

A. Bleck/M. Goedecke/S. Huss/K. Waldschmidt
Praktikum des modernen VLSI-Entwurfs

Praktikum des modernen VLSI-Entwurfs

Eine Einführung in die Entwurfsprinzipien und -beschreibungen, unter besonderer Berücksichtigung von VHDL;
mit einer umfangreichen Anleitung zum Praktikum

Von Dipl.-Inf. Andreas Bleck
Universität Frankfurt/Main
Dipl.-Inf. Michael Goedecke
Prof. Dr.-Ing. Sorin A. Huss
Techn. Hochschule Darmstadt
und Prof. Dr.-Ing. Klaus Waldschmidt
Universität Frankfurt/Main



B. G. Teubner Stuttgart 1996

Die Texte, Schaubilder und Figuren dieses Buches wurden mit größter Gewissenhaftigkeit erstellt. Trotzdem können Irrtümer und Fehler nicht vollständig ausgeschlossen werden. Verlag, Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen. Für Verbesserungsvorschläge und Hinweise sind die Autoren und der Verlag dankbar.

Mentor, Cadence, Synopsys und andere sind eingetragene Warenzeichen und gesetzlich geschützt.

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Praktikum des modernen VLSI-Entwurfs : eine Einführung in die Entwurfsprinzipien und -beschreibungen, unter besonderer Berücksichtigung von VHDL ; mit einer umfangreichen Anleitung zum Praktikum / von Andreas Bleck ... – Stuttgart : Teubner, 1996

ISBN 978-3-519-02296-1 ISBN 978-3-322-94669-0 (eBook)

DOI 10.1007/978-3-322-94669-0

NE: Bleck, Andreas

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 1996

Einband: Peter Pfitz, Stuttgart

Vorwort

Der moderne Mensch ist täglich von zahlreichen mikroelektronischen Systemen umgeben. Dies reicht von der Steuerung eines Rasierapparates über die umfangreiche Elektronik in Kraftfahrzeugen mit bis zu 100 Prozessoren bis hin zu Computern. Für den Entwurf fast aller technischen Produkte ist es daher von entscheidender Bedeutung, diese eingebetteten mikroelektronischen Komponenten effizient entwickeln zu können. Das vorliegende Buch dient dazu, diese Grundfertigkeit zukünftigen Informatikern und Ingenieuren zu vermitteln.

VLSI-Entwurf ist keine schwarze Kunst, sondern beruht auf einer wohlstrukturierten Methodik. Es ist gerade diese Methodik, die es erlaubt, digitale Hardware enormer Komplexität in immer kürzerer Zeit und mit einer bemerkenswerten Fehlerfreiheit zu entwerfen. Das vorliegende Buch entwickelt daher in einem ausführlichen ersten Teil diese Methodik. Die Autoren erläutern die wesentlichen Entwurfsprinzipien, wobei deutlich wird, daß die Strukturierung der Entwurfsaufgabe in wohldefinierte Sichten und Abstraktionsebenen ein wesentliches methodisches Gerüst darstellt. Eingefügt in dieses Gerüst lassen sich verschiedene Entwurfsstile und darauf aufbauend Entwurfsabläufe entwickeln, die dann schließlich durch Werkzeuge unterstützt werden. Die Autoren machen in vorbildlicher Weise deutlich, daß der Entwurf digitaler Hardware letztlich die Transformation von Entwurfsbeschreibungen bedeutet. Ein recht ausführlicher Abschnitt ist daher Entwurfssprachen und dabei insbesondere der Sprache VHDL gewidmet. Ein wesentlicher positiver Aspekt von Beschreibungen mit Sprachen wie VHDL ist, daß es sich auf jeder Abstraktionsebene um ausführbare Spezifikationen handelt, wobei ich an dieser Stelle vereinfachend ein Modell auf einer Abstraktionsebene als Spezifikation für die darunterliegende Ebene auffassen will. Diese Ausführbarkeit in Form von Simulation stellt bis heute einen Schlüssel zu der erzielbaren Entwurfssicherheit dar, auch wenn sich formale Verifikationsmethoden zunehmend ihren Platz erobern. Die weithin akzeptierte Kanonisierung des Hardwareentwurfs erlaubte es, für die verschiede-

