

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**GUILHERME CAVICHIOLO MOREIRA BARBOSA**

**RECUPERAÇÃO DE SINAL DE CLOCK PARA SISTEMAS DE COMUNICAÇÃO  
POR LUZ VISÍVEL**

**CURITIBA**

**2026**

**GUILHERME CAVICHIOLO MOREIRA BARBOSA**

**RECUPERAÇÃO DE SINAL DE CLOCK PARA SISTEMAS DE COMUNICAÇÃO  
POR LUZ VISÍVEL**

**Clock Recovery for Visible Light Communication Systems**

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Elétrica do curso de Engenharia Elétrica da Universidade Tecnológica Federal do Paraná (UTFPR).  
Orientador: Prof. Dr. Juan Camilo Castellanos Rodriguez  
Coorientador: Prof. Dr. Paulo De Tarso Neves Junior

**CURITIBA**

**2026**



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**GUILHERME CAVICHIOLO MOREIRA BARBOSA**

**RECUPERAÇÃO DE SINAL DE CLOCK PARA SISTEMAS DE COMUNICAÇÃO  
POR LUZ VISÍVEL**

Trabalho de conclusão de curso de graduação  
apresentado como requisito para obtenção do  
título de Bacharel em Engenharia Elétrica do  
curso de Engenharia Elétrica da Universidade  
Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 25/Junho/2026

---

Juan Camilo Castellanos Rodriguez  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Paulo de Tarso Neves Júnior  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Alexandre Pohl  
Doutorado  
Universidade Tecnológica Federal do Paraná

**CURITIBA**  
**2026**

## RESUMO

A comunicação por luz visível (Visible Light Communication – VLC) tem se destacado como uma alternativa promissora às tecnologias de comunicação por radiofrequência, especialmente em ambientes internos que demandam elevada largura de banda, imunidade a interferências eletromagnéticas e integração com sistemas de iluminação existentes. Nesse contexto, a recuperação de sincronismo representa uma etapa fundamental para a correta reconstrução dos dados transmitidos. Este trabalho apresenta o desenvolvimento e a validação de um componente de clock recovery compatível com sistemas VLC baseados na norma IEEE 802.15.7, empregando modulação VPPM (Variable Pulse Position Modulation) e implementação integralmente digital em FPGA. A arquitetura proposta consiste em um All-Digital Phase-Locked Loop (ADPLL) descrito em VHDL para a plataforma FPGA DE10-Nano, utilizando um oscilador controlado numericamente (NCO), detector de fase e controlador PI digital. O sistema foi projetado para recuperar um sinal de clock de 3,75 MHz a partir de um trem de bits VPPM, mantendo uma defasagem próxima de  $90^\circ$  em relação às transições do sinal recebido, de forma a maximizar a margem temporal de amostragem. A validação foi realizada por meio de simulações comportamentais e gate level no Quartus Prime e ModelSim, seguidas de testes experimentais em bancada utilizando gerador de funções, osciloscópio digital e analisador lógico Signal Tap. Os resultados demonstraram que o componente ocupa menos de 1% dos recursos lógicos da FPGA e é capaz de recuperar um clock com jitter pico a pico de 7,8 ns a partir de um sinal de entrada com jitter de 44,7 ns. Os ensaios de resposta ao degrau indicaram tempo típico de subida inferior a  $3 \mu s$  e tempo de acomodação geralmente inferior a  $50 \mu s$ . No teste de taxa de erro de bits, não foram observados erros durante a decodificação de mais de 24 mil bits transmitidos. Conclui-se que a arquitetura ADPLL proposta atende aos requisitos estabelecidos para sistemas VLC compatíveis com a IEEE 802.15.7, contribuindo para preencher uma lacuna identificada na literatura relacionada à recuperação de clock para sinais VPPM em aplicações de comunicação por luz visível.

Palavras-chave: comunicação por luz visível; clock recovery; phase locked loop; modulação por posição de pulso variável.

## ABSTRACT

Visible Light Communication (VLC) has emerged as a promising alternative to conventional radio frequency-based communication technologies, offering high bandwidth, immunity to electromagnetic interference, and the possibility of integration with existing lighting infrastructures. In this context, clock recovery plays a fundamental role in ensuring the correct reconstruction of transmitted data, particularly in systems compliant with the IEEE 802.15.7 standard. This work presents the development and validation of a clock recovery component based on an All-Digital Phase-Locked Loop (ADPLL), entirely implemented in VHDL for the DE10-Nano FPGA platform, intended for recovering Variable Pulse Position Modulation (VPPM) signals operating at 3.75 MHz, according to the requirements of the PHY II layer defined by IEEE 802.15.7. The proposed architecture employs a digital phase detector, a proportional-integral (PI) controller, and a numerically controlled oscillator (NCO) to synchronize the recovered clock with the transitions of the received VPPM bitstream, positioning the sampling instants approximately  $90^\circ$  away from signal transitions. The system was initially validated through behavioral and gate-level simulations using Quartus Prime and ModelSim, demonstrating control-loop convergence and successful data recovery. Synthesis results indicated a logic resource utilization below 1% of the available FPGA capacity, highlighting the low implementation cost of the proposed solution. Subsequently, experimental tests were conducted to evaluate signal integrity, transient response, and bit error rate performance. Eye diagram analysis showed that the ADPLL was capable of recovering a clock signal with only 7.8 ns peak-to-peak jitter from an input signal exhibiting 44.7 ns peak-to-peak jitter. Step-response measurements revealed a typical rise time below  $3 \mu s$  and a settling time generally below  $50 \mu s$ . In bit error rate tests, no errors were observed during the decoding of more than 24,000 transmitted bits, indicating reliable operation of the proposed component. The obtained results demonstrate the feasibility of a fully digital ADPLL architecture for clock recovery in VPPM-based VLC systems. Furthermore, this work contributes to addressing a gap identified in the literature, as few studies have specifically focused on clock recovery solutions for VPPM signals compliant with the IEEE 802.15.7 standard.

Keywords: visible light communication; clock recovery; phase locked loop; variable pulse position modulation.

## LISTA DE FIGURAS

Figura 1 – Diagrama de blocos dos sinais de entrada e saída de um circuito de <i>Clock and Data Recovery</i> .....	12
Figura 2 – Diagrama de blocos da topologia Phase Locked Loop .....	13
Figura 3 – Diagrama de blocos da topologia All Digital Phase Locked Loop .....	14
Figura 4 – Diagrama de blocos da topologia Delay Locked Loop .....	15
Figura 5 – Diagrama de blocos da topologia de Oversampling .....	16
Figura 6 – Diagrama de blocos da topologia de Phase Interpolation .....	17
Figura 7 – Natureza eletrônica dos componentes de clock recovery da revisão bibliográfica .....	20
Figura 8 – Conceitos base das arquiteturas dos componentes de clock recovery da revisão bibliográfica .....	21
Figura 9 – Tipos de modulação dos sinais de dados dos componentes de clock recovery da revisão bibliográfica .....	21
Figura 10 – Faixa de frequências dos sinais de dados dos componentes de clock recovery da revisão bibliográfica .....	22
Figura 11 – Diagrama caixa preta do componente ADPLL .....	23
Figura 12 – Diagrama caixa preta do componente ADPLL .....	24
Figura 13 – Diagrama caixa preta do PLL de propriedade intelectual da Intel .....	25
Figura 14 – Diagrama caixa preta do oscilador controlado numericamente .....	26
Figura 15 – Diagrama caixa preta do bloco de detecção de fase .....	27
Figura 16 – Diagrama caixa preta do bloco de operações iniciais do controlador PI ..	28
Figura 17 – Diagrama caixa preta do bloco de operações finais do controlador PI ....	28
Figura 18 – Diagrama caixa preta do bloco de recuperação de dados .....	29
Figura 19 – Relatório de Compilação do código VHDL no Software Quartus Prime 18.1 .....	30
Figura 20 – Captura de tela da simulação gate level realizada do componente ADPLL	31
Figura 21 – Captura de tela da simulação gate level realizada do oscilador controlado numericamente .....	32
Figura 22 – Diagrama de blocos do teste de diagrama de olho .....	34
Figura 23 – Foto da montagem física do teste de diagrama de olho .....	34
Figura 24 – Diagrama de olho centrado em bordas do trem de bits VPPM de entrada.	35
Figura 25 – Diagrama de olho centrado em bordas do sinal de <i>clock</i> recuperado .....	35
Figura 26 – Diagrama de blocos do teste de resposta ao degrau .....	36
Figura 27 – Foto da montagem física do teste de resposta ao degrau .....	36
Figura 28 – Curva de resposta ao degrau unitário quando sinal de <i>clock</i> recuperado começa com 4,6° de fase em relação ao trem de bits VPPM .....	37
Figura 29 – Diagrama de blocos do teste de taxa de erro de bit .....	38
Figura 30 – Foto da montagem física do teste de taxa de erro de bit .....	39
Figura 31 – Interface MATLAB para decodificação do trem de bits VPPM (em preto) com base no sinal de <i>clock</i> recuperado (em azul), visão ampliada .....	40
Figura 32 – Interface MATLAB para decodificação do trem de bits VPPM (em preto) com base no sinal de <i>clock</i> recuperado (em azul), visão geral .....	40

## LISTA DE TABELAS

<b>Tabela 1 – Parâmetros da resposta ao degrau medidos conforme variação de diferença de fase inicial.....</b>	<b>38</b>
--	-----------

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>8</b>
<b>1.1</b>	<b>A Comunicação por Luz Visível</b> .....	<b>8</b>
<b>1.2</b>	<b>Oportunidades da Tecnologia VLC</b> .....	<b>8</b>
<b>1.3</b>	<b>Desafios da tecnologia VLC</b> .....	<b>9</b>
<b>1.4</b>	<b>Delimitação de Escopo</b> .....	<b>10</b>
<b>1.5</b>	<b>Objetivos do Trabalho</b> .....	<b>11</b>
1.5.1	Objetivo Geral .....	11
1.5.2	Objetivos Específicos .....	11
<b>1.6</b>	<b>Metodologia e Classificação da Pesquisa</b> .....	<b>11</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b> .....	<b>12</b>
<b>2.1</b>	<b>Clock Recovery</b> .....	<b>12</b>
2.1.1	Phase Locked Loop .....	13
2.1.2	All Digital Phase Locked Loop .....	14
2.1.3	Delay Locked Loop .....	15
2.1.4	Oversampling .....	16
2.1.5	Phase Interpolation .....	17
<b>2.2</b>	<b>Estado da Arte</b> .....	<b>18</b>
2.2.1	Avanço Tecnológico .....	18
2.2.2	Análise Qualitativa .....	20
<b>3</b>	<b>DESCRIÇÃO DO SISTEMA</b> .....	<b>23</b>
<b>3.1</b>	<b>Funcionamento geral do componente ADPLL</b> .....	<b>23</b>
<b>3.2</b>	<b>Funcionamento do PLL de propriedade intelectual da Intel</b> .....	<b>25</b>
<b>3.3</b>	<b>Funcionamento do oscilador controlado numericamente</b> .....	<b>26</b>
<b>3.4</b>	<b>Funcionamento do bloco de detecção de fase</b> .....	<b>27</b>
<b>3.5</b>	<b>Funcionamento do controlador PI</b> .....	<b>27</b>
<b>3.6</b>	<b>Funcionamento do bloco de recuperação de dados</b> .....	<b>29</b>
<b>3.7</b>	<b>Síntese e resultados de simulação</b> .....	<b>30</b>
<b>4</b>	<b>PROCEDIMENTOS E RESULTADOS EXPERIMENTAIS PRÁTICOS</b> .....	<b>33</b>
<b>4.1</b>	<b>Análise de integridade de sinal através do diagrama de olho</b> .....	<b>34</b>
<b>4.2</b>	<b>Caracterização da curva de resposta a degrau</b> .....	<b>36</b>
<b>4.3</b>	<b>Teste de taxa de erro de bits</b> .....	<b>38</b>
<b>5</b>	<b>CONCLUSÃO</b> .....	<b>41</b>
	<b>REFERÊNCIAS</b> .....	<b>44</b>
	<b>APÊNDICE A – Tabela de Categorização da Revisão Bibliográfica</b> .....	<b>49</b>
	<b>APÊNDICE B – Código VHDL do componente ADPLL</b> .....	<b>51</b>
	<b>APÊNDICE C – Passo a passo para configuração básica de um componente PLL pela seção de propriedade intelectual Intel dentro do software Quartus Prime</b> .....	<b>59</b>
	<b>APÊNDICE D – Passo a passo para configuração básica do Signal Tap Logic Analyzer dentro do software Quartus Prime</b> .....	<b>64</b>
	<b>APÊNDICE E – Código VHDL do arquivo testbench</b> .....	<b>69</b>

## 1 INTRODUÇÃO

### 1.1 A Comunicação por Luz Visível

Através de transmissões imperceptíveis aos olhos humanos, sistemas de comunicação por luz visível (VLC; do inglês *Visible Light Communication*) podem operar na ordem de megabits por segundo a metros de distância ou até gigabits por segundo a centímetros de distância (Ricci; Caputo; Mucchi, 2023, p.1). Esse tipo de comunicação compreende faixas de frequências de 400 THz a 800 THz que até então são pouco exploradas pela área de telecomunicações e são isentas de regulamentação, oferecendo ampla largura de banda (Rehman *et al.*, 2019).

O ano de 2011 foi um marco excepcional por conta da primeira demonstração de um sistema *Light Fidelity* (LiFi). Harald Haas introduziu o termo LiFi na conferência de inovações TEDGlobal, enquanto apresentava seu sistema de comunicação por luz visível que alcançava uma taxa de transmissão de 10 megabits por segundo (Khandal; Jain, 2014). Esse avanço contribuiu com as definições do padrão IEEE 802.15.7 – também publicado em 2011 – que especifica os requisitos físicos (PHY) e de controle de acesso ao meio (MAC) para comunicação por luz visível. Através do levantamento feito por Barreira (2024, p.4), é possível inferir que o número de trabalhos publicados sobre VLC triplicou após somente 5 anos dessa data. Em 2011 haviam menos de 150 trabalhos publicados sobre comunicação por luz visível e, em 2016, esse número ultrapassava o marco de 450 publicações. Devido aos avanços relacionados à comunicação por luz visível serem recentes, há espaço para o desenvolvimento de novas tecnologias como sistemas VLC que podem ser integrados à instalação de iluminação já existentes de ambientes industriais ou domésticos.

### 1.2 Oportunidades da Tecnologia VLC

A aplicação de sistemas VLC em ambientes domésticos ou industriais demonstra potencial, sobretudo quando o sistema de comunicação possibilita integração com a infraestrutura de iluminação já instalada no local. Esse fator torna-se destaque devido a redução de mão de obra e custos relacionados a roteamento de cabos, conforme salientado pelo estudo de Mas-Machuca *et al.* (2022). O estudo compara diferentes redes de comunicação industriais com base em radiofrequência ou luz visível, indicando os sistemas mais vantajosos para contextos diversos. É estimado que infraestruturas de WiFi com ampla cobertura – máximo de 10 metros até o ponto de acesso mais próximo – possuam um *Total Cost of Ownership* (TCO) mais caro que uma rede de comunicação VLC. Por fim, o estudo indica que tecnologias de comunicação por luz visível apresentam taxas de transmissão que se equiparam as de sistemas WiFi.

O desenvolvimento da comunicação VLC também possibilita a redução de níveis de radiação em ambientes sensíveis a interferência eletromagnética. Pauli e Kamp (2019) explicam o caso do hospital de Perpignan, na França, que implantou um sistema de comunicações VLC em seu prédio. Após realizar medições, foi revelado que o hospital teve seus níveis de radiação re-

duzidos de 6000 mW/m para 250 mW/m. Apesar do nível de radiação recomendado pela *World Health Organization* (WHO) ser de até 620 mW/m, o hospital somente passou a seguir essa recomendação após substituir cerca de 700 roteadores de WiFi por um sistema de comunicação por luz visível.

### 1.3 Desafios da tecnologia VLC

Desenvolver *transceivers* e *drivers* de LED compatíveis com luminárias de aplicação comum é uma ideia bastante competitiva sob um aspecto financeiro. Porém, os LED's disponíveis no mercado tem pouca afinidade com o protocolo de comunicação VLC devido a natureza de seus fenômenos de luminescência. A luz base de LED's *high brightness* brancos em sua maioria é azul e é gerada através da eletroluminescência – um fenômeno que cessa quase imediatamente após a interrupção da corrente elétrica. Entretanto, para atingir a cor branca, também são empregados materiais fosforescentes que absorvem a luz azul e emitem luz amarela. O fenômeno de fosforescência por sua vez não cessa imediatamente, portanto a luminária continua a emitir luz por um período mesmo após a interrupção de corrente elétrica. Esse é um agravante para a comunicação VLC, visto que a frequência mínima de operação prevista pela norma IEEE 802.15.7 para equipamentos internos é 3,75 MHz (camada física PHY II, tabela 77). O perfil de fosforescência das luminárias comuns do mercado limita a resposta em frequência do sinal a partir de 2 ou 3 MHz (Albuquerque *et al.*, 2021).

Compreendido o contexto da saturação de sinal, é imprescindível empregar um componente de *clock recovery* robusto e confiável. Em um sistema de comunicações, o sinal de *clock* é responsável por definir os instantes em que cada bit deve ser lido, assegurando a correta interpretação dos dados recebidos. Porém, sistemas de comunicação óptica sem fio contam com somente um canal para a transmissão simultânea do sinal de dados e do sinal de *clock*. Esse único canal modula o fluxo luminoso de uma luminária na faixa de kHz a MHz para transmitir uma mensagem através de luz visível, ou seja, ondas eletromagnéticas na faixa de THz (Ghassemi; Abrams; Little, 2014). Durante a recepção, o sinal óptico é convertido para sinal elétrico e processado por circuitos de detecção de envelope. Nesse momento, as componentes em alta frequência (luz visível) são eliminadas, permitindo a regeneração do sinal base. Em sequência, os circuitos de *clock recovery* entram em ação, extraindo o sinal de temporização embutido para garantir a correta amostragem e reconstrução dos dados.

Por fim, manter o fluxo luminoso das luminárias constante e evitar oscilações de luz perceptíveis a olhos humanos, independentemente da sequência de bits transmitidos, é indispensável para sistemas VLC. A norma IEEE 802.15.7 introduz parâmetros máximos de cintilação: *maximum flickering-time period* (MFTP) e também apresenta estratégias para lidar com suas ocorrências. Adicionalmente, um recurso desejável para o sistema é poder ajustar a intensidade do sinal irradiado pelas luminárias conforme o ambiente de instalação. Controlar o fluxo luminoso torna o sistema mais econômico em ambientes escuros e aumenta sua confiabilidade em ambientes com iluminação natural.

## 1.4 Delimitação de Escopo

A começar pelo sistema VLC, este estudo adota as seguintes imposições para garantir conformidade com a IEEE 802.15.7 e compatibilidade com infraestruturas comuns de iluminação domésticas e industriais, conforme critérios apresentados na seção 1.3:

- Conformidade com especificações da camada física PHY II da IEEE 802.15.7;
- Frequência de sinal de clock de 3,75 Mhz;
- Continuidade do fluxo luminoso conforme parâmetros de MTFP;
- Uso de luminárias tubulares LED comerciais comuns de no mínimo 18 W e 1500 Lm.

A partir dessas premissas, é possível fazer inferências sobre o componente de *clock recovery* a ser desenvolvido. O padrão IEEE 802.15.7 define três tipos principais de modulação para comunicação por luz visível: OOK (*On-Off Keying*), VPPM (*Variable Pulse Position Modulation*) e CSK (*Color Shift Keying*), dos quais entende-se que o primeiro e o último estão além do escopo desse estudo. A modulação CSK é somente possível para sistemas que empregam luminárias LED capazes de controlar separadamente as cores vermelho, verde e azul. Ao combinar as cores em intensidade diferentes, o LED emite luz em 4, 8 ou até 16 cores diferentes, aumentando a capacidade de transmissão de dados. Porém, essas luminárias LED ainda são pouco comuns em instalações domésticas ou industriais. Quanto à modulação OOK, trens de pulso que contenham muitos bits 0 ou 1 em sequência podem causar cintilação, interrompendo a continuidade do fluxo luminoso.

Já a modulação VPPM destaca-se por sua simplicidade, taxa de transmissão de dados elevada e amplitude de pulso contínua. Apesar de ser descrita como somente uma comunicação de banda base (Lathi; Ding, 2009), ou estratégia de codificação de fonte (Haykin, 2006), esse conceito comumente conhecido por modulação representa os bits através da posição de pulsos. O bit 0 é representado por um pulso modulado no início do período do sinal de clock e o bit 1 pelo pulso no fim do período do sinal de clock. Dessa forma, a intensidade luminosa permanece a mesma independente do bit sendo transmitido, evitando oscilações de luz perceptíveis a olhos humanos. Adicionalmente, a largura dos pulsos - conhecida também como *duty cycle* - pode ser ajustada conforme a necessidade do sistema, colaborando com o ajuste de intensidade luminosa conforme o ambiente. Ultimamente, VPPM é a única dentre as modulações previstas pela camada física PHY II da IEEE 802.15.7 que prevê frequência de sinal de clock de 3,75 Mhz.

Segundo Razavi (2002), a qualidade do sinal de saída de um componente de *clock recovery* é medida por 3 fatores: frequência correspondente à taxa de dados do sinal, fase não nula em relação ao sinal de entrada e, por fim, propagação mínima de *jitter*. A começar pela propagação de *jitter*, define-se que a arquitetura do componente de *clock recovery* a ser desenvolvido por este estudo será digital e não agregará circuitos analógicos. Com essa decisão, espera-se que os aspectos construtivos do componente contribuam o mínimo possível com o

valor total de *jitter* propagado. Devido à largura de pulso reconfigurável do VPPM, entende-se que o componente deve realizar ajuste de fase conforme o *duty cycle* do sinal, evitando leituras durante, antes ou após a transição de estado dos pulsos. Afim de viabilizar a reconfiguração do componente, define-se a placa FPGA DE10 NANO como plataforma de desenvolvimento devido suas características digitais e reconfiguráveis. Por fim, o componente por padrão deve apresentar um sinal de *clock* de 3,75 Mhz em sua saída, adaptando sua frequência a variações leves conforme o sinal de entrada.

## 1.5 Objetivos do Trabalho

### 1.5.1 Objetivo Geral

Projetar um sistema de *clock recovery* para um sistema de comunicação por luz visível em ambientes industriais ou domésticos internos.

### 1.5.2 Objetivos Específicos

- Realizar a revisão bibliográfica sobre IEEE 802.15.7, VPPM e *clock recovery*;
- Elaborar circuito de *clock recovery* em VHDL para FPGA DE-10 NANO;
- Simular modulação e demodulação com o circuito de *clock recovery*;
- Validar circuito em bancada através de sinal VPPM com *duty cycle* em 50%;

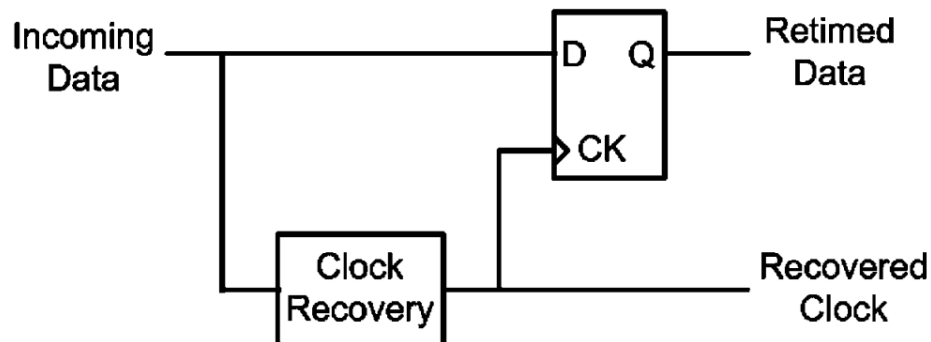
## 1.6 Metodologia e Classificação da Pesquisa

Segundo a taxonomia metodológica de Azevedo e Ensslin (2020), a pesquisa proposta se caracteriza como exploratória e prática, dado que busca aprofundar o conhecimento técnico sobre componentes de *clock recovery* aplicados à comunicação por luz visível, um campo ainda emergente. A lógica utilizada é dedutiva, partindo de referenciais teóricos consolidados sobre modulação VPPM, circuitos de *clock recovery* e o padrão IEEE 802.15.7, para então propor e validar empiricamente uma solução de hardware digital. O processo de investigação é conduzido por meio da coleta e análise de dados primários e quantitativos, obtidos em ambiente de simulação e em bancada de testes, o que caracteriza um método centrado em mensuração e validação objetiva de desempenho. O estudo possui um resultado aplicado, pois visa à construção e aperfeiçoamento de um circuito funcional para aplicações reais em sistemas VLC *indoor*. Por fim, os procedimentos técnicos adotados são experimentais, uma vez que envolvem a implementação prática e a observação controlada de um circuito reconfigurável desenvolvido em FPGA, permitindo iterar o projeto conforme os resultados obtidos.

