

Ralf Gessler  
Thomas Mahr

# **Hardware- Software- Codesign**

**Entwicklung flexibler  
Mikroprozessor-FPGA-  
Hochleistungssysteme**

Mit 134 Abbildungen



# Inhaltsverzeichnis

## Vorwort

## Abkürzungsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Eingebettete Systeme</b>	<b>5</b>
2.1	Definition	5
2.2	Entwicklung	6
2.2.1	Software-Entwicklung	10
2.2.2	Rechnerarchitekturen	10
2.2.3	Rechenbaustein und Hardware-Technologie	11
2.2.4	Rechenmaschine	12
<b>3</b>	<b>Rechenmaschinen</b>	<b>15</b>
3.1	Turing-Maschine	15
3.2	IC-Technologie	20
3.2.1	Schaltungsebene	20
3.2.2	Layoutebene	31
3.3	Rechenbausteine	31
3.4	Übersicht Rechenmaschinen	33
3.5	Mikroprozessoren	36
3.5.1	Der erste Mikroprozessor	36
3.5.2	Grundlegende Funktionsweise der Mikroprozessoren	38
3.5.3	Mikroprozessor-Architekturen	45
3.5.4	Mikroprozessor-Arten	61
3.5.5	Digitale Signalprozessoren	64
3.5.6	Ausgewählte Prozessoren	73
3.6	Programmierbare Logikschaltkreise	75
3.6.1	Einordnung Schaltkreise	75
3.6.2	Kenngroßen	76
3.6.3	PLD-Architekturen	77
3.6.4	Konfiguration	82
3.6.5	FPGA-Architekturen	86
3.6.6	Diskussion	100

<b>4</b>	<b>Software-Entwicklung (Mahr)</b>	<b>103</b>
4.1	Was ist Software-Entwicklung?	104
4.1.1	Begriffsdefinitionen	104
4.1.2	Software-Entwicklung und Bergwanderung	111
4.2	Die Entwicklung von Software	114
4.2.1	Das Problem verstehen lernen	115
4.2.2	Eine Lösung finden – Entwurf der Software	123
4.2.3	Die Lösung umsetzen – Implementierung der Software	134
4.2.4	Die Lösung testen	156
4.3	Vorgehensmodelle	159
4.3.1	Das Wasserfall-Modell – ein historisches Missverständnis	160
4.3.2	Iterativ-inkrementelle Entwicklung	162
<b>5</b>	<b>Digitale Schaltungstechnik (Gessler)</b>	<b>167</b>
5.1	Begriffsbestimmung	167
5.2	Entwurf digitaler Schaltungen	170
5.2.1	Ebenen	170
5.2.2	System-, Algorithmen- und Register-Transfer-Ebene	171
5.2.3	Logikebene	172
5.2.4	Generelles	179
5.3	Rechnergestützter Schaltungsentwurf und -implementierung	184
5.4	Beschreibungssprachen	189
5.4.1	VHDL	190
5.4.2	Verilog	198
<b>6</b>	<b>Hardware-Software-Codesign (Mahr)</b>	<b>203</b>
6.1	Vergleich Software-Entwicklung und Schaltungsentwurf	204
6.2	Synergie: Software-Methodik und Schaltungsentwurf	209
<b>7</b>	<b>Hybride Architekturen (Gessler)</b>	<b>213</b>
7.1	System On Chip	214
7.1.1	FPSLIC	215
7.1.2	FPGA	215
7.2	cASIP	219
7.3	FPFA	221
7.4	Rekonfigurierbare Architekturen	225
<b>8</b>	<b>Werkzeuge zum Entwurf auf Systemebene (Gessler)</b>	<b>229</b>
8.1	Matlab/Simulink	229
8.1.1	Matlab/Simulink und System-Generator	230
8.1.2	Matlab/Simulink und Embedded Target	233
8.2	SystemC	234
8.2.1	Motivation	235
8.2.2	Entwurf	235

<b>9 UML-basierte Entwicklung für hybride Systeme (Mahr)</b>	<b>239</b>
9.1 Hybride Systeme	239
9.2 Software-Entwicklung für hybride Systeme	241
9.3 Der ideale Codesigner	246
9.3.1 Warum modellieren und abstrahieren?	248
9.3.2 Wie modellieren? grafisch oder schriftlich?	252
9.3.3 Was grafisch modellieren?	254
9.3.4 Transformation	255
9.4 UML-basierter Codegenerator	256
9.4.1 Warum Modellierung in UML?	256
9.4.2 Codegeneratoren	258
9.4.3 Ein einfacher UML-zu-Code-Transformator	258
9.4.4 Beispiel: Blinklicht	261
<b>A Anhang (Gessler)</b>	<b>267</b>
A.1 Zahlensysteme und Arithmetik	267
A.1.1 Darstellung ganzer Zahlen	267
A.1.2 Darstellung reeller Zahlen	268
A.1.3 Arithmetik	270
A.1.4 Emulation von Gleitkommazahlen	272
A.2 Auswahlhilfen	272
A.2.1 Maßzahlen	272
A.2.2 Benchmarks	276
<b>Literaturverzeichnis</b>	<b>277</b>
<b>Stichwortverzeichnis</b>	<b>289</b>