Design de um circuito integrado dedicado de um TRNG baseado em anéis osciladores

Leonardo de Andrade Santos[1], Sibilla Batista da Luz França[1],
[1] Departamento de Engenharia Elétrica, Curitiba, Brasil
leonardandrado@gmail.com

Resumo — A evolução da tecnologia é um passo importante para a criação de sistemas mais rápidos e robustos. Consequentemente, estes sistemas exigem mecanismos de proteção de dados mais sofisticados. Números aleatórios são extremamente importantes em algoritmos criptográficos, uma vez que são peça-chave para o seu funcionamento, sendo utilizados em chaves confidenciais, vetores de inicialização e valores de preenchimento. Devido a ampla utilização em algoritmos de criptografia, diferentes abordagens para o projeto de geradores de números verdadeiramente aleatórios são pesquisadas. Este artigo apresenta o projeto de um circuito integrado dedicado, na tecnologia de 130 nm, de um gerador de numeros verdadeiramente aleatórios baseado em anéis osciladores, desenvolvido em um trabalho de pesquisa anterior. Este gerador é composto por 20 anéis osciladores (quatro em cada estágio), cada um com 3 portas inversoras, operando a uma frequência máxima de 200 MHz. O projeto foi inicialmente implementado em FPGA (Field Programmable Gate Array) e então realizado o design do circuito integrado a partir de um código de descrição de hardware, utilizando as ferramentas da Cadence. O circuito resultante contém um total de 53 células lógicas e área de 653 μm^2. O consumo de potência é de aproximadamente 32 μW .

I. INTRODUÇÃO

A crescente demanda por sistemas de computação cada vez mais rápidos e seguros intensificou o uso da criptografia na proteção das informações. Números aleatórios exercem uma função importante em criptografia, são usados em diversos algoritmos criptográficos para geração de chaves confidenciais, vetores de inicialização e valores de preenchimento

Um gerador pode gerar números pseudo-aleatórios (PRNG – *Pseudo Randomic Number Generator*) ou verdadeiramente aleatórios (TRNG – *True Random Number Generator*). A diferença entre eles é que o PRNG cria uma sequência aleatória, porém em seguida começa a repeti-la, tornando-a previsível. Um exemplo de circuito utilizado para construir um PRNG é o Registrador de Deslocamento com Realimentação Linear (LFSR – *Left Feedback Shift Register*), que consiste em vários flip-flops interligados, sendo a entrada do primeiro flip-flop uma função linear de seu estado anterior [1]. O TRNG consiste em um gerador verdadeiramente aleatório. Um exemplo é o TRNG baseado em anéis osciladores, que é formado por um anel composto por um número ímpar de portas inversoras, no qual a fonte de entropia é o *jitter* (atraso entre as portas inversoras).

Este trabalho consiste em realizar o projeto de um circuito integrado dedicado, na tecnologia de 130 nm, do TRNG proposto por [1], o qual faz uso de anéis osciladores com árvores binarias de portas XOR, proposto por [2]. O design do circuito integrado dedicado para o TRNG seguiu o fluxo de projeto VLSI *(Very Large Scale Integration*), utilizando-se uma implementação em VHDL (*VHSIC Hardware Description Language*) já desenvolvida anteriormente. O artigo segue a seguinte organização: na seção 2 apresenta-se os geradores de números pseudo e verdadeiramente aleatórios, na seção 3 a metodologia e os resultados, e por fim, a conclusão.

II. GERADORES DE NÚMEROS ALEATÓRIOS

A crescente evolução da tecnologia gerou uma grande demanda por sistemas criptográficos mais eficientes e seguros, e consequentemente aumentou a necessidade de geradores de números aleatórios mais rápidos e eficientes.

Segundo [3] números aleatórios são, estatisticamente descritas como variáveis aleatórias dentro de uma distribuição uniforme u(0,1), ou seja, o resultado de u depende de fatores aleatórios. Um método comum de avaliar se uma sequência de bits é ou não aleatória é verificar se o número de zeros e uns é próximo de 50% da amostra completa. Apesar de validar uma taxa de bits em uma certa amostragem apenas esse teste não é suficiente para validar a preditividade do sistema. Portanto, a fim de evitar um comportamento premeditado, se faz necessário o estudo dos geradores de números aleatórios. É necessário mencionar que conhecer o processo de geração dos números aleatórios não implica em conhecer o próximo número da sequência, caso contrário a sequência não seria verdadeiramente aleatória. Para validar a sequência de

números gerados são utilizados um conjunto de testes estatísticos chamado NIST Test Suite [4].

A. Geradores de Números Pseudo-Aleatórios

Os PRNGs utilizam algoritmos determinísticos, ou seja, geram uma sequência curta dentro de uma grande quantidade de bits gerados, sendo então pouco seguros para aplicações criptográficas. Se na sequência de um número apresentada na saída de um PRNG for analisada será obtido um valor nulo para a desvio padrão [3].

