

Architectural Core Salvaging in a Multi-Core Processor for Hard-Error Tolerance

M. D. Powell, A. Biswas, S. Gupta and S. S. Mukherjee, *Architectural Core Salvaging in a Multi-Core Processor for Hard-Error Tolerance*, In the Proceedings of the 36th International Symposium on Computer Architecture (ISCA), June 2009.

Autor: Hilário Viana Bacellar

R.A: 107077 Disciplina: MO401

Resumo

O artigo “Architectural Core Salvaging in a Multi-Core Processor for Hard-Error Tolerance” tem em seu principal tema minimizar as alterações nos núcleos de um processador que venha a falhar em determinadas operações.

O autor do artigo explica que as CPU’s atuais de alto desempenho possuem vários núcleos para o processamento de instruções e que mesmo assim, em caso de falha de um desses núcleos, o desempenho da CPU fica comprometido. Desta forma, é crucial um estudo detalhado sobre a vulnerabilidade dos núcleos em caso de falhas de um ou mais.

Existem algumas técnicas para a detecção de falhas, elas podem ser dadas em *manufacturing time detection* e *manufacturing test*.

O *manufacturing time* se concentra na identificação de partes defeituosas, entretanto, uma adequação dessa técnica é possível ser expandido para isolar uma parte defeituosa do sistema e assim criar uma redundância de cobertura contra falhas.

O *manufacturing test* se concentra na execução de uma varredura no sistema e mecanismos de depuração baseados em granulosidade fina a fim de encontrar falhas conhecidas na micro-arquitetura.

Para implementar o *core salvaging in a multi-core processor* foram levados em consideração alguns pontos chaves:

- *Minimal Core Changes* – o objetivo desse ponto é minimizar as alterações sofridas no núcleo.
- *Migration and Overhead* – o objetivo desse ponto é a detecção de carga dos núcleos.
- *Operating System Transparency* – o objetivo é manter a técnica totalmente baseada em hardware, assim o ponto chave é deixar o mais transparente possível ao SO.

Após a implementação, também foi descrito os passos para a otimização dos recursos:

- *Migration Policy* - O objetivo dessa política de migração é garantir que os segmentos sejam migrados de um núcleo para o outro sem comprometer o sistema.
- *Migrações Triggering* – O objetivo desse passo é acelerar o processo de migração e decodificação de segmentos.
- *Fall-back to Core Disabling* – O objetivo é manter uma janela de tempo para evitar uma migração excessiva entre os núcleos.
- *Running Less Than the Max Number of Threads* – O objetivo é manter o núcleo que ocorreu a falha em modo de espera o maior tempo possível.

Ao termino do artigo o autor mostra que *salvaging core* tem potencial para cobrir 86% da área da unidade de execução e que mostrou provas de conceito que abrange 46% da área da unidade de execução, sendo assim esses 46% equivale à área dos núcleos vulneráveis.