

Prüfungsunterlagen zur Vorlesung

PC-Technologie

Norman Hendrich

Universität Hamburg
Fachbereich Informatik
Vogt-Kölln-Str. 30
D 22 527 Hamburg
hendrich@informatik.uni-hamburg.de

Inhaltsverzeichnis

Allgemeines	1
Definition PC, Design-Guides	2
PC Architektur	4
Interrupt-Controller	7
Speicherbereiche	8
BIOS und DOS	9
Skalierung	11
Literatur	14
Die x86-Architektur	15
Befehlssatz	19
Register	20
Stack	23
Adressierungsarten	24
CISC vs. RISC	33
Befehls-Scheduling	34
superskalare Ausführung	36
x86-64 und IA64	38
SIMD-Erweiterungen	40
MMX	41
3Dnow!	45
SSSE	48
Speicherhierarchie	51
Performance Gap	52
DRAM	53
SDRAM	56
DDR-SDRAM	58
Rambus	59
Cache	63
IRAM	67
SMP-Multiprozessorssysteme	69
SMP	70
MESI	71
Exkurs: ASCII-Red	75
Bussysteme	78
ISA	80
ISA Plug and Play	83
PCI	86
AGP	90
Serielle Busse	91
USB Ziele	92
USB Architektur	93
USB Pakete	96
USB Deskriptoren	98
FireWire	101
Festplatten	105
Platten-Technologie	107
Datenformat: Beispiel FAT	108
IDE, ATAPI	112
SCSI	116
SCSI-Beispielkonfiguration	119
RAID	120
Dateicache	125

CD und DVD	127
CD Prinzip	127
CD Fehlerkorrektur	130
Dateiformate CD-DA, CD-ROM	132
CD-R und CD-RW	135
DVD	140
DVD Kopierschutz	143
DVD-R und DVD-RAM	145
Audio	146
Digitale Signalverarbeitung	148
AC97	149
Virtuelle Studios	151
DirectSound	153
Graphik	156
Anforderungen	157
Renderpipeline	159
Trends	163
DirectX	165
Mobile Geräte	171
Stromverbrauch	173
Displays	178
Vernetzung: IrDA	181
Vernetzung: GSM, Bluetooth	183

Vorlesung 18.215

PC-Technologie

Norman Hendrich

Universität Hamburg, Fachbereich Informatik, TECH

<http://tech-www.informatik.uni-hamburg.de/lehre/pc-technologie/>

PC-Technologie | SS 2001 | 18.214

Motivation und Ziele

- Technologiefortschritt weiterhin exponentiell (Moore's Law)
- Marktdominanz der "Wintel-Plattform"

=> PCs haben Technologieführung übernommen
 => Plattform wird ständig weiterentwickelt
 => immer weitere Anwendungsgebiete
 Beispiel Audio: Software-Synthesizer

(Software-Synthesizer RB-338)



Kennenlernen und Einschätzen von:

- Rechnerarchitektur PC
- Betriebssystemkonzepte
- aktuelle und zukünftige Entwicklungen

PC-Technologie | SS 2001 | 18.214

Themen

	Termine:
	05.04
• x86-Prozessoren	12.04
• MMX, 3DNow!, ISSE, IA64	19.04
• Systemarchitektur, Speicher von EDO bis RDRAM	26.04
	03.05
• Massenspeicher, IDE, SCSI, CDROM, DVD	10.05
• Datenübertragung und Busse, USB, Firewire, Modems	17.05
• Medienverarbeitung, Audio, Video, 3D-Graphik	24.05
	31.05
• Betriebssystemkonzepte und Treiber	07.06
• Altlasten: BIOS, MS-DOS, Win32	(14.06
• DirectX	21.06
	28.06
• Anforderungen für mobile Geräte	05.07
	12.07

PC-Technologie | SS 2001 | 18.214

Definition "PC"?

Was ist überhaupt ein PC?

- | | |
|--------------------------------------|-------------------------------|
| + Das Original, IBM-PC 1981: | 8088, 4.77 MHz, 64 KByte, ... |
| + "Aldi"-PC, 2000: | PIII, 666 MHz, 64 MByte, ... |
| + Microsoft und Intel Spezifikation: | PC98, PC99, PC2001, ... |

=> gemeinsame Merkmale?!

