Sequências Pseudo-Caóticas Geradas Pelo Mapa de Arnold Sobre $\mathbb{Z}_{2^m}$: Análise de Período e Implementação em FPGA

Carlos E. C. Souza, Daniel P. B. Chaves, Cecilio Pimentel e Wallace Nascimento Melo

***Resumo*— Neste trabalho é proposto um método de geração de sequências pseudo-caóticas unidimensionais baseado no mapa de Arnold discreto sobre o anel de inteiros $\mathbb{Z}_{2^m}$. O período das sequências geradas é calculado analiticamente utilizando propriedades das sequências de Fibonacci sobre $\mathbb{Z}_{2^m}$. As sequências pseudo-caóticas são empregadas como um gerador de números pseudo-aleatórios e suas propriedades estatísticas são avaliadas pela bateria de testes estatísticos NIST. Finalmente, o gerador proposto é implementado em FPGA e sua complexidade é analisada.**

***Palavras-Chave*— Sequências pseudo-aleatórias, mapas caóticos, caos discreto, FPGA.**

***Abstract*— In this work we propose a method to generate pseudo-chaotic one-dimensional chaotic sequences based on the discrete Arnold map defined over the integer ring $\mathbb{Z}_{2^m}$. The period of the generated sequences is evaluated analitically using properties of the Fibonacci sequence over $\mathbb{Z}_{2^m}$. The pseudo-chaotic sequences are employed as pseudo-random number generators and their statistical properties are analyzed by the statistical suite NIST. Finally, the proposed pseudo-random number generator is implemented in FPGA and its complexity is analyzed.**

***Keywords*— Pseudo-random sequences, chaotic maps, discrete chaos, FPGA.**

I. Introdução

Geradores de números pseudo-aleatórios (PRNGs, *pseudo-random number generators*) baseados em mapas caóticos vêm sendo considerados potenciais candidatos para o projeto de aplicações criptográficas [1], [2]. Propriedades características destes mapas como sensibilidade às condições iniciais, espectro banda larga, comportamento recursivo e não periódico [3] são apropriadas para estas aplicações. Entretanto, quando o mapa é definido sobre os números reais as operações de ponto flutuante alteram a dinâmica caótica devido à sensibilidade às condições iniciais [4]. Em consequência, a dinâmica caótica nunca é reproduzida fielmente.

Uma proposta de mapas caóticos definidos sobre estruturas discretas é apresentada em [5]. No referido trabalho, mapas caóticos discretos são obtidos por um processo de discretização de mapas caóticos reais definidos de forma a apresentar periodicidade. Desta forma, não existe caos no sentido restrito em mapas definidos sobre estruturas discretas. Porém, dentro de um período o mapa possui características similares ao comportamento caótico e no limite em que o período vai ao infinito o mapa recupera as características do mapa original como por exemplo o expoente de Lyapunov, que é interpretado como a taxa de dispersão entre pontos vizinhos no conjunto discreto [5]. Este tipo de comportamento é denominado em [4], [5] de caos discreto ou pseudo-caos.

Uma alternativa para geração de caos discreto é a utilização de mapas caóticos definidos sobre anéis de inteiros [4], [6]. Além de permitir a reprodução exata da dinâmica, a utilização destes mapas reduz a complexidade computacional do sistema em relação a mapas definidos sobre os reais devido ao uso de operações de ponto fixo. Apesar da aplicação iterativa de um determinado mapa sobre uma condição inicial gerar sequências necessariamente periódicas devido à sua estrutura modular, esta abordagem é vantajosa quando se considera anéis de cardinalidade grande, pois as sequências geradas possuem período longo o suficiente para estas aplicações.

Neste trabalho é proposto um método para construção de sequências unidimensionais baseadas no mapa de Arnold discreto sobre o anel de inteiros módulo 2^m, $m \in \mathbb{N}, m \geq 2$, denotado por $\mathbb{Z}_{2^m}$. As propriedades de período das sequências propostas são analisadas utilizando propriedades das sequências de Fibonnaci sobre $\mathbb{Z}_{2^m}$. É demonstrado analiticamente que o período máximo é dado por $3 \times 2^{m-3}$ e como este período se comporta em relação às condições iniciais. As sequências propostas são utilizadas para projetar um PRNG. O PRNG é implementado em FPGA (*Field Programmable Gate Array*) e sua complexidade aritmética é analisada. Finalmente, as propriedades estatísticas das sequências geradas pelo PRNG proposto são analisadas pela bateria de testes estatísticos NIST [7].

