

Projeto Lógico Automatizado de Sistemas  
Digitais Seqüenciais

Ney Laert Vilar Calazans

5 de Junho de 1998



---

A meus pais, Paulo e  
Mariinha

“Nada ou quase nada existe em nossa educação, que desenvolva no nosso estudante o gosto da pesquisa, da constatação, da revisão dos ‘achados’ – o que implicaria no desenvolvimento da consciência transitivo-crítica. Pelo contrário, a sua perigosa superposição à realidade intensifica no nosso estudante a sua consciência ingênua.

A própria posição da escola, de modo geral acalentada ela mesma pela sonoridade da palavra, pela memorização dos trechos, pela desvinculação da realidade, pela tendência a reduzir os meios de aprendizagem às formas meramente notionais, já é uma posição caracteristicamente ingênua.

Cada vez mais nos convencemos, aliás, de se encontrarem na nossa inexperiência democrática, as raízes deste nosso gosto pela palavra oca. Do verbo. Da ênfase nos discursos. Do torneio da frase. É que toda esta manifestação oratória, quase sempre sem profundidade, revela, antes de tudo, uma atitude mental. Revela ausência de permeabilidade característica da consciência crítica. E é precisamente a criticidade a nota fundamental da mentalidade democrática.”

Paulo Freire, *Educação como Prática da Liberdade*, Paz e Terra, 13<sup>a</sup> Ed, 1982. pp. 94–95.



# Conteúdo

<b>Prefácio</b>	<b>vii</b>
Gênese do Livro . . . . .	ix
Público Alvo . . . . .	x
Impacto sobre Alunos de Graduação e Pós-graduação . . . . .	x
Impacto sobre Profissionais e Pesquisadores . . . . .	x
Tratamento do Tema . . . . .	xi
Justificativa de Inclusão na Escola de Computação . . . . .	xii
Pré-conhecimento Requerido . . . . .	xiii
Técnica, Tecnologia, Método, Metodologia . . . . .	xiv
Estrutura do Volume e Roteiro de Leitura . . . . .	xiv
Relação com a Bibliografia da Área . . . . .	xvi
Contato com o Autor . . . . .	xvii
<b>Agradecimentos</b>	<b>xix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Definição de Sistemas Digitais . . . . .	1
1.2 Plano do Capítulo . . . . .	4
1.3 Projeto e Fabricação de Sistemas Digitais . . . . .	4
1.4 Sistemas Combinacionais e Seqüenciais . . . . .	7
1.4.1 Sistemas Digitais e Armazenamento de Informação . . . . .	8
1.4.2 Sistemas Digitais e o Conceito de Estado . . . . .	10
1.4.3 Definição de Sistemas Combinacionais e Seqüenciais . . . . .	12
1.5 Taxonomia de Sistemas Digitais . . . . .	12
1.6 O Processo de Projeto de Sistemas Digitais . . . . .	17
1.6.1 Uma Palavra sobre Modelos . . . . .	18
1.6.2 Modelos para o Processo de Projeto . . . . .	18
1.6.3 Projeto, Otimização e Complexidade . . . . .	28
1.6.4 Fases do Projeto de Sistemas Digitais . . . . .	30
1.7 Projeto Auxiliado por Computador . . . . .	32
1.7.1 Estrutura Geral de Sistemas de CAD Eletrônico . . . . .	32

1.7.2	CAD Acadêmico versus CAD Comercial . . . . .	33
1.8	Escopo da Obra . . . . .	36
1.9	Conclusões do Capítulo . . . . .	38
1.10	Perspectiva Histórica e Bibliografia . . . . .	38
1.11	Resumo do Capítulo . . . . .	41
<b>2</b>	<b>Fundamentação Teórica</b>	<b>43</b>
2.1	Estruturas Algébricas, Grafos e Funções . . . . .	44
2.1.1	Relações Binárias . . . . .	46
2.1.2	Grafos . . . . .	52
2.1.3	Reticulados e Álgebras Booleanas . . . . .	57
2.1.4	Funções discretas . . . . .	59
2.2	Representações de Funções Discretas . . . . .	64
2.2.1	Representações Cúbicas de Funções Discretas . . . . .	70
2.2.2	Diagramas de Decisão Binária . . . . .	80
2.2.3	Formas Implícitas de Representação . . . . .	83
2.3	Análise de Complexidade de Algoritmos . . . . .	84
2.3.1	Crescimento Assintótico de Funções . . . . .	86
2.3.2	Classes de Complexidade de Problemas . . . . .	91
2.4	Problemas Formais . . . . .	94
2.4.1	Problemas de Grafos . . . . .	96
2.4.2	Problemas de Álgebra Booleana . . . . .	98
2.5	Conclusões do Capítulo . . . . .	99
2.6	Perspectiva Histórica e Bibliografia . . . . .	99
2.7	Resumo do Capítulo . . . . .	101
<b>3</b>	<b>Representação Seqüencial</b>	<b>103</b>
3.1	Um Estudo de Caso de Sistema Seqüencial . . . . .	104
3.2	Modelos Seqüenciais . . . . .	107
3.2.1	Modelos Para o Comportamento Interno . . . . .	109
3.2.2	Modelo Geral . . . . .	113
3.3	Representações Síncronas . . . . .	117
3.4	Representações Assíncronas . . . . .	122
3.4.1	Estados em Sistemas Digitais Assíncronos . . . . .	123
3.4.2	AFTs e ASTGs . . . . .	124
3.4.3	Uma Implementação Assíncrona . . . . .	128
3.5	Conclusões do Capítulo . . . . .	131
3.6	Perspectiva Histórica e Bibliografia . . . . .	131
3.7	Resumo do Capítulo . . . . .	132

