

3

FR1000 におけるチップマルチ プロセッサアーキテクチャの紹介

Introducing the One Chip-multiprocessor FR1000



須賀 敦浩

富士通研究所
suga.atsuhiko@jp.fujitsu.com

今井 賢

富士通研究所
imai.satoshi-02@jp.fujitsu.com

田中 篤志

富士通研究所
tanaka.atsushi@jp.fujitsu.com

早川 文彦

富士通研究所
hayakawa.fumihi@jp.fujitsu.com

多湖 真一郎

富士通研究所
s-tago@jp.fujitsu.com

我々は、これまでコンピュータアーキテクチャで議論されている並列性の粒度を超える大きな並列処理を実行することに適したスレッドレベルのアーキテクチャを導入した FR1000 プロセッサを開発した。FR1000 プロセッサは、8 並列同時実行可能な VLW プロセッサコアを 4 個搭載したチップマルチプロセッサである。このようなマルチコアプロセッサを実行するために使用する OS としては、従来の μ ITRON をコア間通信のサービスコールに拡張した OS を開発した。これにより、従来のタスクベースのプログラミング手法によりマルチコアのプログラミングを行うことができる。この FR1000 の使用例として MPEG2 MP@HL のビデオストリームをソフトウェアのみで実現した。MPEG2 の 1 枚のピクチャにおいて、1 つのラインを 1 つのコアがデコードを行うことにより、4 ライン同時に処理をすることができる。このようなプログラミングを行うことにより、大きな粒度の並列処理を実現することができる。

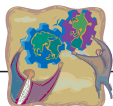
組み込みプロセッサと並列アーキテクチャ

近年のコンピュータアーキテクチャの研究は、90 年代初めに D. W. Wall により指摘されたプロセッサの性能限界¹⁾を超越することを目指して活発になった。90 年代の中頃には、Illinois 大学、Stanford 大学などを始めとする多くの研究者がさまざまな並列性の導入を提唱した。

一般的にコンピュータ上で実行されるプログラムは、制御フローとデータフローから構成される。プログラムを正しくかつ高速に実行するためには、この制御フローを満たしつつデータを高速に処理する必要がある。プログラムの高速な処理を動作周波数に依存することなく実現するには、一度に多くの処理を実行する“並列性”を導入することになる。プログラムの制御を変更する処理は、条件文や繰り返し文および関数などの呼び出しなどの際に発生する。プログラムを並列に実行するためには、
(1) 1 つの制御の中に存在する処理を極力多くする手法
(2) 異なる制御の処理は同時に実行する手法
がある。前者は Predicated Execution という手法が提案され制御依存関係をデータ依存関係に変換することにより命令レベルの並列性を拡大させる手法である。これは、

VLW などにより命令レベルの並列性を利用する場合に有効な手法である。一方、後者は、プログラム上の繰り返し文や関数を 1 つの塊「スレッド」として処理する手法である。Stanford 大学²⁾のチーム等により研究された。これは、今日では、スレッドレベル並列として知られている。スレッドレベル並列性を実現するプロセッサアーキテクチャとしては、1 つのプロセッサコアで同時に複数のスレッドを実行する SMT 手法と 1 チップに複数のプロセッサコアを搭載するチップマルチプロセッサ (CMP) 手法がある。今日このスレッドレベルの並列性を取り込んだコンピュータアーキテクチャの研究は、日本の大学においても、さまざまな研究機関で研究されている。また、産業界においても、近年は MP98³⁾、Cell⁴⁾ に代表される CMP を採用するアーキテクチャが数多く提案されている。

我々は、高解像度のリアルタイム動画処理、イメージング処理に代表されるような低消費電力が要求される組み込みプロセッサの分野に対しても、チップマルチプロセッサ技術を適用させることを考えた。一般的に組み込みシステムでは、機能ごとに特定のハードロジックや DSP を割り当て、システムとしてもそのようなハードマ



