



GERADOR DE SINAIS PSEUDOALEATÓRIOS UTILIZANDO FPGA COM MONITORAMENTO EM TEMPO REAL POR MEIO DO QUARTUS II

Felipe Batista Faro Pinto*¹, Pedro Arthur Bessa Leão¹, Rafael Vinicius Tayette da Nobrega¹, Otávio de Souza Martins Gomes², João Batista Rosolem³, Ulysses Rondina Duarte¹

¹Área Acadêmica de Engenharia Elétrica do Instituto Federal de Minas Gerais, Campus Formiga - IFMG

²Universidade Federal de Itajubá - UNIFEI

³Fundação Centro de Pesquisa e Desenvolvimento em Telecomunicações - CPqD

Resumo - O objetivo deste trabalho é apresentar um tutorial de confecção de um circuito digital gerador de sinal pseudoaleatório no *software* Quartus II, com o uso da *Field Programmable Gate Array* (FPGA), visando a aplicação em comunicações digitais. Para a geração de um sinal *Pseudo Random Binary Signals* (PRBS), foi implementado um diagrama de blocos baseado em *Linear-feedback shift register* (LFSR). Realizou-se a montagem de um *loopback*, ou seja, o sinal gerado foi enviado de um conector SMA da placa FPGA para outro SMA. Utilizando-se da ferramenta *SignalTap II*, presente no Quartus II, foi possível monitorar em tempo real tanto o sinal transmitido quanto o recebido. Para validar o gerador de PRBS, outras configurações de montagem foram elaboradas. Um osciloscópio foi configurado para receber o sinal pseudoaleatório. Através da forma de onda, verificou-se a aleatoriedade do sinal recebido. Para uma possível aplicação em comunicações digitais, utilizou-se dois conectores atenuadores de 3 e 6 dB. Os valores atenuados foram próximos dos esperados.

Palavras-Chave – Eletrônica Digital; FPGA; Quartus II; PRBS.

PSEUDORANDOM SIGNAL GENERATOR USING FPGA WITH REAL-TIME MONITORING THROUGH QUARTUS II

Abstract - The objective of this work is to present a tutorial on how to build a pseudorandom digital signal generator circuit in Quartus II software, using the Field Programmable Gates Matrix (FPGA), administrative to determined in digital communications. To generate a Pseudo-Random Binary Signals (PRBS) signal, a block diagram based on a Linear-feedback shift register (LFSR) was implemented. A loopback was assembled, that is, the generated signal was sent from an SMA connector of the FPGA board to another SMA. Using the SignalTap II tool, present in Quartus II, it was possible to monitor both the

transmitted signal and the pressure in real-time. To validate the PRBS generator, other assembly configurations were elaborated. An oscilloscope is designed to receive the pseudorandom signal. Through the waveform, the randomness of the charged signal was verified. For a possible application in digital communications, use the two 3 and 6 dB attenuator connectors. The attenuated values were the next expected.

Keywords - Digital Electronics; FPGA; Quartus II; PRBS.

NOMENCLATURAS

FPGA *Field Programmable Gate Array*

SMA *SubMiniature version A*

VHSIC *Very High Speed Integrated Circuits*

VHDL *VHSIC Hardware Description Language*

PRBS *Pseudo Random Binary Signals*

LFSR *Linear-feedback shift register*

I. INTRODUÇÃO

A necessidade da comunicação a longas distâncias, serviços de alta qualidade de áudio e vídeo, aumento massivo de usuários, são algumas das motivações que convergem nas evoluções dos sistemas de telecomunicações [1]. De forma básica, uma rede de telecomunicações abrangem os seguintes componentes: um transmissor para enviar os dados através de sinais, que podem ser de natureza elétrica ou óptica; um meio de transmissão para que o sinal enviando pelo transmissor trafegue até o seu destino; e por fim um receptor, o qual recebe os dados do transmissor. Além desses três componentes, existem os sistemas eletrônicos que são imprescindíveis para o devido tratamento nos sinais, como a modulação e codificação, do lado do transmissor, e a demodulação e decodificação, no receptor.