Um método de PRNG desenvolvido foi o Registrador de deslocamento de realimentação linear (LFSR) ilustrado pela Figura 1, o qual seu polinômio característico é $x^n + x^{n-1} + 1$, onde *n* é o número de flip-flops no circuito [5]. Neste exemplo o polinômio característico do LFSR ilustrado $x^3 + x^2 + 1$.

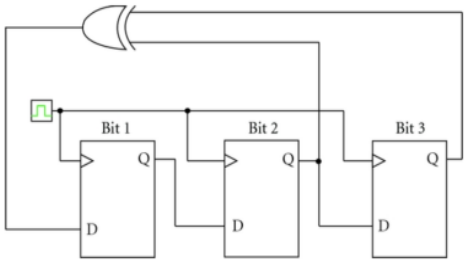

FIGURA 1 – LFSR [5].

Apesar de muitas aplicações possuírem suas demandas de seguranças satisfeitas por PRNGs, determinadas áreas, como a criptografia, exigem um grau de segurança maior. Por conta disto, se vê necessário o estudo e desenvolvimento dos TRNGs uma vez, que seus sistemas são não-determinísticos diferentemente dos PRNGs.

B. Geradores de numeros verdadeiramente randômicos

Os TRNGs não se baseiam em algoritmos determinísticos, e, portanto, são muito mais seguros quando comparados aos PRNGs segundo [1]. São comumente baseados em fenômenos aleatórios de baixo nível, como ruido térmico, efeito fotoelétrico e outros fenômenos quânticos. Esses processos estocásticos são, em teoria, completamente imprevisíveis, ou seja, uma equação que rege tais fenômenos é desconhecida e as afirmações de imprevisibilidade da teoria estão sujeitas a testes experimentais [1]. Dentre as possíveis fontes de aleatoriedade para um TRNG existem o *Delay-Locked Loop* (DLL), *Phase-Locked Loop* (PLL) e o *jitter* (atraso). obtido, por exemplo, a partir de anéis osciladores, que são o tema de estudo deste artigo.

Anéis osciladores são estruturas compostas por um número ímpar de portas inversoras, no qual a saída da última é realimentada na entrada da primeira, conforme ilustrado pela Figura .

FIGURA 2 - ANEL OSCILADOR [2]

Conforme descrito por [2], há dois possíveis estados para a saída do TRNG, um é a saída estável de 0 e 1 e o outro é a saída instável, quando os inversores estão trabalhando. Quando o corre a borda de subida do clock acompanhada do estado instável dos inversores na entrada de dados do flip-flop, a saída será randômica. Segundo [6] TRNGs baseados em anéis osciladores podem apresentar uma boa aleatoriedade sob um longo período de acumulação do *jitter*, porém a taxa de transferência será reduzida e a fonte de consumo de hardware é grande.

Um exemplo de TRNG é o proposto em [2], ilustrado pela Figura 3, operando a uma frequência máxima de 300 MHz.

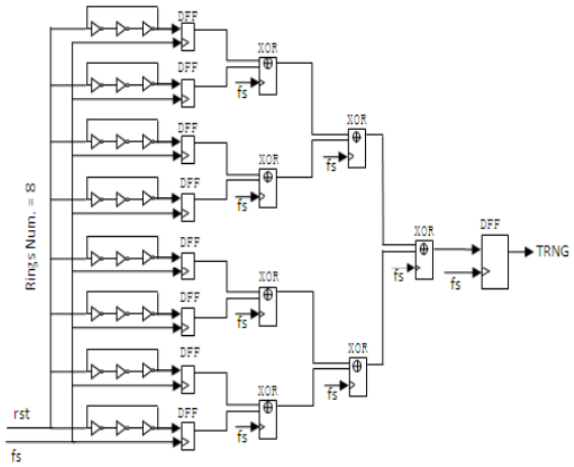

FIGURA 3 - IMPLEMENTAÇÃO PROPOSTA POR [2].

Em [1] também são apresentadas algumas implementações de TRNGs baseados em anéis osciladores. A implementação (2), citada em [1], é composta por apenas 8 anéis osciladores, operando a uma frequência máxima de 200 MHz. Já a implementação (3), apresentada por [1] tem uma arquitetura composta por 20 anéis osciladores, com 3 portas inversoras em cada anel, conforme ilustrado na Figura 4, funcionando em uma frequência de 550 MHz.

