

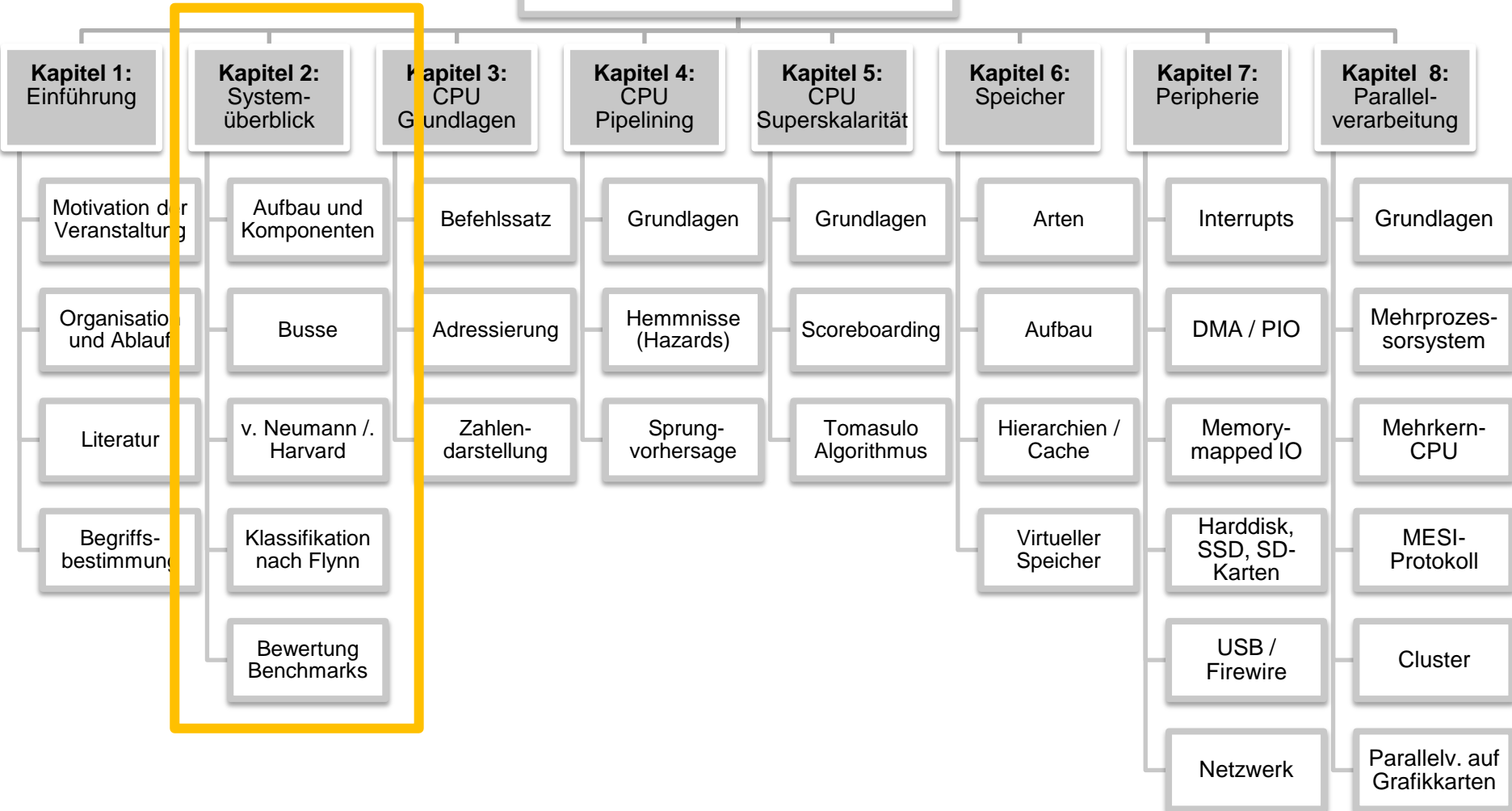
RECHNERARCHITEKTUR

Kapitel 2 – Systemüberblick

Prof. Dr. L. Wischhof <wischhof@hm.edu>

Fakultät 07 – Hochschule München

Rechnerarchitektur



Systemüberblick

Motivation

Typische Fragestellungen:

- Aus welchen Bestandteilen besteht ein Rechner und wie sind diese verknüpft?
- Welche grundsätzlichen Ansätze gibt es hinsichtlich der Architektur?
- Wie kann ich ein Gesamtsystem mit einem anderen Gesamtsystem vergleichen?
 - Welches System ist schneller?
 - Welches ist besser für meinen Anwendungszweck geeignet?



Systemüberblick

Lernziele

- Grundlegenden Aufbau und Komponenten eines Rechnersystems sowie deren Anbindung (Bus/Interface) benennen können
- Vor- und Nachteile der Architektur nach v. Neumann und Harvard angeben können
- Häufig genutzte Benchmarks und deren Einsatzzweck angeben können
- Benchmark-Ergebnisse berechnen und bewerten können



Systemüberblick

Anwendungsbereiche für Rechner

- Unterschiedliche Klassen von Rechnern
- Designherausforderungen (➔ Anforderungen an die Rechnerarchitektur) unterscheiden sich stark

Embedded	Mobilgerät	Desktop	Server	Großrechner
<ul style="list-style-type: none">• Preis• Geringe Stromaufnahme• Anwendungsbezogene Performance	<ul style="list-style-type: none">• Preis• Geringe Stromaufnahme• Schnelle Reaktion auf Benutzereingaben• Medienkodierung und -dekodierung	<ul style="list-style-type: none">• Preisbezogene Performance „Leistung pro Euro“• Grafikgeschwindigkeit	<ul style="list-style-type: none">• Durchsatz• Verfügbarkeit• Skalierbarkeit• Energieeffizienz	<ul style="list-style-type: none">• Preisbezogene Performance• Durchsatz• Energieeffizienz

Quelle: Eigene Darstellung nach [2], S.5



Systemüberblick

Historische Entwicklung am Beispiel des Desktops



	1981 IBM-PC	1984 PC/AT	1988 AT-386/486	1995 Wintel PC
Prozessor:	8088, 8086	80286	80386, 486	Pentium
Takt:	4,77 MHz	6-8 MHz	12-33 MHz	60-300 MHz
Speicher:	16-256 kB	1 MB	4 MB	16 MB
Bus:	PC-Bus	ISA-Bus	ISA, VESA	PCI, ISA
Festplatte:	10 MB, MFM	40 MB, ESDI	100 MB, IDE	1 GB
OS:	DOS 1.0	DOS 3.0	DOS / Win3.x	Windows 95

Bildquellen: Wikipedia (Boffy_b), maximumpc.com, jeweils beispielhafte Ausstattungen/ca.-Angaben

Systemüberblick

Historische Entwicklung am Beispiel des Desktops

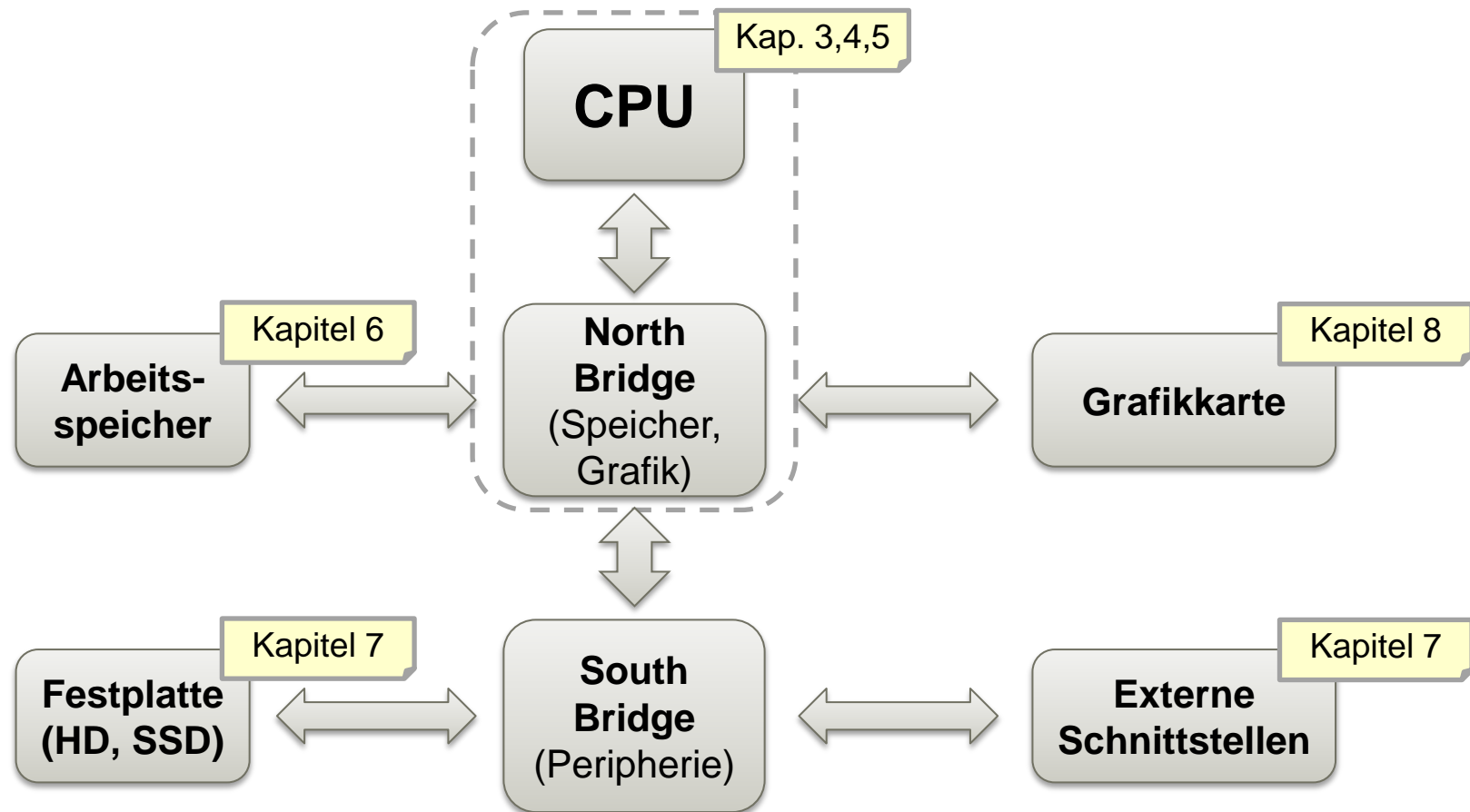


	1999 Media-PC	2002 Inet-PC	2007 Multicore
Prozessor:	Pentium II	Athlon XP	Core 2 Duo
Takt:	233-450 MHz	1-2 GHz	1-3,33 GHz
Speicher:	64 MB	512 MB	2 GB
Bus:	PCI	PCI, AGP	PCIe
Festplatte:	2 GB, ATA	80 GB, SATA	1 TB, SATA
OS:	Windows 98	Windows XP	Win Vista

jeweils beispielhafte Ausstattungen/ca.-Angaben

Systemaufbau

Vereinfachter Aufbau eines Desktops (Schema)



