

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Programa de Pós-Graduação em Engenharia Elétrica

Roger Duarte Costa

**PROJETO E IMPLEMENTAÇÃO DE AUTÔMATOS CELULARES
UNIDIMENSIONAIS EM FPGA**

Belo Horizonte
2018

Roger Duarte Costa

**PROJETO E IMPLEMENTAÇÃO DE AUTÔMATOS CELULARES
UNIDIMENSIONAIS EM FPGA**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Pontifícia Universidade Católica de Minas Gerais, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Carlos Augusto Paiva da Silva Martins

Área de Concentração: Sistemas de Engenharia Elétrica e de Computação

Belo Horizonte
2018

FICHA CATALOGRÁFICA

Elaborada pela Biblioteca da Pontifícia Universidade Católica de Minas Gerais

C837p Costa, Roger Duarte
Projeto e implementação de autômatos celulares unidimensionais em FPGA
/ Roger Duarte Costa. Belo Horizonte, 2018.
89 f. : il.

Orientador: Carlos Augusto Paiva da Silva Martins
Dissertação (Mestrado) – Pontifícia Universidade Católica de Minas Gerais.
Programa de Pós-Graduação em Engenharia Elétrica

1. Autômato celular - Projetos. 2. Sistemas de controle por realimentação. 3. Algoritmos paralelos. 4. VHDL (Linguagem descritiva de hardware). 5. Simulação (Computadores). 6. Teoria dos autômatos. I. Martins, Carlos Augusto Paiva da Silva. II. Pontifícia Universidade Católica de Minas Gerais. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

CDU: 621.38.032

Roger Duarte Costa

**PROJETO E IMPLEMENTAÇÃO DE AUTÔMATOS CELULARES
UNIDIMENSIONAIS EM FPGA**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Pontifícia Universidade Católica de Minas Gerais, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Área de Concentração: Sistemas de Engenharia Elétrica e de Computação

Prof. Dr. Carlos Augusto Paiva da Silva Martins (Orientador) – PUC Minas

Prof.^a Dr.^a Rose Mary de Souza Batalha – PUC Minas

Prof. Dr. Júlio Cezar David de Melo – UFMG

Prof. Dr. Pyramo Pires da Costa Júnior – PUC Minas

Belo Horizonte, 27 de março de 2018

AGRADECIMENTOS

Início os meus agradecimentos por DEUS, já que Ele colocou pessoas tão especiais ao meu lado, sem as quais certamente não teria dado conta!

Aos meus pais, Paulo e Sueli e aos meus irmãos, Ruston e Renzo, meu infinito agradecimento. Sempre acreditaram na minha capacidade e sempre me incentivaram para fazer o melhor de mim.

Ao Prof. Carlos que acreditou no meu potencial de uma forma que eu não acreditava ser capaz de corresponder. Sempre disponível e disposto a ajudar. Fez-me enxergar que existe mais que pesquisadores e resultados por trás de uma dissertação, mas vidas humanas... Você não foi somente um orientador, mas, em alguns momentos, conselheiro, confidente, pai e amigo. Você foi e é referência profissional e pessoal para o meu crescimento. Obrigado por estar ao meu lado e acreditar tanto em mim!

À Prof. Rose Mary de Souza Batalha pelo apoio durante esta caminhada e por ter aceitado ser membro da banca. Obrigado por tudo!

Ao Prof. Dr. Júlio Cezar David de Melo por ter aceitado o convite e pela disponibilidade de participar da conclusão desta etapa importante na minha vida. Muito obrigado!

A meus amigos do mestrado, pelos momentos divididos juntos, especialmente ao Pedro e ao Bernardo, que se tornaram verdadeiros amigos e tornaram mais leve o meu trabalho. Vocês merecem meu eterno agradecimento!

Às funcionárias do Colegiado de Pós-graduação, especialmente à Eliza e à Isabel, pela disponibilidade, simpatia e gentileza. Obrigado pela ajuda!

A todos que contribuíram direta ou indiretamente, para a conclusão desta pesquisa.

