

Informatik – Fachberichte

Band 66: Applications and Theory of Petri Nets. Proceedings, 1982. Edited by G. Rozenberg. VI, 315 pages. 1983.

Band 67: Data Networks with Satellites. GI/NTG Working Conference, Cologne, September 1982. Edited by J. Majus and O. Spaniol. VI, 251 pages. 1983.

Band 68: B. Kutzler, F. Lichtenberger, Bibliography on Abstract Data Types. V, 194 Seiten. 1983.

Band 69: Betrieb von DN-Systemen in der Zukunft. GI-Fachgespräch, Tübingen, März 1983. Herausgegeben von M. A. Graef. VIII, 343 Seiten. 1983.

Band 70: W. E. Fischer, Datenbanksystem für CAD-Arbeitsplätze. VII, 222 Seiten. 1983.

Band 71: First European Simulation Congress ESC 83. Proceedings, 1983. Edited by W. Ameling. XII, 653 pages. 1983.

Band 72: Sprachen für Datenbanken. GI-Jahrestagung, Hamburg, Oktober 1983. Herausgegeben von J. W. Schmidt. VII, 237 Seiten. 1983.

Band 73: GI-13. Jahrestagung, Hamburg, Oktober 1983. Proceedings. Herausgegeben von J. Kupka. VIII, 502 Seiten. 1983.

Band 74: Requirements Engineering. Arbeitstagung der GI, 1983. Herausgegeben von G. Hommel und D. Krönig. VIII, 247 Seiten. 1983.

Band 75: K. R. Dittrich, Ein universelles Konzept zum flexiblen Informationsschutz in und mit Rechensystemen. VIII, 246 pages. 1983.

Band 76: GWAI-83. German Workshop on Artificial Intelligence. September 1983. Herausgegeben von B. Neumann. VI, 240 Seiten. 1983.

Band 77: Programmiersprachen und Programmentwicklung. 8. Fachtagung der GI, Zürich, März 1984. Herausgegeben von U. Ammann. VIII, 239 Seiten. 1984.

Band 78: Architektur und Betrieb von Rechensystemen. 8. GI-NTG-Fachtagung, Karlsruhe, März 1984. Herausgegeben von H. Wettstein. IX, 391 Seiten. 1984.

Band 79: Programmierungsumgebungen: Entwicklungswerkzeuge und Programmiersprachen. Herausgegeben von W. Sammer und W. Remmele. VIII, 236 Seiten. 1984.

Band 80: Neue Informationstechnologien und Verwaltung. Proceedings, 1983. Herausgegeben von R. Traunmüller, H. Fiedler, K. Grimmer und H. Reiner mann. XI, 402 Seiten. 1984.

Band 81: Koordinaten von Informationen. Proceedings, 1983. Herausgegeben von R. Kuhlen. VI, 366 Seiten. 1984.

Band 82: A. Bode, Mikroarchitekturen und Mikroprogrammierung: Formale Beschreibung und Optimierung, 6, 7-227 Seiten. 1984.

Band 83: Software-Fehlertoleranz und -Zuverlässigkeit. Herausgegeben von F. Belli, S. Pfleger und M. Seifert. VII, 297 Seiten. 1984.

Band 84: Fehlertolerierende Rechensysteme. 2. GI/NTG/GMR-Fachtagung, Bonn 1984. Herausgegeben von K.-E. Großpietsch und M. Dal Cin. X, 433 Seiten. 1984.

Band 85: Simulationstechnik. Proceedings, 1984. Herausgegeben von F. Breitenacker und W. Kleinert. XII, 676 Seiten. 1984.

Band 86: Prozeßrechner 1984. 4. GI/GMR/KfK-Fachtagung, Karlsruhe, September 1984. Herausgegeben von H. Trauboth und A. Jaeschke. XII, 710 Seiten. 1984.

Band 87: Mustererkennung 1984. Proceedings, 1984. Herausgegeben von W. Kropatsch. IX, 351 Seiten. 1984.

Band 88: GI-14. Jahrestagung, Braunschweig, Oktober 1984. Proceedings. Herausgegeben von H.-D. Ehrich. IX, 451 Seiten. 1984.

