

Universidade Federal do Paraná

Leonardo de Andrade Santos

RELATÓRIO FINAL

**Projeto de um circuito integrado dedicado de um pré-distorcedor digital
baseado em polinômio de memória**

Relatório apresentado à Coordenação de Iniciação Científica e Tecnológica da Universidade Federal do Paraná como requisito parcial da conclusão das atividades de Iniciação Científica ou Iniciação em Desenvolvimento Tecnológico e Inovação - Edital 2024

Orientador(a): Prof.(a) Sibilla Batista da Luz França

Título do Projeto: Desenvolvimento e implementação em hardware de sistemas digitais modernos usando linguagem de descrição de hardware

CURITIBA

2024

1 Resumo

A evolução da comunicação sem fio permitiu o desenvolvimento de diversas aplicações móveis, como IoT. A melhoria da eficiência energética é essencial para dispositivos móveis e estações de rádio, buscando reduzir perdas e aumentar a autonomia. Porém a eficiência energética reduz a linearidade dos amplificadores, o que afeta modulações de fase e amplitude, aumentando erros de transmissão. Sendo assim, uma alternativa para contornar esse obstáculo, que é implementar um sistema, eficiente energeticamente e linear é a implementação de um (DPD) em cascata com um Amplificador de Potência (PA). Portanto, este trabalho visa implementar um código em VHDL para processar as características de um amplificador em tempo real, minimizando não linearidades e consumo de energia. A modelagem foi feita inicialmente em software e implementada em FPGA (Field-Programmable Gate Array) para validação e análise de desempenho. Durante o desenvolvimento, testes e análises foram realizados com um polinômio de memória de segundo grau e um sinal de amostra memorizado, mostrando uma redução efetiva nas distorções do PA. Foi utilizada a métrica de Erro Quadrático Médio Normalizado (NMSE - Normalized Mean Squared Error) para quantificar a qualidade da modelagem implementada. O NMSE de -23,57 dB mostrou eficiência na redução de distorções. Valores menores de NMSE refletem melhor capacidade de reduzir distorções e preservação da fidelidade do sinal. Assim a próxima etapa consiste na modelagem do PA em vírgula fixa, a fim de atingir o NMSE mais próximo do atingido em vírgula flutuante com a menor resolução em bits, visando alcançar a melhor desempenho em hardware. A modelagem em vírgula fixa com 8 bits otimizou o desempenho, mas que por consequência aumentou o NMSE para -21,23 dB. Com base nos resultados obtidos e no desempenho promissor alcançado, o projeto de apresentou uma solução eficiente e robusta para a pré-distorção digital em amplificadores de potência, promovendo a evolução dos sistemas de comunicação móvel.

Palavras-chave: VHDL, FPGA, DPD

2 Introdução

A evolução dos sistemas de comunicação móveis, impulsionada pela crescente demanda por comunicações mais rápidas e eficientes, tem levado à implementação de uma variedade de serviços, incluindo aplicações multimídia, desenvolvimento web e aplicações IoT (JOHN, 2016). No entanto, essa evolução também trouxe desafios significativos, como a necessidade de melhorar a eficiência energética, tanto para dispositivos móveis, visando aumentar a autonomia da bateria, quanto para estações de rádio base, visando reduzir o consumo de energia devido às perdas de calor. Para atender a essas demandas, estratégias de modulação que alteram tanto a fase quanto a amplitude de ondas portadoras em radiofrequência se tornaram essenciais (KENINGTON, 2000). Além disso, a modulação na amplitude requer linearidade na transmissão para evitar erros e interferências na comunicação entre usuários vizinhos (CRIPPS, 2006). Essa complexa tarefa recai sobre o projetista do PARF (Amplificador de Potência de Rádio Frequência), que enfrenta o desafio de desenvolver um hardware eficiente em termos energéticos e linear ao mesmo tempo, uma vez que esses dois objetivos podem entrar em conflito (CHAVEZ, 2018). Uma solução para contornar esse desafio é a implementação de um pré-distorcedor de Sinais Digital em Banda Base, que visa compensar a distorção causada pelo PARF (CRIPPS, 2006). O DPD (Pre-distorcedor Digital) é conectado em cascata ao PARF e requer um modelo de alta precisão e baixa complexidade computacional para representar as características de transferência direta e inversa do PARF. Existem duas abordagens para modelar o PARF: modelos físicos, que são detalhadas e computacionalmente complexos, e modelos empíricos, que se baseiam em medições de entrada e saída do PARF, com menor complexidade computacional, mas com uma possível diminuição da precisão. Devido às exigências rigorosas de frequência de operação, a paralelização das operações torna-se essencial, e as FPGAs (Matriz de Portas Programáveis em Campo) emergem como uma alternativa viável para a implementação de circuitos pré-distorcedores (PEDRONI, 2010). As FPGAs são dispositivos lógicos programáveis que permitem a reconfiguração física de componentes de eletrônica digital, acelerando processos e suportando operações paralelas e sequenciais. Neste contexto esse projeto foi planejado com os seguintes objetivos geral e específicos:

2.1 Objetivo Geral

Construção e validação de um código em linguagem de descrição de hardware capaz de processar em tempo real as características de transferência de um amplificador. O código deve ser capaz de reproduzir não linearidades e efeitos de memória. Além disso, o código deve ocupar a quantidade mínima de recursos lógicos do circuito digital

que irá sintetizá-lo, buscando reduzir o consumo de energia.

2.2 Objetivos Específicos

Para alcançar o objetivo geral, este trabalho foi desenvolvido com base nos seguintes objetivos específicos:

1. Modelar com precisão o PA em software;
2. Modelar o DPD em software a partir da modelagem do PA;
3. Implementar o DPD em hardware utilizando uma HDL;

3 Pesquisa bibliográfica

A evolução dos sistemas de comunicações sem fio, fomentou a implementação de diversas aplicações móveis. Neste contexto, melhorar a eficiência energética desses sistemas se torna uma característica desejável, tanto para os dispositivos móveis que buscam melhorar a autonomia das baterias quando para os sistemas de comunicação radio base que tendem a perder energia em forma de calor. Sendo assim, o sistema de comunicação pode ser dividido em 3 sub-sistemas principais: Meio transmissor, Receptor e o Transmissor, conforme ilustrado por (SCHUARTZ; LIMA, 2017). No entanto, este trabalho foca exclusivamente no sistema transmissor, ilustrado pela figura 1, mais especificamente no amplificador de potência por ser o componente de maior demanda energética já que é o componente responsável por converter a energia da fonte em energia irradiada pela antena de transmissão. Portanto, a eficiência do sistema de transmissão depende diretamente do desempenho do transmissor.

