

## AxRSU-2<sup>m</sup>: CODIFICADORES APROXIMADOS DE $m$ BITS PARA UNIDADES DE OPERAÇÕES ELEVADAS AO QUADRADO NA BASE 2<sup>m</sup>

NICOLAS OLIVEIRA<sup>1</sup>; MORGANA ROSA<sup>1</sup>; EDUARDO COSTA<sup>2</sup>; RAFAEL SOARES<sup>1</sup>.

<sup>1</sup>Universidade Federal de Pelotas – [ncsdoliveira@inf.ufpel.edu.br](mailto:ncsdoliveira@inf.ufpel.edu.br); [morgana.azevedo@sou.ucpel.edu.br](mailto:morgana.azevedo@sou.ucpel.edu.br); [rafael.soares@inf.ufpel.edu.br](mailto:rafael.soares@inf.ufpel.edu.br)

<sup>2</sup>Universidade Católica de Pelotas – [eduardo.costa@ucpel.edu.br](mailto:eduardo.costa@ucpel.edu.br)

### 1. INTRODUÇÃO

A computação aproximada (AxC) tem se destacado como uma abordagem inovadora para melhorar a eficiência de sistemas computacionais, aproveitando a resiliência intrínseca ao erro em várias aplicações modernas. Em muitos cenários, como processamento de sinais, imagens (GILLANI, 2018), vídeos (GUIDOTTI, 2020), eletrônicos de consumo (SEIDEL, 2021) e aprendizado de máquina (TASOULAS, 2020), não é necessário que todos os cálculos sejam exatos. Em vez disso, existe uma margem de tolerância a erros, o que permite a utilização de técnicas de aproximação que podem reduzir o consumo de energia, a área de hardware e o tempo de processamento. Nesse contexto, o estudo de operações aritméticas fundamentais (ROSA, 2023), como a operação elevada ao quadrado, torna-se um tema de grande relevância (TSAI, 2021). A operação elevada ao quadrado é amplamente utilizada em algoritmos de processamento de sinais, compressão de vídeo e em outras aplicações computacionalmente intensivas. Um exemplo clássico é a métrica *sum of squared differences* (SSD), frequentemente utilizada em sistemas de codificação de vídeo, como o padrão HEVC, para medir a diferença entre quadros consecutivos. No entanto, o cálculo exato de operações ao quadrado pode ser oneroso em termos de energia e área de hardware, especialmente em sistemas embarcados, onde a eficiência energética é uma prioridade. Motivado por essas limitações, o presente estudo foca na proposta de uma unidade de operação ao quadrado baseada na técnica na base 2<sup>m</sup>, denominada AxRSU-2<sup>m</sup> (do inglês, *Approximate Radix-2<sup>m</sup> Squarer Unit*). Esta unidade emprega codificadores aproximados de  $m$  bits para realizar operações elevadas ao quadrado de forma eficiente, reduzindo a complexidade dos cálculos e otimizando o uso de recursos de hardware. A unidade AxRSU-2<sup>m</sup> é uma solução escalável, projetada para operar com diferentes valores de  $m$  (como 2, 3 e 4), correspondendo a operações de Radix-4, Radix-8 e Radix-16, respectivamente. Essa abordagem explora a redução de produtos parciais e a simplificação da árvore de somadores, resultando em ganhos expressivos em termos de economia de energia e redução de área de circuitos. Ao utilizar a AxRSU-2<sup>m</sup>, este trabalho visa demonstrar como a computação aproximada pode ser aplicada de forma eficaz para melhorar o desempenho de sistemas dedicados a operações elevadas ao quadrado, sem comprometer significativamente a precisão necessária em aplicações tolerantes a erros.

