

Inhalt

Vorwort	1
1. Leistungssteigerung im Einzelprozessor	11
1.1 Der herkömmliche Einzelprozessor	11
1.2 Das Leistungsvermögen des Einzelprozessors	13
1.3 Wartezustände vermeiden	16
1.3.1 Ideale Betriebsverhältnisse beim Speicherzugriff	16
1.3.2 Speicherzugriffe beschleunigen	17
1.3.3 Speicher mit geringer Zugriffszeit	19
1.3.4 Nacheilendes Laden (Delayed Load)	21
1.3.5 Verschachtelte Zugriffe (Interleaving)	23
1.3.6 Mehrfache Zugriffsbreite	26
1.3.7 Voreilendes Lesen (Read Prefetch, Speculative Read)	27
1.3.8 Schreibpufferung (Write Posting)	28
1.4 Pipelining	30
1.5 Vektorverarbeitung	33
1.5.1 Skalare und Vektoren	33
1.5.2 Zwei Arten der Vektorverarbeitung	35
1.5.3 Herkömmliche Vektorverarbeitung	37
1.5.4 SIMD-Vektorverarbeitung	39
1.5.5 Lange und kurze Vektoren	42
1.6 Befehlspipelining	44
1.6.1 Befehlsphasen	44
1.6.2 Der Befehl in einem einzigen Taktzyklus	47
1.6.3 Befehlsphasen in eigenen Funktionseinheiten ausführen	47
1.6.4 Konflikte in der Befehlspipeline	53
1.6.5 Verzweigungen und Befehlspipelining	54
1.6.6 Konflikte in der Pipeline – Datenabhängigkeiten	58
1.6.7 Mehrbefehlspipelining (Superpipelining)	60
1.7 Der innewohnende (inhärente) Parallelismus	61
1.7.1 Den innewohnenden Parallelismus erkennen	62
1.7.2 Den innewohnenden Parallelismus ausnutzen	64
1.7.3 Den innewohnenden Parallelismus steuern	65
1.7.4 Implizite Parallelarbeit	67
1.7.5 Explizite Parallelarbeit	71
1.7.6 Mehrfachskalarverarbeitung (Superskalarverarbeitung)	73
1.7.7 Mehrbefehlspipelining (Superpipelining)	75

2. Parallelverarbeitung	79
2.1 Weshalb Parallelverarbeitung?	79
2.2 Ebenen der Parallelverarbeitung	83
2.3 Zur Taxonomie der Parallelverarbeitungssysteme	86
2.3.1 Computersysteme klassifizieren	86
2.3.2 Verbindungsprinzipien	88
2.3.3 Die Körnigkeit des Parallelismus	88
2.3.4 Die Arbeit aufteilen	88
2.4 Anwendungsprobleme parallelisieren	92
2.4.1 Praktisch vollkommen parallelisierbar	92
2.4.2 Hochgradig parallelisierbar	94
2.4.3 In mittlerem Grade parallelisierbar	95
2.4.4 Kaum oder gar nicht parallelisierbar	95
2.4.5 Die Arbeit aufteilen	95
2.4.6 Simultane Parallelverarbeitung	99
2.4.7 Nichtdeterministische Parallelverarbeitung	99
2.4.8 Spekulative Parallelverarbeitung	101
2.4.9 Parallelisierbarkeit als Eigenschaft des Algorithmus	103
2.5 Leistungsgewinn durch Parallelverarbeitung	108
2.6 Konflikte der Parallelverarbeitung	111
2.7 Mehr als ein Prozessor	115
2.7.1 Was ist ein Prozessor?	115
2.7.2 Prozessoren sinnvoll beschäftigen	116
2.7.3 Prozessoren beeinflussen	117
2.7.4 Physische und logische Systemstrukturen	120
2.7.5 Die Cache-Kohärenz	125
2.8 Mehrere kleine oder ein großer?	132
2.8.1 Grenzen der Kostensenkung	133
2.8.2 Kostenoptimierung in der Praxis	135
2.8.3 Hochleistungsmaschinen in Sparbauweise?	136
2.8.4 Mehrere Prozessoren oder Mehrprogrammbetrieb?	138
3. Rechnerarchitektur und Schaltungstechnik	143
3.1 Grundlagen	143
3.1.1 Signalflüsse in der Register-Transfer-Ebene	143
3.1.2 Das Taktsystem	153
3.1.3 Halbleitertechnologien	157
3.2 Maschinen aus programmierbaren Schaltkreisen	165
3.2.1 Programmierbare und hart verdrahtete Schaltungen	165
3.2.2 Software oder FPGA?	172
3.2.3 Hardware und Software gemeinsam entwickeln	175

