

UNIVERSIDADE FEDERAL DE MINAS GERAIS
Escola de Engenharia
Programa de Pós-Graduação em Engenharia Elétrica

Vilmondes Ribeiro Silva

**CONVERSOR DIGITAL PARA TEMPO DO TIPO VERNIER IMPLEMENTADO EM
FPGA**

Belo Horizonte
2025

Vilmondes Ribeiro Silva

**CONVERSOR DIGITAL PARA TEMPO DO TIPO VERNIER IMPLEMENTADO EM
FPGA**

Dissertação submetida à banca examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Dalton Martini Colombo

Coorientador: Prof. Dr. Tomás Perpétuo Corrêa

Belo Horizonte

2025

S586c

Silva, Vilmondes Ribeiro.

Conversor Digital para Tempo do tipo Vernier implementado em FPGA [recurso eletrônico] / Vilmondes Ribeiro Silva. - 2025.

1 recurso online (82 f. : il., color.) : pdf.

Orientador: Dalton Martini Colombo.

Coorientador: Tomás Perpétuo Corrêa.

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Apêndices: f. 69-82.

Bibliografia: f. 65-68.

1. Engenharia elétrica - Teses. 2. Processamento de sinais - Teses. 3. Conversores digitais - Teses. I. Colombo, Dalton Martini. II. Corrêa, Tomás Perpétuo. III. Universidade Federal de Minas Gerais. Escola de Engenharia. IV. Título.

CDU: 621.3(043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
COLEGIADO DO CURSO DE GRADUAÇÃO / PÓS-GRADUAÇÃO EM ENGENHARIA
ELÉTRICA

FOLHA DE APROVAÇÃO

"Conversor Digital Para Tempo do Tipo Vernier Implementado em FPGA"

Vilmondes Ribeiro Silva

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica.

Aprovada em 15 de janeiro de 2025.

Por:

Prof. Dr. Dalton Martini Colombo
(UFMG) - Orientador

Prof. Dr. Tomás Perpetuo Corrêa
DELT (UFMG)

Profa. Dra. Luciana Pedrosa Salles
DEE (UFMG)

Prof. Dr. Hilton de Oliveira Mota
DEE (UFMG)



Documento assinado eletronicamente por **Dalton Martini Colombo, Professor do Magistério Superior**, em 20/01/2025, às 10:18, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

Documento assinado eletronicamente por **Hilton de Oliveira Mota, Professor do Magistério**



Superior, em 20/01/2025, às 13:28, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Luciana Pedrosa Salles, Professora do Magistério Superior**, em 20/01/2025, às 15:55, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Tomás Perpetuo Correa, Membro de comissão**, em 21/01/2025, às 09:24, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site

[https://sei.ufmg.br/sei/controlador_externo.php?](https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0)

[acao=documento_conferir&id_orgao_acesso_externo=0](https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador

3875866 e o código CRC **8A195F0A**.

Dedico este trabalho à minha esposa e nossos filhos, que ao meu lado me renovam as forças e me dão uma nova motivação para continuar a minha jornada nesse mundo.

AGRADECIMENTOS

Agradeço aos meus Pais pela criação e o apoio que me deram.

Agradeço a minha esposa e meus filhos pelo mais puro e incondicional amor, carinho e motivação que me oferecem.

Agradeço também a toda a comunidade da UFMG pela ótima recepção e por todo o conhecimento que me proporcionaram.

Meus sinceros agradecimentos ao orientador, Prof. Dr. Dalton Martini Colombo, e ao coorientador, Prof. Dr. Tomás Perpétuo Corrêa, por toda a orientação, dedicação e paciência desde o início da minha jornada no programa de pós-graduação. Agradeço imensamente o apoio constante, especialmente durante os processos de desenvolvimento e revisão dos trabalhos.

Expresso meus agradecimentos especiais às instituições que contribuíram para o financiamento deste estudo, especialmente à Fundação de Amparo à Pesquisa de Minas Gerais (FAPEMIG, processos APQ-01987-21 e APQ-05837-23), ao Instituto Serrapilheira (Número Serra – processo 2211-42117) e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código Financeiro 001. Agradeço também ao Programa APCI da Sociedade Brasileira de Microeletrônica (SBMicro) pelo apoio prestado.

RESUMO

Neste trabalho, é apresentado um Conversor Digital para Tempo (DTC) baseado em uma plataforma FPGA de baixo custo, utilizando a arquitetura Vernier com osciladores. Um DTC é um circuito que converte informações digitais em um sinal de tempo altamente preciso. O DTC Vernier proposto foi implementado em um chip FPGA Altera Cyclone III e possui resolução ajustável por meio das relações entre os períodos de dois sinais periódicos. A linearidade do sistema foi medida para uma resolução de 990 ps, com valores de Não Linearidade Diferencial variando entre -0,41 e +0,54 LSB, e de Não Linearidade Integral entre -0,82 e +0,17 LSB. O sistema possui um Faixa de tempo total de 99 ns e um consumo de energia estimado em 89 mW. Além disso, as medições demonstraram que o sistema atinge uma resolução máxima estimada em 12,5 ps, utilizando apenas 2% dos recursos do FPGA. Adicionalmente, a síntese lógica e física do circuito proposto foi realizada utilizando uma tecnologia CMOS comercial de 350 nm, com um consumo de potência estimado em 1,75 mW e uma área de silício de 237 μm x 190 μm .

Palavras-chave: conversor digital para o tempo; DTC baseado em plataforma FPGA; princípio Vernier para DTC; circuitos em modo de tempo.

ABSTRACT

This work presents a Digital-to-Time Converter (DTC) based on a low-cost FPGA platform, employing the Vernier with oscillators architecture. A DTC is a circuit that converts digital information into a very accurate time output. This work is implemented on an Altera Cyclone III FPGA chip and features tunable resolution through the period ratio between two PLL-generated signals. The linearity of the system was measured for a resolution of 990 ps, showing DNL and INL values of -0.41 to +0.54 LSB and -0.82 to +0.17 LSB, respectively. The proposed Vernier DTC has a range of 99 ns and an estimated power consumption of 89 mW. Furthermore, measurements demonstrate that the system achieves a maximum resolution of 12.5 ps, utilizing only 2% of the FPGA resources. Additionally, the logical and physical synthesis of the proposed design was carried out using a commercial 350 nm CMOS technology, and the estimated power consumption and silicon area are 1.75 mW, and 237 μm x 190 μm , respectively.

Keywords: digital to time converter; FPGA based DTC; Vernier principle; time mode circuits.

LISTA DE FIGURAS

FIGURA 1: MARGEM DE TENSÃO E ATRASO INTRÍNSECO PARA NÓS TECNOLÓGICOS CMOS (ADAPTADO DE [8]).	14
FIGURA 2: DIAGRAMA DE BLOCOS DE UM SISTEMA DE PROCESSAMENTO DE SINAIS EM MODO DE TEMPO (ADAPTADO DE [11]).	15
FIGURA 3: EXEMPLO DA FUNÇÃO DE TRANSFERÊNCIA DE UM DTC. (ADAPTADO DE [19]).	20
FIGURA 4: ARQUITETURA CONSTANT SLOPE USADA NO TRABALHO [24] (ADAPTADO PELO AUTOR).	23
FIGURA 5: TOPOLOGIA DO DTC PROPOSTO PELO TRABALHO [26]. (ADAPTADO PELO AUTOR).	24
FIGURA 6: DIAGRAMA DE BLOCOS DE UMA ARQUITETURA DE DTC COM LINHAS DE ATRASO VERNIER. (ADAPTADO DE [30]).	27
FIGURA 7: DIAGRAMA DE TEMPO DO FUNCIONAMENTO DE UM DTC GENÉRICO NA ARQUITETURA VERNIER. (ADAPTADO DE [31]).	28
FIGURA 8: DIAGRAMA DE TEMPO EXEMPLIFICANDO O PRINCÍPIO VERNIER COM OSCILADORES. (ADAPTADO DE [16]).	31
FIGURA 9: DIAGRAMA DE TEMPO REPRESENTANDO AS RELAÇÕES ENTRE OS ÍNDICES K E M E A DETECÇÃO DE COINCIDÊNCIA DE FASE. (A) K=6 E M=5, QUE SÃO NÚMEROS PRIMOS ENTRE SI. (B) K=6 E M=4, QUE NÃO SÃO PRIMOS ENTRE SI. (FIGURA DO AUTOR).	32
FIGURA 10: DIAGRAMA DE TEMPO EXEMPLIFICANDO UM SISTEMA ONDE O TERMO (K-M) É DIFERENTE DE 1. (FIGURA DO AUTOR).	33
FIGURA 11: PLATAFORMA DE DESENVOLVIMENTO EM FPGA, TERASIC DEO [34].	35
FIGURA 12: (A) FREQUÊNCIAS DE SAÍDA POSSÍVEIS DE SEREM GERADAS PELO PLL. (B) HISTOGRAMA MOSTRANDO A DISTRIBUIÇÃO DOS SINAIS GERADOS NO PLL POR FAIXA DE FREQUÊNCIA. (FIGURA DO AUTOR).	37
FIGURA 13: DIAGRAMA DA GERAÇÃO DE SINAIS. (FIGURA DO AUTOR).	38
FIGURA 14: DIAGRAMA DE BLOCOS DO DTC VERNIER. (FIGURA DO AUTOR).	40
FIGURA 15: DIAGRAMA DE TEMPO DO DTC VERNIER. (FIGURA DO AUTOR).	40
FIGURA 16: IMAGEM DA TELA DO DIAGRAMA DE TEMPO DA SIMULAÇÃO. (FIGURA DO AUTOR).	41
FIGURA 17: MEDIÇÕES FEITAS NA SIMULAÇÃO, (A) ΔT , (B) T_{\min} . (FIGURA DO AUTOR).	41
FIGURA 18: DIAGRAMA DE BLOCOS DO DETECTOR DE COINCIDÊNCIA DE FASE. (FIGURA DO AUTOR).	42
FIGURA 19: DIAGRAMA DE TEMPO DO DETECTOR DE COINCIDÊNCIA DE FASE, (A) PULSOS MUITO ESTREITOS, (B) COM PULSOS ALARGADOS NO TEMPO. (FIGURA DO AUTOR).	44
FIGURA 20: DIAGRAMA DE BLOCOS DO DTC IMPLEMENTADO NO TRABALHO [16] (ADAPTADO PELO AUTOR)	45
FIGURA 21: DIAGRAMA DE BLOCOS DO DTC IMPLEMENTADO NO TRABALHO [31]. (ADAPTADO PELO AUTOR).	45
FIGURA 22: EXEMPLO VISUAL DA DEFINIÇÃO GERAL DE JITTER. (FIGURA DO AUTOR).	48
FIGURA 23: EXEMPLO DA DEFINIÇÃO DE JITTER PERIÓDICO. (ADAPTADO DE [45]).	49
FIGURA 24: ESTRATÉGIA DE MEDIÇÃO DO PERÍODO USANDO UM SCRIPT DO MATLAB. (FIGURA ELABORADA PELO AUTOR).	49
FIGURA 25: HISTOGRAMA REPRESENTANDO AS MEDIÇÕES DOS PERÍODOS PARA SINAIS SINTETIZADOS NA FAIXA DE 10 MHz PELO PLL (FIGURA ELABORADA PELO AUTOR).	50
FIGURA 26: DISTRIBUIÇÃO NORMAL FEITA COM BASE NA DIFERENÇA ENTRE OS PERÍODOS MEDIDOS E O PERÍODO IDEAL, ONDE A MÉDIA DESSA DISTRIBUIÇÃO É 13,54 ps (FIGURA ELABORADA PELO AUTOR).	51
FIGURA 27: MEDIÇÃO DO PERÍODO T_{\min} ENTRE AS DETECÇÕES DE COINCIDÊNCIA DE FASE. (FIGURA ELABORADA PELO AUTOR).	52
FIGURA 28: (A) NÃO LINEARIDADE DIFERENCIAL (DNL), (B) NÃO LINEARIDADE INTEGRAL (INL).	52
FIGURA 29: COMPARAÇÃO ENTRE A FUNÇÃO DE TRANSFERÊNCIA IDEAL (AZUL) E REAL (LARANJA). (FIGURA ELABORADA PELO AUTOR).	53
FIGURA 30: COMPARAÇÃO ENTRE O LSB ESTIMADO PELA EQUAÇÃO (23) E O LSB IDEAL (990,1 ps). (FIGURA ELABORADA PELO AUTOR).	54
FIGURA 31: (A) HISTOGRAMA PARA O COMANDO DECIMAL DE ENTRADA "100" EM UM DTC CONFIGURADO PARA $\Delta T = 990,1$ ps, (B) CAPTURA DE TELA DA MEDIÇÃO DOS SINAIS DE SAÍDA START (AZUL) E STOP (MAGENTA). (FIGURAS ELABORADAS PELO AUTOR).	55
FIGURA 32: ANÁLISE DA ESTIMAÇÃO DE ΔT , (A) HISTOGRAMA REFERENTE A PALAVRA DIGITAL DE ENTRADA 8000(DECIMAL), (B) HISTOGRAMA REFERENTE A PALAVRA DIGITAL DE ENTRADA 7000(DECIMAL) (FIGURA ELABORADA PELO AUTOR).	56
FIGURA 33: (A) LAYOUT COM TODAS AS CAMADAS A VISTA, (B) LAYOUT SEM AS CAMADAS DE METAL PARA MELHOR VISUALIZAÇÃO DOS ELEMENTOS LÓGICOS. (FIGURA ELABORADA PELO AUTOR).	60

LISTA DE TABELAS

<i>TABELA 1: COMPARAÇÃO DE PERFORMANCE DOS TRABALHOS [25] E [18].</i>	25
<i>TABELA 2: PARÂMETROS USADOS NA SIMULAÇÃO APRESENTADA NA FIGURA 16.</i>	40
<i>TABELA 3: ESPECIFICAÇÕES DOS PARÂMETROS USADOS NAS DUAS CONFIGURAÇÕES DO DTC VERNIER.</i>	47
<i>TABELA 4: TOTAL DE CONSUMO DE ELEMENTOS NO FPGA</i>	57
<i>TABELA 5: DESEMPENHO DO DTC PROPOSTO, COMPARADO COM OUTROS TRABALHOS.</i>	58

LISTA DE ABREVIATURAS E SIGLAS

ATE	<i>Automatic Test Equipment</i> (Equipamento de Teste Automatizado).
ASIC	<i>Application Specific Integrated Circuit</i> (Circuito Integrado de Aplicação Específica).
CML	<i>Current Mode Logic</i> (Lógica em Modo de Corrente elétrica).
CMOS	<i>Complementary Metal-Oxide-Semiconductor</i> (Semicondutor de Óxido Metálico Complementar).
FPGA	<i>Field Programmable Gate Array</i> (Matriz de portas Programáveis).
DFF	<i>flip-flop</i> tipo D.
DNL	<i>Differential Non Linearity</i> (Não Linearidade Diferencial).
DR	<i>Dynamic Range</i> (Faixa de tempo Dinâmica).
DTC	<i>Digital to Time Converter</i> (Conversor de modo Digital para o Tempo).
HDL	<i>Hardware Description Language</i> (Linguagem de Descrição de Hardware).
INL	<i>Integral Non Linearity</i> (Não linearidade Integral).
LSB	<i>Least Significant bit</i> (bit menos significativo).
LUT	<i>Lookup Table</i> (Tabela de consulta).
MSB	<i>Most Significant bit</i> (bit mais significativo).
ns	nano-segundos.
PLL	<i>Phase Locked Loop</i> (Laço Fechado por Fase).
ps	Picossegundos.
PVT	Process-voltage-temperature variations (variações por processo, tensão e temperatura).
PWM	<i>Pulse Width Modulation</i> (Modulação por Largura de Pulso).
RTL	<i>Register Transfer Level</i> (Transferência a Nível de Registradores).
TDC	<i>Time to Digital Converter</i> (Conversor de modo Tempo para o Digital).
TMSP	<i>Time Mode Signal Processing</i> (Processamento de Sinais em Modo de Tempo).
TVC	<i>Time to Voltage Converter</i> (Conversor de modo de Tempo para Tensão).
UFMG	Universidade Federal de Minas Gerais.
UWB	<i>Ultra Wide Band</i> (Banda Ultra Larga).
VDD	Tensão de alimentação do dispositivo.
VHDL	<i>VHSIC Hardware Description Language</i> (Linguagem de descrição de Hardware VHSIC).
VHSIC	<i>Very High Speed Integrated Circuit</i> (Circuito Integrado de Muito Alta Velocidade).
VTC	<i>Voltage to Time Converter</i> (Conversor de modo de Tensão para Tempo).
Vth	<i>Voltage Threshold</i> (Tensão de limiar).
μs	Microsegundos.

SUMÁRIO

1. INTRODUÇÃO	13
1.1. MOTIVAÇÃO E JUSTIFICATIVA	15
1.2. OBJETIVO GERAL	16
1.3. OBJETIVOS ESPECÍFICOS.....	17
1.4. ESTRUTURA DA DISSERTAÇÃO.....	17
2. REVISÃO BIBLIOGRÁFICA	19
2.1. MÉTRICAS DE PERFORMANCE	19
2.1.1. Resolução	19
2.1.2. Não Linearidades	20
2.1.3. Faixa de tempo total	21
2.1.4. Consumo energético e potência dissipada.....	22
2.2. ARQUITETURAS DE DTCs.....	22
2.2.1. Arquiteturas de DTCs Analógicos	22
2.2.2. Projetos de DTCs Digitais.....	23
2.2.3. Projetos de DTCs Baseados em FPGA.....	25
2.2.4. Arquitetura Vernier com osciladores.....	27
3. DESENVOLVIMENTO DO PROJETO	30
3.1. O PRINCÍPIO VERNIER	30
3.2. IMPLEMENTAÇÃO DO DTC NO FPGA	35
3.2.1. Características da Plataforma de desenvolvimento	35
3.2.2. Estratégia de Geração de Sinais.....	36
3.2.3. Arquitetura Proposta	39
3.2.4. Discussão sobre trabalhos anteriores	44
4. ANÁLISE DOS RESULTADOS	47
4.1. AVALIAÇÃO DO JITTER NOS SINAIS SINTETIZADOS NOS PLLS	47
4.2. IMPLEMENTAÇÃO COM 990,1 PS DE RESOLUÇÃO.....	51
4.2.1. Período de coincidência de fase	51
4.2.2. Linearidade	52
4.2.3. Resolução	53
4.3. IMPLEMENTAÇÃO COM 12,5 PS DE RESOLUÇÃO.....	55
4.4. DISCUSSÕES.....	57
4.5. SÍNTESE LÓGICA E FÍSICA USANDO TECNOLOGIA CMOS 350 NM	59
5. CONCLUSÕES E PROPOSTAS FUTURAS	62
5.1. PROPOSTAS FUTURAS	64
REFERÊNCIAS BIBLIOGRÁFICAS	65
APÊNDICE A – DTC VERNIER TOP-LEVEL	69
APÊNDICE B – DETECTOR DE BORDA DE SUBIDA ASSÍNCRONO	72
APÊNDICE C – DETECTOR DE COINCIDÊNCIA DE FASE	74
APÊNDICE D – CONTADOR VERNIER	76
APÊNDICE E – CÓDIGO MATLAB PARA A LEITURA DOS ARQUIVOS CSV DAS MEDIÇÕES FEITAS NO OSCILOSCÓPIO	78
APÊNDICE F – DIAGRAMA E REPORTE DE COMPILAÇÃO DA IMPLEMENTAÇÃO NO FPGA	82

1. INTRODUÇÃO

Um conversor do modo digital para o modo de tempo, também conhecido como *Digital to Time Converter* (DTC), é um dos blocos fundamentais usados no processamento de sinais em modo de tempo. O DTC basicamente tem a função de converter um dado digital em uma variável de tempo, onde essa variável pode ser representada tanto pela largura de um pulso quanto pela diferença de tempo entre os eventos de dois sinais específicos [1].

O conceito de circuitos em modo de tempo surge como uma abordagem alternativa para processar sinais tanto analógicos quanto digitais. É de conhecimento comum que sistemas eletrônicos normalmente são mais populares operando em modo de tensão ou corrente elétrica.

Circuitos em modo de tensão são os mais comumente encontrados, onde o nível de tensão é utilizado como forma de representação das entradas e saídas do sistema. Essa abordagem possui um longo histórico na eletrônica, destacando-se em projetos baseados em amplificadores operacionais e grande parte da eletrônica digital clássica onde o modo de tensão sinaliza os dois estados possíveis destes sistemas (nível alto e baixo) [2].

Circuitos eletrônicos em modo de corrente também são facilmente encontrados, dado o considerável número de dispositivos eletrônicos e sensores que operam com o princípio translinear, onde as relações exponenciais de corrente e tensão são usadas para manipular os sinais de corrente elétricas usados no sistema [3]. Outro exemplo é a lógica digital CML (Current Mode Logic) onde a corrente gerada por uma fonte é chaveada entre dois possíveis ramos do circuito para se alcançar alta taxa de velocidade. [4].

Circuitos em modo de tempo por sua vez também não são recentes. Técnicas que manipulam sinais elétricos no tempo como a modulação por largura de pulso (*Pulse Width Modulation* - PWM), moduladores Delta-Sigma e amplificadores classe D, são exemplos de circuitos clássicos que operam em modo de tempo e tem diversas aplicações tanto em sistemas de controle quanto comunicação [5].

Porém, recentemente é observado um aumento no interesse acadêmico em circuitos em modo de tempo, sobretudo para o uso em novas topologias de circuitos integrados. Uma das motivações para tal interesse é o *Scaling* da tecnologia de circuitos integrados *Complementary Metal-Oxide-Semiconductor* (CMOS). O *Scaling* se refere à tendência da diminuição das dimensões dos transistores CMOS com o objetivo de se melhorar o desempenho e eficiência energética de tais circuitos, gerando sistemas progressivamente mais velozes, com uma taxa de integração cada vez maior e dimensões reduzidas [6].

No entanto, um dos desafios encontrados no *Scaling* se trata da Margem de tensão, também conhecida como *Voltage headroom*. Esse termo se refere sobre a diferença entre a tensão de alimentação (VDD) e a tensão de limiar (V_{th}) dos transistores CMOS.

No *Scaling* CMOS, há uma inevitável tendência para que o *Voltage headroom* diminua progressivamente. As consequências da diminuição dessa margem de tensão são a perda de resolução em modo de tensão, o que dificulta o projeto pelo fluxo analógico, uma vez que não há uma grande extensão entre o ponto de saturação do transistor e a tensão de alimentação, o que torna as novas tecnologias mais apropriadas para o fluxo digital [7]. A Figura 1 mostra um gráfico comparativo entre o atraso de propagação intrínseco e a margem de tensão, os dados são estimados por [8] para diversas tecnologias CMOS (desde 130 nm até 14 nm).

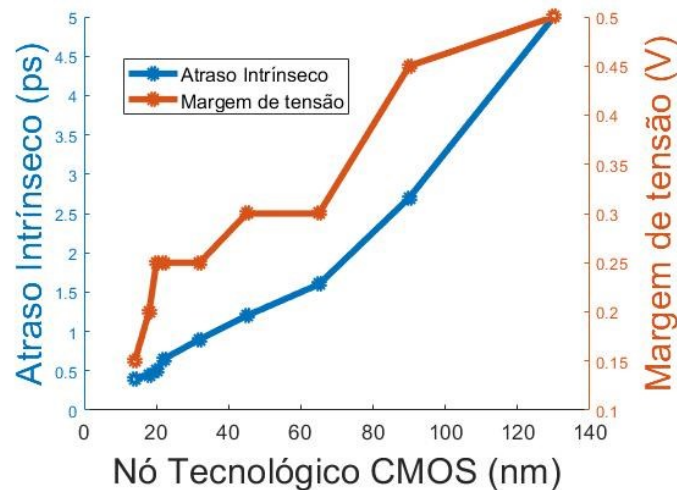


Figura 1: Margem de tensão e atraso intrínseco para nós tecnológicos CMOS (Adaptado de [8]).

Se o *Scaling* da tecnologia CMOS degrada a margem de tensão, por outro lado traz um aumento da resolução em modo de tempo, justamente pela diminuição do atraso de propagação intrínseco, como pode ser visto na Figura 1, o que permite que um circuito em modo de tempo consiga comprimir mais informação em um determinado intervalo. O modo de tempo também tem a vantagem de permitir o processamento de sinais analógicos em um ambiente digital, graças à dualidade que as variáveis de tempo possuem. O sinal em modo de tempo pode ser considerado analógico uma vez que é representado pela largura no tempo de um pulso, mas também é digital uma vez que esse pulso assume apenas dois valores distintos de amplitude [9].

