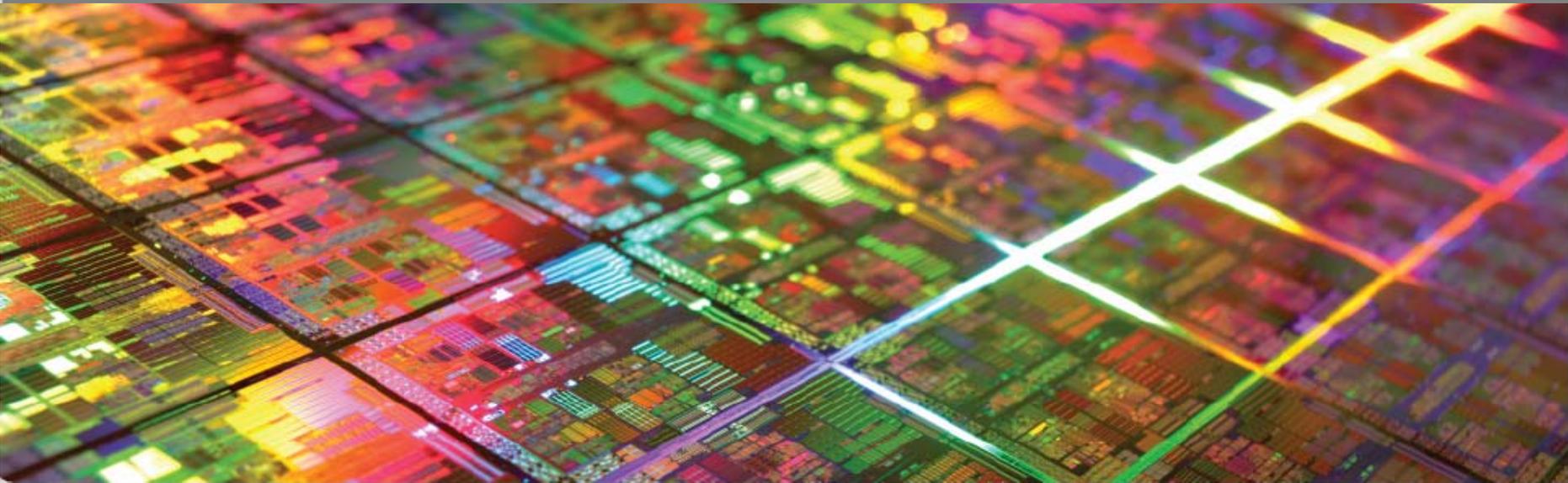


Rechnerstrukturen

Vorlesung im Sommersemester 2010

Prof. Dr. Wolfgang Karl

Fakultät für Informatik – Lehrstuhl für Rechnerarchitektur und Parallelverarbeitung



Persönliches

- Prof. Dr. Wolfgang Karl
 - Professur für Entwurf von Systemen in Hardware/Organisation Innovativer Rechnerarchitekturen,
 - Institut für Technische Informatik der Fakultät für Informatik, Karlsruher Institut für Technologie
 - Büro: Raum 314.1, Technologiefabrik, Haid-und-Neu-Str. 7
 - Tel.: 0721 608 3771
 - Email: karl@kit.edu
 - Sprechstunde:
 - Dienstag, 16:00 – 17:00 Uhr und nach Vereinbarung

Lehrstuhl

- Leitung:
 - Prof. Dr. Wolfgang Karl
- Sekretariat (Technologiefabrik, 2. Stock):
 - Frau Scheffel
 - Frau Murr-Grobe
- Mitarbeiter (Technologiefabrik, 2. Stock):
 - Dipl.-Inform Mario Kicherer
 - Dipl.-Inform. David Kramer
 - Dipl.-Inform. Oliver Mattes
 - Dipl.-Inform. Fabian Nowak
 - Dipl.-Inform. Martin Schindewolf