## 2 REFERENCIAL TEÓRICO

### 2.1 Clock Recovery

Em seu livro "*CMOS Multichannel Single-Chip Receivers for Multi-Gigabit Optical Data Communications*", páginas 127 à 142, os autores Muller e Leblebici explicam que é necessário extrair informações de temporização para que receptores consigam interpretar corretamente os bits de um enlace serial, conforme esquematizado na figura 1. É comum não encontrar sinais de sincronização dedicados nesses sistemas, por essa razão é necessário recuperar o *clock* diretamente do próprio fluxo de dados, processo conhecido popularmente como *Clock and Data Recovery (CDR)*. Adicionalmente, menciona-se que o termo mais indicado para o processo completo é recuperação de *clock* e *retiming* de dados, já que a recuperação do *clock* e a recuperação dos dados são operações conceitualmente distintas. Embora muitas implementações as integrem juntas para melhorar o desempenho do circuito, é possível desenvolver circuitos separados e somente depois conectá-los em um sistema de comunicações.



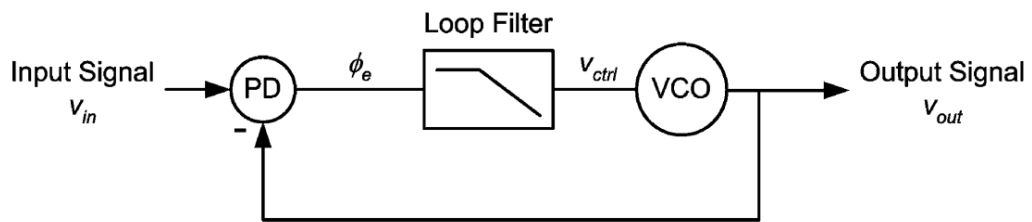
**Figura 1 – Diagrama de blocos dos sinais de entrada e saída de um circuito de *Clock and Data Recovery***

*Fonte: Muller e Leblebici (2007)*

Muller e Leblebici (2007) indicam que, historicamente, a recuperação de *clock* era realizada com estruturas baseadas em filtros ressonantes, utilizando componentes passivos discretos ou dispositivos ressonantes como filtros SAW (*surface acoustic wave*). Com a evolução para sistemas altamente integrados e exigências de redução de tamanho e componentes externos, o uso desses elementos *off-chip* tornou-se indesejável. Segundo os autores, as topologias baseadas em PLL passaram a dominar o mercado de comunicação óptica de longo alcance — onde as tecnologias de fibra ótica se estabeleceram primeiro. Isso se deve à sua capacidade de integração monolítica e ao fato de exigirem pouquíssimos componentes externos, geralmente apenas um capacitor do filtro de malha. Os autores afirmam que, embora existam outras arquiteturas, o PLL continua sendo a topologia de recuperação de *clock* mais utilizada e mais bem compreendida.

### 2.1.1 Phase Locked Loop

Muller e Leblebici (2007) descrevem os três blocos principais de uma topologia PLL: o detector de fase (PD), o filtro de malha (LF) e o oscilador controlado por tensão (VCO). O PD mede o erro de fase entre o sinal de entrada e o sinal gerado pelo VCO, conforme esquematizado na figura 2. Esse erro é integrado pelo filtro de malha de característica passa-baixa, produzindo um sinal de controle que ajusta a frequência do VCO, reduzindo gradualmente o erro de fase e levando o sistema ao travamento (lock).



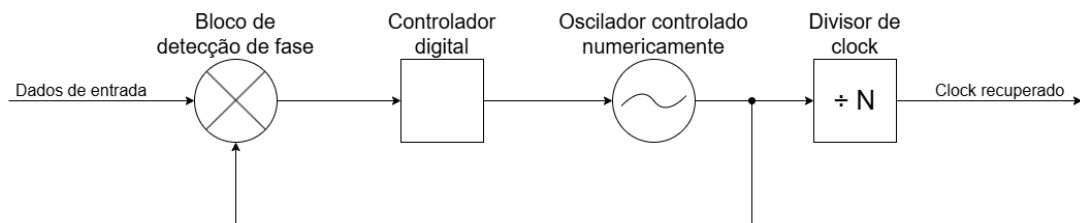
**Figura 2 – Diagrama de blocos da topologia Phase Locked Loop**

*Fonte: Muller e Leblebici (2007)*

O autor explica que sistemas de recuperação de *clock* baseados em PLL apresentam vantagens importantes, como aquisição exata de frequência, capacidade de armazenar a frequência dos dados por longos intervalos por meio do filtro de malha, e uma vasta documentação técnica disponível. Apesar dessas vantagens, os PLLs têm limitações específicas em aplicações multicanaís: cada PLL funciona como um sistema independente, dificultando o compartilhamento de blocos e reduzindo o potencial de otimização geral do receptor. Adicionalmente, o livro destaca que a maioria dos filtros de malha utilizados em PLLs emprega componentes passivos de valores relativamente grandes. Entretanto, topologias mais recentes têm substituído o filtro de malha por controle digital. Topologias PLL de natureza completamente digital trazem vantagens como redução de área ocupada no silício e diminuição de *jitter* propagado no sistema.

### 2.1.2 All Digital Phase Locked Loop

Embora os PLLs tradicionais tenham sido concebidos utilizando blocos analógicos, Muller e Leblebici (2007) explicam que o avanço das tecnologias CMOS e a crescente demanda por sistemas altamente integrados impulsionaram o desenvolvimento dos *All-Digital Phase-Locked Loops* (ADPLLs), ou PLLs completamente digitais. Nessa arquitetura, os blocos analógicos convencionais são substituídos por equivalentes implementados em lógica digital, permitindo que todo o sistema seja sintetizado em *hardware* digital e integrado ao restante do circuito sem a necessidade de componentes analógicos dedicados. Conforme ilustrado na Figura 3, o sinal de dados de entrada é aplicado a um bloco de detecção de fase, responsável por determinar o erro de fase entre as transições dos dados recebidos e o *clock* gerado internamente. Esse erro é então processado por um controlador digital, que desempenha uma função similar a do filtro de malha em PLLs convencionais, realizando o acúmulo do erro de fase por meio de algoritmos digitais, como acumuladores, integradores discretos ou controladores proporcionais-integrais.



**Figura 3 – Diagrama de blocos da topologia All Digital Phase Locked Loop**

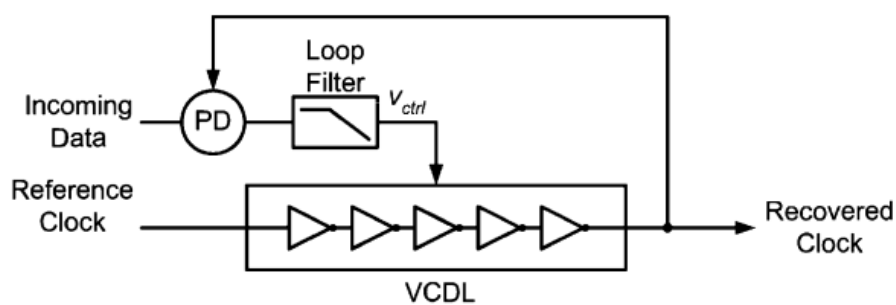
*Fonte: O próprio autor*

O sinal de controle produzido pelo controlador digital é aplicado ao oscilador controlado numericamente (*Numerically Controlled Oscillator – NCO*), bloco responsável pela geração do *clock* interno do sistema. Diferentemente de um VCO, cuja frequência é determinada por uma tensão analógica, o NCO produz sua saída a partir de uma palavra digital de controle, ajustando sua frequência e fase por meio de operações discretas. Em muitas aplicações de recuperação de *clock*, a frequência produzida pelo NCO é posteriormente aplicada a um divisor de *clock*, que gera o sinal de *clock* recuperado na frequência desejada e, simultaneamente, fornece o sinal de realimentação necessário para o fechamento da malha de controle.

Os ADPLLs apresentam diversas vantagens em relação às implementações analógicas tradicionais, incluindo maior portabilidade entre diferentes tecnologias de fabricação, imunidade a variações de processo, tensão e temperatura (PVT), redução de jitter propagado ao sistema, redução da área ocupada em silício e facilidade de reconfiguração por meio de alterações em firmware ou descrição em hardware (Muller; Leblebici, 2007). Além disso, por serem compostos predominantemente por circuitos digitais, os ADPLLs podem beneficiar-se diretamente da contínua redução das dimensões tecnológicas dos processos CMOS, tornando-se uma solução particularmente atrativa para sistemas modernos de recuperação de *clock* e circuitos integrados de comunicação de alta velocidade.

### 2.1.3 Delay Locked Loop

Muller e Leblebici (2007) apresentam a arquitetura *Delay-Locked Loop* (DLL) como uma alternativa à utilização de múltiplos osciladores em um único circuito integrado, por meio da geração centralizada e da distribuição de um sinal de *clock* de referência para todos os canais do receptor. Em cada canal, esse sinal alimenta uma linha de atraso controlada por tensão (*Voltage-Controlled Delay Line* – VCDL), cuja função é ajustar a fase do clock de referência para alinhá-lo ao fluxo de dados recebido, reduzindo a sobrecarga de área de silício em comparação com arquiteturas que empregam um oscilador dedicado para cada canal. Segundo os autores, a estrutura de uma DLL apresenta grande semelhança com uma PLL baseada em oscilador em anel, uma vez que ambas utilizam um detector de fase e um filtro de malha com funcionamento equivalente, conforme esquematizado na figura 4. A principal diferença está no elemento controlado pelo sinal proveniente do filtro de malha: enquanto no PLL o oscilador controlado por tensão altera a frequência do clock, no DLL o VCDL modifica diretamente sua fase por meio do ajuste do atraso introduzido na linha de atraso. Como consequência, o DLL não adiciona um polo à função de transferência da malha, resultando em um sistema de menor ordem, mais estável e sem o acúmulo de *jitter* decorrente da recirculação do sinal em um oscilador. Entretanto, a faixa de ajuste da VCDL é limitada, impossibilitando que a DLL compense diferenças de frequência, mesmo pequenas, entre o receptor e o sinal de dados recebido. Para contornar essa limitação, a literatura apresenta diferentes abordagens, como esquemas de correção da tensão de controle, técnicas baseadas em mistura em quadratura (*quadrature mixing*) e arquiteturas como *Dual Delay-Locked Loop* (DDLL), esta última fundamentada no princípio da interpolação de fase.

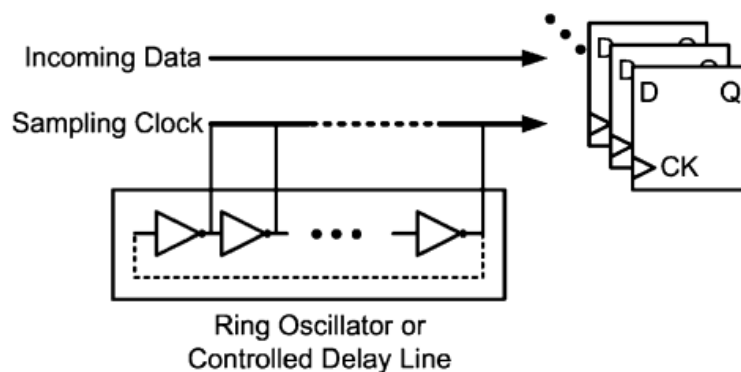


**Figura 4 – Diagrama de blocos da topologia Delay Locked Loop**

Fonte: Muller e Leblebici (2007)

### 2.1.4 Oversampling

Muller e Leblebici (2007) apresentam a arquitetura de *Oversampling* como uma alternativa para recuperação de *clock* favorecida pelo aumento da velocidade das tecnologias CMOS. Nessa arquitetura, um *clock* multifásico, gerado por um PLL baseado em oscilador em anel ou por um DLL, é utilizado para amostrar o sinal de entrada em diferentes instantes ao longo de um mesmo período de *clock*, conforme ilustrado na Figura 5. As amostras obtidas são armazenadas em um conjunto paralelo de elementos de resincronização, tipicamente *flip-flops*, formando uma palavra digital que é processada por um bloco de lógica de decisão. Esse bloco executa um algoritmo de seleção de fase, responsável por identificar a amostra mais representativa do dado recebido e, conseqüentemente, minimizar a probabilidade de erro de detecção. Como a decisão é realizada inteiramente no domínio digital, essa arquitetura dispensa uma malha de realimentação analógica, reduzindo significativamente a quantidade de blocos analógicos necessários. Entretanto, seu desempenho é fortemente influenciado por descasamentos entre os componentes do sistema, especialmente pelo descasamento de atraso nos caminhos de distribuição do *clock* multifásico. Essa sensibilidade torna-se ainda mais crítica quando um mesmo conjunto de fases é distribuído para diversos receptores ao longo do circuito integrado. Além disso, a implementação digital do algoritmo de seleção de fase pode ocupar uma área considerável de silício. Apesar dessa limitação, os autores destacam que a contínua evolução das tecnologias CMOS de dimensões profundamente submicrométricas tende a reduzir esse custo, tornando os receptores por sobreamostragem uma alternativa cada vez mais atrativa em termos de área ocupada e consumo de potência.

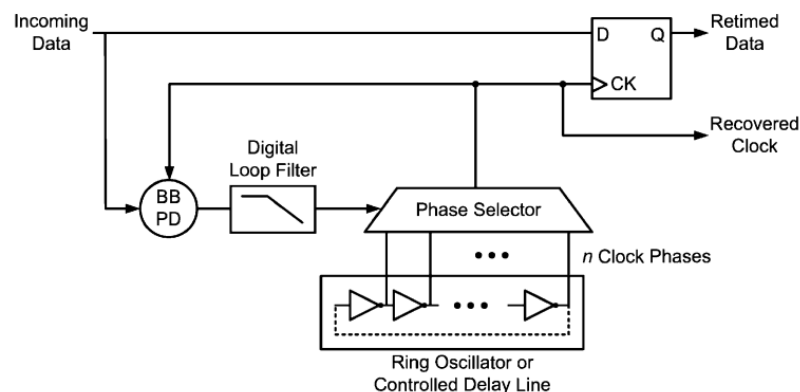


**Figura 5 – Diagrama de blocos da topologia de Oversampling**

*Fonte: Muller e Leblebici (2007)*

### 2.1.5 Phase Interpolation

Muller e Leblebici (2007) apresentam a topologia de Phase Interpolation como uma técnica de recuperação de clock baseada na utilização de um *clock multifásico*. Inicialmente, é realizada uma seleção de fase, na qual é escolhida a fase do *clock* que apresenta o melhor alinhamento com as transições do sinal recebido, minimizando o erro de fase, conforme esquematizado na figura 6. Os autores destacam que essa estratégia possui grande semelhança com os métodos de alinhamento de fase, nos quais o sinal de dados percorre uma linha de atraso, gerando múltiplas versões defasadas que são posteriormente selecionadas por meio de um multiplexador. Entretanto, como a propagação de sinais de dados não periódicos, característicos da operação em modo de rajada (*burst mode*), pode provocar distorções no ciclo de trabalho (*duty cycle*), a técnica de alinhamento de fase torna-se menos adequada que a seleção de fases para esse tipo de aplicação. A resolução da seleção de fase pode ser ampliada por meio da interpolação ponderada entre duas fases consecutivas do *clock*. Nessa abordagem, os bits mais significativos da saída do filtro de malha determinam o intervalo de fase correspondente ao instante de chegada dos dados, enquanto os bits menos significativos definem a ponderação utilizada para interpolar as duas fases adjacentes, proporcionando um ajuste mais fino da fase recuperada. Segundo os autores, essa arquitetura pode ser implementada de forma compacta utilizando um detector de fase *bang-bang*, um filtro de malha digital e um conversor digital-analógico baseado em comutação de corrente. Devido às semelhanças com as arquiteturas de alinhamento de fase e ao fato de o *clock multifásico* poder ser gerado tanto por um PLL quanto por um DLL, implementações dessa técnica recebem diferentes denominações na literatura, como Dual Delay-Locked Loop (DDLL) ou Dual-Loop PLL (DLPLL). Em algumas implementações são empregados *clocks* em quadratura, nas quais o interpolador de fase é denominado rotacionador de fase. Essa configuração requer o roteamento de apenas dois sinais diferenciais de *clock*, reduzindo a sobrecarga de área em comparação com arquiteturas de sobreamostragem e eliminando a necessidade de um DLL dedicada para geração de fases em cada canal.



**Figura 6 – Diagrama de blocos da topologia de Phase Interpolation**

Fonte: Muller e Leblebici (2007)

## 2.2 Estado da Arte

A revisão bibliográfica a seguir foi conduzida utilizando a base de dados Scopus com o objetivo de identificar publicações sobre desenvolvimento de componentes de *clock recovery*. O propósito deste levantamento é entender a evolução dos componentes de *clock recovery* e analisar de maneira qualitativa as tendências das especificações desses componentes. A busca foi limitada por dois termos chaves, o primeiro foram expressões relacionadas à "*Clock and Data Recovery*", garantindo que a publicações encontradas trariam uma descrição de um componente de *clock recovery*. Já o segundo termo chave tinha o intuito de aproximar as publicações do objeto desse estudo, sendo ele um só termo dentre a seguinte seleção: *Visible Light System (VLC)*, modulação VPPM ou *Hardware Description Language (VHDL ou Verilog)*. A expressão de busca foi realizada no dia 28/11/2025 e retornou 63 publicações. Desse total, 17 foram removidas da análise visto que não estavam disponíveis ou o tema apresentado não contemplava o escopo desse estudo: apresentar um componente de *clock recovery* ou sua aplicação em um sistema de comunicação.

### 2.2.1 Avanço Tecnológico

Elrabaa (2006b) apresenta um circuito totalmente digital de recuperação de *clock* capaz de extrair o *clock* embutido em sinais seriais NRZ (*Non Return to Zero*) sem o uso de blocos analógicos, reduzindo o *overhead* e complexidade do componente. O autor ressalta que, diferentemente de PLLs analógicos que sofrem com *jitter*, ocupação de área de silício, dificuldade de portabilidade e longos tempos de travamento, o componente proposto utiliza apenas lógica digital padrão, permitindo fácil descrição e síntese em HDL. O circuito mede a duração do bit por meio de uma linha de atraso digital e forma um oscilador cuja periodicidade corresponde ao período medido. A cada transição de dados, o *clock* é retemporizado, eliminando acúmulo de erro de fase. A arquitetura inclui detectores de borda, linhas de atraso idênticas, um bloco de captura de período e multiplexação de *clock*, e um disparador Schmitt. O estudo elaborado pelo autor fortalece a ideia de que componentes de *clock recovery* digitais tem melhor desempenho do que componentes digitais analógicos, principalmente para sinais de dados de menor frequência. Os resultados das simulações em tecnologia CMOS de 0,13  $\mu\text{m}$  feitos pelo autor mostram operação estável até 2–2,5 Gb/s, com recuperação de *clock* em apenas duas transições de dados e *jitter* menor que 10 ps. O componente apresenta excelente estabilidade, com erro de frequência inferior a 0,8% para diversas taxas de dados. Com menos de 100 portas, o circuito é compacto, altamente portátil e adequado para sistemas com múltiplos links seriais em um único chip.

Liu *et al.* (2013) apresentam o primeiro estudo da seleção com enfoque em componentes de *clock recovery* para sistemas de comunicação por luz visível. Os autores propõem um sistema de comunicação por luz visível baseado em *AC-powered LEDs*, no qual o sinal transmitido é elevado à faixa de 400 kHz e então combinado com o 60 Hz da tomada. Para isso, utiliza-

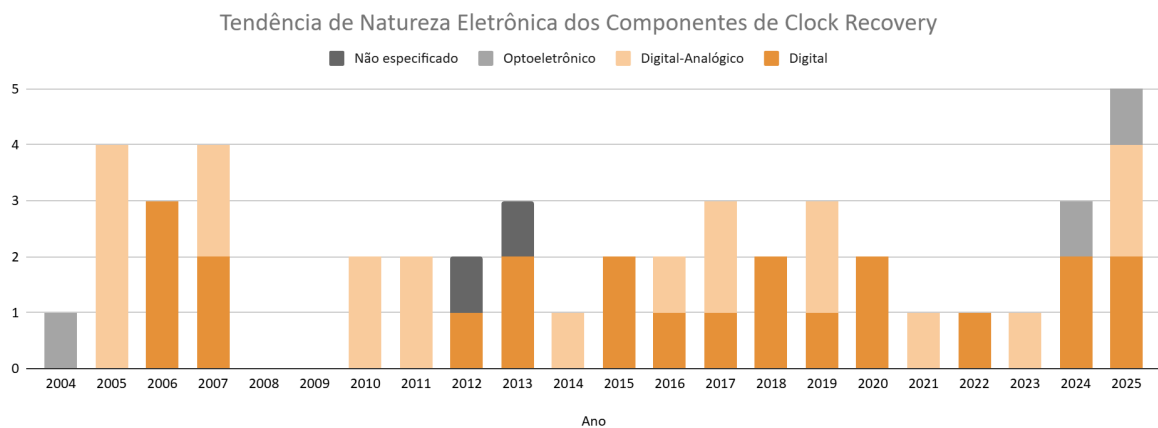
se um *bias-tee* especialmente projetado, capaz de combinar sinais de baixa e alta frequência sem causar distorção ou *spectral overlap*. Esse arranjo permite que o LED seja polarizado corretamente e opere na região linear, mesmo quando usado para iluminação e comunicação simultaneamente. O artigo destaca que o uso de LEDs alimentados em CA para VLC ainda é pouco explorado. Em soluções anteriores, o sinal OOK sofria distorção devido ao ciclo da rede CA e as taxas de dados eram muito baixas. Para superar essas limitações, o trabalho utiliza *up-conversion*, separação espectral via *bias-tee* e *clock recovery* sincronizado com o *AC-power*, permitindo aplicar o sinal transmitido somente durante a porção retificada do ciclo de alimentação. Apesar de não operar conforme parâmetros da camada física PHY II para equipamentos *indoor* da IEEE 802.15.7, o estudo trás percepções significativas para a evolução da tecnologia VLC integrada à infraestrutura de iluminação doméstica. O sistema experimental utiliza um sinal modulado em OOK 200 kbit/s, *up-converted* para 400 kHz, transmitido por um arranjo de 40 LED's alimentado pela combinação CA + mensagem. Após 2 m de transmissão, o sinal recebido passa por processamento digital: filtro passa-faixas, mistura coerente e filtro passa-baixas, permitindo recuperar o sinal de banda base e analisar o diagrama de olho, que apresentou BER (*Bit Error Rate*)  $\approx 10^{-9}$ . A taxa média resultante é 60 kbit/s devido ao uso de retificação de meia onda, podendo chegar a 120 kbit/s com retificação de onda completa.

Liu *et al.* (2020) apresentam um sistema VLC voltado para tags IoT de baixo consumo, cujo sinal de dados é modulado em Manchester encoding. A modulação VPPM e Manchester são similares em sua essência, a diferença está em sinais VPPM ajustarem sua largura de pulso conforme a necessidade e Manchester *encoding* não. O foco dos autores é mostrar uma solução compacta, energeticamente eficiente e adequada para dispositivos alimentados por bateria. O receptor óptico de *front-end* utiliza um amplificador de transimpedância (TIA) baseado em espelho de corrente cascode operando na região próxima à tensão limiar. Em sequência, está um filtro passa-baixas que gera dinamicamente o nível de referência, e por um comparador que restaura o dado digital. Os autores argumentam que essa arquitetura reduz área, consumo e sensibilidade a variações de processo, mantendo desempenho suficiente para 1 Mbps. O texto descreve uma arquitetura totalmente digital, composta por um detector digital de frequência quadricorrelator, um detector de fase de tipo bang-bang, um filtro digital de malha com retenção de ordem zero e mecanismo de controle de ciclos-limite e um oscilador controlado digitalmente (DCO) com laço de auto-calibração. O DCO usa matrizes de transistores PMOS para regulação de corrente em um oscilador em anel de quatro estágios e incorpora correção de *duty cycle* para mantê-lo em 50%. O chip foi fabricado em TSMC 180 nm, ocupando 0,728 mm<sup>2</sup>. As medições mostram *jitter* RMS de 3,312 ns (0,0033 UI) e *phase noise* de -39,54 dBc/Hz (1 kHz), -80,35 dBc/Hz (10 kHz) e -92,26 dBc/Hz (100 kHz). O sistema opera a 1 Mbps com consumo total de 5,58  $\mu$ W, valores que os autores consideram competitivos para aplicações de comunicação por luz visível de baixa velocidade para aplicações IoT.

### 2.2.2 Análise Qualitativa

Com o intuito de apresentar uma visão holística sobre o tema, as 47 publicações selecionadas na seção 2.2 foram examinadas. A análise consistiu na leitura dos resumos, figuras, equações e, quando necessário, de seções específicas dos documentos. O objetivo foi categorizar informações relevantes como: aplicação dos componentes desenvolvidos e sua natureza, conceito base da sua arquitetura, tipo de modulação e frequência do sinal transmitido. O resultado dessa busca está organizado em uma tabela no apêndice A.