O novo paradigma de processamento de sinais em modo de tempo, conhecido como *Time Mode Signal Processing* (TMSP), é ilustrado na Figura 2. Nessa figura, apresenta-se um exemplo de diagrama de blocos de um sistema baseado nesse conceito.

Em um sistema TMSP o DTC serve como um dos circuitos de entrada, onde a entrada digital é convertida para o modo de tempo, processada também em modo de tempo, e então

reconvertida (quando necessário) para o modo digital pelo bloco *Time to Digital Converter* (TDC). O processamento de dados em modo de tempo é realizado por circuitos que operam de maneira digital e processam pulsos de tempo. Mais informações sobre o processamento em modo de tempo podem ser encontradas em [10].

Na Figura 2 os dados tanto analógicos quanto digitais são convertidos para o modo de tempo por seus respectivos conversores representados pelos blocos *Voltage to Time converter* (VTC) para informação analógica em modo de tensão ou *Digital to Time Converter* (DTC) para informação digital, sendo então direcionados para o bloco TMSP, onde são processados em modo de tempo, e finalmente a saída processada é reconvertida pelos conversores de saída *Time to Voltage converter* (TVC) para a saída analógica, ou *Time to Digital Converter* para a saída Digital [10].

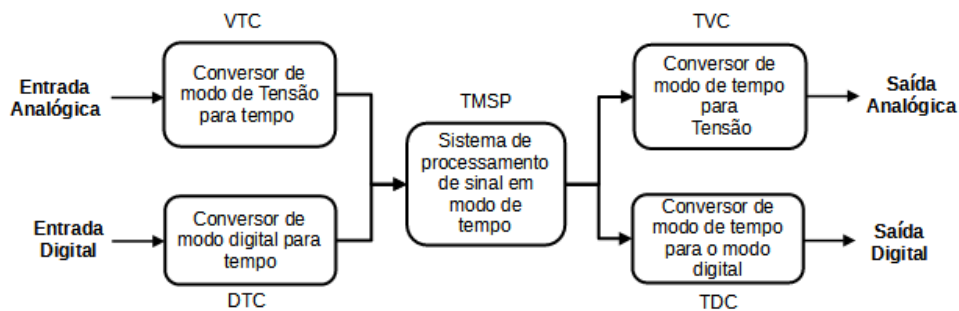


Figura 2: Diagrama de blocos de um sistema de processamento de sinais em modo de tempo (Adaptado de [11]).

Além do conceito descrito na Figura 2, o DTC também é amplamente utilizado em diversas aplicações eletrônicas, especialmente em projetos que demandam sinais de alta precisão temporal. Um exemplo é sua aplicação em equipamentos de teste automatizado, conhecidos como *Automated Test Equipment* (ATE), onde são responsáveis por gerar sinais extremamente precisos para atender a requisitos especializados, sobretudo no teste de circuitos integrados [12]. O DTC também é empregado como gerador de atraso, usado em uma variedade de aplicações incluindo, arquiteturas digitais de *Phase-Locked Loop* (PLL) [13], arquiteturas de radares *Ultra Wide Band* (UWB) com amostragem direta por deslocamento de tempo [14], ou ainda em geradores de sinais e equipamentos de medição como osciloscópios [15].

1.1. MOTIVAÇÃO E JUSTIFICATIVA

A principal motivação deste trabalho surge da crescente demanda por circuitos de alta precisão e eficiência energética em aplicações eletrônicas, incluindo sistemas que operam no

domínio do tempo. Nesse contexto, O DTC desempenha um papel fundamental, pois é utilizado como bloco de entrada de sistemas de processamento em modo de tempo, e tem se mostrado útil em equipamentos de teste, e medição de alta precisão, topologias de PLL digital e de dispositivos de Radar. A investigação de arquiteturas e implementações otimizadas de DTCs é essencial para o avanço dessas aplicações.

Na literatura é observado que os DTCs de alta resolução são preferivelmente projetados em circuitos integrados de aplicação específica também conhecidos por *Application Specific Integrated Circuits* (ASIC). Porém tais dispositivos têm um tempo de desenvolvimento alto, além do elevado custo de fabricação.

A literatura apresenta trabalhos promissores de DTCs implementados em FPGA onde a diferença de desempenho para os dispositivos ASIC está diminuindo tanto pela modernização dos chips FPGA quanto pelas arquiteturas novas usadas para se implementar DTCs nessa plataforma de *hardware* reprogramável [15].

Uma das arquiteturas mais promissoras para se implementar DTCs em FPGA é a que se utiliza do princípio Vernier, que será explicado no Capítulo 3 desse documento. Essa arquitetura se aproveita de recursos facilmente encontrados nos chips FPGA para implementar DTCs em um circuito digital que alia a simplicidade, tamanho reduzido do circuito com alta resolução, boa linearidade na conversão e uma grande faixa de tempo total [16].

Há, porém, uma lacuna na literatura em relação à implementação de DTCs em dispositivos FPGA de baixo custo. Este trabalho explorará o desenvolvimento e avaliação dos resultados de um DTC implementado em uma plataforma FPGA de baixo custo, avaliando a viabilidade e praticidade desta implementação no contexto de seu uso como bloco de entrada para sistemas TMSP.

1.2. Objetivo Geral

O objetivo geral deste trabalho é o desenvolvimento de um DTC baseado em uma plataforma FPGA de baixo custo e utilizando a arquitetura Vernier com osciladores. Tal sistema deverá ter um padrão de linearidade com desvios menores que o valor de largura no tempo de ± 1 bit menos significativo (LSB), e que apresente maior resolução possível para o dispositivo FPGA usado.

1.3. Objetivos específicos

- Revisão bibliográfica sobre as principais arquiteturas de DTC;
- Discussão do princípio Vernier usando osciladores no contexto de um DTC implementado em FPGA;
- Projeto e simulação da estrutura do DTC usando linguagem HDL (VHDL);
- Síntese do DTC no FPGA, verificação do funcionamento e realização de medições de linearidade, estimação do *Jitter* e da potência do circuito;
- Síntese RTL (*Register Transfer Level*) do DTC na tecnologia CMOS 350 nm e *Layout* do projeto do DTC proposto.

Como um dos objetivos futuros, pretende-se implementar o circuito do DTC Vernier em um circuito integrado. Para isso, foi realizada a síntese lógica e física do circuito, permitindo uma estimativa preliminar da área de silício necessária para sua implementação. O foco principal está em aproveitar o DTC projetado neste trabalho como um módulo em um circuito integrado de aplicação específica (ASIC), visando integrar o DTC em conjunto de aplicações que demandam precisão temporal elevada.

1.4. Estrutura da dissertação

Esta dissertação é dividida em 5 capítulos. Neste primeiro capítulo foi apresentado uma introdução sobre a proposta do projeto de um DTC na arquitetura Vernier usando osciladores, bem como as motivações para essa pesquisa e objetivos específicos a serem alcançados para o trabalho proposto.

O capítulo 2 apresenta uma revisão bibliográfica sobre o estado da arte dos DTCs, revisando trabalhos elaborados tanto em fluxo analógico quanto digital, com ênfase nos trabalhos baseados em FPGA.

No capítulo 3 são apresentadas as etapas do desenvolvimento do projeto, começando por uma explicação sobre o princípio Vernier utilizado, passando pela descrição do sistema em linguagem HDL (VHDL), e terminando com a síntese do sistema na plataforma FPGA.

Já no capítulo 4 é apresentada uma análise dos resultados do funcionamento e desempenho do DTC com medições de linearidade, estimação do *Jitter* para os sinais usados

no princípio Vernier, estimação da potência de consumo do sistema e comparação com os trabalhos equivalentes na literatura.

E finalmente no capítulo 5, são mostradas as conclusões deste projeto a partir da análise dos resultados além de propostas de melhorias e trabalhos futuros.

2. REVISÃO BIBLIOGRÁFICA

DTCs podem ser implementados usando o fluxo analógico ou o fluxo digital de projeto. Em ambos os casos, o máximo de desempenho é obtido quando o projeto é implementado em chips customizados de aplicação específica, também conhecidos como *Application-Specific Integrated Circuit* (ASIC) [15]. No entanto, os avanços na tecnologia CMOS, como a diminuição do atraso de propagação, tem possibilitado a queda da diferença de desempenho entre os dispositivos ASIC e FPGA.

Dessa forma, o desenvolvimento baseado em plataformas FPGA torna-se uma alternativa atrativa, pois estas oferecem a capacidade de reconfiguração em nível de *hardware*, o que aumenta tanto a velocidade quanto a flexibilidade do processo de desenvolvimento. Além disso, muitos dispositivos FPGA incorporam uma ampla gama de recursos integrados, como blocos PLL para geração de sinais e outros tipos de *hard blocks* otimizados para operações lógicas e aritméticas, além de processadores dedicados capazes de rodar *software* em conjunto com a aplicação descrita no *hardware* do FPGA, ampliando ainda mais suas possibilidades de aplicação [17].

2.1. Métricas de performance

Esta subseção tem como objetivo apresentar e esclarecer as principais métricas de desempenho utilizadas em DTCs, estabelecendo os fundamentos necessários para a discussão das arquiteturas predominantes. A compreensão dessas métricas é essencial para a avaliação e caracterização das arquiteturas e dos projetos de DTCs apresentados na literatura, permitindo uma análise bem fundamentada.

2.1.1. Resolução

A resolução de um DTC é definida como o menor intervalo de tempo que o sistema é capaz de gerar com precisão. Em outras palavras, a resolução corresponde ao intervalo de tempo associado ao bit menos significativo da representação digital (*Least Significant Bit* - LSB).

A resolução é um dos parâmetros mais importantes para a caracterização deste tipo de sistema, uma vez que quanto menor for seu valor, maior será a densidade de informação do sistema e maior será sua capacidade de representar variações temporais com alta exatidão.

De forma geral, podemos definir a resolução de um conversor DTC como apresentado em (1), onde Δt é a resolução propriamente dita (menor intervalo possível de ser gerado), T_{max} é o intervalo total que o sistema opera (também conhecido como faixa de operação), e N é o número de bits da palavra digital de entrada do sistema.

$$\Delta t = \frac{T_{max}}{2^N - 1} \quad (1)$$

2.1.2. Não Linearidades

A saída do DTC é um pulso com largura no tempo formada pelos incrementos de Δt proporcionais à entrada digital. De uma forma ideal estes pulsos teriam uma relação linear crescente com a entrada, da forma como é apresentada na Figura 3.

A existência de ruído, ou de variações de processo de fabricação, tensão de alimentação, e temperatura de operação (variações PVT), podem interferir no processo de conversão do DTC, gerando uma percepção não linear dos incrementos de Δt para diferentes pontos do sistema.

A Figura 3 mostra um exemplo das não linearidades partindo do gráfico da função de transferência do DTC, é notado que a função de transferência ideal tem uma característica linear entre a entrada e a saída. Os pequenos desvios observados entre os pontos medidos e a função de transferência ideal são as não linearidades [19].

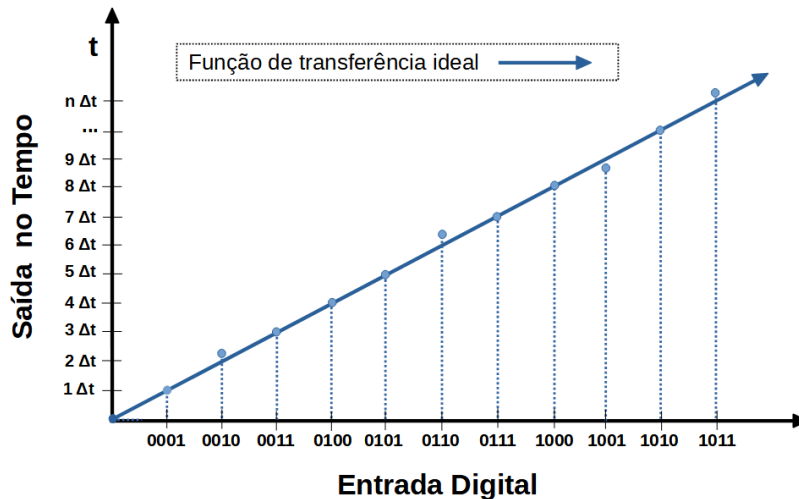


Figura 3: Exemplo da função de transferência de um DTC. (Adaptado de [19]).

Quando esses desvios não ultrapassam o valor do incremento de tempo de $1 \Delta t$, o conversor é considerado monotônico, ou seja, o DTC apresenta uma função de transferência sempre crescente, sem inconsistências que ocasionem reversões em seu comportamento. Isso significa que, à medida que a entrada aumenta, a saída também cresce de forma previsível e consistente [20].

As formas mais comumente usadas para se caracterizar essas não linearidades são através dos erros de não linearidade diferencial (*Differential nonlinearity* – DNL) e da não linearidade integral (*Integral nonlinearity* – INL) [21].

O DNL descreve o desvio da resolução em cada passo quantizado da conversão sempre em relação a um valor ideal (esperado) de resolução, geralmente normalizado para o valor de 1 LSB. Neste trabalho será assumido o DNL na forma da Equação (2), onde T_{out} é a variável de tempo da saída do DTC e “1 LSB” corresponde à resolução Δt do sistema [22].

$$DNL(i) = \left(\frac{T_{out(i+1)} - T_{out(i)}}{LSB} \right) - 1 \quad (2)$$

O INL representa o grau de não linearidade acumulada pelo sistema. Trata-se de uma medida mais rigorosa de não linearidade, pois considera os efeitos das não linearidades acumuladas até o ponto específico de medição. A definição de INL usada neste trabalho pode ser conferida na Equação (3) e, em resumo, é um somatório de todas as não linearidades diferenciais anteriores ao ponto de medição [22].

$$INL(j) = \sum_{i=1}^j DNL(i) \quad (3)$$

Quando ambos DNL e INL são menores que 1 LSB, isso indica que o dispositivo apresenta uma boa fidelidade na conversão, garantindo que cada incremento no código digital corresponde a uma conversão uniforme da entrada digital para a saída em modo de tempo. Um DNL inferior a 1 LSB assegura que não há saltos ou duplicação de níveis diferentes de conversão. Esse padrão é comumente adotado para conversores em aplicações que exigem alta resolução e precisão, como é o caso do DTC proposto nesse trabalho [22].

2.1.3. Faixa de tempo total

A Faixa de tempo total (*Time Range*) refere-se ao intervalo de tempo total que o DTC é capaz de gerar em sua saída. Manipulando a Equação (1) podemos definir uma forma geral para a Faixa de tempo total T_{max} , como pode ser visto em (4).

$$T_{max} = \Delta t(2^N - 1) \quad (4)$$

O tamanho da Faixa de tempo total é uma característica que depende diretamente do tamanho da resolução Δt e da quantidade de conversões possíveis, representada em (4) pelo termo $(2^N - 1)$. Porém a arquitetura escolhida também influencia no tamanho da Faixa de tempo total uma vez que determinadas arquiteturas apresentam diferentes custos para o aumento da Faixa de tempo.

Por exemplo, um DTC que tenha resolução Δt com uma largura menor, precisaria compensar com uma quantidade maior de conversões para possuir uma ampla Faixa de tempo total, no entanto, isso nem sempre é viável dada as limitações apresentadas por cada arquitetura de DTC, como o aumento do tamanho do circuito e suas implicações negativas em outros parâmetros, como o aumento das não linearidades.

Alguns trabalhos apresentam a definição de Faixa dinâmica, também conhecida pelo termo em inglês *Dynamic Range* (DR), sendo esta, uma relação entre o maior e o menor intervalo de tempo possíveis de serem gerados pelo DTC [18]. Geralmente, o valor de DR é expresso em decibéis (dB) e sua definição geral é apresentada em (5).

$$DR = 20 \log_{10} \left(\frac{T_{max}}{\Delta t} \right) \quad (5)$$

2.1.4. Consumo energético e potência dissipada

O consumo energético é um fator crítico tanto em projetos de circuitos integrados, quanto em projetos em FPGAs, influenciando diretamente a eficiência, a durabilidade e a viabilidade econômica de aplicações eletrônicas, e frequentemente serve de parâmetro para quantificar a qualidade de um projeto.

Em circuitos integrados e FPGAs, a dissipação térmica é um desafio significativo, pois o calor excessivo pode afetar a confiabilidade e o desempenho do sistema. Projetos com baixo consumo de potência requerem soluções de dissipação térmica mais simples e baratas, facilitando a integração em sistemas menores e mais compactos [23].

2.2. Arquiteturas de DTCs

2.2.1. Arquiteturas de DTCs Analógicos

Ao revisar trabalhos acadêmicos sobre arquiteturas de DTCs no fluxo de projeto analógico, é evidente a frequente adoção de arquiteturas que aproveitam os fenômenos de carga e descarga em circuitos capacitivos para implementar unidades de atraso (*Delay Line*). Uma das arquiteturas mais comuns para o estado da arte em DTCs no fluxo analógico é a topologia *Constant Slope* [24].

No trabalho [24] o autor usa a arquitetura do tipo *Constant Slope* para criar um DTC com entrada digital de 9-bits e com resolução de 148 fs (femtosssegundos). A Figura 4 apresenta

um exemplo do circuito do DTC proposto pelo referido trabalho. O princípio deste DTC consiste em usar os bits de uma palavra digital de entrada para alimentar um arranjo de capacitores. Esse arranjo é usado para controlar o próximo bloco que é um circuito gerador de rampa e o conjunto desses dois blocos tem um efeito de geração de atraso. Esse gerador de rampa vai inserir um atraso em um sinal periódico que será proporcional à entrada digital deste sistema, deste modo a informação em modo de tempo será a diferença de tempo (atraso) entre o sinal periódico da entrada do gerador de rampa, e do sinal de saída deste.

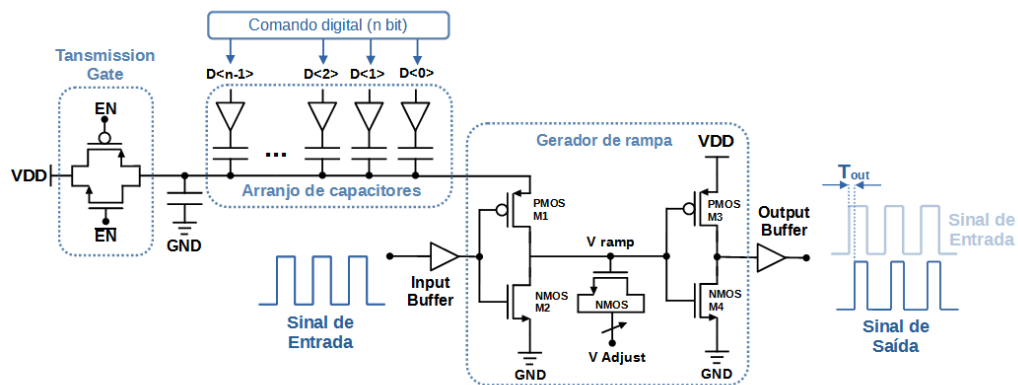


Figura 4: Arquitetura Constant Slope usada no trabalho [24] (Adaptado pelo autor).

Projetos de DTCs em fluxo analógico destacam-se pela personalização desse tipo de circuitos em chips ASIC, oferecendo vantagens significativas em termos de tamanho de chip, consumo energético e desempenho. Essa abordagem é frequentemente associada a DTCs com os mais altos valores de resolução, mas exige maior especialização no projeto, além de custos elevados de produção e maior tempo de fabricação [15]. Ademais, a integração com sistemas predominantemente digitais apresenta desafios adicionais, demandando interfaces bem projetadas para garantir a compatibilidade e o funcionamento adequado do sistema.

2.2.2. Projetos de DTCs Digitais

DTCs implementados usando circuitos digitais costumam empregar contadores ou linhas de atraso programáveis para realizar a conversão de informação digital para o tempo.

O trabalho [26] é um exemplo da utilização de linhas de atraso programáveis para implementar um DTC que serve como um dos componentes de uma aplicação T-DAC (*Time based Digital to Analog Converter*). Um exemplo da arquitetura do DTC implementado em [26] é apresentado na Figura 5.

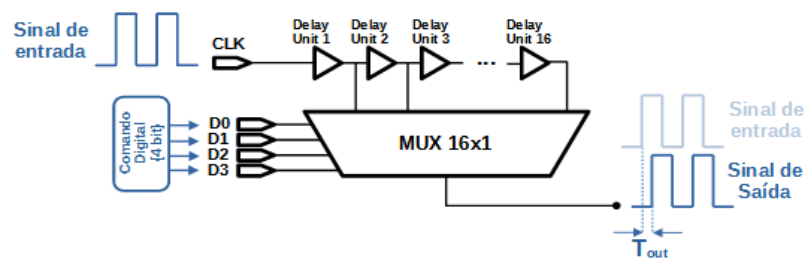


Figura 5: Topologia do DTC proposto pelo trabalho [26]. (Adaptado pelo autor).

A topologia deste DTC consiste em blocos de unidades de atraso (*Delay Units*) ligados às entradas de um multiplexador, onde o comando digital de entrada do DTC é a própria entrada do multiplexador. O funcionamento desse sistema consiste em gerar um atraso em um sinal de entrada que passa pela cadeia de unidades de atraso, que insere o atraso em incrementos discretos proporcionais à entrada digital.

DTCs baseados em linhas de atraso são circuitos simplificados, onde seu princípio reside na distribuição de células de atraso, preferencialmente idênticas para preservar a linearidade do sistema. O valor de atraso depende dos requisitos do projeto e obviamente das capacidades da tecnologia usada. A arquitetura de DTC por linhas de atraso é capaz de oferecer ótimos valores de resolução, considerando principalmente o uso de tecnologias CMOS mais recentes.

No entanto, o tamanho desse circuito tende a aumentar muito caso se queira um DTC com uma grande faixa de tempo total, devido à necessidade do uso de longas linhas de atraso. Essa arquitetura também apresenta problemas para se manter a linearidade da conversão principalmente em sistemas de grande faixa de tempo de saída, devido ao efeito acumulado das não linearidades de cada unidade de atraso [15].

No trabalho [18] são apresentadas alternativas para se controlar os efeitos das não linearidades sobre grandes linhas de atraso. A principal melhoria apresentada em [18] consiste no uso de um sistema de calibração das unidades de atraso e a divisão dessas linhas em dois estágios com o intuito de desacoplar os efeitos acumulativos das não lineares.

Apesar das melhorias, o trabalho [18] ainda apresenta um pico de INL cinco vezes maior que o LSB, evidenciando limitações na mitigação completa das não linearidades.

DTCs implementados pelo fluxo digital tem a vantagem de terem circuitos comparativamente mais simples do que projetos de DTC no fluxo analógico, e por sua vez se beneficiam de tempos de desenvolvimentos mais curtos. Porém projetos de DTC digitais quase sempre ficam atrás dos projetos analógicos no quesito performance, onde os projetos analógicos

geralmente conseguem melhores resultados de resolução, linearidade, consumo de área do chip e de energia.

A *Tabela 1* apresenta a comparação de dois DTCs, um implementado de maneira analógica [25] e outro de maneira digital [18]. A comparação é feita entre os principais dados de performance de ambos. Nota-se que, embora tenham uma alta resolução na faixa dos femtossegundos, o DTC no fluxo analógico do trabalho [25] está à frente do trabalho [18] principalmente no que se refere ao consumo energético, à não linearidade integral (INL), e também na resolução propriamente dita, evidenciando a vantagem dos projetos de DTC analógicos frente aos seus análogos em projetos digitais.

Tabela 1: Comparação de performance dos trabalhos [25] e [18].

Trabalho	Tecnologia CMOS	Resolução (fs)	INL (ps)	Consumo de área (mm ²)	Potência (mW)
[25]	65 nm	189	0,33 (1.7 LSB)	0,1	0,8
[18]	65 nm	330	1,65 (5 LSB)	0,0840	10,3

As topologias digitais baseadas em linhas de atraso são boas opções para a implementação de DTCs, principalmente se forem projetados e fabricados com tecnologias CMOS mais avançadas, que são mais apropriadas para geração de unidades de atraso com reduzidos valores de largura no tempo. Porém essa abordagem sofre para manter a linearidade da conversão em sistemas que tem maior faixa de tempo total, devido ao efeito acumulativo das não linearidades das unidades de atraso.