Fonte: (SCHUARTZ; LIMA, 2017)

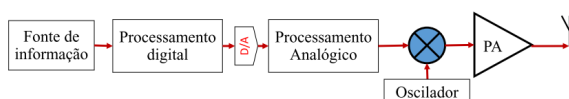


Figura 1 – Sistema de transmissão simplificado

Considerando também que a largura de banda reservada para sistemas de comunicação sem fio é reduzida, torna-se desejável que ela seja utilizada da maneira mais eficiente o possível. Diante desse cenário, segundo (KENINGTON, 2000), só é possível alcançar as maiores taxas utilizando estratégias de modulações, que alterem tanto a fase quanto a amplitude de uma onda portadora em rádio frequência. Ainda segundo (KENINGTON, 2000), a modulação na amplitude, exige linearidade na transmissão afim de evitar erros e interferência na comunicação entre os usuários vizinhos. Ante esse panorama, o projetista do PARF se depara com esse desafio, que é desenvolver um hardware eficiente energeticamente e com uma boa linearidade, o que é um compromisso conflitante, conforme descrito por (CRIPPS, 2006). Esse comportamento se deve, ao fato que um PARF atua de forma eficiente, ou seja, com baixo consumo de energia, na área próxima à de saturação, que é a região em que opera em regimes não lineares, conforme ilustrado pela figura 2.

Fonte: (CHAVEZ, 2018)

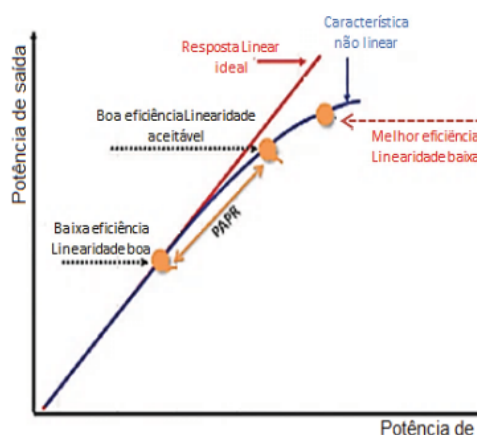


Figura 2 – Curva de saída do amplificador

A fim de contornar esse obstáculo foi adicionado a cadeia de transmissão um método de equalização de sinais, conforme argumentado por (KENINGTON, 2000). Um exemplo de técnica de linearização de sinais é a implementação de um pré-distorcedor de sinais digitais em banda base, o qual apresenta um melhor custo-benefício (KENINGTON, 2000). Essa técnica consiste em distorcer o sinal de entrada utilizando técnicas de processamento digital, antes que esse module uma portadora, de forma compensativa à distorção causada pelo PARF. De maneira sucinta, o DPD é conectado em cascata ao PARF e é projetado de forma que apresenta a função transferência inversa ao PARF. Para isso, é necessário um modelo de alta precisão e baixa complexidade computacional, capaz de representar as características de transferência direta e inversa de um PARF. Isso significa modelar o seu comportamento real utilizando um software. A figura 3 ilustra o processo de um pré-distorcedor digital.

Fonte: (CHAVEZ, 2018)

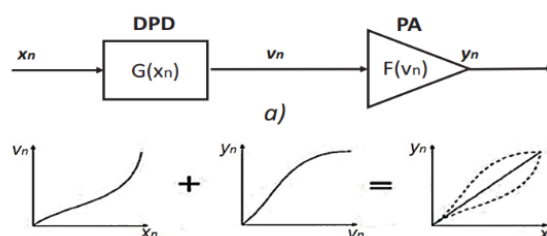


Figura 3 – ilustração do pré-distorcedor em cascata

Segundo (JOHN, 2016), existem duas técnicas utilizadas para fazer essa modelagem. Uma consiste na descrição detalhada do PARF, que implica em uma maior complexidade computacional, esses modelos são conhecidos como modelos físicos. A outra abordagem é conhecida como modelo empírico, este modelo consiste em coletar amostras na entrada e na saída do PARF em domínio temporal, e através destes

dados simulam um modelo matemático do sistema. Uma das vantagens desse método é que ele não exige conhecimento prévio da estrutura do PARF e possui baixa complexidade computacional. Mesmo se todos os parâmetros fossem conhecidos e o equacionamento completo do circuito fosse obtido, uma função inversa poderia ser encontrada, possivelmente muito mais complexa que séries de Volterra. No entanto, sua precisão pode ser ligeiramente afetada pelo modelo adotado. Sendo assim, como a proposta do projeto é a implementação de um DPD em hardware, se faz necessário que o circuito apresente a menor complexidade possível, torna-se mais viável fazer a implementação utilizando modelagem matemática.

3.1 Modelagens Matemáticas

3.1.1 Séries de Volterra

Segundo (LIMA; GHIONE, 2009) série de Volterra pode ser vista como uma extensão multidimensional da série de Taylor para sistemas dinâmicos. A modelagem começa com a representação do sistema através de uma série infinita de integrais convolucionais, onde cada termo da série corresponde a uma ordem de não linearidade e memória.

A saída $y(t)$ de um sistema pode ser expressa pela equação 1:

$$y(t) = h_0 + \sum_{n=1}^{\infty} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_n(\tau_1, \tau_2, \dots, \tau_n) \prod_{i=1}^n x(t - \tau_i) d\tau_i \quad (1)$$

onde h_n são os núcleos de Volterra, que caracterizam a resposta do sistema para a n -ésima ordem de não linearidade e $x(t)$ é a entrada do sistema.

Os núcleos de Volterra h_n são funções de várias variáveis que capturam a dinâmica do sistema em diferentes ordens. Para a maioria das aplicações práticas, a série é truncada para incluir apenas um número finito de termos, já que a identificação de todos os núcleos de uma série infinita é impraticável.

3.1.2 Polinômio de memória

Um modelo simples, utilizado na modelagem comportamental simplificada das séries de Volterra considerando apenas componentes unidimensionais¹ é o MP, que é um modelo compacto, de baixo custo computacional e linear em seus parâmetros. O MP gera baixo erro quando aplicado à PAs que apresentam pouco efeito de memória. O DPD e pós distorsor apresentam característica inversa a do PA (SCHUARTZ; LIMA, 2017), portanto o mesmo modelo pode ser utilizado. A equação 2 apresenta o MP conforme é ilustrado por (SCHUARTZ; LIMA, 2017):

¹ Cada termo do somatório é composto por amostras no mesmo instante, por exemplo: $x(n)|x(n)|, x(n-1)|x(n-1)|$; termos bidimensionais são compostos por amostras em instantes de tempos distintos, como por exemplo: $x(n)|x(n-1)|$

$$y(n) = \sum_{p=1}^P \sum_{m=0}^M h_{p,m} x(n-m) |x(n-m)|^{p-1} \quad (2)$$

Como a proposta do trabalho é a implementação em hardware desse modelo, torna-se necessário paralelizar operações aritméticas de forma a alcançar uma taxa de operação que satisfaça a norma regulamentadora. Nesse contexto, as FPGAs apresentam-se como uma alternativa viável para a implementação de circuitos pré-distorcedores.

3.2 FPGAs

Como descrito em (PEDRONI, 2010), FPGAs são uma classe de dispositivos lógicos programáveis que permitem a reconfiguração física de seus componentes de eletrônica digital por meio de uma linguagem de descrição de hardware. Basicamente, as FPGAs consistem em um conjunto de subcircuitos digitais interconectados, capazes de realizar diversas funções comuns enquanto oferecem um alto nível de flexibilidade. Devido a essas características, FPGAs podem ser utilizadas para aplicações como processamento de imagem em tempo real e aprendizado de máquina.