Band 89: Fachgespräche auf der 14. GI-Jahrestagung. Braunschweig, Oktober 1984. Herausgegeben von H.-D. Ehrich. V, 267 Seiten. 1984.

Band 90: Informatik als Herausforderung an Schule und Ausbildung. GI-Fachtagung, Berlin, Oktober 1984. Herausgegeben von W. Arit und K. Haefner. X, 416 Seiten. 1984.

Band 91: H. Stoyan, Maschinen-unabhängige Code-Erzeugung als semantikerhaltende beweisbare Programmtransformation. IV, 365 Seiten. 1984.

Band 92: Offene Multifunktionale Büroarbeitsplätze. Proceedings, 1984. Herausgegeben von F. Krückeberg, S. Schindler und O. Spaniol. VI, 335 Seiten. 1985.

Band 93: Künstliche Intelligenz. Frühjahrsschule Dassel, März 1984. Herausgegeben von C. Habel. VII, 320 Seiten. 1985.

Band 94: Datenbank-Systeme für Büro, Technik und Wirtschaft. Proceedings, 1985. Herausgegeben von A. Blaser und P. Pistor. X, 519 Seiten. 1985

Band 95: Kommunikation in Verteilten Systemen I. GI-NTG-Fachtagung, Karlsruhe, März 1985. Herausgegeben von D. Heger, G. Krüger, O. Spaniol und W. Zorn. IX, 691 Seiten. 1985.

Band 96: Organisation und Betrieb der Informationsverarbeitung. Proceedings, 1985. Herausgegeben von W. Dirlwanger. XI, 261 Seiten. 1985.

Band 97: H. Willmer, Systematische Software-Qualitätssicherung anhand von Qualitäts- und Produktmodellen. VII, 162 Seiten. 1985.

Band 98: Öffentliche Verwaltung und Informationstechnik. Neue Möglichkeiten, neue Probleme, neue Perspektiven. Proceedings, 1984. Herausgegeben von H. Reiner mann, H. Fiedler, K. Grimmer, K. Lenk und R. Traunmüller. X, 396 Seiten. 1985.

Band 99: K. Küspert, Fehlererkennung und Fehlerbehandlung in Speicherungsstrukturen von Datenbanksystemen. IX, 294 Seiten. 1985.

Band 100: W. Lamersdorf, Semantische Repräsentation komplexer Objektstrukturen. IX, 187 Seiten. 1985.

Band 101: J. Koch, Relationale Anfragen. VIII, 147 Seiten. 1985.

Band 102: H.-J. Appelrath, Von Datenbanken zu Expertensystemen. VI, 159 Seiten. 1985.

Band 103: GWAI-84. 8th German Workshop on Artificial Intelligence. Wingst/Stade, October 1984. Edited by J. Laubsch. VIII, 282 Seiten. 1985.

Band 104: G. Sagerer, Darstellung und Nutzung von Expertenwissen für ein Bildanalysesystem. XIII, 270 Seiten. 1985.

Band 105: G. E. Maier, Exceptionbehandlung und Synchronisation. IV, 359 Seiten. 1985.

Band 106: Österreichische Artificial Intelligence Tagung. Wien, September 1985. Herausgegeben von H. Trost und J. Rettl. VIII, 211 Seiten. 1985.

Band 107: Mustererkennung 1985. Proceedings, 1985. Herausgegeben von H. Niemann. XIII, 338 Seiten. 1985.

Band 108: GI/OCG/ÖGJ-Jahrestagung 1985. Wien, September 1985. Herausgegeben von H. R. Hansen. XVII, 1086 Seiten. 1985.

Band 109: Simulationstechnik. Proceedings, 1985. Herausgegeben von D. P. F. Möller. XIV, 539 Seiten. 1985.

Band 110: Messung, Modellierung und Bewertung von Rechensystemen. 3. GI/NTG-Fachtagung, Dortmund, Oktober 1985. Herausgegeben von H. Beilner. X, 389 Seiten. 1985.

