



Evento	Salão UFRGS 2018: SIC - XXX SALÃO DE INICIAÇÃO CIENTÍFICA DA UFRGS
Ano	2018
Local	Campus do Vale - UFRGS
Título	Método TMR Aproximativo Por Redução de Precisão de Dados
Autor	JUAN SUZANO DA FONSECA
Orientador	FERNANDA GUSMAO DE LIMA KASTENSMIDT

Método TMR Aproximativo Por Redução de Precisão de Dados

Aluno: Juan Suzano da Fonseca

Orientadora: Profa. Dra. Fernanda Lima Kastensmidt

Instituto de Informática - Universidade Federal do Rio Grande do Sul

Este trabalho investiga a implementação de computação aproximada através da manipulação da precisão de dados com o objetivo de analisar como tal técnica pode proporcionar melhores resultados em métodos tradicionais de tolerância a falhas em FPGAs. Neste trabalho foi utilizado o FPGA contido na APSoc Xilinx Zynq-7000.

A representação de dados do tipo ponto-flutuante é, por si só, uma aproximação do valor original na maioria dos casos e funciona dividindo o dado em um bit de sinal, uma parte exponencial e uma parte decimal (mantissa). Segundo a norma IEEE-754 9 bits são destinados à parte inteira (o bit de sinal e a parte exponencial) e 23 bits são destinados à mantissa. Este trabalho explora os resultados, no que diz respeito a área e precisão, da representação de dados utilizando menos bits do que as representações tradicionais.

Utilizando a ferramenta Vivado High Level Synthesis (Vivado HLS), um software fornecido pela Xilinx cujo objetivo é gerar descrições de hardware sintetizáveis a partir de algoritmos em C ou C++, foram pensadas três representações diferentes com 32, 24 e 16 bits de largura cada. Tais representações foram criadas utilizando a biblioteca *ap_fixed.h* que nos permite criar tipos arbitrários de dados. As três representações seguem a lógica da representação em ponto-flutuante na qual aproximadamente 28% dos bits são destinados à parte inteira e 72% à mantissa.

O estudo de caso realizado foi uma multiplicação com redundância modular tripla (TMR) aplicada. Isto é, o sistema criado constitui-se de três multiplicadores em paralelo e um votador bit-a-bit. Foram realizadas diferentes combinações na largura dos multiplicadores utilizando os tipos de dados arbitrários criados para o experimento. As combinações escolhidas foram: 32-32-32, 32-24-24, 32-24-16, 32-16-16, 24-24-24, 24-16-16 e 16-16-16 bits. Foram realizados diversos testes utilizando entradas que exercitassem a maior quantidade de bits possível. Por isso, foram decididos arbitrariamente diferentes valores com 6 casas decimais. Cada operação testada nos sistemas implementados em hardware também foi testada via software no processador ARM contido na placa. O resultado alcançado pelo processador utilizando a representação ponto-flutuante foi considerado como resultado ideal.

Comparando o consumo de recursos consumidos por cada sistema podemos confirmar que a alteração na representação de dados diminuiu o consumo de recursos na implementação do sistema, com exceção do caso 32-32-32 onde um multiplicador de 32 bits utilizando a representação estudada nesse trabalho consome mais recursos do que um multiplicador 32 bits ponto flutuante. A precisão alcançada por cada sistema foi satisfatória. O sistema 32-32-32 alcançou 100% de precisão, apesar de ter sido ineficiente no consumo de hardware. Os sistemas 32-24-24 e 24-24-24 alcançaram uma precisão de 99,9997% consumindo consideravelmente menos hardware. O sistema 32-24-16 obteve uma precisão de 99,998%. Os sistemas 32-16-16, 24-16-16 e 16-16-16 obtiveram o pior resultado com 99,98% de precisão.