FPGAs têm a capacidade de sintetizar arquiteturas complexas de eletrônica digital, resultando em um funcionamento altamente paralelizado que permite um processamento rápido com várias portas de entrada e saída. Além disso, elas também suportam o desenvolvimento de códigos sequenciais.

A estrutura interna de uma FPGA é composta fundamentalmente por blocos lógicos interligados, organizados em uma matriz. Cada bloco é formado por diversos sub-blocos, que por sua vez contêm os componentes mais básicos da hierarquia. FPGAs da Intel e da Xilinx possuem nomenclaturas e organizações diferentes para esses blocos e sub-blocos. Isso é ilustrado na Figura 4 e na Figura 5, que mostram as FPGAs Intel Stratix X e Xilinx Ultrascale+, respectivamente. Embora as arquiteturas sejam fundamentalmente semelhantes, com a disposição em matriz dos blocos e funcionalidades dos componentes fundamentais sendo universais, os blocos lógicos são denominados LAB nas FPGAs da Intel e CLB nas FPGAs da Xilinx. Os sub-blocos são chamados de ALM ou LE, dependendo da FPGA da Intel, e de Slices nas FPGAs da Xilinx.

Fonte: (PEDRONI, 2010)

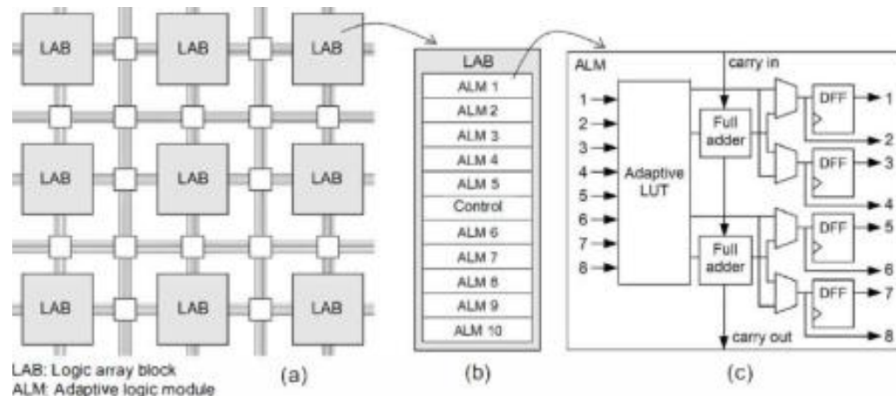


Figura 4 – Estrutura Interna da FPGA Stratix X da Intel

Fonte: (PEDRONI, 2010)

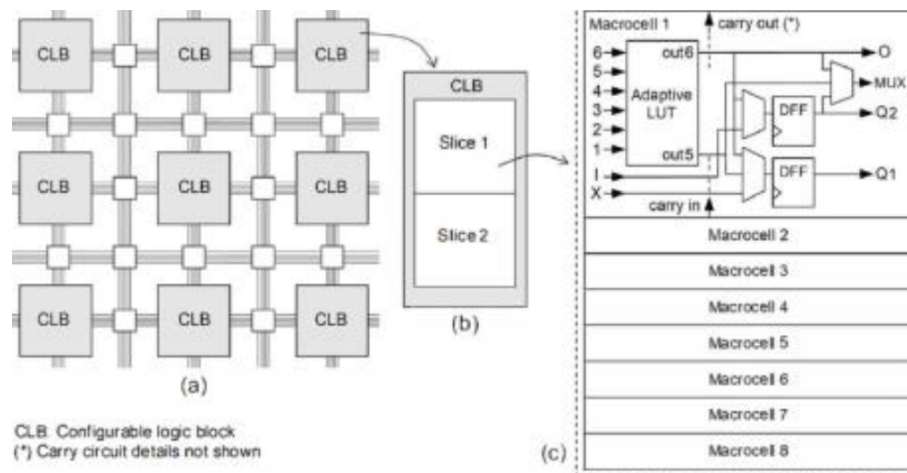


Figura 5 – Estrutura Interna da FPGA Ultrascale+

Os sub-blocos das FPGAs são compostos por LUTs e registradores. As LUTs são compostas por uma árvore binária de multiplexadores 2:1, permitindo o armazenamento de uma função lógica na forma de SOP. Os registradores são os componentes síncronos dos sub-blocos. Além da estrutura mencionada, FPGAs comumente possuem diversos módulos integrados, como CPUs, DSPs, memória Flash, PLLs, que aumentam ainda mais as capacidades do FPGA.

As FPGAs são programadas utilizando uma linguagem de descrição de hardware, sendo o VHDL uma das mais comuns para a síntese de circuitos integrados de alta velocidade. Criada por uma iniciativa financiada pelo Departamento de Defesa dos Estados Unidos em meados dos anos 80, o VHDL foi a primeira linguagem de descrição de hardware padronizada pela IEEE.

A estrutura de um código VHDL consiste em três partes principais: declaração de bibliotecas/pacotes, entidade e arquitetura. Na primeira parte, são listadas as biblio-

tecas e pacotes necessários para o projeto. As bibliotecas padrão incluem a 'std' e a 'work'. A entidade, que é a interface do sistema, descreve as entradas e saídas e é dividida em duas partes: parâmetros e conexões. Os parâmetros são valores constantes, como a largura de um barramento, que são declarados como genéricos. As conexões, por sua vez, definem a transferência de informações e correspondem aos pinos de entrada e saída do circuito. Já a arquitetura é a parte principal do sistema, onde o circuito é descrito. Nessa seção, são definidas as atribuições, operações lógicas e aritméticas, comparações, entre outros. Há também uma parte declarativa da sintaxe, que apresenta uma ampla variedade de declarações possíveis.

Dessa forma, circuitos digitais para processamento de sinais em tempo real são muito utilizados em sistemas de comunicações sem fio. Um exemplo de aplicação são os DPDs para transmissores sem fio. Os DPDs são baseados em operações matemáticas que envolvem uma grande quantidade de somas, produtos e tabelas de busca. Devido às rigorosas exigências de frequência de operação, torna-se fundamental a paralelização das operações necessárias. Nesse contexto, as FPGAs são uma alternativa viável para a implementação de circuitos DPDs.

4 Metodologia

Como mencionado anteriormente, este trabalho tem como objetivo construir e validar um código em linguagem de descrição de hardware para processar em tempo real as características de transferência de um amplificador, reproduzindo suas não linearidades e efeitos de memória. O projeto foi dividido em três etapas principais:

- Estudo do DPD;
- Modelagem em software;
- Implementação em FPGA;

4.1 Estudo dos DPDs

A etapa consistiu no estudo dos DPDs, onde foi feito todo o levantamento sobre os tipos de modelagem dos DPDs. O objetivo deste estudo é entender as diferentes abordagens de modelagem, avaliar seus desempenhos e identificar a mais adequada para a aplicação em amplificadores de potência.

4.2 Implementação em software

Nesta etapa, foi realizada a implementação do modelo DPD em software, utilizando a linguagem de programação Python. Esta linguagem é amigável e amplamente difundida na comunidade acadêmica.

