大域ディペンダブル情報基盤プロジェクト
～ディペンダブルアーキテクチャグループ　坂井・五島研究室～

坂井 修一　五島正裕
情報理工学系研究科電子情報学専攻

あらまし　コンピュータシステムにとって処理速度･消費電力とともに重要なことがディペンダビリティである。われわれはマイクロプロセッサを対象として、ＬＳＩ内にディペンダビリティ機能を組み込み、高性能・省電力･ディペンダビリティの３者をバランス良く向上させるプロセッサアーキテクチャの研究を行っている。今年度は、静的値範囲解析による脆弱性検出、レジスタファイルの書き込み時タイミングエラーの検出･回復、アドレスオフセットに着目したデータフロー追跡による注入攻撃の検出、キャッシュにおけるソフトエラー検出機構などの提案を行い、基本設計・基本評価を行った。さらに、これらを統合する超ディペンダブルプロセッサのアーキテクチャを提案した。

1. はじめに

コンピュータシステムにとって処理速度・消費電力とともに重要なことがディペンダビリティ[14]である。ここでは、ディペンダビリティを安全性と信頼性の両方からなる性質とする。本研究ではマイクロプロセッサレベルでのディペンダビリティを対象とし、プロセッサLSI内部にディペンダビリティ機能を組み込んで、性能とディペンダビリティの両面を向上させるプロセッサアーキテクチャの研究を行う。

本年度はその４年目として、静的値範囲解析による脆弱性検出、レジスタファイルの書き込み時タイミングエラーの検出・回復、アドレスオフセットに着目したデータフロー追跡による注入攻撃の検出、キャッシュにおけるソフトエラー検出機構などの提案を行い、基本設計・基本評価を行った。さらに、これらを統合する超ディペンダブルプロセッサのアーキテクチャを提案した。

2. 静的範囲解析による脆弱性検出

プログラムの脆弱性の原因は作成者がプログラムの動作の可能性を完全には把握できていない点にある. そのため, プログラム作成者に動作の可能性を提示することで, 脆弱性を発見することができると考えられる.

ここでは、静的値範囲解析と範囲付き命令の抽出の2つによって脆弱性を検出する手法を提案した. 静的値範囲解析は, プログラム上の値がとり得る範囲を静的な解析によって求める方法である. この静的値範囲解析によって得られた情報をそのままプログラム作成者に提示することも可能であるが, 大量の情報から脆弱性に関連のあるものを発見することは容易ではない. そこで, 範囲を持ったオペランドをとるような命令を抽出することで, 脆弱性と関連の深い命令のみをプログラム作成者に提示する. これにより, 効率の良い脆弱性の検出が可能となった.

テスト･プログラムを用いた評価を行った結果, いくつかの既知の脆弱性を検出することができた. また, 従来の脆弱性検出手法では対象とされていない脆弱性が検出できる可能性を示した. さらに, 提案手法における計算時間･メモリ消費について評価を行い, 比較的大きなプログラムに対しても適用可能であることを示した[15] .

3. レジスタファイルの書き込み時タイミングエラーの検出・回復

これからのプロセッサでは、ソフトエラーのような値の反転だけではなく、タイミングエラーなども含んだ全ての過渡故障に対応できるアーキテクチャが重要である。ここでは、レジスタファイルに着目し、その書き込み時におけるタイミングエラーを検出、回復する手法を提案した。

タイミングエラーを検出するためには、一度書き込んだ内容をもう一度読み出して正しい書き込み値と比較する必要がある。そのため、本手法ではタイミングエラー耐性の高い小容量のバッファに書き込み値を保持してエラーの検出、回復に用いた(図1)。SPEC2000 ベンチマークでの評価の結果、このバッファが8エントリと小さくても、実行オーバヘッドは4.5%以内と小さく抑えられることがわかった。また、提案した比較方法のうち、ポート・スチールのみでも十分有効であることを示した [16]。

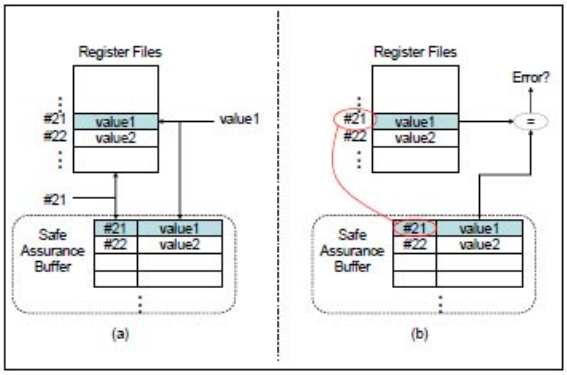

図1. タイミングエラー検出・回復機構

4. アドレスオフセットに着目したデータフロー追跡による注入攻撃の検出

プログラムにはバッファオーバーフローなどの脆弱性が存在し，そのような脆弱性を利用した注入攻撃が行われている．多くの注入攻撃はアドレスを注入することで行っており，本研究では，このような注入攻撃をアドレス注入攻撃と呼ぶ．アドレス注入攻撃を動的に検出する入力データ追跡法は，入力データのアドレスとしての使用を検出し，注入攻撃を回避する．しかし，従来の入力データ追跡法では，アドレスとして使用可能な入力データの識別が曖昧であるため，多くの誤検出，検出漏れが生じる．

本研究で提案する手法では，入力データをアドレスとして使用可能か否かをより厳密に見分ける．入力データをアドレスのオフセットとして用いることのみを許可し，アドレス自体として使用されたら注入攻撃として検出する．そのために，アドレスのデータフロー追跡を行い，記憶領域にあるデータを動的にアドレスと非アドレスに分類する[17].

