

eXamen.press

**eXamen.press** ist eine Reihe, die Theorie und Praxis aus allen Bereichen der Informatik für die Hochschulausbildung vermittelt.

Günter Kemnitz

# Technische Informatik

Band 2: Entwurf digitaler Schaltungen

 Springer

Günter Kemnitz  
TU Clausthal  
Institut für Informatik  
Arbeitsb. Hardwareentwurf u.  
Robotik  
Julius-Albert-Strasse 4  
38678 Clausthal-Zellerfeld  
Deutschland  
kemnitz@in.tu-clausthal.de

ISSN 1614-5216  
ISBN 978-3-642-17446-9      e-ISBN 978-3-642-17447-6  
DOI 10.1007/978-3-642-17447-6  
Springer Heidelberg Dordrecht London New York

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

© Springer-Verlag Berlin Heidelberg 2011

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

*Einbandentwurf:* KuenkelLopka GmbH

Gedruckt auf säurefreiem Papier

Springer ist Teil der Fachverlagsgruppe Springer Science+Business Media ([www.springer.com](http://www.springer.com))

---

# Vorwort

Die Technische Informatik umfasst zwei ganz unterschiedliche Grundausrichtungen, die im applikativen Bereich stark miteinander verflochten sind:

- die technische Basis der Informatik und
- die technischen Anwendungen der Informatik.

Die technische Basis der Informatik ist die Digitaltechnik, die ihrerseits die Grundlage für den Entwurf hochintegrierter Schaltkreise bildet. Hochintegrierte Schaltkreise sind wiederum der Kern der Rechnerhardware.

Die andere, nicht weniger bedeutende Grundausrichtung ist die informationstechnische Erfassung und Steuerung technischer Systeme. Hier ist die Basis die Nachbildung technischer Zusammenhänge durch Zahlen, Algorithmen, Regeln etc., d.h. durch Modelle, die ein Rechner verarbeiten kann. Darauf setzen die stärker applikativ orientierten Gebiete wie die rechnergestützte Mess-, Steuer- und Regelungstechnik sowie die Nachrichtentechnik auf.

Technische Anwendungen stellen spezielle Anforderungen an die Rechnerhardware und die darauf laufende Software, nämlich

- Echtzeitfähigkeit,
- geringe Abmessungen,
- geringer Stromverbrauch,
- hinreichende Verlässlichkeit etc.

An dieser Schnittstelle haben sich wichtige Teilgebiete der Informatik herausgebildet, die sich z.B. unter den Überschriften »Eingebettete Systeme« und »Echtzeitsysteme« mit Systemen aus Hard- und Software beschäftigen, die hauptsächlich für technische Anwendungen gedacht sind. Die Einsatzgebiete solcher Systeme reichen von der Waschmaschine mit Mikrorechner über das Handy bis zum Autopiloten in einem Flugzeug.

Die beiden Grundausrichtungen der Technischen Informatik – technische Basis und technische Anwendungen – repräsentieren unterschiedliche Denkweisen. Die digitale Welt der Informatik unterscheidet nur »0« und »1«. Die

logischen Grundbausteine – UND, ODER, Inverter, ... – sind sehr einfach zu verstehen und ihre Funktionen sind exakt definiert. Aus ihnen werden hierarchisch zuerst kleine Teilsysteme, aus diesen wieder größere Teilsysteme und aus diesen wieder komplette Rechner und Rechnersysteme zusammengesetzt.

*Bei den heutigen hochintegrierten Schaltkreisen mit vielen Millionen von Transistoren gibt es keinen Menschen mehr, der sagen kann, wozu jeder der Transistoren da ist.*

Die Entwurfstechnik für wirklich große Systeme besteht inzwischen vereinfacht darin, einen Algorithmus zu entwickeln, der einen Algorithmus entwickelt, der das System entwickelt. Das ist eine vollkommen andere Denkwelt als bei der Anwendung der Informatik in der Technik.

Die informationstechnische Erfassung, Modellierung und Steuerung technischer Systeme arbeitet immer mit Näherungen. Ein technisches System verhält sich nur zu einem Teil deterministisch. Der deterministische Teil ist meist nichtlinear. Die Nachbildung erfolgt durch große Gleichungssysteme, die sich nur lösen lassen, wenn sie durch lineare Gleichungssysteme angenähert werden. Die Grundlagen hierfür sind die Physik und die Mathematik. Es ist die Genauigkeit, die es zu beherrschen gilt, nicht die Größe.

