社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2003－ＳＬＤＭ－111（23）
2003／10／24

高信頼設計 SPARC64 V マイクロプロセサ

安藤 壽茂† 吉田 裕司† 井上 愛一郎† 杉山 五美† 淺川 岳夫† 森田 国樹† 牟田 俊之†
元車田 強† 岡田 誠之† 山下 英男† 薩川 禎彦† 紺本 明彦† 山下 良一† 杉山 広行†

† 富士通株式会社 〒211-8588 川崎市中原区上小田中 4-1-1
E-mail: † hando@jp.fujitsu.com

あらまし ハイエンドUnixサーバ用プロセサとして，130nm CMOS PD SOIプロセスを用いて第五世代のSPARC64プロセサを開発した。クロック周波数は1.35GHzで動作し，消費電力は約50Wである。基幹業務を遂行するサーバには高い信頼度が要求され，特に多数のプロセサを使用する大規模サーバでは個々のプロセサの信頼度を高めることが必須である。一方，微細化加工の進展に伴ない，回路の動作電圧やスイッチングのエネルギーが減少することにより回路のノイズマージンは低下し，高信頼の安定動作とは相反する環境となっている。このような状況に対処するため，Unix サーバプロセサとしては世界で初めて演算器を含むデータパスにもチェック回路を設け，エラー検出時には再実行により回復を図り高い信頼度を確保する設計を行った。

キーワード コンピュータアーキテクチャ，マイクロプロセサ，エラー検出，高信頼性，SPARC

SPARC64 V Microprocessor designed to achieve High Reliability

Hisashige ANDO† Yuuji YOSHIDA† Aiichiro INOUE† Itsumi SUGIYAMA†
Takeo ASAKAWA† Kuniki MORITA† Toshiyuki MUTA† Tsuyoshi MOTOKURUMADA†
Seishi OKADA† Hideo YAMASHITA† Yoshihiko SATSUKAWA†
Akihiko KONMOTO† Ryouichi YAMASHITA† Hiroyuki SUGIYAMA†

† Fujitsu Ltd. 4-1-1 Kamikodanaka, Nakaharaku, Kawasaki 211-8588 Japan
E-mail: † hando@jp.Fujitsu.com

Abstract Fifth generation SPARC64 processor is developed for high-end unix servers. A semiconductor process used is 130nm CMOS PD SOI process. The processor runs at 1.35GHz clock and dissipates about 50W. A high reliability is required for the high-end server used for the enterprise and/or social infrastructures. Since a large server incorporates many processor chips, a reliability of an individual processor must be very high. On the other hand, an advancing semiconductor scaling is adversely affecting the reliable operation of the circuits. We have added error check circuitry to the logic circuits of the processor in addition to the on chip memory arrays to achieve high reliability under this circumstance. This is the world first use of the error detection and recovery technique on the logic circuits for the unix server processor.

Keyword computer architecture, microprocessors, error detection, reliability, SPARC

1. はじめに

ハイエンド Unix サーバへの適用を主目的として当社のSPARC プロセサの第五世代となる SPARC64 V プロセサを開発した。基幹業務を遂行するハイエンドサーバは一種の社会基盤であり，一日 24 時間，年間 365 日連続稼動する高い信頼度を要求される。一方，ハイエンドサーバは最大128 個のプロセサチップが使用されるので，個々のプロセサチップの平均故障間隔（MTBF）が10年であっても毎月プロセサ故障が発生することになり，高い信頼度要求に応えられない。このため，個々のプロセサの信頼度を高めることが必須である。

これに対して，半導体の微細加工の進展は逆方向に働いている。微細化により電源電圧，回路の動作電圧は減少しノイズマージンが低下するが，電源電流は増加や di/dt は

