

富士通 SPARC64™ VII プロセッサ

Fujitsu Limited

Release 1.0. 1 June 2008

Copyright© 2008 Fujitsu Limited, 4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki, 211-8588, Japan. All rights reserved.

This product and related documentation are protected by copyright and distributed under licenses restricting their use, copying, distribution, and decompilation. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization of Fujitsu Limited and its licensors, if any.

The product(s) described in this book may be protected by one or more U.S. patents, foreign patents, or pending applications.

TRADEMARKS

SPARC® is a registered trademark of SPARC International, Inc. Products bearing SPARC trademarks are based on an architecture developed by Sun Microsystems, Inc.

SPARC64 is a registered trademark of SPARC International, Inc., licensed exclusively to Fujitsu Limited.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Sun, Sun Microsystems, the Sun logo, Solaris, and all Solaris-related trademarks and logos are registered trademarks of Sun Microsystems, Inc.

Fujitsu and the Fujitsu logo are trademarks of Fujitsu Limited.

All other company, product names mentioned may be trademarks or registered trademarks of their respective holders and are used for identification purpose only.

This publication is provided “as is” without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or noninfringement. This publication could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein; these changes will be incorporated in new editions of the publication. Fujitsu Limited may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

目次

1	SPARC64 シリーズについて	1
2	SPARC64 VII 概要	1
3	SPARC64 VII のマイクロアーキテクチャ	2
3.1	マイクロアーキテクチャ詳細	2
3.2	命令フェッチ部	3
3.3	命令実行部	3
4	キャッシュシステム	5
5	RAS 機能 (Reliability, Availability, Serviceability)	7
5.1	内蔵 RAM の RAS	7
5.2	内蔵レジスタと演算機の RAS	8
5.3	同期一括更新方式と命令リトライ	8
5.4	サービス性の向上	9
6	むすび	10
7	参考文献	10

1 SPARC64 シリーズについて

SPARC64 シリーズは富士通が開発している UNIX サーバ用 SPARC プロセッサです。SPARC64 V では、メインフレームクラスの高信頼性技術と GHz を超える周波数を実現しました。このプロセッサは富士通の PRIMEPOWER サーバに使用されました。そして SPARC64 VI は、SPARC64 V をベースに 2 コア×2 スレッド化することで高スループット化を実現しました。このプロセッサは SPARC Enterprise サーバに使用されています。

2 SPARC64 VII 概要

SPARC64 VII は、富士通が開発した SPARC64 シリーズの最新プロセッサです。富士通の 65nm テクノロジーを用い、動作周波数は 2.5GHz、チップサイズは 21.3mm x 20.9mm です。チップは 4 コアを内蔵し、共有の 6MB の L2 キャッシュを持つ構造となっています。動作消費電力は最大 135W です。

SPARC64 VII の設計に当たっては、現行の SPARC64 VI の高性能・高信頼性を保ちつつ、スループット性能の大幅強化を図りました。スループット性能強化のために内蔵するコア数を 2 から 4 に倍増し、さらにマルチスレッド機構を VMT から SMT に変更しました。L2 キャッシュは 4 コアで共用する構成とし、4 つのコアへデータ供給を行うためにスループットを倍増させました。また特に HPC 分野に向け、ハードウェアバリアと呼ぶコア間高速同期機構を実装しました。

一方バスプロトコルを現行 SPARC64 VI と同一にすることで、SPARC64 VI から SPARC64 VII への CPU モジュール単位でのアップグレードを可能としています。

3 SPARC64 VII のマイクロアーキテクチャ

SPARC64 VII のマイクロアーキテクチャについて概要を述べます。コア部のパイプラインの基本構造は SPARC64 VI と同一ですが、SPARC64 VII では、マルチスレッドの実装を VMT (Vertical Multi-Threading) から SMT (Simultaneous Multi-Threading) へ変更しました。図 1 に示すとおり、4 つある各コアの上で二つのスレッドを同時に実行することが可能です。