- + x86-Prozessor, ISA-Bus (nur noch bis 2001)
- + Systemsoftware, BIOS, DOS, Windows (bzw: Linux)
- PDAs, Handhelds, WAP-Handy, ...
- Mac G4, Sun Enterprise 4/450, ...
- Playstation 2, ...

PC-Technologie | SS 2001 | 18.214

Definition "Workstation" vs. "PC"

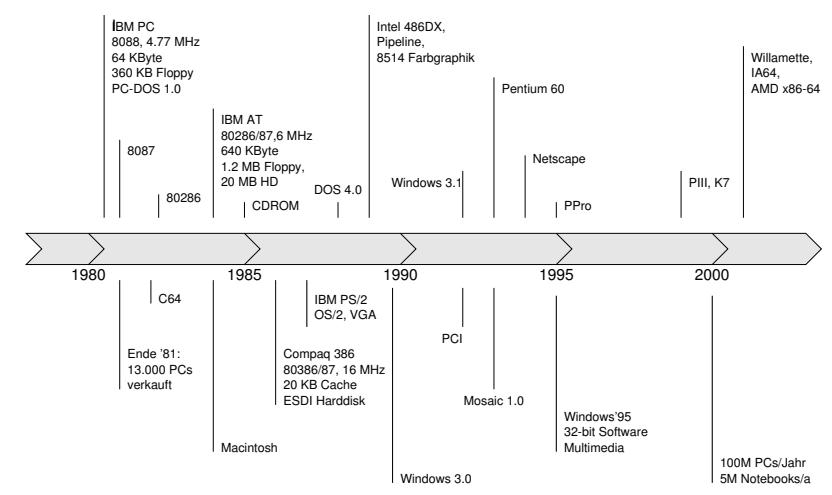
"4M"-Regel (ca. 1985):	Apollo DN-3000 1988:	PC-AT 1988:
1 MIPS	4	0.2
10 MByte Hauptspeicher	8	1
10 MBit/sec. Netzwerk	10	-
10 MPixel Farbgraphik	1024x800x8	640x480x4
Multitasking/Multiuser	ja/ja	nein/nein
Betriebssystem	AEGIS (Unix)	DOS 3.x
Oberfläche	GUI	Text
Monitor	19"	15"
Preis	DM 40.000	DM 10.000

Workstation vs. PC: Performance 03/2001

SPEC CPU2000 Benchmarks (baseline):	SPECint	SPECfp
AMD Athlon 1.2 GHz	443	387
Intel Pentium-III 1.0 GHz (VC820)	407	284
Intel Pentium-IV 1.5 GHz (VC850)	524	549
Compaq Alphaserwer 833 MHz	518	590
HP 9000 j6000	417	433
Sun Blade 900 MHz	438	482
<ul style="list-style-type: none"> keine offiziellen Werte für PowerPC alle anderen RISC weit abgeschlagen 		
<ul style="list-style-type: none"> Programme beanspruchen L1/L2-Cache + Hauptspeicher gleicher Speicher: sehr ähnliche Werte für alle Systeme 		

[www.spec.org/osg/cpu2000, Stand 03/2001]

Timeline



Definition PC: "Windows compatible"

"Windows-compatible" Logo von Microsoft:

- nur bei Einhalten aller Spezifikationen,
- Intel und Microsoft definieren Mindestanforderungen
- www.pcdesguide.org, PC98, PC99, PC2001, ...

