

Inhaltsverzeichnis

1	Einleitung	1
1.1	Was ist Rechnerarchitektur?	1
1.2	Zum Stand der Technik der Hardwarekomponenten	3
1.2.1	Schaltkreistechnik	3
1.2.2	Schaltkreise	5
1.2.3	Speicherbausteine	7
1.2.4	Prozessoren	9
1.3	Motivation für innovative Rechnerarchitekturen	13
1.4	Das Schichtenmodell eines Rechnersystems	13
1.5	Die Konstituenten einer Rechnerarchitektur	21
1.5.1	Definitionen	21
1.5.2	Abstrakte Datentypen	22
1.6	Taxonomie von Rechnerarchitekturen	25
1.6.1	Allgemeine Bemerkungen	25
1.6.2	Operationsprinzipien	27
1.6.3	Strukturen von Parallelrechner-Architekturen	28
	Literatur zu Kapitel 1	31
2	Sequentielle Rechner	33
2.1	Die Struktur der von-Neumann-Maschine	33
2.2	Das Operationsprinzip der von-Neumann-Maschine	35
2.3	Berechnung und Ablaufkontrolle in der von-Neumann-Maschine	38
2.4	Nicht-von-Neumann-Architekturen	44
2.5	Programmstrukturen sequentieller Rechner	45
2.6	Konkurrente Prozesse	51
2.6.1	Schwergewichtige und leichtgewichtige Prozesse, Kontrollfäden	51
2.6.2	Das Petri-Netz-Modell	53
2.7	Grundlagen der Speicherorganisation im von-Neumann-Rechner	55
2.7.1	Der Working Set eines von-Neumann-Programms	55
2.7.2	Schutzmechanismen	58
	Literatur zu Kapitel 2	61
3	Grundzüge der Prozessor-Architekturen	63
3.1	Die minimale von-Neumann-Maschine	63
3.2	Mehrregister-Maschinen	65
3.2.1	Beispiel einer einfachen Mehrregister-Maschine (MC68020)	65
3.2.2	Adressierungsarten	67

3.2.3	Mehrfach-Registersätze	70
3.3	Mikroprogrammierte CISC-Prozessoren	71
3.3.1	Vorbemerkung	71
3.3.2	Mikroprogrammierung	73
3.3.3	Vertikale Verlagerung	78
3.4	Anwendung des Pipeline-Prinzips im Prozessor	80
3.4.1	Die Befehls-Pipeline	80
3.4.2	Pipelining von arithmetischen Operationen	83
3.5	RISC-Prozessoren	86
3.5.1	Das Ziel der RISC-Bewegung	86
3.5.2	Pipeline-Skalarprozessoren und Vektormaschinen	88
3.5.3	Beispiel: SPARC-Prozessor	89
3.6	Bus-Systeme	93
3.6.1	Bus-Arbitrierung	93
3.6.2	Standardbusse	96
	Literatur zu Kapitel 3	97
4	Speicherorganisation und Speicherverwaltung	99
4.1	Speicherhierarchie	99
4.2	Seitenadressierung	100
4.3	Cache-Speicher	102
4.3.1	Vorteile des Cache-Speichers	102
4.3.2	Transparenz der Adressierung: der vollasoziative Cache	104
4.3.3	Der <i>Direct Mapping</i> -Cache	107
4.3.4	Der mengenasoziative Cache	107
4.3.5	Bildung von <i>Working Sets</i>	109
4.3.6	Cache-Kohärenz	111
4.3.7	"Logischer" oder "physikalischer" Cache?	116
4.4	Hauptspeicher	118
4.4.1	Der wortbreite Speicher	118
4.4.2	Speicherverschränkung	120
4.5	Speichersegmentierung	123
4.6	Virtueller Speicher	127
4.7	Einfacher Objektschutz	131
4.7.1	Systemaufsichts-Modus und Benutzer-Modus	131
4.7.2	Speicherverwaltung	132
4.8	Absoluter Objektschutz: Capability-Adressierung	133
4.8.1	Capability-Adressierung und Speichersegmentierung	134
4.8.2	Referenz über Capability-Register	135
4.8.3	Einführung eindeutiger Objekt-Identifikatoren	137
4.8.4	Der Prozessor iAPX432	138
	Literatur zu Kapitel 4	141
5	Konzepte der Parallelarbeit	143
5.1	Parallelität und Datenabhängigkeit	143
5.2	Die Ebenen der Parallelarbeit	147
5.3	Die Anweisungsebene	151