Systemaufbau

Typischer Aufbau eines PC mit North/South Bridge

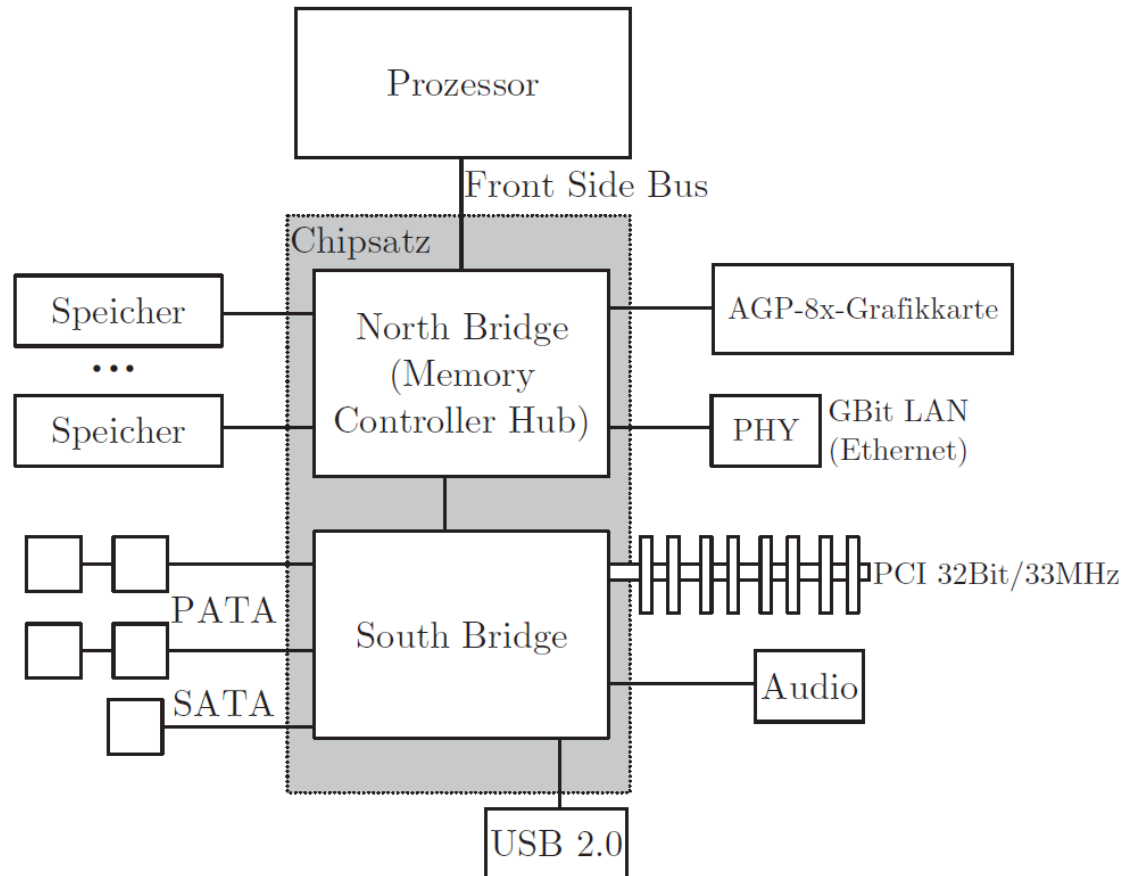
North-Bridge

Anbindung von

- Speicher
- Grafik
- (Netzwerk)

South-Bridge

- Anbindung restlicher Peripherie



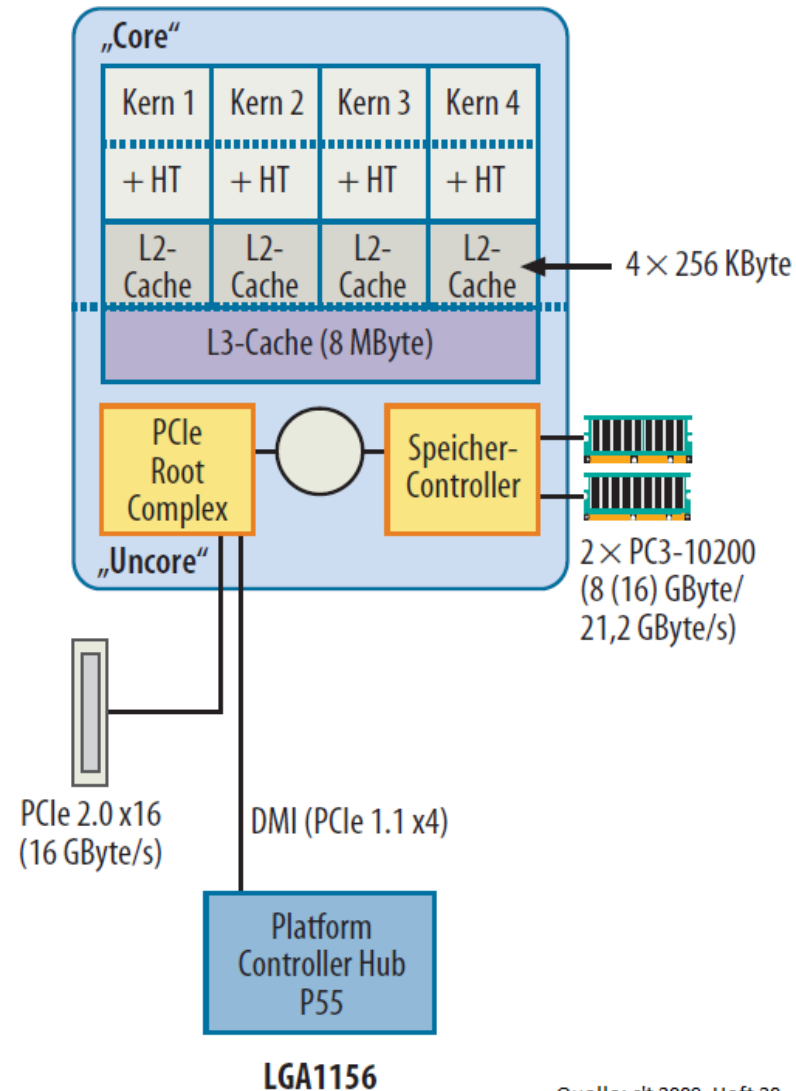
Systemaufbau

Aufbau eines aktuellen PCs am Beispiel Intel i5, i7

Neuere Prozessoren integrieren die North-Bridge Funktionalität in den Prozessor

Beispiel rechts: Intel LGA1156 (Core i5, i7)

Warum ist dies sinnvoll?

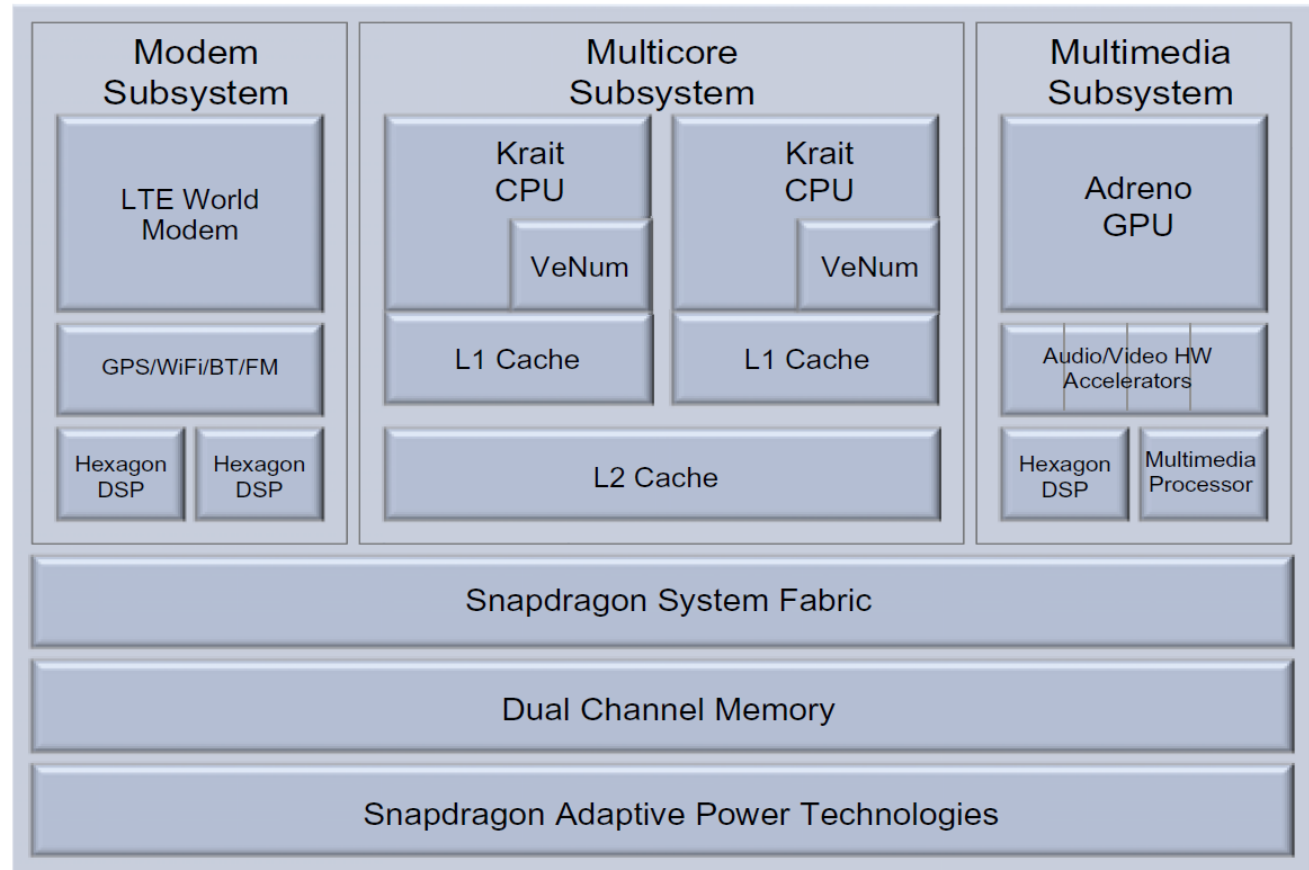


Systemaufbau

Smartphone SoC am Beispiel Snapdragon S4

System on a Chip (SoC) für Smartphone und Tablet mit hoher Integrationsdichte.

eingesetzt z.B. in Samsung Galaxy SIII (LTE-Version)



Quelle: Snapdragon S4 Processors: System on Chip Solutions for a New Mobile Age, White Paper, Qualcomm, Oct. 2011.



Verbindungsstrukturen

Komponenten und Strukturen

Rechner besteht aus drei grundlegenden Komponententypen

- Prozessor
- Speicher
- I/O (Input/Output, Eingabe/Ausgabe)

Wie werden diese verbunden?

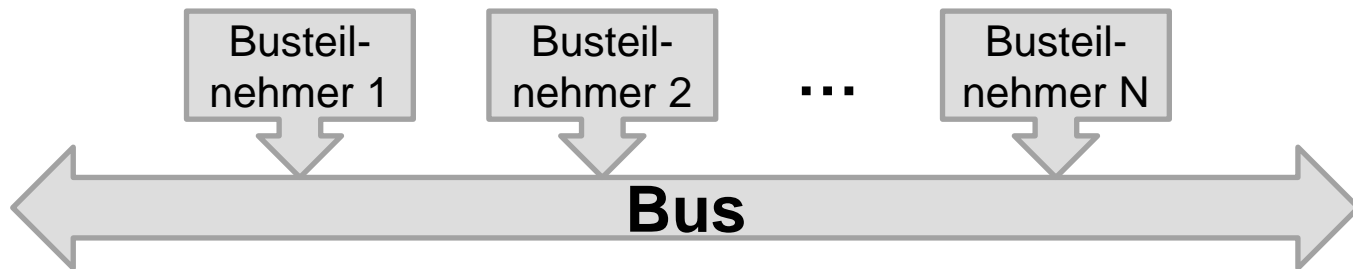
Die zwei am häufigsten verwendeten Strukturen:

1. Bus
2. Punkt-zu-Punkt Verbindungen
(mit paketiertem Datenverkehr)



Verbindungsstrukturen

Bus



- **Geteiltes Medium**

- nur ein Gerät darf zu einem Zeitpunkt senden (→ **Arbitrierung**: Regelung der Busnutzung)
- von einem Gerät gesendete Daten von allen anderen Geräten am Bus empfangbar

