

Inhaltsverzeichnis

| | |
|--|-----------|
| Abkürzungsverzeichnis | ix |
| Formelzeichen | xi |
| 1 Einleitung und Motivation | 1 |
| 1.1 Entwicklung von Steuergeräten im Automobil | 1 |
| 1.2 Ziele der Arbeit | 5 |
| 1.3 Gliederung | 6 |
| 2 Grundlagen | 9 |
| 2.1 Algebro-Differentialgleichungssysteme | 9 |
| 2.2 Numerische Integration | 12 |
| 2.2.1 Das Euler-Cauchy-Verfahren | 12 |
| 2.2.2 Runge-Kutta-Verfahren | 13 |
| 2.2.3 Weitere Ansätze | 14 |
| 2.3 Rechnergestützte Simulation physikalischer Systeme | 14 |
| 2.3.1 Diskrete Ereignissysteme | 17 |
| 2.3.2 Zeitgetriebene Systeme | 19 |
| 2.3.3 Zeitkontinuierliche Systeme | 20 |
| 2.3.4 Hybride Systeme | 22 |
| 2.4 Modelica | 23 |
| 2.4.1 Merkmale | 23 |
| 2.4.2 Übersetzung und Simulation | 28 |
| 2.5 Elektrische Motoren | 30 |
| 2.5.1 Kommutatormaschinen | 30 |
| 2.5.2 Asynchronmaschinen | 33 |
| 2.5.3 Synchronmaschinen | 38 |
| 2.5.4 Reluktanzmaschinen | 41 |
| 2.6 Field-Programmable Gate Arrays | 41 |
| 2.6.1 Übersicht | 41 |
| 2.6.2 Programmierung | 45 |
| 2.6.3 High-Level Synthese | 49 |
| 2.7 Hardware-in-the-Loop Testmethodik | 52 |
| 2.8 Grundbegriffe des Übersetzerbaus | 54 |
| 3 Stand der Technik | 59 |
| 3.1 HiL-Test elektrifizierter Antriebsstränge | 59 |
| 3.1.1 Testmethoden | 59 |
| 3.1.2 FPGA-basierte Simulation elektrischer Maschinen | 60 |

| | | |
|-------|--|-----|
| 3.2 | Sprachen und Werkzeuge für ODE- und DAE-Simulationen | 65 |
| 3.2.1 | MATLAB/Simulink und Stateflow | 65 |
| 3.2.2 | SimScape | 66 |
| 3.2.3 | Modelica-fähige Werkzeuge | 66 |
| 3.2.4 | AMS-Erweiterungen | 67 |
| 3.2.5 | SPICE | 68 |
| 3.2.6 | Saber/MAST und SystemVision | 68 |
| 3.3 | Beschleunigung physikalischer Simulationen | 69 |
| 3.3.1 | Parallelisierung | 69 |
| 3.3.2 | GPGPU | 71 |
| 3.3.3 | FPGA | 72 |
| 3.4 | High-Level Synthese | 73 |
| 3.4.1 | Ansätze | 73 |
| 3.4.2 | Werkzeuge | 85 |
| 4 | Analyse | 87 |
| 4.1 | Anforderungen | 87 |
| 4.1.1 | Modellierungsanforderungen | 87 |
| 4.1.2 | Zeitliche Anforderungen | 93 |
| 4.1.3 | Fazit | 94 |
| 4.2 | Entwurfsunterstützung durch existierende Werkzeuge | 95 |
| 4.2.1 | Modelica und Werkzeuge | 95 |
| 4.2.2 | SimScape | 96 |
| 4.2.3 | Simulink und HDL Coder | 96 |
| 4.2.4 | C to Gates? | 97 |
| 5 | Konzeption | 101 |
| 5.1 | Entwurfsfluss | 101 |
| 5.1.1 | Gesamtkonzept | 101 |
| 5.1.2 | Übergabeformat: Anforderungen | 102 |
| 5.1.3 | Functional Mockup Interface | 103 |
| 5.2 | Extensible Intermediate Language | 104 |
| 5.2.1 | Typsystem | 104 |
| 5.2.2 | Domänenspezifische Operatoren | 105 |
| 5.2.3 | Scheduling-Restriktionen | 106 |
| 5.2.4 | Serialisierung | 106 |
| 5.3 | Kopplung mit FPGA Werkzeugkette | 107 |
| 5.3.1 | Zieltechnologie | 107 |
| 5.3.2 | Generate | 107 |
| 5.4 | Methodik | 108 |
| 5.4.1 | FPGA-freundliches Modellieren | 108 |
| 5.4.2 | Synthesekonzept | 109 |