nen Entwurfsschritte leistungsfähige Werkzeuge bereitzustellen. Dabei wird der Übergang von einer Abstraktionsebene auf eine niedrigere mit dem Begriff Implementation bzw. im vorliegenden Kontext häufig auch mit dem Begriff Synthese bezeichnet. Die standardisierte Sprache VHDL hat dafür einen weithin akzeptierten Rahmen gesetzt, um algorithmische, RT- und Logiksynthese darstellen zu können. Der anschließende physikalische Entwurf wird traditionell von leistungsfähigen Werkzeugen unterstützt. Die Autoren haben es verstanden, im ersten Teil des vorliegenden Buches diesen Entwurfsbereich in seiner Gesamtheit kompakt darzustellen. Hier liegt ein für das Selbststudium oder als Grundlage für eine Vorlesung gleichermaßen hervorragend geeigneter Text vor.

Im zweiten Teil des Buches entwickeln die Autoren einen sehr geschickt angelegten Baukasten zur Durchführung unterschiedlich konzipierter Entwurfspraktika. Dieser Ansatz erscheint mir sehr bemerkenswert. Durch diesen Baukasten wird den lehrenden Personen freigestellt, ob sie eine Top-down-, eine Bottom-up- oder eine Meet-in-the-Middle-Entwurfsmethodik lehren wollen, oder gar eine Mischung daraus. Es werden 13 Einzelaufgaben entwickelt, die in ihrer Komplexität kontinuierlich zunehmen. Für jede der Aufgaben wird das Lehrziel präzise angegeben und eine Anzahl für sich jeweils gut überschaubarer Teilaufgaben gestellt. Konsequenterweise wird dabei durchgängig darauf geachtet, daß der moderne synthesebasierte Entwurfsansatz verfolgt wird und dieser Ansatz von zeitgemäßen Werkzeugen unterstützt wird. Interessant ist, daß bereits innerhalb der Einzelaufgaben wahlweise ein Top-down-, Bottom-up oder Meet-in-the-Middle-Vorgehen selektiv eingeübt werden kann. Die Komplexität der Aufgabenstellung gipfelt im Entwurf eines kleinen hypothetischen Mikroprozessors. Man mag zwar einwenden, daß es in den wenigsten Fällen im zukünftigen Berufsleben der auszubildenden Ingenieure und Informatiker darum gehen wird, Mikroprozessoren zu entwickeln, doch zeigen auch meine Erfahrungen, daß sich mit dieser Aufgabenstellung Studenten besonders gut motivieren lassen. Mit der Notwendigkeit, einen Interpretationsalgorithmus für einen Instruktionssatz zu implementieren, und mit dem Paradigma der Aufspaltung von Steuerwerk und Operationswerk lassen sich anhand eines Mikroprozessorentwurfs auch wesentliche Entwurfsmethoden sehr gut einüben. Die Aufgaben stellen nun das Gerüst dar, auf dessen Basis unterschiedliche Ausprägungen von Praktika entwickelt werden können. Die Autoren stellen sechs wohlüberlegte Standardpraktika dar. Die mögliche Variationsbreite jedoch ist schier

unerschöpflich. Selbst Lehrende, die in ihren Veranstaltungen andere Beispielsysteme realisieren lassen wollen, werden auf wesentliche Teile der vorliegenden Einzelaufgaben zurückgreifen können. Daß dieser Rückgriff durch das elektronische Bereitstellen des zugrundeliegenden VHDL-Materials unterstützt wird, verdient besonders hervorgehoben zu werden.

Mit dem vorliegenden Buch ist es den Autoren gelungen, die Basis für eine weite Verbreitung des Entwurfswissens für digitale mikroelektronische Systeme zu legen. Andererseits werden Lehrende nicht in der konkreten Ausgestaltung ihrer Lehrkonzepte eingeengt. Abhängig von der Verfügbarkeit von Entwurfswerkzeugen und insbesondere vom jeweiligen didaktischen Konzept kann auf der Basis des Buches eine angepaßte Lehrveranstaltung maßgeschneidert werden. Ich habe das Manuskript mit großer Freude gelesen und kann mir nur eine weite Durchdringung bei Informatikern und Ingenieuren wünschen.