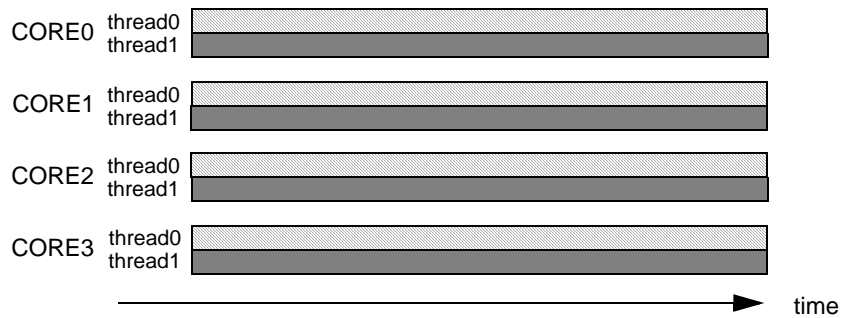


図 1 SPARC64 VII のマルチスレッド

SMT の設計にあたっては、スレッド間の相互干渉をできる限り除くことを主眼におきました。両スレッドが動作している時は、原則としてハードウェアリソースをスレッド毎に分離する構成としています。逆にどちらかのスレッドがアイドル状態のときは、他方のスレッドが一部のリソースを除き両スレッドのリソースを使用可能とすることで、シングルスレッドの性能向上を狙いました。

コアのパイプラインは、両スレッドで共用する構成となっています。但し一方のスレッドがパイプラインストールした場合も、他のスレッドの処理が停滞しないように制御を行っています。命令フェッチ / 命令デコード / コミットステージにおいてはサイクルごとにどちらか一方のスレッドを選択します。

3.1 マイクロアーキテクチャ詳細

以下にマイクロアーキテクチャの詳細について述べます。

SPARC64 VII のコアは、図 2 に示すように、命令フェッチ部と命令実行部に分かれ、命令フェッチ部には命令専用の 1 次キャッシュ (L1I キャッシュ) を含み、実行部にはオペランド専用の 1 次キャッシュ (L1D キャッシュ) を含んでいます。

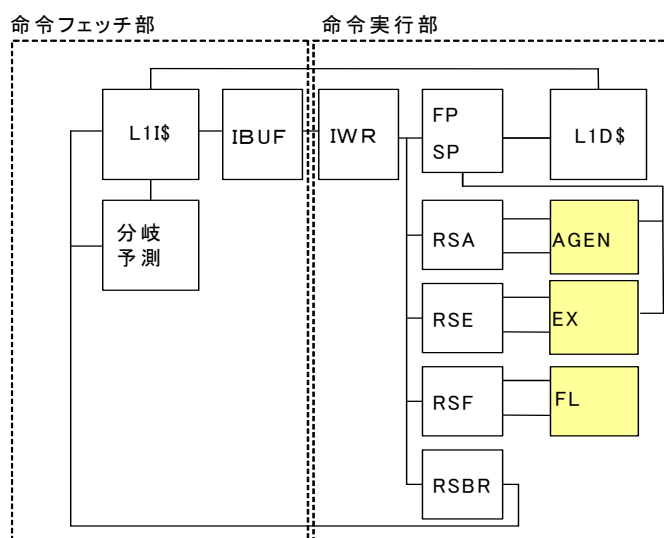


図2 SPARC64 VII コアの機能図

3.2 命令フェッチ部

命令フェッチ部は、命令実行部とは独立して動作し、分岐予測に従って将来実行する命令列を命令バッファ **IBUF** (**I**nstruction **B**U**F**fer)に取り込みます。**IBUF**の容量は256バイトであり、最大64命令の保持が可能です。両スレッドが動作している時は、命令バッファをスレッドごとに均等に分割して使用します。

