

# Betriebssysteme (BS)

## VL 11 – PC Bussysteme

**Daniel Lohmann**

Lehrstuhl für Informatik 4  
Verteilte Systeme und Betriebssysteme

Friedrich-Alexander-Universität  
Erlangen Nürnberg

WS 13 – 15. Januar 2014

[http://www4.cs.fau.de/Lehre/WS13/V\\_BS](http://www4.cs.fau.de/Lehre/WS13/V_BS)

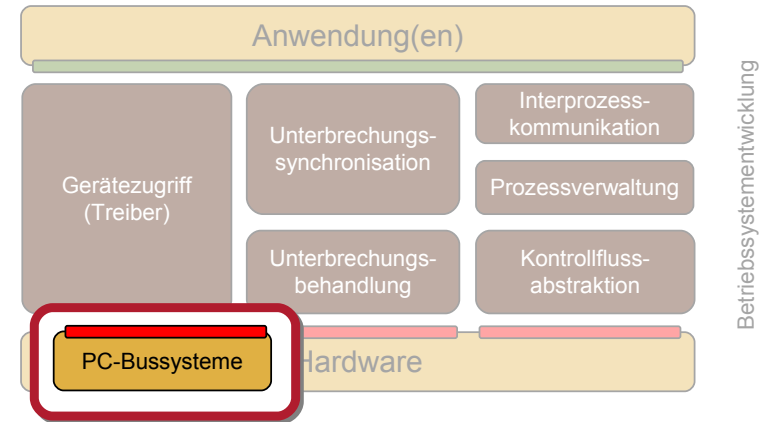


## Agenda

- Einordnung
- Rückblick
- PCI Bus
  - Grundlagen und Konfiguration
  - Initialisierung
  - Das PCI BIOS
- PCI Erweiterungen und Nachfolger
  - AGP
  - PCI-X
  - PCI Express
  - HyperTransport
- Zusammenfassung



## Überblick: Einordnung dieser VL



## Agenda

- Einordnung
- Rückblick
- PCI Bus
  - Grundlagen und Konfiguration
  - Initialisierung
  - Das PCI BIOS
- PCI Erweiterungen und Nachfolger
  - AGP
  - PCI-X
  - PCI Express
  - HyperTransport
- Zusammenfassung



## Rückblick – Bussysteme im PC

- seit es PCs gibt wurden die Anforderungen an den Systembus kontinuierlich größer:

Bussystem	PC	ISA	VLB	MCA	EISA	...
CPU's	ab 8088	ab 286	ab 386	ab 386	ab 386	
typischer Takt	4,7 MHz	8 MHz	25-50 MHz	10-25 MHz	8,33 MHz	
Multi-Master	nein	nein	ja (Version 2)	ja	ja	
Busbreite	8 Bit	16 Bit	32/64 Bit	32 Bit	32 Bit	
Adressraum	1 MB	16 MB	4 GB	4 GB	4 GB	
Transferrate	1 MB/s	4-5 MB/s	40/64 MB/s (Burst)	40 MB/s (Burst)	33 MB/s (Burst)	



## Rückblick – Bussysteme im PC

- seit es PCs gibt wurden die Anforderungen an den Systembus kontinuierlich größer:

Bussystem	...	PCI	AGP	PCI-X	PCI Express	Hypertransport
CPU's		ab 486	ab 486	ab P6	ab PIV (Xeon)	ab Hammer (AMD)
typischer Takt		33/66 MHz	66 MHz	bis 133 MHz	(variabel)	(variabel)
Multi-Master		ja	nein (max 1 Gerät)	ja	Punkt zu Punkt	ja, verschiedene Topologien möglich
Busbreite		32/64 Bit	32 Bit	32/64	bis zu 32 lanes	bis zu 32 links
Adressraum		4 GB/16 EB	4 GB	4 GB/16 EB	4 GB/16 EB	4 GB/16 EB
Transferrate		132/528 MB/s (Burst)	$n \times 266$ MB/s (1x, 2x, ...8x)	1064 MB/s (Burst)	2,5 GBit/s (Burst, pro lane)	1,6 GBit/s (Burst, pro link)