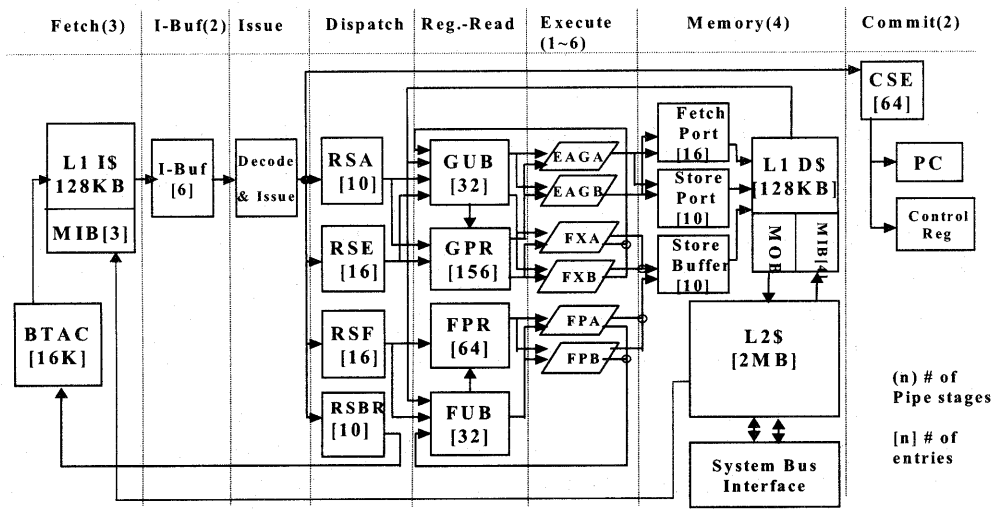

図 1 SPARC64 V プロセサ ブロックダイアグラム

急激に増加し，電源電圧降下やインダクタンスに起因する電源ノイズは増加する。

オンチップのメモリに対してパリティーチェックによるエラー検出や ECC コードを用いて 1bit エラーを訂正することは広く行われている。しかし，180nm プロセスで製造したプロセサの経験ではロジック部のエラーと推定される誤動作が見られており，微細化の進展に伴ない，メモリ部だけでなくロジック部のエラーの検出訂正が重要になると考えた。

2. マイクロアーキテクチャ

本プロセサは SPARC V9 仕様と富士通-Sun で制定した共通命令仕様(JPS1)[1][2]に準拠する64bit SPARC アーキテクチャのプロセサである。基本的なマイクロアーキテクチャは 4 命令同時発行のスーパスカラ方式であり，命令発行は In-Order，実行は Out-of-Order，コミットは In-Order の所謂アウトオブオーダマシンである。本プロセサ[3][4][5]のブロックダイアグラムを図 1 に示す。

命令は 128KB の一次命令キャッシュ(L1I$)からフェッチされ，6 エントリの 32 バイトバッファからなる命令バッファ(I-Buf)に格納される。命令フェッチ部は分岐先アドレスキャッシュ(Branch Target Address Cache)を含む分岐予測機構により次の命令フェッチアドレスを予測してフェッチを続行し I-Buf を満たすように動作する。

命令デコード・発行部は I-Buf から命令を取り出しデコードし，命令の種別ごとに対応する実行パイプラインのリザベーションステーションに格納する。整数命令は RSE リザベーションステーション，浮動小数点演算命令は RSF リザベーションステーション，ロードストア命令は RSA リザベーションステーション，分岐命令は RSBR リザベーションステーションを用いる。

オペランドが揃い，実行パイプラインが空くタイミングになるとリザベーションステーションから命令が取り出され実行が開始される。それぞれの実行パイプラインは独立に動作し，実行パイプライン間での命令の実行は順不同である。また，演算の依存関係によりオペランドが揃わない命令は後回しにし，実行可能な命令を優先して処理するアウトオブオーダ実行を行うことにより効率を高めている。

整数演算とアドレス計算の演算結果は GUB と呼ぶリオーダバッファに格納され，浮動小数点演算の結果は FUB と呼ぶリオーダバッファに格納する。これらのリオーダバッファからアーキテクチャレジスタである GPR，FPR への書き戻しは，分岐方向の予測誤りやハードウェアエラーがないことを確認し命令を完了する時点で行われる。

ロード，ストア命令は EAGA/B 演算器でアドレス計算を行い，その結果をフェッチポート，ストアポートと呼ぶバッファに格納する。また，ストアデータをストアバッファに格納する。ロードストアユニットは，必要な情報(アドレスとストアの場合はストアデータ)の揃った命令から，メモリオーダの制約を満たす範囲でアウトオブオーダで1次データキャッシュ(L1D$)アクセスを実行する。