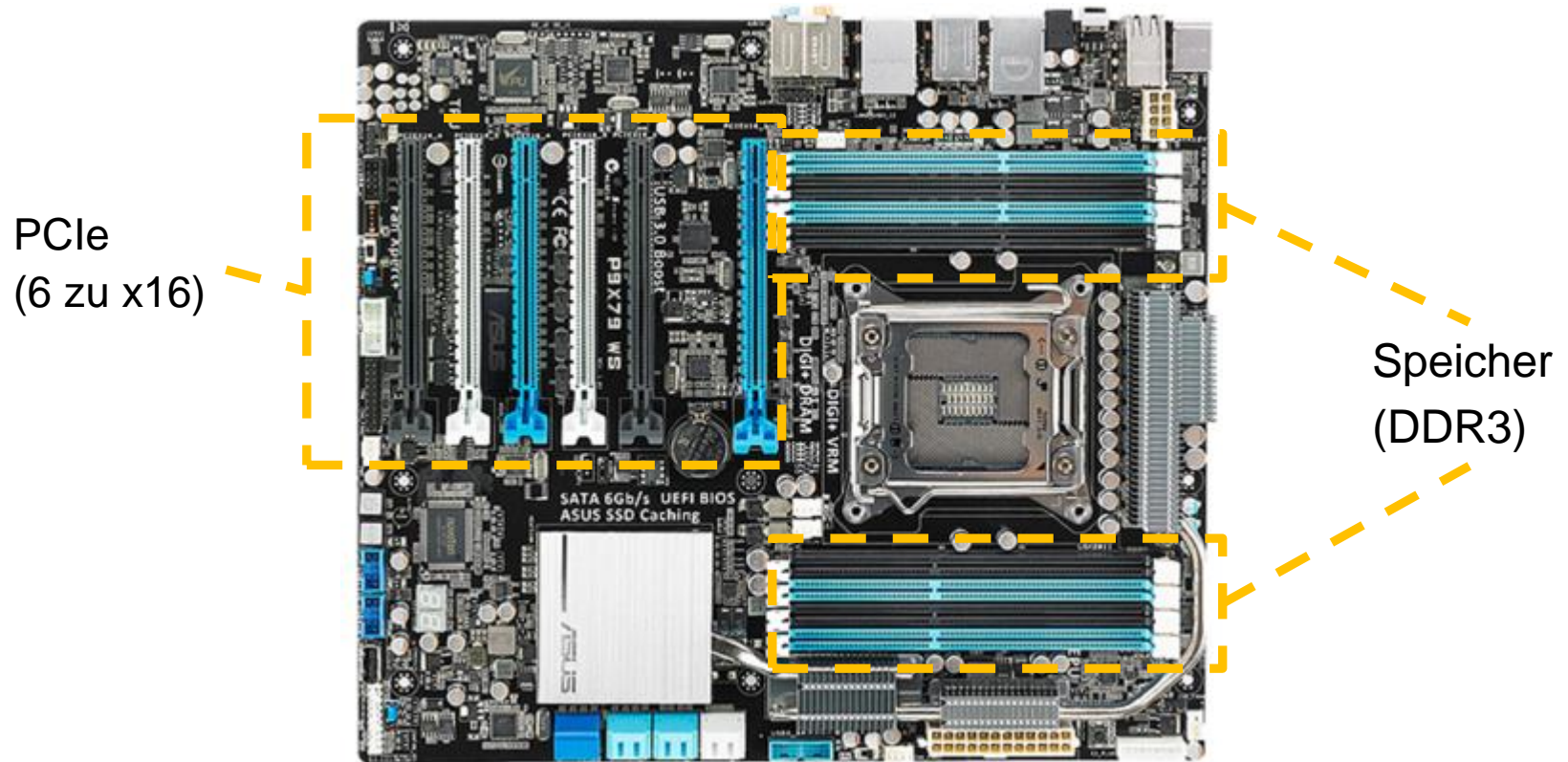
- **Leiterbahnen (bus lines)**

- Jede überträgt ein Bit → mehrere Leiterbahnen erlauben parallele Übertragung mehrerer Bits
- **Busbreite**: Anzahl Leiterbahnen → Anzahl parallel übertragener Bits



Verbindungsstrukturen

Beispiele auf einem typischen Mainboard

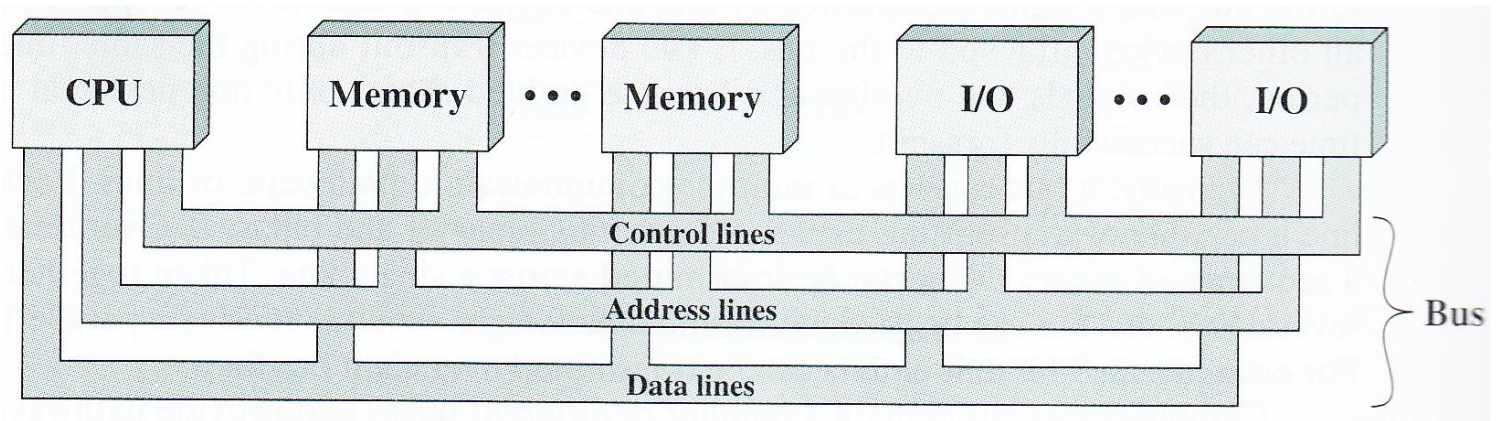


Beispiel: Asus P9X79, Bildquelle: Asus

Verbindungsstrukturen

Busstruktur

- **Datenbus**
 - Datenübertragung zwischen den Komponenten
- **Adressbus**
 - Übertragung der Quell- oder Zieladresse bzw. des I/O Ports
- **Steuerbus (Kontrollbus)**
 - Steuerung des Zugriffs und der Nutzung der Daten- und Adressleitungen, sowie der auszuführenden Operationen



Quelle: [3, S. 108]

Verbindungsstrukturen

Busstruktur: Steuerbus (Kontrollbus)

Typische Steuerleitungen des Steuerbusses beinhalten (siehe [3]):

- **Memory (bzw. I/O) Write**
 - Bewirkt, dass Daten auf dem Bus an adressierte Speicherstelle (bzw. Port) geschrieben werden
- **Memory (bzw. I/O) Read**
 - Bewirkt, dass Daten von der adressierten Stelle (bzw. Port) auf den Bus gelegt werden
- **Transfer ACK**
 - Bestätigung (Acknowledgement) – Daten wurden geschrieben bzw. gelesen
- **Bus Request**
 - Komponente möchte die Kontrolle über den Bus (Zugriff)
- **Bus Grant**
 - Komponente bekommt die Kontrolle über den Bus zugesichert
- **Clock**
 - Zeitsignal zur Synchronisation der Operationen
- **Reset**
 - Initialisierung aller Komponenten am Bus



Verbindungsstrukturen

Klassifizierung von Bussen

- **Typ**
 - Dediziert (z.B. getrennte Leitungen für Daten und Adressen)
 - Multiplexing/“gemultiplext“
- **Methode zur Zugriffssteuerung (Arbitrierung)**
 - Zentral
 - Verteilt
- **Timing**
 - Synchron
 - Asynchron
- **Breite**
 - Adresse
 - Daten

nach [3, S. 112]



Verbindungsstrukturen

Punkt-zu-Punkt Verbindungen: Beispiel QPI

- Breite Busse werden mit steigendem Bustakt problematischer
 - ➔ Arbitrierung und Synchronisation zeitkritisch
 - ➔ aktuell Trend zu Punkt-zu-Punkt Verbindungen
- Punkt-zu-Punkt Verbindungen erlauben höhere Datenraten, geringere Verzögerungen (Latenz) und sind besser skalierbar

Beispiel: Quick Path Interconnect (QPI, Intel 2008)

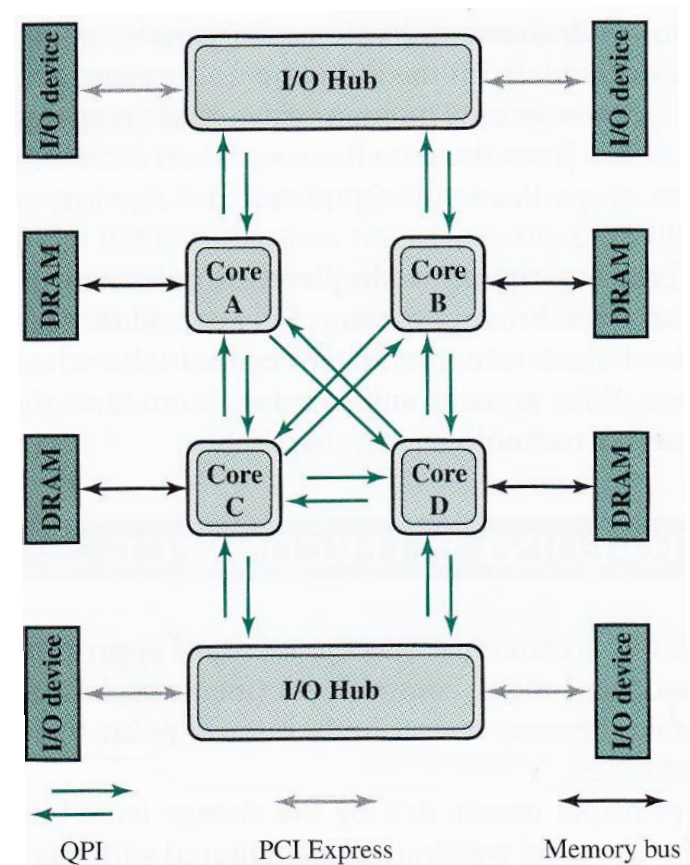
- **Mehrere direkte Verbindungen** zwischen Komponenten anstelle eines Busses
 - **Schichtenbasierte Protokollarchitektur**
 - **Paketorientierte Datenübertragung**
- } siehe auch:
LV Netzwerke I !



Verbindungsstrukturen

Punkt-zu-Punkt Verbindungen: Beispiel QPI

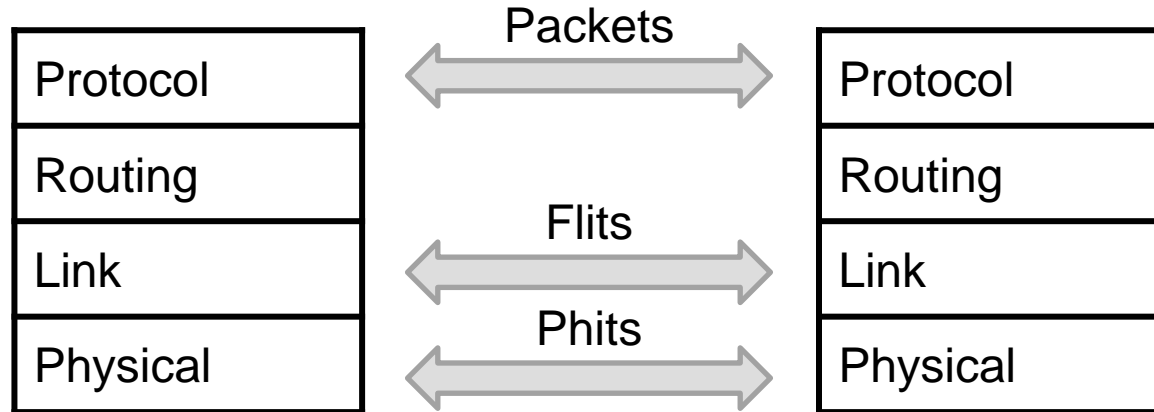
- Multicore Rechner ist Netzwerk unterschiedlicher Komponenten
- Komponenten leiten Daten ggf. an andere weiter
- I/O Hub bindet Peripherie an, konvertiert dazu beispielsweise zwischen QPI und PCIe



Quelle: [3, S. 116]

Verbindungsstrukturen

QPI Schichtenmodell



- **Physical:** physikalische Übertragung in Einheiten von 20 Bits (Phit, physical unit)
- **Link:** zuverlässige Übertragung und Datenflusskontrolle, Übertragung in Einheiten von 80 Bits (Flit, flow control unit)
- **Routing:** Wegewahl und Weiterleitung innerhalb des QPI „Netzwerks“
- **Protocol:** Paketorientierter Datenaustausch, Paket besteht aus ganzzahliger Anzahl an Flits



Verbindungsstrukturen

QPI Physical Layer

QPI Port

- Jeweils 20 Datenleitungen + 1 Clock-Leitung pro Richtung
→ 20 Bit pro Richtung können parallel übertragen werden
- Aktuelle Produkte übertragen 16 GB/s pro Richtung

Übertragung

- Differenzielle Übertragung → störunempfindlich
(Low Voltage Differential Signalling, LVDS)
- 80-bit flits werden nacheinander auf die einzelnen Leitungen verteilt
(multilane distribution)



Verbindungsstrukturen

QPI Link Layer

- Flit besteht aus 72-Bit Nachricht (Payload) und 8-Bit CRC
- **Datenflusskontrolle** (flow control)
 - Vermeidung der Überlastung des Empfängers
 - Während der Initialisierung teilt Empfänger dem Sender Anzahl an Credits mit (entspricht Größe des Sendefensters). Pro gesendetem Flit ein Credit benötigt, nach Verarbeitung durch Empfänger sendet er Credit zurück.
- **Zuverlässige Übertragung** (error control)
 - Sender berechnet CRC, wird vom Empfänger geprüft
 - Im Falle eines Fehlers wird fehlerhaftes Flit inklusive aller nachfolgenden Flits wiederholt.



