

Inhaltsverzeichnis

Abkürzungsverzeichnis	10
Vorwort	15
1 Einleitung	17
1.1 Motivation	17
1.2 Ziele der Arbeit	21
1.3 Aufbau der Arbeit	23
2 Grundlagen	25
2.1 Parallele Rechentechnik	25
2.1.1 Domänen der Berechnung	26
2.1.2 Parallelität der Software-Domäne	28
2.1.3 Speicherorganisation in parallelen Rechnersystemen	32
2.1.4 Rechentechnik mit rekonfigurierbaren Hardwarestrukturen	33
2.1.5 Ableitung und Modellierung heterogener adaptiver Systemarchitekturen	38
2.1.6 Modell eines heterogenen adaptiven Computersystems	41
2.2 Entwurf von Hardware- und Software-Systemen	42
2.2.1 Entwurfssichten, Abstraktionsebenen und Methodik des Systementwurfs	43
2.2.2 Entwurfsprozess anwendungsspezifischer Hardware	46
2.2.3 Dynamisch partiell rekonfigurierbare Schaltungen	50
2.2.4 High-Level-Synthese - Synthese algorithmischer Verhaltensbeschreibungen	51
2.3 Parallele Anwendungskonzepte und Programmiermodelle	64
2.4 Scheduling-Strategien	68
2.4.1 Shortest Job First (SJF)	69
2.4.2 Round-Robin	70
2.4.3 Prioritätsbasiertes Scheduling	70
2.4.4 Multilevel-Feedback Scheduling (MLFS)	71

3	Stand der Technik	73
3.1	Anwendungsentwurf für heterogene Computersysteme	73
3.2	Laufzeitumgebungen und Architekturen	77
3.3	Unterbrechung und Migration	79
4	Entwurf dynamisch verteilter Anwendungen für heterogene Computer	83
4.1	Spezifikation	85
4.2	Codeumsetzung und Partitionierung in eine parallelisierte Anwendung	85
4.3	Architektur-Auswahl	86
4.4	Management-Code-Injektion	87
4.5	Kompilierung und Synthese	89
4.6	HW/SW-Co-Simulation	90
4.7	Eigenschafts- und Komplexitätsanalyse	92
5	Unterbrechung und Migration	95
5.1	Modellbildung und Formalisierung	97
5.2	Kosten für Unterbrechung und Migration	101
5.2.1	Kostenbetrachtungen zusätzlicher Hardwarestrukturen	101
5.2.2	Zeitliche Aufwände einer Unterbrechung	103
5.3	Verteilung von Unterbrechungspunkten	105
5.3.1	Unterbrechungspunkte auf Hochsprachenebene	107
5.3.2	Unterbrechungspunkte auf RT-Ebene	111
6	Laufzeitumgebung und Systemverwaltung	127
6.1	Ausführungsmodell	127
6.2	Anforderungen	129
6.3	Konzeption und Aufbau	130
6.3.1	Programm-Laderoutine, Anwendungsmanager und Speichermanagement	130
6.3.2	Kommunikationsschnittstelle	131
6.3.3	Ressourcenverwaltung und temporale Ablaufplanung der Tasks	133
6.3.4	Übersicht des Ausführungsablaufs einer Anwendung .	140
7	Implementierung der Entwurfsmethodik	143
7.1	Zielformat, Betriebssystem und Laufzeitumgebung	143
7.2	Automatisierung des Entwurfsablaufes	145
7.2.1	Synthesewerkzeuge	145
7.2.2	Anwendungsanalyse und Codeinjektion	146
7.3	Hardware-Erweiterungen	148

7.3.1	Unterbrechungslogik und Kontextspeicher	148
7.3.2	Strombasierte Hardware-Software Schnittstelle	151
7.3.3	Basisstruktur eines Ausführungskontainers in Hardware	152
8	Ergebnisse	155
8.1	Anwendungsbeispiele	155
8.1.1	Auf-/Abwärtszähler	155
8.1.2	MJPEG Kodierung	156
8.1.3	Partikel-Filter	157
8.2	Funktionsnachweis und Schedulinganalyse	160
8.3	Analyse der Unterbrechungspunktverteilung	162
8.4	Gewichtsfaktoren zur Unterbrechungspunktverteilung	166
8.5	Fazit	168
9	Schluss	169
9.1	Zusammenfassung	169
9.2	Ausblick	170
A	Werkzeuge zur High-Level-Synthese	173
A.1	Kommerzielle Werkzeuge	173
A.2	Akademische Ansätze und Werkzeuge	174
	Literatur	176
	Abbildungsverzeichnis	201
	Tabellenverzeichnis	203
	Algorithmensverzeichnis	205
	Thesen	207