
Rechnerstrukturen

Vorlesung im Sommersemester 2006

Prof. Dr. Wolfgang Karl

Universität Karlsruhe (TH)

Fakultät für Informatik

Institut für Technische Informatik



- Prof. Dr. Wolfgang Karl
 - Professur für Entwurf von Systemen in Hardware/Organisation Innovativer Rechnerarchitekturen,
 - Institut für Technische Informatik der Fakultät für Informatik, Universität Karlsruhe (TH)
 - Büro: Raum 159, Geb. 20.20
 - Tel.: 0721 608 3771
 - Email: karl@ira.uka.de
 - Sprechstunde:
 - Dienstag, 16:00 – 17:00 Uhr und nach Vereinbarung

- Übungen:
 - Vertiefung des in der Vorlesung behandelten Stoffs an Beispielen und Aufgaben
 - Übungsleiter: Dr. Rainer Buchty
 - buchty@ira.uka.de
 - Raum: 160, Geb. 20.20
 - Übungen: Dr. Jie Tao
 - tao@ira.uka.de
 - Raum: 161, Geb. 20.20

- Klausurtermin:
 - 7. August 2006, 14:00 Uhr
 - Stoff: Vorlesung und Übung



– Informationen zur Vorlesung unter:

- <http://www.irf.uka.de/karl/rechnerstrukturen.html>

Terminplanung: Vorlesung Rechnerstrukturen SS 2006					
KW	Vorlesung			Übung	
	Mi	Do		Mi	Do
KW 17	26.04.2006	27.04.2006			
KW18	03.05.2006				04.05.2006
KW19	10.05.2006	11.05.2006			
KW20	17.05.2006	18.05.2006			
KW21	24.05.2006				
KW22				31.05.2006	01.06.2006
KW23	07.06.2006	08.06.2006			
KW24	14.06.2006				
KW25		22.06.2006		21.06.2006	
KW26	28.06.2006	29.06.2006			
KW27		06.07.2006		05.07.2006	
KW28	12.07.2006				13.07.2006
KW29	19.07.2006				20.07.2006
KW30	26.07.2006	27.07.2006			
Ort: Tulla-Hörsaal					

- Der Begriff „**Rechnerstrukturen**“:
 - Hardwarestruktur eines Rechners auf einer konzeptionellen Darstellungsebene.
 - Anwendersicht und
 - Operationsprinzip eines Rechners.
- In der Vorlesung ist „Rechnerstrukturen“ im Sinne von Rechnerkonzepten oder schlicht — **Rechnerarchitektur** — gemeint.

- Rechnerarchitektur

- Allgemeine Strukturlehre mit deren Hilfsmittel
- Ingenieurwissenschaftliche Disziplin, die bestehende und zukünftige Rechenanlagen beschreibt, vergleicht, beurteilt, verbessert und entwirft.
- Betrachtet den Aufbau und die Eigenschaften des Ganzen (Rechenanlage), seiner Teile (Komponenten) und seiner Verbindungen (Globalstruktur, Infrastruktur)

1. Grundlagen

- Einführung
- Allgemeine Grundlagen des Entwurfs von Rechenanlagen
- Formen des Parallelismus und Klassifizierungen von Rechnerarchitekturen
- Bewertung von Rechensystemen

2. Prozessortechniken

- Von-Neumann-Architektur
- Von RISC zu Superskalar
- VLIW, EPIC
- Multithreading

3. Multiprozessoren

- Allgemeine Grundlagen, Verbindungsnetze, Leistungsfähigkeit
- Speichergekoppelte Multiprozessoren
 - SMP und DSM
 - Speicherkonsistenz und Cache-Kohärenz
- Nachrichtengekoppelte Multiprozessoren
- Höchstleistungsrechner und Grid-Computing
- Chip-Multiprozessoren

4. Weitere Rechnerstrukturen

- Vektorrechner und Feldrechnerprinzip
- SIMD-Verarbeitung in Mikroprozessoren
- Datenflussprinzip