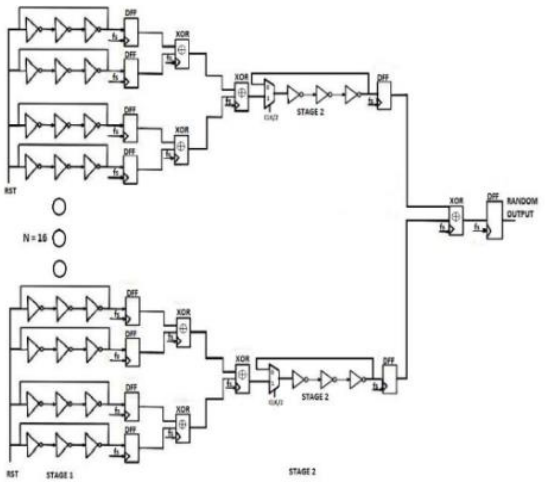

FIGURA 4 – IMPLEMENTAÇÃO 3 DE [1]

III. DESENVOLVIMENTO E RESULTADOS

Este trabalho consistiu no projeto de um circuito integrado de um TRNG desenvolvido em um trabalho de pesquisa anterior, implementado e validado em FPGA(*Field Programmable Gate Array)* [1]. Este TRNG é baseado em anéis osciladores, no qual, a fonte de entropia está nos atrasos das portas inversoras. O projeto foi desenvolvido para a tecnologia de 130 nm, utilizando as ferramentas da Cadence. O fluxo de projeto e verificação de um Circuito Integrado de aplicação especifica (ASIC) pode ser descrito em 5 passos [7].

1. Simulação comportamental: etapa que consiste em verificar se o circuito apresenta a resposta desejada, sem os atrasos da tecnologia. Para essa etapa é utilizada a ferramenta NClaunch;
2. Síntese logica: etapa que consiste na geração da *netlist* do projeto com as portas lógicas da tecnologia de 130 nm, para isso é utilizada a ferramenta do Genus;
3. Simulação pós-síntese: Nessa etapa é verificado se o circuito ainda apresenta o resultado esperado mesmo com os atrasos da tecnologia.
4. Place and Route (PAR): Nessa etapa é gerado o layout do circuito integrado, onde é realizado toda a parte de disposição das células padrão da tecnologia e das trilhas do circuito. Para essa etapa é utilizada a ferramenta Innovus,
5. Simulação pós-PAR: Nessa etapa e feita a última validação do circuito, verificando se os resultados obtidos estão de acordo com os esperados, mesmos com os atrasos inseridos.

O circuito integrado projetado na tecnologia BiCMOS 8HP 130 nm (V_{dd} = 1,2 V), seguiu o fluxo de projeto descrito anteriormente. As informações sobre o número de células lógicas, área e consumo de potência são apresentados na Tabela 1. O layout final do circuito é mostrado na Figura 5.

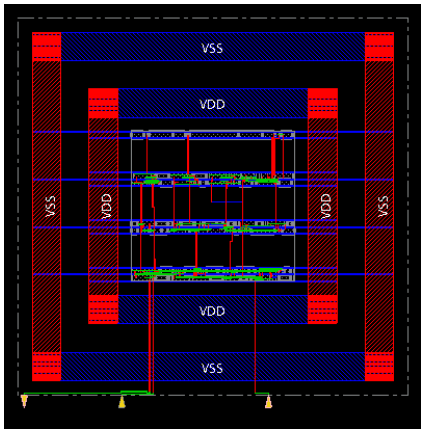

FIGURA 5 – LAYOUT DO GERADOR

TABELA 1. INFORMAÇÕES SOBRE A SÍNTESE DO TRNG

Potência total consumida(µW)	32,412
Área total (µm^2)	653,760
Número total de células	53

As sequências aleatórias obtidas pelo gerador foram validadas pelo conjunto de testes estatísticos NIST, garantindo que o gerador produz números verdadeiramente aleatórios.

IV. CONCLUSÃO

Este trabalho apresentou o projeto de um circuito integrado de um TRNG, baseado em anéis osciladores, desenvolvido em um trabalho de pesquisa anterior. O circuito foi projetado na tecnologia BiCMOS 8HP 130 nm O design do circuito integrado foi realizado a partir da descrição VHDL do gerador e então seguido o fluxo de projeto VLSI apresentado. Foram utilizadas um total de 53 células lógicas e a área do circuito foi de 653 µm^2. O consumo de potência foi de aproximadamente 32 µW.

REFERÊNCIAS

[1] E. Rodrigues De Lima e S. Batista Da Luz França, “Gerador de Números Verdadeiramente Aleatórios Implementado em FPGA”, 2019.

[2] X. Xu e Y. Wang, “High speed true random number generator based on FPGA”, em *Proceedings - 2016 International Conference on Information Systems Engineering, ICISE 2016*, jun. 2016, p. 18–21. doi: 10.1109/ICISE.2016.14.

[3] C. Dutang e D. Wuertz, “A note on random number generation”, 2009. [Online]. Available: http://www-cs-faculty.stanford.edu/

[4] “Random Bit Generation | CSRC”. https://csrc.nist.gov/projects/random-bit-generation/documentation-and-software (acessado ago. 11, 2022).

[5] V. Pedroni, *Eletrônica Digital e VHDL* . 2010.

[6] J. Cui *et al.*, “Design of True Random Number Generator Based on Multi-Stage Feedback Ring Oscillator”, *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, nº 3, p. 1752–1756, mar. 2022, doi: 10.1109/TCSII.2021.3111049.

[7] W. Wolf, “Modern VLSI Design: IP-Based Design, Fourth Edition”.