

## Inhaltsverzeichnis

1	Einleitung . . . . .	1
1.1	Ansätze aus der Literatur . . . . .	2
1.2	Zielsetzung . . . . .	3
2	Beschreibung des Gesamtsystems . . . . .	4
3	Quantitative Analyse . . . . .	6
3.1	Methodik der quantitativen Analyse . . . . .	7
3.1.1	Architekturelevante Fragestellungen . . . . .	7
3.1.2	Statische Analyse . . . . .	8
3.1.3	Dynamische Analyse . . . . .	10
3.2	Ergebnisse aus der Literatur . . . . .	10
3.2.1	Instruktionssatzanalyse . . . . .	11
3.2.2	Hochsprachenprogramm-Analyse . . . . .	12
3.3	Diskussion gängiger Controllerarchitekturen . . . . .	14
3.3.1	Motorola 68HC05 . . . . .	14
3.3.2	Intel 8051 . . . . .	17
3.3.3	Weitere Architekturen . . . . .	20
3.4	Zusammenfassende Bewertung . . . . .	21
4	Design eines 68HC05-Klons . . . . .	23
4.1	Funktionale Modellierung . . . . .	23
4.1.1	Die Architektur-Ebene . . . . .	24
4.1.2	Die Mikroprogramm-Ebene . . . . .	25
4.1.3	Die Register-Transfer-Ebene . . . . .	26
4.2	Verifikation der Modellhierarchien . . . . .	26
4.3	Grundlegende Schaltungstechniken . . . . .	29
4.4	Das Steuerwerk . . . . .	30
4.5	Der Datenpfad . . . . .	33
4.6	Der Interrupt-Controller . . . . .	35
4.7	Der Maskengenerator . . . . .	36
4.8	Diskussion der Erfahrungen . . . . .	36
5	Implikationen der Compiler-Technologie . . . . .	40
5.1	Aufbau des retargierbaren Compilers . . . . .	40
5.2	Ansätze aus der Literatur . . . . .	42

5.2.1	Der RISC-Ansatz	43
5.2.2	Hochsprachen-Maschinen	44
5.3	Die Adressierungsarten	45
5.3.1	Das Adressierungsmodell	46
5.3.2	Virtuelle Register	47
5.3.3	Die Adressierungsräume	49
5.4	Der Kontrollfluß	49
5.5	Der Instruktionsmix	50
5.6	Die implementierte Laufzeitumgebung	51
5.7	Ableitung von Grundsätzen	55
6	Echtzeit-Anwendungen	56
6.1	Reaktion auf externe Ereignisse	56
6.1.1	Ein generischer Interrupt-Controller	60
6.2	Multitasking	61
6.2.1	Scheduling-Strategien	61
6.2.2	Multitasking auf dem 68HC05	63
6.2.3	Ein Implementierungsbeispiel	64
6.2.4	Speicherplatzverwaltung	66
6.3	Prozeßsynchronisation und -kommunikation	67
6.4	Ableitung von Grundsätzen	69
7	Entwurfs- und Verifikationswerkzeuge	70
7.1	Der Architektursimulator	70
7.1.1	Die Modellsynthese	71
7.1.2	Der MicroDebugger	72
7.2	Der ScanDebugger	73
7.2.1	Taktsynchronisation	75
7.2.2	Kosten und Perspektiven	76
7.3	Hardware-Erweiterungen	76
7.4	Schlußfolgerungen	77
8	Vorschlag einer Architektur	78
8.1	Die Speicherorganisation	78
8.1.1	Ein- und Ausgabekanäle	81
8.2	Das Registermodell	82
8.3	Der Instruktionssatz	82
8.3.1	Die Codierung der Instruktionen	87

8.4	Der Entwurf des Datenpfades . . . . .	90
8.4.1	Verkettung von Ressourcen . . . . .	92
8.4.2	Scheduling des Instruktionssatzes . . . . .	92
8.4.3	Struktur des Datenpfades . . . . .	94
8.5	Erweiterungen des Instruktionssatzes . . . . .	96
9	Bewertung der Architektur . . . . .	98
9.1	Der Implementierungsaufwand . . . . .	98
9.2	Assembler-Benchmark . . . . .	100
9.3	Compiler-Unterstützung . . . . .	102
9.4	Echtzeit-Unterstützung . . . . .	103
9.5	Abschließende Bemerkungen . . . . .	104
10	Zusammenfassung . . . . .	105
	Anhang A: Instruktionssätze . . . . .	107
	Anhang B: Statische Instruktionssatznutzung . . . . .	108
	Anhang C: Quellen und Senken im IMS2205 . . . . .	112
	Anhang D: Instruktionssatz der t1-Architektur . . . . .	113
	Bibliographie . . . . .	117