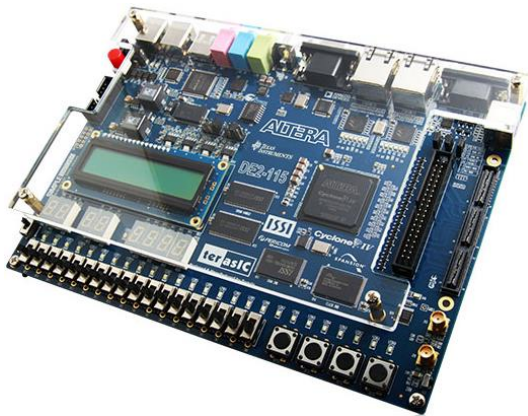
*lipe.pinto97@gmail.com

O uso de microcontroladores que permitem a geração de sinais, processamento de informações, e controle de múltiplos sensores são bastante recorrentes nos projetos de telecomunicações. Uma das plataformas mais utilizadas é o Arduino. Este que utiliza o microcontrolador Atmel, e, por ser uma plataforma acessível e de fácil programação, é muito utilizado em projetos envolvendo sensores e também automação residencial [2]. Outro microcontrolador bastante utilizado é o ESP32, que vem se tornando uma opção atrativa de compra, pois já possui módulo de comunicação Wi-Fi embutido, processador *dual core* e também o sistema *Bluetooth* [3], tudo isso com um custo próximo de um modelo mais básico do Arduino.

No entanto, para aplicações mais críticas na área de telecomunicações, onde há a necessidade do processamento de dados em elevadas taxas (ordem de MHz), além de interfaces que facilitem a transmissão de sinais na frequência de rádio, como a interface a SMA, estes microcontroladores citados não atendem a estas aplicações.

Visando essas aplicações, existem outras plataformas eletrônicas que podem ser integradas em sistema de telecomunicações. Um chip bastante utilizado em projetos envolvendo programação mais avançada, é a *Field Programmable Gate Array* (FPGA), apresentada na Figura 1, que nada mais é do que um hardware que permite ao usuário criar diferentes arranjos lógicos, de modo a realizar uma determinada tarefa [4].

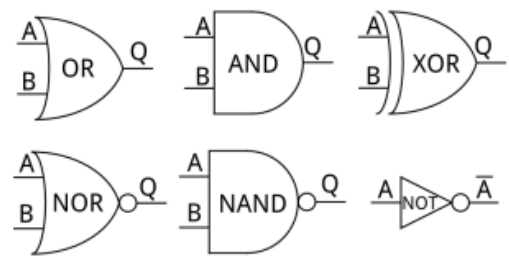
Figura 1: Placa FPGA Altera DE2-115 [5].



Na Figura 1 é apresentado o modelo da Altera DE2-115, e algumas vantagens que esta placa apresenta em comparação com os microcontroladores são: Versatilidade em criar diferentes lógicas em sua matriz; Geração de sinais com frequência de 50 MHz; Interface física diversificada, constituídas de portas SMA, RS-232, USB, entre outras [5].

A programação da FPGA é baseada em diagramas de blocos, onde é possível utilizar de portas lógicas (Figura 2) para gerar um circuito baseado em eletrônica digital. Além de circuito de blocos, há também a linguagem descritiva de hardware que são o VHDL e também o Verilog [6].

Figura 2: Portas lógicas.



A FPGA pode ser aplicada em diversas situações. Em [7], o autor propõe o uso da FPGA para capturar e processar as imagens geradas por um painel de instrumentos digitais, usados em automóveis. Já em [8], a FPGA e a linguagem VHDL fazem parte de um método desenvolvido para uma nova abordagem de ensino direcionada ao aprendizado ativo.

Visando a comunicação digital, um dos circuitos bastante utilizados são os geradores de sinais pseudoaleatórios. Esses tipos de sinais são utilizados como forma de codificar informações [9]. São chamados assim, pois o sinal não é inteiramente aleatório, mas sim com um comprimento finito. Após passar o seu comprimento, a sequência aleatória se repete. Além de ser usado como codificador, um sinal pseudoaleatório pode ser utilizado em testes de validação em redes ópticas, de forma a simular determinada informação requerida por um usuário.

Em [10], um sinal pseudoaleatório é gerado a partir de mapas caóticos. Deste modo, a contribuição se dá com novas estratégias de codificação e pós-processamento. Os autores em [11] desenvolvem um modelo linear para geração de sinal de radiofrequência utilizando *Pseudo Random Binary Signals* (PRBS). O sinal pseudoaleatório é desenvolvido através da FPGA.

