# Realização de Detectores de Fase em Dispositivo Lógico Programável (FPGA)

Alexandre Ribas

Departamento de Engenharia Eletrônica e Computação Instituto Tecnológico de Aeronáutica São José dos Campos, Brasil alexandrefukuda@gmail.com

*Resumo*—Esse trabalho apresenta o desenvolvimento, implementação e comparação de quatro arquiteturas para detecção de fase de sinais senoidais em tempo real com o uso de FPGA. Foram abordados o condicionamento do sinal analógico, a conversão analógico-digital e as quatro topologias de detectores de fase. Toda descrição de como o experimento foi realizado, bem como os materiais utilizados e os níveis de sinal estão apresentados. A comparação entre os quatro métodos foi realizada e apresentada por meio da média obtida e o erro associado. Devido à similaridade dos resultados, futuros trabalhos utilizando outras técnicas poderão ser realizados a fim de otimizar mais a detecção de diferença de fase.

Palavras-chave—detecção de fase, cruzamento em zero, FPGA.

## I. INTRODUÇÃO

Os detectores de fase são dispositivos não-lineares, analógicos ou digitais, que produzem uma saída correspondente à diferença de fase entre dois sinais oscilatórios inseridos em suas entradas [1]. O detector de fase é uma das três partes que constituem o *phase locked loop* (PLL), que é um sistema de controle em malha fechada que gera um sinal de saída com frequência e fase sincronizados com o sinal de entrada [2]; tal sistema pode ser utilizado em diferentes ambientes, como: sistemas de potência, sistemas de comunicação, redes inteligentes, dentre outros [3].

A maioria dos detectores de fase utilizados hoje em dia são digitais, uma vez que sua principal presença seja nas PLLs, que em sua grande parte são totalmente digitais ou híbridas (onde apenas o seu detector de fase é digital) [2]. Alguns tipos de detectores de fase analógicos são: multiplicador de quatro quadrantes, multiplicador de Gilbert e *sample and hold*; e entre os digitais citam-se os métodos: OU-exclusivo (XOR), detector de dois estados, detector de fase-frequência, detector de fase de N estados, multiplicador, análise de correlação, baseado em *flip-flop* JK, detector de *Hogge* e detector *Bang-Bang* [1,4].

Atualmente os métodos digitais apresentam vantagens em relação aos analógicos devido à maior acurácia, baixo custo, maior flexibilidade e menor volume físico [5]; porém,

## Osamu Saotome Departamento de Engenharia Eletrônica e Computação Instituto Tecnológico de Aeronáutica São José dos Campos, Brasil osaotome@ita.br

dentre os vários tipos de detectores existentes não é possível apontar qual é o melhor método, porém pode-se apontar alguns métodos que podem ser mais vantajosos de acordo com os requisitos.

No trabalho realizado em [3], foi implementado um detector de fase-frequência em FPGA que apresentou erro menor que 1,8° para diferenças de fase de 180°; em [5] foi realizada análise de correlação com o algoritmo Coordinate Rotation Digital Computer (CORDIC) e dentro da faixa de 0° a 45° o erro relativo foi menor que 1,4%; no trabalho realizado por [6] o detector de fase utilizado foi baseado em flip-flop JK e implementado em FPGA, apresentando correlação entre a fase medida pelo sistema proposto e pelas medições realizadas com o osciloscópio, de 0,99; embora em [7] foram apresentadas foram duas PLLs, a utilização de diferentes detectores de fase apontou as diferenças entre o detector XOR e o de fase-frequência, onde apesar do XOR tenha implementação mais simples, aponta maiores vantagens com o detector de fase-frequência para a aplicação mencionada em [7].

O presente trabalho propõe a implementação e comparação entre quatro métodos de detecção de fase digitais. Os métodos adotados foram: XOR, detector de fase-frequência, detector de dois estados e *flip-flop* JK; as implementações foram realizadas em dispositivo lógico programável do tipo *Field Programmable Gate Array* (FPGA) e a obtenção dos dados em tempo real.

## II. SISTEMA PROPOSTO

A Fig. 1 apresenta o diagrama de blocos do sistema que foi desenvolvido no presente trabalho, onde existem quatro blocos principais: comparador, conversor analógico-digital (A/D), *buffer* e detecção de fase. As entradas *in\_a* e *in\_b* são introduzidas em comparadores analógicos para que os sinais senoidais sejam transformados em ondas quadradas. O segundo passo é realizar a conversão A/D, seguido por *buffers* para garantir o isolamento entre a placa FPGA e o restante do sistema. O último bloco é composto pela placa FPGA, onde as principais componentes serão: o detector de diferença de fase, contadores, unidade lógica aritmética (ULA) para realizar o cálculo da defasagem em graus e a interface com o usuário.