### 1. METODOLOGIA

A metodologia segue um processo de síntese lógica e co-simulação das unidades de operações. O desenvolvimento dessa metodologia se inicia com a definição da unidade de operação ao quadrado aproximada baseada na técnica Radix-2<sup>m</sup>. Essa unidade, chamada AxRSU-2<sup>m</sup>, explora diferentes valores de  $m$  para

implementar codificadores que operam sobre grupos de  $m$  bits de dados simultaneamente. Foram projetados dois tipos de codificadores: o codificador tipo 1 (AxRSU-2<sup>m</sup>-T1), responsável pelo cálculo da operação ao quadrado, e o codificador tipo 2 (AxRSU-2<sup>m</sup>-T2), que trata a multiplicação de dois valores diferentes. A AxRSU-2<sup>m</sup> foi projetada para ser escalável, permitindo a aproximação dos codificadores e, assim, otimizar a eficiência em termos de área de hardware e consumo de energia. Desenvolveu-se um algoritmo para possibilitar a geração automática de unidades AxRSU-2<sup>m</sup> de diferentes tamanhos e níveis de aproximação. Esse algoritmo cria unidades ao quadrado para qualquer valor de  $m$  e para um número variável de bits de entrada ( $W$ ). Ele ajusta o grau de aproximação por meio de parâmetros, denominados  $k_1$  e  $k_2$ , que determinam quantos codificadores exatos serão substituídos por codificadores aproximados. Ele também aplica deslocamentos à esquerda para evitar operações redundantes, otimizando o cálculo. A avaliação do desempenho das unidades AxRSU-2<sup>m</sup> foi realizada por co-simulação utilizando MATLAB e ModelSim. O MATLAB foi utilizado para gerar modelos de referência exatos, comparando-os com os resultados aproximados obtidos no Modelsim. As simulações usaram vetores de entrada pseudo aleatórios de 16 bits para avaliar a precisão das unidades aproximadas. O hardware foi descrito em VHDL e sintetizado na ferramenta Cadence Genus, utilizando uma biblioteca de células padrão de 65nm e uma tensão de alimentação de 1,25V. Para medir o desempenho e a qualidade dos resultados, foram utilizadas várias métricas de erro, como o erro quadrático médio (MSE), a taxa de erro de bits (BER) e o número de bits eficazes (NoEB). Uma exploração do espaço de projeto foi conduzida para investigar o desempenho das diferentes combinações de codificadores aproximados (AxRSU-4, AxRSU-8 e AxRSU-16) em termos de eficiência energética, área de hardware e qualidade de resultados. Para cada configuração, os parâmetros  $k_1$  e  $k_2$  foram ajustados para obter diferentes níveis de aproximação, e os resultados foram comparados tanto com as soluções exatas quanto com outras propostas da literatura. A análise da frente de Pareto foi utilizada para comparar a relação entre economia de energia, redução de área e qualidade dos resultados, demonstrando que a AxRSU-2<sup>m</sup> oferece um equilíbrio superior em relação a outras soluções de arquiteturas ao quadrado aproximadas.

### **3. Operação Elevada ao Quadrado Aproximada na Base 2<sup>m</sup> (AxRSU-2<sup>m</sup>)**

A técnica Radix-2<sup>m</sup> divide os operandos em blocos de  $m$  bits, aplicando dois tipos de codificadores (T1 e T2) para gerar os produtos parciais de forma aproximada, reduzindo assim a complexidade do circuito e otimizando o consumo de energia e a área de hardware. Esta seção apresenta as configurações da AxRSU-2<sup>m</sup> para três valores de  $m$ : 2, 3 e 4, correspondendo às unidades AxRSU-4, AxRSU-8 e AxRSU-16, respectivamente. AxRSU-4 ( $m=2$ ): A unidade AxRSU-4 é projetada para dividir os operandos em blocos de 2 bits, utilizando os codificadores aproximados T1 e T2 para calcular os produtos parciais. O codificador tipo 1 (AxRSU-4-T1) é responsável por processar os produtos parciais dos blocos iguais (i.e.,  $A^2$ ), enquanto o codificador tipo 2 (AxRSU-4-T2) lida com os blocos de valores diferentes (i.e.,  $A \times B$ ). Esse arranjo permite realizar operações ao quadrado com uma simplificação significativa no número de somadores utilizados na árvore de compressão, o que reduz a área e o consumo de energia. Por