命令実行が滞った場合にも、**IBUF**が一杯になるまで命令フェッチは停止しません。逆に、キャッシュ・ミス等で命令フェッチが停止した場合にも、**IBUF**に命令がある限り、そこから命令を取り出して実行を継続可能です。命令フェッチは毎サイクル起動でき、一度に32バイトすなわち8命令を取り出します。命令実行のスループットはサイクルあたり最大4命令であり、命令フェッチは命令実行の2倍のスループットを確保しています。**IBUF**は命令フェッチと命令実行を分離(デカップリング)することで、大容量の命令1次キャッシュのレイテンシを隠蔽しています。

3.3 命令実行部

命令デコード および発行

命令デコードおよび発行ステージでは、**IWR**上の4つの命令を同時にデコードして、実行に必要な資源(各種リザーベーション・ステーション、フェッチポート・ストアポート、レジスタ更新バッファ)を決定します。そして、その資源に空きがあるかどうかをチェックして、空きがあれば、その資源を割り当てるとともに、0~63の範囲の命令識別子(**IID**, **I**nstruction **I**Dentification)を割り当てて命令を発行します。つまりインフライト命令の最大数は64命令です。一方両スレッド動作時は各スレッドの最大命令数は32命令です。同じサイクルにどちらかのスレッドの命令デコードを行い、交互にスレッドを切り替えます。

命令を発行するとその IWR を開放します。IWR のどのスロットにある命令であっても、リザーベーション・ステーション等の資源の割り当て制限はなく、また、命令種類の組み合わせの制限もありません。このため、資源の空きさえあれば命令を発行可能です。4 命令分の空きが無い場合にも、プログラム順序の命令から可能な分だけを発行します。このように命令発行の停滞を極力排除した方式により、どのようなバイナリコードであっても常に高い並列度を確保しています。

命令実行

デコードされた命令はリザーベーションステーションに登録されます。SPARC64 VII は、整数演算用リザーベーション・ステーション RSE(Reservation Station for Execution) と浮動小数点演算用リザーベーション・ステーション RSF(Reservation Station for Floating point) を装備しています。RSE と RSF は、演算器に対応して、それぞれ 2 つのキューに分かれています。すなわち演算用のリザーベーション・ステーションは、RSEA、RSEB、RSFA、RSFB の 4 種を備えています。リザーベーション・ステーションに格納された各々の命令は、ソースオペランドが用意できたものから、個々のリザーベーション・ステーション毎に順次対応する演算器に投入します。したがって 4 つの演算を同時に投入可能です。投入アルゴリズムは、基本的にはリザーベーション・ステーションにある命令のうち、投入可能で 1 番古い (Oldest Ready) ものを選択する方法を採用しています。しかしロード命令が更新するレジスタをソースオペランドとして演算するような場合には、ロード命令の結果が得られる以前に投機的に投入し、演算ステージで投機的な実行が良かったか否かを判断しています。これを投機的投入 (Speculative dispatch) と呼んでいます。投機的投入より、キャッシュアクセスのパイプラインのレイテンシを隠蔽して演算器の使用効率を高めています。

リザーベーション・ステーションとしては上記の RSE,RSF 以外に分岐命令用の RSBR(Reservation Station for Branch)、及びロードストア命令のアドレス計算用の RSA(Reservation Station for Address generation) があります。

命令コミット

アウト・オブ・オーダーで実行した結果は、すべてソフトウェアから見えない作業用レジスタである GUB、FUB に一旦格納します。プログラムの順序性を保証するために、GPR や FPR などのレジスタやメモリは、コミットステージでプログラム命令順序に従って (イン・オーダーで) 更新します。さらに、PC をはじめとする制御レジスタも、コミットステージで一括して更新します。これにより、正確な割り込みを保証し、いつでも実行中の処理のキャンセルが可能となっています。これを同期一括更新方式と呼び、分岐予測ミスに関わるやり直しを簡単にするばかりではなくて、後の章で説明するように、RAS の向上に寄与しています。同時にコミットできる命令数は最大 4 です。命令コミットステージは 2 つのスレッドで共用されており、サイクルごとにどちらか一方のスレッドを選択しコミット処理を行います。