“Inteligência é a capacidade de se adaptar à mudança”.
Stephen Hawkin

RESUMO

Essa dissertação apresenta implementações de autômatos celulares (CAs) unidimensionais em FPGA em software com código sequencial e em hardware com código sequencial e paralelo afim de comparar para avaliar o desempenho em relação ao tempo de execução. O GCC e o Quartus II foram utilizados para avaliar esta métrica com o auxílio de tabelas e gráficos. O CA foi primeiramente implementado em software utilizando a linguagem C com regra e vizinhança fixa e duas possibilidades de tamanho. Depois, foi implementado em hardware utilizando a linguagem de descrição de hardware VHDL com a utilização da instrução sequencial process e sem a utilização desta instrução também com regra e vizinhança fixa e duas possibilidades de tamanho. Os resultados mostram que a alteração no tamanho do autômato celular pode aumentar ou diminuir o desempenho absoluto ou relativo em hardware em relação ao software na avaliação do tempo de execução em função da definição da aplicação. O CA implementado em hardware sem a utilização do comando process foi capaz de ter melhor desempenho em relação ao tempo de execução em relação as outras implementações.

Palavras-chave: FPGA, Autômato Celular, VHDL.

ABSTRACT

This dissertation presents implementations of cellular automata (CAs) in conjunction with sequential code and in hardware with sequential and parallel code to compare performance against execution time. GCC and Quartus II were used to calculate the metric with the aid of tables and graphs. The CA was implemented in software using a C language as a rule and fixed neighborhood and two possibilities of size. It was then implemented in hardware using a VHDL hardware description language using the sequential instruction and seminar process. The results show that the change in the size of the automaton can increase or decrease the absolute or hardware-related performance in relation to the software in evaluating the execution time of the execution function. The CA implemented in hardware without the use of the process was laptop of the performance in relation to the execution of the interface for the other implementations.

Keywords: FPGA, Cellular Automata, VHDL

LISTA DE FIGURAS

Figura 1: Comparação de desempenho dos três algoritmos paralelos em função do número de processadores.....	19
Figura 2: Resultado experimental com quatro tamanhos diferentes de grade	18
Figura 3: Autômato Lattice Gas.....	18
Figura 4: Diagrama em blocos do CAM proposto.....	25
Figura 5: Diagrama em blocos do CAM proposto.....	26
Figura 6: Estrutura da célula do CAM proposto.....	26
Figura 7: Evolução do CAM ($x = 1$; $y = 1$ e $z = 2$) com a Regra 30 implementado em FPGA para 512 passos de tempo	27
Figura 8: Código C do CA	30
Figura 9: Teste de verificação funcional do CA composto por 8 células e configurado com 2 vizinhos, regra 30 e padrão de entrada com apenas a célula 4 em 1	31
Figura 10: Código C do CA	34
Figura 11: Teste de verificação funcional do CA composto por 64 células e configurado com 2 vizinhos, regra 30 e padrão de entrada com apenas a célula 4 em 1	35
Figura 12: Resultados de comparação de desempenho em software entre o CA de 8 células e o CA de 64 células	38
Figura 13: Diagrama em blocos do CA na ferramenta de projeto	39
Figura 14: Código VHDL do Registrador.....	40
Figura 15: Código VHDL da Unidade de controle	41
Figura 16: Parâmetros de síntese do CA proposto	42
Figura 17: Teste de verificação funcional pós síntese do CA composto por 8 células e configurado com 2 vizinhos, regra 30 e padrão de entrada com apenas a célula 4 em 1	43
Figura 18: Resultado do teste de verificação de desempenho do CA composto por 8 células e configurado com 2 vizinhos, regra 30 e padrão de entrada B	45
Figura 19: Resultado do teste de verificação de desempenho do CA composto por 8 células e configurado com 2 vizinhos, regra 30 e padrão de entrada E	46