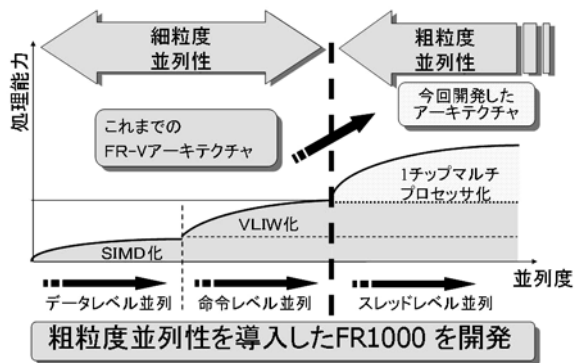
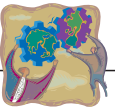


図-1 導入したマルチコアアーキテクチャ

● 1チップマルチコアプロセッサ



● FR550 シリーズ 8並列VLIWプロセッサ

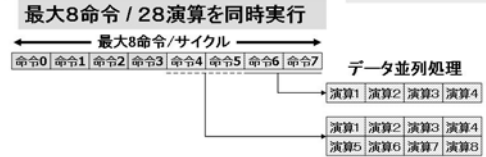


図-3 FR1000 プロセッサの並列性

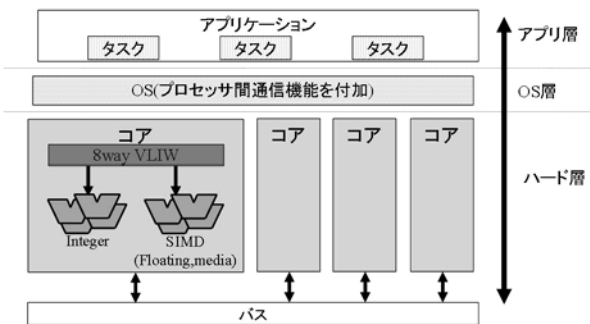


図-2 FR1000 アーキテクチャ

アーキテクチャ

図-2 にチップマルチプロセッサである FR1000 のアーキテクチャの概要を示す。

FR1000 プロセッサは、命令レベル並列性、データレベル並列性およびプロセッサ並列性の3つの異なる並列性を導入するマルチコアプロセッサである。プロセッサコアにおける命令レベル並列性を実現する手法としては、ハードウェアが同時に実行できる命令を抽出するスーパーパスカラ方式とコンパイラなどのソフトウェアが同時に実行できる命令を指定する VLIW 手法がある。我々の命令レベル並列性は、後者を採用している。本プロセッサは、8 並列同時実行可能な VLIW プロセッサコアを4個搭載する。一方、データレベルの並列性は、1つの命令が複数の演算を指示する SIMD (Single Instruction Multiple Data) 方式を取り入れている。本 VLIW プロセッサコアは、16ビット固定小数点演算命令を同時に4演算実行する SIMD 命令を考慮した命令セットを持つ。図-3 に示すように、それぞれのコアは、同時に28演算を実行することができる。したがって、FR1000 プロセッサとしては、1サイクルで112演算を同時実行することができる。

たは DSP を複数保持するということが多い。

ここでいう機能とは、プログラム上1つの意味を持つ粒度になることが多く、上述の繰り返し文や関数といった通常のコンピュータアーキテクチャでいうところの「スレッド」の粒度よりも大きな塊になる。コンピュータアーキテクチャの分野では、命令レベルの並列性よりも小さい並列性であることから細粒度といい、スレッドレベルの並列を粗粒度という。我々が導入した並列性は、通常の粗粒度よりも遥かに大きな塊を対象にしているので、「超粗粒度」といえる。

我々は、SIMD 命令によるデータレベル並列性、VLIW (Very Long Instruction Word) 方式による命令レベル並列性といった細粒度の並列性を継承しつつ、超粗粒度の並列性を抽出するために、4つの VLIW プロセッサコアを1チップに搭載する方式 (CMP アーキテクチャ) を採用した FR1000 を開発した (図-1)⁵⁾。このプロセッサは、上述のようにコア内は、細粒度あるいは中粒度の並列処理を使用したプログラムの塊を4個同時に実行できるような複数の粒度を同時に実現するアーキテクチャを提案するものである。

