

Tempo de transição vs bounces em relés eletromecânicos e a geração de *seeds* randômicos

Carolina Zambelli; João Ranhel

Resumo— Mostramos neste artigo que o tempo na mudança de estado em relés eletromecânicos pode ser um bom gerador de valores aleatórios. Usamos tal fenômeno para gerar *seeds* para *linear feedback shift register* (LFSR). Criamos em FPGA um circuito derivado do LFSR que gera 10^6 vetores por segundo com sequência não linear. Testes estatísticos revelam confiança na aleatoriedade dos valores gerados. Conclui-se que um relé de baixo custo e um circuito digital podem ser bons geradores de vetores binários aleatórios. Surpreendentemente, o tempo na transição entre estados é mais aleatório que o número de *bounces* (repiques) nos contatos do relé.

Palavras-Chave— vetores aleatórios, FPGA, LFSR, relés eletromecânicos, repiques, bounces, tempo de transição de estado.

Abstract— In this article we show that the time for changing states in electromechanical relays can generate random values. We use this phenomenon to generate seeds for linear feedback shift registers (LFSR). We created in FPGA a circuit derived from LFSR that generates 10^6 binary vectors per second with non-linear sequence. Statistical tests reveal confidence in the randomness of the generated values. We conclude that a low-cost relay and a digital circuit can become a good random vector generator. Surprisingly, the transition time in between relay states is more random than the number of bounces in the relay contacts.

Keywords— random vectors, FPGA, LFSR, electromechanical relays, bounce.

I. INTRODUÇÃO

A obtenção de números randômicos é importante em várias áreas da comunicação digital, para simulações como Monte Carlo, para computação evolutiva, geração de ruído em redes neurais; além de números randômicos serem cruciais para criptografia, geração de senhas, dentre outras utilizações. Quase consenso é que computadores só conseguem gerar números pseudorrandômicos. A obtenção de valores realmente aleatórios em geral é associada a algum fenômeno físico, como um gerador quântico, que utiliza o fenômeno de entrelaçamento de átomos [1], lançamento de dados ou moedas, dentre outros.

Geração de valores pseudorrandômicos em computadores costumam usar como base LFSR (*linear feedback shift register*), um circuito eletrônico de deslocamento de bits retroalimentado por portas XOR/XNOR em pontos chave do registrador. Também é usual executar em software algoritmos que aplique “*bitwise operation*” em vetores obtendo o mesmo resultado que nos LFSR. A escolha dos pontos de aplicação de feedback no LFSR pode garantir a geração do número máximo de saídas sem que o circuito repita a mesma sequência binária, que pode ser calculado como $2^L - 1$ valores, sendo L o número de bits deslocados ou de flip-flops no circuito. Ou seja, para polinômios de realimentação primitivos, LFSR são máquinas de estados finitos que passam por $2^L - 1$ estados sem repetição, gerando uma

grande quantidade de valores binários com distribuição uniforme. Porém, uma característica dos LFSR é que não podem passar pelos estados com todos os bits ‘0’, no caso dos feedbacks com XOR, ou com todos os bits ‘1’, no caso do feedback com XNOR. Isso exige que o LFSR inicie com um valor qualquer exceto as condições acima. Esse valor inicial é chamado “*seed*”.

Os circuitos LFSR são amplamente usados como geradores pseudoaleatórios [2],[3]; contudo, devemos garantir que um observador externo seja incapaz de descobrir o valor usado como *seed*; caso contrário é possível descobrir a sequência a ser gerada [2],[4]. Outro fator importante é garantir um deslocamento dos bits em sequência não detectável, de forma não-linear. Garantir esses dois fatores são básicos para que um LFSR possa gerar valores pseudoaleatórios de vetores binários.

Nossa pesquisa descreve um circuito eletrônico de baixa complexidade que explora um fenômeno físico capaz de gerar valores randômicos para *seeds* de LFSRs e outras aplicações. A exploração de um fenômeno físico na geração do *seed* torna imprevisível como o LFSR é iniciado, e funciona como *flip* de moedas ou lançamento de dados honestos.