## Agenda

Einordnung  
Rückblick

PCI Bus

Grundlagen und Konfiguration

Initialisierung

Das PCI BIOS

PCI Erweiterungen und Nachfolger

AGP

PCI-X

PCI Express

HyperTransport

Zusammenfassung



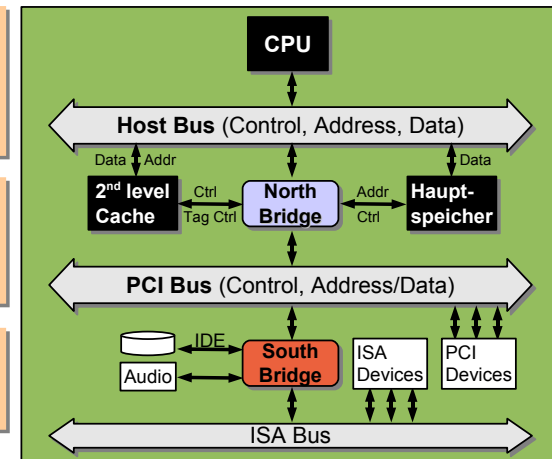
## PCI-basierte PC Systeme

- typische Architektur der ersten PCI Systeme:

Die North Bridge **entkoppelt** Host und PCI Bus. PCI Einheiten und CPU können so parallel arbeiten.

Die PCI Verbindung **zwischen North und South Bridge** wurde später durch etwas schnelleres ersetzt.

Durch die Bridges werden ISA und PCI **transparent** in einem System integriert.

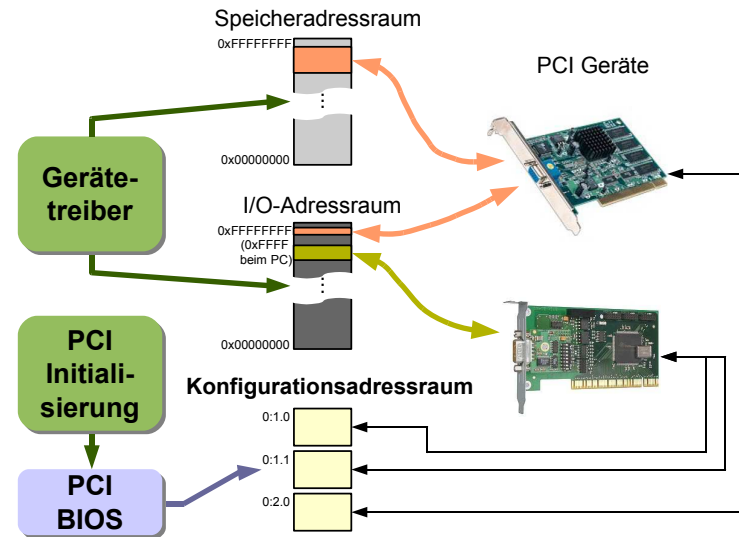


## PCI – die wichtigsten Daten

- Version 1.0 der Spezifikation von Intel (1991)
  - seit 1993 kommen die Spezifikationen von der PCI SIG
- 32/64 Bit, gemultiplexer Adress-/Datenbus
- im *Burst* Modus max. 132 MB/s bzw. 264 MB/s
- CPU-Typ unabhängig
  - PCI gibt es auch in Sparc, Alpha, ARM und PowerPC Systemen
- 4 Interruptleitungen (INTA-D)
- Skalierbarkeit durch *Bridges* und Multifunktionseinheiten
- Multi-Master Fähigkeit (besser als der klassische DMA)
- Schema zur Erkennung und Konfiguration von Geräten (Ressourcenzuweisung)

