式に比べ本アクセラレータ方式を用いた場合には、約 10 倍の性能向上が確認されている (図 11)。

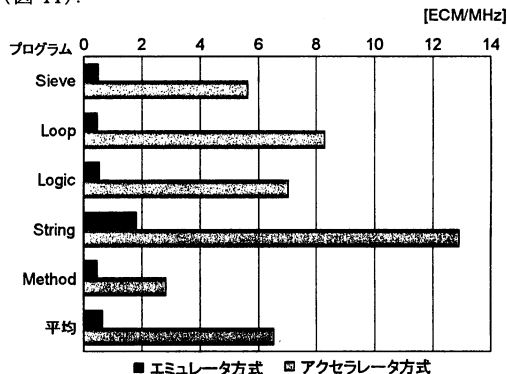


図 11 性能比較

#### 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。この時にもマイクロプロセッサには

電池から電源が供給され、必要な時にすぐ立ち上が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高々100マイクロアンペアといった極めて小さな値が要求される。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て、この漏れ電流が大きくなっている。最近の130nm～90nm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ため、この値を2桁程度下げることが必要である。

2004年に携帯電話向け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。この方式は、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つ。図12に示すようにチップ内部には2種類の電源スイッチを設けている。Rスタンバイに入ると時は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ようにした。これでスタンバイ時の電流を100マイクロアンペア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ミリ秒と速い。このため、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り替えも可能となる。一方、Uスタンバイでは2つの電源スイッチをオフにして待受け時の電流を10マイクロアンペア程度まで下げることができる。

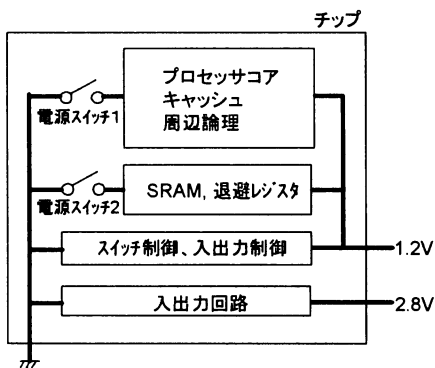


図12 チップ内部の電源構成

SH-Mobile3はチップ内部にゲート幅が約20～80μmの極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が削減でき、電池使用時間を大きく延ばすことができる。

## 6. おわりに

90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に対して、高性能化と低電力化の要求が高まっている。このニーズに応えるべくアーキテクチャ、方式、回路、プロセス面から総合的に技術開発を進められている。今までは、微細加工が進むにつれて電源電圧が下がり、それなりの低電力化も達成できていた。しかし、これからは電源電圧をさらに下げることによって多くの期待はできない。また、トランジスタ特性の向上も漏れ電流の増加という弊害無しには期待できない。このような状況でも、マイクロプロセッサやSoCに対する性能や多機能化の要求は上がっている。今後は、このような難しい問題を解いていく必要がある。これを克服するために応用システムを良く把握しながら、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必要がある。

## 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, 第 81 巻, 10 号, pp.619-622, 10 月, 1999 年.
- [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, 第 82 巻, 10 号, pp.633-636, 10 月, 2000 年.
- [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, 第 84 巻, 10 号, 10 月, pp.647-650, 2002 年.
- [4] 内山, “高性能・低消費電力マイクロプロセッサ,” 日立評論, 第 87 巻, 5 号, pp.89-94, 5 月, 2005 年.