<b>4</b>	<b>Síntese Síncrona</b>	<b>133</b>
4.1	Minimização de Estados . . . . .	136
4.1.1	Definição de Minimização de Estados . . . . .	136
4.1.2	Enunciado do Problema de Minimização de Estados . . . . .	140
4.1.3	Restrições de Compatibilidade e Incompatibilidade . . . . .	141
4.1.4	Restrições de Cobertura . . . . .	142
4.1.5	Restrições de Fechamento . . . . .	143
4.1.6	Geração das Restrições SM . . . . .	144
4.1.7	Complexidade da Geração de Restrições SM . . . . .	148
4.1.8	Algoritmo de Minimização de Estados para ISFSMs . . . . .	150
4.2	Problemas de Codificação . . . . .	156
4.2.1	Os Problemas de Codificação em FSMs . . . . .	157
4.2.2	Restrições de Entrada . . . . .	165
4.2.3	Restrições de Saída . . . . .	174
4.2.4	Complexidade de Geração de Restrições SA . . . . .	178
4.3	Pseudo-Dicotomias . . . . .	179
4.3.1	A Definição de Pseudo-Dicotomia . . . . .	180
4.3.2	Pseudo-Dicotomias e Codificação . . . . .	183
4.4	Pseudo-Dicotomias e Restrições . . . . .	186
4.4.1	Representando Restrições com Pseudo-dicotomias . . . . .	187
4.4.2	O Arcabouço de Pseudo-Dicotomias . . . . .	191
4.5	Problema Codificação Booleana Restrita . . . . .	196
4.5.1	Codificação Booleana Restrita - Enunciado . . . . .	196
4.5.2	O Problema SM/SA Dois Níveis - Enunciado . . . . .	201
4.5.3	Soluções para Problemas de Codificação Restrita . . . . .	203
4.6	O Método ASSTUCE . . . . .	205
4.6.1	A Primeira Iteração . . . . .	206
4.6.2	As Iterações Subseqüentes . . . . .	211
4.6.3	Discussão do Método ASSTUCE . . . . .	212
4.7	Conclusões do Capítulo . . . . .	213
4.8	Perspectiva Histórica e Bibliografia . . . . .	214
4.9	Resumo do Capítulo . . . . .	215
<b>5</b>	<b>Síntese Assíncrona</b>	<b>217</b>
5.1	Síncronos: Vantagens e Inconvenientes . . . . .	219
5.2	Assíncronos: Modelos e Problemas . . . . .	224
5.2.1	Modelagem da Representação de Informação . . . . .	225
5.2.2	Modelagem do Ambiente . . . . .	231
5.2.3	Modelagem de Fenômenos Temporais . . . . .	234
5.2.4	Problemas Temporais . . . . .	239
5.3	Taxonomia Assíncronas . . . . .	240
5.3.1	Uma Proposta de Critérios e Classificações . . . . .	243
5.4	Caracterização de Transitórios . . . . .	246

5.5	Um Estilo de Projeto Assíncrono Moderno . . . . .	248
5.5.1	Pressupostos do Estilo Micropipeline . . . . .	249
5.5.2	Implementação de Controle em Micropipelines . . . . .	252
5.5.3	Estrutura de Sistemas Digitais Micropipeline . . . . .	253
5.6	Conclusões do Capítulo . . . . .	255
5.7	Perspectiva Histórica e Bibliografia . . . . .	256
5.8	Resumo do Capítulo . . . . .	257
<b>6</b>	<b>Análise Seqüencial</b>	<b>259</b>
6.1	Modelo de Análise de Sistemas Digitais . . . . .	261
6.1.1	Análise Abstrata . . . . .	261
6.1.2	Análise Concreta . . . . .	267
6.1.3	Relação Entre as Classes de Tarefas de Análise . . . . .	271
6.2	Verificação de Inclusão de Linguagem . . . . .	273
6.2.1	Autômatos $\omega$ -Regulares . . . . .	273
6.2.2	Verificação Formal com Autômatos-L . . . . .	277
6.2.3	Inclusão de Linguagem $\omega$ -Regular . . . . .	280
6.3	Conclusões do Capítulo . . . . .	286
6.4	Perspectiva Histórica e Bibliografia . . . . .	287
6.5	Resumo do Capítulo . . . . .	289
	<b>Bibliografia</b>	<b>291</b>
	<b>Índice Remissivo</b>	<b>309</b>

# Lista de Figuras

1.1	Modelo genérico de sistema digital. . . . .	2
1.2	Engenharia de Sistemas VLSI. . . . .	5
1.3	Porta lógica E. . . . .	7
1.4	Três circuitos que transitoriamente armazenam informação. . . . .	8
1.5	Circuito combinacional com laço de realimentação. . . . .	11
1.6	Uma taxonomia para ICs. . . . .	16
1.7	O modelo de Suzim. . . . .	20
1.8	O diagrama Y ou modelo de Gajski-Kuhn. . . . .	24
1.9	Exemplo de processo de projeto representado com o diagrama Y. . . . .	26
1.10	O modelo de Medland para o processo de projeto. . . . .	27
1.11	O modelo de Medland como um processo de projeto iterativo. . . . .	29
1.12	Fases do projeto de sistemas digitais e o diagrama Y. . . . .	31
1.13	Estrutura geral de um sistema de CAD eletrônico. . . . .	33
2.1	Três diagramas de Hasse para o Exemplo 2.3. . . . .	52
2.2	Exemplos de grafos. . . . .	54
2.3	Diagrama de Hasse para o Exemplo 2.6. . . . .	71
2.4	Esquemas simbólico e cubo posicional para funções discretas. . . . .	75
2.5	Comportamento de funções usando esquemas mistos. . . . .	79
2.6	Três BDDs representando a função $f(a, b, c) = a \vee (b \wedge \bar{c})$ . . . . .	81
2.7	Um meta-produto de uma disjunção de cubos. . . . .	87
2.8	Exemplo de limites assintóticos para uma função. . . . .	88
2.9	Relação entre as classes de complexidade de problemas. . . . .	95
3.1	Modelos para sistemas de transição de estados finitos. . . . .	110
3.2	Diagrama de blocos do controle de tráfego. . . . .	116
3.3	FSM do controle de tráfego. . . . .	117
3.4	Representação de FSMs síncrona e assíncronas. . . . .	118
3.5	Especificação síncrona de problema. . . . .	121
3.6	Implementação síncrona de um sistema digital. . . . .	122
3.7	Exemplos de diferentes tipos de AFT. . . . .	125
3.8	ASTG e a rede de Petri subjacente. . . . .	127