Paderborn, im November 1995

Franz J. Rammig

Inhalt

I	Grundlagen	19
1	Einleitung	21
2	Entwurfsprinzipien	25
2.1	Ziele und Randbedingungen des VLSI-Entwurfs	25
2.1.1	Beherrschung der Komplexität	25
2.1.2	Ökonomische Aspekte	27
2.1.3	Testbarkeit	29
2.1.4	Entwurfswerkzeuge	30
2.2	Entwurfsdomänen	30
2.3	Abstraktionsebenen und Modellierungskonzepte	33
2.3.1	Abstraktionsebenen	34
2.3.2	Modellierungskonzepte	37
2.4	Entwurfsstile	40
2.4.1	Full-Custom-Entwurf	42
2.4.2	Semi-Custom-Entwurfsstile	43
2.4.2.1	Standardzellen	46
2.4.2.2	Makrozellen	47
2.4.2.3	Gate-Arrays	49
2.4.2.4	Sea-of-Gates	49
2.5	Entwurfsablauf	50
2.5.1	Generierende Aktivitäten	50
2.5.2	Überprüfende Aktivitäten	51
2.5.3	Entwurfsablauf im Y-Diagramm	53
2.5.4	Idealisierte und reale Entwurfsabläufe	54

3	Entwurfsbeschreibungen	56
3.1	Entwurfserfassung	57
3.2	Schaltplaneingabe	58
3.2.1	Schematic-Editoren	58
3.2.1.1	Schematic-Elemente	58
3.2.1.2	Manipulation von Schematic-Elementen	60
3.2.2	Symboleditoren	61
3.3	VHDL	62
3.3.1	Entwurfsobjekte	63
3.3.1.1	Abschnitte eines Entwurfsobjektes	64
3.3.1.2	Hierarchie der Ports	67
3.3.2	Prozeß	69
3.3.2.1	<code>process</code> -Anweisung	69
3.3.2.2	Probleme beim Modellieren mit Prozessen	70
3.3.3	Block	72
3.4	Modellierungsebenen	73
3.4.1	Modellierung auf der PMS-Ebene	74
3.4.1.1	Repräsentation von Signalen	76
3.4.1.2	Signalmultiplexing	78
3.4.1.3	Gleichzeitige Einspeisung	78
3.4.1.4	Zeitlich getrennte Einspeisung	82
3.4.1.5	Initialisierung von Signalen	86
3.4.1.6	Out- und Buffer-Ports	88
3.4.1.7	Übersicht über Datentypen für die Modellierung	89
3.4.2	Modellierung auf der Algorithmischen Ebene	90
3.4.2.1	Eigenschaften eines Modells	91
3.4.2.2	Beschreibung von Signalverzögerungen	92
3.4.2.3	Funktionale Partitionierung, Prozeßmodellgraph	94
3.4.2.4	Beschreibung des zeitlichen Ablaufs	97

10 Inhalt

3.4.2.5	Überprüfung der Eingangsspezifikation	104
3.4.3	Modellierung auf der RT-Ebene	106
3.4.3.1	Beispiel: RT-Modell eines Registersystems	107
3.4.3.2	Überprüfung von zeitlichen Restriktionen	111
3.4.3.3	Beschreibung von Schaltwerken	112
3.4.4	Modellierung auf der Logikebene	113
3.4.4.1	Repräsentation logischer Zustände	113
3.4.4.2	Realisierung von verschiedenen Schaltungstechniken	116
3.4.4.3	Modellierung von Verzögerungen	117
3.4.4.4	Modellierung des Trägheitsverhaltens	123
3.4.4.5	Fehlerüberprüfungen	123
4	Simulation	127
4.1	Elemente einer Simulation	127
4.1.1	Simulator	128
4.1.2	Eingabedaten	128
4.1.3	Ausgabedaten	129
4.2	Ablauf einer Simulation	130
4.3	VHDL-Simulation	135
4.3.1	Simulationssystem	135
4.3.2	Simulationszyklus	137
4.3.3	Validierung mit einer Test-Bench	138
4.3.4	Stimuliprogrammierung	139
4.3.4.1	Waveform-Elemente	140
4.3.4.2	Lesen von Dateien in VHDL	140
4.3.4.3	Load_memory	142
4.3.5	Simulationsausgabe	142
4.3.6	Simulationsdatenhaltung	142
4.3.6.1	Pakete	143
4.3.6.2	Bibliotheken	144