4 キャッシュシステム

SPARC64 VII のキャッシュは、中容量の 1 次キャッシュ (L1 キャッシュ) と大容量の 2 次キャッシュ (L2 キャッシュ) の二階層で構成されています。

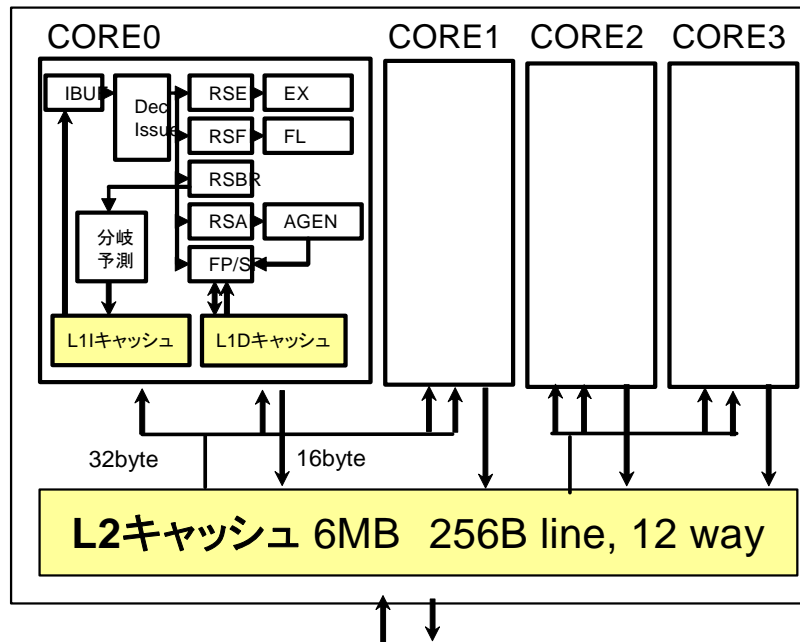


図3 SPARC64 VII コアとキャッシュ

L1 キャッシュは、命令専用のキャッシュ (L1I キャッシュ) とオペランド専用のキャッシュ (L1D キャッシュ) で構成され、いずれも容量 64K バイト / 2 ウェイ・セットアソシアティブ、ブロックサイズは 64 バイトです。L1D キャッシュは、4 バイトアドレス境界で 8 つのバンクに分割されており、2 つのオペランドの同時アクセスを実現しています。L1 キャッシュは、キャッシュインデックスに仮想アドレスを、キャッシュタグには物理アドレスを用いています (VIPT: Virtually Indexed Physically Tagged)。VIPT 方式では、メモリの同じ領域が異なる仮想アドレスでアクセスされると、キャッシュの別のインデックスに登録されて、一貫性を損なうおそれがあります (シノニム問題)。SPARC64 VII では L2 キャッシュとの連携によってハードウェアでシノニム問題を解決しています。

L2 キャッシュは、容量最大 6MB/12 ウェイ・セットアソシアティブ、ブロックサイズは 256 バイトであり、4 コアで共有されています。2 バンクのインタリーブ構成とすることで、1 サイクルあたり 64 バイトのデータを読み出し可能としています。L2 キャッシュから読み出したデータを L1 キャッシュに送るバスは、2 コアあたり 32 バイト幅、L1 キャッシュから L2 キャッシュにデータを送るバスは 1 コアあたり 16 バイト幅です。

L1 キャッシュ、L2 キャッシュとも、ライトバックと呼ばれるストア制御方式であり、書き込みはその階層のキャッシュに対して行います。ライトバック方式では、キャッシュ・ミスした場合には、ストアであってもメモリ上の旧データを一旦キャッシュに持ち込む必要がある反面、キャッシュ・ヒットしたストアは、キャッシュ上だけで完了します。一般的にストアの頻度はかなり高いために、ライトバック方式は、キャッシュ間のトラフィックやメモリアクセスのトラフィックが削減できる点が有利です。