Neste trabalho, apresenta-se um trabalho no formato de tutorial, fornecendo os passos e a metodologia necessária para gerar um sinal PRBS de comprimento desejado, além de transmiti-lo e detectá-lo pelas portas SMA da FPGA. Somado a isso, introduz-se a ferramenta *SignalTap II Logic Analyzer* [12], a qual é responsável por monitorar os dados em tempo real.

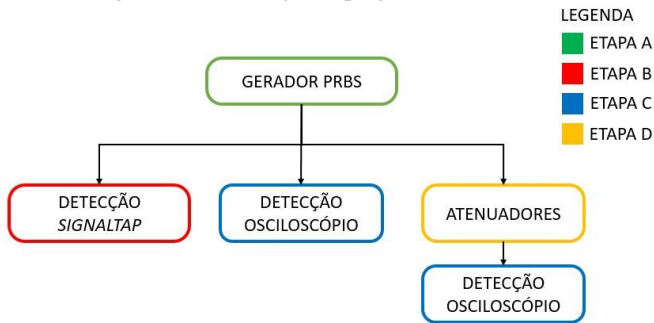
Este trabalho divide-se em: Seção II, onde é apresentada a metodologia empregada para o manuseio do *software* de programação da FPGA, desde a elaboração de um diagrama de blocos, até a implementação no hardware do FPGA. Além disso, apresenta-se os testes práticos elaborados; Na seção III, são expostos os resultados obtidos a partir dos circuitos programados; E na seção IV, as conclusões deste trabalho são discutidas.

II. MATERIAS E MÉTODOS

Este trabalho pode ser dividido em 4 etapas: A primeira etapa é a geração do sinal pseudoaleatório. Nesta etapa será apresentado a lógica de geração do sinal PRBS, assim como a associação dos principais componentes utilizados. Na segunda etapa da metodologia é apresentada a ferramenta *SignalTap II Logic Analyzer* do *software* Quartus, da FPGA, com o objetivo de monitorar o sinal PRBS gerado. Esta ferramenta permite verificar em tempo real as portas da FPGA. Na terceira parte

é mostrada a configuração para medição do sinal PRBS utilizando um osciloscópio. Por fim, na quarta etapa, com o objetivo de aplicar o sinal PRBS gerado em sistemas de comunicações ópticas, são apresentados atenuadores, que reduzem a tensão de saída da FPGA em 3 e 6 dB. A detecção do sinal atenuado é realizada por meio do osciloscópio. Na Figura 3 apresenta-se o fluxograma das 4 etapas citadas.

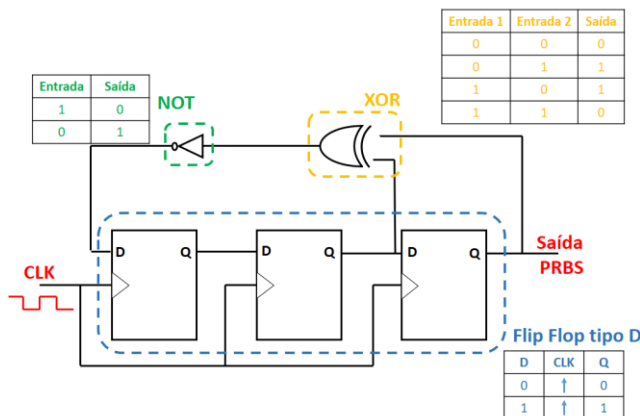
Figura 3: Metodologia empregada neste trabalho.



A. Geração de PRBS

Para geração do Sinal PRBS, utilizou-se Flip-Flop do tipo D, portas lógicas do tipo XOR e NOT. A Figura 4 apresenta, de forma esquemática, a associação destes componentes para geração do sinal. Vale destacar que a utilização das portas lógicas NOT e XOR são utilizadas para gerar aleatoriedade na sequência de bits e a quantidade de flip-flop tipo D são importantes para a quantidade de bits aleatórios gerados. Quanto maior o número de flip-flop, maior a sequência de bits gerados. A proporção entre comprimento da sequência de bits e número de flip-flop é de $2^N - 1$, sendo N o número de Flip [13]. Na Figura 4 também estão apresentadas as informações referentes a Tabela Verdade de cada componente, detalhando a saída das portas lógicas e dos flip flops a partir de suas entradas.

Figura 4: Representação da geração do sinal.