4.3.6.3	Konfigurationen	146
4.4	Logiksimulation	152
4.4.1	Funktionen zur Fehlersuche	153
4.4.1.1	Breakpoints (Unterbrechungspunkte)	153
4.4.1.2	Erkennung von Hazards und Spikes	154
4.4.1.3	Setup- und Hold-Zeiten	155
4.4.1.4	Minimale Impulsbreiten	156
4.4.2	Simulationsprinzipien	156
4.4.2.1	Beispiel: Simulation eines 4-Bit-Registers	157
4.4.2.2	Stimuliprogrammierung	158
4.4.2.3	Darstellung von Simulationsergebnissen	161
4.4.3	Untersuchung des Zeitverhaltens	162
4.4.3.1	Dokumentation des Zeitverhaltens	163
4.4.3.2	Bestimmung der maximalen Taktfrequenz	169
4.5	Planung von Simulationsläufen	172
5	VHDL-Synthese	175
5.1	Logiksynthese	177
5.1.1	Minimieren zweistufiger Logik	177
5.1.2	Minimieren von mehrstufigen Logikrealisierungen	177
5.2	RT-Synthese	181
5.3	Algorithmische Synthese	184
5.4	Synthesefähige Beschreibungen in VHDL	187
5.4.1	Umsetzung von Sprachkonstrukten zur Synthese	187
5.4.1.1	Datentypen	187
5.4.1.2	Instantiierungen	188
5.4.1.3	Inferenz von Speicherelementen	189
5.4.1.4	Variable oder Signal	193
5.4.1.5	Inferenz von Tri-State-Komponenten	194

12 Inhalt

5.4.2	Einfluß von VHDL-Beschreibungen auf das Syntheseergebnis	194
5.4.3	Verhaltensorientierte, synthesefähige Beschreibungen	196
5.4.4	Endliche Automaten (FSM)	197
5.4.4.1	Beschreibungsmethoden in VHDL	198
5.4.4.2	Sonderfall: Zyklische FSM	202
5.4.4.3	Behandlung großer Automaten	203
5.4.4.4	Unterstützung der Synthese	204
5.4.4.5	Zustandskodierung	204
5.5	Überblick über den synthesegetützten Entwurf	205
6	Physikalischer Entwurf	207
6.1	Floorplanning	208
6.2	Plazierung	209
6.3	Verdrahtung	210
6.4	Nachbearbeitung („back annotation“)	211
7	Chip-Test	212
7.1	Testumgebung	212
7.2	Elemente eines Tests	213
	Teil I: Literatur	215
II	Praktikum	217

8	Der Praktikums-Mikroprozessor PMP12	219
8.1	Befehlssatzarchitektur des PMP12	219
8.1.1	Grundlegende Konzepte	220
8.1.2	Maschinenbefehlssatz	223
8.1.3	Befehls- und Datenformate	224
8.1.4	Adressierungsarten	226
8.2	Systemeinbindung	227
8.2.1	Programmstart	228
8.2.2	Kommunikation mit der Umgebung	228
8.2.2.1	Kommunikation beim Schreiben, Ausgabebefehle . .	228
8.2.2.2	Kommunikation beim Lesen, Eingabebefehle	230
9	Aufbau von Praktika	231
9.1	Entwurfsstrategien	231
9.1.1	Top-down-Entwurf	231
9.1.2	Bottom-up-Entwurf	232
9.1.3	Meet-in-the-Middle-Entwurf	233
9.2	Standardpraktika	234
9.2.1	Praktikum P1: Einführungspraktikum	234
9.2.2	Praktikum P2: Schematic-basierter Entwurf	236
9.2.3	Praktikum P3: Meet-in-the-Middle-Entwurf	237
9.2.4	Praktikum P4: Top-down-Entwurf	240
9.2.5	Praktikum P5: Einführung in VHDL	241
9.2.6	Praktikum P6: Einführung in die VHDL-Synthese . .	242