Para essa modelagem, foram coletados sinais de entrada e saída de um amplificador de potência classe AB, que utiliza um HEMT fabricado com tecnologia GaN. O amplificador foi excitado por um sinal portador de frequência de 900 MHz, modulado por um sinal de envelope WCDMA 3GPP com aproximadamente 3,84 MHz de largura de banda. Os dados de entrada e saída do amplificador de potência foram medidos usando um VSA Rohde & Schwarz FSQ com uma taxa de amostragem de 61,44 MHz, conforme disponível em (BONFIM; LIMA, 2016).

Em seguida, realizou-se o cálculo da estimativa do sinal utilizando números com vírgula fixa. Para verificar a precisão dessa estimativa em relação ao sinal original, calculou-se o NMSE. Para essa validação, os dados foram inicialmente divididos em conjuntos de extração e validação, que são dados utilizados para fazer o cálculo do modelo e para fazer a validação do modelo calculado, respectivamente. A matriz de confusão foi calculada com os dados de extração. Esse cálculo é essencial para a extração dos coeficientes do polinômio de memória. Após a extração dos coeficientes, calculou-se o modelo do PA, que foi então validado com os dados de validação. O

NMSE obtido para um polinômio de 2° grau com uma amostra memorizada foi de - 23,57 dB.

Em seguida, o algoritmo foi ajustado para operar com números em vírgula fixa e o número total de bits foi reajustado para atingir a menor resolução possível, buscando o menor NMSE simulado. Por se tratar de um cálculo em vírgula fixa, fez-se necessário uma readequação do resultado obtido entre cada multiplicação de forma a manter a resolução inicial.

4.3 Implementação em FPGA

Essa etapa consiste na implementação do DPD em FPGA. Para isso, é necessário realizar paralelizações nas operações aritméticas. A Figura 6 ilustra como esse processo está dividido entre cada ciclo de clock. A cada ciclo, duas operações são realizadas em paralelo: o sinal atual é elevado ao quadrado e registrado, enquanto ocorre o somatório do produto entre os sinais do mesmo instante de tempo e seus respectivos coeficientes. Esse processo ocorre P vezes para os P graus do polinômio de memória. Portanto, a saída do DPD é incompleta para os primeiros P períodos de clock, pois, nesses primeiros ciclos, realiza-se o cálculo com base em entradas de sinais anteriores que ainda não ocorreram, resultando em uma saída incompleta.

Fonte: Autor

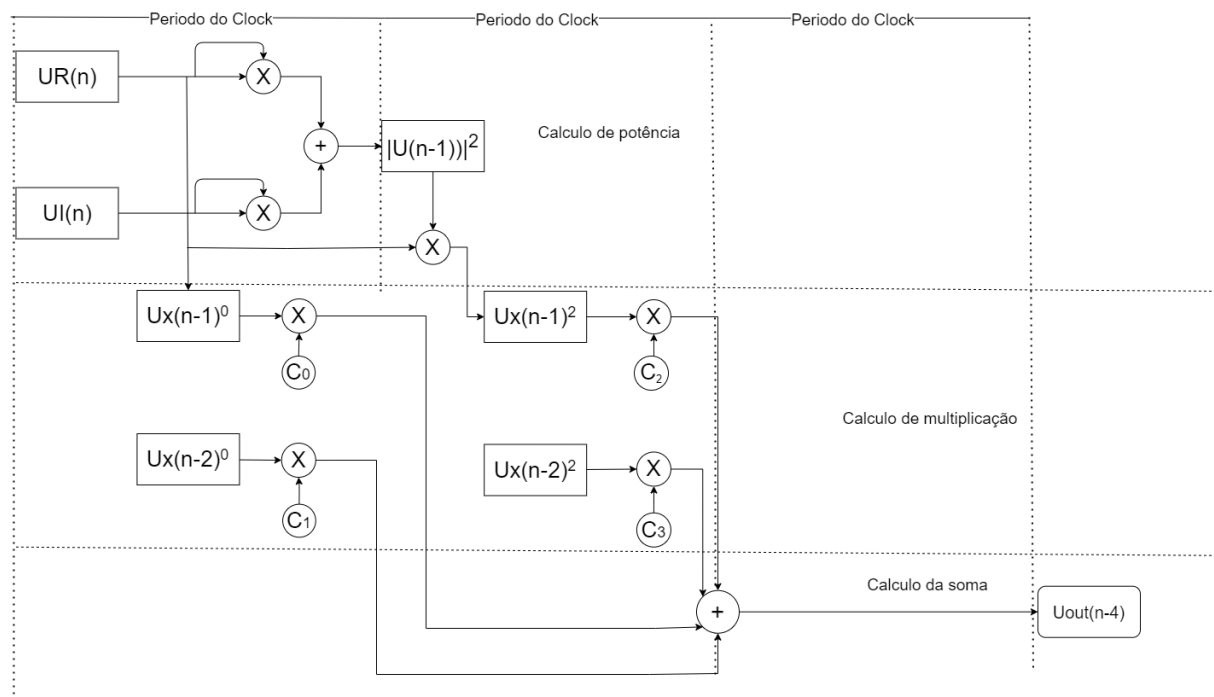


Figura 6 – Processo de cálculo da saída

4.4 Design e Validação do Circuito Lógico

O design da síntese lógica do circuito DPD foi realizado seguindo o fluxo VLSI, utilizando a tecnologia BiCMOS 130 nm 8HP. O processo abrangeu desde a descrição em alto nível até a validação final do circuito.

Inicialmente, a descrição em VHDL foi estruturada com base em um modelo comportamental que prioriza a eficiência computacional e a paralelização das operações. Essa abordagem assegura que o design inicial seja compatível com as restrições impostas pelas etapas subsequentes de síntese e layout.

A síntese lógica foi conduzida na ferramenta Cadence Genus, utilizando células padrão otimizadas para a tecnologia alvo. Durante essa etapa, foram exploradas configurações alternativas de pipeline e estratégias de paralelização para minimizar atrasos críticos e melhorar a taxa de transferência.

Além disso, relatórios detalhados contendo informações sobre área ocupada, consumo de energia e número de células lógicas utilizadas foram gerados ao final da síntese lógica. Esses relatórios serviram como base para avaliar a eficiência do circuito e orientar ajustes nas etapas subsequentes, garantindo que o design final atendesse às especificações do projeto de forma otimizada.

Após a síntese, os arquivos Verilog e SDF gerados foram analisados em simulações pós-síntese com a ferramenta Cadence NCLaunch. A principal preocupação foi verificar o impacto do atraso nas operações críticas do DPD, validando o funcionamento lógico sob as condições especificadas de temporização.

5 Resultados

Como mencionado anteriormente, o trabalho foi dividido em três etapas: a primeira etapa consistiu no estudo dos DPDs e dos métodos de modelagem associados; a segunda etapa envolveu a implementação dessa modelagem em software, utilizando Python; e a terceira etapa foi dedicada à implementação do modelo de DPD escolhido em hardware, empregando a linguagem VHDL. Neste capítulo são exibidos os resultados da etapa 2 e 3 já que a etapa 1 consiste no estudo dos DPDs e nos tipos de modelagens.

