

Leitfäden und Monographien der Informatik

Brauer: **Automatentheorie**

493 Seiten. Geb. DM 58,—

Becker: **Prüfen und Testen von Schaltkreisen**

In Vorbereitung

Dal Cin: **Grundlagen der systemnahen Programmierung**

221 Seiten. Kart. DM 34,—

Ehrich/Gogolla/Lipeck: **Algebraische Spezifikation abstrakter Datentypen**

In Vorbereitung

Engeler/Läuchli: **Berechnungstheorie für Informatiker**

120 Seiten. Kart. DM 24,—

Hentschke: **Grundzüge der Digitaltechnik**

247 Seiten. Kart. DM 36,—

Loeckx/Mehlhorn/Wilhelm: **Grundlagen der Programmiersprachen**

448 Seiten. Kart. DM 44,—

Mehlhorn: **Datenstrukturen und effiziente Algorithmen**

Band 1: Sortieren und Suchen

2. Aufl. 317 Seiten. Geb. DM 48,—

Band 2: Graphenalgorithmen und NP-Vollständigkeit

In Vorbereitung

Messerschmidt: **Linguistische Datenverarbeitung mit Comskee**

207 Seiten. Kart. DM 36,—

Niemann/Bunke: **Künstliche Intelligenz in Bild- und Sprachanalyse**

256 Seiten. Kart. DM 38,—

Pflug: **Stochastische Modelle in der Informatik**

272 Seiten. Kart. DM 38,—

Post: **Entwurf und Technologie hochintegrierter Schaltungen**

247 Seiten. Kart. DM 38,—

Rammig: **Systematischer Entwurf digitaler Systeme**

353 Seiten. Kart. DM 46,—

Richter: **Betriebssysteme**

2. Aufl. 303 Seiten. Kart. DM 38,—

Richter: **Prinzipien der Künstlichen Intelligenz**

359 Seiten. Kart. DM 46,—

Wirth: **Algorithmen und Datenstrukturen**

Pascal-Version

3. Aufl. 320 Seiten. Kart. DM 39,—

Wirth: **Algorithmen und Datenstrukturen mit Modula - 2**

4. Aufl. 299 Seiten. Kart. DM 39,—

Wojtkowiak: **Test und Testbarkeit digitaler Schaltungen**

226 Seiten. Kart. DM 36,—

Preisänderungen vorbehalten



B. G. Teubner Stuttgart

**Leitfäden und Monographien
der Informatik**

**Franz J. Rammig
Systematischer Entwurf
digitaler Systeme**

Leitfäden und Monographien der Informatik

Unter beratender Mitwirkung von

Prof. Dr. Hans-Jürgen Appelrath, Oldenburg
Dr. Hans-Werner Hein, St. Augustin
Prof. Dr. Rolf Pfeifer, Zürich
Dr. Johannes Retti, Wien
Prof. Dr. Michael M. Richter, Kaiserslautern

Herausgegeben von

Prof. Dr. Volker Claus, Oldenburg
Prof. Dr. Günter Hotz, Saarbrücken
Prof. Dr. Klaus Waldschmidt, Frankfurt

Die Leitfäden und Monographien behandeln Themen aus der Theoretischen, Praktischen und Technischen Informatik entsprechend dem aktuellen Stand der Wissenschaft. Besonderer Wert wird auf eine systematische und fundierte Darstellung des jeweiligen Gebietes gelegt. Die Bücher dieser Reihe sind einerseits als Grundlage und Ergänzung zu Vorlesungen der Informatik und andererseits als Standardwerke für die selbständige Einarbeitung in umfassende Themenbereiche der Informatik konzipiert. Sie sprechen vorwiegend Studierende und Lehrende in Informatik-Studiengängen an Hochschulen an, dienen aber auch in Wirtschaft, Industrie und Verwaltung tätigen Informatikern zur Fortbildung im Zuge der fortschreitenden Wissenschaft.