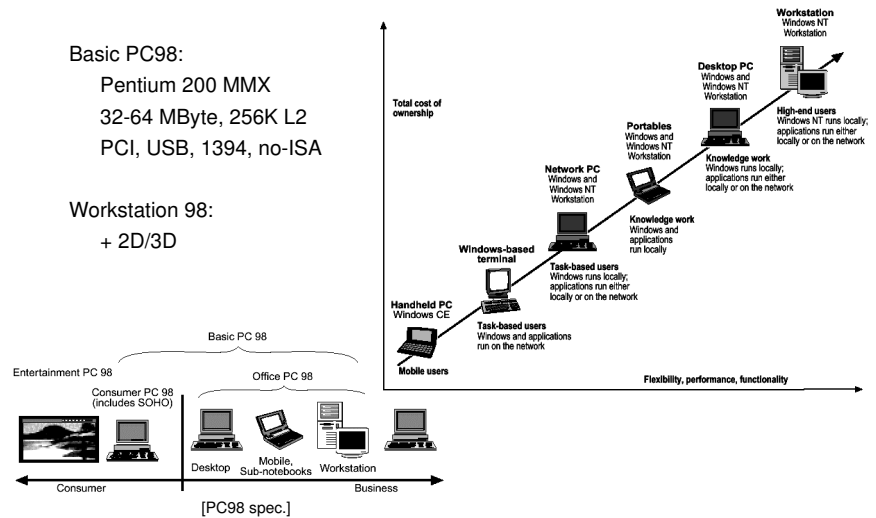


- definiert auch verschiedene Einsatzgebiete: Office, Home, Workstation, Server, ...
- => garantiert Kompatibilität
- => garantiert "Rente" für Intel u. Microsoft durch notwendige Upgrades
- aktueller Trend: Verzicht auf "legacy"-Schnittstellen (serielle/parallele/Joystick/analoge Audio- Ports fallen weg!)

Beispiel: PC98

Basic PC98:
 Pentium 200 MMX
 32-64 MByte, 256K L2
 PCI, USB, 1394, no-ISA

Workstation 98:
 + 2D/3D



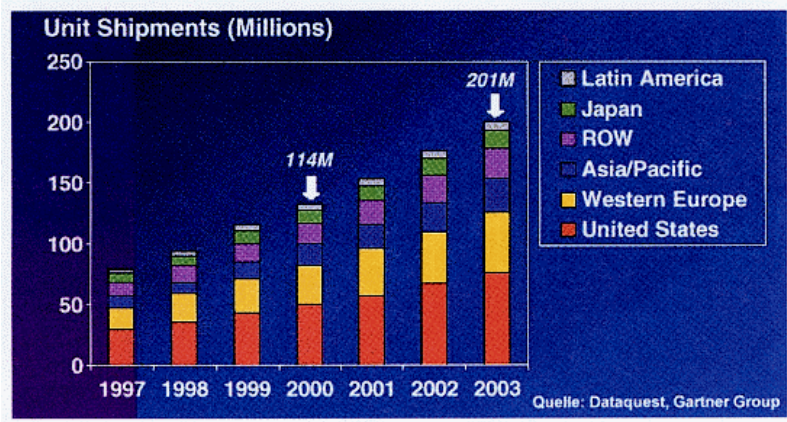
Beispiel: PC2001

Basic PC2001:
 500 MHz
 64 MByte, 128K L2
 PCI, USB, 1394, no-ISA
 keine "legacy" ports, keine Floppy
 4 USB
 1024x768x32 2D
 1024x768x16 + Zbuffer 3D
 1024x768 Video, optional DVD
 digitaler Monitorausgang

Mobile PC2001:
 600 MHz
 64 MByte, 128K L2
 Workstation 2001:
 700 MHz
 128 MByte, > 512K L2
 mehrprozessorfähig