Verbindungsstrukturen

QPI Routing Layer / QPI Protocol Layer

Routing Layer

- Firmware definiert Routing-Tabelle
- In komplexen System kann dynamisch Anpassung des Routings erfolgen. Einflussgrößen:
 - Auslastung
 - Aufteilung der Systemressourcen
 - Umgehen fehlerhafter Komponenten

Protocol Layer

- Nutzdaten, z.B. Cache-Kohärenzprotokoll welches Daten zu einem Cache überträgt



Verbindungsstrukturen

PCI / PCI Express (PCIe)

Peripheral Component Interconnect (PCI)

- Bus zur Vernetzung innerhalb des Rechners bzw. mit Peripherie
- Entwickelt von Intel (1990), Standards veröffentlicht und freigegeben
- Heute von Industriegremium PCI Special Interest Group (SIG) weiterentwickelt.

PCI Express (PCIe)

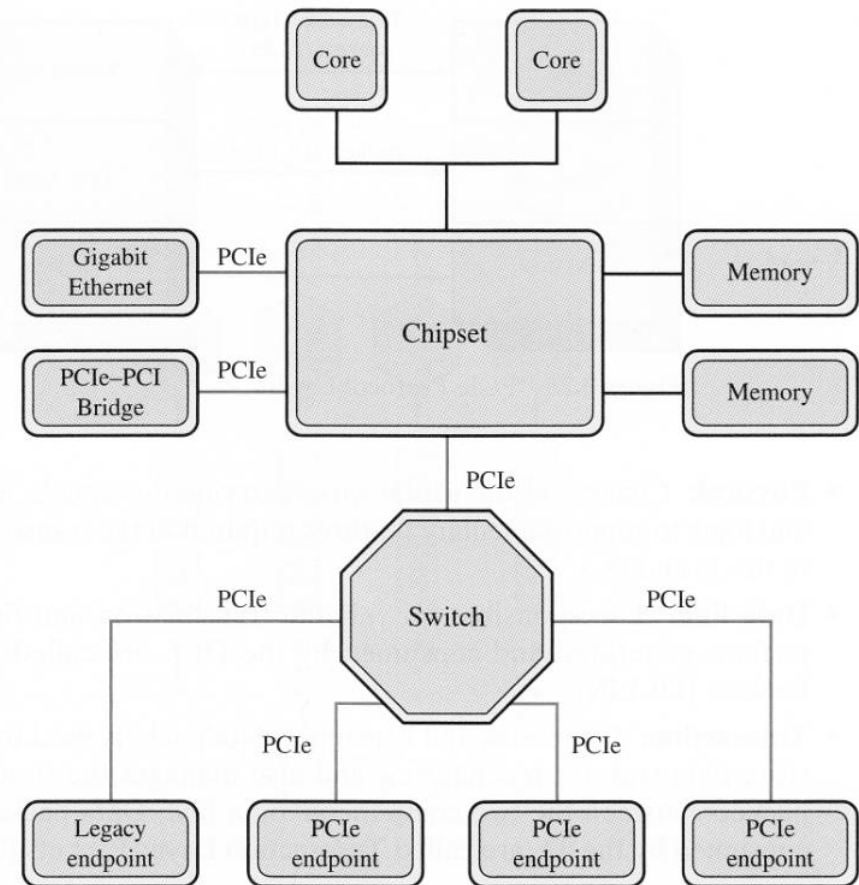
- Weiterentwicklung von PCI
- kein Bus sondern Punkt-zu-Punkt Verbindung
- Erlaubt höhere Datenraten sowie eine Priorisierung von Daten



Verbindungsstrukturen

PCI Express (PCIe): Typische Konfiguration

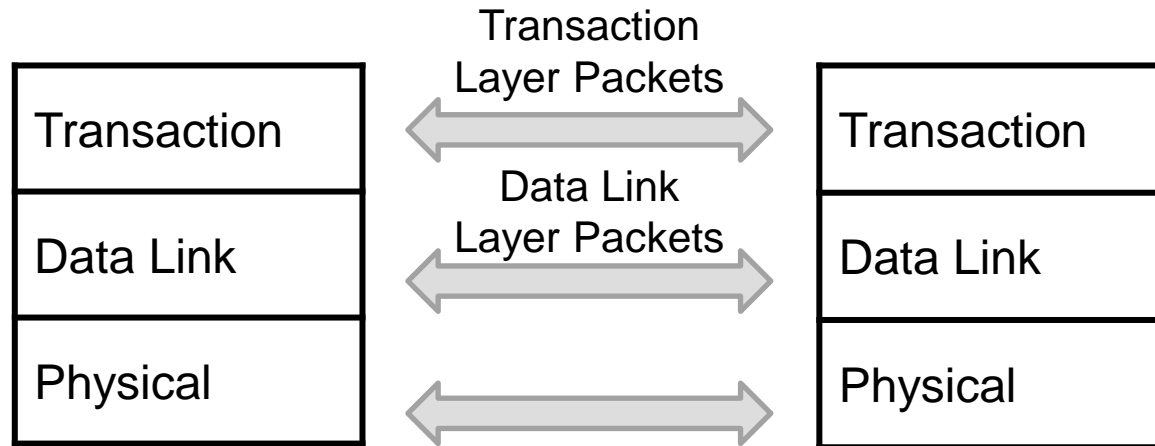
- **Switch:**
Weiterleitung von PCIe Datenströmen
- **PCIe Endpoint:**
Peripheriegerät (z.B.: Gigabit-Ethernet)
- **Legacy Endpoint:**
Endpunkt eines älteren Designs, der zu PCIe migriert wurde.
- **PCIe/PCI Bridge:**
Erlaubt Anschluss von PCI Geräten.



Quelle: [3, S. 121]

Verbindungsstrukturen

PCIe Schichtenmodell



- **Physical:** physikalische Übertragung
- **Link:** zuverlässige Übertragung und Datenflusskontrolle, Einheit: Data Link Layer Packets (DLLPs)
- **Transaction:** Paketorientierter Datenaustausch, Einheit: Transaction Layer Packets (TLPs)



Verbindungsstrukturen

PCIe: Physical Layer

- PCIe Port nutzt **bidirektionale** Leitungen („lanes“)
- Port kann 1,4,6,16 oder 32 lanes besitzen
- Verteilung der Bits auf die einzelnen lanes (vergl. QPI)
- Auf jeder lane werden Daten in Blöcken von 128 Bit übertragen
- Vor Übertragung erfolgt eine Codierung von 128 Bit auf ein 130 Bit Codewort
- Differentielle Übertragung
- Datenrate 16 GB/s (PCIe 3.0, 2010)



Verbindungsstrukturen

PCIe: Data Link Layer

- Data Link Schicht fügt
 - 16-Bit Sequenznummer
 - 32-Bit CRChinzu
- Prüfung an jedem Zwischenknoten
 - ➔ bei Fehler negatives Acknowledgement (NAK) an Sender
- Korrekte DLLPs werden an Transaction Layer weitergeleitet
- Ist dieser nicht der Zielknoten, so wird nächster Knoten bestimmt und Paket wieder an den Link Layer gegeben
- Spezielle Pakete für Link Management
 - Flow Control
 - Power Management
 - ACK/NAK



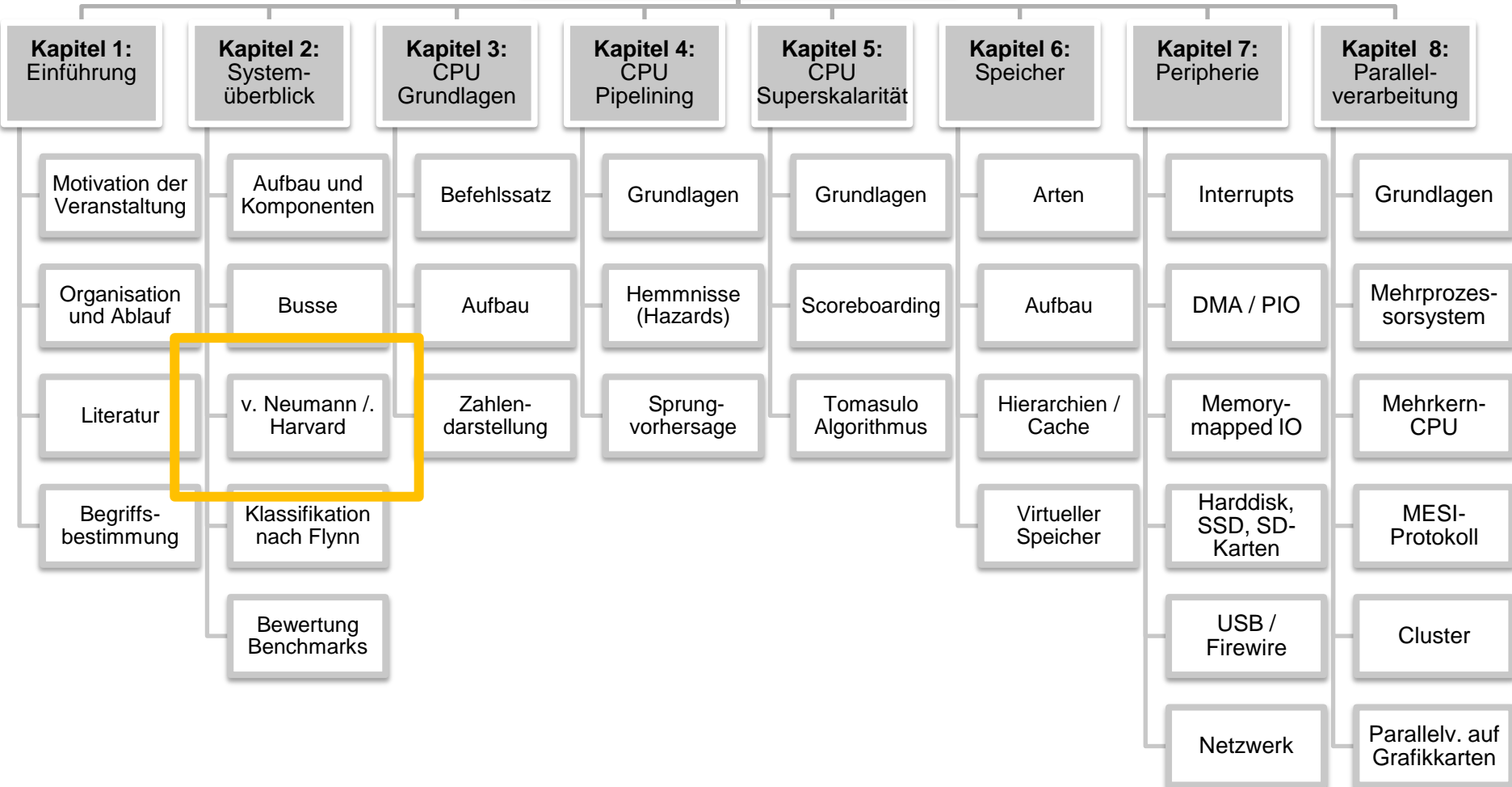
Verbindungsstrukturen

PCIe: Transaction Layer

- Übertragung von Nutzdaten in Form von TLPs
- Datenabfrage häufig zweigeteilt („split transaction“)
 1. Request Paket wird gesendet (mit eindeutiger ID)
 2. Completion Paket dann gesendet wenn Daten vorliegen→ Zeitliche Trennung (Zwischenzeit von anderen Daten nutzbar!)
- 32-Bit und 64-Bit Speicheradressierung unterstützt
- **Transaction Types**
 - Memory: Read Request, Read Lock Request, Write Request
 - I/O: Read Request, Write Request
 - Configuration: Read Request, Write Request (jeweils für Typ 0,1)
 - Message: Request, Request with Data
 - Completion: Completion, Completion with Data, Completion Locked, Completion Locked with Data



