

# 1. RECHNERTECHNIK

## A. CISC-RISC-SPEICHER

### CISC – Complex Instruction Set Computer

→ erster Trend

→ immer komplexer (Aufbau, Befehlsstruktur)

ab 1980 gegenläufiger Trend

### RISC – Reduced Instruction Set Computer

- Implementierung der Prozessorarchitektur vereinfacht
- Befehlssatz  $\leq 128$
- Befehls-Formate  $\leq 4$
- Befehlslänge 128Bit
- Adressierungsarten  $\leq 4$ 
  - Steuerwerk einfacher, mehr Platz
- Mind. 32 allg. verwendbare Register
- Speicherzugriff nur durch Lade-/Speicherbefehle  
(alle anderen Befehle: Operanten aus Register, Ergebnis im Register gespeichert)
  - bis Heute
- Pro Prozessortakt sollte ein Befehl beendet sein  
(Befehlspipelining statt Mikroprogrammierung.)
- Heute ist Befehlspipelinekontrolle nicht mehr in SW
- Inzw. RISC auch bei Architekturen mit Gleitpunktbefehlen
  - benötigen je Befehl 3 Takte
  - da Gleitpunktbefehle intern wie Pipeline behandelt werden
    - pro Takt 1 Ergebnis
- Skalares RISC → wenn RISC Ziel erreicht (im Durchschnitt 1Takt pro Befehl)
- Superskalar → pro Takt bis zu 4 Befehle

## B. CACHE

→ kleiner, schneller Pufferspeicher; enthält Kopien des wohl als nächstes vom Prozessor zugegriffenen Teils des Hauptspeichers

→ Zugriff annähernd so schnell wie auf Register

- Primärcache, direkt auf Chip
- Sekundärcache → auf SDRAM-Technologie (teuer)

→ nur kleiner Teil des Hauptspeichers

→ Zugriff auf Hauptspeicher wird durch Cache verringert

### Speicherverwaltung

→ Inhalt Cache enthält normal Datum des nächsten Zugriffs

→ muss sehr schnell sein → meist vollständig in HW realisiert

### Cache-Controller

→ kopiert Daten autom. in Cache

→ Cache füllt sich

→ schneller Zugriff

### Cache-Treffer

→ angefordertes Datum ist im Cache vorhanden

### Cache Fehler

→ angefordertes Datum nur im Hauptspeicher

→ wird direkt in Cache kopiert

→ unabhängig von der sonstigen Verwaltung

→ etwas langsamer

### Blockrahmen

→ Reihe von Speicherplätzen (Adresstikett, Statusbit)

### Blocklänge

→ Anzahl Speicherplätze im Blockrahmen

### Block

→ Datenportion die in Rahmen passt

(Menge die zw. Cache und Hauptspeicher übertragen wird (16 oder 32 Byte))

### Sätze

→ Menge der Rahmen in Cache

### Assoziativität

→ Anzahl Rahmen pro Satz

→ jeder Block nur in einem Satz, dort in beliebigem Rahmen

### Gesamtzahl der Blockrahmen c

$c = s(\text{Satz}) * n$  (Assoziativität)

### Verschiedene Cacheorganisationen

Vollassoziativ →  $s=1; n=c$   
direkt abgebildet →  $n=1; s=c$   
n-fach assoziativ →  $s = c/n$

### Kleine Assoziativität

→ einfachere Implementierung, da nur wenige Rahmen pro S durchsucht werden müssen

### Verdrängungsstrategie

→ HW, welcher Teil wird bei Fehler überschrieben

### LRU

→ bei assoz. Organisation → ersetzt die am längsten nicht benutzte Speicherportion

n-fach assoziativ

→ S. 55, 56

## **C. PIPELINING**

→ Zerlegung einer Maschinenoperation in Teiloperationen

→ werden taktsynchron von hintereinandergeschalteten Verarbeitungseinheiten bearbeitet

→ jede Verarbeitungseinheit für eine spez. Teiloperation)

### Befehlsbereitstellungsphase

→ Befehl (durch Zähler adressiert) wird aus Hauptspeicher oder Cache in Befehlspeicher geladen

→ Befehlszähler ++

### Decodierphase

→ aus Operationscode/Maschinenbefehl werden Prozessorientierte Steuersignale

### Operandenbereitstellungsphase

→ Operanden werden aus Registern bereitgestellt

### Ausführungsphase

→ Operation wird auf Operanden ausgeführt

### Resultatsspeicherphase

→ Ergebnis wird in Register geschrieben

### Superpipelining

→ feinere Stufen

→ schnellerer interner Takt

→ erlaubt sehr hohe Taktfrequenz für Prozessorchip

→ niedrige Taktfrequenz bei Gesamtsystem → dadurch billig

### Gleitpunktverarbeitungswerke

→ 3 Phasen für eine Operation da zu komplex

### Vektorpipelines

→ mit Vektorbefehl werden mehrere Gleitpunktoperationen auf den Wertepaaren eines Arrays gesteuert

Wegen versch. Befehlsarten bei Superskalaren Rechnern mehrere Pipelines (Befehlsbereitstellungs-, Decodier-, Befehlszuordnungsphase bei allen)

### Superskalare Befehlspipeline

- zur Nutzung der Parallelität 3er Technologien:
  - Superskalar
  - Superpipelining
  - VLIW (Very Long Instruction Word)

### Superskalar

- mehr als ein Befehl pro Takt an Ausführungseinheit zugeordnet
  - Befehlszuordnung arbeitet auf normalen sequentiellen Befehlsstrom
  - Zuordnung durch Befehlszuordnungseinheit, die die Befehle einer spez. Einheit gruppiert und zuordnet wenn Verarbeitung möglich
- Vorraussetzung:
- mehrere Ausführungseinheiten können parallel arbeiten
- Zuordnung:
- 2fach, 3fach, 4fach, vielfach superskalar (je nach max. Befehle je Takt)

Seiten 25,26 Script Bsp. Zu 88110

### Superskalar-Techniken:

- Befehlszuordnung(BZ) und -beendigung in Programmreihenfolge(PR)
- BZ in PR, Befehlsbeendigung außerhalb der Reihenfolge
- BZ außerhalb der PR, Befehlsbeendigung in PR
- BZ und Befehlsbeendigung außerhalb der PR

### Rückordnungspuffer

(wiederherstellen der PR bei Befehlsbeendigung)

- FIFO-Puffer
- Ergebnis nicht direkt in Zielregister sondern im Rückordnungspuffer

Seiten 27,28 Script Bsp. Zu 88110