Dieses Buch behandelt die technische Basis der Informatik: die Modellbildung, die Simulation, den (rechnergestützten) Entwurf und den Test digitaler Schaltungen. Ein Rechner ist in diesem Rahmen eine Beispielschaltung.

Clausthal-Zellerfeld,  
Oktober 2010

*Günter Kemnitz*

---

# Inhaltsverzeichnis

<b>1</b>	<b>Modellbildung und Simulation</b>	<b>1</b>
1.1	Entwurf digitaler Schaltungen	1
1.1.1	Der Umgang mit großen Schaltungen	1
1.1.2	Modelle und Entwurfsraum	4
1.1.3	Entwurfsablauf	6
1.1.4	VHDL als formale Beschreibungsplattform	9
1.1.5	Zusammenfassung und Übungsaufgaben	15
1.2	Funktion, Struktur und Simulation	16
1.2.1	Signale und Datentypen	17
1.2.2	Signalflussplan	20
1.2.3	Imperative Funktionsmodelle	22
1.2.4	Ereignisgesteuerte Simulation	33
1.2.5	Strukturbeschreibung	37
1.2.6	Testrahmen	40
1.2.7	Instanziierung von Komponenten	41
1.2.8	Zusammenfassung und Übungsaufgaben	41
1.3	Laufzeittoleranz	44
1.3.1	Glitches	44
1.3.2	Das Verzögerungsmodell einer Signalzuweisung	46
1.3.3	Simulation mit Halte- und Verzögerungszeiten	48
1.3.4	Laufzeitanalyse	51
1.3.5	Zusammenfassung und Übungsaufgaben	52
1.4	Register	55
1.4.1	Register als Abtastelemente	56
1.4.2	VHDL-Abtastprozesse	57
1.4.3	Verarbeitung plus Abtastung	60
1.4.4	Register-Transfer-Funktion	61
1.4.5	Taktversatz	62
1.4.6	Zusammenfassung und Übungsaufgaben	64
1.5	Asynchrone Eingabe	66
1.5.1	Abtasten asynchroner Signale	66

1.5.2	Entprellen von Tasten . . . . .	67
1.5.3	Asynchrone Initialisierung . . . . .	68
1.5.4	Asynchrone parallele Schnittstelle . . . . .	70
1.5.5	Zusammenfassung und Übungsaufgaben . . . . .	72
1.6	Sequenzielle Schaltungen . . . . .	74
1.6.1	Endlicher Automat . . . . .	74
1.6.2	Vom Zustandsgraph zur VHDL-Beschreibung . . . . .	78
1.6.3	Unzulässige Zustände, Systemabsturz und Watchdog . . . . .	82
1.6.4	Entwurf eines Zahlenschlosses . . . . .	84
1.6.5	Operationsablaufgraph . . . . .	86
1.6.6	Beispiel Wegemessung mit Quadratur-Encoder . . . . .	89
1.6.7	Software-orientierte Ablaufbeschreibung . . . . .	91
1.6.8	Zusammenfassung und Übungsaufgaben . . . . .	92
<b>2</b>	<b>Synthese und Logikoptimierung . . . . .</b>	<b>95</b>
2.1	Register-Transfer-Synthese . . . . .	96
2.1.1	Beschreibung und Extraktion von Registern . . . . .	97
2.1.2	Kombinatorische Schaltungen . . . . .	99
2.1.3	Kombinatorische Schaltungen mit Abtastregistern . . . . .	106
2.1.4	Latches . . . . .	111
2.1.5	Constraints . . . . .	116
2.1.6	Entwurfsfehler und Fehlervermeidung . . . . .	119
2.1.7	Zusammenfassung und Übungsaufgaben . . . . .	122
2.2	Schaltungsvereinfachung auf Basis der Schaltalgebra . . . . .	125
2.2.1	Umformungs- und Vereinfachungsregeln . . . . .	125
2.2.2	Optimierungsziele . . . . .	128
2.2.3	Schaltungsvereinfachung mit Konjunktionsmengen . . . . .	129
2.2.4	KV-Diagramm . . . . .	131
2.2.5	Verfahren von Quine und McCluskey . . . . .	138
2.2.6	Zusammenfassung und Übungsaufgaben . . . . .	141
2.3	Binäre Entscheidungsdiagramme . . . . .	143
2.3.1	Vereinfachung binärer Entscheidungsdiagramme . . . . .	144
2.3.2	Zweistellige Operationen mit ROBDDs . . . . .	146
2.3.3	Umsetzung von ROBDDs in minimierte Schaltungen . . . . .	148
2.3.4	Zusammenfassung und Übungsaufgaben . . . . .	149
2.4	Zahlendarstellung . . . . .	150
2.4.1	Stellenwertsystem . . . . .	151
2.4.2	Vorzeichenbehaftete Zahlen im Zweierkomplement . . . . .	153
2.4.3	Festkommazahlen . . . . .	155
2.4.4	Gleitkommazahlen . . . . .	156
2.4.5	Zahlendarstellung und -verarbeitung in VHDL . . . . .	158
2.4.6	Zusammenfassung und Übungsaufgaben . . . . .	159
2.5	Addierer . . . . .	161
2.5.1	Ripple-Addierer . . . . .	161
2.5.2	Serieller Addierer . . . . .	163