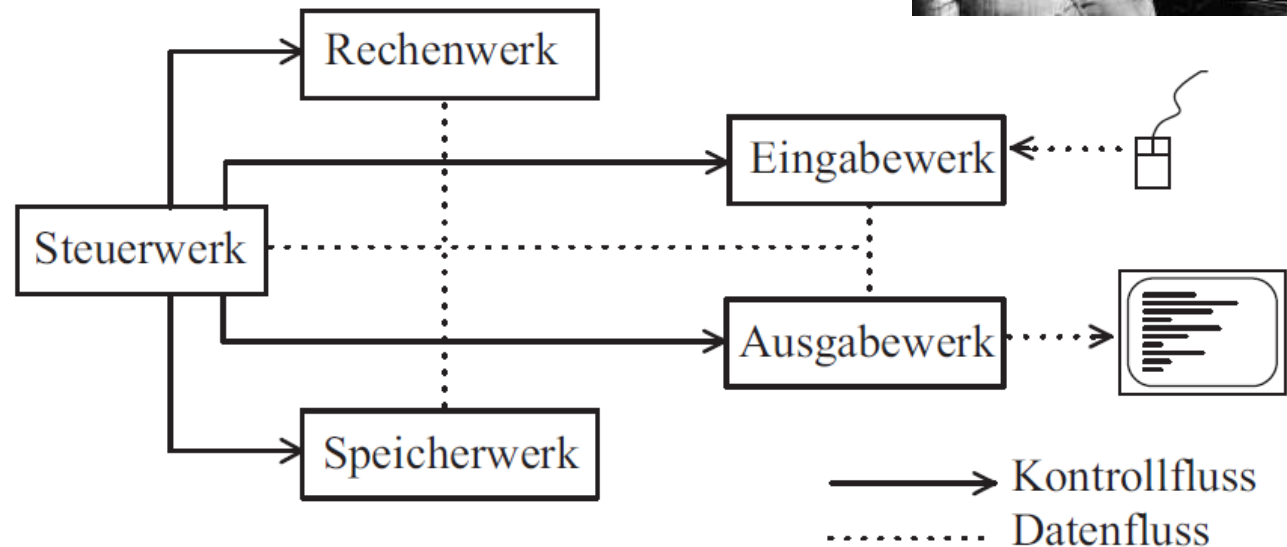
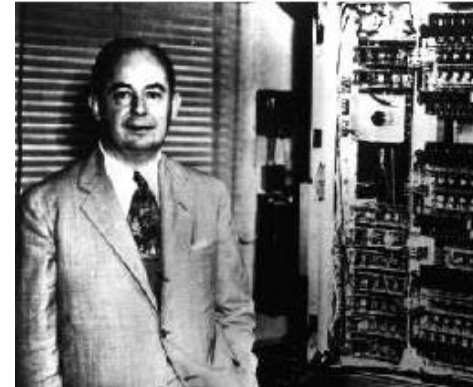
Rechnerarchitektur



Rechnerarchitekturen

Universalrechner nach John v. Neumann (1/3)

- **Rechenwerk**
 - **Steuerwerk**
 - **Speicherwerk**
 - **Eingabewerk**
 - **Ausgabewerk**
- } Prozessor



Rechnerarchitekturen

Universalrechner nach John v. Neumann (2/3)

- Aufbau **unabhängig** von zu bearbeitendem Problem
- Speicher ist in gleich große, fortlaufend nummerierte Zellen eingeteilt
- Programme, Daten, Zwischen- und Endergebnisse **im gleichen Speicher**
- Daten und Programme werden vom Eingabewerk gelesen und im Speicher abgelegt
- Aufeinanderfolgende Befehle befinden sich in aufeinanderfolgenden Speicherzellen
- **Befehlszähler** im Steuerwerk zeigt auf den nächsten auszuführenden Befehl



Rechnerarchitekturen

Universalrechner nach John v. Neumann (3/3)

Ablauf

- Aktueller Befehl wird aus Speicher gelesen und im Befehlsregister des Steuerwerks zwischengespeichert
- Befehl wird dekodiert
 - ➔ Steuersignale veranlassen Ausführung der Operation

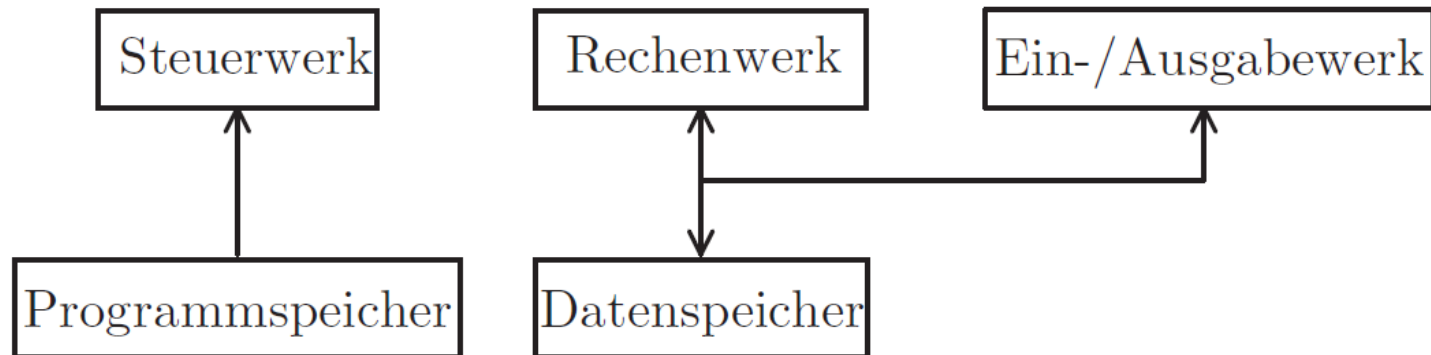
Befehlsarten

- Arithmetische und logische Befehle zur Verknüpfung von Daten
- Transportbefehle (Verschieben von Daten zwischen Komponenten)
- Sprungbefehle (bedingt und unbedingt), Unterprogrammaufrufe
- Ein-/Ausgabebefehle (Kommunikation mit Peripherie)
- Sonstige Befehle (Unterbrechen, Warten, Stop, etc.)



Rechnerarchitekturen

Harvard-Architektur



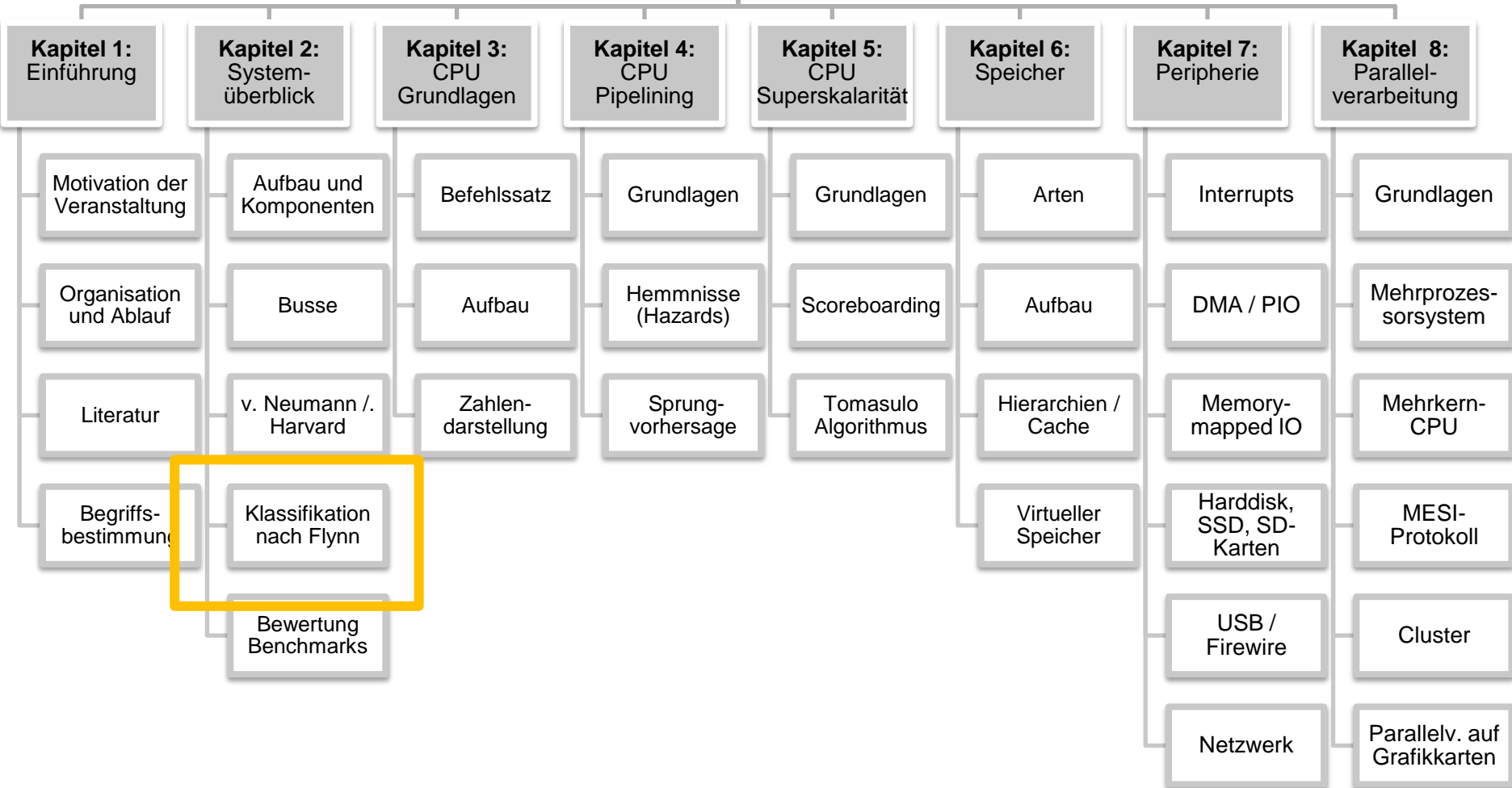
- **Getrennte Speicher** für Programme (Code) und Daten
(in der Praxis z.B. bei einigen Signalprozessoren verwendet)

Modifizierte Harvard-Architektur:

- Getrennte Level-1-Caches für Code und Daten
(→ bei praktisch allen modernen Prozessoren!)



Rechnerarchitektur



Rechnerarchitekturen

Klassifikation nach Flynn (Flynn'sche Taxonomie)

- Von Michael J. Flynn 1966 publiziert
- Einteilung von Rechnerarchitekturen hinsichtlich ihrer Instruktions- und Datenströme
- Heute nur noch eingeschränkt gültig aber oft verwendet

4 Klassen:

- Single Instruction Single Data (SISD)
- Single Instruction Multiple Data (SIMD)
- Multiple Instruction Single Data (MISD)
- Multiple Instruction Multiple Data (MIMD)

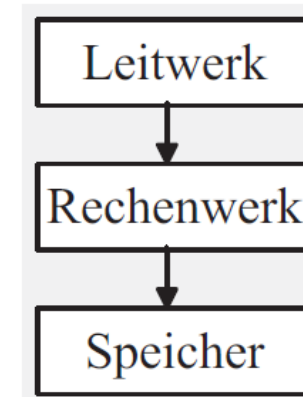


Rechnerarchitekturen

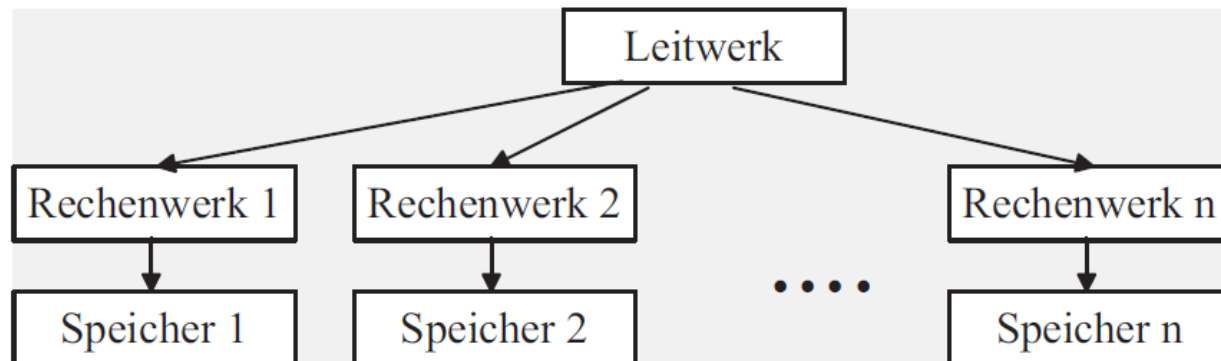
Klassifikation nach Flynn (Flynn'sche Taxonomie)

