

Desenvolvimento de solução embarcada baseada em FPGA do tipo Zynq para sistema de ultrassom por transdutores EMAT



www.ele.puc-rio.br

Aluno: Lucas Grativol Ribeiro

Orientador: Alan Conci Kubrusly
Co-orientador: Felipe Calliari

22/12/2020

1. Introdução
2. Revisão de teórica
3. Desenvolvimento
 - a. Visão Geral
4. Resultados
 - a. Simulação da solução
 - b. Implementação
5. Conclusão
6. Referências

Motivação:

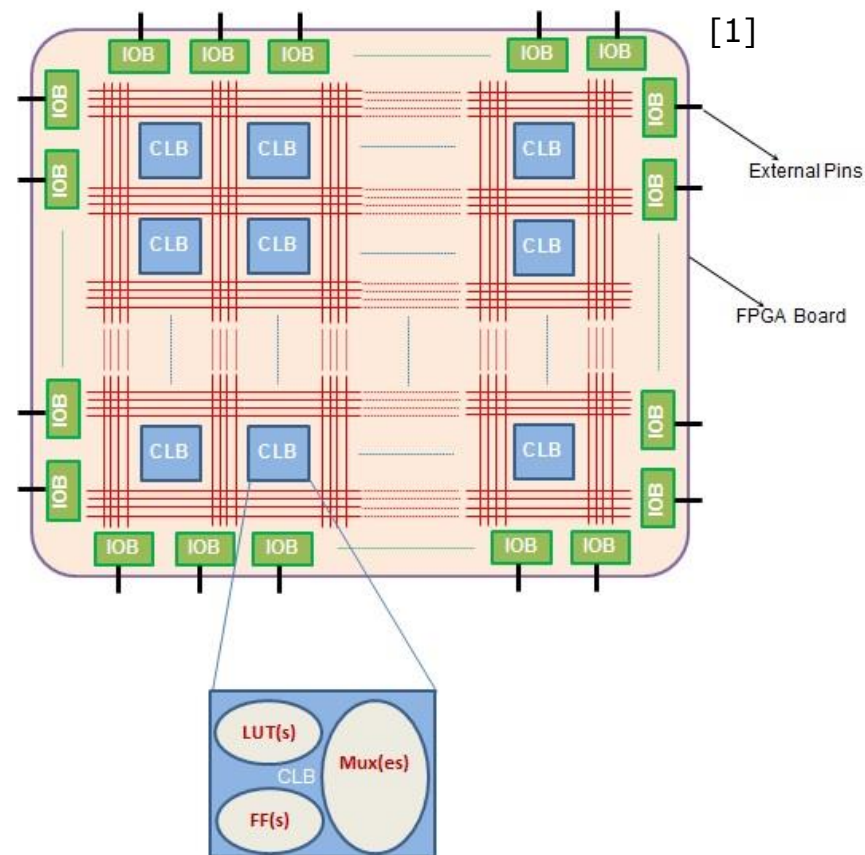
- Adição de funcionalidades aos PIGs (*Pipeline Inspection Gauge*) desenvolvidos pelo CPTI;
- Inspeção não-destrutivas com EMAT (*Electromagnetic Acoustic Transducer*).

Objetivos:

- Construção de uma plataforma para geração de sinais de excitação para testes em EMAT;
- Solução configurável pelo usuário;
- Processamento em tempo real;

Plataforma de desenvolvimento

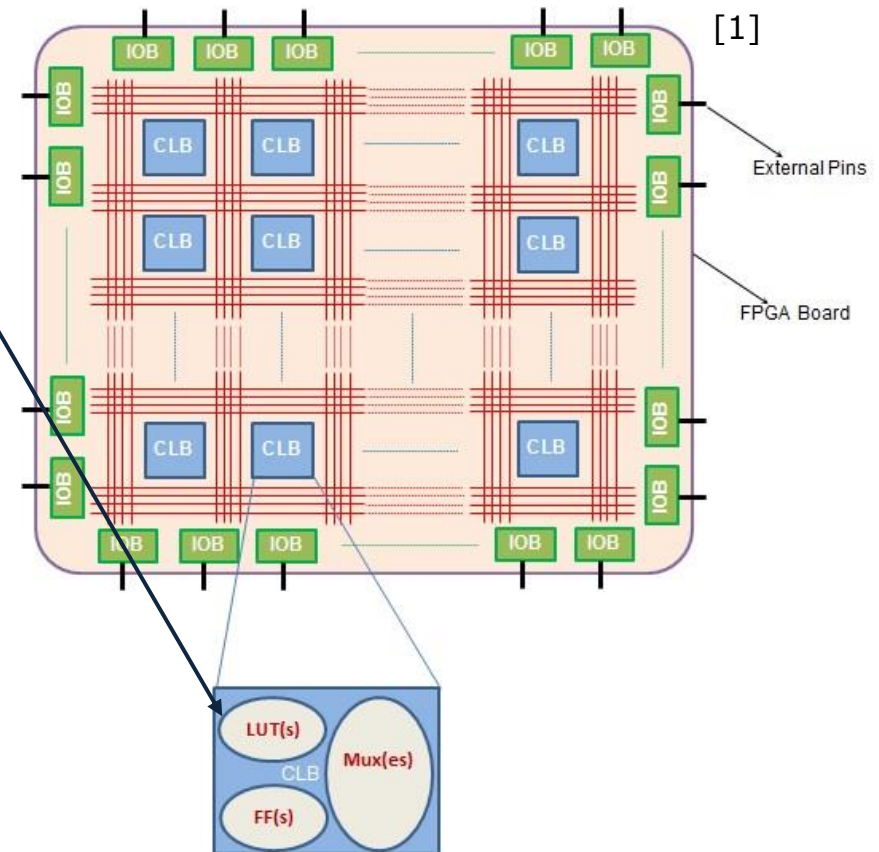
FPGA (Field Programmable Gate Array)



Interior de uma FPGA (ideia geral)

Plataforma de desenvolvimento

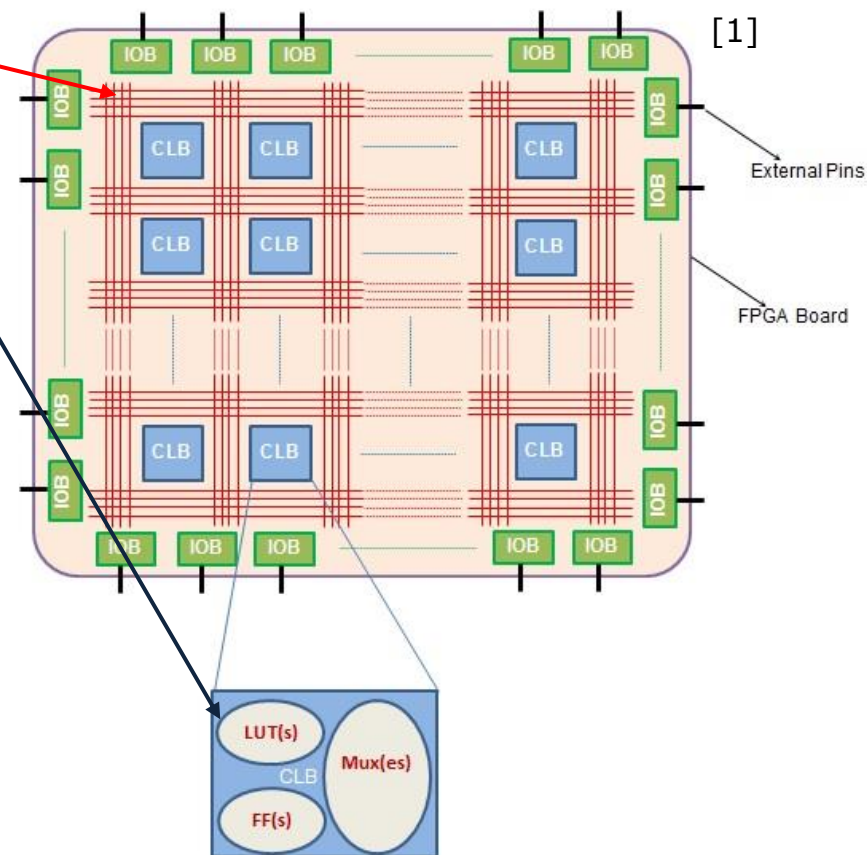
FPGA (Field Programmable Gate Array)



Interior de uma FPGA (ideia geral)

Plataforma de desenvolvimento

FPGA (Field Programmable Gate Array)

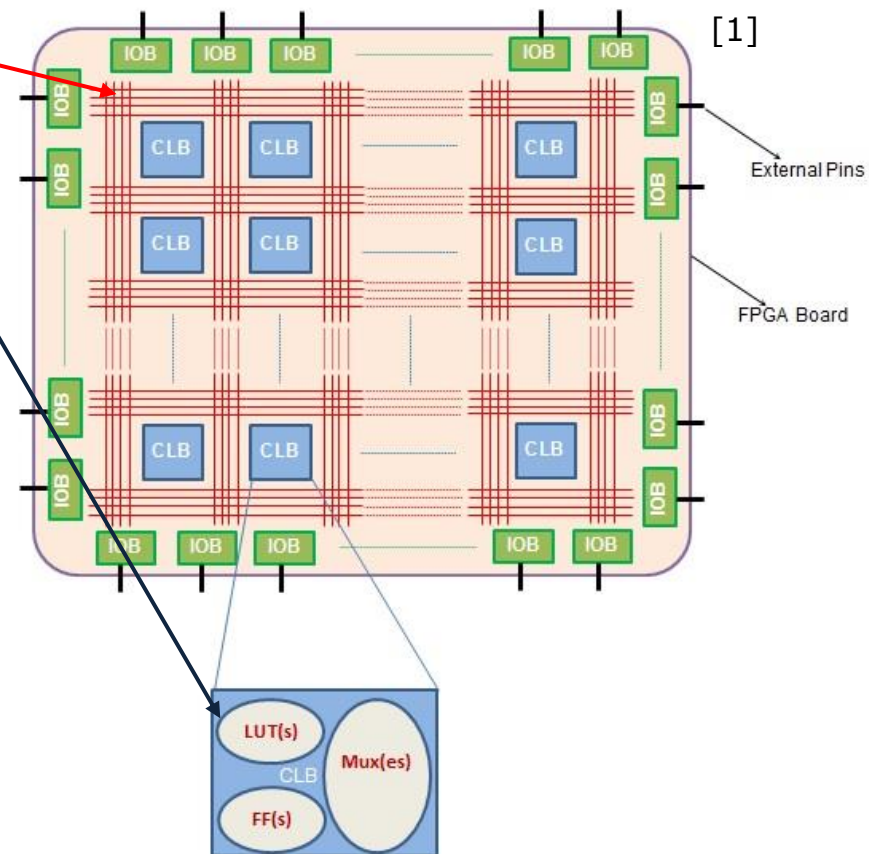


Interior de uma FPGA (ideia geral)

Plataforma de desenvolvimento

FPGA (Field Programmable Gate Array)

Descrição Material
com VHDL-93



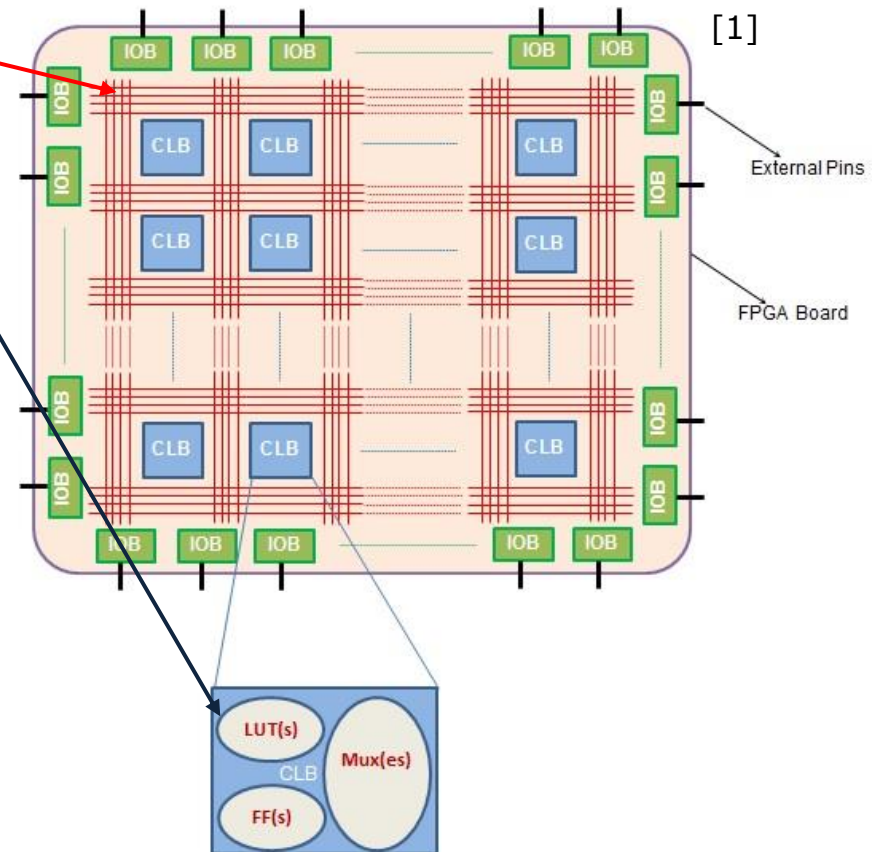
Interior de uma FPGA (ideia geral)

Plataforma de desenvolvimento

FPGA (Field Programmable Gate Array)

Descrição Material
com VHDL-93

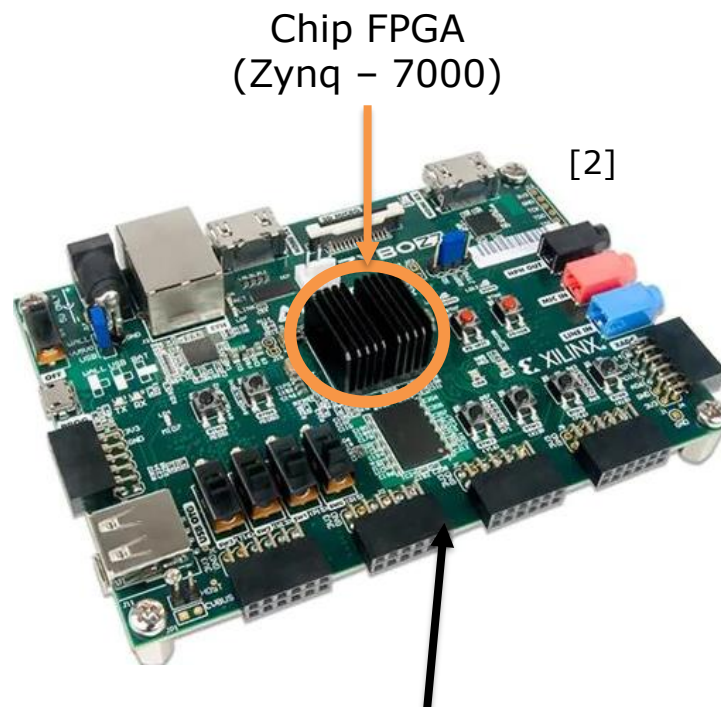
Software
(Síntese, Implementação e
Programação)



Interior de uma FPGA (ideia geral)

Plataforma de desenvolvimento

Zybo Z2-70, fabricada pela Digilent com a família Zynq da Xilinx.