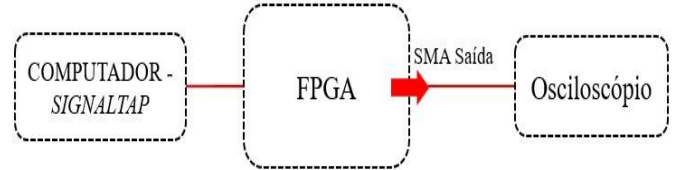


O gerador de sinais pseudoaleatório desenvolvido neste trabalho foi simulado na ferramenta Quartus II e possui 6 Flip-Flops.

B. Ferramenta SignalTap II Logic Analyzer

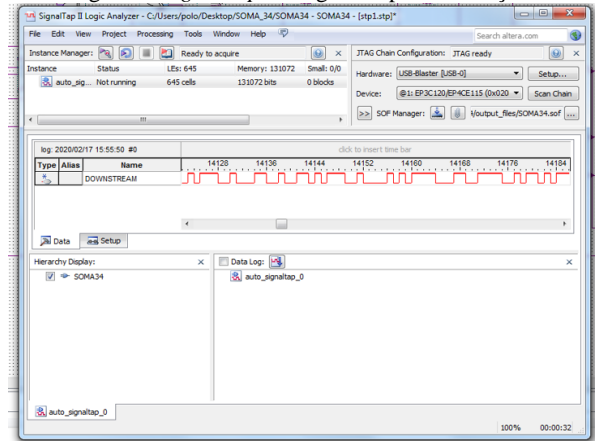
Uma importante ferramenta para monitorar o sinal, em tempo real, da FPGA é *SignalTap II Logic Analyzer*. Esta ferramenta é utilizada neste trabalho para validar a geração do sinal pseudoaleatório. Para utilização dessa ferramenta, a saída da FPGA foi conectada a sua entrada. As portas de entrada e saída são monitoradas pelo PC, utilizando a ferramenta SignalTap do *software* Quartus. A Figura 5 apresenta, de forma esquemática, a configuração experimental.

Figura 5: Configuração experimental para utilização de *SignalTap*.



Também é importante, após acessar a ferramenta *SignalTap*, configurar para que seja possível detectar os sinais. Em [11] estas informações de configuração da ferramenta estão detalhadas. A Figura 6 apresenta a ferramenta configurada para recepção do sinal.

Figura 6: *SignalTap* configurado para utilização.



C. Visualização pelo osciloscópio

Além da utilização da ferramenta *SignalTap* para visualização do sinal gerado, foi utilizado também um osciloscópio. Neste caso, o sinal gerado pela FPGA é visualizado pelo osciloscópio. Esta etapa garante a validação do gerador PRBS, além da detecção realizada pela ferramenta *SignalTap*.

D. Atenuador

Como uma aplicação possível para o gerador de sinais pseudoaleatórios é em sistemas de comunicações ópticas, foram avaliados os níveis de tensão do sinal, por meio do osciloscópio, com atenuadores SMA de 3 e 6 dB. Controlar a amplitude do sinal é importante, permitindo associar este gerador a um *Reflective Semiconductor Optical Amplifier* (RSOA) sem extrapolar os valores de tensão permitidos em

suas portas de entrada. Este dispositivo opera basicamente como amplificadores ópticos semicondutores, provendo altos níveis de ganho óptico [1], logo este componente é muito importante em redes ópticas mais recentes. A Figura 7 apresenta os atenuadores utilizados na validação prática.

Figura 7: Atenuadores SMA.



Para cálculo da atenuação de tensão em dB, utiliza-se a equação 1[14].

$$A = 20 \log (V_{saída}/V_{entrada}) \quad (1)$$

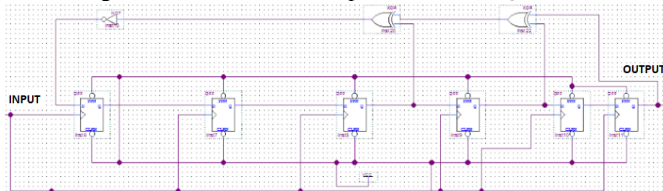
Em que A é a atenuação do sinal, $V_{saída}$ é a tensão de saída do atenuador e $V_{entrada}$ é a tensão de entrada do atenuador.

III. RESULTADOS E DISCUSSÕES

A. Circuito PRBS

Utilizando as informações apresentadas na seção anterior sobre a confecção de um circuito gerador de sinais pseudo aleatórios, elaborou-se um diagrama de blocos no Quartus II com o objetivo de gerar um sequência de 63 bits, e para isso utilizou-se 6 Flip-Flops tipo D, conforme é apresentado na Figura 8.