Fig 1. Diagrama de blocos do sistema proposto.

#### III. MÓDULOS FUNCIONAIS

#### A. Comparador

Uma técnica para a conversão dos sinais senoidais em ondas quadradas consiste em utilizar um comparador para a função de cruzamento em zero ou *zero-crossing detector* (ZCD). Utiliza-se um comparador analógico onde o sinal de entrada é comparado com a referência para indicar quando há passagem do nível negativo para o positivo da amplitude do sinal e vice-versa. É interessante utilizar uma topologia de comparador com histerese, onde é definida uma faixa na qual não há variação do sinal de saída, minimizando os efeitos de ruído perto do nível de transição. Dessa forma, em todo momento que o sinal senoidal cruzar o eixo de *offset* produzirá um sinal de saída com tensão Vcc para transições ascendentes e com o nível de referência para transições descendentes.

### B. Conversor Analógico-Digital

O conversor analógico-digital utilizado foi o MAX1426EAI+ da *Maxim*. Esse conversor possui resolução de 10 bits, taxa de amostragem de até 10 MSPS, entrada diferencial de  $\pm 2V$ , baixa capacitância de entrada e ampla largura de banda de entrada. O próprio conversor possui três tensões de referência interna e alimentações independentes para o sinal analógico e digital, sendo capaz de trabalhar com lógica digital de nível CMOS ou TTL. O sinal digital de saída desse conversor é em complemento de 2 [8].

#### C. Buffers

A utilização de *buffers* entre as saídas digitais do conversor A/D tem a função de realizar o isolamento adequado entre a placa FPGA e o restante do circuito. Esse componente é importante no quesito de prevenir interferências que possam ocorrer nas entradas digitais da placa a ser utilizada, principalmente quando se tratam de sinais com uma frequência mais elevada. Essa recomendação também pode ser observada nas placas de desenvolvimento do fabricante do conversor A/D, que também introduzem *buffers* no caminho do sinal digitalizado até a placa digital a ser utilizada.

## IV. MÓDULOS DE DETECTORES DE FASE

## A. Exclusivo-OU (XOR)

O detector de fase baseado na lógica XOR é o mais simples por tratar-se apenas de uma porta lógica, porém os pulsos de saída estarão representando as transições positivas e negativas, sendo essas com o dobro da frequência. Também não exibem informação em relação a qual sinal estará atrasado ou adiantado. Outra característica é ser sensível ao ciclo de carga do sinal de entrada, onde sua saída pode apresentar erros caso o ciclo dos sinais de entrada sejam diferentes de 50%. A Fig. 2(a) apresenta a porta XOR, onde as entradas são Vi e Vo e a saída nãoinversora está apresentada por Va. Na Fig. 2(b) pode-se observar a resposta da saída, que tem comportamento linear somente entre  $\pm 90^{\circ}$ .



Fig 2. (a) Detector de fase baseado em uma porta XOR e (b) apresenta o comportamento do sinal de saída [1].

## B. Detector de Dois Estados

O detector de dois estados está apresentado na Fig. 3(a), contando com dois *flip-flops* tipo D e uma porta XOR. Em relação ao detector anterior, o detector de dois estados é sensível somente às bordas de subida dos sinais de entrada, uma vez que as entradas Vi e Vo atuam como o sinal de relógio dos *flip-flops*, sendo menos susceptível à mudança de estados causadas por ruídos. Sua vantagem é possuir uma resposta linear em  $\pm 180^{\circ}$ , como pode ser observado na Fig. 3(b).



Fig 3. (a) Detector de fase *two state phase detector* e (b) apresenta o comportamento do sinal de saída [1].

#### C. Detector de Fase-Frequência

A Fig. 4(a) apresenta o detector de fase-frequência, cujo circuito é similar ao anterior por contar com dois *flip-flops* e uma porta lógica. Essa topologia é capaz de gerar sinais somente na borda de subida dos sinais de entrada e possui duas saídas, que são geradas de acordo com qual sinal está atrasado ou adiantado. Esse circuito tem melhor operação

quando a diferença de fase entre os sinais de entrada está mais próxima. Esse detector é sensível somente à borda e insensível ao ciclo de trabalho do sinal de entrada. Produz saída linear em  $\pm 360^{\circ}$ , como pode ser visto na Fig. 4(b).