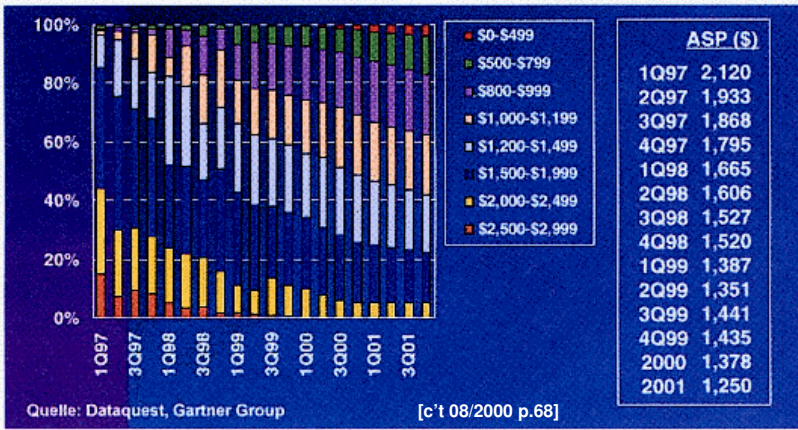
siehe PC2001 Spezifikation

PC: Stückzahlen



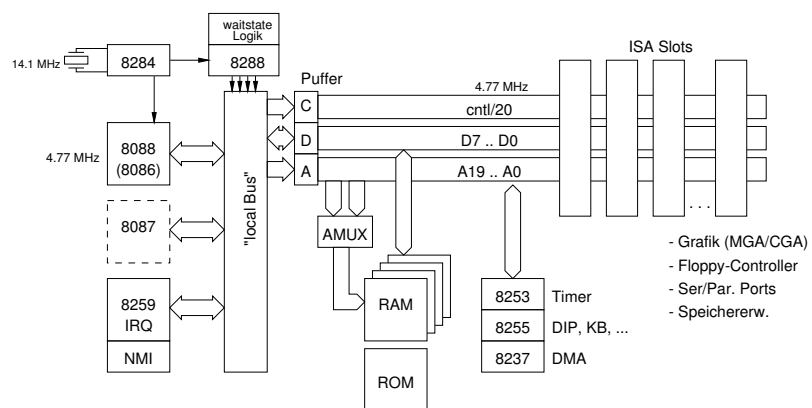
Zwar steigen die PC-Verkäufe laut Dataquest weiter an, die Margen für die Hersteller sollen aber noch weiter sinken. [c't 08/2000 p.68]

PC: Systemkosten



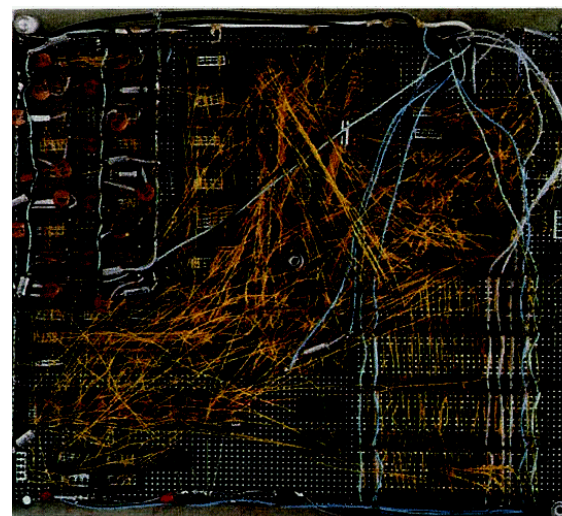
Geht es nach den amerikanischen Marktforschern, wird der Anteil von Rechnern, die einschließlich Monitor unter 1500 US-Dollar kosten, bald fast 80 Prozent der gesamten PC-Verkäufe ausmachen. [c't 08/2000 p.68]

ISA: PC/XT



- Intel 8088 mit Support-Chips (Takt, Timer, DMA, IRQ, ...)
- ein gemeinsamer 8-bit Bus für alle Komponenten

PC: Prototyp des IBM PC



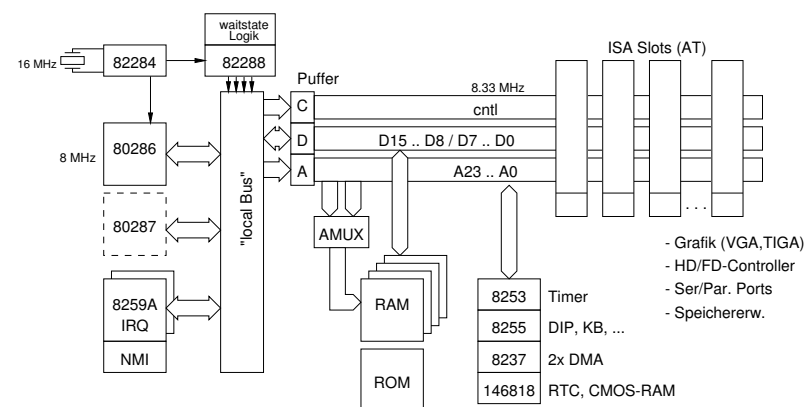
[c't]