5. Entwurf eingebetteter Systeme

- VHDL-Entwurf
- Rekonfigurierbarkeit

6. Zuverlässigkeit, Verfügbarkeit und Fehlertoleranz

7. Ausblick: Aktuelle Forschungsthemen



- Hennessy, J.L., Patterson, D.A.: Computer Architecture: A Quantitative Approach. Morgan Kaufmann, 3.Auflage 2002.
- Patterson, D.A. Hennessy, J.L.: Rechnerorganisation und –entwurf Die Hardware/Software-Schnittstelle. Deutsche Ausgabe herausgegeben von A. Bode, W. Karl, T. Ungerer. Spektrum Akademischer Verlag, Heidelberg, 2005
- U. Bringschulte, T. Ungerer: Microcontroller und Mikroprozessoren, Springer, Heidelberg, 2002
- Theo Ungerer: Parallelrechner und parallele Programmierung, Spektrum-Verlag 1997
- D.Culler, J.P.Singh, A.Gupta: Modern Parallel Computer Architecture, Morgan Kaufmann 1997

- Weitere vom Lehrstuhl angebotene Lehrveranstaltungen:
 - Vorlesung:
 - Mikroprozessoren
 - Dienstag, 9:45-11:15 Uhr, Raum 062, geb. 20.20
 - Inhalt:
 - » Der Schwerpunkt der Vorlesung liegt auf Mikroprozessoren, die in komplexen mehrbusorientierten Mikroprozessorsystemen (PCs, Arbeitsplatzrechnern und symmetrischen Multiprozessorsystemen mit gemeinsamen Speicher) eingesetzt werden. Die Vorlesung soll den Studierenden detaillierte Kenntnisse über die Architektur, Organisation und Programmierung dieser modernen Mikroprozessoren vermitteln. Dabei werden die grundlegenden Eigenschaften dieser Allzweck-Mikroprozessoren an konkreten Fallbeispielen dargestellt.
 - » Moderne Mikroprozessoren nutzen den in Programmen inhärent vorhandenen feinkörnigen Parallelismus aus, weshalb im dritten Teil der Vorlesung die grundlegenden Eigenschaften von parallelen Prozessorarchitekturen (superskalare, superpipelined und VLIW-Architekturen) erläutert werden. Es werden die in der Literatur diskutierten und bereits in Ankündigungen von Mikroprozessoren genannte Techniken besprochen.
 - » In einem Ausblick werden die neuesten Ankündigungen von Mikroprozessoren und Entwicklungstrends diskutiert. Hierzu zählen Multithreading und Multicore-Techniken.

- Weitere vom Lehrstuhl angebotene Lehrveranstaltungen:
 - Praktikum:
 - Prozessorentwurfspraktikum
 - Beim Entwurf von Prozessorarchitekturen steht die Eignung der Architektur für das gewünschte Einsatzgebiet im Vordergrund. Dies ist besonders wichtig für anwendungsspezifische Architekturen, wobei in diesem Bereich typischerweise als zusätzliche Anforderungen eine schnelle Time-to-Market sowie Kosteneffizienz hinzukommen.
 - Diese Eignung wird mithilfe von aufwendigen Simulationsumgebungen untersucht. Bekannte Beispiele hierfür sind pico, SimpleScalar oder Virtutech Simics.
 - Am Lehrstuhl für Rechnerarchitektur und Parallel Programmierung entsteht derzeit eine flexible und modulare Umgebung zur Simulation und Evaluierung von parallelen Systemen. Im Unterschied zu existierenden Simulationsansätzen soll sich diese Umgebung insbesondere durch einfache Erweiterbarkeit sowie universellen Einsatz sowohl für Einzelknoten- als auch parallele Systeme auszeichnen.
 - Im Rahmen dieses Praktikums sollen dieser Umgebung weitere Funktionsmodule hinzugefügt werden, wie sie zur Simulation vollständiger Systeme notwendig sind.
 - Das Praktikum richtet sich daher an engagierte Studenten mit umfangreichen Kenntnissen im Bereich Rechner- und Prozessororganisation sowie in der Programmierung unter C und C++.
 - Vorbesprechung: Donnerstag, 27.04.2006, 14:00 Uhr, Raum 267, Geb. 20.20