Criamos um circuito que desloca um vetor não-linear indetectáveis em um LFSR. Exploramos a maior sequência binária sem repetições. O circuito é capaz de gerar 10^6 vetores binários por segundo. Realizamos testes estatísticos para avaliar o grau de confiança na aleatoriedade dos valores das *seeds*.

Um dos revisores do artigo alertou para o fato de haver na literatura descrições de circuitos geradores de ruído com base na condução de avalanche em uma junção bipolar (e.g. [5], pp. 982). Esse fenômeno físico é usado para gerar bits aleatórios. Agradecemos pela observação. O circuito em questão parece ser mais simples, contudo tem uma parte analógica, usa vários componentes discretos, e gera bits randômicos sequenciais. Nosso circuito é uma opção para obter um vetor de bits a partir de um relé, de preferência um já existente na placa. Apesar de usarmos um FPGA, nosso circuito é pequeno e cabe em um CPLD (*complex programmable logic device*), bem mais barato, ou em componentes discretos da família 74HCxx.

A. Sobre o artigo

Este artigo está dividido da seguinte forma: na seção II mostramos a descrição teórica que guia a construção dos módulos do circuito gerador de randômicos. Na Seção III mostramos a metodologia utilizada na obtenção dos dados. Na seção IV fazemos uma análise de dados e resultados, enquanto na seção V concluímos e apresentamos uma discussão sobre avanços na pesquisa e trabalhos futuros.

II. FUNDAMENTAÇÃO TEÓRICA

Nossa pesquisa partiu de um fenômeno físico conhecido dos engenheiros eletrônicos: o choque mecânico dos contatos de um relé, de uma tecla, um botão ou uma chave ao ser fechada. No

Carolina, Zambelli, CECS, UFABC, carolina.zambelli@aluno.ufabc.edu.br; João Ranhel, CECS, UFABC, joao.ranhel@ufabc.edu.br.

início o contato está em repouso, no estado “desligado”. Na ligação, o contato móvel ganha energia cinética rumo o outro contato estático, então, subitamente, deve assumir uma nova condição de repouso no estado “ligado”. A dissipação da energia cinética do contato móvel ao chocar com o contato estático gera repiques (*bounces*) dos contatos. Em princípio, pensamos que o número de repiques fosse uma fonte de valores randômicos. De fato, ainda que gere valores com alguma aleatoriedade (ver seção IV), o número de repiques é mais determinístico que outro fenômeno observado. Ao medirmos o tempo de fechamento de vários relés percebemos que os dígitos menos significativos da medida de tempo de fechamento e abertura do relé se comporta de forma mais aleatória que o número de repiques. Ao aumentar a resolução da medida do tempo entre estados observamos que o fenômeno pode ser uma fonte confiável para criação de valores randômicos. Outras tentativas de obtenção de *seeds* randômicos foram descartadas; por exemplo, captar uma sequência de uns e zeros da amostragem de estados dos contatos assim que eles se chocam. Essa alternativa gera cadeias de “1111”s ou de “0000”s totalmente dependentes da frequência da amostragem usada.

A medida de tempo de fechamento e abertura dos contatos dos relés se repete com grande precisão; por exemplo, se um relé tem um tempo de fechamento de 1.5 ms esse valor se repete nesta escala temporal pois depende de características intrínsecas da construção. Entre exemplares de mesmo modelo e marca há alguma variabilidade nessa grandeza mas o tempo é constante no mesmo relé. O tempo de abertura dos relés geralmente é menor que o de fechamento. Explicável, porque para manter o contato fechado o relé é construído tal que uma força mantém um dos contatos normalmente fechado (NF). Essa força deve ser vencida no ligamento, mas no desligamento o retorno envolve apenas a força característica da construção.

Medido na escala temporal uma ou duas ordens de grandeza acima do tempo típico da mudança de estado, nota-se que relés não fecham/abrem seus contatos em tempos constantes. Observando-se apenas dígitos menos significativos da medida obtém-se valores que, estatisticamente, podem ser considerados randômico. Escolhemos esses valores para *seed* de um LFSR. Com o envelhecimento, o tempo de abertura e fechamento dos relés pode mudar um pouco, mas não a aleatoriedade detectada.

