

# Projeto em FPGA de um Filtro Digital FIR do Tipo *Sinc*-Janelado Utilizado na Tecnologia F-OFDM

Patrick Dantas Silva Sá Teles, Maykon Renan Pereira da Silva e Flávio Geraldo Coelho Rocha

**Resumo**— Este trabalho tem como objetivo apresentar a implementação de um algoritmo de processamento digital de sinais em um FPGA (*Field-Programmable Gate Array*) utilizando a placa de desenvolvimento Altera DE0. O algoritmo implementado é o de filtro digital do tipo Resposta ao Impulso Finita (FIR – *Finite Impulse Response*), mais especificamente um filtro *sinc*-janelado, cujo protótipo baseia-se em uma função *sinc* no domínio do tempo, com o intuito de obter uma próxima do ideal, retangular no domínio da frequência. Para tornar o filtro realizável e ao mesmo tempo eficiente, essa função *sinc* é multiplicada por uma função de janela propriamente escolhida para prover resultados de interesse tanto no domínio do tempo, quanto no domínio da frequência, o que inclui a análise da Interferência Inter-Simbólica (ISI – *Intersymbol Interference*) e Emissão Fora de Banda (OOBE – *Out-Of-Band Emission*). O filtro implementado pode ser utilizado na Modulação por Divisão Ortogonal de Frequência Filtrada (F-OFDM – *Filtered Orthogonal Frequency-Division Multiplexing*), uma tecnologia candidata a ser utilizada nas redes de comunicações móveis de quinta geração (5G), e consiste basicamente na aplicação de filtros nas sub-bandas do sistema OFDM convencional.

**Palavras-Chave**— F-OFDM, FPGA, Funções de Janela, Filtro *Sinc*-Janelado, OOBE

**Abstract**— This work aims to present the implementation of a digital signal processing algorithm in a Field-Programmable Gate Array (FPGA) using the Altera DE0 development board. The implemented algorithm is the digital filter of the type Finite Impulse Response (FIR), more specifically a windowed-sinc filter, whose prototype is based on a sinc function in the time domain, to obtain an ideal rectangular response in the frequency domain. To make the filter realizable and at the same time efficient, this sinc function is multiplied by a window function properly chosen to provide results of interest both in the time domain and in the frequency domain, which includes Intersymbol Interference (ISI) and Out-Of-Band Emission (OOBE) analysis. The implemented filter can be used in filtered Orthogonal Frequency-Division Multiplexing (F-OFDM), a candidate technology to be used in fifth-generation mobile communications networks (5G), and consists of the application of filters in the sub-bands of the conventional OFDM system.

**Keywords**— F-OFDM, FPGA, Window Functions, Windowed-Sinc Filter, OOBE

## I. INTRODUÇÃO

O FPGA (*Field-Programmable Gate Array*) é um circuito integrado configurado por *software* para implementação de

Patrick D.S.S. Teles, Maykon R.P. da Silva e Flávio G.C. Rocha fazem parte do Grupo INCOMM (*Information and Communication Engineering Group*) da Escola de Engenharia Elétrica, Mecânica e de Computação, Universidade Federal de Goiás, Goiânia, GO, Brasil, e-mail: patrickdsilva99@gmail.com, maykonrenan@discente.ufg.br, flavioger@ufg.br.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior -Brasil (CAPES) - Código de Financiamento 001.

processadores, controladores e decodificadores. As conexões entre as saídas de determinados blocos com as entradas de outros são programáveis através de um protocolo, como *Verilog* e *VHDL* (*VHSIC Hardware Description Language*). O FPGA possui ferramentas de projetos de sistemas para reduzir o ciclo de desenvolvimento, além de providenciar utilização eficiente do dispositivo, proporcionando assim ao projetista a possibilidade de encontrar o ponto ótimo entre tamanho do projeto e velocidade de implementação. Essa tecnologia pode ser utilizada para a programação e implementação de filtros digitais, visto que pode suportar funções de processamento paralelo em *software*, além de maior flexibilidade em *hardware*.