5.1 Modelagem do PA

Para fazer a modelagem em software foi utilizada a linguagem de programação Python. Para isso, separou-se os dados citados na seção 4.2, em dados de extração e dados de validação, os quais são utilizados para extração dos coeficientes do modelo do MP e validação do modelo encontrado, respectivamente. Para fazer a validação do modelo utilizou-se a métrica do NMSE, que consiste em calcular o erro médio quadrado do valor medido pelo VSA (Analisador de Sinal Vetorial) para o valor calculado pelo modelo. Portanto, quanto menor o NMSE mais fiel é o modelo do PA. Nesta etapa obteve-se um NMSE de -23.57 dB, para cálculos em vírgula flutuante, onde o resultado está presente no gráfico da figura 7.

Fonte: Autor

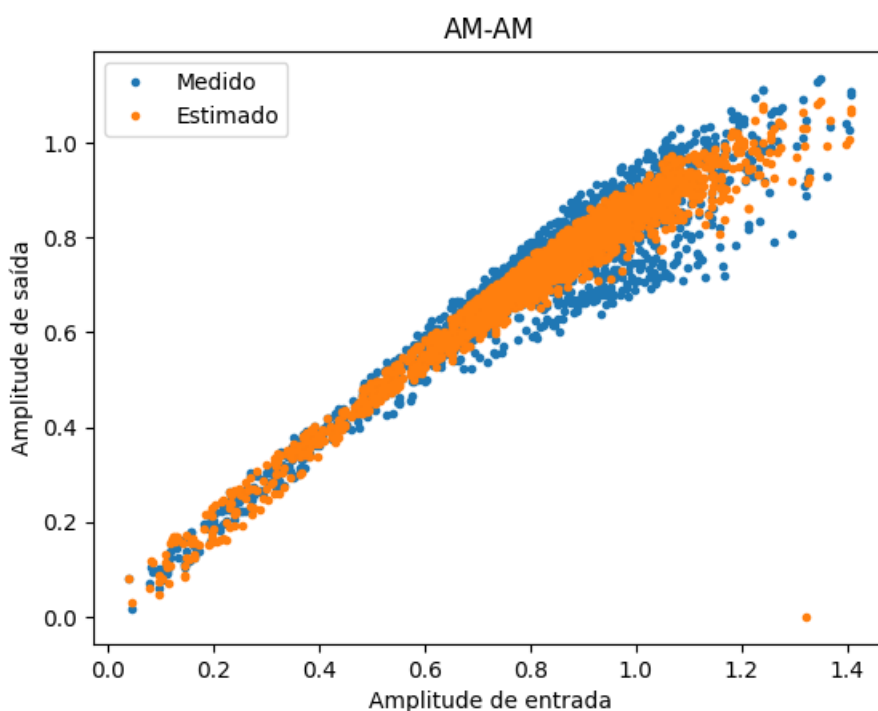


Figura 7 – Modelo do PA em vírgula flutuante

5.2 Definição do número de bits

Após concluída a modelagem matemática, realizou-se a modelagem do PA para então ser feito o levantamento da quantidade de bits necessários para a implementação do DPD em hardware minimizando os erros de quantização. Para isso foi necessário refazer a extração dos coeficientes, mas desta vez com os dados normalizados para valores de 0 a 2^{bits} . O resultado desse levantamento está presente no gráfico na figura 8.

Fonte: Autor

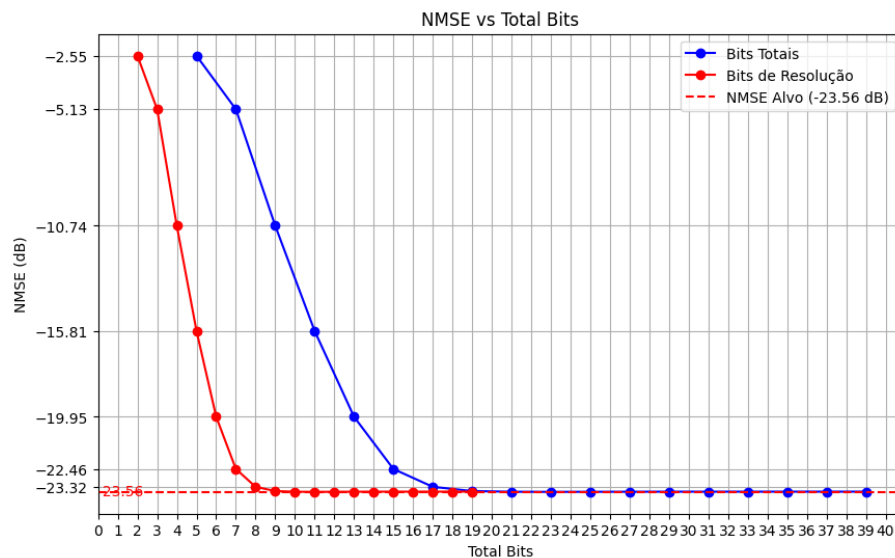


Figura 8 – Gráfico Número de bits x NMSE

Neste gráfico observa-se duas curvas, a curva em azul apresenta a quantidade total de bits contando com os bits de overflow necessárias para as operações de multiplicação, enquanto a curva em vermelho representa a quantidade de bits de resolução do sinal. Analisando este gráfico observou-se que não existem ganhos significativos no erro a partir de 8 bits, portanto foi feita a modelagem do PA utilizando uma resolução de 8 bits o resultado alcançado está ilustrado pela figura 9.

Fonte: Autor

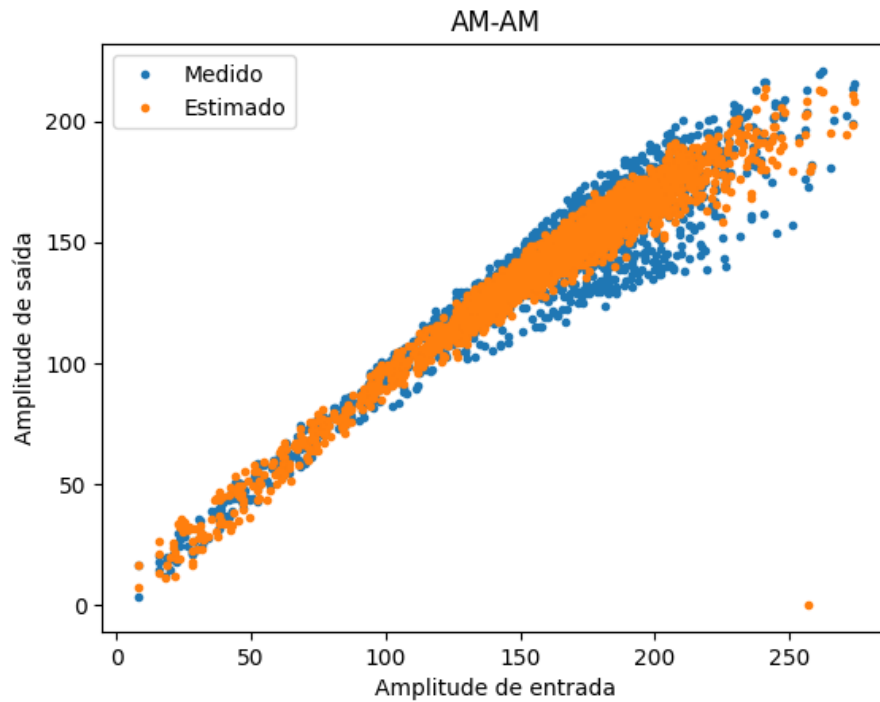


Figura 9 – Modelo do PA em vírgula fixa

5.3 Modelagem do DPD

A partir dos resultados obtidos foi possível fazer a modelagem do DPD, para isso foi feito o mesmo processo de modelagem do PA, porém para alcançar a característica de transferência inversa do PA foi invertido a ordem dos dados de entrada e saída para extração dos coeficientes do DPD. O resultado desta modelagem está ilustrado pela figura 10 a seguir.