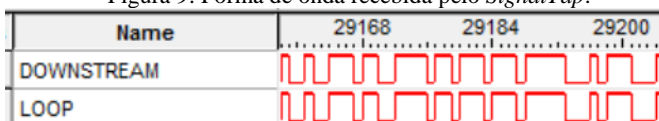
Figura 8: Circuito PRBS implementado no Quartus II.



B. Monitoramento no SignalTap

Após a implementação do circuito digital, o próximo passo foi enviar para a placa da FPGA e utilizar as portas SMA para transmitir e receber de volta o sinal. Com o intuito de monitorar estes dados em tempo real, foi utilizado o *SignalTap*. Na Figura 9 é apresentada a forma de onda digital, visualizada por meio *SignalTap*, dos sinais transmitidos e recebidos.

Figura 9: Forma de onda recebida pelo *SignalTap*.



Foram capturados poucos bits de informação na Figura 9, no entanto, é possível observar que os dados recebidos são

idênticos ao transmitido. Isso já era esperado visto que o *loopback* direto entre as portas SMA não seria capaz de gerar qualquer atraso no sinal.

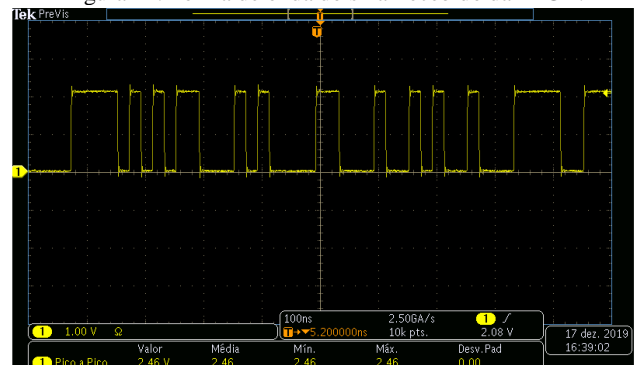
C. Monitoramento do osciloscópio

Além da montagem de *loopback*, outra configuração de experimento foi realizada, na qual consistiu em gerar e transmitir um sinal PRBS pela FPGA e receber no osciloscópio. Nas Figuras 10 e 11 são apresentadas a configuração de montagem e a forma de onda obtida, respectivamente.

Figura 10: Experimento prático.



Figura 11: Forma de onda do sinal recebido da FPGA.



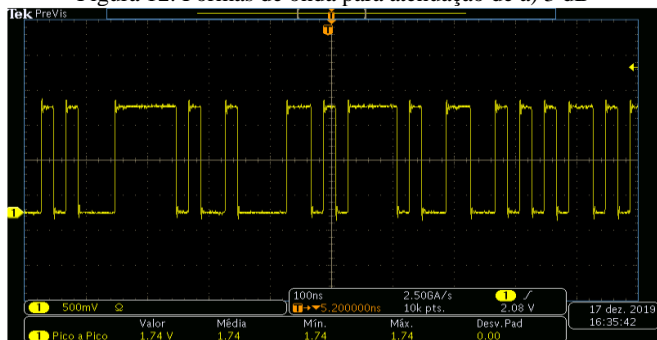
Como pode ser observado na Figura 11, o sinal recebido pelo osciloscópio apresenta aleatoriedade na representação dos bits, alternando entre níveis lógicos baixo e alto sem qualquer padrão aparente.

Além disso, observa-se um nível de tensão de 2,46 V de valor de pico.

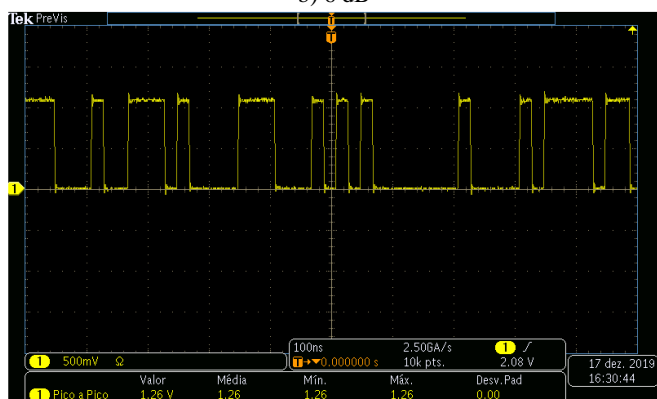
D. Formas de onda com atenuadores

Utilizando atenuadores SMA foram visualizados diferentes níveis de tensão para o sinal PRBS gerado pela FPGA. Nas Figuras 12 (a) e (b), estão apresentadas as formas de ondas obtidas.