A implementação prática de técnicas de processamento digital de sinais em FPGA é de interesse em diversas aplicações, por exemplo, processamento de sinais e imageamento de ultrassom, com viabilidade demonstrada em [1], [2], filtragem digital para imagens em um processador embarcado implementado em uma distribuição Linux projetada para executar em FPGA [3], assim como o desenvolvimento de um modem OFDM (*Orthogonal Frequency-Division Multiplexing*) completo em um FPGA com a linguagem *VHDL* [4]–[6]. Além da tecnologia OFDM, existem estudos de implementação em FPGA de outras formas de onda, como *UFMC* (*Universal Filtered Multi-Carrier*) [7] e *FDM* não ortogonal [8]. Na comunidade acadêmica também podem ser encontrados projetos destinados ao estudo e otimização de arquiteturas e técnicas de desenvolvimento de filtros FIR para a tecnologia F-OFDM [9]. Porém, nota-se que em [9] os autores não avaliam o desempenho do filtro *sinc*-janelado quando implementado em *hardware* (FPGA) em termos dos níveis de OOBE (*Out-Of-Band Emission*) obtidos do sinal F-OFDM em comparação a formulação teórica. A análise da OOBE através da Densidade Espectral de Potência (PSD – *Power Spectral Density*) é fundamental para observar o grau de Interferência entre Portadoras (ICI – *Inter-carrier Interference*).

No sistema de modulação F-OFDM, é aplicado um filtro do tipo *sinc*-janelado nas sub-bandas do sistema modulação OFDM, reduzindo assim o consumo da banda de guarda entre sub-bandas, levando a uma utilização mais eficiente do espectro de frequência disponível. O filtro a ser implementado neste trabalho é um filtro de Resposta ao Impulso de duração Finita (FIR – *Finite Impulse Response*) do tipo *sinc*-janelado.

O restante do artigo está organizado da seguinte forma: A Seção II apresenta a teoria matemática que fundamenta o projeto do filtro digital FIR; A Seção III apresenta a arquitetura do sistema desenvolvido, a obtenção dos coeficientes dos filtros a serem testados no sistema, e, finalmente, a implementação do algoritmo em linguagem *VHDL*; A Seção IV

apresenta o desempenho do filtro *sinc*-janelado implementado em FPGA quando aplicado no sistema F-OFDM; Na Seção V, são apresentadas as conclusões obtidas.

## II. FUNDAMENTAÇÃO TEÓRICA

De acordo com [10], a forma mais direta de implementar um filtro digital LTI (*Linear Time-Invariant*) consiste em processar o sinal de entrada por meio de uma operação chamada de convolução linear, denotada por  $y = x * f$ , onde  $f$  é a resposta ao impulso do filtro, também chamada de núcleo (*kernel*) do filtro,  $x$  é o sinal de entrada, e  $y$  é o sinal de saída. Um filtro FIR consiste em um número finito de valores, portanto a soma das operações de multiplicação feitas em um processo de convolução é finita. Assim, a saída de um FIR de tamanho  $L$ , para uma série temporal de entrada  $x[n]$ , é definida da seguinte maneira:

$$y[n] = x[n] * f[n] = \sum_{k=0}^{L-1} f[k]x[n-k], \quad (1)$$

onde  $f$ , com seu índice  $k$  variando de 0 até  $L-1$ , representa os coeficientes do filtro. Esses valores também correspondem à sua resposta ao impulso. Essa expressão pode ser representada no domínio  $z$  como

$$Y(z) = F(z)X(z), \quad (2)$$

onde  $F(z)$  é a função de transferência do filtro FIR, definida no domínio  $z$  como

$$F(z) = \sum_{k=0}^{L-1} f[k]z^{-k}. \quad (3)$$

As raízes do polinômio  $F(z)$  definem os zeros do filtro. A quantidade de raízes define a ordem  $N$  do filtro, sendo que um filtro de comprimento  $L$  tem ordem  $N = L - 1$ .

### A. Método Sinc-Janelado

Uma função *sinc* no domínio do tempo produz no domínio da frequência a resposta de um filtro passa-baixa ideal.

