

Zabel

Techniken zur Simulation von eingebetteten Systemen mit abstrakten RTOS-Modellen



C-LAB Publication

Herausgegeben von
Published by

Dr. Wolfgang Kern, Siemens AG
Prof. Dr. Franz-Josef Rammig, Universität Paderborn

Das C-LAB - Cooperative Computing & Communication Laboratory - leistet Forschungs- und Entwicklungsarbeiten und gewährleistet deren Transfer an den Markt. Es wurde 1985 von den Partnern Nixdorf Computer AG (nun Siemens AG) und der Universität Paderborn im Einvernehmen mit dem Land Nordrhein-Westfalen gegründet.

Die Vision, die dem C-LAB zugrunde liegt, geht davon aus, daß die gewaltigen Herausforderungen beim Übergang in die kommende Informations- und Wissensgesellschaft nur durch globale Kooperation und in tiefer Verzahnung von Theorie und Praxis gelöst werden können. Im C-LAB arbeiten deshalb Mitarbeiter von Hochschule und Industrie unter einem Dach in einer gemeinsamen Organisation an gemeinsamen Projekten mit internationalen Partnern eng zusammen.

C-LAB - the Cooperative Computing & Cooperation Laboratory - works in the area of research and development and safeguards its transfer into the market. It was founded in 1985 by Nixdorf Computer AG (now Siemens AG) and the University of Paderborn under the auspices of the State of North-Rhine Westphalia.

C-LAB's vision is based on the fundamental premise that the gargantuan challenges thrown up by the transition to a future information and knowledge society can only be met through global cooperation and deep interworking of theory and practice. This is why, under one roof, staff from the university and from industry cooperate closely on joint projects within a common research and development organization together with international partners. In doing so, C-LAB concentrates on those innovative subject areas in which cooperation is expected to bear particular fruit for the partners and their general well-being.

C-LAB Publication

Band 32

Henning Zabel

**Techniken zur Simulation
von eingebetteten Systemen mit
abstrakten RTOS-Modellen**

D 466 (Diss. Universität Paderborn)

Shaker Verlag
Aachen 2010

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Zugl.: Paderborn, Univ., Diss., 2010

Copyright Shaker Verlag 2010

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

Printed in Germany.

ISBN 978-3-8322-9092-4

ISSN 1438-3527

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen

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

Internet: www.shaker.de • E-Mail: info@shaker.de

Techniken zur Simulation von eingebetteten Systemen mit abstrakten RTOS-Modellen

Dissertation

**Schriftliche Arbeit eingereicht bei der
Fakultät für Elektrotechnik, Informatik und Mathematik
der Universität Paderborn zur Erlangung des Grades
Dr. rer. nat.**

von

Henning Zabel

Paderborn

Gutachter:

Prof. Dr. Franz Rammig, Universität Paderborn

Prof. Dr. Andreas Gerstlauer, University of Texas at Austin

Datum der mündlichen Prüfung: 19.03.2010.

Kurzbeschreibung

Die stetig wachsende Rechenleistung von Prozessoren und die höhere Integrationsdichte von Schaltkreisen erlauben die Realisierung von immer mehr und komplexeren Funktionen in immer kleineren und leistungsfähigeren Komponenten. Die gesteigerte Komplexität ist eine Herausforderung an den Entwurf elektronischer Systeme. Das System-Level-Design strukturiert den Entwurf, indem das System, ausgehend von einem abstrakten Applikationsmodell, schrittweise in eine detaillierte Beschreibung eines Schaltkreises oder einer Softwarekomponente überführt wird. Bei jedem Entwurfsschritt wird die Erfüllbarkeit von Anforderungen überprüft und somit werden einzelne Entwurfsschritte frühzeitig bewertet. Eine wichtige Anforderung an Echtzeitsysteme ist ein vorhersagbares Ausführungsverhalten durch die Einhaltung von vorgegebenen Zeitschranken bei der Ausführung von Tasks auf einem Zielprozessor. Abstrakte RTOS-Modelle erlauben hier eine lauffzeiteffiziente Simulation des funktionalen Verhaltens und der Ausführungszeiten. Eine Herausforderung ist es dabei, trotz der Abstraktion, eine zeitlich genaue Simulation von Ausführungszeiten zu erreichen.