Systematischer Entwurf digitaler Systeme

Von der System- bis zur Gatter-Ebene

Von Prof. Dr. rer. nat. Franz J. Rammig
Universität-Gesamthochschule Paderborn

Mit zahlreichen Abbildungen und Beispielen



B. G. Teubner Stuttgart 1989

Prof. Dr. rer. nat. Franz J. Rammig

Von 1969 bis 1973 Studium der Mathematik, Wirtschaftswissenschaften und Informatik an der Universität Bonn mit Abschluß als Diplommathematiker. Anschließend Wiss. Angestellter im Fachbereich Informatik der Universität Dortmund und 1977 Promotion bei Prof. Reusch. Seit 1983 Prof. für praktische Informatik an der Universität-GH Paderborn. Von 1985 bis 1987 Mitglied des Vorstandes von Cadlab, einem von der Universität-GH Paderborn und der Nixdorf Computer AG gemeinsam getragenen Forschungsinstitut.

CIP-Titelaufnahme der Deutschen Bibliothek

Rammig, Franz J.:

Systematischer Entwurf digitaler Systeme : von der System- bis zur Gatter-Ebene / von Franz J. Rammig. – Stuttgart : Teubner, 1989

(Leitfäden und Monographien der Informatik)

ISBN 978-3-519-02265-7

ISBN 978-3-663-01461-4 (eBook)

DOI 10.1007/978-3-663-01461-4

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Das gilt besonders für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

© B. G. Teubner Stuttgart 1989

Gesamtherstellung: Zehnersche Buchdruckerei GmbH, Speyer

Umschlaggestaltung: M. Koch, Reutlingen

Vorwort

Dieses Buch versucht, eine durchgängige Systematik des Hardwareentwurfs über verschiedene Abstraktionsebenen hinweg darzustellen. Dabei wird von einem abstrakten Modell des Entwurfsvorgangs als über mehrere Abstraktionsebenen reichender rückgekoppelter Prozeß ausgegangen. Auf der Basis dieses Modells werden verschiedene Klassen von Entwurfsaktivitäten identifiziert. Es sind dies: Modellierung, Modifikation/Optimierung, Implementation und Verifikation. Die verschiedenen Abstraktionsebenen (Systemebene, algorithmische Ebene, Registertransfer-Ebene, Gatterebene, Schalterebene/ Ebene des symbolischen Layouts, elektrische/Layout-Ebene) werden in verschiedenen Sichten (Verhalten, Struktur, Geometrie, Test) charakterisiert. Dient das erste Kapitel dazu, eine allgemeine Systematik des Hardwareentwurfs zu entwickeln, so werden in den weiteren Kapiteln verschiedene Entwurfsaktivitäten beispielhaft diskutiert.

Das Kapitel 2 ist den verschiedenen Methoden der Hardwaremodellierung gewidmet. Nach einem allgemeinen Überblick wird darin exemplarisch die Breitband-Hardwarebeschreibungssprache DACAPO detaillierter eingeführt. Dies erlaubt, über verschiedene Aspekte des Hardwareentwurfs in einheitlicher Terminologie zu sprechen, und zwar nicht nur über Hardwarebeschreibungen auf unterschiedlichen Abstraktionsebenen, sondern auch über verschiedene Algorithmen des Entwurfsprozesses.

Im Kapitel 3 (Implementierungsaktivitäten) wird mit besonderem Augenmerk der Übergang von der algorithmischen auf die Registertransferebene behandelt. Aber auch verschiedene Methoden des Steuerwerksentwurfs und der Übergang auf die Gatterebene finden Berücksichtigung. Ein ausführliches Entwurfsbeispiel soll zur Illustration dienen. Optimierungsverfahren (Kapitel 4) werden hauptsächlich auf der Registertransferebene, aber auch auf der algorithmischen und Gatterebene diskutiert. Auch hier wird ein Beispiel exemplarisch durchgeführt.

