1. Die Organisation von RISC-Prozessoren 2

1.1. Superskalarität 3

1.2. Parallelverarbeitung 4

1.2.1 Multi Core- und Many Core-CPUs 11

1.2.2 Automatische Parallelisierung im Instruction Window 12

1.3. Dynamische Befehlsausführung 13

1.4. Die Fetch Unit 16

1.4.1 Core-interne Parallelität durch die Fetch Unit 18

1.5. Der Prefetch Buffer 20

1.6. Die Decode Unit 21

1.7. Der 1st Level-Befehlscache 21

1.8. Der 1st Level-Datencache 24

1.9. Die 2nd, 3rd Level Caches 24

1.10. Die Load- und Store Buffer 26

1.11. Die Kopplung von Speicher- mit Befehls-Pipeline 27

1.12. Die Verallgemeinerte Speicherhierarchie 29

1.13. Die Reservierungsregister 30

1.14. Das Scoreboard der vollen Ausbaustufe 32

1.14.1 Der Aufbau des Scoreboard der vollen Ausbaustufe 34

1.14.1.1 Die Liste der Befehlszustände 34

1.14.1.2 Die Liste der ALU-Zustände 36

1.14.1.3 Die Liste der Zustände der Ergebnisregister 37

1.15. Spekulative Befehlsausführung 37

1.15.1 Der Reorder Buffer 39

1.15.1.1 Zusammenfassung der Aufgaben des Reorder Buffers 41

1.15.1.2 Präzise Interrupts durch Reorder Buffer 42

1.15.1.3 Sonderrolle von Load/Store und Sprüngen 43

1.15.1.4 Aufbau des Reorder Buffers 43

1.16. Eager Execution 45

1.16.1 Nachteil von Eager Execution: 47

1.17. Register Renaming 48

1.18. Registerfenster 49

1.18.1 Häufigkeitsverteilung von Befehlen in C-Programmen 51

1.18.1.1 Verschachtelungstiefe 52

1.18.2 Zeitliche Analyse der Verschachtelungstiefen 53

1.18.3 Linear angeordnete Register 57

1.18.4 Zirkular angeordnete Register 58

1.18.5 Zirkulare Register mit Unit-Adressengenerator 61

1.18.6 Bewertung von Registerfenstern 62

1.19. Beispiel für RISC-Prozessorarchitekturen 62

1.19.1 Einfache, superskalare, Tomasulo-basierte CPU 63

1.19.2 RISC-CPU mit spekulativer Befehlsausführung 64

1.19.3 Gesamtschaltbild RISC-CPU 65

1.19.4 Speicherverwaltungseinheiten (MMUs) für RISC-CPUs 66
1.19.4.1 Beispiel des RISC-Prozessors HP Alpha 21264 67
1.19.4.2 Kombinierte RISC/CISC-Architektur der Intel x-86 CPUs 69

2. Die Organisation von Parallelrechnern 70

2.1. Multi Core-CPU 70
2.2. Warum gibt es Multicore-CPUs? 71
2.3. Programmierung von Multicore-CPUs 72
2.4. Many Core-CPU 72
2.5. Multiprozessoren 73
2.6. Multicomputer 73
2.7. Cluster-Computer 74
2.8. Parallelrechner 75
2.9. Warum gibt es Parallelrechner? 75
2.10. Ziele beim Entwurf eines Parallelrechners 77
2.11. Abgrenzung gegenüber verteilten Systemen auf LAN-Basis 77
2.12. Was sind die wesentlichen Punkte bei Parallelrechnern? 78
2.13. Kategorien bei der Parallelrechnerorganisation 79
2.14. Definition Verbindungsnetzwerk 80
2.15. Uniform Memory Access Computer (UMA) 81
2.15.1 Bus/Speicherkopplung 82
2.15.2 Sättigungseffekt bei der Bus-Speicher-Kopplung 84
2.15.3 Erhöhung der Bandbreite bei der Bus/Speicherkopplung 86
2.15.3.1 Konsistenzproblem bei Cache-Kopien 90
2.15.4 Bewertung der Beschleunigungsmaßnahmen bei Bus/Speicher-Kopplung 92
2.15.5 Kosten/Nutzen-Analyse der Beschleunigungsmaßnahmen 94
2.16. Non-Uniform Memory Access Computer (NUMA) 95
2.16.1 Beschleunigung durch Lokalspeicher 95
2.16.2 Beschleunigung durch lokale Caches 97
2.16.3 Prozessor/Core-Kopplung über Multiport Memory 98
2.16.3.1 Beispiel für Multiport Memory-Kopplung 99
2.16.4 Parallelbussysteme 99
2.16.5 2D-Busmatrizen (Kreuzschienenverteiler) 101
2.16.6 Hierarchische Bussysteme 102
2.17. Skalierbarkeit von Verbindungsnetzwerken 103
2.18. Übersicht zur Programmierung von Parallelrechnern 105
2.18.1 Funktionsparallelisierung versus Datenparallelisierung 106
2.18.2 Modelle der Kommunikation 107
2.18.3 Testen paralleler Programme 107
2.18.4 Implementierung der Kommunikationsmodelle 109
2.18.5 Zusammenfassung Parallelrechnerprogrammierung 110

