

TEUBNER-TEXTE zur Informatik Band 15

J. Keller/W. J. Paul

Hardware Design

TEUBNER-TEXTE zur Informatik

Herausgegeben von

Prof. Dr. Johannes Buchmann, Saarbrücken

Prof. Dr. Udo Lipeck, Hannover

Prof. Dr. Franz J. Rammig, Paderborn

Prof. Dr. Gerd Wechsung, Jena

Als relativ junge Wissenschaft lebt die Informatik ganz wesentlich von aktuellen Beiträgen. Viele Ideen und Konzepte werden in Originalarbeiten, Vorlesungsskripten und Konferenzberichten behandelt und sind damit nur einem eingeschränkten Leserkreis zugänglich. Lehrbücher stehen zwar zur Verfügung, können aber wegen der schnellen Entwicklung der Wissenschaft oft nicht den neuesten Stand wiedergeben.

Die Reihe „TEUBNER-TEXTE zur Informatik“ soll ein Forum für Einzel- und Sammelbeiträge zu aktuellen Themen aus dem gesamten Bereich der Informatik sein. Gedacht ist dabei insbesondere an herausragende Dissertationen und Habilitationsschriften, spezielle Vorlesungsskripten sowie wissenschaftlich aufbereitete Abschlußberichte bedeutender Forschungsprojekte. Auf eine verständliche Darstellung der theoretischen Fundierung und der Perspektiven für Anwendungen wird besonderer Wert gelegt. Das Programm der Reihe reicht von klassischen Themen aus neuen Blickwinkeln bis hin zur Beschreibung neuartiger, noch nicht etablierter Verfahrensansätze. Dabei werden bewußt eine gewisse Vorläufigkeit und Unvollständigkeit der Stoffauswahl und Darstellung in Kauf genommen, weil so die Lebendigkeit und Originalität von Vorlesungen und Forschungsseminaren beibehalten und weitergehende Studien angeregt und erleichtert werden können.

TEUBNER-TEXTE erscheinen in deutscher oder englischer Sprache.

Hardware Design

Formaler Entwurf digitaler Schaltungen

Von Dr. Jörg Keller
Prof. Dr. Wolfgang J. Paul
Universität des Saarlandes



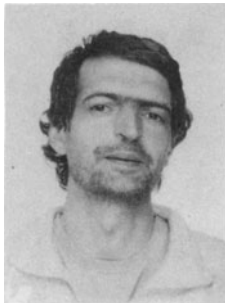
B. G. Teubner Verlagsgesellschaft
Stuttgart · Leipzig 1995



Dr. Jörg Keller

Geboren 1966 in Saarbrücken. Von 1986 bis 1989 Studium der Informatik mit Nebenfach Mathematik an der Universität des Saarlandes, Diplom 1989. Von 1989 bis 1992 und 1993 bis 1995 wissenschaftlicher Mitarbeiter am Lehrstuhl für Rechnerarchitektur der Universität des Saarlandes bei Prof. Dr. W. J. Paul, Promotion 1992. Von 1992 bis 1993 Postdoktorand am Centrum voor Wiskunde en Informatica (CWI), Amsterdam. Seit Mai 1995 Stipendiat der Deutschen Forschungsgemeinschaft am Lehrstuhl für Rechnerarchitektur der Universität des Saarlandes.

Arbeitsschwerpunkte: Hardware Design, parallele Algorithmen, Rechnerarchitektur, verteilte und parallele Systeme.



Prof. Dr. Wolfgang J. Paul

Geboren 1951 in Dudweiler. Von 1969 bis 1972 Studium der Physik, Mathematik und Informatik an der Universität des Saarlandes, 1972 Diplom in Informatik. Von 1972 bis 1974 wissenschaftlicher Mitarbeiter am Institut für Angewandte Mathematik und Informatik der Universität des Saarlandes bei Prof. Dr. G. Hotz, Promotion 1973. Von 1974 bis 1976 Postdoktorand am Computer Science Department der Cornell University, Ithaca, NY (USA). Von 1976 bis 1978 Vertretung des Amtes eines wiss. Rates und Professors, von 1978 bis 1982 wiss. Rat und Professor an der Fakultät für Mathematik der Universität Bielefeld. Von 1982 bis 1986 Research staff member am IBM Forschungslabor in San Jose, CA (USA). Seit 1986 Professor am Fachbereich Informatik der Universität des Saarlandes. 1986 Gottfried Wilhelm Leibniz Förderpreis zusammen mit G. Hotz und K. Mehlhorn.

Arbeitsschwerpunkte: Hardware Design, Komplexitätstheorie, Rechnerarchitektur.