Figura 20: Resultado do teste de verificação de desempenho do CA composto por 8 células e configurado com 2 vizinhos, regra 30 e padrão de entrada A	47
Figura 21: SW versus FPGA	48
Figura 22: Diagrama em blocos do autômato celular	49
Figura 23: Código VHDL do Registrador	50
Figura 24: Código VHDL da Unidade de controle	51
Figura 25: Parâmetros de síntese do CA proposto	52
Figura 26: Teste de verificação funcional pós síntese do CA composto por 64 células e configurado com 2 vizinhos, regra 30 e padrão de entrada com apenas a célula 32 em 1	53
Figura 27: Resultado do teste de verificação de desempenho do CA composto por 64 células e configurado com 2 vizinhos, regra 30 e padrão de entrada B	55
Figura 28: Resultado do teste de verificação de desempenho do CA composto por 64 células e configurado com 2 vizinhos, regra 30 e padrão de entrada A	55
Figura 29: SW versus FPGA	58
Figura 30: Resultados de comparação de desempenho em hardware entre o CA de 8 células e o CA de 64 células	59
Figura 31: Resultados de comparação de síntese em hardware entre o CA de 8 células e o CA de 64 células	60
Figura 32: Diagrama em blocos do CA na ferramenta de projeto	63
Figura 33: Código VHDL do Registrador	63
Figura 34: Código VHDL da Unidade de controle	65
Figura 35: Parâmetros de síntese do CA proposto	66
Figura 36: Teste de verificação funcional pós síntese do CA composto por 8 células e configurado com 2 vizinhos, regra 30 e padrão de entrada com apenas a célula 4 em 1	67
Figura 37: Resultado do teste de verificação de desempenho do CA composto por 8 células e configurado com 2 vizinhos, regra 30 e padrão de entrada B	70
Figura 38: Resultado do teste de verificação de desempenho do CA composto por 8 células e configurado com 2 vizinhos, regra 30 e padrão de entrada A	71
Figura 39: SW versus FPGAS	73
Figura 40: Diagrama em blocos do autômato celular	74
Figura 41: Código VHDL do Registrador	74

Figura 42: Código VHDL da Unidade de controle	76
Figura 43: Parâmetros de síntese do CA proposto	77
Figura 44: Teste de verificação funcional pós síntese do CA composto por 64 células e configurado com 2 vizinhos, regra 30 e padrão de entrada com apenas a célula 32 em 1	78
Figura 45: Resultado do teste de verificação de desempenho do CA composto por 64 células e configurado com 2 vizinhos, regra 30 e padrão de entrada B	79
Figura 46: Resultado do teste de verificação de desempenho do CA composto por 64 células e configurado com 2 vizinhos, regra 30 e padrão de entrada A	80
Figura 47: SW versus FPGAS.....	82
Figura 48: Resultados de comparação de desempenho em hardware entre o CA de 8 células e o CA de 64 células deste capítulo e do Capítulo 3.....	83
Figura 49: Resultados de comparação de síntese em hardware entre o CA de 8 células e o CA de 64 células deste capítulo e do Capítulo 3.....	84

LISTA DE TABELAS

Tabela 1: Detalhes da implementação em hardware do trabalho anterior	2118
Tabela 2: Detalhes da implementação em hardware deste trabalho.....	18
Tabela 3: Detalhes da implementação em hardware deste trabalho para uma grade 10x10, 15x15 e 20x20, respectivamente	18
Tabela 4: Ganho de velocidade do modelo Lattice Gas	24
Tabela 5: Ganho de velocidade do modelo CGL.....	24
Tabela 6: Síntese dos temas dos trabalhos relacionados	27
Tabela 7: Regra 30.....	31
Tabela 8: Resultados do teste da verificação de desempenho no Raspberry Pi 3.....	32
Tabela 9: Resultados do teste da verificação de desempenho no Intel-Core-i7	33
Tabela 10: Regra 30.....	35
Tabela 11: Resultados do teste da verificação de desempenho no Raspberry Pi 3 ..	36
Tabela 12: Resultados do teste da verificação de desempenho no Intel-Core-i7.....	37
Tabela 13: Regra 30.....	42
Tabela 14: Resultado dos testes do CA proposto considerando todos os padrões de entrada apresentados	47
Tabela 15: Resultado dos testes do CA proposto considerando todos os padrões de entrada apresentados	57
Tabela 16: Regra 30.....	67
Tabela 17: Resultado dos testes do CA proposto considerando todos os padrões de entrada apresentados	72
Tabela 18: Resultado dos testes do CA proposto considerando todos os padrões de entrada apresentados	81