5.3.1	Übergang zur Maschinenbefehlsebene, allgemeine Optimierungen	151
5.3.2	Software-Pipelining	155
5.4	Erkennen von Parallelität auf der Anweisungsebene	157
5.4.1	Datenabhängigkeitsanalyse in Basisblöcken	157
5.4.2	Datenabhängigkeiten über Verzweigungen hinweg	159
5.4.3	Datenabhängigkeiten in Schleifen	161
5.5	Die Prozeßebene	162
5.5.1	Schwergewichtige und leichtgewichtige Prozesse	162
5.5.2	Parallelarbeit auf der Prozeßebene	162
5.6	Nutzung der Anweisungs-Parallelität	163
5.6.1	Übersicht	163
5.6.2	Konstrukte für die parallele Ausführung mehrerer Anweisungen	163
5.6.3	Schleifen-Parallelisierung	165
5.7	Synchronisation von Parallelarbeit	169
5.7.1	FORK-JOIN-Synchronisation von Kontrollfäden	169
5.7.2	Lock-Step-Betrieb und Barrieren-Synchronisation	171
5.7.3	Synchronisation kooperierender Prozesse	172
5.7.4	Synchronisation durch objektorientierte Variablentypen	175
5.8	Datenstrukturen und Parallelarbeit	175
5.8.1	Explizite Datenparallelität und Datenstruktur-Architekturen	175
5.8.2	Wirkungsgrad der Nutzung der Datenparallelität	177
5.8.3	Vergleich zwischen Pipeline und Array von Rechenelementen	181
5.8.4	Datenparallelität auf der Ebene kooperierender Prozesse	183
5.8.5	Eine Sprache zur datenparallelen Programmierung (HPF)	183
	Literatur zu Kapitel 5	184
6	Superskalare Prozessoren und VLWI-Maschinen	187
6.1	Übersicht	187
6.2	Der erste superskalare Prozessor: CDC6600	187
6.3	Moderne superskalare Mikroprozessoren	191
6.3.1	Allgemeines	191
6.3.2	Intel i80860XP	192
6.3.3	Motorola MC88110	194
6.3.4	Digital 21064 (Alpha)	198
6.4	Das Problem des Multiport-Registerfiles	202
6.5	Das Compilerproblem	203
6.6	Die VLWI-Maschine	205
6.6.1	Grundprinzip der VLIW-Maschine	205
6.6.2	Compiler-Optimierungen und Operationsplanung	206
6.6.3	Die Speicherorganisation der VLIW-Maschine	207
6.6.4	Vergleich der VLIW-Maschine mit dem superskalaren Prozessor	207
	Literatur zu Kapitel 6	210
7	SIMD-Architekturen	211
7.1	Vektormaschinen und Anordnungen von Rechenelementen	211
7.2	Das Operationsprinzip der Vektormaschinen	212
7.3	Leistungskenngrößen der Vektormaschine	214