| | | |
|----------|--|------------|
| 6 | Umsetzung | 111 |
| 6.1 | Das System#-Framework | 111 |
| 6.1.1 | Übersicht | 111 |
| 6.1.2 | SysDOM | 112 |
| 6.1.3 | Die generative Modellierungsebene | 116 |
| 6.1.4 | Die deskriptive Modellierungsebene | 116 |
| 6.1.5 | Elaboration | 122 |
| 6.1.6 | Typsystem | 122 |
| 6.1.7 | Simulationskernel | 124 |
| 6.1.8 | Code-Generierung | 126 |
| 6.2 | Dekompilierung von deskriptivem Code | 127 |
| 6.2.1 | Strukturelle Analyse | 127 |
| 6.2.2 | Isolation des Laufzeitverhaltens | 128 |
| 6.2.3 | Methoden ohne „async“ Modifier | 129 |
| 6.2.4 | Methoden mit „async“ Modifier | 137 |
| 6.3 | High-Level Synthese in System# | 139 |
| 6.3.1 | Übersicht | 139 |
| 6.3.2 | Programmtransformationen | 140 |
| 6.3.3 | Komponentenbibliothek | 143 |
| 6.3.4 | Scheduling | 144 |
| 6.3.5 | Ressourcen-Allokation und Binding | 145 |
| 6.3.6 | Interconnect-Allokation | 146 |
| 6.3.7 | Kontrollpfad-Synthese | 147 |
| 6.3.8 | Syntheseplan | 152 |
| 6.4 | Domänen-spezifische Bausteine | 154 |
| 6.4.1 | Behandlung linearer Systeme | 154 |
| 6.4.2 | Behandlung nichtlinearer Systeme | 157 |
| 6.4.3 | Xilinx IP Cores | 157 |
| 6.5 | Automatisierte Auslegung von Festkommatentypen | 158 |
| 6.5.1 | Einführung | 158 |
| 6.5.2 | Verwandte Arbeiten | 160 |
| 6.5.3 | Bestimmung der Wertebereiche | 161 |
| 6.5.4 | Sensitivitätsanalyse | 162 |
| 6.5.5 | Wortbreitenbestimmung | 166 |
| 6.6 | Synthesewerkzeug | 169 |
| 6.6.1 | Übersicht | 169 |
| 6.6.2 | Mensch-Maschine-Schnittstelle | 170 |
| 6.6.3 | Verifikation und Logiksynthese | 173 |
| 7 | Praxis-Erprobung: Elektromotoren | 175 |
| 7.1 | Ausgewählte Beispiele | 175 |
| 7.2 | Inbetriebnahmen bei Projektpartnern | 178 |
| 7.3 | Parametrierung von Festkommaarithmetik | 180 |
| 7.4 | Synthese-Ergebnisse | 186 |
| 7.4.1 | Latenzkonfiguration arithmetischer Operatoren | 187 |

| | | |
|-------|---|-----|
| 7.4.2 | FSM vs. HMA | 192 |
| 7.4.3 | Performanz-steigernde Maßnahmen | 195 |
| 7.5 | Vergleich mit existierenden Lösungen | 197 |
| 7.5.1 | E/A-Schnittstelle zur Signalkonditionierung | 198 |
| 7.5.2 | E/A-Schnittstelle zur Teststeuerung | 198 |
| 7.5.3 | Sicherheitsebene | 199 |
| 7.5.4 | Motorsimulation | 199 |
| 7.5.5 | Inverter | 200 |
| 7.5.6 | Fazit | 200 |
| 8 | Zusammenfassung | 203 |
| 9 | Ausblick | 207 |
| A | Verzeichnisse | 211 |
| A.1 | Eigene Veröffentlichungen (E) | 211 |
| A.2 | Betreute studentische Arbeiten (B) | 213 |
| A.3 | Fremdliteratur | 215 |
| A.4 | Standards und Normen (Std) | 238 |
| A.5 | Webquellen (WWW) | 239 |
| A.6 | Abbildungsverzeichnis | 242 |
| A.7 | Tabellenverzeichnis | 243 |