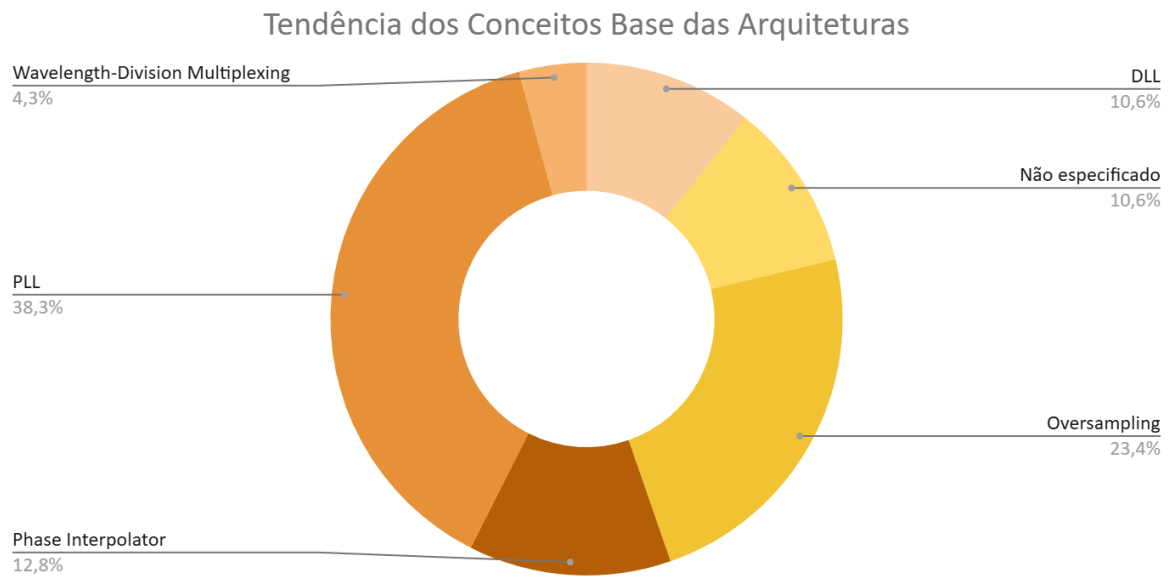
Dentre as aplicações de componentes de *clock recovery* estão os sistemas de comunicação óptica digital, comunicação óptica em espaço livre, comunicação por satélite, instrumentação de sensores, comunicação SATA, transmissão de áudio e principalmente comunicação por luz visível. A utilização nesses sistemas concentra-se principalmente na parte de *data recovery* e serialização e deserialização de dados. Analisando a figura 7, é possível inferir que existe um equilíbrio entre a tendência de desenvolver componentes puramente digitais (22 publicações) e mistas, ou seja, parte digitais parte analógicas (20 publicações).



**Figura 7 – Natureza eletrônica dos componentes de clock recovery da revisão bibliográfica**

*Fonte: O próprio autor*

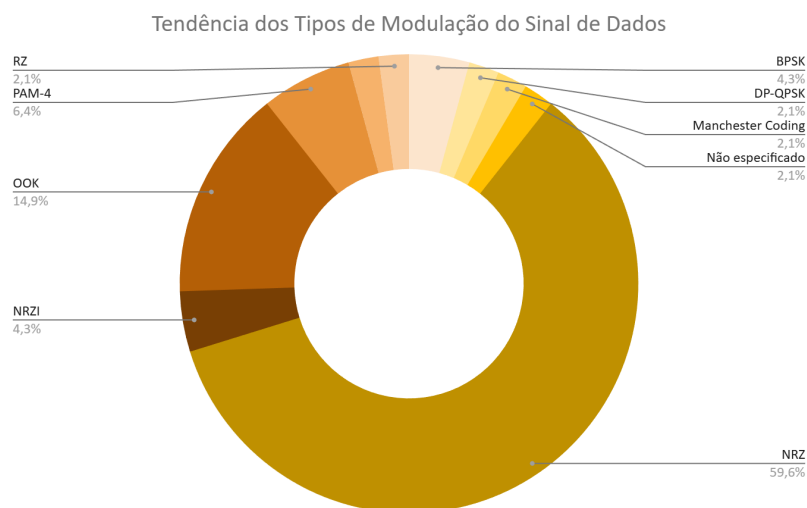
O conceito base de *Phase Locked Loop* foi implementado em mais de um terço dos componentes analisados, como mostra a figura 8, reforçando a afirmação de Muller e Leblebici (2007) que consagra esse conceito como uma arquitetura clássica de *clock recovery*. Adicionalmente, *Oversampling* demonstra bastante recorrência, ocupando uma parcela superior a um quinto dos conceitos base aplicados nos componentes.



**Figura 8 – Conceitos base das arquiteturas dos componentes de clock recovery da revisão bibliográfica**

*Fonte: O próprio autor*

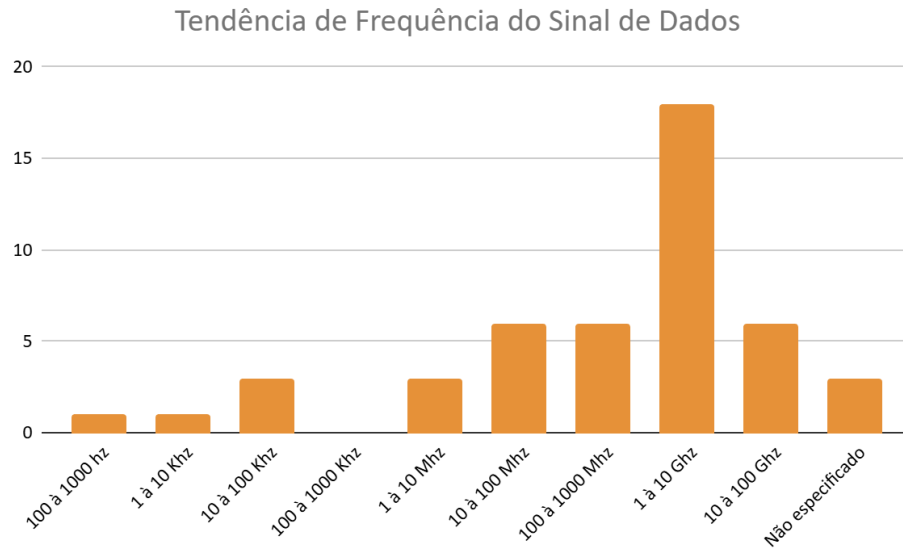
De acordo com a figura 9, mais da metade dos componentes foram projetados para receber sinais modulados no padrão NRZ. Somando esse número aos números de OOK, a quantidade beira três quartos do grupo amostral. Apesar de não terem sido encontradas publicações especificamente para sinais modulados em VPPM, há publicações que abordam componentes de *clock recovery* para sistemas de comunicação por luz visível e até para manchester *encoding*, uma modulação essencialmente similar ao VPPM, conforme visto na 2.2.1.



**Figura 9 – Tipos de modulação dos sinais de dados dos componentes de clock recovery da revisão bibliográfica**

*Fonte: O próprio autor*

Por fim, é possível notar uma tendência do desenvolvimento de componentes de *clock recovery* cuja frequência do sinal de dados esteja na faixa de 1 a 10 Ghz na figura 10. Das 47 publicações analisadas, somente 5 apresentavam um sinal de dados com frequência inferior a 1 MHz.



**Figura 10 – Faixa de frequências dos sinais de dados dos componentes de clock recovery da revisão bibliográfica**

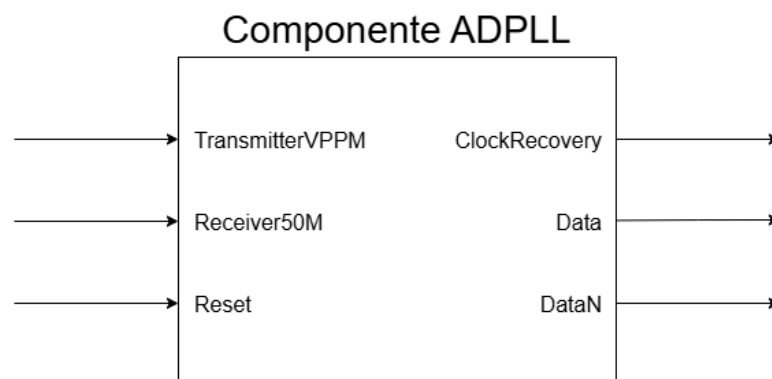
*Fonte: O próprio autor*

### 3 DESCRIÇÃO DO SISTEMA

Para esse estudo, optou-se por desenvolver uma topologia baseada no conceito de *Phase Locked Loop* de natureza totalmente digital devido suas características robustas e aplicação consagrada pela literatura, conforme abordado na seção 2.1. A arquitetura do componente foi descrita inteiramente em VHDL e seu código está disponível para consulta no apêndice B. Para compreender o funcionamento do componente *All Digital Phase Locked Loop* (ADPLL), este capítulo apresenta um diagrama caixa preta geral, um diagrama de blocos geral e seis diagramas caixa preta do seu funcionamento interno. Em sequência, é apresentado o relatório de compilação do código VHDL e suas simulações.

#### 3.1 Funcionamento geral do componente ADPLL

Assim como apresenta o diagrama da figura 11, os sinais de entrada desse sistema são: *TransmitterVPPM* que é o trem de bits VPPM lido pelo receptor, *Receiver50* que é o sinal de *clock* interno de 50 MHz da plataforma FPGA e *Reset* que é o sinal de *reset* ativo alto para esse componente. Os sinais de saída desse sistema são *ClockRecovery* que é o sinal de *clock* 3,75 MHz recuperado a partir do trem de bits VPPM sincronizado a uma defasagem de 90°, *Data* que é o sinal de dados decodificado e *DataN* que é a negação do sinal de dados.



**Figura 11 – Diagrama caixa preta do componente ADPLL**

*Fonte: O próprio autor*

Os sinais de entrada do ADPLL serão processados pelos seguintes blocos internos: PLL do megawizard de propriedade intelectual da Altera, bloco de detecção de fase, bloco de operações iniciais do controlador PI, bloco de operações finais do controlador PI, bloco do oscilador controlado numericamente (NCO) e bloco de recuperação de dados. Os sinais internos do componente ADPLL e suas interligações entre esses blocos serão descritas nas próximas sessões e podem ser conferidos graficamente no diagrama de blocos apresentado na figura 12.

# Componente ADPLL

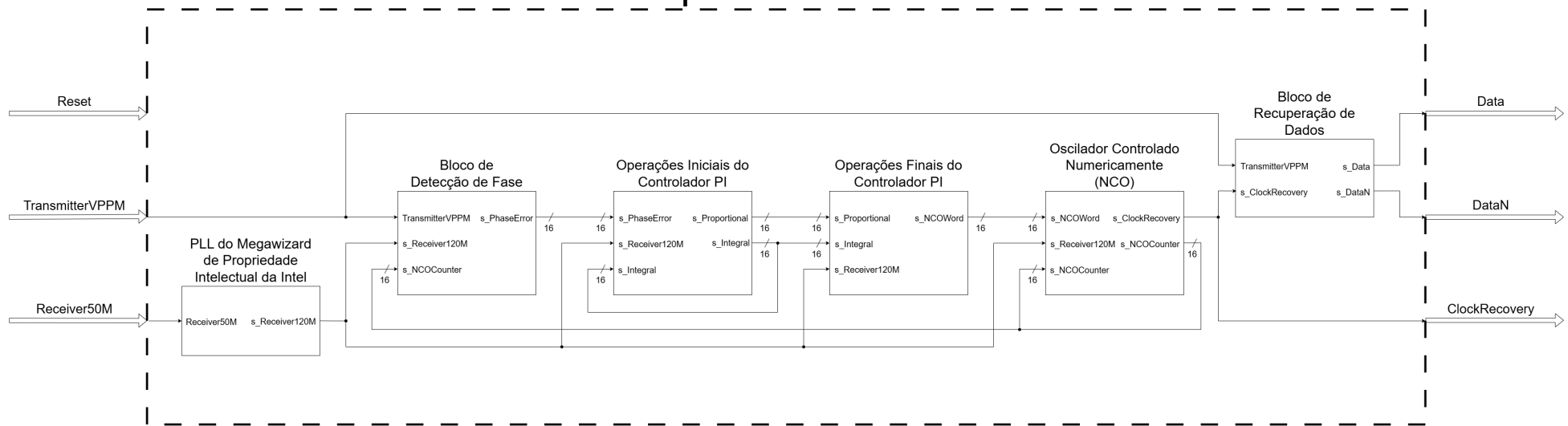
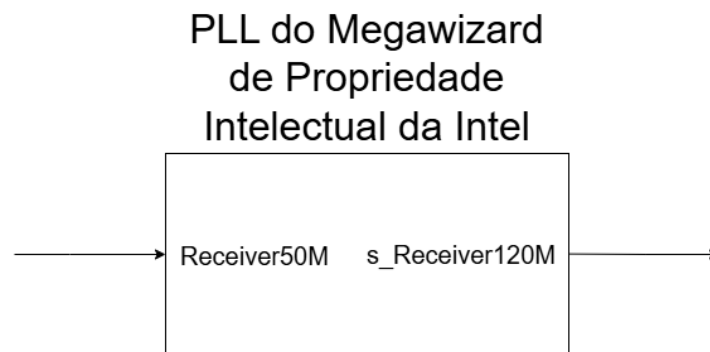


Figura 12 – Diagrama caixa preta do componente ADPLL

Fonte: O próprio autor

### 3.2 Funcionamento do PLL de propriedade intelectual da Intel

Para explicar o funcionamento interno do componente ADPLL, é essencial frisar que seu sinal de saída é derivado do próprio sinal de *clock* interno da FPGA. A base de temporização de placas DE10 NANO é uma onda quadrada de 50 MHz que pode ser multiplicada ou dividida e aplicada em diferentes circuitos internos. Esse recurso de derivação de *clock* está disponível também para o desenvolvimento de códigos VHDL. No software Quartus Prime, na seção de recursos de propriedade intelectual da Intel, é possível configurar um componente PLL para gerar o sinal de clock necessário. Porém, conforme explicado no passo a passo presente no apêndice C, o megawizard do Quartus Prime cria componentes PLL cujo *clock* de saída seja no mínimo 5 MHz. Portanto, o componente PLL de propriedade intelectual da Intel sozinho não é capaz de chegar a uma frequência de 3,75 MHz, sendo necessário implementar em sequência uma divisão de *clock* no código VHDL. Devido a 50 MHz não ser múltiplo de 3,75 MHz, o PLL de propriedade intelectual da Intel empregado no ADPLL primeiramente aumenta a frequência do *clock* para 120 MHz, assim como indicado pelo diagrama de caixa preta na figura 13. O sinal de entrada *Receiver50M* será o próprio sinal de *clock* interno da FPGA e o sinal de saída *s\_Receiver120M* o sinal de *clock* resultante da multiplicação. Para concluir este raciocínio, é importante notar que o sinal de *clock* de 120 MHz será dividido até 3,75 MHz posteriormente durante as etapas de sincronização do sinal de *clock* com o trem de bits VPPM.

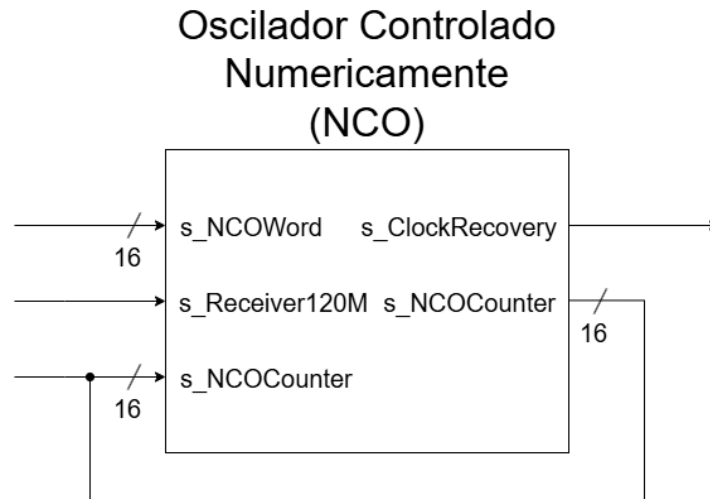


**Figura 13 – Diagrama caixa preta do PLL de propriedade intelectual da Intel**

*Fonte: O próprio autor*

### 3.3 Funcionamento do oscilador controlado numericamente

Uma maneira comum de dividir o *clock* de 120 MHz seria conectá-lo a um contador de 5 bits e isolar seu dígito mais significativo. A frequência de 120 MHz é 32 vezes mais alta que a frequência de 3,75 MHz, portanto o dígito mais significativo do contador agiria sob uma frequência de 3,75 MHz. Porém, como é necessário sincronizar o *clock* recuperado com as bordas de subida e descida do trem de bits VPPM, somente dividir o *clock* não é suficiente. O ADPLL precisa ser capaz de comparar os sinais e decidir se o *clock* recuperado deve ser atrasado, adiantado ou se manter na fase atual. Portanto, a arquitetura de um oscilador controlado numericamente (NCO) foi implementada, assim como indicado no diagrama caixa preta da figura 14. O sinal de entrada *s\_NCOCOUNTER* é um contador a complemento de dois, resolução de 5 bits e é incrementado a cada borda de subida do sinal de entrada *s\_Receiver120M*. O contador é acrescido por um incremento padrão - que representa uma parcela de  $1/32$  do valor cheio do contador - junto ao valor do sinal *s\_NCOWord* que é um valor de correção positivo ou negativo. O valor de correção está em constante atualização e é gerado por uma malha de controle composta por um bloco de detecção de fase e um controlador PI que decidem se é necessário atrasar, adiantar ou manter a fase atual do *clock* recuperado. A saída do NCO, o sinal *ClockRecovery*, é o dígito mais significativo do contador.



**Figura 14 – Diagrama caixa preta do oscilador controlado numericamente**

*Fonte: O próprio autor*

### 3.4 Funcionamento do bloco de detecção de fase

A malha de controle do componente ADPLL é composta por 3 etapas. A primeira delas é um bloco de detecção de fase, cujo objetivo é avaliar a sincronia entre o trem de bits VPPM e o sinal de *clock* recuperado. A cada borda de subida do sinal *s\_Receiver120M*, o bloco de detecção de fase verifica se houve uma mudança no estado lógico do sinal *TransmitterVPPM*. Assim, é possível determinar quando houve uma borda de subida ou descida. Essa supervisão atua como uma detecção de borda. Assim que uma mudança é detectada, o valor do sinal *s\_NCOCOUNTER* é amostrado. A leitura deve indicar metade do valor cheio do contador, seja positivo ou negativo, para que a borda de subida do sinal de *clock* recuperado ocorra o mais distante das bordas de subida ou descida do trem de bits VPPM. Em outras palavras, a amostra do contador deve indicar que o *clock* recuperado está 90 adiantado ou 90 graus atrasado das bordas do trem de bits VPPM. Essa amostra é propagada para a próxima etapa da malha de controle como o sinal *s\_PhaseError*, que é o erro de fase do sinal de *clock* recuperado em relação ao trem de bits VPPM.

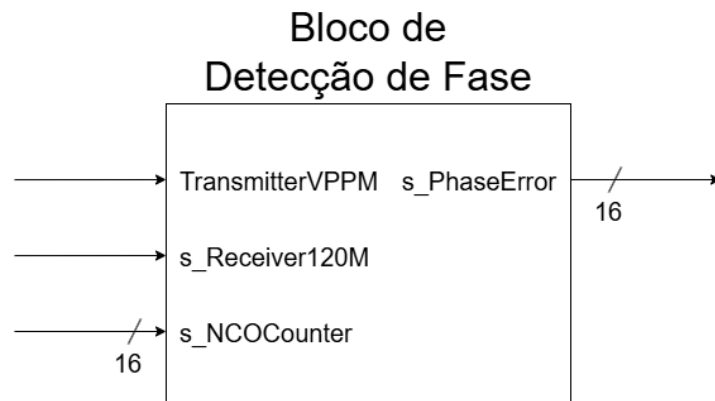


Figura 15 – Diagrama caixa preta do bloco de detecção de fase

Fonte: O próprio autor

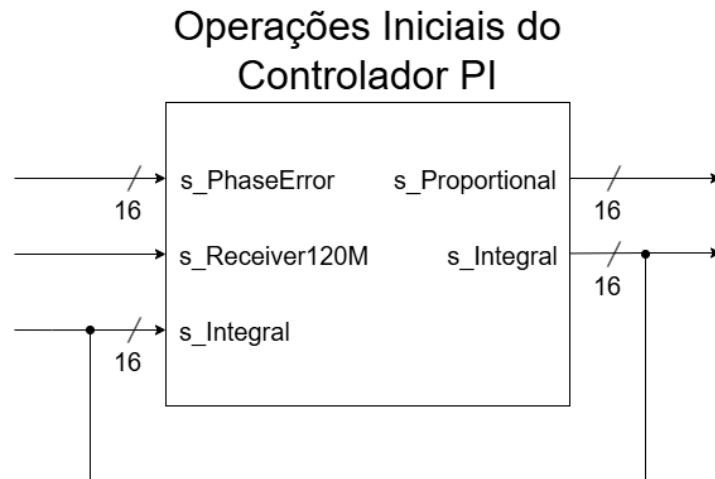
### 3.5 Funcionamento do controlador PI

O objetivo da segunda etapa da malha de controle é calcular as parcelas proporcional e integral da correção a ser aplicada no NCO. Na borda de subida do sinal *Receiver120M* após a primeira etapa, o controlador PI utiliza o valor do sinal *s\_PhaseError* para calcular os valores das parcelas proporcional e integral. A equação 1 apresenta os cálculos realizados para definir a parcela proporcional e a equação 2 apresenta os cálculos realizados para definir a parcela integral.

$$\text{Parcela Proporcional} = \frac{\text{Erro de fase}}{2^{K_p}} \quad (1)$$

$$\text{Parcela Integral} = \text{Parcela Integral} + \frac{\text{Erro de fase}}{2^{K_i}} \quad (2)$$

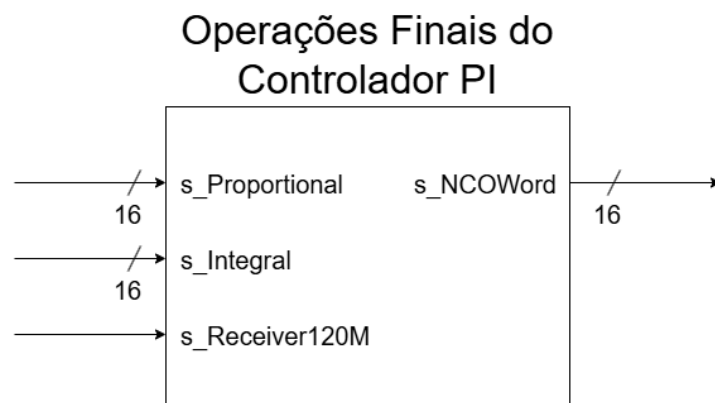
Durante a execução das simulações apresentadas na seção 3.7, os valores de ganho de KP e KI foram ajustados para 9 e 12, respectivamente. Esses valores se mantiveram ao longo dos testes realizados e apresentados no capítulo 4. Os sinais de saída desse bloco são *s\_Proportional* que representa a parcela proporcional e *s\_Integral* que representa a parcela integral, assim como mostra o diagrama caixa preta da figura 16.



**Figura 16 – Diagrama caixa preta do bloco de operações iniciais do controlador PI**

*Fonte: O próprio autor*

Na última etapa da malha de controle, os valores de *s\_Proportional* e *s\_Integral* são somados durante a borda de subida do sinal *Receiver120M* após a segunda etapa. O resultado dessa soma é apresentado como o sinal *s\_NCOWord* na figura 17 e ele representa a correção da malha de controle aplicada ao NCO. Esse valor de correção então é somado junto ao incremento padrão do NCO a cada borda de subida do *clock* de 120 MHz, sendo atualizado novamente somente após o trem de bits VPPM passar por uma borda de transição novamente.

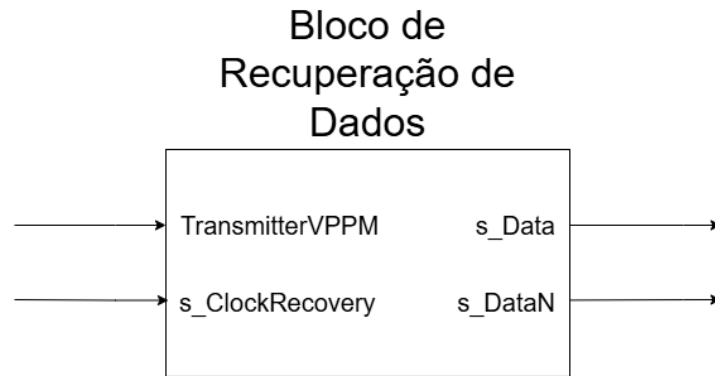


**Figura 17 – Diagrama caixa preta do bloco de operações finais do controlador PI**

*Fonte: O próprio autor*

### 3.6 Funcionamento do bloco de recuperação de dados

Por fim, o bloco de recuperação de dados realiza uma amostragem do sinal *TransmitterVPPM* a cada borda de subida do sinal *s\_Receiver120M*. Essa amostragem é disponibilizada pelo sinal de saída *Data* e pelo seu valor negado, *DataN*.



**Figura 18 – Diagrama caixa preta do bloco de recuperação de dados**

*Fonte: O próprio autor*

Ambos os sinais de saída serão necessários para o sistema VLC visto a existência de ambiguidade na interpretação de trem de bits VPPM com bits alternados. Dada uma amostragem em um momento aleatório de uma transmissão, não é possível determinar instantaneamente se uma borda de transição do trem de bits coincide com a borda de subida do sinal de *clock* originado pelo transmissor. Essa informação é importante pois a borda de subida do sinal de *clock* originado pelo transmissor é utilizada para modular o trem de bits VPPM. Sem essa informação, um trem de bits VPPM originado pela sequência '10101010' pode ser interpretado como '01010101' pelo receptor. Ambas sequências de bits geram uma forma de onda similar a uma onda quadrada e, portanto, há ambiguidade na hora de determinar em qual instante deve ser realizada a leitura dos dados. Devido a essa ambiguidade entre bordas de subida e descida, o sistema VLC deve empregar um *start byte* característico no começo de cada pacote de dados. É necessário haver uma definição de quantos e quais bits serão transmitidos ao início de cada pacote de dados para que componentes do sistema VLC subsequentes ao ADPLL possam identificar se a leitura de dados está sendo feita corretamente ou se os dados estão negados.