A. Definição de um gerador randômico

De acordo L'Ecuyer [6], um gerador é uma estrutura que pode ser definida por: $\mathcal{G} = (\mathcal{S}, s_0, T, U, G)$; onde \mathcal{S} é um conjunto finito de estados, $s_0 \in \mathcal{S}$ é o estado inicial, $T: \mathcal{S} \rightarrow \mathcal{S}$ é uma função de transição, U é um conjunto finito de símbolos de saída, e $G: \mathcal{S} \rightarrow U$ é a função de saída. No gerador em questão, \mathcal{S} é o conjunto de estados de um LFSR ($2^L - 1$ estados, sendo L o número de bits do registrador) e s_0 é o que descrevemos no artigo: o estado inicial (*seed*) do gerador.

A segunda parte da pesquisa consistiu em criar um circuito LFSR que gere em hardware $10^6 - 1$ valores randômicos binários de tal sorte que alguém que analise a sequência, vetor por vetor, não possa determinar qual será o valor seguinte. Ou seja, a função de transição $T: \mathcal{S} \rightarrow \mathcal{S}$ não pode ser identificada analisando-se os vetores de saída U . O circuito tem como saída U vetores binários que podem ser inteiros de k bits; assim, $G: \mathcal{S} \rightarrow U$ é a função que separa k bits do LFSR e expõe na forma de vetor binário. Em um LFSR com deslocamento linear, mesmo com uma $G: \mathcal{S} \rightarrow U$ que cause embaralhamento físico dos bits, alguém com um analisador lógico pode identificar a sequência de

deslocamento e prever quais serão os próximos valores. Um bom gerador deve ter uma função de transição $T: \mathcal{S} \rightarrow \mathcal{S}$ que gere valores de saída imprevisíveis. Devido a restrições de espaço, o artigo se limita a descrever a obtenção do *seed* (s_0). Há várias soluções de construção do LFSR com L bits, sequenciador não-linear e funções distintas que geram vetores U com k bits.

B. Métodos de Avaliação Estatística

A fim de verificar a natureza aleatória do conjunto de saída U , devemos nos certificar de algumas propriedades. A primeira é a uniformidade dos resultados, isto é, garantir que a sequência gerada segue a lei dos grandes números (lembrando que o problema em questão é de natureza discreta) como é o caso de um lançamento de uma moeda não viciada [7]. A segunda é garantir independência, no caso do lançamento de moedas, seria o mesmo que garantir que o resultado de um lançamento, não afetará o próximo lançamento.

A verificação de tais propriedades é feita através de testes estatísticos. Entretanto, a interpretação de seus resultados, e a quantidade de testes necessários para assegurar a qualidade de um gerador randômico, está intimamente ligada à sua aplicação [8],[6]. Como exemplo, para fins de criptografia, uma das aplicações mais rigorosas, é recomendado em [9] um conjunto de 15 testes para garantir a “qualidade” do gerador randômico.

Geradores genuinamente randômicos artificiais são raros e complexos; por isso os geradores pseudoaleatórios são muito utilizados. Como foi dito, para estes geradores, além de se testar a natureza aleatória da sequência gerada (a saída U) é imprescindível garantir não ser possível determinar a *seed*; caso contrário, seria fácil adivinhar o próximo U . Como estamos tratando das *seeds*, ou seja, da parte s_0 do gerador randômico \mathcal{G} , podemos flexibilizar a quantidade de testes estatísticos se comparado com criptografia, por exemplo. Para isso, selecionamos 4 testes: (a) Teste Qui-Quadrado: para análise de frequências (com regras de Freedman-Diaconis e Styrge's para definição do número de *bins*). (b) Teste de Correlação serial (*autocorrelação*): para analisar independência das realizações, dos valores obtidos. (c) Teste de Durbin Watson: também para analisar independência entre valores obtidos. (d) Teste Espectral: para identificação de padrões no domínio da frequência. Os resultados são apresentados na seção IV.

III. PROCESSOS E MÉTODOS

Foram utilizados na pesquisa dois kits FPGA/SoC Cyclone V (DE10-Nano) da Intel/Terasic. Usamos ainda relés das marcas TONGLING modelos /JQC-377-S-Z, SONGLE /SRD-05VDC-SL-C e HONG WEI /JQC3F-05VDC-C. Os relés têm mesma dimensão, formato e parâmetros de operação; e as medidas obtidas diferem minimamente, assim supomos que eles possuem mesma origem. Assim, focamos a pesquisa em 4 módulos de relés JQC-377-S-Z, com 4 relés cada módulo.