A função *sinc* discreta é definida como

$$f[i] = \frac{\text{sen}(2\pi f_c i)}{i\pi}, \quad (4)$$

sendo  $f_c$  a frequência de corte, de maneira que todas as frequências abaixo dela são passadas com amplitude unitária, enquanto frequências mais altas são bloqueadas. A banda de passagem é constante até a frequência de corte, a atenuação fora da banda de passagem é infinita, e a transição entre as duas é infinitesimal.

Porém, é impossível armazenar infinitas amostras de um sinal, portanto é necessário limitar a função *sinc* no domínio do tempo com uma janela, obtendo então uma função *sinc* janelada. A equação que representa o processo de janelamento do *sinc* no domínio do tempo discreto é definida como

$$h[i]w[i] = f[i], \quad (5)$$

sendo  $i$  a amostra,  $h[i]$  a resposta ao impulso do *sinc*,  $w[i]$  a janela e  $f[i]$  o filtro *sinc*-janelado.

O processo de convolução, introduzido na seção anterior e detalhado na Equação 6, é definido como

$$x[i] * f[i] = y[i], \quad (6)$$

sendo  $x[i]$  o sinal de entrada do sistema,  $f[i]$  o filtro *sinc*-janelado e  $y[i]$  o sinal filtrado.

A convolução entre um sinal de entrada e um filtro com resposta ao impulso dada por uma função *sinc* gera uma resposta em frequência passa-baixas ideal, que por sua vez é não causal. Uma solução é truncar a função *sinc* (ou seja, fazer janelamento com uma simples janela retangular), obtendo assim uma resposta em frequência não ideal, com presença de oscilações indesejadas na banda de passagem, porém causal. O método do janelamento com janelas de formatos diferentes da retangular melhora esse resultado, suavizando o decaimento da resposta em frequência [11].

### B. Sistemas OFDM e F-OFDM

Na Figura 1 é possível visualizar uma sub-banda OFDM, que consiste em várias subportadoras ortogonais multiplexadas. Assim, cada subportadora, cujo formato corresponde a uma função *sinc*, se sobrepõe parcialmente sem interferência a subportadoras adjacentes. Por isso, para que seja possível filtrar uma sub-banda OFDM é ideal que o filtro tenha uma resposta em frequência de formato retangular.

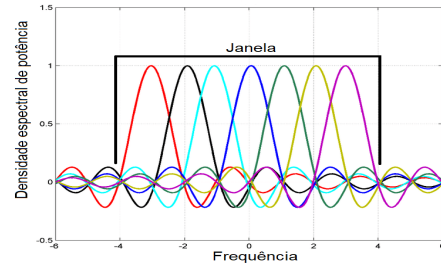


Fig. 1: PSD de uma sub-banda OFDM [12].

No sistema OFDM convencional, os lóbulos laterais da resposta em frequência decaem em torno da frequência de corte  $f_c$ . Esse decaimento ocorre a uma taxa de  $1/f_c$ , que é considerado um decaimento lento. Desta forma, o espectro de frequência não é muito bem localizado, dificultando sua coexistência com outros sistemas ou com outras sub-bandas adjacentes. A tecnologia OFDM apresenta taxas consideráveis de OOB. Portanto, sempre haverá dispersão do sinal que está no domínio da frequência. Para prevenir que as OOB causem ICI, é preciso afastar as sub-bandas através da banda de guarda.

O sistema F-OFDM reduz a banda de guarda aplicando filtros nas sub-bandas para reduzir as OOB e consequentemente, aumentar a eficiência espectral. Para permitir filtragem nas sub-bandas de símbolos modulados pela tecnologia OFDM e obter os benefícios prometidos pela F-OFDM, filtros propriamente projetados são necessários. No geral, o projeto de um filtro envolve a troca (*trade-off*) entre as características do domínio do tempo e do domínio da frequência, além de depender diretamente da complexidade da implementação [13].