### 3.7 Síntese e resultados de simulação

Para validar a arquitetura proposta, o código VHDL foi compilado e passou por simulações comportamentais no software Quartus Prime. Uma vez que os resultados foram validados, o código VHDL passou por outras simulações do tipo *gate level*, que além de levar em consideração os aspectos funcionais da lógica do código, também leva em consideração aspectos físicos e limitações construtivas da própria FPGA. Ao comparar resultados, percebeu-se que o componente ADPLL atuou de forma similar em ambas simulações. Portanto, optou-se por apresentar somente os resultados das simulações *gate level* visto que elas trazem as mesmas informações que os resultados das simulações comportamentais.

A compilação do código VHDL mostrada na figura 19 indicou uma utilização de recursos lógicos inferior à 1%, indicando que sua implementação é bastante econômica e ideal para sistemas que precisam acomodar múltiplos componentes VHDL em somente uma FPGA. Por sua vez, as simulações feitas no software ModelSim do Quartus Prime foram capazes de confirmar o bom funcionamento do ADPLL e sua capacidade de sincronizar seu sinal de *clock* com o trem de bits VPPM.

Flow Status	Successful - Tue Jun 02 21:09:06 2026
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	ADPLL
Top-level Entity Name	ADPLL
Family	Cyclone V
Device	5CSEBA6U2317
Timing Models	Final
Logic utilization (in ALMs)	46 / 41,910 (< 1 %)
Total registers	71
Total pins	4 / 314 ( 1 %)
Total virtual pins	0
Total block memory bits	0 / 5,662,720 ( 0 %)
Total DSP Blocks	0 / 112 ( 0 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	1 / 6 ( 17 %)
Total DLLs	0 / 4 ( 0 %)

**Figura 19 – Relatório de Compilação do código VHDL no Software Quartus Prime 18.1**

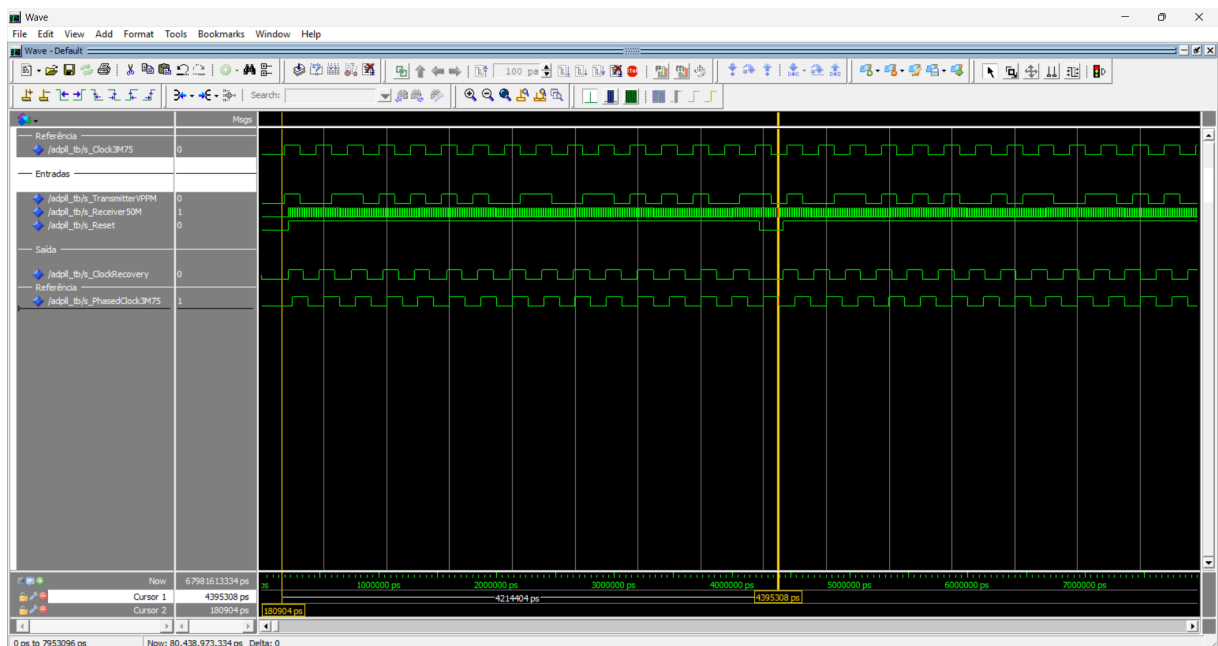
*Fonte: O próprio autor*

Para executar as simulações, foi necessário um arquivo auxiliar VHDL apelidado de *testbench* que descreve quais sinais são aplicados às entradas do componente ADPLL. O arquivo utilizado para os testes apresentados abaixo pode ser encontrado no apêndice E. O primeiro sinal especificado pelo *testbench* é uma onda quadrada de 3,75 MHz que origina a temporização da mensagem enviada pelo transmissor do sistema VLC. A partir desse sinal, um trem de bits VPPM é modulado, transportando a seguinte mensagem:

ABCDEFGHIJKLMNPOQRSTUVWXYZabcdefghijklmnopqrstuv

Esse trem de bits VPPM é conectado a entrada do componente ADPLL, assim como também uma onda quadrada de 50 MHz que simula o *clock* interno da FPGA DE10 NANO e um sinal de *reset*. Um último sinal de referência é gerado pelo *testbench*: uma onda quadrada de 3,75 MHz com 90° de defasagem em relação ao trem de bits VPPM para ajudar na visualização da qualidade do sinal de saída do ADPLL.

Os resultados da simulação *gate level* do ADPLL mostrados na figura 20 demonstram um cenário em que o sinal de *reset* foi iniciado em estado baixo e transicionou para estado alto 33,3 ns após o início da transmissão VPPM. Dessa forma, o ADPLL iniciou sua sincronização quando a borda de subida do *clock* recuperado estava 45° atrasado em relação à borda de subida do trem de bits VPPM (observar valores dos sinais após o primeiro cursor na figura 20, marcado por uma linha vertical amarela). Nesse cenário, o objetivo do componente ADPLL é atrasar mais o sinal de *clock* recuperado para que ele atinja uma fase de 90° atrasado, garantindo que a leitura dos dados seja realizada no momento mais distante das bordas de subida e descida do trem de bits.



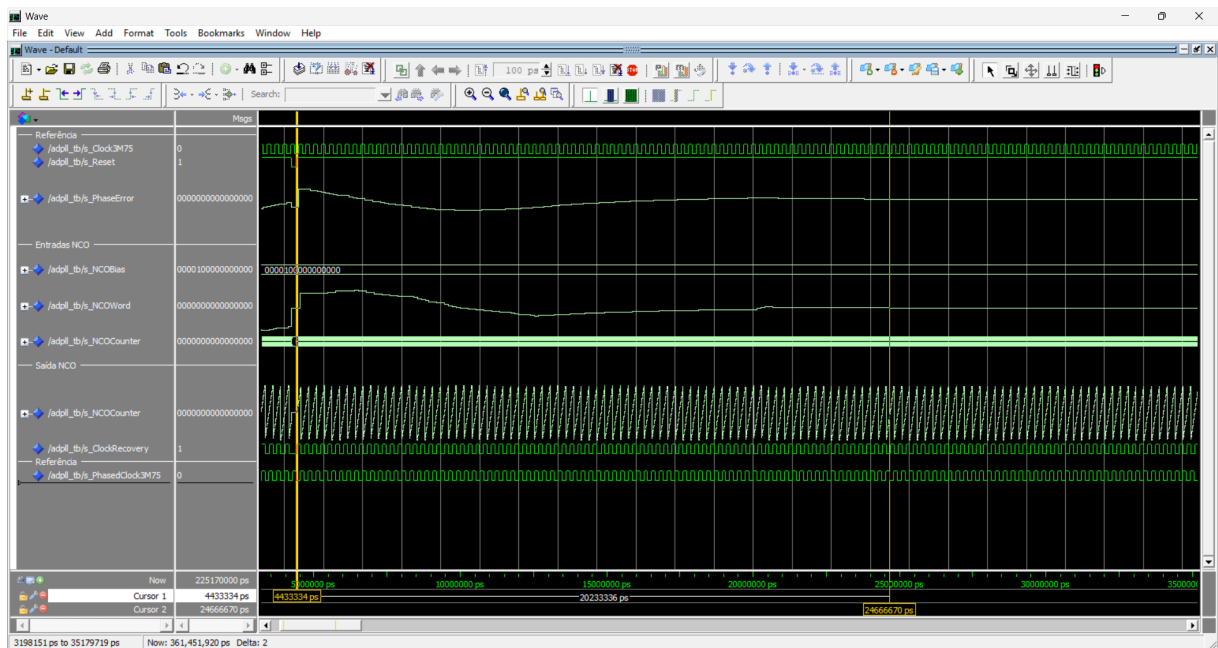
**Figura 20 – Captura de tela da simulação gate level realizada do componente ADPLL**

*Fonte: O próprio autor*

Após 4  $\mu s$ , o sinal de *clock* recuperado demonstrou uma defasagem próxima à 90°, similar a do sinal gabarito gerado pelo *testbench*. Então, o sinal de *reset* transicionou para estado baixo, desligando o ADPLL, e 200 ns depois transicionou para estado alto novamente. Porém, nesse cenário a borda de subida do sinal de *clock* recuperado estava 45° adiantada em relação à borda de subida do trem de bits VPPM (observar valores dos sinais após o segundo cursor na figura 20, marcado por uma linha vertical amarela). Nesse contexto, o objetivo do componente ADPLL é adiantar mais o sinal de *clock* recuperado para que ele atinja uma fase de 90° adiantado. Após mais 4  $\mu s$ , é possível notar que as bordas de subida e descida do

o sinal de *clock* recuperado estão sincronizadas com as do sinal gabarito gerado pelo *testbench*. Entretanto, os estados de um sinal estão negados em relação ao outro. Isso se deve ao fato de ser somente possível definir a posição das bordas do *clock* referência do transmissor, mas não qual borda originalmente era de subida e qual é de descida.

Já a simulação *gate level* apresentada na figura 21 demonstra o comportamento interno do ADPLL sob o mesmo cenário descrito previamente. O primeiro cursor (linha vertical amarela) indica o último instante em que o *reset* ficou em estado baixo. Nesse momento, o ADPLL iniciou a sincronização do sinal de *clock* recuperado com as bordas de transição do trem de bits VPPM a uma fase inicial de  $45^\circ$  adiantado. A primeira onda de cima para baixo expressada de maneira analógica na figura 21 indica o erro de fase, a segunda indica o valor de correção aplicada ao NCO e a terceira é o próprio valor do contador do NCO. É possível notar que após  $20,2 \mu s$ , onde o segundo cursor vertical está posicionado, não há mais variação no valor do erro de fase. Nesse momento, a soma da parcela proporcional e da parcela integral da malha de controle estão resultando em um valor nulo. Portanto, não há mais correção da malha de controle e o erro de fase se mantém. Dessa forma, entende-se que o ADPLL entrou em regime permanente com um erro de fase de  $9,8^\circ$  atrasado.



**Figura 21 – Captura de tela da simulação gate level realizada do oscilador controlado numericamente**

*Fonte: O próprio autor*

#### 4 PROCEDIMENTOS E RESULTADOS EXPERIMENTAIS PRÁTICOS

Afim de validar os resultados obtidos em simulação e confirmar o bom funcionamento do componente PLL, os seguintes testes experimentais foram propostos: análise de integridade de sinal através de diagrama de olho, caracterização da curva de resposta a degrau e teste de taxa de erro de bits. Os sinais de entrada, configuração de equipamentos, variáveis controle e objetivos de cada teste diferem entre si e serão descritos detalhadamente nas próximas sessões. Entretanto, os equipamentos e programas de instrumentação utilizados serão os mesmos. Portanto, as descrições a seguir são aplicáveis para os três testes.

A começar pelo gerador de funções arbitrárias, utilizou-se o modelo AFG1022 da Tektronix que apresenta 2 canais independentes, largura de banda de 25 MHz, resolução de 14 bits, taxa de amostragem de 125 MS/s e memória de onda arbitrária de 8.000 pontos. Em seguida, optoacopladores modelo 6N137 são utilizados para isolar o gerador de funções. A alimentação desses circuitos integrados é realizada pelos pinos 3.3V e GND da FPGA. Por sua vez, a plataforma FPGA utilizada foi a DE10 Nano da Intel, cujo chip é um Intel Cyclone V SoC (5CSEBA6U23I7), contando com 110.000 elementos lógicos disponíveis para lógica programável, processador dual-core ARM cortex-A9 e memória embarcada de 5570 Kbits. Já o osciloscópio utilizado foi de modelo DHO814 da Rigol que apresenta 4 canais, largura de banda de 100 MHz, taxa de amostragem de até 1.25 GS/s e resolução de 12 bits. Por fim, utilizou-se também o analisador lógico Signal Tap da Intel da versão 18.1.0 do Quartus Prime, que dispoem de até 128 mil amostras por canal, permite trabalhar com múltiplas instâncias (limitadas pela máxima alocação de memória da FPGA) e até 2048 canais por instância. A descrição de configuração básica do Signal Tap pode ser consultada através das instruções do apêndice D.

#### 4.1 Análise de integridade de sinal através do diagrama de olho

O objetivo deste teste prático é analisar a integridade do sinal de saída do PLL durante o regime permanente. O indicador escolhido para monitorar o resultado foi o *jitter* de pico a pico, medido através de diagrama de olho. O gerador de funções produziu um trem de bits VPPM em 3,75 MHz de 5 bytes (31, 32, 33, 34 e 35 em ASCII) repetidamente em seu canal um. Esse sinal foi conectado ao circuito isolador para depois ser conectado ao osciloscópio no canal um e ao componente PLL para iniciar a sincronização. Então, após a operação do PLL entrar em regime permanente, o sinal de *clock* recuperado foi lido pelo canal dois do osciloscópio, conforme demonstrado no diagrama de blocos da figura 22 e na montagem física apresentada na figura 23.

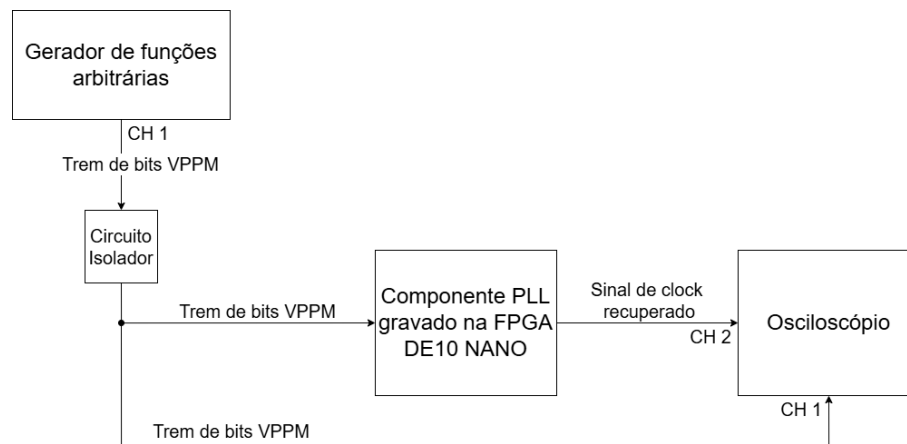


Figura 22 – Diagrama de blocos do teste de diagrama de olho

Fonte: O próprio autor

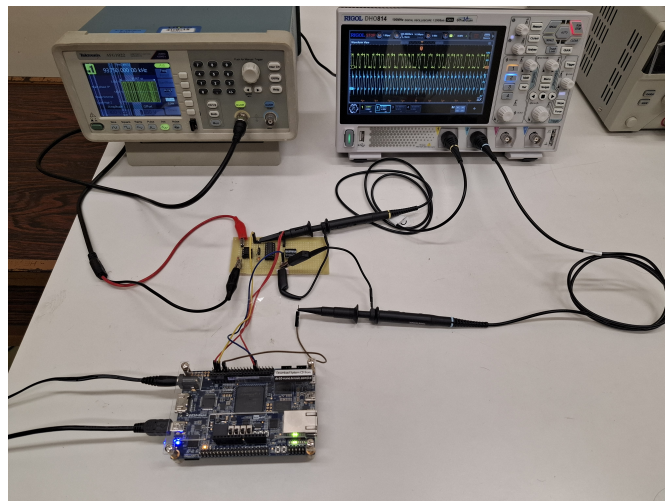
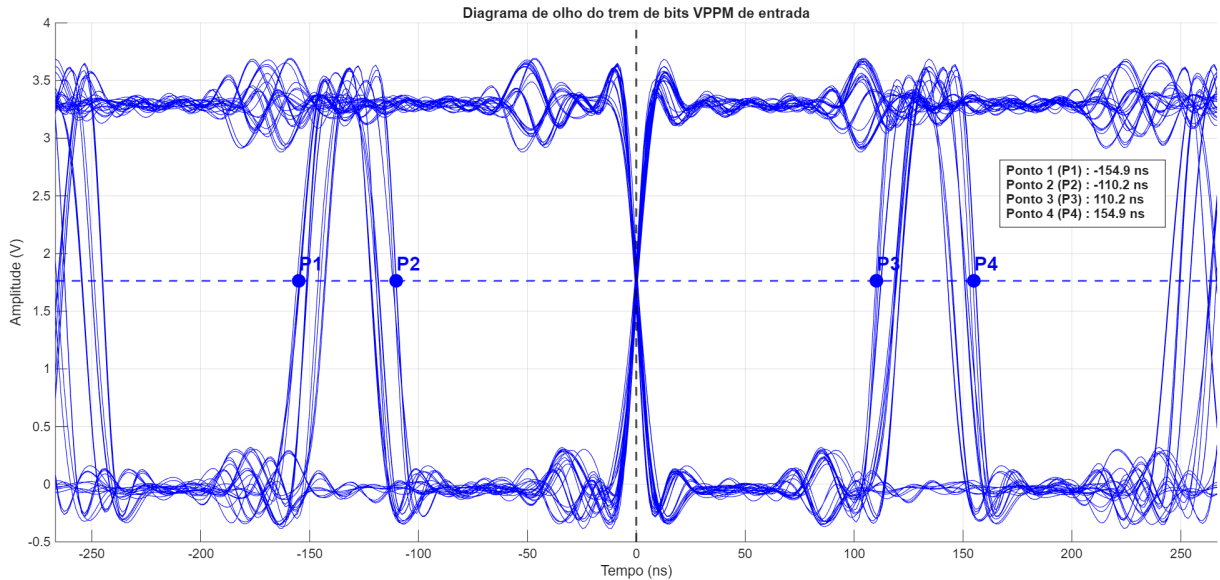


Figura 23 – Foto da montagem física do teste de diagrama de olho

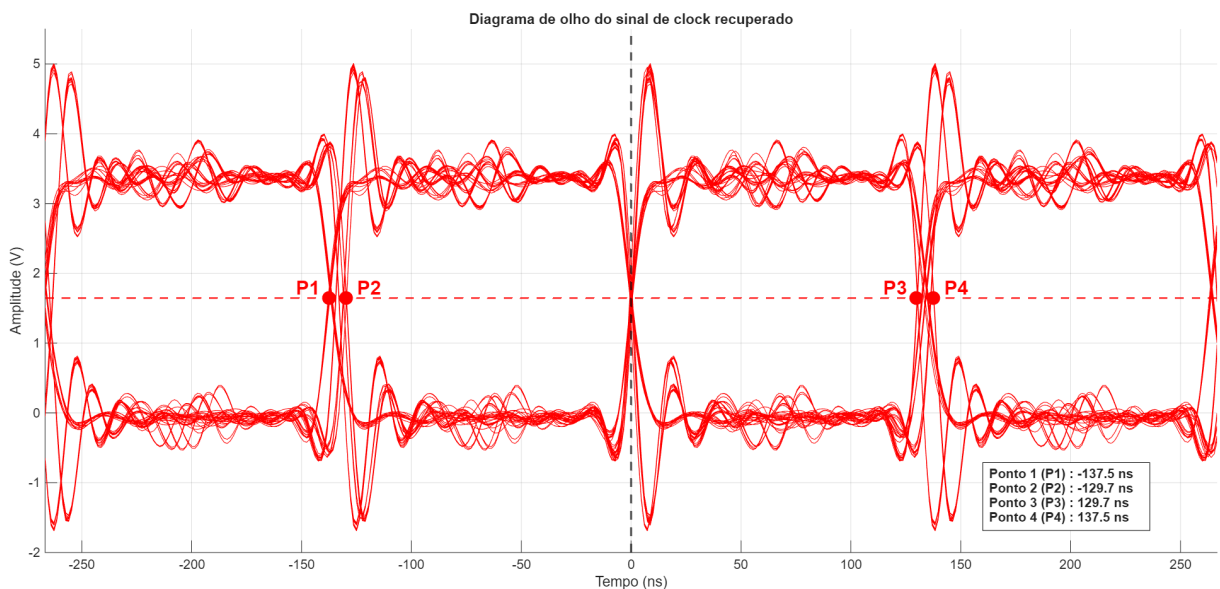
Fonte: O próprio autor

Nesse teste, o osciloscópio fez aquisições durante um período de  $10 \mu\text{s}$  sob uma taxa de amostragem de 625 MS/s. Apesar das medições do trem de bits VPPM de entrada indicarem um *jitter* de pico a pico de 44,7 ns, o ADPLL foi capaz de recuperar um sinal de *clock* com somente 7,8 ns de *jitter* de pico a pico, conforme mostram as figuras 24 e 25.



**Figura 24 – Diagrama de olho centrado em bordas do trem de bits VPPM de entrada**

*Fonte: O próprio autor*

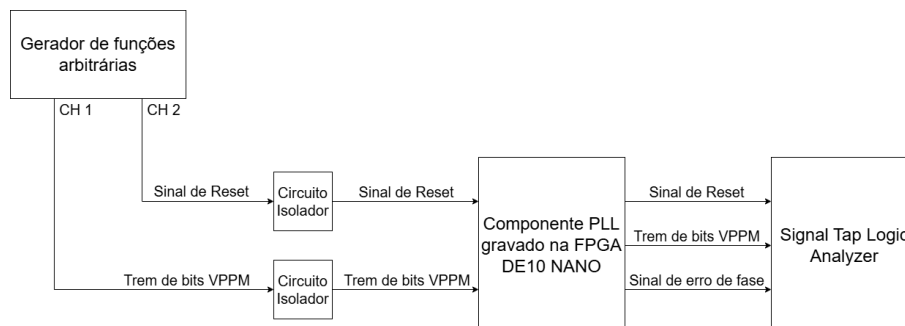


**Figura 25 – Diagrama de olho centrado em bordas do sinal de *clock* recuperado**

*Fonte: O próprio autor*

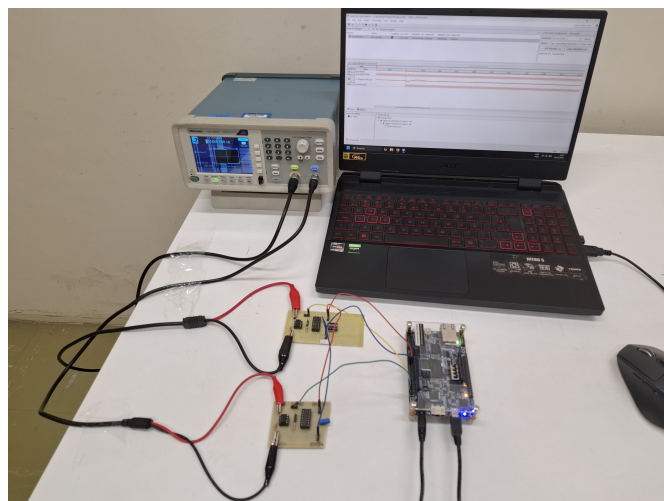
## 4.2 Caracterização da curva de resposta a degrau

O objetivo deste teste prático é caracterizar o comportamento do PLL durante seu regime transitório e sua capacidade de atingir regime permanente. Os indicadores escolhidos para monitorar o resultado foram o tempo de subida ( $T_s$ ), sobressinal ( $M_p$ ), tempo de pico ( $T_p$ ), erro de regime ( $E_r$ ) e tempo de acomodação ( $T_a$ ). Para este estudo, a definição do tempo de acomodação foi baseada no critério de 5% do erro de regime. O gerador de funções produziu um sinal VPPM 3,75 MHz composto exclusivamente por bits zero em seu canal um e uma onda quadrada de 1.875 KHz em seu canal dois para atuar como *reset* do PLL durante estado lógico alto. Ambos sinais foram conectados aos circuitos isoladores, depois ao componente PLL e lidos pelo analisador lógico Signal Tap, como mostra a figura 26 e na montagem física apresentada na figura 27. O Signal Tap foi configurado para sensibilizar seu *trigger* a cada borda de descida do sinal de *reset* e também para ler o sinal interno de erro de fase do bloco de detecção de fase do PLL. Adicionalmente, o sinal interno de *clock* de 120 MHz do PLL foi utilizado como taxa de amostragem do Signal Tapper.