Organisatorisches

- Informationen zur Vorlesung unter:
 - <http://capp.itec.kit.edu/teaching/>

KW	Vorlesung			Übung	
	Di 09:45-11:15 HSaF	Do 14:00-15:30 Tulla		Di 09:45-11:15 HSaF	Do 14:00-15:30 Tulla
KW15	13.04.2010				
KW16	20.04.2010	22.04.2010			
KW17	27.04.2010				29.04.2010
KW18	04.05.2010	06.05.2010			
KW19	11.05.2010	Feiertag			Feiertag
KW20		20.05.2010		18.05.2010	
KW21	25.05.2010	27.05.2010			
KW22		Feiertag		01.06.2010	Feiertag
KW23	08.06.2010	10.06.2010			
KW24	15.06.2010	17.06.2010			
KW25				22.06.2010	24.06.2010
KW26	29.06.2010	01.07.2010			
KW27	06.07.2010	08.07.2010			
KW28				13.07.2010	15.07.2009

Organisatorisches

■ Übungen:

- Vertiefung des in der Vorlesung behandelten Stoffs an Beispielen und Aufgaben
- Übungsleiter: David Kramer
 - david.kramer@kit.edu
 - Raum: Technologiefabrik, 2. Stock, 315.1

■ Klausurtermin:

- 11. August 2010, 08:00 Uhr
 - Stoff: Vorlesung und Übung

Vorlesung Rechnerstrukturen

- 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: Disziplin

- 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)

Aufbau der Vorlesung

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, Superskalartechniken
- VLIW, EPIC
- Multithreading

Aufbau der Vorlesung

3. Multiprozessoren

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

4. Weitere Rechnerstrukturen

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

5. Zuverlässigkeit, Verfügbarkeit und Fehlertoleranz

6. Ausblick: Aktuelle Forschungsthemen

Hinweis

■ Vom Lehrstuhl angebotene Lehrveranstaltungen:

■ Vorlesungen:

- Rechnerorganisation
- Digitaltechnik und Entwurfsverfahren
- Rechnerstrukturen
- Mikroprozessoren I,
- Mikroprozessoren II
- Heterogene parallele Rechnerstrukturen

} Im Turnus mit
Prof. Dillmann, Prof. Hanebeck, Prof. Henkel

} Regelmäßig im Sommersemester

} Regelmäßig im Wintersemester

■ Praktika:

- Multicore-Programmierung
- Multicore-Technologie
- Basispraktikum Technische Informatik: Hardware-naher Systementwurf

■ Praxis der Softwareentwicklung

■ Seminare

Literatur

- 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, 2. Auflage 2007
- Theo Ungerer: Parallelrechner und parallele Programmierung, Spektrum-Verlag 1997
- D.Culler, J.P.Singh, A.Gupta: Modern Parallel Computer Architecture, Morgan Kaufmann 1997

Vorlesung Rechnerstrukturen

Kapitel 1: Grundlagen

1.1 Einführung, Begriffsklärungen

Begriffsklärung

■ 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.

■ 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.

Begriffsklärung

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

} Implementierung

Begriffsklärung

- 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
 - ...

Begriffsklärung

- 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

Vorlesung Rechnerstrukturen

- Kapitel 1: Grundlagen
 - 1.2 Entwurf von Rechenanlagen - Entwurfsfragen

Rechnerarchitektur (Disziplin)

- 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

Entwurfsfragen

- 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
 - Server im kommerziellen Bereich
 - Rechen- und datenintensive Anwendungen
 - Hohe Anforderungen an die Verfügbarkeit und Zuverlässigkeit
 - Skalierbarkeit
 - Große Datei-Systeme, Ein-/Ausgabesysteme

Entwurfsfragen

- 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
 - Echtzeitanforderungen
 - Abwägen der Anforderungen an die Rechenleistung, Speicherbedarf, Kosten, Energieverbrauch

Entwurfsfragen

- 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
 - ...
- Leistungsfähigkeit