本手法はRed Hat Linux 6.2 を載せたx86 エミュレータを用いて実装し，検出漏れ，誤検出，速度及びハードウエア容量のオーバヘッドの評価を行った．本研究によってアドレス注入攻撃を高い精度で検出することができ，注入攻撃を動的に検出する手法として最も精度の高いシステムを実現できた.

5. キャッシュにおけるソフトエラー検出機構

連想メモリ(Content Addressable Memory、CAM)は、キャッシュのタグ部として広く使われている。CAM の検索では、タグのデータは陽に読み出されることはないため、ソフトエラーによるフォールスミスの検出は困難であった。本研究では、連想度の高いキャッシュにおけるフォールスミスを検出する手法について提案した。

提案手法では、タグ部を2つの領域に分割し、タグが片方だけマッチした場合にバックアップチェックを行う(図2)。さらに、バックアップチェックを減らすためのタグのコード化についても提案した [5]。

本方式によるハードウエアの変更によって、キャッシュアクセスのレーテンシを増加させることはない。また、フォールスミスによって生じる実行時間の増加は、実質的に無視できる範囲にとどまることがシミュレーションによって示された。

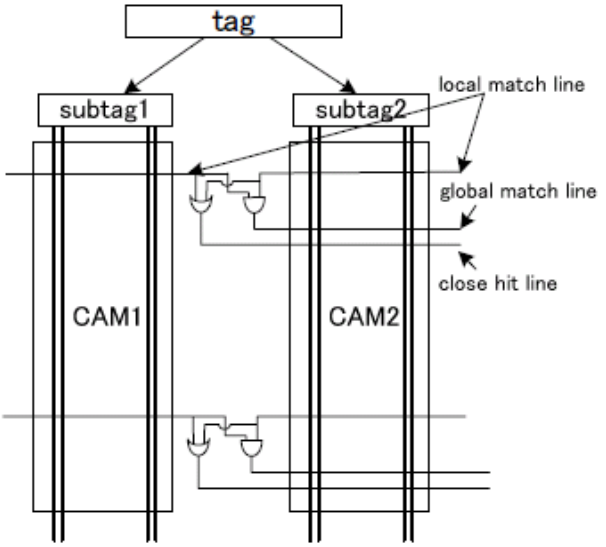

図２．キャッシュにおけるソフトエラー検出機構

6. 統合アーキテクチャ

本研究でこれまでに提案・評価したディペンダブルプロセッサの要素技術を統合し、全体を制御する機構を組み入れた統合ディペンダブルアーキテクチャ（超ディペンダブルプロセッサ）(図3)の開発に着手した。

超ディペンダブルプロセッサは、CPU の中でディペンダビリティを保証する部分(ディペンダビリティマネジャ)と実行コアを分離し、後者によって従来のプロセッサと同じ速度でプログラムを実行し、同時に前者によってその実行を監視する。

7．まとめ

本年度は、静的値範囲解析による脆弱性検出、レジスタファイルの書き込み時タイミングエラーの検出・回復、アドレスオフセットに着目したデータフロー追跡による注入攻撃の検出、キャッシュにおけるソフトエラー検出機構などの提案を行い、基本設計・基本評価を行った。さらに、これらを統合する超ディペンダブルプロセッサのアーキテクチャを提案した。

最終年度は、ディペンダビリティ制御機構の研究開発を進め、統合アーキテクチャの具体化・詳細化を行う予定である。

発表文献

[1] 葛 毅, 櫻井 隆雄, ルォン ディン フォン, 阿部 公輝, 坂井 修一:"インターリーブ型剰余乗算回路の評価" 電子情報通信学会論文誌, Vol.J88-A, No.12, pp. 1497 - 1505, Dec, 2005.

[2] Naoya Hatta, Niko Demus Barli, Chitaka Iwama, Luong Dinh Hung, Daisuke Tashiro, Shuichi Sakai and Hidehiko Tanaka: "Bus Serialization for Reducing Power Consumption" 情報処理学会論文誌コンピューティングシステム(ACS 13), Vol.47, No.SIG3, Mar, 2006 (掲載決定).

[3] Luong Dinh Hung and Shuichi Sakai: "Dynamic Estimation of Task Level Parallelism with Operating System Support", 情報処理学会論文誌コンピューティングシステム(ACS 14), Vol.48, No.SIG4, Apr, 2006 (掲載決定).