ISA: PC/XT Eigenschaften

Original-IBM PC:

- Intel 8088, 4.77 MHz (Turbo-Versionen bis 10 MHz)
- real-mode, 1 MB Adressraum
- nutzt alle damals verfügbaren Support-Chips
- ein gemeinsamer Bus
- 20 bit Adressen (1MB), 8 bit Daten, diverse Steuerleitungen
- RAM / ROM mit am zentralen Bus
- RAM-Refresh über Timer und DMA
- 8 Interrupt-Quellen, 3 DMA-Kanäle frei
- weitere Peripherie (Grafik!) über Slots
- nur CPU und DMA als Busmaster

ISA: PC/AT



- Intel 80286 mit passenden Support-Chips
- gemeinsamer Bus, 8/16-bit Transfers

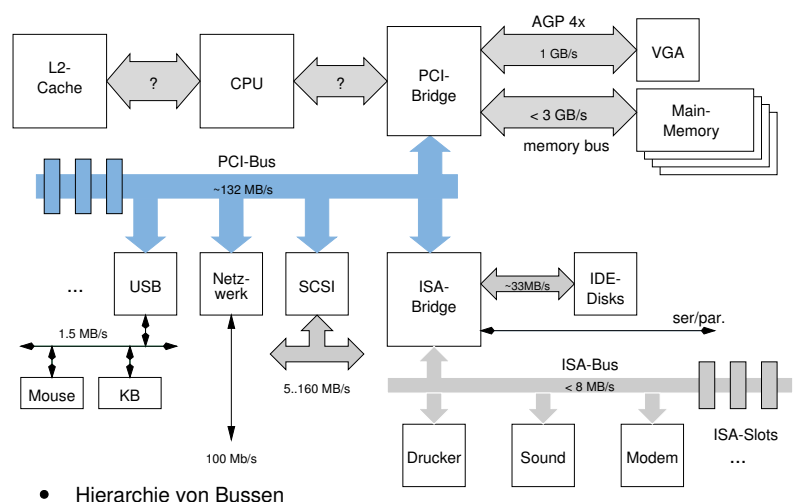
ISA: PC/AT Eigenschaften

- 80286/80287-Prozessor, plus passende Support-Chips
- 16-bit Daten, 24-bit Adressen
- real-mode oder protected-mode

- neue Slots, abwärtskompatibel für 8-bit XT-Karten
- eingeschränktes Busmastering möglich
- max. Bustakt 8.33 MHz ("ISA Standard")

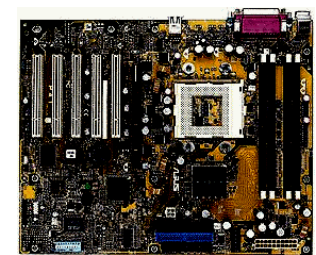
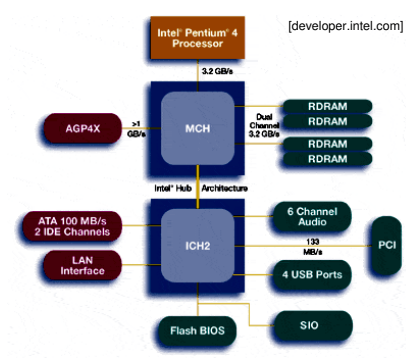
- 15 Interrupt-Kanäle
- insgesamt 7 DMA-Kanäle, davon 4x 8-bit, 3x 16-bit

PC: Pentium-PC



- Hierarchie von Bussen

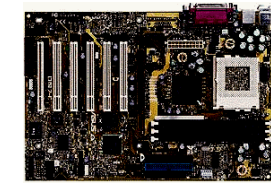
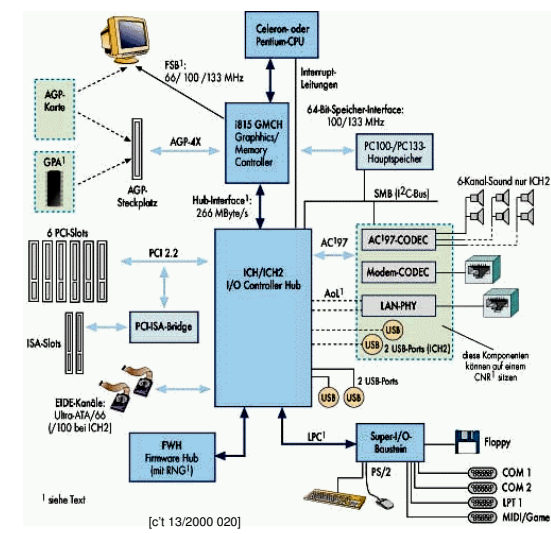
PC: Pentium-IV, Intel 850