Das Kapitel 5 ist der Verifikation/Evaluation/Validierung gewidmet. Trotz ihrer in Zukunft sicherlich zentralen Bedeutung wird dabei die formale Verifikation relativ knapp und nur einführend behandelt. In diesem Buch sollte eine Konzentration auf heute praktisch und in Breite einsetzbare Hilfsmittel vorgenommen werden. Dies trifft für die "Ti-

mingverifikation" sicherlich zu, die aus diesem Grund auch etwas ausführlicher behandelt wird. Der Simulation als Hilfsaktivität für eine Verifikation wird breiter Raum eingeräumt. Hier werden verschiedene Simulationskonzepte verglichen, Simulationsszenarios behandelt und auf das Problem der Mehrebenensimulation eingegangen.

Das wichtige Thema der Testverfahren wird in dem abschließenden Kapitel 6 behandelt. Hier finden nach einer einleitenden Diskussion der Testproblematik Methoden der Testmustergenerierung, der Fehler-simulation, des testfreundlichen Entwurfs und der Selbsttestverfahren Erwähnung.

Alle Kapitel sind bewußt reich mit illustrierenden Beispielen versehen, wobei nach Möglichkeit auf eine einheitliche Notation geachtet wurde. Hier erwies sich die Abstützung auf die Breitband-Hardwarebeschreibungssprache DACAPO als sehr hilfreich.

Das Buch eignet sich sowohl für Informatiker in der Praxis als auch für Studenten der Informatik an Universitäten und Technischen Hochschulen.

Große Teile des Buches sind während meines Forschungsfreisemesters, das ich bei XEROX PARC, Palo Alto verbracht habe, entstanden. Die sehr anregende Atmosphäre dieses Forschungsinstitutes und die ausgezeichneten Arbeitsmöglichkeiten dort haben dieses Buch wesentlich beeinflusst. Ich möchte mich herzlich bei meiner Frau und Herrn Kollegen Waldschmidt für die kritische Durchsicht des Manuskriptes bedanken. Mein Dank gilt ganz besonders auch Frau S. Alejandro, deren Eifer dem Buch seine endgültige Form verliehen hat.

Inhaltsverzeichnis

1 Entwurfsprozeß	11
1.1 Makroskopisches Modell des Entwurfsprozesses	11
1.2 Abstraktionsebenen	13
1.3 Mikroskopisches Modell des Entwurfsprozesses	32
1.4 Literatur	37
2 Modellierungskonzepte und Entwurfssprachen	42
2.1 Modellierungskonzepte	42
2.1.1 Objektorientierte Modellierung	42
2.1.2 Imperative Sicht	46
2.1.2.1 Zeitbehaftete Interpretierte Petri-Netze	47
2.1.2.2 Communicating Sequential Processes (CSP)	50
2.1.3 Reaktive Sicht	55
2.1.4 Stimulierte Gleichungen	57
2.1.5 Modellierungskonzepte und Abstraktionsebenen	60
2.2 Sprachkonzepte	60
2.2.1 Dedizierte Sprachen	61
2.2.1.1 Dedizierte Sprachen für die Systemebene	61
2.2.1.2 Dedizierte Sprachen für die algorithmische Ebene	64
2.2.1.3 Dedizierte Sprachen für die Registertransferebene	66
2.2.1.4 Dedizierte Sprachen für die Gatterebene	66
2.2.1.5 Dedizierte Sprachen für die Schalterebene/Symbolisches Layout	67
2.2.1.6 Dedizierte Sprachen für die Elektrische/Layout-Ebene	69
2.2.2 Sprachfamilien	71
2.2.3 Breitbandsprachen	74
2.3 Die Hardwarebeschreibungssprache DACAPO III	75
2.3.1 DACAPO III Grundlagen	75
2.3.2 Beschreibungen in DACAPO III auf der algorithmischen Ebene	86