3. Grundlagen statischer und dynamischer Netze 110

3.1. Eng gekoppelte Systeme 114
3.1.1 DSM UMA-Architektur 114
3.1.2 DSM NUMA-Architektur 116
3.1.2.1 Transparenter Zugriff auf entfernten Hauptspeicher 117
3.1.3 SVM NUMA-Architektur 119
3.1.4 COMA-Architektur 121
3.1.5 Programmierung eng gekoppelter Systeme 122
3.1.5.1 Zugriffssynchronisation für gemeinsame Variable 123
3.1.5.2 Semaphore 124
3.1.5.3 Peer-to-Peer-Kommunikation über Semaphore 125
3.1.5.4 Netzwerkweite Semaphore (=Entfernte Semaphore) 126
3.1.5.5 Monitore 127
3.1.5.6 Critical Sections 128
3.2. Lose gekoppelte Systeme 128
3.2.1 Botschaftenaustausch 129
3.2.1.1 Virtuelle Kommunikationskanäle mittels D-Controller 130
3.2.2 Programmierung lose gekoppelter Systeme 132
3.2.2.1 Synchrone/asynchrone Kommunikation über Botschaften 132
3.2.2.2 Prozess-Synchronisation durch Botschaften 132
3.2.2.3 Produzenten und Konsumenten von Daten 136
3.2.2.4 Peer-to-Peer-Kommunikation über Rendezvous 137
3.2.2.5 Bewertung von READ/WRITE und von (a)synchronem SEND/ RECEIVE 137
3.3. Erweiterte Konstruktionsprinzipien von Netzen 138
3.3.1 Parallelschaltung 138
3.3.2 Hierarchie 140
3.3.3 Rekursion 141
3.3.4 Modularisierung 141
3.3.5 Modularisierung und Rekursion gemeinsam 142
3.4. Verbindungstypen bei eng und lose gekoppelten Systemen 143
3.5. Datentransport bei eng und lose gekoppelten Systemen 147
3.5.1 Rahmen/Pakete- und Nachrichtenformate 148
3.5.2 Paket-Routing bzw. Rahmen-Switching 149
3.5.3 Transportart/Flusssteuerung 151
3.5.3.1 Store-and-Forward Routing 151
3.5.3.2 Virtual-Cut-Through Routing 152
3.5.3.3 Wormhole Routing 153
4. Statische Verbindungsnetzwerke 153
4.1. Symmetrie bei statischen Netzen 158
4.2. Metriken bei statischen Netzen 159
4.3. Konstruktion eines n-dimensionalen Überwürfels (Hypercube) 161
4.4. Konstruktion eines de Bruijn-Graphen 162
4.5. Routing in statischen Netzen 164
4.6. Das Deadlock-Problem bei Interprozessor-Kommunikation 165
4.6.1 Verklemmung aufgrund eines belegten Kanals 166
4.6.2 Verklemmung aufgrund zweier belegter Puffer 166

4.6.3 Verklemmung aufgrund vier belegter Puffer 167
4.6.4 Verklemmung aufgrund vier belegter Kanäle 167
4.6.5 Verklemmung trotz getrennter Sende- und Empfangspuffer 168

5. Dynamische Verbindungsnetzwerke 169

5.1. Permutationsfunktionen 169

5.1.1 Die Perfect Shuffle-Permutation 170
5.1.2 Die Butterfly-Permutation 171
5.1.3 Die Reversal-Permutation 172
5.1.4 Die inverse Perfect Shuffle-Permutation 173
5.1.5 Die Supershuffle-Funktion 174
5.1.6 Die Subshuffle-Funktion 175
5.1.7 Die Super-/Subbutterfly-Funktion 176
5.1.8 Ternäre und quaternäre Shuffle-Permutation 178