7.4	Speicherverschränkung	216
7.4.1	Konsequente Adressierung	216
7.4.2	Verschoebene Speicherung (<i>skewed storage</i>)	217
7.4.3	Adressierung mit Zufallsadressen	218
7.4.4	Vektorregister	219
7.5	Frühe Vektormaschinen: STAR-100, CYBER 203, CRAY-1	220
7.5.1	STAR-100 und CYBER 203	220
7.5.2	CRAY-1	222
7.6	Moderne Multiprozessor-Vektormaschinen: CRAY-X MP	226
7.6.1	Zentraler Speicher	226
7.6.2	Prozessor	227
7.7	Die Programmierung von Vektormaschinen	229
7.8	Konzepte der ILLIAC IV als Beispiel eines RE-Arrays	231
7.8.1	Grundstruktur	231
7.8.2	Die Organisation der ILLIAC IV	233
7.8.3	Programmierung der ILLIAC IV	235
7.9	Die Connection Machine CM2	238
	Literatur zu Kapitel 7	241
8	Typenkenning, Datenstruktur-Architekturen, Sprach-Architekturen	243
8.1	Architekturen mit Typenkenning	243
8.1.1	Vorteile der Typenkenning	243
8.1.2	Typengesteuerte Befehlsausführung	245
8.1.3	Unterstützung der Mechanismen höherer Programmiersprachen	245
8.1.4	Unterstützung des Betriebssystems	246
8.1.5	Probleme der Typenkenning und Wege zu ihrer Überwindung	248
8.2	Aufbau von Datenstrukturtypen, DRAMA-Prinzip	249
8.2.1	Das DRAMA-Prinzip	249
8.2.2	DRAMA-Prinzip und Typenkenning	250
8.2.3	Beispiel einer DRAMA-Machine	251
8.3	Datenstruktur-Architekturen	253
8.3.1	Allgemeines über Datenstruktur-Architekturen	253
8.3.2	Definitionen	254
8.3.3	Datenstrukturspeicher mit parallelem Zugriff	257
8.3.4	Beispiel einer frühen Datenstruktur-Architektur: STARLET	258
8.3.5	Beispiel einer modernen Datenstruktur-Architektur: DATYPAR	261
8.3.6	Beispiel einer Spezialrechner-Architektur für die Bildanalyse	267
8.4	Sprach-Architekturen	269
8.4.1	Übersicht	269
8.4.2	Vorteile der Array-Sprachen und Datenstruktur-Architekturen	270
8.4.3	Die Lisp-Maschine ULM	272
8.4.4	Der parallele Prolog-Rechner POPE	278
8.5	Assoziative Rechner	283
8.5.1	Assoziativspeicher	283
8.5.2	Assoziativrechner	284
8.5.3	Der assoziative Feldrechner STARAN	285
	Literatur zu Kapitel 8	288

9	Datenfluß-Architekturen	291
9.1	Übersicht	291
9.2	Grundlagen der Datenflußarchitekturen	293
9.2.1	Das reine Datenflußprinzip	293
9.2.2	Elementares Datenflußschema, elementarer Datenflußprozessor	297
9.2.3	Verallgemeinertes Datenflußschema	301
9.3	Statische Datenflußarchitekturen	303
9.3.1	Das Operationsprinzip der statischen Datenflußarchitektur	303
9.3.2	Die statische Datenflußarchitektur des MIT	305
9.3.3	Die LAU-Architektur	307
9.4	Dynamische Datenflußarchitekturen	311
9.4.1	Das Operationsprinzip der dynamischen Datenflußarchitektur	311
9.4.2	Herstellung der Laufzeitumgebung	313
9.4.3	Kennung der Informationseinheiten (<i>Tagged Token</i>)	314
9.4.4	Die Manchester-Maschine	316
9.4.5	Das Drosselungsproblem	319
9.5	Verfeinerungen der dynamischen Datenflußarchitektur	320
9.5.1	Zuordnung der Akteure zu den Prozessoren	320
9.5.2	Die I-Structure	320
9.5.3	Das Prinzip des expliziten Markenspeichers	323
9.5.4	Der Monsoon-Rechner	325
9.6	Hybride Datenflußarchitekturen	327
9.6.1	Nachteile der feinkörnigen Datenflußarchitekturen	327
9.6.2	Die Grundidee der hybriden Datenflußarchitekturen	328
9.6.3	Das LGDG-Prinzip einer Berechnung	329
9.6.4	Die LGDG-Maschine von <i>Dai</i>	334
9.7	Reduktionsmaschinen	338
	Literatur zu Kapitel 9	341
10	Grundlagen der MIMD-Architekturen	345
10.1	Allgemeine Gesichtspunkte	345
10.1.1	Die Hauptformen von MIMD-Architekturen	345
10.1.2	Programmiermodell und abstrakte Maschine	348
10.1.3	Die Granularität der Parallelarbeit	350
10.1.4	Das Kommunikationssystem	351
10.1.5	Kommunikationslatenz	352
10.1.6	Die Grundaufgabe des Verbindungsnetzes	354
10.1.7	Synchronisation zur korrekten Programmausführung	355
10.1.8	Strategien der Zuweisung der Aufträge zu den Knoten	359
10.2	Verbindungsnetz-Topologien	360
10.2.1	Kriterien für eine Taxonomie von Verbindungsnetzen	360
10.2.2	Beispiele für Verbindungsnetze	364
10.2.3	Busverbindungen	366
10.2.4	Ringstrukturen	369
10.2.5	Gitterverbindungen (Nächste Nachbarn)	371
10.2.6	Mehrstufige Verbindungsnetze	374
10.2.7	Hyperwürfel	377