O restante do artigo está dividido como segue. Na Seção II o mapa de Arnold discreto é apresentado. Na Seção III as sequências propostas são definidas e analisadas. O PRNG e FPGA são detalhados na Seção IV. Por fim, as conclusões são apresentadas na Seção V.

II. O Mapa de Arnold Discreto

O mapa de Arnold discreto $\Gamma : \mathbb{Z}_q \times \mathbb{Z}_q \rightarrow \mathbb{Z}_q \times \mathbb{Z}_q$ é definido por

$$\Gamma(x, y) = (2x + y, x + y) \pmod{q} \tag{1}$$

em que $q = p^m$, p é um número primo e $\mathbb{Z}_q$ é o anel de inteiros módulo q. A aplicação iterativa de Γ a partir de uma condição inicial $(x_0, y_0) \in \mathbb{Z}_q \times \mathbb{Z}_q$ gera sequências bidimensionais $s =$

Os autores são do Departamento de Eletrônica e Sistemas, Universidade Federal de Pernambuco, Recife-PE, e-mails: {carlos.ecsouza, cecilio, daniel.chaves, wallace.nmelo}@ufpe.br. Este trabalho foi parcialmente financiado pela CAPES, CNPq e FACEPE.

$\{(x_0, y_0), (x_1, y_1), (x_2, y_2), \ldots\}$ em que cada par $(x_n, y_n) \in s$ é dado de forma recursiva por

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \pmod q. \quad (2)$$

Definindo

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \quad (3)$$

e

$$\mathbf{B} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \quad (4)$$

tal que $\mathbf{B}^2 = \mathbf{A}$ então

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \mathbf{B}^{2n} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \pmod q. \quad (5)$$

Como Γ é definido sobre uma estrutura modular, segue que Γ é necessariamente periódico. A análise do espectro de períodos para o mapa de Arnold generalizado é feita em [8], [9].

A. Análise de Período de Γ

As potências de $\mathbf{B}$ podem ser escritas utilizando-se os elementos da sequência de Fibonacci $F = \{0, 1, 1, 2, 3, 5, 8, 13, \ldots\}$, definida por $F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2},\ n \geq 2$. A n-ésima potência de $\mathbf{B}$ em função dos números de Fibonacci é dada por [10]

$$\mathbf{B}^n = \begin{bmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{bmatrix} \quad (6)$$

e portanto (5) pode ser escrita como

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} F_{2n+1} & F_{2n} \\ F_{2n} & F_{2n-1} \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \pmod q. \quad (7)$$

O período do mapa de Arnold discreto é definido pelo menor valor T tal que $\mathbf{A}^T \equiv \mathbb{I}_2 \pmod q$ em que $\mathbb{I}_2$ é a matriz identidade de ordem dois [8]. Neste caso, $(x_{n+T}, y_{n+T}) \equiv (x_n, y_n) \pmod q, \forall n$. De (7) temos que a condição de período de Γ está associada ao período da sequência de Fibonacci módulo q, conhecido por período de Pisano e denotado por $\pi(q)$. Quando $\pi(p) \neq \pi(p^2)$ então $\pi(p^m) = \pi(p) \times p^{m-1}$ [11].

Da condição de período, $F_{\pi(q)} = F_{0+\pi(q)} = F_0 = 0$ e da relação de recorrência da sequência de Fibonnaci segue que $F_{\pi(q)+1} = F_{\pi(q)-1} = 1$, logo $\mathbf{B}^{\pi(q)} \equiv \mathbf{I}_2 \pmod q$. Desta forma,

$$\begin{bmatrix} x_{\pi(q)/2} \\ y_{\pi(q)/2} \end{bmatrix} = \mathbf{B}^{\pi(q)} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \pmod q \quad (8)$$

$$= \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \pmod q \quad (9)$$

e portanto o período de Γ é dado por $\pi(q)/2$.

III. Sequências-z

Nesta seção são definidas as sequências unidimensionais $\{z_n\}$ a partir do mapa de Arnold discreto. É feita uma análise da periodicidade das sequências $\{z_n\}$ e estabelecidas as condições que geram sequências com período máximo. Desta forma, pode-se descrever a estrutura do espaço de chaves gerado pelo conjunto de condições iniciais possíveis. Para isto, serão utilizadas duas conhecidas relações para a sequência de Fibonacci: a identidade de Catalan

$$F_n^2 - F_{n+r}F_{n-r} = (-1)^{n-r}F_r^2,\ r \in \{1, 2, 3, \ldots\} \quad (10)$$

e a identidade de Cassini

$$F_{n-1}F_{n+1} - F_n^2 = (-1)^n \quad (11)$$

que é um caso particular da identidade de Catalan para $r = 1$.