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Keller, Jörg:

Hardware-Design : formaler Entwurf digitaler Schaltungen /

von Jörg Keller ; Wolfgang J. Paul. –

Stuttgart ; Leipzig : Teubner, 1995

(Teubner-Texte zur Informatik ; Bd. 15)

ISBN 978-3-8154-2065-2 ISBN 978-3-322-93438-3 (eBook)

DOI 10.1007/978-3-322-93438-3

NE: Paul, Wolfgang J. ; GT

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 Verlagsgesellschaft Leipzig 1995

Umschlaggestaltung: E. Kretschmer, Leipzig

Unseren Gymnasiallehrern gewidmet, insbesondere

Rudolf Conradi

Franz Röder

Erwin Saar

Engelbert Walle

Vorwort

Das vorliegende Lehrbuch beschäftigt sich in mathematisch präziser Weise mit einem ganz und gar praktischen Thema, nämlich dem Entwurf von Hardware. Kapitel 1 enthält eine Diskussion mathematischer Grundbegriffe. In den Kapiteln 2 bis 4 werden die notwendigen theoretischen Grundlagen über Boole'sche Ausdrücke, Schaltkreiskomplexität und Rechnerarithmetik behandelt. Der Übergang von der abstrakten Schaltkreistheorie zum Entwurf konkreter Schaltungen findet nahtlos in Kapitel 5 statt, wo aus den Verzögerungszeiten von Gattern das zeitliche Verhalten von Flipflops und anderen Speicherbausteinen abgeleitet wird. Kapitel 6 enthält dann das vollständige Design eines einfachen Rechners.

Das Lehrbuch ist aus Vorlesungen des zweiten Autors entstanden. Kapitel 2 bis 6 enthalten den Stoff für eine einsemestrige Anfängervorlesung. Kapitel 1 ist mehr ein Nachschlagewerk für nagende Fragen, die sich früher oder später einstellen. Kapitel und Abschnitte, die mit einem Stern gekennzeichnet sind, kann man überspringen. Die Darstellung großer Teile der Kapitel 1, 5 und 6 ist neu. Die Kapitel 2 bis 4 haben einen erheblichen Anteil an fortgeschrittenem Material. Es hat sich jedoch gezeigt, daß Erstsemester damit keine besonderen Schwierigkeiten haben.

Im Einzelnen enthält Kapitel 1 eine Entwicklung des formalen Mengenkonzepts in einer für Studienanfänger verständlichen Form. Die Sätze und Beweise in diesem Kapitel sind von vorne bis hinten klassische Mathematik. Der Standpunkt, von dem aus wir sie interpretieren, ist bewußt übermütig und nicht klassisch. Wir wollen damit den Studierenden schmackhaft machen, von Zeit zu Zeit auch Wohlvertrautes von einem frischen — natürlich nicht unsinnigen — Standpunkt aus zu betrachten. Das ist insbesondere beim Forschen manchmal sehr nützlich.

In Kapitel 2 beschränken wir uns bei der Behandlung Boole'scher Ausdrücke nicht auf die Funktionszeichen \wedge , \vee und \sim , sondern wir lassen zusätzlich abzählbar viele frei definierbare Funktionszeichen zu. Dieses Vorgehen ist aus der Logik bekannt. Der zusätzliche Aufwand ist sehr gering. Der Vorteil gegenüber dem üblichen Vorgehen ist, daß man den Kalkül rechtfertigen kann, *mit dem man nachher tatsächlich rechnet*.

In Kapitel 3 rechtfertigen wir — ebenfalls mit wenig Aufwand — das formale Rechnen mit Gattern in Schaltkreisen.

An fortgeschrittenem Material enthalten Kapitel 2 und 3 kombinatorische Abzählargumente sowie eine komplette Behandlung von Resolutionssatz und Kompaktheitsatz der Aussagenlogik. Was üblicherweise bei der Behandlung des Resolutionskalküls Zeit kostet, sind nicht die Beweise, sondern die Begriffsbildung. Genau diese Begriffsbildung ist aber im Anschluß an die Optimierung von disjunktiven Normalformen bereits abgeschlossen: das Verfahren von Quine-McCluskey ist ein spezielles Resolutionsverfahren. Man braucht nur noch zu dualisieren.

In Kapitel 4 über Rechnerarithmetik behandeln wir nicht nur die Verfahren, die für Dezimalzahlen schon aus der Schule bekannt sind, sondern auch viel effizientere Verfahren wie Conditional-Sum Addierer, Carry-Lookahead Addierer, Wallace-Tree Multiplizierer und Multiplizierer nach Karatsuba/Ofman. Auf die beiden fortgeschrittenen Addierer kommen wir ausdrücklich in Kapitel 6 zurück, um den dort konstruierten Rechner schneller zu machen.