- Weitere vom Lehrstuhl angebotene Lehrveranstaltungen:
 - Seminare:
 - Software-Aspekte für dynamisch rekonfigurierbare Architekturen
 - Traditionelle Architekturen basieren auf festgelegten Einzelkomponenten, insbesondere die CPU-Architektur ist unveränderlich. Für die Programmentwicklung bedeutet dies, dass der Compiler einmal an diese Architektur angepasst werden muß, entsprechend muss die Anwendung einmal für die Zielplattform übersetzt werden.
 - Heutzutage werden bereits zunehmend so genannte ASIPs eingesetzt. Hierbei wird basierend auf Code-Analyse und korrespondierender CPU-Beschreibung eine an die jeweilige Anwendung angepasste CPU-Architektur samt hierfür tauglichem Compiler erzeugt.
 - Zukünftig sollen vermehrt rekonfigurierbare Systeme eingesetzt werden, welche zur Laufzeit in ihren Eigenschaften verändert werden können. Beispielsweise kann eine CPU in Art und Anzahl der Funktionseinheiten sowie auch ihrem Befehlssatz zur Laufzeit modifiziert werden. Hier ist der klassische Weg der Einmalerzeugung von angepasstem Compiler und Übersetzung der Anwendung nicht länger gangbar.
 - Im Rahmen dieses Seminars werden ausgehend von herkömmlichen Programmentwicklungsmethoden für statische Systeme aktuelle Verfahrensweisen diskutiert, die zur Anwendungsentwicklung für und auf rekonfigurierbaren Systemen dienen können. Hierbei stehen die Fragestellungen im Vordergrund, wie die Rekonfiguration die Programmerstellung und -übersetzung beeinflusst, und wie für solche Systeme universell gültige Compilatoren erzeugt werden können.
 - Vorbesprechung: Donnerstag, 27.04.2006, 14:30 Uhr, Raum 267, Geb. 20.20

- Weitere vom Lehrstuhl angebotene Lehrveranstaltungen:
 - Seminare:
 - Leistungsoptimierung von Rechensystemen
 - Zur vollständigen Ausnutzung der Rechenkapazität bedarf es oft Leistungsoptimierungen, die sich auf verschiedene Systemkomponenten beziehen. Auf Hardwareebene sind dies beispielsweise die Prozessorarchitektur mit Speichersubsystem (insbesondere Cache) bzw. auf Softwareebene Compiler und die jeweilige Implementierung der Algorithmen. Ebenfalls eine große Rolle spielen Verbindungsstrukturen und dort eingesetzte Kommunikationsprotokolle.
 - Bestmögliche Ausnutzung der Ressourcen bedingt in aller Regel eine Leistungssteigerung, auch eine Steigerung der Energieeffizienz lässt sich durch optimierte Ressourcenverwendung erzielen. Diese Optimierung wird auf unterschiedlichen Ebenen durch unterschiedliche Techniken erzielt, Beispiele hierfür sind Hyperthreading, rekonfigurierbare Hardware, adaptive Compiler und intelligente Aufgabenverwaltung.
 - Vorbesprechung: Donnerstag, 27.04.2006, 15:00 Uhr, Raum 267, Geb. 20.20

Kapitel 1: Grundlagen

1.1 Einführung, Begriffsklärungen



- Der Begriff „Rechnerarchitektur“
 - Definition nach Amdahl, Blaauw, Brooks (1967)

„Computer architecture is defined as the attributes and behavior of a computer as seen by a machine-language programmer. This definition includes the instruction set, instruction formats, operation codes, addressing modes, and all registers and memory locations that may be directly manipulated by a machine language programmer. Implementation is defined as the actual hardware structure, logic design, and data path organization of a particular embodiment of the architecture.“
 - Beschreibung der Attribute und des funktionalen Verhaltens eines Systems, wie es von einem Anwender, der in Maschinensprache programmiert, gesehen wird.

- **Der Begriff „Rechnerarchitektur“**
 - Definition nach Amdahl, Blaauw, Brooks (1967)
 - Rechnerarchitektur
 - spezifiziert die konzeptionelle Struktur und das funktionale Verhalten
 - und betrifft nicht Details der Hardware und der technischen Ausführung des Rechners
 - Die Definition behandelt nur das äußere Erscheinungsbild des Rechners und klammert die internen Vorgänge ausdrücklich aus.

- **Der Begriff „Rechnerarchitektur“**
 - Definition nach Amdahl, Blaauw, Brooks (1967)
 - Rechnerarchitektur
 - spezifiziert die konzeptionelle Struktur und das funktionale Verhalten
 - und betrifft nicht Details der Hardware und der technischen Ausführung des Rechners
 - Die Definition behandelt nur das äußere Erscheinungsbild des Rechners und klammert die internen Vorgänge ausdrücklich aus.