4.1	Tabelas de compatibilidade inicial e final. . . . .	146
4.2	Grafos de junção e de compatibilidade. . . . .	147
4.3	Algoritmo de compatíveis primos. . . . .	153
4.4	Restrições de entrada em FSMs. . . . .	168
4.5	Minimização de tabelas de cubos. . . . .	172
4.6	Codificações e minimização de tabelas de cubos. . . . .	173
4.7	Tabelas de cubos e restrições de dominância. . . . .	175
4.8	Tabelas de cubos e restrições disjuntivas. . . . .	177
4.9	Codificação de estados e PLA final para FSM LION9. . . . .	186
4.10	STT e restrições de entrada para a FSM $\mathcal{A}$ . . . . .	199
5.1	Um diagrama de blocos típico de um sistema síncrono. . . . .	220
5.2	Tempo de hold e setup: definição e violação. . . . .	221
5.3	Escorregamento do sinal de relógio. . . . .	223
5.4	Portas lógicas trilha simples e trilha dupla. . . . .	226
5.5	Porta lógica XOR, o OU da lógica de eventos. . . . .	229
5.6	Protocolos de comunicação assíncronos. . . . .	230
5.7	Diagrama de tempos para um modelo de inversor com atraso. . . . .	235
5.8	Modelo físico de um inversor. . . . .	236
5.9	Comparação de atrasos inerciais inteiros binário e ternário. . . . .	237
5.10	Modelos de distribuição de atrasos em circuitos assíncronos. . . . .	238
5.11	Exemplo de corrida. . . . .	240
5.12	Exemplo de transitório. . . . .	241
5.13	Elemento-C, o E da lógica de eventos. . . . .	248
5.14	Outros operadores da lógica de eventos. . . . .	250
5.15	Elemento de memória controlado por eventos. . . . .	251
5.16	A estrutura de controle básica em micropipelines. . . . .	253
5.17	Estrutura geral de uma implementação micropipeline. . . . .	254
6.1	Hierarquia de atividades de análise. . . . .	260
6.2	Simulação e verificação comparadas. . . . .	262
6.3	Arquitetura de um sistema de emulação. . . . .	270
6.4	Relação entre diferentes tipos de tarefas de análise. . . . .	272
6.5	Exemplo de autômato $\omega$ -regular. . . . .	274
6.6	Primeiro exemplo de autômato-L. . . . .	275
6.7	Autômato-L para infinitas ocorrências de “Bach”. . . . .	277
6.8	Ilustração das operações $\lim(\mathcal{L})$ e $\mathcal{L}^\omega$ . . . . .	278
6.9	Autômatos que aceitam $\lim(\mathcal{L})$ e $\mathcal{L}^\omega$ . . . . .	280
6.10	Um exemplo de autômato produto. . . . .	282
6.11	Procedimento INCLUSÃO_DE_LINGUAGEM. . . . .	283
6.12	Ilustração de não-inclusão no conjunto ciclo. . . . .	284
6.13	Teste de inclusão de linguagem sobre autômato produto. . . . .	286

# Lista de Tabelas

2.1	Tabela-verdade para o Exemplo 2.7. . . . .	73
2.2	Três tabelas cúbicas para o Exemplo 2.7. . . . .	74
2.3	Conjunção componente a componente de cubos. . . . .	77
2.4	Operação supercubo componente a componente. . . . .	78
2.5	Valores para crescimento assintótico de funções. . . . .	90
2.6	Tempo de execução de algoritmos em máquina de 100 MIPS. . . . .	91
3.1	Exemplos de expressões e linguagens regulares. . . . .	113
3.2	AFT parcial, sem tráfico na Picada do Mato. . . . .	128
3.3	AFT primitiva final para cruzamento da BR 2000. . . . .	129
3.4	AFT final reduzida para cruzamento da BR 2000. . . . .	131
4.1	Tabela de transição de estados. . . . .	145
4.2	Compatíveis primos para o exemplo 4.1. . . . .	154
4.3	Tabela de transição de estados minimizada. . . . .	157
4.4	Tabela de transição da FSM BEECOUNT. . . . .	167
4.5	STT para a máquina LION9. . . . .	184
4.6	Regras para definir os elementos $e_{ij}$ da matriz avaliação $E$ . . . . .	208
5.1	Codificação dupla trilha com conjuntos de dados alternantes. . . . .	227
6.1	Métodos de prova empregados em verificação formal. . . . .	266



# Prefácio

## Gênese do Livro

Esta obra, como tantas outras, surgiu da insatisfação, bem como das necessidades geradas por esta. A insatisfação derivou entre outras razões, da baixa disponibilidade de livros em português sobre o tema projeto de sistemas digitais. Entenda-se aqui que se trata da disponibilidade de livros que contemplem o estado da arte no assunto.