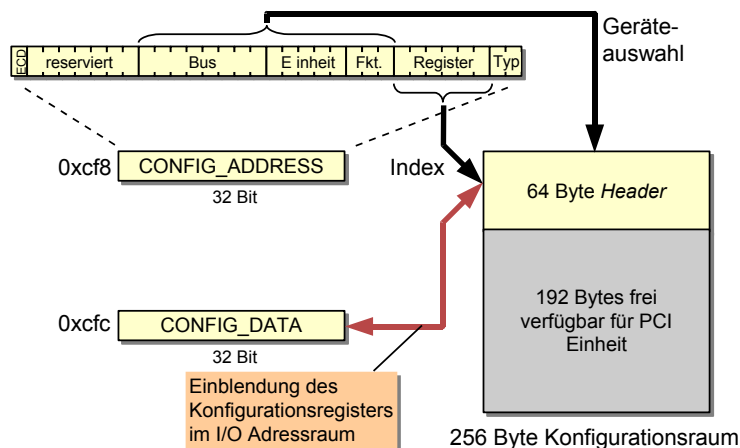


## Interaktion mit PCI Geräten



## Der PCI Konfigurationsadressraum (1)

- beim PC wird der Konfigurationsadressraum indirekt über I/O-Ports angesprochen:



## Der PCI Konfigurationsadressraum (2)

- Format des 64 Byte *Headers*:

Die **Einheiten-ID** 0xffff bedeutet 'nicht vorhanden'

Am **Header** Bit 7=1 kann man Multifunktions-einheiten erkennen

**BIST** erlaubt einen Selbsttest des Geräts

	31	16 15	0
0x00	Einheiten-ID		Hersteller-ID
0x04	Status		Befehl
0x08	Klassencode		Revision
0x0c	BIST	Header	Latenz
0x10	CLG		
0x14	Basisadressregister		
0x18			
0x1c			
0x20			
0x24			
0x28			
0x2c			
0x30			
0x34			
0x38			
0x3c	MaxLat	MinGNT	INT-Pin   INT-Leit.

Die **Einheiten-ID** und **Revision** identifizieren das Gerät eindeutig. **Hersteller-ID** und **Klassencode** sind Zusatzinformationen.

Mit dem **Befehl** lässt sich das Gerät aktivieren und deaktivieren.

Hier wird festgelegt, welche Adressbereiche die Einheit belegt. Gleichzeitig erfährt das System, wie groß der benötigte Adressraum ist.



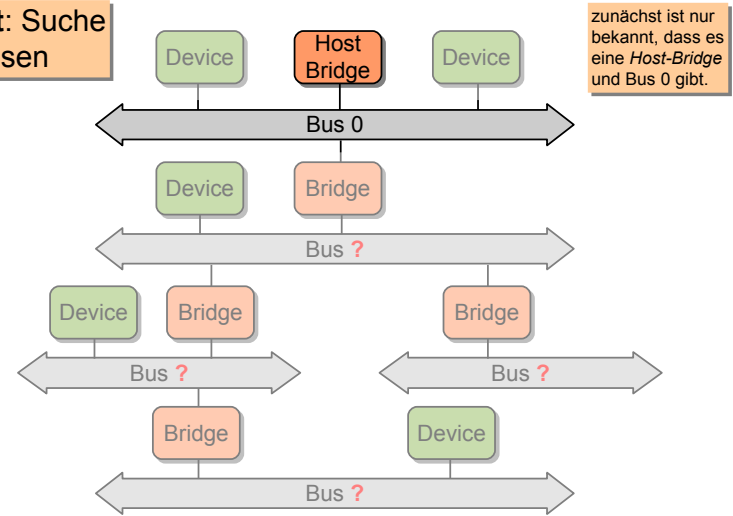
# PCI Initialisierung

- Bevor PCI Geräte durch ihre Gerätetreiber angesprochen werden können, muss folgendes erfolgt sein:
  - Konfigurierung der Basisadressregister der Geräte
  - Konfigurierung der PCI-Bridges
    - Speicherfensterregister – hängt von den Geräten unterhalb ab!
    - Busnummern (Primary, Secondary, Subordinate)
    - Subordinate ist die Nummer des letzten Busses unterhalb (downstream) der Bridge
- Das BIOS bzw. Betriebssystem muss die PCI Busstruktur schrittweise erforschen und initialisieren
  - bereits belegte Busnummern und Adressbereiche dürfen auf keinen Fall doppelt vergeben werden!



