

<b>1. Übersicht zu den Prozessorfamilien</b>	<b>2</b>
<b>2. Grundlagen der Rechnerorganisation</b>	<b>3</b>
<b>2.1. Aufbau eines Rechners in Ebenen</b>	<b>3</b>
<b>2.2. Die Ebene der elektronischen Bauelemente</b>	<b>5</b>
<b>2.3. Die Gatterebene</b>	<b>5</b>
<b>2.3.1 Einfache Gatter</b>	<b>6</b>
2.3.1.1 Schaubilder und Wertetafeln einfacher Gatter	6
<b>2.3.2 Elementare Gatter</b>	<b>7</b>
<b>2.3.3 EXOR-Gatter</b>	<b>8</b>
<b>2.4. Die Ebene der Funktionsblöcke</b>	<b>9</b>
<b>2.4.1 3-zu-8-Dekodierer</b>	<b>11</b>
<b>2.4.2 8-zu-1-Multiplexer</b>	<b>12</b>
<b>2.4.3 4-Bit-Komparator</b>	<b>13</b>
<b>2.4.4 1-Bit-Volladdierer</b>	<b>14</b>
<b>2.4.5 N-Bit-Addierer</b>	<b>15</b>
<b>2.4.6 N-Bit-Addierer/Subtrahierer</b>	<b>17</b>
<b>2.4.7 Einfache logische Einheit</b>	<b>18</b>
<b>2.4.8 Einfache arithmetisch/logische Einheit</b>	<b>19</b>
<b>2.4.9 8-Bit-ALU</b>	<b>20</b>
<b>2.4.10 Verallgemeinerte ALUs</b>	<b>21</b>
<b>2.4.11 Flip Flops</b>	<b>22</b>
2.4.11.1 1-Bit-Registerspeicher mit Setzen/Rücksetzen	23
2.4.11.2 Taktgesteuertes D-Latch	24
2.4.11.3 Taktflankengesteuertes D-Flip-Flop	26
2.4.11.4 Kommerzielle Registerspeicher	27
<b>2.4.12 Zähler</b>	<b>29</b>
2.4.12.1 Modulo-k-Zähler	29
2.4.12.2 Asynchroner Modulo-k-Zähler	30
2.4.12.3 Synchroner Modulo-k-Zähler	32
<b>2.4.13 Synthese von Funktionsblöcken</b>	<b>34</b>
<b>2.4.14 Schaltwerksynthese am Beispiel eines Synchronzählers</b>	<b>35</b>
2.4.14.1 Aufstellen der Wahrheitstabelle aus den Zählerzuständen	35
2.4.14.2 Bestimmung der Übergangsfunktion	36
2.4.14.3 Vereinfachen der Übergangsfunktion	37
2.4.14.4 Realisierung der vereinfachten Übergangsfunktion	38
2.4.14.5 Verbinden des Schaltnetzes mit dem Register	38
2.4.14.6 Anwendungen von Synchron-Zählern	40
<b>2.4.15 Synchron-Zähler mit permutierten Zählfolgen</b>	<b>40</b>
<b>2.4.16 Endliche Automaten</b>	<b>41</b>
2.4.16.1 Endliche Automaten mit Eingabe	42
2.4.16.2 Implementierung von endlichen Automaten mit Eingabe	44
2.4.16.3 Endliche Automaten mit Ein- und Ausgabe	45
2.4.16.4 Moore-Automat	46
2.4.16.5 Mealy-Automat	48
<b>2.5. Die Komponentenebene</b>	<b>49</b>
<b>2.5.1 Hauptspeicher</b>	<b>50</b>
2.5.1.1 Linear adressierte Speicherzellen	50
2.5.1.2 SRAM- und DRAM-Speicher	51