5.2. Kreuzschalter 179

5.2.1 Kreuzschalter mit Broadcast 179
5.2.2 Kreuzschalter als Kreuzschienenverteiler 180
5.2.3 Die Verbindungsmöglichkeiten des 2x2 Kreuzschienenverteilers 181
5.2.4 Ein fxs-Schalter und ein fxs-Kreuzschienenverteiler sind identisch 182
5.2.5 Funktion eines 2x2-Kreuzschalters beim Durchgang eines Datenrahmens 183
5.2.6 Die Exchange-Permutation ε 184
5.2.7 Die Subexchange-Permutation 185

5.3. Die klassischen LogN-Netze 186

5.3.1 Das Shuffle-Exchange-Netzwerk 186
5.3.2 Das Omega-Netz 187
5.3.2.1 Self Routing beim Omega-Netz 188
5.3.2.2 Kollision zweier Pfade im Omega-Netz 188
5.3.3 Das Flip-Netz 189
5.3.3.1 Analogie zwischen dem Flip-Netz und dem Signalflussgraph der Pease FFT 190
5.3.3.2 Analogie zwischen dem Flip-Netz und der Transposition einer Matrix 191
5.3.4 Das Indirect Binary n-Cube-Netz 192
5.3.4.1 Funktion eines 16x16 Indirect Binary n-Cube-Netzes 193
5.3.4.2 Analogie zwischen dem Indirect Binary n-Cube-Netz und dem Hypercube 195
5.3.4.3 Analogie zwischen dem Indirect Binary n-Cube-Netz und der Cooley-Tukey FFT 199
5.3.5 Das Generalized Cube-Netz 200
5.3.5.1 Routing im Generalized Cube-Netz 201
5.3.6 Das Baseline-Netz 202
5.3.6.1 Routing im Baseline-Netz 203
5.3.6.2 Funktion des Baseline-Netzes (N=16, n=4) 204
5.3.7 Das inverse Baseline-Netz 205
5.3.7.1 Funktion des inversen Baseline-Netzes 206
5.3.8 Der Butterfly-Banyan 207
5.3.9 Der inverse Butterfly-Banyan 208

5.3.10 Definitionsgleichungen der logN-Netze 209
5.3.11 Routing in logN-Netzen 210
5.3.12 Äquivalenz der logN-Netze 211
5.3.12.1 Umwandlung eines Flip-Netzes in ein Omega-Netz 214
5.3.12.2 Umwandlung eines Omega-Netzes in ein Butterfly-Banyan 215
5.4. Allgemeine Banyans 216
5.5. Regelmäßige Banyans 223
5.5.1 Einziger Banyan der Größe n=1 und f=s=2 223
5.5.2 Verallgemeinerung zum (f,s,1)-Banyan ist ein fxs-Kreuzschienenverteiler 223
5.6. Klassifikation der Banyans 224
5.7. Regelmäßige und rechteckige Banyans 224
5.7.1 Die SW-Banyans 226
5.7.1.1 Der (2,2,3)-SW-Banyan 226
5.7.1.2 Der (2,2,3)-SW-Banyan in der Aktivknoteninterpretation 227
5.7.1.3 Funktionsweise des (2,2,3)-SW-Banyans in der Aktivknoteninterpretation 228
5.7.1.4 Der (2,2,3) Banyan in der Passivknoteninterpretation 229
5.7.1.5 Konstruktion des (2,2,3) Banyans in der Passivknoteninterpretation 230
5.7.1.6 Funktionsweise des Netzes $N_{\textbf{neu 231}}$
5.7.1.7 Topologierhaltende Umwandlung des Netzes $N_{\textbf{neu 232}}$
5.8. Anwendungen von SW-Banyans 233
5.8.0.1 Parallelrechner mit 64-Prozessoren in „Fat Tree-Topologie“ 233
5.8.1 Topologie des (4,2,2)-Banyans mit N_1=32 und N_2=16 und N_3=8 235
5.8.2 Systematische Konstruktion von Banyans 236
5.8.2.1 Sequenz der ersten drei SW-Banyans (2,2,1), (2,2,2) und (2,2,3) 236
5.8.2.2 Additive Konstruktion eines (2,2,2)- SW-Banyans aus 4 Binärbäumen 236
5.8.2.3 Rekursive Konstruktion eines (2,2,2)-SW-Banyans aus (2,2,1)-SW-Banyans 237
5.8.3 Routing in einem regelmäßigen und rechteckigen Banyan 238
5.8.3.1 Routing im (2,2,n)-SW-Banyan 238
5.8.4 Die CC-Banyans 239
5.8.4.1 Routing im (2,2,n)-Cylindrical Cross Hatched-Banyan 243
5.9. Das Clos-Netz 244
5.10. Das Benes-Netz 249
5.11. Das doppelte Baseline-Netz nach Wu und Feng 254
5.12. Das Lee-Netz 255