# PCI Initialisierung unter Linux

## 1. Schritt: Suche nach Bussen

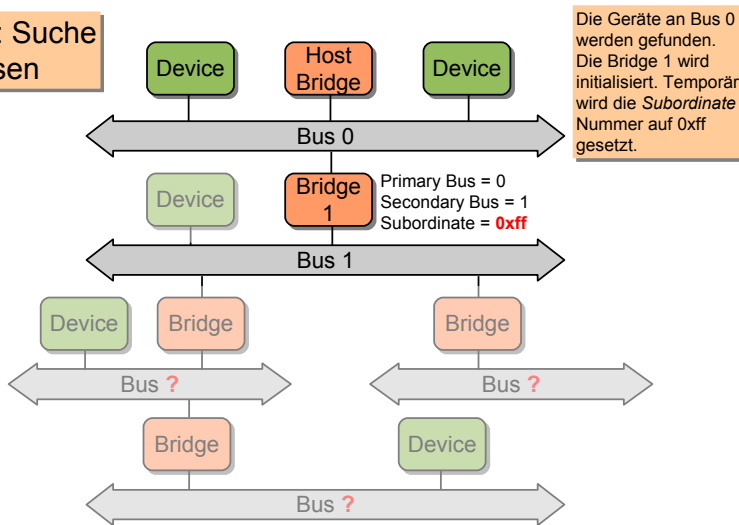


zunächst ist nur bekannt, dass es eine Host-Bridge und Bus 0 gibt.



# PCI Initialisierung unter Linux

## 1. Schritt: Suche nach Bussen

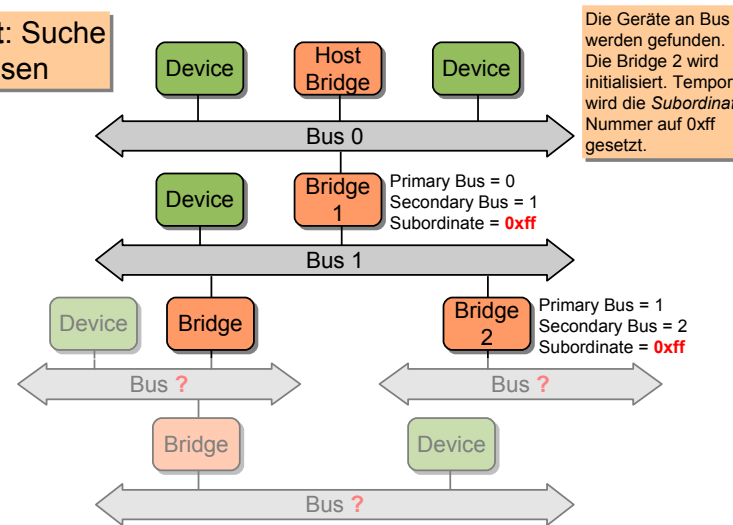


Die Geräte an Bus 0 werden gefunden. Die Bridge 1 wird initialisiert. Temporär wird die Subordinate Nummer auf 0xff gesetzt.



# PCI Initialisierung unter Linux

## 1. Schritt: Suche nach Bussen

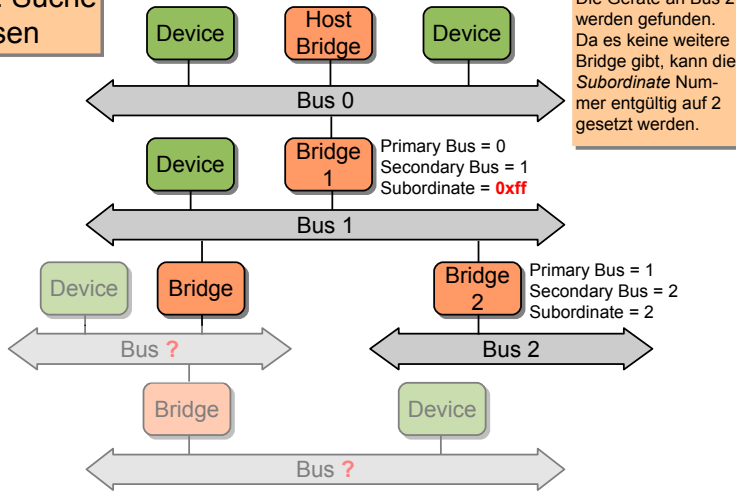


Die Geräte an Bus 1 werden gefunden. Die Bridge 2 wird initialisiert. Temporär wird die Subordinate Nummer auf 0xff gesetzt.



