

Universidade de Brasília - UnB
Faculdade do Gama - FGA
Engenharia Eletrônica

**IMPLEMENTAÇÃO EM CHIP DE SISTEMAS
CELULARES AUTÔMATOS DEDICADOS À
EMULAÇÃO DA PROPAGAÇÃO DE ONDAS
ACÚSTICAS EM SISTEMAS FÍSICOS**

Autor: Henrique Gomes de Moura
Orientador: Daniel M. Muñoz Arboleda

Brasília, DF
2022



Henrique Gomes de Moura

**IMPLEMENTAÇÃO EM CHIP DE SISTEMAS
CELULARES AUTÔMATOS DEDICADOS À
EMULAÇÃO DA PROPAGAÇÃO DE ONDAS
ACÚSTICAS EM SISTEMAS FÍSICOS**

Monografia submetida ao curso de graduação em Engenharia Eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Universidade de Brasília - UnB

Faculdade do Gama - FGA

Orientador: Daniel M. Muñoz Arboleda

Brasília, DF

2022

Henrique Gomes de Moura

IMPLEMENTAÇÃO EM CHIP DE SISTEMAS CELULARES AUTÔMATOS DEDICADOS À EMULAÇÃO DA PROPAGAÇÃO DE ONDAS ACÚSTICAS EM SISTEMAS FÍSICOS/ Henrique Gomes de Moura. – Brasília, DF, 2022-
96 p. : il. (algumas color.) ; 30 cm.

Orientador: Daniel M. Muñoz Arboleda

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade do Gama - FGA , 2022.

1. autômatos celulares. 2. sistemas em chip. I. Daniel M. Muñoz Arboleda.
II. Universidade de Brasília. III. Faculdade do Gama. IV. IMPLEMENTAÇÃO
EM CHIP DE SISTEMAS CELULARES AUTÔMATOS DEDICADOS À
EMULAÇÃO DA PROPAGAÇÃO DE ONDAS ACÚSTICAS EM SISTEMAS
FÍSICOS

CDU -a definir-

Henrique Gomes de Moura

IMPLEMENTAÇÃO EM CHIP DE SISTEMAS CELULARES AUTÔMATOS DEDICADOS À EMULAÇÃO DA PROPAGAÇÃO DE ONDAS ACÚSTICAS EM SISTEMAS FÍSICOS

Monografia submetida ao curso de graduação em Engenharia Eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Trabalho aprovado. Brasília, DF, 21 de abril de 2022:

Daniel M. Muñoz Arboleda
Orientador

**Prof. Dr. Carlos Humberto Llanos
Quintero (ENM/FT/UnB)**
Convidado 1

**Prof. Dr. Fabiano Araújo Soares
(FGA/UnB)**
Convidado 2

Brasília, DF
2022

Dedico este trabalho à minha amada família, minha companheira Stella Alves Mourão e meu filho Samuel Mourão de Moura.

Resumo

Os organismos artificiais celulares são circuitos que encapsulam e controlam arquiteturas integradoras, capacitadas para a execução de todos os cálculos requeridos sobre o domínio discreto escolhido. Os autômatos celulares constituem uma alternativa para modelagem de sistemas físicos em domínio discreto. Neste trabalho, o fenômeno físico considerado como estudo de caso é a propagação de ondas sonoras através de um meio ideal e estacionário. O trabalho foi realizado em duas etapas, sendo a primeira voltada para o desenvolvimento de um sistema celular autômato (CA) aplicável à emulação de ondas planas unidimensionais, para aplicação em circuitos reconfiguráveis, e a segunda voltada para a apresentação de propostas de implementação em chip. O sistema CA desenvolvido, denominado Camphs1D, é composto de duas células autômatas, baseadas em IP-cores de operadores aritméticos em 27 bits, capazes de operar todos os fenômenos básicos de onda em meios unidimensionais. As mencionadas células foram desenvolvidas a partir de duas estratégias distintas, sendo a primeira descrita diretamente em VHDL, no software Vivado, e a segunda descrita em C/C++, no software Vivado HLS, onde foram usados IP-cores de ponto flutuante de 32 bits da Xilinx. Foram utilizadas, para efeito de validação de resultados, simulações em *Testbench* VHDL, simulações no software SDK (Xilinx), e implementações físicas integradas com um processador ARM, para acesso à memória DDR disponível em placa. Todas as validações atingiram seus objetivos, relacionados à comparações com os mesmos resultados obtidos por modelos de referência, desenvolvidos a partir da solução de d'Alembert para a equação unidimensional da onda.

Palavras-chaves: autômatos celulares; malha de guias digitais de ondas; sistemas em chip.

Abstract

Artificial cellular organisms are circuits able to encapsulate and control integrative architectures, capable of performing all required calculations on the chosen discrete domain. Cellular automata are an alternative for modeling physical systems in a discrete domain. In this work, the physical phenomenon considered, as a case study, is the sound wave propagation through an ideal and stationary medium. The work was carried out in two stages, the first focused on the development of a CA system, applied to one-dimensional plane wave propagation, in a reconfigurable circuitry, and the second focused on its on-chip implementation proposals. The developed CA system, called Camphs1D, is composed of two automata cells, based on IP-cores of 27-bit arithmetic operators, capable of operate all basic wave phenomena in one-dimensional media. The aforementioned cells were developed from two different strategies, the first one directly described in VHDL, using Vivado software, and the second one using C/C++, using Vivado HLS software, where Xilinx 32-bit floating point IP-cores were used. In order to validate results, simulations were used in Testbench VHDL, simulations in SDK software (Xilinx), and physical implementations integrated with a software ARM terminator and DDR memory access. All validations achieved their goals, when compared with the reference models, developed from the solution of d'Alembert for the one-dimensional wave equation.

Key-words: cellular automaton; digital waveguide mesh; systems on chip.

Lista de ilustrações

Figura 1 – Simulação da propagação de ondas planas unidimensionais com presença de obstáculos. Os fenômenos básicos da onda estão considerados em respeito à solução exata do problema.	19
Figura 2 – Esquema em alto-nível dos componentes do autômato de von Neumann (CASTRO; CASTRO, 2008).	23
Figura 3 – Exemplo de resultado obtido do jogo da vida de Conway. A coloração escura das células, divididas em uma matriz quadrada, indica que as mesmas estão vivas, enquanto o contraste de cor indica que as mesmas estão mortas. O estado das células modifica no tempo, em função de suas vizinhanças.	24
Figura 4 – Resultados para aplicação da regra 30 dos autômatos de Wolfram.	25
Figura 5 – Representação de uma célula autômata conectada a seus 4 vizinhos, formando uma condição de vizinhança de Von Neumann. Também estão representadas as direções das forças elásticas que agem entre a célula central e seus vizinhos. FONTE: Huerta-Trujillo et. al (2015).	25
Figura 6 – Arquitetura utilizada no trabalho de Lima e Ferreira (2013).	27
Figura 7 – Digitalização da solução de d’Alembert para a Onda.	30
Figura 8 – Fenômenos de transmissão e reflexão de ondas planas unidimensionais.	31
Figura 9 – Representação dos fenômenos de transmissão e reflexão de ondas a partir das guias digitais de ondas.	34
Figura 10 – Primeira versão da célula autômata de dispersão, utilizada na arquitetura Camphs1D.	37
Figura 11 – Primeira versão da célula autômata geradora, utilizada na arquitetura Camphs1D.	37
Figura 12 – Primeira versão da célula autômata híbrida, utilizada na arquitetura Camphs1D.	37
Figura 13 – Célula autômata GCA1D. Para representarmos a célula autômata HCA1D, temos que retirar os sinais SET_STIM, SET, GAIN e MODE e acrescentar os sinais CROSS_LEFT e CROSS_RIGHT.	41
Figura 14 – Bloco operacional do sistema CA Camphs1D. Neste exemplo, temos uma célula receptora, indicada pelas siglas GCA RCE, externalizando seus sinais <i>rightout</i> e <i>leftout</i>	41
Figura 15 – Blocos operacionais das células autômatas GCA1D e HCA1D. As células são compostas de multiplicadores e somadores em ponto flutuante de 27 bits, além de portas lógicas, multiplexadores e registradores.	42

Figura 16 – Esboço da arquitetura Camphs1D05G1. Temos 5 células autômatas, distribuídas em dois tipos(G/H), e 2 sinais receptores (RCE).	45
Figura 17 – Diagrama de tempo utilizado no organismo artificial Camphs1D05G1.	46
Figura 18 – Esboço da arquitetura Camphs1DB9G2.	46
Figura 19 – Diagrama de tempo utilizado no organismo artificial Camphs1DB9G2.	48
Figura 20 – Diagrama esquemático representativo da validação de Camphs1DB9G2. Para maior detalhamento dos sinais envolvidos na arquitetura principal consultar Figura 14.	50
Figura 21 – Esboço da primeira estratégia de implementação de Camphs1D. Temos a necessidade de integrar o core CA com uma unidade de memória ROM. Demais IP-cores são necessários para o funcionamento e <i>debug</i> de sinais.	51
Figura 22 – Estrutura básica de um projeto gerado no Vivado HLS C-level.	53
Figura 23 – Esboço da arquitetura Camphs1D05G1RAV. Além do core CA, temos memória ROM embarcada, um IP-core ILA, e um IP-core VIO para introdução de sinais. Temos também uma pequena máquina de estados controlando a unidade de memória ROM.	54
Figura 24 – Esboço da arquitetura Carom1D05G1. Este componente deve armazenar três sinais de controle, tendo um sinal 2 bits (MOD), um com 27 bits (VAL), e um último sinal com apenas 1 bit (STA).	55
Figura 25 – Esboço da arquitetura de Camphs1D05G1AL. Este design foi automaticamente constituído pela ferramenta Vivado 2018.3 Webpack.	57
Figura 26 – Esboço do CORE de Camphs1D05G1AL. A interface <i>AXI4-Lite</i> promove a introdução e externalização dos sinais no core CA, que opera ciclicamente até o estado final desejado.	57
Figura 27 – Esboço da arquitetura de Camphs1D05G1AS. Este design foi automaticamente constituído pela ferramenta Vivado 2018.3 Webpack.	59
Figura 28 – Esboço do CORE de Camphs1D05G1AS. A interface <i>AXI4-Lite</i> promove a introdução e externalização do estado final do core CA, enquanto a interface <i>AXI4-Stream</i> externaliza os sinais receptores.	59
Figura 29 – Simulação comportamental da organismo artificial Camphs1D05G1. Em tela, um visão geral das variáveis envolvidas.	61
Figura 30 – Resultados esperados para o sistema Camphs1D05G1, com base nos fenômenos básicos da onda. Cada linha apresentada representa as saídas da equação de d’Alembert, submetida ao sinal de entrada (célula 3), em cada um dos 5 nós da solução discreta. Temos também representadas as duas direções de movimento das ondas acústicas.	62
Figura 31 – Resultado esperado para o sistema Camphs1D05G1, com base nos fenômenos básicos da onda. Em tela, temos os sinais de estímulo gerados.	63

Figura 32 – Resultado esperado para o sistema Camphs1D05G1, com base nos fenômenos básicos da onda. Em tela, as amplitudes de onda transmitidas pelas células autômatas.	63
Figura 33 – Decodificação dos valores registrados nos receptores de Camphs1D05G1. Fragmento do código desenvolvido no software Octave.	65
Figura 34 – Amostragem dos valores registrados nos receptores de Camphs1D05G1. Fragmento do código desenvolvido no software Octave.	65
Figura 35 – Resultado da decodificação dos valores registrados nos canais receptores de Camphs1D05G1. Temos em tela os 12 primeiros instantes de tempo do sistema CA.	66
Figura 36 – Posicionamento dos receptores “A ”e “B ”em Camphs1DB9G2, nas saídas das células 20 e 150.	67
Figura 37 – Resultado da decodificação dos valores registrados nos canais do receptor “A ”em Camphs1DB9G2 e modelo de referência.	67
Figura 38 – Resultado da decodificação dos valores registrados nos canais do receptor “B ”em Camphs1DB9G2 e modelo de referência.	68
Figura 39 – Consumo estimado de recursos para Camphs1D05G1.	69
Figura 40 – Consumo estimado de recursos para Camphs1DB9G2.	69
Figura 41 – Análise de tempo obtida para Camphs1D05G1RAV, para o caminho crítico de <i>setup</i> . Observamos na imagem os valores dos parâmetros <i>Arrival Time</i> e <i>Required Time</i> , bem como a folga positiva encontrada. Também está indicado o caminho percorrido na placa.	70
Figura 42 – Análise de tempo obtida para Camphs1D05G1RAV, para o caminho crítico de <i>hold</i> . Observamos na imagem os valores dos parâmetros <i>Arrival Time</i> e <i>Required Time</i> , bem como a folga positiva encontrada. Também está indicado o caminho percorrido na placa.	71
Figura 43 – Simulação comportamental de Camphs1D05G1RAV.	72
Figura 44 – Resultados coletados pelo ILA-core de Camphs1D05G1RAV, para a porta esquerda das células autômatas receptoras. À esquerda temos os resultados obtidos na porta esquerda, enquanto à direita temos os resultados obtidos na porta direita.	72
Figura 45 – Análise de tempo obtida para Camphs1D05G1AL. Em tela, temos os caminhos críticos de <i>setup</i> e <i>hold</i> (origens e destinos), indicando folgas positivas.	73
Figura 46 – Resultados obtidos no software SDK para Camphs1D05G1AL. Em tela estão os resultados da Co-simulação praticada.	74
Figura 47 – Análise de tempo obtida para Camphs1D05G1AS. Em tela, temos os caminhos críticos de <i>setup</i> e <i>hold</i> (origens e destinos), indicando folgas positivas.	75

Figura 48 – Resultados obtidos no software SDK para Camphs1D05G1AS. Em tela estão os resultados da Co-simulação praticada.	76
Figura 49 – Tabela de consumo de recursos para as implementações físicas praticadas.	77
Figura 50 – <i>Layout</i> obtido da implementação de Camphs1D05G1RAV.	78
Figura 51 – <i>Layout</i> obtido da implementação de Camphs1D05G1AL.	79
Figura 52 – <i>Layout</i> obtido da implementação de Camphs1D05G1AS.	79
Figura 53 – Mapa de classes utilizado na solução fragmentada operada por Camphs1DH1G4.	87
Figura 54 – Mapa de classes utilizado na solução fragmentada operada por Camphs1DH1G4.	88

Lista de tabelas

Tabela 1	– Consumo de memória em bits das estratégias MGDO e CA. A sigla I_{32} representa inteiros com 32 bits, F_{32} representa pontos flutuantes em 32 bits, N_j representa o número de nós ou juntas de dispersão e a sigla N_s representa o número de passos temporais.	38
Tabela 2	– Análise de perfil das soluções MGDO e CA.	39
Tabela 3	– Sinais das utilizados nas células Autômatas GCA1D e HCA1D.	40
Tabela 4	– Amostragem de um sinal senoidal de $400Hz$ a $3200Hz$, para operação das células geradoras no organismo artificial Camphs1DB9G2.	47
Tabela 5	– Sinais de controle embarcados na memória ROM, necessários para o funcionamento de Camphs1D05G1RAV.	54
Tabela 6	– Parâmetros adotados na implementação com restrições de tempo.	56
Tabela 7	– Valores de referência para Camphs1D05G1AL. Excitação e resultados após 11 interações (contando a excitação).	60
Tabela 8	– Valores de referência para Camphs1D05G1AS. Excitação e resultados após 10 interações (contando a excitação).	60
Tabela 9	– Valores registrados nos canais de saída de Camphs1D05G1.	64
Tabela 10	– Conteúdo dos arquivos de controle utilizados na simulação comportamental do organismo artificial Camphs1D05G1.	65
Tabela 11	– Erro médio quadrático calculado para os sinais receptores de Camphs1DB9G2, em relação a saídas semelhantes coletadas no modelo de referência.	67
Tabela 12	– Tabela de consumo de recursos para as implementações dos organismos (A) Camphs1D05G1RAV, (B) Camphs1D05G1AL e (C) Camphs1D05G1AS.	78

Lista de abreviaturas e siglas

FSM	Finite State Machines
PF	Ponto flutuante
PS	Processor System
PL	Programmable Logic
LUT	Lookup Table
FF	Flip-Flop
IOB	Bounded Input/Output
DSP	Digital Signal Processor
CA	Celular Autômata / Célula Autômata
POO	Programação Orientado à Objeto
EDO	Equações diferenciais ordinárias
FDTD	Finite Difference Time Domain
MGDO	Malha de Guias Digitais de Ondas
ILA	Integrated Logic Analyser
BMG	Block Memory Generator
SDK	Software Development kit
BRAM	Block RAM
AXI-DMA	AXI Direct Memory Access
START	Sinal normalmente utilizado com gatilho para o funcionamento de componentes eletrônicos
MODE	Sinal utilizado para definir o comportamento (dispersão ou geração de ondas) da célula autômata GCA1D
READY	Sinal geralmente utilizado para indicar o término da operação de um componente eletrônico
Camphs1D	Cellular Automaton for Unidimensional Physical Simulation

- Camphs1D05G1 Primeiro organismo artificial obtido a partir de Camphs1D, com 5 células sendo uma geradora.
- Camphs1DB9G2 Segundo organismo artificial obtido a partir de Camphs1D, com 185 células sendo duas geradoras.
- Camphs1D05G1RAV Terceiro organismo artificial obtido a partir de Camphs1D, com 5 células sendo uma geradora, Memória ROM, Integrated Logic Analyser (ILA) e Virtual Input/Output (VIO).]
- Camphs1D05G1AL Quarto organismo artificial obtido a partir de Camphs1D, com 5 células sendo uma geradora, com ARM integrado e barramento de dados do tipo *AXI4-Lite*.
- Camphs1D05G1AS Quinto organismo artificial obtido a partir de Camphs1D, com 5 células sendo uma geradora, com ARM integrado e barramento de dados do tipo *AXI4-Stream*.

Sumário

1	INTRODUÇÃO	16
1.1	Definição do Problema	16
1.2	Objetivos	18
1.3	Aspectos Metodológicos	18
1.4	Contribuições deste trabalho	20
1.5	Conteúdo e organização dos capítulos	20
2	AUTÔMATOS CELULARES	22
2.1	Autômatos celulares aplicados à modelagem de sistemas físicos	25
2.2	Aceleração em Hardware de sistemas CA	26
3	ONDAS PLANAS UNIDIMENSIONAIS	29
3.1	Equação Linear Unidimensional da Onda	29
3.2	Digitalização da solução de d'Alembert para a Onda	30
3.3	Modelagem dos fenômenos básicos de onda	31
3.4	Modelo de referência	34
4	METODOLOGIA	36
4.1	O Sistema CA Camphs1D	36
4.1.1	Solução de software para sistema CA Camphs1D	37
4.1.2	Análises de perfil das soluções por MGDO e CA	38
4.1.3	Projeto do sistema CA Camphs1D	39
4.1.3.1	Blocos operacionais: células autômatas	39
4.1.3.2	Arquitetura principal: integração de componentes	40
4.1.4	Notas sobre o desenvolvimento e aplicações do sistema CA Camphs1D	43
4.2	Projeto das simulações comportamentais	43
4.2.1	Organismo artificial Camphs1D05G1	45
4.2.1.1	Diagrama de tempo: sinais de controle para Camphs1D05G1	45
4.2.2	Organismo artificial Camphs1DB9G2	46
4.2.2.1	Diagrama de tempo: sinais de controle para Camphs1DB9G2	46
4.2.3	Padronização do sistema Camphs1D: rotina vCAgen	48
4.2.4	Validação das simulações comportamentais	49
4.3	Projeto das implementações em placa	50
4.3.1	Uma breve descrição da ferramenta Vivado HLS C-level	52
4.3.2	O organismo artificial Camphs1D05G1RAV	53
4.3.2.1	Funcionamento da memória embarcada em placa e vetores de teste	54

4.3.2.2	Análise de tempo realizada para Camphs1D05G1RAV	55
4.3.3	O organismo artificial Camphs1D05G1AL	56
4.3.4	O organismo artificial Camphs1D05G1AS	58
4.3.4.1	Validação das implementações realizadas em placa	59
5	RESULTADOS	61
5.1	Simulações comportamentais	61
5.1.1	Organismo artificial Camphs1D05G1	61
5.1.2	Dados de entrada, saída e decodificações realizadas em Camphs1D05G1	63
5.1.3	Organismo artificial Camphs1DB9G2	66
5.1.4	Dados de entrada, saída e decodificações realizadas em Camphs1DB9G2	68
5.1.5	Relatórios de utilização de recursos após sínteses lógicas	68
5.2	Implementações em Placa	70
5.2.1	Organismo artificial Camphs1D05G1RAV	70
5.2.2	Organismo artificial Camphs1D05G1AL	72
5.2.3	Organismo artificial Camphs1D05G1AS	73
5.2.4	Relatório de utilização de recursos após as implementações	75
6	CONCLUSÕES	80
	REFERÊNCIAS	82
	APÊNDICES	85
	APÊNDICE A – PROPOSTA DE UMA ESTRATÉGIA CA FRAGMENTADA PARA CAMPHS1D	86
	APÊNDICE B – MODELING WAVE PROPAGATION USING CELLULAR AUTOMATA ON CHIP	90

1 Introdução

1.1 Definição do Problema

Cientistas de todas as áreas investigam fenômenos da natureza através dos mais diversos modelos matemáticos. Modelos matemáticos podem ser classificados com respeito à forma de se tratar o tempo, o espaço e o grau de realismo no tratamento das incertezas inerentes.

Dentre as inúmeras alternativas existentes, as equações diferenciais ordinárias (EDO) representam uma maneira clássica para abordagem de problemas de modelagem em domínio contínuo, de espaço e tempo. Entretanto, em muitas ocasiões, suas resoluções acabam se concentrando em regiões de assumida linearização e aceitável convergência numérica, fato que limita a generalização de soluções. Por este motivo, o domínio completo, i.e., sem restrições de tempo ou espaço, de uma solução pode não ser viável ou factível, dificultando os avanços de tal metodologia frente às necessidades impostas pela natureza dos fenômenos estudados.

Vejamos o caso da modelagem da propagação de ondas em meios elásticos. Através da resolução da equação geral da onda é possível obter funções que descrevam todos os fenômenos básicos de onda, incluindo a interferência, transmissão, reflexão e difração pelo meio. Entretanto, o grau de realismo da solução, no que se diz respeito às geometrias envolvidas, acaba dificultando a resolução desta equação no domínio contínuo, e levando de maneira conveniente às resoluções por métodos numéricos e funções de interpolação. Este fato denota uma “fraqueza” na solução que, dominada por aproximações numéricas e rígidas regras de convergência para as suas derivadas parciais, acaba sendo localizada a uma região do espaço e tempo. Em outras palavras, estamos lidando com um sistema complexo de equações diferenciais parciais, exuberante em sua formulação, para no fim termos que nos acostumar com severas restrições matemáticas e soluções controladas.

Deste modo surgem os questionamentos a seguir. Seria mesmo necessário, para toda ou qualquer situação, sempre considerarmos como sendo a melhor solução a modelagem de sistemas físicos a partir do tratamento de sistemas de equações diferenciais e/ou integrais? O tratamento de domínios contínuos de espaço e tempo são, em todas as situações, preferidos aos métodos discretos? Tais respostas devem ser resolvidas por cada projetista, em função de suas preferências e habilidades com os números. Mas é impossível negar que a melhor solução sempre converge para a forma mais simplificada e abrangente, ao mesmo tempo, no tratamento da realidade buscada.

Abandonando o *continuum* do cálculo diferencial e integral, qualquer solução pode

ser elaborada em um domínio discreto do espaço e tempo. Pode não parecer em um primeiro momento mas o domínio discreto é mais realista do que o domínio contínuo, pelo simples fato do mundo ser, aos olhos e sentidos humanos, definitivamente discreto.

Um sinal cardíaco nunca pode ser entendido através de uma “equação divina”, mas sim através de procedimentos de amostragem e sensores elétricos. De forma semelhante, o estudo dos mecanismos de fratura se constitui, em primeiro lugar, de relações físico-matemáticas elaboradas, para depois se ajustar à realidade a partir de testes de identificação, i.e., observações discretas de parâmetros físicos.

Voltando-se para a modelagem discreta de sistemas físicos e/ou matemáticos, é possível pensar em uma grade ou malha que armazene em suas interseções informações relevantes sobre um fenômeno em estudo. Se consideramos que em cada interseção se tenha uma célula, e que a mesma se comporte conforme sua vizinhança ao longo do tempo, transitando entre estados, mudando as quantidades matemáticas representativas de sua essência, conforme regras simples de sobrevivência e morte, temos então um sistema celular autômato. Segundo Haderler and Muller ([HADELER; MULLER, 2017](#)), um sistema celular autômato pode ser visto com uma representação simples e direta da solução de um sistema por equações integro-diferenciais. Entretanto, um sistema celular autômato tem uma enorme vantagem sobre a solução analítica, que é o fato de ser totalmente discreto e livre de critérios de convergência espaciais ou temporais. Em outras palavras, as células autômatas simplesmente vivem e evoluem em um espaço finito, controladas pelas suas próprias regras, influenciadas unicamente pelas suas próprias vizinhas.

Felizmente é possível provar que um sistema de equações diferenciais parciais pode ser transformado em um sistema CA, ou vice e versa. Esta prova de conceito está apresentada neste trabalho, tomando como estudo de caso a propagação de ondas acústicas estimuladas em um meio elástico ideal em regime estacionário, regida pela conhecida solução de d’Alembert para a equação unidimensional da onda. A Figura 1, a seguir, apresenta um instante aleatório de uma simulação realizada pelo modelo de referência, constituído por um fonte acústica e alguns obstáculos.

A aplicação de sistemas CA apresenta algumas vantagens, em relação à aplicação de sistemas de EDO. Tais vantagens estão listadas a seguir.

1. Não possuem limitações numéricas.
2. Estão diretamente envolvido com a emulação do(s) fenômeno(s) físico(s).
3. Não possuem restrições de cunho geométrico.
4. Não necessitam de estratégias numéricas para convergência e estabilidade.

Por outro lado, o desenvolvimento empírico de sistemas CA pode dificultar a aplicação e entrega de soluções com restrições de tempo (tempo imposto pelo mercado de

trabalho). Também precisamos reforçar que o projeto e a aplicação de sistemas CA requer um elevado grau de abstração/observação, sobre o(s) fenômeno(s) observado(s).

O estudo de caso realizado por este trabalho foi motivado pelo elevado custo computacional relacionado à resolução do sistema de EDO, por diferenças finitas (FDTD) (TREFETHEN, 1994). Sabe-se que um sistema CA é, em sua natureza, mais simples, podendo ser aplicado na retratação de sistemas físicos (CASTRO; CASTRO, 2008). Neste sentido, a implementação em hardware se justifica no fato deste sistema operar em ciclos repetitivos, em algumas situações, com grande número de células autômatas envolvidas.

De maneira resumida, duas questões estão sendo levantadas neste trabalho. A primeira, se refere à simplificação da solução para o transporte de ondas acústicas em sistemas físicos, outrora obtida por métodos tradicionais aplicados à resolução de EDO. A segunda questão se refere à possibilidade de aceleração da solução em hardware, o que representa um grande benefício para o mercado em potencial, tendo em tela o elevado custo computacional atribuído à solução discreta deste problema.

1.2 Objetivos

O objetivo central deste trabalho é a implementação em chip de um sistema celular autômato devidamente constituído para a emulação da propagação de ondas acústicas em sistemas físicos, em regime ideal e estacionário. Os objetivos específicos deste trabalho, são relacionados a seguir.

1. Definição e programação, utilizando linguagem C, de um modelo de referência para um sistema CA em domínio unidimensional.
2. Implementação, em hardware reconfigurável, utilizando um kit de desenvolvimento FPGA, do sistema CA em domínio unidimensional.
3. Simulação comportamental de um sistema CA em domínio unidimensional, dedicado à resolução do mesmo problema considerado no modelo de referência.
4. Desenvolvimento de um sistema em chip, com integração software/hardware (SoC FPGA), dedicado à implementação de um sistema CA em domínio unidimensional.

1.3 Aspectos Metodológicos

Em primeiro lugar, falando sobre o modelo de referência, o mesmo foi realizado em duas etapas. Na primeira etapa, o método da MGDO (malha de guias digitais de ondas) foi utilizado na simulação da propagação de ondas planas unidimensionais em um duto, com comprimento linear de 20 metros, conforme apresentado na Figura 1. A

MGDO é uma das técnicas mais aplicadas na reprodução dos fenômenos considerados, e está devidamente implementada nos trabalhos de Moura (MOURA, 2006).

Dentro do duto é preciso considerar uma membrana com impedância acústica diferenciada do meio, de maneira a se produzir fenômenos de transmissão e reflexão de ondas. As extremidades do tubo também receberam membranas flexíveis, devidamente calculadas para refletirem uma parcela de ondas no interior do tubo.

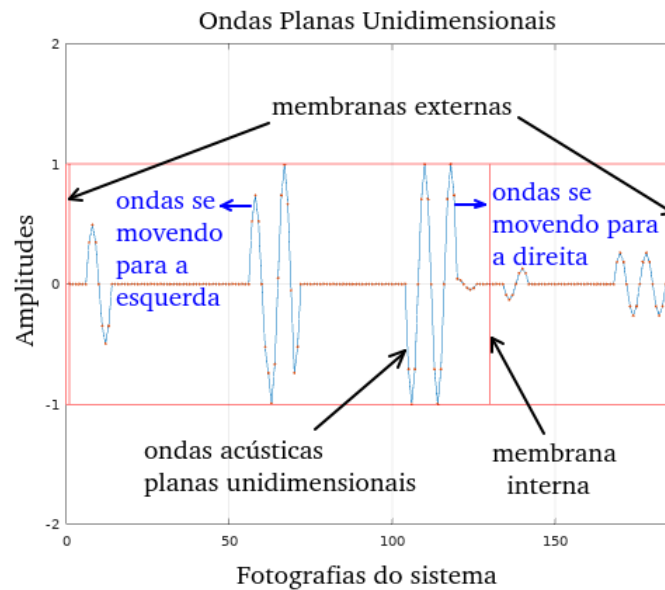


Figura 1 – Simulação da propagação de ondas planas unidimensionais com presença de obstáculos. Os fenômenos básicos da onda estão considerados em respeito à solução exata do problema.

Na segunda etapa, um novo código foi constituído para o problema utilizando-se de um sistema CA dedicado. Os resultados do sistema CA foram confrontados com os resultados da MGDO. Após a validação do sistema CA, o mesmo foi utilizado como modelo de referência para os demais trabalhos realizados em circuitaria reconfigurável.

As duas soluções foram avaliadas, não somente em relação à convergência numérica, mas também através de análises de perfis de código. Tais análises são úteis para verificação da viabilidade da solução CA obtida, frente aos custos operacionais e recursos necessários à implementação da MGDO.

Na sequência deste trabalho, um kit de desenvolvimento em *Field-programmable Gate Array* (FPGA) foi utilizado para validações em hardware, em sistema em chip (SoC FPGA). Inicialmente, simulações comportamentais foram realizadas sobre as células autômatas projetadas, para validar o sistema CA Camphs1D. Para as verificações funcionais da implementação física do sistema celular autômato (CA) proposto, foram utilizados um IP-core de um analisador lógico integrado (ILA) e um IP-core para uso de memórias ROMs, fornecidos pela Xilinx. O ILA core permite rastrear sinais de interesse a fim de

verificar o correto comportamento dos diversos componentes que constituem as células do sistema.

De maneira oportuna, a ferramenta Vivado HLS C-level foi testada para encapsular IP-cores de sistemas celulares autômatas, tornando-os reutilizáveis em outros projetos. Os sistemas foram implementados utilizando-se da integração entre o microprocessador ARM Cortex-A9 Dual Core e barramentos on-chip AXI de médio e alto desempenho, com o intuito de oferecer maior liberdade, e maior volume de dados transferidos, na modelagem autômata celular de sistemas físicos.

1.4 Contribuições deste trabalho

Foi submetida uma publicação no *34rd Symposium on Integrated Circuits and Systems Design*, organizado pela Universidade de Campinas-SP, no ano de 2021. A publicação apresentou resultados inéditos para a modelagem de um sistema celular autômata unidimensional, usando circuitos reconfiguráveis (vide Apêndice B).

Em decorrência desta primeira publicação, foi iniciada, no LabMicro da Faculdade UnB Gama-DF, uma nova linha de pesquisa na área de modelagem autômata celular de sistemas físicos. A linha possui outros dois trabalhos vinculados, sendo 1 trabalho de conclusão de curso e 1 trabalho de mestrado. Os temas escolhidos abordam aplicações variadas dos autômatas celulares propostos, bem como o desenvolvimento de novas células autômatas, capazes de modelar domínios maiores e mais abrangentes.

1.5 Conteúdo e organização dos capítulos

No capítulo 2 o leitor encontrará um breve histórico sobre os sistemas CA, desde suas primeiras aparições à algumas das mais recentes aplicações relacionadas ao tema deste trabalho. No capítulo 3 o leitor será apresentado ao problema físico abordado. Nesta primeira parte do trabalho, a propagação de ondas acústicas planas unidimensionais será tratada, utilizando-se da conhecida solução de d'Alembert para a onda. Uma estratégia tradicional baseada na resolução por FDTD das EDO de movimento será apresentada, através de um modelo de referência constituído em linguagem C.