Quando se ministra cursos a nível de graduação e pós-graduação, raramente o docente se contenta em seguir passo a passo um livro-texto por melhor que este seja, especialmente se o assunto coincide com suas áreas de atuação em pesquisa. Ele acaba por complementar o curso com outras obras, artigos, exercícios ou sua visão pessoal do tema. A inexistência de uma obra que atenda a visão pessoal do assunto pelo autor constituiu um segundo motivo de insatisfação.

Assuntos como projeto de sistemas digitais apresentam um grau de dinamismo tamanho que pouco tempo se passa antes de uma técnica de projeto recém proposta ser melhorada, ou mesmo superada por outra mais rápida, mais econômica ou ambos. Como tal, a panóplia de técnicas disponíveis supera amplamente o que se pode definir, analisar e comparar em uma única obra. Assim, esta obra, como qualquer outra da área, não pode alegar cobrir todas as técnicas de projeto, ou mesmo um conjunto abrangente destas. O livro presente é uma visão particular da área, com esperança de constituir um panorama significativo do estado da arte em projeto de sistemas digitais seqüenciais.

A disciplina de projeto lógico encontra-se hoje firmemente estabelecida em currículos de graduação no contexto nacional, seja de cursos de Ciência da Computação e Informática, seja de cursos de Engenharia Elétrica e da Computação. Contudo, a ênfase é, ainda hoje, freqüentemente colocada no aprendizado de técnicas manuais de projeto, insuficientes e, no mais das vezes, inadequadas para servir de base à discussão dos problemas de projeto de sistemas VLSI. Visto que estes últimos são o bloco construtivo básico de sistemas computacionais modernos, e que de sua compreensão depende a melhor exploração dos recursos

destes sistemas, vale a pena salientar aqui a necessidade de mudança na forma de abordar o processo de projeto de sistemas digitais. As técnicas de projeto lógico automatizado possuem uma longa história. Por exemplo, métodos de otimização combinacional e seqüencial automatizados foram propostos desde a década de 50 [132, 150]. O advento de sistemas VLSI fomentou o uso amplo e a evolução de tais métodos apenas a partir de meados da década de 80, quando a eficiência computacional dos mesmos aumentou sobremaneira [22, 23, 62, 69]. Mesmo em vista deste último fato, nos parece injustificado que mesmo livros-texto empregados de forma ampla em cursos de graduação, tais como as obras de Hayes e Mano [95, 127] não contemplem a discussão, nem superficial, das técnicas modernas de projeto, principalmente se considerarmos que já existem livros que endereçam o tema adequadamente, tais como o de Hill e Peterson [98].

## **Público Alvo**

### **Impacto sobre Alunos de Graduação e Pós-graduação**

Um dos objetivos do livro é expor o público alvo oriundo de cursos de graduação, talvez pela primeira vez, ao corpo de métodos e representações de projeto lógico seqüencial automatizado em uso corrente, junto a alguns desenvolvimentos recentes da área.

Do ponto de vista do público oriundo de cursos de pós-graduação, enxergamos uma carência freqüente no corpo discente novato em pesquisa na área de projeto de sistemas digitais: a pouca familiaridade com a quantificação da eficiência de métodos e formas de representação empregadas. Este livro salienta o quão imprescindível é o uso de técnicas de análise de complexidade para avaliar a qualidade dos algoritmos e estruturas de dados, ou seja, das respectivas implementações de métodos e formas de representação.

### **Impacto sobre Profissionais e Pesquisadores**

As técnicas e métodos a serem discutidos no livro, conquanto desenvolvidas primordialmente para atender necessidades específicas em projeto lógico automatizado de sistemas digitais, possuem campo de aplicação em outros domínios. Por exemplo, a formulação dada a problemas de codificação Booleana [41] no livro é geral, e já foi aplicada ao projeto físico, bem como a vários subproblemas distintos de projeto lógico combinacional e seqüencial. Diagramas de decisão binária reduzidos e ordenados (do inglês, *reduced ordered binary decision diagrams* ou ROBDDs), associados às técnicas implícitas de representação, causaram um grande impacto nos últimos anos pelo aumento, em várias ordens de grandeza, do tamanho máximo de problemas solúveis por computador.

O livro apresenta os paradigmas gerais subjacentes aos métodos e estruturas mostrados, com o intuito de fomentar no pesquisador participante do curso a investigação da viabilidade de empregar tais paradigmas de sucesso em projeto lógico a outras áreas.

Uma característica adicional, que torna a disciplina de projeto lógico bastante atraente, é a quantidade de material à disposição no domínio público. A maioria dos métodos discutidos aqui estão implementados sob a forma de programas de computador e contam com um conjunto considerável de exemplos que auxiliam na comparação entre métodos de projeto alternativos. A maioria destes programas são de origem acadêmica, mas sua qualidade viabiliza o emprego no ambiente produtivo. Por outro lado, a funcionalidade de sistemas de projeto comerciais usados na indústria pode ser complementada ou acrescida pela cuidadosa integração de ferramentas de domínio público.

No livro, será dada atenção também à discussão do quanto as técnicas apresentadas alcançaram penetração em sistemas comerciais e o quanto estes últimos se encontram defasados com relação ao estado da arte em pesquisa. O objetivo será permitir que profissionais usuários da disciplina de projeto lógico possam avaliar a necessidade e a viabilidade de incrementar sistemas de CAD comerciais através da agregação de ferramentas disponíveis.

## Tratamento do Tema

O tratamento do tema é teórico-prático. Citamos acima que as técnicas de projeto lógico automatizado são pouco difundidas a nível de cursos de graduação. Um dos motivos para justificar este fato é a base teórica exigida para compreender conceitos convoluídos da álgebra de chaveamento e da teoria de autômatos finitos. Estruturas algébricas abstratas como semi-grupos, campos e reticulados são parte integrante de vários dos modelos empregados em projeto lógico seqüencial automatizado. Representações implícitas exigem também uma dose razoável de manipulação de conceitos abstratos. Logo, o livro não pode prescindir de uma boa dose de teoria, sob pena de tornar a discussão conceitual sem fundamentação.