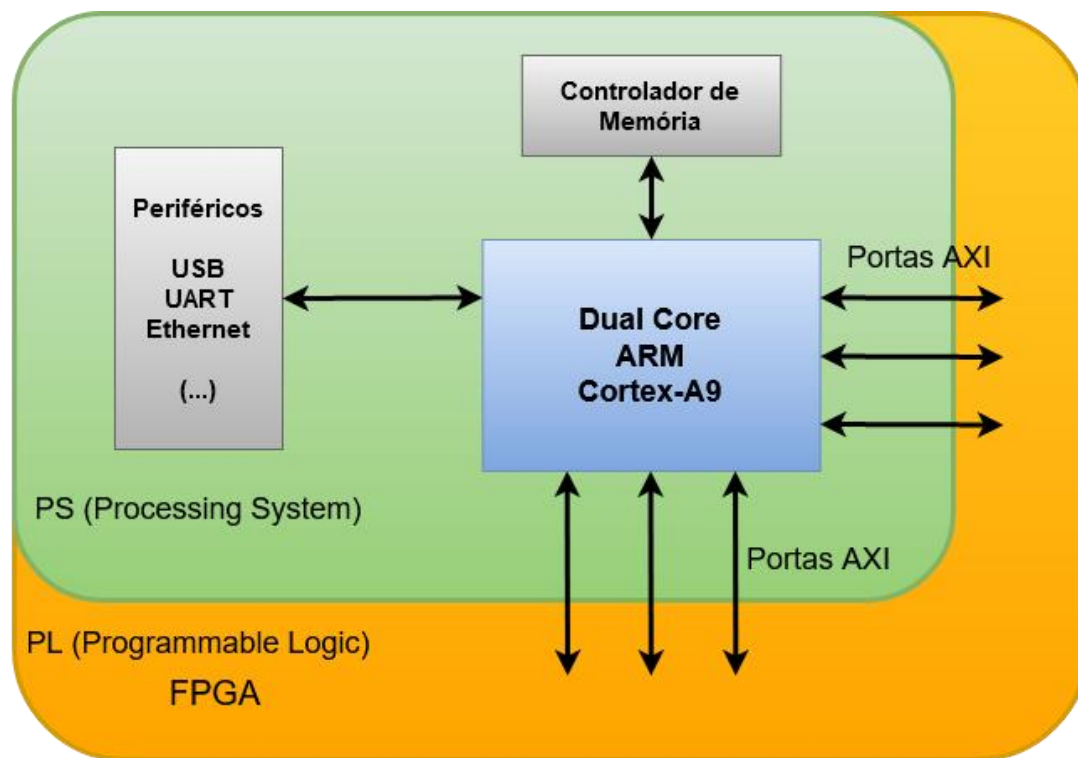


Placa de desenvolvimento

Placa Zybo Z2-70

Plataforma de desenvolvimento

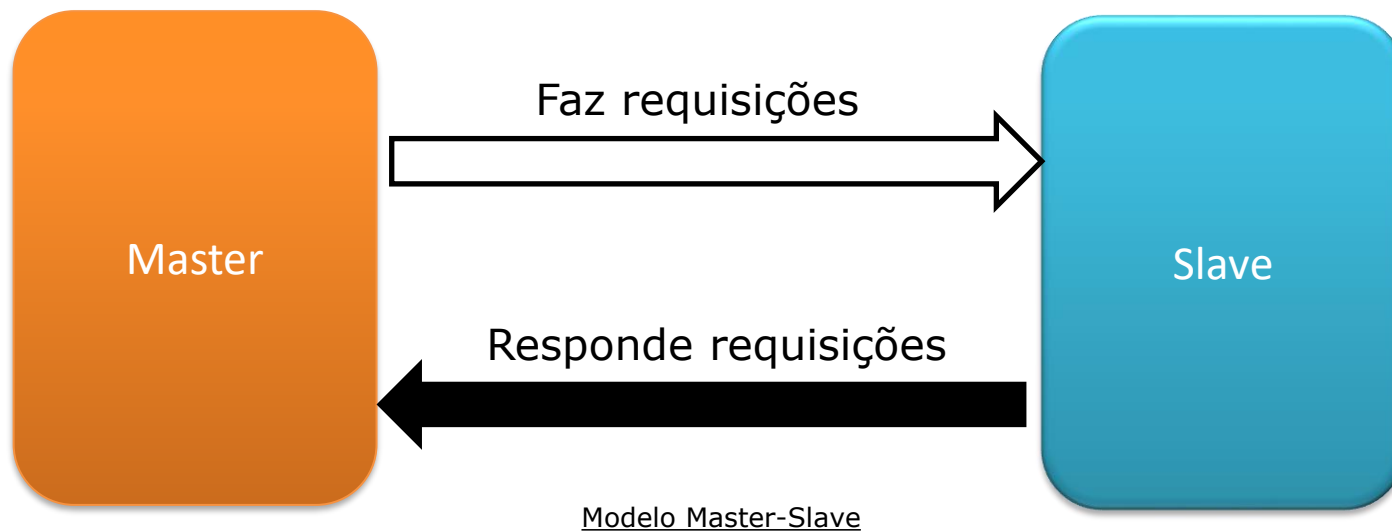
Arquitetura família Zynq-7000



Arquitetura da família Zynq

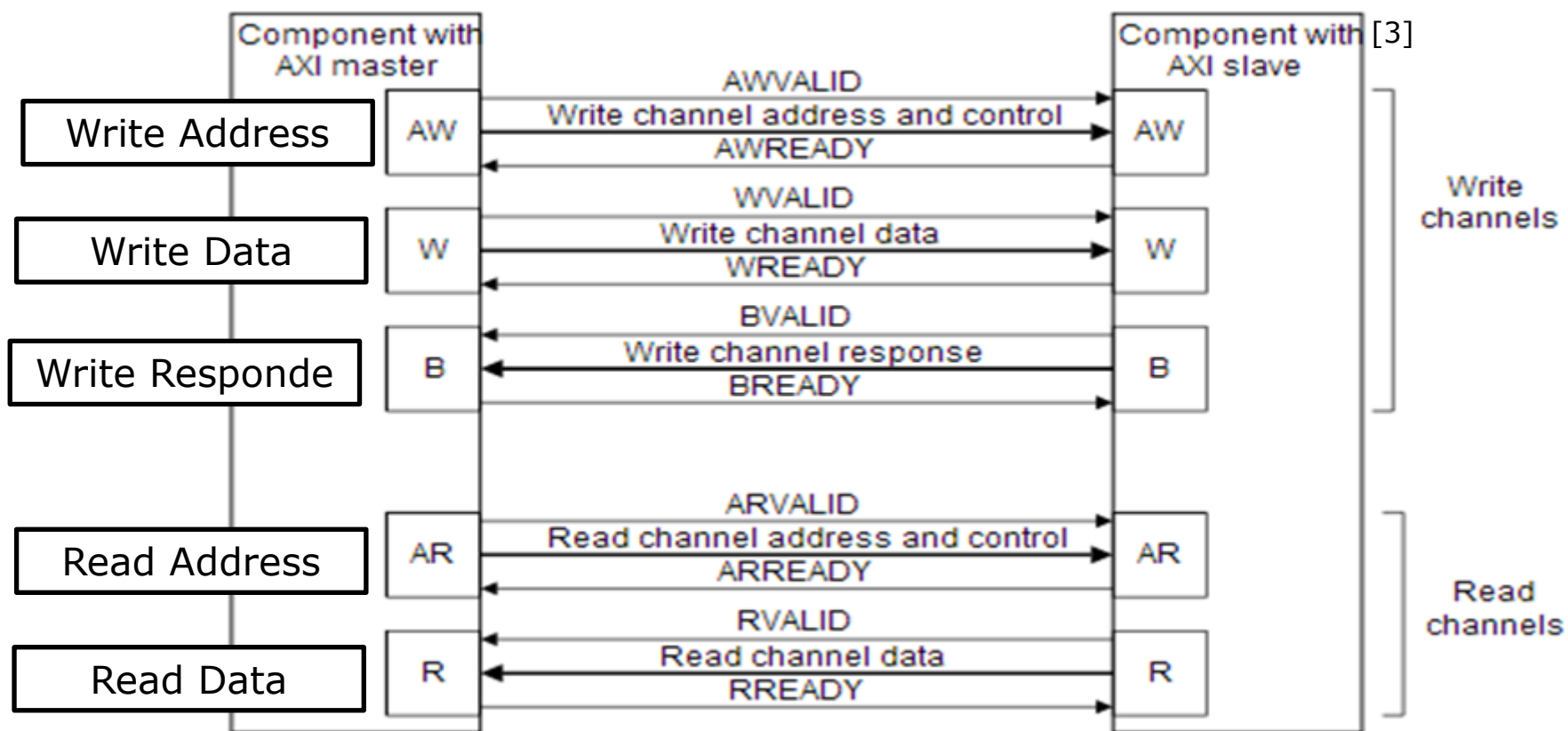
Protocolo AXI (Advanced eXtensible Interface)

- Desenvolvido pela ARM, adotado como padrão pela Xilinx;
- Modelo Master-Slave
- Handshake "ready-valid"
- Dois tipos principais: Memory Mapped e Stream



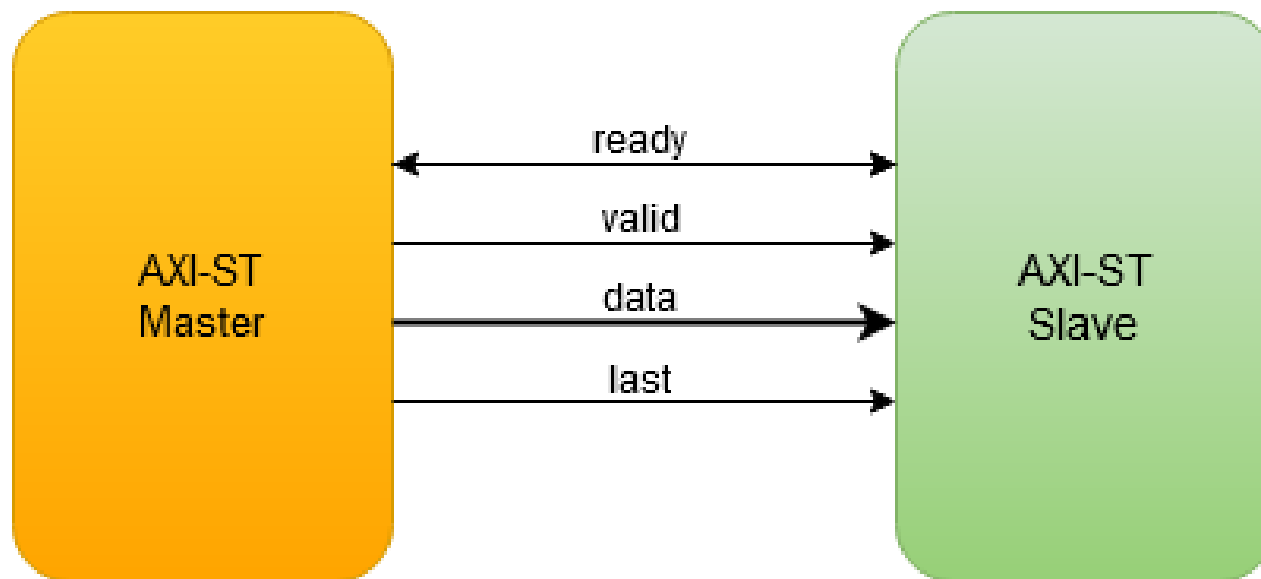
Protocolo AXI (Advanced eXtensible Interface)

AXI-MM (Memory Mapped)

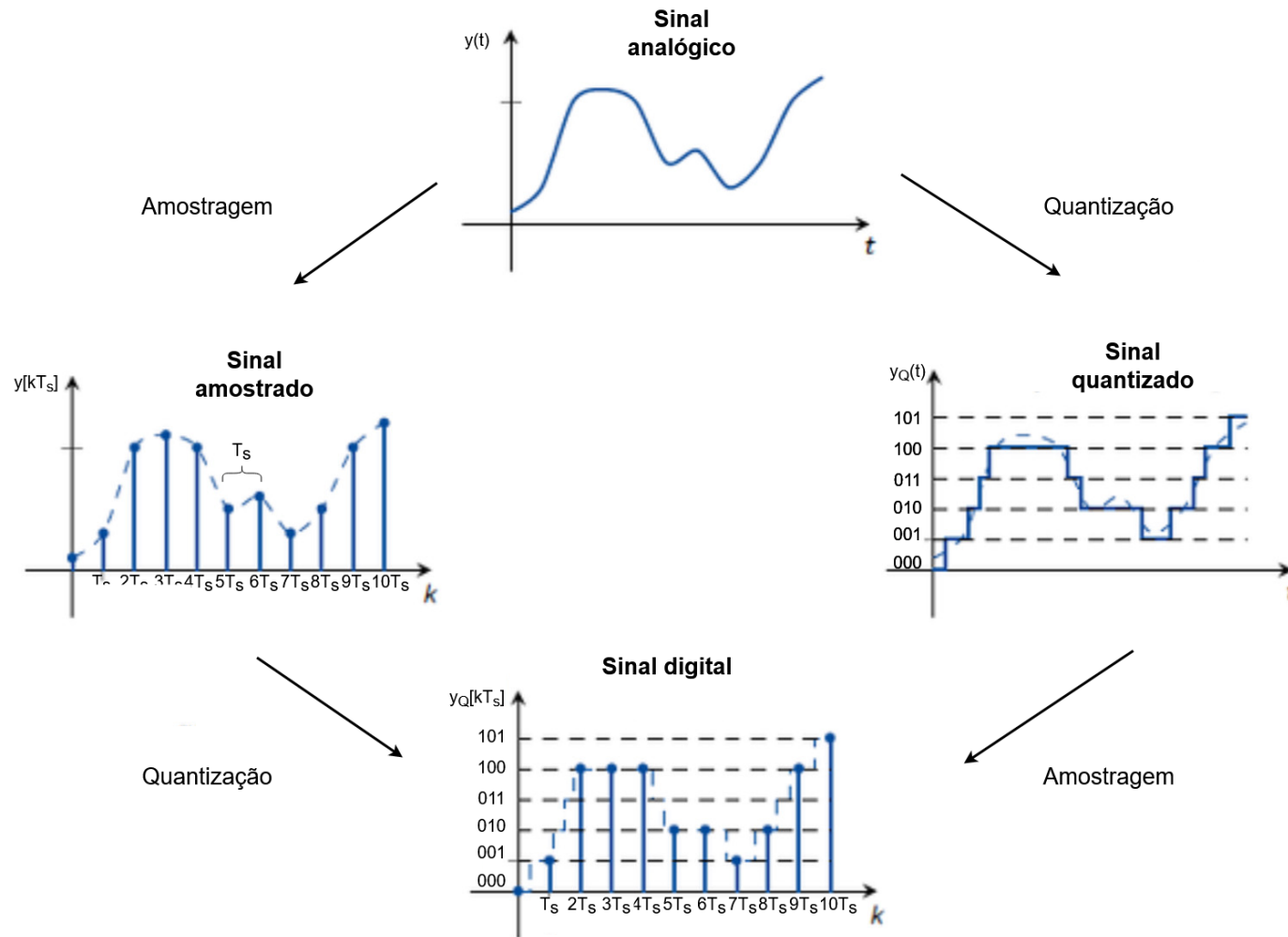


Protocolo AXI (Advanced eXtensible Interface)

AXI-ST (Stream)

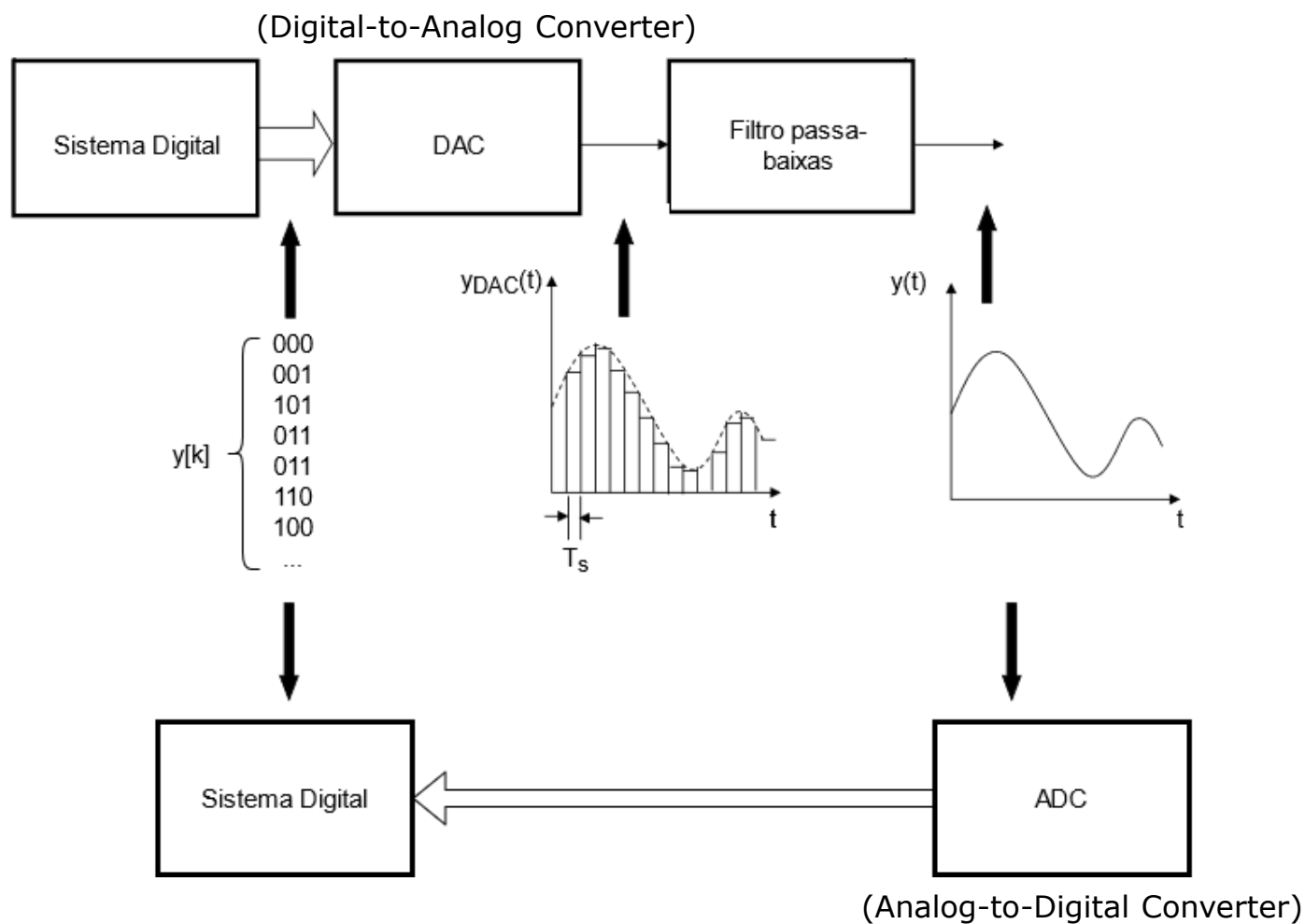
Protocolo/Interface AXI-ST

Síntese de Sinais



Sinal Digital a partir de um sinal analógico

Síntese de Sinais



Síntese de Sinais

Sinal Arbitrário

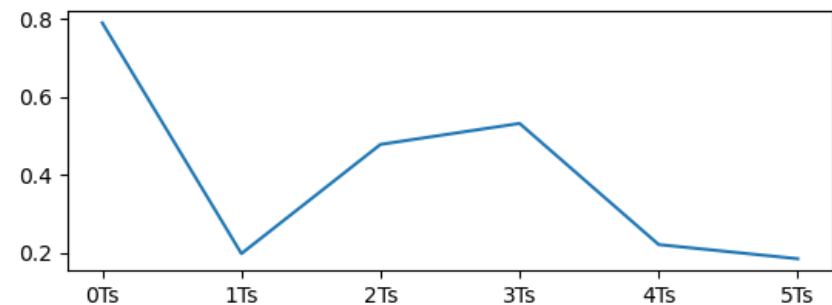
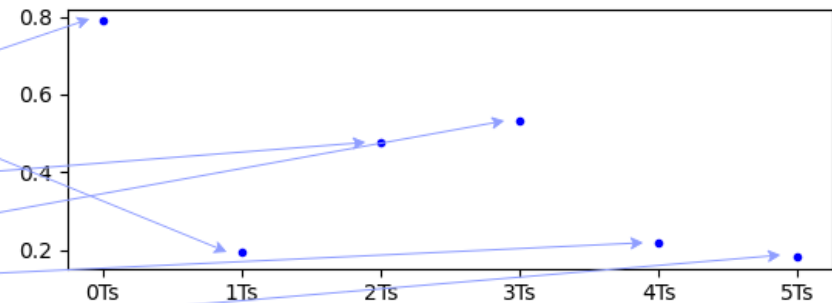
Vantagens:

- Solução genérica;
- Ajustes em software.

Desvantagens:

- Interface de acesso;
- Alto gasto de recursos.