Fonte: Autor

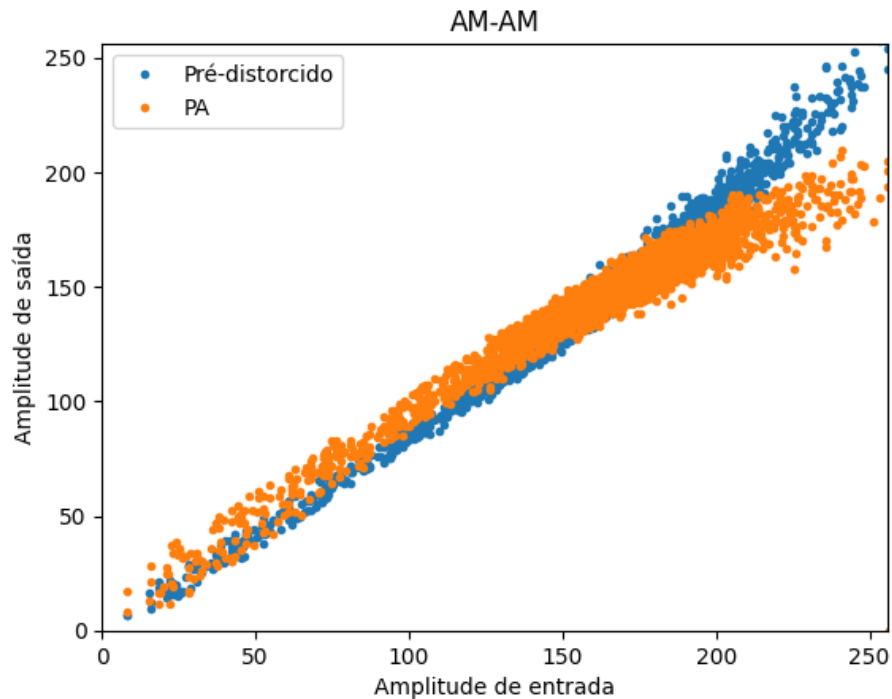


Figura 10 – Modelo do DPD em vírgula fixa

5.4 Implementação do DPD em FPGA

E por fim está sendo desenvolvido a implementação do código em VHDL para FPGA. Para que essa arquitetura de hardware apresentasse uma boa performance, todas as operações aritméticas (soma e multiplicação) são realizadas de forma síncrona. Então foi necessário dividir cada uma em processos distintos. A saída de um processo alimenta um *buffer*, que serve como entrada para o próximo processo. A Figura 11 ilustra essa arquitetura de maneira simplificada.

Fonte: Autor

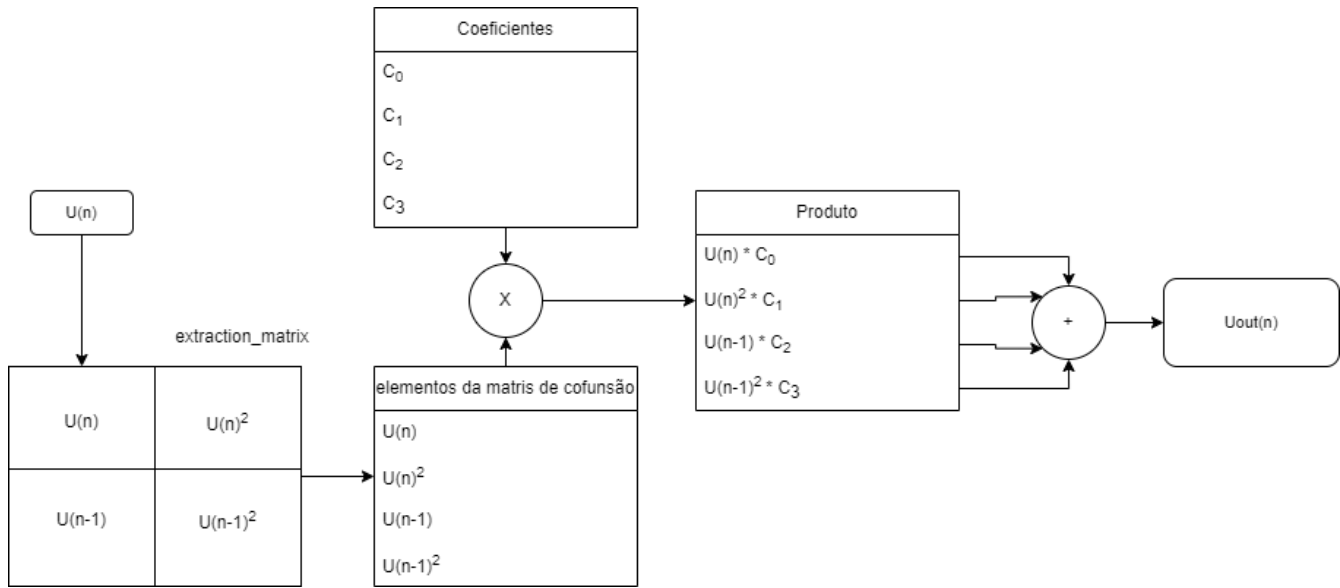


Figura 11 – Processo de cálculo da saída

No primeiro ciclo de clock, o sinal de entrada é registrado e, em seguida, elevado ao quadrado em P^2 graus do polinômio. Após essa etapa, o resultado é adicionado a um buffer de matriz de extração, que processa todos os sinais da amostra. Por fim, cada valor é multiplicado pelos seus respectivos coeficientes e somado, compondo o sinal de saída.

Essa descrição de hardware foi implementada FPGA Virtex5 XC5VLX50T, operando a uma frequência de 62,5 MHz, cujo os recursos lógicos estão sendo mostrados na tabela 1 a seguir.

Recursos	Quantidade	Percentual
Registradores	150	1%
LUTs	692	2%
DSP48E	4	8%

Tabela 1 – Utilização dos recursos do FPGA no projeto analisado.

Para fazer essa simulação foi utilizado um *testbench* com as mesmas entradas simuladas no python. Essa simulação foi feita no Xilinx ISE cujo o resultado está ilustrado na Figura 12.

