GERAÇÃO DE NÚMEROS PSEUDO ALEATÓRIOS COM UM MAPA CAÓTICO DE ESPAÇO DISCRETO

JOÃO INÁCIO MOREIRA BEZERRA[1]; ALEXANDRE MOLTER;[2]
VINÍCIUS VALDUGA DE ALMEIDA CAMARGO[2]
RAFAEL IANKOWSKI SOARES[2]
[1]*Universidade Federal de Pelotas – jimbezerra@inf.ufpel.edu.br*
[2]*Universidade Federal de Pelotas – alexandre.molter@ufpel.com.br*
[2]*Universidade Federal de Pelotas – vvacamargo@inf.ufpel.edu.br*
[2]*Universidade Federal de Pelotas – rafael.soares@inf.ufpel.edu.br*

1. INTRODUÇÃO

Geradores de números pseudo-aleatórios (PRNG) compõem uma técnica determinística que usa fórmulas matemáticas ou tabelas pré-calculadas para gerar sequências de números com propriedades aleatórias. PRNGs são amplamente usados em áreas como jogos eletrônicos, processamento de sinais, simulações estocásticas e criptografia, onde a segurança das chaves depende da aleatoriedade. Com o aumento de dispositivos conectados à internet, o uso de PRNGs na criptografia está crescendo (AL-MHADAWI, 2023). No contexto da criptografia, o uso de sistemas caóticos como fonte para PRNG tem atraído grande interesse dos pesquisadores. Essa atenção se deve às características inerentes dos sistemas caóticos, como sensibilidade às condições iniciais, comportamento aleatório e ergodicidade, que se alinham bem com as necessidades dos sistemas criptográficos (BEZERRA, 2023).

Os geradores de números aleatórios são classificados em dois tipos: (i) Geradores de Números Verdadeiramente Aleatórios (TRNG) e (ii) Geradores de Números Pseudo-Aleatórios (PRNG) (IBRAHIM, 2022). TRNGs são não determinísticos, baseados em características de componentes eletrônicos, como transistores e memristores, e oferecem vantagens como menor consumo de área e energia em comparação aos PRNGs, além de maior taxa de transferência. No entanto, como os valores produzidos pelos TRNGs não podem ser reproduzidos, eles não são adequados para criptografia, que exige valores reproduzíveis para a decifragem. Por outro lado, PRNGs são amplamente utilizados em aplicações criptográficas por seu comportamento determinístico, mas imprevisível (BEZERRA, 2023). Recentes arquiteturas de PRNG caóticos foram propostas, como por exemplo SIVARAMAN (2020) e RAMAKRISHNAN (2022), contudo estas arquiteturas apresentam limitações, como a dependência de sistemas caóticos de alta dimensionalidade ou arquiteturas de espaço contínuo, limitações que aumentam o consumo de energia do bloco PRNG e que já haviam sido discutidas por PREISHUBER (2018).

Em arquiteturas de espaço contínuo, as variáveis de estado são fracionárias, mas ao serem integradas em sistemas discretos com PRNGs, devem ser convertidas em inteiros. Essa conversão gera desafios de desempenho e segurança. Em termos de desempenho, a resolução de cifras caóticas de alta dimensão em tempo contínuo exige métodos numéricos, como o Forward Euler ou Runge-Kutta de 4ª ordem, que envolvem muitas operações, aumentando o consumo de área, energia e a latência. Esses métodos impedem a execução paralela, reduzindo a eficiência energética. Em termos de segurança, a conversão

de valores contínuos para aproximações discretas degrada o sistema, levando os valores caóticos a ciclos previsíveis, o que pode ser mitigado, mas requer mais área.

Uma arquitetura de espaço discreto, por sua vez, trabalha apenas com valores exatos, não lidando com os desafios relativos à aproximações. Além disso, não requisita o uso de métodos numéricos de alta complexidade, sendo tratada apenas como uma função recursiva. Deste modo, este trabalha apresenta o resultado da síntese ASIC (*Application Specific Integrated Circuit*) de um mapa caótico descrito em um espaço discreto. Comparado ao estado da arte da implementação ASIC de sistemas caóticos, apresentou-se uma redução de aproximadamente 80% no consumo de energia em comparação ao estado da arte.