2.3.3 Beschreibungen in DACAPO III auf der Systemebene	97
2.3.4 Beschreibungen in DACAPO III auf der Registertransferebene	118
2.3.5 Beschreibungen in DACAPO III auf der Gatter/Schaltebene	125
2.3.6 "Behavioral"-Beschreibungen in DACAPO III	133
2.4 Literatur	135
3 Implementationsaktivitäten	142
3.1 Systemebene zur algorithmischen Ebene	142
3.2 Algorithmische Ebene zur Registertransferebene	143
3.2.1 Monolithische Dekomposition	145
3.2.1.1 Ein vollständiges Beispiel zur monolithischen Dekomposition	161
3.2.2 Parallele Dekomposition	176
3.2.3 Hierarchische Steuerwerksdekomposition	177
3.3 Registertransferebene zur Gatterebene	180
3.3.1 Steuerwerksentwurf	180
3.3.1.1 Fest verdrahtete Implementierung von Steuerwerken	182
3.3.1.1.1 Implementierung in krauser Logik	185
3.3.1.1.2 Implementation durch Array-Logik	190
3.3.1.2 Mikroprogrammierte Steuerwerksimplementation	194
3.3.2 Datenpfadentwurf	199
3.4 Literatur	199
4 Optimierungsaktivitäten	205
4.1 Optimierung auf der Systemebene	205
4.2 Optimierung auf algorithmischer Ebene	206
4.2.1 Optimierung von Basisblöcken	206
4.2.2 Optimierung von Schleifen	208
4.3 Optimierung auf der Registertransferebene	209
4.3.1 Eine Heuristik zur Zustandsminimierung von Steuerwerken	210
4.3.1.1 Beispiel einer Optimierung auf RT-Ebene	220
4.4 Optimierung auf der Gatterebene	235
4.5 Literatur	252

5 Evaluierung, Validierung, Verifikation	255
5.1 Formale Verifikation	255
5.1.1 Formale Verifikation von Verhaltenseigenschaften	259
5.1.2 Verifikation des Zeitverhaltens getakteter Systeme	261
5.2 Simulation	266
5.2.1 Generierung ausführbarer Objektmodelle und deren Ausführung	268
5.2.1.1 Interne Modellierungskonzepte	268
5.2.1.1.1 Abbildung Algorithmischer Konstrukte von DACAPO III	275
5.2.1.1.2 Abbildung von DACAPO III-Konstrukten der Systemebene	277
5.2.1.1.3 Abbildung von DACAPO-Konstrukten auf Registertransferebene	280
5.2.1.1.4 Abbildung von DACAPO-Konstrukten auf Gatter/Schaltebene	280
5.2.1.2 Simulationstechniken	281
5.2.1.2.1 Streamline Code Simulation (SCS)	282
5.2.1.2.2 Äquivalente Iteration (EI)	287
5.2.1.2.3 Critical Event Scheduling (CES)	293
5.2.2 Simulationsszenarios	303
5.2.2.1 Modellierung der Umgebung	304
5.2.2.2 Ergebnisanalyse	306
5.2.3 Mehrebenensimulation	308
5.2.3.1 Multisimulatoransatz	308
5.2.3.1.1 Datenaustausch	309
5.2.3.1.2 Synchronisation	311
5.2.3.1.3 Benutzerschnittstelle	313
5.2.3.2 Breitbandsimulatoren	314
5.3 Literatur	315

6. Testmethoden	320
6.1 Begriffsbestimmungen	320
6.2 Strukturorientierte Testverfahren	321
6.2.1 Fehlermodelle	321
6.2.2 Testmustererzeugung für das Haftfehlermodell	323
6.2.3 Fehlersimulation	331
6.2.3.1 Fehlersimulation mit dem SCS-Algorithmus	331
6.2.3.2 Fehlersimulation mit dem CES-Algorithmus	333
6.2.3.2.1 Parallele Fehlersimulation	333
6.2.3.2.2 Deduktive und Concurrent-Fehlersimulation	333
6.3 Funktionsorientierte Testverfahren	335
6.4 Testfreundlicher Entwurf	336
6.4.1 Strukturelle Maßnahmen zur Erhöhung der Testbarkeit	336
6.4.2 Selbsttest	340
6.5 Literatur	347
Sachregister	351