

## MO401 - Trabalho 1 – Resumo de artigo

### Generating Test Programs to Cover Pipeline Interactions

Dang, T. N., Roychoudhury, A., Mitra, T. and Mishra, P; DAC 2009

Resumido por **Camila Satsu de Amorim Yokoigawa** (RA 107006)

Os sistemas embarcados estão cada vez mais complexos devido ao aumento da demanda por altos requisitos computacionais que representam a inclusão de complexidade acompanhada de características de alta performance, como caches e pipelines, dentro da arquitetura de computadores. Porém, interações não triviais entre essas características são a causa mais comum de erros na implementação de processadores.

Neste contexto, o artigo *Generating Test Programs to Cover Pipeline Interactions*, apresentado em 2009 no 46º evento DAC, *Design Automation Conference*, apresenta um modelo de especificação de arquitetura em alto nível baseada na comunicação estendida de máquinas de estado finito, ou FSM, *Finite State Machine*, de processadores para a geração automática de conjuntos de testes que possibilitam a simulação de todas as possibilidades de interação de pipeline. Este método de geração de testes reduz o tamanho dos conjuntos de testes para validação comparada à outras abordagens de geração de testes, reduzindo drasticamente os esforços de validação, que gastam uma significativa parte do processo de design.

Para gerar automaticamente os conjuntos de teste a arquitetura é modelada por meio de modelos de operação de máquina de estado de dois níveis: o nível operacional e o nível de hardware, que, respectivamente, descreve os movimentos de instruções através dos estágios da pipeline e recursos de hardware, como janelas de instruções e unidades funcionais. É definida a métrica de cobertura que assegura que todos os componentes de pipeline sejam usados, capturando-se todos os cenários comportamentais dentro do pacote de teste por meio de modelos de pipelines dados como uma coleção de EFSM, *Extended Finite State Machine* – máquina de estado finita com variáveis, para a construção de uma FSM global. É usado O Algoritmo 1 para gerar o conjunto de testes *on-the-fly*.

A área de geração de teste para arquiteturas é muito bem estudada, não faltando trabalhos relacionados como a ferramenta Genesys desenvolvida pela IBM, técnicas de cobertura de testes e uso de model checkers para gerar programas de testes para processadores de pipeline.

Entretanto, essas abordagens falham na cobertura das possibilidades de interações de pipeline e na automatização do trabalho de validação, pontos fortes da abordagem trabalhada pelos autores, que garante 100% de cobertura das interações de pipeline com confiabilidade, produzindo conjunto de testes executáveis pequenos, com menos de 1% do tamanho dos conjuntos de testes produzidos pelas demais abordagens.