Asus P4T (www.asuscom.de)

- Hierarchie von Bussen
- Chipsatz mit ähnlicher Komplexität wie Prozessor
- keine ISA-Unterstützung mehr

PC: "Solano" Chipsatz, i815



ASUS CUSL2 (www.asuscom.de)

Hardware vs. Software . . .

Hardware = "fest, schwer zu ändern, . . ."

- dramatische Evolution der PC-Hardware
- 8088 mit 8-bit Bus zum Pentium-III
- alle Komponenten um Größenordnungen verbessert

Ausnahmen bedingt durch Softwarekompatibilität (!)

- originale Interrupt- und I/O-Architektur erhalten
- sogar Bugs müssen vererbt werden (A20-Gate usw.)
- immer noch Engpässe mit I/O-Adressen und Interrupts
- PC2001 fordert (endlich) Verzicht auf "legacy" Komponenten
- aber immer noch interrupt-sharing usw.

PC-Technologie | SS 2001 | 18.214

PC: I/O-Konzept

- Trennung zwischen Speicher- und I/O-Bereichen
- nutzt die x86-Befehle für I/O-Transfers (inportb, outport, ...)
- nutzt das x86-Buskonzept:
 - gemeinsame Daten- und Adressleitungen
 - separate Steuerleitungen für Speicher und I/O
 - nur 64 KByte I/O-Adressraum
- 8086/8088 erlauben 1 MByte Adressraum für Speicher
- Aufteilung in 640 KByte RAM, oberhalb ROM und Graphik
- Interrupt-Architektur basiert auf dem Intel 8259 Controller
- zunächst nur acht Interruptebenen

PC-Technologie | SS 2001 | 18.214

PC: I/O-Adressen im AT

000 - 01F	DMA controller 1	8237
020 - 03F	Interrupt controller 1	8259
040 - 05F	Timer	8254
060 - 06F	Keyboard controller	8041
070 - 07F	real-time clock	
080 - 09F	DMA page register	
0A0 - 0BF	Interrupt controller 2	8259
0c0 - 0DF	DMA controller 2	8237
0F0	coprocessor, clear busy	80287
0F1	coprocessor reset	
0F8 - 0FF	coprocessor registers	
1F0 - 1F8	hard disk	
200 - 207	game i/o	
278 - 27F	parallel printer port 2	
2F8 - 2FF	serial port 2	
300 - 31F	prototype card	
360 - 36F	reserved	
370 - 378	parallel printer port 1	
380 - 38F	sdhc bisynchronous 2	
3A0 - 3AF	sdhc bisynchronous 1	
3B0 - 3BF	monochrome graphics	
3C0 - 3CF	reserved	
3D0 - 3DF	CGA graphics adapter	
3F0 - 3F7	diskette controller	
3F8 - 3FF	serial port 1	

nur 10 bit dekodiert ...

PC-Technologie | SS 2001 | 18.214

PC: IO-Adressen, Beispiel



- I/O-Adressraum gesamt nur 64 KByte
- je nach Zahl der I/O-Geräte evtl. fast voll ausgenutzt
- eingeschränkte Autokonfiguration über PnP-BIOS

PC-Technologie | SS 2001 | 18.214

PC: Interrupt-Konzept

flexibles Interrupt-Konzept der 8086-Familie:

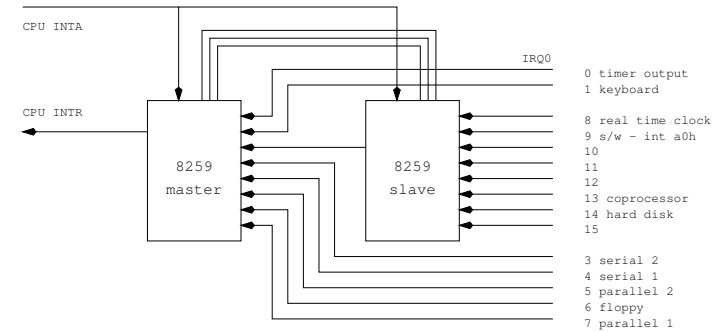
- ein Interrupt-Eingang am Prozessor
- zusätzlicher Eingang für NMI-Interrupt (non maskable)
- spezielle Buszyklen bei Reaktion auf INT-Signal
- Prozessor liest Interrupt-Nummer vom Bus
- 8259-Controller implementiert das zugehörige Busprotokoll
- oder Software-Interrupt auslösen
- eigene x86-Befehle (INTO, INT 3, INT n, BOUND)
- Interrupt-Nummer wird im AX-Register übergeben
- Standard-Mechanismus zum Aufruf von BIOS/DOS-Funktionen
- 256 Interrupt-Vektoren, ab Adresse 00000 im Hauptspeicher
- Vektornummer abhängig von IRQ-Quelle oder INT-Argument

PC: 8259 Interrupt-Controller

programmable interrupt controller Intel 8259

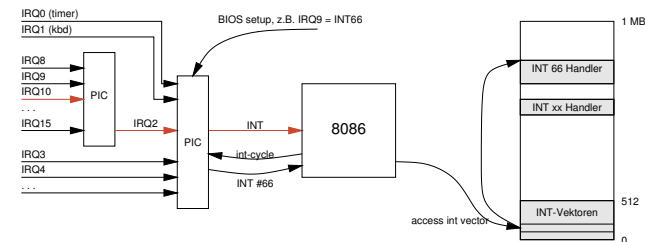
- Hilfsbaustein für die 8085/8086-Familie
- übernimmt Mapping von IRQ-Leitung zu Interrupt-Nummer
- Mapping per Software programmierbar
- kaskadierbar bis 8 Bausteine (56 Interrupt-Eingänge)
- nur ein Baustein im ursprünglichen PC
- zwei-Chip Kaskade seit PC/AT
- Hardware einfach erweiterbar, aber SW (DOS) leider nicht
- Multiprozessorsysteme erfordern verbesserten Controller (APIC)
- Details und INT-Sequenz: siehe 8259-Datenblatt (.pdf)

PC: Interrupts im AT



- nur vier freie Interrupts - ohne Sound, Graphik, ...
- weitere Kaskadierung wäre leicht möglich
- scheitert aber an Softwareunterstützung

x86: Interrupts im real-mode



```

void INT_66_handler() {
    save_registers_to_stack();
    read_master_PIC();
    if (master_PIC_active) { // hardware interrupt
        read_slave_PIC(); // but which one?
        switch( slave_PIC ) {
            case slave_IRQ8: // handle_RTC_interrupt
            case slave_IRQ9: // handle s/w int a0h
            case slave_IRQ10: // free
        }
        reset_slave_PIC();
        reset_master_PIC();
    }
    else { // software interrupt 66
        restore_registers();
        IRET;
    }
}
    
```

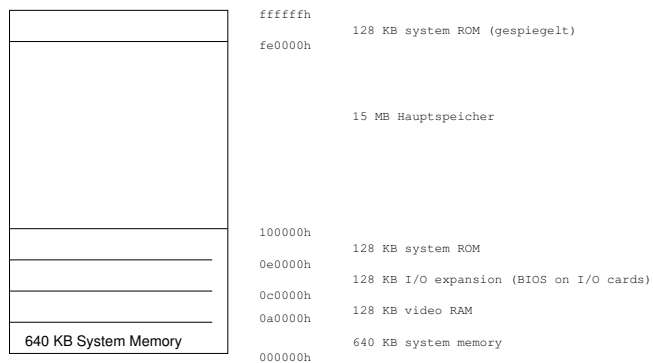
- BIOS programmiert den PIC 8259
- Umsetzung IRQ auf INT-Nummern
- INT-Vektoren ab Adresse 0