Single Instruction Single Data (SISD)

z.B. klassischer v. Neumann Rechner



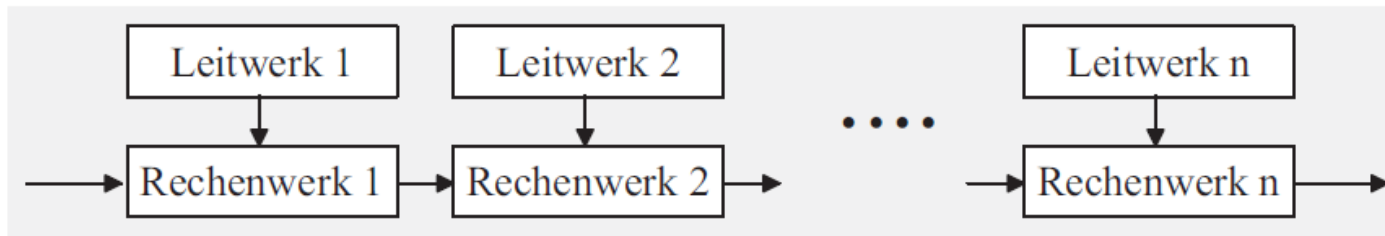
Single Instruction Multiple Data (SIMD)



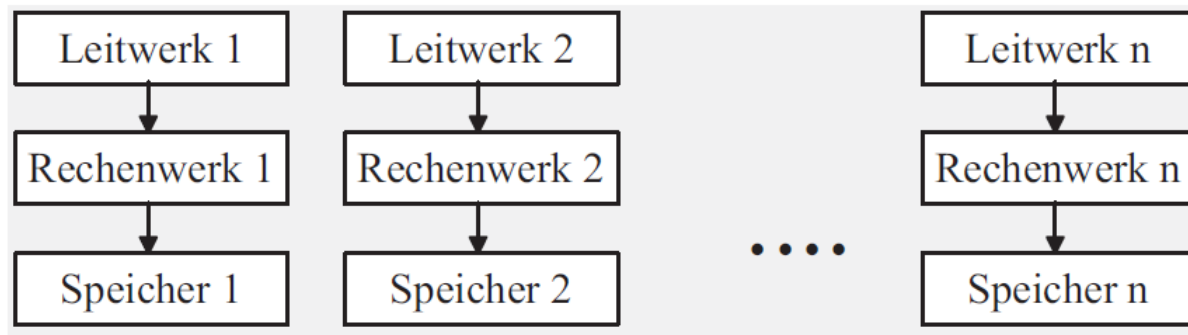
Rechnerarchitekturen

Klassifikation nach Flynn (Flynn'sche Taxonomie)

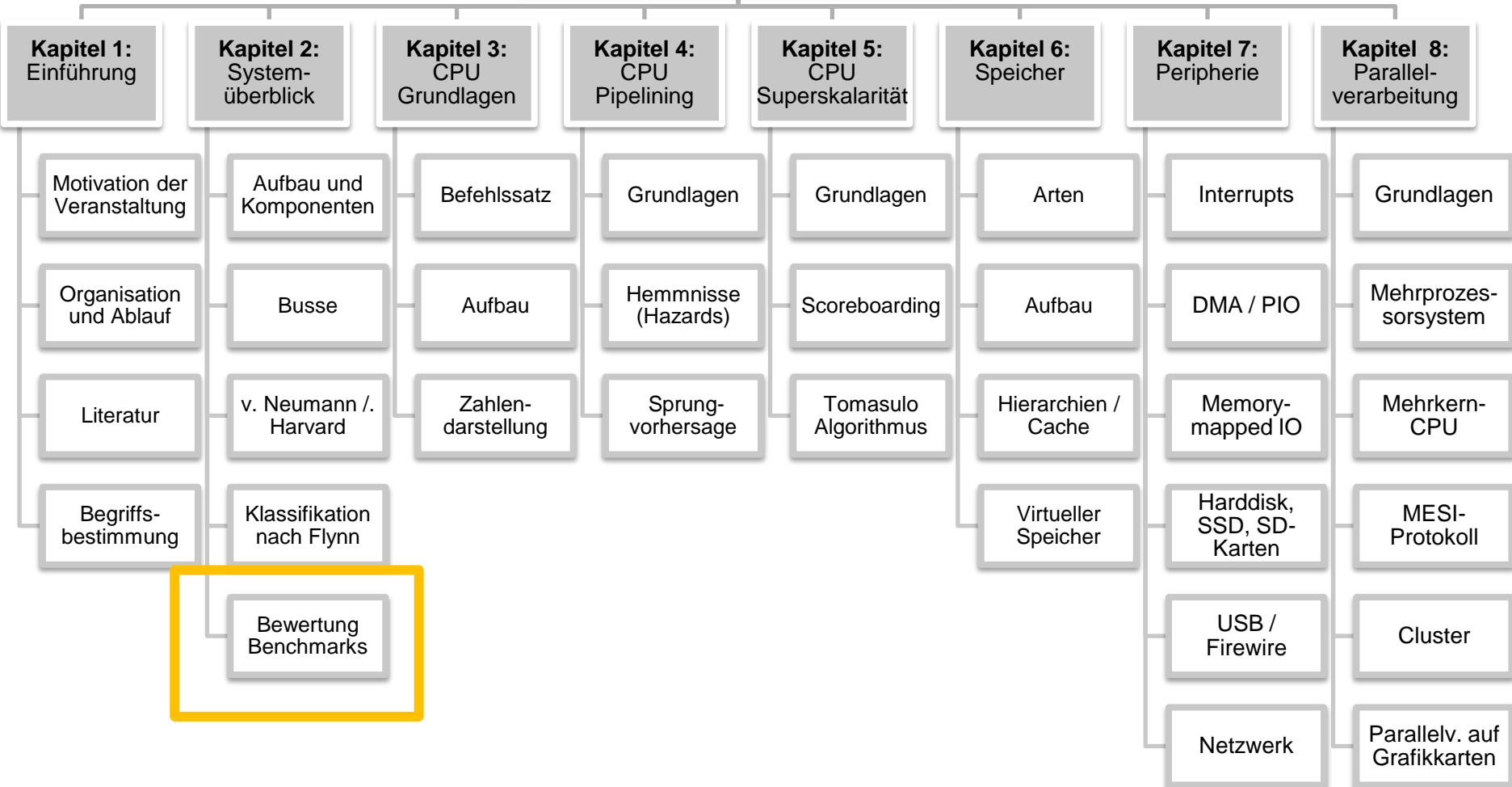
Multiple Instruction Single Data (MISD)



Multiple Instruction Multiple Data (MIMD)



Rechnerarchitektur



Leistungsmessung

Einfache Leistungsmaße

Taktfrequenz

- Lange Zeit das übliche **Maß für Marketing** von Prozessoren
- **Keine Aussage** über Bearbeitungsdauer eines Befehls!
- Prozessortakt üblicherweise schneller als Bus-/Speichertakt

Instructions **P**er **S**econd (IPS) / **M**illion Instructions **P**er **S**econd (MIPS)

- Anzahl der Instruktionen pro Sekunde (bei MIPS: in Millionen)

$$\text{MIPS} = \frac{n}{t \cdot 10^6}$$

n : Anzahl Instruktionen
 t : Ausführungszeit in s

- Klassisches Maß, heute selten verwendet, da
 - Abhängig vom Befehlssatz
 - Programmabhängig (auch auf dem gleichen System!)
 - Nicht proportional zur benötigten CPU Zeit



Leistungsmessung

Einfache Leistungsmaße

Floating Point Operations Per Second (FLOPS)

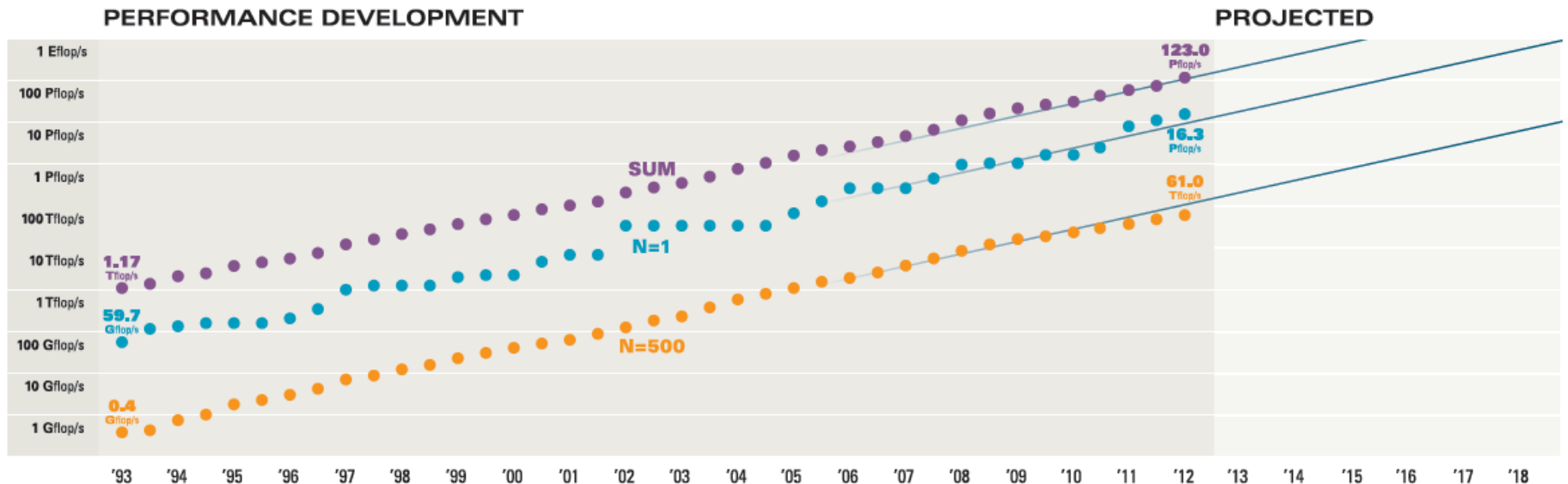
- Häufig für Supercomputer genutztes Maß (siehe nächste Folie)
- Gleiche Einschränkungen in der Aussagekraft wie bei MIPS
- Beispiele:
 - Sequoia (USA, 2012): $16,33 \cdot 10^{15}$ FLOPS = 16,33 PFLOPS
= 16330 TFLOPS
 - Grafikkarte mit Vektorprozessor (ATI Radeon HD 5970):
 $4,64 \cdot 10^{12}$ FLOPS = 4,64 TFLOPS



Leistungsmessung

Top 500 Liste der Supercomputer (Ausschnitt)