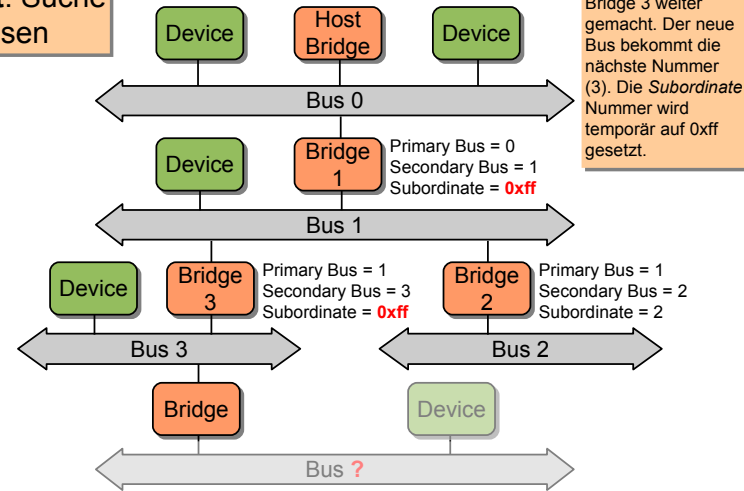
# PCI Initialisierung unter Linux

## 1. Schritt: Suche nach Bussen



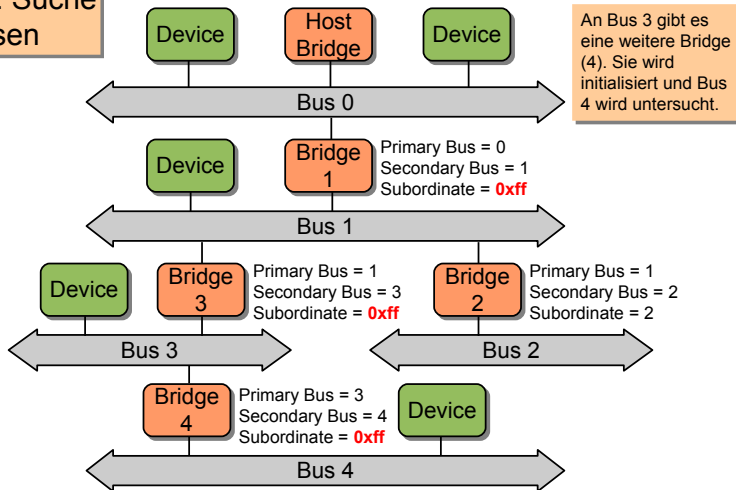
# PCI Initialisierung unter Linux

## 1. Schritt: Suche nach Bussen



# PCI Initialisierung unter Linux

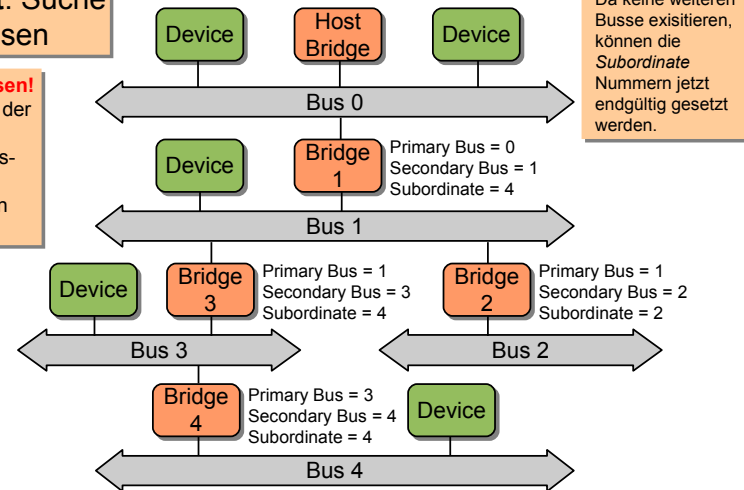
## 1. Schritt: Suche nach Bussen



# PCI Initialisierung unter Linux

## 1. Schritt: Suche nach Bussen

**Abgeschlossen!**  
Ab jetzt kann der komplette Konfigurations-Adressraum angesprochen werden.



## PCI Initialisierung unter Linux

### Algorithmus:

- Ausrichtung der aktuellen I/O und Speicheradressen auf die nächste 4K bzw. 1M Grenze
- für jedes Gerät des akt. Busses (in aufsteigender Reihenfolge der I/O Speicher-Anforderungen):
  - Reservierung der I/O und Speicheradressen
  - Aktualisierung der globalen I/O und Speicherzeiger
  - Initialisierung und Aktivierung des Geräts
- rekursive Anwendung des Algorithmus für alle angeschlossenen *Bridges*
- Ausrichtung der resultierenden Adressen (wie oben)
- Programmierung und Aktivierung der *Bridge*

### 2. Schritt:

Zuweisung der Adressen



## Das PCI BIOS – Überblick

- Festlegung durch PCI SIG (1993, Vorlage von Intel 1991)
- auf PCs normalerweise vorhanden, bei anderen Rechnertypen eher selten anzutreffen
- konfiguriert die PCI *Bridges* und Geräte beim Systemstart
  - minimal, falls ein "Plug&Play Betriebssystem" installiert ist
  - sonst komplett
- nach dem *Booten* erlaubt das PCI BIOS ...
  - die Suche von PCI Geräten nach Geräteklasse oder Typ
  - den Zugriff auf den Konfigurationsadressraum
- der Zugriff erfolgt über ...