Neste trabalho vamos considerar $p = 2$ e vamos denotar $T = \pi(2^m)$. Todas as operações subsequentes estão definidas no anel $\mathbb{Z}_{2^m}$. Neste caso, $F \equiv \{0, 1, 1, 0, 1, 1, 0, \ldots\} \pmod 2$. Claramente, $\pi(2) = 3$ e consequentemente $\pi(2^m) = 3 \times 2^{m-1}$. Desta forma, conclui-se que o período de Γ definido em $\mathbb{Z}_{2^m}$ é dado por $T/2 = \pi(2^m)/2 = 3 \times 2^{m-2}$.

A. Definição de $\{z_n\}$

Definimos a sequência-z, denotada por $\{z_n\} = \{z_0, z_1, z_2, \ldots\},\ z_n \in \mathbb{Z}_{2^m},\ \forall n$, por

$$z_n = x_n y_n \pmod{2^m} \quad (12)$$

em que $(x_n, y_n) \in \mathbb{Z}_{2^m} \times \mathbb{Z}_{2^m}$ é o n-ésimo par gerado pela aplicação iterativa de Γ sobre uma condição inicial $(x_0, y_0) \in \mathbb{Z}_{2^m} \times \mathbb{Z}_{2^m}$. Utilizando (7) e (12) obtemos

$$\begin{aligned} z_n &\equiv x_0^2 F_{2n}F_{2n+1} + x_0 y_0 (F_{2n+1}F_{2n-1} + F_{2n}^2) \\ &+ y_0^2 F_{2n}F_{2n-1} \pmod{2^m}. \end{aligned} \quad (13)$$

B. Sequências-z Identicamente Nulas

O elemento z_n é função de (x_0, y_0) e dos números de Fibonacci. Dependendo destes valores a operação modular pode dar origem a sequências identicamente nulas. Estas condições são detalhadas a seguir.

Proposição 1: Se $(x_0, y_0) = (0, 0)$, $z_n \equiv 0 \pmod{2^m}$, $\forall n$.

Demonstração: Segue diretamente de (13). ■

Proposição 2: Se $(x_0, y_0) = (0, y_0)$ e $2^k | y_0, k \in \mathbb{Z}_{2^m}$ ou $(x_0, y_0) = (x_0, 0)$ e $2^k | x_0, k \in \mathbb{Z}_{2^m}$ então $z_n \equiv 0 (\mathrm{mod} 2^m) \forall n$ se $2k - m \geq 0$.

Demonstração: Seja $(x_0, y_0) = (0, y_0)$ tal que $2^k | y_0$. Podemos escrever $(x_0, y_0) = (0, 2^k y_0')$, em que $y_0' = y_0/2^k$, então $z_n \equiv 2^{2k} y_0'^2 F_{2n}F_{2n-1} \pmod{2^m}$. Portanto $z_n \equiv 0 \pmod{2^m} \forall n$ se e somente se $2^m | 2^{2k}$, ou seja, $2k - m \geq 0$. O caso $(x_0, y_0) = (x_0, 0)$ é análogo. ■

Proposição 3: Se $2^{k_1} | x_0$ e $2^{k_2} | y_0$ então $z_n \equiv 0 \pmod{2^m}$ se $2 \cdot \min(k_1, k_2) - m \geq 0$.

Demonstração: Seja $(x_0, y_0) = (2^{k_1} x_0', 2^{k_2} y_0')$. De (13) temos $z_n \equiv 2^{2k_1} x_0'^2 F_{2n}F_{2n+1} + 2^{2k_2} y_0'^2 F_{2n}F_{2n-1} + 2^{k_1+k_2} x_0' y_0' (F_{2n}^2 + F_{2n+1}F_{2n-1}) \pmod{2^m}$. Seja $k_{min} = \min(k_1, k_2)$ então $2^{2k_{min}} | 2^{2k_1}, 2^{2k_{min}} | 2^{2k_2}$, e $2^{2k_{min}} | 2^{k_1+k_2}$. Logo, se $2k_{min} - m \geq 0$, $z_n \equiv 0 \pmod{2^m} \forall n$. ■

A seguir as propriedades de período das sequências-z são analisadas.

