

Entwurf und interaktive Hardware-/Software-Partitionierung komplexer heterogener Systeme

Vom Fachbereich 18
der Technischen Universität Darmstadt
zur Erlangung der Würde eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Dissertation

Dipl.-Ing.
Thomas Hollstein
geboren am 30. Januar 1966

Referent:	Prof. Dr. Dr. h.c. mult. M. Glesner
Korreferenten:	Prof. Dr. H. Eveking Prof. Dr. R. Ubar
Tag der Einreichung:	25. Januar 2000
Tag der mündlichen Prüfung:	30. Juni 2000

D17

Darmstädter Dissertationen

Berichte aus der Elektrotechnik

Thomas Hollstein

**Entwurf und interaktive Hardware-/Software-
Partitionierung komplexer heterogener Systeme**

D 17 (Diss. TU Darmstadt)

Shaker Verlag
Aachen 2001

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Hollstein, Thomas:

Entwurf und interaktive Hardware-/Software-Partitionierung komplexer heterogener Systeme/Thomas Hollstein.

Aachen : Shaker, 2001

(Berichte aus der Elektrotechnik)

Zugl.: Darmstadt, Techn. Univ., Diss., 2001

ISBN 3-8265-8956-4

Copyright Shaker Verlag 2001

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 3-8265-8956-4

ISSN 0945-0718

Shaker Verlag GmbH • Postfach 1290 • 52013 Aachen

Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9

Internet: www.shaker.de • eMail: info@shaker.de

Abstract

Increasing integration capabilities of advanced semiconductor technologies combined with the demand of increasing functionality within modern “informed” products from several fields of applications define real challenges for future microelectronic system design. Furthermore there is a trend for ubiquitous computing with local system integration of computational power into the application system (embedded microelectronic systems), which delivers a demand for highly integrated microelectronic components. Application fields like mobile computing, mobile communication, automotive and avionics applications, intelligent embedded industrial control systems and advanced medical devices can be regarded among others. Therefore the on-chip integration (“System-on-Chip (SoC)”) of heterogeneous microelectronic components becomes a topic of increasing importance. In future, the components to be integrated on a “System-on-Chip” can be microprocessors, custom digital circuits, memory and interface modules, analog and radio frequency components as well as advanced analog/digital and digital/analog converters. Hardware/software co-design is a research area, which focuses the design of integrated mixed hardware/software systems.

One of the key issues in hardware/software co-design is an efficient hardware/software partitioning of a given system specification. The partitioning problem is very complex, since during the partitioning progress a certain amount of constraints has to be considered as well as a more or less complex cost function. Existing approaches suffer from a lack of controllability (black box behaviour) or don’t provide advanced algorithmic support at all.

In this thesis, a new semi-interactive system-level multistage hardware/software partitioning method (HiPART) is presented, which is part of the DICE co-design environment (**D**armstadt **I**nteractive **C**o-Design of **E**mbedded **S**ystems). Based on a VHDL/C system co-specification on behavioral (process) level, a common concurrent control/data flow graph is constructed. The design frontend allows several code transformations in order to prepare the specification for the HW/SW co-synthesis process. The partitioning method contains four phases, which can be controlled in a transparent way by an advanced graphical user interface (GUI): Pre-clustering (realising a user-determined partitioning granularity), clustering (coarsening of the partitioning graph), partitioning, refinement. The clustering phase guarantees the scalability of the method to applications of very different sizes. For the central partitioning steps two alternative algorithms have been implemented, which are capable to provide excellent results with respect to the cost function: A new dynamic and adaptive version of *Simulated Annealing* and *Tabu Search*. The capabilities of this new partitioning methodology are proven by application to dedicated application examples.

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Fachgebiet “Mikroelektronische Systeme” des Instituts für Datentechnik der Technischen Universität Darmstadt.

Mein besonderer Dank gilt Herrn Prof. Dr. Dr. h.c. mult. Manfred Glesner, der diese Arbeit ermöglicht hat und seiner Unterstützung mit wertvollen Anregungen in zahlreichen Diskussionen.

Herrn Prof. Dr. Hans Evekling (TU Darmstadt) und Herrn Prof. Dr. Raimund Ubar (Technische Universität Tallin, Estland) danke ich für die freundliche Übernahme des Korreferats und das Interesse an der vorliegenden Arbeit.