L1D$は 128KB で2way セットアソシアティブ構成になっている。L1D$は 4 個のミスバッファ(Move In Buffer と表記)を持ち，キャッシュミスの場合にはこのバッファを割り当てて 2 次キャッシュにアクセスを起動する。そして，L1D$自体は次のアクセス要求の処理に移るので，最大 3 個のミスを処理中の状態でも新規のアクセスを受け付けるノンブロッキング動作が可能な構成となっている。L1I$側にも 3 個のミスバッファを持ち，同様にノンブロッキング動作を行う。これらの 1 次キャッシュをミスしたアクセスは二次キャッシュに送られる。また，ミスバッファがフルの状態でも，新規のミスをプリフェッチ要求として 2 次キャッシュに送りアクセスレーテンシを短縮している。

二次キャッシュは命令，データ共通のキャッシュであり，容量は2MB で 4way セットアソシアティブ構成である。2 次キャッシュをミスしたアクセスや I/O アクセス等はシステムバスインタフェースを経由してチップ外部へ送られる。

Commit Stack Entry(CSE)は命令の発行から完了までの状態を管理するリング構造の 64 エントリのバッファである。ハードウェアエラーが検出された場合には，CSE の状態に基づき，仕掛り中の命令の中間結果を全て破棄し，実行を完了した命令の次の命令からやり直すことによりハードウェアによるエラー回復を実現する。

3. プロセサチップの実装

3.1. 概要

プロセサチップの諸元を表 1 に示す。使用した半導体プロセスは 130nm PD SOI CMOS プロセスであり，8 層の銅配線と FSG 層間絶縁膜を用いている。1-4 層の配線はマクロの内部配線や近距離配線に用いており，配線ピッチは 0.45um である。5-6 層配線はこの 2 倍の 0.9um ピッチ，7-8 層配線はその 2 倍の 1.8um ピッチとして CAD での取り扱いを容易にしている。チップサイズは 18.14mmx15.99mm であり，191M トランジスタを集積している。チップは 5858 個の半田バンプでほぼ全域を覆われている。そのうち，269 個のバンプが信号入出力に使用され，残りはグランド，プロセサコア電源(Vdd)と I/O 用の 1.8V 電源(Vddh)に使用されている。

図 2 にチップの顕微鏡写真を示す。2MB の L2$のデータアレイとそのコントロール回路がチップの右側の約 40%の領域を占めている。システムバスの I/O バッファとコントロール回路が L2$を除いた部分のチップの下辺を占めている。そして，このコントロール回路の上に L2$のタグアレイが置かれている。これら以外のチップの左上の 1/3 程度の部分がプロセサコアである。

図 1 のブロックダイアグラムに沿って述べると，プロセサコアの左下の部分に L1I$と BTAC が置かれている。物理的にはどちらも 128KB で SRAM で作られている。そして I-Buf からリザベーションステーションの部分は I-unit 領域に含まれている。チップの左上の FP 領域に FPR, FUB と FPA/B の浮動小数点演算器が含まれ，FX 領域には GPR, GUB と FXA/B の整数演算器が含まれている。EAGA/B, Fetch Port, Store Port, Store Buffer は Load Store 領域に含まれている。また，このブロック図には表記されていないメモリ管理ユニット(MMU)もこの部分に含まれている。そしてプロセサコアの右上に L1D$とそのコントロール回路が置かれている。また，分岐ユニット，PC，CSE などプロセサ全体を管理する機構は I-unit 領域に含まれている。

3.2. 回路設計

我々にとって PD SOI プロセスを使ったプロセサ開発は初めてであり，リスクを軽減するため，ダイナミック回路の使用は極力させる方針を取った。論理回路は全てスタティック

表 1 チップ実装諸元

半導体テクノロジ	130nm CMOS SOI, 8層銅配線 電源電圧 1.32V
チップ	18.14mm x 15.99mm バンプ数5858 (269 I/O)
トランジスタ数	191M (うち論理回路 19M)
クロック	1.35GHz
バスインタフェース	270MHz SDR/DDR, 16B幅, 双方向
消費電力	～50W @1.35GHz