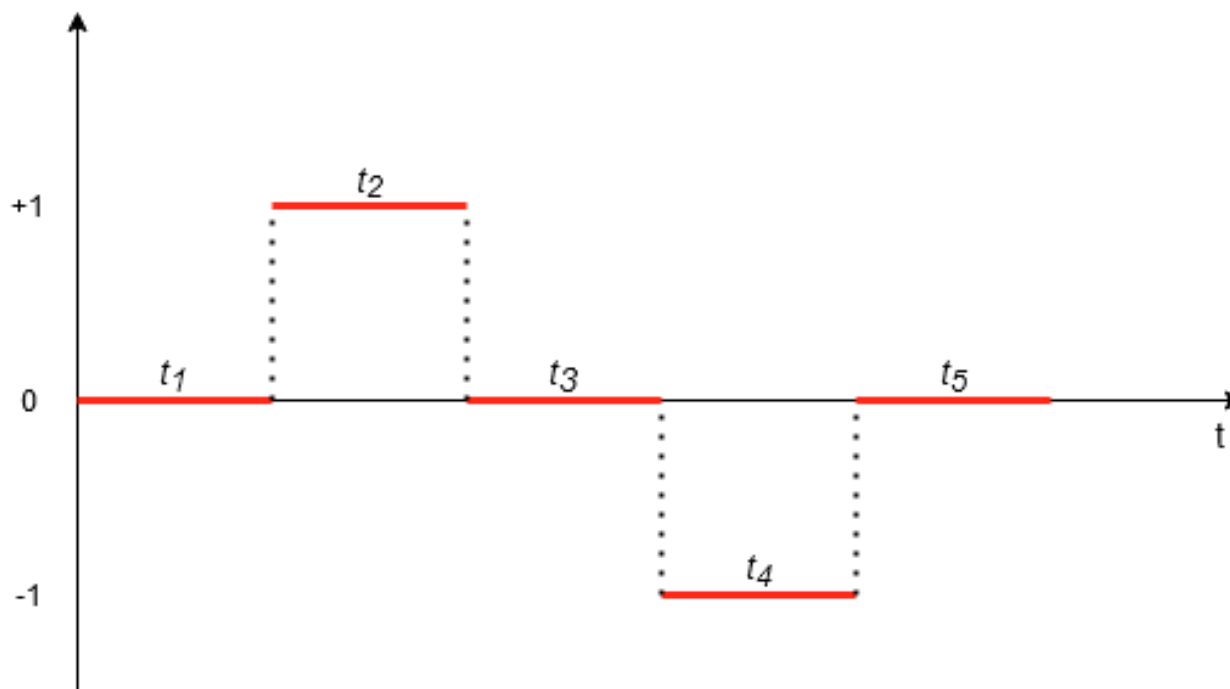
Tabela de pontos	
Índice	Ponto
0	0.0445
1	0.9964
2	0.6644
3	0.3192
4	0.9215
5	0.7684



Exemplo método da tabela

Síntese de Sinais

Sinal Quadrado



Sinal quadrado

Síntese de Sinais

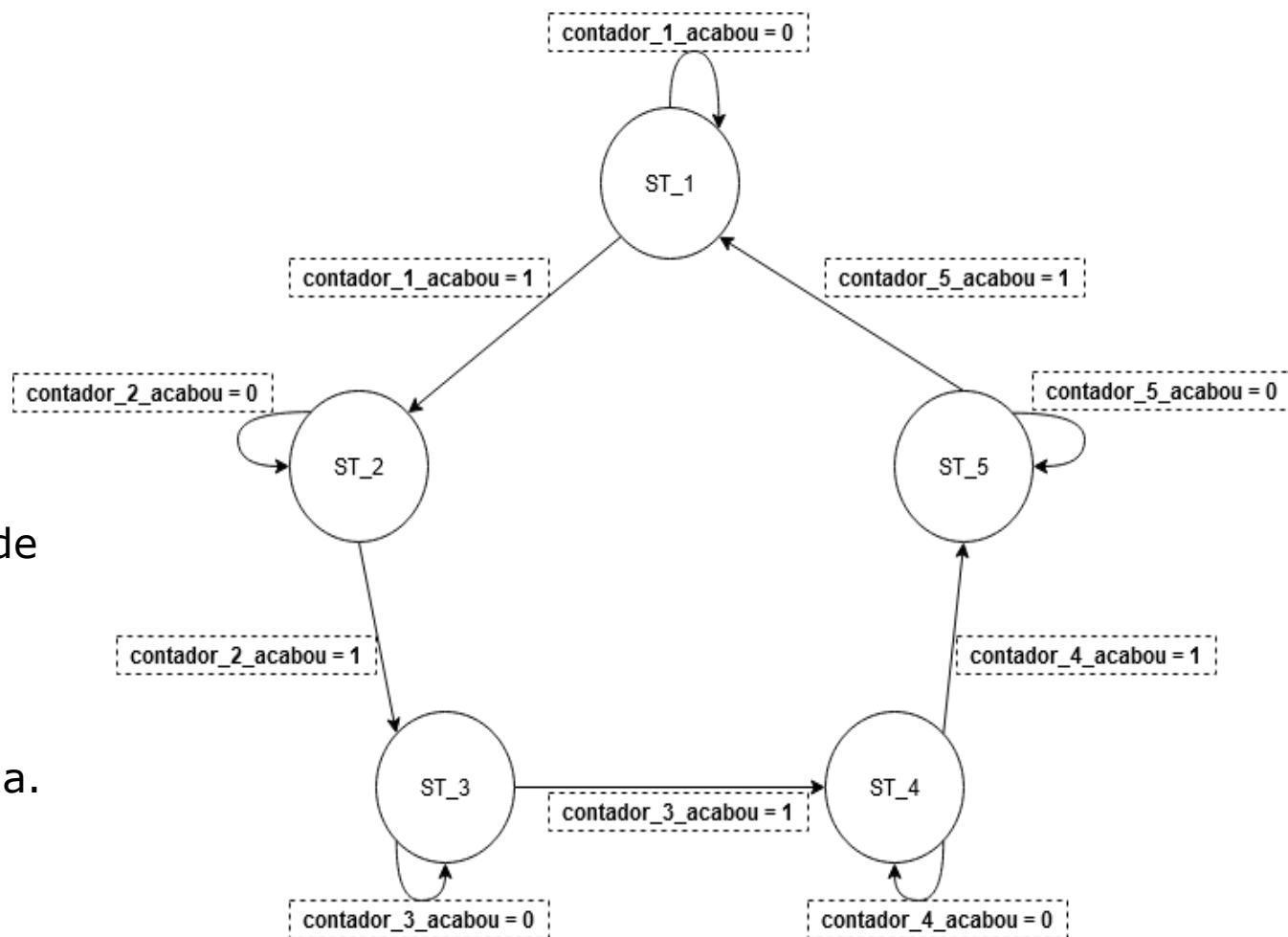
Sinal Quadrado

Vantagens:

- Fácil descrição;
- Baixa necessidade de recursos.

Desvantagem:

- Definição limitada.

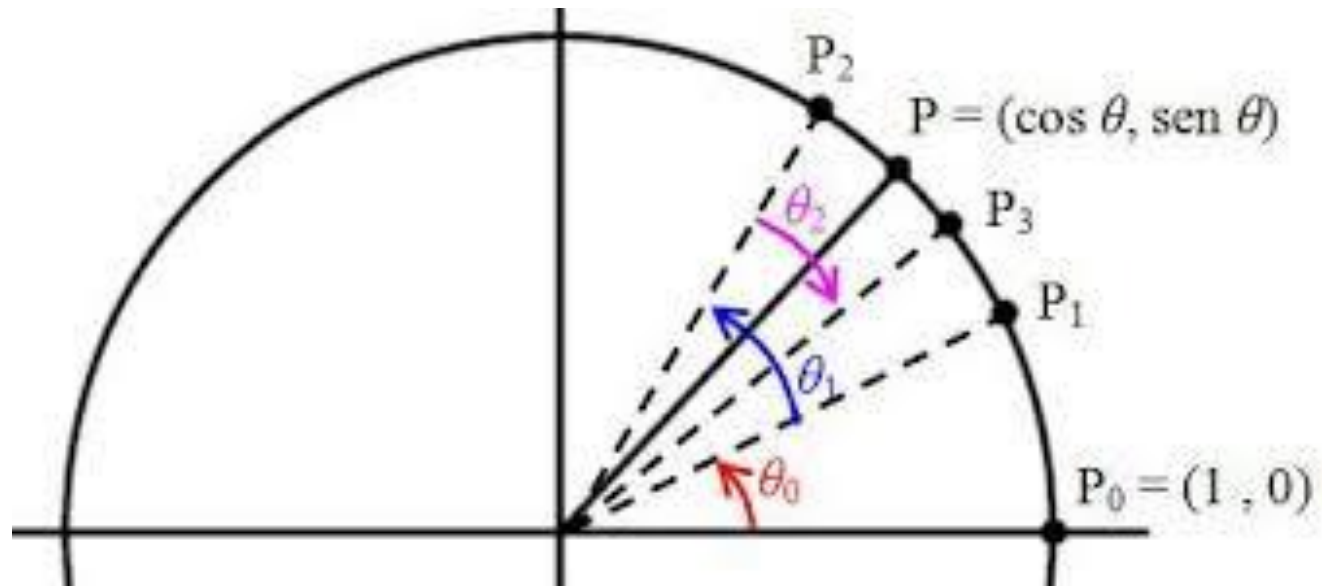


Exemplo de FSM para produção de um sinal quadrado

Síntese de Sinais

Sinal Seno e Cossenos

Algoritmo CORDIC (**CO**ordinate **R**otation **DI**gital **C**omputer)



Exemplo do algoritmo CORDIC

Síntese de Sinais

Sinal Seno e Cossenos

Vantagens:

- Baixa necessidade de recursos;
- Seno e cosseno calculados no mesmo ciclo.

Desvantagem:

- Aumento de precisão, aumento de complexidade (recursos);
- Algoritmo limitado entre $\left[-\frac{\pi}{2}; \frac{\pi}{2}\right]$.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

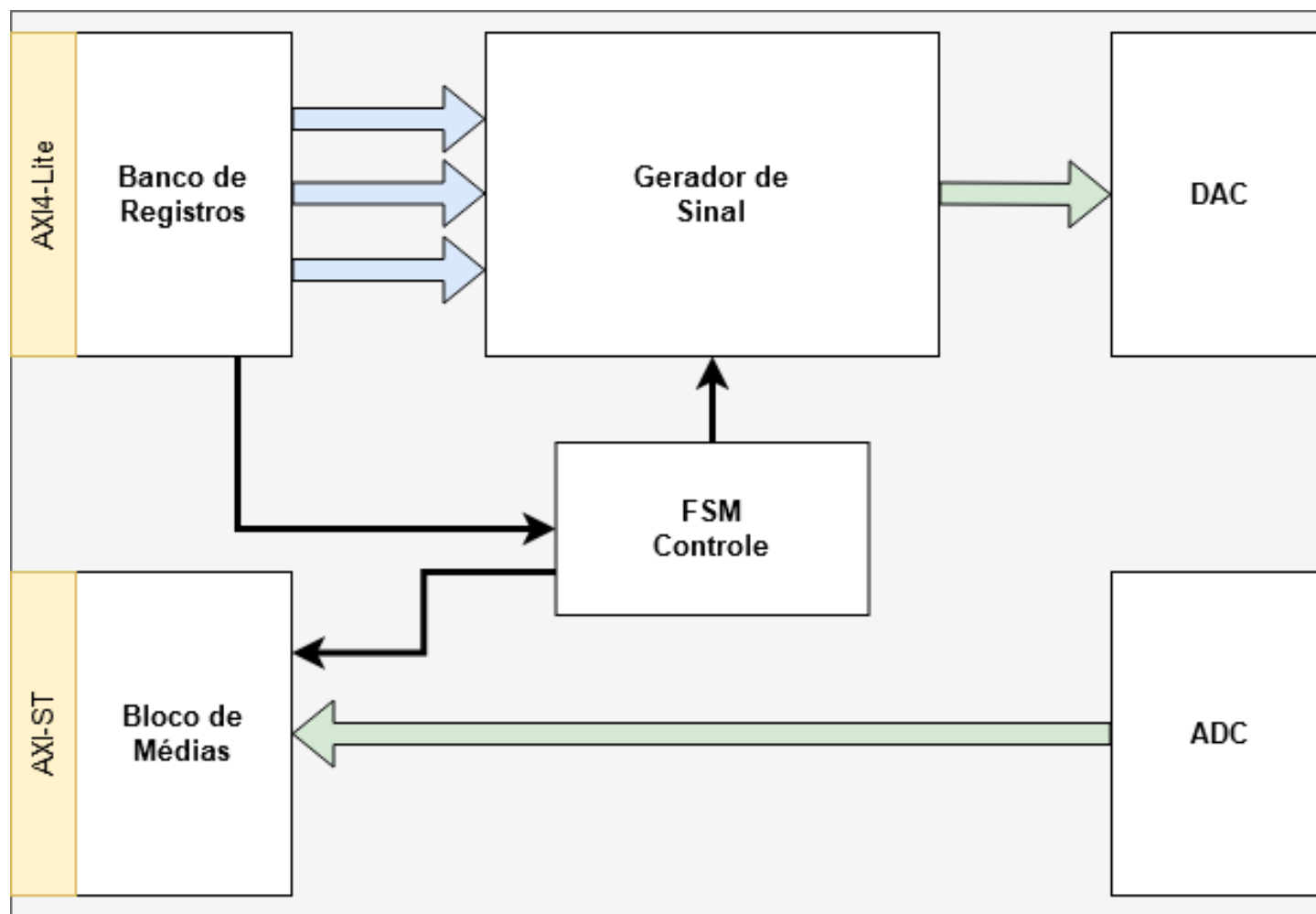
$$V' = (\cos(\theta))^{-1} \begin{bmatrix} 1 & -\tan(\theta) \\ \tan(\theta) & 1 \end{bmatrix} V$$

$$\theta_k = \arctan(2^{-k}) \quad \theta = \sum_{k=0}^{\infty} d_k \cdot \theta_k$$

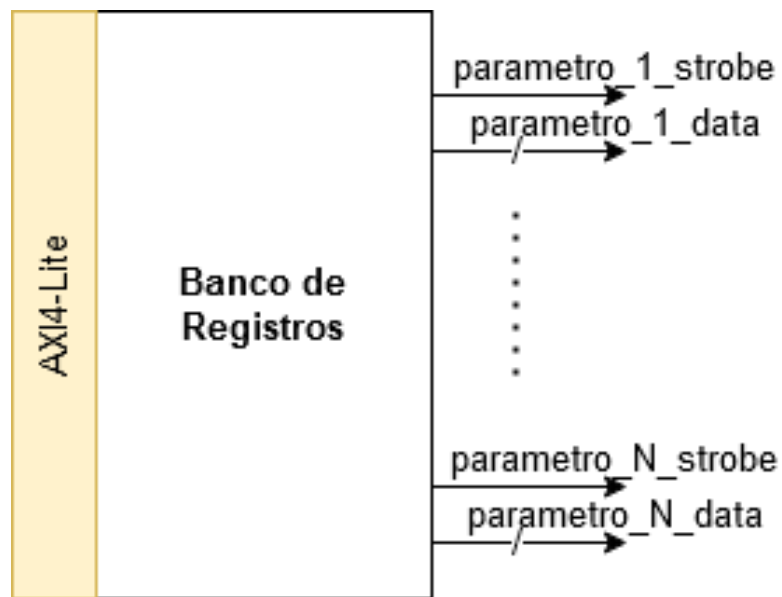
$$V' = A_k \begin{bmatrix} 1 & -2^{-k} \\ 2^{-k} & 1 \end{bmatrix} V$$

$$A_k = (\cos(\arctan(2^{-k})))^{-1}$$

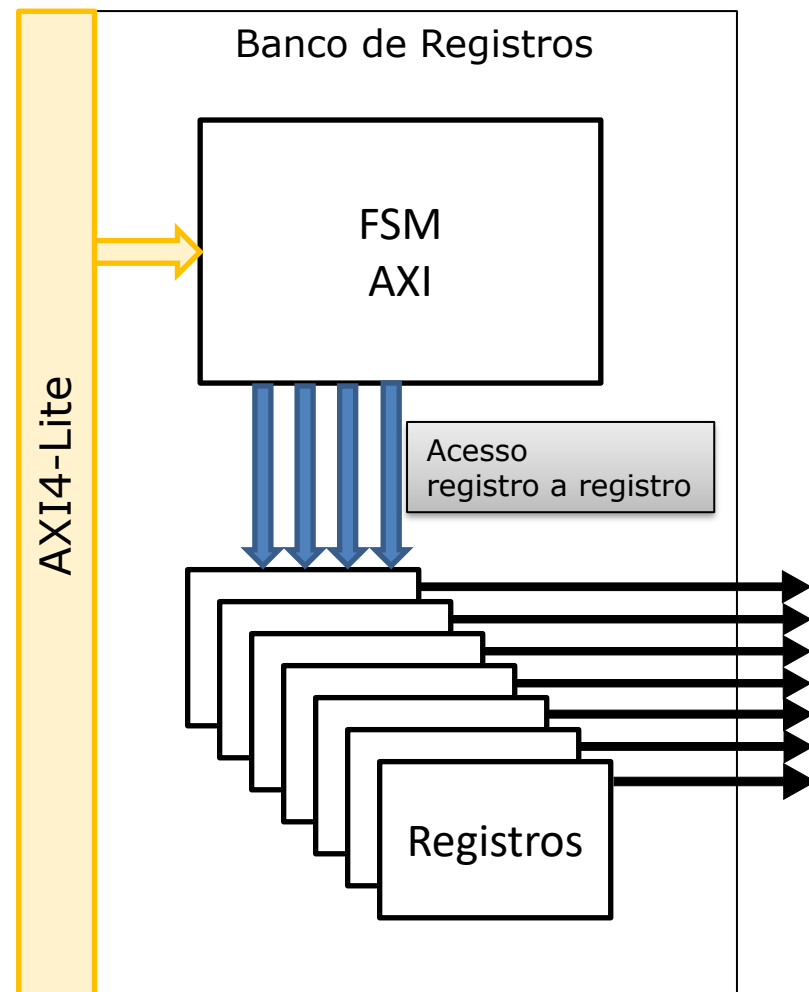
$$\begin{cases} x_{k+1} = A_k(x_k - d_k 2^{-k} y_k) \\ y_{k+1} = A_k(y_k + d_k 2^{-k} x_k) \\ z_{k+1} = z_k - d_k \arctan(2^{-k}) \end{cases}$$