**Figura 26 – Diagrama de blocos do teste de resposta ao degrau**

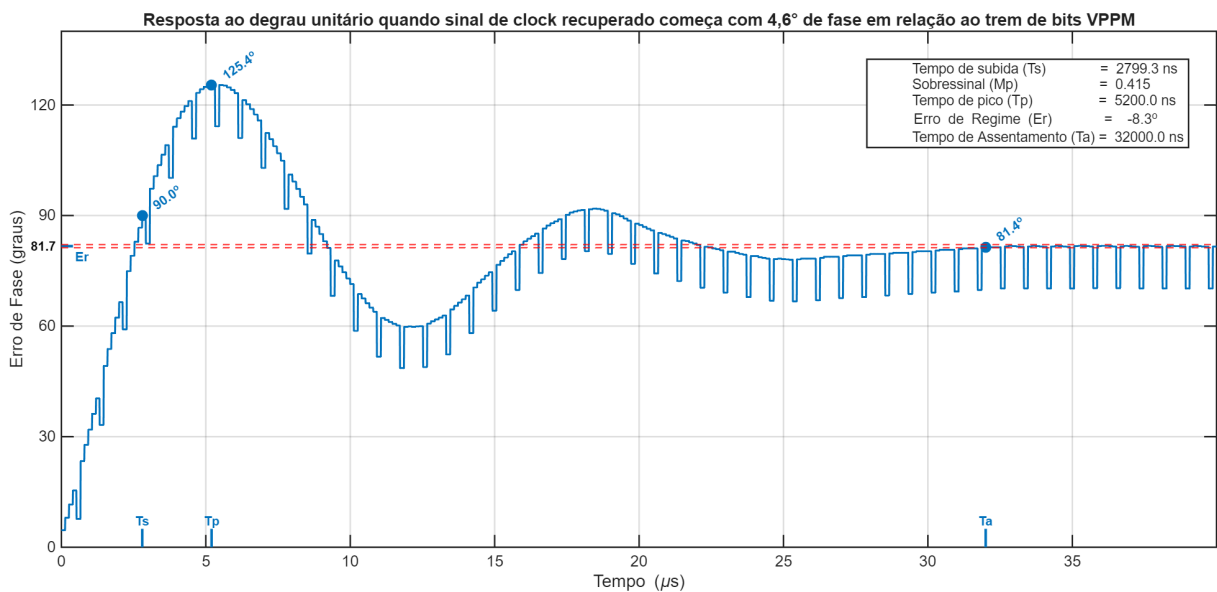
*Fonte: O próprio autor*



**Figura 27 – Foto da montagem física do teste de resposta ao degrau**

*Fonte: O próprio autor*

Nesse teste, a diferença de fase ao longo do tempo entre o trem de bits VPPM e o sinal de *clock* recuperado foi medida em 6 diferentes cenários, revelando que o ADPLL entra em regime permanente em não mais que  $100 \mu s$ . A figura 28 mostra graficamente a aquisição dos indicadores no cenário em que o sinal de *clock* recuperado iniciou a sincronização a uma fase de  $4,6^\circ$  em relação às bordas de subida/descida do trem de bits VPPM. Em outras palavras, nesse cenário o sinal de *clock* estava adiantado em relação ao ponto ótimo de leitura de bits ( $90^\circ$ ) e o bloco de controle do ADPLL agiu para atrasá-lo. Em  $2,8 \mu s$  ( $T_s$ ), atingiu-se o valor desejado de erro de fase, porém houve sobressinal ( $M_p$ ) de até  $0,415$  em  $5,2 \mu s$  ( $T_p$ ) e o componente ADPLL entrou em regime permanente somente em  $32 \mu s$  ( $T_a$ ), com um erro de regime de  $-8,3^\circ$  ( $E_r$ ).



**Figura 28 – Curva de resposta ao degrau unitário quando sinal de clock recuperado começa com  $4,6^\circ$  de fase em relação ao trem de bits VPPM**

*Fonte: O próprio autor*

Para cada um dos 6 cenários testados, a fase inicial entre os sinais foi ajustada para diferentes valores entre  $0^\circ$  e  $180^\circ$  com  $30^\circ \pm 11,25^\circ$  de espaçamento e 64 mil aquisições foram realizadas. O resultado desses testes pode ser conferido através dos indicadores resultantes disponibilizados na tabela 1. Através desses dados, é possível prever o perfil de comportamento do ADPLL. Em questões de performance, espera-se que o ADPLL atinja a fase desejada em  $2,9 \mu s$  ( $T_s$ ), mas com um sobressinal ( $M_p$ ) de até  $63^\circ$  entre  $5 \mu s$  a  $6 \mu s$  ( $T_p$ ). Já as oscilações de sobressinal em geral devem cessar em  $44,8 \mu s$  ( $T_a$ ), mas caso contrário não devem ultrapassar  $100 \mu s$ . Por fim, o erro de regime esperado é  $8,3^\circ$  ( $E_r$ ), sem ultrapassar o valor de  $20^\circ$ .

Diferença de fases inicial	Tempo de subida (Ts)	Sobressinal (Mp)	Tempo de pico (Tp)	Erro de Regime (Er)	Tempo de acomodação (Ta)
173,2°	2798,6 ns	0,754	6000,0 ns	19,9°	37475,0 ns
158,5°	2794,7 ns	0,750	5866,7 ns	8,4°	54000,0 ns
123,8°	2395,1 ns	1,133	5741,7 ns	8,3°	33075,0 ns
56,2°	3593,3 ns	0,187	4933,3 ns	8,3°	67200,0 ns
22,5°	3065,9 ns	0,373	5600,0 ns	8,3°	44800,0 ns
4,6°	2799,3 ns	0,415	5200,0 ns	8,3°	32000,0 ns

**Tabela 1 – Parâmetros da resposta ao degrau medidos conforme variação de diferença de fase inicial**

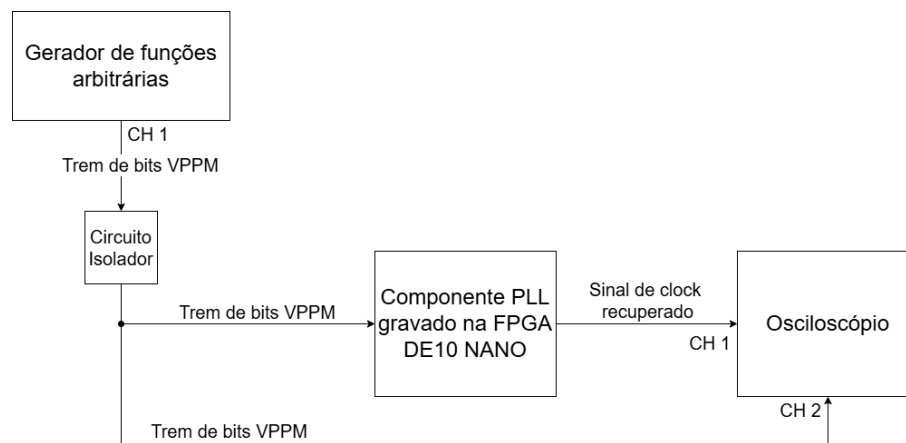
*Fonte: O próprio autor*

### 4.3 Teste de taxa de erro de bits

O objetivo deste teste prático é determinar uma quantidade máxima de erros esperado dentro de um intervalo de bits transmitidos. O indicador escolhido para monitorar o resultado foi a taxa de erro de bit (BeR). Em seu canal um, o gerador de funções produziu um trem de bits VPPM repetidamente, o qual foi composto pela seguinte sequência de 125 bits gerados por um algoritmo PRBS-7:

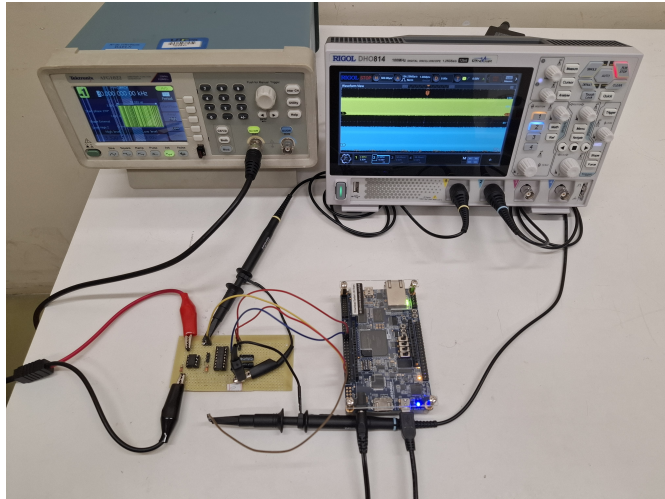
```
0000010000011000010100011110010001011001110101001111101000011110
00100100110110101101111011000110100101110111001100101010111111
```

Esse sinal foi conectado ao circuito isolador e depois ao osciloscópio no canal um, conforme demonstrado no diagrama de blocos da figura 29 e na montagem física apresentada na figura 30. O osciloscópio foi configurado para amostrar os sinais a uma taxa de 156,25 MSa/s, guardando em sua memória 1 milhão de amostras.



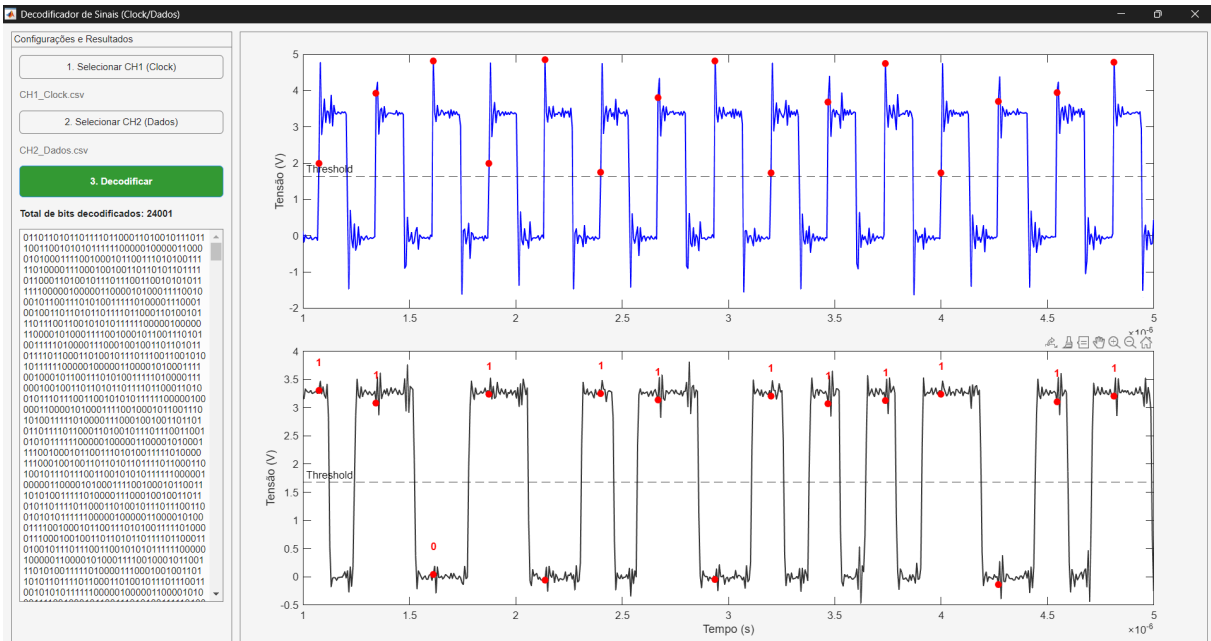
**Figura 29 – Diagrama de blocos do teste de taxa de erro de bit**

*Fonte: O próprio autor*



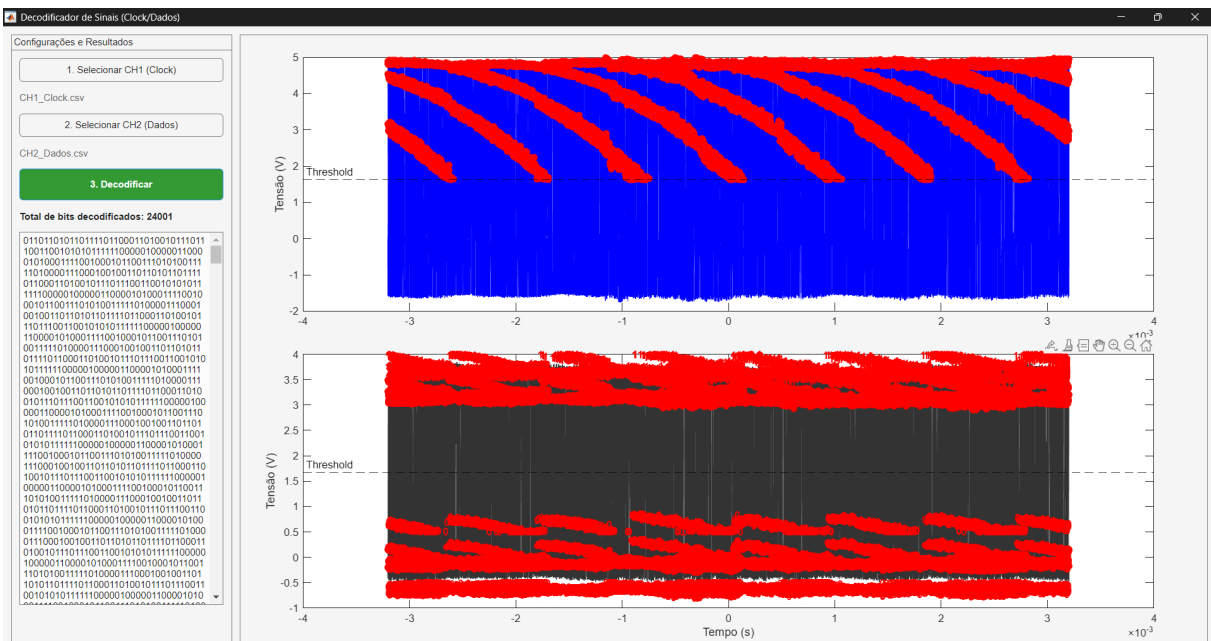
**Figura 30 – Foto da montagem física do teste de taxa de erro de bit**  
*Fonte: O próprio autor*

Nesse teste, o osciloscópio realizou uma leitura durante 6,4 ms e coletou 41 a 42 amostras a cada período do sinal de saída do PLL. Em sequência, um código MATLAB foi utilizado para comparar os dados de entrada com o *clock* recuperado, resultando na decodificação de 24001 bits conforme mostram as figuras 31 e 32. Nessas figuras, o sinal em azul representa o sinal de *clock* recuperado e os pontos vermelhos indicam os momentos em que a interface MATLAB detectou uma borda de subida. Há 24001 pontos vermelhos, um para cada momento em que foi realizada a leitura de um bit. Já o sinal em preto representa o trem de bits VPPM na entrada do ADPLL. A posição dos pontos vermelhos indica que a leitura foi feita durante estado lógico alto ou baixo do trem de bits. Leituras em estado lógico baixo indicaram valores de até 1V e leituras em estado lógico alto indicaram valores de pelo menos 3V. Apesar dos valores das leituras variarem por conta do ruído do sinal produzido pelo gerador de funções, nenhuma leitura foi realizada na faixa de 1.5V a 2.5V durante uma transição de estado lógico. Dessa forma, não houve complicações de ambiguidade no processo de decodificação da sequência de bits. Após carregar os dados na interface MATLAB, uma comparação foi realizada entre a sequência de bits decodificados com a sequência PRBS-7 original. A correspondência foi total, não houveram bits lidos a mais ou a menos e nem erros de leitura. Esse resultado indica que o componente PLL isolado é capaz de operar no mínimo com um BER inferior a  $10^{-3}$ .



**Figura 31 – Interface MATLAB para decodificação do trem de bits VPPM (em preto) com base no sinal de *clock* recuperado (em azul), visão ampliada**

*Fonte: O próprio autor*



**Figura 32 – Interface MATLAB para decodificação do trem de bits VPPM (em preto) com base no sinal de *clock* recuperado (em azul), visão geral**

*Fonte: O próprio autor*

## 5 CONCLUSÃO

A comunicação por luz visível vem se consolidando como uma alternativa promissora às tecnologias tradicionais baseadas em radiofrequência, especialmente em ambientes internos que demandam elevada largura de banda, imunidade a interferências eletromagnéticas e integração com sistemas de iluminação já existentes. Nesse contexto, a recuperação de sincronismo representa uma das etapas fundamentais para garantir a correta reconstrução dos dados transmitidos. Diante desse cenário, este trabalho teve como objetivo desenvolver e validar um componente de clock recovery compatível com sistemas VLC baseados na norma IEEE 802.15.7, empregando modulação VPPM e implementação integralmente digital em FPGA.

Os resultados obtidos demonstram que o objetivo geral do trabalho foi alcançado. O componente ADPLL desenvolvido em VHDL foi capaz de recuperar o sinal de clock a partir de um trem de bits VPPM operando a 3,75 MHz, mantendo sincronismo adequado para a leitura dos dados. As simulações realizadas permitiram verificar o funcionamento da arquitetura proposta, enquanto os experimentos práticos confirmaram sua capacidade de operação em condições reais de hardware. A implementação apresentou baixo consumo de recursos lógicos, ocupando menos de 1% da capacidade da FPGA utilizada, característica que favorece sua aplicação em sistemas maiores e mais complexos.

Os ensaios experimentais evidenciaram resultados consistentes. A análise por diagrama de olho mostrou que o componente foi capaz de recuperar um sinal de clock com jitter pico a pico de apenas 7,8 ns a partir de um sinal de entrada que apresentava jitter de 44,7 ns. Esse resultado demonstra a capacidade do ADPLL de atenuar variações temporais presentes no sinal recebido, contribuindo para aumentar a margem de decisão dos circuitos subsequentes de recuperação de dados. Da mesma forma, os testes de resposta ao degrau revelaram comportamento estável e previsível, com tempo típico de subida inferior a 3 s e tempo de acomodação geralmente inferior a 50 s. Os testes de taxa de erro de bits também indicaram desempenho satisfatório, não sendo observados erros durante a decodificação de mais de 24 mil bits transmitidos, evidenciando a viabilidade funcional da arquitetura proposta.

Sob a perspectiva do estado da arte, os resultados obtidos ocupam uma posição particular. A revisão bibliográfica realizada identificou que a maior parte dos componentes de clock recovery desenvolvidos nas últimas décadas foi projetada para sistemas de comunicação operando em frequências da ordem de gigahertz, utilizando predominantemente modulações NRZ e arquiteturas destinadas a enlaces ópticos de alta velocidade. Em contraste, foram encontradas poucas publicações voltadas especificamente para comunicação por luz visível e nenhuma delas apresentou uma solução dedicada à recuperação de clock para sinais modulados em VPPM conforme os requisitos da camada PHY II da IEEE 802.15.7. Dessa forma, a principal contribuição deste trabalho não está associada à obtenção de taxas de transmissão superiores às encontradas na literatura, mas sim ao preenchimento de uma lacuna tecnológica relacionada à implementação de um circuito de recuperação de clock voltado especificamente para sistemas VLC compatíveis com a norma estudada.

Outro diferencial importante da solução proposta é sua natureza integralmente digital. Enquanto parte significativa dos trabalhos encontrados utiliza arquiteturas analógicas ou mistas, frequentemente dependentes de VCOs, filtros analógicos e processos específicos de fabricação, o ADPLL desenvolvido neste estudo foi implementado exclusivamente com lógica digital sintetizável em VHDL. Essa característica favorece a portabilidade entre plataformas FPGA, reduz a dependência de parâmetros tecnológicos específicos e simplifica futuras modificações ou expansões da arquitetura. Além disso, a baixa utilização de recursos lógicos demonstra que a solução pode ser incorporada a sistemas VLC completos sem comprometer significativamente a disponibilidade de hardware para outras funções.

Apesar dos resultados positivos, algumas limitações devem ser reconhecidas. A primeira delas está relacionada ao erro de regime permanente observado durante os ensaios, que apresentou valores típicos próximos de  $9^\circ$ . Embora esse erro não tenha comprometido a recuperação dos dados, ele indica a existência de espaço para aperfeiçoamentos no controlador PI e na estratégia de detecção de fase utilizada. Outra limitação está associada à ambiguidade inerente entre bordas de subida e descida do sinal VPPM, impossibilitando a identificação direta da polaridade dos dados sem a utilização de um padrão de sincronização ou start byte previamente conhecido. Adicionalmente, os testes foram realizados em ambiente controlado utilizando sinais elétricos gerados por instrumentos de bancada, não contemplando diretamente efeitos ópticos como atenuação do canal, interferência luminosa externa, dispersão espacial ou limitações impostas por luminárias LED comerciais.

Entretanto, essas limitações não invalidam a aplicabilidade da solução proposta. Pelo contrário, os resultados experimentais demonstram que o componente é capaz de estabilizar seu clock recuperado em torno da condição ideal de aproximadamente  $90^\circ$  em relação às transições do sinal VPPM. Essa característica é particularmente importante, pois posiciona os instantes de amostragem na região de máxima abertura temporal do sinal recebido, minimizando a probabilidade de erros decorrentes de jitter ou de transições de estado. Em sistemas VLC, essa condição representa um requisito fundamental para a correta recuperação dos dados, permitindo afirmar que o componente desenvolvido apresenta capacidade operacional compatível com aplicações reais de comunicação por luz visível.

Como perspectivas futuras, recomenda-se a integração do ADPLL com blocos completos de recuperação de dados e decodificação VPPM, permitindo a construção de um receptor VLC totalmente funcional. Também são sugeridos estudos envolvendo canais ópticos reais, utilizando luminárias LED comerciais e fotodetectores, bem como a investigação de técnicas avançadas de controle que possam reduzir o erro de regime permanente e acelerar o processo de sincronização. Do ponto de vista de implementação, uma oportunidade de aprimoramento consiste na reestruturação da máquina de estados e da arquitetura interna do código VHDL, buscando uma descrição mais enxuta, eficiente e otimizada em termos de desempenho e utilização de recursos lógicos. Outra linha de pesquisa relevante é o desenvolvimento de versões do componente capazes de operar com sinais VPPM de diferentes razões cíclicas, permitindo a recuperação de clock em sistemas cujos pulsos apresentem proporções distintas de 50%, como

25%, 75% ou outros valores definidos pelos requisitos de comunicação e controle de luminosidade. Adicionalmente, estudos voltados à desambiguação da polaridade do clock recuperado podem ampliar a funcionalidade do sistema, possibilitando identificar não apenas a posição temporal das bordas associadas ao sinal de referência do transmissor, mas também distinguir quais transições correspondem originalmente às bordas de subida e quais às bordas de descida. Por fim, a validação do sistema sob diferentes condições de iluminação ambiente, níveis de ruído e cenários de interferência contribuiria para uma avaliação ainda mais abrangente da robustez da arquitetura proposta.

Conclui-se, portanto, que o componente desenvolvido atende aos requisitos estabelecidos no início deste trabalho, demonstrando que uma arquitetura ADPLL implementada em VHDL pode realizar a recuperação de clock de sinais VPPM compatíveis com a IEEE 802.15.7 de forma eficiente, econômica e reconfigurável. Além de apresentar resultados experimentais satisfatórios, o trabalho contribui para o avanço das pesquisas em comunicação por luz visível ao propor uma solução direcionada a uma lacuna identificada na literatura, fortalecendo a viabilidade da utilização de sistemas VLC em aplicações futuras de comunicação digital em ambientes internos.

## REFERÊNCIAS

- AHMED, S. I.; KWASNIEWSKI, T. A. Efficient simulation of jitter tolerance for all-digital data recovery circuits. *Em: IEEE MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS (MWSCAS)*. 2007. , 2007. p. 1070–1073. DOI: <https://doi.org/10.1109/MWSCAS.2007.4488745>.
- AHMED, S. I.; ORTHNER, K.; KWASNIEWSKI, T. A. Behavioral test benches for digital clock and data recovery circuits using verilog-a. *Em: PROCEEDINGS OF THE IEEE CUSTOM INTEGRATED CIRCUITS CONFERENCE (CICC)*. 2005., 2005. , 2005. p. 297–300. DOI: <https://doi.org/10.1109/CICC.2005.1568664>.
- ALBUQUERQUE, V. M. de *et al.* A simple resonant switched-capacitor LED driver employed as a fast pulse-based transmitter for VLC applications. **IEEE Journal of Emerging and Selected Topics in Power Electronics**, v. 9, n. 1, DOI: <https://doi.org/10.1109/JESTPE.2020.2965505>.
- ALSER, M. H. *et al.* Design and fpga implementation of pll-based quarter-rate clock and data recovery circuit. *Em: 2012 IEEE 4TH INTERNATIONAL CONFERENCE ON INTELLIGENT AND ADVANCED SYSTEMS (ICIAS)*. 2012, Kuala Lumpur, Malaysia. Kuala Lumpur, Malaysia: , 2012. p. 825–830. DOI: <https://doi.org/10.1109/ICIAS.2012.6306128>.
- ASHARI, Z. M.; NORDIN, A. N.; IBRAHIMY, M. I. Design of a 5 ghz phase-locked loop. *Em: 2011 IEEE REGIONAL SYMPOSIUM ON MICRO AND NANOELECTRONICS (RSM)*. 2011, Kota Kinabalu, Malaysia. Kota Kinabalu, Malaysia: , 2011. p. 167–171. DOI: <https://doi.org/10.1109/RSM.2011.6088316>.
- ASSAAD, M.; CUMMING, D. R. S. Cmos ic design and verilog-a modelling of 10-gb/s pll-based deserializer for inter-chip communication in soc. *Em: INTERNATIONAL SYMPOSIUM ON SYSTEM-ON-CHIP*. 2007. , 2007. DOI: <https://doi.org/10.1109/ISSOC.2007.4427420>.
- ASSAAD, M.; HARB, A. A synthesizable serial link for point-to-point communication in soc/noc. *Em: PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON MICROELECTRONICS (ICM)*. 2017., 2018. , 2018. p. 1–4. DOI: <https://doi.org/10.1109/ICM.2017.8268824>.
- BARREIRA, A. R. **Sistema Protótipo de Comunicação via Luz Visível (VLC/LiFi) para Fins Educacionais**. 2024. 120 p. Dissertação (Mestrado) — Universidade Federal Fluminense Niterói 2024.
- CHEN, Y. *et al.* Verilog hdl modeling and design of 10 gb/s serdes full-rate cdr in 65 nm cmos. **High Technology Letters**, v. 20, n. 2, DOI: <https://doi.org/10.3772/j.issn.1006-6748.2014.02.005>.
- DEVI, M. S.; SAKETH, M.; MUTHUMEENAKSHI, K. Design of costas loop for carrier recovery mechanism of bpsk demodulation. *Em: PROCEEDINGS OF THE 2017 INTERNATIONAL CONFERENCE ON INTELLIGENT COMPUTING AND CONTROL SYSTEMS (ICICCS)*. 2017. , 2017. p. 424–429. DOI: <https://doi.org/10.1109/ICCONS.2017.8250758>.
- ELRABAA, M. E. S. A digital clock re-timing circuit for on-chip source-synchronous serial links. *Em: PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON MICROELECTRONICS (ICM)*. 2006. , 2006. p. 206–209. DOI: <https://doi.org/10.1109/ICM.2006.373303>.
- ELRABAA, M. E. S. A portable clock recovery circuit (crc) for systems-on-chip serial data communication. *Em: PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON MICROELECTRONICS (ICM)*. 2006. , 2006. p. 198–201. DOI: <https://doi.org/10.1109/ICM.2006.373301>.