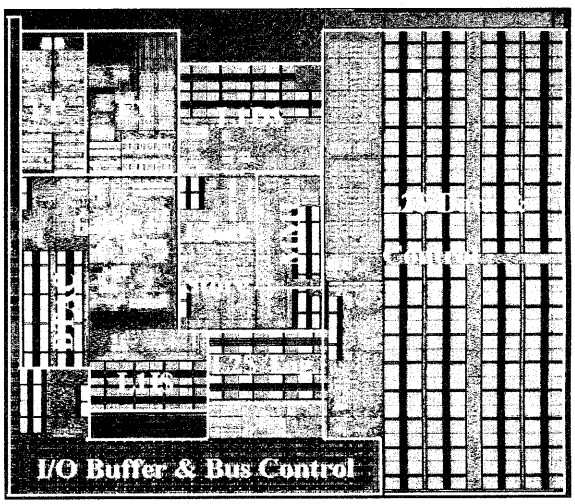

図 2 チップ顕微鏡写真

CMOS 回路を使用している。ダイナミック回路を用いているのは SRAM マクロ内部で，ビット線の部分とセンスアンプにプリチャージを使う回路を用いている。

マクロ設計もフルカスタム設計の部分を減らし，スタンダードセルや小さなフルカスタムマクロを部品として人手で密に配置配線を行ってシュードマクロ(Pseudo Macro)を構成するという手法を用いた。構成要素のスタンダードセルや小規模カスタムマクロの回路特性を把握しておけば，大規模なマクロのノイズマージンの確保が出来，タイミングも構成要素の遅延特性とシュードマクロを構成する配線容量から計算することができる。従って，大規模なフルカスタムマクロを開発する場合に較べ，回路設計の工数を大幅に削減することができた。

スタンダードセルは 0.25V の通常 Vth トランジスタによるものと，0.17V の低 Vth トランジスタによるものの合計で 579 種を開発した。低 Vth トランジスタを用いるセルはリーク電流が大きいのでタイミングクリティカルな部分に使用を制限しており，結果として全スタンダードセル数の 5.75%が低 Vth トラン

ジスタセルとなった。

また，これに加えて67種のカスタムマクロ，22種のセミカスタムマクロと4種のSRAMを開発した。これは442種のカスタムマクロと50種のSRAMを開発したと報告[6]されているフルカスタム設計のマイクロプロセサ開発に比較すると非常に少ない開発量である。また，カスタムマクロは最大のものでも12Kトランジスタであり，一般的なプロセサでのカスタムマクロに較べて規模が小さいものである。

3.3. 省電力設計

本プロセサの開発に当たっては，(1)主にスタティック回路を使用すること，(2)クロック系の電力削減，(3)論理ブロックごとのクロックゲーティングにより消費電力の削減を図った。

スタティック回路はダイナミック回路に較べて出力信号のトグル確率が低く，負荷となる配線やゲート容量を充放電する電力を減少できる。また，クロックを必要としないのでクロック供給系の電力も削減できる。特に，本プロセサでは各ラッチへのクロックのスキューをタイミングチューニングの必要に応じて選択できる設計法を用いており，他のプロセサのようにスキューを減らすためにチップ全域を覆う大きなクロックグリッドを用いる設計は行っていない。スタティック回路の使用と相俟ってクロック系の負荷が減少し，クロック電力を節減している。

クロックゲーティングは電力削減に広く適用されている手法であり，次のクロックサイクルに次段のパイプラインを動作させる必要が無いことが判明した場合には，次段の論理ブロックへのクロック供給を止めている。これによりクロック電力と論理ブロック内部のスイッチングによる電力を節減している。

これらの手法の適用により，当初のサンプルでは1.2V電源，1.3GHzクロック動作で34.7Wの消費電力を実測した。しかし，製品への適用では高い周波数で動作するチップの比率を増加させるため電源電圧を10%高めて1.32V動作とし，クロック周波数を1.35GHzとした。この状態での消費電力は約50Wである。

4. 高信頼度設計

4.1. 背景

当社の180nm世代の半導体プロセスで製造したプロセサの経験では，ECCにより検出訂正されるエラーと比較して1/3～1/10程度の比率でSRAMエラーではないエラーが発生している。これらのエラーの大部分は再現しない故障であり，なんらかの原因でプロセサの中のSRAM以外の回路が間欠的にエラーした可能性が高い。

