

Model Checking Probabilístico para comparação de técnicas de mitigação de SEU em FPGAs

Viny Cesar Pereira; Valdivino Alexandre de Santiago Júnior

vinycpereira@gmail.com

O uso de dispositivos lógicos programáveis como Field Programmable Gate Arrays (FPGAs) em aplicações espaciais cresceu fortemente nos últimos anos devido à sua flexibilidade, custo de desenvolvimento e desempenho, mas existe uma exposição excessiva aos raios cósmicos presentes no ambiente e os efeitos da radiação podem causar erros transientes quando partículas carregadas atingem a superfície dos componentes. Tais efeitos são chamados de Single Event Effects (SEEs), que, em sua forma não destrutiva conhecida como Single Event Upset (SEU), pode atingir as células de memória e causar uma inversão no valor lógico armazenado, ou seja, um bit flip. Diversas técnicas de detecção, mitigação e correção de SEU surgiram no passado como forma de evitar falhas, mas a maioria dos testes presentes na literatura foram conduzidos após implementar as técnicas em FPGAs e simular os upsets com ferramentas de injeção de falhas, o que pode ser uma abordagem custosa, já que os resultados só são apresentados ao final das simulações. Por outro lado, modelos estocásticos/probabilísticos podem ser utilizados nos estágios iniciais de um projeto sem a necessidade de implementação, com grande potencial para amortizar o custo do projeto como um todo. Um dos métodos de Verificação Formal que lida com sistemas de comportamento estocástico é Model Checking Probabilístico onde, ao invés da garantia absoluta da corretude dificilmente atingida por Model Checking clássico/funcional, é capaz de garantir, de acordo com uma probabilidade especificada e de uma maneira menos exigente, a corretude do sistema. Esse trabalho apresenta uma comparação relacionada à dependabilidade entre três técnicas conhecidas para mitigação de SEU em FPGAs: Triple Modular Redundancy (TMR), Scrubbing e código de Hamming. Model Checking Probabilístico foi usado com Cadeias de Markov em Tempo Contínuo e Lógica Estocástica Contínua. Resultados dessa comparação mostram que, em termos de disponibilidade, o código de Hamming apresentou o melhor desempenho mantendo o sistema por mais tempo em modo operacional mesmo com a pior taxa de falhas. Na confiabilidade, o que mais afetou o Scrubbing foi o tamanho do intervalo entre as correções enquanto a segurança está mais relacionada com a capacidade do sistema de detectar os componentes degradados. O TMR apresentou os piores resultados pois permite que os

upsets se acumulem e deve ser combinado com alguma técnica de correção, como o código de Hamming ou Scrubbing.

Verificação Formal. Model Checking Probabilístico. Single Event Upset. Field Programmable Gate Array.