MMAx : Multiplicador Modular Aproximado Didático

LOURENÇO DA CRUZ MÜLLING[1]; MORGANA M. A. ROSA[1]; EDUARDO COSTA[2]; RAFAEL IANKOWSKI SOARES[1]

[1] Universidade Federal de Pelotas – ldcmulling@inf.ufpel.edu.br,
[1]morganamacedoazevedodarosa@gmail.com,
[1]rafael.soares@inf.ufpel.edu.br,
[2]Universidade Católica de Pelotas - eduardo.costa@ucpel.edu.br

1. INTRODUÇÃO

Nas últimas décadas o uso de reticulados (do inglês, lattice) surge como uma base matemática atrativa para concepção de algoritmos de criptografia, inclusive para garantir segurança na era dos computadores quânticos (LYUBASHEVSKY; PEIKERT; REGEV, 2013). Recentemente Regev (2005) definiu o problema de aprendizado com erros (do inglês, learning with errors - LWE) e provou que este problema tem propriedades para garantir a robustez da segurança no pior caso. O problema LWE tem se mostrado versátil, servindo como base para algoritmos de chave pública entre outras aplicações. Ring LWE (RLWE) é um esquema baseado no aumento da dificuldade em solucionar o problema do vetor curto (do inglês, Shortest vector problem) em reticulados, além de utilizar chaves criptográficas com maior número de bits. Este esquema envolve aritmética polinomial e amostragem Gaussiana discreta como funções primitivas, desafiando os limites da computação convencional. No entanto, a multiplicação de polinômios e a redução modular exata podem ser intensivas em recursos de tempo e energia. Nesse contexto, técnicas de computação aproximada emergem como soluções viáveis a fim de buscar reduções modulares aproximadas para parâmetros bem definidos do RLWE, almeja-se criar criptossistemas mais eficientes, promovendo a economia de energia, velocidade de processamento e eficiência de área (D. -E. -S. Kundi, A. Khalid, S. Bian, C. Wang, M. O'Neill and W. Liu, 2022). Este trabalho visa contribuir não apenas para a segurança pós-quântica, mas também para a viabilidade prática de implementações criptográficas em dispositivos do mundo real, abrindo caminho para avanços significativos na segurança digital em um cenário em constante evolução.

2. METODOLOGIA

As operações matemáticas realizadas sobre reticulados são baseadas em aritmética modular. Neste contexto, este trabalho explora a construção de operadores aritméticos como divisão, multiplicação e subtração modulares, os quais obtêm-se o resto de uma operação. Fazendo uso de circuitos aproximados no estado da arte encontrados na literatura, tais como os somadores ETA, LOA, COPY B e Truncation, o divisor Goldschmidt e de ponto fixo e os multiplicadores DRUM, ROBA, LoBA e TRUNC, que serão implementados e testados dentro da aplicação.

A proposta deste trabalho é analisar em termos de acurácia os multiplicadores modulares aproximados (MMAx) e propor novos circuitos MMAx que equilibrem economia de área e energia em relação a bons

resultados de Acurácia, potência e Área. Para as reduções modulares construiu-se os circuitos em VHDL, e simulou-se em Matlab em conjunto com o Questasim, visando otimizações nas reduções modulares aproximadas através de pequenas mudanças nesses circuitos que serão avaliadas para mensurar a acurácia. O Matlab é usado para gerar 10.000 entradas com valores aleatórios de 16 bits de largura, e medindo o impacto de cada componente adicionado a um circuito preciso de forma individual, sendo comparada as entradas através da métrica SSIM a qual calcula o índice de similaridade estrutural.

Nos casos em que o nível de Acurácia varia entre 70% e 100%, pode-se considerar que há alta precisão dos resultados e baixo risco de erro, o que traz mais segurança para a tomada de decisão (Kundi, 2020).

3. RESULTADOS E DISCUSSÃO

Criou-se um circuito que faça uma operação de divisão de maneira tradicional como é ensinado didaticamente em escolas, o qual segue um processo passo a passo que envolve a distribuição dos dígitos do dividendo (número a ser dividido) para encontrar o quociente (resultado) e o resto (se houver), utilizando os circuitos aproximados, o qual está representado na **Figura 1**.

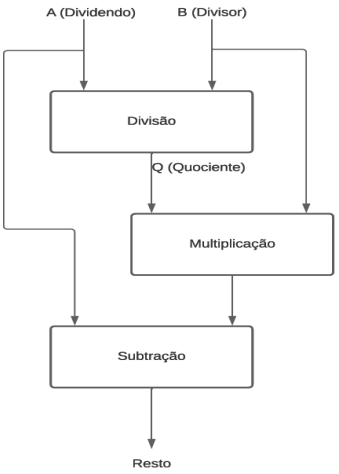

Figura 1. Esboço de uma operação aritmética modular para o RLWE construída em VHDL.

Inicialmente, realizamos uma análise do custo computacional dessa aplicação ao empregar todos os componentes em sua forma precisa. Posteriormente, substituímos esses componentes por circuitos aproximados. Com o auxílio das ferramentas Matlab e Questasim, conduzimos uma avaliação da precisão consumo de energia e área dos multiplicadores, divisores e somadores, comparando-os com os circuitos de alta precisão originalmente implementados, como podemos ver na **Tabela 1**.