一方ライトバック方式では最新データをキャッシュ上に握っているために、そのプロセッサでエラーが発生すると、被害がプロセッサ内に止まらずにシステムに及ぶ危険性があります。SPARC64 VII は強力な RAS 機能でこの問題に対処しています。

また SPARC64 VII では、新たにハードウェアバリア機構を実装しました。ハードウェアバリア機構は CPU チップ内でコア間の同期を取るための機構であり、従来のソフトウェアによる同期方式に比べて高速な同期処理が可能となっています。この機能は、特に HPC(High Performance Computing) 分野に有用です。

5 RAS 機能 (Reliability, Availability, Serviceability)

SPARC64 VII は、メインフレームに匹敵する RAS を実現しています。確実にエラーを検出すること、エラーの影響範囲を限定すること、回復処理を試みること、エラーの記録を残すこと、ソフトウェアに通知することなど、RAS の基本的な事項を徹底しています。これにより SPARC64 VII は、ミッションクリティカルな UNIX サーバのプロセッサとして、高度な信頼性、可用性、サービス性、データ保全性を提供しています。

5.1 内蔵 RAM の RAS

RAM はプロセッサ内で最もエラーの発生頻度が高い部分です。SPARC64 VII においては、RAM の 1 ビット・エラーはソフトウェアの介入なしに全てハードウェアで自動的に訂正するため、ソフトウェアへの影響は皆無です。

種別		エラー検出方法 保護方法	エラー訂正方法
L 1 命令キャッシュ	データ	パリティ	無効化及び再読み込み
	タグ	パリティ+二重化	二重化データの再書き込み
L 1 データキャッシュ	データ	SECEDED ECC	ECC による 1 ビットエラー訂正
	タグ	パリティ+二重化	二重化データの再書き込み
L 2 キャッシュ	データ	SECEDED ECC	ECC による 1 ビットエラー訂正
	タグ	SECEDED ECC	ECC による 1 ビットエラー訂正
命令 TLB		パリティ	無効化
データ TLB		パリティ	無効化
ブランチヒストリ		パリティ	分岐予測失敗からの回復

SECEDED : Single Error Correction Double Error Detection

L1 キャッシュ、L2 キャッシュ、TLB はウェイ単位での縮退が可能です。エラーの発生回数を機能単位ごとにカウントし、単位時間あたりのエラー回数が上限値を越えると縮退して、それ以降はそのウェイを使用しないようにします。縮退はハードウェアで自動的に行いますが、これに際してコヒーレンスを維持するための保証動作もハードウェアで自動的に行います。すなわち、L1D キャッシュの縮退されるウェイのダーティラインを全て L2 キャッシュに書き戻す動作、L2 キャッシュの縮退されるウェイのダーティラインをメモリに書き戻す動作はハードウェアで自動的に実行します。ウェイ縮退は、ソフトウェアにダメージを与えることなく実行され、ソフトウェアは処理速度の低下を除いて動作に影響を受けません。

5.2 内蔵レジスタと演算機の RAS

SPARC64 VII はレジスタや演算器もエラー保護することで、データ保全性に万全を期しています。

種別		エラー検出方法 保護方法
レジスタ	整数レジスタ	SECEDED ECC
	浮動小数点レジスタ	パリティ
	PC, PSTATE 等	パリティ

種別		エラー検出方法 保護方法
演算器	演算入出力レジスタ	パリティ
	加減算, 除算, シフト グラフィックス演算	パリティ予測
	乗算	パリティ予測 + 剰余チェック

整数アーキテクチャレジスタではより信頼度を高めるため、SPARC64 VII から ECC を採用しました。エラー発生時には ECC 回路でエラーを訂正します。浮動小数点アーキテクチャレジスタや他のレジスタはパリティビットが付加されています。また演算器にパリティ予測回路、剰余チェック回路などを設け、出力結果へとパリティを伝播させます。万一パリティエラーを検出した場合は、次に述べる通りハードウェアが自動的に命令を再実行し回復を試みます。この機能を命令リトライと呼んでいます。