Ainda assim há o fato de que fabricação de um chip digital ou analógico exige um alto custo financeiro e longo tempo de desenvolvimento, o que pode impactar negativamente na entrega de resultados em aplicações onde o tempo de desenvolvimento é um fator crítico.

2.2.3. Projetos de DTCs Baseados em FPGA

Os dispositivos ASIC possuem vantagens intrínsecas em relação aos dispositivos FPGA, especialmente no que diz respeito ao consumo de área de chip e ao consumo energético, uma vez que os ASICs são circuitos personalizados projetados para atender a aplicações específicas [17]. Contudo, a diferença de desempenho entre ASICs e FPGAs tem diminuído gradativamente a cada nova geração da tecnologia CMOS empregada na fabricação dos FPGAs. Esse avanço também é impulsionado pela integração de novos recursos nos sistemas FPGA, que aumentam sua eficiência e competitividade.

Atualmente, os chips FPGA oferecem uma ampla gama de funcionalidades que ampliam significativamente sua aplicação em sistemas complexos. Entre essas funcionalidades, destacam-se os dispositivos conhecidos como *System on a Chip* (SoC), que integram diversos componentes de um sistema digital em um único chip, frequentemente incluindo um bloco FPGA (SoC FPGA).

SoC FPGAs contam com recursos avançados, como blocos de geração de sinais (PLL ou gerenciadores de *clock*) e blocos com funções predefinidas (*Hard Blocks*). Esses *Hard Blocks* desempenham um papel fundamental na melhoria do desempenho geral dos FPGAs, otimizando o uso da área do chip, reduzindo o tempo de desenvolvimento e simplificando o processo de compilação. Exemplos de *Hard Blocks* incluem blocos de memória, multiplicadores/divisores e núcleos de processadores. Tais funcionalidades ampliam as capacidades e a eficiência dos FPGAs em diversas aplicações [27].

Essa evolução tecnológica tem permitido que os dispositivos FPGA alcancem resultados cada vez mais competitivos em relação aos ASIC. Por exemplo, DTCs implementados em ASIC alcançam resoluções na ordem de femtossegundos, enquanto aqueles implementados em FPGA estão geralmente limitados à faixa de picossegundos. Entretanto, em termos de linearidade e faixa dinâmica, projetos baseados em FPGA têm demonstrado resultados equivalentes aos obtidos com ASICs, evidenciando o potencial crescente dos FPGAs em aplicações de alta precisão.

Em geral, as arquiteturas de DTCs em FPGA estão ligadas a métodos que derivam, ou do uso de linhas de atraso, ou do uso de contadores, ou ainda de uma mescla entre essas duas opções. O trabalho [28] apresenta um projeto de DTC implementado em FPGA inspirado em arquiteturas de células de atraso, mas se aproveitando de um *Hard Block* proprietário IDELAY2, disponível nos chips FPGA da Fabricante XILINX. Tal trabalho oferece um DTC com resolução de 39 ps, e uma faixa de tempo total de 5 ns, utilizando um sinal de *clock* de 200MHz, porém não apresenta informações sobre as não linearidades. Já o trabalho [29] apresenta uma arquitetura muito similar para integrar uma aplicação LIDAR (*Light Detection and Ranging*), com uma resolução de 52 ps.

Uma arquitetura DTC que usa linhas de atraso é proposta em [30] para a implementação de um DTC, também baseado em FPGA. Trata-se da linha de atraso Vernier que em princípio, se utiliza de duas linhas de atraso programáveis para gerar a variável de tempo de saída. O diagrama esquemático da Figura 6 exemplifica a ideia desse princípio.

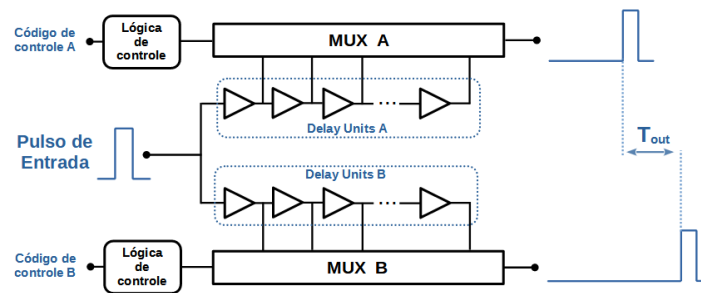


Figura 6: Diagrama de blocos de uma arquitetura de DTC com linhas de atraso Vernier. (Adaptado de [30]).

No DTC do trabalho [30], as linhas de atraso *A* e *B* não são idênticas, sendo que as unidades de atraso *A* possuem um valor ligeiramente diferente das unidades de atraso *B*. Esse DTC opera com sinais de *clock* na faixa de 200 MHz, alcançando uma resolução de 1,02 ps e uma faixa de tempo total de 590 ns, com consumo de potência estimado em 165 mW. Os valores máximos de DNL e INL apresentados no trabalho foram respectivamente -0,2 LSB e 0,62 LSB, o que são excelentes resultados, especialmente considerando uma arquitetura baseada em linhas de atraso.

No entanto, a síntese desse circuito em FPGA resultou em um projeto de grandes dimensões, ocupando aproximadamente 7680 elementos lógicos o que limita a margem de projeto para chips de FPGA de baixo custo, tendo em conta que o DTC frequentemente é usado como parte de aplicações mais complexas.

Para fins de comparação, FPGAs de baixo custo geralmente possuem entre 1 mil e 15 mil elementos lógicos, o que torna a arquitetura em [30] mais adequada para dispositivos FPGA com maior capacidade de recursos, normalmente associados a um custo mais elevado. A elevada utilização de elementos lógicos dificulta a implementação dessa topologia em FPGAs de baixo custo, especialmente quando é necessário compartilhar recursos com outras aplicações no mesmo chip.

2.2.4. Arquitetura Vernier com osciladores

Recentemente outro tipo de arquitetura usando o princípio Vernier vem ganhando destaque [31]. Trata-se do princípio Vernier usando osciladores. Tal arquitetura é capaz de entregar bons valores de resolução e linearidade junto com um baixo consumo energético, usando um circuito relativamente simples e de baixo uso de elementos lógicos.

Observa-se que o princípio Vernier utilizado na implementação de DTCs é inspirado na escala Vernier, amplamente empregada em dispositivos de medição analógicos, como o paquímetro. Esse método recebe o nome em homenagem a Pierre Vernier, matemático francês

do século XVI, que desenvolveu essa escala com base no instrumento de navegação Nônio, criado pelo navegador português Pedro Nunes para medições precisas de ângulos [32].

O princípio Vernier para construção de DTCs se baseia no uso de dois sinais periódicos, que chamaremos de sinais A e B , onde suas frequências são muito próximas. Esses sinais oscilam até que, em determinado momento, suas fases coincidam. Se as frequências forem suficientemente próximas, é possível estabelecer uma regularidade nas diferenças de fase que ocorrem após essa coincidência inicial, criando uma relação cíclica em que novas coincidências de fase acontecem após um número fixo de oscilações.

A partir da coincidência de fase, que representa o momento “zero” do sistema, observa-se que, após cada ciclo de oscilação, a diferença de fase entre os sinais aumenta progressivamente.

Se os incrementos no deslocamento de fase observado entre os sinais A e B tiverem uma relação linear e crescente, é possível projetar um sistema capaz de contar as oscilações desses sinais, onde essa contagem pode ser determinada por uma entrada digital, permitindo a geração de um sinal de saída que contenha uma largura no tempo que seja proporcional à entrada digital.

Na Figura 7, o sinal de saída é formado pela diferença no tempo entre os sinais S_{START} e S_{STOP} , que indicam o final da contagem dos ciclos de oscilação dos sinais A e B . Se os sinais forem estáveis o suficiente, a saída apresentará uma ótima linearidade e proporcionalidade com a entrada digital.

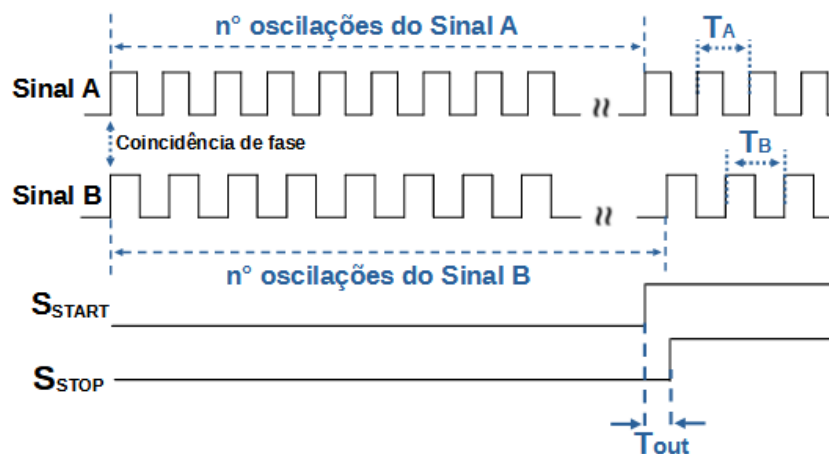


Figura 7: Diagrama de tempo do funcionamento de um DTC genérico na arquitetura Vernier. (Adaptado de [31]).

Ao ajustar as frequências dos sinais periódicos utilizados no princípio Vernier, é possível manipular a diferença de fase gerada entre eles, resultando em deslocamentos temporais suficientemente regulares para serem percebidos de forma crescente e linear.

Neste trabalho será desenvolvido um DTC baseado na arquitetura Vernier usando osciladores. Nossa proposta visa tornar esse sistema acessível para aplicações que não demandam dispositivos de última geração, ou que necessitam de uma solução com baixa ocupação de elementos lógicos no chip FPGA, contudo, mantendo resolução e linearidade comparáveis aos trabalhos de referência no estado da arte dessa arquitetura.

3. DESENVOLVIMENTO DO PROJETO

Neste capítulo, o princípio Vernier será explorado em maior profundidade. Inicialmente, será apresentada uma explicação fundamentada em rigor matemático, complementada por diagramas de tempo ilustrativos que permitem a visualização do funcionamento do sistema. Na sequência, serão discutidos os detalhes relacionados às capacidades do FPGA utilizado e à implementação da arquitetura nesse dispositivo.

3.1. O princípio Vernier

O princípio Vernier usado em nosso trabalho muitas vezes é referido em outros trabalhos acadêmicos usando nomenclaturas ligeiramente diferentes. Para a elaboração de uma apresentação deste princípio usamos como referências principais os artigos [16], [31] e [33]. O artigo [16] menciona o princípio com o termo “resolução por deslocamento de fase quantificado” ou *Quantified Phase Shift Resolution* (QPSR), já o artigo [31] utiliza o próprio termo “Princípio Vernier”, e o artigo [33], que é mais antigo, usa a nomenclatura *Greatest Common Factor Frequency* para explicar de forma detalhada as relações de fase entre dois sinais periódicos.

Para diferenciar a arquitetura Vernier proposta em nosso trabalho, da arquitetura de linhas de atraso Vernier apresentada na subseção 2.2.3, chamaremos esse princípio de “Vernier com osciladores”, onde tal nomenclatura também é empregada no artigo [15].

O princípio básico de funcionamento envolve a utilização de dois sinais periódicos, denominados “*A*” e “*B*” que podem ser vistos na Figura 8. Esses sinais possuem frequências próximas e oscilam continuamente até que suas bordas de subida coincidam. Após esta coincidência de fase, no próximo ciclo de oscilação, ocorre uma pequena diferença de tempo Δt causada pela diferença dos períodos dos dois sinais. Além disso, esta diferença de fase aumenta proporcionalmente com cada ciclo de oscilação subsequente até que ocorra uma nova coincidência de fase. Esta diferença de tempo Δt entre as bordas ascendentes dos sinais periódicos *A* e *B* é usada na topologia do Vernier DTC, representando a informação no modo de tempo.

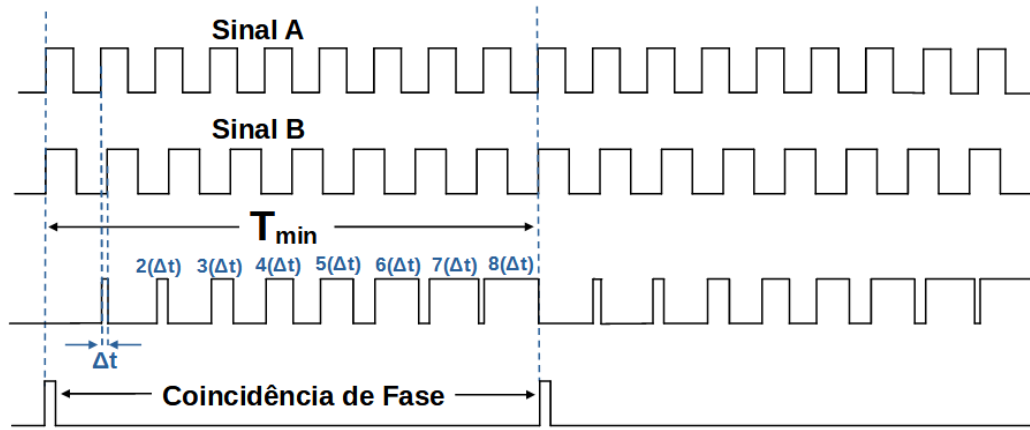


Figura 8: Diagrama de tempo exemplificando o princípio Vernier com osciladores. (Adaptado de [16]).

De forma simplificada, podemos expressar a saída em modo de tempo T_{out} como apresentado na Equação (6).

$$T_{out} = n \Delta t \quad (6)$$

Onde n é o número de ciclos após a coincidência de fase e Δt é a menor largura de tempo possível (esse é o mesmo valor de LSB). Este Δt , conforme visto na Figura 8, é obtido no primeiro ciclo após a coincidência de fase dos sinais e pode ser calculado pela simples diferença entre os períodos dos sinais A e B (T_A e T_B respectivamente), assim como é descrito na Equação (7).

$$\Delta t = T_B - T_A = \frac{1}{f_B} - \frac{1}{f_A} \quad (7)$$

Como pode ser visto na Figura 8, os sinais A e B compartilham um período mínimo em comum (T_{min}), que representa o período entre as coincidências de fases. Da mesma forma, esses sinais também compartilham uma frequência máxima em comum f_{max} que está relacionada a T_{min} , conforme dado pela Equação (8). Onde T_{min} é entendido como o intervalo de tempo em que é possível transmitir cada informação (uma única conversão para cada período T_{min}).

$$f_{max} = \frac{1}{T_{min}} \quad (8)$$

Podemos relacionar a frequência dos sinais A e B com f_{max} e T_{min} usando as Equações de (9) à (12), onde K e M são índices que devem ser inteiros positivos e primos entre si.

$$f_A = K f_{max} \quad (9)$$

$$T_{min} = K T_A \quad (10)$$

$$f_B = M f_{max} \quad (11)$$

$$T_{min} = M T_B \quad (12)$$

Se K e M não são primos entre si, então eles podem ser decompostos em fatores que são primos entre si. Portanto, a coincidência de fases sempre ocorrerá, levando em consideração um número de oscilações K e M cujos valores são primos entre si. Esses índices representam respectivamente o número de oscilações dos sinais A e B dentro do período T_{min} .

Na Figura 9 (a) é apresentado um exemplo da relação entre os índices K e M , onde $K=6$ e $M=5$, como estes valores são números inteiros e primos entre si, a relação da coincidência de fase se mantém da forma como é apresentada.

A Figura 9 (b) apresenta um caso em que $K=6$ e $M=4$, números estes que não são primos entre si. Nesse cenário, a coincidência de fase ocorre de forma equivalente a um caso simplificado com $K=3$ e $M=2$. Isso acontece porque a relação entre os índices K e M pode ser reduzida ao menor denominador comum, o que reflete em um período T_{min} mais curto.

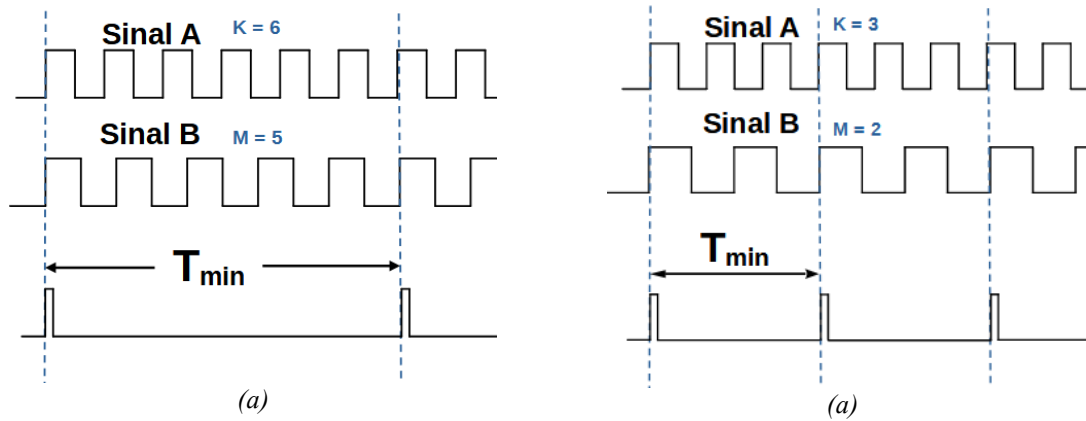


Figura 9: Diagrama de tempo representando as relações entre os índices K e M e a detecção de coincidência de fase. (A) $K=6$ e $M=5$, que são números primos entre si. (B) $K=6$ e $M=4$, que não são primos entre si. (Figura do autor).

Assumindo $K > M$ e considerando as Equações (10) e (12), podemos inferir que o período T_A é menor que o período T_B , ou seja, o sinal A possui uma frequência ligeiramente maior em relação ao sinal B . Nestas condições, dentro do período T_{min} , só é possível ter um número finito de diferenças de fase entre estes sinais que será igual ao número de oscilações do sinal de menor frequência.

Para estabelecer os limites de T_{out} , podemos reescrever a Equação (6) de forma a obter a Equação (13). Esse passo nos permitirá compreender e delimitar os valores extremos que T_{out} pode assumir dentro do período de coincidência entre as fases T_{min} .

$$T_{out} = n\Delta t$$

$$T_{out} = n(T_B - T_A) \quad (13)$$

Manipulando as Equações (10) e (12), e substituindo em (13), obtemos a Equação (14).

$$\begin{aligned}
 T_{out} &= n \left(\frac{T_{min}}{M} - \frac{T_{min}}{K} \right) \\
 T_{out} &= n (T_{min}) \left(\frac{K - M}{KM} \right) \\
 T_{out} &= n (K - M) \left(\frac{T_{min}}{KM} \right) \tag{14}
 \end{aligned}$$

Observe que as Equações (14) e (6) são iguais, mas escritas em formatos diferentes, onde n representa o número de oscilações que se passaram após a coincidência entre as fases dos sinais A e B , e o termo $(K-M)(T_{min}/KM)$ é igual a Δt . Para garantir o funcionamento ideal do princípio de Vernier, isto é, garantindo que Δt tenha uma largura no tempo mais estreita possível, a expressão $(K-M)$ deve ser igual a 1.

Quando essa condição é satisfeita, a resolução de tempo do sistema é a mais alta possível. Por outro lado, se $(K-M) \neq 1$, significa que as frequências dos sinais A e B não estão suficientemente próximas, o que resulta em uma menor resolução de tempo.

A Figura 10 apresenta um exemplo onde o termo $(K-M) \neq 1$, note que a variável Δt assume um valor de menor resolução em comparação com o sistema apresentado na Figura 8 onde o termo $(K-M) = 1$.

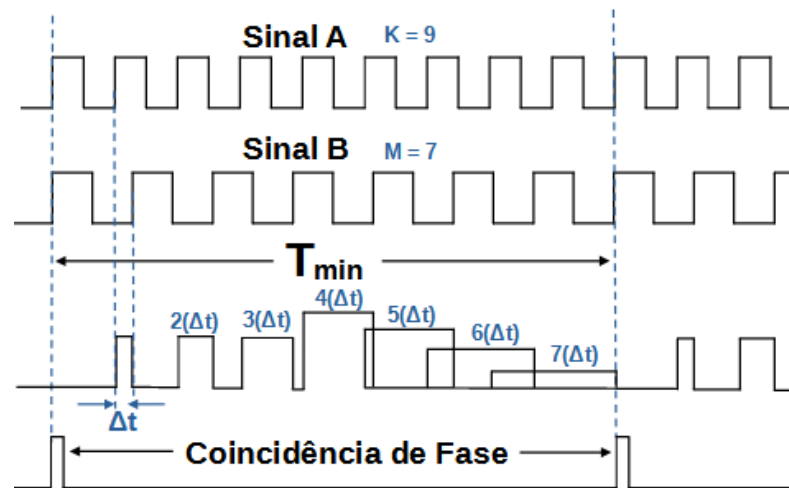


Figura 10: Diagrama de tempo exemplificando um sistema onde o termo $(K-M)$ é diferente de 1. (Figura do autor).

A quantidade de diferenças de fase quantificadas dentro do período T_{min} será sempre igual ao número de oscilações do sinal de frequência mais baixa. Em nosso exemplo na Figura 8, o sinal B é o sinal mais lento com uma contagem de $M = 8$ oscilações dentro de T_{min} , que é exatamente igual ao número de diferenças de fase observadas entre os sinais A e B dentro do período T_{min} .

O valor mínimo que T_{out} pode assumir é quando a variável n for igual zero, o que faz com que o termo $n(K-M)$ também seja igual a zero, não sendo contabilizado nenhuma oscilação dos sinais A e B .

Por outro lado, o valor máximo que a saída T_{out} pode assumir dentro do período T_{min} é quando o termo $n(K-M) = M$, esse será o ponto de contabilização da última oscilação dos sinais A e B antes da próxima coincidência de fase destes dois sinais. Portanto, quando o T_{out} for máximo ele atingirá uma largura de tempo igual à duração do período do sinal de maior frequência que, no caso da Figura 8, é o período do sinal A como apresentado na Equação (15), obtida através de manipulações das Equações (14) e (10).

$$\begin{aligned} T_{out} &= n(K-M) \left(\frac{T_{min}}{KM} \right) \\ T_{out} &= M \left(\frac{T_{min}}{KM} \right) \\ T_{out} &= \left(\frac{T_{min}}{K} \right) = T_A \end{aligned} \quad (15)$$

Portanto, considerando o período T_{min} , podemos enunciar a inequação apresentada em (16), onde o termo $n(K-M)$ assume valores de 0 a M .

$$0 \leq n(K-M) \leq M \quad (16)$$

Utilizando as Equações (6) e (13), e considerando o termo $(K-M) = 1$, a expressão da Equação (17) é definida. Manipulando (17) com o auxílio das Equações (9) à (12), obtêm-se as expressões úteis apresentadas na Equação (18), tais expressões auxiliam no cálculo das principais variáveis até aqui apresentadas e conseqüentemente ajudam no projeto desta arquitetura de DTC.

$$T_{out} = n\Delta t$$

$$\begin{aligned} T_{out} &= n(K-M) \left(\frac{T_{min}}{KM} \right) \\ n\Delta t &= n(K-M) \left(\frac{T_{min}}{KM} \right) \\ \Delta t &= \left(\frac{T_{min}}{KM} \right) \end{aligned} \quad (17)$$

$$\Delta t = \left(\frac{T_{min}}{KM} \right) = T_B - T_A = \frac{1}{K} \cdot T_B = \frac{1}{M} \cdot T_A = \frac{f_{max}}{f_A \times f_B} \quad (18)$$

A resolução do DTC Vernier proposto em nosso trabalho depende das relações entre os sinais A e B . À medida que a frequência destes sinais se aproxima, e respeitando a condição para que os índices K e M sejam inteiros e primos entre si, a largura de tempo mínima Δt diminui, aumentando a resolução do sistema. Simultaneamente, o período comum T_{min} entre estes dois

sinais aumenta, indicando que são possíveis mais diferenças de fase entre as oscilações dos sinais A e B dentro período T_{min} , o que naturalmente expande a faixa dinâmica do sistema (*Dynamic Range*) sendo possível utilizar uma entrada digital com mais bits.

3.2. Implementação do DTC no FPGA

Este projeto de DTC, utiliza o princípio Vernier com osciladores, e sua implementação é possível de ser feita na maioria dos FPGAs comerciais que dispõem de blocos PLL. Essa característica torna a arquitetura particularmente atraente para aplicações em FPGAs, aproveitando os recursos disponíveis para viabilizar uma implementação que alia tamanho reduzido junto com boas métricas de performance. Nesta seção, serão discutidos os aspectos da implementação do projeto do DTC Vernier no FPGA, abordando tanto os desafios quanto as soluções adotadas para otimizar a resolução do sistema.