Entwurfsfragen

- Zielsetzungen
- Zuverlässigkeit
 - Gewährleistung einer minimalen Verfügbarkeit des Systems
 - Bei Ausfällen von Komponenten muss ein betriebsfähiger Kern bereit sein
 - Vielfach Verwendung redundanter Komponenten
 - Wichtig für sicherheitskritische Anwendungen
 - Wichtig im kommerziellen Bereich

Entwurfsfragen

- Zielsetzungen
- Energieverbrauch, Leistungsaufnahme
 - Mobile Geräte
 - verfügbare Energiemenge durch Batterien und Akkumulatoren begrenzt
 - möglichst lange mit vorhandener Energie auskommen
 - möglichst wenig Energie soll in Wärme umgesetzt werden, um eine Überhitzung zu vermeiden
 - Green IT
 - Rechnerhersteller bieten „green HW“ an:
 - niedriger Energieverbrauch
 - ökologische Produktion
 - einfaches Recycling

Entwurfsfragen

- Gestaltungsgrundsätze (nach Ungerer, 2000)
- Konsistenz:
 - Eigenschaft eines Systems mit folgerichtiger, schlüssiger Aufbau
 - Bei Kenntnis eines Teil des Systems muss der Rest vorhersagbar sein
 - Beispiel:
 - Bei Erweiterung des Befehlssatzes ist die Realisierung eines neuen Befehls weitgehend durch die bereits vorliegenden Befehle festgelegt
- Orthogonalität / Modularität
 - Funktional unabhängige Teilelemente sind unabhängig voneinander spezifiziert und realisiert.
 - Beispiel:
 - Befehlssatz ist orthogonal, wenn sich jeder arithmetisch-logische Befehl mit jeder Adressierungsart kombinieren lässt

Entwurfsfragen

- Gestaltungsgrundsätze (nach Ungerer, 2000)
- Symmetrie
 - Eigenschaft eines Systems, mathematisch symmetrische Eigenschaften des Systems auch symmetrisch zu entwerfen
 - Beispiel:
 - Die Verwendung der Subtraktion sollte in gleichartiger Weise erfolgen können, wie die Addition, d.h. mit denselben Daten- und Befehlsformaten
- Angemessenheit
 - Die Elemente eines Systems sind angemessen, wenn ihre Funktionen bei der Lösung der vorgesehenen Problemstellung ausgeschöpft werden
 - Beispiel:
 - Abstimmung von Prozessor und Speicher

Entwurfsfragen

- Gestaltungsgrundsätze (nach Ungerer, 2000)
- Transparenz
 - Ein System ist transparent, wenn verschiedene Funktionen des Gesamtsystems unsichtbar bleiben
 - Beispiel:
 - Prozesse im Hintergrund
- Virtualität
 - Eigenschaft eines Systems, Funktionen anzubieten, die real gar nicht vorhanden sind
 - Beispiel:
 - Virtuelle Speicherverwaltung

Entwurfsfragen

- Gestaltungsgrundsätze (nach Ungerer, 2000)
- Benutzerfreundlichkeit
 - Beziehung zwischen einem Rechensystem und Nutzer:
 - Anwendungsprogrammierer:
 - problemorientierte Erstellung von Anwendungen
 - Administration
 - Verwaltung, Wartung verteilter Systemressourcen
 - Bedienung durch nicht-technisches Personal, Anwender
 - Gestaltung der Schnittstelle zwischen dem Rechensystem und seinem Benutzer
 - Ziel der Software-Entwicklung
 - Techniken der benutzerfreundlichen Oberfläche erst durch spezielle Hardware-Techniken möglich

Entwurfsfragen

- Gestaltungsgrundsätze (nach Ungerer, 2000)
- Erweiterbarkeit / Skalierbarkeit
 - Charakterisiert ein System, das in jedem Zustand die Erweiterung seiner Fähigkeiten zulässt, aber in jedem Zwischenzustand korrekt funktionsfähig ist
 - Installation weiterer Ausbaustufen durch Hinzufügen von Komponenten