2.5.1.3	Kommerzielle SRAM und DRAM-Speicherchips	52
2.5.1.4	Unterschiede zwischen SRAM und DRAM	53
2.5.1.5	Aufbau eines SRAMs	55
<b>2.5.2</b>	<b>Sonstige Speicher: EEPROMs, Flash RAMs und MRAMs</b>	<b>56</b>
<b>2.5.3</b>	<b>Busse</b>	<b>57</b>
2.5.3.1	Bussysteme im Rechner	58
2.5.3.2	Adress-, Daten- und Steuerbusleitungen	59
<b>2.5.4</b>	<b>Peripherie</b>	<b>61</b>
2.5.4.1	PIO	61
2.5.4.2	Anschluss von PIO und Peripherie an CPU-Busse	64
2.5.4.3	Ansteuerung eines peripheren Bausteins durch den Prozessor	65
2.5.4.4	Festplatten	66
2.5.4.5	Mechanischer Aufbau einer Festplatte	68
<b>2.6.</b>	<b>Die Ebene der Rechnerorganisation</b>	<b>69</b>
<b>2.6.1</b>	<b>Die von-Neumann-Maschine</b>	<b>69</b>
2.6.1.1	Arbeitsweise der von-Neumann-Maschine	71
<b>2.6.2</b>	<b>Die Harvard-Architektur</b>	<b>72</b>
<b>2.6.3</b>	<b>Verbesserungen der von-Neumann-Maschine</b>	<b>74</b>
2.6.3.1	Der von-Neumann-Flaschenhals	75
2.6.3.2	Der Akkumulator als schneller Zwischenspeicher (historisch)	76
2.6.3.3	Busse als wesentliche Elemente der Rechnerorganisation	77
2.6.3.4	Peripheriebus und Systembus	78
2.6.3.5	Steuereinheiten für die Peripherie	79
<b>2.6.4</b>	<b>Organisation eines fortgeschritteneren Rechners</b>	<b>81</b>
2.6.4.1	Embedded Controller	83
<b>2.6.5</b>	<b>Abläufe in einem Rechner</b>	<b>85</b>
<b>2.6.6</b>	<b>Adressierung der Peripherie</b>	<b>85</b>
2.6.6.1	Eigener I/O-Adressraum	86
2.6.6.2	Memory-Mapped I/O	87
2.6.6.3	Bewertung von Memory Mapped- bzw. eigenem I/O-Adressraum	87
<b>2.6.7</b>	<b>Polling</b>	<b>87</b>
2.6.7.1	Vorteile und Nachteile von Polling	88
<b>2.6.8</b>	<b>Interrupts</b>	<b>89</b>
2.6.8.1	Hardware-mäßige Organisation eines Interruptsystems	90
2.6.8.2	Auftreten eines Interrupts	90
2.6.8.3	Ablauf eines Interrupts	92
2.6.8.4	Aufruf der Interrupt-Service-Routine	94
2.6.8.5	Vektorisierter Interrupt	94
2.6.8.6	Erweitertes Interruptsystem	97
2.6.8.7	Vor- und Nachteile von Interrupts	98
<b>2.6.9</b>	<b>Direct Memory Access (DMA)</b>	<b>98</b>
2.6.9.1	Eigenschaften von DMA	99
2.6.9.2	DMA-Controller	100
2.6.9.3	Initialisierung des DMA-Controllers	100
2.6.9.4	DMA-Transfer	101
2.6.9.5	Zeitlicher und funktionaler Ablauf eines DMA-Transfers	102
2.6.9.6	Zwei Implementierungsvarianten von DMA	104
<b>3.</b>	<b>Die Zentraleinheit (CPU)</b>	<b>104</b>