10.2.8	Hierarchien von Crossbars	379
10.2.9	Eins-zu-N-Kommunikation in Verbindungsnetzen (Broadcast)	380
10.3	Verhaltensparameter von Verbindungsnetzen	381
10.3.1	Benötigte Verbindungsbandbreite	381
10.3.2	Aufwand und Verbindungslatenz	382
10.3.3	Blockierungsverhalten	383
10.4	Übersicht über die behandelten MIMD-Architekturen	389
	Literatur zu Kapitel 10	389
11	Systeme mit gemeinsamem Speicher	393
11.1	Schleifen-Parallelisierung	393
11.1.1	Das Ausführungsmodell	393
11.1.2	Das PAX-Protokoll	396
11.1.3	Der Parallelrechner Alliant FX/2800	399
11.2	Speicherkopplung mit vielen Kontrollfäden	401
11.2.1	Gründe für diese Architekturform	401
11.2.2	HEP-Architektur und TERA-Maschine	401
11.2.3	Die PRAM-Architektur	407
11.3	MIMD-Architekturen mit verteiltem gemeinsamem Speicher	410
11.3.1	Das Operationsprinzip	410
11.3.2	Beispiel: Die DASH-Architektur	412
11.4	Vielfädige Architekturen mit verteiltem Speicher	416
11.4.1	Gründe für vielfädige Architekturen	416
11.4.2	Das Operationsprinzip der vielfädigen Architekturen	416
11.4.3	Maßnahmen für schnelle Umgebungswechsel	417
11.4.4	Herstellung des globalen Adreßraums	419
11.4.5	Die *T-Maschine	419
11.4.6	Programmierbeispiel für die *T-Maschine	420
11.4.7	Der *T-Prozessor 88110MP	422
	Literatur zu Kapitel 11	424
12	MIMD-Architekturen mit verteiltem Speicher	427
12.1	Die großen Herausforderungen	427
12.2	Die Knotenarchitektur	430
12.2.1	Superskalare Knoten oder Vektorknoten?	430
12.2.2	Der Kommunikationsprozessor	433
12.2.3	Multiprozessor-Knoten	436
12.3	Verbindungsnetze für massiv-parallele Rechner	437
12.3.1	Crossbar-Topologien	437
12.3.2	Anschluß des Knotens	441
12.4	Synchrone oder asynchrone Kommunikation?	442
12.5	Das parallele Knoten-Betriebssystem	443
12.5.1	Die Rolle des Betriebssystems in nachrichtenorientierten Systemen	443
12.5.2	Die Struktur des parallelen Betriebssystems PEACE	444
12.5.3	Implementierung	447
12.5.4	Die PEACE-Familie von Betriebssystemkernen	448
12.6	Der virtuelle gemeinsame Speicher (VGS)	450

12.6.1	Das Operationsprinzip des virtuellen gemeinsamen Speichers	450
12.6.2	Architekturen mit virtuellem gemeinsamem Speicher	452
12.6.3	Ein Realisierungsmodell für sequentielle Konsistenz	453
12.6.4	Adaptive Konsistenz	456
12.7	Die Programmierung nachrichtenorientierter Systeme	460
12.7.1	Das nachrichtenorientierte Programmiermodell	460
12.7.2	Programmierwerkzeuge für nachrichtenorientierte Architekturen	464
12.7.3	Parallelisierende Compiler	467
12.8	Ausblick: Neue Programmiermodelle für massiv-parallele Systeme	469
	Literatur zu Kapitel 12	472
	Sach- und Namensverzeichnis	475