Weiterhin gilt mein Dank allen Mitarbeiterinnen und Mitarbeitern des Fachgebiets “Mikroelektronische Systeme”, insbesondere meinen Kollegen der CAD-Gruppe, Herrn Dr. Michael Gasteier, Herrn Dr. Michael Münch und Herrn Dr. Andreas Kirschbaum, welche mir in vielen Diskussionen und mit konstruktiver Kritik wertvolle Anregungen und Unterstützung gaben. Ganz besonders danken möchte ich auch Herrn Dr. Wolfgang Ecker (Siemens AG/Infineon AG München) für viele Anregungen und die wertvolle Zusammenarbeit im Bereich der Hardwareestimation. Herrn Prof. Dr. Norbert Wehn (Universität Kaiserslautern) gilt mein Dank für entscheidende Anregungen in der Initialphase der Dissertation. Weiterhin gilt mein Dank meinem Kollegen Herrn Dr. Jürgen Becker für konstruktive Diskussionen in der Endphase der Arbeit. Zusätzlich danke ich allen Studenten, die im Rahmen ihrer Studien- und Diplomarbeiten zum Gelingen dieser Arbeit beigetragen haben.

Ferner danke ich der Siemens AG München für die Bereitstellung des Programms VOTAN. Letzteres hat zu einer nicht unerheblichen Reduktion des ohnehin umfangreichen Gesamtimplementierungsaufwands der in dieser Arbeit vorgestellten Entwurfsumgebung beigetragen.

Nicht zuletzt danke ich auch meiner Familie für ihr Verständnis und ihre Unterstützung während der Anfertigung dieser Arbeit.

Thomas Hollstein

Darmstadt, den 30. Juni 2000

Inhaltsverzeichnis

1	Einführung	1
1.1	Aktuelle Trends der Mikroelektronik	1
1.2	Halbleitertechnologie als Schrittmacher für die Systementwicklung	3
1.3	Herausforderungen an computergestützte Entwurfsverfahren	4
1.4	Motivation für den integrierten Entwurf von Hardware und Software	6
1.5	Übersicht: Aufgabenstellung und Ziele der vorliegenden Arbeit	7
2	Methoden des Systementwurfs	9
2.1	Entwurf auf Systemebene	9
2.2	Eingebettete Systeme	12
2.3	Anwendungsspezifische integrierte Schaltungen	13
2.4	Entwurfsprozeß für digitale Systeme	14
2.5	Systemspezifikation und Modellbildung	14
2.5.1	Strukturmodelle	16
2.5.2	Zustandsmodelle	16
2.5.3	Datenflußmodelle	18
2.5.4	Hardwarebeschreibungssprachen	18
2.6	Systemvalidierung	21
2.7	Syntheseschritte auf Systemebene	22
2.8	Hardwaresynthese und optimierende Transformationen	23
2.8.1	High-Level-Synthese	23
2.8.2	Register-Transfer- und Logiksynthese	24
2.9	Testen digitaler Hardwarekomponenten	24
2.10	Softwaresynthese	25
3	Hardware/Software Co-Design	27
3.1	Einführung	27

3.2	Aufgabenstellungen: Definition und Klassifikation	28
3.3	Entwurfsablauf	30
3.4	Teilprobleme	31
3.4.1	Spezifikation	31
3.4.2	Funktionale Validierung	31
3.4.3	HW-/SW-Partitionierung	32
3.4.4	Kommunikationssynthese	34
3.5	Vorstellung und Vergleich existierender Entwurfsumgebungen	34
3.5.1	Entwurfsumgebungen und Methodologien	34
3.5.2	Zusammenfassende Bewertung	49
4	DICE: Eine integrierte Hardware-/Software-Entwurfsumgebung	51
4.1	Entwurfsmethodik und Zielapplikationen	51
4.2	Entwurfsablauf	53
4.3	Modellierung gemischter HW-/SW-Systeme	55
4.3.1	Systemmodell	55
4.3.2	Modellarchitektur	55
4.3.3	Restriktionen	57
4.3.4	Zeitmodellierung	57
4.3.5	Prozeßkommunikation	58
4.4	Co-Simulation durch Simulatorkopplung	59
4.5	DICE Frontend-Tools	60
4.6	Paralleler CDFG und interaktive Benutzerschnittstelle	64
4.7	HW-/SW-Partitionierung mit HiPART	64
4.8	Kommunikationssynthese	65
4.8.1	Minimierung dezidierter Verbindungen	66
4.8.2	Generierung von Bustopologien	67
4.9	Systemimplementierung und Rapid Prototyping	68
4.9.1	Prototyping-Hardware	68
4.9.2	Synthese der Routing-Konfiguration	69
4.9.3	Hardware-Monitor	69
5	Schätzung der Entwurfsqualität	71
5.1	Einführung	71
5.2	Schätzung von Hardwarekosten	72