2.5.3	Schneller Übertragsdurchlauf	167
2.5.4	Hierarchische Addierer	168
2.5.5	Carry-Save-Addierer	171
2.5.6	Zusammenfassung und Übungsaufgaben	172
2.6	Weitere Rechenwerke	173
2.6.1	Subtrahierer	173
2.6.2	Zähler	174
2.6.3	Negation und Betragsbildung	174
2.6.4	Multiplizierer	176
2.6.5	Dividierer	177
2.6.6	Vergleicher	182
2.6.7	Verschiebung und Rotation	182
2.6.8	Synthese-Suchmuster und Optimierung	184
2.6.9	Zusammenfassung und Übungsaufgaben	187
<b>3</b>	<b>VHDL im Detail</b>	<b>189</b>
3.1	Imperative Beschreibungsmittel	189
3.1.1	Operatoren, Ausdrücke und Zuweisungen	191
3.1.2	Fallunterscheidungen	194
3.1.3	Schleifen	195
3.1.4	Funktionen	198
3.1.5	Unreine Funktionen und globale Variablen	204
3.1.6	Operatorfunktionen	205
3.1.7	Prozeduren	206
3.1.8	Nebenläufige Prozeduren	209
3.1.9	Zusammenfassung und Übungsaufgaben	212
3.2	Anwendungsspezifische Datentypen	214
3.2.1	Zahlentypen	214
3.2.2	Aufzählungstypen	218
3.2.3	Physikalische Typen	219
3.2.4	Attribute von elementaren Datentypen	220
3.2.5	Felder	221
3.2.6	Zuordnungslisten für Feldelemente	224
3.2.7	Bitvektoren	226
3.2.8	Verbund	230
3.2.9	Zusammenfassung und Übungsaufgaben	233
3.3	Ein- und Ausgabe	235
3.3.1	Das Dateikonzept von VHDL	235
3.3.2	Textausgabe	236
3.3.3	Texteingabe	240
3.3.4	Zusammenfassung und Übungsaufgaben	242
3.4	Beschreibungsschablonen für digitale Schaltungen	244
3.4.1	Auslagerung kombinatorischer Funktionen in Packages	244
3.4.2	Bäume statt Ketten	250
3.4.3	Blockspeicher	252