2. METODOLOGIA

A formulação do mapa caótico de espaço discreto, levando em conta a implementação de 32 *bits,* foi proposta por Lambic(2020) e está expressa na Equação (1) que segue.

$$f(x_n) = x_{n+1} = (z << (z\%32))|(z >> (32 - z\%32)), \qquad (1)$$

$$z = (2^{-16}x_n + 1) \times (x_n\%2^{16} + 1) + 1.$$

Na equação acima, x_n representa a variável de estado do mapa caótico, sem a presença de uma variável de controle. Embora não haja um parâmetro de controle, uma análise exaustiva das condições iniciais revelou que o mapa não possui pontos fixos, exibindo um comportamento caótico independentemente da condição inicial.

Do ponto de vista de uma arquitetura de *hardware,* o mapa caótico de espaço discreto definido de acordo com a Equação (1) apresenta as seguintes vantagens em relação aos mapas ou sistemas caóticos de espaço contínuo.

- Trabalha apenas com valores inteiros, eliminando a necessidade de operações complexas de multiplicação e divisão com frações, além da quantização de valores contínuos. Na Equação (1), as operações de divisão e resto tem como argumento potências de 2, de forma que podem ser implementadas como deslocamento de *bits,* que é consideravelmente menos custoso do que um *hardware* dedicado para as operações de divisão e resto.
- Garante a reprodutibilidade em diversas arquiteturas de FPGA e ASIC, já que a variável de estado *x* é um valor inteiro exato, em vez de um valor fracionário aproximado. Assim, eliminam-se erros de arredondamento e aproximação que poderiam variar entre dispositivos e arquiteturas.
- Evita a degradação dinâmica, por ser composto exclusivamente de valores inteiros, prevenindo erros de arredondamento e aproximação. Assim, o mapa caótico mantém suas características, independentemente do número de iterações necessárias em uma aplicação, como a criptografia de imagens por exemplo.

A descrição do *hardware* do cifrador proposto destaca-se em relação a outros trabalhos na literatura por sua baixa latência, com apenas três ciclos de *clock.* Trabalhos como TUNA (2019) e SAMBAS (2022), em que sistemas caóticos de

espaço caótico são implementados, com a latência chegando a atingir 50 ciclos de *clock*. Os três ciclos de *clock* necessários para a descrição do circuito relativo ao mapa caótico de espaço discreto são apresentados à seguir.

- **Ciclo 1:** Leitura do valor obtido na iteração anterior.
- **Ciclo 2:** Seleção de *bits* do valor lido no ciclo anterior, acumular cada um desses valores por um, e aplicar a operação de multiplicação.
- **Ciclo 3:** Incrementa o resultado da multiplicação do ciclo anterior por um, aplica o deslocamento e as operações XOR.

3. RESULTADOS E DISCUSSÃO

Nesta seção, apresenta-se e discute-se os resultados da síntese ASIC do PRNG que tem o mapa caótico discreto definido pela Equação (1) como fonte. As arquiteturas foram modeladas em VHDL e sintetizadas em diferentes frequências utilizando a ferramenta de síntese Cadence Genus. Uma biblioteca comercial de células padrão de baixo consumo, ST 65 nm, com uma tensão de alimentação de 1,25 V, foi empregada, além de uma frequência operacional de 100 MHz.

Na Tabela 1, é realizada a comparação da síntese ASIC da arquitetura proposta com o estado da arte da síntese ASIC de PRNG. É especialmente notável que a arquitetura proposta reduziu o consumo de energia de KOPPARTHI (2022) em 79.56%, com o uso do mapa caótico de espaço discreto ao invés de um sistema caótico de espaço contínuo. A arquitetura proposta apresenta um *throughput* menor em virtude do multiplicador gerado pela ferramenta de síntese ter sido usado. Com o uso de multiplicadores especiais, esta métrica pode ser melhorada sem um aumento no consumo de potência, reduzindo significativamente o consumo de energia da arquitetura.