Solução proposta – Diagrama de blocosDiagrama de blocos da solução proposta

Solução proposta

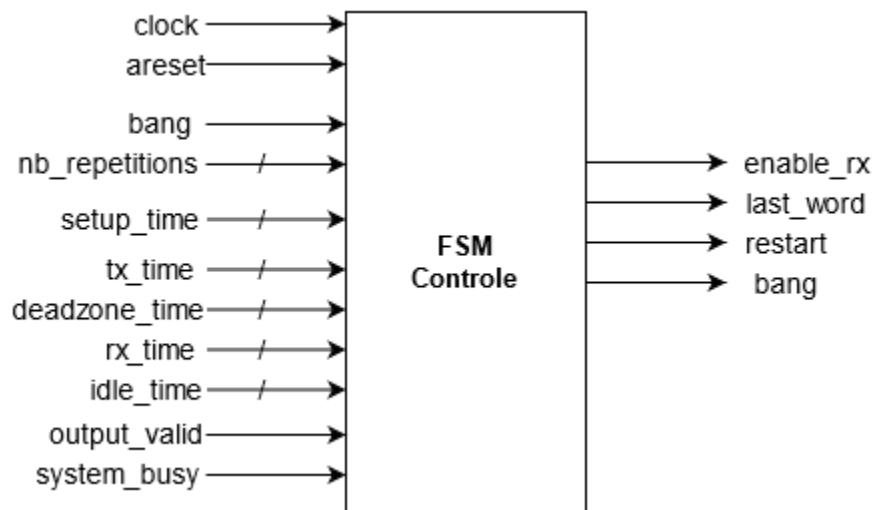


Interface do banco de registros

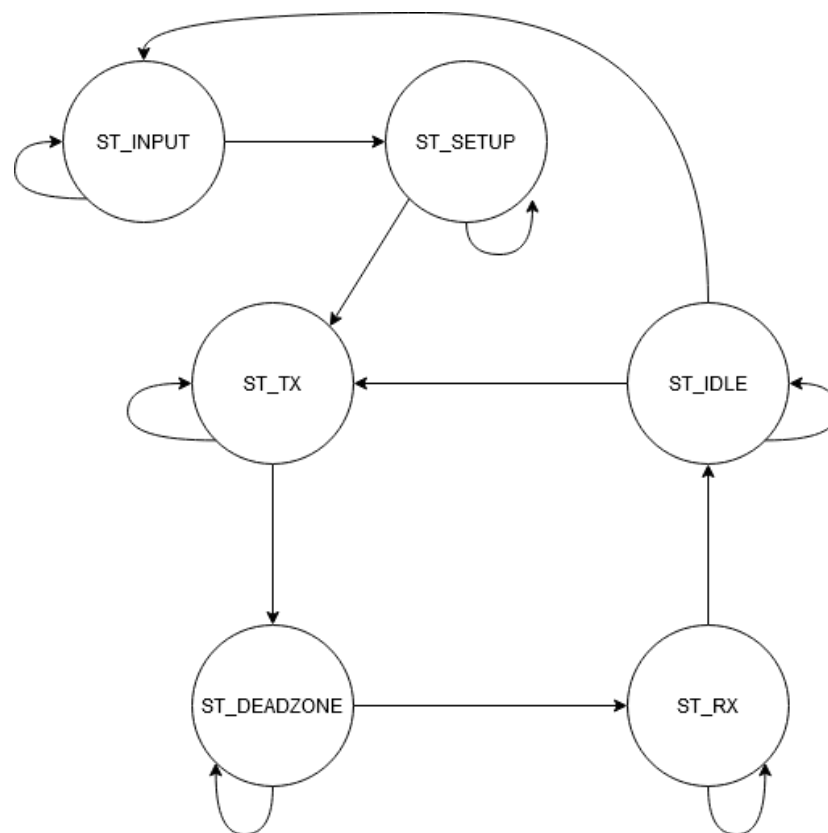


Arquitetura do banco de registros

Solução proposta



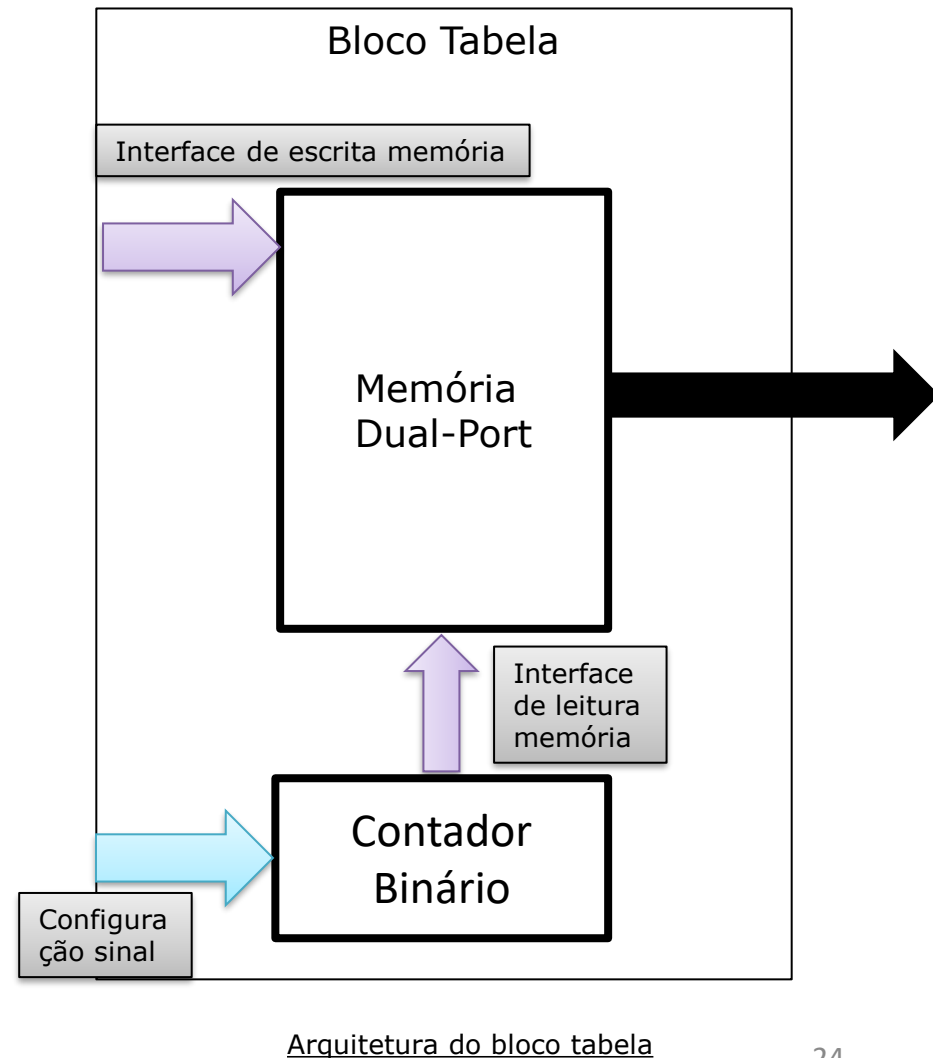
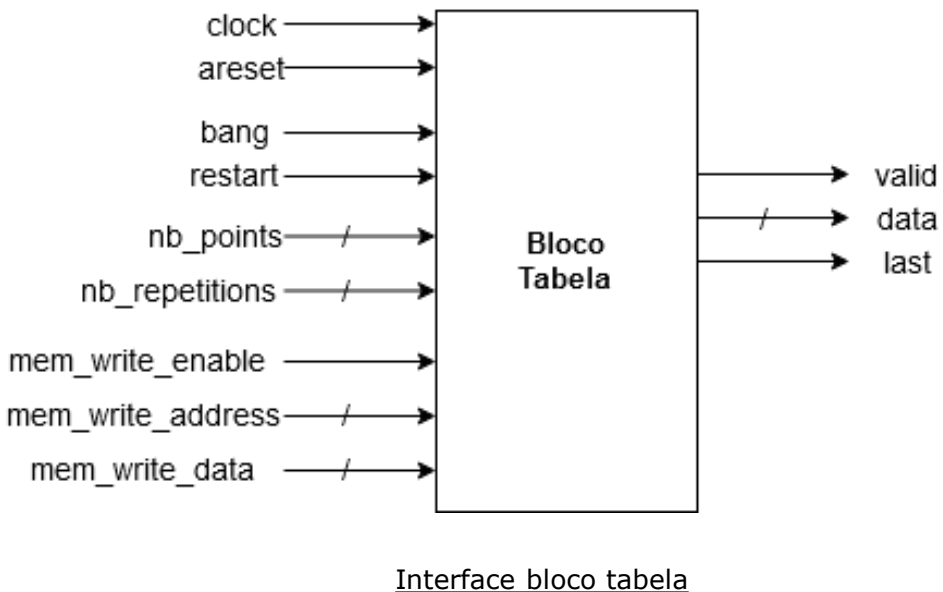
Interface da FSM de Controle



FSM da FSM de Controle

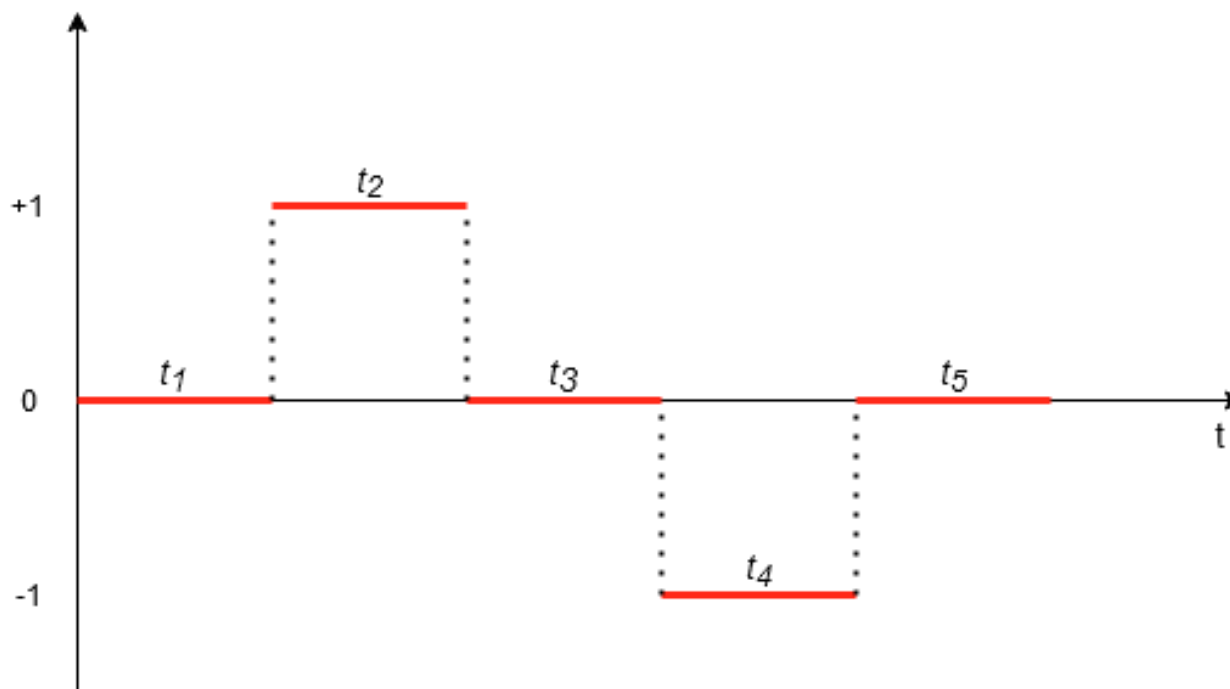
Solução proposta

Bloco Tabela



Solução proposta

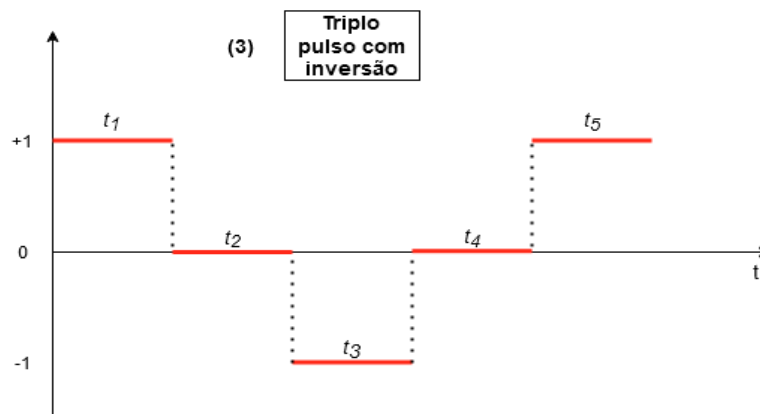
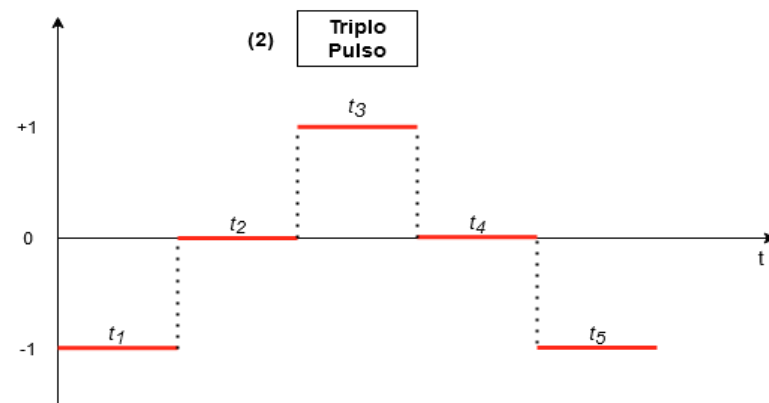
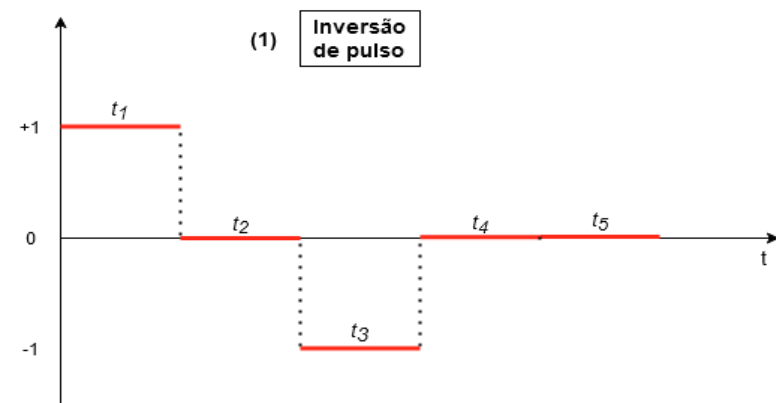
Pulser



Sinal quadrado

Solução proposta

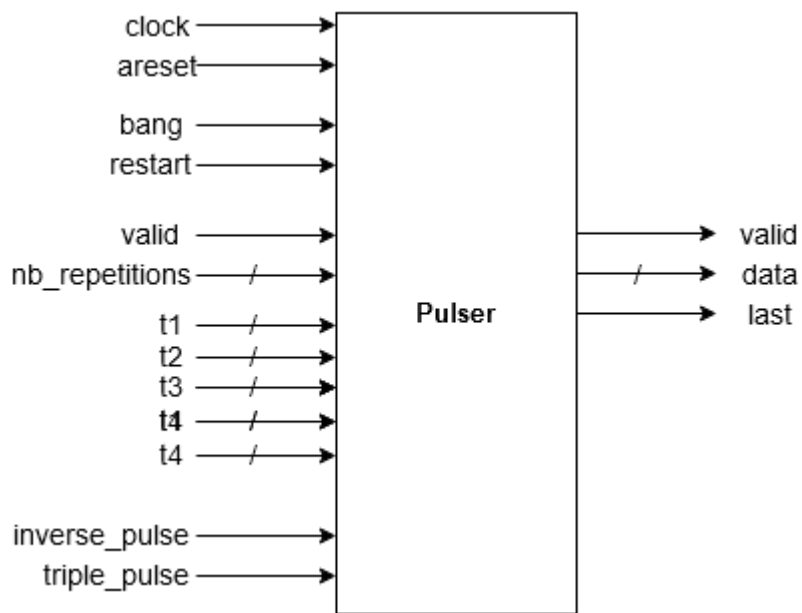
Pulser



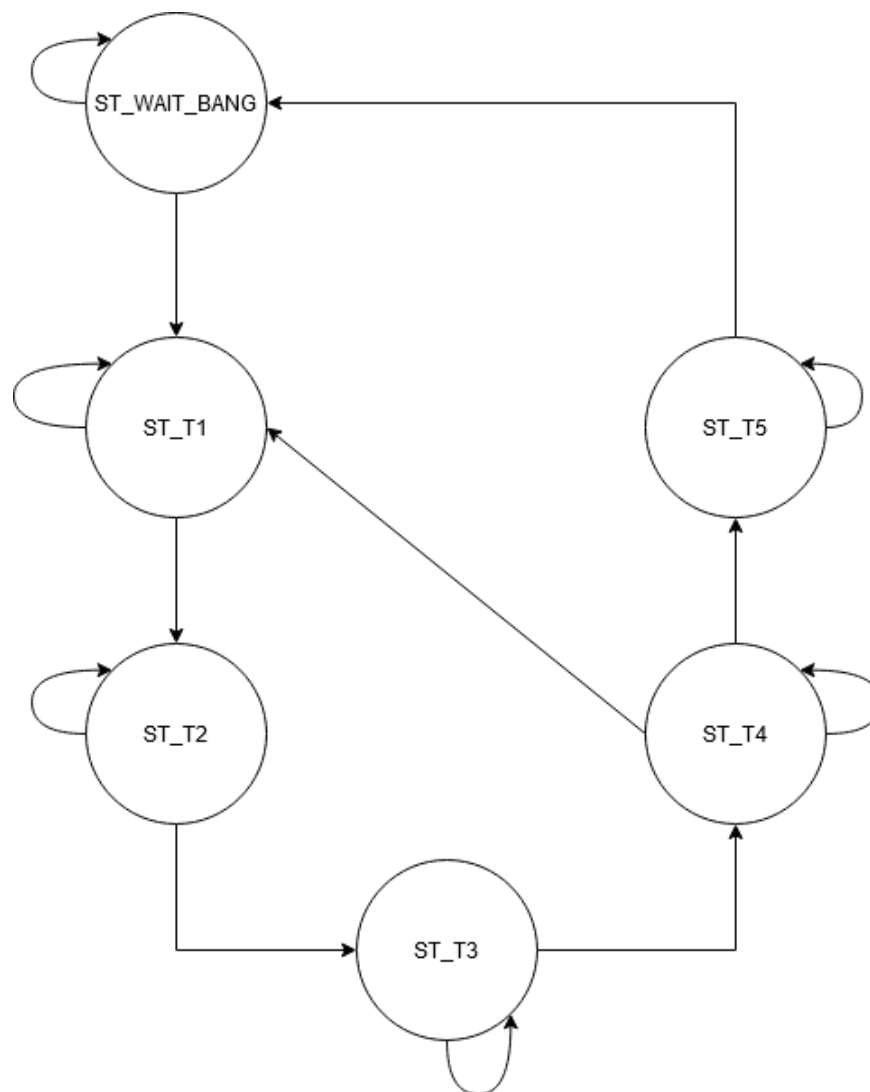
Modos de operação adicionais do Pulser

Solução proposta

Pulser



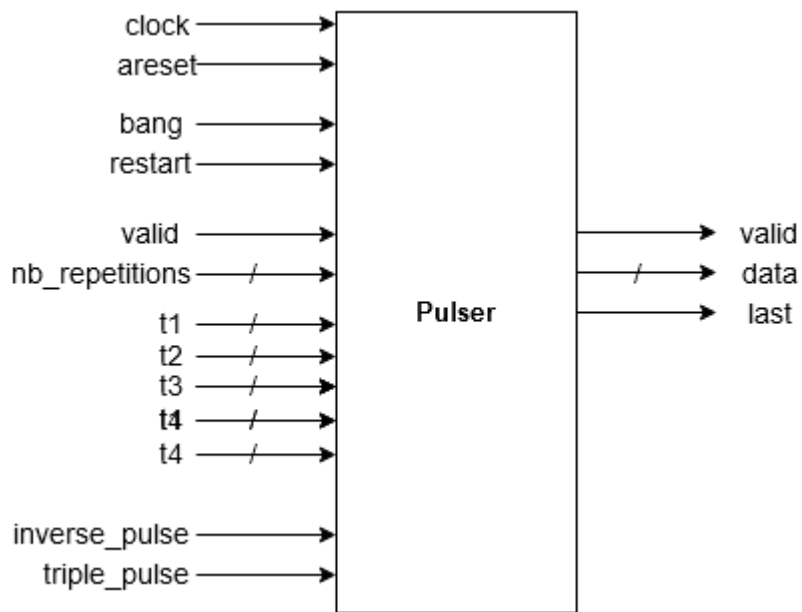
Interface do Pulser



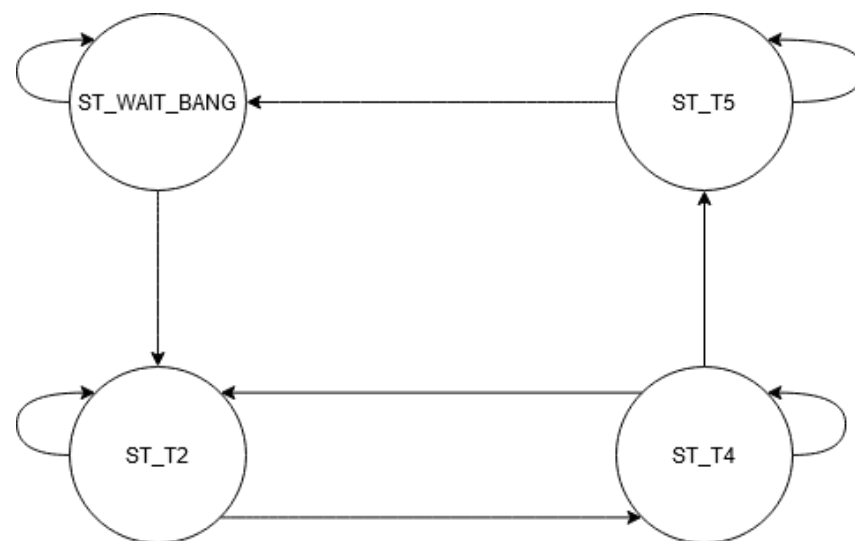
FSM do Pulser

Solução proposta

Pulser



Interface do Pulser



Exemplo FSM do Pulser. Tempos t1 e t3 nulos.