3.4.4	Ein objektorientiertes FIFO-Modell . . . . .	262
3.4.5	UART – schrittweise Modellentwicklung . . . . .	265
3.4.6	Entwicklung eines CORDIC-Rechenwerks . . . . .	278
3.4.7	Zusammenfassung und Übungsaufgaben . . . . .	286
3.5	Methoden und Funktionsbausteine für den Test . . . . .	288
3.5.1	Pseudo-Zufallstest und Mehrversionsvergleich . . . . .	289
3.5.2	Vorab berechnete Eingaben und Soll-Werte . . . . .	294
3.5.3	Spezifikationstest . . . . .	296
3.5.4	Zusammenfassung und Übungsaufgaben . . . . .	300
<b>4</b>	<b>Vom Transistor zur Schaltung . . . . .</b>	<b>301</b>
4.1	Entwurf und Modellierung von CMOS-Gattern . . . . .	301
4.1.1	MOS-Transistoren als Schalter . . . . .	301
4.1.2	Geschaltete Transistornetzwerke . . . . .	305
4.1.3	Signale mit mehreren Quellen . . . . .	307
4.1.4	FCMOS-Gatter . . . . .	309
4.1.5	Bussignale und deaktivierbare Treiber . . . . .	312
4.1.6	Gatter mit Pull-Up- und Pull-Down-Elementen . . . . .	313
4.1.7	Transfergatter und Multiplexer . . . . .	315
4.1.8	Geometrische Beschreibung . . . . .	317
4.1.9	Zusammenfassung und Übungsaufgaben . . . . .	320
4.2	Zeitverhalten . . . . .	322
4.2.1	Simulation des elektrischen Verhaltens . . . . .	323
4.2.2	Verzögerungsparameter . . . . .	329
4.2.3	Parameterbestimmung mit Ringinvertern . . . . .	332
4.2.4	Gatter mit mehreren Eingängen . . . . .	333
4.2.5	Simulation mit geschalteten Transistorbreiten . . . . .	335
4.2.6	Reduktion auf zwei Zeitparameter . . . . .	339
4.2.7	Gepufferte CMOS-Gatter . . . . .	341
4.2.8	Zusammenfassung und Übungsaufgaben . . . . .	342
4.3	Speicherzellen, Latches und Register . . . . .	344
4.3.1	Dynamische Speicherzellen . . . . .	344
4.3.2	RS-Flipflop . . . . .	345
4.3.3	D-Flipflops und Latches . . . . .	346
4.3.4	Übernahme an der Taktflanke . . . . .	347
4.3.5	Register aus Master-Slave-Flipflops . . . . .	349
4.3.6	Taktversorgung . . . . .	351
4.3.7	Zusammenfassung und Übungsaufgaben . . . . .	355
4.4	Schreib-Lese-Speicher . . . . .	357
4.4.1	SRAM . . . . .	358
4.4.2	Mehrportspeicher . . . . .	365
4.4.3	Assoziativspeicher . . . . .	369
4.4.4	Dynamische Speicher (DRAM) . . . . .	371
4.5	Festwertspeicher . . . . .	375
4.6	Programmierbare Logikschaltkreise . . . . .	376

4.6.1	Programmierbare Tabellenfunktionen . . . . .	377
4.6.2	Programmierbare UND-Matrix . . . . .	378
4.6.3	Weitere programmierbare Elemente . . . . .	379
4.6.4	Zusammenfassung . . . . .	380
<b>5</b>	<b>Komplexe Beispielentwürfe . . . . .</b>	<b>381</b>
5.1	Pipeline-Verarbeitung und Speicherengpass . . . . .	381
5.1.1	Prinzip der Pipeline-Verarbeitung . . . . .	381
5.1.2	Ausbalancieren einer Pipeline . . . . .	383
5.1.3	Engpass Speicher . . . . .	384
5.2	FIR-Filter mit Blockspeichern . . . . .	384
5.2.1	Planung der Speicherstruktur und der Pipeline-Abläufe	385
5.2.2	Der gesamte Operationsablauf . . . . .	387
5.2.3	Datentypen und Bearbeitungsmethoden . . . . .	388
5.2.4	Das erste komplette Simulationsmodell . . . . .	390
5.2.5	Ersatz der Zahlentypen durch Bitvektortypen . . . . .	393
5.2.6	Ein- und Ausgabe über Signale . . . . .	395
5.2.7	Umformung in eine Automatenbeschreibung . . . . .	397
5.2.8	Zusammenfassung und Übungsaufgaben . . . . .	402
5.3	Point-of-Interest-Berechnung . . . . .	402
5.3.1	Die Zielfunktion . . . . .	403
5.3.2	Grobkonzept der Hardware-Struktur . . . . .	403
5.3.3	Optimierung der 2D-FIR-Filter . . . . .	405
5.3.4	Entwurf der Filter-Pipeline . . . . .	406
5.3.5	Nachladen des Bildausschnittsspeichers . . . . .	408
5.3.6	Suche der betragsmäßig größten Filterantworten . . . . .	410
5.3.7	Klassifikation . . . . .	412
5.3.8	Sortieren der klassifizierten Punkte . . . . .	412
5.3.9	Zusammenfassung . . . . .	414
5.4	Entwurf eines RISC-Prozessors . . . . .	414
5.4.1	Die Pipeline für Verarbeitungsbefehle . . . . .	415
5.4.2	Lade- und Speichereinheit . . . . .	417
5.4.3	Sprung-Pipeline . . . . .	418
5.4.4	Entwurf eines Befehlssatzes für einen Minimalprozessor	420
5.4.5	Konstanten- und Datentypdefinitionen . . . . .	423
5.4.6	Das Rechenwerk . . . . .	424
5.4.7	Test des Rechenwerks . . . . .	427
5.4.8	Schaltung zur Auswertung der Sprungbedingung . . . . .	429
5.4.9	Das Pipeline-Objekt . . . . .	430
5.4.10	Die Übergangsfunktion für einen Pipeline-Schritt . . . . .	432
5.4.11	Test der Übergangsfunktion . . . . .	435
5.4.12	Zusammenfassung . . . . .	438