### III. IMPLEMENTAÇÃO

Na Figura 2, pode-se observar a ação da janela de Hamming para o janelamento suave da função *sinc* a ser aplicada na filtragem de um sinal de entrada arbitrário. Neste trabalho, um arquivo foi gerado pelo Matlab e utilizado para popular um conjunto de vetores binários, mais especificamente, uma matriz de binários com 512 linhas e 512 colunas foi gerada de maneira que fossem necessários 10 *bits* para cada elemento do vetor, considerando que  $W_{in} = 10$  e  $2^{W_{in}-1} = 512$ . Assim, cada um dos 512 elementos pode armazenar valores de  $-512$  a  $511$ . Na Figura 3, está claro o número de elementos lógicos utilizados, assim como a quantidade de multiplicadores integrados e *bits* de memória utilizados, entre outras informações. Percebe-se que a operação de convolução necessita de uso intensivo de multiplicadores. Isso é esperado, como ficará explícito a seguir.

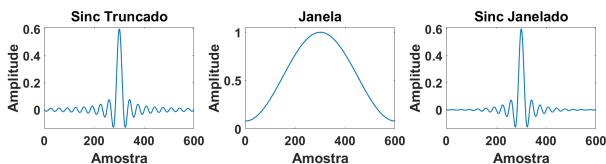


Fig. 2: Aplicação da função de janela de Hamming no sinal *sinc* truncado.

Flow Status	Successful - Mon Dec 07 23:30:45 2020
Quartus II Version	9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name	DE0_FIR_Filter
Top-level Entity Name	DE0_FIR_Filter
Family	Cyclone III
Device	EP3C16F484C6
Timing Models	Final
Met timing requirements	N/A
Total logic elements	12,309 / 15,408 ( 80 % )
Total combinational functions	11,599 / 15,408 ( 75 % )
Dedicated logic registers	6,325 / 15,408 ( 41 % )
Total registers	6325
Total pins	252 / 347 ( 73 % )
Total virtual pins	0
Total memory bits	9,176 / 516,096 ( 2 % )
Embedded Multiplier 9-bit elements	112 / 112 ( 100 % )
Total PLLs	0 / 4 ( 0 % )

Fig. 3: Visualização dos resultados do sistema no Quartus II.

Para um filtro de tamanho  $L$ ,  $\log_2(L)$  *bits* adicionais são necessários para a soma de produtos do tipo *unsigned*, e  $\log_2(L) - 1$  *bits* de guarda para aritmética com tipos *signed* (Esse tipo difere do *std\_logic\_vector* por possuir um *bit* exclusivo para o sinal do valor binário, e por utilizar a biblioteca *numeric\_std*, mais indicada para operações aritméticas com vetores binários).

A implementação manual do algoritmo do filtro foi feita em código escrito em VHDL utilizando 512 elementos, com cada elemento sendo um vetor de 10 *bits*. Esses valores são facilmente ajustáveis pela definição dos dados genéricos na definição da entidade VHDL. Com o intuito de ser conciso, o algoritmo não é mostrado no artigo, mas pode ser facilmente obtido pelos emails de contato dos autores. Para executar o programa, o *hardware* irá ligar fisicamente elementos lógicos disponíveis dentro do chip, e todos os processos definidos no código serão executados simultaneamente. Portanto, em toda subida de *clock*, serão executadas todas as operações

de todos os processos que dependem da subida do *clock* (*rising\_edge*(*clk*)). Além disso, é importante detalhar que o tipo *std\_logic\_vector*, um vetor de binários, é utilizado preferencialmente para a transmissão de sinais entre os elementos do projeto, portanto será sempre utilizado como entrada e saída nas entidades, e o *signed* é utilizado para operações matemáticas (definidas pela biblioteca *numeric\_std*) entre sinais.