Trabalho	Tecnologia	Área (μm²)	Frequência de operação (MHz)	Potência (mW)	*Throughput* (Gbps)	Energia (pJ/bit)
Este	65 nm	10643	100	2.07	1.07	1.93
KOPPARTHI (2022)	90 nm	78100	106	10	1.7	9.44
Garcia (2019)	180 nm	196820	125	24.1	1	24.1

Tabela 1: Comparação dos resultados da síntese ASIC do PRNG proposto com o estado da arte da literatura.

4. CONCLUSÕES

Neste trabalho, a implementação ASIC de um mapa caótico de espaço discreto foi apresentada, mostrando que o uso da arquitetura no espaço discreto reduz o consumo de energia do bloco gerador de sequências pseudo-aleatórias, por meio do aumento no *throughput* e da redução no consumo de potência.

Estes resultados podem ser melhorados com o uso de multiplicadores específicos para aplicações *low-power*, ou ainda multiplicadores aproximadores,

já que o gargalo da arquitetura se encontra na operação de multiplicação. Como trabalho futuro, serão testadas outras arquiteturas de multiplicadores, além de inserir o bloco PRNG em arquiteturas mais complexas, como por exemplo cifradores.

5. REFERÊNCIAS BIBLIOGRÁFICAS

AL-MHADAWI, Mohammed M.; ALBAHRANI, Ekhlas Abbas; LAFTA, Sadeq H. Efficient and secure chaotic PRNG for color image encryption. **Microprocessors and Microsystems**, v. 101, p. 104911, 2023.

BEZERRA, João Inácio Moreira et al. A novel simultaneous permutation–diffusion image encryption scheme based on a discrete space map. **Chaos, Solitons & Fractals,** v. 168, p. 113160, 2023.

GARCIA-BOSQUE, Miguel et al. A 1 gbps chaos-based stream cipher implemented in 0.18 μm CMOS technology. **Electronics**, v. 8, n. 6, p. 623, 2019.

IBRAHIM, Hebatallah M. et al. Memristor-based PUF for lightweight cryptographic randomness. **Scientific reports**, v. 12, n. 1, p. 8633, 2022.

KOPPARTHI, Venkata Reddy et al. Hardware architecture of a digital piecewise linear chaotic map with perturbation for pseudorandom number generation. **AEU-International Journal of Electronics and Communications**, v. 147, p. 154138, 2022.

LAMBIĆ, Dragan. A new discrete-space chaotic map based on the multiplication of integer numbers and its application in S-box design. **Nonlinear Dynamics**, v. 100, n. 1, p. 699-711, 2020.

PREISHUBER, Mario et al. Depreciating motivation and empirical security analysis of chaos-based image and video encryption. **IEEE Transactions on Information Forensics and Security**, v. 13, n. 9, p. 2137-2150, 2018.

RAMAKRISHNAN, Balamurali et al. Image encryption with a Josephson junction model embedded in FPGA. **Multimedia Tools and Applications**, v. 81, n. 17, p. 23819-23843, 2022.

SAMBAS, Aceng et al. A novel 3D chaotic system with line equilibrium: multistability, integral sliding mode control, electronic circuit, FPGA implementation and its image encryption. **IEEE Access**, v. 10, p. 68057-68074, 2022.

SIVARAMAN, R.; RAJAGOPALAN, Sundararaman; AMIRTHARAJAN, Rengarajan. FPGA based generic RO TRNG architecture for image confusion. **Multimedia Tools and Applications**, v. 79, n. 19, p. 13841-13868, 2020.

TUNA, Murat et al. High speed FPGA-based chaotic oscillator design. **Microprocessors and Microsystems**, v. 66, p. 72-80, 2019.