3.2.1. Características da Plataforma de desenvolvimento

O DTC Vernier foi desenvolvido em linguagem de descrição de *hardware* (*Hardware Description Language – HDL*), tendo sido usado mais especificamente a linguagem VHDL. A simulação foi realizada com o simulador GHDL, em conjunto com o visualizador de formas de onda GTKWave.

A implementação utilizou a placa de desenvolvimento DE0 da TERASIC apresentada na Figura 11, que incorpora o chip FPGA da família Cyclone III EP3C16F484C6N da antiga ALTERA (atualmente Intel) [34]. Tal placa de desenvolvimento já inclui a interface de comunicação e o *hardware* necessário para o desenvolvimento e testes dos projetos em FPGA.

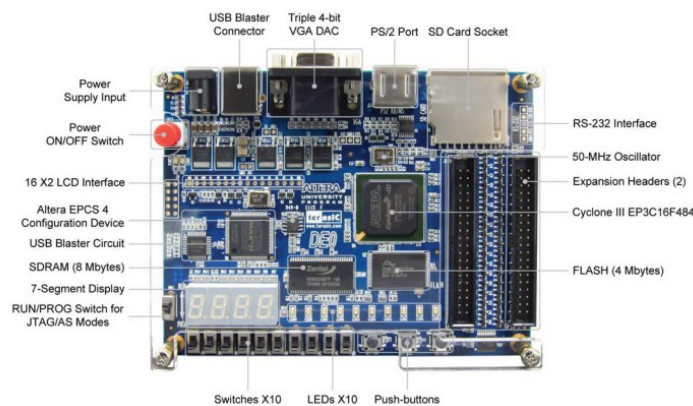


Figura 11: Plataforma de desenvolvimento em FPGA, Terasic DE0 [34].

A síntese para o FPGA foi realizada no software Quartus II, que é disponibilizado pelo fabricante do chip FPGA (ALTERA/INTEL). Essa ferramenta de *software* oferece um sistema integrado com um conjunto de funcionalidades que incluem, ferramentas para análise de tempo, síntese, otimização e gerenciador de blocos IP (*Intellectual Properties* – IP core) que são blocos paramétricos, pré-processados e pré-sintetizados de propriedade intelectual, que incluem os blocos PLL usados em nosso trabalho.

O projeto sintetizado foi caracterizado usando um osciloscópio *Agilent MSO-X 2004A* com largura de banda de 70 MHz e taxa de amostragem de 2 GSa/s (Giga amostras por segundo) [35]. Isso significa que o osciloscópio faz uma leitura a cada 500 ps, o que torna desafiadora a medição adequada de um sistema de resolução muito alta, que é o caso do DTC proposto cuja resolução pode chegar à valores de 12,5 ps. Se o osciloscópio fosse usado para medir a largura de pulso de 12,5 ps, efeitos de *aliasing* (efeitos resultantes de uma baixa taxa de amostragem) resultariam em uma saída que não corresponderia ao estado real do sistema.

Levando em conta a limitação da taxa de amostragem do osciloscópio utilizado, a topologia proposta de DTC foi implementada em duas versões: uma de "alta resolução" (12 ps) e uma de "baixa resolução" (990,1 ps). A versão de "baixa resolução" pode ser mais bem caracterizada pelo osciloscópio, já que exige uma menor taxa de amostragem, conforme definido pelo teorema de *Nyquist* [36].

3.2.2. Estratégia de Geração de Sinais

O DTC proposto na topologia Vernier é um sistema que gera sinais de alta resolução, podendo gerar sinais tão breves quanto algumas dezenas de picossegundos. Para tanto é necessário que os sinais usados em seu princípio sejam apropriadamente estáveis e com baixo nível de ruído. Esta subseção oferece uma descrição do arranjo criado para a geração dos sinais usados na referida arquitetura.

Conforme a informação disponibilizada pela ficha técnica [37] e do manual dos blocos PLL [38], a frequência do sinal de saída do PLL é descrita pela Equação (19), onde f_{out} é justamente a frequência do sinal de saída, f_{ref} é a frequência do sinal de referência, W é o índice multiplicador e N é o índice divisor. Ambos os índices devem ser inteiros, positivos e diferentes de 0.

$$f_{out} = f_{ref} \times \frac{W}{N} \quad (19)$$

Os índices W e N podem variar de 1 a 512, e a ficha técnica [37] indica uma frequência máxima de saída de 472,5 MHz. No entanto, quando múltiplas saídas são habilitadas em um único módulo PLL (onde cada bloco PLL pode possuir até cinco saídas individuais), a capacidade de geração é impactada. Isso limita a faixa de frequências que podem ser sintetizadas para cada saída separadamente, o que inviabiliza o uso de um único PLL para gerar os sinais A e B , dado que essa limitação afeta a propriedade de se sintetizar sinais com frequência próxima entre si.

Outra dificuldade encontrada na geração dos sinais foi a respeito do sinal de referência f_{ref} que é importante para a síntese de sinais no PLL. A placa de desenvolvimento DE0 oferece um oscilador a cristal de 50 MHz para ser usado ou como sinal de *clock* do sistema, ou como sinal de referência para a síntese de sinais no PLL. Verificou-se que o sinal de referência de 50 MHz não possibilitava sintetizar sinais A e B com frequências próximas o suficiente para se alcançar um DTC com uma resolução abaixo de 100 ps.

A solução encontrada foi utilizar um bloco PLL exclusivamente para gerar um sinal de referência f_{ref} para ser usado na síntese dos sinais A e B . Foi constatado que com $f_{ref} = 50,4$ MHz foi possível de se sintetizar sinais A e B com frequências próximas o suficiente para atingir uma resolução de 12.5 ps.

A Figura 12 (a) ilustra as frequências que podem ser sintetizadas em um módulo PLL quando apenas uma saída está habilitada. O gráfico é baseado na Equação (19) onde é fixado o valor de $f_{ref} = 50,4$ MHz, e os índices W e N são variados com valores de 1 a 512.

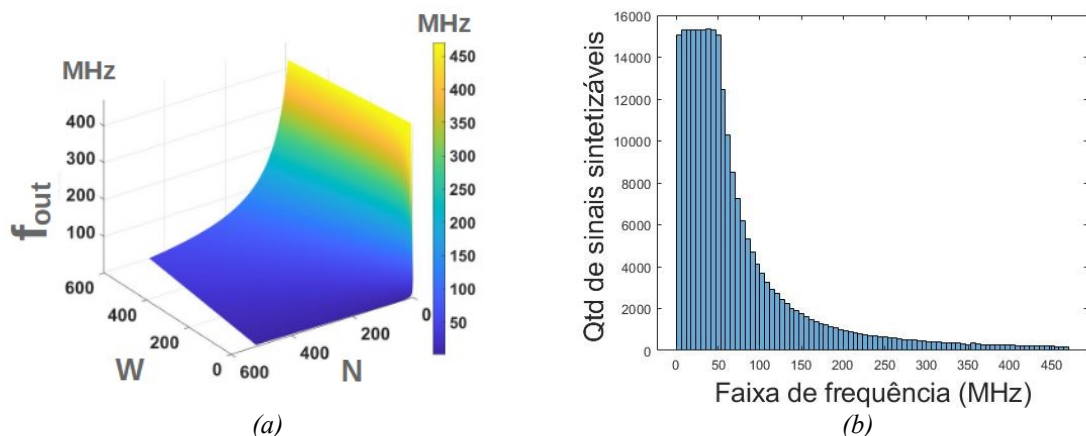


Figura 12: (a) Frequências de saída possíveis de serem geradas pelo PLL. (b) Histograma mostrando a distribuição dos sinais gerados no PLL por faixa de frequência. (Figura do autor).

Na Figura 12 (a), destaca-se que, entre as frequências sintetizadas pelo bloco PLL, a área mais favorável para selecionar sinais com frequências próximas está na região mais plana do gráfico, caracterizada pela ausência de inclinações abruptas no eixo vertical. Nessas condições, alterações nos parâmetros W e N resultam em variações menores nas frequências

sintetizadas. Com base na análise visual do gráfico, essa região corresponde à área azul-escura e está localizada abaixo de 50 MHz.

Já na Figura 12 (b), é apresentada a distribuição da quantidade de sinais sintetizáveis por faixa de frequência. Observa-se que a região abaixo de 50 MHz possui a maior densidade de sinais sintetizáveis, correspondendo a aproximadamente 60% do total. Essa maior densidade evidencia que essa faixa abaixo de 50 MHz é a mais adequada para a obtenção de sinais com frequências próximas entre si.

Os sinais utilizados no princípio Vernier devem manter uma ótima relação de regularidade entre suas frequências. Imprecisões como as ocasionados por sinais ruidosos podem comprometer a detecção de coincidência de fase ou a linearidade da saída, afetando diretamente a precisão e o próprio funcionamento do sistema. Desvios momentâneos de frequência causados por índices altos de *Jitter* podem influenciar negativamente diversas etapas do DTC baseado no princípio Vernier com osciladores, com destaque para a detecção de coincidência de fase. Portanto, assegurar a estabilidade das frequências é essencial para garantir que o sistema opere com a resolução desejada.

Uma forma de assegurar a regularidade de fase entre os sinais é garantir que eles sejam síncronos. Dois sinais periódicos são considerados síncronos quando são derivados da mesma fonte ou conseguem manter uma relação de fase constante entre si ao longo do tempo [39]. Sinais gerados a partir de um PLL são síncronos com o sinal de referência, pois estes compartilham da mesma fonte [40].

Deste modo para garantir que todos os sinais usados no princípio Vernier tenham uma boa relação de sincronia entre si, é necessário que todos tenham o mesmo sinal de referência. Portanto, na arquitetura proposta, a mesma referência de frequência será utilizada para acionar os sinais *A*, *B* e o sinal de *clock* do detector de fase, conforme mostrado na Figura 13.

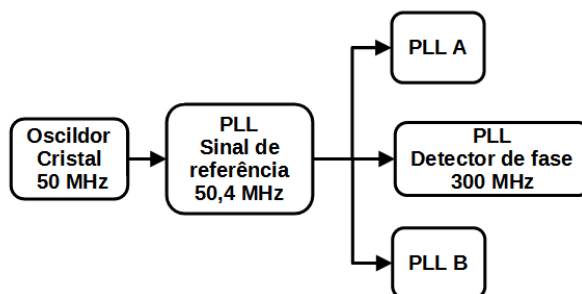


Figura 13: Diagrama da geração de sinais. (Figura do autor).

Nota-se também, na Figura 13, que é usado um oscilador a cristal da placa de desenvolvimento DE0 de 50 MHz para gerar um novo sinal de referência de 50,4 MHz com o

objetivo de se alcançar uma maior resolução do DTC Vernier através da a síntese de sinais com a frequência mais próxima o possível.

Todos os sinais utilizados no DTC têm como referência um sinal base de 50,4 MHz. Essa estratégia atende a um requisito fundamental para que os sinais gerados apresentem regularidade de fase entre si, sendo síncronos em relação a um ponto comum. Além disso, a boa qualidade dos blocos PLL disponíveis no chip FPGA, que apresentam baixo valor de *Jitter*, ajuda na confiabilidade do funcionamento da arquitetura implementada.

A configuração adotada é essencial para garantir a regularidade temporal entre os sinais. Experimentos com outras configurações para a geração desses sinais, usando sinais de fontes externas, ou com sinais de referência diferentes para geração dos três sinais citados, resultaram em instabilidades que comprometeram o correto funcionamento do DTC Vernier.

Alternativas como geradores externos de sinais, incluindo geradores de função e osciladores a cristal montados individualmente, foram experimentados, mas não se mostraram funcionais nos testes realizados. Essas fontes externas sofrem com a captação de ruídos e atrasos de propagação na linha de transmissão, em um nível que impede a manutenção de uma regularidade temporal aceitável dos sinais. Essas irregularidades acabam por comprometer a detecção de coincidência de fase, uma etapa fundamental para o funcionamento do DTC Vernier.

3.2.3. Arquitetura Proposta

A implementação do DTC Vernier no FPGA, é apresentada no diagrama de blocos da Figura 14. Nesse diagrama, destaca-se, no retângulo pontilhado em azul, o arranjo dos PLLs responsável pela geração dos sinais necessários ao funcionamento do sistema. Esse arranjo é responsável tanto pelo sinal de *clock* utilizado no detector de coincidência de fase quanto pela geração dos sinais *A* e *B*.

Estes sinais são direcionados ao bloco detector de coincidência de fase. Uma vez detectada tal coincidência, este gera um pulso que serve como sinal de partida para os contadores *A* e *B* iniciarem a contagem dos ciclos dos seus respectivos sinais até que a contagem atinja o valor definido pela palavra digital de entrada. Ao final da contagem, um pulso é gerado para cada contador, e a diferença de tempo entre as bordas de subida destes pulsos representa a informação no formato de tempo T_{out} .

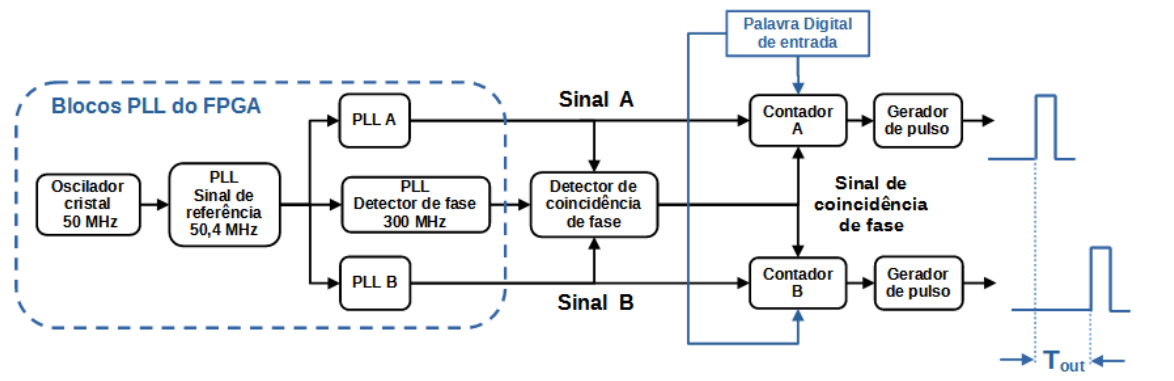


Figura 14: Diagrama de blocos do DTC Vernier. (Figura do autor).

A Figura 15 mostra um exemplo do diagrama de tempo representando a operação do DTC Vernier. A entrada digital é carregada nos contadores após o recebimento do sinal de coincidência de fase e neste momento é iniciada a contagem das oscilações de cada sinal. Essa entrada digital equivale a variável n na Equação (6). Assim que a contagem for concluída, os contadores geram um pulso para sinalizar o início (*Start*) e o final (*Stop*) da variável de tempo de saída T_{out} .

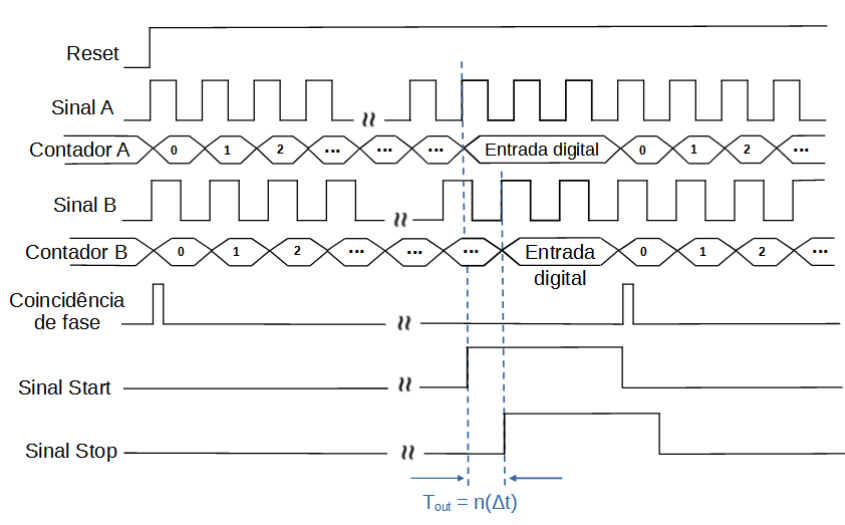


Figura 15: Diagrama de tempo do DTC Vernier. (Figura do autor).

Para fins de comparação, a Figura 16 apresenta os diagramas de tempo obtidos por meio da simulação do DTC Vernier. Na simulação, foram utilizados os parâmetros especificados na Tabela 2, garantindo que os valores de T_{min} , f_{max} e Δt estejam alinhados com a teoria descrita na Subseção 3.1. Essa configuração permite validar o comportamento teórico do DTC e verificar sua consistência em relação aos resultados esperados.

Tabela 2: Parâmetros usados na simulação apresentada na Figura 16.

Sinal A	Sinal B	K	M	T_{min}	f_{max}	Δt
285,71 kHz	250 kHz	8	7	28 μ s	35,71 kHz	500 ns

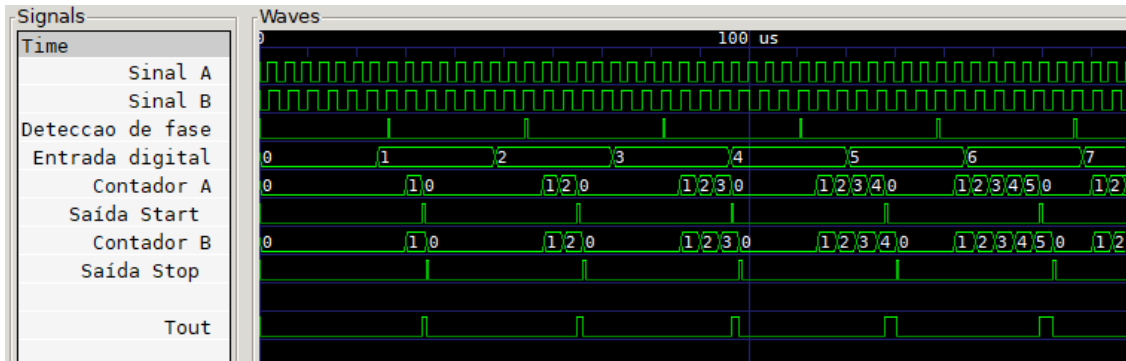


Figura 16: Imagem da tela do diagrama de tempo da simulação. (Figura do autor).

Na simulação mostrada na Figura 16, foi utilizada uma resolução baixa (500 ns) propositalmente para facilitar a visualização dos sinais. Neste diagrama, os sinais A e B oscilam continuamente, e o detector de coincidência de fase (terceira linha, "Detecção de fase") identifica o instante em que as bordas de subida desses sinais coincidem, gerando um pulso de detecção. A Figura 17 mostra medições feitas na simulação, onde é apresentado na Figura 17 (a) a medição de Δt que é igual à largura do bit menos significativo LSB, e na Figura 17 (b) é apresentado a medição de T_{min} . Ambas as medições correspondem com os valores teóricos apresentados na Tabela 2.

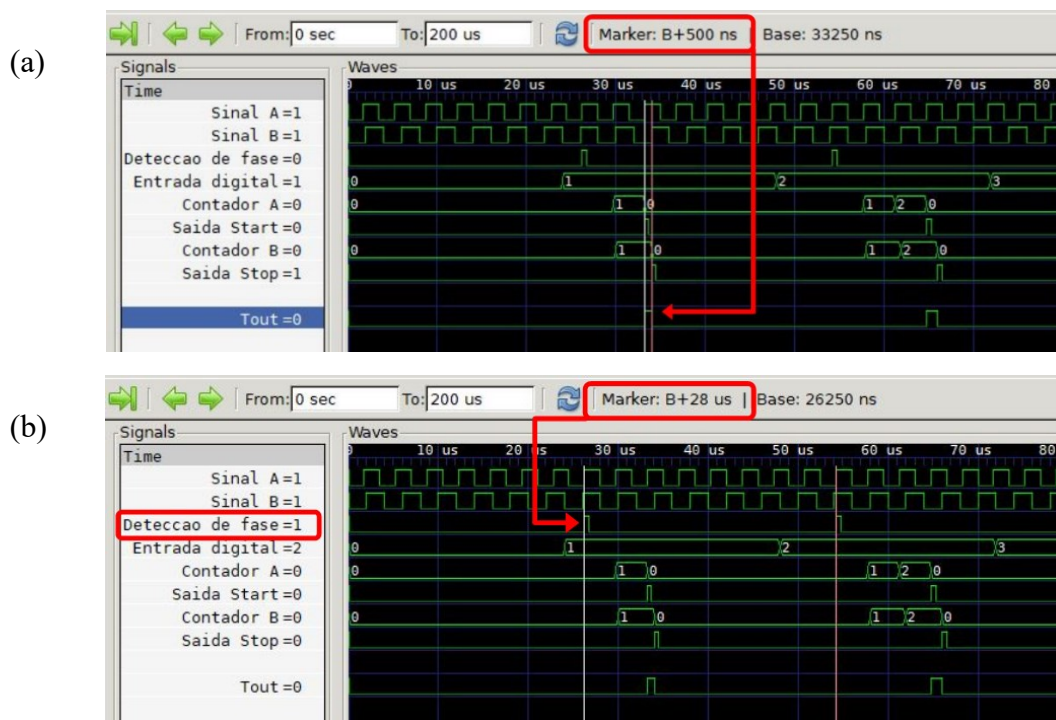


Figura 17: Medições feitas na simulação, (a) Δt , (b) T_{min} . (Figura do Autor).

O pulso de detecção da coincidência de fase é o sinal para carregar o valor da entrada digital nos contadores A e B , e no mesmo momento, a contagem das oscilações de seus respectivos sinais é iniciada até atingirem o valor da entrada digital.

Quando a contagem é concluída, cada contador gera um pulso: o contador *A* produz o pulso “Saída *Start*” e o contador *B* gera o pulso “Saída *Stop*”. A diferença de tempo entre as bordas de subida desses dois pulsos corresponde à conversão da palavra digital de entrada em uma informação correspondente em modo de tempo.

O sinal T_{out} , exibido na simulação, é meramente ilustrativo, pois o DTC é capaz de gerar variações de tempo muito breves, na ordem de alguns picossegundos. Tais variações excedem em muito a capacidade da taxa de subida da maioria dos sistemas digitais como o FPGA utilizado neste trabalho, que possui uma taxa de subida de aproximadamente $350 \mu V/ps$ [37]. Essa característica inviabiliza a representação direta de T_{out} através de um pulso, pois o mesmo se tornaria excessivamente distorcido, principalmente para representações de menor valor do comando digital de entrada.

A correta operação do DTC depende da detecção de uma precisa e correta coincidência de fase, onde o bloco responsável por tal detecção é apresentado na Figura 18. A detecção no momento errado levará a um erro na conversão, e a falha na detecção aumentará o tempo de latência do DTC (tempo entre o início e o final de uma conversão), pois a conversão será iniciada somente na próxima coincidência de fase, que ocorrerá após passado o período T_{min} .

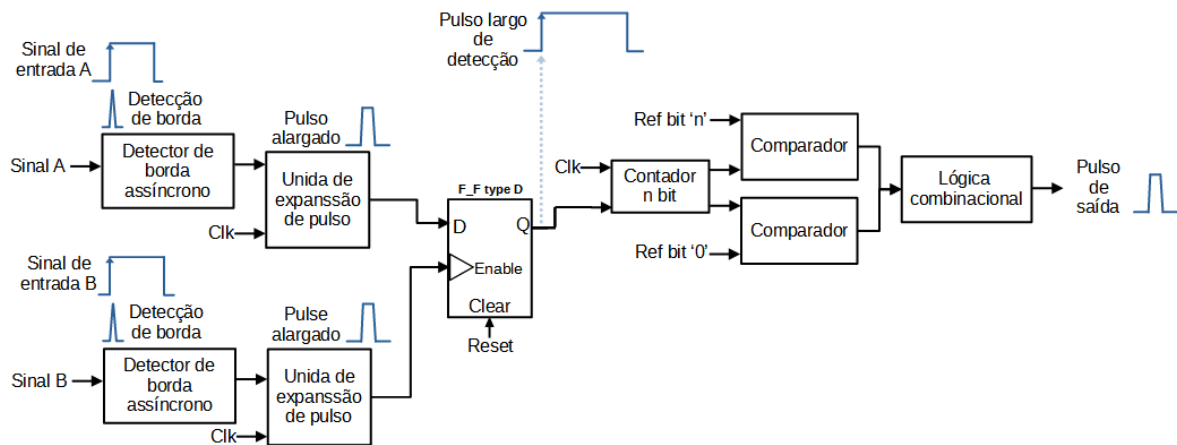


Figura 18: Diagrama de blocos do detector de coincidência de fase. (Figura do autor).