5.3 同期一括更新方式と 命令リトライ

命令実行部で説明したように、SPARC64 VII は同期一括更新方式を採用しています。エラーが検出されると、その時点で実行途中の処理を全てキャンセルします。コミットまでの間の途中結果は廃棄可能であり、エラーに遭遇することなく完了した命令が更新した結果だけがプログラマブルな資源に残ります。従って、エラーによるプログラマブルな資源の破壊を防止するばかりではなく、エラー検出後にハードウェアで命令リトライを行うことが可能となります。ハングアップの場合にも、滞っている処理をいったん捨てて、はじめからやり直すことが出来るので、回復の可能性がります。

命令リトライはエラーをトリガーとして、自動的に起動します。リトライ中は、1 命令ずつ実行することで正常実行の可能性を上げ、正常に実行完了すると、通常の実行状態に自動的に復帰します。この間ソフトウェアに介入は不要であり、命令リトライが成功すれば、エラーのソフトウェアへの影響はありません。命令リトライは閾値に達するまで繰り返し、これを越えた場合には、エラーの発生を割り込みによって、ソフトウェアに通知されます。

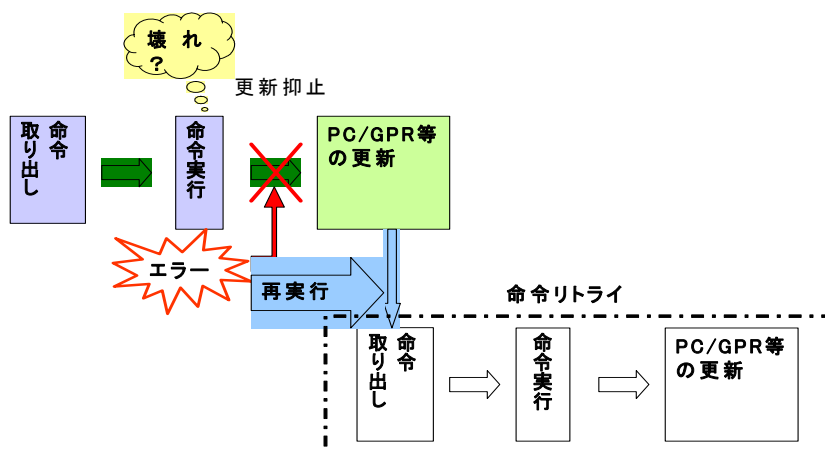


図4 SPARC64 VII 命令リトライ

5.4 サービス性の向上

SPARC64 VII は、ありとあらゆる個所にエラーのチェックを装備しています。エラーの発生時には、専用のインタフェースによってシステムに通知します。この通知により、SCF (System Control Facility) ファームウェアは、専用のインタフェースを使用して、エラーログを収集し、その解析を行います。これらの動作は、ソフトウェアに影響を与えずにバックグラウンドで行われます。

これらの機構によって、SPARC64 VII を搭載するシステムは、運用を継続しながら故障個所と故障種別を迅速かつ正確に特定し、予防保守に有用な情報を得て、サービス性の向上を図ることができます。

6 むすび

SPARC64 VII は、現行の SPARC64 VI の高性能・高信頼性を保ちつつ、クアドコアおよび SMT の採用により、スループット性能の大幅強化を実現しました。また 4 コアで L2 キャッシュを共用する構成とし、ハードウェアによるコア間高速同期機構を実装しました。

SPARC64 VII はシングルスレッド性能とスループット性能を両立させており、ビジネス・科学技術計算いずれの分野においてもその性能をフルに発揮することが可能です。

7 参考文献

富士通株式会社: UNIX サーバ用プロセッサ「SPARC64 V」, Aug 2004