

Inhaltsverzeichnis

1	Einleitung	1
1.1	Begriffe und Abgrenzung	1
1.2	Anwendungsgebiete	5
1.3	Wachsende Relevanz von eingebetteten Systemen	8
1.4	Struktur dieses Buches	10
2	Spezifikationssprachen	13
2.1	Anforderungen	13
2.2	Berechnungsmodelle	17
2.3	StateCharts	18
2.3.1	Modellierung von Hierarchie	19
2.3.2	Zeitbedingungen	24
2.3.3	Kantenbeschriftungen und StateMate-Semantik	25
2.3.4	Bewertung und Erweiterungen	28
2.4	Allgemeine Spracheigenschaften	29
2.4.1	Synchrone und asynchrone Sprachen	29
2.4.2	Prozess-Konzepte	30
2.4.3	Synchronisation und Kommunikation	30
2.4.4	Spezifikation von Zeitbedingungen	31
2.4.5	Nicht-Standard-Ein-/Ausgabegeräte	32
2.5	SDL	32
2.6	Petrinetze	39
2.6.1	Einführung	39
2.6.2	Bedingungs-/Ereignisnetze	42
2.6.3	Stellen-/Transitionen-Netze	43

2.6.4	Prädikat-/Ereignis-Netze	47	3.4.1	Überblick	108
2.6.5	Bewertung	49	3.4.2	Anwendungsspezifische integrierte Schaltkreise (ASICs)	109
2.7	<i>Message Sequence Charts</i>	49	3.4.3	Prozessoren	110
2.8	UML	51	3.4.4	Rekonfigurierbare Logik	126
2.9	Prozessnetze	55	3.5	Speicher	129
2.9.1	Taskgraphen	55	3.6	Ausgabe	131
2.9.2	Asynchroner Nachrichtenaustausch	58	3.6.1	D/A-Wandler	132
2.9.3	Synchroner Nachrichtenaustausch	60	3.6.2	Aktuatoren	133
2.10	Java	63	4	Eingebettete Betriebssysteme, <i>Middleware</i> und Scheduling	135
2.11	VHDL	64	4.1	Schranken von Ausführungszeiten	136
2.11.1	Einführung	64	4.2	Scheduling in Echtzeitsystemen	137
2.11.2	<i>Entities</i> und <i>Architectures</i>	66	4.2.1	Klassifikation von Scheduling-Algorithmen	138
2.11.3	Mehrwertige Logik und IEEE 1164	68	4.2.2	Aperiodisches Scheduling	141
2.11.4	VHDL-Prozesse und Simulations-Semantik	75	4.2.3	Periodisches Scheduling	149
2.12	SystemC	80	4.2.4	Ressourcen-Zugriffs-Protokolle	155
2.13	Verilog und SystemVerilog	81	4.3	Eingebettete Betriebssysteme	158
2.14	SpecC	83	4.3.1	Allgemeine Anforderungen	158
2.15	Weitere Sprachen	84	4.3.2	Echtzeitbetriebssysteme	160
2.16	Ebenen der Hardware-Modellierung	87	4.4	<i>Middleware</i>	164
2.17	Vergleich der Sprachen	90	4.4.1	Echtzeit-Datenbanken	164
2.18	Verlässlichkeitsanforderungen	92	4.4.2	Zugriff auf entfernte Objekte	165
3	Hardware eingebetteter Systeme	95	5	Implementierung eingebetteter Systeme: Hardware-/Software-Codesign	169
3.1	Einführung	95	5.1	Organisation der Nebenläufigkeit auf Task-Ebene	172
3.2	Eingabe	96	5.2	<i>High-Level</i> -Optimierungen	176
3.2.1	Sensoren	96	5.2.1	Umwandlung von Gleitkomma- in Festkomma-Darstellung	176
3.2.2	Sample-and-Hold-Schaltungen	99	5.2.2	Einfache Schleifentransformationen	177
3.2.3	A/D-Wandler	100	5.2.3	Kachel-/Blockweise Verarbeitung von Schleifen	179
3.3	Kommunikation	102	5.2.4	Aufteilen von Schleifen	182
3.3.1	Anforderungen	103	5.2.5	Falten von Feldern	184
3.3.2	Elektrische Robustheit	104	5.3	Hardware-/Software-Partitionierung	186
3.3.3	Garantieren von Echtzeitverhalten	105	5.3.1	Einleitung	186
3.3.4	Beispiele	106			
3.4	Verarbeitungseinheiten	108			

5.3.2	COOL	187
5.4	Compiler für eingebettete Systeme	197
5.4.1	Einführung	197
5.4.2	Energieoptimierende Compiler	198
5.4.3	Compiler für digitale Signalverarbeitung	202
5.4.4	Compiler für Multimedia-Prozessoren	205
5.4.5	Compiler für VLIW-Prozessoren	205
5.4.6	Compiler für Netzwerkprozessoren	206
5.4.7	Compiler-Erzeugung, retargierbare Compiler und Untersuchung des Entwurfsraums	207
5.5	Versorgungsspannungs-Anpassung und Energie-Management ..	207
5.5.1	Dynamische Anpassung der Versorgungsspannung	207
5.5.2	Dynamisches Power-Management	211
6	Evaluierung und Validierung	213
6.1	Einleitung	213
6.2	Pareto-Optimalität	215
6.3	Simulation	217
6.4	Rapid Prototyping und Emulation	218
6.5	Leistungsbewertung	219
6.6	Bewertung des Energieverbrauchs	226
6.7	Risiko- und Verlässlichkeits-Analyse	227
6.8	Formale Verifikation	233
6.9	Testen	236
6.9.1	Betrachteter Bereich	236
6.9.2	Testfreundlicher Entwurf	237
6.9.3	Selbsttestprogramme	240
6.10	Fehlersimulation	241
6.11	Fehlerinjektion	242
	Literaturverzeichnis	245
	Sachverzeichnis	259