5.2.1	Kostenfaktoren	73
5.2.2	Zielarchitektur	73
5.2.3	Performanzfaktoren	74
5.2.4	Chipfläche	78
5.3	Schätzung von Softwarekosten	81
5.3.1	Profiling	82
5.3.2	Ausführungszeit	82
5.3.3	Programmspeicher	82
5.3.4	Datenspeicher	82
5.3.5	Gängige Estimationstechniken in Syntheseumgebungen	83
5.4	Grundlagen des in DICE verwirklichten Estimationskonzeptes	86
5.5	Schätzung von HW-Kosten in DICE	87
5.5.1	Zielarchitektur	87
5.5.2	SPIDER: Automatische Erstellung einer Datenbasis	88
5.5.3	Profiling	88
5.5.4	ESTIMA: CDFG-basierte Schätzung von Hardwarekosten	89
5.6	Schätzung von SW-Kosten in DICE	94
6	Partitionierungsverfahren	95
6.1	Einführung	95
6.2	Systempartitionierung: Aufgabenstellungen	96
6.3	Wahl einer geeigneten Spezifikationsebene	96
6.3.1	Partitionierung auf verschiedenen Abstraktionsebenen	98
6.3.2	Partitionierung abstrakter anwendungsorientierter Spezifikationen	98
6.3.3	Funktionale Partitionierung verhaltensorientierter Spezifikationen	98
6.3.4	Implementierungsnahe Spezifikationen	99
6.4	Komplexität	100
6.5	Klassifizierung von Partitionierungsalgorithmen	100
6.6	Partitionierungsalgorithmen	102
6.6.1	Clusterverfahren	102
6.6.2	Gruppenmigrationsverfahren	107
6.6.3	Eigenvektorverfahren	114
6.6.4	Lineare Optimierung	117
6.6.5	Branch-and-Bound-Verfahren	119
6.6.6	Tabu Search	122

6.6.7	Simulated Annealing	124
6.6.8	Genetische Algorithmen	130
6.6.9	Vergleich	131
7	HiPART: Ein neuer semiinteraktiver Partitionierungsansatz	133
7.1	HiPART: Einführung	133
7.2	HiPART: Definition des Partitionierungsproblems	134
7.3	Wahl der Granularität	135
7.4	Interaktives Konzept	136
7.5	Ablauf der Partitionierung	137
7.6	Pre-Clustering	137
7.6.1	Ablauf des Pre-Clustering	140
7.6.2	Berechnung von Datenabhängigkeiten	141
7.6.3	Laufzeitanalyse	145
7.7	Clustering	146
7.7.1	Problemdefinition	147
7.7.2	Verwendete Nähefunktion	149
7.7.3	Verwendeter Algorithmus	152
7.7.4	Parametereinstellungen	152
7.7.5	Laufzeitanalyse	153
7.8	Partitionierung	154
7.8.1	Problemdefinition	155
7.8.2	Kostenfunktion	156
7.8.3	Partitionierungsvorgang	163
7.8.4	Laufzeitanalyse	167
7.9	Optimierung mit verfeinerter Granularität	167
7.9.1	Optimierungsvorgang	168
7.9.2	Laufzeitanalyse	172
7.10	Alternatives Partitionierungsverfahren: Tabu Search	172
7.11	Anwendungsbeispiele und Ergebnisse	178
7.11.1	Verwendete Testspezifikationen	178
7.11.2	Konvergenz der Partitionierungsalgorithmen	180
7.11.3	Laufzeit und Partitionierungsergebnisse	184
7.12	Parametereinstellungen: S.A. und Tabu Search	188

8 Codemigration: C \leftrightarrow VHDL	193
8.1 Übersetzung von C nach VHDL	193
8.1.1 Globalisierung lokaler Variablen	194
8.1.2 Einfügen von Schnittstellenoperationen	197
8.1.3 Übersetzung nach VHDL	197
8.1.4 Ergebnisse	199
8.1.5 Sprachspezifische Eigenschaften und Restriktionen	200
8.2 Übersetzung von VHDL nach C	209
9 Zusammenfassung und Ausblick	211
9.1 Ergebnisse dieser Arbeit	211
9.2 Ausblick	213
A Verzeichnis der verwendeten Abkürzungen	215
B Verzeichnis der verwendeten Bezeichnungen	217
C Beschreibung der CST-Datenstrukturen	221
D Beschreibung der CCDFG-Datenstrukturen	231
Abbildungsverzeichnis	237
Tabellenverzeichnis	241
Literaturverzeichnis	243
Eigene Veröffentlichungen	255
Betreute Studien- und Diplomarbeiten	259