3.3	Einzweckmaschinen	181
3.4	Rekonfigurierbare Hardware	186
3.5	Ressourcen im Verbund	193
3.5.1	Verarbeiten und transportieren	193
3.5.2	RISC: eine Feinstrukturanalyse	198
3.5.3	Pipelining	201
3.5.4	Pipelining oder Mehrfachverarbeitung?	216
3.5.5	Grundsätzliche Leistungsgrenzen	225
3.6	Grundsatzlösungen der Mehrfachverarbeitung	227
3.6.1	Eine Taxonomie	228
3.6.2	Befehlsverteilung	229
3.6.3	Befehlsumsetzung	229
3.6.4	Befehlswandlung	231
3.6.5	Befehlszuweisung	234
3.6.6	Mehrbefehlspipelining (Superpipelining)	241
3.7	Multitasking in der Hardware	242
4.	Universal- und Spezialmaschinen	251
4.1	Universalmaschinen erweitern und beschleunigen	251
4.1.1	Physische und programmseitige Schnittstellen	251
4.1.2	Beschleunigungseinrichtungen im Überblick	254
4.1.3	Was Beschleunigungseinrichtungen leisten	256
4.1.4	Beschleuniger oder Universalprozessor?	260
4.2	Maschinen für hochkomplexe Algorithmen	261
4.2.1	Architekturprinzipien	261
4.2.2	Mehrere kleine oder eine große?	263
4.2.3	Neue Universalmaschinen?	267
4.2.4	Universal- und Verbundmaschinen	269
4.3	Spezialmaschinen	270
4.3.1	Entwicklungsziele	270
4.3.2	Prinziplösungen	275
4.3.3	Leistungsklassen	278
4.3.4	Architekturkonzepte	282
4.4	Datenflußmaschinen	286
4.4.1	Steuerflußprinzip und Datenflußprinzip	286
4.4.2	Datenflußgraphen	287
4.4.3	Das universelle Datenflußprinzip der Theorie	296
4.4.4	Datenflußsteuerung in Prozessoren	304
4.5	Datenstrukturmaschinen	318

5. Universalrechner verbessern	325
5.1 Grundsätzliche Alternativen	325
5.1.1 Das prinzipielle Modell der Informationsverarbeitung.....	325
5.1.2 Leistungsgrenzen des herkömmlichen Universalrechners	326
5.1.3 Die Leistungsgrenzen überwinden	330
5.1.4 Zur Kritik des Universalrechners	333
5.1.5 Alternativen im Überblick.....	334
5.2 Alternative Befehlssatzarchitekturen	341
5.2.1 Elementare Befehlswirkungen	342
5.2.2 Komplexe Befehlswirkungen	345
5.2.3 VLIW-, Mikro- und Ressourcenvektorbefehle	349
5.2.4 Lange Befehle dimensionieren	352
5.3 Mikrobefehle	355
5.4 Ressourcenvektorbefehle	359
5.5 Alternativen der Programmablaufsteuerung	369
5.5.1 Die Programmablaufsteuerung verbessern.....	369
5.5.2 Steuerautomaten und Operationsautomaten.....	371
5.5.3 Spekulativ parallel arbeiten	382
5.6 Alternativen der Speicherorganisation.....	384
5.6.1 Speicher und Verarbeitung im Verbund.....	384
5.6.2 Aktive Speicher	388
5.6.3 Speicher in den Verarbeitungsressourcen	390
5.6.4 Die Wiederkunft des Umlaufspeichers?.....	391
5.6.5 Assoziative Speicher	394
5.7 Alternative Wirkprinzipien	399
5.7.1 Alternative Informationsstrukturen	399
5.7.2 Alternative Informationsdarstellung	403
5.7.3 Mehrwertige Logik, Schwellwertlogik	406
5.7.4 Asynchroner Betrieb	406
5.7.5 Seriell übertragen und verarbeiten	407
5.7.6 Analoge Signaldarstellung und Verarbeitung	407
Anhang 1: Historische Beispiele	413
1.1 IA-32: Historische Prozessoren der Personalcomputer	413
1.2 IA-64: Expliziter Parallelismus.....	421
1.3 AP120: Beispiel einer Datenstrukturmaschine	427
Anhang 2: Serielle Verarbeitungsschaltungen	432
2.1 Bitseriell verarbeiten	432
2.2 Serielle Mehrbitverarbeitung	436

Anhang 3: Datenflußschaltungen	441
3.1 Die Erfahrungsbasis	442
3.2 Das Speicherproblem	444
3.3 Datenflußgraphen und Operationswerke	448
3.4 Die Maschine im Überblick	454
3.5 Ablaufsteuerung	459
3.6 Die Datenflußschaltung als Pipeline	470
3.7 Eine erweiterte Topologie	474
3.8 Ein Blick in ein Operationswerk	477
Anhang 4: Universelle Datenflußmaschinen	480
4.1 Datenflußmaschinen implementieren	480
4.2 Ein Implementierungsbeispiel	485
4.3 Bedingungen und Prädikate	495
4.4 Weiterführende Überlegungen	500
4.4.1 Grundsätzliche Grenzen der universellen Datenflußmaschinen	500
4.4.2 Assoziativzellen weiterentwickeln	501
4.4.3 Assoziativzellen und Ressourcen	501
4.4.4 Das Verkettungsproblem	502
4.4.5 Assoziative oder adressierte Zugriffe?	506
4.4.6 Digitale Analogrechner	507
Literatur- und Quellenverzeichnis	509
Index	517