- GHASSEMI, A.; ABRAMS, D.; LITTLE, T. D. C. A tutorial on visible light communication: Modulation and signal processing. *Em: 2014 IEEE PHOTONICS CONFERENCE (IPC)*. 2014, San Diego, CA, USA. San Diego, CA, USA: , 2014. p. 1–2.
- HAO, N.; ZHANG, M.; ZHANG, Y. An ook based visible light communication system for short distance. *Em: 7TH INTERNATIONAL ICST CONFERENCE ON COMMUNICATIONS AND NETWORKING IN CHINA (CHINACOM)*. 2012, Kunming, China. Kunming, China: , 2012. p. 306–308. DOI: <https://doi.org/10.1109/ChinaCom.2012.6417496>.
- HAYKIN, S. **Introdução aos Sistemas de Comunicação**. 2. ed. 2006.
- HE, J. *et al.* Efficient sampling scheme based on length estimation for optical camera communication. **IEEE Photonics Technology Letters**, v. 31, n. 11, DOI: <https://doi.org/10.1109/LPT.2019.2910118>.
- HEO, D.-H. *et al.* An analysis of 32-gb/s and full-rate phase interpolator based clock and data recovery. *Em: PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON ELECTRONICS, INFORMATION AND COMMUNICATION (ICEIC)*. 2024. , 2024. DOI: <https://doi.org/10.1109/ICEIC61013.2024.10457116>.
- HONG, D. S.; EL-GAMAL, M. N. An efficient design and verification workflow for evaluating the proper operation of clock and data recovery systems. *Em: IEEE MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS (MWSCAS)*. 2005., 2005. , 2005. p. 1827–1830. DOI: <https://doi.org/10.1109/MWSCAS.2005.1594478>.
- HSIEH, M.-T.; SOBELMAN, G. E. Clock and data recovery with adaptive loop gain for spread spectrum serdes applications. *Em: IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS (ISCAS)*. 2005. , 2005. p. 4883–4886. DOI: <https://doi.org/10.1109/ISCAS.2005.1465727>.
- HUO, Q. *et al.* Fpga-based 120 mbps online visible light communication system with rgb leds. *Em: 2016 15TH INTERNATIONAL CONFERENCE ON OPTICAL COMMUNICATIONS AND NETWORKS (ICOON)*. 2017. , 2017. DOI: <https://doi.org/10.1109/ICOON.2016.7875598>.
- JURADO-VERDU, C. *et al.* Rgb synchronous vlc modulation scheme for occ. *Em: 2018 11TH INTERNATIONAL SYMPOSIUM ON COMMUNICATION SYSTEMS, NETWORKS AND DIGITAL SIGNAL PROCESSING (CSNDSP)*. 2018. , 2018. DOI: <https://doi.org/10.1109/CSNDSP.2018.8471829>.
- KAKEHBRA, Z. *et al.* A fast-lock, low jitter, high-speed half-rate cdr architecture with a composite phase detector (cpd). *Em: PROCEEDINGS OF THE 26TH INTERNATIONAL CONFERENCE ON MIXED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS (MIXDES)*. 2019. , 2019. p. 141–146. DOI: <https://doi.org/10.23919/MIXDES.2019.8787120>.
- KANG, J.; LEE, C. Digital clock data recovery circuit for s/pdif. *Em: 2016 IEEE ASIA PACIFIC CONFERENCE ON CIRCUITS AND SYSTEMS (APCCAS)*. 2017. , 2017. p. 325–326. DOI: <https://doi.org/10.1109/APCCAS.2016.7803965>.
- KHANDAL, D.; JAIN, S. Li-Fi (light fidelity): The future technology in wireless communication. **International Journal of Computer Applications**, v. 98, n. 9,
- KUBÍČEK, M. Simulation of digital clock and data recovery of strongly disturbed signals. *Em: 17TH INTERNATIONAL CONFERENCE RADIOELEKTRONIKA*. 2007. , 2007. DOI: <https://doi.org/10.1109/RADIOELEK.2007.371478>.

KUMAR, M. K.; PASUPATHY, K. R.; BINDU, B. Design of finfet based all-digital dll for multiphase clock generation. *Em: 2015 IEEE 12TH INTERNATIONAL CONFERENCE ON ELECTRONICS, ENERGY, ENVIRONMENT, COMMUNICATION, COMPUTER CONTROL (INDICON)*. 2016. , 2016. DOI: <https://doi.org/10.1109/INDICON.2015.7443371>.

KUMAR, V.; KHOSLA, M. Design of a low power delay locked loop based clock and data recovery circuit. *Em: ANNUAL IEEE INDIA CONFERENCE (INDICON)*. 2011, Hyderabad, India. Hyderabad, India: , 2011. DOI: <https://doi.org/10.1109/INDCON.2011.6139507>.

LATHI, B. P.; DING, Z. **Modern Digital and Analog Communication Systems**. 4. ed. 2009.

LIU, M.-C. *et al.* A low-power reference-less clock/data recovery for visible light communication devices requiring low data throughput. **Microsystem Technologies**, v. 26, n. 1, DOI: <https://doi.org/10.1007/s00542-019-04541-w>.

LIU, Y. F. *et al.* Ac-based phosphor led visible light communication by utilizing novel signal modulation. **Optical and Quantum Electronics**, v. 45, n. 10, DOI: <https://doi.org/10.1007/s11082-013-9716-y>.

MAS-MACHUCA, C. *et al.* Techno-economics of LiFi compared to Wi-Fi in industrial IoT applications. *Em: IECON 2022 – 48TH ANNUAL CONFERENCE OF THE IEEE INDUSTRIAL ELECTRONICS SOCIETY*. 2022, Brussels, Belgium. Brussels, Belgium: , 2022. p. 1–5.

MENDOZA, E. J. A. *et al.* A 10gb/s pi-based quarter rate all-digital cdr with improved linearity. *Em: PROCEEDINGS OF THE IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS (ISCAS)*. 2025. , 2025. DOI: <https://doi.org/10.1109/ISCAS56072.2025.11044203>.

MULLER, P.; LEBLEBICI, Y. **CMOS Multichannel Single-Chip Receivers for Multi-Gigabit Optical Data Communications**. DOI: <https://doi.org/10.1007/978-1-4020-5912-4> 2007. ISBN 9781402059117. DOI: <https://doi.org/10.1007/978-1-4020-5912-4>.

MULLER, P. *et al.* Top-down design of a low-power multi-channel 2.5-gbit/s/channel gated oscillator clock-recovery circuit. *Em: DESIGN, AUTOMATION AND TEST IN EUROPE CONFERENCE (DATE)*. 1., 2005. , 2005. p. 258–263. DOI: <https://doi.org/10.1109/DATE.2005.315>.

NASSAR, A. *et al.* Multichannel clock and data recovery: A synchronous approach. **IEEE Transactions on Circuits and Systems II: Express Briefs**, v. 57, n. 5, DOI: <https://doi.org/10.1109/TCSII.2010.2047308>.

PATEL, K.; ASHOK, R.; GUPTA, S. Equalizer-free clock recovery for pam-4 optical interconnects. *Em: INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING AND COMMUNICATIONS (SPCOM)*. 2020. , 2020. DOI: <https://doi.org/10.1109/SPCOM50965.2020.9179629>.

PATIL, J.; HE, L.; JONES, M. Clock and data recovery for a 6 gb/s serdes receiver. *Em: 2010 3RD IEEE INTERNATIONAL CONFERENCE ON COMPUTER SCIENCE AND INFORMATION TECHNOLOGY (ICCSIT)*. 2010, Chengdu, China. Chengdu, China: , 2010. p. 217–221. DOI: <https://doi.org/10.1109/ICCSIT.2010.5564973>.

PAULI, G.; KAMP, J. **LiFi: Communication at the Speed of Light and the Emergence of the Internet of People**. 1. ed. 2019.

PEYCHET, S.; DESGREYS, P.; WARE, C. Optoelectronic pll simulation and optimisation with vhdl-ams. *Em: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL TECHNOLOGY (ICIT)*. 1., 2004. , 2004. p. 50–55.

- RADHA, M. *et al.* An implementation of serial interface engine with transceiver using verilog hdl. *Em: PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON ENERGY, MATERIALS AND COMMUNICATION ENGINEERING (ICEMCE)*. 2023. , 2023. DOI: <https://doi.org/10.1109/ICEMCE57940.2023.10433977>.
- RAZAVI, B. Challenges in the design of high-speed clock and data recovery circuits. **IEEE Communications Magazine**, v. 40, n. 8,
- REHMAN, S. U. *et al.* Visible light communication: A system perspective – overview and challenges. **Sensors**, v. 19, n. 5,
- RICCI, S.; CAPUTO, S.; MUCCHI, L. Fpga-based visible light communications instrument for implementation and testing of ultralow latency applications. **IEEE Transactions on Instrumentation and Measurement**, v. 72,,
- RODRIGUES, L. *et al.* A packet-based analog m-cap visible light communication system for internet of things. *Em: PROCEEDINGS OF THE 14TH INTERNATIONAL SYMPOSIUM ON COMMUNICATION SYSTEMS, NETWORKS AND DIGITAL SIGNAL PROCESSING (CSNDSP)*. 2024. , 2024. p. 389–394. DOI: <https://doi.org/10.1109/CSNDSP60683.2024.10636661>.
- ROSHNA, T. R. *et al.* Design and implementation of digital costas loop and bit synchronizer in fpga for bpsk demodulation. *Em: 2013 INTERNATIONAL CONFERENCE ON CONTROL, COMMUNICATION AND COMPUTING (ICCC)*. 2013, Thiruvananthapuram, India. Thiruvananthapuram, India: , 2013. p. 39–44. DOI: <https://doi.org/10.1109/ICCC.2013.6731621>.
- SAWABY, A. M. *et al.* A 10 gb/s serdes transceiver. *Em: PROCEEDINGS OF THE 3RD NOVEL INTELLIGENT AND LEADING EMERGING SCIENCES CONFERENCE (NILES)*. 2021. , 2021. p. 389–393. DOI: <https://doi.org/10.1109/NILES53778.2021.9600520>.
- SHIN, K.-B. *et al.* Verilog synthesis of usb 2.0 full-speed device phy ip. *Em: 2013 INTERNATIONAL SOC DESIGN CONFERENCE (ISOCC)*. 2013, Busan, South Korea. Busan, South Korea: , 2013. p. 162–165. DOI: <https://doi.org/10.1109/ISOCC.2013.6863961>.
- SONG, S.-J.; NAM, H. Short-distance visible light communication with simple clock data recovery for audio applications. **Journal of the Society for Information Display**, v. 23, n. 10, DOI: <https://doi.org/10.1002/jsid.375>.
- SOULIOTIS, G.; TSIMPOS, A.; VLASSIS, S. Phase interpolator-based clock and data recovery with jitter optimization. **IEEE Open Journal of Circuits and Systems**, v. 4,, DOI: <https://doi.org/10.1109/OJCAS.2023.3295649>.
- SUNDRIYAL, A. *et al.* Design and synthesis of physical layer module of usb 3.0. *Em: PROCEEDINGS OF THE 2025 INTERNATIONAL CONFERENCE ON EMERGING TECHNOLOGIES IN COMPUTING AND COMMUNICATION (ETCC)*. 2025. , 2025. DOI: <https://doi.org/10.1109/ETCC65847.2025.11108447>.
- TARPARA, E. G.; SONI, J.; PATEL, N. Fpga based 1-channel fiber optic analog signal link using 8b/10b encoding scheme. *Em: 2014 INTERNATIONAL CONFERENCE ON ADVANCES IN ELECTRONICS, COMPUTERS AND COMMUNICATIONS (ICAECC)*. 2015. , 2015. DOI: <https://doi.org/10.1109/ICAECC.2014.7002451>.
- THAHIR, T. *et al.* Low-power clock-gated pam4 serdes design for high data rate communication system. *Em: PROCEEDINGS OF THE 7TH INTERNATIONAL CONFERENCE ON INVENTIVE MATERIAL SCIENCE AND APPLICATIONS (ICIMA)*. 2025. , 2025. p. 372–378. DOI: <https://doi.org/10.1109/ICIMA64861.2025.11073935>.

WANG, X. *et al.* Experimental investigation of ultra-fast cdr scheme for high-speed free-space optical data transmission with multiple access. **IEEE Access**, v. 13,, DOI: <https://doi.org/10.1109/ACCESS.2025.3565527>.

WANG, Z.; LIANG, L. An all digital cmos serial link transceiver with 3x over-sampling based data recovery. *Em: 6TH IEEE INTERNATIONAL WORKSHOP ON SYSTEM-ON-CHIP FOR REAL-TIME APPLICATIONS (IWSOC)*. 2006. , 2006. p. 15–19. DOI: <https://doi.org/10.1109/IWSOC.2006.348256>.

XU, S. *et al.* Real-time full-duplex digital video communication system based on parallel-connected micro-led array detection. **Acta Optica Sinica**, v. 45, n. 22, DOI: <https://doi.org/10.3788/AOS251131>.

XU, W. *et al.* Behavioral modeling to circuit design steps of an injection locked cdr in 0.18  $\mu\text{m}$ -cmos. *Em: 2016 10TH IEEE INTERNATIONAL CONFERENCE ON ANTI-COUNTERFEITING, SECURITY AND IDENTIFICATION (ASID)*. 2016. , 2016. p. 96–99. DOI: <https://doi.org/10.1109/ICASID.2016.7873925>.

ZHAI, Z. *et al.* Highly compatible sub-symbol level laser communication/ranging integrated system. **Infrared and Laser Engineering**, v. 53, n. 8, DOI: <https://doi.org/10.3788/IRLA20240194>.

ZHAO, Y. *et al.* A cdr system based on improved second order digital filter, hysteretic voter and phase interpolator. **Chinese Journal of Electronics**, v. 28, n. 6, DOI: <https://doi.org/10.1049/cje.2019.08.006>.

**APÊNDICE A – Tabela de Categorização da Revisão Bibliográfica**

Publicação	Aplicação	Funcionalidade	Estrutura	Natureza	Conceito Base da Arquitetura	Modulação do Sinal	Frequência do sinal de dados	Faixa de Frequência dos dados	Plataforma de Testes	Validação de Conceito
(Peychet; Desgreys; Ware, 2004)	Comunicação óptica digital (fibra óptica)	Clock Recovery	VHDL-AMS	Optoeletrônico	PLL	RZ	10 GHz	10 à 100 GHz	Simulação por Software	Simulações comportamentais
(Ahmed; Othman; Kwasniewski, 2005)	Comunicação serial de ampla aplicação	Clock and Data Recovery	Verilog	Digital-Analógico	Oversampling	NRZ	2.5 GHz	1 à 10 GHz	Simulação por Software	Simulações comportamentais
(Hong; El-gamal, 2005)	Comunicação óptica digital (fibra óptica)	Clock Recovery	VHDL-AMS	Digital-Analógico	PLL	NRZ	2.375 GHz	1 à 10 GHz	Simulação por Software	Simulações comportamentais
(Muller et al., 2005)	Comunicação óptica digital (fibra óptica)	Clock and Data Recovery	Verilog	Digital-Analógico	PLL	NRZ	3 GHz	1 à 10 GHz	Simulação por Software	Simulações comportamentais
(Hsieh; Sobelman, 2005)	Componente SerDes de Interface SATA	Clock and Data Recovery	Verilog	Digital-Analógico	Phase Interpolator	NRZ	30 à 33 kHz	10 à 100 kHz	Simulação por Software	Simulações comportamentais
(Etrabaa, 2006a)	Comunicação serial de ampla aplicação	Clock Recovery	Projetado no software T-spice	Digital	Oversampling	NRZ	2 GHz	1 à 10 GHz	Simulação por Software	Simulações comportamentais
(Wang; Liang, 2006)	Comunicação serial de ampla aplicação	Clock and Data Recovery	Verilog	Digital	DLL	NRZ	1 GHz	1 à 10 GHz	Simulação por Software	Simulações comportamentais
(Etrabaa, 2006b)	SSL instrumentation/sensor data transmission	Clock Recovery	Projetado no software T-spice	Digital	Oversampling	NRZ	2 GHz	1 à 10 GHz	Simulação por Software	Simulações comportamentais
(Kubiček, 2007)	Comunicação serial de ampla aplicação	Clock Recovery	VHDL-AMS	Digital-Analógico	PLL	NRZ	155.52 MHz	100 à 1000 MHz	Simulação por Software	Simulações comportamentais
(Kubiček, 2007)	Comunicação serial de ampla aplicação	Clock Recovery digital descrito em VHDL	VHDL	Digital	Oversampling	NRZ	155.52 MHz	100 à 1000 MHz	Simulação por Software	Simulações comportamentais
(Asaad; Cumming, 2007)	Componentes SerDes de ampla aplicação	Clock and Data Recovery	Verilog	Digital-Analógico	PLL	NRZ	10 Gb/s Quarter Rate (2.7 GHz por fase)	1 à 10 GHz	Simulação por Software	Simulações comportamentais
(Ahmed; Kwasniewski, 2007)	Componentes SerDes de ampla aplicação	Data Recovery	Verilog	Digital	Oversampling	NRZ	2.5 GHz	1 à 10 GHz	Simulação por Software	Simulações comportamentais
(Nassar et al., 2010)	Comunicação serial de ampla aplicação	Clock and Data Recovery	Verilog	Digital-Analógico	PLL	NRZ	2.5 GHz	1 à 10 GHz	Simulação por Software	Simulações comportamentais
(Pali; Ho; Jones, 2010)	Componente SerDes de Interface SATA	Clock Recovery	Verilog-AMS	Digital-Analógico	PLL	NRZ	5 à 8.4 GHz	1 à 10 GHz	Simulação por Software	Simulações comportamentais
(Kumar; Khosla, 2011)	Comunicação serial de ampla aplicação	Clock Recovery	Verilog	Digital-Analógico	PLL	NRZ	1 GHz	1 à 10 GHz	Simulação por Software	Simulações comportamentais
(Ashari; Nordin; Ibrahimy, 2011)	Comunicação óptica digital (fibra óptica)	Clock and Data Recovery	Verilog-AMS	Digital-Analógico	PLL	NRZ	5 GHz	1 à 10 GHz	Simulação por Software	Simulações comportamentais
(Atser et al., 2012)	Comunicação serial de ampla aplicação	Clock Recovery	Verilog	Digital	PLL	NRZ	Quarter Rate até 440 MHz (não especificado)	Não especificado	Simulação por Software	Simulações comportamentais
(Hao; Zhang; Zhang, 2012)	Comunicação por luz visível	Clock and Data Recovery	Não especificado	Não especificado	OOK	NRZ	10 MHz	1 à 10 MHz	Não especificado	Testes em bancada
(Shin et al., 2013)	Demodulação de sinal BPSK (usual em satélites e localização GPS)	Clock and Data Recovery	VHDL	Digital	PLL	BPSK	44 kHz	10 à 100 MHz	ACTEL PROASIC3E FPGA	Simulações comportamentais e testes em bancada
(Roshma et al., 2013)	Comunicação USB 2.0	Clock and Data Recovery	Verilog	Digital	Não especificado	NRZI	60 MHz (5 phases, 12 MHz)	10 à 100 MHz	Xilinx Vertex-5	Testes em bancada
(Liu et al., 2013)	Comunicação por luz visível	Clock and Data Recovery	Não especificado	Não especificado	OOK	NRZ	60 kHz	10 à 100 MHz	Não especificado	Não especificado
(Chen et al., 2014)	Componente SerDes de comunicação óptica digital	Clock and Data Recovery	Verilog	Digital-Analógico	PLL	NRZ	9.95 à 11.5 GHz	1 à 10 GHz	Simulação por Software	Simulações comportamentais
(Tarpata; Sori; Palei, 2015)	Componente de conversão de sinal analógico para digital ótico	Clock and Data Recovery	VHDL	Digital	Oversampling	NRZ	50 MHz e 200 MHz	10 à 100 MHz	Xilinx Spartan 3E	Simulações comportamentais e testes em bancada
(Song; Nam, 2015)	Comunicação por luz visível	Clock and Data Recovery	Não especificado	Não especificado	OOK	NRZ	32 kHz	10 à 100 kHz	Não especificado	Simulações comportamentais e testes em bancada
(Kumar; Pasupathy; Bindu, 2016)	Comunicação serial de ampla aplicação	Clock Recovery	Verilog	Digital	PLL	NRZ	Não especificado	5 GHz	Simulação por Software	Simulações Comportamentais
(Xu et al., 2016)	Comunicação wireless UWB	Clock and Data Recovery	Verilog	Digital-Analógico	PLL	NRZ	250 MHz	100 à 1000 MHz	Simulação por Software	Simulações Comportamentais
(Kang; Lee, 2017)	Comunicação de áudio protocolo S/PDIF	Clock and Data Recovery	Verilog	Digital-Analógico	Não especificado	NRZ	150 à 370 MHz	100 à 1000 MHz	Simulação por Software	Simulações Comportamentais
(Huo et al., 2017)	Comunicação por luz visível	Clock and Data Recovery	Não especificado	Não especificado	OOK	NRZ	40 MHz	10 à 100 MHz	DE2-115	Testes em bancada
(Devi; Saketh; Muthumeenakshi, 2017)	Demodulação de sinal BPSK (usual em satélites e localização GPS)	Clock and Data Recovery	VHDL	Digital-Analógico	PLL	BPSK	Não especificado	Não especificado	Simulação por Software	Simulações comportamentais
(Asaad; Harb, 2018)	Componente SerDes de ampla aplicação	Clock and Data Recovery	Verilog	Digital	PLL	NRZ	167.32 Mb/s a 193.6 Mb/s (41.83–48.4 MHz por fase)	10 à 100 MHz	DE2-70	Simulações comportamentais e testes em bancada
(Lurado-verdu et al., 2018)	Comunicação por luz visível (OCC system)	Clock and Data Recovery	Não especificado	Digital	Wavelength-Division Multiplexing	OOK	0.99 à 3.84 kHz	1 à 10 kHz	Não especificado	Simulações comportamentais e testes em bancada
(Kakehbra et al., 2019)	Comunicação serial de ampla aplicação	Clock and Data Recovery	Verilog	Digital-Analógico	PLL	NRZ	5 GHz	1 à 10 GHz	Simulação por Software	Simulações comportamentais
(Ho et al., 2019)	Comunicação por luz visível (OCC system)	Clock and Data Recovery	Projetado no Matlab	Digital	Wavelength-Division Multiplexing	OOK	672 Hz	100 à 1000 Hz	Não especificado	Simulações comportamentais e testes em bancada
(Zhao et al., 2019)	Comunicação serial de ampla aplicação	Clock and Data Recovery	Verilog	Digital-Analógico	Phase Interpolator	NRZ	3.125 GHz	1 à 10 GHz	Não especificado	Testes em bancada
(Liu et al., 2020)	Comunicação por luz visível	Clock and Data Recovery	ASIC	Digital	PLL	Manchester Coding	1 MHz	1 à 10 MHz	Tape Out Chip	Simulações comportamentais e testes em bancada
(Patel; Ashok; Gupta, 2020)	Comunicação óptica digital (fibra óptica)	Clock and Data Recovery	Verilog	Digital	PLL	PAM-4	100 GHz	10 à 100 GHz	Simulação por Software	Simulações comportamentais
(Sawaby et al., 2021)	Componente SerDes de alta velocidade	Clock and Data Recovery	Verilog	Digital-Analógico	PLL	NRZ	10 GHz	10 à 100 GHz	Simulação por Software	Simulações comportamentais
(Radna et al., 2023)	Comunicação serial de ampla aplicação	Clock and Data Recovery	Verilog	Digital	PLL	NRZ	Não especificado	Não especificado	Simulação por Software	Simulações comportamentais
(Soulitis; Tsimpos; Vlassis, 2023)	Comunicação serial de ampla aplicação	Clock and Data Recovery	Verilog	Digital-Analógico	Phase Interpolator	NRZ	5.83 GHz	10 à 100 GHz	Simulação por Software	Simulações comportamentais
(Rodríguez et al., 2024)	Comunicação por luz visível	Clock and Data Recovery	Não especificado	Optoeletrônico	Oversampling	QAM	50 kHz	10 à 100 kHz	Não especificado	Testes em bancada
(Hao et al., 2024)	Comunicação serial de ampla aplicação	Clock and Data Recovery	Verilog	Digital	Phase Interpolator	NRZ	32 GHz	10 à 100 GHz	Simulação por Software	Simulações comportamentais
(Zhai et al., 2024)	Free Space Optical Communication	Clock Recovery	Não especificado	Digital	Oversampling	DP-QPSK	156.25 MHz	100 à 1000 MHz	Altera Stratix V	Simulações comportamentais e testes em bancada
(Wang et al., 2025)	Free Space Optical Communication	Clock and Data Recovery	Não especificado	Digital-Analógico	Phase Interpolator	PAM-4	10 GHz	10 à 100 GHz	Não especificado	Simulações comportamentais e testes em bancada
(Xu et al., 2025)	Componente SerDes de ampla aplicação	Clock and Data Recovery	Verilog	Digital	Oversampling	PAM-4	112 MHz	100 à 1000 MHz	Simulação por Software	Simulações comportamentais
(Thahir et al., 2025)	Comunicação USB 3.0	Clock and Data Recovery	Verilog	Digital	PLL	NZRI	5 GHz	1 à 10 GHz	Simulação por Software	Simulações comportamentais
(Sundriyal et al., 2025)	Comunicação por luz visível	Clock and Data Recovery	Não especificado	Optoeletrônico	Não especificado	OOK	2 MHz	1 à 10 MHz	Não especificado	Testes em bancada
(Mendoza et al., 2025)	Comunicação serial de ampla aplicação	Clock and Data Recovery	Verilog-AMS	Digital-Analógico	Phase Interpolator	NRZ	2.5 GHz	1 à 10 GHz	Simulação por Software	Simulações comportamentais