In Kapitel 5 werden aus Gattern nacheinander Bausteine wie R/S-Flipflops, D-Latches, D-Flipflops und SRAMs konstruiert. Aus den Verzögerungszeiten der verwendeten Gatter werden Parameter wie Setup-Zeit, Hold-Zeit und minimale Pulsweite dieser Bausteine hergeleitet.

In Kapitel 6 schließlich wird aus kommerziell erhältlichen, niedrig integrierten Bausteinen und PALs systematisch ein vollständiger Schaltplan einschließlich Speicheranbindung und Busprotokoll für einen einfachen Rechner entwickelt.

Trotz aller Bemühungen sind vermutlich Fehler verblieben. Für Hinweise auf Fehler (am einfachsten durch electronic mail an hadesign@cs.uni-sb.de) sind wir dankbar. Eine elektronische Liste der bekannten Fehler kann ebenfalls unter obiger Adresse angefordert werden.

Saarbrücken, im Juni 1995

Jörg Keller

Wolfgang J. Paul

Inhaltsverzeichnis

1	Mathematische Grundlagen	13
1.1	Über Gott und die Welt	13
1.2	Elementare Konzepte	17
1.2.1	Natürliche Zahlen	17
1.2.2	Induktion	19
1.2.3	Variablen und der naive Mengenbegriff	20
1.2.4	Funktionen und Relationen	23
1.2.5	Zeichenreihen	29
1.2.6	Zeichensätze	33
1.3	Rechnen	36
1.3.1	Vollständig geklammerte Ausdrücke	38
1.3.2	Auswertung von Ausdrücken	43
1.3.3	Identitäten	46
1.3.4	Unvollständig geklammerte Ausdrücke	50
1.4	Zahlen	59
1.4.1	Ganze Zahlen, rationale Zahlen, reelle Zahlen	59
1.4.2	Zahlendarstellungen	62
1.4.3	Abzählbarkeit	64
1.5	Formale Logik	71
1.5.1	Prädikate	71
1.5.2	Formale Fassung des Mengenkonzepts	75
1.5.3	Beweissysteme	83
1.6	Übungen	86
2	Boole'sche Ausdrücke	90
2.1	Mächtigkeit endlicher Mengen	90
2.2	Rechnen mit Boole'schen Ausdrücken	94
2.2.1	Vollständig geklammerte Ausdrücke	96
2.2.2	Einsetzungen	98
2.2.3	Identitäten und Ungleichungen	99
2.2.4	Lösen von Gleichungen	104

2.2.5	Der Darstellungssatz	105
2.3	Kosten von Ausdrücken	107
2.3.1	Definitionen	107
2.3.2	*Allgemeine Schranken	109
2.4	Polynome und Resolution	113
2.4.1	Polynome und Primimplikanten	113
2.4.2	Das Verfahren von Quine–McCluskey	116
2.4.3	Monome und n -Würfel	119
2.4.4	Bestimmung von Minimalpolynomen	128
2.4.5	*Nullstellen und Erfüllbarkeit	131
2.4.6	*Erfüllbarkeit von konjunktiven Normalformen	135
2.4.7	*Resolutionsbeweise	138
2.5	Übungen	144
3	Schaltkreise	147
3.1	Gerichtete Graphen und Schaltkreise	147
3.1.1	Gerichtete Graphen	147
3.1.2	Gatter	149
3.1.3	Schaltkreise	150
3.2	Rechnen mit Schaltkreisen	152
3.2.1	Einsetzungen	152
3.2.2	Identitäten und berechnete Funktionen	155
3.2.3	Anfangsschaltkreise	157
3.3	Darstellungssatz	158
3.4	Schaltkreiskomplexität	162
3.4.1	Komplexitätsmaße	162
3.4.2	Assoziativität und balancierte Bäume	164
3.5	*Schaltkreise und Boole'sche Ausdrücke	168
3.5.1	Subfunktionen	170
3.5.2	Das Kriterium von Neciporuk	172
3.5.3	Selektoren	174
3.5.4	Funktionen mit vielen Subfunktionen	177
3.6	Übungen	179
4	Arithmetik	182
4.1	Zahlendarstellungen	182
4.1.1	Stellenwertsysteme	182
4.1.2	Binärzahlen	184
4.1.3	2's-Complement-Darstellung	185
4.2	Addierer	188
4.2.1	Halb- und Volladdierer	188
4.2.2	Carry-Chain Addierer	190

