

Informatik-Fachberichte

Herausgegeben von W. Brauer
im Auftrag der Gesellschaft für Informatik (GI)

82

Arndt Bode

Mikroarchitekturen
und Mikroprogrammierung:
Formale Beschreibung
und Optimierung



Springer-Verlag
Berlin Heidelberg New York Tokyo 1984

Autor

Arndt Bode

Institut für Mathematische Maschinen und Datenverarbeitung (III)

Universität Erlangen-Nürnberg

Martensstraße 3, 8520 Erlangen

CR Subject Classifications (1982): B.1.1, B.1.2, B.1.4, B.5.2, C.0, D.3.2, D.3.4

ISBN-13: 978-3-540-13380-3

e-ISBN-13: 978-3-642-69701-2

DOI: 10.1007/978-3-642-69701-2

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically those of translation, reprinting, re-use of illustrations, broadcasting, reproduction by photocopying machine or similar means, and storage in data banks. Further, storage or utilization of the described programmes on data processing installations is forbidden without the written permission of the author. Under § 54 of the German Copyright Law where copies are made for other than private use, a fee is payable to "Verwertungsgesellschaft Wort", Munich.

© by Springer-Verlag Berlin Heidelberg 1984

Vorwort und Danksagung

"Microprogramming is the implementation of hopefully reasonable systems through interpretation of unreasonable machines."
ROSIN, 1969

Vielleicht geht von dieser Definition der Mikroprogrammierung als Vermittlerin zwischen der von technologischen Randbedingungen diktierten "unwirtlichen" Hardware und einer möglichst benutzerfreundlichen Software-Schnittstelle die Faszination dieser Disziplin aus, die heute einen wichtigen Platz im Rahmen der Informatik einnimmt.

Die vorliegende Arbeit soll einen Beitrag dazu leisten, die Mikroprogrammierung theoretisch zu durchdringen und sie durch geeignete Verfahren letztendlich auch einem größeren Kreis von Informatikern zugänglich zu machen. Ausgehend von einem formalen Modell zur Beschreibung von Mikroarchitekturen werden Algorithmen entwickelt, die die Reduktion des Mikroprogrammspeicherbedarfs auf systematische Weise erlauben. Es wird auch gezeigt, daß diese Aufgabenstellung in der heutigen Praxis des (V)LSI-Entwurfes weitgehend durch ad-hoc Verfahren gelöst wird.

Den Schwerpunkt der Arbeit bilden jedoch Algorithmen zur automatischen Kompaktifizierung von Mikroprogrammen auf der Basis des eingeführten Modells. Diese Verfahren sollen die Implementierung und effiziente Übersetzung maschinenunabhängiger höherer Mikroprogrammiersprachen in Zukunft ermöglichen.

Meine Tätigkeit am Institut für Mathematische Maschinen und Datenverarbeitung (IMMD) III der Universität Erlangen-Nürnberg hat mir den Zugang zum gesamten Bereich der Technischen Informatik, speziell zur Rechnerarchitektur und Mikroprogrammierung eröffnet. Für diesen Zugang und die beständige Förderung, die ich durch ihn und seine Mitarbeiter und Mitarbeiterinnen erfahren habe, möchte ich zuallererst Herrn Professor Händler danken.

Die besonders anregende Atmosphäre am IMMD III ist nicht zuletzt auch auf die Durchführung realisierungsorientierter Projekte wie EGPA (Erlangen General Purpose Array, gefördert seit 1978 durch das BMFT) zurückzuführen, das durch die Initiative von Herrn Professor Händler, Herrn Professor Herzog, Herrn Professor Hofmann und Herrn Professor Schneider ermöglicht wurde. So ist auch eine wesentliche Anregung zu dieser Arbeit aus den Implementierungsaufgaben im Rahmen des EGPA-Teilprojektes "Mikroprogramme für die Vertikalverarbeitung und die Rechnerkommunikation" entstanden.

Für die Hinführung zur Informatik überhaupt möchte ich jedoch auch allen Mitgliedern des damaligen Instituts für Informatik und der späteren Fakultät für Informatik der Universität Karlsruhe, insbesondere meinem Doktorvater, Herrn Professor Alfred Schmitt, danken.