本稿では、CMP アーキテクチャを採用した FR1000 のチップアーキテクチャおよび、CMP 向けソフトウェアアーキテクチャについて述べる。

チップマルチプロセッサである FR1000 プロセッサのそれぞれのプロセッサコアは、シングルプロセッサ用の OS を動作させることができる。我々は、μITRON ベースの OS を利用している。μITRON では、プログラムの1つの固まりをタスクという。このタスクを制御したりタスク間の通信を行うためにサービスコールが用意されている。我々は、このシングルコアの OS の仕組みをマルチコア用に拡張した。異なるプロセッサコアで実行しているタスク同士は、専用のサービスコールを用いて通信することができる。このようなアーキテクチャをとることにより、アプリケーション層から見るとシングルコアプロセッサ上のマルチタスクプログラミング手法とほぼ同様の手法でマルチコア向けのプログラミングを実装することができる。

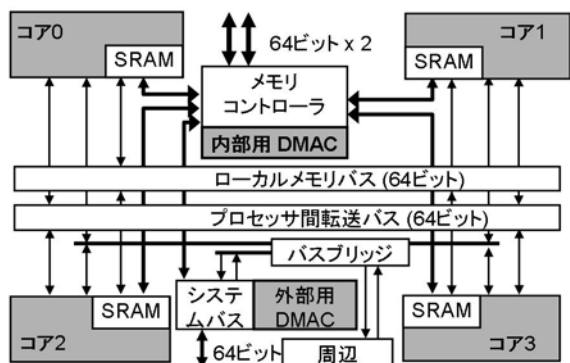


図-4 FR1000 ブロック図

FR1000 インプリメントアーキテクチャ

FR1000 のブロック図を **図-4** に示す。FR1000 は、4つのプロセッサコア、メモリコントローラ、システムバス、DMA コントローラ、周辺機器およびこれらを接続するバスで構成されている。

FR1000 プロセッサに搭載されるプロセッサコアは、整数演算命令と浮動小数点演算命令および16ビットの固定小数点演算命令であるメディア命令を実行する。FR1000 プロセッサは533MHz動作時のピーク性能は59.7GOPSである。

チップマルチプロセッサ方式を採用しているプロセッサは、複数のコアがバスにアクセスするため、バスで競合が発生する傾向にある。FR1000 では4つのプロセッサコアおよび2つのDMA コントローラがメモリアクセスを行う際に競合する。この競合を回避するために、FR1000 では**図-4**に示すバスアーキテクチャになっている。

(1) 2チャンネルメモリインタフェース

FR1000 では、4つのプロセッサコアがメモリにアクセスしたとしても、性能低下を避けるため、2チャンネルのメモリインタフェースを実装した。各コアはメモリインタフェースと専用のクロスバーで接続し、複数のコアが同時にメモリにアクセス可能とした。

(2) コア内蔵SRAM

外部メモリアクセス削減のために、コアのローカルメモリとしてSRAMを内蔵した。すべてのコアはすべての内蔵SRAMと専用のクロスバーで接続し、複数のコアが同時に内蔵SRAMにアクセス可能とした。

(3) コア間通信向け専用バス

コア間通信を高速に行うために、専用のコア間通信制御機構を搭載した。

(4) 64ビットシステムバスインタフェース

高解像度の画像データなど、大容量のデータを転送するために、64ビット幅のシステムバスインタフェースを搭載した。

(5) DMA コントローラ

チップマルチプロセッサ方式のプロセッサは、プログ

ラムを実行する準備としてメモリ上のデータをローカルメモリに転送することが多くなる。組み込みプロセッサは、このデータの転送と同時に、すでに処理の終わったメモリ上のデータを一定の時間内に外部に転送する必要がある。そのような複数のデータ転送を実現するために、2種類のDMA コントローラを搭載した。1つは、プロセッサコア間、プロセッサコア-外部メモリ間、メモリーメモリ間転送を行う内部用DMACである。もう1つは、メモリーシステムバス間の転送を行う外部用DMACである。内部用DMACと外部用DMACのチャンネル数は、それぞれ16チャンネルである。このような2種類のDMAアーキテクチャを用いることにより、従来のシステムソフトにおけるドライバなどがDMAを使用しながらも、アプリケーションプログラムは、チップマルチプロセッサ特有のメモリ転送のためにハードウェアリソースを十分に使用することができる。