Apresentamos a seguir avaliações referentes a 16 relés de mesma marca e modelo. Nossa intenção é medir dispersões em elementos da mesma população. Os relés foram acionados por fonte separada dos kits FPGA. O módulo de relé possui isolador óptico entre o sinal de comando do FPGA e os transistores que alimentam as bobinas, conforme Fig.1. Assim, as saídas do FPGA apenas enviam sinais de comando para os transistores (~2 mA) que acionam os relés; eliminando a influência da carga

representada pelas bobinas e o circuito no FPGA. Os relés foram acionados por fonte de +5V e consomem ~70 mA.

Foram medidos os tempos de fechamento (NF=>NA) e abertura (NA=>NF) dos relés, além do número de repiques nos contatos. Como foi dito, descartamos medidas que pouco contribuíram para o avanço da pesquisa, como amostragem dos repiques. Para medir tempo e número de repiques ligamos os contatos fixos dos relés para +3V3 com resistores pull-up, e o contato móvel comum para terra, conforme Fig.1.

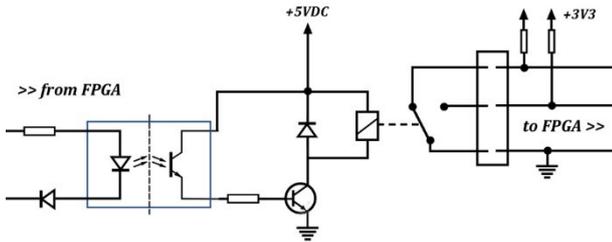


Fig. 1. Setup do experimento. O FPGA comanda ligamento do relé via acoplador óptico. Os relés são alimentados separadamente. As medidas são feitas nos contatos sem carga, com resistores pull-up ligados ao FPGA.

Alguns circuitos foram sintetizados no “fabric” do FPGA para tomar medidas com precisão. Os circuitos são paralelos e independentes. Os resultados foram acessados via barramento Avalon pelo processador ARM embarcado no SoC Cyclone V. O ARM roda um programa criado em Qt sobre Linux (distro Xfce), que comanda os circuitos e lê os resultados de um buffer de memória interna do FPGA.

Para os 16 relés, coletamos 1000 leituras do (a) tempo de ligamento e (b) de desligamento, (c) número de repiques no ligamento e (d) no desligamento. A fim de estabilizar os relés, usamos intervalos de 1500 ms para ligar e desligar os relés antes de um novo ciclo de medição. Tempos maiores, e.g. 4 segundos entre ciclos, não influenciam; embora tempos mais curtos que 600 ms causem interferências nas medidas, provavelmente porque os relés estão ainda vibrando.

Nas medidas de tempo usamos o seguinte procedimento: (a) os contadores de tempo são zerados e o relé é ligado; (b) no ligamento, o contato NF está fechado, então o circuito digital lê um nível ‘0’ porque o NF está conectado ao contato comum que está ligado à terra; (c) assim que o contato NF “descola”, o sinal NF torna-se ‘1’ e o circuito dispara o timer cujo clock é de 100 MHz; (d) enquanto o contato móvel percorre o caminho entre NF e NA os sinais dos contatos NA e NF ficam ambos em nível ‘1’; (e) assim que NA torna-se ‘0’ significa que o contato móvel atingiu o alvo, então o contador é parado. A tomada de tempo no desligamento é idêntica, invertendo-se NA e NF.

A medição do número de repiques foi feita dentro de uma janela de até 200 ms a partir do primeiro contato; ou seja, o circuito conta quantas vezes ocorreu uma borda de descida dos sinais NA (ou NF) dentro de um intervalo de 200 ms. A borda de descida é devido ao fato de que cada vez que os contatos se tocam o sinal é levado para nível ‘0’ por ter tocado o terminal comum. Foram usados registradores de 32 bits nas medidas de tempos de mudança de estado e repiques. Para cada valor lido o programa aplica um filtro ($V=V \& 0xFF$;) em C, o que significa extrair os 8 bits menos significativos da medida em binário.