[4] Hidetsugu Irie, Naoya Hattori, Masanori Takada, Naoya Hatta, Takeshi Toyoshima, Shuichi Sakai: "Distributed Speculative Memory Forwarding" IEEE Symp. on Low-Power and High-Speed Chips(COOL Chips VIII),, pp.473-482, Apr, 2005.

[5] Luong Dinh Hung, Masahiro Goshima and Shuichi Sakai: "Mitigating Soft Errors in Highly Associative Cache with CAM-based Tag", IEEE International Conference on Computer Design (ICCD 2005), California, USA, Vol.2005, pp.342-347, Oct, 2005.

[6] Luong Dinh Hung and Shuichi Sakai: "Dynamic Estimation of Task Level Parallelism with Operating System Support", International Symposium on Parallel Architectures, Algorithms, and Networks (ISPAN 2005), Vol.2005, pp.358-363, Dec, 2005.

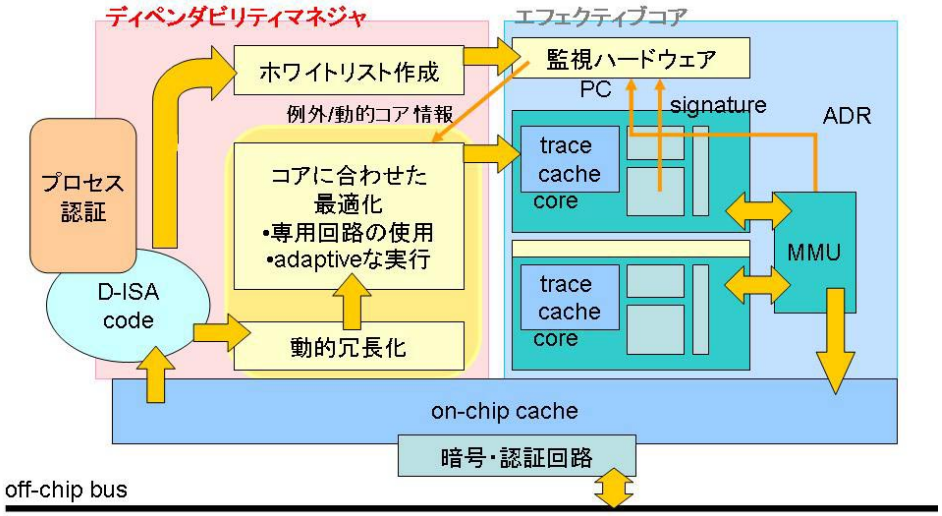

図 3．超 ディペンダブルプロセッサ

[7] Yuya Ueno, Luong D. Hung, Masanori Takada, Daisuke Tashiro and Shuichi Sakai: "Improvement of Signature-based Phase Detection and its Application to Power Reduction in Caches" 電子情報通信学会技術研究報告 RECONF2005-1～14, Vol.105, No.42, pp.25-30, May, 2005.

[8] 清水 一人, 高田 正法, 入江 英嗣, 坂井 修一: "プログラムの振る舞い秘匿のための動的アドレス変換" 情報処理学会研究報告書 計算機アーキテクチャ研究会 2005-ARC-164, Vol.2005, No.80, pp.19-24, Aug, 2005.

[9] 門馬 太平, ルォン ディン フォン, 田代 大輔, 坂井 修一: "スレッド投機実行のためのキャッシュコヒーレンシプロトコルの検証", 情報処理学会研究報告書 計算機アーキテクチャ研究会 2005-ARC-164, Vol.2005, No.80, pp.103-108, Aug, 2005.

[10] Luong D. Hung, Masahiro Goshima and Shuichi Sakai: "Technique to Mitigate Soft Errors in Caches with CAM-based Tags" 電子情報通信学会技術研究報告 DC2005-18, 於 武雄市文化会館, Vol.105, No.227, pp.31-36, Aug, 2005.

[11] 坂井 修一: "新世代プロセッサアーキテクチャの展開", 情報処理学会誌, Vol.46, No.10, pp.1100-1103, Oct, 2005.

[12] 五島 正裕: "スーパスカラ/VLIW プロセッサとスループット指向 MT プロセッサ"情報処理学会誌, Vol.46, No.10, pp.1104-1110, Oct, 2005.

[13] 入江 英嗣:"クラスタ型プロセッサ", 情報処理学会誌, Vol.46, No.10, pp.1111-1117, Oct, 2005.

[14] 坂井修一:"ディペンダブル情報処理基盤"、電子情報通信学会、計算機システム研究会 招待講演、Apr, 2005.

[15] 初田 直也:"静的値範囲解析による脆弱性検出手法"、修士論文, 東京大学大学院情報理工学系研究科, Feb. 2006.

[16] 荻野 健:"レジスタファイルの書き込み時タイミングエラーの検出・回復手法"、卒業論文, 東京大学工学部, Feb. 2006.

[17] 勝沼 聡:"アドレスオフセットに着目したデータフロー追跡による注入攻撃の検出"、卒業論文, 東京大学工学部, 2006.