ソフトウェアアーキテクチャ

チップマルチプロセッサを使用するプログラムは、複数のプロセッサコアに処理を割り当てるために、ソフトウェアを分割することになる。プロセッサの持つ性能を發揮するためには、ソフトウェアの分割において以下の2点を考慮する必要がある。

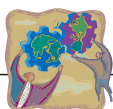
- (1) 各コアの処理量を均等にすること
- (2) コア間通信の処理量を少なくすること

我々は、このようなプロセッサのソフトウェアの例として、MPEG2復号化ソフトウェアに関して、チップマルチプロセッサ向けアーキテクチャを開発した。

図-5に示すように、MPEG2復号化ソフトウェアをMPEG2が規定するスライス層で分割する。各コアにスライス1ライン分を割り当てることで、各コアの処理時間を均等にすることができる。また、MPEG2では同一フレーム内のスライス間でデータ依存がない。そのため、スライス層で分割した場合は、プロセッサ間通信の処理時間を少なくすることができる。

チップマルチプロセッサ向けMPEG2復号化ソフトウェアのフローチャートを**図-6**に示す。コア0が行う処理は、前処理、ピクチャヘッダ解析、他コアの起動、スライス層以下の復号化、他コアの完了待ちである。コア0以外のコアが行う処理は、スライス層以下の復号化とコア0との同期である。

プロセッサの性能を最大限に引き出すためには、メインメモリへのアクセスを削減する必要がある。MPEG2復号化の処理においては、復号後のデータをいったん内蔵SRAMに格納し、その後内部DMACが内蔵SRAM



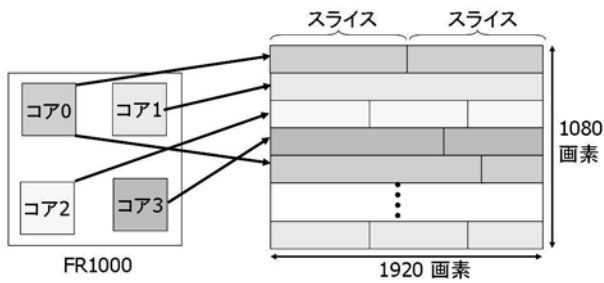
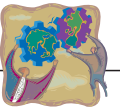