O compilador da interface de desenvolvimento Quartus II busca a forma mais eficiente de encaixar o projeto no dispositivo pretendido, e suprime tudo que é desnecessário do arquivo final que será programado no FPGA. Com isso, através do desenvolvimento do sistema foi possível perceber que o total de recursos que o projeto demanda depende da variável *Wout*, que representa a precisão em *bits* da saída do sistema, mas depende mais diretamente da quantidade de *bits* do vetor de saída que são de fato utilizados. O vetor de saída tem tamanho *Wout*, mas o compilador somente reservará blocos de processamento do FPGA para os *bits* da saída que forem lidos. Com a ligação dos 10 *bits* menos significativos aos pinos de entrada e saída do *hardware*, aproximadamente 80% dos recursos do FPGA são utilizados, e com a utilização de 11 *bits*, aproximadamente 99% dos recursos são utilizados.

### IV. RESULTADOS

O filtro foi implementado em VHDL com o objetivo de ser genérico, com precisão regulável. Nesta sessão é introduzida a técnica utilizada para simular a entidade em ambiente de teste. As simulações são facilmente comprovadas na placa DE0 através da utilização do *display* de 7 segmentos vinculado à variável de saída, assim comprovando que o dispositivo e a simulação estão apresentando processamentos idênticos. Neste capítulo são demonstrados os resultados obtidos ao utilizar tipos diferentes de filtros e técnicas de janelamento com o algoritmo implementado, assim como a análise de tempo de execução da operação de convolução no FPGA com o sistema simplificado e otimizado para execução rápida e direta. Os testes a serem executados pelo *software* ModelSim são especificados por arquivos de *testbench*, também escritos em VHDL. Esses arquivos têm o objetivo de simular o *clock* que rege o funcionamento de todo o sistema, assim como instanciar entidades, mapeando suas variáveis a variáveis de teste, com o objetivo de tornar os valores acessíveis por todo o período de execução.

Para testar a funcionalidade do filtro implementado em FPGA, os componentes real e imaginário do sinal OFDM devem passar individualmente pelo processo de convolução. Com a utilização da biblioteca *textio* é possível realizar manipulação de arquivos em *testbenches* implementadas em VHDL. Com isso, é possível visualizar graficamente a saída do sinal filtrado no FPGA, importando os arquivos que contém os componentes real e imaginário do sinal filtrado e atribuindo-os a um vetor de números complexos no Matlab.

Na Figura 4, é possível visualizar o filtro digital do tipo *sinc*-janelado projetado para filtrar uma sub-banda OFDM, além de sua resposta a uma entrada retangular. Esse filtro utiliza uma janela de Hamming. É possível visualizar o resultado da

convolução no tempo entre o filtro e o vetor que representa os valores reais do sinal OFDM na Figura 5, obtida através do *software* de simulação ModelSim. O arquivo de saída, para ser exportado para o Matlab, é preenchido com os valores correspondentes ao sinal de saída da operação.

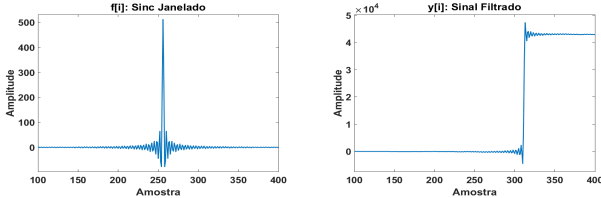


Fig. 4: Filtro projetado para filtrar sub-bandas OFDM e sua resposta a um sinal retangular. Formas de onda obtidas no ModelSim e exportadas para o Matlab.

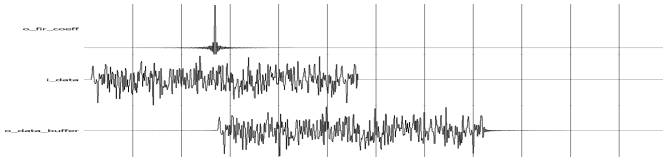


Fig. 5: Visualização do filtro, sinal de entrada e sinal de saída no ModelSim.

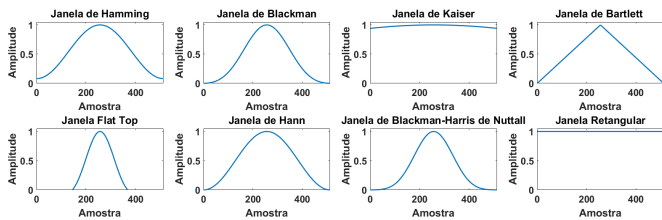


Fig. 6: Principais funções de janelamento.