Na descrição deste bloco, seguiu-se boas práticas de desenvolvimento para FPGAs. Os dispositivos FPGA possuem recursos internos otimizados para operar com sinais de *clock*, sendo muito eficientes em projetos digitais síncronos. Esses recursos, como *flip-flops* dedicados e redes de distribuição de *clock* (*clock trees*), foram projetados para minimizar atrasos e maximizar a confiabilidade em tarefas relacionadas ao *clock*, incluindo a detecção de bordas [41].

Entretanto, esses recursos são limitados em número e são projetados exclusivamente para sinais de *clock*. O uso desses elementos para sinais que não representem o *clock* do sistema é desaconselhado, especialmente em cenários onde ocorre cruzamento de domínio de *clock*, ou seja, a troca de informações entre sistemas que operam com frequências ou fases de *clock* diferentes. O uso inadequado pode introduzir erros de sincronização e comportamento indesejado no sistema [42].

Utilizou-se detectores de borda assíncronos para amostrar a borda de subida dos sinais *A* e *B*. A escolha por este tipo de detector é necessária para evitar atrasos nos pulsos de detecção.

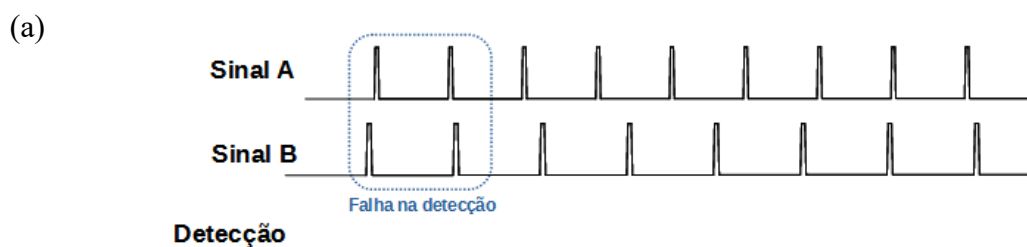
Enquanto detectores síncronos sincronizam o evento detectado com a borda ativa do *clock* do sistema, o que pode introduzir um atraso adicional devido à sincronização, os detectores assíncronos operam independentemente do *clock* do sistema. Dessa forma, eles eliminam o risco de atrasos causados por essa sincronização, permitindo que as bordas dos sinais sejam detectadas imediatamente.

O detector de borda assíncrono gera um pulso de saída muito estreito e, para alargá-lo, foi usado um gerador de pulso baseado em contadores digitais.

O elemento fundamental para a detecção da coincidência de fase é o *flip-flop* tipo *D* (DFF). Os pulsos vindos da etapa de detecção da borda de subida são inseridos na entrada *D* e no *enable* do DFF. Preferencialmente o sinal com a frequência menor deve ser conectado no *enable*, e o de frequência maior deve ser colocado na entrada *D*. Quando os dois pulsos estão no nível alto, a saída *Q* também comuta para o nível alto.

A Figura 19 mostra o diagrama de tempo do detector de coincidência de fase. A Figura 19 (a) exemplifica quando os pulsos são demasiadamente estreitos podendo causar falhas na detecção, principalmente por violar os tempos de *setup* e *hold* do DFF, essas violações podem provocar o fenômeno da metaestabilidade, onde a saída do flip-flop torna-se imprevisível, comprometendo o funcionamento do sistema.

Em contraste, a Figura 19 (b) demonstra que, ao ajustar a largura dos pulsos para valores adequados, é possível solucionar esse problema, garantindo a estabilidade e a confiabilidade na detecção da coincidência de fase.



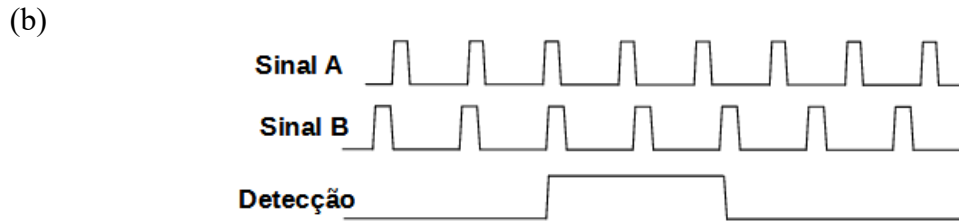


Figura 19: Diagrama de tempo do detector de coincidência de fase, (a) pulsos muito estreitos, (b) com pulsos alargados no tempo. (Figura do autor).

O método de detecção de coincidência de fase usado nesse trabalho, também é mencionado em [31], porém tal trabalho não o implementa devido ao já mencionado problema com metaestabilidade em configurações de sistema de alta resolução (1.53 ps de resolução em [31]), o que implica em tempos de transição desafiadores, pois os sinais *A* e *B* estão muito aproximados. Em nosso trabalho, o DTC chega a valores de 12,5 ps de resolução.

Foi verificado que configurações abaixo de 12,5 ps apresentaram instabilidades ao serem medidas no osciloscópio. Tal fenômeno reforça a hipótese de que a metaestabilidade na detecção de coincidência de fase pode causar instabilidade no sistema do DTC Vernier, principalmente em sistemas de com alta resolução.

O restante do circuito acoplado na saída *Q* do *flip-flop* tipo *D* serve para gerar um pulso estreito que será a saída do detector de coincidência de fase.

3.2.4. Discussão sobre trabalhos anteriores

No princípio Vernier com osciladores a detecção da coincidência de fase é uma etapa importante do processo, pois a correta conversão da entrada digital somente será atingida com a correta identificação do momento em que as fases dos dois sinais periódicos coincidirem.

Neste sentido, além de se obter inovações nos parâmetros de performance como a resolução, a linearidade e o consumo de energia, algumas das inovações para essa arquitetura também partem da inovação na estratégia de detecção da coincidência entre as fases.

Os artigos [16] e [31] apresentam trabalhos de DTC com o princípio Vernier usando osciladores, onde conseguem um sistema com resolução de cerca de 4 ps e 1,6 ps respectivamente. Estes trabalhos são implementados em dispositivos FPGA de alta performance como o Xilinx Virtex 6 [16] e o Altera STRATIX III [31].

A principal diferença entre esses trabalhos é a estratégia de detecção da coincidência de fase entre os sinais usados no princípio Vernier, onde ambos os trabalhos conseguem eliminar

a necessidade de um bloco de circuito exclusivo para a tal detecção, se aproveitando dos recursos e capacidades disponíveis nos chips FPGA de alta performance.

O trabalho [16] elimina o uso de um bloco específico de detecção de coincidência entre as fases usando as funcionalidades do bloco gerenciador de *clock*.

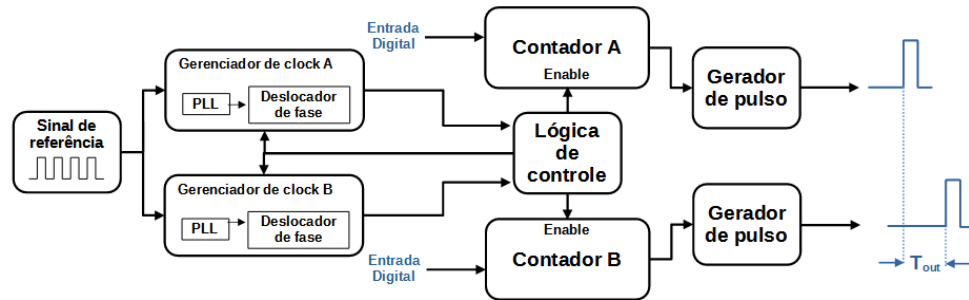


Figura 20: Diagrama de Blocos do DTC implementado no trabalho [16] (Adaptado pelo autor)

A Figura 20 apresenta um exemplo do diagrama de blocos do DTC do trabalho [16]. Os sinais usados no princípio Vernier são gerados no gerenciador de *clock*. Quando é solicitado uma conversão pelo sistema, o gerenciador de *clock* é capaz de deslocar as fases dos sinais até que atinjam a coincidência. Embora seja uma estratégia eficiente, essa funcionalidade é proprietária do chip FPGA usado, o que torna difícil de reaproveitar essa proposta para uma implementação em outros modelos de chips FPGA ou mesmo projetos de chips ASIC.

O trabalho [31] também consegue eliminar o bloco de detecção de coincidência de fase usando uma estratégia de geração aproveitando da capacidade dos blocos PLL do chip FPGA STRATIX III que tem uma flexibilidade maior de síntese, quando comparado com os blocos PLL de chips de baixo custo, como o chip Cyclone III usado em nosso trabalho.

No trabalho [31] os dois sinais são gerados por PLLs distintos. Usando um sinal de referência que é igual ou múltiplo do período T_{min} de coincidência entre as fases. Deste modo a necessidade de se detectar a coincidência entre as fases é eliminada pois as fases dos sinais e do sinal de referência coincidem em toda a borda de subida do sinal de referência.

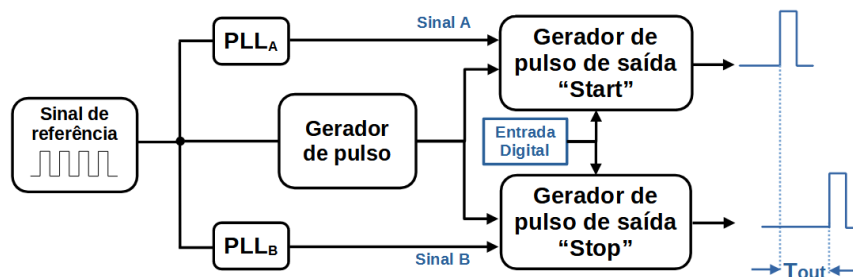


Figura 21: Diagrama de Blocos do DTC implementado no trabalho [31]. (Adaptado pelo autor).

Embora seja uma estratégia eficaz, essa abordagem é mais adequada às capacidades de síntese dos PLLs presentes no FPGA STRATIX III, que permitem gerar sinais em uma faixa

ampla de frequências a partir de uma referência de frequência relativamente baixa. No trabalho [31] por exemplo, os melhores resultados foram obtidos utilizando sinais na faixa de 1,25 GHz para os dois sinais usados no princípio Vernier, muito além da capacidade máxima de síntese do 470 MHz do FPGA CYCLONE III. Como visto anteriormente, no FPGA empregado neste trabalho, a faixa mais adequada para gerar sinais de frequências próximas é abaixo de 50 MHz.

A principal diferença da proposta deste trabalho é a estratégia de detecção de coincidência de fase que foi projetada levando em consideração as capacidades do chip FPGA CYCLONE III. Nossa estratégia se difere dos trabalhos [16] e [31] por usar um bloco de detecção da coincidência de fase entre os sinais A e B , onde este bloco se utiliza de um conjunto de detectores de borda assíncronos, unidade de expansão de pulso e um *flip-flop* tipo D formando o circuito apresentado na Figura 18. Nossa proposta é verificada como funcional até uma resolução de 12,5 ps.

4. ANÁLISE DOS RESULTADOS

Neste capítulo, apresentamos uma análise dos resultados obtidos em nossos experimentos com o DTC Vernier. Primeiramente, será avaliado o nível de *Jitter* presente nos sinais *A* e *B*. A seguir, o sistema será configurado em duas versões de resolução: 990,1 ps e 12,5 ps. Em seguida, avaliamos a não linearidade diferencial (DNL) e integral (INL) da versão de 990,1 ps do sistema. Finalmente, será feita uma estimativa da variação da resolução ao longo das conversões do DTC em ambas as configurações do sistema, para verificar a influência de um possível acúmulo de erro na conversão dos valores mais altos da palavra digital de entrada. Essas medições e análises são importantes para validar a eficácia do DTC Vernier proposto, destacando suas capacidades em diferentes configurações de resolução.

A Tabela 3 apresenta os parâmetros usados para gerar os sinais *A* e *B* nas duas configurações do DTC Vernier, de 990,1 ps e 12,49 ps respectivamente. Na Tabela 3 também pode-se conferir os valores esperados de resolução Δt , do período de coincidência entre as fases T_{min} , da faixa de tempo T_{MAX} , e da faixa dinâmica DR.

Tabela 3: Especificações dos parâmetros usados nas duas configurações do DTC Vernier.

Osc. DE0 (MHz)	PLL ref. (MHz)	PLL A (MHz)	PLL B (MHz)	Δt (ps) esperado	T_{min} (μ s)	T_{MAX} (ns)	DR (dB)
50	50,4	10,1	10	990,1	10	99,01	40 ^(a)
50	50,4	10,00125	10	12,49	800	99,98	78 ^(b)

(a) - O número de conversões possíveis (100 conversões possíveis dentro do período T_{min}) é superior a 6 bits, mas inferior a 7 bits.

(b) - O número de conversões possíveis (8000 conversões possíveis dentro do período T_{min}) é superior a 12 bits, mas inferior a 13 bits.

As duas versões do DTC Vernier servirão para caracterizar esse sistema em concordância com o equipamento disponível para medição. Como foi explicado na subseção 3.2.1, o osciloscópio usado tem uma taxa de amostragem de 2 GSa/s, Ele é, portanto, mais adequado para a medição da versão do sistema de 990,1 ps, onde realizaremos as medições de não linearidades Diferencial e Integral. Para a versão de 12,49 ps avaliaremos a estimativa da resolução e da variabilidade temporal.

4.1. Avaliação do *Jitter* nos sinais sintetizados nos PLLs

De uma forma genérica, *Jitter* em sinais periódicos refere-se às variações temporais indesejadas na instantes de ocorrência dos pulsos de um sinal em relação ao seu tempo ideal [43]. A Figura 22 representa um exemplo visual do *Jitter* em um sinal.

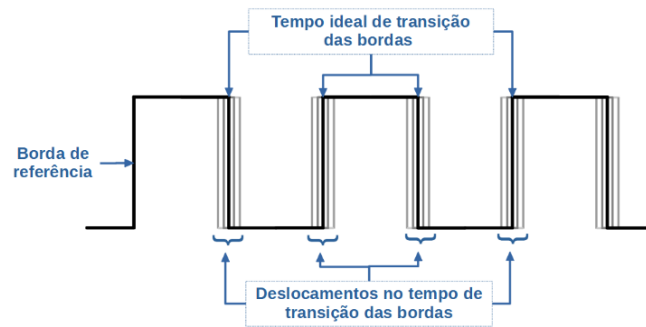


Figura 22: Exemplo visual da definição geral de Jitter. (Figura do autor).

Há algumas definições mais específicas de *Jitter* que atendem aos requisitos de aplicações mais especializadas como sistemas de comunicação e sistemas digitais, porém é comum se categorizar os tipos de *Jitter* separando-os em dois grupos principais [44];

- *Jitter* Randômico, onde as variações temporais nas bordas de transição do sinal periódico surgem de processos verdadeiramente randômicos, como o ruído térmico ou o ruído de cintilação (*Flicker*);

- *Jitter* Determinístico, onde as variações seguem padrões que acompanham fenômenos e fontes de ruídos não aleatórios, como injeção de ruído pela fonte de alimentação, pela interferência intersimbólica ou por *cross-talking*.

Como o DTC depende da medição precisa das oscilações dos sinais *A* e *B*, um nível elevado de *Jitter* nestes sinais pode resultar em erros, reduzindo a precisão ou a linearidade do sistema, uma vez que o princípio Vernier implementado neste trabalho utiliza contadores digitais para amostrar a quantidade de oscilações destes sinais periódicos. Portanto, a caracterização do *Jitter* das fontes geradoras de sinal usadas na arquitetura Vernier é relevante para se avaliar os limites de performance do DTC.

A metodologia usada para se caracterizar o *Jitter* usando o equipamento disponíveis, (osciloscópio mencionado na subseção 3.2.1) foi elaborada levando-se em consideração a definição de *Jitter* periódico. Ele é conhecido por ser uma importante definição de *Jitter* para sistemas digitais, uma vez que o mesmo leva em conta efeitos determinísticos e randômicos [45]. A definição de *Jitter* periódico (J_T) é apresentada na Equação (20), onde T_{ciclo} representa o ciclo medido, e T_{ideal} representa o valor esperado para o período.

$$J_T = T_{ciclo} - T_{ideal} \quad (20)$$

A Figura 23 representa graficamente a definição de *Jitter* periódico. Note que esse *Jitter* é medido em comparação ao período de cada ciclo com um valor de período ideal, levando-se em consideração as bordas de subida. Dado a natureza aleatória observada no *Jitter* é possível

utilizar uma densidade de probabilidade para entender onde estão concentrados os valores de desvios encontrados nas medições de *Jitter*.

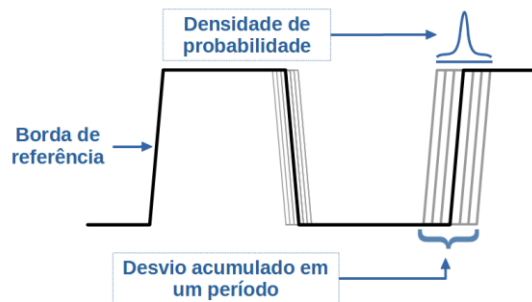


Figura 23: Exemplo da definição de *Jitter* periódico. (adaptado de [45]).

Para evitar a interferência de um viés humano nas medições, foi elaborada uma estratégia de medição desses ciclos individuais. Nessa estratégia, os ciclos são gravados em arquivos CSV com a utilização de um osciloscópio e um script do *Matlab* é usado para coletar o valor do período desses ciclos. Os arquivos CSV são lidos individualmente e suas formas de onda são plotadas em conjunto com uma linha mediana, conforme apresentado na Figura 24. Os pontos de interseção entre a linha mediana e as curvas do arquivo CSV são coletadas, com o auxílio da função *Intersections* (Schwarz, 2010) disponível em [46].

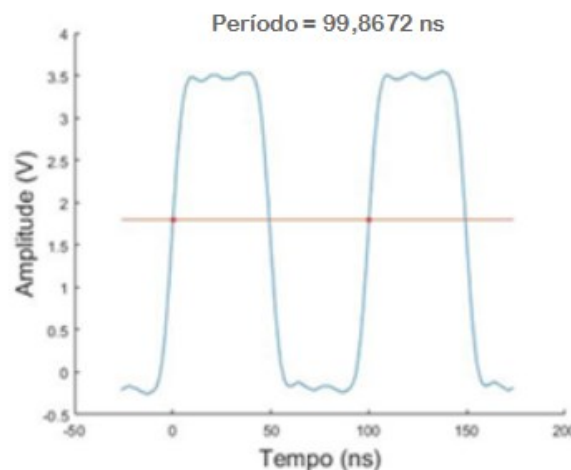


Figura 24: Estratégia de medição do período usando um script do *Matlab*. (Figura elaborada pelo autor).

Foram realizadas 1000 medições do período do sinal de 10 MHz, e o histograma da Figura 25 foi construído a partir dessas medições. Nota-se que a grande maioria das amostras se concentra em torno de 100 ns, que é justamente o período esperado para um sinal periódico com frequência de 10 MHz. A distribuição do histograma da Figura 25 apresenta uma forma aproximadamente simétrica em torno do valor médio perto de 100 MHz, o que evidencia uma natureza aleatória das variações do período.

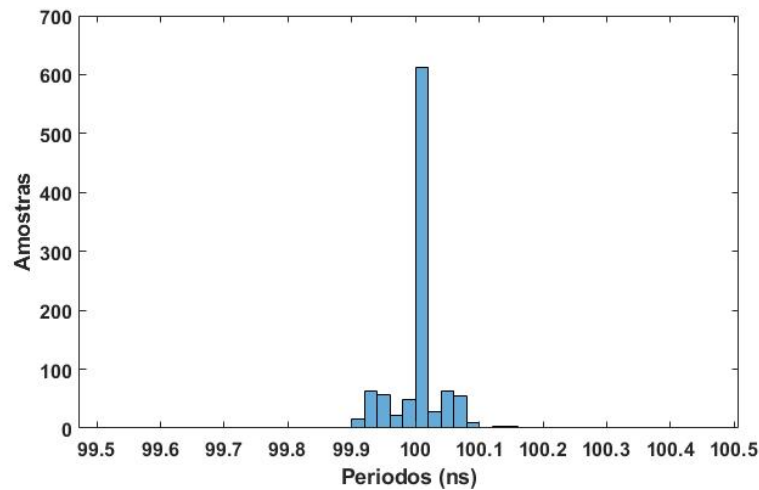


Figura 25: Histograma representando as medições dos períodos para sinais sintetizados na faixa de 10 MHz pelo PLL (Figura elaborada pelo autor).

A Equação (20) foi aplicada aos 1000 valores medidos e, com esses resultados, foi criada uma série de dados da medição instantânea de *Jitter* periódico. Para esta série, foram calculados a média μ e desvio padrão σ , e foi gerado a curva de distribuição normal, utilizando a função de densidade de probabilidade (DP), como apresentado na Figura 26. As equações utilizadas são apresentadas respectivamente em (21), (22) e (23). O valor médio obtido é de cerca de 13,54 ps, com um desvio padrão σ de 52,3 ps.

$$\mu = \frac{1}{N} \left(\sum_{i=1}^N J_i \right) \quad (21)$$

$$\sigma = \sqrt{\frac{1}{N-1} \left(\sum_{i=1}^N |J_i - \mu|^2 \right)} \quad (22)$$

$$DP(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (23)$$

Definimos, então, esse valor médio de 13,5 ps como o valor de *Jitter* periódico médio para os sinais gerados pelo PLL na região de 10 MHz. Esse valor será assumido para os sinais *A* e *B*, devido ao fato de que os sinais usados no princípio Vernier possuem frequências muito próximas.

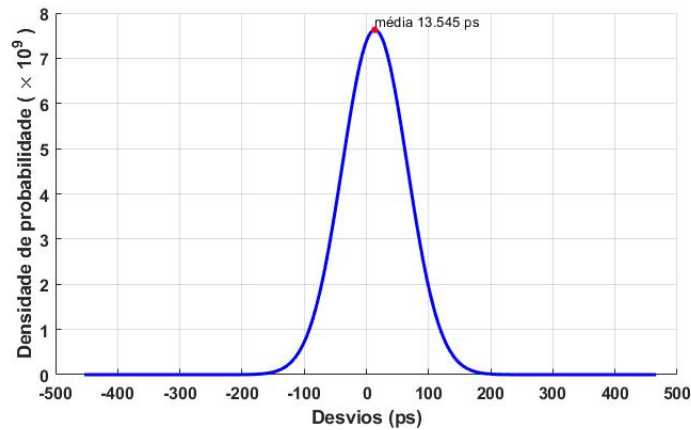


Figura 26: Distribuição Normal feita com base na diferença entre os períodos medidos e o período ideal, onde a média dessa distribuição é 13,54 ps (Figura elaborada pelo autor).

4.2. Implementação com 990,1 ps de resolução

Para validar a topologia proposta e nossa metodologia de caracterização, o circuito foi primeiramente sintetizado com resolução de 990,1 ps.

Conforme discutido na seção anterior, nosso osciloscópio tem uma frequência de amostragem de 2 GS/s onde a taxa de amostragem é de pelo menos 1 amostra a cada 500 ps. Tal taxa está no limite para se amostrar um sinal, no entanto, apenas os 3 menores valores da entrada digital são tão estreitos no tempo e, portanto, as entradas digitais restantes satisfazem a frequência de amostragem mínima para nosso sistema.

Para obter uma medição de alta qualidade, geralmente é desejado ter uma frequência de amostragem 10 vezes maior que a largura de banda do sinal. Como consequência, certamente poderíamos esperar uma medição precisa para sinais maiores de que 9,9 ns (correspondente a palavra digital de entrada de $b1010 - 10$ em decimal), como será discutido nesta seção.

4.2.1. Período de coincidência de fase

A primeira medição foi o período de coincidência de fase (T_{min}), conforme mostrado na Figura 27. A medição do osciloscópio indicou um T_{min} de 10 μ s, que condiz com o valor obtido na Equação (8) para uma resolução DTC de 990,1 ps. Tal valor está conforme com o estimado na Tabela 3 para esta configuração, demonstrando o bom funcionamento do detector proposto para esta arquitetura de DTC Vernier.