C. Análise de Período de $\{z_n\}$

Lema 1: $F_{T/2} \equiv 0 \pmod{2^m}$ para $m > 2$.

Demonstração: Usando (10) para $n = T = \pi(2^m)$ e $r = T/2$ temos $F_T^2 - F_{T+T/2}F_{T-T/2} = (-1)^{T-T/2}F_{T/2}^2$. Note que $(-1)^{T/2} = 1$ pois $T/2$ é par $\forall\, m > 2$. Lembrando que $F_T = F_{\pi(2^m)} \equiv 0 \pmod{2^m}$ segue que $-F_{T/2}^2 \equiv F_{T/2}^2 \pmod{2^m}$ e portanto $F_{T/2} \equiv 0 \pmod{2^m}$ para $m > 2$. ■

Lema 2: $F_{T/2+1}^2 \equiv F_{T/2-1}^2 \equiv 1 \pmod{2^m}$.

Demonstração: De (11) para $N = T/2$ temos $F_{T/2-1}F_{T/2+1} \equiv 1 \pmod{2^m}$. Da relação de recorrência da sequência de Fibonnaci temos que $F_{T/2+1} = F_{T/2-1}$, logo $F_{T/2+1^2} \equiv F_{T/2-1}^2 \equiv 1 \pmod{2^m}$. ■

A seguir apresentamos o resultado principal sobre o período das sequências-z.

Teorema 1: Se (x_0, y_0) não gera a sequência identicamente nula e é tal que 2^k não divide x_0 e y_0 simultaneamente, então o período de $\{z_n\}$ é igual $T/4 = 3 \times 2^{m-3} = \pi(2^m)/4$.

Demonstração: Temos que

$$\begin{bmatrix} x_{T/4} \\ y_{T/4} \end{bmatrix} = \mathbf{B}^{T/2} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} F_{T/2+1} & F_{T/2} \\ F_{T/2} & F_{T/2-1} \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}. \quad (14)$$

Pelo Lema 1 e Lema 2

$$\begin{bmatrix} x_{T/4} \\ y_{T/4} \end{bmatrix} \equiv \alpha \mathbf{I}_2 \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \pmod{2^m} \quad (15)$$

em que $\alpha = F_{T/2+1} = F_{T/2-1}$. Logo

$$z_{T/4} \equiv \alpha^2 x_0 y_0 \equiv x_0 y_0 \pmod{2^m} \quad (16)$$

e consequentemente $z_{T/4+i} = z_i \forall i$. Agora considere a sequência $(\mathbf{I}, \mathbf{B}^2, \mathbf{B}^4, \ldots, \mathbf{B}^{T/2-2})$. Esta sequência é um grupo cíclico com gerador $\mathbf{B}^2$. Vamos assumir que existe algum $T' < T/4$ que é período de $\{z_n\}$. A condição de período implica que $z_n = x_0 y_0$. Para que esta condição seja verdadeira para todo (x_0, y_0), as equações a seguir devem ser satisfeitas

$$F_{2T'}F_{2T'+1} \equiv 0 \pmod{2^m} \quad (17)$$

$$F_{2T'}F_{2T'-1} \equiv 0 \pmod{2^m} \quad (18)$$

$$F_{2T'+1}F_{2T'-1} + F_{2T'}^2 \equiv 1 \pmod{2^m}. \quad (19)$$

Caso $F_{2T'+1} \equiv 0 \pmod{2^m}$ então $F_{2T'}^2 \equiv 1 \pmod{2^m}$ e portanto $F_{2T'} \neq 0$. Por outro lado, $F_{2T'} \neq 0$ implica que $F_{2T'-1} \equiv 0 \pmod{2^m}$ e toda a sequência é nula. Da mesma forma se $F_{2T'-1} \equiv 0 \pmod{2^m}$ toda a sequência é nula. Caso $F_{2T'} \equiv 0 \pmod{2^m}$ então $F_{2T'+1}F_{2T'-1} \equiv 1 \pmod{2^m}$. Pela relação de recorrência da sequência de Fibonnaci, $F_{2T'-1} \equiv F_{2T'+1} \pmod{2^m}$ e podemos escrever $F_{2T'+1}^2 \equiv 1 \pmod{2^m}$. Esta equação possui 4 possíveis soluções: $1, 2^{m-1}-1, 2^{m-1}+1, 2^m - 1$. Da propriedade de grupo cíclico das potências de $\mathbf{B}^2$, a solução 1 não pode ocorrer pois $\mathbf{B}^{T/2} \equiv \mathbf{I}_2 \pmod{2^m}$ e todos os elementos de um grupo cíclico são distintos. Segue que $\mathbf{B}^{T'}$ é da forma $\beta \mathbf{I}_2$ com $\beta = 2^{m-1}-1, 2^{m-1}+1, 2^m-1$. Portanto, $\mathbf{B}^{2T'} = \beta^2 \mathbf{I}_2 \equiv \mathbf{I}_2 \pmod{2^m}$. Como $2T' < T/2$, é uma contradição. Concluímos que $T/2$ é o período de $\{z_n\}$ e não existe outro período menor que este. ■