  - den BIOS Interrupt 0x1a (*Real Mode*)
  - das "BIOS32 Service Directory" (*Protected Mode*)



## Das PCI BIOS – im Protected Mode

- das BIOS32 Service Directory erlaubt (im Prinzip) den Zugriff auf beliebige BIOS Komponenten
- es liegt irgendwo im Bereich von 0xE0000-0xFFFFF

Offset	Größe	Beschreibung
0x00	4 Bytes	Signatur "_32_"
0x04	4 Bytes	physikalische Einstiegsadresse (für call)
0x08	1 Byte	BIOS32 Version (0)
0x09	1 Byte	Länge der Datenstruktur / 16 (1)
0x0a	1 Byte	Prüfsumme
0x0b	5 Byte	reserviert (0)

- mit dem BIOS32 Service kann man testen, ob ein PCI BIOS vorhanden ist.



## Das PCI BIOS – Funktionsumfang

- folgende Funktionen umfasst das PCI-BIOS laut Spezifikation:

Funktionsname	Argumente	Resultate
<i>PCI BIOS Present</i>	-	ja/nein, letzte Busnr., Init.-Mechanismus
<i>Find PCI Device</i>	Device ID, Vendor ID, Index	Bus/Dev./Func. Nr.
<i>Find PCI Class Code</i>	Class Code, Index	Bus/Dev./Func. Nr.
<i>Generate Special Cycle</i>	Bus Nr.	-
<i>Get Interrupt Routing Opt.</i>	Pufferspeicher	Routing Möglichkeiten
<i>Set PCI Hardware Interrupt</i>	Bus Nr., Device Nr., Int.-Pin, Int.-Nr.	-
<i>Read Configuration Byte/Word/DWord</i>	Bus/Dev./Func./Reg. Nr.	gelesenes Byte/Word/DWord
<i>Write Configuration Byte/Word/DWord</i>	Bus/Dev./Func./Reg. Nr., zu schreibendes Byte/Word/DWord	-



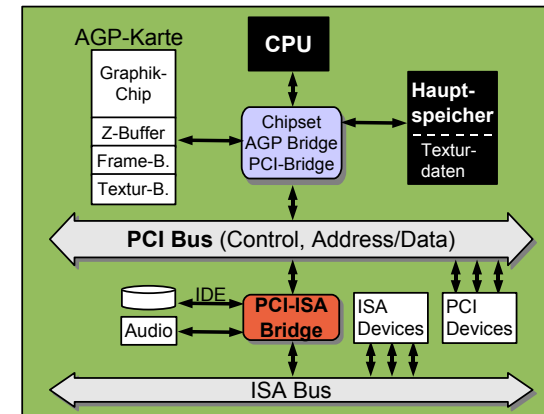
## Agenda

- Einordnung
- Rückblick
- PCI Bus
  - Grundlagen und Konfiguration
  - Initialisierung
  - Das PCI BIOS
- PCI Erweiterungen und Nachfolger
  - AGP
  - PCI-X
  - PCI Express
  - HyperTransport
- Zusammenfassung