O desenvolvimento de todos os sistemas CA, aplicados à solução do problema proposto, está apresentado no capítulo 4. As principais regras do sistema CA, bem como suas expressões matemáticas básicas, estão apresentadas. Neste capítulo o leitor também encontrará os blocos operacionais necessários para a implementação e integração das células autômatas e dos organismos artificiais utilizados neste trabalho.

No capítulo 5 o leitor terá acesso aos resultados obtidos e às comparações realizadas com o modelo de referência. Serão apresentados os resultados de uma análise de utilização

de recursos e os layouts de implementação em placa, referentes às arquiteturas testadas.

O capítulo 6 traz as conclusões obtidas para o sistema Camphs1D, bem como um breve resumo sobre os desafios esperados para a continuação dos trabalhos.

2 Autômatos Celulares

Em primeiro lugar, se faz oportuna a apresentação de uma definição informal, porém satisfatória, para o entendimento de um sistema CA padrão. Um sistema CA pode assim ser denominado se possuir as seguintes propriedades fundamentais (CASTRO; CASTRO, 2008):

1. Consistir em uma matriz, malha ou grade de células.
2. Evoluir em passos discretos de tempo.
3. Cada célula é caracterizada por um estado pertencente a um conjunto finito de estados.
4. A relação com a vizinhança é local e uniforme.

Após tais informações é possível imaginar uma enorme quantidade de aplicações numéricas semelhantes, em vários aspectos, a um sistema CA. Entretanto, é importante mencionar que a principal e decisiva característica de um sistema CA não está mencionada acima, mas sim a seguir:

A natureza funcional de um sistema CA é constituída por regras simples, desprezadas de trajetórias, lugares geométricos e/ou demais funções que possam impedir a evolução espontânea de suas células autômatas.

O polivalente matemático Von Neumann (1903 - 1957) foi quem primeiro discutiu sobre a aplicação de células autômatas artificiais, em seus estudos sobre as "máquinas de cálculo". Os resultados de seus estudos na área foram publicados em 1966 por Arthur W. Burks, depois de sua morte (NEUMANN, 1966). Von Neumann tratava as células autômatas como uma realização matemática de organismos naturais, capazes de operar regras básicas de sobrevivência, tais como a reprodução e evolução.

Von Neumann esteve motivado pela possibilidade de idealizar um construtor universal, capaz de operar tarefas complexas tais como a replicação autônoma de robôs. O construtor universal de Von Neumann foi definido com 29 estados, capazes de transportarem sinais e executarem operações lógicas visando à sua completa auto-replicação. O DNA artificial obtido era constituído por uma fita de células capazes de decodificarem uma sequência de bits (NEUMANN, 1966). Um esquema da solução proposta se encontra na Figura 2.

Von Neumann estabeleceu, em uma malha bidimensional, padrões de vizinhança e suas interações com a célula central. No padrão mais simples, cada célula se interage com sua vizinhança em suas fronteiras norte, sul, leste e oeste (totalizando 5 células

envolvidas). A quantidade de células envolvidas em um vizinhança de Von Neumann com dimensão d e raio r é definida pelo o número de Delannoy $D(d, r)$ (JOACHIM, 2007).

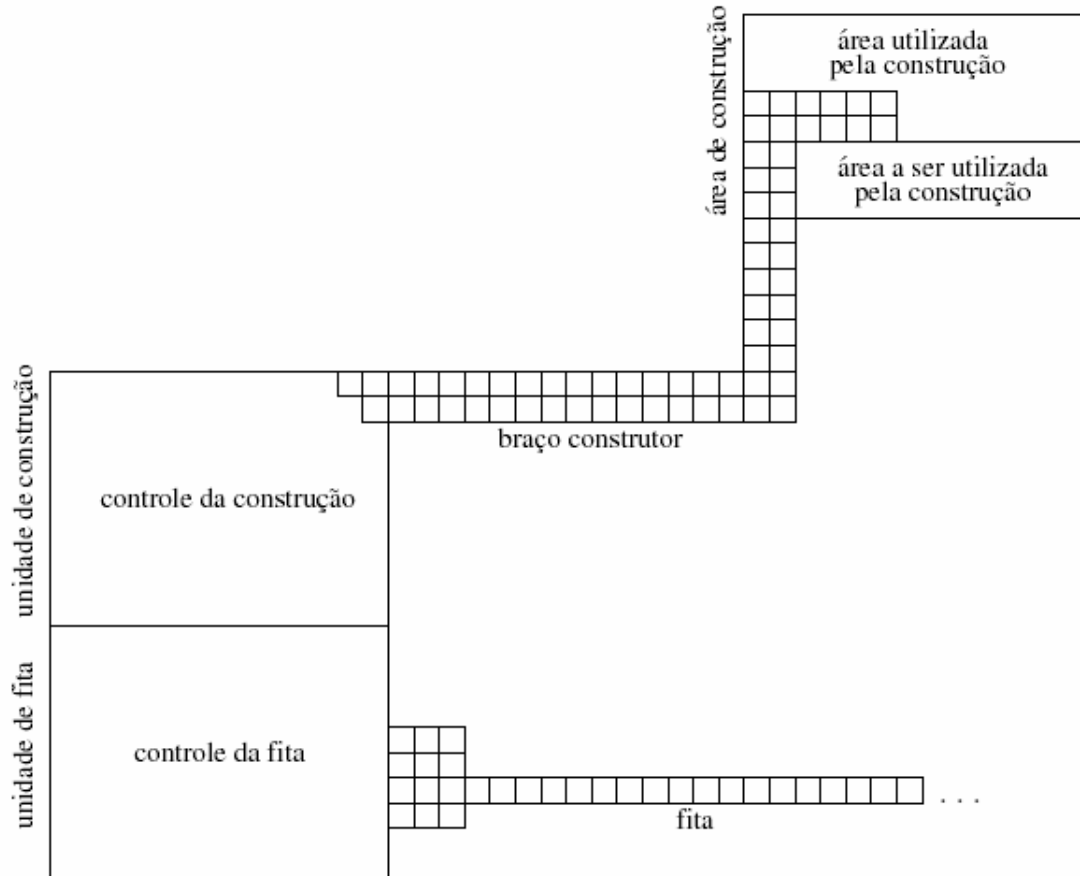


Figura 2 – Esquema em alto-nível dos componentes do autômato de von Neumann (CASTRO; CASTRO, 2008).

Após esta implementação, variadas aplicações começaram a surgir nas mais diferentes áreas da ciência. Talvez o mais famoso exemplo de um sistema CA seja, devido à sua didática e simplicidade, o "Jogo da Vida" de Conway (GARDNER, 1970) (BOINGO, 2011). Nesta aplicação, as CA possuem 2 estados, definidos por um bit. Deste modo, cada célula muda de estado, em cada instante de tempo, a depender de 4 elegantes regras:

1. Qualquer célula viva com menos de dois vizinhos vivos morre de solidão.
2. Qualquer célula viva com mais de três vizinhos vivos morre de superpopulação.
3. Qualquer célula morta com exatamente três vizinhos vivos se torna uma célula viva.
4. Qualquer célula viva com dois ou três vizinhos vivos continua no mesmo estado para a próxima geração.

O funcionamento do jogo da vida pode ser observado em diversos canais do YouTube, bastando para tal realizar uma busca simples. Um estudo mais denso sobre as possibilidades e aplicações deste sistema pode ser encontrado em bibliografias especializadas (ADAMATZKY, 2010) (FLITNEY; ABBOTT, 2010). Uma imagem do jogo da vida pode ser apreciada na Figura 3, a seguir.

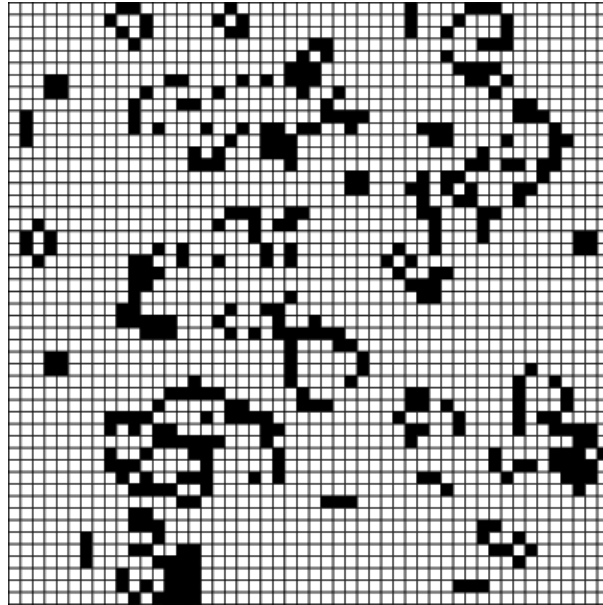


Figura 3 – Exemplo de resultado obtido do jogo da vida de Conway. A coloração escura das células, divididas em uma matriz quadrada, indica que as mesmas estão vivas, enquanto o contraste de cor indica que as mesmas estão mortas. O estado das células modifica no tempo, em função de suas vizinhanças.

Outra aplicação de grande difusão foi apresentada por Stephen Wolfram, em 1983 (WOLFRAM, 2020). Os autômatos de Wolfram são células organizadas em um espaço unidimensional, com três vizinhos e dois estados. A depender de sua vizinhança e em respeito às regras utilizadas, novas linhas são construídas abaixo da anterior. Desta forma o sistema evolui a cada instante de tempo (WOLFRAM, 1994), produzindo arquiteturas elegantes e inesperadas. Um exemplo deste tipo de CA pode ser apreciado na Figura 4, a seguir.

Inúmeras outras aplicações poderiam ser citadas aqui ao longo dos anos. As aplicações de sistemas CA se expandem para diversas áreas da ciência, incluindo a física, a biologia, geociência e astronomia (CHOPARD; DROZ, 1998) (MELOTTI, 2009) (PERES, 2005) (PERDANG; LEJEUNE, 1996). Neste momento serão colocadas em evidência aplicações relacionadas ao objeto de estudo deste trabalho, que representa a propagação de ondas acústicas em sistemas físicos.

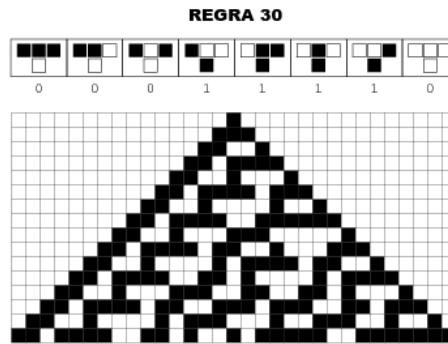


Figura 4 – Resultados para aplicação da regra 30 dos autômatos de Wolfram.

2.1 Autômatos celulares aplicados à modelagem de sistemas físicos

Huerta-Trujillo ([TRUJILLO et al., 2015](#)) aplicaram um sistema CA dedicado à representação de vibração em membranas elásticas, disposto em um espaço bidimensional retangular (vide Figura 5). A implementação foi realizada em paradigma de programação orientada à objeto (POO), usando linguagem C++, fato que permitiu uma simplificada representação de parâmetros físicos encapsulados em cada célula. As deformações foram observadas ao longo do tempo respeitando regras estabelecidas nas vizinhanças de cada célula.

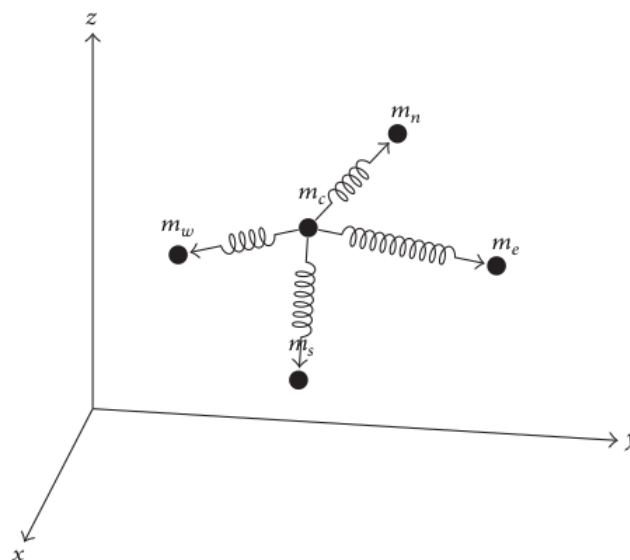


Figura 5 – Representação de uma célula autômata conectada a seus 4 vizinhos, formando uma condição de vizinhança de Von Neumann. Também estão representadas as direções das forças elásticas que agem entre a célula central e seus vizinhos. FONTE: Huerta-Trujillo et. al (2015).

Outra aplicação em destaque foi apresentada por Janowicz ([JANOWICZ; ORŁOWSKI, 2006](#)), em um estudo sobre a propagação de ondas eletromagnéticas em meios dispersivos. Para tal, o modelo de Drude ([HAN, 2011](#)) foi utilizado em um sistema CA unidimensi-

onal, capaz de emular a transferência de quantidades de ondas nas vizinhanças de cada célula. Observa-se aqui que foi atribuída às células uma grande quantidade de estados, de tal modo a armazenarem um conjunto de números reais necessário para operação do modelo matemático intracelular.

Uma terceira aplicação pode ser ilustrada nos trabalhos de Belotti (COLOMBO, 2017), sobre a aplicação de autômatos celulares na simulação de processos de microfabricação. Este trabalho apresentou um sistema CA capaz de simular a evolução topográfica de uma superfície de silício, através do atendimento ao princípio de Huygens (EMILE; BRUNEL, 2003). A CA foi definida a partir de 3 valores reais, um valor inteiro e um bit. O valor inteiro foi utilizado para representar o material da célula (por exemplo, Si, Al, etc). Os valores reais representaram distâncias e o bit um valor condicional que permite modificar a célula. Nota-se que, mais uma vez, um padrão se repete na definição das células autômatas aplicadas em sistemas físicos, no que tange à armazenagem de propriedades físicas (espaço finito de estados) combinada com regras intracelulares para transição de estado.

No que se refere à acústica, os sistemas CA podem ser atrativos devido a basicamente duas características buscadas na grande maioria das aplicações numéricas. A primeira característica se refere à solução espacial das equações de movimento da onda acústica, em meios elásticos. Os autômatos celulares atuam em uma matriz de células, facilmente adaptável às geometrias do problema, e com potencial de realizar os cálculos necessários para o problema. A segunda característica se refere ao custo computacional, provocado devido à excessiva quantidade de pontos na discretização do meio. Mais uma vez, os autômatos celulares se mostram atraentes, pois representam unidades matemáticas simplificadas e portanto, mais fáceis de serem embarcadas e/ou armazenadas em memória.

2.2 Aceleração em Hardware de sistemas CA

De acordo com Bakhteri (BAKHTERI; CHENG; SEMMELHACK, 2020) os sistemas CA podem ser aplicados à qualquer modelagem física calculada por um computador. Neste sentido, uma das mais nobres aplicações CA se faz na aceleração de operações matemáticas em hardware. Bakhteri utilizou como estudo de caso a implementação do "Jogo da Vida" de Conway em FPGA, comparando-a com soluções similares em software para uso de GPUs. Seus resultados apontaram que a implementação do jogo em um FPGA, com recursos básicos, operou 36.7 vezes mais rápida do que semelhante implementação em GPU e 2.908 mais rápida do que a implementação padrão otimizada para uso regular em um microprocessador.

Halbach et. al. (HALBACH; HOFFMANN; RÖDER, 2004) apresentam um traba-

lho comparativo bem detalhado sobre um sistema CA em chip, e sua implementação em software. Os autores relataram que os esforços utilizados em uma implementação em chip podem ser significativamente maiores, quando comparados com suas respectivas versões em software. Porém os autores concluem que a solução em hardware apresenta-se mais viável, na ordem de milhares de vezes (HALBACH; HOFFMANN; RÖDER, 2004).

Lima e Ferreira (LIMA; FERREIRA, 2013) apresentaram uma proposta de um sistema genérico CA, em FPGA, composto por uma FSM de alto nível, blocos de memória RAM, bancos de registradores, um microprocessador Microblaze, além de uma rede bidimensional de células autômatas e demais componentes (vide Figura 6). O sistema foi projetado para atender, em espaços unidimensionais e bidimensionais, diversos tipos de condições de contorno (entre vizinhos e nas fronteiras do sistema celular), excitações periódicas ou nulas, e a possibilidade, sem restrições, de definição de estados intracelulares. Toda a lógica pode ser programada pelo usuário a partir de operações binárias em cada célula. Mais uma vez, o "Jogo da Vida" de Conway foi utilizado como objeto de estudo para demonstração de resultados.

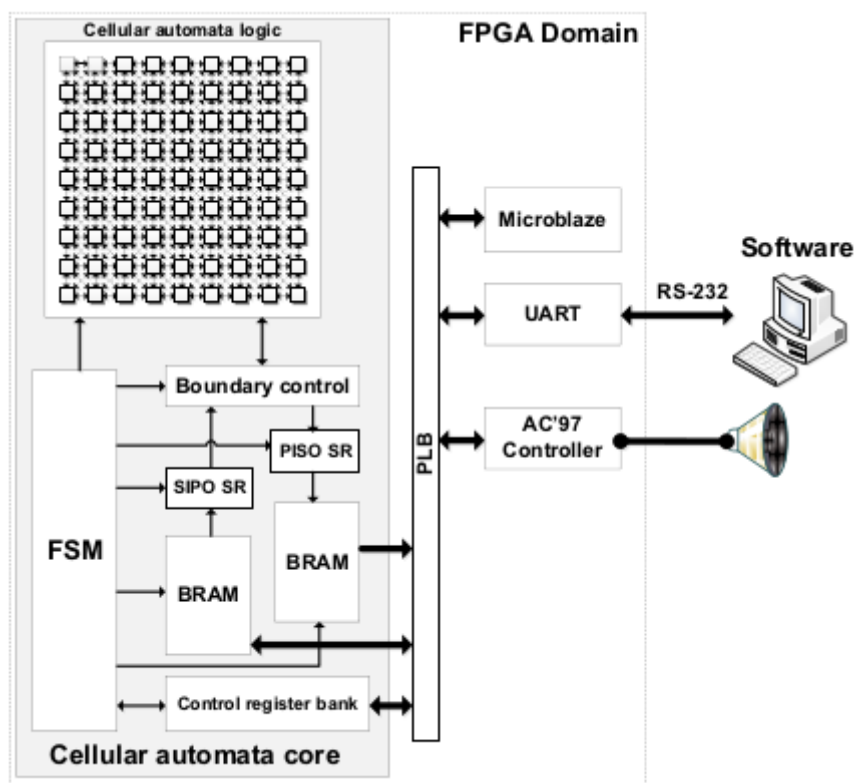


Figura 6 – Arquitetura utilizada no trabalho de Lima e Ferreira (2013).

Os autores obtiveram resultados positivos para a aceleração em hardware do "Jogo da Vida" de Conway, em relação à solução semelhante em software. Em uma grade de 32×32 células a aceleração foi de 39.4 vezes, em uma grade de 40×40 células foi de 74 vezes e, finalizando, em uma grade de 56×56 células foi de 168 vezes. Nota-se que a aceleração

se mostrou diretamente proporcional aos acréscimos sobre o custo computacional. Outros resultados, tais como o relatório de utilização de recursos e a frequência de atualização, também foram apresentados no mesmo trabalho.

De maneira sucinta e resumida é possível verificar, a partir das várias aplicações apresentadas, que o sistema CA necessário para realização da primeira parte deste trabalho deve, muito provavelmente, possuir as seguintes características:

1. células dispostas em um espaço unidimensional com dois vizinhos cada.
2. vizinhança de Von Neumann com dimensão e raio unitário (3 células envolvidas).
3. enorme quantidade de estados para descrição das quantidades de onda em trânsito pelo sistema.
4. transferência de informações por “contato” intercelular, de maneira a representar a dinâmica das ondas acústicas.
5. permitir excitações periódicas ou não nas fronteiras de uma ou mais células (termo fonte).
6. assumir um princípio matemático capaz de definir a dinâmica intracelular.
7. condições de contorno nulas nas fronteiras do sistema.

A última característica é necessária para se considerar condições anecoicas nas fronteiras do sistema. Uma condição anecoica é aquela na qual as ondas se perdem por completo nas fronteiras do sistema, em outras palavras, uma condição na qual toda a energia da onda seja absorvida sem parcelas refletidas ([KUTTRUFF, 2007](#)).

3 Ondas Planas Unidimensionais

Neste capítulo serão apresentadas as equações de movimento e de interferência de ondas com o meio de propagação. Os resultados a seguir foram obtidos da acústica linear, na qual o meio de propagação é considerado ideal e a condução térmica entre os elementos de volume adjacentes (partículas do meio) é desprezada. Desta forma, o processo termodinâmico relacionado à propagação das ondas sonoras, considerado neste trabalho, é isentrópico.

3.1 Equação Linear Unidimensional da Onda

Considerando uma onda plana propagando-se somente na direção positiva de x , é possível escrever (HALL, 1987) (BLACKSTOCK, 2000),

$$\frac{\partial^2 p(x, t)}{\partial x^2} - \frac{1}{c_0^2} \frac{\partial^2 p(x, t)}{\partial t^2} = 0. \quad (3.1)$$

Para esta EDO é possível considerar uma solução expressa em função de duas variáveis linearmente dependentes, $p = p(\xi, \eta)$. Após a substituição desta solução e algumas manipulações matemáticas, é possível encontrar uma solução que representa a superposição de duas parcelas de ondas concorrentes no mesmo ponto do espaço,

$$p = f(x - c_0 t) + g(x + c_0 t). \quad (3.2)$$

Conclui-se assim que a solução para esta equação de movimento dever ser representada por duas novas variáveis, definidas por $(x - c_0 t)$ e $(x + c_0 t)$. Este resultado foi obtido pelo matemático Jean le Rond d'Alembert, em meados do século XVIII.

Considerando o movimento apenas da parcela $(x - c_0 t)$, observa-se que o argumento da pressão sonora $p = p(x - c_0 t) = f(\psi)$ se mantém constante, caso seja mantida uma variação simultânea do tempo e do espaço. A este termo constante da-se o nome de fase de onda. Também podemos expressar a velocidade de propagação da perturbação (onda sonora),

$$c_0 = \frac{\Delta x}{\Delta t}. \quad (3.3)$$

Neste momento é conveniente expressar o valor deste parâmetro em termos das propriedades termodinâmicas do meio,

$$c = \sqrt{\gamma RT}. \quad (3.4)$$

Sabendo que a constante universal dos gases, $R = R_0 = 8314[J/Kg^\circ K]$, $\gamma_{ar} = 1,4$ e considerando a temperatura em 20° , tem-se o valor de $344[m/s]$. Este valor é uma importante referência para a correta definição da frequência de atualização do sistema CA.

3.2 Digitalização da solução de d'Alembert para a Onda

Uma das maneiras mais comuns de se resolver o problema proposto é a digitalização da solução de d'Alembert, expressa na equação 3.2, e visualizada a seguir na Figura 7. A digitalização desta solução permite a sua resolução numérica por FDTF (TREFETHEN, 1994). Deste modo se derivam os desenvolvimentos de MGDO, aplicada à modelagem de sistemas físicos (MURPHY; MULLER, 2002).

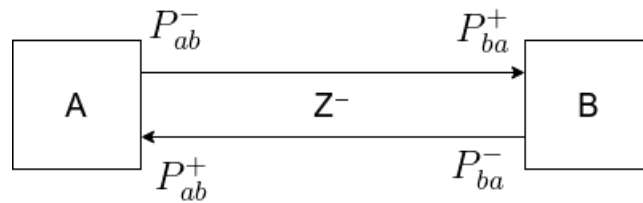


Figura 7 – Digitalização da solução de d'Alembert para a Onda.

A formulação completa da MGDO pode ser encontrada no seu trabalho progênito, desenvolvido por Smith (SMITH, 1992), no CCRMA da Universidade de Stanford/USA. É preciso compreender aqui que a solução de d'Alembert se faz a partir da soma das parcelas em trânsito $p^+(n)$ e $p^-(n)$, de modo a constituir o sinal de pressão sonora $p(n)$ em cada instante de tempo. Também é possível perceber que as parcelas de pressão sonora se cruzam ao longo das guias digitais, durante um atraso de tempo definido pela velocidade de propagação das ondas no meio (intervalo Δt de atualização da malha).

Considerando de modo generalizado as possíveis direções de propagação de ondas, bem como os fenômenos de interação das ondas com o meio, é possível derivar uma equação para atualização dos valores de pressão sonora em cada junta 'J' de um dado domínio discreto,

$$p_J = \sum_{i=1}^{2N} p_i^+ \psi_i. \quad (3.5)$$

Esta equação nos diz que o valor de pressão sonora em cada junta do sistema corresponde à soma das parcelas de pressão sonoras incidentes, provenientes de todas as

direções. Em outras palavras, as guias digitais de ondas são elementos unidimensionais úteis para implementação da solução de d'Alembert ao longo do espaço e, portanto, podem representar sistemas físicos tridimensionais.

Tomemos o exemplo de MGDO bidimensional. Duas direções de propagação de ondas são consideradas para representar as fronteiras norte, sul, leste e oeste de cada junta da malha. Nesta situação temos $N = 2$ guias de ondas cruzando cada junta e, conseqüentemente, 4 parcelas de ondas incidentes, ponderadas pelos seus respectivos coeficientes de transmissão.

A equação 3.5 é a referência matemática principal para o desenvolvimento das células autômatas, pois guarda a essência da solução de d'Alembert e os mecanismos de transmissão de ondas através do meio.

3.3 Modelagem dos fenômenos básicos de onda

Para que os cálculos indicados na equação 3.5 sejam realizados é preciso obter valores, em cada junta da MGDO, para os coeficientes de transmissão, ψ_i . Este trabalho foi realizado por Moura em sua teste de mestrado, apresentada à Universidade Federal de Uberlândia em 2006 (MOURA, 2006). A principal variável, ou variáveis utilizadas neste cálculo são os valores de impedância que caracterizam o meio.

Para a compressão mínima das expressões apresentadas a seguir é preciso uma rápida estruturação do problema da interação de ondas sonoras com os obstáculos presentes no meio. Um obstáculo é tão simplesmente uma mudança significativa de impedância entre dos pontos do meio.

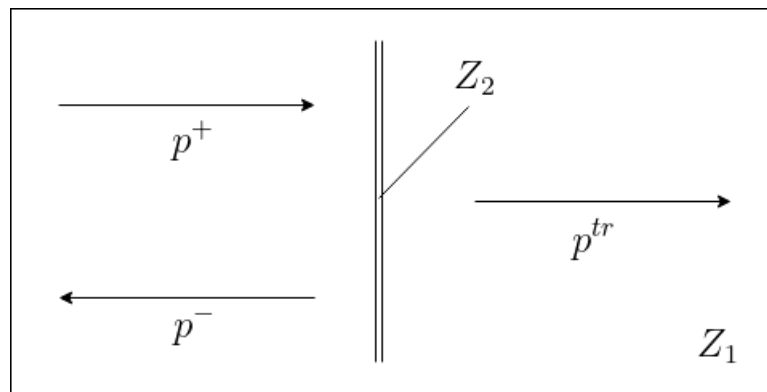


Figura 8 – Fenômenos de transmissão e reflexão de ondas planas unidimensionais.

Na Figura 8 temos as parcelas $p^+ = p^+(x_1 - c_0 t_1)$, incidente, $p^- = p^-(x_1 + c_0 t_1)$, refletida e $p^{tr} = p^+(x_2 - c_0 t_2)$, transmitida através do obstáculo com impedância Z_2 (percebe-se que o meio possui impedância Z_1). Este parâmetro representa a impedância característica do meio e significa a magnitude necessária, à pressão sonora, para que se produza uma

velocidade de partícula unitária no meio (KUTTRUFF, 2007). A impedância característica do meio pode ser determinada pela razão entre a pressão sonora e a velocidade de partículas,

$$\frac{p}{u} = \pm \rho_0 c_0. \quad (3.6)$$

Neste momento devemos considerar uma solução particular para a pressão sonora incidente no obstáculo. Para o caso de ondas monocromáticas, tem-se,

$$p(x, t) = p_0 e^{i[wt - kx]}, \quad (3.7)$$

$$v(x, t) = \frac{p_0}{\rho_0 c} e^{i[wt - kx]}. \quad (3.8)$$

O termo $(wt - kx)$ representa a fase da onda, e permite acoplar as variações de tempo e espaço. Observa-se que a equação 3.8 descreve a velocidade das partículas no meio, de acordo com a equação 3.6. A onda refletida no obstáculo possui uma mudança de amplitude e fase, em função de um coeficiente de reflexão, r . A reversão do movimento da onda é considerada no número de onda, $k = w/c_0$, e o sinal da velocidade das partículas altera também conforme a equação 3.6,

$$p_r(x, t) = r p_0 e^{i[wt + kx]}, \quad (3.9)$$

$$v_r(x, t) = -r \frac{p_0}{\rho_0 c} e^{i[wt + kx]}. \quad (3.10)$$

Para se obter a pressão sonora na superfície do obstáculo basta somar as expressões acima, para $x = 0$,

$$p(0, t) = (1 + r) p_0 e^{i[wt + kx]}, \quad (3.11)$$

$$v(0, t) = (1 - r) \frac{p_0}{\rho_0 c} e^{i[wt + kx]}. \quad (3.12)$$

Estes resultados podem ser substituídos na equação 3.6 para equacionar o valor da impedância característica do obstáculo,

$$Z_2 = Z_1 \frac{1 + r}{1 - r}. \quad (3.13)$$

E a partir deste resultado é possível obter uma expressão para cálculo do coeficiente de reflexão, r , em função dos valores de impedância do meio e do obstáculo,

$$r = \frac{Z_2 - Z_1}{Z_2 + Z_1}. \quad (3.14)$$

O coeficiente de transmissão de ondas, através do obstáculo, é diferente de zero para valores de reflexão menores do que a unidade. Aplicando a lei de conservação de energia, tem-se que a soma da energia refletida e a energia transmitida é igual à energia incidente (absorção nula no obstáculo). Logo, a transmissão de ondas através do obstáculo é dada por,

$$t = \frac{2Z_1}{Z_2 + Z_1}. \quad (3.15)$$

Neste momento, se faz necessária a equação proveniente do equilíbrio das forças geradas pela passagem das ondas acústicas pelo meio. Esta formulação pode ser integralmente encontrada em (SMITH, 1992).

$$\psi_i = \frac{2R_i}{R_J + \sum_{i=1}^{2N} R_i}. \quad (3.16)$$

A equação 3.16, acima, nos permite obter os coeficiente de transmissão necessários à equação 3.5. Nela, os valores de impedância foram renomeados para R_i , tal que $p^+(n) = Ru^+(n)$ e $p^-(n) = Ru^-(n)$. Este resultado é análogo ao encontrado para o tudo de impedância de Morse (1936) e também pode ser apreciado em Markle and Gray (MARKEL; GRAY, 1976).

Agora temos condições de verificar, por completo, a equação 3.5. Combinando as equações 3.5 e 3.16, tem-se a equação 3.17, para a qual temos esboçada a solução na Figura 9.

$$p_J = \left(\frac{2R_1}{R_1 + R_2} \right) p_{left}^+ + \left(\frac{2R_1}{R_1 + R_2} \right) p_{right}^+. \quad (3.17)$$

Através das equações 3.17, 3.16 e 3.15 é possível verificar que as mesmas representam a solução exata do problema de transmissão de ondas, em meio ideal. Vale ressaltar ainda na equação 3.16, que a mesma possui a representação da impedância de junta, R_J . Para que o meio seja ideal, este valor deve ser anulado nesta equação. De maneira contrária, assumindo um valor não nulo para R_J , é possível simular um mecanismo simplificado de absorção sonora (perda energética).

Por fim, Moura apresenta a formulação completa da MGDO em presença de obstáculos, expressa para a velocidade das partículas no meio,

$$v_J(n) = \sum_{i=1}^{2N} \psi_i v_i(n-1) - \left\{ \left[\sum_{i=1}^{2N} \psi_i \right] - 1 \right\} v_J(n-2). \quad (3.18)$$

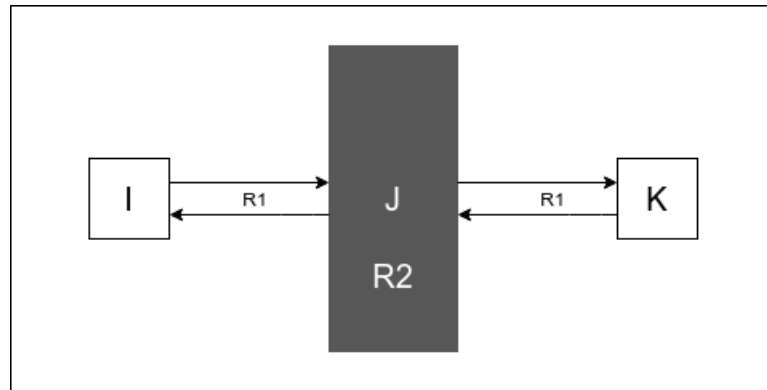


Figura 9 – Representação dos fenômenos de transmissão e reflexão de ondas a partir das guias digitais de ondas.

Na qual a variável n representa o tempo discreto. Os valores de velocidade de partícula, em cada junta J do sistema, evoluem em função de seus vizinhos e de seus valores passados. Para o caso de um termo de fonte, esta quantidade de onda deve ser forçada no ponto. A equação 3.18 foi desenvolvida utilizando-se do método FDTD, e servirá de base para o desenvolvimento do modelo de referência definido no tópico seguinte.