Por outro lado, as figuras de mérito associadas ao nível de abstração lógico são por demais concretas (área, atraso de propagação, potência dissipada, testabilidade), e contam com interpretações precisas. Por exemplo, uma boa estimativa para área é o número de literais de uma expressão de chaveamento, ou o número de vértices de um ROBDD. Esta objetividade de medidas, aliada à disponibilidade extensa de estudos de caso para validar as técnicas desenvolvidas, e ao hábito corrente de implementar as técnicas de projeto sob a forma de programas, garantem a faceta prática do tratamento do tema.

Quanto aos conteúdos, devemos mencionar que o corpo estável de conhecimentos da área de projeto de sistemas digitais ainda é bastante reduzido.

Na presente obra, optamos por investigar em detalhe um número limitado de aspectos específicos do tema. Em particular, optamos por colocar a ênfase principal em métodos de síntese seqüencial e alguma ênfase em métodos de análise, concentrando-nos em aspectos de verificação formal. Referências extensas para abordagens complementares ou distintas são incluídas, visando permitir uma exploração mais ampla do espectro de técnicas de projeto lógico seqüencial existentes.

## **Justificativa de Inclusão na Escola de Computação**

“O projeto de sistemas digitais é atribuição de engenheiros eletrônicos”. Esta frase pode ter sido uma verdade inquestionável há apenas 15 anos atrás. Hoje, pelo menos no que diz respeito a sistemas digitais VLSI, a realidade é bastante distinta. Vejamos alguns fatos da área que dão suporte a esta última afirmação.

Sistemas modernos de projeto auxiliado por computador (*Computer Aided Design* ou CAD) para sistemas digitais VLSI podem comportar de 150 a 200 programas que manipulam múltiplas bibliotecas de módulos, o todo integrado através de um banco de dados de projeto e um sistema de controle do fluxo de concepção (formando os chamados CAD *frameworks*). Tais sistemas chegam a ser comercializados por quantias que excedem US\$500.000,00. Ainda assim, nenhum destes sistemas de CAD pode ser classificado como completo. Cada um dos programas de tal sistema pode ter sido subproduto de achados de pesquisa recente, documentados em dissertações de mestrado ou teses de doutorado. Apenas a título de exemplo, podemos citar três teses de doutorado desenvolvidas nos últimos anos, todas dedicadas quase que exclusivamente ao problema de codificação Booleana aplicada ao nível lógico de abstração: o trabalho de Avedillo na Universidade de Sevilha [11], o de Villa, na Universidade de Berkeley [186], e aquele desenvolvido pelo autor na Universidade de Louvain [40]. E certamente esta lista não é exaustiva, mesmo considerando o curto período considerado e o tema restrito.

Como se pode notar, a complexidade da área de projeto VLSI requer uma estreita cooperação entre implementadores de programas de auxílio ao projeto e projetistas de “hardware”. No primeiro grupo, encontra-se pessoal com formação em eletrônica, mas também pessoal de Ciência da Computação, além de áreas limítrofes, como Engenharia da Computação. No segundo grupo, encontra-se até mesmo pessoal de Ciência da Computação, por diversos motivos, entre os quais podemos citar:

- a automatização crescente dos níveis inferiores de projeto viabiliza a abstração de grande parte dos detalhes elétricos do sistema digital alvo, habilitando ao projeto pessoal com formação em construção de algoritmos e treino com ferramentas de projeto;

- da mesma forma, o advento de componentes personalizáveis VLSI tais como os FPGAs [30], compostos de matrizes de células descritas sobretudo no nível lógico de abstração, capacitam a execução de projetos por parte de leigos em microeletrônica e eletrônica;
- o projeto no nível sistêmico, embora ainda a nível de pesquisa, atinge hoje o ponto de automatizar a implementação integrada de um sistema digital composto de software e hardware, exigindo a cooperação entre pessoal de eletrônica e de computação, seja para a produção de programas de auxílio ao projeto, seja para o adequado emprego das mesmas na obtenção de um sistema digital o mais otimizado possível;
- os níveis superiores de abstração de projeto lidam com primitivas de descrição tais como filas, pilhas e árvores, mais próximas do treinamento de pessoal de computação que daquele do pessoal de eletrônica, o que é particularmente verdade no contexto de cursos de engenharia no Brasil, em função das características destes últimos, como a existência, até bem pouco tempo, de um currículo mínimo obrigatório.

Consideramos que a discussão acima justifica ser o tema do livro, uma área de aplicação com amplo espectro de possibilidades de atuação para informatas, tanto quanto para engenheiros.

Quanto ao assunto específico do livro, projeto lógico seqüencial, trata-se de tema amplo, pois abrange um conjunto significativo de métodos de síntese, otimização e análise de sistemas digitais no nível lógico de abstração. O tema também é atual, como atesta a disponibilidade de poucas obras de alta qualidade versando sobre o assunto [90, 7, 66, 133, 99].

## Pré-conhecimento Requerido

O público alvo do livro compreende alunos de graduação e pós-graduação, bem como pesquisadores e profissionais. O pré-conhecimento técnico esperado para abordar os conteúdos não é extenso.