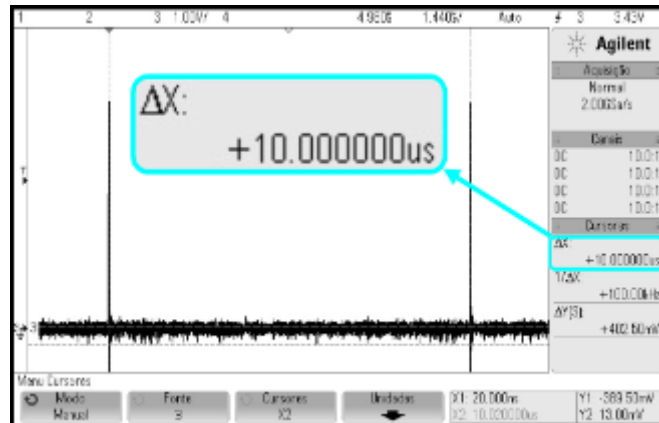


Figura 27: Medição do período T_{min} entre as detecções de coincidência de fase. (Figura elaborada pelo autor).

4.2.2. Linearidade

A segunda medição foi realizada para avaliar a linearidade do sistema. A palavra digital de entrada é representada por n em (6), seus valores foram transmitidos de forma sequencial e crescente, e cada valor de saída foi medido. As Equações (17) e (18) definem a não linearidade diferencial (DNL) e a não linearidade integral (INL), respectivamente. Nessas equações, T_{out} representa a largura de tempo gerada na saída do DTC, e LSB representa a largura de tempo ideal do bit menos significativo, que é igual a Δt na Equação (6). A Figura 28 apresenta os gráficos da não linearidade diferencial – DNL (a), e da não linearidade integral – INL (b) respectivamente.

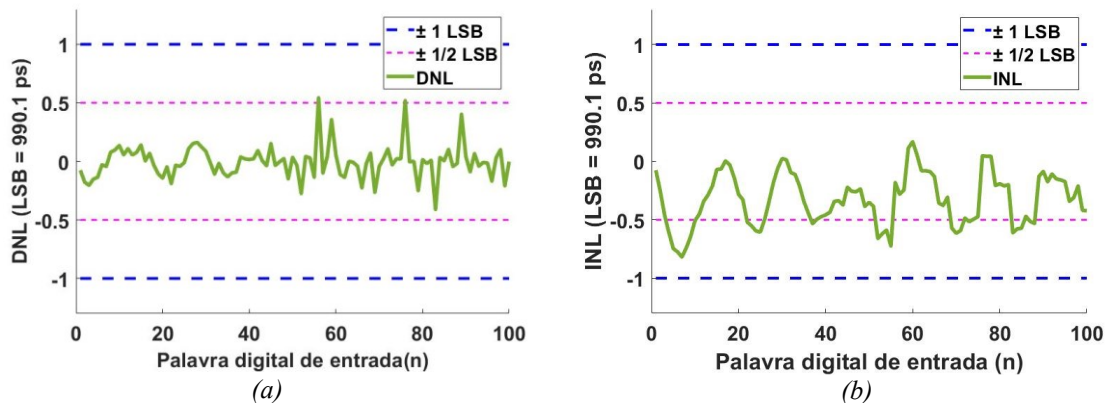


Figura 28: (a) não linearidade diferencial (DNL), (b) não linearidade integral (INL). (Figura elaborada pelo autor).

Observa-se que essa implementação apresenta boa linearidade, com DNL e INL dentro dos limites de ± 1 LSB. A norma IEEE 1241 [47] define métodos de teste para conversores ADC/DAC, onde os conceitos podem ser estendidos para o DTC. Embora a norma não especifique valores ideais para DNL e INL, ela menciona que um DNL = -1 LSB representa que um código foi perdido, uma vez que para este caso o valor de saída foi igual ao valor do

código de entrada anterior, e um $DNL = +1$ LSB representa um salto de duas vezes a largura de tempo esperada na saída. Dessa forma é compreendido que o DNL dentro do limite de ± 1 LSB impediria saltos ou perdas em relação ao valor ideal da variável de tempo de saída do conversor.

Na indústria, como regra geral, conversores utilizados em aplicações de alta precisão costumam apresentar DNL inferior a $\pm 0,5$ LSB, esse limite está relacionado a monotonicidade da função de transferência do DTC, ou seja, a função de transferência será sempre crescente e mais aproximada de um comportamento linear. Dispositivos DTC de uso geral geralmente mantêm o INL abaixo de ± 1 LSB, no entanto, esses valores dependem principalmente dos requisitos específicos da aplicação final do DTC [48].

A maior parte do gráfico do DNL da Figura 28 (a) apresenta variações de DNL abaixo do limite de $\pm 0,5$ LSB apresentando apenas duas leituras com valores ligeiramente superiores a este limite. Como resultado, a possibilidade de erros de conversão é minimizada, apresentando uma boa relação entre a linearidade e a monotonicidade de nosso sistema.

A Figura 29 apresenta o gráfico comparativo entre as funções de transferência ideal (representada pela curva azul) e real (representada pelos pontos em laranja) do DTC Vernier, em um formato semelhante ao ilustrado na Figura 3. Observa-se que a função de transferência real segue de forma muito próxima a curva ideal, o que evidencia a boa linearidade do sistema, apesar de possíveis desvios decorrentes de limitações práticas e não idealidades inerentes ao circuito.

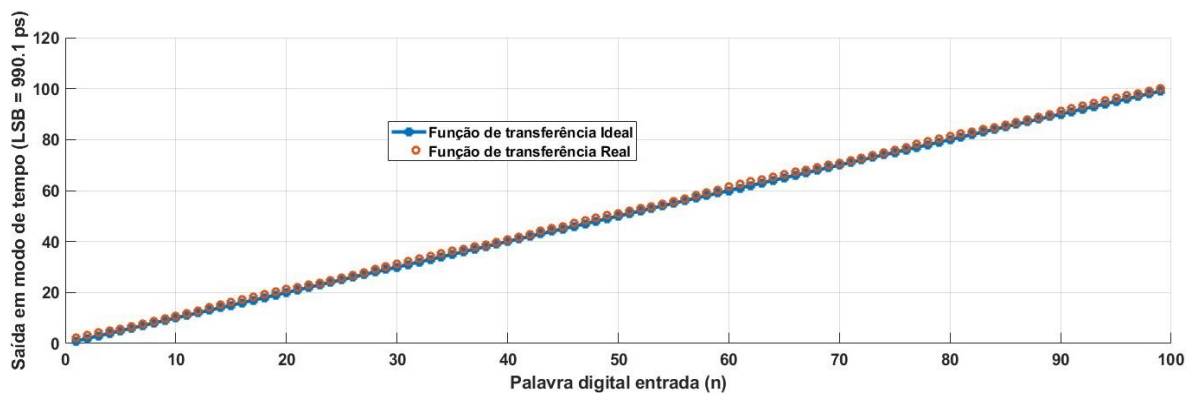


Figura 29: Comparação entre a Função de transferência ideal (azul) e Real (laranja).
(Figura elaborada pelo autor).

4.2.3. Resolução

A terceira medição envolveu a estimativa da resolução Δt usando a Equação (24). O valor da resolução foi avaliado para cada valor de saída T_{out} , e os resultados são apresentados na Figura 9.

$$\Delta t = \frac{T_{out}}{n} \quad (24)$$

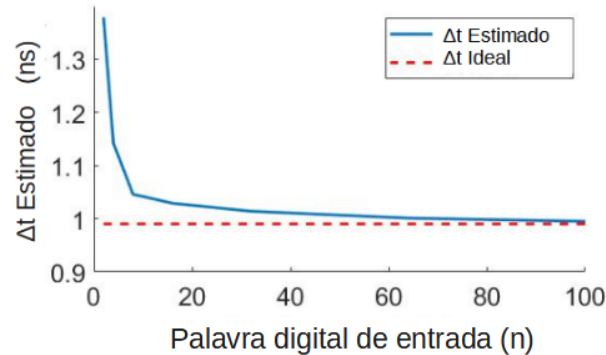


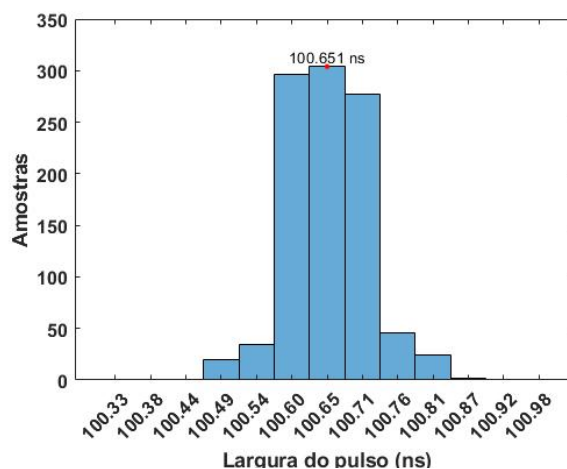
Figura 30: Comparação entre o LSB estimado pela Equação (23) e o LSB Ideal (990,1 ps).
(Figura elaborada pelo autor).

Na Figura 30 é observado que, embora os desvios sejam pequenos e não excedam 1 LSB, é notado que as palavras digitais de entrada de menor valor exibem desvios maiores entre as resoluções ideal e estimada em comparação com as entradas maiores.

Este resultado é esperado porque, para os comandos digitais de entrada iniciais, o pulso de saída é mais curto, e a frequência de amostragem do osciloscópio é apenas ligeiramente maior do que o mínimo necessário. À medida que a largura de saída aumenta, a qualidade da medição tende a melhorar.

Para estimar a resolução média, avaliamos a variação da resolução Δt em várias medições da maior faixa de tempo total possível da variável de saída. Especificamente, 1000 medições foram feitas utilizando o comando de entrada máximo (valor decimal 100), e o pulso de saída (idealmente 99,01 ns) foi registrado, conforme mostrado na Figura 31 (a).

(a)



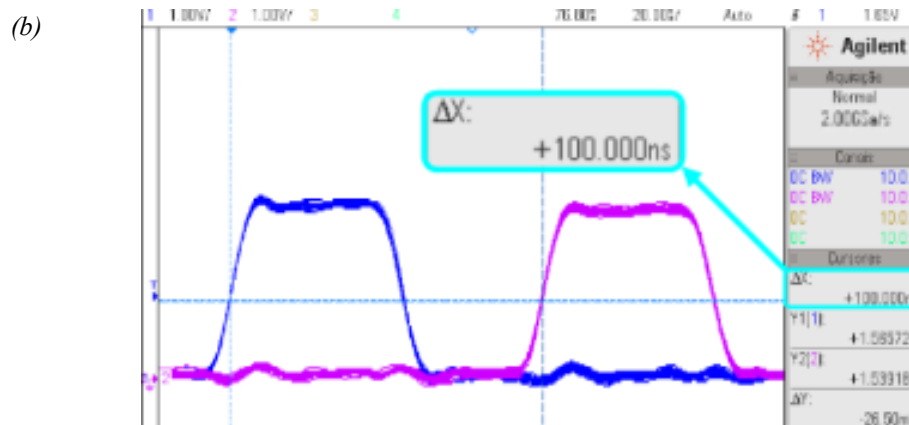


Figura 31: (a) Histograma para o comando decimal de entrada “100” em um DTC configurado para $\Delta t = 990,1$ ps, (b) Captura de tela da medição dos sinais de saída Start (azul) e Stop (magenta). (Figuras elaboradas pelo autor).

A Figura 31 (b) exibe uma imagem do osciloscópio de uma dessas medições usadas para construir o histograma na Figura 31 (a). Na imagem do osciloscópio, a forma de onda Azul representa o sinal *Start*, enquanto a forma de onda magenta representa o sinal *Stop*.

Ao analisar o histograma da Figura 31 (a), constatamos que o valor médio da largura do pulso é de aproximadamente 100,65 ns. Aplicando esse valor na Equação (24), obtém-se uma resolução média de 1006,5 ps. Essa resolução média apresenta uma diferença percentual de 1,64% em relação ao valor esperado de 990,1 ps, o que indica uma pequena discrepância em comparação com o valor teórico.

Ao considerar o pior caso registrado no histograma, onde o centro do intervalo (*bin*) correspondente é de 100,81 ns, a diferença percentual em relação ao valor esperado aumentou ligeiramente para 1,86%. Mesmo nesse cenário extremo, a variação observada na resolução é inferior a 2%.

Essa análise revela que, mesmo nos piores resultados, a variação na resolução esperada do sistema é mínima, mantendo-se dentro de um limite de menos de 2% muito longe do limite de 1 LSB (100%) onde erros de conversão e a perda de resolução do DTC seriam muito mais prováveis. Esse resultado é um forte indicativo de que as variações acumuladas ao longo da contagem dos ciclos de ambos os sinais *A* e *B* não impactam de maneira significativa as conversões realizadas pelo sistema. Em outras palavras, o sistema demonstra uma robustez considerável, garantindo a confiabilidade das conversões.

4.3. Implementação com 12,5 ps de resolução

Para estimar a resolução na implementação de alta resolução, foi usado o método indireto usando a Equação (24) tal como foi feito na seção anterior, já que a medição direta é inviável, conforme já foi detalhadamente abordado na Subseção 3.2.1. Nesse contexto, pretende-se medir o valor da faixa de tempo total, que é 99,92 ns, correspondente ao último valor decimal (8000) do DTC Vernier projetado.

É relevante destacar que, para garantir a alta resolução desejada, as frequências dos sinais A e B foram ajustadas para diferirem em menos de 1%, resultando em uma diferença de aproximadamente 1,25 kHz, como detalhado na Tabela 3. Essa pequena discrepância permite que os passos da diferença de fase quantificada atinjam o valor de apenas 12,49 ps.

Com essa configuração foi gerado o histograma, conforme ilustrado na Figura 32 (a). Este histograma é baseado em 1000 medições obtidas a partir da entrada digital que representa o comando decimal 8000. Para esse valor de entrada, a expectativa teórica é de uma largura de pulso de saída de 99,92 ns.

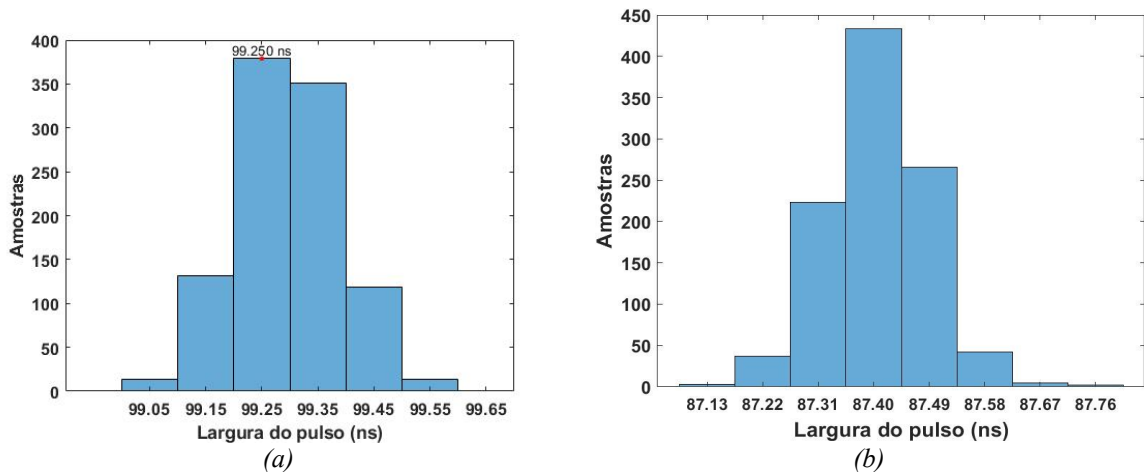


Figura 32: Análise da estimação de Δt , (a) histograma referente a palavra digital de entrada 8000(decimal), (b) histograma referente a palavra digital de entrada 7000(decimal) (Figura elaborada pelo autor).

Com base na Figura 32 (a), para a variação do pior caso, a faixa de tempo total é de 99,05 ns, e o Δt estimado é de cerca de 12,38 ps, o que representa uma diferença de aproximadamente 0,87% do valor esperado de Δt de 12,49 ps. Essa medição também indica uma estabilidade temporal ainda maior em comparação ao sistema configurado para $\Delta t = 990,1$ ps apresentado na seção anterior.

A estimação de Δt utilizando a maior largura de tempo possível, embora seja mais fácil de realizar devido à ampla largura temporal, representa teoricamente o caso mais rigoroso para a avaliação dos efeitos do Jitter acumulado. Nesse cenário, Δt seria avaliado após os sinais A e B completarem 8000 ciclos de oscilação, o que tornaria o acúmulo das variações ao longo dessas oscilações mais evidente. Como contraprova dessa estimativa, realizamos uma nova série de medições (1000 amostras) para o valor de entrada digital correspondente a 7000 em decimal.

Os resultados dessas medições estão apresentados no histograma da Figura 32(b). Nesse caso, a estimativa do Δt médio foi de 12,48 ps, valor muito próximo do ideal (12,49 ps), com uma diferença percentual de apenas 0,32% do pior resultado em relação ao valor ideal de Δt .

Embora seja possível configurar os PLLs para gerar os sinais A e B de modo que, teoricamente, ultrapassem o limite de resolução de 12,49 ps apresentado nesta seção, observou-se que o DTC Vernier proposto enfrenta dificuldades para manter a estabilidade quando o sistema opera com resoluções inferiores a este valor.

A principal hipótese para esse comportamento é a limitação do bloco de detecção de coincidência de fase. Em sistemas com alta resolução, as frequências dos sinais A e B tornam-se muito próximas, o que resulta em tempos de borda de subida também extremamente próximos por vários ciclos consecutivos. Essa proximidade aumenta a probabilidade de ocorrerem problemas de metaestabilidade. Para além disso o nível de *Jitter* médio (13,5 ps) observado na geração dos sinais, demonstra que esse para nossa plataforma FPGA de baixo custo esse é o limite de resolução.

4.4. Discussões

O consumo total de elementos no FPGA é apresentado na *Tabela 4*. O número de elementos lógicos e registradores usados nessa topologia de DTC Vernier representa menos de 2% da capacidade do FPGA (15.408 elementos lógicos). O número de pinos usados representa cerca de 5% do total de 347 pinos, e todos os 4 blocos PLL disponíveis foram utilizados.

Tabela 4: total de consumo de elementos no FPGA

Elementos lógicos	Registradores	Pinos	PLL
222	130	19	4

É importante destacar que o DTC Vernier foi implementado utilizando a linguagem de descrição VHDL, e o número de elementos lógicos empregados é definido pela interpretação da ferramenta de síntese. Em futuras otimizações desse projeto, uma abordagem que foque na simplificação de certas partes da lógica pode resultar em uma maior eficiência no uso de recursos lógicos do FPGA. Especificamente, a adoção de um projeto menos genérico, com resolução fixa, pode reduzir o uso de tipos de dados mais complexos, como vetores com mais de 8 bits, o que, por sua vez, diminuiria a necessidade de elementos lógicos adicionais.

Ainda assim, nossa implementação utiliza uma quantidade de elementos lógicos similar à observada em outros trabalhos de referência no estado da arte para Conversores DTC baseados em FPGA, como pode ser conferido na Tabela 5.

A Tabela 5 compara o desempenho do DTC proposto neste trabalho com trabalhos anteriores. A potência estimada vem da ferramenta de projeto Quartus II. Os valores escolhidos para DNL e INL representam as piores variações positivas e negativas apresentadas nos respectivos gráficos da Figura 28, enquanto o trabalho [16] apresenta apenas os piores valores globais para DNL e INL.

Tabela 5: Desempenho do DTC proposto, comparado com outros trabalhos.

	Este Trabalho	[51]	[49]	[50]	[16]	[31]
Tecnologia	FPGA	FPGA	FPGA	FPGA	FPGA	FPGA
Arquitetura	Vernier com osciladores	Digital clock manager (DCM)	Carry chains	IDELAYE2 primitive	QPSR	Vernier com osciladores
Potência (mW)	89,23	-	1100	-	196	3,04
Resolução (ps)	990,1 ~ 12,5	625 ~ 78	38,6	52	3,93	1,58
DNL (LSB)	-0,41 ~ 0,54	-	-0,18 ~ 0,24	-	0,36	-0,086 ~ 0,75
INL (LSB)	-0,82 ~ 0,17	-	-0,02 ~ 0,01	-	2,61	-0,93 ~ 0,75
Faixa de tempo total	99,92 ns	-	2,1 ns	5 ns	43 s	59,3 min
Elementos lógicos	222	-	668	-	199	422
Registradores Dedicados	98	-	146	-	84	84

(a) O DNL e o INL foram medidos para a implementação com resolução de 990,1 ps.

A comparação dos trabalhos apresenta diferenças significativas no desempenho das arquiteturas de DTC baseadas em FPGA. Nosso trabalho se destaca-se pelo baixo consumo de energia, onde esse consumo é estimado pela ferramenta de software Quartus II, e o reporte da implementação no FPGA que contém os dados estimados de consumo é apresentado no Apêndice F deste documento.

Com apenas 89,23 mW que supera a maioria das alternativas, como por exemplo o trabalho [49], que apresenta um consumo elevado de 1100 mW, e [16], com 196 mW. Somente o trabalho [31] apresenta um consumo menor, de 3,04 mW, o que reflete o uso de uma arquitetura mais simplificada sem um bloco adicional para detecção da coincidência de fase, porém possivelmente tal trabalho não leva em conta o consumo dos blocos PLL.

Em termos de resolução temporal, o DTC proposto atinge valores na faixa de 990,1 ps à 12,5 ps, demonstrando uma ampla capacidade de ajuste. Comparativamente, apenas o trabalho [16] e [31] apresentam uma resolução melhor, atingindo respectivamente 3.93 ps e 1,58 ps,

enquanto os demais trabalhos, como [49] e [50], apresentam resoluções de 38,6 ps e 52 ps, respectivamente. Essa flexibilidade do trabalho proposto posiciona-o como uma solução intermediária entre simplicidade e alta precisão.

A linearidade também foi avaliada em termos de DNL e INL. O trabalho proposto mostra uma variação de -0,41 a 0,54 LSB para o DNL e de -0,82 a 0,17 LSB para o INL, indicando boa estabilidade e precisão, comparável a [49] e [31], que apresentam variações ligeiramente menores. Por outro lado, trabalhos como [16] registraram maiores desvios na linearidade, como o INL de 2,61 LSB.

Adicionalmente, a faixa de tempo total alcançada pelo DTC proposto (99,92 ns) é expressiva e supera a de trabalhos como [49] e [50], que possuem faixas muito menores (2,1 ns e 5 ns, respectivamente). No entanto, trabalhos como [31] destacam-se pela alta capacidade de alcance no tempo da variável de saída, com uma faixa de tempo total de até 59,3 minutos, porém faixas de tempo muito elevadas são aproveitadas em aplicações muito específicas uma vez que tornariam a latência do sistema (período entre cada conversão) muito alta.

Por fim, em termos de recursos utilizados no FPGA, o trabalho proposto consome apenas 222 elementos lógicos e 98 registradores dedicados, o que é eficiente em comparação a [49], que utiliza 668 elementos lógicos, e a [31], que consome 422 elementos lógicos. Esse baixo uso de recursos torna a arquitetura ideal para projetos onde a otimização de espaço no FPGA é uma prioridade.

4.5. Síntese lógica e física usando tecnologia CMOS 350 nm

Para esta síntese, foi aproveitado o código VHDL, usado na implementação para FPGA descrita anteriormente. A síntese RTL do circuito foi realizada utilizando a ferramenta *Cadence Genus* [53], configurada com a tecnologia CMOS de 350 nm e o *Process design kit* (PDK) da AMS. O layout foi projetado no software *Cadence Innovus* [54], também configurado para o mesmo processo CMOS. Diferentemente da implementação em FPGA, a ferramenta de síntese Genus produziu um circuito com um total de 331 elementos lógicos.

A diferença observada decorre da presença, nos FPGAs, de blocos lógicos otimizados, como *Lookup tables* (LUT), multiplexadores, *carry chains*, *flip-flops* e memórias. Esses blocos são pré-fabricados e altamente reutilizáveis, permitindo implementações mais eficientes [52]. Em contraste, a síntese física em ASICs baseia-se em bibliotecas de células padrão, que atendem

a necessidades mais genéricas, e são menos otimizadas em comparação com os blocos lógicos disponíveis nos FPGAs [55].