Alguns exemplos de diferentes tipos de funções de janelamento podem ser vistos na Figura 6, sendo que a Janela Retangular é simplesmente a truncagem do sinc no comprimento igual ao tamanho da janela, neste caso com 512 amostras.

Para analisar o desempenho de diferentes funções de janela após o processo de filtragem no sinal OFDM, é conveniente utilizar a ferramenta `plotbrowser` do Matlab, ativando-a com o comando `plotbrowser('on')`, proporcionando uma forma rápida e intuitiva de comparar individualmente o impacto que o janelamento causa na função *sinc*, explícito principalmente quando qualquer função de janela é posta lado a lado com a Janela Retangular, como pode ser visto na Figura 7a. A janela que apresentou o melhor desempenho em diminuição das OOBs foi a de Hamming, sendo que em segundo lugar ficam as janelas de Hann e Flat-top, dispostas na Figura 7b. Percebe-se que a janela de Hamming apresenta o melhor desempenho em redução de OOBs e rápido decaimento de amplitude em frequências acima da frequência de corte.

Para comparar a filtragem do sinal em *hardware* e no Matlab, pode-se observar na Figura 8 o sinal OFDM filtrado no FPGA com valores de entrada (sinais de entrada e coeficientes do filtro) de 10 e 11 *bits*, e o mesmo sinal filtrado no Matlab.

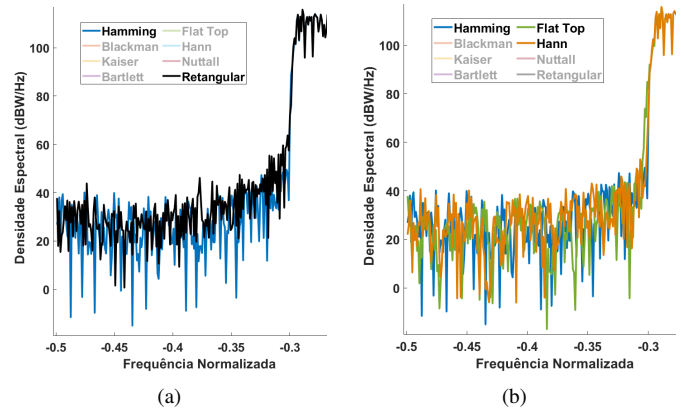


Fig. 7: Comparação de funções de janelamento quando aplicadas a um filtro *sinc*-janelado utilizado para filtrar uma sub-banda OFDM em FPGA com precisão de 11 *bits*.

Tecnologia de Modulação	Valor médio das OOBs
OFDM	-56.23 dBW/Hz
F-OFDM Matlab	-138.78 dBW/Hz
F-OFDM FPGA 10-bit	-107.37 dBW/Hz
F-OFDM FPGA 11-bit	-115.58 dBW/Hz

TABELA I: Tabela de valores médios de OOBs para cada caso exposto na Figura 8.

Os sinais estão sobrepostos com o sinal original OFDM para facilitar a comparação dos valores de OOBs.

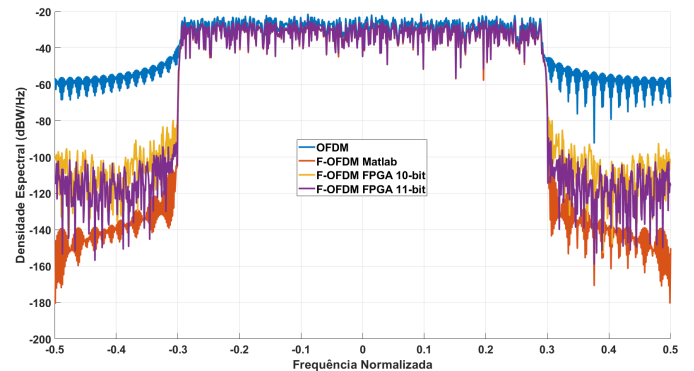


Fig. 8: Densidade espectral de potência da tecnologia de modulação OFDM e F-OFDM sobrepostos, com filtro implementado em FPGA e em Matlab.