O programa de uma primeira disciplina de graduação versando sobre lógica digital, tal como previsto no currículo de referência da Sociedade Brasileira de Computação (SBC), é suficiente para cobrir os pré-requisitos relacionados a hardware. Com relação a algoritmos e estruturas de dados, o livro pressupõe alguma exposição prévia ao tema, a nível introdutório. Ou seja, espera-se que os leitores sejam versados em desenvolvimento de algoritmos e conheçam estruturas de dados elementares tais como pilhas, filas, árvores e listas e algum contato com uma linguagem de programação procedural tal como C ou PASCAL. A familiaridade com princípios de análise de complexidade de algoritmos, tais

como crescimento assintótico de funções e noções de NP-completude [55], auxilia a leitura, mas não constitui um pré-requisito, visto que uma introdução está contida no livro. Quanto ao pré-conhecimento de matemática, o único fundamental assumido será o de teoria de conjuntos elementar. Contato anterior com estruturas algébricas, tipicamente reticulados e grupos, são úteis para colocar os conceitos mais complexos em uma perspectiva formal, mas nada disto será exigido para uma leitura completa da obra. Unicamente para o tema de verificação formal, requer-se que o leitor conte com conhecimentos rudimentares de lógica teórica, embora uma curta introdução seja provida no livro.

## Técnica, Tecnologia, Método, Metodologia

É freqüente a confusão ao interpretar e empregar os termos *técnica*, *tecnologia*, *método* e *metodologia*. Vale a pena procurar estabelecer um sentido preciso para os mesmos, antes de empregá-los. Em seu trabalho sobre elaboração de métodos de projeto de sistemas [56], Rocha Costa define *técnica* como “um conjunto de operações muito bem caracterizadas, as quais devem ser realizadas em uma determinada ordem, sobre determinados objetos, para que se possa obter, com esses objetos, um determinado resultado (novos objetos ou transformações dos objetos já existentes)”. O mesmo autor define *método* como “um conjunto de técnicas organizadas para a obtenção de determinado resultado, quando da manipulação de determinados objetos”. O importante destas definições, mais que sua não-ambigüidade é o estabelecimento de uma relação de hierarquia entre técnica e método, pela expressão do segundo termo a partir do primeiro. Como exemplo prático, podemos ver que um algoritmo de classificação tal como “quicksort” é a implementação de uma técnica segundo a definição, enquanto que os mecanismos que levam da especificação de um controlador como uma máquina de estados finita até sua implementação usando portas lógicas e biestáveis do tipo JK, constitui um método, em vista da composição de técnicas necessárias (e.g., minimização e codificação de estados, geração de equações de excitação para cada uma das entradas dos biestáveis).

Aurélio dá definições adequadas para os termos restantes [75]. Tecnologia é o “conjunto de conhecimentos, especialmente princípios científicos que se aplicam a um determinado ramo de atividade” e metodologia, o “estudo dos métodos e, especialmente, dos métodos das ciências”. Nota-se aí a justificativa para parte da confusão. Não há a mesma relação entre os dois últimos termos que entre os dois primeiros, apesar da semelhança etimológica das raízes. Como exemplo, o trabalho de Rocha Costa é um trabalho de metodologia, a transmissão de dados usando o TCP/IP é uma tecnologia, da mesma forma que a implementação de circuitos integrados digitais TTL (“transistor-transistor logic”).

## Estrutura do Volume e Roteiro de Leitura

Os conteúdos do livro estão distribuídos em seis capítulos. Os três primeiros capítulos apresentam a base necessária para o desenvolvimento do tema da obra nos três capítulos finais. O Capítulo 1 inicia com a definição do tema e o posicionamento deste com relação ao processo de produção moderno de sistemas digitais. Neste, introduz-se definições para sistemas combinacionais e seqüenciais, apresenta-se uma classificação geral de sistemas digitais e estuda-se um conjunto de modelos que podem ser empregados para racionalizar e representar o processo de projeto de sistemas digitais. O mesmo Capítulo ainda sugere uma visão moderna das fases de projeto de sistemas digitais complexos, caracterizando a disciplina de projeto lógico como central à discussão subsequente. O Capítulo conclui discutindo o emprego de sistemas de projeto auxiliado por computador na concepção de sistemas digitais, em vista das fases de projeto caracterizadas.

Apresentar a fundamentação teórica do tema do livro é papel do Capítulo 2. Grafos, estruturas algébricas e funções são os principais fundamentos matemáticos apresentados. A seguir, discute-se em detalhe um conjunto de formas de representação de funções discretas, adequado para manipular representações lógicas de sistemas digitais. Estuda-se então alguns dos problemas teóricos fundamentais, aos quais se pode reduzir os problemas de síntese e análise de sistemas digitais seqüenciais. Finalmente, introduz-se os princípios e as notações a empregar para a análise dos algoritmos.

A primeira parte encerra-se ao caracterizar as principais formas de representação de sistemas seqüenciais, no Capítulo 3, com base nas formas de representação de funções discretas do Capítulo 2.

A segunda parte da obra particiona os temas segundo dois conceitos ortogonais: tipos de atividades do projeto seqüencial e formas de implementação de sistemas seqüenciais. Os Capítulos 4 e 5 tratam das atividades de síntese e otimização, enquanto que o Capítulo 6 lida com a fase de análise. A forma de implementação síncrona é assunto do Capítulo 4, enquanto que sistemas assíncronos são tratados no Capítulo 5. As duas formas de implementação são tratadas em conjunto para as atividades de análise, no Capítulo 6.

A síntese e otimização de sistemas digitais síncronos é sem dúvida uma das áreas mais desenvolvidas em pesquisa e desenvolvimento na atualidade. Em conseqüência, seria muito difícil em uma obra de médio porte como esta ser suficientemente abrangente a ponto de cobrir o assunto de maneira geral apenas no Capítulo 4. Opta-se então por explorar em detalhe um arcabouço teórico particular para o qual muitos problemas podem ser mapeados. O arcabouço denomina-se Codificação Booleana Restrita, ou BCE (do inglês, *Boolean Constrained Encoding*) [40]. Este é o mais longo dos capítulos do livro, e junto com o Capítulo 2 o que requer maior fluência teórica para sua compreensão.