4.2.3	Incrementer	193
4.2.4	Conditional-Sum Addierer	194
4.2.5	Carry-Lookahead Addierer	200
4.3	Subtraktion	205
4.3.1	Addition von 2's-Complement-Zahlen	205
4.3.2	Addierer und Subtrahierer	207
4.4	*Multiplizierer	208
4.4.1	Multiplikation nach der Schulmethode	209
4.4.2	Wallace-Tree Multiplizierer	214
4.4.3	Multiplikation nach Karatsuba und Ofman	220
4.5	Aufbau von Arithmetikeinheiten	223
4.5.1	Bit Slice Designs	228
4.6	Übungen	230
5	Speicher und Tristate-Bausteine	234
5.1	Physikalische Eigenschaften von Gattern	235
5.1.1	Logische und physikalische Signale	236
5.1.2	Fanout	237
5.1.3	Verzögerungszeiten	239
5.1.4	*Transferfunktionen	242
5.1.5	Kapazitive Last	244
5.1.6	Worst Case Timing-Analyse	246
5.1.7	Spikefreies Umschalten von Gattern	251
5.2	Flipflops	252
5.2.1	R/S-Flipflop	252
5.2.2	D-Latch	255
5.2.3	D-Flipflop	258
5.3	Bausteine mit Flipflops	261
5.3.1	Register	261
5.3.2	Zähler	262
5.4	Statischer Speicher	263
5.4.1	Mehrfaches OR	264
5.4.2	Treiberbäume	264
5.4.3	Dekodierer	269
5.4.4	Aufbau eines statischen Speichers	270
5.5	Tristate-Treiber, Busse und Pipelines	276
5.5.1	Speicher und Bus Contention	281
5.5.2	Wired OR	283
5.6	Übungen	284
6	Ein einfacher Rechner	287
6.1	Die abstrakte RESA-Maschine	288

6.1.1	Load und Store	290
6.1.2	Compute Befehle	291
6.1.3	Immediate Befehle	292
6.1.4	Indexregister	293
6.1.5	Sprungbefehle	293
6.2	Instruktionssatz	296
6.2.1	Instruktionsformate	297
6.2.2	Load Befehle	298
6.2.3	Store Befehle	299
6.2.4	Compute Befehle	299
6.2.5	Sprungbefehle	300
6.2.6	Erzeugung großer Konstanten und Shifts	301
6.2.7	Selbstmodifikation und von Neumann-Architektur	303
6.2.8	Sign Extension	303
6.3	Datenpfade	304
6.4	Idealisierte Timing-Diagramme	314
6.5	PLAs, PROMs und PALs	317
6.5.1	PLAs	319
6.5.2	PROMs und EPROMs	319
6.5.3	PALs	320
6.5.4	PALASM	322
6.6	Kontrolllogik	326
6.6.1	Clock und Phasensignale	327
6.6.2	Clocksignale	328
6.6.3	Output enable Signale	330
6.6.4	Kontrolle der ALU und Sign-Extension	334
6.6.5	Laden des Befehlszählers	336
6.6.6	Asynchrone Signale und Reset	338
6.6.7	Ansteuerung des Speichers	340
6.7	Exakte Timing-Analyse	342
6.7.1	Kontrolllogik	343
6.7.2	Vermeidung von Bus contention	344
6.7.3	Inkrementieren des Befehlszählers	345
6.7.4	Compute-Befehle	346
6.7.5	LOADIN1 und STOREIN1	349
6.7.6	Jump	351
6.7.7	Zykluszeit und Befehlsrate	351
6.7.8	ALUs mit Carry Lookahead und Conditional Sum	351
6.7.9	Kontrollsignale mit 3.5 Zyklen Dauer	355
6.7.10	Verkürzung des Fetch-Zyklus	357
6.7.11	Einschalten des Stroms	359
6.8	Ein- und Ausgabe	360

6.8.1	I/O-Ports und Interfaces	361
6.8.2	UART	363
6.8.3	EPROM	365
6.8.4	Datenpfade und Memory Map	365
6.8.5	Busprotokoll und Wait-Zyklen	368
6.8.6	Zustandsdiagramme	374
6.8.7	Kontrolle des Speichers	378
6.9	Übungen	385
Anhang		393
A Verwendete Bauteile		393
A.1	Gatter	393
A.2	Multiplexer	393
A.3	Register	394
A.4	Zähler	395
A.5	Treiber	396
A.6	ALU Bausteine	396
A.7	PALs	397
A.8	Speicherbausteine	398
A.9	Ein- und Ausgabe	399
B PAL-Gleichungen		400
B.1	CPU-Kontrolle	400
B.2	SRAM-Kontrolle	401
B.3	EPROM-Kontrolle	402
B.4	UART-Kontrolle	403
Literaturverzeichnis		405
Index		408