<b>6</b>	<b>Lösungen zu den Übungsaufgaben</b> .....	439
6.1	Modellbildung und Simulation .....	439
6.1.1	Entwurf digitaler Schaltungen .....	439
6.1.2	Funktion, Struktur und Simulation .....	440
6.1.3	Laufzeittoleranz .....	442
6.1.4	Register .....	444
6.1.5	Asynchrone Eingabe .....	446
6.1.6	Sequenzielle Schaltungen .....	447
6.2	Synthese und Logikoptimierung .....	450
6.2.1	Register-Transfer-Synthese .....	450
6.2.2	Schaltungsvereinfachung auf Basis der Schaltalgebra ...	452
6.2.3	Binäre Entscheidungsdiagramme .....	456
6.2.4	Zahlendarstellung .....	457
6.2.5	Addierer .....	459
6.2.6	Weitere Rechenwerke .....	459
6.3	VHDL im Detail .....	463
6.3.1	Imperative Beschreibungsmittel .....	463
6.3.2	Anwendungsspezifische Datentypen .....	465
6.3.3	Ein- und Ausgabe .....	466
6.3.4	Beschreibungsschablonen für digitale Schaltungen .....	469
6.3.5	Methoden und Funktionsbausteine für den Test .....	471
6.4	Vom Transistor zur Schaltung .....	472
6.4.1	Entwurf und Modellierung von CMOS-Gattern .....	472
6.4.2	Zeitverhalten .....	474
6.4.3	Speicherzellen, Latches und Register .....	476
6.5	Komplexe Beispielenwürfe .....	476
6.5.1	FIR-Filter mit Blockspeichern .....	476
<b>A</b>	<b>Ergänzungen zu VHDL</b> .....	479
A.1	Syntax, Schlüsselworte, Bezeichner, Konstanten .....	479
A.1.1	Syntaxregeln für die Beschreibung von Sprachkonstrukten .....	479
A.1.2	Schlüsselworte .....	479
A.1.3	Syntaxregeln für Bezeichner .....	480
A.1.4	Zeichen- und Zeichenkettenkonstanten .....	481
A.1.5	Darstellung von Zahlenwerten .....	481
A.1.6	Vordefinierte Attribute .....	482
A.2	Die Bibliothek »Tuc« .....	484
A.2.1	Textausgabe (Tuc.Ausgabe) .....	484
A.2.2	Kontrollierter Simulationsabbruch (Tuc.StopSim_pack)	486
A.2.3	Texteingabe (Package Tuc.Eingabe) .....	487
A.2.4	Arithmetische Operationen (Tuc.Numeric_Sim und Tuc.Numeric_Synth) .....	489
A.2.5	Pseudo-Zufallstest (Tuc.Zufallstest) .....	492

**Sachverzeichnis** ..... 493

**Literaturverzeichnis** ..... 501