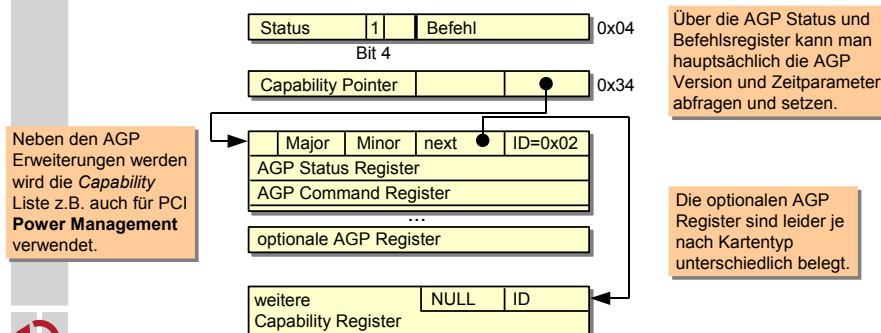
## AGP – Hardware

- *Accelerated Graphics Port* (1997)
- schnelle 1:1 Anbindung **einer** (3D) Graphikkarte
  - (theoretische) N x 266 MB/s Transferrate für AGP 1x, 2x, 4x, ...



## AGP – Initialisierung

- AGP Karte und *Bridge* präsentieren sich im System wie eine PCI-to-PCI *Bridge* und ein normales PCI Gerät
  - volle Software-Kompatibilität
- spezielle AGP Register lassen sich über die *Capability* Liste im Konfigurationsraum ansprechen:



## PCI-X (eXtended)

- Erweiterung des PCI Busses (1999)
  - von der PCI Special Interest Group (SIG) im PCI 3.0 Standard festgeschrieben
- erlaubt eine größere Bandbreite bei voller Kompatibilität
  - der PCI-X Bus benutzt den Arbeitsmodus des **langsamsten** Geräts

PCI-Kartentyp	PCI (konventionell)			PCI-X	
	33 MHz	33 MHz	66 MHz	66 MHz	133 MHz
Bus-Frequenz	33 MHz	33 MHz	66 MHz	66 MHz	133 MHz
Spannung	5 V	3,3 V/univ.	3,3 V/univ.	3,3 V/univ.	3,3 V/univ.
<b>Mainboard</b>					
PCI	33 MHz	33 MHz	33 MHz	33 MHz	33 MHz
PCI	66 MHz	-	33 MHz	66 MHz	33/66 MHz
PCI-X	66 MHz	-	33 MHz	33/66 MHz	66 MHz
PCI-X	100 MHz	-	33 MHz	33/66 MHz	66 MHz
PCI-X	133 MHz	-	33 MHz	33/66 MHz	66 MHz

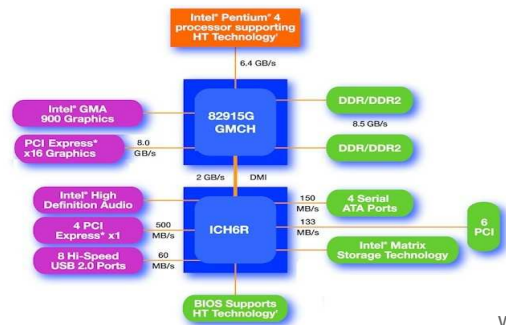
- neben der Takterhöhung gibt es auch *Split Transactions*
  - zugänglich wiederum über die *Capabilities* Liste





## PCI Express

- ... hat technisch wenig mit dem PCI Bus zu tun
- bidirektionale, serielle Punkt-zu-Punkt Verbindungen
  - Bandbreite pro Lane je Richtung: 512 MB/s, 8GB/s bei x16!
- ein typisches PC System mit PCI Express Geräten (i915)