O *testbench* empregado nesta simulação gera um arquivo de texto contendo os sinais de saída, permitindo a validação desses sinais calculados pela FPGA simulados no ISE em comparação com os resultados calculados em Python. A diferença entre os sinais foi avaliada utilizando a métrica NMSE (-16,77 dB). A figura 13 ilustra o resultado dessa comparação.

² P representa o grau máximo do polinômio utilizado no modelo matemático, que define o número de termos não lineares considerados no processamento do sinal.

Fonte: Autor

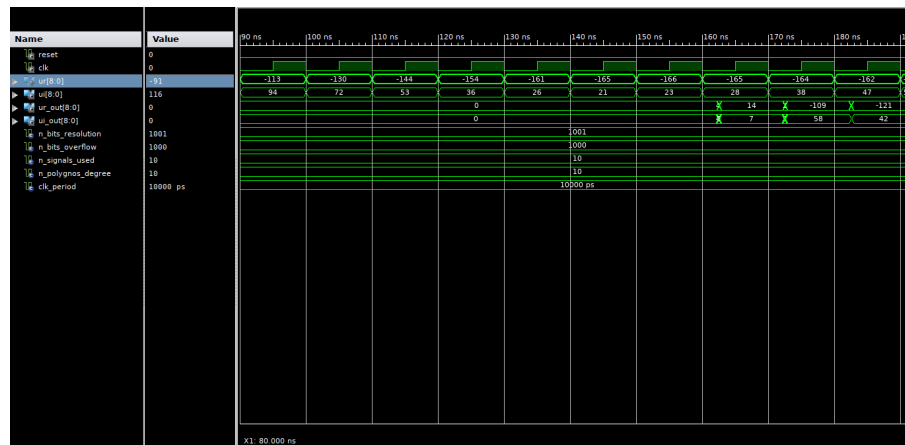


Figura 12 – Simulação ISE

Fonte: Autor

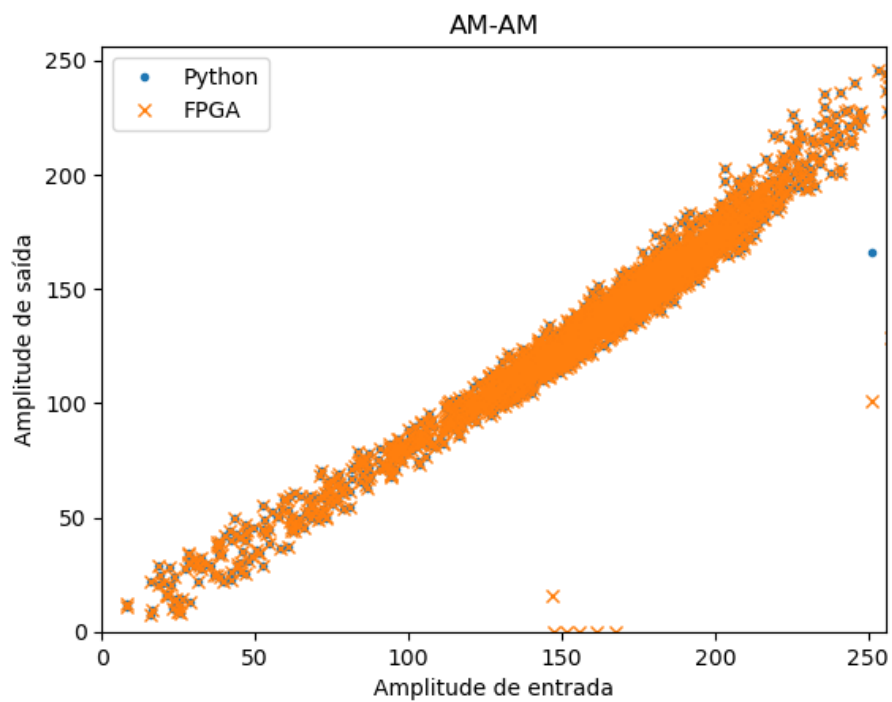


Figura 13 – Simulação FPGA

5.5 Síntese lógica

Por fim foi feito a síntese lógica do circuito e a simulação pós síntese lógica. A primeira parte consistiu na síntese lógica no Genus cujo o resultado está disponível na Figura 14. Já os relatórios de consumo, área e afins estão disponíveis na tabela 2.

Fonte: Autor

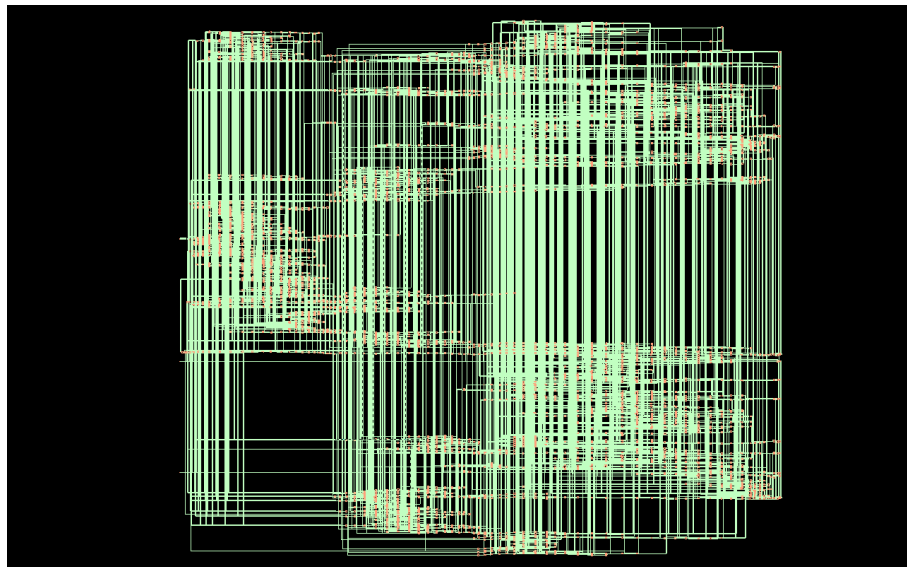


Figura 14 – Circuito lógico

Recursos	Valor
Células lógica	1567
Consumo	1.6 mW
Área	28116 μm^2

Tabela 2 – Utilização dos recursos de Células Lógicas.

Esse circuito foi simulado utilizando o mesmo *testbench* empregado na etapa anterior de implementação na FPGA, porém desta vez no ambiente NcLaunch do Cadence, conforme ilustrado na figura 15. A simulação gerou um arquivo de texto com os sinais de saída, permitindo a comparação e validação dos resultados calculados em Python e simulados na FPGA pelo ISE. A métrica NMSE também foi aplicada nessa simulação para validar os dados obtidos no NcLaunch em relação aos calculados em Python, resultando em um NMSE de -17,53 dB. A figura 16 apresenta o resultado dessa comparação.

Esse circuito foi simulado atuando a uma taxa de operação de 33,34 MHz, ou seja, a síntese lógica apresentou um desempenho pior que o apresentado pela FPGA.