Na Tabela I, estão dispostos os valores médios das emissões fora de banda em cada caso. É possível concluir que, apesar de constatado que a filtragem realizada no Matlab é superior, o projeto prático apresenta reduções consideráveis de OOBs. A diferença em performance se dá devido aos arredondamentos obrigatórios de cada valor dos vetores de entrada e de coeficientes do filtro, para que possam ser representados por vetores binários. Percebe-se também que quanto maior a precisão dos valores de entrada no FPGA, menores são as emissões fora de banda.

Com a análise de tempo da interface de desenvolvimento



utilizada, é possível averiguar o período de *clock* efetivo no FPGA. Levando em consideração todos os esforços do *software* para encaixar o projeto no *hardware*, é possível que o ciclo de *clock* tenha diminuído do máximo padrão de 50 MHz a até um *clock* máximo de 10 MHz.

Com precisão de 10 *bits*, o valor máximo de *clock* obtido foi 34,68 MHz, com período de 28,836 ns. Utilizando o SignalTap do Quartus II é possível verificar diretamente do FPGA as alterações de qualquer natureza em qualquer porta do projeto. Na Figura 9 é possível verificar que a execução completa do sistema, do momento em que o botão de *reset* é pressionado até a filtragem completa do sinal OFDM, tem um tempo de duração de 45,30  $\mu$ s. Com precisão de 11 *bits*, o valor máximo de *clock* obtido foi 34,13 MHz, com período de 29,291 ns. A filtragem completa do sinal OFDM apresentou tempo de duração de 46,02  $\mu$ s. Com 10 *bits* de precisão, o projeto utiliza por volta de 80% dos recursos, enquanto que com 11 *bits* aproximadamente 99% dos recursos são reservados. Com a precisão de 12 bits, o projeto se torna grande demais para ser encaixado no modelo de FPGA utilizado.

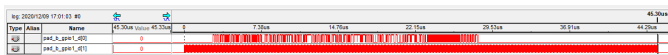


Fig. 9: Visualização de tempo de execução.

Na Tabela II estão dispostos os dados referentes a precisão, recursos utilizados e tempos de execução com as precisões de *bits* testadas em *hardware*. Analisando a tabela, é fácil notar que o tempo de processamento não necessariamente aumenta com mais blocos lógicos utilizados, e sim está diretamente relacionado com a velocidade do *clock*, confirmando na prática a capacidade de processamento paralelo do FPGA.

$W_{in}$	Recursos utilizados	Velocidade máxima de <i>clock</i>	Tempo de execução
10 <i>bits</i>	≈80%	34,68 MHz	45,30 $\mu$ s
11 <i>bits</i>	≈99%	34,13 MHz	46,02 $\mu$ s

TABELA II: Tabela de precisão de *bits*, recursos utilizados, velocidade máxima de *clock* obtida e tempo total de execução para filtragem completa de sinal OFDM.

## V. CONCLUSÃO

Neste trabalho, foi feita a implementação prática de um algoritmo de convolução no tempo de um sinal de entrada e um outro sinal composto de coeficientes gerados em Matlab que representam um filtro FIR, com parâmetros variados. Além disso, foi desenvolvido um sistema de testes para as entidades construídas, com o objetivo de analisar as formas de onda formadas pelos vetores de *bits* obtidos na saída do algoritmo por todo o período de execução.

Percebe-se que o projeto em FPGA do algoritmo de processamento digital de sinal escolhido apresenta complexidades pontuais, apesar da implementação do algoritmo principal ser rápida e intuitiva. A mudança de paradigma ao relacionar

desenvolvimento em *software* e *hardware* é muito significativa, e um novo conceito de ordem de execução precisa ser internalizado por parte do desenvolvedor do sistema.