3.4 Modelo de referência

Para definir o modelo de referência foi necessário escolher um sistema físico unidimensional. O sistema escolhido está descrito abaixo:

- comprimento linear da malha igual a 20 metros.
- velocidade de propagação das ondas no meio = $344m/s$.
- impedância característica do meio igual a 400 Rayls.
- uma membrana (obstáculo) com impedância característica igual a 2666.7 Rayls.
- duas membranas nas extremidades da malha, com impedância característica de 1200 Rayls.
- frequência de atualização da malha igual a 3200 Hz.
- tempo total da simulação igual a 0.05 segundos (160 quadros).
- frequência de operação das perturbações igual a 400 Hz. Isso implica em uma representação senoidal com 8 pontos por período.
- Duas perturbações senoidais inseridas em dois pontos das malha, com durações de 1 e 2 ciclos completos.

Parte das definições de sinais e das estratégias algorítmicas aplicadas no código do modelo de referência foi negligenciada neste trabalho, por não fazer parte do escopo do mesmo.

A simulação foi realizada através do pacote G++ do linux, aplicando uma MGDO unidimensional. A Figura 1, apresentada na introdução deste trabalho, representa as geometrias do problema. Uma animação gráfica das soluções temporais obtidas foi realizada no software Octave, a partir de um processo automático de escrita e leitura de arquivos TXT. A animação pode ser apreciada no link a seguir,

<https://youtu.be/lkZPeOSbf7g>

É importante dizer que este modelo de referência é transitório, pois o mesmo será utilizado apenas para validar o sistema CA programado em software. Após tal validação, o código em C utilizado para emular o sistema CA em software passará a ser o modelo de referência definitivo. De posse do modelo de referência definitivo será possível validar os sistemas CA desenvolvidos neste trabalho, descritos em VHDL.

4 Metodologia

Neste capítulo serão apresentadas as etapas necessárias para o projeto de um sistema CA unidimensional, passando por detalhes construtivos até as implementações realizadas em circuito reconfigurável.

4.1 O Sistema CA Camphs1D

Em um primeiro momento, 3 padrões básicos de células autômatas foram projetados com base em observações realizadas sobre o fenômeno em estudo, tendo em vista a equação 3.5. Juntos, os padrões projetados podem emular o comportamento esperado de toda a solução, incluindo os fenômenos de interação de ondas com obstáculos. Em seguida, será apresentado o resultado final da arquitetura projetada, através de um projeto RTL. A arquitetura Camphs1D foi finalizada com apenas 2 tipos distintos de células autômatas, ou seja, 2 componentes descritos em VHDL.

O sistema Camphs1D é composto por 3 células, ou padrões básicos de células: (1) células de dispersão, (2) células geradoras, e (3) células híbridas. Em acordo com a equação 3.5, e tendo em vista a dinâmica esperada para o sistema CA, as seguintes regras foram estabelecidas:

1. As quantidades vetoriais da onda são “transmitidas” através das células de dispersão, de maneira isolada, ao longo de cada direção de movimento ($x+$, $-x$).
2. uma célula geradora não deve receber quantidades vetoriais de onda de sua vizinhança. Seu estado energético é forçado por um agente externo, presente no organismo artificial. Suas quantidades energéticas devem ser transmitidas em todas as direções, a cada instante de tempo.
3. uma célula híbrida recebe quantidades de ondas de sua vizinhança de modo semelhante às células de dispersão. Porém, as mesmas são capazes de replicar as energias recebidas ao longo de uma direção para a outra, de maneira a se comportarem de forma semelhante a uma célula geradora.

Com base nas regras apresentadas foi possível desenhar os três padrões de células autômatas. Os resultados podem ser apreciados nas figuras 10, 11 e 12.

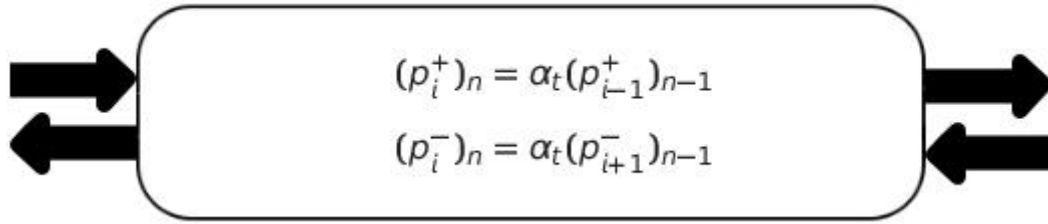


Figura 10 – Primeira versão da célula autônoma de dispersão, utilizada na arquitetura Camphs1D.

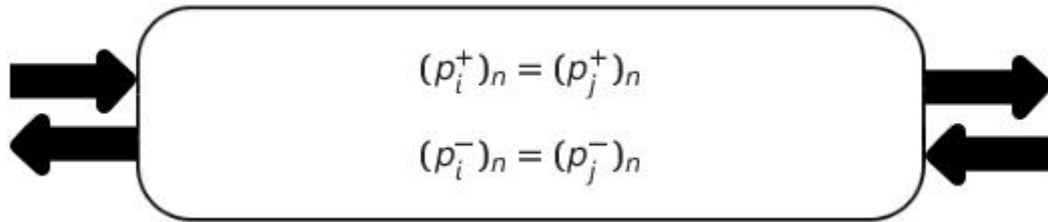


Figura 11 – Primeira versão da célula autônoma geradora, utilizada na arquitetura Camphs1D.

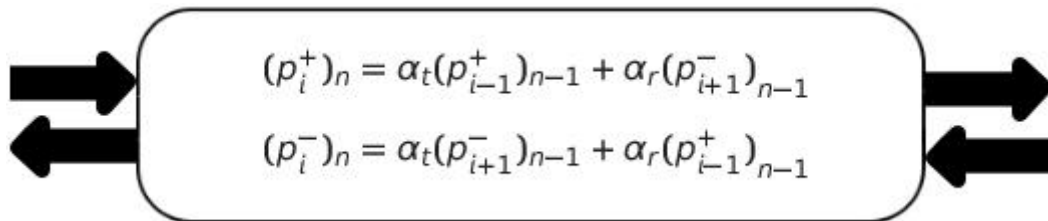


Figura 12 – Primeira versão da célula autônoma híbrida, utilizada na arquitetura Camphs1D.

É possível observar que a dinâmica das células autômatas apresentadas responde de maneira idêntica às regras previamente estabelecidas. O próximo passo é a implementação em software do sistema CA em questão, visando a comparação com o modelo de referência adotado no capítulo 3, para aplicação de uma MGDO.

4.1.1 Solução de software para sistema CA Camphs1D

O principal objetivo de um sistema CA é a simplicidade. Para tal, as células foram diretamente conectadas de maneira a formarem duas guias de ondas, cada uma para uma direção de movimento. Não foram necessários parâmetros de controle ou equações de movimento, apenas as células autômatas e os sinais de fonte.

Outra simplificação considerada foi a divisão da simulação em 3 partes concatenadas. Na primeira parte, duas fontes sonoras são consideradas em um conjunto de 66 células autômatas, composto por 64 células de dispersão e 2 células geradoras - as demais células foram desabilitadas. Na segunda parte, 82 células autômatas foram utilizadas,

Tabela 1 – Consumo de memória em bits das estratégias MGDO e CA. A sigla I_{32} representa inteiros com 32 bits, F_{32} representa pontos flutuantes em 32 bits, N_J representa o número de nós ou juntas de dispersão e a sigla N_s representa o número de passos temporais.

Estratégia	Escalares	Vetores	Total
MGDO-FDTD	$4 \times I_{16} + 8 \times F_{32}$	$9 \times N_J + 3 \times N_s$	2483 bits
sistema CA	$4 \times I_{16} + 8 \times F_{32}$	$6 \times N_J$	1442 bits

tendo apenas 1 célula geradora no conjunto - uma das células geradoras interrompe o seu funcionamento se transformando em uma célula de dispersão. Na parte final, 187 células autômatas foram utilizadas (185 em funcionamento e duas desligadas), dentre as quais temos 3 células híbridas e o restante em células de dispersão. As células híbridas sempre estiveram presentes no sistema, porém desabilitadas até o início da terceira parte desta simulação. Após a implementação de todo o código (Camphs1D) foi possível realizar uma análise de consumo de memória. Os resultados se encontram na Tabela 1.

Na Tabela 1, visando uma comparação mais justa, somente as variáveis essenciais, às duas estratégias, foram contabilizadas. O consumo em escalares é idêntico por estar relacionado com os parâmetros básicos da simulação, tais como os sinais de fonte e as geometrias envolvidas. As estratégias se diferem significativamente em relação aos vetores, pois os mesmos representam as equações envolvidas e seus coeficientes.

Ainda na Tabela 1, as variáveis I_{16} e F_{32} representam o inteiro de 16 bits e o ponto flutuante de 32 bits, respectivamente. As variáveis N_J e N_s representam quantidades de pontos flutuantes - foram utilizados, nestas simulações, $N_J = 187$ e $N_s = 160$. Deste modo, percebe-se um menor consumo de bits para o sistema CA, pois o mesmo se utiliza de uma quantidade menor de vetores envolvidos em seu cálculo.

A comparação entre o sistema CA implementado em C e a estratégia da MGDO, também implementada em C, foi satisfatória. A precisão encontrada na comparação foi superior a 7 casas decimais.

4.1.2 Análises de perfil das soluções por MGDO e CA

As análises de perfil para as soluções por MGDO e CA foram realizadas utilizando-se do compilador G++, do linux, e da ferramenta GPROF. Para facilitar a visualização e a contagem de tempo das execuções unitárias de cada programa, os mesmos foram executados de maneira automática por 1000 vezes. Cada execução corresponde a 160 evoluções temporais de um sistema composto por 185 células autômatas. Um resumo se encontra na Tabela 2, a seguir.

Vale ressaltar aqui a relevância do projeto abordado neste trabalho. Como se pode observar, os procedimentos de cálculo apresentados são densos, ainda que em um caso

Tabela 2 – Análise de perfil das soluções MGDO e CA.

Parâmetro	Solução por FDTD	Solução por CA
Númerod de execuções	1000	1000
Tempo total de execução	3.36 seg.	3.68 seg.
Qtde de multiplicações	89446	57121
Qtde de somas	59526	864
Qtde de divisões	9	0

simplificado. Em uma simulação padrão aplicada a uma sala com aproximadamente 16 m^2 e pé direito de 2.5 m, podemos ter uma quantidade de juntas/células da ordem de 6 dígitos¹. Ou seja, trata-se de uma aplicação com elevado custo computacional, necessitando de paralelização e/ou acelerações em hardware.

4.1.3 Projeto do sistema CA Camphs1D

Após da definição do conjunto de regras deste sistema CA, e da definição matemática de suas células, o projeto RTL foi iniciado. Em paralelo, cada célula foi descrita em VHDL e testada através de simulações comportamentais. Este estágio foi importante para a produção do CORE CA, utilizando-se dos IP-cores aplicados às multiplicações e somas com representação aritmética de ponto flutuante em 27 bits. Os IP-cores de cálculo aritmético em ponto flutuante em 27 bits (FPUPACK27), foram desenvolvidos no laboratório LEIA/GRACO da Universidade de Brasília (MUÑOZ et al., 2010a) (MUÑOZ et al., 2010b).

4.1.3.1 Blocos operacionais: células autômatas

A seguir serão apresentados os blocos operacionais de cada uma das células desenvolvidas. Neste estágio do projeto, os três padrões de comportamento celular idealizados foram convertidos em duas únicas células, denominadas “células autômatas globais” e “células autômatas híbridas”. O termo “global” se deve ao fato destas células representarem a maior parte da arquitetura principal do sistema CA.

Começamos pelo resumo das portas consideradas em cada um dos componentes (células autômatas). As informações estão dispostas na Tabela 3, a seguir. Um maior detalhamento de todos os sinais apresentados nestas tabelas poderá ser encontrado no capítulo 4.

O bloco operacional da célula autômata GCA1D está apresentado na Figura 15, a seguir. Esta célula pode executar os trabalhos de geração de ondas e dispersão de ondas, geralmente executados nesta ordem. Algumas células GCA1D nunca são solicitadas como geradoras de ondas, naturalmente por não ocuparem posição de fontes no sistema.

¹ Para simular cenários acústicos reais precisamos, no mínimo, cobrir a faixa de frequências ($20Hz \approx 10KHz$). Para isso, precisamos de uma frequência de atualização de malha igual a $20KHz$.

Tabela 3 – Sinais das utilizados nas células Autômatas GCA1D e HCA1D.

Porta	IN/OUT	Bits	Função
RESET	IN	1	Inicialização de registradores
MODE	IN	2	Geração, dispersão ou “OFF ”
SOURCE	IN	27	Valor imposto (fonte)
GAIN	IN	27	Ganho interno (fonte)
SCE_STIM	IN	1	Estímulo para geração
LEFT_STIM	IN	1	Estímulo para dispersão
RIGHT_STIM	IN	1	Estímulo para dispersão
LEFT	IN	27	Coef. transm. de ondas à esq.
RIGHT	IN	27	Coef. transm. de ondas à dir.
LEFTIN	IN	27	Valor de ondas à esquerda
RIGHTIN	IN	27	Valor de ondas à direita
RIGHT_READY	OUT	1	Disponibilidade da info.
LEFT_READY	OUT	1	Disponibilidade da info.
RIGHTOUT	OUT	27	Valor de saída à dir.
LEFTOUT	OUT	27	Valor de saída à esq.
CROSSLEFT	IN	27	Coef. de geração de ondas
CROSSRIGHT	IN	27	Coef. de geração de ondas
CLK	IN	1	Funcionamento síncrono

O projeto do componente GCA1D, apresentada na Figura 13 utiliza 3 multiplicadores em ponto flutuante em 27 bits, 2 MUXs 3×3 , 2 portas lógicas OR, e um registrador de 1 bit. Também temos um banco registrador de 27 bits armazenando uma constante, normalmente configurada para o valor nulo.

O componente GCA1D opera, de maneira síncrona, em quatro ciclos de relógio, para atender à latência da arquitetura principal. A arquitetura principal promove a integração das células, dentro do organismo artificial CA, em um ambiente funcional programado com sinais externos e recursos para entrada e saída de dados.

Em seguida temos apresentado também na Figura 15 o bloco operacional da célula autômata HCA1D. Esta célula não realiza trabalho duplo, mas híbrido, operando dispersões de ondas e interação com obstáculos. Em seu projeto, foram utilizados 4 multiplicadores e 2 somadores. De maneira semelhante à GCA1D, esta célula autômata também opera com 4 ciclos de relógio.

4.1.3.2 Arquitetura principal: integração de componentes

A arquitetura a seguir é responsável pela integração celular, de maneira a automatizar o funcionamento do sistema CA. Os sinais de entrada e saída de informações devem ser controlados pela arquitetura maior, também chamada de *organismo CA artificial*. O organismo CA artificial será representado, neste trabalho, por um arquivo de simulação (*Testbench* em VHDL).

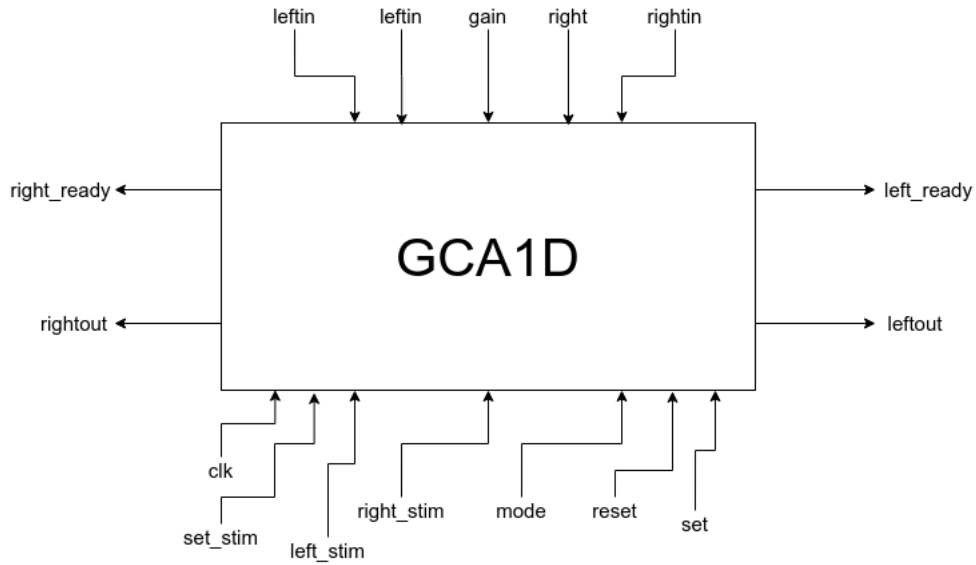


Figura 13 – Célula autômata GCA1D. Para representarmos a célula autômata HCA1D, temos que retirar os sinais SET_STIM, SET, GAIN e MODE e acrescentar os sinais CROSS_LEFT e CROSS_RIGHT.

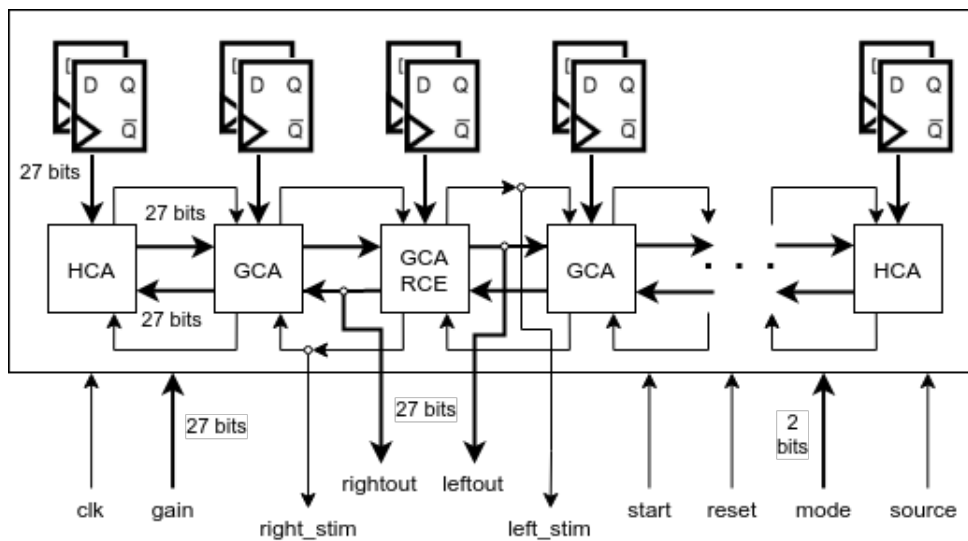


Figura 14 – Bloco operacional do sistema CA Camphs1D. Neste exemplo, temos uma célula receptora, indicada pelas siglas GCA RCE, externalizando seus sinais *rightout* e *leftout*.

A arquitetura principal está apresentada na Figura 14, acima. A título de exemplo, temos nesta arquitetura uma integração aleatória com apenas uma célula HCA1D.

Nesta arquitetura principal temos ainda bancos de registradores para cada uma das células autômatas. Tais informações representam os coeficientes utilizados pelo sistema CA, capazes de decodificarem as informações de impedância para as células autômatas.

Os sinais CLK, GAIN, RESET, MODE (2 bits) e SOURCE (27 bits) são diretamente passados para as células autômatas correspondentes. Cada célula autômata se

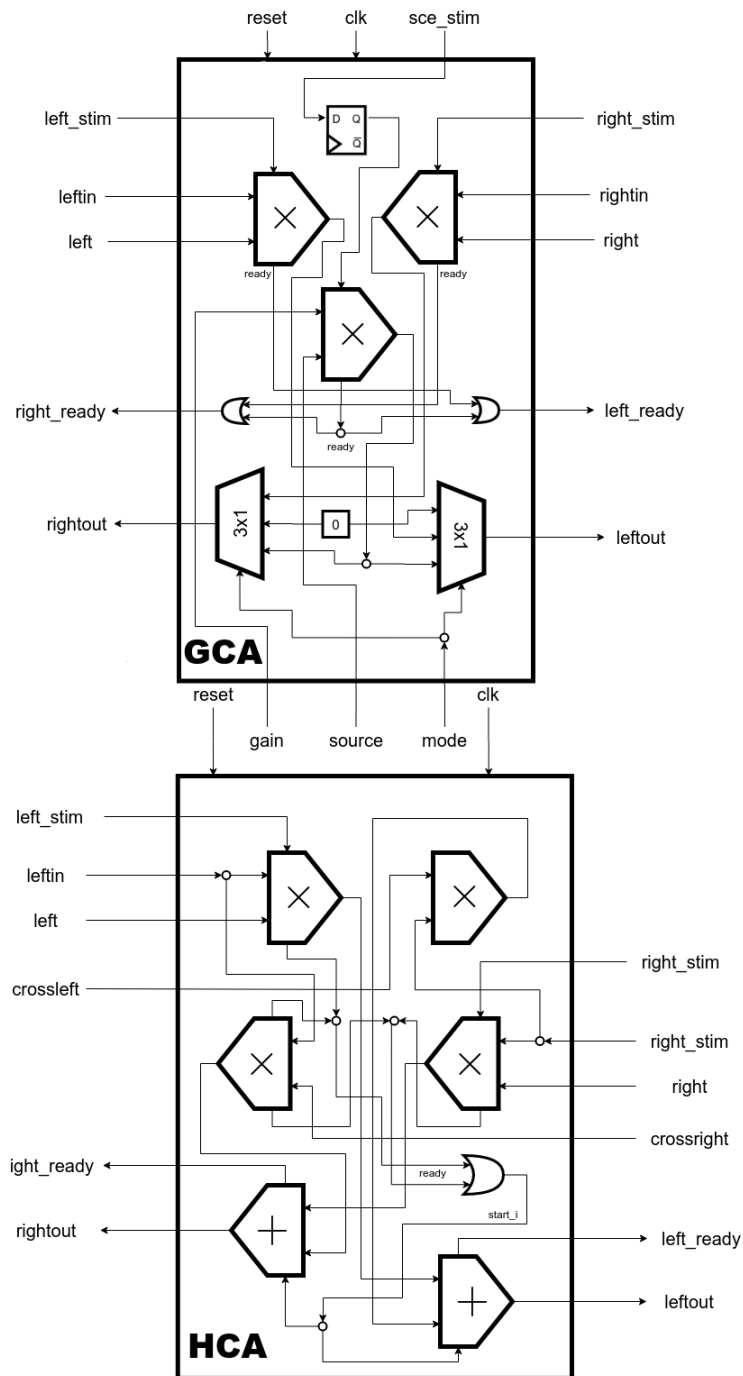


Figura 15 – Blocos operacionais das células autômatas GCA1D e HCA1D. As células são compostas de multiplicadores e somadores em ponto flutuante de 27 bits, além de portas lógicas, multiplexadores e registradores.

comunica com sua vizinhança através de 4 sinais, sendo 2 sinais de READY (1 bit para cada direção de propagação) e 2 sinais de onda (27 bits). Falta mencionar o sinal de START, que foi estabelecido para iniciar todas as células GCA1D em modo de geração.

Deste modo, uma célula autômata só opera seus cálculos quando estimulada pela vizinhança, em uma direção específica. Esta característica representa uma vantagem para a implementação em hardware, pois o sistema CA projetado trabalha de modo a permitir

o “descanso” de células autômatas. Em termos de software, o mesmo poderia ser obtido a partir do uso de matrizes esparsas, porém com uso de lógica mais complexa.

4.1.4 Notas sobre o desenvolvimento e aplicações do sistema CA Camphs1D

Inicialmente, um sistema CA tem as suas células desenhadas para operarem da forma mais simples e rápida possível, evitando a necessidade de lógica adicional e demais recursos que possam consumir ciclos extra de relógio.

Após os testes unitários, usando de simulação comportamental, a célula autômata GCA1D apresentou uma operação em apenas 3 ciclos de relógio, em sua atividade de geração de ondas. Por outro lado, a célula autômata HCA1D, por envolver maior quantidade de operações, apresentou uma latência de 4 ciclos de relógio.

Para atender ao sincronismo desejado, foi adicionado à célula autômata GCA1D um atraso de 1 ciclo de relógio (registrador intermediário). Deste modo, a célula passou a operar com 4 ciclos de relógio - independentemente de estar em modo de dispersão ou geração.

Foi dada à célula GCA1D a habilidade de permanecer desativada durante o funcionamento do sistema CA. Desta forma, uma célula autômata GCA1D pode absorver 100% da energia incidente, sem causar prejuízos ao funcionamento das células vizinhas. Este recurso é útil para definição de condições perfeitamente anecoicas, fato que é difícil de conseguir em técnicas baseadas na resolução numérica de EDO.

4.2 Projeto das simulações comportamentais

Em primeiro lugar vamos nos aprofundar nos detalhes do sistema CA projetado. Trata-se de um circuito sequencial, síncrono, capaz de transmitir, pelo tempo que for preciso, os sinais de onda através de uma malha de componentes interligados (células autômatas). O sistema CA projetado nesta primeira parte do trabalho não necessita de uma FSM, mas sim de sequências binárias que indicam o modo de funcionamento das células geradoras. Todas as demais células autômatas operam conforme sua natureza, a cada ciclo do sistema, sem a necessidade de qualquer intervenção externa.

As mencionadas sequências binárias representam os valores para os sinais de controle deste sistema CA. O primeiro deles, denominado MODE, pode provocar uma mudança no modo de operação das células GCA1D, fazendo com que as mesmas parem de gerar ondas, para apenas transmitirem sinais. O caminho reverso também é possível.

O segundo sinal de controle a ser mencionado é o RESET. Este sinal é de extrema importância para o correto funcionamento do CORE das células autômatas. Todas as células são compostas por somadores e multiplicadores em 27 bits, e suas portas de

saída estão permanentemente conectadas às portas de entrada das células vizinhas. Deste modo, quando uma célula autômata é estimulada, em qualquer direção, a mesma retira as informações de entrada de suas células vizinhas e processa as suas saídas.

Em outras palavras, o CORE das células autômatas não identifica se a onda está se movendo da esquerda para a direita ou da direita para esquerda, ele apenas processa os dados que estão nas suas portas de entrada. Por este motivo, o sinal de RESET deve garantir que todas as portas de saída, de todas as células autômatas, sejam carregadas de zeros. Deste modo, o sistema todo irá funcionar adequadamente na presença de ondas se movendo em qualquer direção.

Existem outros sinais de extrema importância para o sistema CA, que são os estímulos para que as células autômatas trabalhem. Os estímulos indicam a direção de propagação das ondas sonoras, fazendo com que uma informação seja impulsionada na direção correta, desde a fonte até à sua completa dissipação nas fronteiras do sistema CA².

Felizmente, os sinais de estímulo não precisam ser controlados. Os estímulos precisam ser, apropriadamente, conectados entre as células, para depois operarem simplesmente conforme os sinais de READY das mesmas.

Uma célula autômata GCA1D possui 3 sinais de estímulo, sendo um para uma possível atividade de geração e outros 2 para suas direções de propagação de ondas. As células autômatas HCA1D possuem apenas 2 sinais de estímulo. As células HCA1D são recomendadas para modelagem de fronteiras e obstáculos, enquanto as células GCA1D representam o meio de transporte de ondas.

Neste momento o conceito completo de um “organismo artificial” pode ser completamente entendido. Trata-se de um ambiente que encapsula uma arquitetura principal (integradora de células autômatas) fornecendo-a sinais de controle adequados para a operação automática. Conforme o pensamento de Von Neumann, um sistema CA representa, matematicamente, um organismo natural dotado de funções primitivas, básicas para a sua existência, movido sem a necessidade de uma lógica complexa.

Dois organismos artificiais foram testados através de simulações comportamentais. O primeiro, denominado Camphs1D05G1, foi utilizado para descrever completamente o funcionamento do sistema Camphs1D, enquanto o segundo, Camphs1DB9G2, foi utilizado em contraste com o modelo de referência adotado neste trabalho. Simulações automáticas com leitura e escrita de arquivos foram aplicadas nos dois casos para decodificação e observação dos resultados obtidos.

² A dissipação de energia sonora ocorre nas fronteiras devido aos fenômenos de transmissão de ondas. Somente as parcelas refletidas nas fronteiras do sistema retornam energia para o mesmo.

4.2.1 Organismo artificial Camphs1D05G1

O primeiro organismo artificial CA testado possui uma arquitetura integradora com cinco células autômatas, sendo 2 HCA1D e 3 GCA1D. A célula autômata central foi utilizada como termo fonte, durante o instante inicial de operação do sistema. Foi possível observar, através de uma simulação automática com leitura e escrita de arquivos, os sinais de saída em uma das células do sistema.

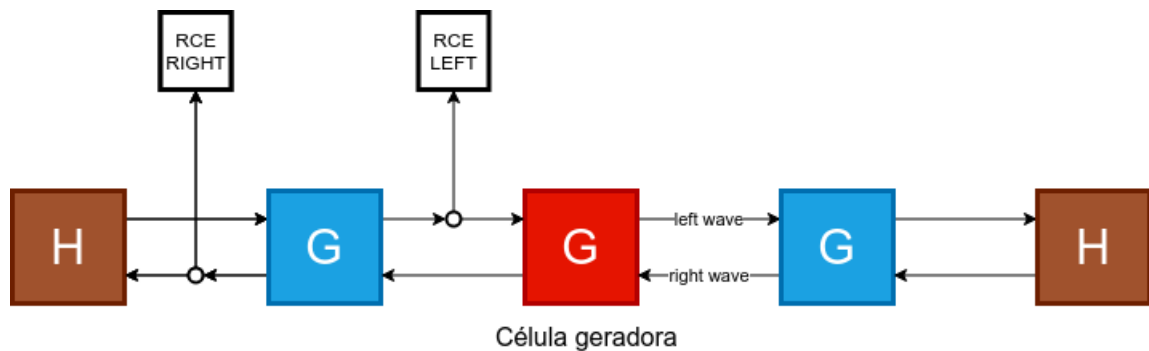


Figura 16 – Esboço da arquitetura Camphs1D05G1. Temos 5 células autômatas, distribuídas em dois tipos(G/H), e 2 sinais receptores (RCE).

4.2.1.1 Diagrama de tempo: sinais de controle para Camphs1D05G1

Os sinais de controle utilizados na simulação comportamental do organismo Camphs1D05G1 estão apresentados na Figura 17, a seguir. Nota-se que o sinal de RESET foi disparado logo no início da simulação, seguido por 2 sinais de START, espaçados de 4 ciclos de relógio. Este espaçamento é necessário para suportar a latência das células autômatas.

Os sinais MODE, START e SOURCE foram fornecidos através da leitura automática de arquivos TXT. A leitura é realizada até que o final do arquivo seja encontrado, sempre na borda de subida do relógio - todo o sistema opera na borda de subida do relógio.

A transição do sinal MODE (“10 ”= geração, “01 ”= dispersão, outros = “OFF ”) é efetuada após 10 ciclos de relógio, modificando o comportamento da célula autômata geradora. Deste modo, quarto ciclo de relógio depois do segundo sinal de START, esta célula autômata necessariamente passará a ignorar o sinal SOURCE, para se portar com uma célula de dispersão.

O sinal SOURCE foi programado para inserir o valor $20d0000_h$ (20_d) durante 5 ciclos de relógio. Em seguida este valor se altera permanentemente para zero, antes do próximo disparo da célula geradora. O ganho foi fixado em 2090000_h (10_d).

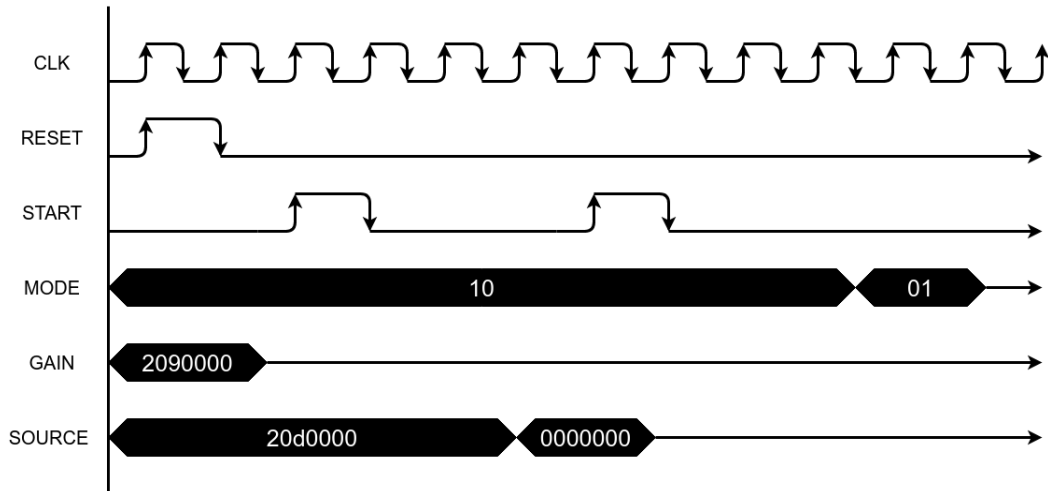


Figura 17 – Diagrama de tempo utilizado no organismo artificial Camphs1D05G1.