Solução proposta

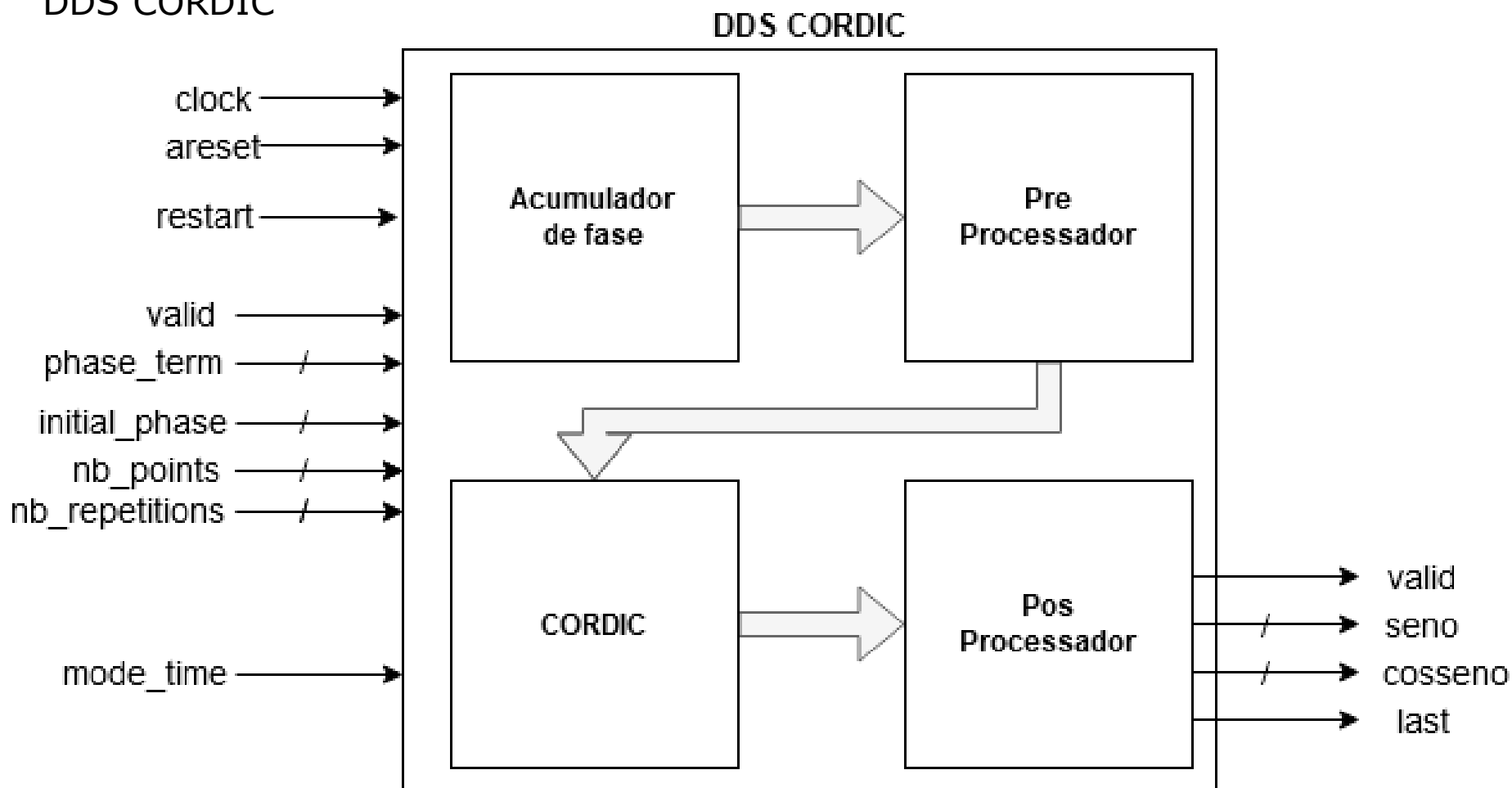
DDS CORDIC

Geração de senos e cossenos precisa de entidades de suporte:

- Acumulador de fase, para gerar fase de $[0; 2\pi]$;
- Pré-processador, corrigir fase para 1º e 4º quadrante;
- Algoritmo CORDIC;
- Pós-processador, para corrigir o sinal do seno e do cosseno (\pm).

