

# Método TMR Aproximativo Por Redução de Tamanho de Dados

Juan Suzano da Fonseca

Orientadores: Fernanda Kastensmidt, Gennaro Rodrigues, Adria Oliveira

## Resumo

Este trabalho investiga o uso de computação aproximada em funções matemáticas visando a redução de área no projeto de circuitos. A aproximação é feita através da redução de bits na representação de dados. Este trabalho aplica essa proposta em uma função de multiplicação de matrizes protegida por triplicação (TMR). Os resultados mostram que a aproximação pode, em muitos casos, reduzir significativamente a área e manter uma excelente precisão no resultado. Para tolerância a falhas, uma redução de área pode diminuir significativamente a suscetibilidade do circuito.

## 1. Introdução

- Segundo o padrão IEEE-754 a representação de dados float contém 32 bits que são divididos em: 1 bit de sinal, 8 bits de expoente e 23 bits de mantissa.
- Quanto maior o expoente mais números inteiros podem ser representados.
- Quanto maior a mantissa maior a precisão alcançada em números decimais.
- O que queremos é: Reduzir a área consumida por uma aplicação, tornando o circuito mais barato e tolerante a falhas, através da redução de bits na apresentação de dados.

## 2. Investigação

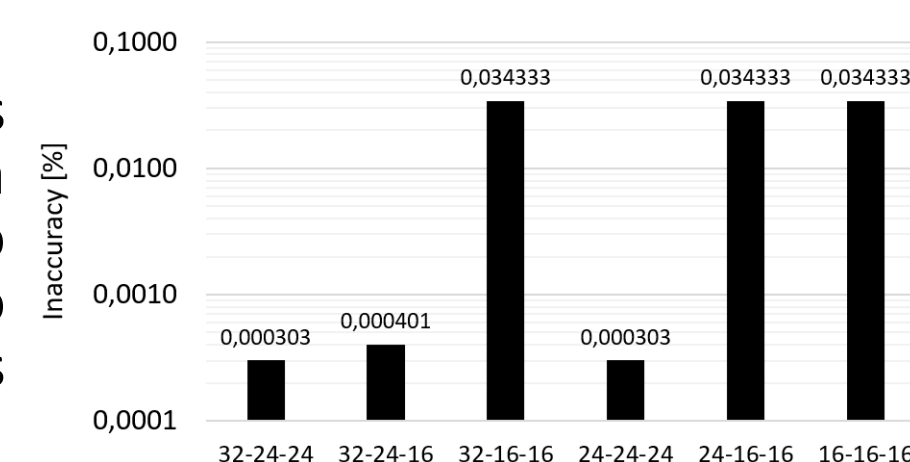
- Estudo de caso: multiplicador de matrizes com dados em ponto flutuante representados com de 32, 24 e 16 bits
- Implementação de TMR (Triple Modular Redundancy) na multiplicação de matrizes usando diversas configurações de largura de dados.
- 7 configurações: 32-32-32, 32-24-16, 32-16-16, 24-24-24, 24-16-16 e 16-16-16 onde cada valor representa a largura de dados de um dos módulos de redundância do TMR, ou seja, a configuração 32-16-16 conta com um módulo que opera com 32 bits e outros dois que operam com 16 bits.
- Para a multiplicação de matrizes 2x2 cada módulo contém 8 multiplicadores com a largura de dados especificadas pelo módulo.

## 3. Implementação

- A multiplicação de matrizes e o votador foram descritos em C e o VHDL gerado automaticamente pela ferramenta Vivado HLS.
- As representações de dados foram criadas usando a biblioteca *ap\_fixed.h*.
- Usando a biblioteca, foi definido o tamanho da representação, a quantidade de bits destinada a parte inteira (sinal e expoente) e a quantidade de bits destinada a mantissa.
- O TMR possui um votador bit-a-bit. Portanto, é necessário que todos os valores votados tenham o mesmo tamanho. Logo, quando necessário, foram implementados conversores.
- Os conversores atuam convertendo, se necessário, os menores dados para o tamanho do maior.

## 4. Análise da precisão

- Cada multiplicador apresenta um erro (%) previsto dependendo da representação usado em sua implementação.
- Naturalmente, em um ambiente não-suscetível a falhas dois multiplicadores de mesma largura de dados apresentam o mesmo resultado.
- Em configurações com dois multiplicadores de mesma largura de dados e um diferente, o votador bit-a-bit seleciona o resultado alcançado pelos multiplicadores iguais.



## 5. Análise da Área

- Cada multiplicador consome uma quantidade de DSPs dependendo da sua Largura de dados.
- Multiplicadores com largura de dados de 32, 24 e 16 bits consomem, respectivamente 4, 2 e 1 DSPs.
- Para a multiplicação de matrizes 2x2 o consumo de DSPs é previsível pois a quantidade de multiplicadores envolvidos na operação é fixa e a quantidade de DSPs necessários para implementar cada multiplicador é conhecida.

8 multiplicadores por módulo, cada multiplicador consome 1 DSP e são implementados 3 módulos iguais.

Design TMR	Área			Tclk: 10ns
	DSP48E	FF	LUT	
32-32-32	96	1985	888	9
32-24-24	64	1859	761	9
32-24-16	56	1763	595	9
32-16-16	48	1759	945	9
24-24-24	48	1815	1609	8
24-16-16	32	1841	1305	6
16-16-16	24	1032	689	6

Votar dados em 24 bits se mostrou mais custoso do que votar dados em 32 ou 16 bits. Isso ocorre pois o algoritmo em C que descreve o votador de 24 bits precisou ser mais complexo do que o algoritmo que descreve os votadores de 32 e 16 bits.

## 4. Conclusão

A técnica investigada pode reduzir consideravelmente o consumo de área do circuito enquanto mantém uma precisão muito boa, permanecendo acima de 99.9% em todos os casos avaliados. Trabalhos futuros envolvem a investigação do impacto de diferentes tipos de votadores na precisão alcançada assim como testes de injeção de falhas para atestar a confiabilidade do sistema.