exemplo, ao aplicar a AxRSU-4 em uma entrada de 8 bits, os produtos parciais são gerados utilizando três codificadores T1 e quatro codificadores T2. As aproximações inseridas pelos codificadores T1 e T2 permitem reduzir o número de adições necessárias, utilizando 15 meio somadores (HA - *Half Adder*), quatro somadores completos (FA - *Full Adders*) e uma adição final com um somador com propagação de *carry* (CPA - *Carry Propagate Adder*) de 10 bits. Essa configuração resulta em uma redução de até 55% no consumo de energia em comparação com soluções exatas, mantendo uma precisão aceitável para várias aplicações. A unidade AxRSU-8 ( $m=3$ ) é projetada para processar dados de 3 bits simultaneamente, utilizando codificadores T1 e T2 para realizar a operação ao quadrado de forma aproximada. O AxRSU-8-T1 gera os produtos parciais para os blocos iguais, enquanto o AxRSU-8-T2 cuida dos produtos diagonais, calculando as multiplicações entre dois valores diferentes. Ao aplicar a AxRSU-8 a uma entrada de 9 bits, os blocos são divididos em três partes, com o uso de dois codificadores T1 e dois T2. Assim como no caso do AxRSU-4, a simplificação no número de produtos parciais reduz a complexidade da árvore de somadores, que agora utiliza 14 HAs e três FAs, além de um CPA de 10 bits. A redução de complexidade permite economias de energia superiores a 70% em comparação com soluções exatas, sem comprometer a qualidade dos resultados em aplicações onde a tolerância ao erro é permitida. A unidade AxRSU-16 ( $m=4$ ) divide os operandos em blocos de 4 bits, promovendo uma redução ainda maior na quantidade de produtos parciais gerados e, consequentemente, na complexidade da árvore de somadores. Nessa configuração, o AxRSU-16-T1 lida com as operações de dois valores iguais, enquanto o AxRSU-16-T2 processa as multiplicações entre dois valores diferentes. Ao aplicar a AxRSU-16 a uma entrada de 8 bits, os produtos parciais são gerados utilizando um codificador T1 e um T2. Essa configuração resulta em uma árvore de compressão com seis HAs e uma adição final com um CPA de 7 bits. A AxRSU-16 oferece um equilíbrio ainda mais vantajoso entre consumo de energia e área de circuito, com economias de energia que podem atingir até 95%, mantendo uma precisão adequada para diversas aplicações de processamento de sinais e imagens.

#### 4. AxRSU SÍNTESE E ACURÁCIA DOS RESULTADOS

Esta seção apresenta os principais resultados obtidos para a AxRSU-2<sup>m</sup>. Para a AxRSU-4, foram avaliadas 260 combinações de codificadores, para AxRSU-8, 65 combinações, e para AxRSU-16, 34 combinações. Essas arquiteturas foram comparadas com a unidade ao quadrado exata e com soluções aproximadas da literatura (GILLANI, 2018). A AxRSU-2<sup>m</sup> superou essas soluções, com novos resultados na frente de Pareto para eficiência energética e área, em todas as métricas de erro (BER, MSE, NoEB). A unidade AxRSU-8 obteve as maiores economias de energia (até 95.57%) e área (até 55.70%) em relação ao projeto exato. Os resultados principais foram: AxRSU-8 com  $k_1 = 5$  e  $k_2 = 10$  atingiu economias de energia de 95.57% e redução de área de 55.70%, com erro BER de 75.53%, e erro MSE 99.99% menor comparado à solução SquASH M1-S1 (GILLANI, 2018); AxRSU-8 com  $k_1 = 1$  e  $k_2 = 0$  obteve economia de energia de 90.68% e área de 47.36%, com BER de 3.75%; AxRSU-4 com  $k_1 = 8$  e  $k_2 = 28$  apresentou economia de energia de 86.72% e área de 66.73%, com BER de 10.45%.