Fig 4. (a) Detector de fase do tipo *phase-frequency detector* e (b) apresenta o comportamento do sinal de saída [1].

## D. Flip-flop JK

O detector baseado em *flip-flop* JK, apresentado na Fig. 5(a), consiste basicamente em inserir os sinais de entrada Vi e Vo nas estradas J e K. Esse circuito também é sensível à frequência, tendo a vantagem de ter uma resposta de  $\pm 180^{\circ}$ . Outra característica é que o JK não é sensível ao ciclo de trabalho do sinal de entrada, sendo uma alternativa bastante interessante quando o sinal de entrada for muito susceptível a distorções. A Fig. 5(b) apresenta a linearidade do sinal de saída, que é na faixa de  $\pm 180^{\circ}$ .



Fig 5. (a) Detector de fase baseado em um flip-flop JK e (b) apresenta o comportamento do sinal de saída.

#### V. IMPLEMENTAÇÃO EM HARDWARE

A arquitetura proposta apresentada pela Fig. 1 utilizou sinais de entrada  $in_a$  e  $in_b$ , de caráter senoidal, que foram provenientes de um gerador de funções *Agilent 33522 30-MHz Function/Arbitrary Waveform Generator*. As características dos sinais de entrada foram: frequência de 1 kHz; amplitude de 4 Vpp; offset de 2,25 V; e a fase foi variante de 0 a 180 com espaçamento de 15°.

Em seguida, esses sinais foram introduzidos em comparadores analógicos para obtenção de ondas quadradas através do método de cruzamento em zero. Para esse circuito, foram utilizados amplificadores operacionais LM7171BIN da *Texas Instruments*, que são de alta velocidade, possuindo alto *slew-rate*, e portando adequados para tal função.

As ondas quadradas foram digitalizadas pelo MAX1426. Para a utilização do conversor A/D, os sinais de habilitação e relógio foram providos pela placa FPGA. A taxa de conversão é proporcional ao sinal de relógio, que por sua vez foi de 1 MHz.

Em seguida, na etapa de *buffer*, foram utilizados circuitos integrados (CIs) da *Texas Instruments* CD4050BE, que contém em cada CI seis *buffers* não inversores capazes de operar em nível lógico CMOS ou TTL e com tempo de transição suficientemente baixos para não comprometer a boa funcionalidade do sistema.

A placa FPGA utilizada para implementação dos detectores de fase foi a *Digilent Basys3 Artix-7 XC7A35T-1CPG236C* e o *software* utilizado foi o *Xilinx Vivado* 2017.4.

Foram então implementadas as quatro arquiteturas de detecção de fase, cada uma separadamente, utilizando linguagem de descrição de *hardware* (VHDL). Como se pôde observar na Fig. 1, o primeiro bloco realizado pelo FPGA foi o circuito de detecção de fase e, a partir dos pulsos que o mesmo gerou, foram introduzidos em um contador operando em frequência de 10 MHz. Os pulsos referentes à diferença de fase foram utilizados como habilitador do contador síncrono, ou seja, enquanto os pulsos estavam em nível lógico alto, o contador era incrementado e quando houve a transição dos pulsos para nível lógico baixo, tal valor era enviado a um registrador.

O passo seguinte consistiu em utilizar dez registradores dinâmicos de modo a obter a média aritmética das dez últimas contagens e através de uma unidade lógica e aritmética, expressar tal valor em graus no display de 7 segmentos que a própria placa FPGA dispõe.

Por fim, os valores de diferença de fase utilizados como referência foram observados através de um osciloscópio *Agilent DSO-X 2012A* e os valores a serem comparados foram registrados a partir do valor expresso pelo display da placa FPGA. Os sinais senoidais recebidos pelo gerador de funções foram verificados através do osciloscópio no mesmo ponto em que foram conectados ao circuito de cruzamento em zero.

#### VI. RESULTADOS EXPERIMENTAIS

Os dados obtidos através do sistema proposto estão apresentados na Tabela 1. Durante a captura de dados, houve variação do valor expresso pelo sistema, dessa forma, foram registados os valores máximo e mínimo para cada grau requerido e obtenção da média aritmética. O erro apresentado foi através do cálculo do desvio padrão, indicando qual seria o valor do erro caso fosse realizada apenas uma medição para cada caso.