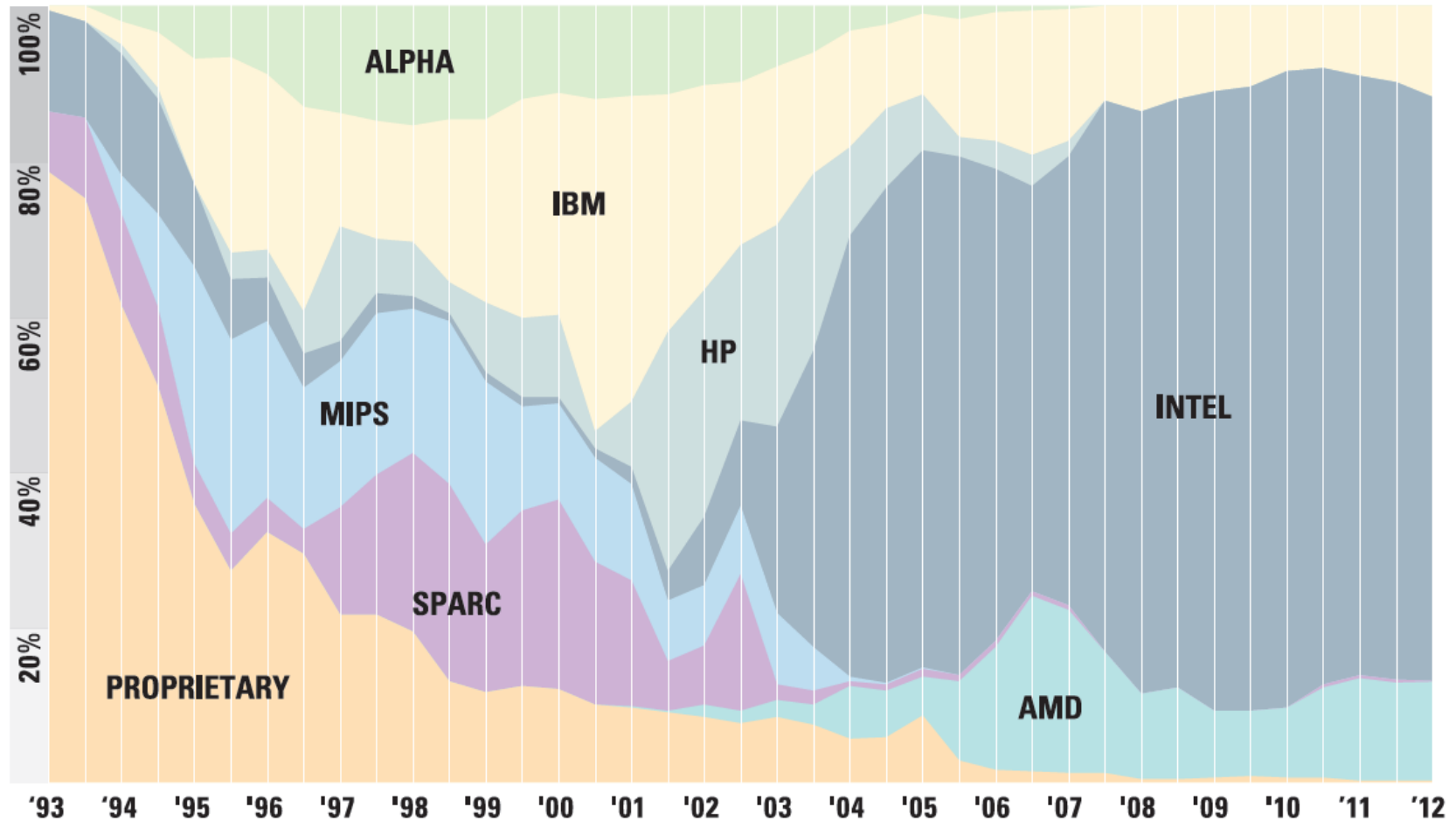
	NAME	SPECS	SITE	COUNTRY	CORES	R _{max} Pflop/s
1	Sequoia	IBM BlueGene/Q, Power BQC 16C 1.60 GHz, Custom interconnect	DOE / NNSA / LLNL	USA	1,572,864	16.33
2	K computer	Fujitsu SPARC64 VIIIfx 2.0GHz, Tofu interconnect	RIKEN AICS	Japan	705,024	10.51
3	Mira	IBM BlueGene/Q, Power BQC 16C 1.60 GHz, Custom interconnect	DOE / SC / ANL	USA	786,432	8.153
4	SuperMUC	IBM iDataPlex DX360M4, Xeon E5-2680 8C 2.70GHz, Infiniband QDR	Leibniz Rechenzentrum	Germany	147,456	2.897
5	Tianhe-1A	NUDT YH MPP, Xeon X5670 6C 2.93 GHz, NVIDIA 2050	NUDT/NSCC/Tianjin	China	186,368	2.566



Quelle: <http://www.top500.org>, Stand: 9/2012

Leistungsmessung

Top 500 Liste: Anteil der Prozessorarten



Quelle: <http://www.top500.org>, Stand: 9/2012

Leistungsmessung

Einfache Leistungsmaße

Cycles Per Instruction (CPI)

- Anzahl benötigter Takte pro ausgeführter Instruktion
- Aussagekräftig wenn als Durchschnittswert zusammen mit der Taktfrequenz genannt
- Zusammenhang zwischen MIPS in CPI:

$$\text{MIPS} = \frac{n}{t \cdot 10^6} = \frac{f}{\text{CPI} \cdot 10^6}$$

n : Anzahl Instruktionen
 t : Ausführungszeit in s
 f : Takt in Hz

Instructions Per Cycle (IPC)

- Kehrwert der CPI



Leistungsmessung

Einfache Leistungsmaße

Leistungsaufnahme (pro ausgeführter Instruktion)

- Z.B. in Joule/Instruktion oder GOps/Watt
- Beispiele
 - Intel P4: 0,1 GOps/Watt
 - DSP: 5 GOps/Watt
 - Supercomputer BlueGene/Q (IBM Thomas J. Watson Research):
2,1 GFLOPS/Watt

Green 500 List: <http://www.green500.org>



Leistungsmessung

Einfache Leistungsmaße

Frames Per Second (FPS)

- Anzahl pro Sekunde dargestellter Bilder (z.B. im Gaming Bereich)
- Einfach messbar z.B. mit dem Tool FRAPS <http://www.fraps.com>

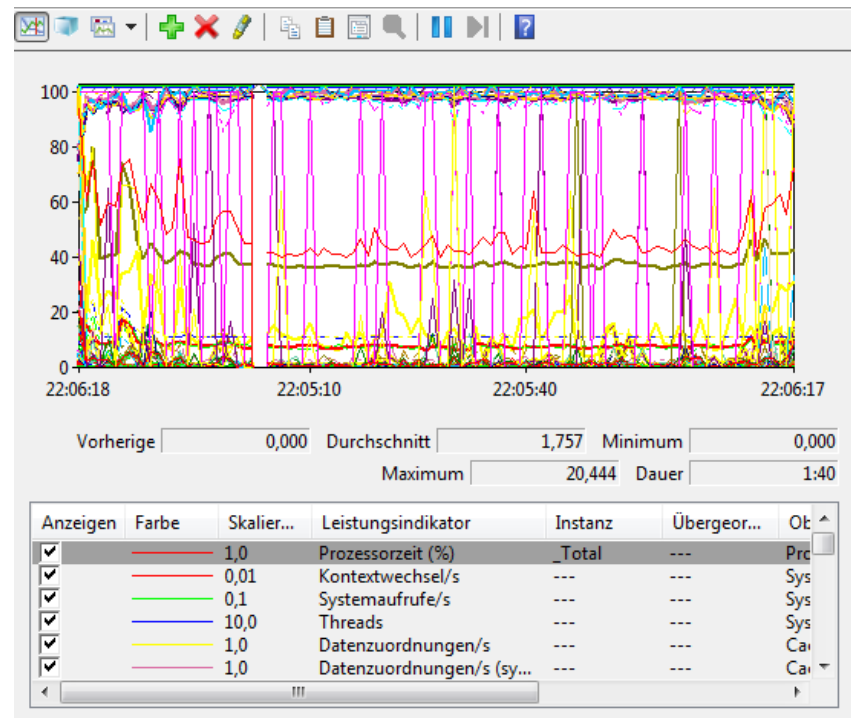
Arbeitsweise:
Vermutlich mittels
Hooking in die swap-
buffer oder draw
Funktion der Grafik
Module



Leistungsmessung

Auslastungsmessungen

- Messung von charakteristischen Parametern, die Auslastung des Systems beschreiben
- Beispiel Windows: perfmon (Windows Leistungsüberwachung)
 - CPU Auslastung, Frequenz, Kontextwechsel
 - Netzwerkzugriffe
 - HD-Zugriffe und Datenraten
 - Speicherauslastung
 - Systemaufrufe



- Beispiel Linux: top

Leistungsmessung

Kommerzielle Leistungsmessung

Problematisch ist

- Neutralität der Messungen bzw. Ergebnisdarstellung
- Praxisrelevanz der Benchmarks
(→ vs. Optimierung des Systems auf den Benchmark?)

Werte hängen stets von drei Größen ab

- Compiler
- Prozessor (CPU)
- Speicherarchitektur und -ausstattung



Leistungsmessung

Beispiel: Einfluss des Compilers (1/2)

Folgendes C-Programm wurde mit gcc für MMIX übersetzt:

```
#include <stdio.h>
int main ( )
{
    register int i , j =0;
    for ( i =0; i <100; i ++ )
        j += i ;
    printf ( "%d\n " , j ) ;
    return 0;
}
```

- Was wird berechnet?
- Wie beurteilen Sie den generierten Code? (siehe folgende Folie)



Leistungsmessung

Beispiel: Einfluss des Compilers (2/2)

Vom gcc für das
Beispiel auf der
vorheriger Seite
generierter Code:

12	main	IS	@	43	ADDU	\$0,\$2,1	
13		SUBU	\$254,\$254,32	44	SUBU	\$1,\$253,12	
14		STOU	\$253,\$254,24	45	STTU	\$0,\$1,0	
15		ADDU	\$253,\$254,32	46	JMP	L:2	
16		GET	\$1,rJ	47	L:3	IS	@
17		SUBU	\$0,\$253,32	48		SUBU	\$2,\$253,20
18		STOU	\$1,\$0,0	49		LDT	\$2,\$2,0
19		SETL	\$0,0	50		SLU	\$0,\$2,32
20		SUBU	\$2,\$253,20	51		SR	\$0,\$0,32
21		STTU	\$0,\$2,0	52		GETA	\$4,LC:0
22		SETL	\$2,0	53		SET	\$5,\$0
23		SUBU	\$1,\$253,12	54		PUSHJ	\$3,printf
24		STTU	\$2,\$1,0	55		SUBU	\$0,\$253,32
25	L:2	IS	@	56		LDO	\$0,\$0,0
26		SUBU	\$1,\$253,12	57		PUT	rJ,\$0
27		LDT	\$1,\$1,0	58		SETL	\$0,0
28		SLU	\$0,\$1,32	59		LDO	\$253,\$254,24
29		SR	\$0,\$0,32	60		ADDU	\$254,\$254,32
30		CMP	\$0,\$0,99	61		POP	1,0
31		BNP	\$0,L:5				
32		JMP	L:3				
33	L:5	IS	@				
34		SUBU	\$2,\$253,20				
35		LDT	\$1,\$2,0				
36		SUBU	\$2,\$253,12				
37		LDT	\$2,\$2,0				
38		ADDU	\$0,\$1,\$2				
39		SUBU	\$1,\$253,20				
40		STTU	\$0,\$1,0				
41		SUBU	\$2,\$253,12				
42		LDT	\$2,\$2,0				



Leistungsmessung

Einfluss einer Komponente – Amdahls Gesetz

Für die Beschleunigung gilt:

$$\text{Beschleunigung} = \frac{\text{Ausführungszeit}_{\text{alt}}}{\text{Ausführungszeit}_{\text{neu}}}$$

Wie wirkt sich die Beschleunigung einer Komponente auf das Gesamtsystem aus?

- Beschleunigung eines Teilsystems um einen Faktor B^* wirkt nur dort, wo das Teilsystem zur Verarbeitung beiträgt.
- Hat das Teilsystem einen Laufzeitanteil A_v ($0 \leq A_v \leq 1$), so gilt

$$\text{Beschleunigung} = \frac{1}{(1 - A_v) + \frac{A_v}{B^*}}$$

(Amdahls Gesetz)



Leistungsmessung

Kommerzielle Leistungsmessung

Beispiele für Benchmarks zur Beurteilung kommerzieller Systeme
(*Benchmark: engl. für Bewertung, Bezugsmarke*)

Prozessor-Benchmarks

- Whetstone
(National Physical Laboratory in UK, 1976)
- Dhrystone
(Dr. Reinhold Weicker, 1984)
- CoreMark
(Shay Gal-On, EEMBC, 2009)

System-Benchmarks

- SPEC
(Standard Performance Evaluation Corporation, 1988)
- SYSmark
(Business Applications Performance Corporation, 1991)
- TPC – Bewertung von Transaktionssystemen/Datenbanksystemen
(Transaction Processing Performance Council, 1988)



Leistungsmessung

Prozessor Benchmarks: Dhrystone (1984)

- Bestandteile wie typisches Anwendungsprogramm
 - Integer-Arithmetik, Verzweigungen, String-Operationen, Speicherzugriffe
- Referenzsystem: VAX 11/780 von DEC (1 MIPS, 1757 Dhrystone/s)
- Angabe der Testergebnisse in Dhrystone/s oder Dhrystone Instructions per Second (DMIPS)
- Umrechnung über Referenzsystem: 1 DMIPS entspricht 1757 Dhrystone/s
- Beispiele

Dhrystone 2.1
keine Optimierung

AMD 386	40 Mhz	4,5 DMIPS
Intel Pentium III	1000 Mhz	465 DMIPS
Intel Core i7 930	2800-3066 Mhz	1660 DMIPS
ARM v7-A9	1500 Mhz	768 DMIPS

Quelle: <http://www.roylongbottom.org.uk/dhrystone%20results.htm> , Stand: 9/2012

Reinhold P. Weicker. 1984. Dhrystone: a synthetic systems programming benchmark. *Commun. ACM* 27, 10 (October 1984), 1013-1030.