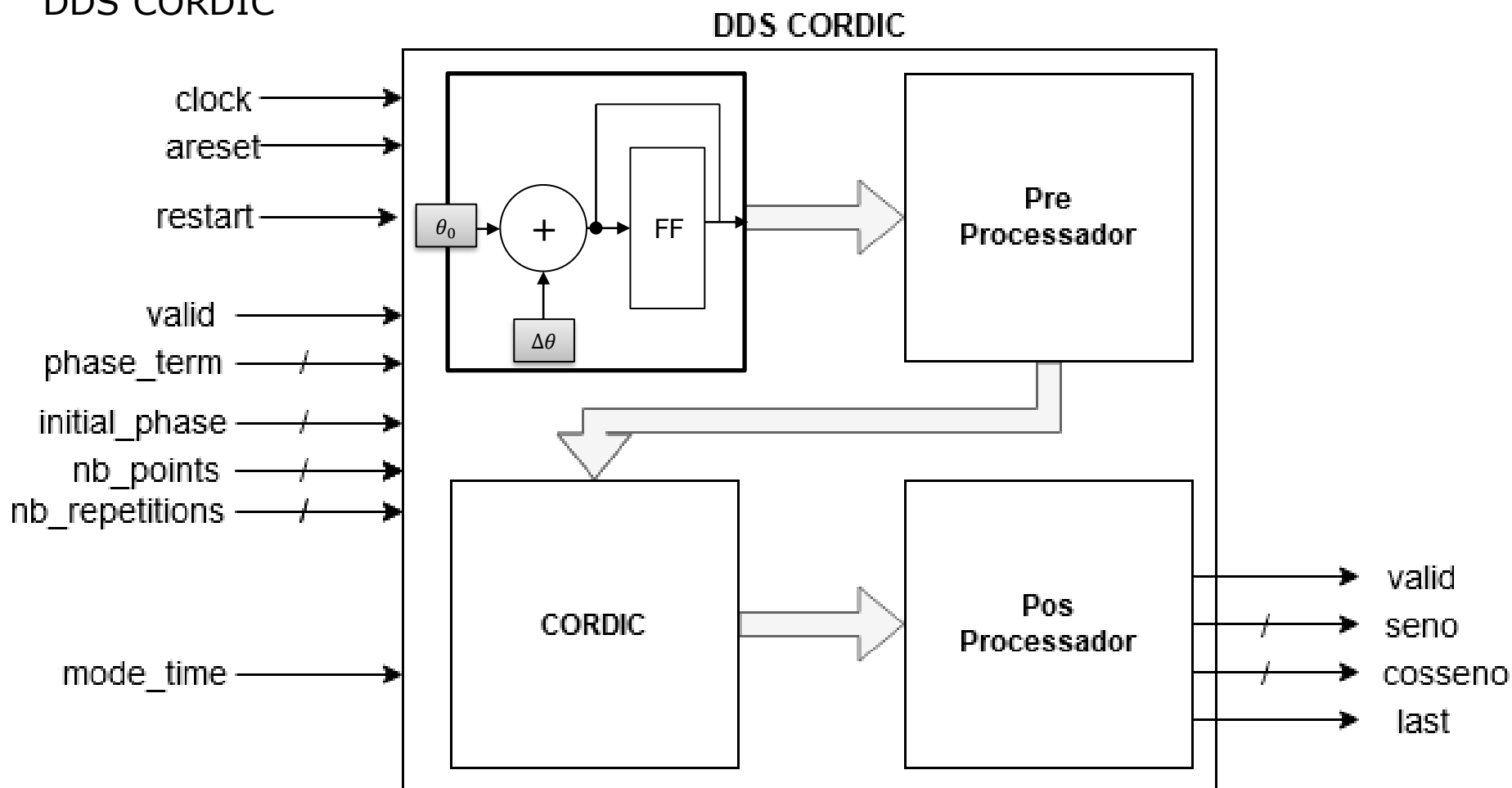
Solução proposta

DDS CORDIC



Solução proposta

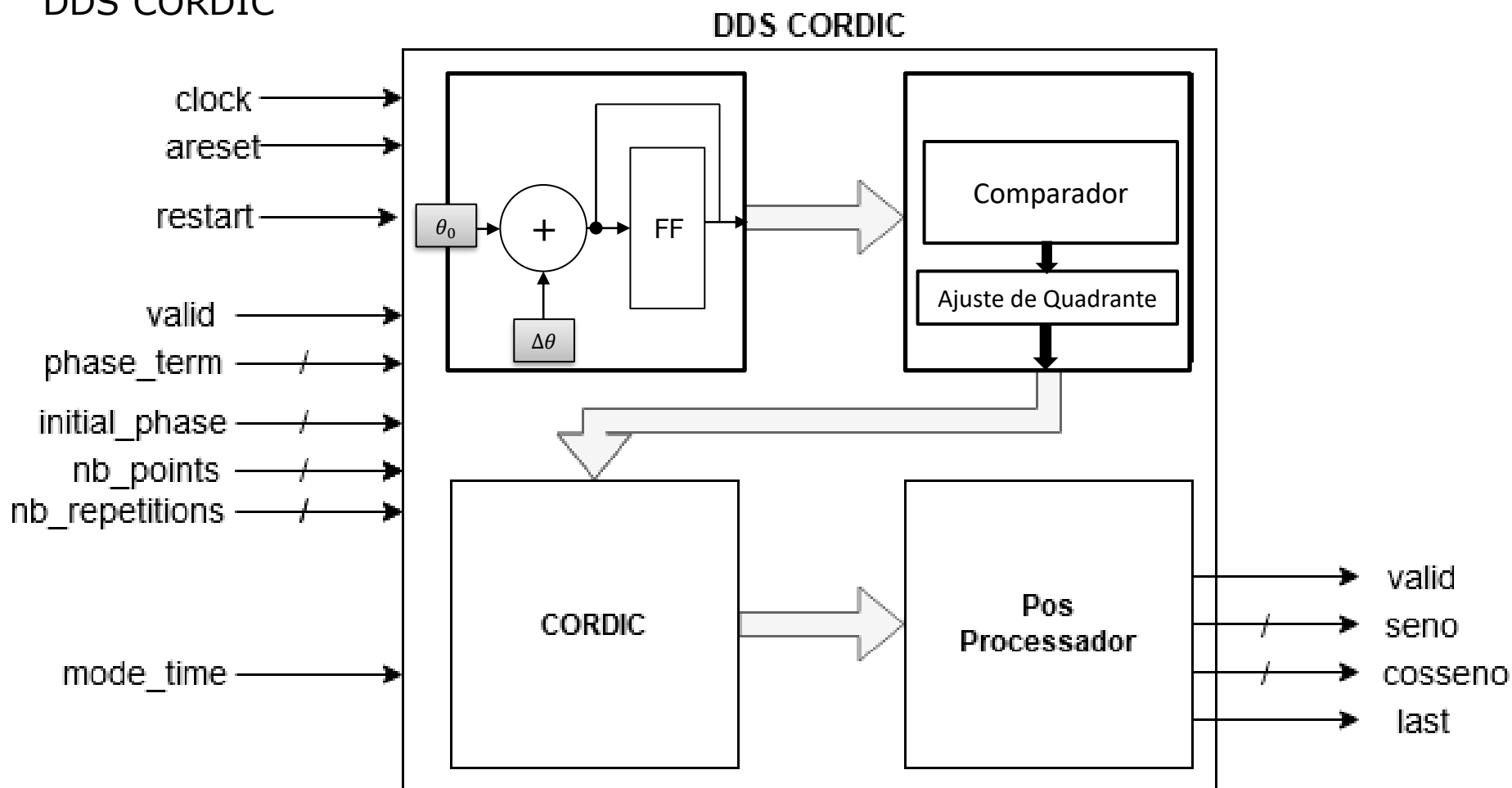
DDS CORDIC



Arquitetura do DDS CORDIC

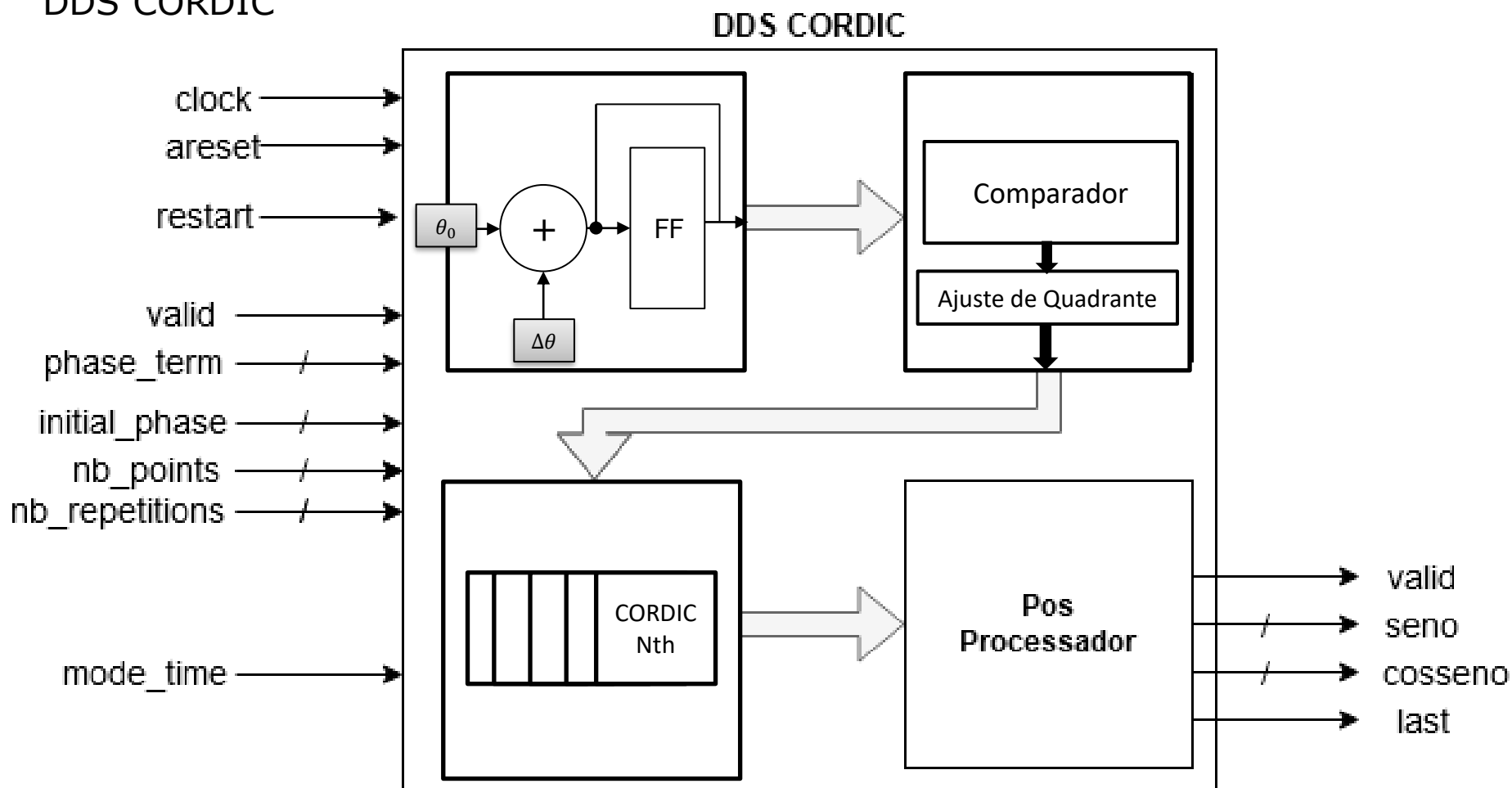
Solução proposta

DDS CORDIC



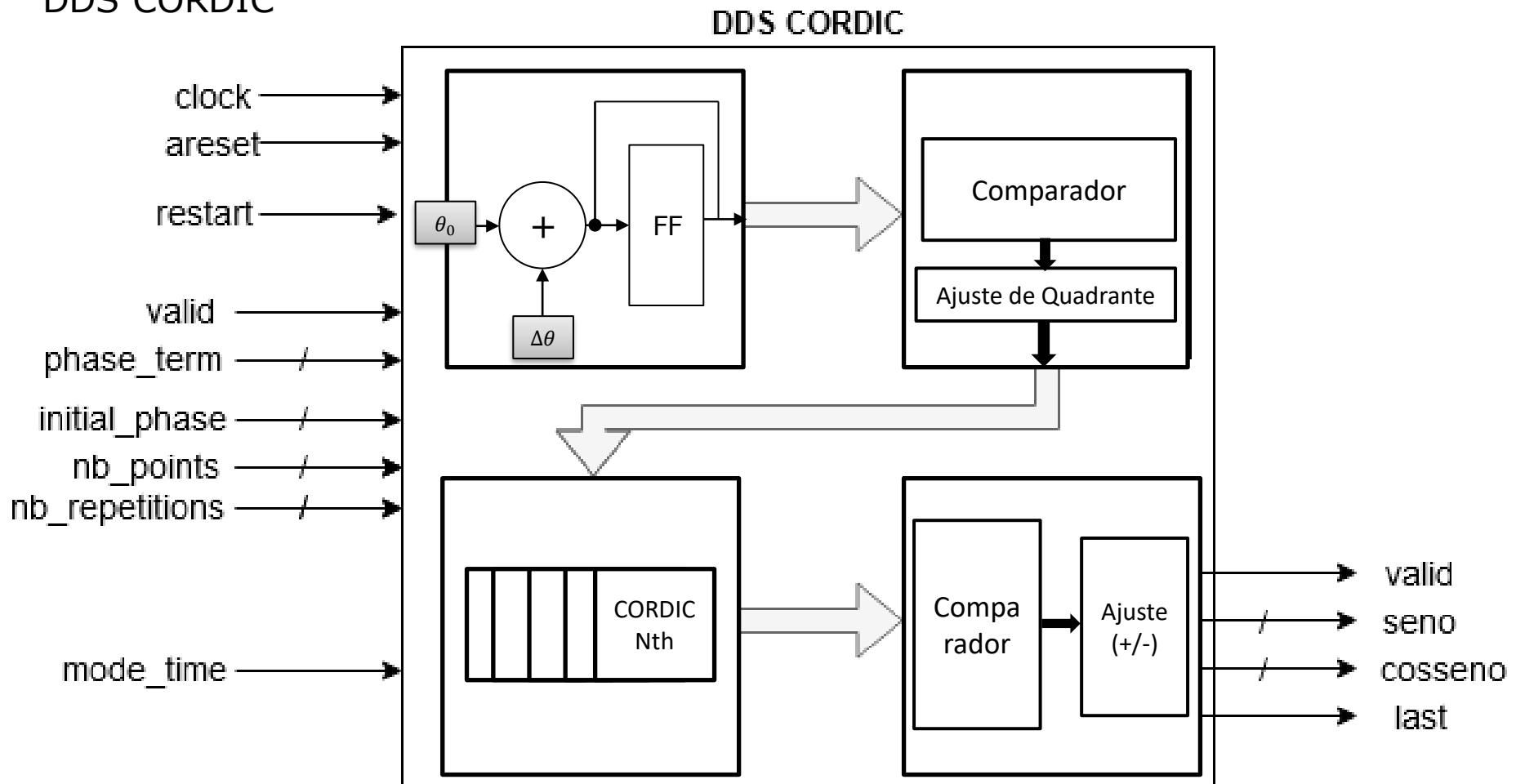
Solução proposta

DDS CORDIC



Solução proposta

DDS CORDIC

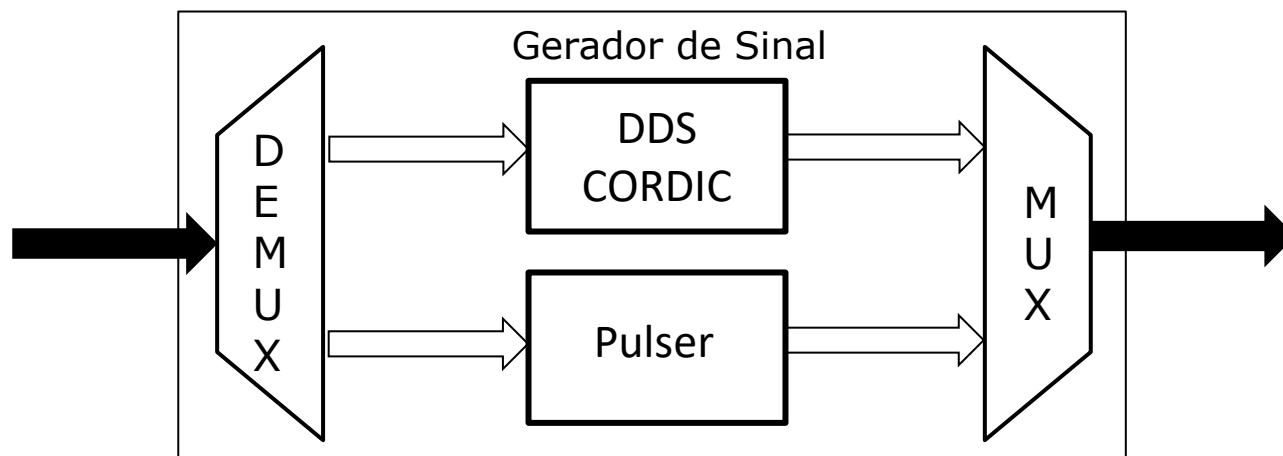


Arquitetura do DDS CORDIC

Solução proposta



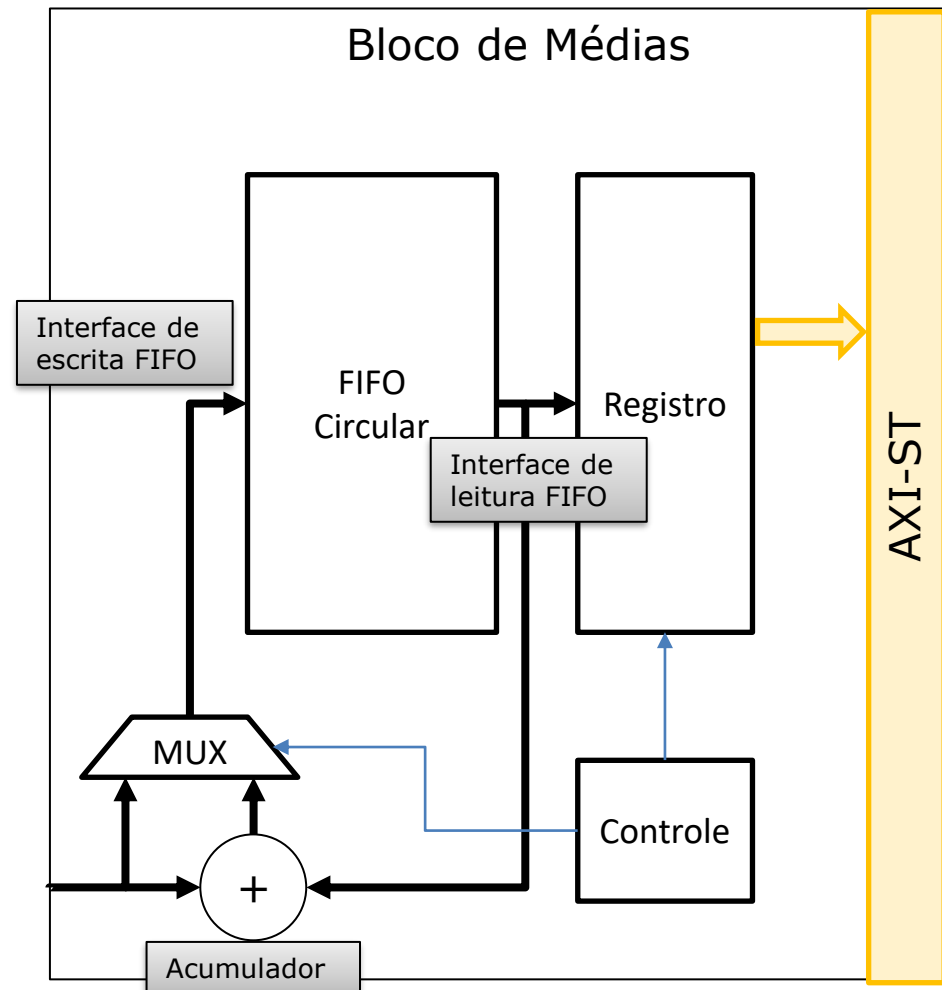
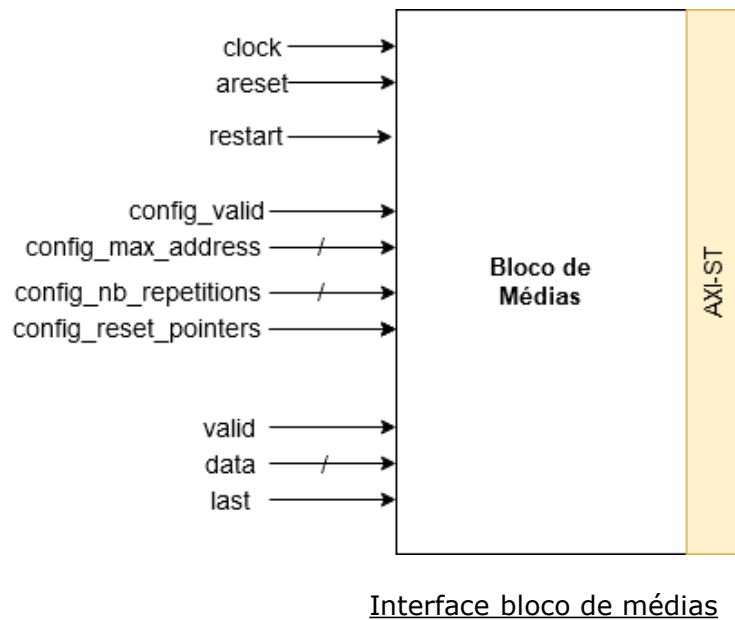
Interface do Gerador de Sinal



Arquitetura do Gerador de Sinal

Solução proposta

Bloco de Médias



Arquitetura do bloco médias

Janelas – Soma de cossenos

$$\text{Tukey : } \begin{cases} w[n] = \frac{1}{2} \left[1 - \cos\left(\frac{2\pi n}{\alpha L}\right) \right], & 0 \leq n < \frac{\alpha L}{2} \\ w[n] = 1, & \frac{\alpha L}{2} \leq n < \frac{N}{2} \\ w[N - n] = w[n], & 0 \leq n \leq \frac{N}{2} \end{cases}$$

Definição janela Tukey

$$\text{Hanning : } w[n] = a_0 - a_1 \cdot \cos\left(\frac{2\pi n}{N}\right),$$

$$a_1 = 1 - a_0 \text{ e } a_0 = 0,5$$

$$\text{Hamming : } w[n] = a_0 - a_1 \cdot \cos\left(\frac{2\pi n}{N}\right),$$

$$a_1 = 1 - a_0 \text{ e } a_0 = 0,53836$$

Definição janela Hanning e Hamming

$$\text{Blackman: } w[n] = a_0 - a_1 \cdot \cos\left(\frac{2\pi n}{N}\right) + a_2 \cdot \cos\left(\frac{4\pi n}{N}\right),$$

$$a_0 = \frac{1 - \alpha}{2}; a_1 = \frac{1}{2}; a_2 = \frac{\alpha}{2}$$

Definição janela Blackman

Blackman-Harris:

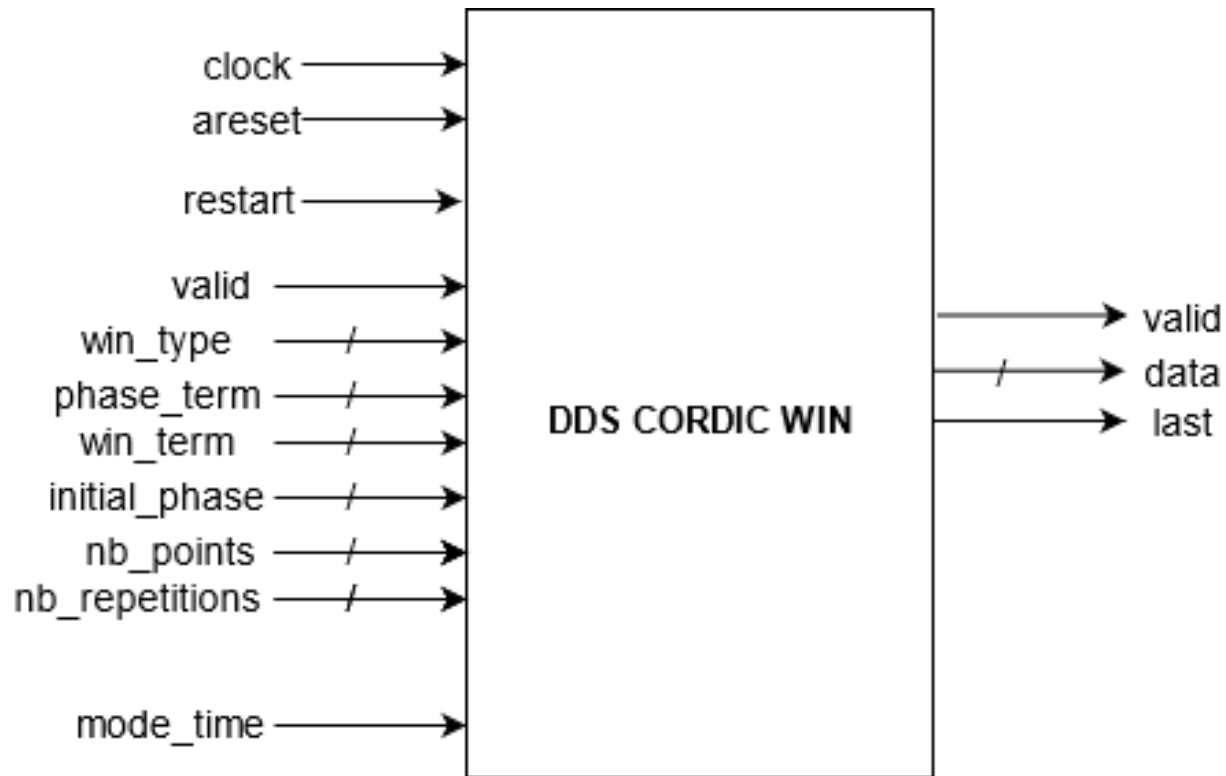
$$w[n] = a_0 - a_1 \cdot \cos\left(\frac{2\pi n}{N}\right) + a_2 \cdot \cos\left(\frac{4\pi n}{N}\right) + a_3 \cdot \cos\left(\frac{6\pi n}{N}\right),$$

$$a_0 = 0,3635819; a_1 = 0,4891775;$$

$$a_2 = 0,1365995; a_3 = 0,0106411$$

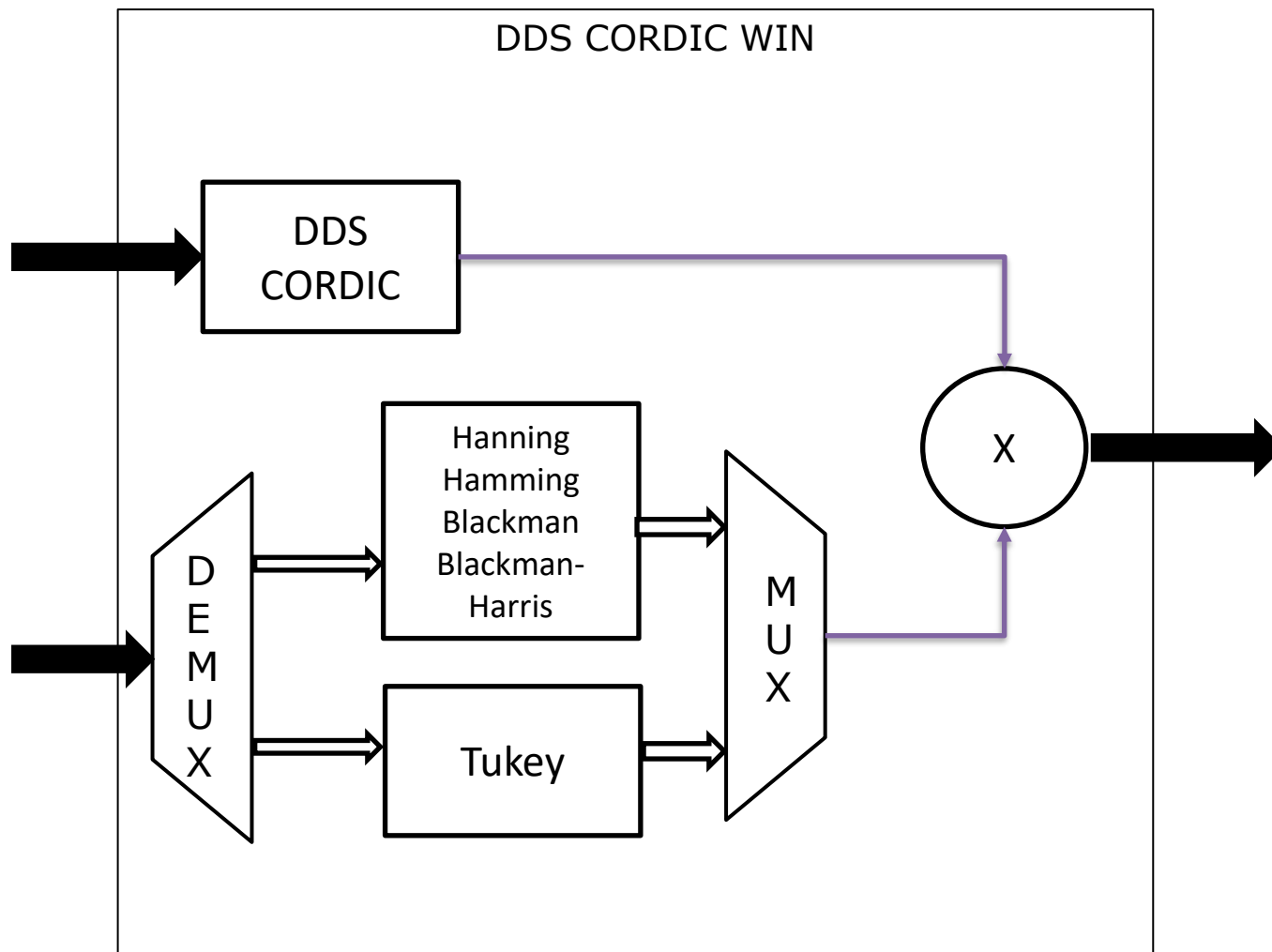
Definição janela Blackman-Harris

Janelas – Soma de cossenos



Interfaces DDS CORDIC WIN

Janelas – Soma de cossenos

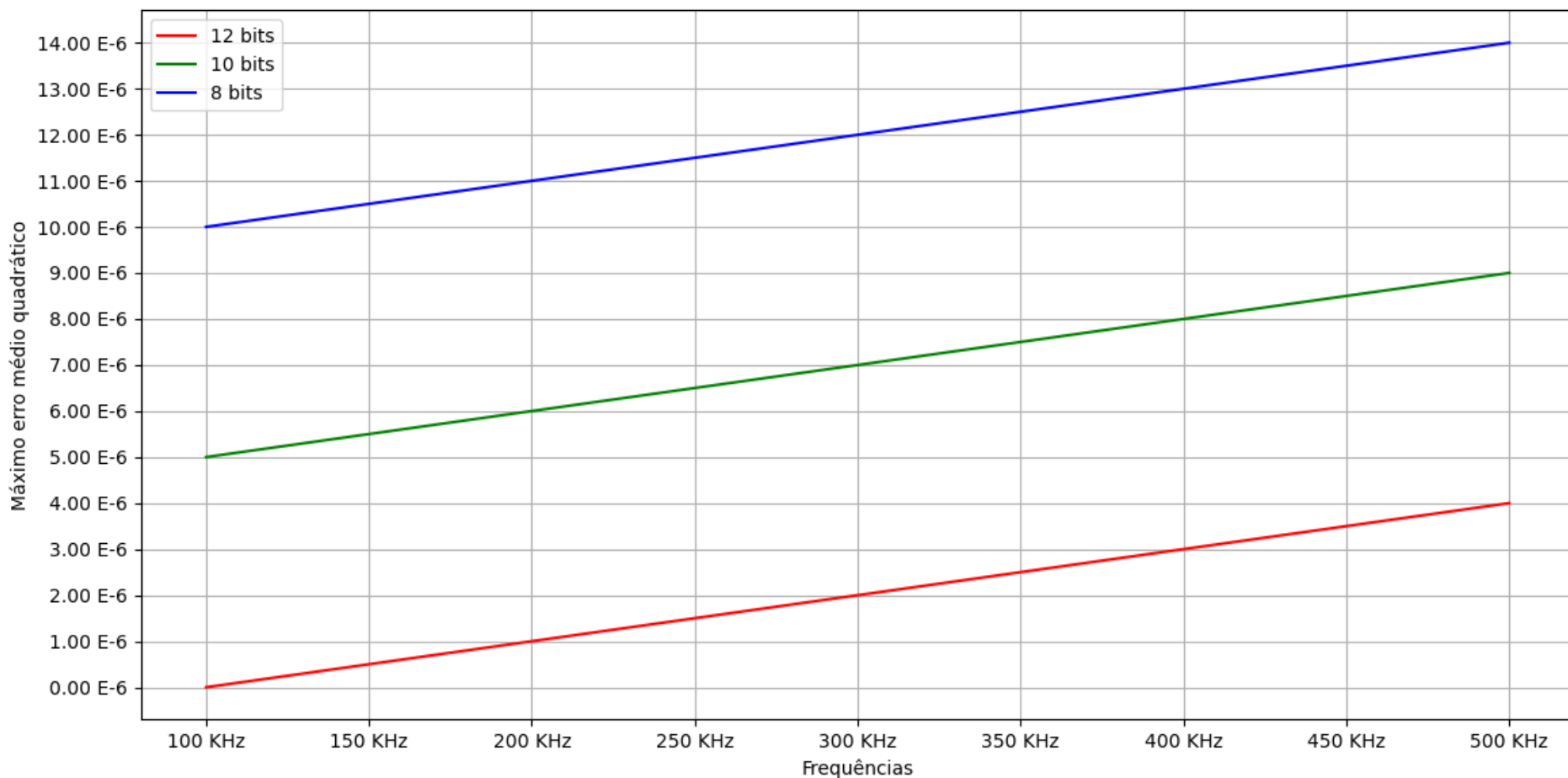


Testes

Metodologia:

- Construção de *testbenchs* em VHDL antes das entidades;
- Comparação com modelo em Python;
- Modelagem das operações aritméticas com ponto-fixo, considerando números entre $[-1; +1]$.