Informatik-Fachberichte 146

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

Werner Damm

Entwurf und Verifikation mikroprogrammierter Rechnerarchitekturen



Springer-Verlag
Berlin Heidelberg New York
London Paris Tokyo

Autor

Werner Damm
Fachbereich 10, Praktische Informatik
Universität Oldenburg
Ammerländer Heerstraße 114-118, D-2900 Oldenburg

CR Subject Classifications (1987): B.1.2, B.1.4, D.2.10, F.3.1

ISBN 978-3-540-18320-4 ISBN 978-3-642-51137-0 (eBook)
DOI 10.1007/978-3-642-51137-0

CIP-Kurztitelaufnahme der Deutschen Bibliothek. Damm, Werner: Entwurf und Verifikation mikroprogrammierter Rechnerarchitekturen / Werner Damm. – Berlin; Heidelberg; New York; Tokyo: Springer, 1987
(Informatik-Fachberichte; 146)
ISBN 978-3-540-18320-4

NE: GT

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 Fassung vom 24. Juni 1985 zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

© by Springer-Verlag Berlin Heidelberg 1987

VORWORT

Die Mikroprogrammierung hat im Entwurf von Rechnerarchitekturen einen festen Platz als vermittelnde Ebene zwischen Software und Hardware. Aus ihrer ursprünglichen Rolle in der Realisierung von Leitwerken von Rechenanlagen hat sie sich weiterentwickelt zu einem flexiblen Werkzeug, dessen heutige Bedeutung ebenso stark in der Entwicklung von Spezialarchitekturen liegt, wo sie eine effiziente Möglichkeit der Realisierung von komplexen auf die jeweilige Anwendung angepaßten Spezialinstruktionen bietet. Heutige Anwendungsbeispiele reichen von der Entwicklung von Prolog-Architekturen bis zur Anwendung in der Bildverarbeitung.

Trotz dieser etablierten Stellung der Mikroprogrammierung hat die Entwicklung von Unterstützungswerkzeugen mit der zunehmenden Anwendungsbreite nicht Schritt gehalten. Die extreme Nähe zur Hardware gestattet zwar einerseits eine sehr effiziente Implementierung; andererseits steht diese Hardwarenähe der Entwicklung von Unterstützungswerkzeugen auf Grund der Komplexität der hardwarenahen Programmierung entgegen.

Das vorliegende Buch stellt einen Ansatz zur Entwicklung und Verifikation von Mikroprogrammen vor, der diese Lücke überbrücken soll. Mit der Einführung einer auf diesen Bereich abgestimmten Rechnerbeschreibungssprache und einem Ansatz, in höhere Mikroprogrammiersprachen systematisch gerade so viele hardwarenahe Aspekte einzubetten, daß eine Übersetzung in effizienten Mikrocode möglich ist, wird die Brücke zu einer Firmwareentwicklung auf einem aus dem Software Engineering bekannten Niveau geschlagen, ohne die Effizienzvorteile der Mikroprogrammierung einzubüßen. Die sich daraus ergebende hierarchische Methode zur Firmwareentwicklung wird ausführlich an Hand einer konkreten Fallstudie vorgestellt.

Neben diesem praktischen Aspekt stellt die vorliegende Arbeit die Relevanz von formalen Methoden der Informatik in diesem konkreten Anwendungsgebiet der Mikroprogrammierung heraus.

Dazu wird auf die für die Mikroprogrammierung besonders brisante Frage der Fehlerfreiheit der Mikroprogramme eingegangen. Die besondere Hardwarenähe bedingt eine zusätzliche Komplexität dieser Ebene der Programmierung, die leicht zu Kodierfehlern führen kann. Andererseits müssen gerade an Mikroprogramme, die auf einer der innersten Systemebenen stehen, hohe Zuverlässigkeitsanforderungen gestellt werden. Diese können letztlich nur durch formale Verifikationsverfahren erfüllt werden, wie sie im zweiten Teil dieser Arbeit vorgestellt werden. Besonderes Gewicht wurde darauf gelegt, die Verifikation während des Entwurfs zu ermöglichen, so daß unmittelbar das Wissen des Mikroprogrammierers auch im Verifikationsprozeß ausgenutzt werden kann.