Fase	Topologias Implementadas			
	Dois Estados	Fase-freq	XOR	JK
15	14,96±0,28	14,97±0,25	14,99±0,03	14,94±0,25
30	30,09±0,25	30,09±0,25	30,00±0,03	30,06±0,25
45	45,02±0,02	45,03±0,00	44,98±0,03	45,00±0,00
60	59,97±0,25	59,97±0,25	59,99±0,03	59,94±0,25
75	75,08±0,23	75,09±0,25	74,99±0,01	75,06±0,25
90	90,02±0,02	90,03±0,00	89,98±0,03	90,00±0,00
105	104,94±0,25	104,97±0,25	104,99±0,03	105,12±0,40
120	120,08±0,23	120,09±0,25	120,00±0,03	120,29±0,13
135	135,03±0,00	135,03±0,00	134,98±0,03	135,65±0,40
150	149,97±0,25	149,97±0,25	149,99±0,03	150,30±0,05
165	165,09±0,25	165,09±0,25	165,00±0,03	165,29±0,03
180	180,00±0,00	180,03±0,00	179,98±0,03	180,02±0,02

TABELA I. DADOS OBTIDOS

<sup>a.</sup> Todos os valores expressos são em graus.

#### VII. CONCLUSÃO

É possível observar que a comparação das quatro topologias não apresentou nenhuma discrepância significativa, onde todos os métodos apresentaram boa precisão e baixo valor de erro.

A carga computacional utilizada pela FPGA não foi levantada em questão, uma vez que a quantidade de LUTs e *flip-flops* foi bem baixa e em nenhum caso comprometendo mais que 2% da capacidade da placa.

A utilização do mesmo bloco de contador nas quatro topologias fez com a implementação de cada topologia pudesse ter resultados similares aos outros. Dessa forma, pretende-se utilizar esse sistema para sinais de radiofrequência e nesse caso será necessário implementar diferentes técnicas no contador devido à frequência do sinal ser próxima ao sinal de relógio máximo da FPGA.

Outro ponto a ser estudado será a comparação desses métodos utilizando lógica combinacional e *flip-flips* com algoritmos que calculam o ângulo de fase por meio de coordenadas, como por exemplo o algoritmo CORDIC.

#### REFERENCES

- D. Abramovitch, "Phase-locked loops: a control centric tutorial," *Proceedings of the 2002 American Control Conference* (*IEEE Cat. No.CH37301*), Anchorage, AK, USA, 2002, pp. 1-15 vol.1. doi: 10.1109/ACC.2002.1024769
- [2] S. Edway and R. K. Manjunath, "Design and simulation of FPGA based all digital phase locked loop (ADPLL)," 2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Tumkur, 2017, pp. 259-263. doi: 10.1109/ICATCCT.2017.8389144
- [3] P. Kumar, V. Kumar and R. Pratap, "Design and implementation of phase detector on FPGA," 2017 6th International Conference on Computer Applications In Electrical Engineering-Recent Advances

*(CERA)*, Roorkee, 2017, pp. 108-110. doi: 10.1109/CERA.2017.8343310

- [4] Jianjun Wei and Xinzhang Jia, "Two-stage phase/frequency detecting CPPLL," 2007 Australasian Telecommunication Networks and Applications Conference, Christchurch, 2007, pp. 303-306. doi: 10.1109/ATNAC.2007.4665262
- [5] T. Wang, Y. Hou, S. Tang, H. Lei and Z. Deng, "Measuring phase difference of sinusoidal signals based on FPGA," 2017 13th IEEE International Conference on Control & Automation (ICCA), Ohrid, 2017, pp. 1039-1042. doi: 10.1109/ICCA.2017.8003204
- [6] M. A. G. Lopez et al, "FPGA-based phase and module impedance meter," 2013 Journal of Physics: Conference Series, 477, 12006. https://doi.org/10.1088/1742-6596/477/1/012006
- [7] S. Li and Y. Zhou, "Phase-locked loop of inverter based on FPGA," 2012 5th International Conference on BioMedical Engineering and Informatics, Chongqing, 2012, pp. 1490-1493. doi: 10.1109/BMEI.2012.6513128
- [8] Maxim Integrated Products, MAX1426 datasheet, 2011