Erro CORDIC



Erro médio quadrático máximo, para 8, 10 e 12 bits

Solução proposta – Banco de registros



Opções:

- Mapeamento de registros;
- Mapeamento de array de registros;
- Mapeamento de interface memória;

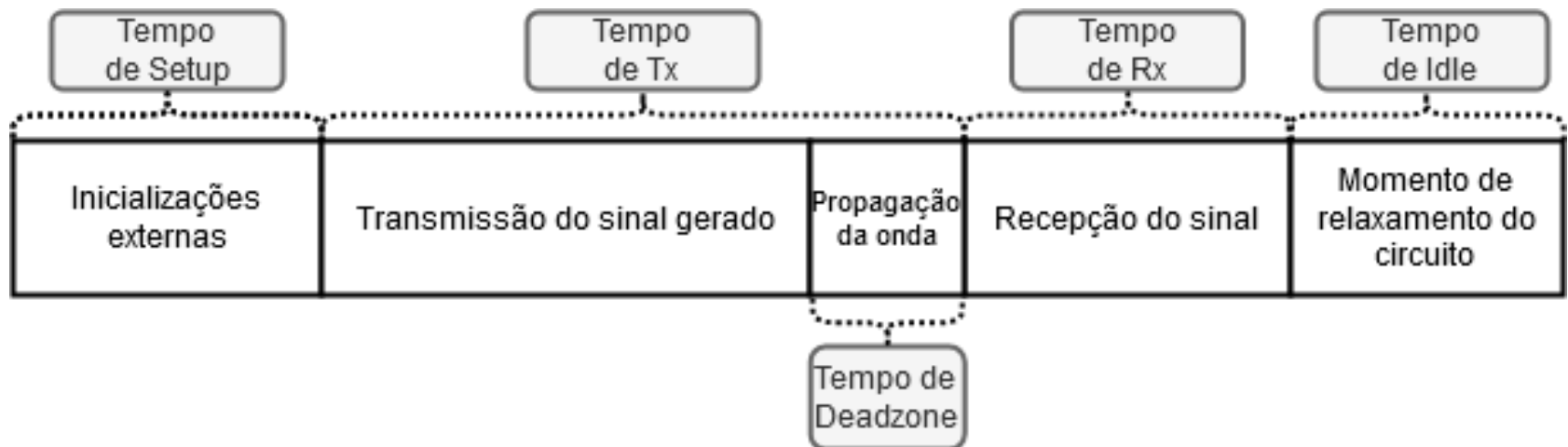
Solução proposta – Banco de registros

Offset	Nome	Descrição	Tamanho	Acesso	Reset
0x0	version	Número da versão do código. Usado para controle de versão.	8 bits	READ_ONLY	0x0
0x4	bang	Botão de "start", com auto-reset, imitando um tick.	1 bit	WRITE_ONLY	0x0
0x8	sample_frequency	Frequência de amostragem. Valor predefinido de 100 MHz.	27 bits	READ_WRITE	0x5F5E100
0xC	wave_nb_periods	Número de períodos do sinal a ser gerado.	8 bits	READ_WRITE	0x0
0x10	wave_nb_points	Número de pontos em um período. Usado no DDS CORDIC.	32 bits	READ_WRITE	0x0
0x14	wave_config	Seletor do sinal desejado. "0" -> DDS CORDIC ; "1" -> <i>pulser</i>	1 bit	READ_WRITE	0x0
0x24	fsm_nb_repetitions	Número de repetições do sinal a ser gerado, controlado pela FSM.	6 bit	READ_WRITE	0x0
0x28	fsm_setup_timer	Número de ciclos na zona Setup	18 bits	READ_WRITE	0x0
0x2C	fsm_tx_timer	Número de ciclos na zona Tx	18 bits	READ_WRITE	0x0
0x30	fsm_deadzone_timer	Número de ciclos na zona Deadzone	18 bits	READ_WRITE	0x0
0x34	fsm_rx_timer	Número de ciclos na zona Rx	18 bits	READ_WRITE	0x0
0x38	fsm_idle_timer	Número de ciclos na zona Idle	18 bits	READ_WRITE	0x0
0x3C	<i>pulser_t1</i>	Tamanho do degrau em t1	10 bits	READ_WRITE	0x0
0x40	<i>pulser_t2</i>	Tamanho do degrau em t2	10 bits	READ_WRITE	0x0
0x44	<i>pulser_t3</i>	Tamanho do degrau em t3	10 bits	READ_WRITE	0x0
0x48	<i>pulser_t4</i>	Tamanho do degrau em t4	10 bits	READ_WRITE	0x0
0x4C	<i>pulser_t5</i>	Tamanho do degrau em t5	10 bits	READ_WRITE	0x0
0x50	<i>pulser_config</i>	Sinal de configuração para as opções de pulso invertido e pulso triplo. [0] <i>inverted_pulse</i> [1] <i>triple_pulse</i>	2 bits	READ_WRITE	0x0
0x54	dds_phase_term	DDS CORDIC phase term	32 bits	READ_WRITE	0x0
0x58	dds_nb_points	Número de pontos em 1 período do seno	18 bits	READ_WRITE	0x0
0x5C	dds_init_phase	DDS CORDIC initial phase	32 bits	READ_WRITE	0x0
0x60	dds_mode	DDS CORDIC opção mode_time	1 bit	READ_WRITE	0x0



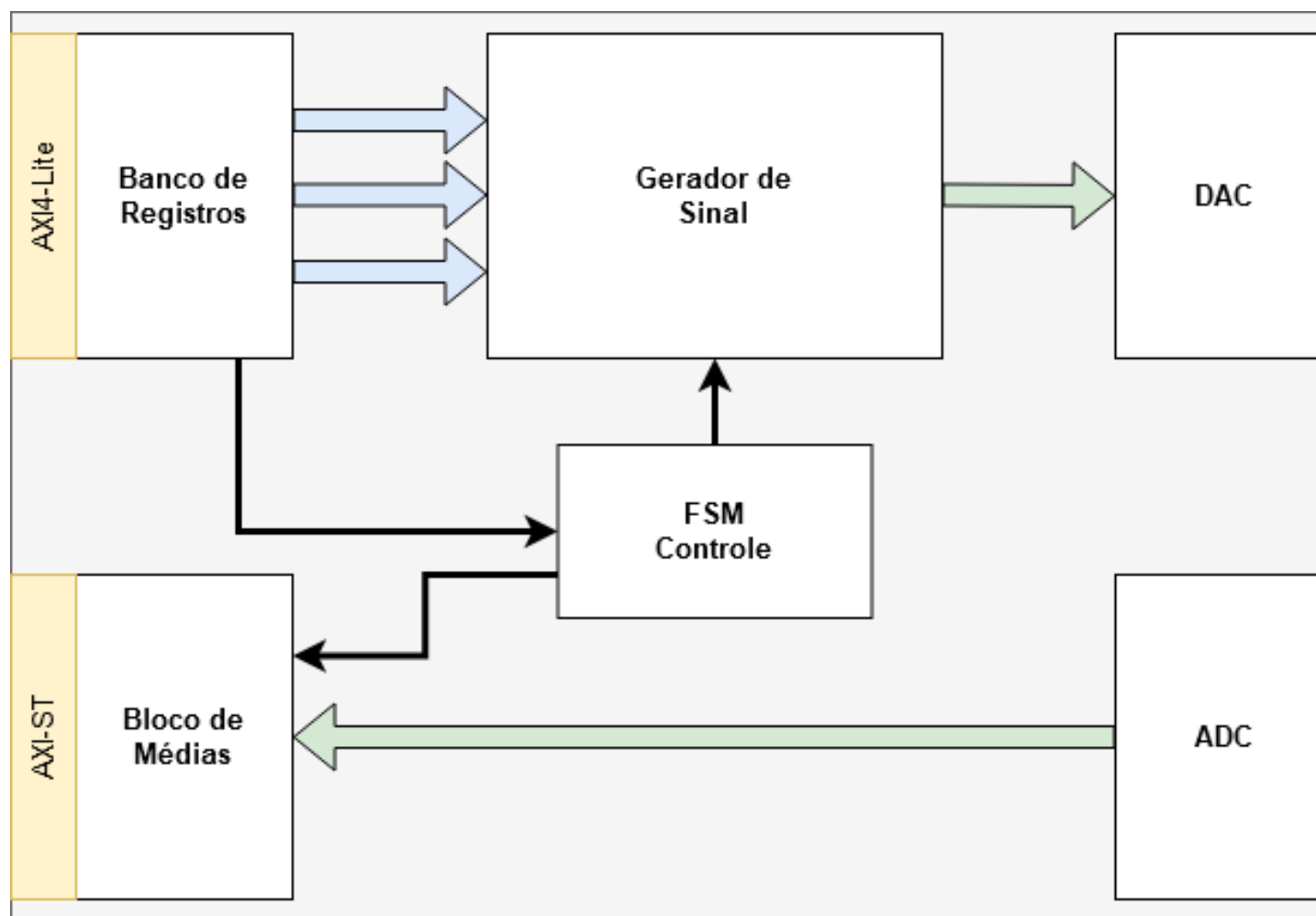
Solução proposta

Tempos de Controle



Cinco tempos de controle, aplicados na FSM de Controle

Demonstração

Solução proposta – Diagrama de blocosDiagrama de blocos da solução proposta

Arquitetura discutida

Recurso	Utilização	Taxa de utilização dos recursos da FPGA
LUT	1978	3.72 %
FF	1436	1.35 %
BRAM	30	21.43 %

- 100 MHz;
- Amostras com 10 bits;
- Máximo de 64K (65536) amostras.

Resultado da implementação na Zybo Z2-70, da solução completa

Trabalhos Futuros:

- Teste em carta;
- Melhora precisão e reposta do CORDIC;
- Implementação de janelas;
- Implementação das funções de interpolação e decimação;

Recapitulação:

1. Construção de uma plataforma para geração de sinais de excitação para testes em EMAT;
2. Solução configurável pelo usuário;
3. Processamento em tempo real;

Conclusão:

1. A solução foi criada, mas a plataforma carece da conexão em PS e PL.
2. A solução é genérica;
3. A FPGA possibilita o processamento em tempo real.

- [1] - "Purpose and Internal Functionality of FPGA Look-Up Tables", <https://www.allaboutcircuits.com/technical-articles/purpose-and-internal-functionality-of-fpga-look-up-tables/>, acessado em 20 de Dezembro de 2020 às 10h00.
- [2] - "Zybo Z7Board Reference Manual", https://reference.digilentinc.com/_media/reference/programmable-logic/zybo-z7/zybo-z7_rm.pdf, acessado em 20 de Dezembro de 2020 às 10h00.
- [3] - "AMBA AXI Protocol", <https://referencedesigner.com/blog/amba-axi-protocol/2567/>, acessado em 20 de Dezembro de 2020 às 10h10.
- [4] - "airhdl", <https://airhdl.com/index.jsp>, acessado em 20 de Dezembro de 2020 às 10h10.
- [5] - LogiCORE, I. P. "FIR Compiler v5. 0, Xilinx." *Inc., San Jose, CA, USA* (2010).
- [6] - "scopefir", <https://iowegian.com/scopefir/>, acessado em 21 de Dezembro de 2020 às 09h25.

Obrigado pela atenção!

Upsampler e Downsampler

Interpolação (Aumento da frequência de amostragem):

1. Para um aumento de M vezes (M inteiro);
2. Inserir $M-1$ zeros, a cada M amostras;
3. Filtrar o resultado com um FIR;

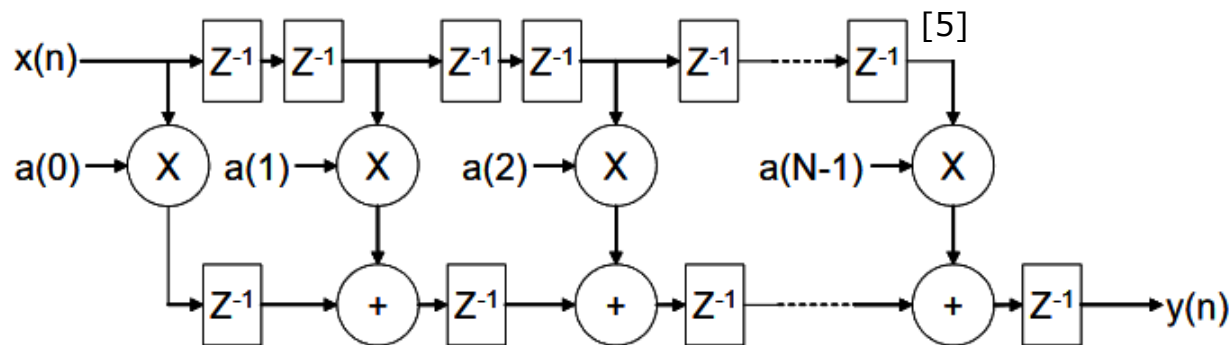
Decimação (Redução da frequência de amostragem):

1. Para uma redução de P vezes (P inteiro);
2. Filtrar o sinal com um FIR;
3. Retirar $P-1$ amostras, a cada P amostras;

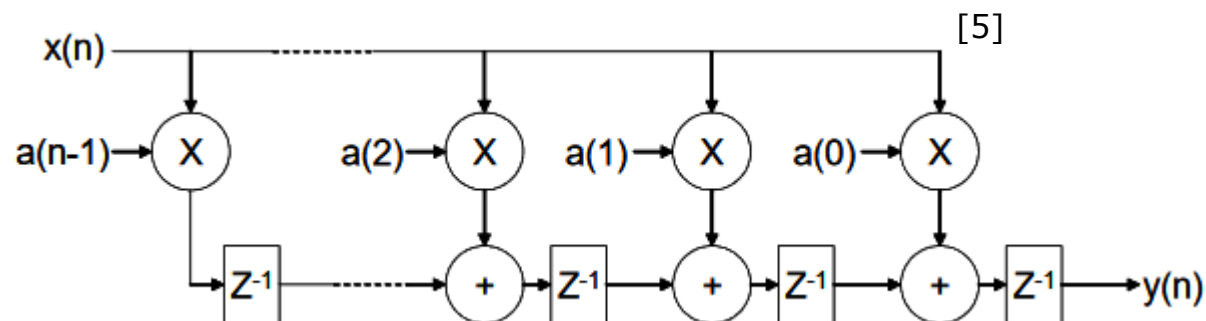
FIR (Finite Impulse Response)

$$y[n] = \sum_{k=0}^{N_{taps}-1} coef[k].x[n-k]$$

$$a[n] = coef[n]$$



Arquitetura FIR direto

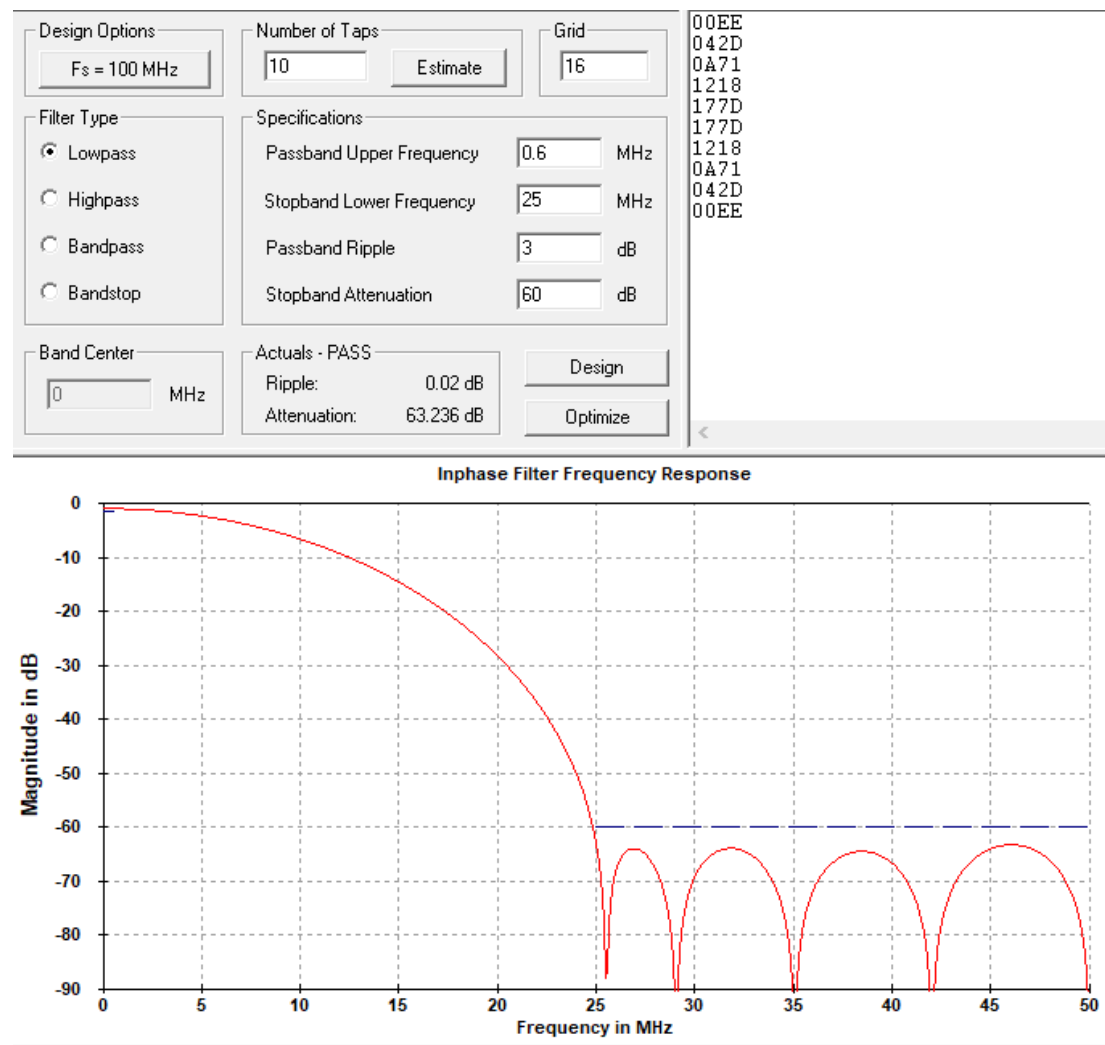


Arquitetura FIR transposto

Coeficientes do FIR

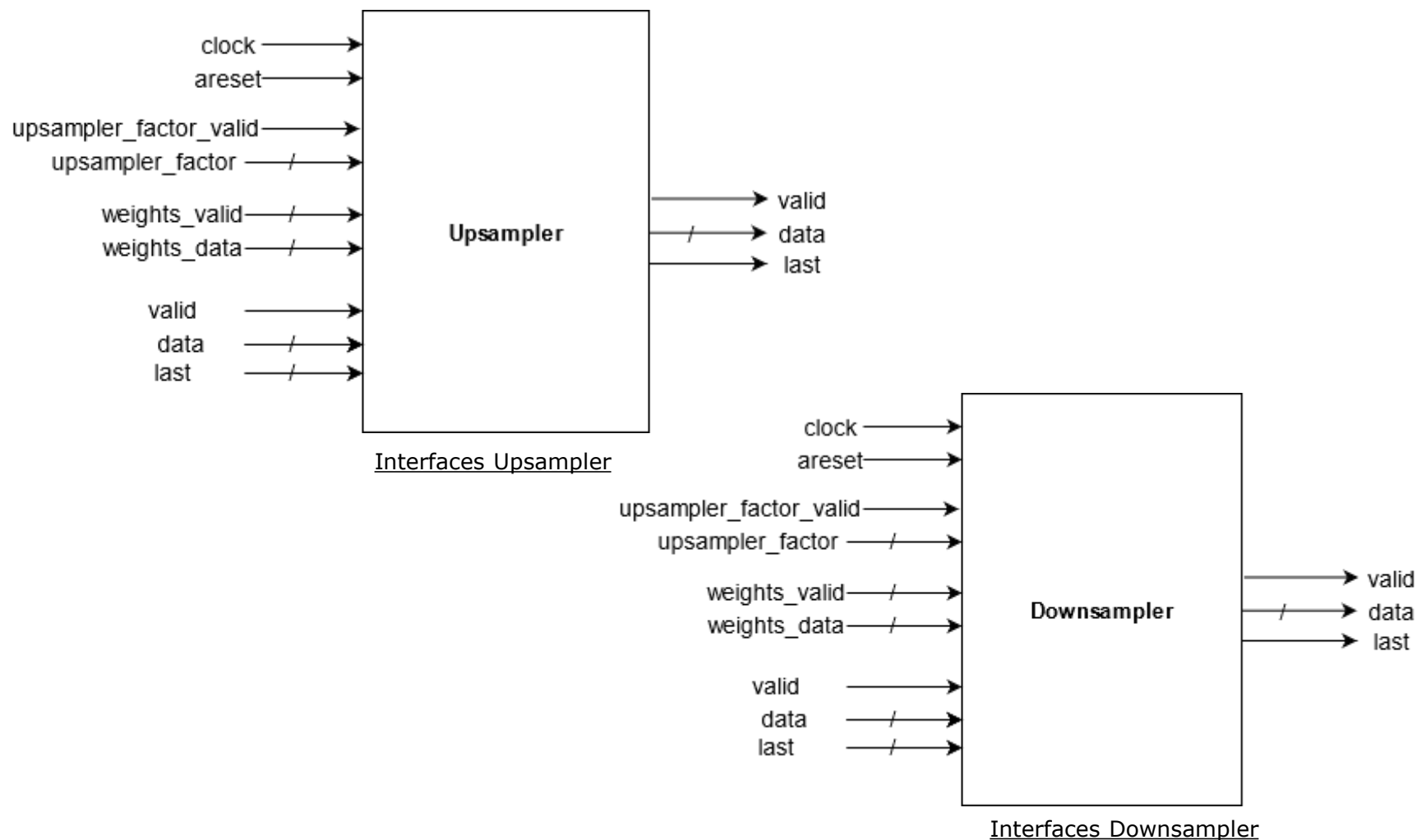
[6]