半導体の微細化の進展により，電源電圧が低下し回路の信号振幅，ノイズマージンは低下している。しかし，チップ全体の消費電力は増加しており，電源電圧の低下と相俟って電源電流は大幅に増加している。電源電流の増加は抵抗性の電源電圧降下を増加させ，また，回路の高速化と相俟って電源電流のdi/dtは急速に増加し，設計努力にも係わらずパッケージやチップ内インダクタンスに起因するノイズが増加傾向にある。

加えて，微細化による素子サイズの縮小は寄生容量を減少させ，アルファ線や中性子などの衝突に起因する注入電荷による誤動作の耐性を減少させている。SOIプロセスによりバルクからの電荷ファネリングが無くなり耐性の向上が期待されるが，一方，チャネル領域に衝突した場合はバイポーラ効果で影響が増大するので，結果としてバルクと比較して改善は1桁程度という報告[7]もあり，SOIも放射線起因のエラー対策の決め手にはなっていない。

このような状況の中で高い信頼度を確保するには，プロセサのエラー検出，回復能力を強化することが必要であると考えた。当社の従来のプロセサではオンチップのSRAMアレイに対しては，ECCやパリティーチェックによりエラー検出訂正を行ってきたが，本プロセサではこれを一歩進めて，チップ内のレジスタやレジスタファイル，データパス，演算器にもパリティーチェックを付加し，エラー検出，回復を行ってる。

表2 エラー検出と回復手段のまとめ

	エラー検出	回復
L1I$ Data	パリティーチェック	内容破棄と再アクセス
TLB	パリティーチェック	
BTAC	パリティーチェック	回復不要，エラー通知だけ
L1I$ & L1D$ Tag	パリティーチェック +2重化	エラーなしの出力を利用して回復処理
L1D$ Data	SECDEDコード	ECCによるエラー訂正
L2$ Data & Tag	SECDEDコード	
Registers	パリティーチェック	命令再実行
ALU, Shifter, VIS	パリティー計算と比較	
Mult/Div	レジジューチェック +パリティー計算と比較	

4.2. エラー検出機構

表2にプロセサ各部のエラー検出機構とエラー回復手段の一覧を示す。メモリのエラー検出，回復手段は，それぞれのアレイの使われ方のより最適な手法を選択した。L1I$データアレイ，TLBとBTACはパリティーチェックでエラーを検出する方法を用いた。L1I$はオリジナルがメモリに存在するので，エラーを検出した場合はL1I$の内容を破棄して再度フェッチすることにより回復が可能である。TLBも同様にメモリにあるページ変換テーブルから正しい内容を再構成できる。従って，これらの2種のアレイはパリティーチェックだけでエラー回復が可能である。

BTACは性能向上のヒント情報であり，エラーがあってもプログラム実行の正しさには影響しない。パリティーチェックはハードウェア故障の検出，ロギングに用いられる。

L1I$とL1D$のタグアレイはオリジナル情報を含むので，エラー回復機能を必要とする。これらのアレイはL2$側に同一情報を保持するアレイがあり，一方でエラーを検出した場

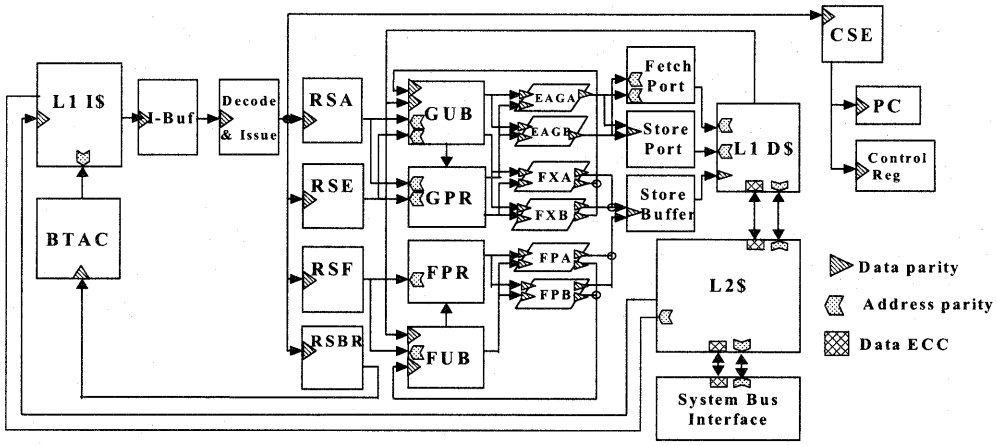

図 4 パリティーチェッカ配置図

合，他方のアレイと通信し，エラーの無いアレイの情報を用いてエラー回復を行う。L1D$データと L2$のデータ，タグアレイは SECDED(Single Error Correction Double Error Detection)コードを適用し，ECC によりエラー訂正を行っている。