4.2.2 Organismo artificial Camphs1DB9G2

O organismo artificial Camphs1DB9G2 foi projetado com 187 células autômatas, submetido às mesmas condições de contorno do modelo de referência. Duas células GCA1D foram utilizadas em modo de geração, de maneira semelhante ao modelo de referência. Outras 3 células HCA1D foram utilizadas para representar os obstáculos existentes no modelo de referência. Todos os cuidados em relação aos sinais de controle, incluindo os sinais fornecidos às células geradoras, foram estabelecidos para que o mesmo comportamento pudesse ser estimulado.

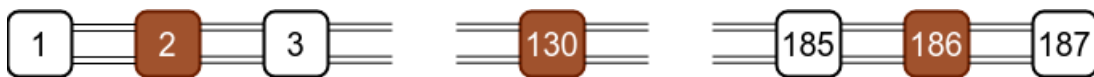


Figura 18 – Esboço da arquitetura Camphs1DB9G2.

A célula autômata utilizada como receptor foi a de número 20. As células autômatas geradoras foram as de número 44 e 94. As células autômatas 1 e 187 ficaram desligadas durante todo o tempo (podem ser eliminadas deste sistema).

Foram gerados 2 sinais senoidais em acordo com a frequência de amostragem simulada pelo sistema CA ($f_s = 3200Hz$). Os valores discretos em PF 27 bits, para estes sinais, estão apresentados na tabela 4.

4.2.2.1 Diagrama de tempo: sinais de controle para Camphs1DB9G2

Os sinais de controle utilizados na simulação comportamental do organismo artificial Camphs1DB9G2 estão mostrados na Figura 19, a seguir. Neste caso, são 2 células autômatas geradoras, por isso temos 2 sinais START, 2 sinais MODE e 2 sinais SOURCE, denominados pelo índice das respectivas células geradoras.

Tabela 4 – Amostragem de um sinal senoidal de $400Hz$ a $3200Hz$, para operação das células geradoras no organismo artificial Camphs1DB9G2.

Célula autômata (PFU 27bits)	Hexadecimais	Seno 400 Hz (3200 Hz)
001111110011010100000100111	1F9A827	0.70710563659668
001111111000000000000000000	1FC0000	1
001111110011010100000100111	1F9A827	0.70710563659668
000000000000000000000000000	0000000	0
101111110011010100000100111	5F9A827	-0.70710563659668
101111111000000000000000000	5FC0000	-1
101111110011010100000100111	5F9A827	-0.70710563659668
000000000000000000000000000	0000000	0

Nota-se que os sinais de fonte foram carregados com os valores apresentados na tabela 4. O tempo de operação da célula geradora 44 corresponde ao dobro do tempo da célula 94. Os sinais MODE foram definidos adequadamente para permitirem o funcionamento das células geradoras.

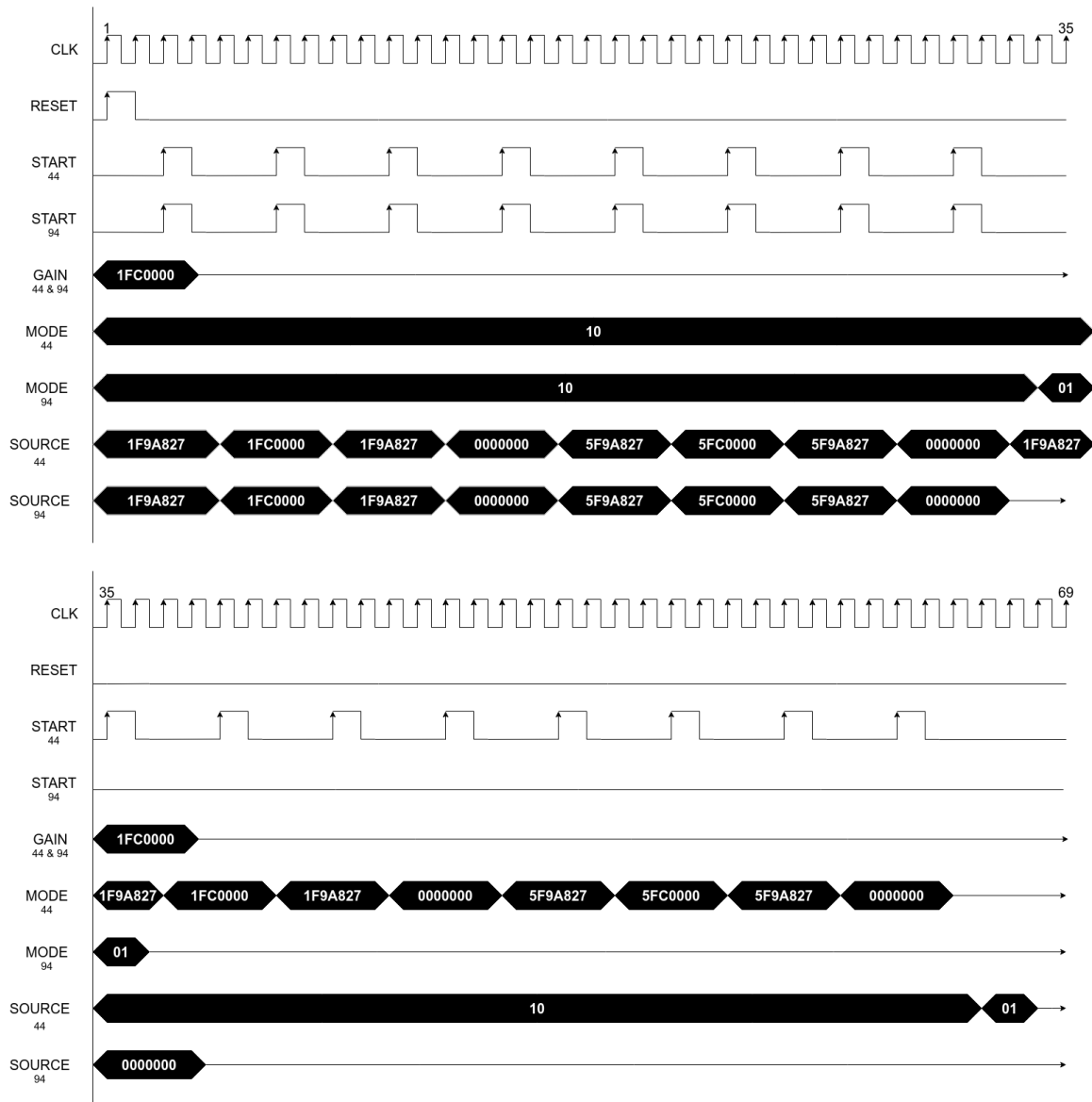


Figura 19 – Diagrama de tempo utilizado no organismo artificial Camphs1DB9G2.

4.2.3 Padronização do sistema Camphs1D: rotina vCAgen

As arquiteturas principais, constituídas nos organismos artificiais Camphs1D05G1 e Camphs1DB9G2, são responsáveis por toda a integração de componentes. A arquitetura principal de Camphs1D05G1 totalizou cerca de 200 linhas de código VDLH para apenas 5 componentes. Esta verificação levantou a necessidade de se padronizar o processo de construção de novas arquiteturas, pois a quantidade excessiva de linhas pode inserir dificuldades e riscos de erros em um processo manual de descrição.

Deste modo, foi gerado uma rotina computacional dedicada à realização de uma arquitetura padrão para sistemas Camphs1D, em linguagem VHDL - denominada vCAgen. A arquitetura principal de Camphs1DB9G2 foi gerada automaticamente a partir deste procedimento, resultando em um código com mais de 4200 linhas. A rotina vCA-

gen foi desenvolvida no software Octave, e possui cerca de 100 linhas orientadas para a definição dos sinais e geometrias utilizadas na construção de uma arquitetura principal. Através deste processo de padronização, é possível realizar, automaticamente, as seguintes tarefas:

- definição de todos os sinais utilizados
- instanciação de todos os componentes (células autômatas).
- atribuição de todos os sinais de controle, visando a integração celular.
- atribuição de sinais de fonte, para o controle do comportamento de uma ou mais células autômatas globais geradoras.
- atribuição de sinais receptores, para externalização das saídas de uma ou mais células autômatas globais de dispersão.

4.2.4 Validação das simulações comportamentais

As simulações comportamentais foram realizadas através de *Test Benches*, desenvolvidos em VHDL. As informações pertencentes à cada fonte inserida no sistema foram alimentadas através de um processo de leitura automática de arquivos TXT, em acordo com o respectivo diagrama de tempo.

Os resultados das simulações foram escritos em arquivos de texto (TXT), a cada borda de subida do relógio. Tais resultados englobam todos os registros dos sinais de estímulo e saídas de valores, em PF 27 bits, da(s) célula(s) receptora(s). Os valores de saída estão, neste estágio, divididos em 2 canais, referentes às ondas transmitidas nos sentidos esquerdo e direito.

Em software, os arquivos de saída são decodificados e processados de maneira a se produzir um único vetor com os quantidades resultantes das ondas passantes na(s) célula(s) receptora(s). Os principais trechos do arquivo utilizado na decodificação das saídas mencionadas podem ser apreciados no capítulo seguinte.

Nas simulações praticadas, os arquivos de saída dos organismos artificiais devem ser re-amostrados através de um sinal de relógio sincronizado de duas formas possíveis: (1) através do sinal READY da(s) célula(s) receptora(s) ou (2) através de um sinal de relógio sincronizado com a resposta do sistema CA. Este procedimento é necessário para se desprezar valores registrados durante a latência do sistema.

A partir de análises gráficas foi possível validar os resultados, juntamente com o MSE obtido entre as soluções em VHDL e linguagem C.

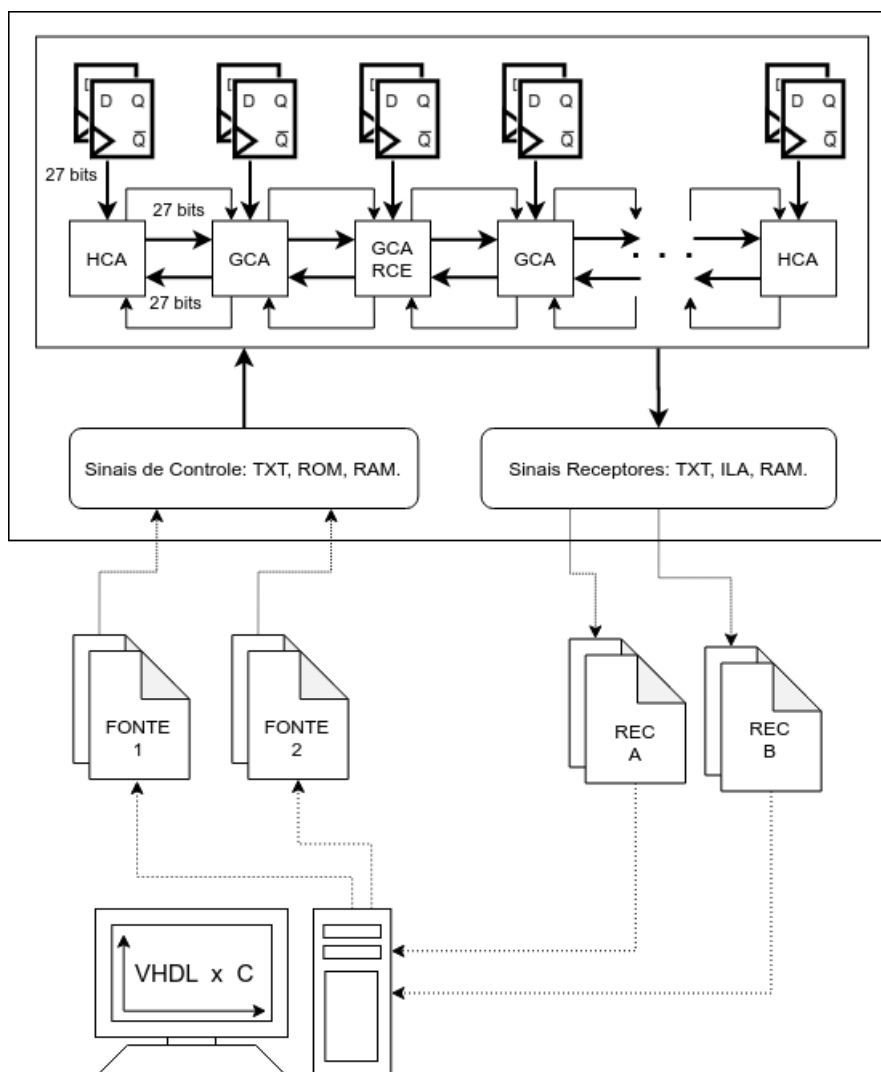


Figura 20 – Diagrama esquemático representativo da validação de Camphs1DB9G2. Para maior detalhamento dos sinais envolvidos na arquitetura principal consultar Figura 14.

4.3 Projeto das implementações em placa

Em diversas situações de projeto, temos a necessidade de criar, encapsular e utilizar IP-cores que representem o funcionamento específico de funções/algoritmos matemáticos, para uso embarcado em FPGA. Desta forma, a primeira implementação do sistema foi realizada, através da integração dos IP-cores Block Memory Generation (BMG), Integrated Logic Analyser (ILA) e Virtual Input/Output (VIO), junto ao organismo artificial Camphs1D05G1 (IP-cores já pré-definidos na ferramenta). Nesta direção, foi formatada a primeira estratégia de implementação, apresentada na Figura 21.

Nesta estratégia, temos a construção de um organismo artificial que embarca o CORE das CA e unidades de memória, para que todos os sinais de controle sejam adequadamente administrados, em placa. Temos também a inclusão dos IP-cores BMG (Block

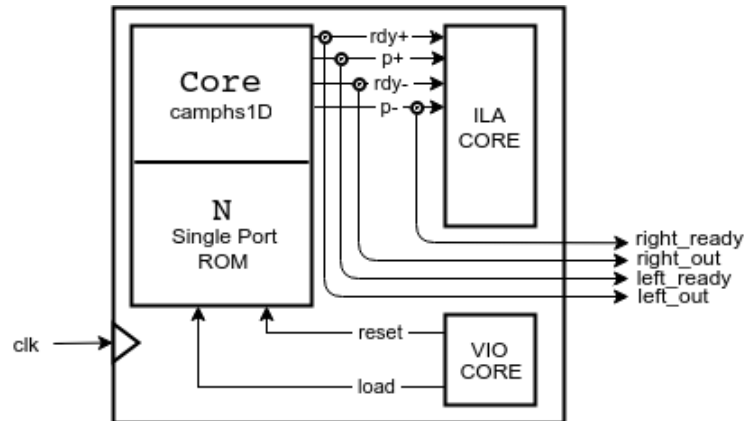


Figura 21 – Esboço da primeira estratégia de implementação de Camphs1D. Temos a necessidade de integrar o core CA com uma unidade de memória ROM. Demais IP-cores são necessários para o funcionamento e *debug* de sinais.

Memory Generator) e ILA (Integrated Logic Analyser), possibilitando a manipulação do sistema CA e depuração de resultados. Trata-se de um sistema síncrono, devido as necessidades requeridas pelo(s) componente(s) de memória ROM, fornecidos pelo IP-core BMG. Os 4 sinais de saída apresentados se referem aos sinais de READY, LEFT_OUT e RIGHT_OUT de uma única célula receptora (mais células poderia ser consideradas, a depender do kit FPGA utilizado), mas poderiam ser suprimidos.

Pensando de uma maneira mais estruturada, o organismo Camphs1D05G1 também poderia ser configurado e encapsulado como um IP-core, para uso em projetos sistema em chip (SoC) FPGA. Um projeto SoC FPGA nos permite integrar recursos de hardware e software, em uma mesma placa FPGA, com o intuito de ampliarmos o volume de dados considerados nos projetos. Os recursos de software são obtidos a partir da disponibilidade, em silício, de um microprocessador ARM, integrado aos recursos lógicos programáveis de um kit FPGA.

A segunda estratégia vem explorar estas funcionalidades, através da ferramenta Vivado HLS C-level, disponível no Vivado 2018.3 HLs Webpack, da fabricante Xilinx. No próximo tópico, serão apresentadas as vantagens e limitações do uso da ferramenta mencionada, alternativamente ao uso direto da linguagem VHDL.

Com relação à placa de desenvolvimento, foi utilizada uma ARTY Z7-20, da fabricante Xilinx. Dentre os seus principais recursos, nos interessa destacar o seu processador ARM ZYNQ SoC 650MHz Dual-Core Cortex-A9, integrado a uma parte lógica programável equivalente a uma Artix-7 FPGA. Esta placa possui 512 MB de memória DDR3, com barramento de 16 bits @ 1050 MBps.

4.3.1 Uma breve descrição da ferramenta Vivado HLS C-level

A fabricante de chips Xilinx oferece, como alternativa para o desenvolvimento de IP-cores, a ferramenta Vivado HLS C-level. Através desta ferramenta é possível desenvolver um IP-core a partir de um código C/C++, porém com alguns cuidados ou restrições em relação ao seu uso. As principais limitações estão listadas abaixo.

- Não podemos trabalhar com alocação dinâmica de memória.
- Não é permitido trabalhar com funções recursivas.
- Não é permitida a utilização de funções de leitura e entrada de arquivos (I/O).
- não podemos trabalhar com a biblioteca padrão para *containers* tais como vetores, mapas de busca otimizada, entre outros (STD).

Em contraste com estas limitações, temos os benefícios gerados pelo uso da ferramenta Vivado HLS C-level. Dentre os principais, podemos listar:

- Maior produtividade para desenvolvedores de hardwares.
- Disponibilidade de recursos otimizados para cálculos em pontos flutuantes, com 32-bits ou 64-bits.
- Possibilidade de verificação em nível de software (C/C++) e co-simulação, considerando a implementação RTL.
- Possibilidade de se trabalhar com diretrizes programáveis capazes de configurar e aumentar a eficiência da síntese lógica.

O projeto de um IP-core se inicia, no Vivado HLS C-level, com a codificação de um arquivo C/C++ que traga a função matemática em foco, incluindo a operação de parâmetros de entrada, que podem ser deste de endereços de memória (ponteiros) até estruturas complexas de dados. As funções podem ser do tipo void (sem retorno de dados), ou de tipos padrões tais como char, short, int, float, double, dentre outros. Uma vez definidas, tais funções podem ser testadas através de um segundo arquivo C/C++, descrito para operar como testador (Test Bench). A estrutura básica de um projeto no Vivado HLS está apresentada na Figura 22.

Na Figura 22 é possível verificar que o Vivado HLS possui os arquivos organizados em uma árvore padrão, de maneira a separar a lógica de funcionamento do IP-core da lógica de teste, que fica na aba *Test Bench*. A aba denominada *solution1* - neste nome é fornecido pelo usuário no início de um novo projeto - guarda todos os arquivos necessários

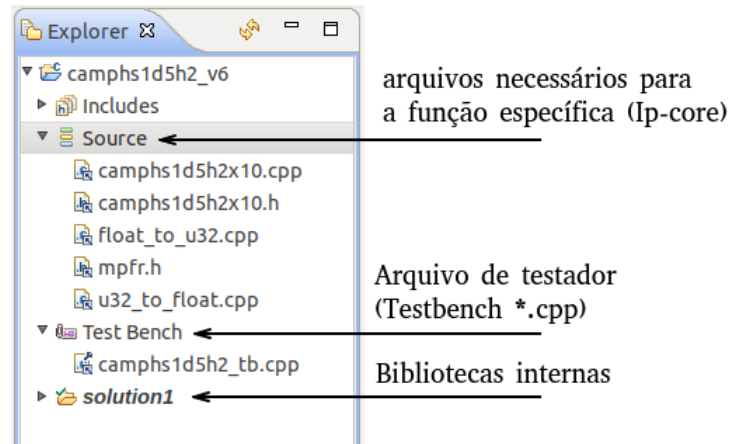


Figura 22 – Estrutura básica de um projeto gerado no Vivado HLS C-level.

para a sintetização do IP-core, bem como as bibliotecas necessárias para a operação de interfaces de dados em nível C, tais como o *AXI4-Lite* e o *AXI4-Stream*.

Após a codificação do IP-core, e de seu arquivo de teste, o Vivado HLS C-level nos oferece alguns recursos para depuração da solução. Os principais recursos são: (1) a depuração em software, a partir da compilação e execução do código, no console da ferramenta, e a (2) co-simulação software/hardware, que sintetiza a solução em RTL, comparando-a com a solução descrita em software. Existem ainda recursos mais avançados de depuração, tal como análise dos sinais de ondas, que podem nos oferecer uma visão mais concreta sobre o funcionamento do circuito sintetizado.

Felizmente, a comunidade do Vivado HLS C-level é rica em exemplos e projetos, fato que possibilita o desenvolvimento dos mais diversificados projetos, envolvendo diferentes tipos de estruturas e interfaces de dados. Também existem tutoriais bem estruturados para a correta utilização da linguagem C/C++, bem como de diretrizes que se façam convenientes para as sintetizações praticadas. Este trabalho não se aprofundará neste sentido, mantendo-se restrito resolução do problema considerado.

4.3.2 O organismo artificial Camphs1D05G1RAV

O sistema Camphs1D05G1 foi escolhido, devido à sua simplicidade, para a primeira implementação em placa. Para tal, foi necessária a elaboração de uma nova arquitetura, que considera o sistema mencionado integrado com demais componentes capazes de fornecer os sinais de controle, para o organismo operar, e validar dos resultados obtidos. Deste modo, foi construído o organismo artificial Camphs1D05G1RAV, composto pelos módulos que aparecem na Figura 23, a seguir.

A construção desta arquitetura começa pela consideração do core Camphs1D05G1, já explorado nas simulações comportamentais realizadas. Em seguida, precisamos de uma memória alocada em placa, para fornecer os sinais de controle necessários para a operação

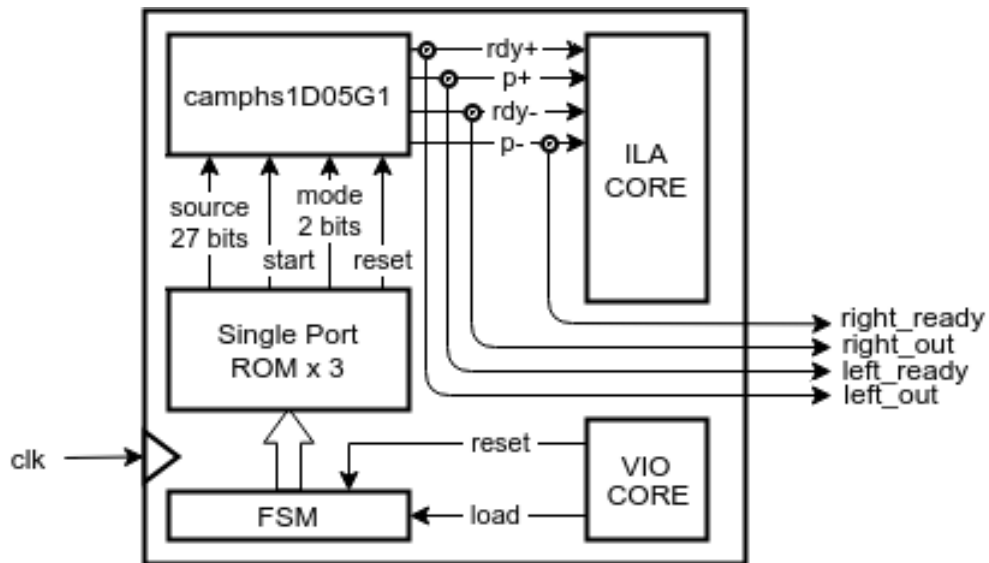


Figura 23 – Esboço da arquitetura Camphs1D05G1RAV. Além do core CA, temos memória ROM embarcada, um IP-core ILA, e um IP-core VIO para introdução de sinais. Temos também uma pequena máquina de estados controlando a unidade de memória ROM.

de Camphs1D05G1. Tal recurso pode ser implementado através da memória ROM (BMG), que se trata de um recurso para a implementação de memórias do tipos ROM e RAM. Neste parte do projeto, foram instanciadas três memórias ROM do tipo Single-port, sendo uma para cada sinal de controle.

Cada memória ROM necessita, por sua vez, de uma máquina de estados para sua operação, de acordo com a latência do organismo Camphs1D05G1 - unidade controladora. Tal máquina, é controlada pelo Virtual Input/Output IP-core, através dos sinais de RESET e LOAD. Para finalizar, temos o ILA-core conectado às saídas de Camphs1D05G1, visualizando os sinais oriundos de uma célula receptora, bem como os sinais de READY gerados em suas portas de saída, direita e esquerda.

4.3.2.1 Funcionamento da memória embarcada em placa e vetores de teste

As memórias embarcadas em placa (Single-port ROM) devem ser criadas, ou instanciadas, a partir de três arquivos de texto, relacionados a cada um dos sinais de controle MODE, START e RESET. O conteúdo dos arquivos está apresentado na tabela 5, a seguir.

Tabela 5 – Sinais de controle embarcados na memória ROM, necessários para o funcionamento de Camphs1D05G1RAV.

MODE	START	VALUE
10	1	010000110100100000000000000000
10	0	010000110100100000000000000000
10	1	010000110100100000000000000000
01	0	000000000000000000000000000000

Já foi dito que cada memória ROM deve operar de acordo com a latência do sistema camphs1D05G1. Por motivos de simplificação, as três unidades de memória foram encapsulados em uma arquitetura chamada carom1D05G1, diagramada na Figura 24.

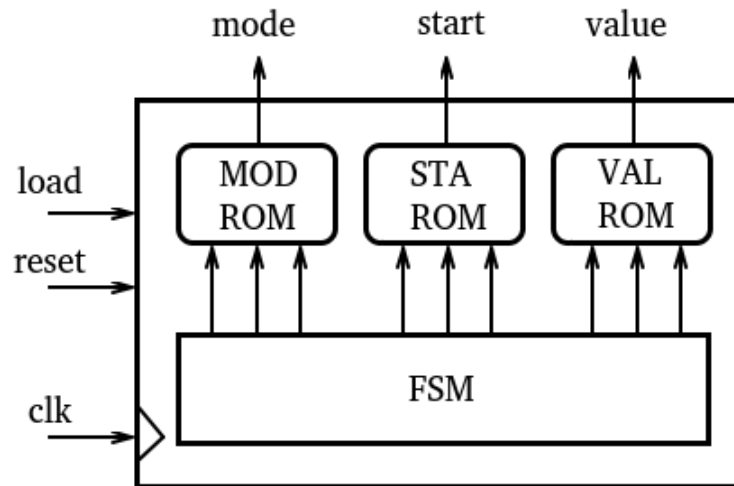


Figura 24 – Esboço da arquitetura Carom1D05G1. Este componente deve armazenar três sinais de controle, tendo um sinal 2 bits (MOD), um com 27 bits (VAL), e um último sinal com apenas 1 bit (STA).

O funcionamento do sistema ocorre através de dois sinais, RESET e LOAD, fornecidos pelo VIO IP-core, diretamente ao módulo Carom1D05G1]. Na subida do relógio, com o sinal de RESET em nível alto, Camphs1D05G1 retorna os endereços de memória para suas posições iniciais. Na subida do relógio, com o sinal de LOAD em nível alto, Camphs1D05G1 dispara as três memórias em paralelo, fornecendo três valores de endereço a serem lidos. Cada memória ROM possui a latência de dois ciclos de relógio, portanto, a latência de Carom1D05G1 é de três ciclos de relógio.

Conforme se vê na tabela 5, dois sinais de fonte foram introduzidos no sistema, em formato de pulsos, e com um intervalo de descanso de 1 ciclo operacional (que na prática se equivale a três ciclos de relógio). Desta maneira, se espera obter, para Camphs1D05G1RAV, um *throughput* de exatos 3 ciclos de relógio. Vale ressaltar que os sinais de saída indicados na Figura 23 foram mantidos no circuito, porém sem necessidade, devido à utilização do ILA-core.

4.3.2.2 Análise de tempo realizada para Camphs1D05G1RAV

Na primeira tentativa de implementação do organismo artificial Camphs1D05G1RAV, sem restrições de tempo, foi obtido uma folga negativa (Setup) de aproximadamente 22 ns no total, com folgas críticas variando entre 1 e 2 ns. O caminho mais crítico indicava um problema entre o componente Carom1D05G1 e a célula autômata de número 3, definida

como fonte. O problema foi resolvido a partir da inclusão de um arquivo com restrições de tempo, para as seguintes definições (time-constraint.xdc):

```

1: # Clock signal
2: create_clock -period 10.00 -name clk -waveform {0.000 5.000} [get_ports clk]
3:
4: # input delays
5: set_input_delay -clock clk -max 2.0 [get_ports reset]
6: set_input_delay -clock clk -min 1.0 [get_ports reset]
7:
8: set_input_delay -clock clk -max 2.0 [get_ports start]
9: set_input_delay -clock clk -min 1.0 [get_ports start]

```

As informações apresentadas podem ser visualizadas na tabela 6. Os resultados desta implementação, no que se diz respeito aos novos caminhos críticos obtidos, estão apresentados no capítulo de resultados.

Tabela 6 – Parâmetros adotados na implementação com restrições de tempo.

Parâmetro	Valor	DUTY	MAX	MIN
Relógio : clk	100 MHz	50%	NA	NA
Input Delay p/ RESET (clk)	NA	NA	2.0	1.0
Input Delay p/ START (clk)	NA	NA	2.0	1.0

4.3.3 O organismo artificial Camphs1D05G1AL

O organismo artificial Camphs1D05G1AL possui um CORE de 5 células autômatas, o mesmo utilizado no primeiro organismo artificial apresentado neste trabalho. Temos também um barramento de dados do tipo *AXI4-Lite*, para transferência de dados entre a memória DDR3 embarcada e o CORE CA. Um esboço deste organismo está apresentado na Figura 25, seguir.

Neste arquitetura, o CORE CA (PL) foi programado juntamente com uma interface *AXI4-Lite*, de maneira a permitir a comunicação com o PS (*ARM Processor System*). O sistema como um todo possui, além do IP-core Camphs1D05G1AL, alguns módulos necessários para funcionamento do ARM, que são o *Axi Interconnect* e o *Processor System*, além das portas de acesso à memória DDR3.

O *Axi Interconnect* promove a conexão entre 1 ou mais dispositivos mestres de memória mapeada (ARM) com 1 ou mais dispositivos escravos de memória mapeável (Camphs1D05G1AL). O acesso de memória via mapeamento de endereços é uma das características do barramento de dados *AXI4-Lite*, e por este motivo a sua utilização para grandes volumes de dados não é indicada - nestes casos, o *AXI4-Stream* deve ser

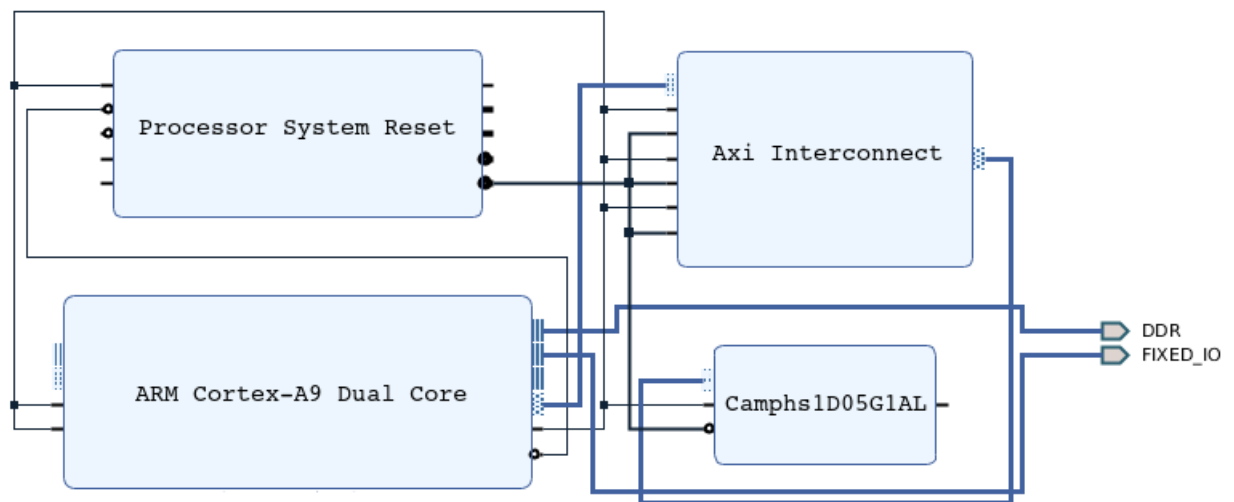


Figura 25 – Esboço da arquitetura de Camphs1D05G1AL. Este design foi automaticamente constituído pela ferramenta Vivado 2018.3 Webpack.

preferido. Para finalizar, o *Processor System Reset* permite que o PS realize a manipulação dos sinais RESET, em todos os componentes do sistema.

O organismo artificial Camphs1D05G1AL é indicado para situações nas quais o sistema CA seja estimulado sem a presença de um receptor, entre as suas células. Deste modo, o sistema efetuará a evolução celular em hardware, durante o número de ciclos requeridos pelo PS. Trata-se de um solução de elevada eficiência e velocidade, uma vez que o IP-core (escravo) não é interrompido durante seu ciclo interno de funcionamento.