Obteu-se os resultados baseados em ASIC onde as arquiteturas foram descritas em VHDL e sintetizadas usando a ferramenta de síntese Genus da Cadence na frequência de 500 MHz. As sínteses utilizaram a biblioteca de células padrão comercial ST 65nm de baixa potência com tensão de alimentação de

1,25V. Utilizou-se a ferramenta Cadence Incisive para simular todas as listas de redes considerando o arquivo SDF para atrasos precisos na propagação do sinal e falhas temporais. A simulação gera um arquivo *Toggle Count Format* (TCF), carregado na ferramenta de síntese para extração de energia realista. A metodologia de estimativa de potência utiliza a ferramenta de síntese Genus no modo PLE para gerar a netlist em nível de porta Verilog e o formato SDF.

Redutor Modular com exploração da multiplicação aproximada									
		SSIM	Cells (K gates)	Area total(µm²)	Economia de area total (%)	leakage (µW)	dinâmica (µW)	total potência (µW)	Economia de potencia total (x)
Baseline	*, +, /	1	16570	138078	0	31,684	3.491,862	3.598,546	0
Multiplicador	AxRMU-4 k=1	0.9978	8663	47220	65,80193804	20,786	1.545,431	1.566,218	2,297602249
	DRUM	1	5319	28355	79,46450557	13,367	1.477,765	1.491,132	2,413298085
	LoBA	0.9885	4741	24753	82,07317603	11,812	1.231,434	1.243,246	2,894476234
	TRUNC	0,2595	4782	24398	82,33027709	12,161	1.294,966	1.307,126	2,753021514
	ROBA	0,1139	7204	43094	68,79010414	20,995	1.570,739	1.591,734	2,260770958
Divisor	GLD	0,9996	5115	49648	64,04351164	24,160	1.657,935	1.682,095	2,139323879
	ponto fixo	1	4555	49284	64,30713075	23,899	1.643,369	1.667,268	2,158348868
Somador	copy K=1	1	6659	39009	71,74857689	17,943	2.556,260	2.574,202	1,397926814
	copy K=2	1	6657	38992	71,76088877	17,935	2.555,027	2.572,962	1,398600523
	copy K=3	0.9978	6655	38975	71,77320065	17,926	2.553,681	2.571,607	1,399337457
	copy K=4	0.9901	6653	38959	71,78478831	17,917	2.552,170	2.570,087	1,400165053
	copy K=5	0.9835	6651	38942	71,79710019	17,909	2.550,506	2.568,416	1,401075994
	copy K=6	0.9754	6649	38926	71,80868784	17,901	2.548,813	2.566,714	1,402005054
	copy K=7	0.9527	6647	38909	71,82099972	17,893	2.546,961	2.564,854	1,40302177
	copy K=8	0.9091	6645	38893	71,83258738	17,885	2.545,415	2.563,300	1,403872352
	ETA K=1	1	6659	39005	71,7514738	17,941	2.555,916	2.573,857	1,398114192
	ETA K=2	0.9978	6662	39007	71,75002535	17,945	2.556,308	2.574,253	1,397899119
	ETA K=3	0.9835	6664	39010	71,74785266	17,947	2.556,334	2.574,281	1,397883914
	ETA K=4	0.9901	6666	39012	71,74640421	17,949	2.556,196	2.574,145	1,397957769
	ETA K=5	0.9754	6668	39015	71,74423152	17,951	2.556,244	2.574,195	1,397930615
	ETA K=6	0.9527	6670	39017	71,74278306	17,952	2.556,362	2.574,315	1,397865452
	ETA K=7	0.9092	6672	39020	71,74061038	17,954	2.556,073	2.574,027	1,398021854
	ETA K=8	0.8565	6674	39022	71,73916192	17,955	2.555,717	2.573,672	1,398214691
	LOA K=1	1	6660	39016	71,74350729	17,945	2.556,536	2.574,481	1,397775319
	LOA K=2	1	6659	39004	71,75219803	17,940	2.555,615	2.573,554	1,3982788
	LOA K=3	0.9978	6658	38992	71,76088877	17,933	2.554,533	2.572,466	1,398870189
	LOA K=4	0.9901	6657	38980	71,76957951	17,927	2.553,309	2.571,237	1,399538821
	LOA K=5	0.9835	6656	38967	71,77899448	17,922	2.551,945	2.569,867	1,400284917
	LOA K=6	0.9754	6655	38955	71,78768522	17,916	2.550,549	2.568,465	1,401049265
	LOA K=7	0.9527	6654	38943	71,79637596	17,911	2.548,989	2.566,900	1,401903463
	LOA K=8	0.9092	6653	38931	71,8050667	17,905	2.547,713	2.565,618	1,402603973
	Trunc K=1	1	6661	39018	71,74205884	17,946	2.556,873	2.574,819	1,397591831
	Trunc K=2	1	6657	38996	71,75799186	17,934	2.554,937	2.572,872	1,398649447
	Trunc K=3	0.9978	6653	38973	71,77464911	17,923	2.552,891	2.570,814	1,3997691
	Trunc K=4	0.9901	6650	38947	71,79347905	17,909	2.550,580	2.568,490	1,401035628
	Trunc K=5	0.9835	6647	38920	71,81303321	17,895	2.547,844	2.565,739	1,402537826
	Trunc K=6	0.9754	6644	38894	71,83186315	17,881	2.545,050	2.562,930	1,404075024
	Trunc K=7	0.9527	6641	38868	71,85069309	17,867	2.541,997	2.559,865	1,405756163
	Trunc K=8	0.9091	6638	38842	71,86952302	17,854	2.539,352	2.557,205	1,407218428