Manche Anregung zu dieser Arbeit ist der Diskussionsbereitschaft vieler, auch studentischer Mitarbeiter des IMMD und am EGPA-Projekt zu verdanken. Aus ihrem Kreis möchte ich nur Frau Barbara Albert erwähnen, mit der ich während mancher Stunde Probleme der Mikroprogrammierung gewälzt habe.

Schließlich möchte ich auch nicht versäumen, Frau Petra Wagner für die tadellose Erstellung des maschinengeschriebenen Manuskriptes zu danken.

Erlangen, September 1983

Arndt Bode

Inhalt

	Seite
<u>1. Motivation und Übersicht</u>	7
<u>2. Mikroprogrammierung, eine verbale und eine formale</u>	12
<u>Einführung</u>	
2.1 Mikroprogrammierung, eine verbale Einführung	12
2.1.1 Mikroprogrammierung: eine Realisierung des Leitwerkes von Rechnern	12
2.1.2 Mikroinstruktion, Mikrooperation und ihre Ansteuerung	15
2.1.3 Kodierung der Steuerinformation: Parallelismus auf der Ebene der Mikroinstruktion	18
2.1.4 Der Mikroinstruktionszyklus, Pipelining auf Mikroinstruktionsebene	23
2.1.5 Mikroprogrammierbarkeit	31
2.1.6 Mikroprogrammierung und vertikale Verlagerung	32
2.2 Formale Beschreibung der Mikroprogrammierung: das Tupelmodell	37
2.2.1 Beispiel - Rechner - Bausteine	39
2.2.2 Das Tupelmodell	43
2.2.3 Relationen zwischen Mikrooperationen und Mikrooperations-Bündeln	57
2.2.3.1 Datenabhängigkeit	58
2.2.3.2 Ressourcen-Konflikte	72
<u>3. Reduktion der Wortlänge von Mikroinstruktionen</u>	75
3.1 Motivation und Problemstellung	75
3.2 Varianten der Kodierung der Steuerinformation	77
3.2.1 Mikroprogrammierte Rechner	78
3.2.1.1 Vollständige Kodierung der Mikroinstruktion	78
3.2.1.2 Keine Kodierung der Mikroinstruktion	80
3.2.1.3 Minimale Kodierung der Mikroinstruktion	81
3.2.2 Mikroprogrammierbare und dynamisch mikropro- grammierbare Rechner	83

	Seite	
3.3	Bestimmung der minimalen Kodierung von Mikroinstruktionen ("Optimale Verzorgung")	85
3.3.1	Lösungen im Rahmen der Schaltkreistheorie	92
3.3.2	Lösungen im Rahmen der linearen Programmierung	101
3.4	Pragmatische Ansätze zur Wortlängenreduktion	108
3.4.1	Grundlegende Verfahren zur Wortlängenreduktion	108
3.4.2	Verfahren zur Reduktion des Mikroprogramm- speicherplatzes für VLSI-Bausteine	115
3.4.3	Ein Verfahren zur Reduktion der Zustände von Leitwerken	122
<u>4.</u>	<u>Reduktion der Anzahl von Mikroinstruktionen in Mikro-</u> <u>programmen: lokale und globale Kompaktifizierung</u>	125
4.1	Motivation und Problemstellung	125
4.2	Lokale Kompaktifizierung	132
4.2.1	Branch and bound-Algorithmus	133
4.2.2	Branch and bound-Algorithmus mit Heuristiken	144
4.2.3	Kritischer Pfad Algorithmus	148
4.2.4	First-come-first-served (FCFS-)Algorithmus	155
4.2.5	Wertung	159
4.3	Globale Kompaktifizierung	160
4.3.1	Blockorientierte globale Kompaktifizierung	162
4.3.2	Pfadorientierte globale Kompaktifizierung	168
4.4	Höhere Mikroprogrammiersprachen, Problemstellung	186
4.4.1	Mikroprogrammiersprachen, Übersicht	186
4.4.2	Offene Probleme höherer Mikroprogrammiersprachen	195
4.4.2.1	Binden der Ressourcen von Mikroarchitekturen	195
4.4.2.2	Registerzuteilung, Speicherzugriffe	197
4.4.2.3	Umbenennen von Variablen	199
4.4.2.4	Phasenkopplung	200
<u>5.</u>	<u>Ausblick</u>	203
<u>6.</u>	<u>Literatur</u>	206
	<u>Stichwortverzeichnis</u>	217
	<u>Abkürzungsverzeichnis</u>	224