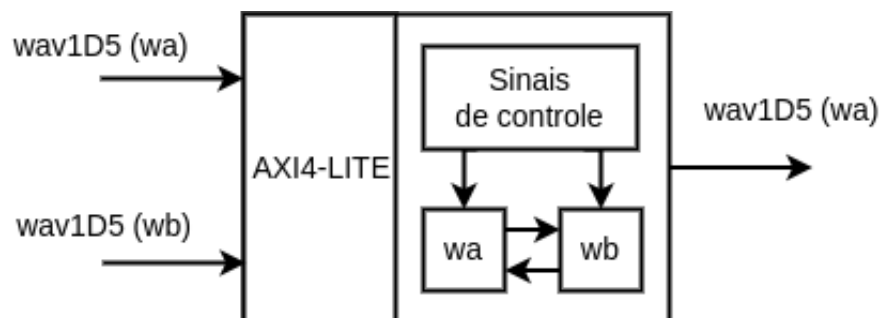


Figura 26 – Esboço do CORE de Camphs1D05G1AL. A interface *AXI4-Lite* promove a introdução e externalização dos sinais no core CA, que opera ciclicamente até o estado final desejado.

Como se pode ver na Figura 26, O CORE de Camphs1D05G1AL temos dois parâmetros de entrada e um parâmetro de saída, do tipo *wav1D5*. Trata-se de uma estrutura de dados, definida pelas linhas de código seguintes (*camphs1D05G1AL.h*).

```
1: #ifndef __WAVEGUIDE_1D5__
2: #define __WAVEGUIDE_1D5__
3:
4: struct wav1D5 {
5:     float lg0;
6:     float lg1;
7:     float lg2;
8:     float lg3;
9:     float lg4;
10:    float rg0;
11:    float rg1;
12:    float rg2;
13:    float rg3;
14:    float rg4;
15: }
16:
17: #endif
```

A estrutura `wav1D5` representa 2 guias de ondas (esquerda e direita) distribuídas em 5 células autônomas. Internamente, `camphs1D05G1AL` possui todos os parâmetros necessários para a sua execução, programada para um determinado número de ciclos operacionais. As estruturas `wa` e `wb` se interagem de maneira a evoluir os seus valores, até que `wa` atinja o seu estado final. A interface *AXI4-Lite* é utilizada 2 únicas vezes, sendo a primeira para leitura dos valores iniciais e a segunda para retorno dos valores finais.

4.3.4 O organismo artificial `Camphs1D05G1AS`

O organismo artificial `Camphs1D05G1` também possui um CORE de 5 células autômatas, o mesmo utilizado no primeiro organismo artificial apresentado neste trabalho. Temos também os barramentos de dados dos tipos *AXI4-Lite* e *AXI4-Stream*, operando conjuntamente, para transferência de dados entre a memória DDR3 embarcada e o CORE CA. Um esboço deste organismo está apresentado na Figura 27, seguir.

Os módulos *Axi Interconnect* e *Processor System Reset* operam de maneira semelhante ao primeiro caso apresentado. Temos ainda o módulo *Axi Smart Connect* como uma interface importante para a comunicação de dados, entre o escravo, IP-core `Camphs1D05G1AS`, e o mestre, PS, com funções semelhantes ao *Axi Interconnect*.

Este sistema é mais complexo do que o primeiro apresentado, devido à presença de uma interface *AXI4-Stream* que conecta as saídas de uma célula receptora diretamente à memória. Desta maneira, o IP-core `Camphs1D05G1AS` externaliza seus resultados paralelamente às suas operações internas, a partir de uma única chamada da interface *AXI4-Lite*

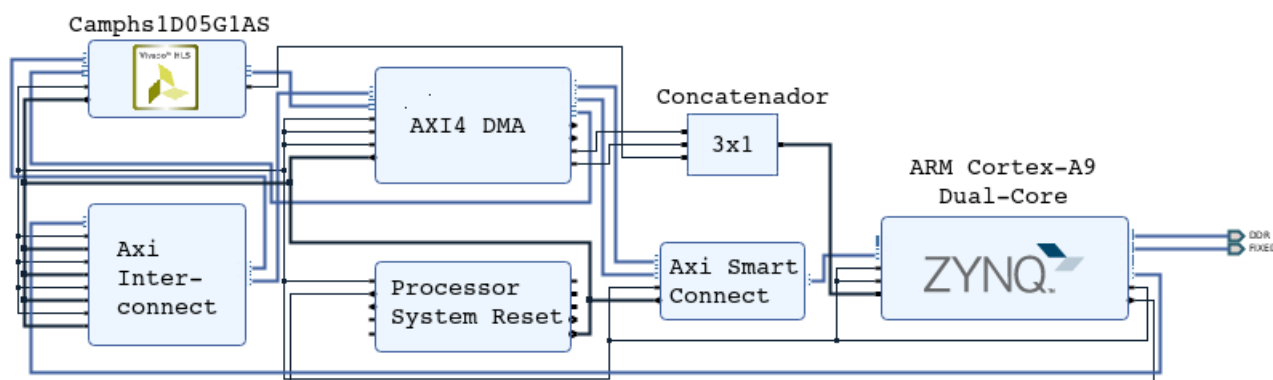


Figura 27 – Esboço da arquitetura de Camphs1D05G1AS. Este design foi automaticamente constituído pela ferramenta Vivado 2018.3 Webpack.

- passagem das estruturas iniciais. Um esboço do CORE utilizado está apresentado na Figura 28, a seguir.

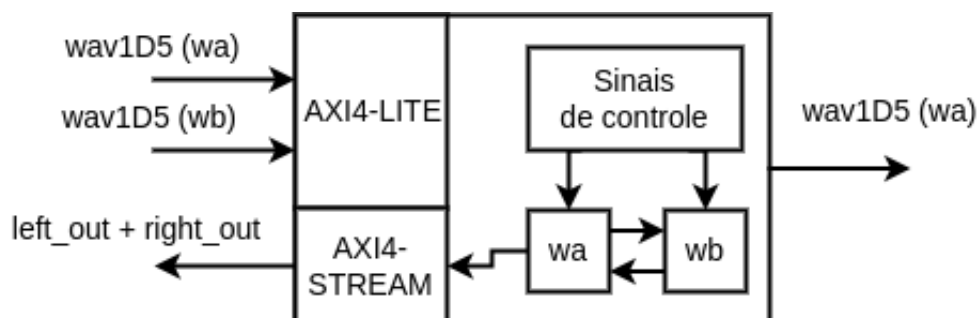


Figura 28 – Esboço do CORE de Camphs1D05G1AS. A interface *AXI4-Lite* promove a introdução e externalização do estado final do core CA, enquanto a interface *AXI4-Stream* externaliza os sinais receptores.

Observe que Camphs1D05G1AS retorna o valor combinado das saídas direita e esquerda da célula receptora. Estes valores podem ser graficamente visualizados e validados, posteriormente.

O organismo artificial Camphs1D05G1AS é indicado para situações nas quais se tenha a necessidade de captar os sinais de uma célula autômata receptora, paralelamente ao funcionamento do organismo artificial. Para a interface *AXI4-Lite* realizar tal tarefa seria necessário realizar um *loop* externo, em nível de software, com chamadas sucessivas do IP-core Camphs1D05G1AL. Esta estratégia não seria otimizada e, portanto, deve ser evitada.

4.3.4.1 Validação das implementações realizadas em placa

O modelo de referência adotado para validação dos resultados, para ambos os organismos, pode ser obtido analiticamente, devido à sua simplicidade. Os testes foram

realizados a partir de um sinal de estímulo (célula geradora) unitário, disparado uma única vez, capaz de fazer o organismo artificial funcionar por um período pré-definido de tempo.

Ambos os organismos artificiais apresentados devem retornar, ao final do seu ciclo de operação, os valores correspondentes às suas saídas direita e esquerda. Estes valores podem ser diretamente comparados com os valores obtidos pelo do modelo de referência. O organismo artificial Camphs1D05G1AS também retorna tais valores, porém com o acréscimo de um vetor contendo os valores resultantes obtidos pela célula autômata receptora. Os valores esperados pelo modelo de referência se encontram a seguir.

Tabela 7 – Valores de referência para Camphs1D05G1AL. Excitação e resultados após 11 interações (contando a excitação).

Vetores	Células Autômatas				
	#1	#2	#3	#4	#5
Excitação	0.0000	0.0000	1.0000	0.0000	0.0000
<i>Left-going wave</i>	0.1250	0.0000	1.0000	0.0000	0.1250
<i>Right-going wave</i>	0.1250	0.0000	1.0000	0.0000	0.1250

Tabela 8 – Valores de referência para Camphs1D05G1AS. Excitação e resultados após 10 interações (contando a excitação).

Vetores	Células Autômatas				
	#1	#2	#3	#4	#5
Excitação	0.0000	0.0000	1.0000	0.0000	0.0000
<i>Left-going wave</i>	0.0000	0.0000	1.0000	0.2500	0.0000
<i>Right-going wave</i>	0.0000	0.2500	1.0000	0.0000	0.0000

Tempo	Valores lidos pelo receptor em #2
1	0.0000
2	1.0000
3	0.0000
4	0.5000
5	0.0000
6	0.5000
7	0.0000
8	0.2500
9	0.0000
10	0.2500

5 Resultados

Este capítulo se inicia com os resultados das simulações comportamentais realizadas sobre os organismos artificiais Camphs1D05G1 e Camphs1DB9G2. Outros detalhes do projeto das arquiteturas mencionadas, tais como geradores de código e conversores numéricos, também serão apresentados. Em seguida, serão apresentados os relatório de consumo, bem como os resultados obtidos a partir das implementações realizadas em placa.

5.1 Simulações comportamentais

5.1.1 Organismo artificial Camphs1D05G1

Os resultados desta primeira simulação foram úteis para a validação de todas as funcionalidades do sistema CA desenvolvido. As figuras seguintes trazem imagens obtidas da simulação comportamental realizada no software VIVADO, para os sinais de controle apresentados no capítulo anterior.

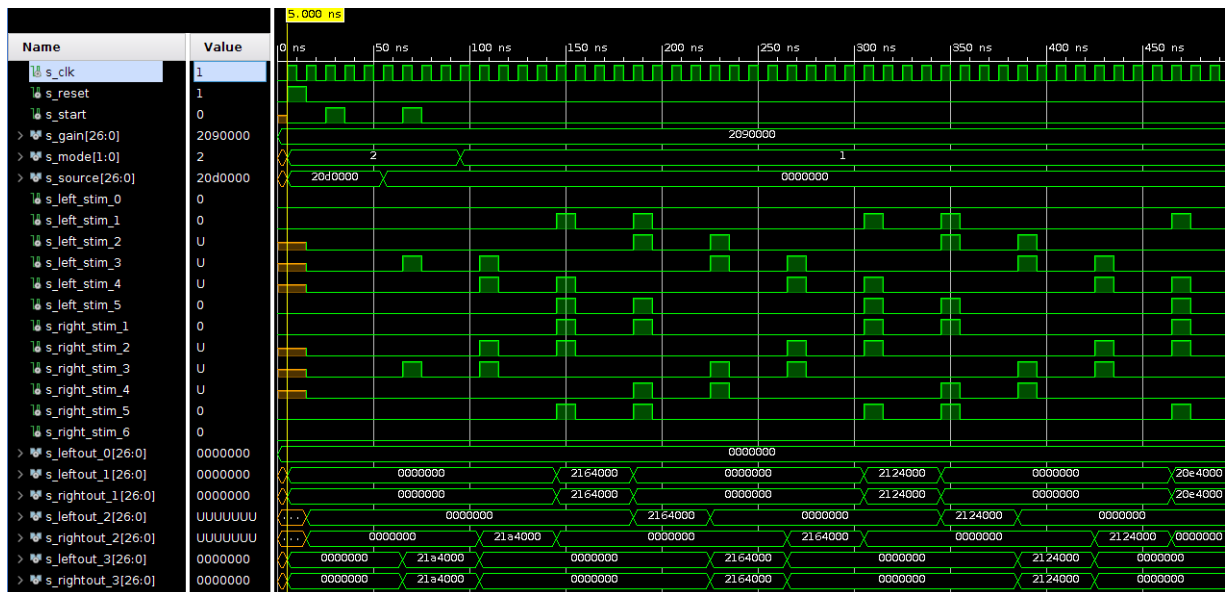


Figura 29 – Simulação comportamental da organismo artificial Camphs1D05G1. Em tela, um visão geral das variáveis envolvidas.

Um sinal impulsivo foi gerado pela célula GCA1D central. Após mudança de valor do sinal MODE, conectado à esta célula, esta célula autômata passou a operar apenas com a dispersão das ondas transmitidas.

É possível observar na Figura 29 um padrão de resposta que se repete com o tempo, na Figura dos sinais $s_left_stim_*$ e $s_right_stim_*$. Estes sinais representam os estímulos de saída de ondas nas células autômatatas. Este comportamento é desejado devido às sucessivas reflexões de ondas geradas pelas células HCA1D posicionadas nos extremos do sistema.

Na Figura 30, a seguir, serão apresentados os primeiros eventos esperados para este sistema CA, em acordo com o modelo de referência (no que tange o fenômeno observado).

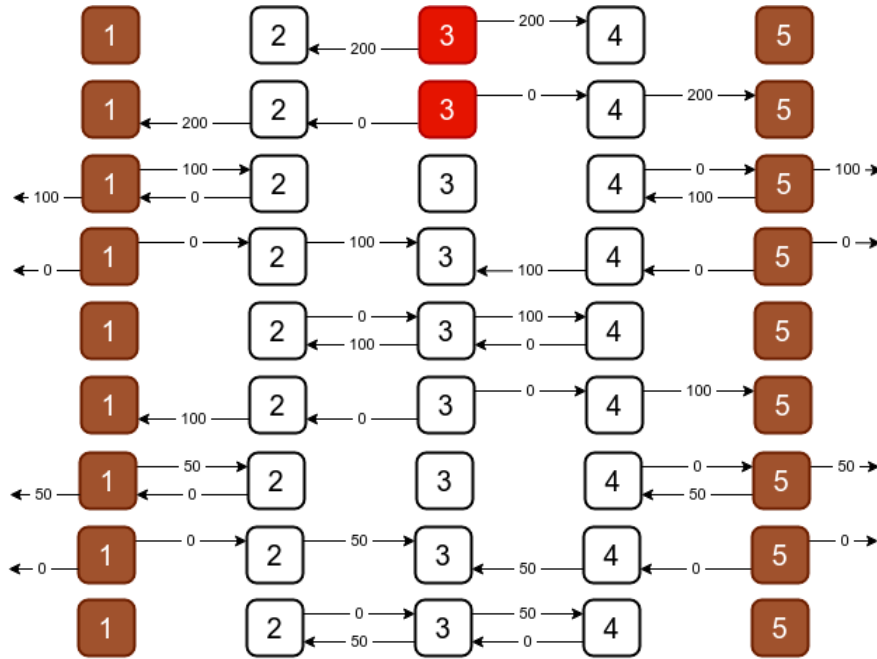


Figura 30 – Resultados esperados para o sistema Camphs1D05G1, com base nos fenômenos básicos da onda. Cada linha apresentada representa as saídas da equação de d’Alembert, submetida ao sinal de entrada (célula 3), em cada um dos 5 nós da solução discreta. Temos também representadas as duas direções de movimento das ondas acústicas.

O coeficientes internos às células HCA1D foram ajustados para emularem um coeficiente de reflexão $r = 0.5$. O padrão repetido visto na Figura 29 fica agora mais fácil de ser compreendido, pois percebemos que as ondas sonoras devem, gradualmente, perder 50% das suas amplitudes a cada reflexão.

Na Figura 31 podemos comprovar a sequência de geração de estímulos esperada - é possível verificá-la na Figura 30. Primeiramente a célula 3 (fonte) gera estímulos nas duas direções, com amplitude igual a 200. Em seguida, a célula 4 gera um estímulo para a sua célula a direita ($s_left_stim_4$), juntamente com as células 2, que gera um estímulo para a sua vizinha à esquerda ($s_right_stim_2$). Neste mesmo ciclo, a célula 3 gera novamente seus estímulos bidirecionais, desta vez para o valor 0. Deste modo, a simulação prossegue, atendendo perfeitamente às expectativas para com a solução.

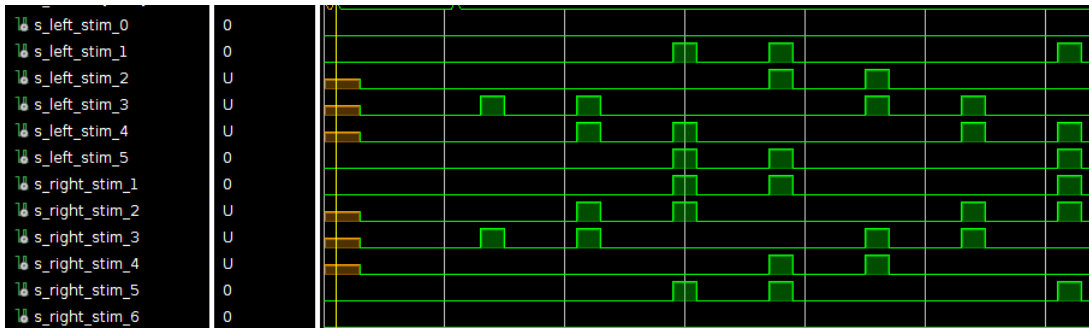


Figura 31 – Resultado esperado para o sistema Camphs1D05G1, com base nos fenômenos básicos da onda. Em tela, temos os sinais de estímulo gerados.

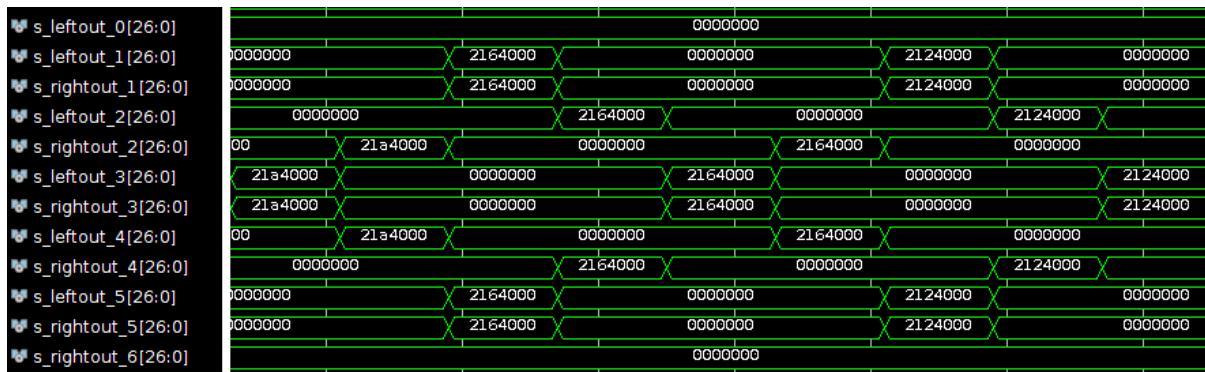


Figura 32 – Resultado esperado para o sistema Camphs1D05G1, com base nos fenômenos básicos da onda. Em tela, as amplitudes de onda transmitidas pelas células autômatas.

Na Figura 32 observamos as amplitudes das ondas transmitidas pelas células autômatas, em acordo com os estímulos gerados pelas suas vizinhas. Os valores em hexadecimal $21a4000_h$, 2164000_h e 2124000_h representam os decimais 200_d , 100_d e 50_d , respectivamente.

Para finalizar esta exposição, é possível checar as saídas dos canais receptores posicionados na célula 2, e sequência observada coincide perfeitamente com o padrão esperado. Os resultados podem ser apreciados na Tabela 9, a seguir.

Cada ciclo do sistema CA, indicado na Tabela 9, corresponde a um intervalo de 4 ciclos de relógio. Os ciclos de 1 a 9 correspondem exatamente com os resultados apresentados na Figura 30, e foram contabilizados a partir do primeiro sinal de START.

5.1.2 Dados de entrada, saída e decodificações realizadas em Camphs1D05G1

Serão apresentados aqui apenas os conteúdos dos arquivos utilizados para a introdução dos sinais de controle. O procedimento de saída de valores, nos canais receptores, também será discutido.

O organismo artificial Camphs1D05G1 foi projetado para executar a leitura de 3

Tabela 9 – Valores registrados nos canais de saída de Camphs1D05G1.

S_LEFT_OUT		
Ciclo	Valor Obtido	Valor esperado
1	0000000 _h	0000000 _h
2	0000000 _h	0000000 _h
3	0000000 _h	0000000 _h
4	2164000 _h	2164000 _h
5	0000000 _h	0000000 _h
6	0000000 _h	0000000 _h
7	0000000 _h	0000000 _h
8	2124000 _h	2124000 _h
9	0000000 _h	0000000 _h
10	0000000 _h	0000000 _h
11	0000000 _h	0000000 _h
S_RIGHT_OUT		
Ciclo	Valor Obtido	Valor esperado
1	0000000 _h	0000000 _h
2	21a4000 _h	21a4000 _h
3	0000000 _h	0000000 _h
4	0000000 _h	0000000 _h
5	0000000 _h	0000000 _h
6	2164000 _h	2164000 _h
7	0000000 _h	0000000 _h
8	0000000 _h	0000000 _h
9	0000000 _h	0000000 _h
10	2124000 _h	2124000 _h
11	0000000 _h	0000000 _h

arquivos de dados: `sce_value.txt`, `sce_start.txt` e `sce_modes.txt`. Os arquivos são lidos nas bordas de subida do relógio - o sinal de RESET foi acionado na primeira borda de subida do relógio para inicializar todos os registradores internos. O conteúdo dos arquivos estão apresentados na Tabela 10.

A cada borda de subida do relógio, o sistema também registra os canais de recepção, em vias separadas, para a esquerda e direita. Por este motivo, os arquivos devem ser decodificados e sincronizados para com a correta frequência de resposta do sistema - a latência do sistema é de 4 ciclos de relógio. O mecanismo de decodificação dos valores de saída está apresentado no fragmento de código a seguir.

Na Figura 33 observamos o bloco de decodificação dos valores registrados nos canais de saída do Camphs1D05G1. Primeiramente, os valores são convertidos para PF 64 bits (linhas 22 e 23) para serem armazenados em vetores (linhas 24 e 25). Camphs1D05G1 também externaliza os sinais de estímulo gerados pela célula receptora, para que estes valores sejam utilizados na amostragem dos registros de saída (conversões nas linhas 26 e 27, com armazenagem nas linhas 28 e 29).

Tabela 10 – Conteúdo dos arquivos de controle utilizados na simulação comportamental do organismo artificial Camphs1D05G1.

Linhas dos Arquivos de Controle		
<i>sce_modes.txt</i>	<i>sce_start.txt</i>	<i>sce_value.txt</i>
10 _b	0 _b	0100000100100000000000000000 _b
10 _b	0 _b	0100000100100000000000000000 _b
10 _b	1 _b	0100000100100000000000000000 _b
10 _b	0 _b	0100000100100000000000000000 _b
10 _b	0 _b	0100000100100000000000000000 _b
10 _b	0 _b	0000000000000000000000000000 _b
10 _b	1 _b	
10 _b	0 _h	
10 _b		
01 _b		

```

21 for output_i=1:N
22   left_f= bin2float(EW,FW,cell2mat(left_rcedata(output_i)));
23   right_f= bin2float(EW,FW,cell2mat(right_rcedata(output_i)));
24   left_vector(output_i)= left_f;
25   right_vector(output_i)= right_f;
26   left_rdy= str2num(cell2mat(left_rceready(output_i)));
27   right_rdy= str2num(cell2mat(right_rceready(output_i)));
28   left_ready(output_i)= left_rdy;
29   right_ready(output_i)= right_rdy;
30 endfor

```

Figura 33 – Decodificação dos valores registrados nos receptores de Camphs1D05G1. Fragmento do código desenvolvido no software Octave.

```

31 % amostragem
32 left_ready_idx= find(left_ready==1);
33 right_ready_idx= find(right_ready==1);
34 left_vector= left_vector(left_ready_idx);
35 right_vector= right_vector(right_ready_idx);

```

Figura 34 – Amostragem dos valores registrados nos receptores de Camphs1D05G1. Fragmento do código desenvolvido no software Octave.

Na Figura 34 observamos o processo de amostragem dos valores registrados nos canais receptores de Camphs1D05G1. Nas linhas 32 e 33, os vetores de dados são amostrados pelos sinais de estímulo gerados pela célula receptora. Deste modo, os vetores foram sincronizados com a frequência de saída de dados.

Na Figura 35 observamos as 12 primeiras saídas da célula autômata receptora de Camphs1D05G1. As curvas representam os canais de saída esquerdo e direito. Vale lembrar que os valores apresentados se referem apenas aos momentos nos quais a célula

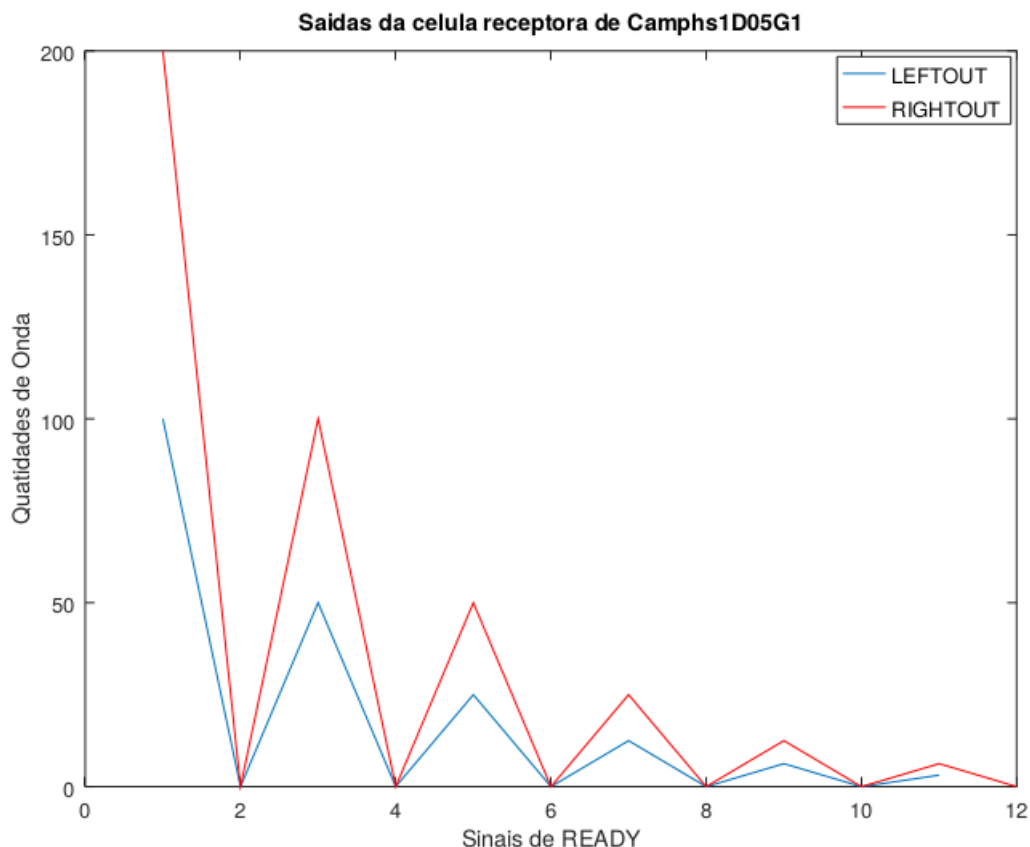


Figura 35 – Resultado da decodificação dos valores registrados nos canais receptores de Camphs1D05G1. Temos em tela os 12 primeiros instantes de tempo do sistema CA.

produziu saídas, ou seja, os valores estão sincronizados com os estímulos gerados pela célula. Verifica-se que os resultados atendem às expectativas teóricas apresentadas na Figura 30.

5.1.3 Organismo artificial Camphs1DB9G2

Todos os detalhes relacionados à operação dos sinais, bem como respostas do sistema CA Camphs1D já foram explorados na primeira simulação comportamental realizada. Sendo assim, considerando o maior volume de resultados, a validação desta simulação foi realizada através do uso de 2 receptores, “A ”e “B ”, respectivamente posicionados nas células de números 20 e 150 do sistema CA.

Os mesmos receptores foram posicionados no modelo de referência, que desta vez está representado pelo sistema CA equivalente validado no capítulo 2. Em cada receptor, os valores de saída, provenientes de cada canal, foram somados de maneira a se obter a quantidade de onda resultante através da célula.

É possível observar nas Figuras 37 e 38, os sinais resultantes obtidos pelos receptores, oriundos das soluções em VHDL e C. inicialmente, podemos verificar que o receptor

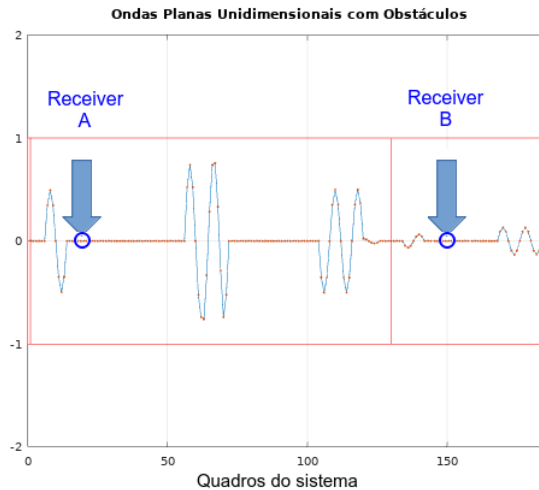


Figura 36 – Posicionamento dos receptores “A ”e “B ”em Camphs1DB9G2, nas saídas das células 20 e 150.

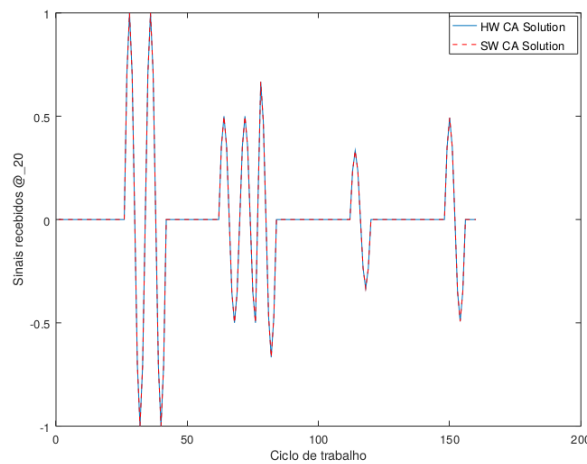


Figura 37 – Resultado da decodificação dos valores registrados nos canais do receptor “A ”em Camphs1DB9G2 e modelo de referência.

“A ”percebe a passagem de exatamente 5 ondas, enquanto o receptor “B ”percebe 3 ondas. Estas contagens podem ser verificadas no vídeo realizado para o modelo de referência, cujo link se encontra no capítulo 2. Os resultados, para os 160 ciclos iniciais de trabalho, coincidem satisfatoriamente com o modelo de referência. O MSE obtido para os 2 sinais se encontram na Tabela 11.

Tabela 11 – Erro médio quadrático calculado para os sinais receptores de Camphs1DB9G2, em relação a saídas semelhantes coletadas no modelo de referência.

Receptor	MSE
A	3.45^{-13}
B	1.53^{-14}

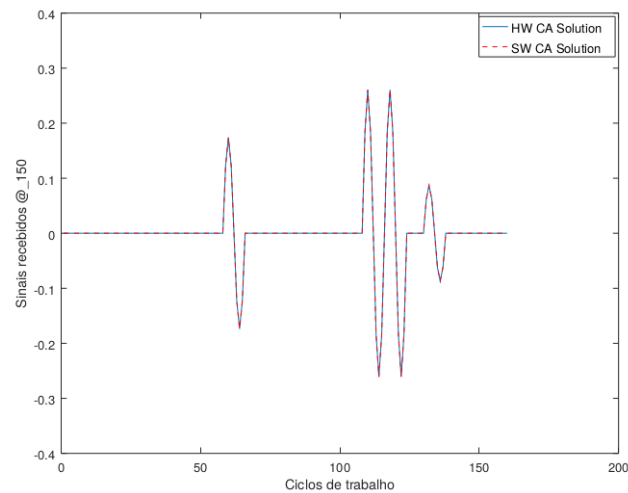


Figura 38 – Resultado da decodificação dos valores registrados nos canais do receptor “B” em Camphs1DB9G2 e modelo de referência.

5.1.4 Dados de entrada, saída e decodificações realizadas em Camphs1DB9G2

Por abordar duas fontes sonoras, a quantidade de arquivos de leitura é o dobro da quantidade utilizada em Camphs1D05G1. De fato, sempre teremos 3 arquivos de leitura para cada fonte sonora considerada no sistema CA Camphs1D. Os arquivos foram construídos manualmente, com base no diagrama de tempo aplicado, mas podem, alternativamente, ser programados a partir de um conjunto de parâmetros de simulação.

Desta vez, a amostragem do arquivo de saída de Camphs1DB9G2 foi realizada através de um sinal gerado em software, sincronizado com o diagrama de tempo utilizado na simulação. Deste modo, 160 saídas foram coletadas, correspondentes às bordas de subida do relógio [3, 7, 11, ..., 639]. Os vetores, referentes a cada canal de recepção (direito e esquerdo), foram simplesmente somados para produzir os valores resultantes de onda.

A decodificação dos sinais binários coletados, PF em 27 bits para PF em 64 bits, foi realizada de maneira semelhante à primeira simulação, através do fragmento apresentado na Figura 33.