In dieser Arbeit werden die folgenden Verbesserungen für abstrakte RTOS-Modelle vorgestellt: Neben dem Scheduling für die Echtzeittasks wird zusätzlich das Scheduling für Interrupt-Service-Routinen (ISRn) modelliert und simuliert. Die Scheduling-Strategien lassen sich dabei unabhängig voneinander festlegen, um die hardwareabhängige Ausführungsreihenfolge von ISRn unabhängig von dem RTOS-Scheduling zu evaluieren. Die abstrakte RTOS-Simulation erfordert die Abstraktion der Ausführungszeiten von Tasks auf einem Zielprozessor. In dieser Arbeit wird eine effiziente Methode vorgestellt, die eine Abstraktion von optimiertem Assemblercode erlaubt und die Simulation von Task-Migrationen zwischen verschiedenen Zielprozessoren unterstützt. Die Effizienz dieser Methode resultiert aus ihrem geringen Speicherplatzbedarf und den kurzen Zugriffszeiten während der Simulation. Ein Ergebnis der Simulation mit abstrakten RTOS-Modellen sind die Antwortzeiten von Tasks. Als Maß für die Qualität der Simulation werden diese Werte mit Referenzwerten, z.B. aus Messungen, verglichen. Zusätzlich ist eine hohe Übereinstimmung der Kommunikationszeitpunkte und deren identische Reihenfolge erforderlich. Die Abstraktion der Tasks kann sich negativ auf diese Übereinstimmung auswirken, was allein durch den obigen Vergleich von Antwortzeiten nicht feststellbar ist. In dieser Arbeit wird eine Methode vorgestellt, die die Übereinstimmung auch bei starken Abstraktionen erhöht.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Ziele der Arbeit	3
1.3	Struktur der Arbeit	5
2	Grundlagen	7
2.1	Entwurf elektronischer Systeme	7
2.1.1	Transaction Level Modeling	8
2.1.2	TLM-Ebenen der OSCI	9
2.1.3	TLM-Ebenen nach Cai/Gajski	10
2.1.4	Software und TLM	12
2.2	Laufzeitanalyse	13
2.2.1	Ausführungszeiten von Tasks	14
2.2.2	Worst-Case-Execution-Time-Analyse	16
2.2.3	Instruction-Set-Simulation	17
2.2.4	Analytische Verfahren	18
2.3	Fazit	19
3	Abstrakte RTOS Simulation	21
3.1	Motivation	21
3.2	Ziele der Arbeit	24
3.3	Simulation auf Task-Ebene	25
3.3.1	Basisblöcke	25
3.3.2	Segmente	26
3.3.3	Instrumentierung von Quelltext	28
3.3.4	Probleme der Abstraktion	29
3.3.5	Verwandte Arbeiten	31
3.4	Abstraktion des RTOS	32
3.4.1	Zustandsmodell und Scheduler	34
3.4.2	Prinzipien der Simulation	35
3.4.3	Berücksichtigung von Interrupts	36
3.4.4	Verwandte Arbeiten	36
3.5	Anwendungen	38
3.5.1	Entwurfsraumexploration	39
3.5.2	Software Synthese und Refinement	39

3.6	Fazit	40
4	Getrennte Modellierung von Schedulern	43
4.1	Motivation	43
4.2	Behandlung der Zeit	46
4.3	Scheduling von ISRn	48
4.4	Integration in die Simulation	51
4.5	Fazit	52
5	Annotation von Ausführungszeiten	55
5.1	Motivation	55
5.2	Annotation von Laufzeiten	58
5.3	Speicherung von Laufzeiten	60
5.4	Effiziente Annotation	64
5.5	Definition von Lösungsgraphen	66
5.6	Annotation und Lösungsgraphen	68
5.7	Kombination von Graphen	71
5.8	Grammatik zur Markierung und Annotation	71
5.9	Aufbau der Lösungsgraphen	74
5.10	Kreisfreiheit der Lösungsgraphen	78
5.11	Kreisfreiheit von Sequenzen	82
5.12	Kreisfreiheit von bedingten Verzweigungen	83
5.13	Kreisfreiheit von Schleifen	86
5.14	Kreisfreiheit von Funktionen	91
5.15	Fazit	95
6	Kompensation der Überabschätzung	97
6.1	Motivation	97
6.2	Vergleich von Ablaufreihenfolgen	101
6.3	Vergleich Antwortzeiten	106
6.4	Anpassung der Periode	109
6.5	Annotation der Fehler	113
6.6	Fazit	115
7	Evaluierung	119
7.1	Laufzeiten der Zeitannotation	119
7.2	Evaluierung des Rauschens	124
7.2.1	Motivation	124
7.2.2	Abstraktion der Hardware	125
7.2.3	Abstraktion der Software	126
7.2.4	Vergleich von Software und Hardware	128
7.2.5	Beschreibung der Beispiele	129
7.2.6	Darstellung der Simulationsergebnisse	134
7.2.7	Ergebnisse: Verteilte Lichtsteuerung	137
7.2.8	Ergebnisse: RTOS-Benchmark	142

7.2.9 Vergleich mit anderen Arbeiten	155
7.2.10 Fazit	158
8 Zusammenfassung	161
A Lösungsgraphen	165
B Messdaten der Evaluierung	169
B.1 Überdeckungskurven der Konfigurationen A-D	169
B.2 Messkurven zum Simulationsaufwand	178
B.3 Modifizierte Markierung der u16mod Funktion	180
B.4 RTOS-Benchmark exakt	181
Abbildungsverzeichnis	187
Publikationen	189
Literaturverzeichnis	189