Ao contrário do Capítulo 2, os dois últimos são escritos sob a forma de re-

visões abrangentes de modelos e do estado da arte nos temas sistemas assíncronos e análise de sistemas digitais seqüenciais, respectivamente. Além disto, os Capítulos 5 e 6 têm em comum com o Capítulo 2 o fato de cada um deles discutir em detalhe dois métodos de solução de problemas típicos da área respectiva e ampla quantidade de referências onde o leitor pode abordar métodos adicionais em detalhe.

Cada um dos seis capítulos possui três Seções finais comuns a todos: Conclusões, Perspectiva Histórica e Bibliografia, Resumo.

Os Capítulos 4, 5 e 6 são em tese independentes e podem assim ser lidos em qualquer ordem. Para leitores a nível de graduação, os Capítulos 1 e 3 são pré-requisitos para a leitura de qualquer dos Capítulos da segunda parte do livro, e os conteúdos do Capítulo 2 são fundamentais para o Capítulo 4. Ainda considerando público de graduação, o Capítulo 3 necessita de alguns conhecimentos teóricos desenvolvidos no Capítulo 2, mas o leitor com alguma base em álgebra Booleana pode evitar este último em uma primeira leitura, usando seus conhecimentos para intuitivamente apreender os conceitos do Capítulo 3. Leitores no nível de pós-graduação, pesquisadores e profissionais, se experientes em sistemas digitais, podem iniciar a leitura diretamente na segunda parte do Livro, consultando os Capítulos 2 e 3 sempre que necessário o esclarecimento ou a definição precisa de conceitos. Contudo, aconselha-se a estes leitores avançados a leitura do Capítulo 1, para uma visão global do assunto e familiarização com os modelos e a terminologia particulares empregados na obra.

## Relação com a Bibliografia da Área

O projeto lógico combinacional foi tema de livro escrito pelo Prof. Dr. Ricardo Jacobi, intitulado *Síntese de Circuitos Lógicos Combinacionais* [110], publicado no contexto da 10<sup>a</sup> Escola de Computação, realizada em 1996, pelo DCC-UNICAMP, em Campinas. O livro proposto aqui é uma obra companheira daquela. Juntos, estes volumes constituem um apanhado amplo da disciplina de projeto lógico moderno de sistemas digitais, endereçando sistemas combinacionais e seqüenciais, sistemas síncronos e assíncronos, síntese e análise de sistemas. Um elemento não tratado em qualquer das obras, devido primordialmente a questões de espaço e volume dos textos, é o teste de sistemas digitais.

Já foi mencionado antes que o tema de projeto lógico de sistemas digitais não é abordado de forma global em qualquer obra de conhecimento do Autor. Esta inexistência é sem dúvida fruto do extenso desenvolvimento presente da área, bem como da relativa juventude das últimas descobertas no setor. Cabe também mencionar que mesmo limitando a atenção ao projeto lógico de sistemas digitais seqüenciais, não seria possível encontrar uma obra suficientemente abrangente para servir de referência única. No que concerne a síntese de sistemas síncronos, dois livros podem ser citados como tendo servido de base à

proposta: o de Hachtel e Somenzi, *Logic Synthesis and Verification Algorithms* [90] e o de Ashar, Devadas e Newton, *Sequential Logic Synthesis* [7]. Ambos livros contemplam uma ampla discussão de técnicas aplicáveis à implementação de sistemas usando lógica em dois níveis e lógica multinível, a primeira apresentando uma discussão adicional sobre verificação formal de sistemas descritos sob a forma de estruturas de transição de estados finitas.

Para a verificação de sistemas síncronos, utilizamos a primeira obra. A obra de McMillan, *Symbolic Model Checking* [133] endereça o problema de verificação formal de ambos, sistemas síncronos e assíncronos, sendo assim de interesse primordial para a presente proposta. Uma obra alternativa interessante é o livro de Ghosh, Devadas e Newton, *Sequential Logic Testing and Verification* [85], que provê uma visão comparativa de verificação e teste de sistemas seqüenciais síncronos.

Para a síntese de sistemas assíncronos, usamos quatro livros: os dois primeiros são a quarta edição de *Computer Aided Logical Design with emphasis on VLSI*, de Hill e Peterson [98] e a referência clássica de Unger, *Asynchronous Sequential Switching Circuits* [182]. O primeiro livro é limitado, pois dirige-se a público de graduação iniciante, e o segundo, malgrado a qualidade da abordagem, está defasado com relação aos avanços do último quarto de século. Os dois últimos livros contemplam desenvolvimentos recentes no tema: *Algorithms for Synthesis and Testing of Asynchronous Circuits*, de Lavagno e Sangiovanni-Vincentelli [117] e o mais recente *Asynchronous Circuits* de Brzozowski e Seger [35].

Além das obras mencionadas, uma quantidade de referências sobre cada um dos temas abordado no livro são citados nos Capítulos específicos.

Finalmente, é opinião do Autor que a relação entre técnicas de síntese de sistemas síncronos e assíncronos não foi devidamente explorada na bibliografia consultada. Isto ocorre mesmo que na maior parte das vezes esta relação seja bastante clara, como no caso da solução dos problemas de codificação de estados e minimização de estados [40]. Neste aspecto, a proposta presente pretende acrescentar uma visão comparativa original do projeto lógico de sistemas seqüenciais.