**APÊNDICE B – Código VHDL do componente ADPLL**

```
1
2 -----
3 -- BLOCO DE IMPORTAÇÃO DE BIBLIOTECAS
4 -----
5 library ieee;
6 use ieee.std_logic_1164.all;
7 use ieee.numeric_std.all;
8 -----
9
10
11 -----
12 -- BLOCO DE DEFINIÇÃO DA ENTIDADE
13 -----
14 entity ADPLL is
15     port (
16         TransmitterVPPM      : in std_logic;
17         Receiver50M           : in std_logic;
18         Reset                 : in std_logic;
19         ClockRecovery         : out std_logic;
20         Data                  : out std_logic;
21         DataN                 : out std_logic
22     );
23 end entity;
24 -----
25
26
27 -----
28 -- BLOCO DE DEFINIÇÃO DA ARQUITETURA
29 -----
30 architecture a_ADPLL of ADPLL is
31 -----
32
33
34 -----
35 -- BLOCO DE DEFINIÇÃO DO COMPONENTE DA INTEL
36 -----
37 component PLL50M120M is
38     port (
39         refclk      : in std_logic;
40         rst         : in std_logic;
41         outclk_0    : out std_logic;
42         locked      : out std_logic
43     );
44 end component PLL50M120M;
45 -----
46
47
48
49
50
```

```

51
52 -----
53 -- BLOCO DE DEFINIÇÃO DOS SINAIS
54 -----
55 -- Sinais independentes
56 -----
57 signal s_Receiver120M      : std_logic      := '0';
58 signal s_Receiver120MLock : std_logic      := '0';
59 signal s_NCOCounter       : signed(15 downto 0) := (others => '0');
60 signal s_ClockRecovery    : std_logic      := '0';
61 signal s_Data             : std_logic      := '0';
62 signal s_DataN            : std_logic      := '0';
63 -----
64 -- Sinais do Estado 0 - Phase Detection
65 -----
66 constant s_PhaseTargetPos50DTC : signed(15 downto 0) := "01" & (13 downto 0 => '0');
67 constant s_PhaseTargetNeg50DTC : signed(15 downto 0) := "11" & (13 downto 0 => '0');
68 signal s_S0Mem                : std_logic      := '0';
69 signal s_S0Count              : unsigned(1 downto 0) := "00";
70 signal s_S1Flag               : std_logic      := '0';
71 signal s_PhaseError           : signed(15 downto 0) := (others => '0');
72 -----
73 -- Sinais do Estado 1 - Primeiras Operações do Controlador PI
74 -----
75 constant KP                    : integer        := 9;
76 constant KI                    : integer        := 12;
77 signal s_S1Mem                 : std_logic      := '0';
78 signal s_S1Count              : unsigned(1 downto 0) := "00";
79 signal s_S2Flag               : std_logic      := '0';
80 signal s_Proportional          : signed(15 downto 0) := (others => '0');
81 signal s_Integral              : signed(15 downto 0) := (others => '0');
82 -----
83 -- Sinais do Estado 2 - Últimas Operações do Controlador PI
84 -----
85 -- Incremento padrão de 1/32 partes do contador a cada borda de subida do sinal de clock 120Mhz.
86 constant s_NCOBias            : signed(15 downto 0) := "00001" & (10 downto 0 => '0');
87 signal s_S2Mem                : std_logic      := '0';
88 signal s_S2Count              : unsigned(1 downto 0) := "00";
89 -- Correção de fase aplicada junto ao incremento padrão, atualizada pela malha de controle.
90 signal s_NCOWord              : signed(15 downto 0) := (others => '0');
91 -----
92
93
94 -----
95 -- BLOCO DE INÍCIO DE ARQUITETURA
96 -----
97 begin
98 -----
99
100

```

```

101
102 -----
103 -- BLOCO DE CRIAÇÃO E INTERLIGAÇÃO DO PLL DA INTEL
104 -----
105 PLL : PLL50M120M
106     port map(
107         refclk    => Receiver50M,
108         rst       => '0',
109         outclk_0  => s_Receiver120M,
110         locked    => s_Receiver120MLock
111     );
112 -----
113
114
115 -----
116 -- BLOCO DE ITERAÇÃO DO NCO
117 -----
118     process(s_Receiver120M)
119     begin
120         if rising_edge(s_Receiver120M) then
121             -- Reset dos sinais do bloco de iteração do NCO
122             if (Reset = '0') then
123                 s_NCOCOUNTER <= (others => '0');
124             -- Operação padrão do bloco de iteração do NCO
125             else
126                 s_NCOCOUNTER <= s_NCOCOUNTER + s_NCOWord + s_NCOBias;
127             end if;
128         end if;
129     end process;
130     s_ClockRecovery <= s_NCOCOUNTER(15);
131 -----
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150

```

```

151
152 -----
153 -- BLOCO DA PRIMEIRA ETAPA DA MALHA DE CONTROLE - PHASE DETECTION
154 -----
155 process(s_Receiver120M)
156 begin
157     if rising_edge(s_Receiver120M) then
158         -- Reset dos sinais deste bloco
159         if (Reset = '0') then
160             s_S0Mem      <= '0';
161             s_S0Count    <= "00";
162             s_S1Flag     <= '0';
163             s_PhaseError <= (others => '0');
164         -- Operação padrão deste bloco
165         else
166         -- Condição de entrada deste estado
167             if (TransmitterVPPM /= s_S0Mem) then
168         -- Condição inicial de loop deste estado
169                 if (s_S0Count = "00") then
170                     s_S0Count      <= s_S0Count + 1;
171         -- Cálculo de PhaseError durante porção positiva do NCOCOUNTER
172                     if (s_NCOCOUNTER(15) = '0') then
173                         s_PhaseError <= s_PhaseTargetPos50DTC - s_NCOCOUNTER;
174         -- Cálculo de PhaseError durante porção negativa do NCOCOUNTER
175                     else
176                         s_PhaseError <= s_PhaseTargetNeg50DTC - s_NCOCOUNTER;
177                     end if;
178         -- Condição de fim deste estado
179                 elsif (s_S0Count = "11") then
180                     s_S0Mem      <= not(s_S0Mem);
181                     s_S0Count    <= "00";
182                     s_S1Flag     <= not(s_S1Flag);
183         -- Condição de loop deste estado
184                 else
185                     s_S0Count      <= s_S0Count + 1;
186                 end if;
187             end if;
188         end if;
189     end if;
190 end process;
191 -----
192
193
194
195
196
197
198
199
200

```

```

201
202 -----
203 -- BLOCO DA SEGUNDA ETAPA DA MALHA DE CONTROLE - OPERAÇÕES INICIAIS DO CONTROLADOR PI
204 -----
205 process(s_Receiver120M)
206 begin
207     if rising_edge(s_Receiver120M) then
208         -- Reset dos sinais deste bloco
209         if (Reset = '0') then
210             s_S1Mem          <= '0';
211             s_S1Count        <= "00";
212             s_S2Flag         <= '0';
213             s_Proportional    <= (others => '0');
214             s_Integral        <= (others => '0');
215         -- Operação padrão deste bloco
216         else
217             -- Condição de entrada deste estado
218             if (s_S1Flag /= s_S1Mem) then
219                 -- Condição inicial de loop deste estado
220                 if (s_S1Count = "00") then
221                     s_S1Count          <= s_S1Count + 1;
222                     s_Proportional      <= shift_right(s_PhaseError, KP);
223                     s_Integral          <= shift_right(s_PhaseError, KI) + s_Integral;
224                 -- Condição de fim deste estado
225                 elsif (s_S1Count = "11") then
226                     s_S1Mem            <= not(s_S1Mem);
227                     s_S1Count          <= "00";
228                     s_S2Flag           <= not(s_S2Flag);
229                 -- Condição de loop deste estado
230                 else
231                     s_S1Count          <= s_S1Count + 1;
232                 end if;
233             end if;
234         end if;
235     end if;
236 end process;
237 -----
238
239
240
241
242
243
244
245
246
247
248
249
250

```

```

251
252 -----
253 -- BLOCO DA TERCEIRA ETAPA DA MALHA DE CONTROLE - OPERAÇÕES FINAIS DO CONTROLADOR PI
254 -----
255 process(s_Receiver120M)
256 begin
257     if rising_edge(s_Receiver120M) then
258 -- Reset dos sinais deste bloco
259         if (Reset = '0') then
260             s_S2Mem      <= '0';
261             s_S2Count    <= "00";
262             s_NCOWord    <= (others => '0');
263 -- Operação padrão deste bloco
264         else
265 -- Condição de entrada deste estado
266             if (s_S2Flag /= s_S2Mem) then
267 -- Condição inicial de loop deste estado
268                 if (s_S2Count = "00") then
269                     s_S2Count    <= s_S2Count + 1;
270                     s_NCOWord    <= s_Proportional + s_Integral;
271 -- Condição de fim deste estado
272                 elsif (s_S2Count = "11") then
273                     s_S2Mem      <= not(s_S2Mem);
274                     s_S2Count    <= "00";
275 -- Condição de loop deste estado
276                 else
277                     s_S2Count    <= s_S2Count + 1;
278                 end if;
279             end if;
280         end if;
281     end if;
282 end process;
283 -----
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300

```

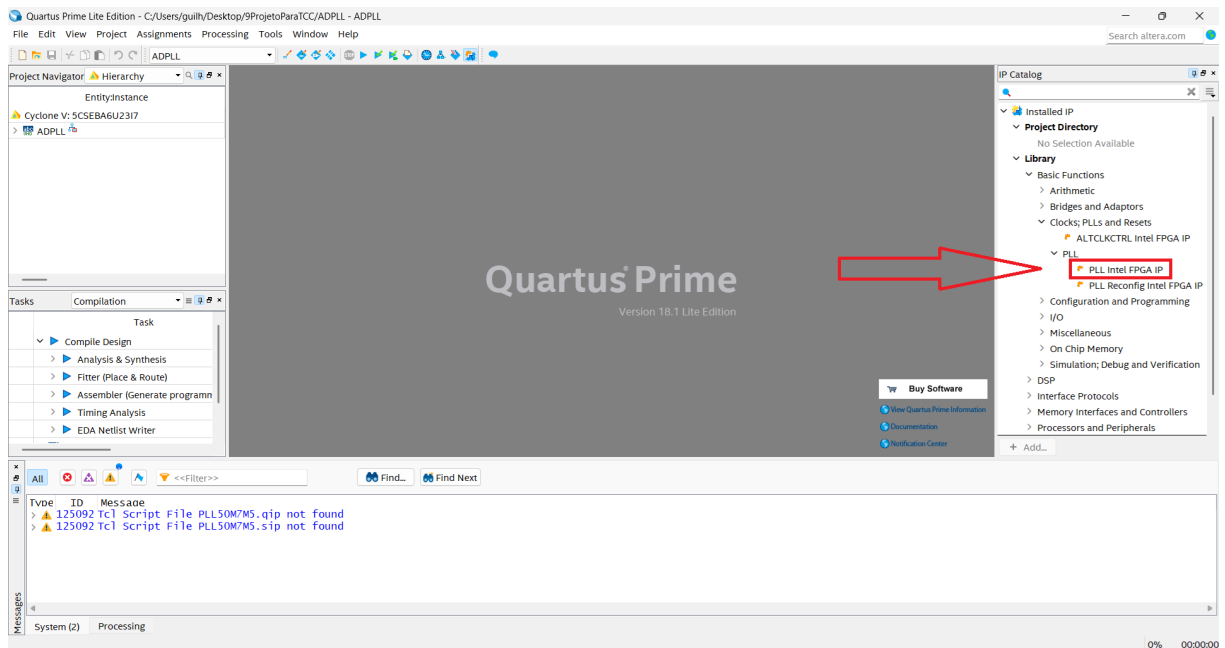
```

301
302 -----
303 -- BLOCO DE DATA RECOVERY
304 -----
305 process(s_ClockRecovery)
306 begin
307     if rising_edge(s_ClockRecovery) then
308 -- Reset dos sinais do bloco de Data Recovery
309         if (Reset = '1') then
310             s_Data    <= '0';
311             s_DataN   <= '0';
312         else
313             s_Data    <= TransmitterVPPM;
314             s_DataN   <= not(TransmitterVPPM);
315         end if;
316     end if;
317 end process;
318 -----
319
320
321 -----
322 -- BLOCO DE CONEXÕES DOS SINAIS DE SAÍDA
323 -----
324 ClockRecovery <= s_ClockRecovery;
325 Data    <= s_Data;
326 DataN   <= s_DataN;
327 -----
328
329
330 -----
331 -- BLOCO DE FIM DE ARQUITETURA
332 -----
333 end architecture;
334 -----

```

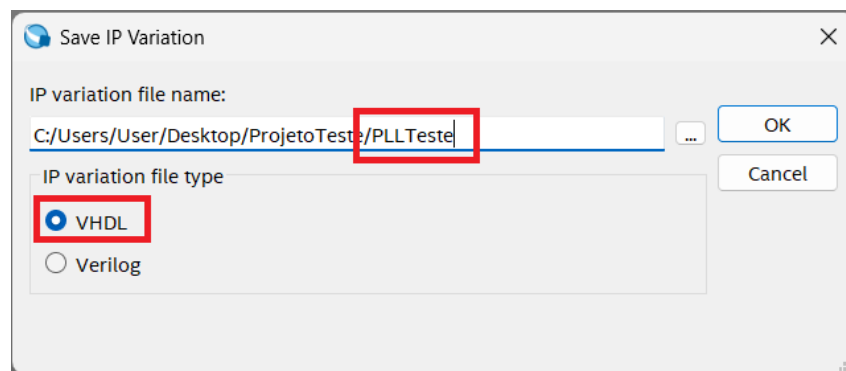
**APÊNDICE C – Passo a passo para configuração básica de um componente PLL pela seção de propriedade intelectual Intel dentro do software Quartus Prime**

Após criar um projeto VHDL no Quartus Prime, é possível configurar multiplicadores ou divisores de *clock* a partir do catálogo de propriedade intelectual da Intel e do *clock* interno da plataforma FPGA. Para esse estudo, foi utilizada uma plataforma DE10 NANO de chip 5CSEBA6U2317, cujo *clock* interno é 50 MHz, e o resultado da multiplicação é 120 MHz. O primeiro passo foi acessar o menu de propriedade intelectual e abrir a opção "PLL Intel FPGA IP", conforme mostra a imagem abaixo.



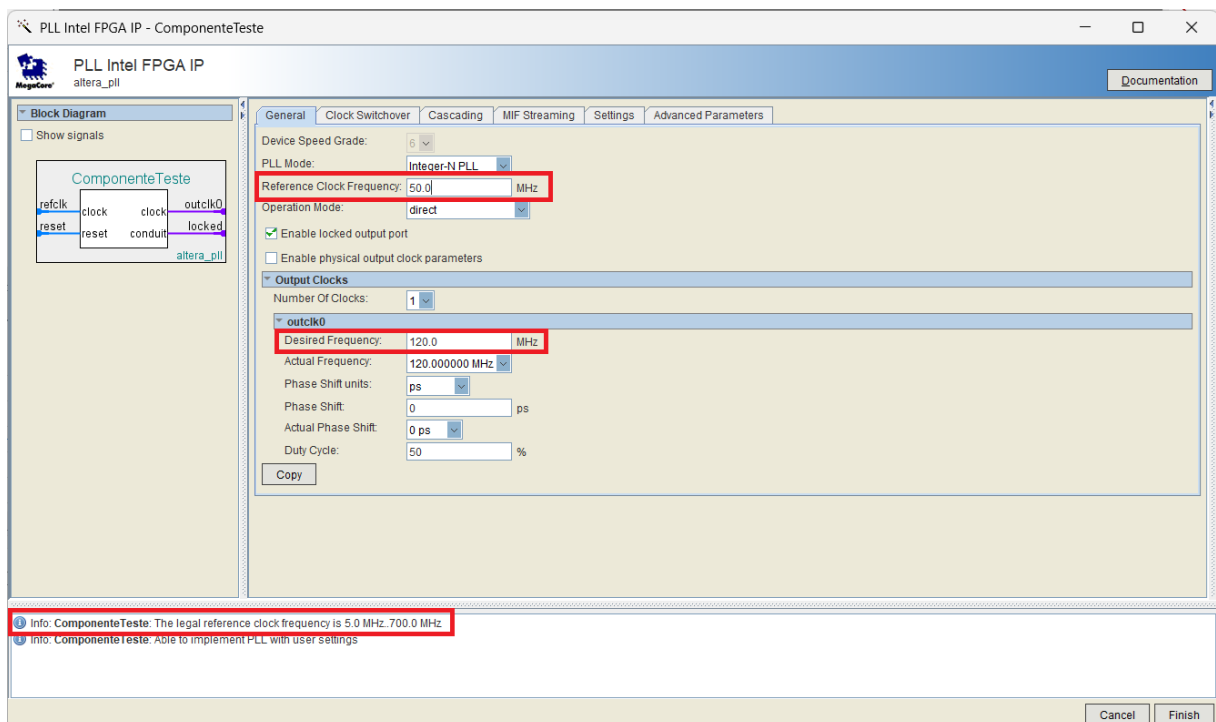
Fonte: O próprio autor

Em sequência, é necessário dar um nome ao componente PLL a ser criado e selecionar se ele deve ser descrito em VHDL ou Verilog, conforme mostra a imagem abaixo.



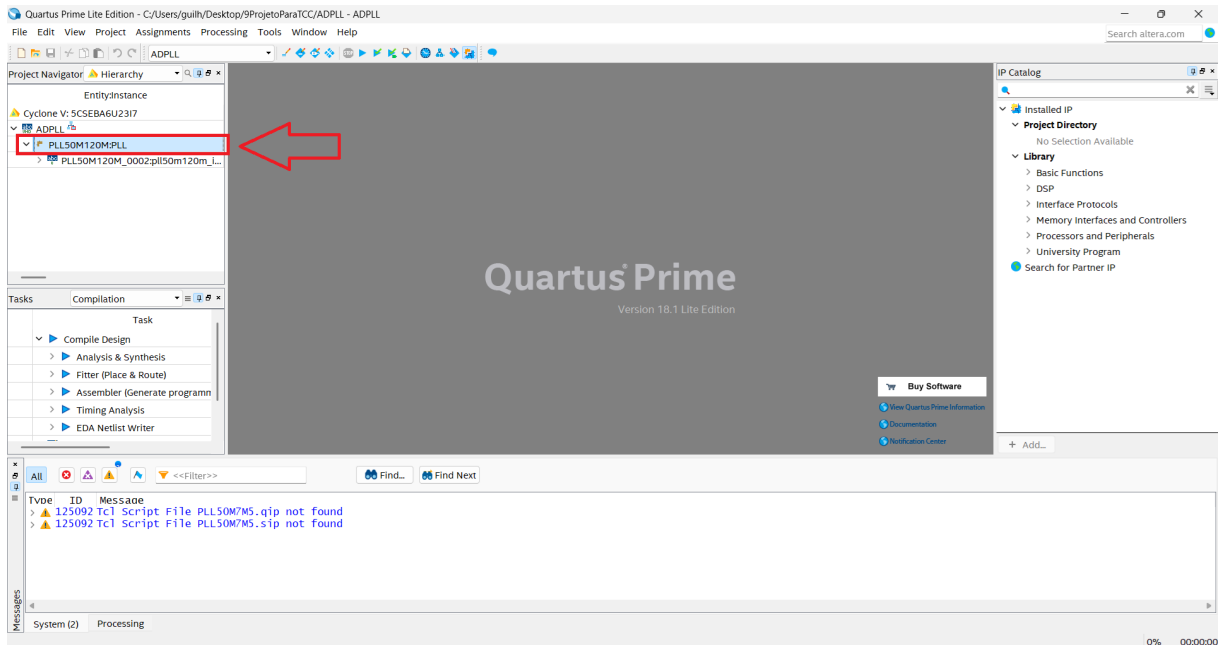
Fonte: O próprio autor

A próxima tela que o software abrirá permite configurar o componente PLL. Dentre os campos apresentados na tela, os essenciais são o "Reference Clock Frequency" e o "Desired Frequency". A frequência de entrada e a de saída devem ser preenchidas nesses campos, respectivamente. Porém, é necessário se atentar às limitações das configurações. Para este estudo, componentes PLL foram limitados pelo software à faixas de frequência de 5 MHz a 700 MHz, conforme indicado pelas notificações na imagem abaixo. Quando as configurações estiverem prontas, basta selecionar a opção "Finish" no canto inferior direito da tela para que o Quartus Prime gere o componente e o adicione ao seu projeto



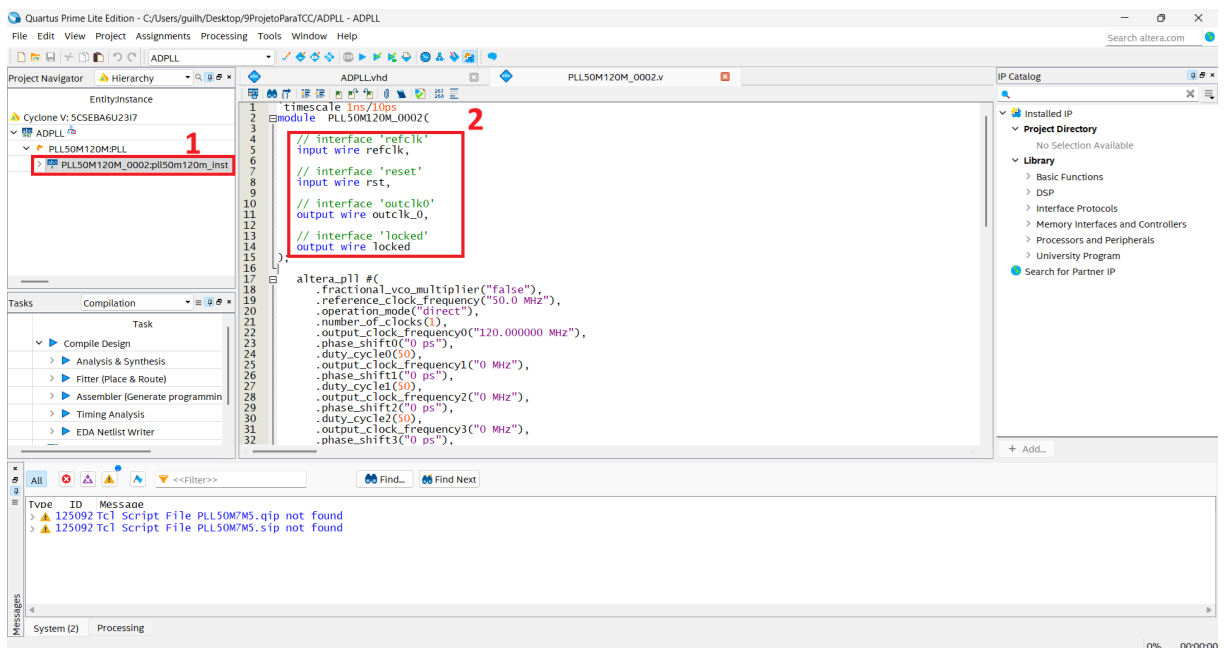
Fonte: O próprio autor

Assim que o processo for finalizado, a instância do componente de propriedade intelectual deve aparecer na aba de navegação de projeto, como por exemplo o componente PLL50M120M indicado na imagem abaixo.