データレジスタやレジスタファイルにはパリティービットを付け，データと一緒にパリティービットを伝達し，受端でパリティーチェックを行っている。図 4 に示すようにプロセサ内の主要なデータ伝送経路の受端にはパリティーチェッカーが配置されている。

ALU などの演算器では演算と並行して，入力データから演算結果のパリティーを計算する回路を設け，この出力と演算結果から再計算したパリティービットとの一致確認を行っている。乗算器では

Mod3(A*B) = Mod3(Mod3(A)*Mod3(B))

の関係を利用し，乗算結果から計算した Mod3 と入力から計算した Mod3 の一致を確認している。

図 5 に各ユニットに含まれるラッチ数とパリティーの付加状況を示す。制御系のラッチはパリティーチェックを付けることが困難であり，これらのラッチはパリティーチェックなしである。I-unit は制御回路の比率が高いので付加率が低いが，その他のユニットでは大部分にパリティーチェックが付加されている。“その他”と記された部分は実行状態のログを取る部分や I/O の試験を行うバウンダリスキャンが中心であり，通常の命令実行には影響がないのでパリティーチェックを省いている。結果として全ラッチの 77%がパリティーチェックで保護されており，“その他”の部分を除いて考えるとこの比率は 83%となる。

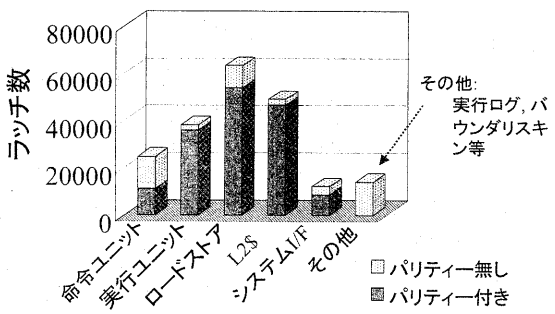

図 5 ユニット別ラッチ数とパリティー付加状況

4.3. 命令再実行

これらのデータパスや演算器でエラーが検出された場合には命令再実行によるエラー回復が行われる。その手順を図 6 に示す。エラーが発生しパリティーチェッカからの信号がエラー検出部に到達した時点で命令実行が停止される。この時点で実行が完了しコミットした命令の結果は既に GPR や FPR などのアーキテクチャレジスタに書き戻されている。

メモリアクセスなどの時間のかかる命令が実行中であるケースに備え，十分な時間，命令発行を停止した後，コミットしていない命令の中間状態を保存しているリザベーションステーションやリオーダバッファなどの内容を破棄し，プロセサをコミットしていない命令の実行開始前の状態に戻す。ハードウェアは，どの命令の実行過程でエラーが発生したかを正確に検出するのでは無く，単にエラーが発生した命令がエラー検出以前にコミットしてしまうことが無いように作られている。従って，図 6 のようにエラーが命令 n で発生しても n-1 の命令も破棄してしまうことが起こり得る。しかし，命令 n-1 を再実行するコストは僅かであり，発生頻度も少ないので無視できる。

命令実行の再開にあたり，最初は，コミットしていない最初の命令を 1 命令だけ取り出して実行する。そして，この命令がエラー無く完了したら通常のスーパスカラ実行に戻る。