10	Praktische Aufgaben	243
10.1	Kaskadierbare 4-Bit-Addierereinheit	245
10.1.1	Grundlagen	245
10.1.2	Bottom-up-Entwurfsdurchführung	247
10.1.2.1	Aufgabenstellung	247
10.1.2.2	Entwurfsdurchführung mit graphischem Editor . . .	249
10.1.2.3	Entwurfsdurchführung mit VHDL	250
10.1.2.4	Validierung des Schematic-basierten Entwurfs . . .	252
10.1.2.5	Validierung des VHDL-Entwurfs	253
10.1.2.6	Auswertung (Nachbereitung)	253
10.1.3	Top-down-Entwurfsdurchführung	255
10.1.3.1	Aufgabenstellung	255
10.1.3.2	Modellierungsarten	256
10.1.3.3	Entwurfsdurchführung	258
10.1.3.4	Parametrisierbare Beschreibungen	263
10.2	Kaskadierbare Arithmetik-Logik-Einheit (ALU)	265
10.2.1	Grundlagen	265
10.2.2	Aufgabenstellung	266
10.2.3	Bottom-up-Entwurfsdurchführung	267
10.2.3.1	Entwurfsdurchführung	269
10.2.3.2	Validierung	270
10.2.3.3	Auswertung (Nachbereitung)	270
10.2.4	Top-down-Entwurfsdurchführung	271
10.2.4.1	Parallele Berechnung der Funktionen	272
10.2.4.2	Bit-Slices	278
10.3	12-Bit-Parallelregister und Register-Stack	281
10.3.1	Grundlagen	281
10.3.1.1	Flanken- und pegelgesteuerte Register	281
10.3.1.2	Stack	282

10.3.2	Aufgabenstellung	283
10.3.2.1	Register	283
10.3.2.2	Register-Stack	284
10.3.3	Bottom-up-Entwurfsdurchführung	285
10.3.3.1	Entwurfsdurchführung mit graphischem Editor . . .	286
10.3.3.2	Entwurfsdurchführung mit VHDL	288
10.3.3.3	Validierung	292
10.3.3.4	Auswertung (Nachbereitung)	292
10.3.4	Top-down-Entwurfsdurchführung	293
10.3.4.1	Entwurfsdurchführung	293
10.3.4.2	Validierung	294
10.4	Scan-Path-fähiges D-Flip-Flop	295
10.4.1	Grundlagen	295
10.4.1.1	Testfreundlicher Entwurf	295
10.4.1.2	Dynamische Taktung nach dem Master-Slave-Prinzip	297
10.4.1.3	Quasistatische CMOS-Logik	299
10.4.2	Bottom-up-Entwurfsdurchführung	300
10.4.3	Validierung	302
10.4.4	Auswertung (Nachbereitung)	303
10.5	12-Bit-Schieberegister	304
10.5.1	Grundlagen	304
10.5.2	Aufgabenstellung	306
10.5.3	Bottom-up-Entwurfsdurchführung	308
10.5.3.1	Entwurfsdurchführung mit graphischem Editor . . .	308
10.5.3.2	Entwurfsdurchführung mit VHDL	308
10.5.3.3	Validierung	310
10.5.3.4	Auswertung (Nachbereitung)	310
10.5.4	Top-down-Entwurfsdurchführung	311
10.5.4.1	Beschreibung von Schiebeoperationen in VHDL . . .	311