## Contato com o Autor

O Autor é na atualidade professor adjunto no Instituto de Informática da Pontifícia Universidade Católica do Rio Grande do Sul, PUCRS. Em 1994, ao retornar de programa de doutoramento, criou e desde então lidera o Grupo de Apoio ao Projeto de Hardware (GAPH), voltado para atividades de pesquisa em métodos e ferramentas projeto auxiliado por computador para sistemas digitais. O Autor e o grupo GAPH podem ser contactados através dos endereços

eletrônicos

**calazans@inf.pucrs.br** e **gaph@inf.pucrs.br**,

respectivamente. Maiores informações sobre o Autor e o GAPH podem ser obtidas via Internet nas páginas

**<http://www.inf.pucrs.br/~calazans>** e

**<http://www.inf.pucrs.br/~gaph>**.

# Agradecimentos

A autoria individual de um livro é estritamente falando, uma mentira. Embora cada uma das frases e idéias de tal obra tenham sido fruto em última análise da mente do autor, estas jamais viriam à luz do dia sem o concurso de uma equipe imensa de colaboradores, seja na formação das idéias, na habilitação do desenvolvimento do trabalho intelectual, na provisão de meios físicos para que o trabalho se realize, ou no apoio moral e afetivo. Abaixo, procuro fazer jus à contribuição desta equipe.

Quanto à formação de idéias, gostaria de citar dois nomes de significância: o Prof. Dr. Altamiro Susin (anteriormente, Suzim) e o Prof. Dr. Marc Davio. O primeiro introduziu-me ao tema sistemas digitais e influenciou-me na formação dos conceitos do que seja pesquisa, sua importância e sua ética. O segundo, meu orientador de doutorado (*in memoriam*), fez-me descobrir, primeiro através de sua obra, e depois pessoalmente, a beleza e importância do raciocínio abstrato em qualquer campo do conhecimento, além de ser uma fonte de idéias inesgotável, mesmo após sua morte. O Prof. Susin e o Prof. Luigi Carro do Depto. de Engenharia Elétrica da UFRGS gentilmente franquearam acesso às publicações e material não-publicado referente ao modelo proposto pelo primeiro, e discutido na Seção 1.6.2.1 deste livro.

No que diz respeito à habilitação do trabalho intelectual, tenho três reconhecimento importantes a apontar: os órgãos de fomento à pesquisa, à PUCRS, e a meus alunos. No primeiro caso, tenho a agradecer particularmente ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), à Coordenadoria de Aperfeiçoamento de Pessoal de Ensino Superior (CAPES) e à Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS). Há quase vinte anos tenho podido contar com o suporte destas instituições na realização de atividades de ensino e pesquisa, desde a iniciação científica até os projetos de pesquisa em que hoje participo, passando pelo apoio durante meus cursos de mestrado e doutorado. Segundo, a Pontifícia Universidade Católica do Rio Grande do Sul, meu local de trabalho, tem investido enorme quantidade de recursos na qualidade de ensino e pesquisa, e este esforço tem produzido um ambiente dinâmico, em constante evolução, que dá suporte ao trabalho

de produção intelectual e produção de recursos humanos altamente capacitados. Terceiro, meus alunos de graduação e pós-graduação têm provido uma motivação fundamental à busca do conhecimento. É o desafio de participar efetivamente de sua formação que me conduz à pesquisa, mais que qualquer outro incentivo. A eles, meu mais profundo respeito.

O provimento de meios físicos para realizar esta obra deve ser imputado a duas instituições: à organização da Escola de Computação de 1998, e ao Instituto de Informática da PUCRS. No primeiro caso, gostaria de agradecer aos organizadores pela oportunidade de publicar e divulgar este trabalho, e em particular aos avaliadores anônimos que revisaram minha proposta inicial e dedicaram porção preciosa de seu tempo à cuidadosa elaboração de sugestões, críticas e comentários, sugestões que espero ter atendido nesta versão final. O Instituto de Informática (II) da PUCRS foi fundamental na realização deste trabalho, provendo suporte técnico e administrativo e uma boa dose de indulgência no tratamento de meus prazos para tarefas internas, durante a escrita do presente livro. Meus agradecimentos individuais à Profa. Iára Claudio, diretora do II, que vem contribuindo há anos para projetar a Unidade e a própria PUCRS nos mais variados cenários de ensino, pesquisa e extensão. Agradeço aos colegas professores do II, que acompanharam, tão ansiosamente como eu, o desenrolar desta obra, sendo sempre uma fonte de apoio. Gostaria também de agradecer ao orientando Waldir Walter Jr., pelas contribuições críticas à escrita do Capítulo 5, bem como ao acadêmico André Duque Madeira, bolsista de iniciação científica sob minha orientação, que contribuiu para a formatação do texto final com seus conhecimentos de L<sup>A</sup>T<sub>E</sub>X.

Uma obra de fôlego como esta, exige constância e estabilidade. Agradeço a Karin, minha companheira, esposa, amiga e tanto mais, pelo amor, pelo inesgotável suporte e pela infindável compreensão. Foram tantos fins de semana, tantas horas roubadas de nosso convívio após o trabalho, que mesmo sendo ela própria professora e pesquisadora, sei como foi difícil suportar a privação de momentos de lazer que propiciei.

Devo salientar que meus pais, Mariinha e Paulo foram, em última análise, grandes responsáveis pela realização deste trabalho. Durante toda a vida, eles trabalharam com fervor quase religioso em prol da educação de seus filhos, mesmo quando os recursos para tanto eram escassos, mesmo sob as mais dolorosas situações pessoais familiares. Confesso-me incapaz de compreender a amplitude de seus esforços, restando-me apenas contemplá-los e admirá-los como exemplares únicos do gênero humano. A eles dedico esta obra.

Finalmente, se os eventuais méritos existentes neste trabalho devem ser compartilhados com todos que citei e tantos outros, os erros, falácias, e omissões só podem ser a mim imputados.

Ney Laert Vilar Calazans