IV. ANÁLISES E RESULTADOS

Primeiramente, vamos descrever por que desistimos do uso do número de repiques como gerador de seed. A Fig.2 mostra o boxplot obtido em 1000 leituras em 16 relés, alimentados e comandados da mesma forma. A parte superior mostra o número de repiques no ligamento e a inferior no desligamento. Tanto no ligamento quanto no desligamento observamos uma alta dispersão nos dados, evidenciado pelo intervalo interquartil, além da intensa presença de possíveis outliers (observações acima da cauda de distribuição). A tabela II no apêndice mostra os dados resumo obtidos no experimento.

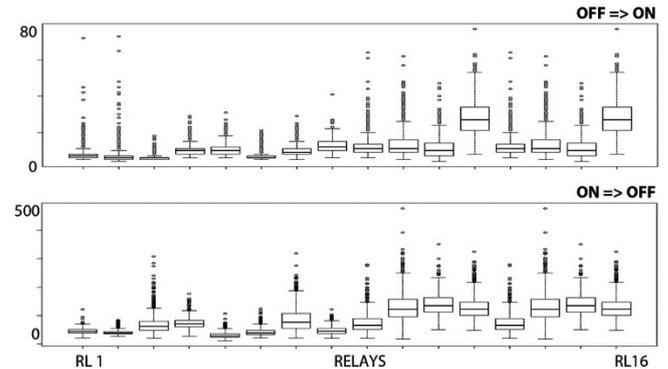


Fig. 2. Boxplot com distribuição do número de bounces de 16 relés, 1000 leituras. Há grande variação entre repiques obtidos entre relés.

No primeiro teste estatístico realizado revelou-se a impossibilidade do uso de repiques. A Fig.3 mostra um gráfico típico de autocorrelação entre as medidas de repiques.

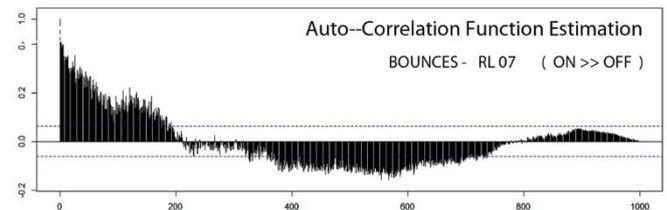


Fig. 3. Gráfico de autocorrelação do número de repiques no relé RL07 no desligamento, sendo o eixo x o número de lags e no y o valor de autocorrelação. A alta correlação entre um valor e outro indica que uma medida pode ter correlação com um evento anterior.

A medida de autocorrelação, $[-1,1]$, mostra quanto uma medida num tempo ‘t’ pode ser explicada por uma medida em $t-x$, sendo x o atraso ou lag. No caso dos repiques, podemos observar que para valores de lag de 1 até 200 (eixo x), existe autocorrelação (eixo y) significativa, que são os valores de acima da linha tracejada; linha esta que, por padrão, representa 5%. Dessa forma, descartamos o uso de repiques como seeds.

Por outro lado, a Fig.4 mostra o boxplot com a distribuição de 1000 valores de leitura do tempo de fechamento/abertura com apenas com os 8 bits menos significativos da medida. A parte superior mostra o valor extraído do tempo de ligamento e a parte inferior mostra valores obtidos no desligamento. Note na distribuição dos dados binários que qualquer relé pode gerar do número 0x00 até 0xFF. Diferente da Figura 2, neste caso não existem ocorrências de outliers e nem alta dispersão dos dados. No tempo de fechamento e abertura a distribuição é simétrica.

Um histograma das medidas de tempo de fechamento e abertura já filtradas pode ser visto na Fig.5. Note que as alturas das barras (quantidade de observações em cada categoria) estão muito próximas, evidenciando uma distribuição uniforme, o que é bom sinal para um valor randômico porque a probabilidade de se escolher uma *seed* no intervalo de 0x00 até 0xFF é proporcional ao tamanho deste intervalo.