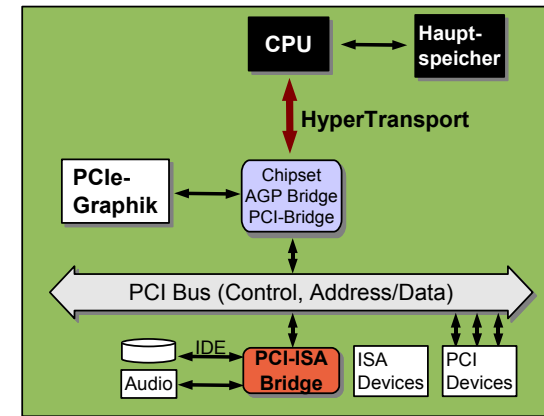
www.intel.com

\* Hyper-Threading (HT) Technology requires a computer system with an Intel Pentium 4 processor supporting HT Technology, and a HT Technology enabled chipset, BIOS and operating system. Performance will vary depending on the specific hardware and software you use. See www.intel.com/info/hyperthreading for more information including details on which processors support HT Technology.

11-29

## HyperTransport

- (AMD-)CPU integriert Speichercontroller und L2-Cache
- standardisierte Kommunikation** mit North Bridge: HyperTransport



11-30

## HyperTransport

- Versionen 1.0 (2001), 1.1, 2.0 und 3.0 (2006)
  - Konsortium: u.a. AMD, Apple, Cisco, NVIDIA, Sun
- bidirektional, Punkt-zu-Punkt, *Links* mit 2-32 Bit, Taktung bis zu 2,6GHz (DDR)
- je nach Version und Konfiguration bis zu 20,8 GB/s
  - bei aktuellen AMD Sockel-939-Prozessoren: HT 2.0 mit 4GB/s
- Gerätekonfiguration wie bei PCI
- weitere Anwendungen neben FSB-Ersatz
  - CPU-Kommunikation in AMD-Multiprozessor-Systemen
  - Chipsatz-Kommunikation (Northbridge ↔ Southbridge)
  - Kommunikation mit Coprozessoren: HTX
- Konkurrenz von Intel kam relativ spät
  - Intel QuickPath Interconnect (QPI, seit Ende 2008 (i7), 24-32 GB/s)

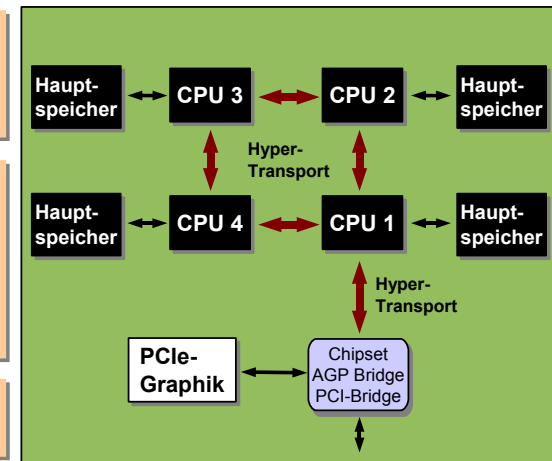
## HyperTransport in MP-Systemen

- NUMA (Non-Uniform Memory Architecture)

Die CPUs (u.U. mit mehreren Cores) kommunizieren untereinander via HyperTransport.

**Globaler Adressraum:** An andere CPUs angebundener Hauptspeicher kann adressiert werden, die Latenz ist jedoch höher.

Das Betriebssystem muss Tasks geeignet verteilen.





## Agenda

Einordnung

Rückblick

PCI Bus

Grundlagen und Konfiguration

Initialisierung

Das PCI BIOS

PCI Erweiterungen und Nachfolger

AGP

PCI-X

PCI Express

HyperTransport

Zusammenfassung



## Zusammenfassung

- im Bereich der PC Bussysteme dominiert seit Jahren PCI
- die neuesten Entwicklungen (PCI Express) haben kaum noch Ähnlichkeit mit dem PCI Bus von 1991
  - serielle Punkt-zu-Punkt Verbindungen und *Switches*
- neben den physikalischen Eigenschaften definiert PCI auch ein Programmiermodell
  - I/O- und Speicheradressräume
  - Konfigurierung und Initialisierung über Konfigurationsadressraum
  - Bus-Hierarchien
- auch die neuesten Entwicklungen sind auf der Ebene des Programmiermodells zu PCI kompatibel