Além dos aspectos a se considerar em desenvolvimento e na implementação em *hardware*, a alteração dos parâmetros do filtro FIR alteram profundamente o resultado do algoritmo, mas também impacta o cálculo da utilização de recursos do FPGA, assim como a sua capacidade máxima de processamento ao considerar toda a complexidade do programa a ser nele implementado, aspecto essencial para garantir a utilização devida da capacidade que FPGAs têm de ágil processamento paralelo. A implementação apresentada neste trabalho abre o caminho para a implementação de melhorias técnicas, tendo em vista que o desempenho de sistemas escritos em linguagem de descrição de *hardware* depende muito da experiência do programador. Além disso, uma análise funcional detalhada do algoritmo pode tornar possível sua evolução com o fim de torná-lo inteligente, introduzindo uma análise adaptativa aos coeficientes do filtro para um aumento ainda maior da eficiência espectral, almejada para a evolução das tecnologias de comunicação móvel.

## REFERÊNCIAS

- [1] A. Assef, J. Maia, B. Ferreira, H. Souza, G. Bassan, A. De, M. Filho, and E. Costa, "Projeto de um filtro digital fir passa-baixa em fpga para aplicações de processamento de sinais de ultrassom," 04 2016.
- [2] B. M. Ferreira, "Modelagem e implementação de um sistema de processamento digital de sinais baseado em fpga para geração de imagens por ultrassom usando simulink." Master's thesis, Universidade Tecnológica Federal do Paraná, Curitiba, 2017. [Online]. Available: <http://repositorio.utfpr.edu.br/jspui/handle/1/2874>
- [3] J. I. Galvan-Tejada, C. E. Galvan T, J. M. Celaya-Padilla, and J. R. Delgado C, "Digital filter implementation over fpga platform with linux os," *Procedia Engineering*, vol. 35, pp. 223 – 229, 2012, international Meeting of Electrical Engineering Research 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S18770581201819X>
- [4] T. M. Siqueira, "Implementação de um Modem ofdm em fpga," Vitória, 2004.
- [5] S. L. e João Filipe Lopes, "Implementação em fpga da modulação/desmodulação ofdm 802.11p," Master's thesis, Universidade de Aveiro, Toledo, 2011. [Online]. Available: <https://ria.ua.pt/bitstream/10773/8606/1/248799.pdf>
- [6] A. C. Woss, "Implementação de um algoritmo de comunicação embarcado em dispositivo lógico programável com aplicação de técnica de multiplexação em frequência," Toledo, 2014. [Online]. Available: [http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/3705/1/TD\\_COELE\\_2014\\_1\\_%2006.pdf](http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/3705/1/TD_COELE_2014_1_%2006.pdf)
- [7] R. Jafri, J. Majid, L. Zhang, M. A. Imran, and M. N. ul Islam, "Fpga implementation of ufmc based baseband transmitter: Case study for lte 10mhz channelization," *Hindawi*, p. 12, 2018, wireless Communications and Mobile Computing, vol. 2018. [Online]. Available: <https://www.hindawi.com/journals/wcmc/2018/2139794/>
- [8] R. Grammenos and I. Darwazeh, "Fpga design considerations for non-orthogonal fdm signal detection," 2010.
- [9] M. M. Shahbaz, A. Wakeel, Junaid-ur-Rehman, and B. Khan, "Fpga based implementation of fir filter for fofdm waveform," in *2019 2nd International Conference on Communication, Computing and Digital Systems (C-CODE)*, 2019, pp. 226–230.
- [10] U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*, 4th ed. Springer Publishing Company, Incorporated, 2014.
- [11] H. Kaplan and A. Nacaroglu, "A new window function for fir filter design," in *2015 23rd Signal Processing and Communications Applications Conference (SIU)*, 2015, pp. 2458–2461.
- [12] Y. Al-jawhar, "An implementation of peak to average power ratio reduction for multicarrier system (orthogonal frequency division multiplexing)," Ph.D. dissertation, 11 2017.
- [13] M. R. P. da Silva, "Propostas de um modelo para o tráfego de redes de comunicações e de funções de janela para o filtro fir utilizado na tecnologia f-fofdm," 2019.