É importante destacar que a análise não está comparando a área ocupada pelas soluções em FPGA e ASIC. Tal comparação de área entre essas soluções seria inadequada uma vez que em implementações em FPGA, os elementos lógicos, como *Lookup Tables (LUTs)* e *flip-flops*, ocupam significativamente mais área em comparação com elementos lógicos equivalentes em tecnologias CMOS dedicadas. As comparações em trabalhos acadêmicos indicam que circuitos implementados em FPGAs podem consumir até 35 vezes mais área do que em ASICs baseados em células padrão CMOS [56].

Essa diferença substancial deve-se à natureza reconfigurável dos FPGAs, que requerem recursos adicionais para flexibilidade, resultando em maior consumo de área em comparação com implementações CMOS otimizadas para funções específicas.

O resultado da síntese RTL foi aproveitado para se projetar a implementação física. A Figura 33 (a) apresenta o *layout* do circuito, com uma área total de $236 \mu\text{m} \times 189 \mu\text{m}$. O consumo estimado de energia é consideravelmente reduzido levando-se em conta a estimativa de consumo da implementação no FPGA, sendo de apenas 1,75 mW. A Figura 33 (b) mostra o mesmo *layout*, mas sem as camadas de metal que realizam as interligações do sistema deixando à mostra um mapa da distribuição do circuito.

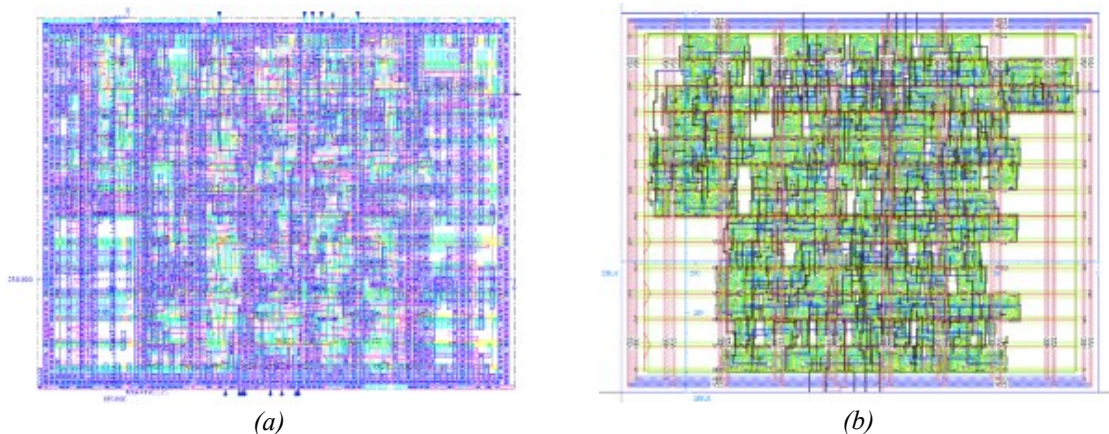


Figura 33: (a) *Layout* com todas as camadas a vista, (b) *layout* sem as camadas de metal para melhor visualização dos elementos lógicos. (Figura elaborada pelo autor).

É importante destacar que, tanto na síntese quanto no *layout*, os circuitos responsáveis pela geração dos sinais *A* e *B* não foram incluídos. Essa omissão evidencia uma das limitações dessa arquitetura: a dependência de uma fonte externa para a geração dos sinais de entrada, do princípio Vernier. Essa necessidade impõe desafios adicionais, pois a fonte deve possuir alta precisão e estabilidade para garantir o correto funcionamento do DTC Vernier. A ausência desses circuitos de geração impacta também as estimativas de consumo de área e energia do

sistema como um todo, uma vez que esses componentes não foram considerados nas análises realizadas.

A escolha e implementação da fonte geradora exigem atenção especial no *layout* do circuito, pois é necessário integrar soluções robustas contra ruídos e *Jitter*. Além disso, o roteamento das linhas de transmissão dos sinais deve ser cuidadosamente planejado para minimizar atrasos, de modo a não comprometer a regularidade entre as relações de fase dos sinais A e B e, conseqüentemente, o funcionamento do DTC Vernier.

5. CONCLUSÕES E PROPOSTAS FUTURAS

Este trabalho apresentou o desenvolvimento e a caracterização de um Conversor Digital-para-Tempo (DTC) utilizando o princípio Vernier com osciladores, implementado em uma FPGA de baixo custo, onde nossa proposta consegue atingir um patamar de alta resolução que se aproxima de trabalhos que representam o estado da arte neste tipo de arquitetura, aliando versatilidade de configuração da resolução do sistema com bons valores de linearidade e de faixa de tempo total, com um ótimo valor de consumo energético considerando DTCs implementados em FPGA.

Neste trabalho inicialmente, introduzimos o conceito de DTC e sua relevância para sistemas de circuitos em modo de tempo. Exploramos as motivações para o uso desses sistemas, destacando o ganho de resolução proporcionado pelo paradigma em modo de tempo, especialmente em projetos baseados em tecnologias avançadas de semicondutores. Nessas tecnologias, os valores reduzidos de atraso intrínseco de propagação viabilizam sistemas com alta capacidade de resolução temporal.

Discutimos as principais métricas de desempenho utilizadas para avaliar DTCs, como resolução, linearidade, faixa de tempo total e consumo energético. As topologias mais comuns de DTC foram analisadas tanto no fluxo analógico quanto no digital. Apesar de os projetos analógicos em ASIC oferecerem as melhores performances, com resoluções na faixa de femtossegundos e baixos consumos de energia e área, é destacado que as arquiteturas digitais também têm vantagens como topologias que empregam circuitos menos complexos quando comparado às alternativas analógicas, bem como o tempo reduzido de desenvolvimento, que é característico de projetos no fluxo digital, em especial é destacado a praticidade de implementação utilizando dispositivos FPGA. Em particular, observamos que as arquiteturas baseadas em linhas de atraso programáveis, muito utilizadas para implementação de DTCs em FPGA, enfrentam desafios relacionados à linearidade e o tamanho do circuito, fato este que torna a topologia Vernier com osciladores uma alternativa capaz de entregar um DTC de alta resolução e boa linearidade com um tamanho de circuito reduzido para implementação em FPGA.

A topologia Vernier foi detalhadamente explorada neste trabalho. Apresentamos descrições teóricas, gráficos temporais e equacionamentos que ilustram o funcionamento, capacidades e limitações de DTCs baseados nesse princípio. Para a implementação no FPGA (placa DE0 com chip Cyclone III), consideramos as restrições dos recursos disponíveis e

configuramos o sistema para atender às taxas de amostragem do osciloscópio utilizado na caracterização.

Um ponto central do princípio Vernier é a geração dos sinais A e B , que devem ter frequências muito próximas. Observamos que a melhor faixa de operação para a geração desses sinais utilizando os blocos PLL disponíveis no FPGA está abaixo de 50 MHz. Além de gerar os sinais é necessário que estes tenham uma ótima regularidade de fase entre si, para tal é também apresentado um arranjo que permite que os sinais gerados sejam todos síncronos entre si.

A arquitetura desenvolvida inclui a geração dos sinais A e B , o bloco de detecção de coincidência de fases e os contadores responsáveis por amostrar os ciclos dos sinais usados no princípio Vernier e gerar os pulsos de saída. O bloco de detecção de coincidência de fase é um diferencial de nosso trabalho, utilizando uma estratégia de detecção simplificada em termos de consumo de componentes, é baseado em um *flip-flop* tipo D, mas que emprega também estratégias de expansão de pulsos para melhorar a precisão e mitigar problemas de metaestabilidade.

Na análise do Jitter na faixa de 10 MHz, que é a faixa que foi utilizada para a geração dos sinais A e B , é identificado um valor médio de desvio de 13 ps em relação ao período ideal, com um valor de desvio padrão de 52 ps. Esse valor médio de desvios tende a definir um provável limite prático para a resolução do DTC na plataforma FPGA utilizada, uma vez que foi observado instabilidade ao tentar implementar um sistema com resolução abaixo de 12,5 ps.

Para a caracterização, foram implementadas duas versões do sistema: uma com resolução de 990 ps, usada para medições de não linearidades, e outra com resolução estimada de 12,5 ps. A versão de maior resolução foi utilizada para se efetuar as medições de não linearidades uma vez que tal implementação se encaixa nos parâmetros de taxa de amostragem do osciloscópio disponível. A resolução do sistema de 12,5 ps é estimada de forma indireta utilizando a Equação (24) e sua variabilidade é estimada através de histogramas montados com 1000 medições da faixa de tempo total, onde é esperado o maior efeito das variações acumuladas, nessa análise é constatado que a diferença percentual para o valor ideal é inferior à 1% fato este que comprova a exatidão e precisão do DTC Vernier.

Ao comparar nossa implementação com outros trabalhos em FPGA, observamos vantagens em termos de consumo energético (89 mW) e utilização de elementos lógicos (220). Embora nossa resolução de 12,5 ps seja inferior à alguns trabalhos do estado da arte, estes geralmente utilizam FPGAs mais avançados e de custo elevado, o que torna nossa proposta adequada para implementar DTC de alta resolução em plataformas FPGA de baixo custo. A análise das não linearidades revelou valores de DNL entre -0,41 e 0,54 LSB, e de INL entre -

0,82 e 0,17 LSB, com uma faixa de tempo de 99,92 ns. Esses resultados de linearidade superam alguns trabalhos, enquanto a faixa de tempo se mostra intermediária entre os trabalhos analisados indicando um ótimo custo-benefício de nossa proposta.

Por fim, realizamos a síntese lógica e física da descrição em VHDL do DTC Vernier para a tecnologia CMOS 350 nm (AMS). O núcleo do DTC utilizou 331 elementos lógicos e apresentou uma área total de $236 \mu\text{m} \times 189 \mu\text{m}$, com consumo energético estimado de 1,75 mW. É importante destacar que a geração dos sinais A e B não foi incluída nesta síntese física, ainda assim a síntese física de nossa versão do DTC Vernier tem um potencial de .

Os resultados apresentados demonstram a viabilidade do uso de FPGAs de baixo custo para implementar DTCs de alta resolução. Nosso trabalho contribui com uma arquitetura eficiente de DTC Vernier para aplicações em sistemas em modo de tempo, utilizando fluxo de projeto digital e plataforma FPGA .

5.1. Propostas Futuras

Futuras atualizações deste sistema podem explorar diferentes abordagens para expandir sua faixa de tempo total. Uma possibilidade promissora é a implementação de técnicas que utilizem contagem de oscilações adicionais dos ciclos de um dos sinais A e B , de maneira a aproveitar melhor o intervalo de tempo do período T_{min} . Essa abordagem tem o potencial de ampliar significativamente a faixa de tempo total do DTC Vernier, aproveitando melhor os recursos já existentes no sistema e ajustando o funcionamento dos sinais gerados.

Outro exemplo de linha de investigação consiste na integração do princípio Vernier com osciladores em combinação com linhas de atraso controladas digitalmente. Nesta arquitetura híbrida, a saída do DTC Vernier seria conectada a uma linha de atraso programável, que permitiria refinar ainda mais a resolução do sistema. Essa combinação pode potencialmente mitigar limitações de não linearidades observadas em arquiteturas puramente baseadas em longas linhas de atraso. Além disso, tal abordagem tem o potencial de melhorar a resolução do sistema para valores ainda mais próximos de DTCs de grande performance como são por exemplo os implementados em ASIC, mas mantendo-se dentro das limitações de plataformas FPGA de baixo custo.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ROBERTS, Gordon W.; ALI-BAKHSHIAN, Mohammad. A brief introduction to time-to-digital and digital-to-time converters. *IEEE Transactions on Circuits and Systems II: Express Briefs*, v. 57, n. 3, p. 153-157, 2010.
- [2] RAVINUTHULA, Vishnu et al. Time - mode circuits for analog computation. *International Journal of Circuit Theory and Applications*, v. 37, n. 5, p. 631-659, 2009.
- [3] RAVINUTHULA, Vishnu et al. Time - mode circuits for analog computation. *International Journal of Circuit Theory and Applications*, v. 37, n. 5, p. 631-659, 2009.
- [4] ALIOTO, Massimo; PALUMBO, Gaetano. Model and design of bipolar and MOS current-mode logic: CML, ECL and SCL digital circuits. Springer Science & Business Media, 2006.
- [5] OULMANE, Mourad; ROBERTS, Gordon W. A CMOS time amplifier for femto-second resolution timing measurement. In: 2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No. 04CH37512). IEEE, 2004. p. I-509.
- [6] KUHN, Kelin J. Considerations for ultimate CMOS scaling. *IEEE transactions on Electron Devices*, v. 59, n. 7, p. 1813-1828, 2012.
- [7] YUAN, Fei (Ed.). CMOS time-mode circuits and systems: fundamentals and applications. CRC Press, 2018.
- [8] PANDIT, Soumya; MANDAL, Chittaranjan; PATRA, Amit. Nano-scale CMOS analog circuits: models and CAD techniques for high-level design. CRC Press, 2018.
- [9] M. Ali-Bakhshian, Digital processing of analog information adopting time-mode signal processing, Ph.D. dissertation, McGill University, 2013.
- [10] LOCATELLI, Pedro Sartori; COLOMBO, Dalton Martini; EL-SANKARY, Kamal. Time-domain multiply-accumulate unit. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, v. 31, n. 6, p. 762-775, 2023.
- [11] ALI-BAKHSHIAN, Mohammad. Digital processing of analog information adopting time-mode signal processing. 2013.
- [12] JUNG, Dong-Hoon et al. All-digital process-variation-calibrated timing generator for ATE with 1.95-ps resolution and maximum 1.2-GHz test rate. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, v. 26, n. 6, p. 1015-1025, 2018.
- [13] GOVINDARAJ, Vivek et al. Dtc-assisted all-digital phase-locked loop exploiting hybrid time/voltage phase digitization. In: 2019 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS). IEEE, 2019. p. 81-84.

- [14] LAI, Chang-Ming et al. A UWB impulse-radio timed-array radar with time-shifted direct-sampling architecture in 0.18- μ m CMOS. *IEEE Transactions on Circuits and Systems I: Regular Papers*, v. 61, n. 7, p. 2074-2087, 2014.
- [15] DANG, Haoyang et al. Review of FPGA-Based Digital-to-Time Converters. In: 2023 IEEE 16th International Conference on Electronic Measurement & Instruments (ICEMI). IEEE, 2023. p. 243-252.
- [16] ZHANG, Min; WANG, Hai; LIU, Yan. Digital-to-time converter with 3.93 ps resolution implemented on FPGA chips. *IEEE Access*, v. 5, p. 6842-6848, 2017.
- [17] KUON, Ian; ROSE, Jonathan. Measuring the gap between FPGAs and ASICs. In: *Proceedings of the 2006 ACM/SIGDA 14th international symposium on Field programmable gate arrays*. 2006. p. 21-30.
- [18] ELMALLAH, Ahmed et al. A 1.6 ps peak-INL 5.3 ns range two-step digital-to-time converter in 65nm CMOS. In: 2018 IEEE Custom Integrated Circuits Conference (CICC). IEEE, 2018. p. 1-4.
- [19] BALESTRIERI, E. et al. DAC static parameter specifications—some critical notes. In: *Proc. of 10th IMEKO TC4 Workshop on ADC Modelling and Testing*, Gdynia and Jurata, Poland. 2005. p. 81-86.
- [20] CIRCUITS—PART, Semiconductor Devices—Integrated. 4: Interface Integrated Circuits. *IEC*, v. 60, p. 748-4, 1997.
- [21] CONG, Yonghua; GEIGER, Randall L. Formulation of INL and DNL yield estimation in current-steering D/A converters. In: 2002 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2002. p. III-III.
- [22] MAXIM, Dallas Semiconductors. INL/DNL Measurements for High-Speed Analog-to-Digital Converters (ADCs). *APPLICATION NOTE 283*, Sep, 2000.
- [23] Low-Power IC Design: Techniques and Best Practices. Disponível em: <<https://www.ansys.com/blog/low-power-ic-design-techniques-best-practices>> Acessado em 14 jun. 2024.
- [24] CHEN, Peng et al. A 31- μ W, 148-fs Step, 9-bit Capacitor-DAC-Based Constant-Slope Digital-to-Time Converter in 28-nm CMOS. *IEEE Journal of Solid-State Circuits*, v. 54, n. 11, p. 3075-3085, 2019.
- [25] RU, Jiayoon Zhiyu et al. A high-linearity digital-to-time converter technique: Constant-slope charging. *IEEE journal of solid-state circuits*, v. 50, n. 6, p. 1412-1423, 2015.
- [26] HASSAN, Ali H. et al. A 0.002 - mm² 8 - bit 1 - MS/s low - power time - based DAC (T - DAC). *IET Circuits, Devices & Systems*, v. 15, n. 8, p. 738-744, 2021.
- [27] JACOB, Nisha et al. Securing FPGA SoC configurations independent of their manufacturers. In: 2017 30th IEEE International System-on-Chip Conference (SOCC). IEEE, 2017. p. 114-119.

- [28] CORNA, N. et al. Programmable Delay-Line with High-Resolution Time Steps Implemented in a Digital-to-Time Converter IP-Core for FPGAs and SoCs. In: 2020 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC). IEEE, 2020. p. 1-3.
- [29] CORNA, N. et al. Multi-channel high-resolution digital-to-time pattern generator ip-core for fpgas and socs. In: 2021 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC). IEEE, 2021. p. 1-4.
- [30] WANG, Hai; ZHANG, Min; LIU, Yan. High-resolution digital-to-time converter implemented in an FPGA chip. Applied Sciences, v. 7, n. 1, p. 52, 2017.
- [31] CHEN, Poki et al. FPGA vernier digital-to-time converter with 1.58 ps resolution and 59.3 minutes operation range. IEEE Transactions on Circuits and Systems I: Regular Papers, v. 57, n. 6, p. 1134-1142, 2009.
- [32] KWAN, Alistair. Vernier scales and other early devices for precise measurement. American Journal of Physics, v. 79, n. 4, p. 368-373, 2011.
- [33] ZHOU, Wei. The greatest common factor frequency and its application in the accurate measurement of periodic signals. In: Proceedings of the 1992 IEEE frequency control symposium. IEEE, 1992. p. 270-273.
- [34] BOARD, Altera DE. Version 1.00 Copyright© 2009 Terasic Technologies.
- [35] INFINIIVISION, Agilent. X-Series Oscilloscopes, ; 2011. 2000.
- [36] SEMERIA, Marcelo. Los tres teoremas: Fourier-nyquist-shannon. Serie Documentos de Trabajo, 2015.
- [37] ALTERA. Datasheet: Cyclone-III Device Handbook. EP3C16F484C6N plataform FPGA User Guide. Document Publication, 2012.
- [38] ALTERA. ALTPLL (Phase-Locked Loop) IP Core User Guide, Altera Corporation, Document Publication 2017.
- [39] INSTRUMENTS, N. Synchronous communications and timing configurations in digital devices. 2013.
- [40] SHU, Keliu; SÁNCHEZ-SINENCIO, Edgar. CMOS PLL synthesizers: analysis and design. Springer Science & Business Media, 2006.
- [41] LI, Wuxi et al. Simultaneous placement and clock tree construction for modern FPGAs. In: Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. 2019. p. 132-141.
- [42] CHEN, Doris et al. A comprehensive approach to modeling, characterizing and optimizing for metastability in FPGAs. In: Proceedings of the 18th annual ACM/SIGDA international symposium on Field programmable gate arrays. 2010. p. 167-176.

- [43] HANCOCK, Johnnie et al. Jitter—understanding it, measuring it, eliminating it part 1: Jitter fundamentals. *High Frequency Electronics*, v. 4, n. 4, p. 44-50, 2004.
- [44] NOTE, Tektronix Application. *Understanding and Characterizing Timing Jitter*. 2007.
- [45] JENKINS, J. *Specifying a PLL Part 2: Jitter Basics*. Australia: Perceptia Devices Australia Pty Ltd, 1 jan. 2020.
- [46] SCHWARZ, Douglas. *Fast and robust curve intersections*. MATLAB Central File Exchange. Available online: <https://www.mathworks.com/matlabcentral/fileexchange/11837-fast-and-robust-curve-intersections>, 2010.
- [47] TILDEN, Steven J.; LINNENBRINK, Thomas E.; GREEN, Philip J. *Overview of IEEE-STD-1241" standard for terminology and test methods for analog-to-digital converters"*. In: *IMTC/99. Proceedings of the 16th IEEE Instrumentation and Measurement Technology Conference (Cat. No. 99CH36309)*. IEEE, 1999. p. 1498-1503.
- [48] *Understanding Data Converters 1995 Mixed-Signal Products Application Report SLAA013*. [s.l: s.n.]. Disponível em: <<https://www.ti.com/lit/an/slaa013/slaa013.pdf>>. Acesso em: 4 jun. 2024.
- [49] CUI, Ke; LI, Xiangyu; ZHU, Rihong. *A high-resolution programmable Vernier delay generator based on carry chains in FPGA*. *Review of Scientific Instruments*, v. 88, n. 6, 2017.
- [50] CORNA, N. et al. *Multi-channel high-resolution digital-to-time pattern generator ip-core for fpgas and socs*. In: *2021 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*. IEEE, 2021. p. 1-4.
- [51] NAVARRO, Denis et al. *Synchronous FPGA-based high-resolution implementations of digital pulse-width modulators*. *IEEE transactions on power electronics*, v. 27, n. 5, p. 2515-2525, 2011.
- [52] SINGH, Satwant et al. *The effect of logic block architecture on FPGA performance*. *IEEE Journal of Solid-State Circuits*, v. 27, n. 3, p. 281-287, 1992.
- [53] *Genus User Guide for Legacy UI*. [s.l: s.n.]. Disponível em: <<https://picture.iczhiku.com/resource/eetop/wYIDduQHAKlkSxcN.pdf>>. Acesso em 26 dez. 2024.
- [54] *Innovus User Guide*. [s.l: s.n.]. Disponível em: <<https://iccircle.com/static/upload/img20240414154902.pdf>>. Acesso em: 26 dez. 2024.
- [55] BHATNAGAR, Himanshu. *Advanced ASIC chip synthesis*. Springer Science & Business Media, 2002.
- [56] WONG, Henry; BETZ, Vaughn; ROSE, Jonathan. *Comparing FPGA vs. custom CMOS and the impact on processor microarchitecture*. In: *Proceedings of the 19th ACM/SIGDA international symposium on Field programmable gate arrays*. 2011. p. 5-14.