図-5 スライス層分割

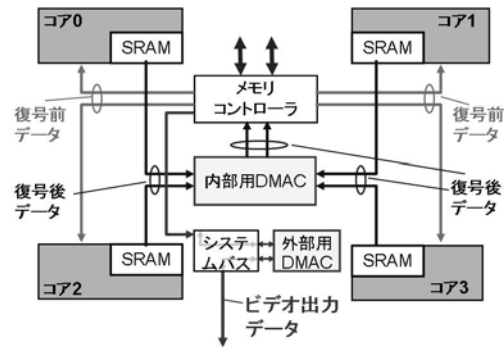


図-7 MPEG2 復号化時の DMAC 動作状況

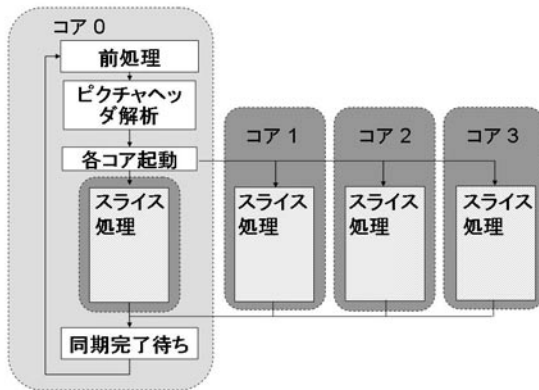


図-6 マルチコア向け MPEG2 復号化フローチャート

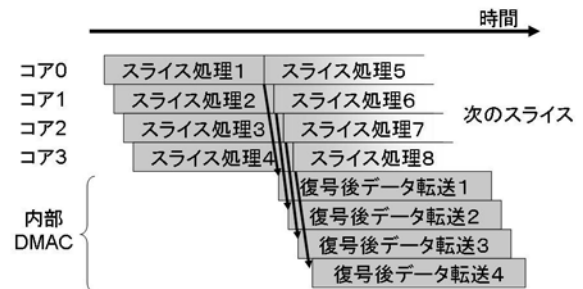


図-8 MPEG2 復号化時の内部 DMA 転送内容

からメインメモリへバースト転送する方式がとれる (図-7)。この内部 DMAC によるデータ転送はスライス処理後に1つのスライスを単位として行われる (図-8)。MPEG2 の復号化の処理は、それなりの工夫をすることにより比較的高い命令レベル並列度を持つプログラミングをすることができる。1つのピクチャは、複数のスライスから構成される。我々が提案しているチップマルチプロセッサアーキテクチャは、内蔵 SRAM を上手く利用しながら、細粒度の並列性を持つソフトウェアの塊を同時に複数実行するというプログラミングに最適である。

チップマルチプロセッサの実行性能

チップマルチプロセッサを用いた際の動作時性能の例として、データ転送を行う際の性能と前述の MPEG2 の復号処理の性能を説明する。

FR1000 プロセッサの DMA 転送の性能に関して、異なるコアにある内蔵 SRAM 間の転送とメモリ上のある領域のデータをメモリ上の別の領域へ転送を行いながらメモリと外部デバイスとの間のデータ転送を行うという複数のデータ転送時の動作性能を示す。このような複合的なデータ転送は、実際のシステムを動作させるためには、よく使われるケースである。

本プロセッサは、メモリ間転送およびメモリ-外部デ

バイス間転送ともにそれぞれ 1GB/s の性能を実現し、たとえ競合が発生した場合においても 1GB/s の転送性能を得ることができる (図-9)。実評価ボード上でのこの測定結果は、チップマルチプロセッサ向けのバスアーキテクチャとしては、前述の2種類の DMA と各ハードウェアリソースをクロスバー接続という FR1000 プロセッサで採用したバスアーキテクチャの方式が、実際の動作時に効果を発揮することを示している。

本プロセッサで MPEG2 MP@HL の復号化の性能を示す。復号化するデータ形式としては、画像サイズは 1920 × 1088、ビットレートは 20Mbps を用いている。

本プロセッサ上で、前述のようなプログラミングを行った MPEG2 MP@HL の復号化処理は、約 75% の処理能力で実現することができる (図-10)。

本プロセッサにおける1つのコアで MPEG2 MP@ML (画像サイズ 720 × 480、ビットレート 8Mbps) を復号化するのに必要な周波数は 190MHz である。また、1つのコアで MPEG2 MP@HL を復号化するためには、1GHz 程度の周波数が必要である。

このように、高い処理能力を持つプロセッサコアを複数搭載するチップマルチプロセッサは、シングルプロセッサでは数 GHz 程度の動作周波数が必要であったソフトウェアの処理を比較的低い周波数で実現することができる。低い消費電力で多くの機能を実現することができ

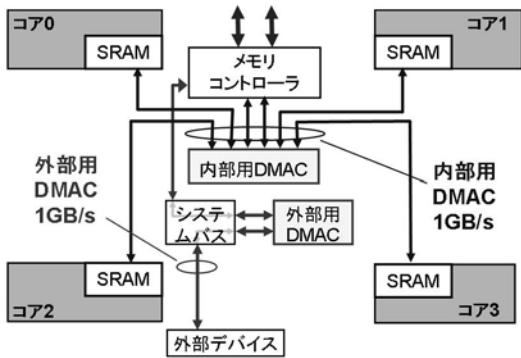


図-9 データ転送性能

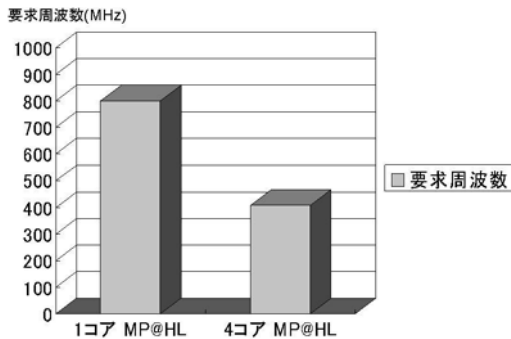


図-10 MPEG2 MP@HL 性能評価結果

るので、本プロセッサアーキテクチャは、低価格で高性能を求められる組み込みシステムに適している。

まとめ

本稿では、チップマルチプロセッサの例として、我々が開発した8並列同時実行可能なVLIWプロセッサコアを4個内蔵したチップマルチプロセッサを説明した。FR1000プロセッサは、チップマルチプロセッサ方式を採用するマルチコアプロセッサであり、各コアに128KBの内蔵SRAMを搭載している。