ScopeFIR

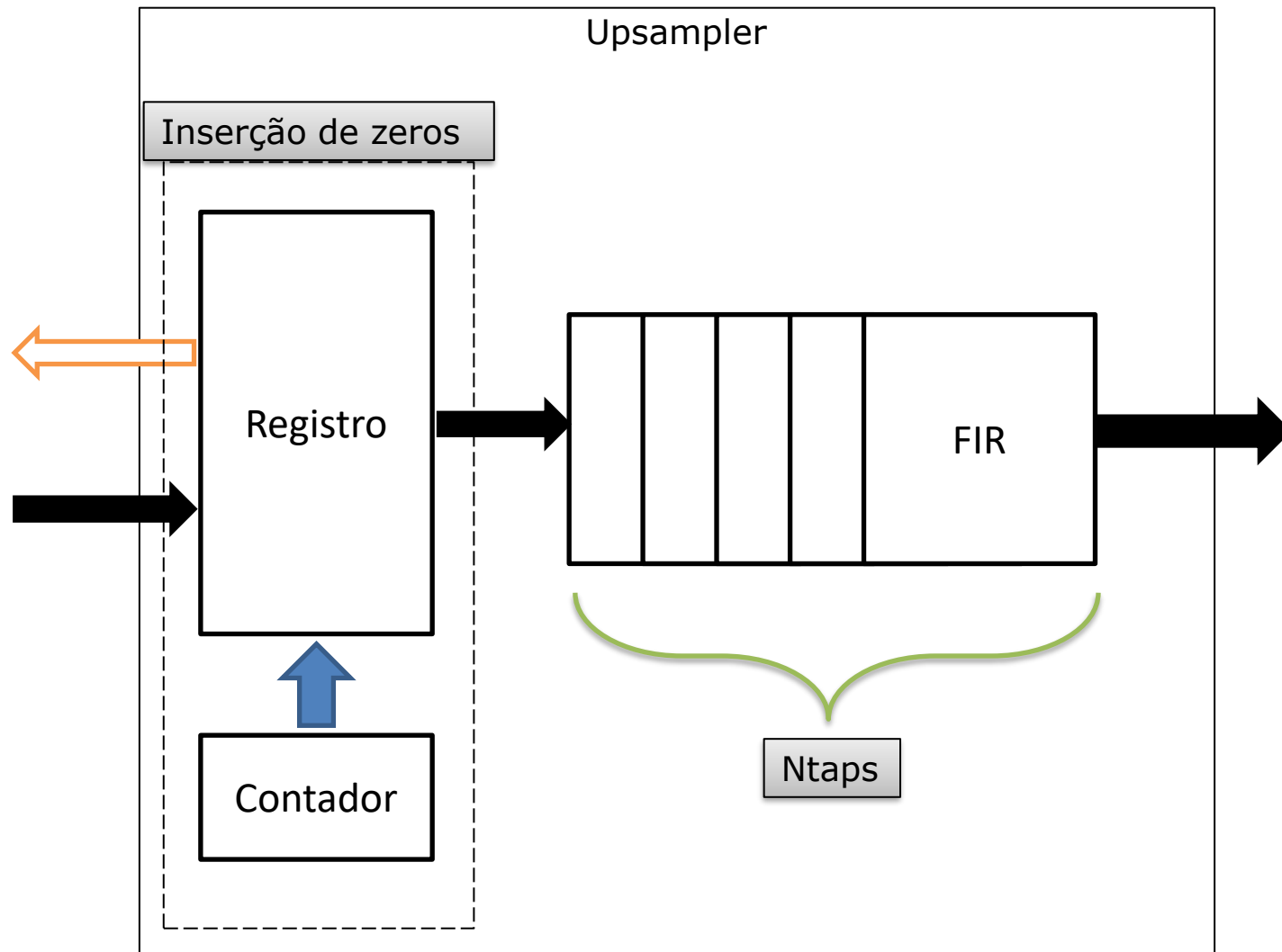


Cálculo coeficientes de um FIR, usando o ScopeFIR

Upsampler e Downsampler

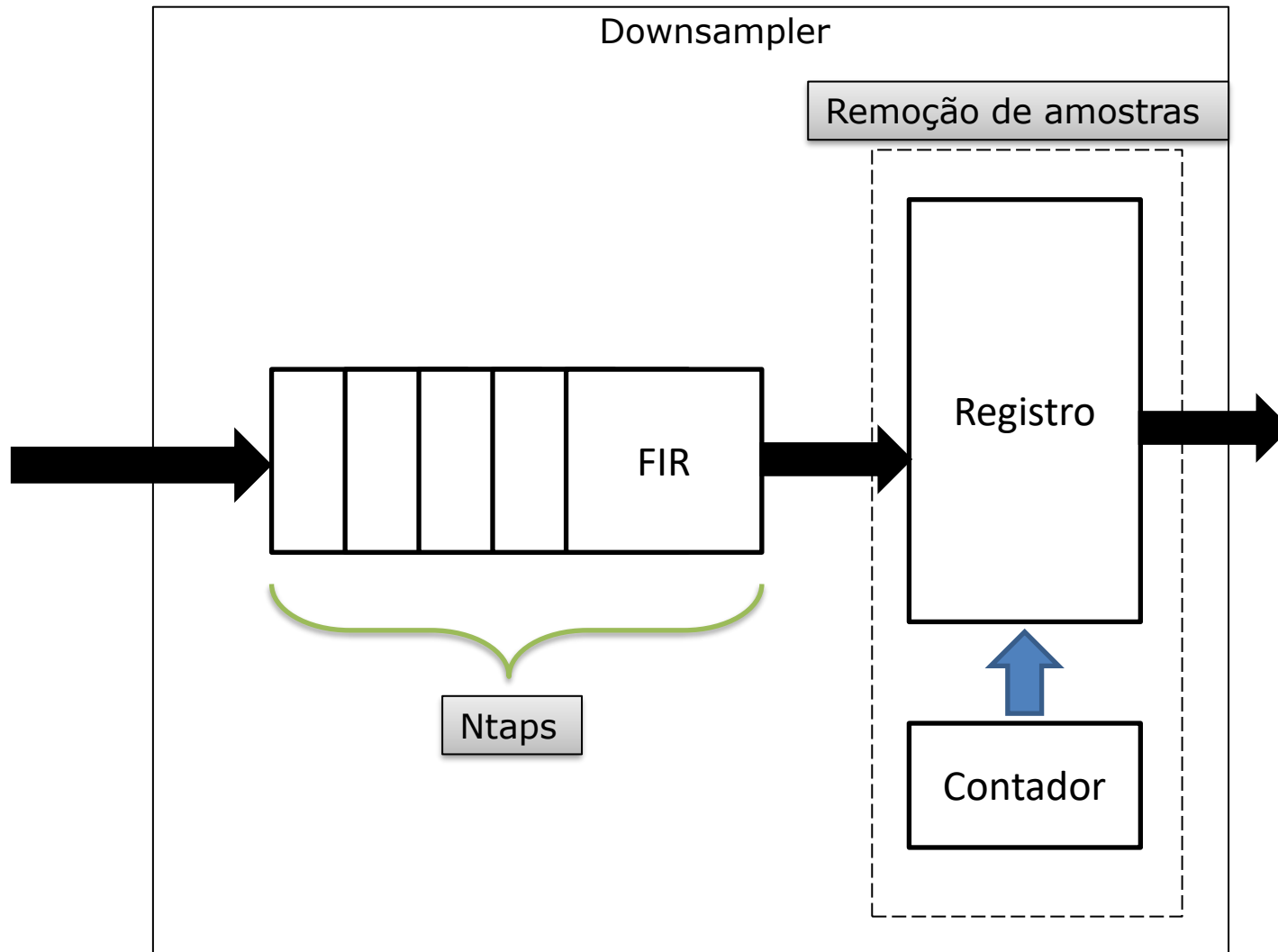


Upsampler



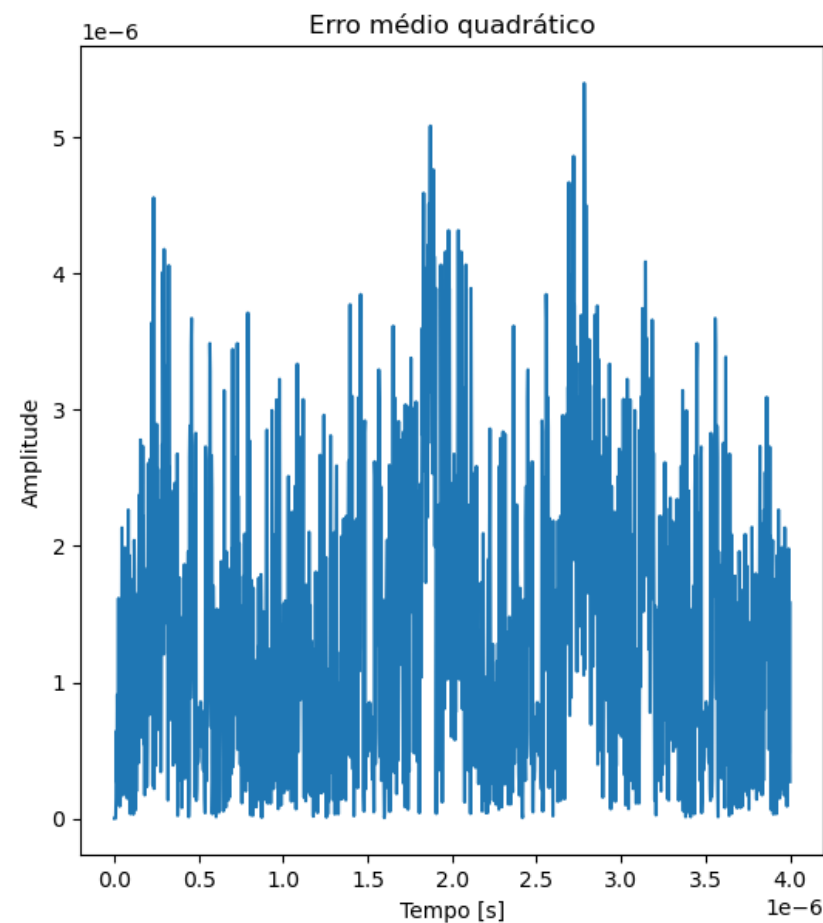
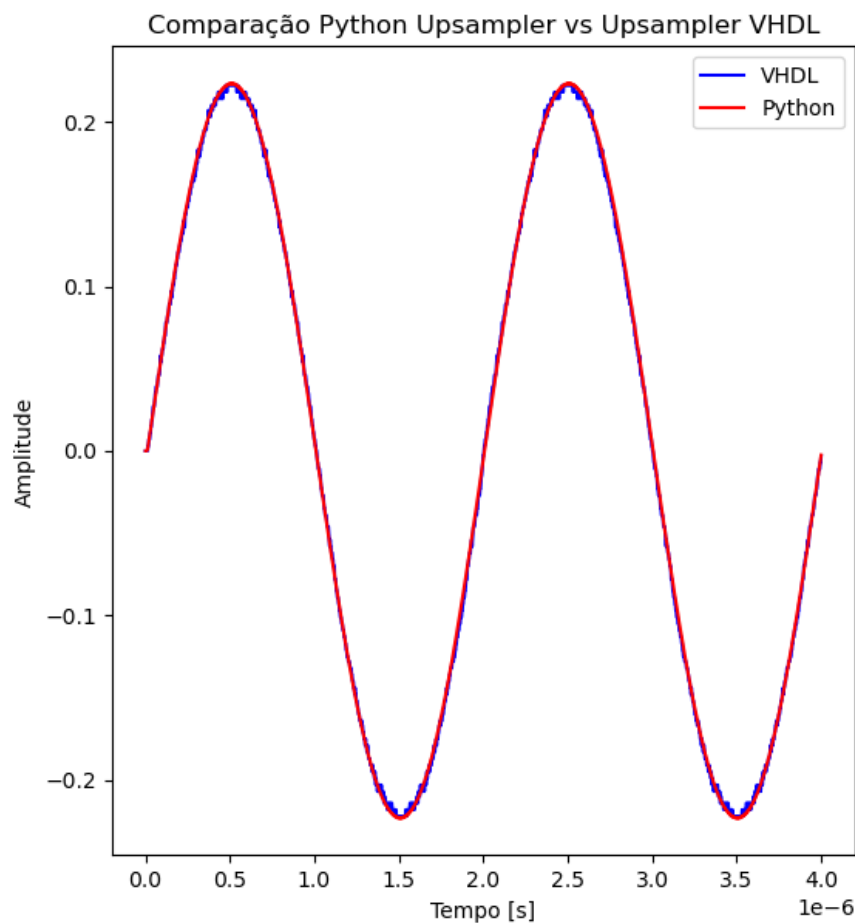
Arquitetura Upsampler

Downsampler



Arquitetura Downsampler

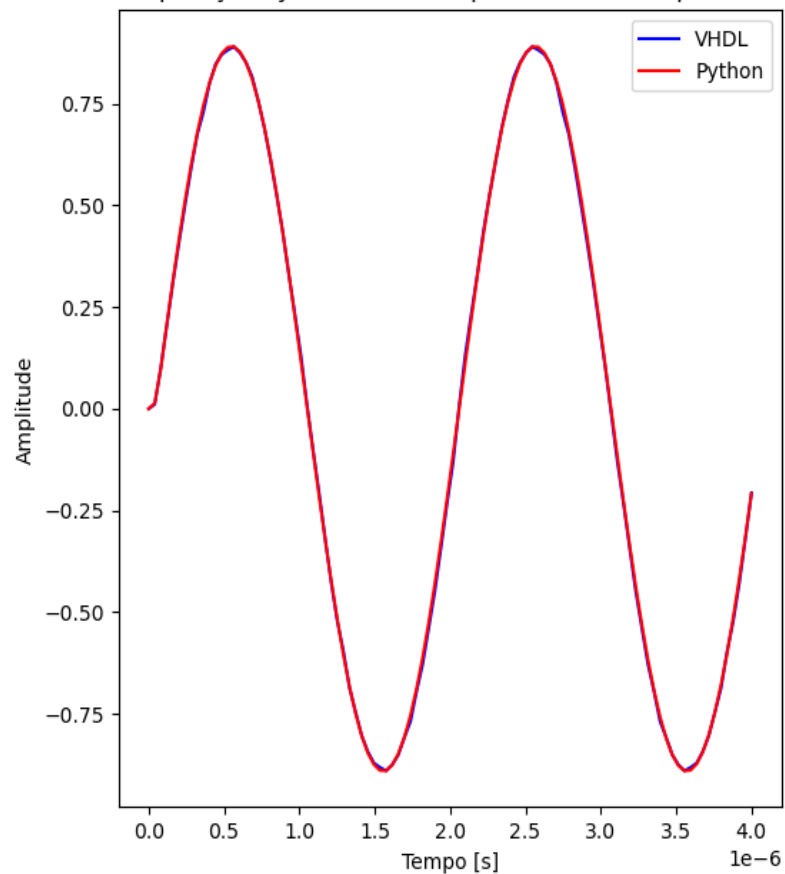
Teste - Upsampler



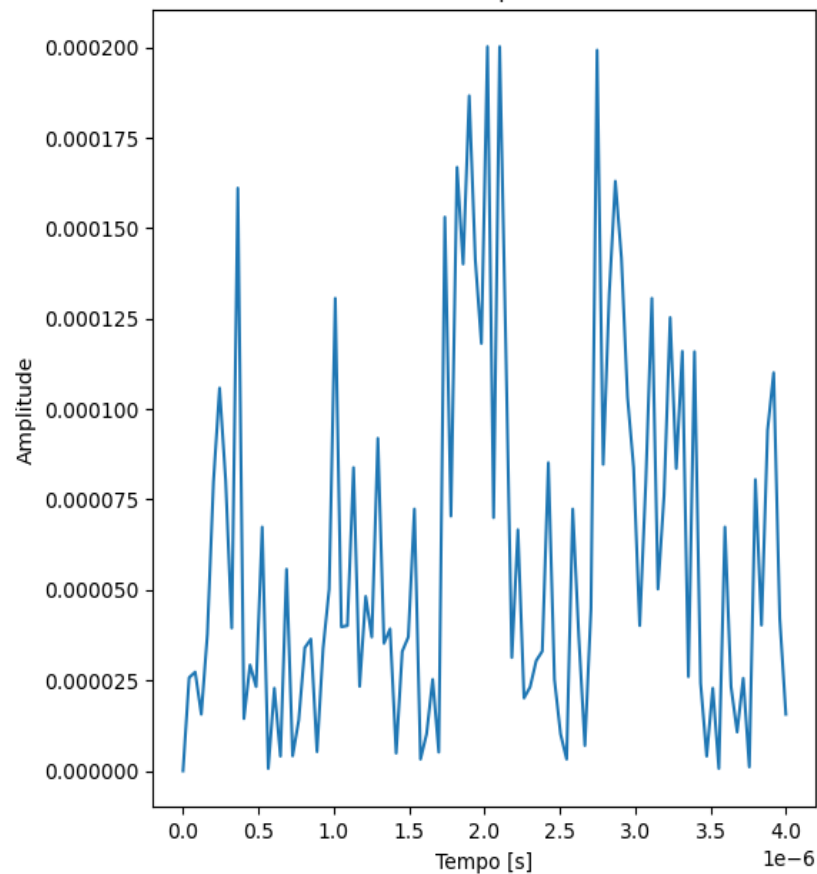
Resultado comparação com Python - Upsampler

Teste - Downsampler

Comparação Python Downsampler vs Downsampler VHDL



Erro médio quadrático



Resultado comparação com Python - Downsampler