Die Entwicklung dieser Verifikationsmethoden erfordert fortgeschrittene formale Begriffsbildungen, die auf diesen formalen Ebenen die der Mikroprogrammierung innewohnende Komplexität transparent machen. Trotzdem erweist sich die entwickelte Theorie als so prägnant, daß die Frage der Fehlerfreiheit des Mikrocodes vollständig in der Axiomatisierung erfaßt werden kann. Der zweite Teil dieser Arbeit ist daher auch als Studie gedacht, wie in vergleichbaren Anwendungssituationen Methoden der theoretischen Informatik erfolgreich zur Lösung konkreter Anwendungsprobleme eingesetzt werden können.

Die Mikroprogrammierung stellt - von einem weiteren Standpunkt aus gesehen - letztlich nur eine Implementierungsebene im Entwurf von Rechnerarchitekturen dar. Mit der Komplexität zur Zeit entwickelter 5th Generation Computer stellt sich die Frage nach Entwurfsunterstützung im zunehmenden Maße für den gesamten Rechnerentwurfsprozeß. In der Tat werden 70 % der Entwicklungszeit heute in den höheren Ebenen des Rechnerentwurfs eingesetzt, in denen CAD-Werkzeuge fast gänzlich fehlen. Die vorliegende Arbeit kann in diesem Kontext als ein Schritt in Richtung einer Entwicklungsumgebung angesehen werden, die auf die Unterstützung des gesamten Entwurfsprozesses von Rechnersystemen abzielt. Erst damit wird Rechnerarchitektur von einer "craft" zu einer "art".

INHALTSVERZEICHNIS

1	Einleitung	1
2	Grundbegriffe der Firmwareverifikation	7
2.1	Ebenen einer Rechnerarchitektur	7
2.2	Mikroprogrammierung	8
2.3	Mikroprogrammierte Rechnerarchitekturen	13
2.4	Grundlagen der axiomatischen Verifikation von Firmware	20
3	Entwurf mikroprogrammierter Rechnerarchitekturen	26
3.1	Formale Beschreibung von Rechnerarchitekturen	26
3.1.1	Grundkonzepte von AADL	27
3.1.2	Formale Beschreibung von Mikroarchitekturen	43
3.1.3	Spezifikation von Zielarchitekturen	50
3.2	Die S*-Familie höherer Mikroprogrammiersprachen	59
3.3	Hierarchischer Entwurf von Rechnerarchitekturen	67
3.3.1	Grundprinzipien des hierarchischen Entwurfs mikroprogrammierter Rechnerarchitekturen	67
3.3.2	Die Speicherabbildung	75
3.3.3	Modifikation der hierarchischen Entwurfsmethode	79
4	Verifikation mikroprogrammierter Rechnerarchitekturen	86
4.1	Die Generierung der axiomatischen Spezifikation einer Operation	88
4.1.1	Die schwächste ableitbare Vorbedingung einer Operation	89
4.1.2	Die axiomatische Spezifikation einer Operation	108
4.1.3	Die Einbindung der axiomatischen Spezifikation als Anforderung an die Implementierung	119

VIII

4.2	Eine axiomatische Definition der S*-Familie	137
4.2.1	Axiomatisierung von Mikrooperationen	138
4.2.2	Konfliktanalyse in Mikroarchitekturen und ihre Einbettung in Beweisregeln für die parallele Ausführung von Mikrooperationen	146
4.2.3	Axiomatisierung prozessorabhängiger Kontrollstrukturen von S*	175
4.2.4	Axiomatisierung prozessorunabhängiger Kontrollstrukturen und der Dokumentationsbefehle von S*	184
	Zusammenfassung	192
	Danksagung	193
	Fußnoten	194
	Literaturverzeichnis	205
A1	Anhang 1	
A1.1	Spezifikation der Makroarchitektur der NOVA 1200	216
A1.2	Formale Beschreibung der Mikroarchitektur der MICRODATA 1600 ...	235
A1.3	Definition der Zwischenarchitektur	253
A2	Anhang 2	
	Die Syntax von S*	259
A3	Anhang 3	
	Konfliktanalyse zwischen dynamischen Speicherausdrücken	265
A4	Anhang 4 : Ein Beispielbeweis	
A4.1	Diskussion des Beweises	273
A4.2	Schematische Darstellung des Beweises	280
A4.3	Berechnung der schwächsten Vorbedingung	286
A4.4	Einige Vereinfachungsregeln	309
	Stichwortverzeichnis	313
	Verzeichnis der Abbildungen	326