FR1000プロセッサを用いたソフトウェアの例としては、MPEG2 MP@HLの復号化処理を約400MHzで実現することができ、そのときの消費電力は3Wである。MPEG2 MP@MLの約6倍の処理量があるMPEG2 MP@HL復号化を、およそ2倍程度の消費電力で実現し、高性能化と低消費電力化を同時に達成している。

このように、高い処理能力を持つプロセッサコアを複数搭載するチップマルチプロセッサは、シングルプロセッサでは数GHz程度の動作周波数が必要であったソフトウェアの処理を比較的低い周波数で実現することができる。組み込みシステムは、低い周波数と適正な価格で機能を実現しなくてはならない。チップマルチプロセッサの技術は、このような組み込みシステムの要件を満た

コア	4コア 8-way VLIW アーキテクチャ
メモリ	32KB+32KB/ コア (D- キャッシュ, I- キャッシュ) 128KB/ コア (ローカルメモリ)
DMA コントローラ	16 チャンネル (内部転送), 16 チャンネル (外部転送)
インターフェース	メインメモリ IF266MHz 64bit × 2ch システムバス 178MHz 64bit
テクノロジー	90-nm CMOS, 9層メタル
トランジスタ数	28M (ロジック), 55M (メモリ)
動作周波数	533MHz@1.2V
消費電力	3.0W@1.2V, 533MHz
パッケージ	900ピンFCBGA

表-1 チップ仕様

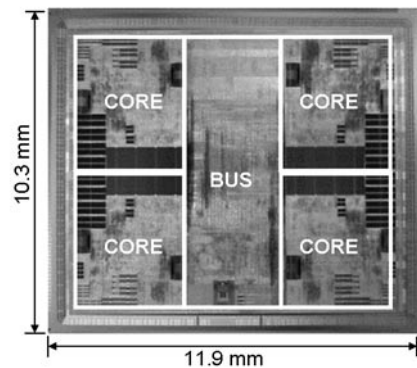


図-11 1チップマルチプロセッサのチップ写真

す技術であることを本稿では示した。

しかしながら、ここで紹介したようなチップマルチプロセッサの潜在能力を発揮するためには、アプリケーションを機能ごとに分割するコーディングスタイルが必須である。今後、さまざまなアプリケーションの並列化の研究が発展することを願う次第である。

最後に、本プロセッサの諸元を表-1に、チップ写真を図-11に示す。本プロセッサは、90nm CMOS 9メタル層のプロセス、900ピンのフリップチップパッケージを用いて開発し、チップサイズは11.9mm × 10.3mmである。プロセッサコアは533MHz、メモリインターフェースは266MHz、システムバスインターフェースは178MHzで動作する。533MHz動作時のピーク性能は59.7GOPSである。

参考文献

- 1) Wall, D. : Limits of Instruction-Level Parallelism, WRL Technical Note TN-15 (Dec. 1990).
- 2) Oplinger, J., Heine, D. and Lam, M. S. : In Search of Speculative Thread-Level Parallelism, Proceedings of the 1999 International Conference on Parallel Architectures and Compilation Techniques (PACT '99), Newport Beach, CA (Oct.1999).
- 3) Edahiro, M., Matsushita, S., Yamashita, M. and Nishi, N. : A Single-Chip Multiprocessor for SmartTerminals, IEEE MICRO, pp.12-20 (July 2000).
- 4) Pham, D. et al.: The Design and Implementation of a First-Generation CELL Processor, ISSCC Dig. Tech. Papers, pp.184-185 (Feb 2005).
- 5) Shiota, T. et al.: A 51.2GOPS 1.0GB/s-DMA Single-chip Multi-processor Integrating Quadruple 8-way VLIW Processors, ISSCC Dig. Tech. Papers, pp.18-19 (Feb 2005).

(平成 17 年 12 月 12 日受付)