16 Inhalt

10.5.4.2	Entwurfsdurchführung	312
10.5.4.3	Validierung	312
10.6	Akkumulatoreinheit mit Status-Flags	313
10.6.1	Grundlagen	313
10.6.1.1	Statuseinheit	314
10.6.1.2	Operationen	316
10.6.1.3	Zeitverhalten (Timing)	318
10.6.2	Aufgabenstellung	319
10.6.3	Bottom-up-Entwurfsdurchführung	322
10.6.3.1	Entwurfsdurchführung	322
10.6.3.2	Validierung	323
10.6.3.3	Auswertung (Nachbereitung)	325
10.6.4	Top-down-Entwurfsdurchführung	328
10.7	Parallel ladbarer 12-Bit-Dualzähler	329
10.7.1	Grundlagen	329
10.7.2	Aufgabenstellung	331
10.7.3	Bottom-up-Entwurfsdurchführung	332
10.7.3.1	Entwurfsdurchführung	332
10.7.3.2	Validierung	334
10.7.3.3	Auswertung (Nachbereitung)	334
10.7.4	Top-down-Entwurfsdurchführung	334
10.7.4.1	Darstellung von Zahlen	334
10.7.4.2	Entwurfsdurchführung	335
10.7.4.3	Auswertung	336
10.8	Operationswerk	337
10.8.1	Grundlagen	337
10.8.1.1	Kontrollpfad	338
10.8.1.2	Datenpfad	338
10.8.2	Aufgabenstellung	339

10.8.3	Bottom-up-Entwurfsdurchführung	339
10.8.3.1	Vorgaben	339
10.8.3.2	Entwurfsdurchführung	345
10.8.3.3	Validierung	348
10.8.3.4	Auswertung (Nachbereitung)	349
10.8.4	Top-down-Entwurfsdurchführung	351
10.8.4.1	Entwurfsdurchführung	353
10.8.4.2	Validierung	355
10.9	PLA-Steuerwerk	356
10.9.1	Grundlagen	356
10.9.2	Aufgabenstellung	359
10.9.3	Bottom-up-Entwurfsdurchführung	361
10.9.3.1	Vorgaben	361
10.9.3.2	Entwurfsdurchführung	363
10.9.3.3	Validierung und Nachbereitung	369
10.9.4	Top-down-Entwurfsdurchführung	369
10.9.4.1	Möglichkeiten der Entwurfsdurchführung	369
10.9.4.2	Schnittstelle des Steuerwerks	370
10.9.4.3	Generierung der Automatenbeschreibung	371
10.9.4.4	Kodierung	377
10.9.4.5	Realisierung der Funktionen	378
10.9.4.6	Validierung	379
10.10	Mikroprozessorsystem als integrierte Schaltung	380
10.10.1	Grundlagen	380
10.10.2	Aufgabenstellung	380
10.10.3	Bottom-up-Entwurfsdurchführung	383
10.10.3.1	Entwurfsdurchführung	383
10.10.3.2	Validierung	384
10.10.3.3	Auswertung (Nachbereitung)	386

18 Inhalt

10.10.4	Top-down-Entwurfsdurchführung	386
10.10.4.1	Zielarchitekturen	387
10.10.4.2	Strukturelle Partitionierung	388
10.11	Verhaltensorientierte Validierung des Mikroprozessorsystems . .	391
10.11.1	Grundlagen	391
10.11.2	Aufgabenstellung	392
10.11.3	Top-down-Entwurfsdurchführung	393
10.11.3.1	Beschreibung auf der PMS-Ebene	393
10.11.3.2	Randbedingungen beim algorithmischen Entwurf . .	394
10.11.3.3	Beschreibung des RAM	395
10.11.3.4	Beschreibung der Ein- und Ausgabekomponenten . .	399
10.11.3.5	Modellierung des Mikroprozessors	400
10.11.4	Validierung	404
10.11.4.1	Assembler	404
10.11.4.2	Simulation	405
10.12	Physikalischer Entwurf des Mikroprozessor-ASICs	408
10.12.1	Grundlagen	408
10.12.2	Entwurfsdurchführung	408
10.12.2.1	Entwurfseingabe	408
10.12.2.2	Simulation	411
10.12.2.3	Plazierung und Verdrahtung	412
10.12.2.4	Nachbearbeitung	414
10.12.2.5	Fabrikationsvorbereitung	415
10.13	Test des gefertigten IC	417

Teil II: Literatur

419