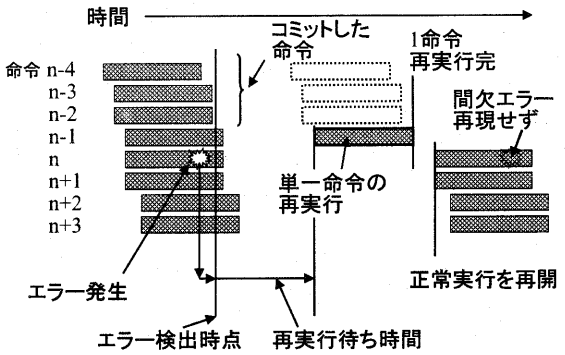

図 6 命令再実行手順

エラーの原因がα線や中性子などの一過性のものである場合は，再実行時にまた発生することは無く，回復が可能である。

また，最初の命令の実行開始時点ではプロセサ内のバッファやキューなどのリソースは全て開放されており，かつ，1命令だけの実行であるのでリソース競合が生じない。また，チップ内の回路動作も少ないので電気的なノイズも少なく，エラー原因がチップ内のノイズやリソース競合時に顕在化するバグであったとしてもこの命令の再実行が成功する確率は高い。また，図 6 の例のようにエラーは命令 n の実行中に発生した場合でも，命令 n の再実行時点ではリソース競合やノイズ発生は命令 n-1 の時点よりは大きいものの，当初のエラー発生時に較べると少なく再実行が成功する確率は高い。更に，命令 n の再実行時にもエラーが発生したとすると，このときには命令 n-1 はコミットしているので，次の再実行は命令 n を最初の命令として単一命令の実行モードとなり，成功の可能性を高める。

一方，エラーの原因が固定故障である場合には再実行を行っても回復は出来ない。この場合は同一命令の再実行の繰り返しを監視するカウンタが 4 回に達した時点でハードエラーとして監視機構に通知する。

約 20%のパリティーチェックが付加されていないラッチやそれに付随する回路のエラーは検出できない。また，演算器のパリティーチェックも 1 箇所のエラーが複数ビットの出力に影響を与えるケースでは検出できない場合がある。更に，パリティーチェック付きのレジスタやレジスタファイルの内容ビットがエラーにより書き換わってしまった場合は，再実行を行っても回復は出来ない。従ってメモリ以外の論理回路部分のエラー検出，回復は完全ではないが，プロセサのロジック部のエラーのうち 80～90%程度のケースで検出，回復が可能と見積もっている。

このエラー検出，回復の効果を検証するためには 10^7～10^8 コンポーネント時間の動作が必要であり，32CPU のサーバ 35 台を連続運転して 1 年～10 年を必要とする。従って，効果の実証は今後の課題である。

5. まとめ

ハイエンド unix サーバ用プロセサとしてメモリアレイに加えてレジスタ，データパス，演算器にもパリティーチェック回路を付加し，エラー検出を行い，命令の再実行によりエラー回復を行うプロセサを開発した。論理回路部分のエラー検出，回復を行う設計は unix 用プロセサとしては世界初である。プロセサチップとしては 130nm PD SOI CMOS プロセスを使用し，1.35GHz クロック，約 50W の消費電力を実現した。

謝辞: 本プロセサの開発は論理，回路，CAD，テストを含む設計チームの努力の成果であり，著者として名前を上げることが出来なかった多数のチームメンバの貢献なしには為し得なかったことをここに記し，感謝の意を表します。

文 献

[1] "SPARC Joint Programming Specification (JPS1): Commonality", http://primepower.fujitsu.com/en/docs/JPS1-R1.0.4-Common-pub.pdf

[2] "SPARC JPS1 Implementation Supplement: Fujitsu SPARC64 V", http://primepower.fujitsu.com/en/docs/JPS1-R1.0-SPARC64V-pub.pdf

[3] A.Inoue "Fujitsu's New SPARC64 V for Mission-Critical Servers", Microprocessor Forum 2002, Oct. 2002.

[4] H.Ando, Y.Yoshida et. al "A 1.3GHz Fifth Generation SPARC64 Microprocessor", in ISSCC Tech. Dig., Feb. 2003, pp.246-247.

[5] H.Ando, Y.Yoshida et. al "A 1.3GHz Fifth Generation SPARC64 Microprocessor", in DAC Tech. Dig., June. 2003, pp.702-705

[6] C.Anderson et. al "Physical Design of a Fourth-Generation POWER GHz Microprocessor", in ISSCC Tech. Dig., Feb. 2001, pp232-233.

[7] F. Irom et. al, "Single-event upset in commercial silicon-on-insulator PowerPC microprocessors", in Proc. SOI conference, Oct. 2002, pp203-204.