- 3.1. CPU-Register 105**
- 3.2. CPU-interne Busse 106**
- 3.3. Die Steuereinheit (Control Unit) 108**
  - 3.3.1 Mikroprogrammierte Steuereinheit 109**
- 3.4. Interaktion zwischen den CPU-Komponenten 111**
  - 3.4.1 Die Befehlsholphase der CPU 112**
  - 3.4.2 Die Phasen „Operand holen“ und „Befehl ausführen“ 113**
  - 3.4.3 Zusammenspiel von Steuereinheit, PC, IR, MAR und MDR 114**
  - 3.4.4 Zusammenspiel von CPU mit Hauptspeicher 115**
  - 3.4.5 Zusammenspiel von CPU, Hauptspeicher und Schaltern 116**
- 3.5. Die Hauptbefehlsschleife aus der Sicht des Steuerwerks 117**
- 4. CISC-Prozessoren (ca.1971 - 1990) 118**
  - 4.1. Der Befehlssatz eines CISC-Prozessors 119**
  - 4.2. Adressierungsarten bei CISC- und RISC-CPU's 123**
  - 4.3. Orthogonaler Befehlssatz 125**
  - 4.4. Architekturen bei CISC-Prozessoren 126**
    - 4.4.1 Stack-Architektur 126**
      - 4.4.1.1 Bewertung der Stackarchitektur 127
    - 4.4.2 Register/Speicher-Architektur 128**
      - 4.4.2.1 Bewertung der Register/Speicher-Architektur 129
  - 4.5. Heutiger Status der CISC-Prozessoren 131**
- 5. Caches 132**
  - 5.1. Speicherhierarchie 132**
  - 5.2. Funktionsprinzip von Caches 133**
    - 5.2.1 Befehlsungleichverteilung 134**
    - 5.2.2 Lokalität von Befehlen und Daten 135**
  - 5.3. Der Aufbau von Caches 137**
    - 5.3.1 Vollasoziativer Cache 139**
    - 5.3.2 Direkt-abbildender Cache 140**
      - 5.3.2.1 Arbeitsweise des direkt-abbildenden Cache 142
      - 5.3.2.2 Adressabbildung beim direkt-abbildenden Cache mit k Worten 143
    - 5.3.3 M-Wege Mengenassoziativer Cache 144**
  - 5.4. Datentransfer zwischen Cache und Hauptspeicher 147**
    - 5.4.1 Fall 1: CPU möchte lesen 148**
      - 5.4.1.1 Ersetzungsstrategien bei voll- und bei mengenassoziativem Cache 149
    - 5.4.2 Fall 2: CPU möchte einen Befehl oder ein Datum schreiben 150**
      - 5.4.2.1 Zu schreibendes Hauptspeicherwort ist als Kopie im Cache (Cache Hit) 151
      - 5.4.2.2 Zu schreibendes Hauptspeicherwort ist nicht als Kopie im Cache (Cache Miss) 152
    - 5.4.3 Bewertung der CPU-Schreibstrategien in den Hauptspeicher 153**

- 5.4.4 **Das Paging-Problem bei Write Back und Fetch-on-Write 154**
  - 5.4.4.1 Übersicht zum Paging 154
  - 5.4.4.2 Lösung des Paging-Problems 156
- 5.4.5 **Das IO-Problem bei Write Back und Fetch-on-Write 158**
- 5.4.6 **Zusammenfassung Datentransfer 159**
- 5.5. **Split Caches (Teilung des Cache) 159**
- 6. **Virtueller Hauptspeicher und Speicherverwaltung 160**
  - 6.1. **Problem der nicht zusammenhängenden Adressräume 161**
  - 6.2. **Lösung von Problem 1 163**
  - 6.3. **Adressabbildung durch die MMU 165**
  - 6.4. **Die Größe von virtuellem Hauptspeicher 165**
  - 6.5. **Verdrängung einer Hauptspeicherseite 167**
    - 6.5.1 **Vorteile von Virtual Memory 169**
    - 6.5.2 **Prozessor- und Hauptspeicheradressen, Cache- und Page-Adressierung 170**
    - 6.5.3 **Einordnung der MMU in die Schichten und Module eines Rechners 171**
    - 6.5.4 **Praktische Implementierung von virtuellem Speicher 174**
    - 6.5.5 **Adressabbildung über Seitentabellen 176**
      - 6.5.5.1 Schritte der Adressabbildung 177
    - 6.5.6 **Adressabbildung mit TLB-Cache 180**
    - 6.5.7 **Mindestanforderungen an Page Table und TLB 180**
      - 6.5.7.1 **Erweiterte Anforderungen an den TLB-Cache 182**
  - 6.6. **Speicherschutz durch MMU und Betriebssystem 182**
  - 6.7. **Zusammenfassung Virtual Memory 183**
- 7. **RISC-Prozessoren 184**
  - 7.1. **Definition der Prozessorleistung 185**
  - 7.2. **Architektonische Unterschiede RISC-CISC 186**
  - 7.3. **Kommerzielle Beispiele von RISC-Prozessoren 188**
  - 7.4. **Zusammenfassung RISC-Design-Prinzipien 188**
- 8. **Beschleunigung der Befehlsausführung bei RISC-Prozessoren 189**
  - 8.1. **Pipelining 190**
    - 8.1.1 **Pipeline-Prinzip 192**
    - 8.1.2 **Mit und Ohne Pipelining 193**
    - 8.1.3 **Raum-Zeitdiagramm für den Durchlauf eines Befehls 195**
    - 8.1.4 **Vergleich ohne/mit Pipelining 196**
    - 8.1.5 **Maße bei einer Pipeline 197**
      - 8.1.5.1 **Speedup (Beschleunigung) 198**
    - 8.1.6 **Arithmetisches-, Phasen- und Befehls-Pipelining 199**
    - 8.1.7 **Superpipelining 200**
      - 8.1.7.1 **Vorteil von Superpipelining 201**
      - 8.1.7.2 **Nachteil von Superpipelining 202**