##### 4.1 RESULTADO DO AXRSU EM ESTUDO DE CASO

Para ilustrar a eficácia da AxRSU-2<sup>m</sup>, foram compradas em dois estudos de caso.: O primeiro envolveu a métrica de diferenças quadráticas somadas (SSD) usada

em codificadores de vídeo, como o padrão HEVC. O segundo estudo de caso utilizou um acelerador de hardware baseado no método *Square Accumulate* (SAC), voltado ao processamento de vídeo. Os resultados mostraram que a AxRSU-2<sup>m</sup> alcançou economias de energia de até 95,57% e reduções de área de até 66,73%, mantendo uma precisão adequada para aplicações que toleram erros. Assim, a metodologia adotada combina o projeto design automatizado de unidades de operação ao quadrado aproximadas, a co-simulação com ferramentas de modelagem e síntese de hardware, e uma avaliação abrangente do desempenho utilizando métricas de erro. O uso de codificadores aproximados na AxRSU-2<sup>m</sup> permite explorar a relação entre eficiência energética, área de circuito e precisão de forma otimizada, proporcionando uma solução eficaz para sistemas que exigem hardware otimizado e podem tolerar erros.

## 5. CONCLUSÕES

Este trabalho propôs e implementou a AxRSU-2<sup>m</sup>, uma nova unidade aproximada para operações ao quadrado, introduzindo diversos codificadores aproximados para produtos parciais verticais e diagonais, denominados AxRSU-2<sup>m</sup>-T1 e AxRSU-2<sup>m</sup>-T2. Os parâmetros k1 e k2 ajustam o número de codificadores aproximados, destacando-se os melhores resultados com AxRSU-8 e AxRSU-4. Os estudos de estudo de caso, que incluíram um acelerador de hardware SSD e uma unidade de SAC, demonstraram economias de 46,20% a 95,57% em energia e de 37,68% a 66,73% na área do circuito, com uma precisão de 3,76% a 75,53% de correlação cruzada.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

- R. FERREIRA et al., **An approximate square unit based on radix-2<sup>m</sup> architecture**, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 27, no. 10, pp. 2301-2312, 2019.
- G. Gillani, M. Hanif, M. Krone, S. Gerez, M. Shafique, and A. Kokkeler, **SquASH: Approximate Square-Accumulate With Self-Healing**, IEEE Access, vol. 6, pp. 49 112–49 128, 2018.
- V. Guidotti, G. Paim, L. M. Rocha, E. Costa, S. Almeida, and S. Bampi, **Power-Efficient Approximate Newton–Raphson Integer Divider Applied to NLMS Adaptive Filter for High-Quality Interference Canceling**, Circuits, Systems, and Signal Processing, vol. 39, pp. 5729–5757, 2020.
- H. Seidel, M. M. A. da Rosa, G. Paim, E. A. C. da Costa, S. J. M. Almeida, and S. Bampi, **Approximate Pruned and Truncated Haar Discrete Wavelet Transform VLSI Hardware for Energy-Efficient ECG Signal Processing**, IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 68, no. 5, pp. 1814–1826, 2021.
- Z. Tasoulas, G. Zervakis, I. Anagnostopoulos, H. Amrouch, and J. Henkel, **Weight-Oriented Approximation for Energy-Efficient Neural Network Inference Accelerators**, IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67, no. 12, pp. 4670–4683, 2020.
- M. da Rosa, G. Paim, P. L. d. Costa, E. A. C. d. Costa, R. I. Soares, and S. Bampi, **AxPPA: Approximate Parallel Prefix Adders**, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 31, no. 1, pp. 17–28, 2023.
- K. Tsai, Y.-J. Chang, C.-H. Wang, and C.-T. Chiang, **Accuracy-Configurable Radix-4 Adder With a Dynamic Output Modification Scheme**, IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 68, no. 8, pp. 3328–3336, 2021.