LISTA DE SIGLAS E ABREVIATURAS

CA	Cellular Automaton (singular), Cellular Automata (plural)
CAM	Cellular Automaton with Memory
FPGA	Field Programmable Gate Array
LUT	Look Up Table
PRNG	Pseudo-Random Number Generator

SUMÁRIO

1. INTRODUÇÃO	14
1.1. Contexto.....	14
1.2. Objetivos.....	16
1.3. Contribuição.....	16
1.4. Justificativa.....	16
1.5. Estrutura do trabalho.....	17
2. TRABALHOS RELACIONADOS	18
2.1. Autômato Celular.....	18
2.1.1. Conway's game of life.....	19
2.2. Aceleradores em <i>FPGA</i>	20
2.3. Autômato Celular em <i>FPGA</i>	22
2.4. Considerações finais.....	27
3. IMPLEMENTAÇÕES DO AUTÔMATO CELULAR EM SOFTWARE E EM HARDWARE	29
3.1. Implementações em <i>software</i>	29
3.1.1. Autômato celular unidimensional de 8 bits	29
3.1.1.1. Resultados da Verificação Funcional.....	31
3.1.1.2. Implementação e execução no Raspberry Pi 3.....	32
3.1.1.3. Implementação e execução no Intel-Core-i7.....	33
3.1.2. Autômato celular unidimensional de 64 bits.....	33
3.1.2.1. Resultados da Verificação Funcional.....	35
3.1.2.2. Implementação e execução no Raspberry Pi 3.....	36
3.1.2.3. Implementação e execução no Intel-Core-i7.....	37
3.1.3. Discussão SW	37
3.2. Implementações em <i>hardware</i>	38
3.2.1. Autômato celular unidimensional com 8 células	39
3.2.1.1. Configuração em blocos do CA na ferramenta de projeto.....	39
3.2.1.2. Registrador.....	39
3.2.1.3. Unidade de controle.....	40
3.2.1.4. Síntese.....	41
3.2.1.5. Resultados da verificação funcional.....	42
3.2.1.6. Resultados da verificação de desempenho.....	44
3.2.2. Autômato celular unidimensional com 64 células	49
3.2.2.1. Configuração em blocos do CA na ferramenta de projeto.....	49
3.2.2.2. Registrador.....	49
3.2.2.3. Unidade de controle.....	50
3.2.2.4. Síntese.....	51
3.2.2.5. Resultados da verificação funcional.....	52
3.2.2.6. Resultados da verificação de desempenho.....	53
3.2.3. Discussão HW	58
3.3. Considerações finais.....	60
4. EXPLORAÇÃO DO ESPAÇO DE PARÂMETROS DO CA	62
4.1. Implementações em <i>hardware</i>	62
4.1.1. Autômato celular unidimensional com 8 células.....	62
4.1.1.1. Configuração em blocos do CA na ferramenta de projeto.....	62

4.1.1.2. Registrador	63
4.1.1.3. Unidade de controle.....	64
4.1.1.4. Síntese	65
4.1.1.5. Resultados da verificação funcional	66
4.1.1.6. Resultados da verificação de desempenho	68
4.1.2. Autômato celular unidimensional com 64 células.....	73
4.1.2.1. Configuração em blocos do CA na ferramenta de projeto	73
4.1.2.2. Registrador	74
4.1.2.3. Unidade de controle.....	74
4.1.2.4. Síntese	77
4.1.2.5. Resultados da verificação funcional	77
4.1.2.6. Resultados da verificação de desempenho	78
4.1.3. Discussão HW.....	82
4.2. Considerações finais	85
5. CONCLUSÃO	86
REFERÊNCIAS.....	88