Tabela 1. Resultados de acurácia e economia de energia e área.

Observamos que, no caso dos multiplicadores, sua eficiência é notável quando tratamos de multiplicação de 16 bits por 16 bits, isso com números significativos. No entanto, quando utilizados na aplicação, a entrada inicialmente passa por uma divisão que retorna um valor inteiro sem arredondamento. Como resultado, obtemos um número relativamente pequeno que será multiplicado pelos 16 bits que são do divisor. Nesse cenário, os circuitos LoBA e DRUM demonstram maior eficiência. O circuito TRUNC tende a eliminar a maior parte dos bits menos significativos, direcionando-os para zero. Por outro lado, o circuito ROBA realiza um arredondamento, aproximando o resultado mais próximo do divisor (entrada B).

No caso dos somadores, temos a flexibilidade de ajustar o número de bits para aproximação, identificado como 'K'. À medida que aumentamos o valor de 'K', ocorre uma redução no hardware, resultando em menor consumo de energia. No entanto, essa redução está diretamente relacionada à diminuição da precisão do circuito. É interessante observar que mesmo com 'K' definido para metade dos bits válidos, ainda conseguimos obter resultados de precisão notável.

Mesmo com os circuitos individualizados, já temos versões de circuitos de redução modular aproximada, assim como os circuitos implementados podem

ser combinados para obtenção de um hardware ainda menor. Ao explorarmos as combinações que podem ser feitas dentro da aplicação, conseguimos reduzir ainda mais os custos computacionais.

4. CONCLUSÕES

Em conclusão, as técnicas de reduções modulares computacionais desempenham um papel vital na busca por sistemas mais eficientes em termos de energia e uso de área. À medida que avançamos em direção a um mundo cada vez mais conectado, onde dispositivos da Internet das Coisas (IoT) e sistemas embarcados são onipresentes, a economia de energia e a otimização de espaço tornam-se indispensáveis. As reduções modulares aproximadas oferecem uma abordagem promissora para alcançar esses objetivos, permitindo que aplicações que não dependem de precisão absoluta operem de maneira mais eficiente, sem comprometer significativamente a qualidade das saídas. Essa eficiência energética e de área não apenas contribui para a sustentabilidade e durabilidade de dispositivos e sistemas, mas também promove avanços significativos em setores críticos, como a segurança cibernética e o processamento de dados em tempo real. Portanto, a pesquisa e implementação contínuas dessas técnicas são essenciais para atender às crescentes demandas da tecnologia moderna.

5. REFERÊNCIAS BIBLIOGRÁFICAS

RABAEY, J. M.; CHANDRAKASAN, A.; NIKOLIC, B. Digital Integrated Circuits. 3rd.ed. USA: Prentice Hall Press, 2008.

LYUBASHEVSKY, V.; PEIKERT, C.; REGEV, O. On ideal lattices and learning with errors over rings. Journal of the ACM (JACM), [S.l.], v.60, n.6, p.1–35, 2013.

D. -E. -S. Kundi, A. Khalid, S. Bian, C. Wang, M. O'Neill and W. Liu, "AxRLWE: A Multilevel Approximate Ring-LWE Co-Processor for Lightweight IoT Applications," in IEEE Internet of Things Journal, vol. 9, no. 13, pp. 10492-10501, 1 July1, 2022, doi: 10.1109/JIOT.2021.3122276.

BARRETT, P. Implementing the Rivest Shamir and Adleman public key encryption algorithm on a standard digital signal processor. In: ADVANCES IN CRYPTO-LOGY—CRYPTO'86: PROCEEDINGS, 2000. Anais. . . [S.l.: s.n.], 2000. p.311–323.

LYUBASHEVSKY, V.; PEIKERT, C.; REGEV, O. On ideal lattices and learning with errors over rings. Journal of the ACM (JACM), [S.l.], v.60, n.6, p.1–35, 2013.

REGEV, O. On lattices, learning with errors, random linear codes, and cryptography. J. ACM, 56(6):1-40, 2009. Preliminary version in STOC 2005.

D. E. S. Kundi et al., "AxMM: Area and Power Efficient Approximate Modular Multiplier for R-LWE Cryptosystem," 2020 IEEE ISCAS, Seville, Spain, 2020, pp. 1-5, doi: 10.1109/ISCAS45731.2020.9180839.