Teorema 2: Se (x_0, y_0) não gera a sequência identicamente nula e $2^k | x_0$ e $2^k | y_0$ então o período de $\{z_n\}$ é dado por $T/(4 \times 2^{2k})$.

Demonstração: Considere $(x_0, y_0) = (2^k x_0', 2^k y_0')$ em que $x_0' = x_0/2^k$ e $y_0' = y_0/2^k$, então z_n é dado por

$$\begin{aligned} z_n \;\equiv\; & 2^{2k}[(x_0')^2 F_{2n}F_{2n+1} + x_0'y_0'(F_{2n+1}F_{2n-1} + F_{2n}^2) \\ + \;& (y_0')^2 F_{2n}F_{2n-1}] \pmod{2^m}. \end{aligned} \quad (20)$$

A condição de período para $\{z_n\}$ é

$$z_n \equiv 2^{2k} x_0' y_0' \pmod{2^m}. \quad (21)$$

Assumindo que $2k < m$, $\mathrm{mdc}(2^m, 2^{2k}) = 2^{2k}$ e portanto

$$z_n/2^{2k} \equiv x_0' y_0' \pmod{2^m/2^{2k}}. \quad (22)$$

Logo, a sequência-z gerada por $(2^k x_0', 2^k y_0')$ é equivalente a outra sequência-z com condição inicial (x_0', y_0') que é periódica para $q' = 2^{m-2k}$ com período $(T/4)/2^{2k} = T/(4 \times 2^{2k})$. ■

Proposição 4: O número de condições iniciais que geram sequências-z com período máximo $T/4$ é $3 \times 2^{2m-2}$.

Demonstração: O número de possíveis condições iniciais $(x_0, y_0) \in \mathbb{Z}_{2^m} \times \mathbb{Z}_{2^m}$ é $2^m \times 2^m = 2^{2m}$. O número de elementos em $\mathbb{Z}_{2^m}$ que não são fatores de 2^m é $\phi(2^m) = 2^{m-1}$, em que $\phi(\cdot)$ é a função phi de Euler [12]. Portanto, o número de condições iniciais que não geram período máximo é $(2^m - 2^{m-1})^2 = 2^{2m-2}$. Finalmente, o número de condições iniciais que geram sequências-z com período máximo é $2^{2m} - 2^{2m-2} = 3 \times 2^{2m-2}$. ■

IV. Geração de Números Pseudoaleatórios e Implementação em FPGA

Nesta seção as sequências-z são utilizadas para projetar um PRNG e sua complexidade computacional é avaliada por implementação em FPGA.

A. Parâmetros de Síntese e de Análise do PRNG

Inicialmente, uma sequência $\{z_n\}$ com período máximo é gerada a partir de uma condição inicial (x_0, y_0) aleatória que se enquadra na condição do Teorema 1. Os inteiros $z_i \in \{z_n\}$ são representados na forma binária $b_{m-1}b_{m-2}\ldots b_2 b_1 b_0$, $b_j \in \{0, 1\}$, com m bits para $q = 2^m$. Em particular, foram utilizadas representações binárias de 32 e 64 bits. Sequências binárias são geradas pela concatenação de bits extraídos dos z_i's associados a uma sequência de inteiros $\{z_n\}$. Para analisar as propriedades estatísticas das sequências binárias obtidas, foi utilizada a bateria de testes estatísticos NIST versão SP800-22 [7]. Cada teste do NIST é avaliado com nível de significância $\alpha = 0{,}01$, sendo este o valor recomendado em [7]. O teste é realizado para conjuntos de 10^2 subsequências binárias em que cada uma possui comprimento 10^6.