5.1.5 Relatórios de utilização de recursos após sínteses lógicas

Em primeiro lugar vale dizer que todos os códigos em VHDL foram preparados no VIVADO, visando um kit alvo de desenvolvimento Diligent ARTY Z7 ([DILIGENT, 2012](#)). Todas as sínteses foram realizadas com sucesso.

Como se pode notar na Figura 39, para a arquitetura Camphs1D05G1, temos a estimativa de utilização de 903 LUTs, 553 FFs e 9 DSP. O recurso IO (IOB) não importa neste momento pois o mesmo representa o conjunto de elementos utilizados para entrada e saída de dados nas simulações, e a estratégia de IO será significativamente modificada

para as implementações físicas.

Resource	Utilization	Available	Utilization %
LUT	903	53200	1.70
FF	553	106400	0.52
DSP	9	220	4.09
IO	115	125	92.00

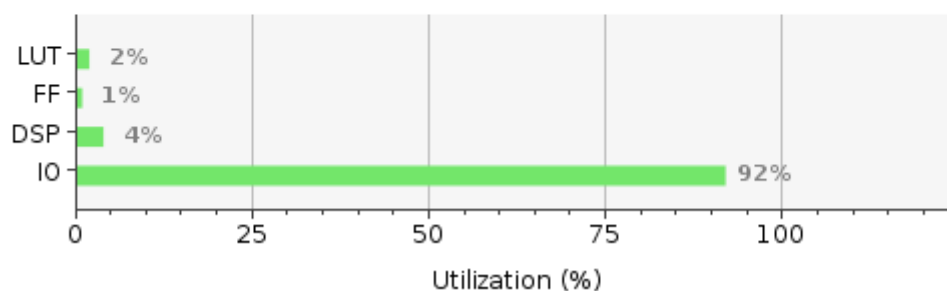


Figura 39 – Consumo estimado de recursos para Camphs1D05G1.

Resource	Utilization	Available	Utilization %
LUT	23531	53200	44.23
FF	21928	106400	20.61
IO	228	125	182.40

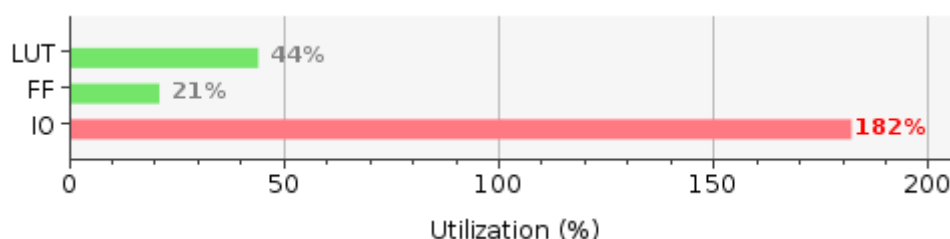


Figura 40 – Consumo estimado de recursos para Camphs1DB9G2.

Na Figura 40 temos o resumo dos recursos de utilização de Camphs1DB9G2, sendo 23531 LUTs e 21928 FFs. A ferramenta não indicou o uso de DSPs, provavelmente pelo fato da estimativa ter superado a quantidade de componentes disponíveis. Neste estágio, o consumo de recursos nos serve apenas como direcionamento para as implementações a serem realizadas, pois não é possível estimar com precisão tais recursos, uma vez que a ferramenta não implementou o sistema em placa. No tópico seguinte, temos relatórios mais completos, e confiáveis, de consumo de recursos, referentes às implementações praticadas em placa, nas etapas finais deste trabalho.

5.2 Implementações em Placa

5.2.1 Organismo artificial Camphs1D05G1RAV

As análises de *timing* foram realizadas de tal modo a identificar os caminhos críticos de *setup* e *hold*, indicando as folgas positivas obtidas, para um relógio empregado de 100MHz. Na figura 41 temos a visualização do caminho crítico de *setup*, em placa, além dos valores de tempo querido e de chegada. Neste caso, o tempo requerido (*Arrival Time*) foi maior do que o tempo de chegada (*Required Time*), apresentando uma folga de 1.162ns, como desejado. Este caminho crítico está compreendido entre o registrador associado com o sinal de fonte (origem) necessário para a operação da célula autômata #3 (destino).

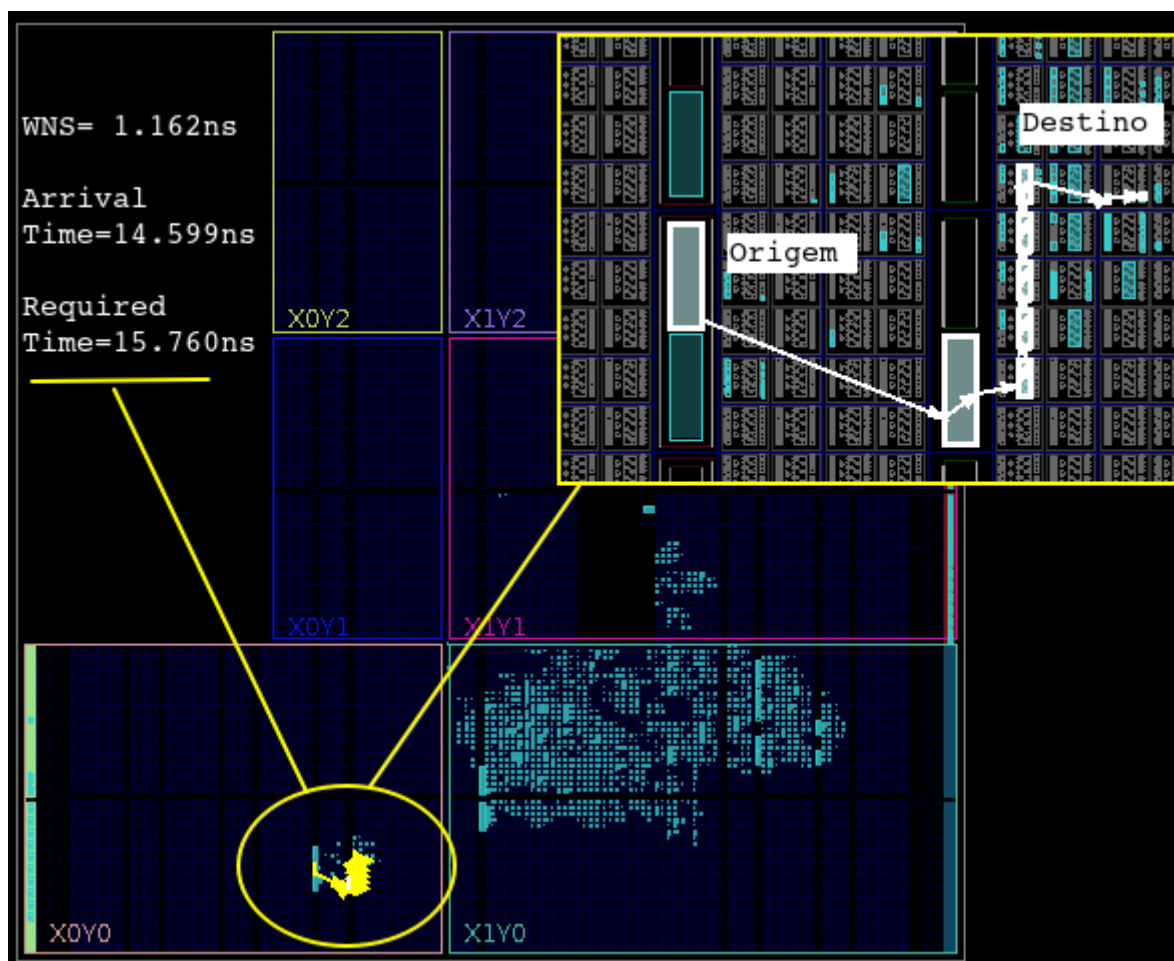


Figura 41 – Análise de tempo obtida para Camphs1D05G1RAV, para o caminho crítico de *setup*. Observamos na imagem os valores dos parâmetros *Arrival Time* e *Required Time*, bem como a folga positiva encontrada. Também está indicado o caminho percorrido na placa.

Na figura 42 temos a visualização do caminho crítico de *setup*, em placa, além dos valores de tempo querido e de chegada. Neste caso, o tempo requerido (*Arrival Time*) foi menor do que o tempo de chegada (*Required Time*), apresentando uma folga de 0.060ns,

como desejado. Desta vez, o caminho crítico está compreendido na implementação do IP-core ILA (origem e destino). O relógio empregado se mantém 1m 100MHz.

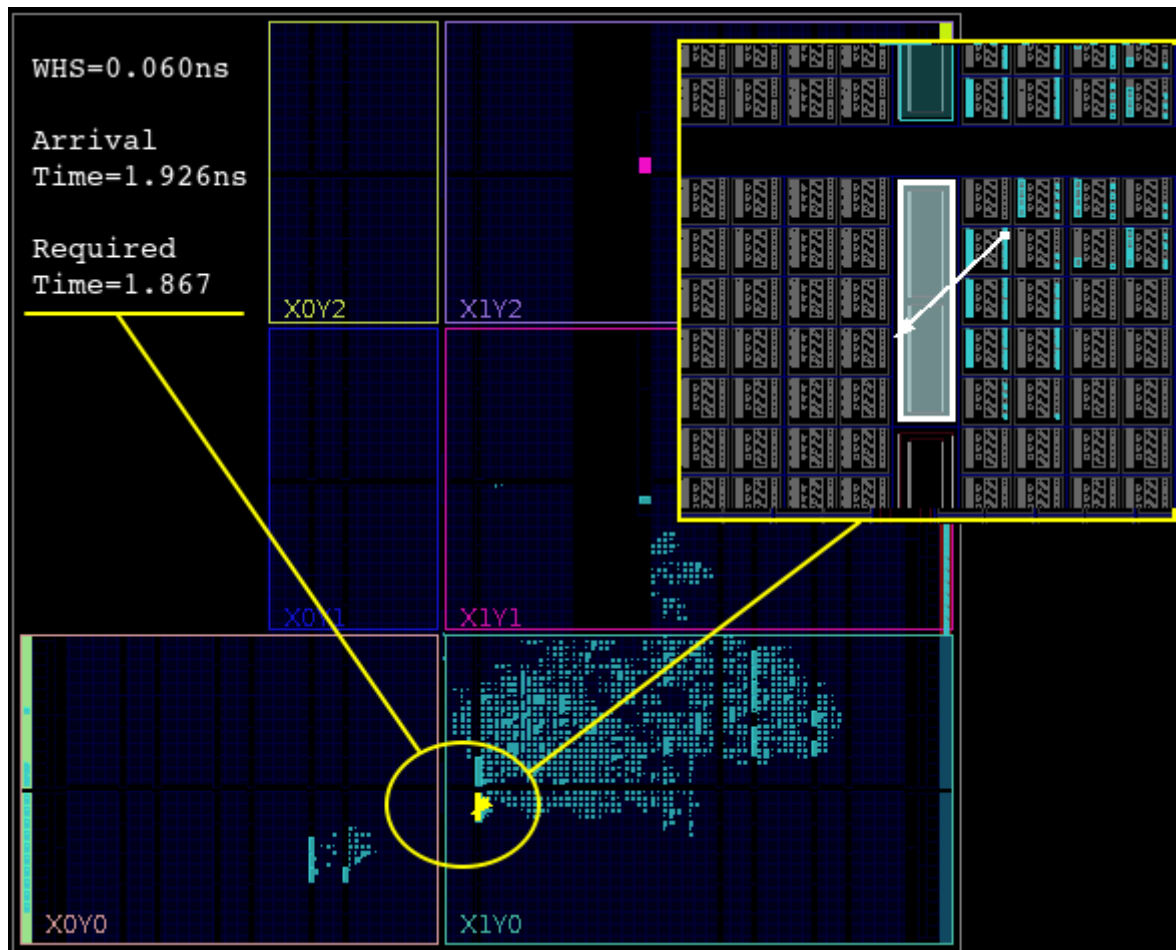


Figura 42 – Análise de tempo obtida para Camphs1D05G1RAV, para o caminho crítico de *hold*. Observamos na imagem os valores dos parâmetros *Arrival Time* e *Required Time*, bem como a folga positiva encontrada. Também está indicado o caminho percorrido na placa.

Devido ao fato do organismo em foco possuir o mesmo CORE de Camphs1D05G1, o seu modelo de referência é o mesmo apresentado na Figura 30. Para efeito de comparação, foi então realizada uma simulação comportamental equivalente, e os resultados se encontram na Figura 43.

Na Figura 43 é possível verificar que o *throughput* do sistema é de 3 ciclos de relógio. Este fato é devido à latência da memória ROM. Os resultados obtidos pelas células receptoras de Camphs1D05G1RAV, nas mesmas posições arbitradas em Camphs1D05G1, coincidem com aqueles apresentados na Figura 35. Os mesmos podem ser visualizados na Figura 44. A seguir, na Tabela 12, temos relatório de consumo de recursos para o organismo em questão.

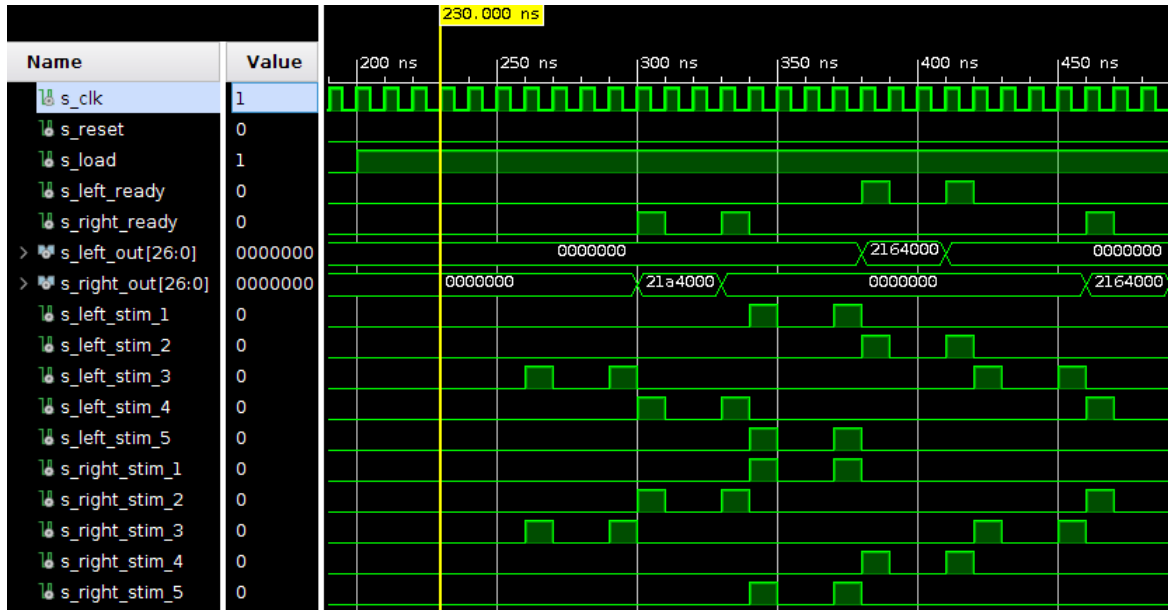


Figura 43 – Simulação comportamental de Camphs1D05G1RAV.

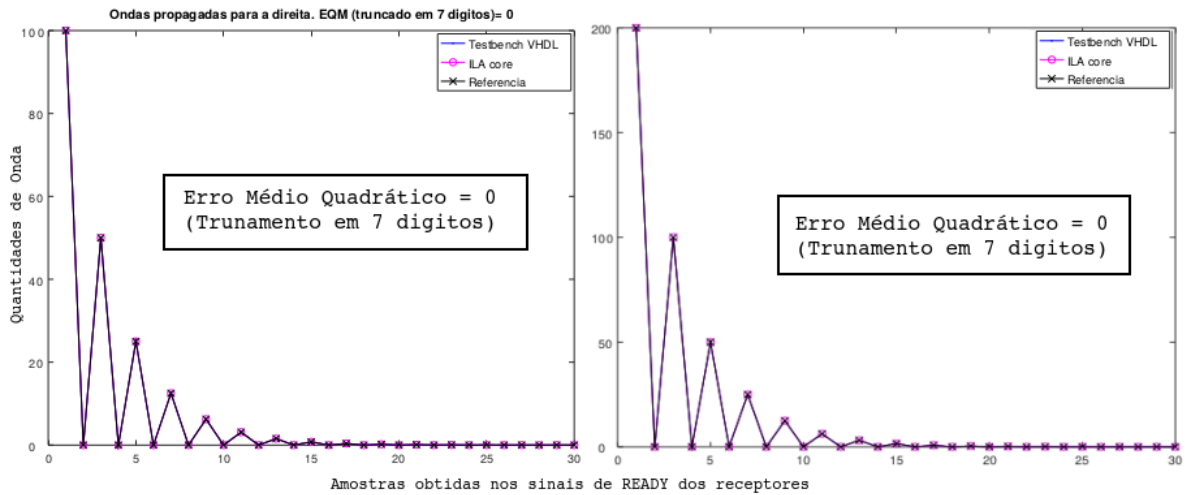


Figura 44 – Resultados coletados pelo ILA-core de Camphs1D05G1RAV, para a porta esquerda das células autômatas receptoras. À esquerda temos os resultados obtidos na porta esquerda, enquanto à direita temos os resultados obtidos na porta direita.

5.2.2 Organismo artificial Camphs1D05G1AL

A análise estática de *timing*, para o organismo artificial Camphs1D05G1AL, está apresentada na Figura 45. Na ferramenta Vivado 2018.3 Webpack, é possível destacar os caminhos críticos de *setup* e *hold*. Para *setup*, o parâmetro *Required Time* é maior do que o parâmetro *Arrival Time*, com folga de 2.775ns, como desejado. Neste caso, a origem está no PS, enquanto o destino está no IP-core Camphs1D05G1AL. Para *hold*, o parâmetro *Arrival Time* deve maior do que o parâmetro *Required Time*, fato que ocorre com folga positiva de 0.043ns. Neste segundo caso, a origem e o destino está na implementação do

AXI interconnect.

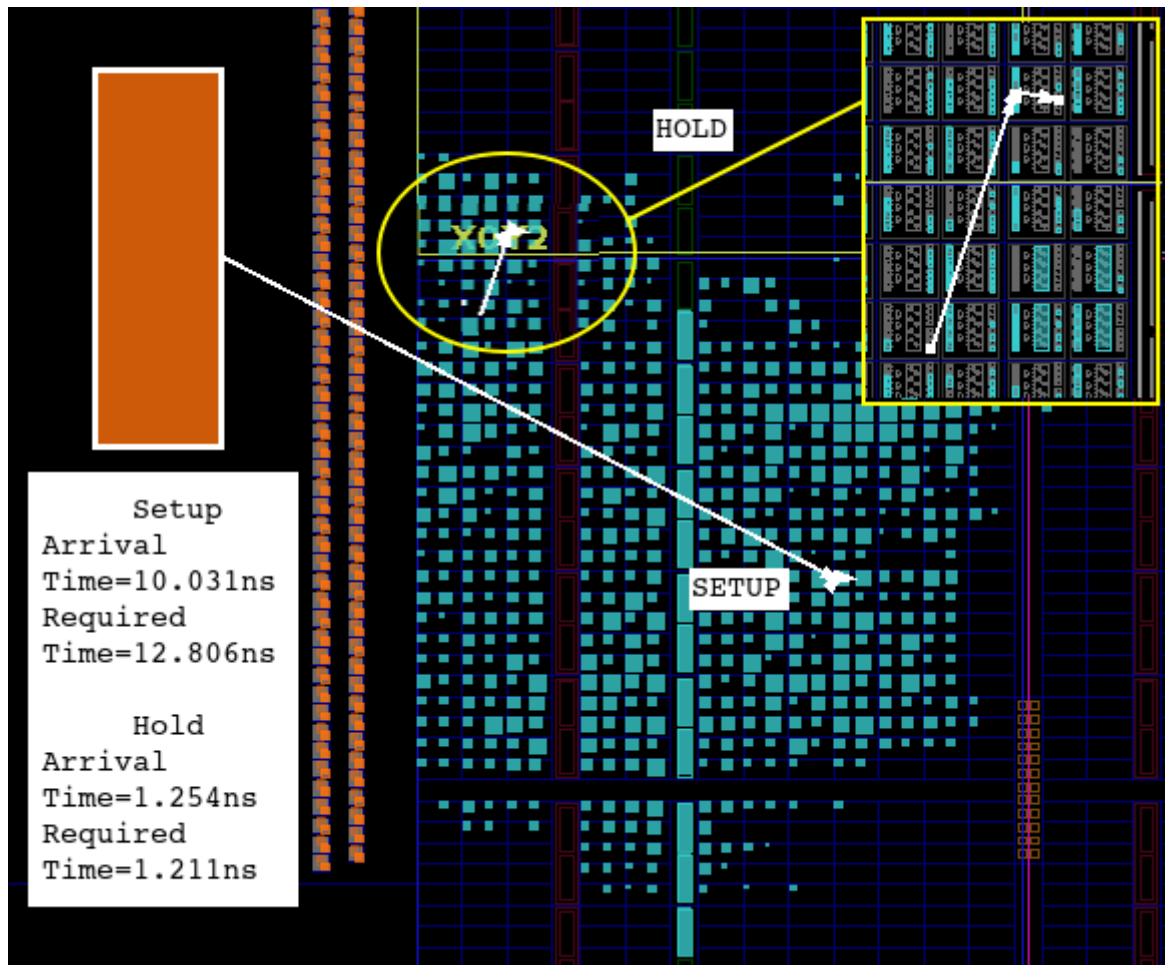
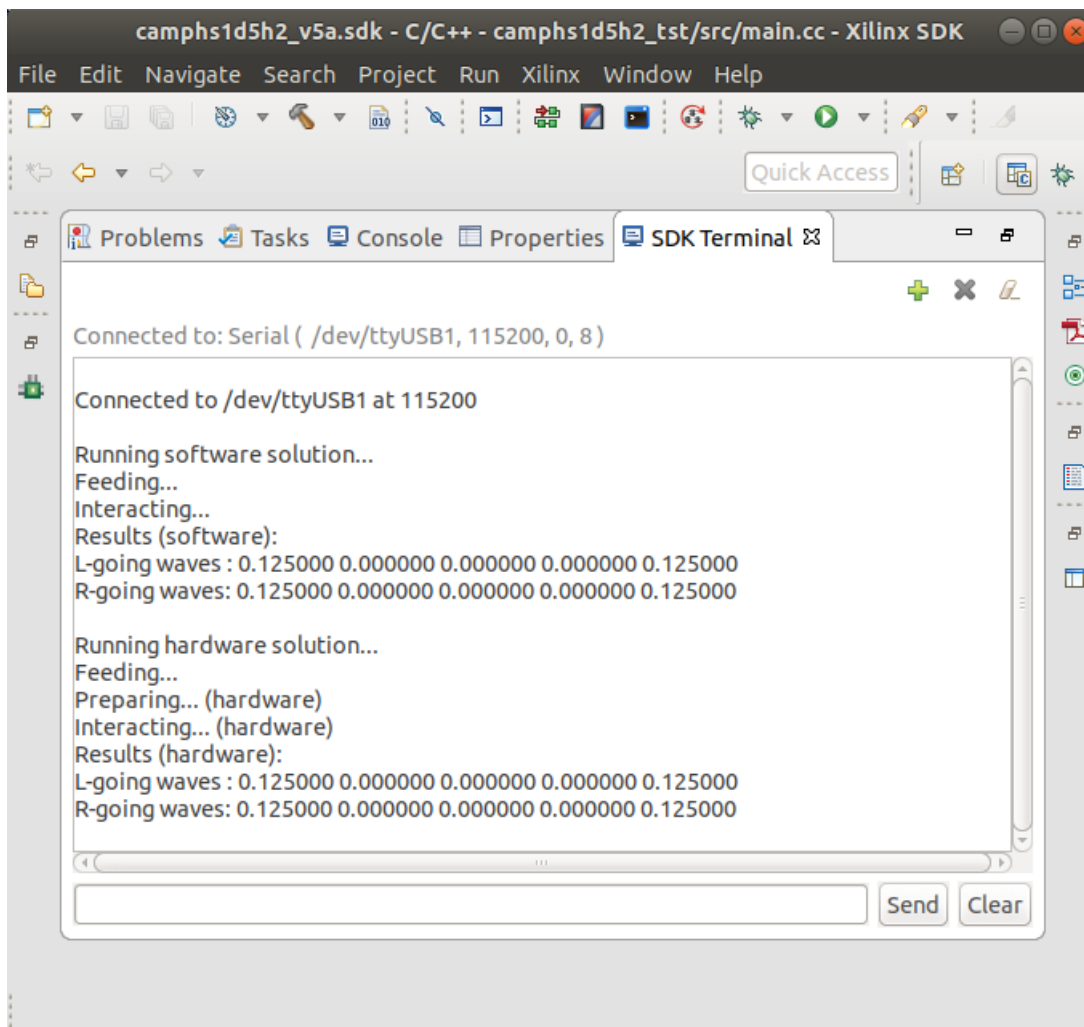


Figura 45 – Análise de tempo obtida para Camphs1D05G1AL. Em tela, temos os caminhos críticos de *setup* e *hold* (origens e destinos), indicando folgas positivas.

A seguir, na Figura 46, estão apresentados os resultados obtidos a partir da implementação do organismo artificial Camphs1D05G1AL. No software SDK, foi possível comparar a solução de software e hardware obtidas para o mencionado organismo. As soluções coincidem, conforme apresentado no modelo de referência apresentado na Tabela 7. O relatório de consumo de recursos, para o organismo em questão, está apresentado na Tabela 12, a seguir.

5.2.3 Organismo artificial Camphs1D05G1AS

A análise estática de *timing*, para o organismo artificial Camphs1D05G1AS, está apresentada na Figura 47, a seguir. Para *setup*, temos uma folga positiva de 0.905ns, enquanto para *hold* temos uma folga positiva de 0.021ns. O caminho crítico de *setup* possui sua origem e destino na implementação do IP-core Camphs1D05G1AS, enquanto o caminho crítico de *hold* possui sua origem e destino na implementação do *IP AXI*



```
camphs1d5h2_v5a.sdk - C/C++ - camphs1d5h2_tst/src/main.cc - Xilinx SDK
File Edit Navigate Search Project Run Xilinx Window Help
Quick Access
Problems Tasks Console Properties SDK Terminal
Connected to: Serial ( /dev/ttyUSB1, 115200, 0, 8 )
Connected to /dev/ttyUSB1 at 115200
Running software solution...
Feeding...
Interacting...
Results (software):
L-going waves : 0.125000 0.000000 0.000000 0.000000 0.125000
R-going waves: 0.125000 0.000000 0.000000 0.000000 0.125000
Running hardware solution...
Feeding...
Preparing... (hardware)
Interacting... (hardware)
Results (hardware):
L-going waves : 0.125000 0.000000 0.000000 0.000000 0.125000
R-going waves: 0.125000 0.000000 0.000000 0.000000 0.125000
Send Clear
```

Figura 46 – Resultados obtidos no software SDK para Camphs1D05G1AL. Em tela estão os resultados da Co-simulação praticada.

Direct Memory Access (AXI-DMA). O sinal de relógio de 100 MHz foi mantido para esta implementação.

Na Figura 48, a seguir, estão apresentados os resultados obtidos a partir da implementação do organismo artificial Camphs1D05G1AS. No software SDK, foi possível comparar a solução de software e hardware obtidas para o mencionado organismo. Note que os resultados contemplam o estado final das células autônomas e as saídas no receptor. As soluções coincidem, conforme apresentado no modelo de referência apresentado na Tabela 8. A estratégia utilizada foi a Co-simulação hardware/software, oferecida pela ferramenta. deste modo, a solução é apresentada nas duas formas disponíveis, para que a implementação, em hardware, seja avaliada. O relatório de consumo de recursos para o organismo artificial Camphs1D05G1AS, bem como para os demais implementados, está apresentado na Tabela 12, a seguir.

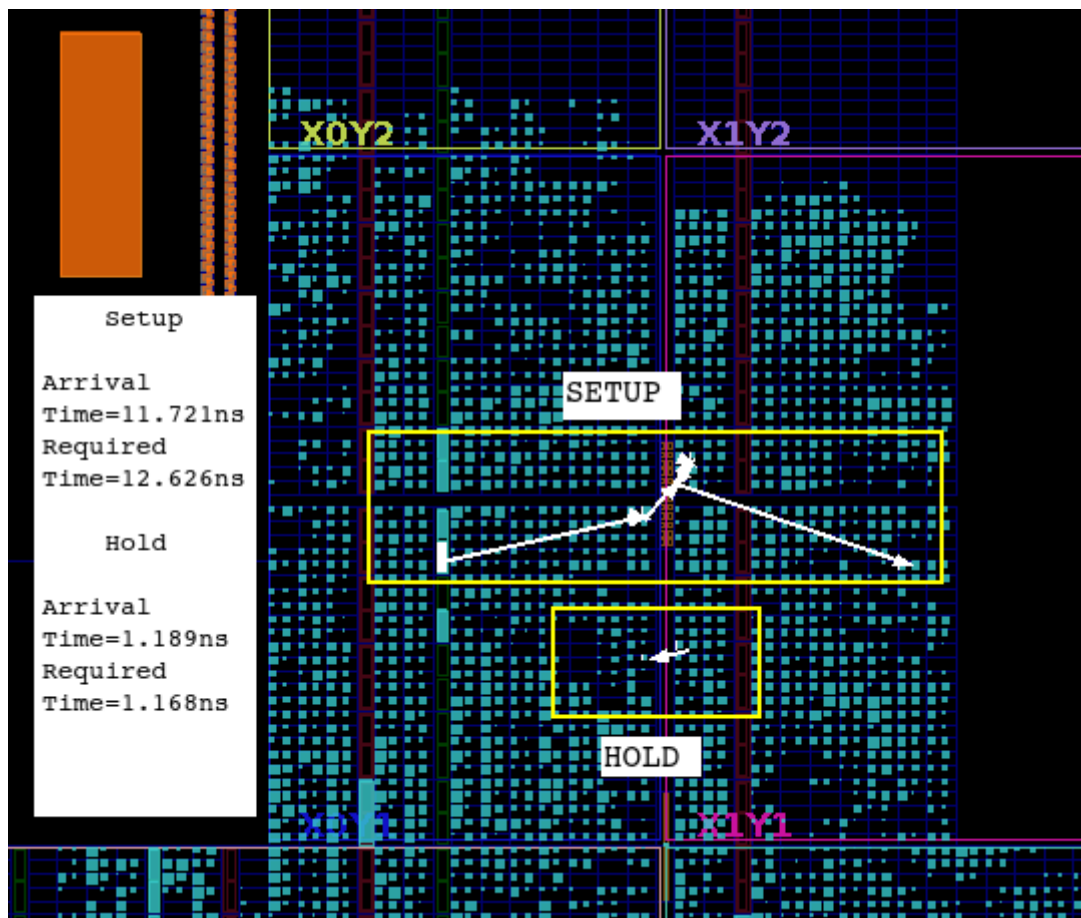
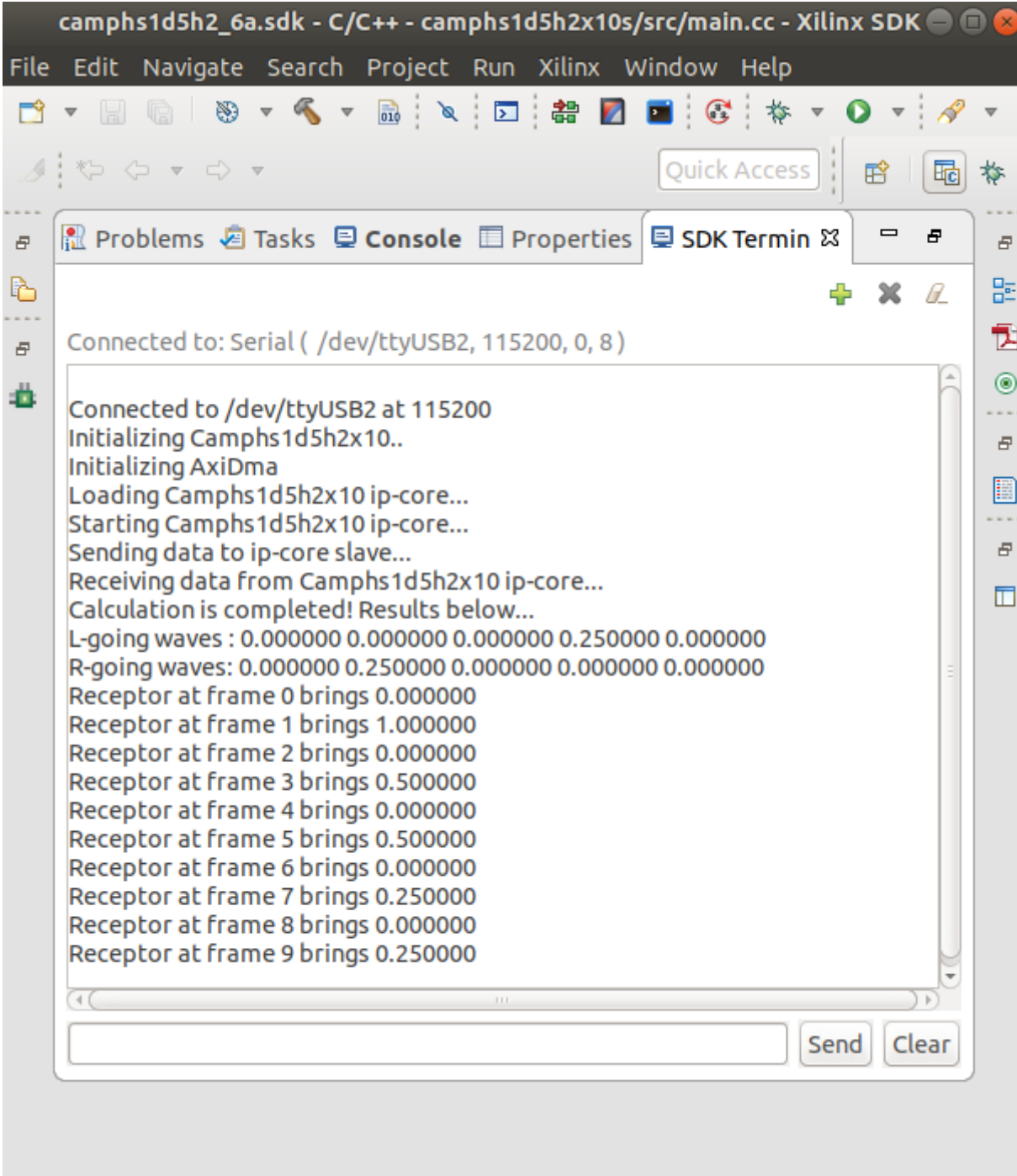


Figura 47 – Análise de tempo obtida para Camphs1D05G1AS. Em tela, temos os caminhos críticos de *setup* e *hold* (origens e destinos), indicando folgas positivas.