Leistungsmessung

Prozessor Benchmarks: Coremark (2009)

- Entwickelt von Embedded Microprocessor Benchmark Consortium (EEMBC)
- Ziel: Ablösung des Dhrystone durch neuen Benchmark der CPU bei heutigen Arbeitslasten besser charakterisiert
- Bestandteile
 - Matrixmanipulation
 - Suchen und Sortieren in einer Liste
 - Zustandsautomat (Prüfung eines Eingabedatenstroms)
 - CRC-Berechnung
- Angabe der Testergebnisse in Coremark (Durchläufe pro Sekunde) pro MHz



Leistungsmessung

Prozessor Benchmarks: Coremark (2009)

- Vorgegebenes Format zur Angabe der Ergebnisse:

CoreMark *V : N / C / P / M*

V: Version

N: Anzahl Iterationen pro Sekunde

C: Compiler-Version und -Flags

P: Parameter (z.B.: der Speicherkonfiguration)

M: Art und Anzahl bei paralleler Ausführung

Beispiel:

CoreMark 1.0 : 21054.3 / GCC4.2.4 -O3 / Stack / 4:PThreads

- Oft ebenfalls angegeben: CoreMark pro MHz
- Beispiele:

Prozessor	Takt	CoreMark	CoreMark/MHz
AMD 386	40 Mhz	24,27	0,61
AMD Phenom II X4	2600 Mhz	24828,46	9,55
Intel Core i7 3930K	3200 Mhz	150962,39	47,17
Infineon TC1796	150 Mhz	337,61	2,25

Quelle: <http://www.coremark.org> , Stand: 9/2012, Angabe aus Platzgründen nicht standardkonform



Leistungsmessung

Standard Performance Evaluation Corp. (SPEC)

- Benchmarks für CPU, Java, Grafik, Server
- Am bekanntesten ist CPU-Benchmark
(aktuell: CPU2006 V1.2, siehe <http://www.spec.org/cpu2006/>)

CPU-Benchmark

- Mehrere Suiten für verschiedene Zwecke definiert (Integer, Fließkomma, ...)
- Strenge Regeln („Run and Reporting Rules“) für Ermitteln und Veröffentlichen konkreter Werte (base/peak)
- Integer-Benchmark-Suite umfasst z.B.: Perl, bzip2 Kompression, Go-Spiel, Mustererkennung, Videokompression, XML-Verarbeitung, usw.



Leistungsmessung

Ermittlung von SPEC-Werten

- Referenzwerte für bestimmtes Referenzsystem werden vorgegeben (aktuell: SPARC Ultra Enterprise 2, 296 MHz UltraSPARC II, 1997)
- Alle Programme der Suite werden auf zu testendem System ausgeführt und die Laufzeiten gemessen
- Die einzelnen Werte werden auf das Referenzsystem bezogen und dann das geometrische Mittel errechnet:

Wenn r_i (für $1 \leq i \leq n$) die gemessenen Zeiten für die n Programme der Suite auf dem Referenzsystem sind und b_i die Messwerte auf dem zu testenden System, dann wird der SPEC-Wert S folgendermaßen berechnet:

$$S = \sqrt[n]{\prod_{i=1}^n \frac{r_i}{b_i}}$$



Leistungsmessung

Beispiel: SPEC-CINT 2000

Programm	r_i	b_i	Verhältnis
164.zip (Datenkompression)	1400	163	8,6
175.vpr (Schaltungsentwurf)	1400	217	6,45
176.gcc (C-Compiler)	1100	108	10,22
181.mcf (Optimierung)	1800	381	4,72
186.crafty (Schach)	1000	94,2	10,61
197.parser (Textverarbeitung)	1800	290	6,20
252.eon (Visualisierung)	1300	104	12,46
253.perlbnk (PERL Interpreter)	1800	199	9,03
254.gap (Gruppentheorie)	1100	161	6,83
255.vortex (OO-Datenbank)	1900	150	12,65
256.bzip2 (Datenkompression)	1500	206	7,27
300.twolf (Routing Simulation)	3000	380	7,90

Einfaches Beispiel zur Berechnung von S:

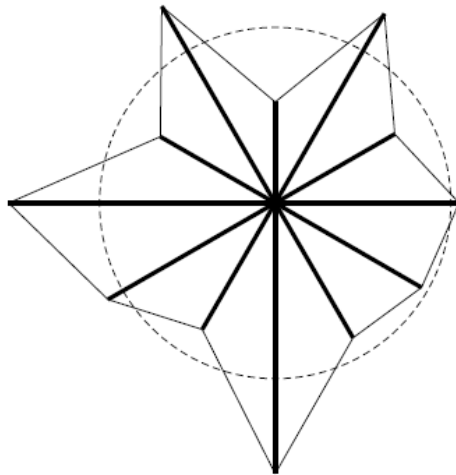
Bestünde die Suite nur aus gzip, vpr, gcc, dann $S = \sqrt[3]{\prod_{i=1}^3 8,6 * 6,45 * 10,22}$



Leistungsmessung

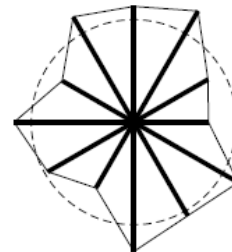
Starplots zur Darstellung von SPEC-Ergebnissen

Dell Power Edge 3250



SPEC 8,24

Compaq Alpha Server DS20E Model 6



SPEC 4,44

Quelle: iX



Leistungsmessung

BAPCo SYSmark 2012

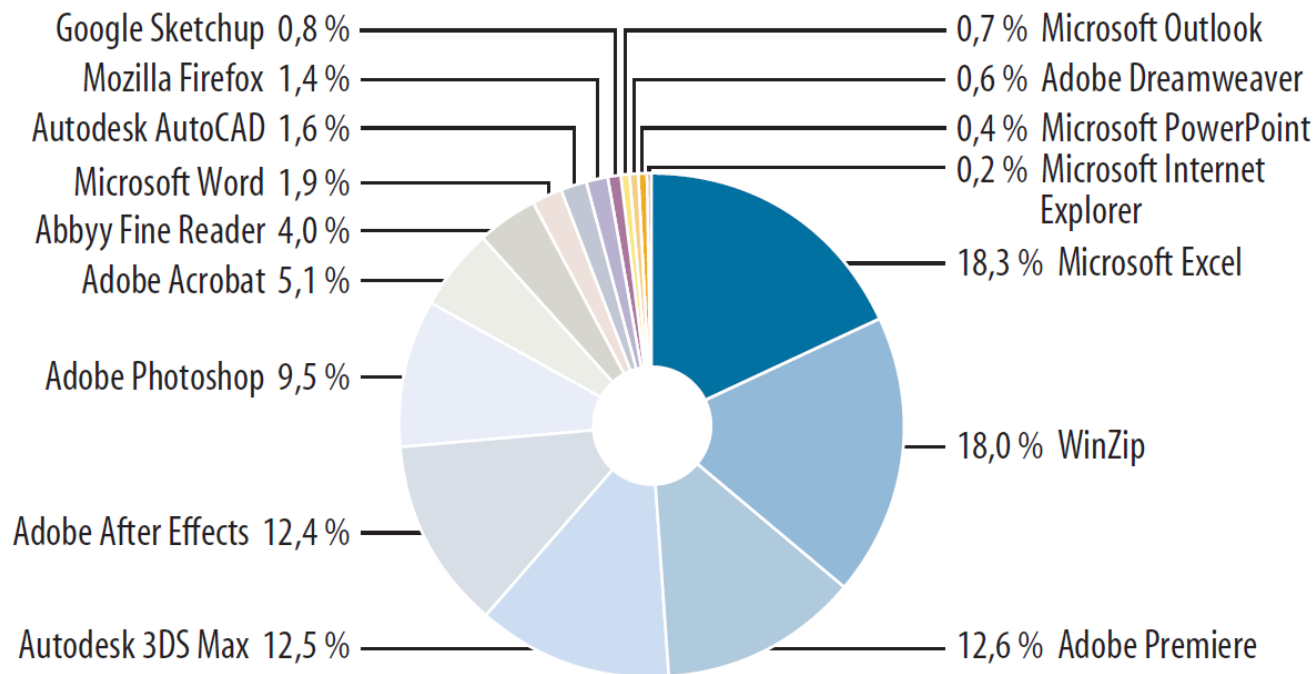
- Benchmark des Industriegremiums „Business Applications Performance Corporation (BAPCo)“
- Mitglieder u.a. Intel, Microsoft
- Empfohlen zum Leistungsvergleich von Desktop-Rechnern in öffentlichen Einrichtungen und Großunternehmen
- Besteht aus gängigen Windowsanwendungen in 6 Szenarien
 - Office Productivity
 - Media Creation
 - Web Development
 - Data/Financial Analysis
 - 3D Modeling
 - System Management
- Gesamtergebnis: geometrisches Mittel aller 6 Teilmessungen
- Referenzsystem (Intel Core i3) hat 100 Punkte in jeder Teilmessung



Leistungsmessung

BAPCo SYSmark 2012

Bestandteile des SYSmark 2012 und ihr Anteil an der Gesamtpunktzahl:



Quelle: c't 2012, Heft 3

Leistungsmessung

Transaction Processing Performance Council (TPC)

- Beurteilung von Gesamtsystemen (Transaktionsverarbeitung, Datenbankmanagementsysteme)
- Benchmarks simulieren typische Last transaktionsbasierter Systeme (z.B. Banktransaktionen, Bestellvorgänge)
- Messwerte:
 - Performance (transactions-per-minute, tmp)
 - Preis des Systems (USD) / Performance
 - Energiebedarf (Watt) / Performance
- Weitere Informationen: <http://www.tpc.org>



Systemüberblick

Zusammenfassung

- Unterschiedliche Komponenten eines Rechners werden über Busse oder Punkt-zu-Punkt Verbindungen verbunden
Praxisbeispiele: QPI, PCIe
- Architekturansatz nach v. Neumann vs. Harvard-Architektur
- Klassifikation nach Flynn: Einteilung von Rechnerarchitekturen hinsichtlich ihrer Instruktions- und Datenströme
- Einfache Leistungsmaße für Rechner: Takt, MIPS, FLOPS, CPI/IPC, Leistungsaufnahme pro Instruktion
- Verschiedene Benchmarks und ihre Einsatzgebiete
- Schwierigkeit der objektiven Bewertung der Leistungsfähigkeit durch Abhängigkeiten von Compiler, realitätsnahen Benchmark,...
- Auswirkung der Beschleunigung einer Komponente auf die Performance des Gesamtsystems (Amdahls Gesetz)



Kontrollfragen zu diesem Kapitel

- Was sind die Vorteile von Punkt-zu-Punkt Verbindungen gegenüber einem Bus?
- Funktionsweise des v. Neumann Rechners? Welchem Architekturansatz (v. Neumann/Harvard) folgt ein heutiger PC?
- Was sind einfache Leistungsmaße für Rechner?
- Welche Benchmarks können zur Bewertung der CPU bzw. des Gesamtsystems genutzt werden?
- Wie wird der SPEC Wert eines Systems berechnet?
- Was kann dazu führen, dass ein laut Benchmark schnelleres System A in der Praxis eine schlechtere Performance als ein System B aufweist?
- Sie verdoppeln in einem System den Takt des Speicherbusses. Wovon hängt es ab, wie sich dies auf die Gesamtperformance auswirkt? Wie berechnen Sie die Gesamtbeschleunigung?



Danksagung und Quellen

- Dieser Foliensatz basiert inhaltlich in großen Teilen auf einem älteren von Prof. Axel Böttcher, Hochschule München, entwickelten Foliensatz zur Rechnerarchitektur sowie dem entsprechenden Buch [1].
- Sämtliche Fehler im Foliensatz hingegen entstammen meiner Feder – falls Sie Fehler finden, bin ich Ihnen für einen kurzen Hinweis dankbar.
- Eine Liste weiterer Quellen finden Sie im Abschnitt „Empfohlene Literatur“ des Foliensatzes zu Kapitel 1.