Fixado um valor de m, é possível determinar um subconjunto de bits extraídos das amostras z_i para os quais a sequência binária gerada seja aprovada no NIST. A cardinalidade desse subconjunto é denotada por k. Foram realizadas simulações computacionais não exaustivas com a bateria de testes NIST para determinar valores elevados de k. No caso $m = 32$, o valor máximo obtido é $k = 8$ bits/amostra em que é extraído o bloco $b = b_0 b_{10} b_3 b_6 b_9 b_1 b_2 b_7$ de cada z_i e no caso $m = 64$ obtém-se $k = 16$ bits/amostra com a extração do

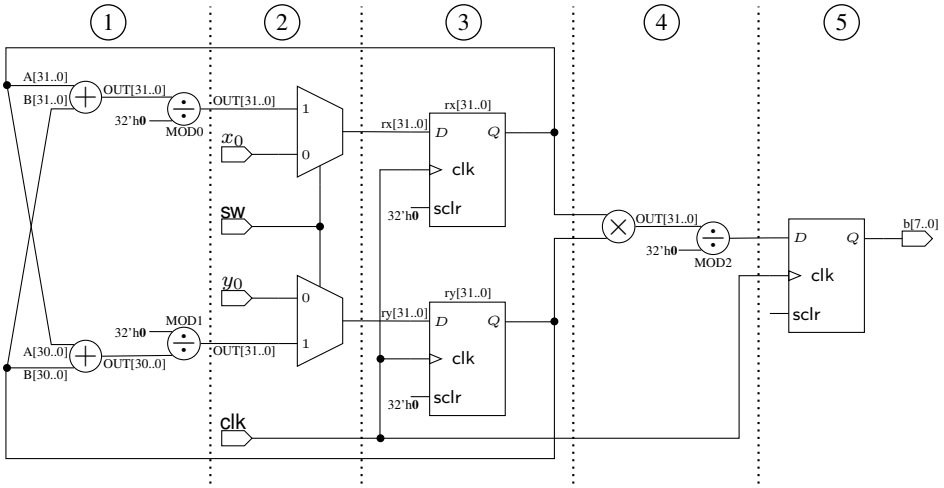

Fig. 1. RTL utilizado para geração de sequências binárias a partir de uma sequência-z para $m = 32$.

TABELA I
TESTE NIST PARA AS SEQUÊNCIAS BINÁRIAS OBTIDAS DE $\{z_n\}$ PARA $m = 32$ ($k = 8$) E 64 ($k = 16$).

Statistical test	$m = 32$	$m = 64$
Frequency	0.99	1
Block Frequency	0.99	0.99
Cumulative Sums*	0.99	1
Runs	0.99	0.99
Longest Run	0.99	0.97
Rank	0.98	0.99
FFT	1	1
Non-Ovla. Temp.*	0.96	0.96
Ovla. Temp.	1	0.97
Universal	0.98	0.99
Approximate Entropy	1	0.98
Ran. Exc.*	0.97	0.97
Ran. Ex. Var.*	0.97	0.97
Serial*	0.97	1
Linear Complexity	1	1

bloco $b = b_{16}b_{37}b_{35}b_{30}b_{40}b_{17}b_{14}b_{32}b_{39}b_{20}b_{12}b_{25}b_{13}b_{22}b_{15}b_{31}$ de cada z_i. Na Tabela I é apresentada a proporção obtida em cada teste estatístico realizado pelo NIST para m igual a 32 e 64, indicando que a qualidade estatística é similar para $m = 32$ e $m = 64$. Nos testes múltiplos (indicados por *) é mostrado apenas o valor mínimo do teste.

B. Implementação em FPGA

A implementação do mapa é facilmente escalonável para qualquer $q = 2^m$, como apresentado no código Verilog em Algoritmo 1 para $m = 32$. Para tanto, é necessário especificar adequadamente o parâmetro m e os k bits selecionados de z_i. Os testes empregam a FPGA Cyclone® IV EP4CE115F29C7 da Intel®, sendo apresentado na Fig. 1 o RTL (*register transfer level*) associado ao código em Algoritmo 1. Como destacado na figura, o circuito pode ser dividido em cinco níveis. O primeiro calcula os valores de x_{n+1} e y_{n+1} a partir de x_n e y_n, respectivamente, como especificado em (2). O segundo, formado por dois MUX 2×1, seleciona se as operações são realizadas sobre os valores x_0 e y_0 (condições iniciais) ou sobre valores subsequentes. No nível três, formado por um par de flip-flops tipo D, os valores de x_n e y_n são armazenados por um período de clock, durante o qual os valores de x_{n+1} e y_{n+1} são calculados no nível um e o valor de z_n é calculado no nível quatro. Por fim, na próxima borda de subida do clock, a saída do flip-flop no nível cinco é atualizada para o valor z_n, quando as saídas dos flip-flops no nível três também são atualizadas para os valores x_{n+1} e y_{n+1}.