5.2.4 Relatório de utilização de recursos após as implementações

Na Figura 49 temos os recursos necessários para a implementação dos organismos artificiais aqui construídos. Nota-se, no organismo artificial Camphs1D05G1RAV, que temos um consumo de quase 50% dos recursos de I/O, aproximadamente 20% dos recursos de BRAM, além de 4% dos DSP, 4% dos FF, 4% de LUT e 1% de LUTRAM. Ainda que boa parte dos recursos esteja considerada para os IP-cores ILA, temos uma utilização considerável, fato que indica a necessidade de uma estratégia que integre os benefícios da aceleração em hardware (FPGA) com recursos de memória (software), visando uma maior escalabilidade da solução.

A Figura 49 também contempla os recursos necessários para o organismo artificial Camphs1D05G1AL. Vale lembrar que, desta vez, temos apenas o IP-core Camphs1D05G1 integrado com um barramento de dados do tipo *AXI4-Lite*, pois o PS está implementado em silício e não consome os recursos reconfiguráveis do kit. Vemos que foram consumidos 5% de DSP, 3% de LUT, 3% de BUFG, 2% de FF e 1% de LUTRAM. Ou seja, um consumo bem menor e mais próximo de uma aplicação típica do sistema CA Camphs1D.



```
camphs1d5h2_6a.sdk - C/C++ - camphs1d5h2x10s/src/main.cc - Xilinx SDK
File Edit Navigate Search Project Run Xilinx Window Help
Quick Access
Problems Tasks Console Properties SDK Termin
Connected to: Serial ( /dev/ttyUSB2, 115200, 0, 8 )
Connected to /dev/ttyUSB2 at 115200
Initializing Camphs1d5h2x10..
Initializing AxiDma
Loading Camphs1d5h2x10 ip-core...
Starting Camphs1d5h2x10 ip-core...
Sending data to ip-core slave...
Receiving data from Camphs1d5h2x10 ip-core...
Calculation is completed! Results below...
L-going waves : 0.000000 0.000000 0.000000 0.250000 0.000000
R-going waves: 0.000000 0.250000 0.000000 0.000000 0.000000
Receptor at frame 0 brings 0.000000
Receptor at frame 1 brings 1.000000
Receptor at frame 2 brings 0.000000
Receptor at frame 3 brings 0.500000
Receptor at frame 4 brings 0.000000
Receptor at frame 5 brings 0.500000
Receptor at frame 6 brings 0.000000
Receptor at frame 7 brings 0.250000
Receptor at frame 8 brings 0.000000
Receptor at frame 9 brings 0.250000
Send Clear
```

Figura 48 – Resultados obtidos no software SDK para Camphs1D05G1AS. Em tela estão os resultados da Co-simulação praticada.

Agora vamos ao consumo de recursos obtido da implementação do organismo artificial Camphs1D05G1AS, que novamente podem ser observados na Figura 49. Neste caso, temos o IP-core Camphs1D05G1 integrado com aos barramentos de dados *AXI4-Lite* e *AXI4-Stream*, para promover o *streamming* de dados do PL para o PS. Vemos que foram consumidos 5% de DSP, 13% de LUT, 3% de BUFG , 8% de FF e 4% de LUTRAM. Tam-

bém tivemos um consumo moderado, porém ligeiramente maior do que no caso anterior apresentado - o barramento *AXI4-Lite* consumiu uma maior quantidade de recursos do kit.

Resource	Utilization	Available	Utilization %
LUT	2236	53200	4.20
LUTRAM	151	17400	0.87
FF	3968	106400	3.73
BRAM	3.50	140	2.50
DSP	9	220	4.09
IO	57	125	45.60
BUFG	2	32	6.25

Camphs1D05G1RAV

Resource	Utilization	Available	Utilization %
LUT	1495	53200	2.81
LUTRAM	60	17400	0.34
FF	2136	106400	2.01
DSP	12	220	5.45
BUFG	1	32	3.13

Camphs1D05G1AL

Resource	Utilization	Available	Utilization %
LUT	6849	53200	12.87
LUTRAM	635	17400	3.65
FF	8609	106400	8.09
BRAM	2	140	1.43
DSP	12	220	5.45
BUFG	1	32	3.13

Camphs1D05G1AS

Figura 49 – Tabela de consumo de recursos para as implementações físicas praticadas.

Na Figura 50 temos os resultados da implementação de Camphs1D05G1RAV, indicando a porção da placa ocupada pelos componentes no FPGA. Vemos que a ocupação do ILA-core é significativa, chegando a ter a mesma proporção do IP-core Camphs1D05G1. Na Figura 52, temos os resultados obtidos para Camphs1D05G1AL. Notamos que a maior parte está destinada à implementação do IP-core Camphs1D05G1, ou seja, trata-se de um projeto mais enxuto. Já na Figura 51, vemos uma grande parte dos recursos serem destinada à implementação do barramento *AXI4-Stream* e ao *AXI-DMA* (recurso que viabiliza a comunicação direta entre a memória e o IP-core Camphs1D05G1). Os demais

Tabela 12 – Tabela de consumo de recursos para as implementações dos organismos (A) Camphs1D05G1RAV, (B) Camphs1D05G1AL e (C) Camphs1D05G1AS.

Implementações ->	A	B	C
Consumo dinâmico	0.030 W	1.274 W	1.313 W
Relógios	50%	1 %	2 %
BRAM	21%	-	<1%
Sinais	14%	1%	1 %
Outros	12%	2 %	<1%
PS	0%	96%	94 %
Consumo estático	0.106 W	0.134 W	0.136 W

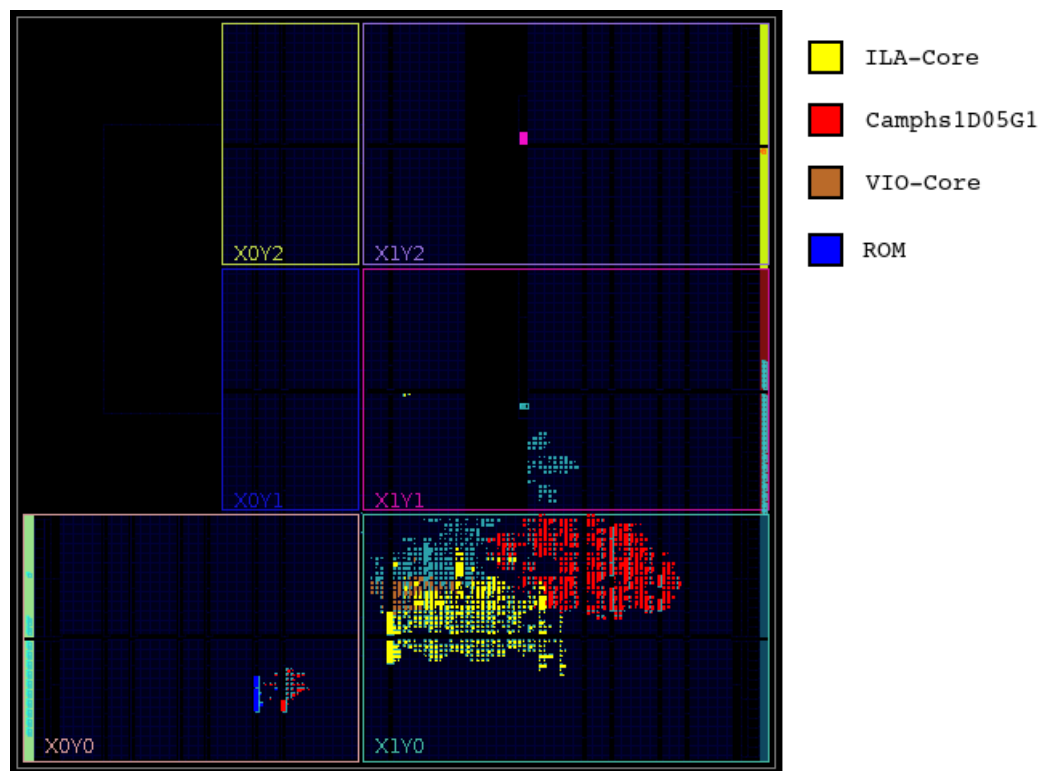


Figura 50 – *Layout* obtido da implementação de Camphs1D05G1RAV.

recursos se assemelham com o caso anterior, até mesmo porque temos aqui os mesmos recursos, e funcionalidades, do *AXI4-Lite* implementados.

Todos os relatórios de consumo de recursos podem ser apreciados na Tabela 12, acima. Nota-se que o consumo energético de Camphs1D05G1RAV é prioritariamente estático, o que significa que o kit gasta mais energia para manter a implementação dos recursos utilizados do que, propriamente, para manter o tráfego dos sinais envolvidos. Para os organismos artificiais Camphs1D05G1AL e Camphs1D05G1AS, é possível verificar que o consumo é prioritariamente dinâmico, ou seja, a maior parte de energia relacionada com o tráfego dos sinais envolvidos. O custo energético é maior do que o caso anterior, apresentado para Camphs1D05G1RAV, pelo fato de termos o PS envolvido na solução.

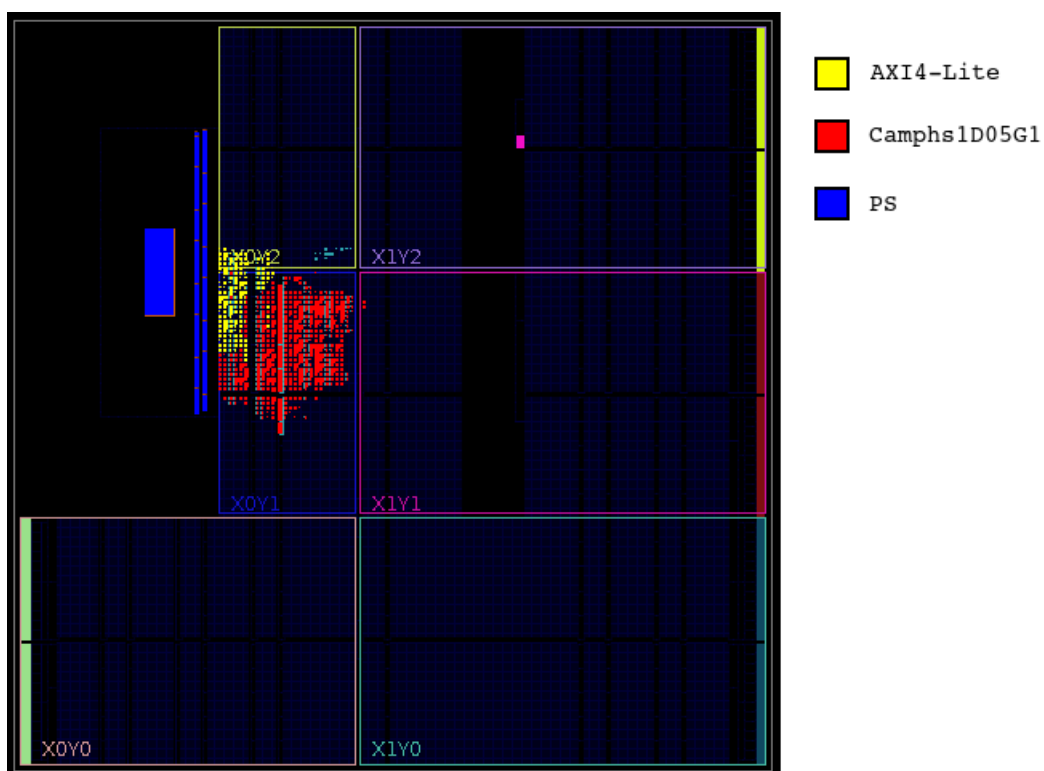


Figura 51 – *Layout* obtido da implementação de Camphs1D05G1AL.

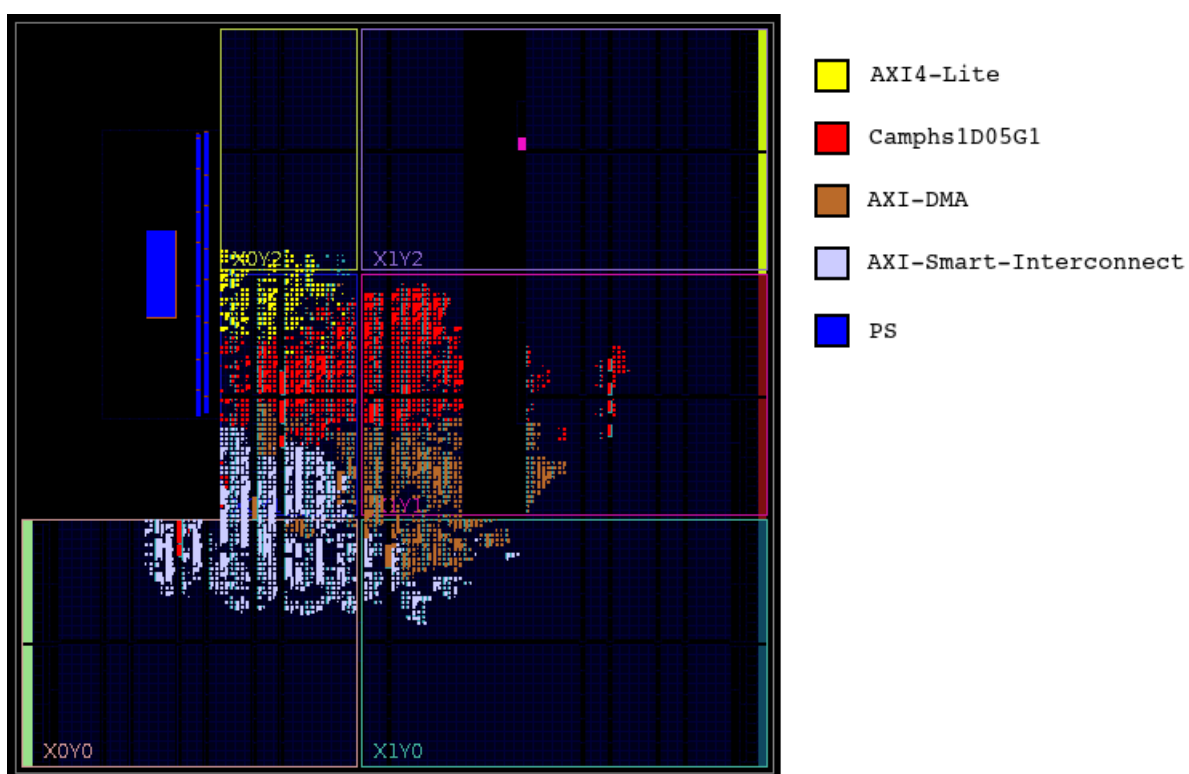


Figura 52 – *Layout* obtido da implementação de Camphs1D05G1AS.

6 Conclusões

O sistema CA é mais flexível e independente de restrições de ordem numérica e geométrica, fato que o coloca em vantagem em relação à solução tradicional da MGDO, na modelagem da propagação de ondas em sistemas físicos. É possível destacar as principais vantagens:

- possibilidade de se trabalhar com células desligadas. A MGDO, bem como outros métodos baseados em FDTD, não permite tal condição.
- fácil alteração do comportamento celular, bastando para isso modificar regras ou sinais de controle. A resolução numérica de uma EDO requer condições mais rígidas para convergência e estabilidade, para que se tenha coerência nos resultados obtidos,
- apresenta melhor desempenho computacional por estar diretamente envolvido com a discretização do fenômeno emulado. Em um processo envolvendo a resolução de EDO, temos geralmente uma quantidade significativa de cálculos auxiliares, necessários para aproximações numéricas, procedimentos de otimização, dentre outros.

Por outro lado, a formulação de um sistema CA pode exigir um grau mais elevado de criatividade e observação, para que se obtenha boa consistência na emulação do fenômenos físicos explorados. O engenheiro/projetista deve ser capaz de escolher, para cada situação, a estratégia que apresente o melhor equilíbrio entre o grau de realismo requerido e o custo da solução.

Todos os objetivos almejados para este trabalho foram atingidos. Na primeira parte do trabalho, o sistema CA Camphs1D foi desenvolvido, considerando duas células autômatas, com funcionamentos distintos, denominadas GCA1D e HCA1D. Estas células são capazes de operar todos os fenômenos básicos da onda, reproduzindo o comportamento de uma MGDO padrão. Códigos geradores de texto em VHDL, bem como funções decodificadoras de valores binários, foram produzidos para suportar as validações numéricas, que foram realizadas com sucesso.

Com relação às implementações em placa, mais especificamente sobre os sistemas CA SoC FPGA desenvolvidos, este trabalho traz algumas contribuições paralelas. Foi possível verificar a possibilidade de uma aceleração em hardware, realizada pelo PS através do uso de uma interface do tipo *AXI4-Lite*. Também foi possível integrar a este sistema uma interface do tipo *AXI4-Stream* com acesso direto a memória, permitindo realizar um fluxo de dados de entrada, paralelamente à execução dos Co-processadores.

No que se diz respeito somente às interfaces de dados, é possível levantar vantagens e desvantagens relacionadas às suas utilizações. A interface *AXI4-Lite* requer menor quantidade de recursos e, portanto, possibilita a implementação de um CORE CA maior. Por outro lado, esta interface é de baixa velocidade (em relação ao *AXI4-Stream*), possui um menor volume de dados e só permite que o estado final das CA seja externalizado.

A interface *AXI4-Stream*, em contraste com a interface *AXI4-Lite*, oferece uma maior quantidade de dados, com maior velocidade de comunicação. Temos a possibilidade de externalizar sinais (receptores) paralelamente à execução do CORE CA. Por outro lado, a implementação desta interface requer maior quantidade de recursos, e esta restrição acarreta na implementação de um CORE CA reduzido, em relação ao caso anterior.

A estratégia utilizada em todo este trabalho, no que se diz respeito ao desenvolvimento das arquiteturas, contemplou a implementação de um sistema CA completo, em placa. Esta estratégia não se mostrou satisfatória, quando pensada em termos do consumo de recursos implementados. A justificativa, para esta conclusão, está na baixa quantidade de células que podemos implementar no circuito, trabalhando em paralelo - a modelagem de um sistema físico, visando uma aplicação no mundo real, pode necessitar uma quantidade milhares de vezes maior do que as quantidades praticadas neste trabalho.

Ainda que as células autômatas sejam componentes síncronos, os organismos artificiais produzidos podem ser considerados assíncronos, pois os mesmos oferecem todas as condições necessárias para as células operarem em função dos sinais propagados. Deste modo, o sinal de relógio, praticado nas implementações, dita a velocidade de operação do sistema CA, e a latência de suas células autômatas (em ciclos de relógio).

Pensando na transposição deste problema, visando a escalabilidade para aplicações maiores, foi desenvolvida uma solução alternativa para o sistema CA Camphs1D. A denominada estratégia CA fragmentada está apresentada no apêndice I, desenvolvida em nível de software, por representar uma prototipagem inicial. Como parte dos trabalhos futuros, a implementação, em chip, da estratégia CA fragmentada foi desdobrada para um outro trabalho de conclusão de curso (em andamento). Tais implementações devem, obrigatoriamente, ser obtidas em projetos SoC FPGA, devido os benefícios do uso do acesso a memórias do tipo RAM - tal como foi implementado para os organismos artificiais Camphs1D05G1AL e Camphs1D05G1AS.

Referências

- ADAMATZKY, A. Game of life cellular automata. Springer, 2010. Citado na página 24.
- BAKHTERI, R.; CHENG, J.; SEMMELHACK, A. Design and implementation of cellular automata on fpga for hardware acceleration. In: . Third International Conference on Computing and Network Communications (CoCoNet'19): [s.n.], 2020. v. 171, p. 9. Citado na página 26.
- BLACKSTOCK, D. T. Fundamentals of physical acoustics. 2000. Citado na página 29.
- BOINGO, O. epic conway's game of life. 2011. Disponível em: <<https://www.youtube.com/channel/UCgqt1RE0k0Mir0LoyJRy2lg>>. Citado na página 23.
- CASTRO, M. L. A.; CASTRO, R. de O. Autômatos celulares: Implemenrações de von neumann, conway e wolfram. In: . Revista de Ciências Exatas e Tecnologia: [s.n.], 2008. III, n. 3, p. 89–106. Citado 4 vezes nas páginas 7, 18, 22 e 23.
- CHOPARD, B.; DROZ, M. Cellular automata modeling of physical systems. 1998. Citado na página 24.
- COLOMBO, F. B. Aplicação de autômatos celulares para simulação de processos de microfabricação. Universidade de São Paulo, 2017. Citado na página 26.
- DILIGENT. 2012. Disponível em: <<https://store.digilentinc.com/arti-z7-apsoc-zynq-7000-development-board-for-makers-and-hobbyists/>>. Citado na página 68.
- EMILE, O.; BRUNEL, M. Huygens' principle and young's experiment in the propagation of light beams. In: . American Journal of Physics: [s.n.], 2003. v. 71. Citado na página 26.
- FLITNEY, A. P.; ABBOTT, D. Towards a quantum game of life. Springer London, 2010. Citado na página 24.
- GARDNER, M. The fantastic combinations of john conway's new solitaire game "life". 1970. Disponível em: <<https://web.stanford.edu/class/sts145/Library/life.pdf>>. Citado na página 23.
- HADELER, K.-P.; MULLER, J. Cellular automata: Analysis and applications. 2017. Citado na página 17.
- HALL, D. E. Basic acoustics. 1987. Citado na página 29.
- HAN, F. Problems in solid state physics with solutions. Dalian University of Technology, China, 2011. Citado na página 25.
- JANOWICZ, M.; ORłOWSKI, A. Cellular automaton approach to electromagnetic wave propagation in dispersive media. Proceedings of The Royal Society A Mathematical Physical and Engineering Sciences, 2006. Citado na página 25.

- JOACHIM, S. Delannoy and tetrahedral numbers. In: . [S.l.: s.n.], 2007. v. 48, n. 3. Citado na página 23.
- KUTTRUFF, H. Acoustic an introduction. Taylor & Francis Group, 2007. Citado 2 vezes nas páginas 28 e 32.
- LIMA, A. C.; FERREIRA, J. C. Automatic generation of cellular automata on fpga. In: . IX Jornadas sobre Sistemas Reconfiguráveis: [s.n.], 2013. IX, n. 1, p. 51–58. Citado na página 27.
- MARKEL, J. D.; GRAY, A. H. Linear prediction of speech. 1976. Citado na página 33.
- MELOTTI, G. Aplicação de autômatos celulares em sistemas complexos: Um estudo de caso em espalhamento de epidemias. Universidade Federal de Minas Gerais, 2009. Citado na página 24.
- MOURA, H. G. Simulação da propagação bidimensional de ondas acústicas através de uma malha de guias digitais de ondas. Universidade Federal de Uberlândia, 2006. Citado 2 vezes nas páginas 19 e 31.
- MURPHY, D. T.; MULLER, J. Digital waveguide mesh modelling of room acoustic: Improved anechoic boundaries. Proc. Of the 5th Int. Conference on Audio Effects (DAFX-02), Hamburg, 2002. Citado na página 30.
- MUÑOZ, D. M. et al. Fpga based floating-point library for cordic algorithms. Universidade de Brasília, 2010. Citado na página 39.
- MUÑOZ, D. M. et al. Tradeoff of fpga design of a floating-point library for arithmetic operators. In: . Journal Integrated Circuits and Systems: [s.n.], 2010. v. 5, n. 1, p. 42–52. Citado na página 39.
- NEUMANN, J. V. Theory of self-reproducing automata. University of Illinois, 1966. Citado na página 22.
- PERDANG, J.; LEJEUNE, A. Cellular automaton experiments on local galactic structure. i. model assumptions. In: . Astronomy Astrophysics Supplement Series: [s.n.], 1996. v. 119, p. 231–248. Citado na página 24.
- PERES, M. C. P. O. M. Modelos urbanos baseados em autômatos celulares: integrando ambiente natural e o crescimento urbano. Universidade de Pelotas, 2005. Citado na página 24.
- SMITH, J. O. Physical modeling using digital. In: . [S.l.: s.n.], 1992. v. 16, n. 4, p. 74–91. Citado 2 vezes nas páginas 30 e 33.
- TREFETHEN, L. N. *Finite Difference And Spectral Methods For Ordinary And Partial Differential Equations*. Cornell University, 1994. Citado 2 vezes nas páginas 18 e 30.
- TRUJILLO, H. et al. Simulation of a vibrant membrane using a 2-dimensional cellular automaton. In: . [s.n.], 2015. v. 2015, n. 542507, p. 8. Disponível em: <<https://www.hindawi.com/journals/ddns/2015/542507/>>. Citado na página 25.
- WOLFRAM, S. *Cellular Automata And Complexity: Collected Papers*. Westview Press, 1994. Citado na página 24.

WOLFRAM, S. Elementary cellular automaton. 2020. Disponível em: <<https://mathworld.wolfram.com/ElementaryCellularAutomaton.html>>. Citado na página 24.

Apêndices

APÊNDICE A – Proposta de uma estratégia CA fragmentada para Camphs1D

A estratégia CA fragmentada proposta neste trabalho, para o sistema CA Camphs1D, tem como objetivo de subdividir o domínio em blocos menores, processados em um *loop*, através de um CORE CA implementado em hardware. Sendo assim, a implementação do CORE CA pode ter uma quantidade bem reduzida de células autônomas, pois o este CORE será reutilizado até o processamento de todas as células autômatas do domínio considerado.

O termo “fragmentada” também faz alusão ao fato de que, no interior do CORE CA, as células autômatas não estão conectadas entre si. Temos apenas um aglomerado de células autômatas, capazes de operar, simultaneamente, os sinais de entrada do CORE CA. Em outras palavras, o CORE CA deve possuir um barramento de dados capaz de enviar as informações que, outrora, seriam recebidas por estas células caso elas estivessem posicionadas, e integradas, dentro do domínio completo. De maneira semelhante, nas saídas do deste CORE CA temos os mesmos sinais que, outrora, deveriam ser repassados para as suas células vizinhas, caso as mesmas estivessem conectadas no domínio completo.

Percebe-se que temos uma inversão de funções. No caso da estratégia completa (adotada nas implementações deste trabalho) as células autônomas estão integradas em hardware, respeitando a geometria considerada. Por este motivo, não precisamos nos preocupar, em nível de software, com a integração dos sinais processados. No caso da estratégia fragmentada, os sinais processados estão integrados através de um algoritmo descrito em software e, portanto, não precisamos nos preocupar com a integração celular, em nível de hardware.

A seguir será apresentada a solução CA fragmentada do mesmo problema atacado pelo organismo artificial Camphs1DB9G2. Para esta solução, foi desenhado um CORE CA com apenas 5 células autônomas, capazes de resolver o domínio de 185 células proposto. Para efeito de uma melhor prototipação do circuito, foi utilizado um paradigma de programação orientada a objeto (POO). Todos os artefatos deste projeto estão apresentados a seguir.

Na Figura 53 temos o diagrama de classes implementado na solução fragmentada proposta. Nesta estratégia, o organismo artificial denominado Camphs1DH1G4, composto por 1 célula autômata do tipo HCA1D e 4 células autômatas do tipo GCA1D, deverá ser capaz de operar todas as interações celulares realizadas pelo organismo Camphs1DB9G2, em sua forma integral.

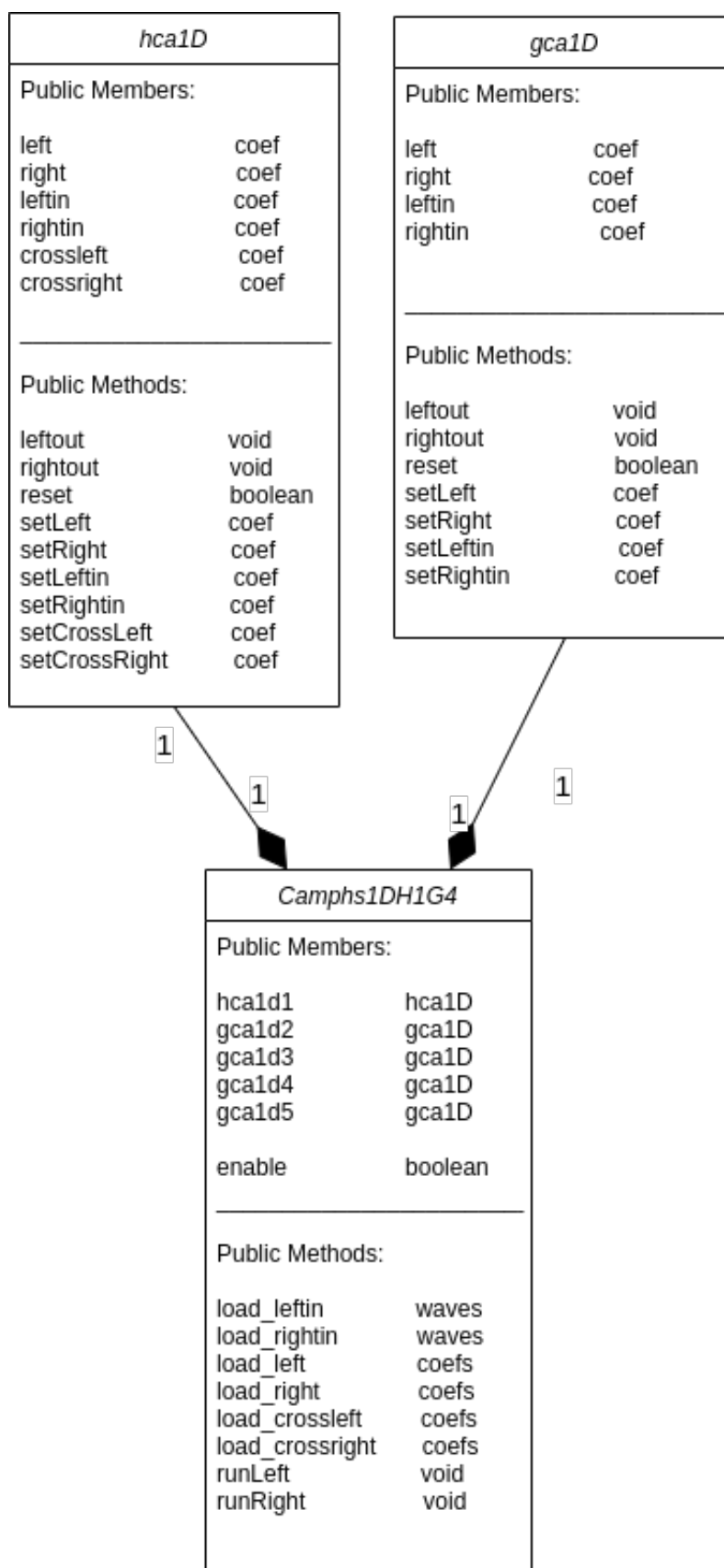


Figura 53 – Mapa de classes utilizado na solução fragmentada operada por Camphs1DH1G4.