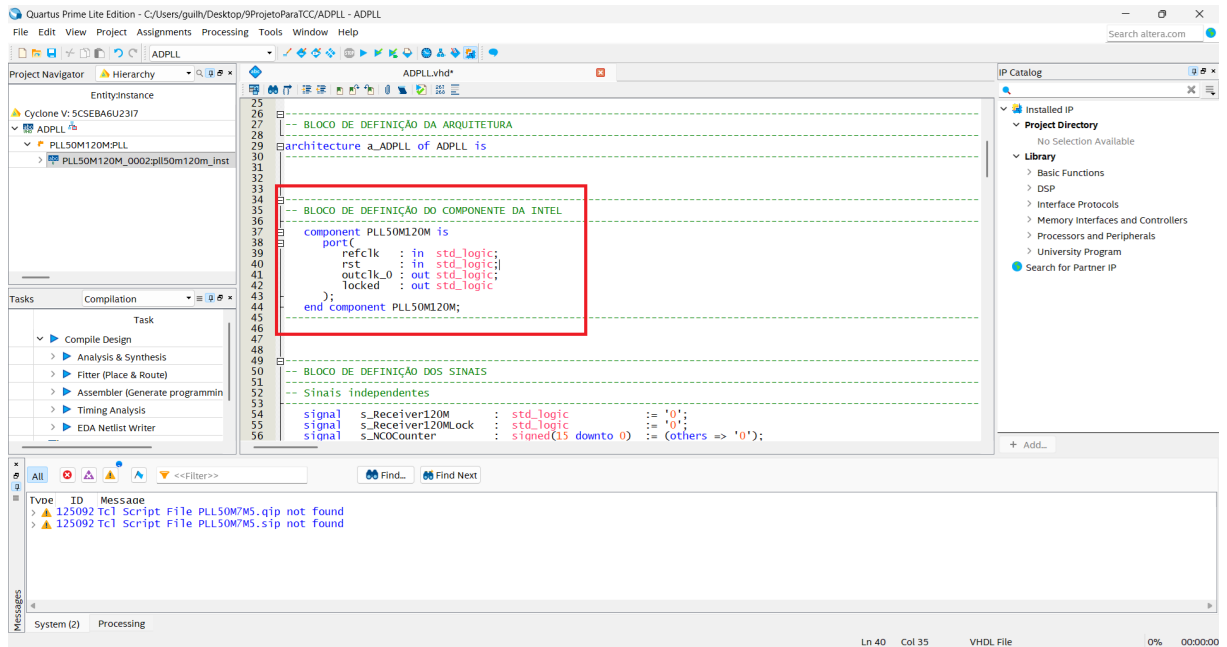
Fonte: O próprio autor

Por fim, é necessário abrir o arquivo do PLL e indentificar suas entradas e saídas para adicionar o componente ao arquivo *toplevel* de seu projeto, conforme mostra a imagem abaixo.



Fonte: O próprio autor

A partir deste ponto, o componente PLL pode ser tratado como um arquivo VHDL instanciado abaixo do arquivo *toplevel* de seu projeto. Para utilizá-lo, basta declará-lo dentro do arquivo *toplevel* e ligar sinais a ele.

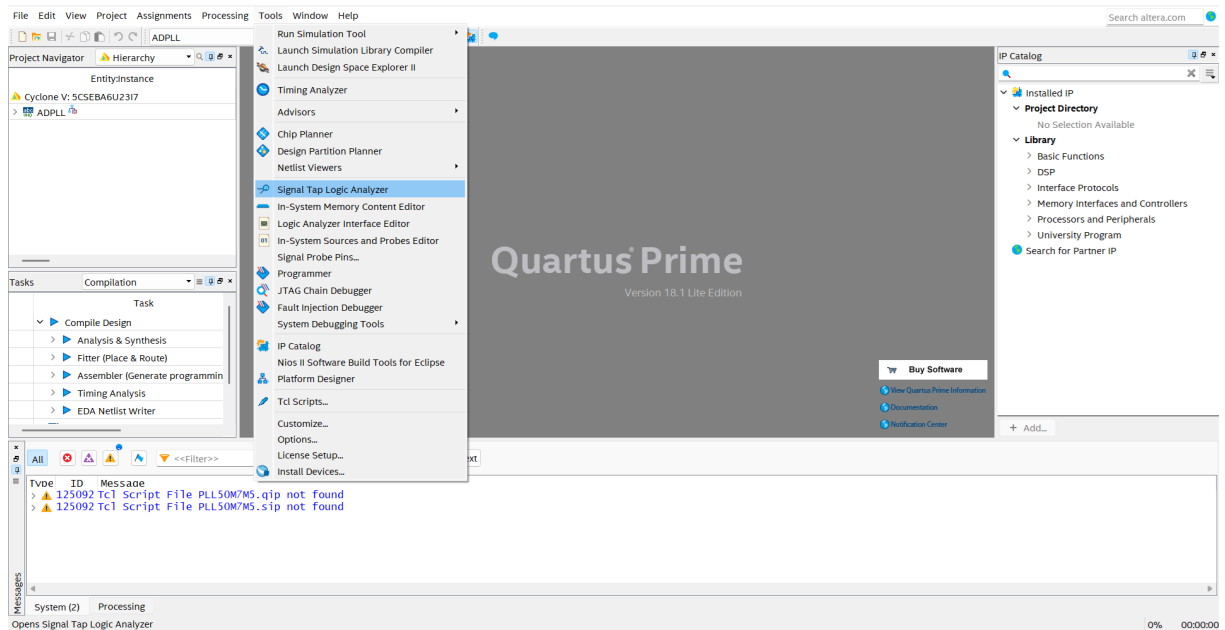


Fonte: O próprio autor

**APÊNDICE D – Passo a passo para configuração básica do Signal Tap Logic Analyzer dentro do software Quartus Prime**

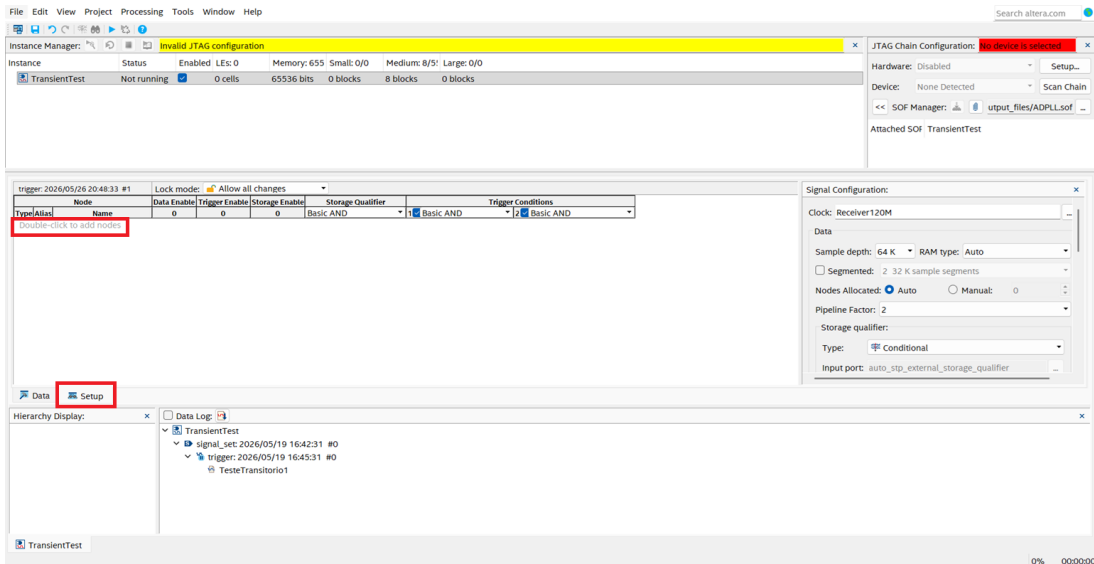
Após validar resultados de simulação de um projeto VHDL no Quartus Prime, é possível configurar um analisador lógico para ler os sinais da FPGA em tempo real durante testes. Esse método é particularmente útil para sinais paralelos, cuja quantidade de bits supera a quantidade de pontas de prova de um osciloscópio. O Signal Tap é capaz de fazer a leitura desses sinais e enviá-los através de sua porta serial ao software Quartus Prime com o próprio cabo USB utilizado para gravação do código.

O primeiro passo é fazer uma compilação completa do seu código VHDL. Assim, o software Quartus Prime saberá quais sinais existem em seu componente VHDL. Essa etapa é necessária para ser capaz de selecionar quais sinais o Signal Tap vai realizar a leitura. Em seguida, é necessário abrir a janela de configuração do Signal Taper, conforme mostra a imagem abaixo.



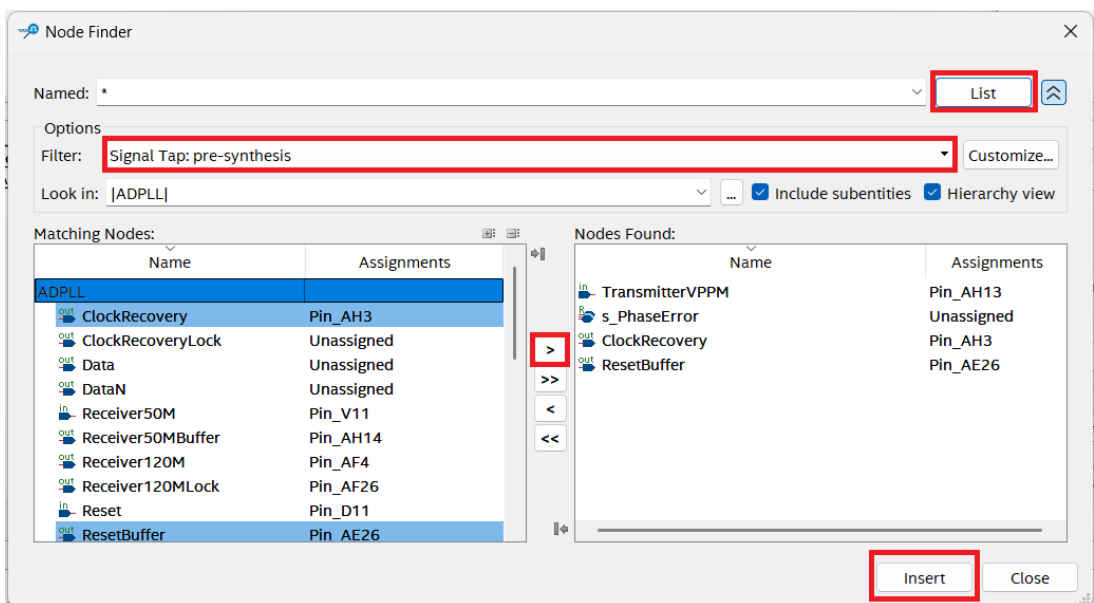
Fonte: O próprio autor

Em sequência, é necessário selecionar quais sinais serão lidos. Esse processo pode ser realizado na aba "Setup" ao clicar duas vezes nas linhas em branco da tabela "Nodes", conforme mostra a imagem abaixo.



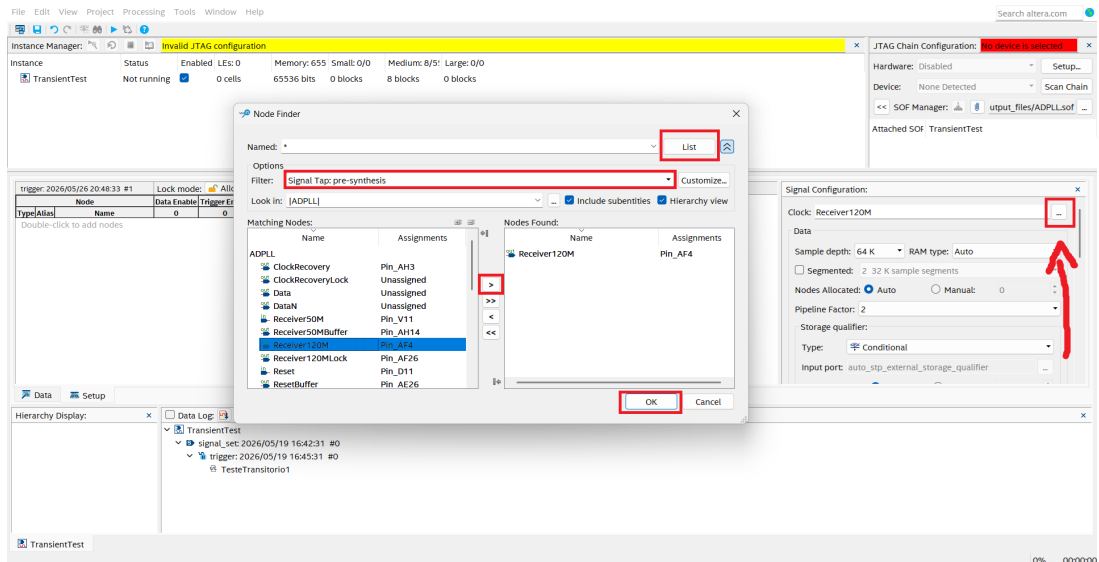
Fonte: O próprio autor

A próxima janela que abrir permitirá a escolha dos sinais a serem lidos. Para encontrar os sinais descritos em seu código, selecione a opção "Signal Tap: pre-synthesis" e aperte o botão "List". Em seguida, selecione os sinais a serem lidos e mova-os para a lista da direita com o botão "»". Termine o processo apertando o botão "Insert", conforme mostra a imagem abaixo.



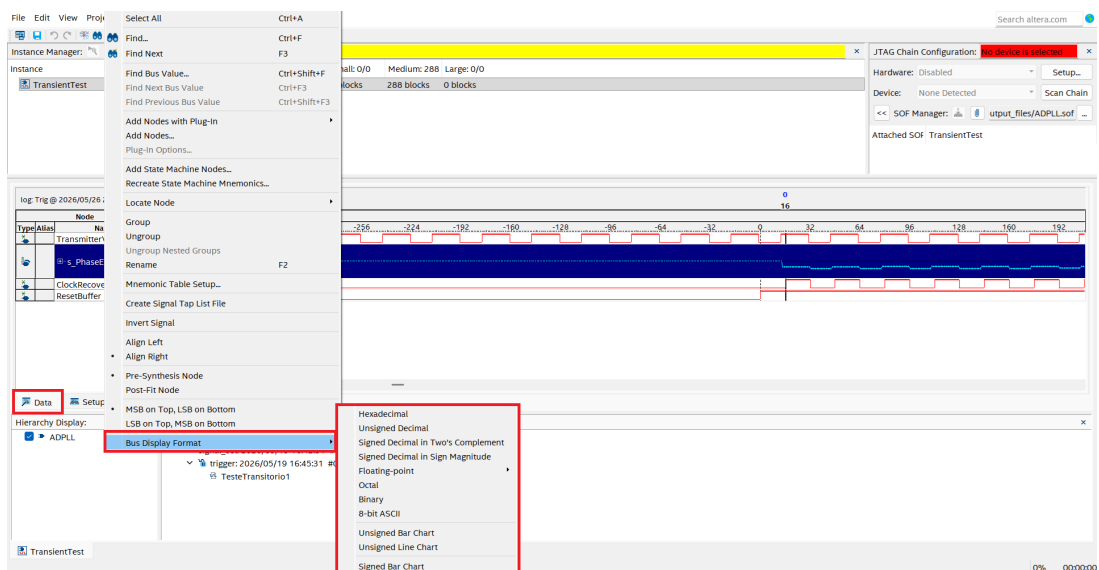
Fonte: O próprio autor

O próximo passo é selecionar um sinal que servirá como taxa de amostragem do signal tapper. No caso desse estudo, foi selecionado o clock de 120 MHz por ser o sinal de maior frequência gerado pela FPGA. Aperte o botão "..." conforme mostra a imagem abaixo e selecione seu sinal através da busca da lista "Signal Tap: pre synthesis", conforme feito na etapa anterior.



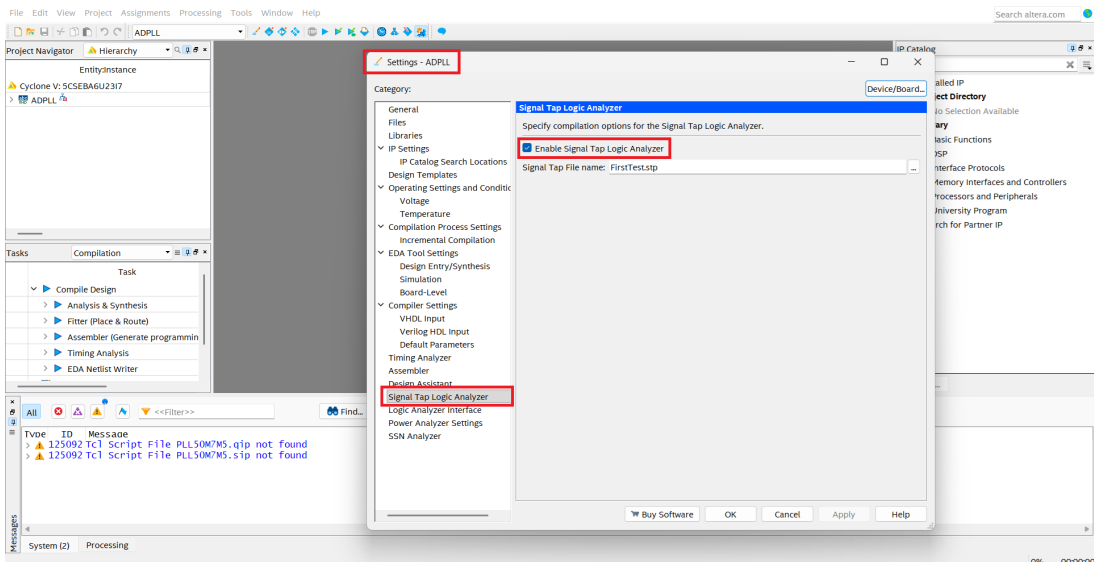
Fonte: O próprio autor

Na aba "Data", é possível configurar o formato de exibição dos dados a serem lidos, conforme mostra a imagem abaixo. Salve as configurações do Signal Tap como um arquivo .stp e feche o menu do Signal Tap.



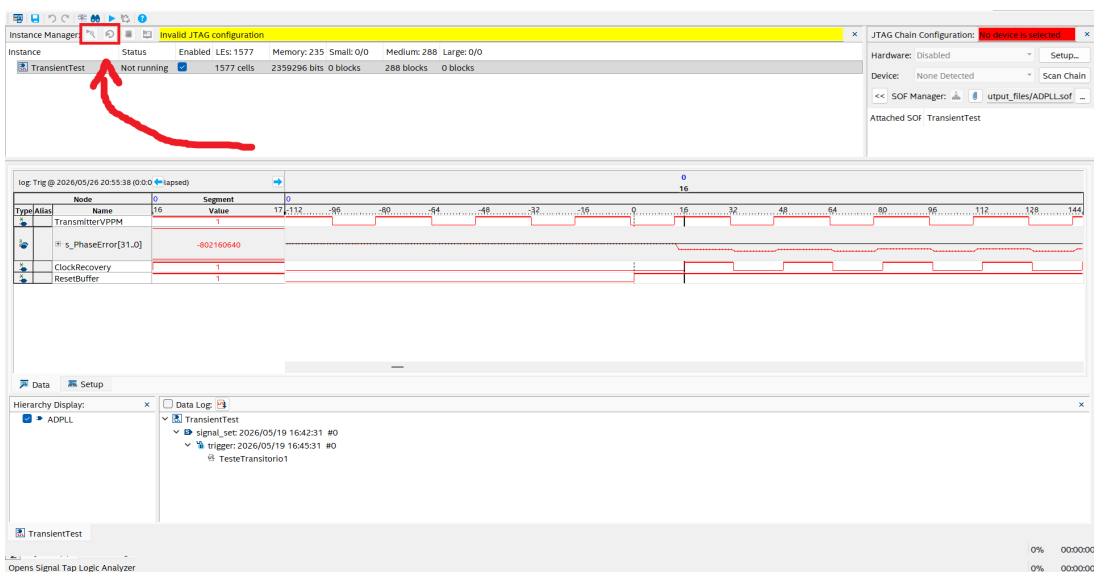
Fonte: O próprio autor

Por fim, confira se o Signal Tap está habilitado em seu projeto, conforme mostra a imagem abaixo. Após terminar de configurar o Signal Tap, faça uma compilação completa do código VHDL novamente. Esse processo incluirá os recursos lógicos necessários do Signal Tap durante a compilação. O seu código e o código complementar do Signal Tap serão incluídos automaticamente no arquivo .sof gerado pela compilação.



Fonte: O próprio autor

Na próxima vez que um arquivo .sof for gravado na FPGA, basta abrir novamente o menu do Signal Tap e selecionar a opção "Run Analysis" para fazer uma única aquisição ou "Autorun Analysis" para fazer aquisições continuamente, conforme mostra a imagem abaixo.



Fonte: O próprio autor

**APÊNDICE E – Código VHDL do arquivo testbench**

```
1
2 -----
3 -- BLOCO DE IMPORTAÇÃO DE BIBLIOTECAS
4 -----
5 library ieee;
6 use ieee.std_logic_1164.all;
7 use ieee.numeric_std.all;
8 use std.textio.all;
9 -----
10
11
12 -----
13 -- BLOCO DE DEFINIÇÃO DA ENTIDADE
14 -----
15 entity ADPLL_tb is
16 end ADPLL_tb;
17 -----
18
19
20 -----
21 -- BLOCO DE DEFINIÇÃO DA ARQUITETURA
22 -----
23 architecture a_ADPLL_tb of ADPLL_tb is
24 -----
25
26
27 -----
28 -- BLOCO DE DEFINIÇÃO DO COMPONENTE ADPLL
29 -----
30 component ADPLL
31 port (
32     TransmitterVPPM    : in  std_logic;
33     Receiver50M        : in  std_logic;
34     Reset              : in  std_logic;
35     ClockRecoveryLock  : out std_logic;
36     Data               : out std_logic;
37     DataN              : out std_logic
38 );
39 end component;
40 -----
41
42
43
44
45
46
47
48
49
50
```

```

51
52 -----
53 -- BLOCO DE DEFINIÇÃO DOS SINAIS
54 -----
55 signal s_TransmitterVPPM,
56         s_Receiver50M,
57         s_Reset,
58         s_Clock3M75,
59         s_PhasedClock3M75,
60         s_ClockRecovery,
61         s_Data,
62         s_DataN          : std_logic          := '0';
63 signal s_StrVPPM       : signed(767 downto 0) := "100110101010100110011010101001101001101
64                                                     0101001011001101010011101010011010100110
65                                                     011001101010010110100110101001010110011
66                                                     010011010101001101001101001100110100110
67                                                     011010011010011001011001101001011010100
68                                                     110100101100110011010010101101001101001
69                                                     010101100110011010101010011001101010011
70                                                     001100110100110100110011010010110011001
71                                                     1001101010011001100110011001100110011001011
72                                                     010011001100101011001100101101010100110
73                                                     010110100110011001011001101001011010101
74                                                     001100101101010011010010110101001011001
75                                                     011010011010100101101001100110010110100
76                                                     101101001011010010101100101100110101010
77                                                     010110011010011001011001100110100101100
78                                                     110010110010110010110101001011001011001
79                                                     100101100101011010010110010101011001010
80                                                     110101010100101011010100110010101101001
81                                                     101001010110100101100101011001101010010
82                                                     101100110011001010110010110";
83 -----
84
85
86 -----
87 -- BLOCO DE INÍCIO DA ARQUITETURA
88 -----
89 begin
90 -----
91
92
93
94
95
96
97
98
99
100

```

```

101 -----
102 -- BLOCO DE CRIAÇÃO E INTERLIGAÇÃO DO ADPLL
103 -----
104 TopLevel      :      ADPLL
105   port map(
106     TransmitterVPPM  => s_TransmitterVPPM,
107     Receiver50M      => s_Receiver50M,
108     Reset            => s_Reset,
109     ClockRecoveryLock => s_ClockRecoveryLock,
110     Data             => s_Data,
111     DataN            => s_DataN
112   );
113 -----
114
115
116 -----
117 -- BLOCO DE EMULAÇÃO DO SINAL DE CLOCK DE 3.75 MHZ DO TRANSMISSOR
118 -----
119 process
120 begin
121   wait for 200000 ps;
122   while true loop
123     s_Clock3M75 <= not(s_Clock3M75);
124     wait for 133333 ps;
125   end loop;
126 end process;
127 -----
128
129
130 -----
131 -- BLOCO DE EMULAÇÃO DO SINAL DE CLOCK DE 3.75 MHZ DO TRANSMISSOR DEFASADO EM 90°
132 -----
133 process
134 begin
135   wait for 200000 ps;
136   wait for 66667 ps;
137   while true loop
138     s_PhasedClock3M75 <= not(s_PhasedClock3M75);
139     wait for 133333 ps;
140   end loop;
141 end process;
142 -----
143
144
145
146
147
148
149
150

```



```
201
202 -----
203 -- BLOCO DE EMULAÇÃO DO SINAL DE CLOCK DE 50 MHZ DO RECEPTOR
204 -----
205 process
206 begin
207     wait for 200000 ps;
208 -- Fase de 45° graus
209     wait for 33334 ps;
210     s_Receiver50M <= not(s_Receiver50M);
211     while true loop
212         wait for 10 ns;
213         s_Receiver50M <= not(s_Receiver50M);
214     end loop;
215 end process;
216 -----
217
218
219
220 -----
221 -- BLOCO DE EMULAÇÃO DO SINAL DE RESET DO RECEPTOR
222 -----
223 process
224 begin
225     wait for 200000 ps;
226 -- Fase de 45° graus
227     wait for 33334 ps;
228     s_Reset <= not(s_Reset);
229     wait for 4000000 ps;
230     s_Reset <= not(s_Reset);
231     wait for 200000 ps;
232     s_Reset <= not(s_Reset);
233     wait;
234 end process;
235 -----
236
237
238 -----
239 end architecture;
240 -----
```