APÊNDICE A – DTC Vernier Top-Level

```

1  --Autor: Vilmondes Ribeiro
2
3  library ieee;
4  use ieee.std_logic_1164.all;
5  use ieee.numeric_std.all;
6  --use ieee.math_real.all;
7
8  entity dtc4 is
9  port (
10 clk : in std_logic;
11 reset : in std_logic;
12 --i_osc1 : in std_logic; --PLL A
13 --i_osc2 : in std_logic; --PLL B
14 i_dado : in std_logic_vector(9 downto 0);
15 o_start : out std_logic; -- sinal de saida start
16 o_stop : out std_logic; -- sinal de saida stop
17 o_pulso_saida : out std_logic; --inapropriado pela baixa taxa de subida
18
19 o_phase : out std_logic; --apenas para medição
20
21 o_oscA : out std_logic; --apenas para medição
22 o_oscB : out std_logic; --apenas para medição
23 o_fast_clk : out std_logic --apenas para medição
24 );
25 end entity;
26
27 architecture rtl of dtc4 is
28 --Componente declaration
29 -----
30 component edge_detec is
31 port (
32 clk : in std_logic; --i
33 entrada : in std_logic;
34 reset : in std_logic; --i
35 saida : out std_logic
36 );
37 end component;
38 -----
39 component phase_detec is
40 port (
41 clk : in std_logic; --i
42 reset : in std_logic; --i
43 i_osc1 : in std_logic; --i
44 i_osc2 : in std_logic; --i
45 o_saida : out std_logic --s
46 );
47 end component;
48 -----
49 component fase_detec is
50 port (
51 clk : in std_logic;
52 osc1 : in std_logic;
53 osc2 : in std_logic;
54 reset : in std_logic;
55 saida : out std_logic
56 );
57 end component;
58 -----
59 component contador_start is
60 port (
61 clk : in std_logic; --i
62 reset : in std_logic; --i
63 i_clk : in std_logic; --i
64 bcd : out std_logic_vector (7 downto 0); --s
65 i_dado : in std_logic_vector(9 downto 0); --i
66 i_inicio_cont : in std_logic; --s --s_phase_coincidence

```

```

67 o_final_cont : out std_logic; --o --sinais de start e stop
68 o_cont : out std_logic_vector(9 downto 0) --s
69 );
70 end component;
71 -----
72 component pll1 is
73 port (
74 areset : IN STD_LOGIC := '0';
75 inclk0 : IN STD_LOGIC := '0';
76 c0 : OUT STD_LOGIC
77 );
78 end component;
79 -----
80 component pll2 is
81 port (
82 areset : IN STD_LOGIC := '0';
83 inclk0 : IN STD_LOGIC := '0';
84 c0 : OUT STD_LOGIC
85 );
86 end component;
87 -----
88 component pll3 is
89 port (
90 areset : IN STD_LOGIC := '0';
91 inclk0 : IN STD_LOGIC := '0';
92 c0 : OUT STD_LOGIC
93 );
94 end component;
95 -----
96 component pll4 is
97 port (
98 areset : IN STD_LOGIC := '0';
99 inclk0 : IN STD_LOGIC := '0';
100 c0 : OUT STD_LOGIC
101 );
102 end component;
103 -----
104 --signal instantiation
105 signal s_reset : std_logic := '0';
106 signal s_novo_clk : std_logic := '0';
107 signal s_phase_coincidence : std_logic := '0';
108 signal s_pulso : std_logic := '0';
109 signal s_bcd1 : std_logic_vector (7 downto 0) := (others => '0'); --s
110 signal s_bcd2 : std_logic_vector (7 downto 0) := (others => '0'); --s
111 signal s_dado : std_logic_vector (9 downto 0) := (others => '0'); --s
112 signal s_start : std_logic := '0';
113 signal s_stop : std_logic := '0';
114 signal s_oscA : std_logic := '0';
115 signal s_oscB : std_logic := '0';
116 signal s_fast_clk : std_logic := '0';
117 signal s_cont1 : std_logic_vector (9 downto 0) := (others => '0'); --s
118 signal s_cont2 : std_logic_vector (9 downto 0) := (others => '0'); --s
119 -----
120
121 begin
122
123 s_reset <= not reset;
124
125 inst_pll1: pll1 port map (s_reset, s_novo_clk, s_fast_clk); --clk sys
126 inst_pll2: pll2 port map (s_reset, s_novo_clk, s_oscA); --Fa --voltar s_novo_clk
para clk
127 inst_pll3: pll3 port map (s_reset, s_novo_clk, s_oscB); --Fb --voltar s_novo_clk
para clk
128 inst_pll4: pll4 port map (s_reset, clk, s_novo_clk);
129
130 inst_fase_detec: fase_detec port map (s_fast_clk, s_oscA, s_oscB, s_reset,
s_phase_coincidence);
131

```

```
132 inst_cont_start: contador_start port map (s_fast_clk, s_reset, s_oscA, s_bcd1,
s_dado, s_phase_coincidence, s_start, s_cont1);
133 inst_cont_stop: contador_start port map (s_fast_clk, s_reset, s_oscB, s_bcd2,
s_dado, s_phase_coincidence, s_stop, s_cont2);
134
135 o_oscA <= s_oscA;
136 o_oscB <= s_oscB;
137 o_fast_clk <= s_fast_clk;
138 o_start <= s_start;
139 o_stop <= s_stop;
140 s_dado <= i_dado;
141
142 o_phase <= s_phase_coincidence;
143
144 process (s_start, s_stop)
145 begin
146 if s_start = '1' then
147 s_pulso <= '1';
148 elsif s_stop = '1' then
149 s_pulso <= '0';
150 else
151 null;
152 end if;
153 end process;
154
155 o_pulso_saida <= not s_pulso;
156
157 end architecture;
```

APÊNDICE B – Detector de borda de subida assíncrono

```

1  --Autor: Vilmondes
2
3  library ieee;
4  use ieee.std_logic_1164.all;
5  use ieee.numeric_std.all;
6
7  entity a_edge_detec is
8  port (
9  reset : in std_logic;
10 clk : in std_logic;
11 entrada : in std_logic;
12 saida : out std_logic --pulso de detecção
13 );
14 end entity;
15
16 architecture rtl of a_edge_detec is
17
18 signal s_cont : unsigned(22 downto 0) := (others => '0');
19 signal s_a, s_saida : std_logic := '0';
20
21 begin
22
23 -----
24 process (clk, entrada)
25 begin
26 if entrada = '0' then
27 s_cont <= (others => '0');
28 elsif rising_edge(clk) then
29 s_cont <= s_cont+1;
30 end if;
31 -----
32 --AQUI SE CONTROLA A LARGURA DO PULSO
33 if s_cont = "10" then -- NÃO usar valores 1111, 11, 111
34 s_a <= '1'; --s_a indica o fim do pulso de detecção
35 else
36 s_a <= '0';
37 end if;
38 -----
39 if s_cont = "00" and entrada = '1' then
40 s_saida <= '1';
41 elsif s_a = '1' then
42 s_saida <= '0';
43 else
44 null;
45 end if;
46 end process;
47 -----
48
49 --divisor de clock
50 --p_divisor: process (clk, reset)
51 --begin
52 --if reset = '1' then
53 --cont_div <= (others => '0');
54 --elsif rising_edge(clk) then
55 --cont_div <= cont_div +1;
56 --end if;
57 --end process;

```

```
58 --clk_out <= clk;  
59 --clk_div <= cont_div(3);  
60  
61 saida <= s_saida;  
62  
63 end architecture;
```

APÊNDICE C – Detector de coincidência de fase

```

1  --Detector de fase
2
3  library ieee;
4  use ieee.std_logic_1164.all;
5  use ieee.numeric_std.all;
6
7  entity fase_detec is
8  port (
9  clk : in std_logic;
10 osc1 : in std_logic;
11 osc2 : in std_logic;
12 reset : in std_logic;
13 saida : out std_logic
14 );
15 end entity;
16
17 architecture rtl of fase_detec is
18 -----
19 signal borda_dsubida_oscl : std_logic := '0';
20 signal borda_dsubida_osc2 : std_logic := '0';
21 signal s_saida : std_logic := '0';
22 -----
23 --detector de borda de subida
24 component edge_detec is
25 port (
26 clk : in std_logic;
27 entrada : in std_logic;
28 reset : in std_logic;
29 saida : out std_logic
30 );
31 end component;
32 -----
33 --detector de borda de subida ASSINCRONO!!
34 component A_edge_detec is
35 port (
36 reset : in std_logic;
37 clk : in std_logic;
38 entrada : in std_logic;
39 saida : out std_logic --pulso de detecção
40 );
41 end component;
42 -----
43 --detector de borda de subida ASSINCRONO!!
44 component a_edge_detec_down is
45 port (
46 reset : in std_logic;
47 clk : in std_logic;
48 entrada : in std_logic;
49 saida : out std_logic --pulso de detecção
50 );
51 end component;
52 -----
53
54 begin
55
56 instancial : a_edge_detec port map (reset, clk, osc1, borda_dsu-
bida_oscl);

```

```
57 instancia2 : a_edge_detec port map (reset, clk, osc2, borda_dsu-
bida_osc2);
58
59 inst_saida : a_edge_detec port map (reset, clk, s_saida, saida);
60
61 detc_fase: process (borda_dsubida_osc1, borda_dsubida_osc2, reset)
62 begin
63 if reset = '1' then
64 s_saida <= '0';
65 elsif (borda_dsubida_osc1 and borda_dsubida_osc2) = '1' then
66 if borda_dsubida_osc2 = '1' then
67 s_saida <= '1';
68 else
69 s_saida <= '0';
70 end if;
71 end if ;
72 end process;
75
76 end architecture;
```

APÊNDICE D – Contador Vernier

```

1  --Autor: vilmondes ribeiro
2
3  library ieee;
4  use ieee.std_logic_1164.all;
5  use ieee.numeric_std.all;
6
7  entity contador_start is
8  port (
9  clk : in std_logic;
10 reset : in std_logic;
11 i_clk : in std_logic;
12 bcd : out std_logic_vector (7 downto 0);
13 i_dado : in std_logic_vector(9 downto 0);
14 i_inicio_cont : in std_logic;
15 o_final_cont : out std_logic;
16 o_cont : out std_logic_vector(9 downto 0)
17 );
18 );
19 end entity;
20
21
22 architecture rtl of contador_start is
23 --Componente declaration
24 -----
25 component edge_detec is
26 port (
27 clk : in std_logic;
28 entrada : in std_logic;
29 reset : in std_logic;
30 saida : out std_logic
31 );
32 end component;
33 -----
34 --detector de borda de subida ASSINCRONO!!
35 component A_edge_detec is
36 port (
37 reset : in std_logic;
38 clk : in std_logic;
39 entrada : in std_logic;
40 saida : out std_logic --pulso de detecção
41 );
42 end component;
43 -----
44 --gerador de pulso (usado no start e stop)
45 component geradorPulso is
46 port (
47 reset : in std_logic;
48 clk : in std_logic;
49 entrada : in std_logic;
50 saida : out std_logic
51 );
52 end component;
53 -----
54
55 --signal instantiation
56 signal s_borda : std_logic := '0';
57 signal s_enable : std_logic := '0';
58 signal s_inicio_cont : std_logic := '0';
59 signal s_limite : std_logic := '0';
60 signal s_cont : unsigned (9 downto 0) := (others => '0') ;
61 signal s_dado : unsigned (9 downto 0) := (others => '0') ;
62 signal s_final_cont : std_logic := '0';
63
64
65 begin

```

```

66 s_inicio_cont <= i_inicio_cont;
67 o_final_cont <= s_final_cont;
68
69 int_edge_detec: a_edge_detec port map (reset, clk, i_clk, s_borda);
70
71 int_pulso_finalconta: geradorPulso port map (reset, clk, s_limite, s_final_cont);
72
73 --enable system
74 sinal_enable: process (s_inicio_cont, s_final_cont )
75 begin
76 if s_inicio_cont = '1' then
77 s_enable <= '1';
78 s_dado <= unsigned(i_dado);
79 elsif s_final_cont = '1' then
80 s_enable <= '0';
81 else
82 null;
83 end if;
84 end process;
85
86
87 --counter
88 -----
89 contador: process (s_borda, reset, s_enable, s_dado)
90 begin
91 if (reset = '1') or (s_enable = '0') then
92 s_cont <= (others => '0');
93 elsif rising_edge(s_borda) then
94 s_cont <= s_cont + 1;
95 s_limite <= '0';
96 if s_cont = s_dado then
97 s_cont <= (others => '0');
98 s_limite <= '1';
99 end if ;
100 end if ;
101
102 end process;
103 o_cont <= std_logic_vector(s_cont);
104 -----105
106 --binary to 7seg converter
107 BCD_conversao: process (s_cont)
108 begin
109 bcd <= "11111111";
110 case s_cont is
111 when "000000000" => bcd(1)<='0';
bcd(2)<='0';bcd(3)<='0';bcd(4)<='0';bcd(5)<='0';bcd(6)<='0'; --0
112 when "000000001" => bcd(1)<='0';bcd(2)<='0'; --1
113 when "000000010" =>
bcd(0)<='0';bcd(1)<='0';bcd(4)<='0';bcd(3)<='0';bcd(6)<='0'; --2
114 when "000000011" =>
bcd(0)<='0';bcd(1)<='0';bcd(2)<='0';bcd(3)<='0';bcd(6)<='0'; --3
115 when "000000100" => bcd(1)<='0';bcd(2)<='0';bcd(5)<='0';bcd(6)<='0'; --4
116 when "000000101" =>
bcd(0)<='0';bcd(2)<='0';bcd(3)<='0';bcd(5)<='0';bcd(6)<='0'; --5
117 when "000000110" =>
bcd(0)<='0';bcd(2)<='0';bcd(3)<='0';bcd(4)<='0';bcd(5)<='0';bcd(6)<='0'; --6
118 when "000000111" => bcd(0)<='0';bcd(1)<='0';bcd(2)<='0'; --7
119 when "000001000" => bcd(1)<='0';
bcd(2)<='0';bcd(3)<='0';bcd(4)<='0';bcd(5)<='0';bcd(6)<='0'; --8
120 when "000001001" => bcd(0)<='0';
bcd(1)<='0';bcd(2)<='0';bcd(5)<='0';bcd(6)<='0'; --9
121 when others =>
122 null;
123 end case;
124 end process;
132 end architecture;

```

APÊNDICE E – Código Matlab para a leitura dos arquivos CSV das medições feitas no osciloscópio

```

1 %Autor: Vilmondes Ribeiro
2
3 clear;
4 clc;
5 format shortEng
6 disp('INÍCIO DO SCRIPT!');
7 %-----
8 %DETERMINA A QUANTIDADE DE ARQUIVOS .CSV QUE O DIRETÓRIO CONTÉM
9 %lista a quantidade de arquivos .csv
10 directorio = dir;
11 qtdFileCSV = 0;
12 disp('Conta qtd de arquivos csv...');
13 for i=1:length(diretorio)
14 if strfind(diretorio(i).name, 'scope_')==1
15 qtdFileCSV = qtdFileCSV+1 ;
16 end
17 end
18 %-----
19 %VETOR COM OS NOMES DOS ARQUIVOS
20 fileNames = []; %Formando um vetor de nomes para a leitura dos arquivos
21 disp('Pega os nomes dos arquivos csv...');
22 for i=0:qtdFileCSV
23 if i<=qtdFileCSV
24 fileName = ['scope_'+string(i)+'.CSV'];
25 fileNames = [fileNames fileName];
26 end
27 end
28 %-----
29 %CALCULO DA RESOLUÇÃO (deltaT) DE ACORDO COM Fa, Fb, TMIN,
30 disp('Calculos...');
31 fa = 10e6;
32 fb = 10.1e6 %153ps
33 deltat = ((1/fa)-(1/fb))
34 fmax = deltat*fa*fb
35 tmin = 1/fmax
36 K = fa/fmax
37 M = fb/fmax
38
39 %-----
40 %LAÇO FOR PARA:
41 %LEITURA DOS ARQUIVOS .CSV DO DIRETÓRIO
42 %PLOTAGEM DAS FORMAS DE ONDA
43 %CALCULO DA COORDENADA X DA INTERSEÇÃO DA FORMA DE ONDA COM O PTO %MÉDIO
44
45 diferencas = [];
46 %leitura dos arquivos e separação dos dados
47 disp('leitura dos arquivos...');
48 for i=1:qtdFileCSV
49 %printf(" Lendo o arquivo .csv: \n");
50 arquivo = dlmread(fileNames(i), ',', 2, 0 );
51 %printf("\n Dados de tempo (segundos): \n");
52 tempo = arquivo([1:end], 1);
53 %printf("\n Dados do sinal de Start (tensão): \n");
54 signal_start = arquivo([1:end], 2);
55 %printf("\n Dados do sinal de Stop (tensão): \n");
56 signal_stop = arquivo([1:end], 3);
57 %-----
58 end
59 %LINHA USADA PARA A INTERSEÇÃO DO PTO DE LEITURA DOS SINAIS START E STOP
60 referencia = 1.8; %Linha mostrando o ponto médio do sinal de start e stop
61 disp('Cria a linha média...');
62 for i = 1:length(tempo)
63 if i<=length(tempo)
64 linha(i) = referencia;

```

```

65 end
66 end
67 %-----
68 for i=1:qtdFileCSV
69 %printf(" Lendo o arquivo .csv: \n");
70 arquivo = dlmread(fileNames(i), ',', 2, 0 );
71 %printf("\n Dados de tempo (segundos): \n");
72 tempo = arquivo([1:end], 1);
73 %printf("\n Dados do sinal de Start (tensão): \n");
74 signal_start = arquivo([1:end], 2);
75 %printf("\n Dados do sinal de Stop (tensão): \n");
76 signal_stop = arquivo([1:end], 3);
77 pto2 = intersections(tempo, signal_start, tempo, linha);
78 pto = intersections(tempo, signal_stop, tempo, linha);
79 if length(pto)==0 || length(pto2)==0
80 diferencas = [diferencas 0];
81 else
82 diferenca = (pto2(1)-pto(1));
83 diferencas = [diferencas diferenca];
84 end
85 %-----
86 % DESCOMENTE ESSE BLOCO PARA PLOTAR O GRAFICO DE TODOS OS ARQUIVOS .CSV
87 % if i==100
88 % figure(i)
89 % disp("plotando medição: "+i);
90 % hold on
91 % plot(tempo*1e9, signal_start, '-', tempo*1e9, linha, '-');
92 % plot(tempo*1e9, signal_stop, '-');
93 % plot(pto(1)*1e9, referencia, 'ro', 'markersize', 3)
94 % plot(pto2(1)*1e9, referencia, 'ro', 'markersize', 3)
95 % str = 'Tout = ' + string(diferencas(i)*-1e9) + ' ns';
96 % title(str);
97 % xlabel('Tempo (ns)', 'FontSize', 16);
98 % ylabel('Amplitude (V)', 'FontSize', 16);
99 % legend({'Start', 'Linha média', 'Stop'});
100 % ax=gca;
101 % ax.FontSize = 16; % Set the font sizehold off
102 % hold off
103 % end
104 %-----
105 end
106 diferencas = diferencas*-1;
107 %-----
108 %cálculo do DNL e INL
109 DNL = [];
110 INL = [];
111 INL2 = [];
112 passoIdeal = [];
113 %passoIdeal = 1.739e-10;
114 comandos = [];
115 disp('Calculo do DNL e INL...');
116 for i=1:(length(diferencas)-1)
117 comandos = [comandos i];
118 passoIdeal = [passoIdeal (comandos(i)*deltat)];
119 %esse if serve para desconsiderar a primeira diferença do DNL
120 if i<1
121 DNL(i) = 0;
122 else
123 %DNL
124 DNL = [DNL ((diferencas(i+1) - diferencas(i) )/(deltat))-1];
125 end;
126
127 INL = [INL sum(DNL(1:i))];
128
129 end
130 %-----
131 %Linha mostrando o limite LSB+ e LSB-
132 lsb = deltat; %DNL e INL já estão em relação ao LSB

```

```

133 lsb = 1;
134 lsbPlus = [];
135 lsbMinus = [];
136 for i = 1:length(DNL)
137 lsbPlus(i) = lsb;
138 lsbMinus(i) = lsb*(-1);
139 end
140 %-----
141 % INL
142 figure(qtdFileCSV+1)
143 hold on
144 %plot(comandos, lsbPlus, '-', comandos, lsbMinus, '-'); %linas limite do LSB
145 plot(comandos, lsbPlus, '--','LineWidth', 4); %linas limite do LSB
146 plot(comandos, lsbMinus, '--','LineWidth', 4); %linas limite do LSB
147 % plot(comandos, DNL,'LineWidth', 2); disp('Plot DNL');
148 plot(comandos, INL,'LineWidth', 4); disp('Plot INL');
149 % str3 = 'Não Linearidades Diferencial e Integral (DNL e INL)';
150 str3 = '';
151 % str4 = '1 LSB = ' + string(deltat) + ' segundos';
152 str4 = ' (LSB = 990.1 ps)';
153 title(str3);
154 %subtitle(str4);
155
156 ylim([-1.3 1.3]);
157 legend({'+LSB','-LSB', 'INL'});
158 ylabel(str4,'FontSize', 24);
159 xlabel('Input word (n)','FontSize', 24);
160 ax=gca;
161 ax.FontWeight = 'bold'; % Set the font size
162 ax.FontSize = 24; % Set the font size
163 hold off
164 %-----
165 % DNL
166 figure(qtdFileCSV+2)
167 hold on
168 %plot(comandos, lsbPlus, '-', comandos, lsbMinus, '-'); %linas limite do LSB
169 plot(comandos, lsbPlus, '--','LineWidth', 4); %linas limite do LSB
170 plot(comandos, lsbMinus, '--','LineWidth', 4); %linas limite do LSB
171 plot(comandos, DNL,'LineWidth', 4); disp('Plot DNL');
172 % plot(comandos, INL,'LineWidth', 2); disp('Plot INL');
173 % str3 = 'Não Linearidades Diferencial e Integral (DNL e INL)';
174 str3 = '';
175 % str4 = '1 LSB = ' + string(deltat) + ' segundos';
176 str4 = ' (LSB = 990.1 ps)';
177 title(str3);
178 %subtitle(str4);
179
180 ylim([-1.3 1.3]);
181 ax=gca;
182 ax.XAxis.FontWeight = 'bold'; % Peso da fonte do eixo X
183 ax.XAxis.FontSize = 18; % Tamanho da fonte do eixo X
184 ax.FontWeight = 'bold'; % Set the font size
185 ax.YAxis.FontWeight = 'bold'; % Peso da fonte do eixo Y
186 ax.YAxis.FontSize = 18; % Tamanho da fonte do eixo Y
187 ax.FontSize = 24; % Set the font size
188
189 legend({'+LSB','-LSB', 'DNL'});
190 ylabel(str4,'FontSize', 24);
191 xlabel('Input word (n)','FontSize', 24);
192 hold off
193 % -----
194
195 % Coloca uma linha no zero
196 figure(qtdFileCSV+2)
197 hold on
198 DNL = 0;
199 INL = 0;
200 for i=1:(length(diferencas)-1)

```

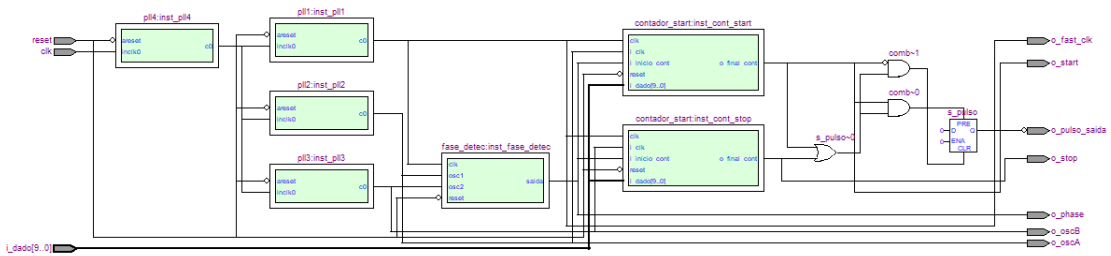
```

201 %comandos = [comandos i];
202 %passoIdeal = [passoIdeal (comandos(i)*deltat)];
203 %esse if serve para desconsiderar a primeira diferença do DNL
204
205 if i==1
206 DNL(i) = 0;
207 else
208 %DNL = [DNL ((diferencas(i+1) - diferencas(i) )/(passoIdeal(1)))-1]];
209 DNL = [DNL (diferencas(i) - (deltat))]; %este que é o certo !!
210 end;
211 %INL = [INL ((diferencas(i) - (deltat*i))/(deltat))];
212 INL = [INL sum(DNL(1:i))]; %este que é o certo !!
213 %INL = [INL (sum(INL)+DNL(i))];
214 end
215 plot(comandos, lsbPlus, '--','LineWidth', 2); %linas limite do LSB
216 plot(comandos, lsbMinus, '--','LineWidth', 2); %linas limite do LSB
217 plot(comandos, DNL,'LineWidth', 2); disp('Plot DNL');
218 % plot(comandos, INL,'LineWidth', 2); disp('Plot INL');
219 str3 = '';
220 % str4 = '1 LSB = ' + string(deltat) + ' segundos';
221 str4 = ' (LSB = 990.1 ps)';
222 title(str3);
223 %subtitle(str4);
224 xlabel('Digital input word(n)','FontSize', 16);
225 ylabel(str4,'FontSize', 16);
226 ylim([-1.3 1.3]);
227 legend({'+LSB', '-LSB', 'DNL'});
228 ax=gca;
229 ax.FontSize = 16; % Set the font size
230 hold off
231 %-----
232 %FINAL DO SCRIPT!!
233 disp('Script completado com sucesso!');

```

APÊNDICE F – Diagrama e reporte de compilação da implementação no FPGA

Diagrama de blocos gerado a partir dos códigos VHDL referentes ao DTC Vernier:



Reporte de compilação referente à síntese do DTC Vernier:

Flow Status	Successful - Tue Aug 06 19:58:35 2024
Quartus II Version	9.1 Build 350 03/24/2010 SP 2 SJ Web Edition
Revision Name	dtc4
Top-level Entity Name	dtc4
Family	Cyclone III
Device	EP3C16F484C6
Timing Models	Final
Met timing requirements	N/A
Total logic elements	222 / 15,408 (1 %)
Total combinational functions	222 / 15,408 (1 %)
Dedicated logic registers	138 / 15,408 (< 1 %)
Total registers	138
Total pins	19 / 347 (5 %)
Total virtual pins	0
Total memory bits	0 / 516,096 (0 %)
Embedded Multiplier 9-bit elements	0 / 112 (0 %)
Total PLLs	4 / 4 (100 %)