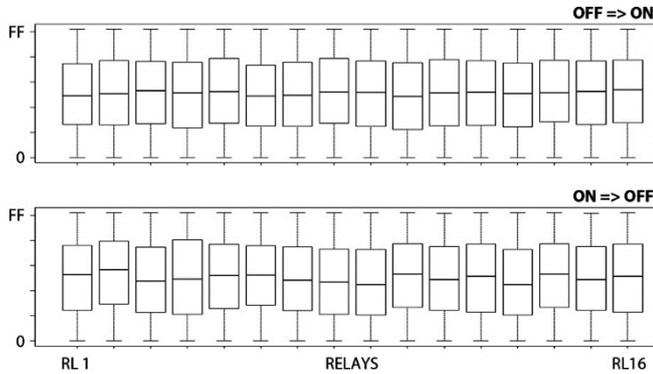


Fig. 4. Boxplot com a distribuição de medidas filtradas (0x0 até 0xFF) de 16 relés, 1000 leituras. Note-se a homogeneidade da distribuição.

Para *seeds* de 16, 32, 64 bits tomamos mais de uma leitura e deslocamos 8 bits lidos em cada leitura. A tabela I no apêndice mostra os dados resumo com medidas de tempo obtidas no ligamento (OFF=>ON) e desligamento (ON=>OFF).

Para validar uniformidade, além da visualização gráfica, foi aplicado o teste qui-quadrado usando duas técnicas para definição do número de *bins*. Para cada relé calculamos dois valores qui-quadrado, comparando os resultados ao valor crítico.

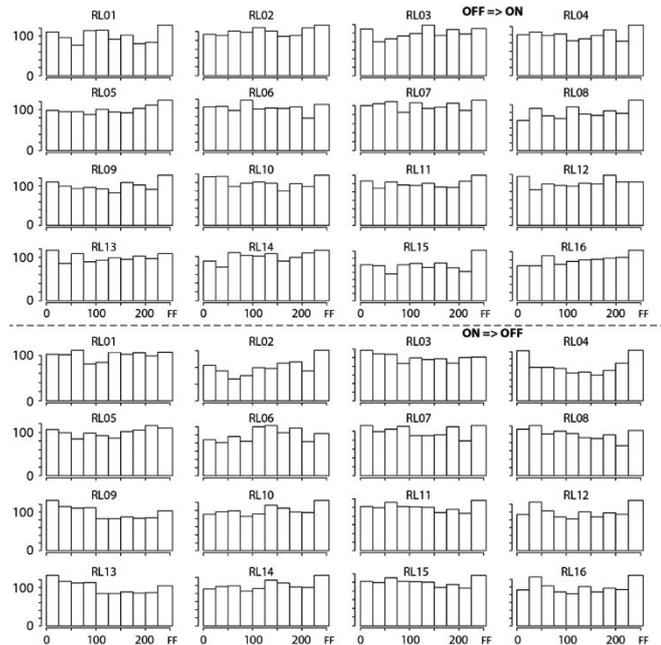


Fig. 5. Histograma com distribuição dos valores obtidos (0x0 até 0xFF) de 16 relés, 1000 leituras, sendo o eixo x os bins do histograma e o eixo y a quantidade absoluta. Note-se a distribuição \approx uniforme.

Pela regra de Freedman-Diaconis apenas dois experimentos falharam com diferença apenas na segunda casa decimal. Assim,

os resultados dos testes são positivos o que evidencia um comportamento uniforme do fenômeno.

A Fig.6 mostra um gráfico de autocorrelação entre medidas filtradas obtidas no ligamento e desligamento do relé RL07, que foi o pior caso que obtivemos.

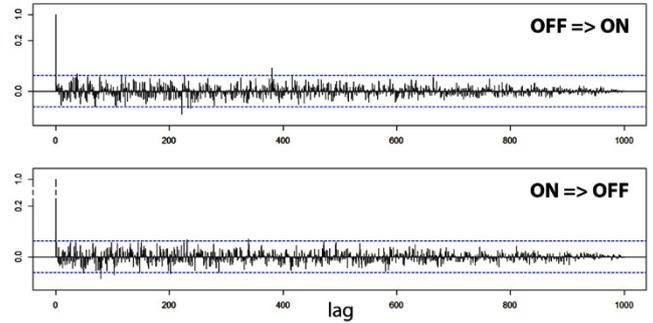


Fig. 6. Gráfico de autocorrelação do relé #7. Note o baixo indicador de correlação. Outros relés têm curvas semelhantes ou melhores que estas.