Fonte: Autor

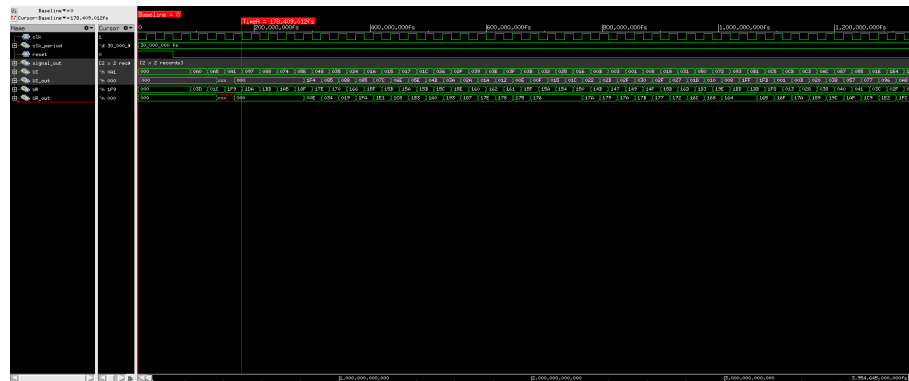


Figura 15 – Simulação NcLaunch

Fonte: Autor

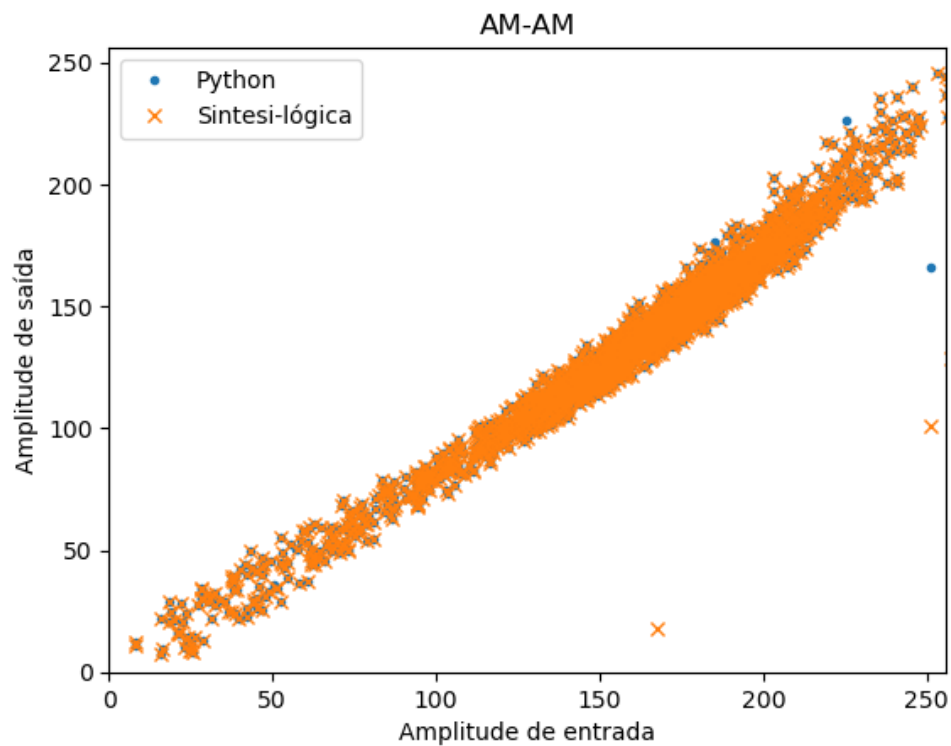


Figura 16 – Comparação dos sinais calculados em Python e no NcLaunch

6 Conclusão

A evolução dos sistemas de comunicação sem fio tem promovido a implementação de diversos serviços móveis, tornando essencial que esses sistemas operem com máxima eficiência. Nesse cenário, a implementação de um DPD em cascata com o PA surge como uma alternativa de baixo custo e interessante para melhorar o desempenho desses sistemas. O objetivo deste trabalho de iniciação científica foi a implementação e validação de um código em linguagem de descrição de hardware capaz de processar em tempo real as características de transferência de um amplificador. O código deve ser capaz de reproduzir não linearidades e efeitos de memória. Além disso, o código deve ocupar a quantidade mínima de recursos lógicos do circuito digital que irá sintetizá-lo, buscando reduzir o consumo de energia. Para isso, o projeto foi dividido em três etapas: estudo do DPD e da modelagem matemática, modelagem do DPD em software e implementação do DPD em hardware. Sendo assim a primeira etapa de desenvolvimento do projeto foi a modelagem do PA em vírgula flutuante, utilizando o método do MP, para fazer essa modelagem utilizou-se um polinômio de 2º grau com uma amostra de memória, para fazer a validação dessa modelagem utilizou-se a métrica do NMSE. Nesta etapa obteve-se um NMSE de -23,57 dB. A próxima etapa consiste em otimizar a quantidade de células lógicas utilizadas no processo limitando o número de bits utilizados. Nesta etapa, constatou-se que a partir de 8 bits não houve uma melhora significativa no NMSE. Dessa forma, essa foi a resolução em bits utilizada para a amostragem dos sinais. E por fim foi feito a modelagem do DPD em software o qual apresentou um comportamento inverso em relação o do PA, assim satisfazendo as necessidades. Iniciou-se a etapa de implementação do modelo em hardware do circuito em VHDL e validação em FPGA Virtex5 XC5VLX50T, que utilizou um total de 150 registradores, 692 LUTs e 4 unidades DSP48E, operando a uma frequência de 62,5 MHz. Em seguida seguiu-se para a etapa de síntese lógica a qual resultou em um circuito com 1567 células lógica, com uma area total de $28116 \mu m^2$ e um consumo de energia de 1,6 mW, atuando a uma taxa de operação de 33,34 MHz. Conclui-se, portanto, que o projeto alcançou os resultados esperados, com uma implementação eficaz do DPD em hardware, exceto pela taxa de operação da síntese lógica esperar-se uma melhor performance, o que não foi observado nas simulações.

7 Referências

BONFIM, E. J.; LIMA, E. G. D. *A Modified Two Dimensional Volterra-Based Series for the Low-Pass Equivalent Behavioral Modeling of RF Power Amplifiers*. 2016. 27-35 p.

CHAVEZ, J. H. *Estudo comparativo entre as arquiteturas de identificação de pré-distorcedores digitais através das aprendizagens direta e indireta*. 2018.

CRIPPS, S. *RF Power Amplifiers for Wireless Communications*. [S.l.: s.n.], 2006.

JOHN, E. *Modelagem comportamental de amplificadores de potência de radiofrequência usando termos unidimensionais e bidimensionais de séries de Volterra*. 2016.

KENINGTON, P. *High Linearity RF Amplifier Design*. [S.l.: s.n.], 2000.

LIMA, E. G. de; GHIONE, G. *Behavioral modeling and digital base-band predistortion of RF power amplifiers*. 2009.

PEDRONI, V. *Eletrônica Digital e VHDL*. [S.l.: s.n.], 2010.

SCHUARTZ, L.; LIMA, E. *Polinômios com Memória de Complexidade Reduzida e sua Aplicação na Pré-distorção Digital de Amplificadores de Potência*. 2017.