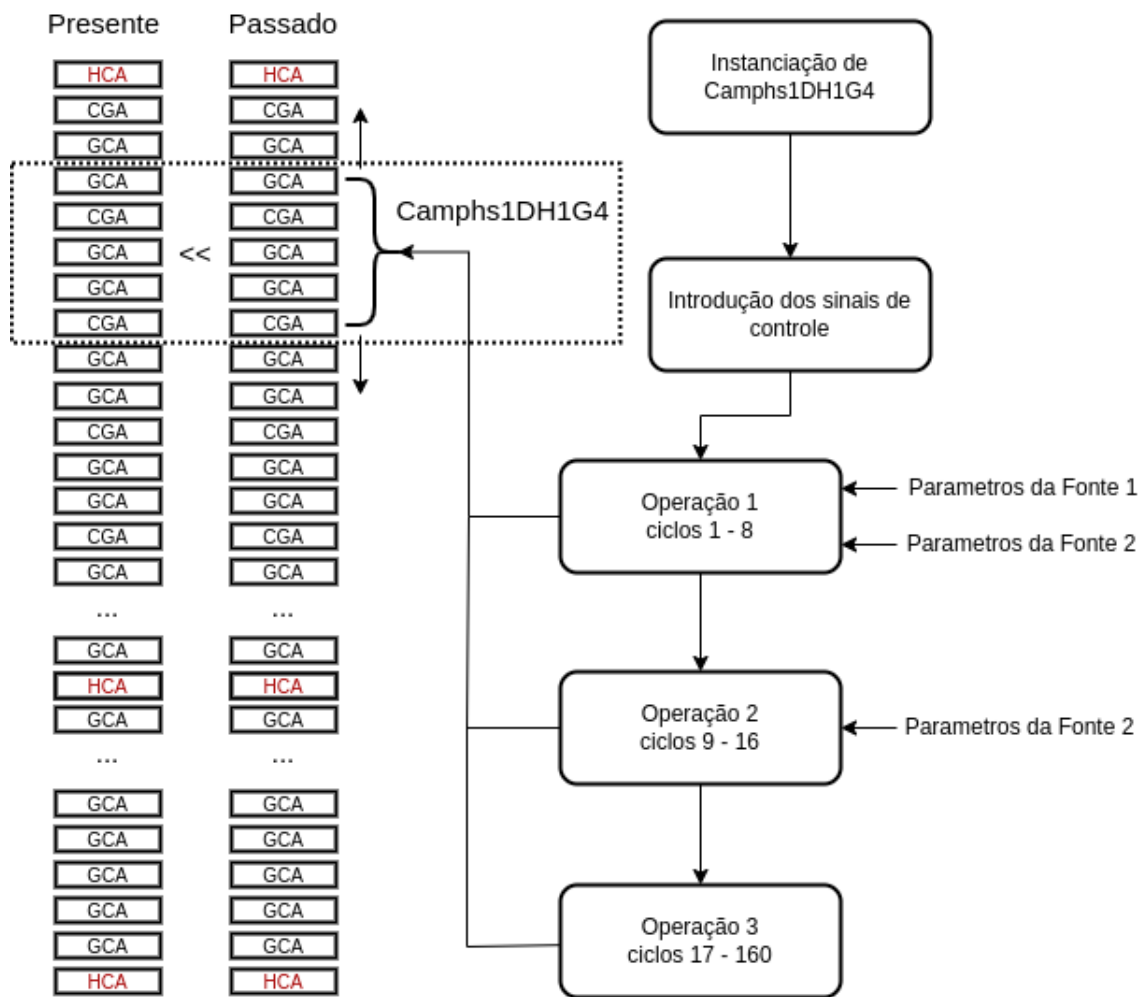


Figura 54 – Mapa de classes utilizado na solução fragmentada operada por Camphs1DH1G4.

O paradigma POO nos permite visualizar a solução em software de uma maneira mais próxima da realidade do circuito, pois temos o benefício do encapsulamento de informações, variáveis, funções e demais tratamentos de sinais. Seguindo este pensamento, os objetos representam os componentes que devem se interagir no circuito, operando os sinais envolvidos.

A estratégia fragmentada está apresentada, de maneira simplificada, na Figura 54. Temos três estágios de operação, um para cada configuração de fonte. Em cada estágio, para cada ciclo de relógio, o componente Camphs1DH1G4 percorre o vetor de memória processando os sinais, entre as células autômatas, como se suas células estivessem realmente naquelas posições. O resultado é imediatamente armazenado em outro vetor de memória, que tem o papel de guardar o estado presente. Após a varredura, o vetor de memória presente transfere seus valores para o vetor de memória passado, e um novo ciclo de relógio se inicia.

Temos, nesta simulação, 187 células autômatas “abstratas” sendo operadas por

apenas 5 células “reais”. A solução fragmentada apresentou a mesma solução do modelo de referência, adotada para validação do organismo artificial Camphs1DB9G2. Por outro lado, o organismo Camphs1DH1G4 necessitou de 46 operações para realizar o trabalho que o organismo Camphs1DB9G2 executa a cada ciclo de relógio.

APÊNDICE B – Modeling wave propagation using cellular automata on Chip

O trabalho intitulado *Modeling wave propagation using cellular automata on Chip* foi publicado no 34th *Symposium on Integrated Circuits and Systems Design* (SBCCI 2021). O mesmo pode ser apreciado nas páginas seguintes.

Modeling wave propagation using cellular automata on Chip

1st Henrique G. Moura

Automotive Engineering Undergraduate Course
University of Brasília
Brasília, Brazil.
hgmoura@unb.br

2nd Daniel M. Muñoz

Electronic Engineering Undergraduate Course
University of Brasília
Brasília, Brazil.
damuz@unb.br

Abstract—Cellular automata (CA) are often used for physical modelings in which space and time are discrete by assumption. An artificial organism can be considered as a circuit that controls a high-level integrated CA architecture, developed for some mathematical purpose. In this work, propagation and reflection wave phenomenon has been reproduced using CA systems, in a one-dimensional lossless medium. The main objective here was to develop a modeling tool, called *vCAgen*, which provides the VHDL code for the CA-based system implementation. The CA-based system called *Camphs1D*, composed of 185 cells and two receiver channels, was mapped on a Xilinx Zynq Ultrascale 7EV1156 device, using 27-bit width floating-point arithmetic representation. The proposed *Camphs1D* was effectively mapped on the Zynq device, achieving an operational frequency of 100 MHz and throughput of 25 MOPS. Numerical comparisons allow us to conclude that the proposed CA-system, which is based on simple arithmetic operations, achieves the same results that a reference model based on the well known d’Alembert one-dimensional discrete wave solution, with a Mean Square Error (MSE) in the order of 10^{-13} .

Index Terms—Cellular automata, Wave phenomenon, SoC FPGA.

I. INTRODUCTION

Physical modelings of acoustics waves, electromagnetic waves, fluid dispersion, and its interactions with obstacles in space, commonly make use of partial differential equations (PDEs) [20]. Traditional methods for solving PDEs such as finite difference methods (FDM) require a fine discretization of space for achieving accurate solutions and are very dependent on initial and boundary conditions [16] [18]. For this kind of solution, large computational resources are required, which makes it inefficient for modeling complex PDE systems.

In order to overcome the challenges imposed by continuous space-time models, the cellular automata (CA) can model complex scenarios using simple preconditions over the behavior of spatial and temporally discrete units, in a wide variety of fields. However, there is a problem regarding the computational costs of CA-based systems in cases of huge discrete domains, which results in exhaustive simulations. This issue is a drawback for designing systems that commonly imposes the necessity of solving the inverse problem and then involves hundreds or thousands of simulations of the associated forward model. One possible solution to that problem is the hardware acceleration

of the CA-based models, overcoming the problem of repetitive and enormous calculations [1] [9].

This work presents a modeling tool for one-dimensional CA systems called *vCAgen*, which provides the VHDL code, allowing the obtained models to be implemented in system-on-a-chip (SoC) devices, accelerating its execution time. In particular, this work proposes a new CA model for one-dimensional discrete domains, forced by monochromatic harmonic signals, in a lossless medium. Obstacles have been used to model the frontiers and a membrane inside the linear grid, allowing the representation of basic wave phenomena and their constructive interactions.

A reference software model was built using C language, to reproduce a traditional numerical solution for the partial differential one-dimensional wave equation, in discrete domain as well. The implemented cell’s core, for the designed CA system, called *Camphs1D*, makes use of floating point arithmetic IP-cores with custom precision, to hand over all the arithmetical needed processes. It is important to highlight that the custom precision provided by the IP-cores, allow the designer to analyze the trade-off between hardware resources consumption and numerical precision. For a deep study on the floating point IP-cores, readers can refer to the work in [13], [14].

In order to validate the proposed *Camphs1D* solution, a CA system, composed of 185 cells and two source channels, was mapped on a Zynq Ultrascale 7EV1156 device from Xilinx. Behavioral simulations demonstrate the accuracy of the proposed CA-system with respect to the PDE-based model, achieving a mean square error (MSE) of about $10E-13$.

II. REFERENCE MODEL

A plane wave is a kind of a pure mathematical function, and it does not exist in the real world. However, we must consider these wave functions as the first step in any matter of computation. Considering a harmonic wave traveling along positive x -axis [10] [3],

$$\frac{\partial^2 p(x, t)}{\partial x^2} - \frac{1}{c_0^2} \frac{\partial^2 p(x, t)}{\partial t^2} = 0, \quad (1)$$

The well-known d’Alembert solution for plane waves [2] states that a wave quantity, defined by Equation 1, $p(x, t)$, can be

represented by a superposition of harmonic concurrent patterns at the same point in the space, as follows,

$$p = f(x - c_0t) + g(c + c_0t). \quad (2)$$

It is shown in Fig. 1 an aleatory instant of the transient solution obtained from the reference model. It was constituted of one acoustic source and some obstacles, in a lossless or ideal medium. Some harmonic signals are traveling in both directions which are about to hit an obstacle. Note that equations 1 and 2 do not consider reflections due to obstacles, but only wave dispersion and its constructive interactions. To do it, some numerical efforts and additional theories are needed, however, that development is out of the scope of this paper.

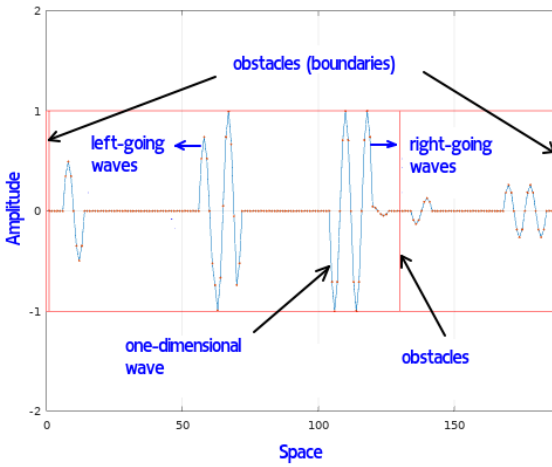


Fig. 1. Simulation of the propagation of an uni-dimensional plane wave.

In this work, several assumptions were made to construct a reference model for a uni-dimensional plane wave and its interactions with the medium and obstacles [7]. The mathematical characteristics of the reference model are shown below.

- Mesh size of 20 meters with 187 cells.
- Velocity of sound wave equal to $344 \sim m/s$.
- Characteristic impedance equal to $400 \sim \text{Rayls}$.
- A membranes with impedance set to $2667 \sim \text{Rayls}$.
- Two membranes at the frontiers of the mesh with impedance set to $1200 \sim \text{Rayls}$.
- Mesh update frequency set to $3200 \sim \text{Hz}$.
- Simulation time equal to 0.05 seconds (160 frames).
- Input frequencies equal to $400 \sim \text{Hz}$. That means 8 points to plot per period.
- Two sinusoidal inputs at different locations, with 1 and 2 complete periods.

The whole numerical formulation can be found in the references [12], [17], [6]. Sign definitions and algorithms applied to the reference model have been neglected in this work because it is out of scope. In summary, this work uses the d'Alembert solution for plane waves as a reference model,

which is computationally solved by an FDM technique, thus allowing for numerical comparisons with the solution achieved by the proposed cellular automata model, explained in the next section.

III. CELLULAR AUTOMATA MODELING

A. Background

In an informal way, but satisfactory, a CA system can be adopted wherever the following characteristics exist [4].

- 1) It consists of an n -dimensional grid of cells.
- 2) It evolves over discrete time-steps.
- 3) Each cell is a kind of finite-state unit whose operation or output values are determined only by its current state and its neighborhood.
- 4) The neighborhood relations are local and uniform.

In the meantime, the main characteristic of a CA system (one which supports this work) falls within its nature. A CA system is supposed to operate in a truly simple mode, with no geometry restrictions, no trajectories or functions that could stop the spontaneous evolution of each cell.

The multipurpose mathematician Von Neumann was the pioneer in creating a CA system, in the late forties, during his studies over computing machines. The results of this work were published after his death, in 1966, by Arthur W. Burks, [15]. Von Neumann treated his CA systems as a mathematical idealization of natural organisms, including surviving rules, reproducing mechanisms, and temporal evolution. In the seventies, the CA systems became more popular with Conway's "Game of Life" [8]. This simple system needs only three survival rules to show formidable and unexpected results.

Over the years, CA systems have been applied as a versatile modeling tool in different fields of knowledge such as acoustics [19], electromagnetism [11] and fabrication processes [5].

B. CAMPHSID: Cellular Automata Modeling of One-dimensional Physical Systems

The CA system called *CampHSID* was designed to represent, as simple as possible, the basic wave phenomena obtained from the reference model simulation. In terms of its nature, this CA system brings the following operating rules.

- 1) Wave quantities must be "transmitted" through dispersion cells along both wave guides direction ($+x, -x$), separately. A dispersion cell does not change its behavior in life.
- 2) A generating cell must not receive but force wave quantities to the neighborhood, in each discrete time, in both directions. For that, values must be provided to that cell by an external agent, inside the artificial organism. A generating cell might have its behavior changed by the artificial organism to a dispersion cell, any time, either forward or backward.
- 3) A hybrid cell must receive wave quantities like a dispersion cell and transmit wave quantities like a generating cell. A hybrid cell does not change its behavior in life.

These rules could be materialized through the figures below. These images bring some mathematical relations, according to

TABLE I
CODE PROFILING FOR BOTH STRATEGIES: FDTD AND CA.

Parameter	FDTD solution	CA Solution
Executions number	1000	1000
Time processing	3.36 sec.	3.68 sec.
Multiplication operations	89446	57121
Summation operations	59526	864
Division operations	9	0

the finite differential approaching used in the reference model. The superscript symbols '+' and '-' represent right-going and left-going waves portions, respectively, while the subscribed symbol 'n' represents time.

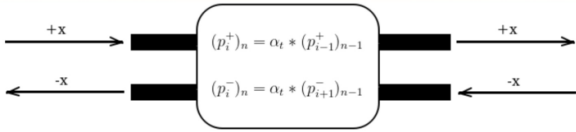


Fig. 2. Rule 1 - Dispersion mechanism.

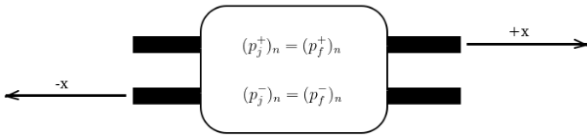


Fig. 3. Rule 2 - Generating mechanism.

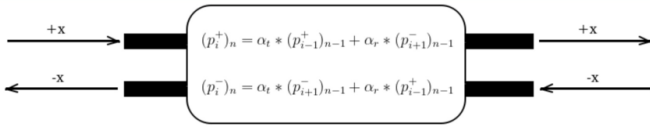


Fig. 4. Rule 3 - Hybrid mechanism.

Just before the VHDL implementation of Camphs1D, a very important and quickly investigation was accomplished, by using of code profiling. Both strategies, FDTD and Cellular automata, had been compared in a software environment. As it can be seen in Table I, the results highlight the CA approaching, pointing out a good opportunity to work with floating point operations, on chip.

Following these rules, during the VHDL description of Camphs1D, it was realized that all of them could be better represented by two components, or two cells. Both cellular automatas, GCA1D and HCA1D, are show in Fig. 5. On one hand, the cellular automata GCA1D has been designed to generate and disperse wave quantities, according to rules 1 and 2. Some of them will never be requested as a generating cell in life because it might not exist any source wave at its position. The GCA1D's core makes use of three floating-point multipliers, two 3x3 multiplexers, two OR gates, and one register. The floating-point operations are supported by the FPUPACK library [13] [14], using a 27-bit width representation. We have

also a register bank of 27 bits to store a constant value, normally set to zero. The GCA1D cells work in synchronous mode, with a latency of four clock cycles, due to internal floating-point operations. On the other hand, the automata HCA1D is in charge of wave dispersion at the surfaces of the medium, which means, wave reflections from the obstacles. This cell makes use of four floating-point multipliers and two adders. Because of the same reasons mentioned before, this cell must work with the same latency of four clock cycles. For a more detailed description regarding all signals in Fig. 5, see Table II.

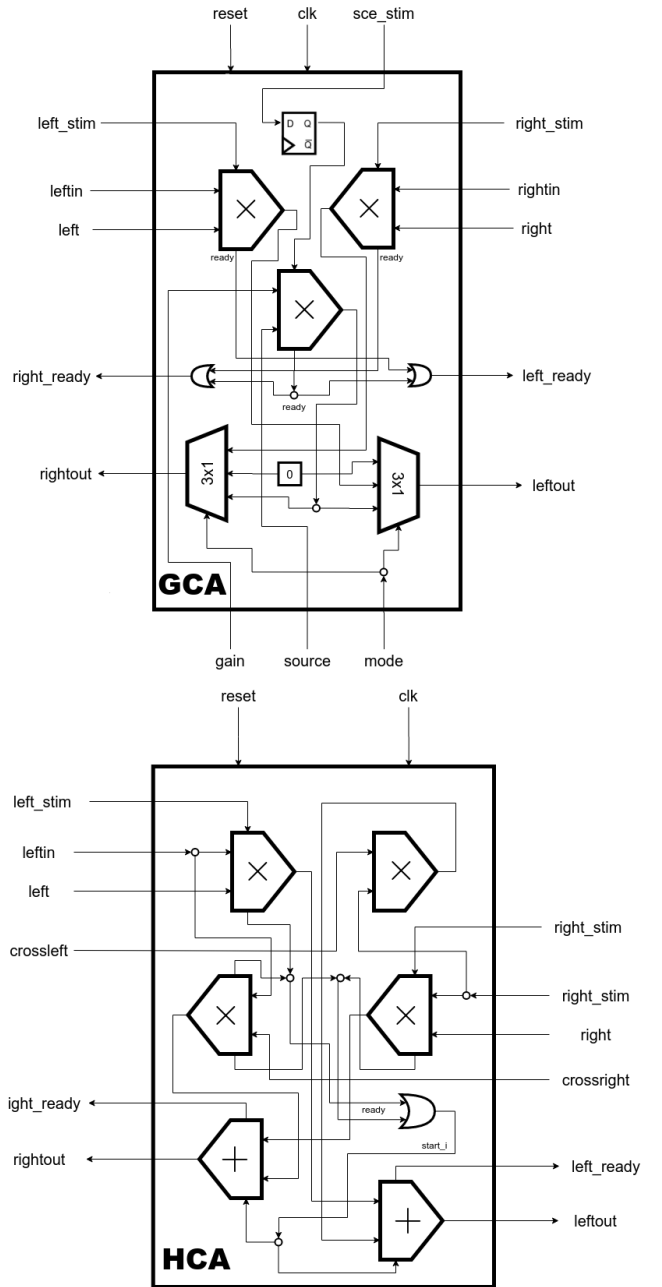


Fig. 5. Operational blocks used in the GCA1D and HCA1D design.

TABLE II
DESCRIPTION OF USED SIGNALS FOR THE GCA1D AND HCA1D CELLS -
VCAGEN JOB.

Signal/Gate	IN/OUT	Bits	Description
CLK	IN	1	clock signal
RESET	IN	1	Register/device initialization
MODE	IN	2	Generating/Dispersion/OFF
SOURCE	IN	27	Forced value for sources
GAIN	IN	27	internal gain for sources
SCE_STIM	IN	1	generating stimulus
LEFT_STIM	IN	1	Dispersion stimulus (left)
RIGHT_STIM	IN	1	Dispersion stimulus (right)
LEFT	IN	27	left wave coefficient
RIGHT	IN	27	right wave coefficient
LEFTIN	IN	27	wave quantities from the left
RIGHTIN	IN	27	wave quantities from the right
CROSSLEFT	IN	27	left wave coefficient
CROSSRIGHT	IN	27	right wave coefficient
RIGHT_READY	OUT	1	calculation is done
LEFT_READY	OUT	1	calculation is done
RIGHTOUT	OUT	27	wave quantities to the right
LEFTOUT	OUT	27	wave quantities to the left

IV. CAMPHS1D GENERAL ARCHITECTURE

As stated before, an artificial organism is an integrated and auto-sufficient system, able to control, by using some input-output data, all the cells that compose the organism. An artificial organism can also run for an indeterminate period of time.

The main architecture of the CA-based organism is responsible for the integration of the cells in a functional environment, fully programmed and adjusted to perform the whole computation of the system. It is important to highlight that the main architecture description may be a tough task, because of the high number of cells, components, and routing signals. In order to make it easier and less costly, a modeling tool for one-dimensional CA systems called *vCAgen* was developed in Octave. The *vCAgen* tool is able to describe any typical *Camphs1D* artificial organism, including its sources, receivers, and topology. The *vCAgen* tool is also able to automatically generate the VHDL code that describes the whole CA system. The following list shows the current characteristics of the *vCAgen* tool.

- Control signals definitions.
- Cell to cell interaction, connecting READY signals to its adjacent waveguides.
- Source values attributions according to its respective data banks and MODE signal operation.
- Signal capturing at the GCA1D receiver cells.
- Automatic VHDL component instantiation.

In this work, the *vCAgen* tool was used to create an artificial organism, called *Camphs1DB9G2*, composed of 185 cells and two source channels. Such a configuration is minimally satisfactory to reproduce the reference model in an FPGA chip. The main architecture of the *Camphs1D* is shown in Fig. 6. Note that the cells form two waveguides, each one for a single direction. Several registers are connected to each cell, providing control signals for configuring its expected behavior.

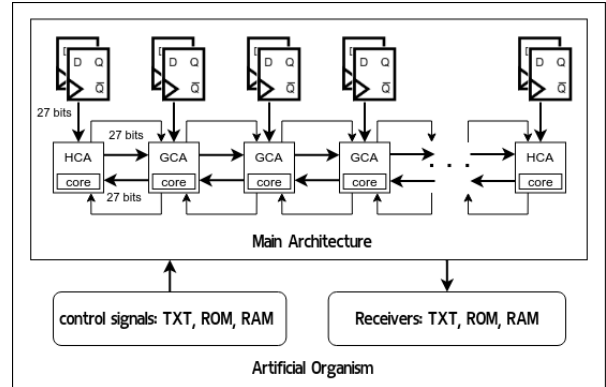


Fig. 6. Operational block used in the design of *Camphs1D* general architecture.

Currently, the *vCAgen* tool provides two ways of validating the behavior of the *Camphs1D* architecture, by injecting the input data and control signals using a TXT file (behavioral simulation) or by using ROMs in case of physical implementation. The signals used by *vCAgen* are shown in II. The output data can be exported to TXT files or can be stored in RAM memory and accessed using the Integrated Logic Analyzer (ILA) core from Xilinx. Eventually, the Virtual Input-Output (VIO) can be used to set inputs and verify outputs.

V. EXPERIMENTAL PROCEDURE

The implemented *Camphs1DB9G2* was validated using behavioral simulation. The input data (TXT files) includes *START*, *MODE* and *SOURCE* signals. The output data were collected by the receivers at the 20th and 150th positions, and at every clock cycle, for both directions or waveguides, were written in the respective TXT file.

The *Camphs1DB9G2* system has tree HCA1D cells at 2nd, 130th, and 186th positions. It has also two GCA1D cells at 44th and 94th positions, performing two and one complete wave periods, respectively. See Table III for details regarding the generated values. Others cells, namely non-wave-generating and non-hybrid, are GCA1D-type also, but perform only dispersions in both directions, according to their neighborhood stimulus. Two of them have their waves quantities written to TXT files (receivers), in a hexadecimal representation.

The experiment starts with the initialization of the registers, which is done by a *RESET* signal. Then, the CA system is ready to run at any time according to the very first *START* signal. The first generating cell transmits its wave quantities and its *READY* signals, which are connected to the neighborhood, and so on, indefinitely.

The number of lines on each control signal (TXT files) must be synchronized to the system throughput, which is of four clock cycles multiplied by the number of wave discrete events. Thus, the output data needs to be filtered and decoded to be compared to the reference model, in software. Finally, the proposed architecture was characterized in terms of resources

TABLE III

SIMPLE DISCRETE SINUSOIDAL SIGNAL IN TUNE OF 400 ~Hz BY USING A 3200 ~Hz UPDATE FREQUENCY. THESE VALUES WERE FORCED AT SOURCE POINTS IN CAMPHS1DB9G2.

Binaries in a 27-bit format	Hexadecimals	400 ~Hz@3200 ~Hz
001111110011010100000100111	1F9A827	0.70710563659668
001111111000000000000000000	1FC0000	1
001111110011010100000100111	1F9A827	0.70710563659668
000000000000000000000000000	0000000	0
101111110011010100000100111	5F9A827	-0.70710563659668
101111111000000000000000000	5FC0000	-1
101111110011010100000100111	5F9A827	-0.70710563659668
000000000000000000000000000	0000000	0

consumption, numerical error and precision when compared to the reference model, operational frequency, and execution time (latency and throughput).

VI. RESULTS

All results obtained from the *Camphs1DB9G2* organism were correlated to the reference model. The results provided 160 time frames (each one with 185 values). Figs. 7 and 8 present the obtained CA results in the same plot with the reference model.

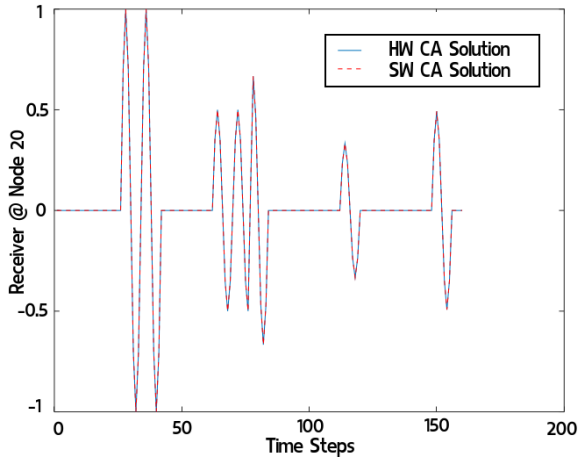


Fig. 7. Decoded results obtained from the “A” channel of Camphs1DB9G2 and its reference values.

In order to compare the differences between the *Camphs1DB9G2* and its reference model, the mean square error (MSE) criterion was adopted. The obtained CA solution, for both receivers, comes too close to the reference model with MSE in the tune of 1×10^{-13} or less, as shown in Table IV. The low margin of errors occurs because of the rounded applied data. Since the implemented model is vulnerable to the effects of digitization and numerical losses, it is not expected for an aleatory sampling data.

It is worth mentioning that all VHDL codes have been synthesized for the Zynq Ultrascale device of the ZCU104 development kit (chip xczu7evffvc1156-2-e). Resources utilization of the main architecture and each cell can be appreciated in Table V.

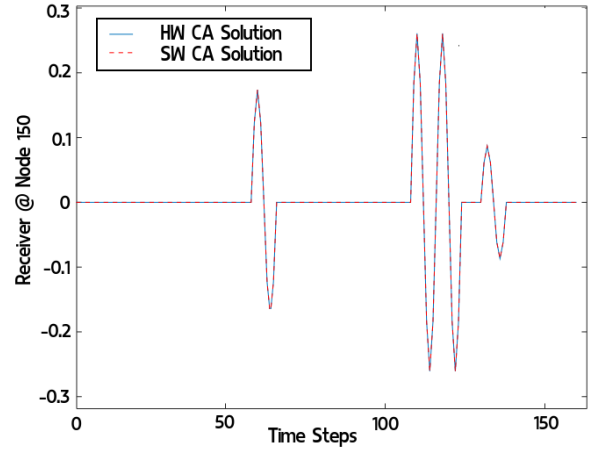


Fig. 8. Decoded results obtained from the “B” channel of Camphs1DB9G2 and its reference values.

TABLE IV
MEAN SQUARE ERRORS OBTAINED FROM *Camphs1DB9G2* IN BOTH RECEIVER CHANNELS.

Receptor	MSE
A	3.45×10^{-13}
B	1.53×10^{-14}

As mentioned before, the cell’s core uses floating point multipliers and adders with 27 bits. It lead us to connect the cells outputs with their neighborhood, directly. Another important point it that there is any state machine (FSM) inside the mentioned CA. This operational characteristic improves the *Camphs1D* speed. Consequently, the systems scalability ends up being a function of the core amount that works in parallel.

The execution time of the proposed *Camphs1D*, and its possible artificial organisms, depends on the latency and throughput of the cells as well as on the clock frequency. Because of the latency of four clock cycles of the *GCA* and *HCA* cells, the achieved throughput of the cells at a clock frequency of 100 MHz is around 25 MFLOPS. The latency of a *Camphs1D* organism depends on its length and the position of the receivers; therefore, after the initial latency, each receiver can provide a valid output every four-clock cycles, achieving

TABLE V
RESUME OF RESOURCES CONSUMPTION FOR *Camphs1DB9G2*.

Architecture	LUTs (230400)	Registers (460800)	DSPs % (1728)
Main architecture	22422 (9.73 %)	21924 (4.76 %)	372 (21.53 %)
GCA cell	114 (0.049 %)	118 (0.026 %)	2 (0.116 %)
HCA cell	825 (0.358 %)	220 (0.048 %)	4 (0.231 %)
HCA cell (course)	303 (0.132 %)	83 (0.018 %)	1 (0.058 %)

25 MFLOPS.

VII. CONCLUSION

This work presented a novel architecture for cellular automata (CA) systems that can be used to model any one-dimensional grid of cells that evolves over discrete time steps. The developed CA system, called *Camphs1D*, has shown itself to be flexible and scalable, taking advantage of regular structures, that can be easily integrated through a high-level modeling tool called *vCAgen*. The proposed *vCAgen* tool is able to provide the VHDL code of the CA system and makes use of custom precision floating-point operators. In particular, this tool was used to implement a case study of a one-dimensional CA composed of 185 cells, allowing plane-wave phenomena to be simulated. Additional advantages are regarding circuit's simplicity when compared to alternative solutions in software. See more benefices listed below.

- It is possible to consider on-off states for cells, which is a tough task in software, because of the numerical nature of EDO discretization.
- Cell's mode is easily handled, requiring only to apply external signals to control it. In contrast, EDO-based solutions need more effort and rigid conditions for consistency and convergence.
- CA systems seem to be more efficient for being involved to physical phenomena digitization. EDO solutions need more complex mathematical structures and algorithms to perform in a similar way.

On the other hand, modeling CA systems may require an appropriate degree of creativity and observation to obtain consistent results. The engineer/researcher must be able to choose, in each situation, the best approach equilibrium between realism and solution cost.

The only pointed disadvantage is regarding the necessary resources, which are highly costly. A trade-off between hardware acceleration and cost must be considered for each case application. Aiming to a far equilibrium, future steps go to the use of integrated hardware and software processors. Thus, it is expected to have DDR memories and a software processor, working to expand the CA system modeling capability.

REFERENCES

- [1] Rabia Bakhteri, Julian Cheng, and Alex Semmelhack. Design and implementation of cellular automata on FPGA for hardware acceleration. volume 171, page 9, Third International Conference on Computing and Network Communications (CoCoNet'19), 2020.
- [2] D. T. Blackstock. Fundamentals of physical acoustics. 2000.
- [3] David T. Blackstock. Fundamentals of physical acoustics. 2000.
- [4] Marcia Luciana Agueña Castro and Reinaldo de Oliveira Castro. Autômatos celulares: Implementações de Von Neumann, Conway e Wolfram. volume III, pages 89–106, Revista de Ciências Exatas e Tecnologia, 2008.
- [5] Fábio Belotti Colombo. Aplicação de autômatos celulares para simulação de processos de microfabricação. 2017.
- [6] Moura H. G. Simulação da propagação bidimensional de ondas acústicas através de uma malha de guias digitais de ondas. 2006.
- [7] Moura H. G. One-dimensional digital waveguide mesh - ftdt solution. <https://youtu.be/lkZPeOSbf7g>, Nov 2020.
- [8] Martin Gardner. The fantastic combinations of John Conway's new solitaire game "life". 1970.
- [9] Mathias Halbach, Rolf Hoffmann, and Patrick Röder. FPGA implementation of cellular automata compared to software implementation. 2004.
- [10] Donald E. Hall. Basic acoustics. 1987.
- [11] Maciej Janowicz and Arkadiusz Orłowski. Cellular automaton approach to electromagnetic wave propagation in dispersive media. 2006.
- [12] Heinrich Kuttruff. Acoustic an introduction. 2007.
- [13] Daniel M. Muñoz, Diego F. Sanchez, Carlos H. Llanos, and Mauricio Ayala-Rincón. FPGA based floating-point library for cordic algorithms. 2010.
- [14] Daniel M. Muñoz, Diego F. Sanchez, Carlos H. Llanos, and Mauricio Ayala-Rincón. Tradeoff of FPGA design of a floating-point library for arithmetic operators. volume 5, pages 42–52, Journal Integrated Circuits and Systems, 2010.
- [15] J. V. Neumann. Theory of self-reproducing automata. 1966.
- [16] R. D. Skeel and M. Berzins. A method for the spatial discretization of parabolic equations in one space variable. 1990.
- [17] J. O. Smith. Physical modeling using digital waveguides. volume 16, pages 74–91, Center for Computer Research in Music and Acoustics (CCRMA), 1992.
- [18] L. N. Trefethen. Finite difference and spectral methods for ordinary and partial differential equations. 1994.
- [19] Huerta Trujillo, J. C. Chimal-Eguía, L. Muñoz Salazar, and J. A. Martínez Nuño. Simulation of a vibrant membrane using a 2-dimensional cellular automaton. volume 2015, page 8, 2015.
- [20] C. R. Wylie. Matemáticas superiores para ingeniería. 1982.