Nas medidas de tempo de transição entre estados, todos os relés apresentam curvas semelhantes à da Fig.6, indicando baixo grau de correlação entre um valor e a medida anterior. Note que diferente dos resultados da Fig. 3, os valores de correlação estão abaixo do tracejado, indicando autocorrelação estatisticamente insignificante para os valores de lag entre 0 a 1000 [10] [4].

Para complementar a análise de autocorrelação com um teste paramétrico, foi aplicado o teste de Durbin-Watson em todas as medidas de tempo de mudança de estados. O resultado deste teste pode variar de -4 a $+4$, sendo os valores próximos de 2 evidências de autocorrelação nula. Nos testes que realizamos, todos os resultados ficaram dentro do intervalo $[1.88, 2.05]$, evidenciando correlação nula, complementado os resultados da análise gráfica da Fig. 6.

Por fim, realizamos análise no domínio da frequência através da Transformada Discreta de Fourier, que é um método não paramétrico para estimação de espectro [11].

O objetivo é detectar padrões repetitivos com certo grau de proximidade que, se observados, indicam não aleatoriedade [9]. A Fig. 7 mostra gráficos espectrais de dois relés nos quais se vê que nenhum padrão discriminante pode ser destacado. Gráficos similares foram obtidos para todos os relés.

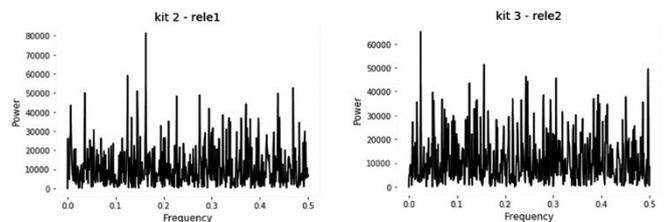


Fig. 7. Gráfico de espectro para relé 1 do kit 2; e relé 2 do kit 3.

V. CONCLUSÕES

Em relés eletromecânicos, o tempo de mudança de estado dos contatos é aproximadamente constante; porém, quando se aumenta a resolução do medidor de tempo nota-se que há grande variação nos valores dos algoritmos menos significativos da

medida. Surpreendente é que esses valores são mais aleatórios que os repiques (bounces) dos contatos dos relés. Assim, filtrando-se os valores menos significativos dos tempos de fechamento e abertura dos relés obtivemos um gerador de valores randômicos que usamos como *seed* para um circuito LFSR. Os LFSR são conhecidos e muito usados para geração de números pseudorrandômicos. Criamos um circuito em FPGA que, além de alimentar o circuito com valores obtidos por meio da técnica descrita no artigo, o circuito possui ainda dois mecanismos para garantir aleatoriedade dos valores gerados: (a) em intervalos indeterminados o circuito troca o *seed*, e (b) o circuito realiza uma sequência de deslocamentos não linear e não identificável no vetor de saída binário randômico.

Os valores obtidos para a geração da *seed*, foram submetidos a testes estatísticos para certificação quanto a aleatoriedade. Os resultados obtidos se mostraram satisfatórios. Os testes de aleatoriedade da sequência de vetores gerada pelo circuito LFSR não foram incluídos neste trabalho por falta de espaço. O foco do artigo é mostrar que um fenômeno físico observado em circuitos relativamente simples pode funcionar como gerador de valores aleatórios (*seeds*) confiáveis.