O circuito calcula um elemento z_i da sequência $\{z_n\}$ para cada período do sinal de clock. Um conjunto de k bits é extraído de cada z_i (com a seleção dos bits como especificado na Subseção IV-A), podendo ser enviados sequencialmente e, assim, formando uma sequência binária. A complexidade do circuito é concentrada essencialmente na operação produto para o cálculo de z_n, no nível quatro. Como os valores são representados na base dois e reduzidos módulo uma potência de dois, a operação $2x_n$ na expressão para o cálculo de x_{n+1} do mapa é realizada pelo deslocamento por um bit do bloco associado a x_n na direção do bit mais significativo, preenchendo o bit menos significativo com zero. Portanto, essa operação não agrega complexidade ao circuito.

Para a aquisição dos dados gerados, foi utilizado o aplicativo Signal Tap Analyser do Quartus®. Foram feitas análises considerando as representações de 32 e 64 bits, usando os mesmos parâmetros da etapa de simulação. O perfil de utilização da FPGA pode ser visto na Tabela II, em que é apresentado o número absoluto de unidades para quatro tipos de recursos em termos de m, além do percentual de utilização desses com relação à quantidade disponível na FPGA. Observa-se que os recursos empregados não crescem proporcionalmente com m. Isso permite um aumento da taxa em bits/s do PRGN (dada por (BITS/AMOSTRA)×FREQUÊNCIA DO CLOCK) sem o acréscimo proporcional dos recursos de hardware e, consequentemente, do consumo de energia. Portanto, o circuito apresentado é menos complexo e mais eficiente energeticamente por bit gerado para $m = 64$. Como esperado, já

TABELA II
RECURSOS EMPREGADOS DA FPGA PARA SÍNTESE DO PRNG.

Recurso	$m = 32$	$m = 64$
Funções Combinacionais (%)	673(< 1)	919(< 1)
Registradores Lógicos Dedicados (%)	901(< 1)	997(< 1)
Multiplicador 9 Bits (%)	6(1)	20(4)
Pinos (%)	10(4)	18(6)

ALGORITMO 1
CÓDIGO VERILOG DO PRNG COM 32 BITS.

```
module PRNG
/* Parameters and wires */
#(parameter[31:0] m = 32,
parameter k = 8,
parameter x_0 = 7,
parameter y_0 = 5 )

(
input wire CLK, SWITCH,
output wire [k-1:0]b
);
/* Auxiliary variables and registers*/
integer MOD_q=2**m;
reg [m-1:0]rx;
reg [m-1:0]ry;
reg [m-1:0]rz;
/* Implementation of Arnold's map */
always @ (posedge CLK) begin
 if (SWITCH) begin
  rx <= (2*rx + ry) % MOD_q;
  ry <= (rx + ry) % MOD_q;
  rz <= (rx * ry) % MOD_q;
 end else begin
  rx <= x_0;
  ry <= y_0;
  rz <= (rx * ry)% MOD_q;
 end
end
/* Specification of the outputs */
assign b  = {rz[0], rz[10], rz[3], rz[6],
             rz[9], rz[1], r[2], rz[7]};
endmodule
```

que o circuito replica exatamente a sequência simulada, os resultados obtidos com o NIST para a sequência gerada por simulação e para a obtida pelo circuito são idênticos. Estes resultados são apresentados na Tabela I. A análise lógica, a síntese e a implementação do PRNG empregando o mapa da Arnold também foram gerados com o Vivado®, o ambiente de desenvolvimento de projeto da Xilinx. Foi selecionado o dispositivo XC7S100 da família SPARTAN. Neste caso, é possível comparar a complexidade do circuito gerado com a de outras proposta que utilizam a mesma plataforma [13]. O emprego de aritmética em ponto fixo levou a uma redução de pelo menos três vezes no número de LUT's, cinco vezes no número de Slices e a aproximadamente o mesmo número de DSP's. Foram comparados o mapa de Arnold (aritmética em ponto fixo com 64 bits) com os mapas de Bernoulli, tent e zigzag (aritmética em ponto flutuante com 64 bits).