- 8.1.8 Blasenfreie Pipeline 202**
- 8.1.9 Steuerflusskonflikte 202**
- 8.2. Sprungvorhersage (Branch Prediction) 206**
  - 8.2.1 Branch Unit 207**
  - 8.2.2 Branch Target Cache (Sprungzielspeicher) 208**
  - 8.2.3 Aufgabenverteilung Sprungvorhersage/Sprungzielvorhersage 209**
  - 8.2.4 Vorteil von Sprungvorhersage + Sprungzielvorhersage 212**
  - 8.2.5 Nachteil von Sprungvorhersage + Sprungzielvorhersage 212**
  - 8.2.6 Maximale Beschleunigung der Befehlsausführung 212**
  - 8.2.7 Ergänzungen zum Sprungzielspeicher (Branch Target Cache) 219**
  - 8.2.8 Probleme bei der Sprungzielvorhersage 220**
  - 8.2.9 Zusammenfassung Branch Unit und Branch Target Cache 221**
  - 8.2.10 Wie funktioniert die Sprungvorhersage? 222**
  - 8.2.11 Statische Sprungvorhersage 224**
    - 8.2.11.1 Prepare-to-branch-Technik 224
    - 8.2.11.2 Assume Backward Branches as Taken 225
  - 8.2.12 Dynamische Sprungvorhersage 225**
    - 8.2.12.1 Sprungvorhersage mit Branch History Table und endlichem Automaten 226
    - 8.2.12.2 Alternative Implementierung des EA 231
    - 8.2.12.3 Verbesserte adaptive Sprungvorhersage 232
    - 8.2.12.4 Prinzip der verbesserten adaptiven Sprungvorhersage 233
    - 8.2.12.5 Ablauf der verbesserten adaptiven Sprungvorhersage 233
    - 8.2.12.6 Blockdiagramm der verbesserten adaptiven Sprungvorhersage 235
  - 8.2.13 Mehrfache Sprungzielvorhersage 235**
  - 8.2.14 Sprungzielvorhersage bei Pentium II und PowerPC 604e 237**
    - 8.2.14.1 PowerPC 604e 237
    - 8.2.14.2 Pentium II 239
  - 8.2.15 Aktuelle Methoden der Sprungvorhersage und der Sprungzielvorhersage 240**
    - 8.2.15.1 Gshare-Methode 241
    - 8.2.15.2 Agrees-Methode 243
  - 8.2.16 Allgemeine Methoden zur Steigerung der Trefferquote bei der Sprungvorhersage 243**
  - 8.2.17 Prädikation 244**
  - 8.2.18 Eager Execution-Technik 245**
  - 8.2.19 Was kostet es, wenn die Sprungvorhersage fehlschlägt? 246**
    - 8.2.19.1 Modellierung der Wirklichkeit 246
    - 8.2.19.2 Elimination von b 249
    - 8.2.19.3 Elimination von n 250
    - 8.2.19.4 Elimination von t bzw. f 251
    - 8.2.19.5 Elimination von k 252
    - 8.2.19.6 Elimination von p und q 252
    - 8.2.19.7 Resultat 253

## **9. Grenzen der Beschleunigung im Falle von Pipelining 253**

- 9.1. Liste der Pipeline-Probleme 254**
- 9.2. Ressourcenkonflikt 255**

### **9.3. Datenflusskonflikte 256**

#### **9.3.1 Kategorisierung der Datenflusskonflikte 257**

9.3.1.1 RAW-Datenflusskonflikt 257

9.3.1.2 Write-After-Read-Konflikt (WAR) 260

9.3.1.3 Write After Write-Konflikt (WAW) 261

## **10. Hinausschieben der Pipeline-Grenzen durch Scoreboarding und Tomasulo 262**

### **10.1. Konfliktbehebung in der Pipeline 264**

#### **10.2. Scoreboard 265**

#### **10.3. Der Tomasulo-Mechanismus 266**

10.3.1 Operand Forwarding 268

10.3.2 Reservierungsregister 268

10.3.3 Common Data Bus 270

10.3.4 Vereinfachter Aufbau der Reservierungsregister 271

10.3.5 Ablauf einer Berechnung mit Reservierungsregistern 272

10.3.6 Realer Aufbau der Reservierungsregister 272

#### **10.4. In-Order-Execution 275**

10.4.1 Zusammenfassung 278