REFERÊNCIAS

- [1] S. Pironio *et al.*, “Random numbers certified by Bell’s theorem,” *Nature*, vol. 464, no. 7291, pp. 1021–1024, 2010.
- [2] T. Tuncer and E. Avaroğlu, “Random number generation with LFSR based stream cipher algorithms,” in *2017 40th Intern. Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2017 - Proceedings*, 2017, pp. 171–175.
- [3] M. Sahithi, B. Muralikrishna, M. Jyothi, K. Purnima, A. J. Rani, and N. N. Sudha, “Implementation of Random Number Generator Using LFSR for High Secured Multi Purpose Applications,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 3, no. 1, pp. 3287–3290, 2012.
- [4] P. L’Ecuyer, “Testing random number generators,” in *Proceedings of the 24th conference on Winter simulation - WSC ’92*, 1992, pp. 305–313.
- [5] Horowitz Paul and Hill Winfield, *A Arte da Eletrônica: Circuitos Eletrônicos e Microeletrônica*, 3a. ed, Porto Alegre: Bookman, 2017.
- [6] P. L’Ecuyer, “Uniform random number generation,” *Ann. Oper. Res.*, vol. 53, no. 1, pp. 77–120, Dec. 1994.
- [7] P. Martin-Löf, “The definition of random sequences,” *Inf. Control*, vol. 9, no. 6, pp. 602–619, Dec. 1966.
- [8] Adam Goucher, Tim Riley, “Beautiful Testing: Leading Professionals Reveal How They Improve Software”, O’Reilly, 2010.
- [9] A. Rukhin *et al.*, “A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications,” National Institute of Standards and Technology, Gaithersburg, MD, Apr. 2010.
- [10] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice - Rob J Hyndman, George Athanasopoulos - Google Livros*, 2nd ed., vol. 1. OTexts - Online Open-Access Textbooks, 2018.
- [11] M. Pastell, *Measurements and Data Analysis for Agricultural Engineers using Python*. Helsinki, Finland: Leanpub.com, 2016.

APÊNDICE

As tabelas mostram as medidas-resumo do experimento. A Tabela I mostra valores dos tempos de mudança de estado dos contatos (x100), ou seja, de fechamento e abertura dos relés.

TABELA I. VALORES FILTRADOS: P/ 1000 MEDIÇÕES, EM HEXADECIMAL (SENDO A MEDIDA ORIGINAL X100 MS), MEDIANA (X100 MD) E MÉDIA (X100 MN) OBTIDOS NO LIGAMENTO E DESLIGAMENTO DOS 16 RELÉS.

ID Relé	OFF => ON (intervalo [0,FF])		ON => OFF (intervalo [0,FF])	
	md	mn	md	mn
1	123.0	126.5	132.0	126.9
2	126.5	128.5	142.0	135.4
3	133.0	129.3	119.0	122.4
4	128.5	126.7	123.0	125.7
5	131.0	130.5	130.0	128.4
6	122.5	123.7	131.0	129.8
7	124.0	125.6	121.0	123.7
8	130.5	131.6	117.0	120.2
9	129.5	128.1	112.5	117.5
10	121.5	123.5	133.0	130.1
11	128.5	127.7	122.0	124.4
12	130.0	127.5	128.5	126.6
13	127.0	125.7	112.5	117.5
14	129.0	130.3	133.0	130.1
15	131.5	130.1	122.0	124.4
16	135.0	131.9	128.5	126.6

A Tabela II mostra o número de repiques medidos no fechamento e na abertura dos contatos dos relés.

TABELA II. CONTAGEM DE REPIQUES: MÍN (<), MÁX (>), MEDIANA (MD) E MÉDIA (MN) OBTIDOS NO LIGAMENTO E DESLIGAMENTO DOS RELÉS.

ID Relé	OFF => ON				ON => OFF			
	<	md	mn	>	<	md	mn	>
1	5	7	7.9	72	21	44	44.9	122
2	4	6	7.3	73	26	38	40.2	84
3	5	6	5.9	18	20	62	70.3	308
4	6	10	10.2	29	26	71	73.3	177
5	6	10	10.5	31	11	29	30.3	107
6	5	6	7.0	21	22	39	42.4	124
7	5	9	9.8	29	21	76	85.9	319
8	6	12	12.8	41	20	45	47.2	122
9	6	11	12.1	64	20	66	73.4	279
10	5	11	13.7	62	17	122	131.1	477
11	4	10	11.3	47	50	135	139.8	351
12	8	27	28.1	77	48	122	128.5	325
13	6	11	12.1	64	20	66	73.4	279
14	5	11	13.7	62	17	122	131.1	477
15	4	10	11.3	47	50	135	139.8	351
16	8	27	28.1	77	48	122	128.5	325