V. CONCLUSÕES

Neste trabalho foi proposto um mapa unidimensional baseado no mapa de Arnold discreto sobre o anel de inteiros $\mathbb{Z}_{2^m}$. As propriedades de período das sequências geradas pelo mapa proposto, denominadas sequências-z, foram analisadas com a utilização de propriedades das sequências de Fibonnaci módulo 2^m. Observa-se que as sequências-z possuem um período máximo igual a $T/4$, em que T é o período de Pisano. O mapa proposto foi implementado em FPGA para avaliação dos recursos de hardware necessários para a implementação do PRNG. Como o mapa é definido sobre aneis de inteiros, sua implementação utiliza operações de ponto fixo, requerendo menor complexidade computacional comparado a mapas caóticos definidos sobre os reais, cuja iteração emprega operações do ponto flutuante.

Uma vez estabelecida a análise de período, pode-se escolher valores grandes de m, de tal forma que as sequências geradas nunca se repetem na prática. Também observa-se que as sequências propostas são aprovadas no NIST, mesmo considerando uma taxa de bits por amostra superior à taxa normalmente utilizada na literatura, que é de um bit por amostra. A análise de recursos mostra que para $m = 64$ as quantidades de funções combinacionais e de registradores lógicos não são o dobro das necessárias quando se utiliza $m = 32$. Portanto, o circuito pode ser implementado com valores elevados de m sem, com isso, levar a exaustam dos recursos da FPGA.

REFERÊNCIAS

[1] L. Kocarev and S. Lian, *Chaos-based Cryptography: Theory, Algorithms and Applications*, ser. Studies in Computational Intelligence. Springer, 2011.

[2] A. Beirami and H. Nejati, "A framework for investigating the performance of chaotic-map truly random number generators," *IEEE Trans. Circuits and Syst. II: Exp. Briefs*, vol. 60, no. 7, pp. 446–450, July 2013.

[3] S. Strogatz, *Nonlinear Dynamics and Chaos with Applications to Physics, Biology, Chemistry, and Engineering*, ser. Studies in Nonlinearity Series. Westview Press, 2001.

[4] B. Chirikov and F. Vivaldi, "An algorithmic view of pseudochaos," *Physica D: Nonlinear Phenomena*, vol. 129, no. 3, pp. 223 – 235, May 1999.

[5] L. Kocarev, J. Szczepanski, J. M. Amigo, and I. Tomovski, "Discrete chaos-I: Theory," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 53, no. 6, pp. 1300–1309, June 2006.

[6] B. Yang and X. Liao, "Some properties of the logistic map over the finite field and its application," *Signal Processing*, vol. 153, pp. 231 – 242, Dec. 2018.

[7] L. E. B. III *et al.*, *SP 800-22 Rev. 1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. Gaithersburg, MD, United States: Nat. Inst. Std. & Technol., 2010.

[8] F. Chen, K. Wong, X. Liao, and T. Xiang, "Period distribution of generalized discrete Arnold cat map for $N = p^e$," *IEEE Trans. Inf. Theory*, vol. 58, no. 1, pp. 445–452, Jan 2012.

[9] F. Chen, K. Wong, X. Liao, and T. Xiang, "Period distribution of the generalized discrete Arnold cat map for $N = 2^e$," *IEEE Trans. Inf. Theory*, vol. 59, no. 5, pp. 3249–3255, May 2013.

[10] F. J. Dyson and H. Falk, "Period of a discrete cat mapping," *The American Mathematical Monthly*, vol. 99, no. 7, pp. 603–614, Aug. 1992.

[11] D. D. Wall, "Fibonacci series modulo m," *The American Mathematical Monthly*, vol. 67, no. 6, pp. 525–532, June 1960.

[12] G. H. Hardy, E. M. Wright, D. R. Heath-Brown, and J. H. Silverman, *An Introduction to the Theory of Numbers*, 6th ed. Oxford University Press, 2008.

[13] L. G. de la Fraga, E. Torres-Pérez, E. Tlelo-Cuautle, and C. Mancillas-López, "Hardware implementation of pseudo-random number generators based on chaotic maps," *Nonlinear Dynamics*, vol. 90, no. 3, pp. 1661–1670, Nov 2017.