- Der Begriff „Rechnerarchitektur“
 - Heutige Sichtweise (Hennessy/Patterson, 2003):
 - Rechnerarchitektur umfasst
 - Befehlssatzarchitektur (Instruction Set Architecture)
 - Organisation
 - Hardware

} Implementierung

- Der Begriff „Rechnerarchitektur“
 - Heutige Sichtweise (Hennessy/Patterson, 2003):
 - Befehlssatzarchitektur (Instruction Set Architecture)
 - Beschreibung der Attribute und des funktionellen Verhaltens eines Rechners
 - Sichtweise des Maschinenprogrammierers
 - Schnittstelle zwischen Hardware und Software
 - Spezifikation der Befehlssatzarchitektur
 - » Ausführungsmodell
 - » Datenformate, Datentypen
 - » Adressierungsarten
 - » Befehlsformat und Befehlssatz
 - » Logischer Adressraum
 - » Unterbrechungssystem
 - » ...

- Der Begriff „Rechnerarchitektur“
 - Heutige Sichtweise (Hennessy/Patterson, 2003):
 - Organisation
 - Höhere Aspekte des Rechnerentwurfs:
 - Entwurf der internen CPU
 - » Art und Anzahl der internen Ausführungseinheiten
 - » Art und Stufenzahl der Befehlspipeline
 - » Grad und Verwendung der Superskalartechnik, VLIW, EPIC, Multithreading
 - Speicher- und Cachesystem
 - Busstruktur
 - ...
 - Hardware
 - Betrifft die speziellen Hardware-Eigenschaften des Rechners, einschließlich dem Logik-Entwurf bis hin zur Verpackungstechnik

Kapitel 1: Grundlagen

1.1 Einführung, Entwurfsfragen



- Entwurf einer Rechenanlage
 - Ingenieurmäßige Aufgabe der Kompromissfindung zwischen
 - Zielsetzungen
 - Einsatzgebiet, Anwendungsbereich, Leistung, Verfügbarkeit ...
 - Randbedingungen
 - Technologie, Größe, Geld, Energieverbrauch, Umwelt,...
 - Gestaltungsgrundsätzen
 - Modularität, Sparsamkeit, Fehlertoleranz ...
 - Anforderungen
 - Kompatibilität, Betriebssystemanforderungen, Standards

- Zielsetzungen

- Anwendungsbereiche:

- Technisch-wissenschaftlichen Bereich

- Strömungsmechanik
 - Materialforschung
 - ...

- Kommerzieller Bereich

- Datenbankanwendungen
 - WEB, Suchmaschinen
 - Optimierung von Geschäftsprozessen,
 - Unterstützung von Geschäftsentscheidungen (Risikoanalyse)
 - ...

- Eingebettete Systeme

- Verarbeitung digitaler Medien
 - Automatisierungstechnik
 - Automobil
 - Telekommunikation
 - ...

- Zielsetzungen

- Einsatzgebiete

- Desktop Computing

- PCs bis Workstations (\$1000 - \$10000)
 - Günstiges Preis-/ Leistungsverhältnis
 - Ausgewogene Rechenleistung für ein breites Spektrum von Anwendungen, einschließlich interaktiver Anwendungen (Graphik, Video, Audio) oder WEB-Anwendungen

- Server

- Höchstleistungsrechner
 - » Hohe Leistungsfähigkeit bzgl. Gleitkommaverarbeitung, Graphik
 - Server im kommerziellen Bereich
 - » Datenbankanwendungen, Transaktionsverarbeitung
 - » WEB-Server, Unterstützung von WEB-Diensten
 - Rechen- und datenintensive Anwendungen
 - Hohe Anforderungen an die Verfügbarkeit und Zuverlässigkeit
 - Skalierbarkeit
 - Große Datei-Systeme, Ein-/Ausgabesysteme

- Zielsetzungen
 - Einsatzgebiete

- Eingebettete Systeme (Embedded Systems)

- Mikroprozessorsysteme, eingebettet in Geräten, daher nicht unbedingt sichtbar
 - Beispiele: Automobil, Unterhaltungselektronik, Telekommunikation, Haushaltsgeräte, ...
 - » Rechensysteme sind auf spezielle Aufgabe zugeschnitten
 - » Hohe Leistungsfähigkeit für spezielle Anwendung
 - » Spezialprozessoren, Prozessorkerne mit anwendungsspezifischen Komponenten
 - Breites Preis-/Leistungsspektrum
 - » Von einfachen 8-, 16-Bit Microcontrollern bis komplexe Spezialprozessoren
 - Abwägen der Anforderungen an die Rechenleistung, Speicherbedarf, Kosten, Energieverbrauch
 - Echtzeitanforderungen