Figura 12: Formas de onda para atenuação de a) 3 dB



b) 6 dB



Através da Eq. 1 foram calculados os níveis de tensão esperado para cada valor de atenuação. Na Tabela 1 onde relaciona os níveis de tensões medido e o esperado, para cada valor de atenuação.

Tabela 1: Níveis de tensão para cada valor de atenuação.

Atenuação (dB)	Medido (V)	Esperado (V)
3	1,74	1,74
6	1,26	1,23

IV. CONCLUSÕES

Neste trabalho foi apresentada a metodologia empregada para gerar e monitorar em tempo real um sinal pseudoaleatório utilizando a FPGA. Utilizando o *software* Quartus II, foi possível gerar um sinal do comprimento desejado, a partir do diagrama de blocos construído. E através da ferramenta *SignalTap*, visualizou-se o sinal transmitido e recebido pelas portas SMA da placa.

O gerador apresentado foi confeccionado com o intuito de aplicar a um circuito óptico, no entanto, este pode ser inserido em diversas aplicações, que já foram mencionadas na seção de Introdução.

REFERÊNCIAS

[1] DUARTE, Ulysses Rondina. **Técnicas Combinadas de Autoalimentação e Reuso de Comprimento de Onda em Transmissores Refletivos para Redes WDM-PON**. 2015. Tese (Doutorado em Ciências) – Escola de Engenharia de São Carlos-USP. São Paulo, 2015.

[2] THOMSEN, Adilson. O que é Arduino? **FILIFELOP**, 2014. Disponível em: <https://www.filipeflop.com/blog/o-que-e-arduino/>. Acesso em: 20 de Jul. de 2021.

[3] Conhecendo o ESP32. **CURTOCIRCUITO**, 2018. Disponível em: <https://www.curtocircuito.com.br/blog/Categoria%20IoT/conhecendo-esp32>. Acesso em: 20 de Jul. de 2021.

[4] CURVELLO, André. Primeiros passos de FPGA com a Papilio One. **FILIFELOP**, 2017. Disponível em: <https://www.filipeflop.com/blog/fpga-no-modo-spartan-com-papilio-one/>. Acesso em: 20 de Jul. de 2021.

[5] Altera DE2-115 *Development and Education Board*. **TERASIC**, 2013. Disponível em: <https://www.terasic.com.tw/en/>. Acesso em: 19 de Jul. de 2021.

[6] CARDOSO, F; FERNANDES, M. **FPGA e fluxo de projeto**. Campinas. 2007. Notas de aula. UNICAMP.

[7] DUARTE, J. **Aquisição de imagens através de FPGA**. 2015. Dissertação de mestrado, Departamento de Engenharia Eletrotécnica e de Computadores - Universidade de Coimbra, Coimbra (Portugal), 2015.

[8] AZEVEDO, F. *et al.* FPGA como ferramenta de ensino de engenharia. **FUPAC**, 2017. Disponível em: <https://ri.unipac.br/repositorio/wp-content/uploads/2019/09/2017-02-Filipe-Samuel-Teixeira-de-Azevedo.pdf>. Acesso em: 19 de Jul. de 2021.

[9] ARTUZI, W. **Espalhamento de Espectro**. Paraná. 2014. Notas de aula. UFPR.

[10]CORREIA, D. S. O. **Geração de números pseudo-aleatórios empregando mapas caóticos**. 2016. Dissertação de mestrado, Centro de Tecnologia e Geociências - Universidade Federal de Pernambuco, Recife-PE, 2016.

[11]H. Fang *et al.* "Developing a Linear Model of RF Power Generators with Pseudo Random Binary Signals (PRBS)," *14th International Conference on Control, Automation and Systems (ICCAS 2014)*, 2014, pp. 1158-1162.

[12]ALTERA. Design Debugging Using the SignalTap II Logic Analyzer.. Quartus II Handbook Version. 13.1. Vol. 3, Novembro, 2013.

[13]PHABRIX. Using Pseudo-Random Binary Sequences to Stress Test Serial Digital Interfaces. Disponível em: < https://phabrix.com/ftp/App_Notes/Stress_Whitepaper.pdf > Acesso em: 02 de Set. 2021.

[14]O decibel, ou melhor: os decibéis. Nota de aula. UFRGS. 1997.