PC: Speicherbereiche im PC

ursprüngliche IBM Entwurfsentscheidungen:

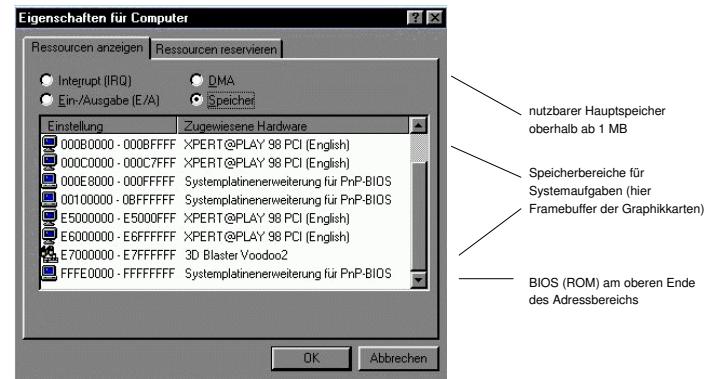
- 8086/8088 Adressraum ist 1 MByte
- für RAM, ROM, memory-mapped I/O
- zunächst 128 KB ROM am oberen Ende (wg. Reset)
- zunächst 640 KB RAM (ausgeliefert mit 64 KB)
- außerdem 128 KB Adressraum für Graphik
- Rest für spätere Erweiterungen reserviert
- entsprechende Aufteilung in DOS "hardkodiert"
- später diverse Erweiterungen auf "hohe" Speicherbereiche

PC: Speicherbereiche im AT



- 80286 adressiert bis 16 MByte Speicher
- unterer Bereich bis 1 MB ist PC-kompatibel
- ROM wird an obere Adressen gespiegelt

PC: Speicherbereiche, Beispiel



- Windows 9x erlaubt bis 4 GByte Adressraum
- Adressen 00000000h bis ffffffffh
- Aufteilung 1 GB / 1 GB / 2 GB

BIOS: Grundfunktionen

BIOS / Betriebssystemfunktionen:

- realisiert über x86 INT Befehl
- Register AX enthält die Interrupt-"nummer"
- andere Register verwendet zur Parameterübergabe
- zugeordnete Nummern:

BIOS:	00h .. 1Fh	z.B. 13h Disk-I/O
DOS:	20h .. 40h	z.B. 23h CNTL-C Handler
Anwender:	40h .. FFh	z.B. 4Fh SCSI, 6Fh Novell, ...

```

MOV AX, 05h      ; Funktionsnummer nach AX
MOV DL, "a"      ; Datenwerte nach DX, lower Byte
INT 21h          ; Software-Interrupt
                 ; gibt Zeichen "a" auf PRN aus
    
```

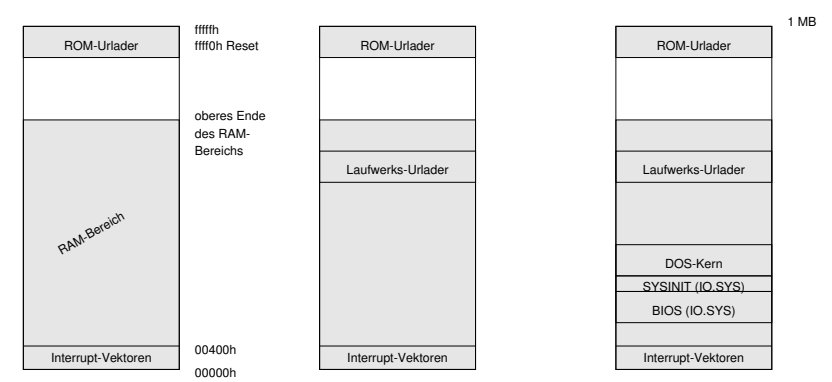
BIOS: Funktionen

00h	divide error	10h	graphic
01h	single-step, debugging	11h	bios get equipment list
02h	non-maskable interrupt	12h	bios get memory size
03h	breakpoint	13h	disk I/O
04h	into detected overflow	14h	serial ports
05h	print screen	15h	tape recorder / extensions
06h	invalid opcode (80286)	16h	keyboard I/O
07h	processor extension (reserved)	17h	printer I/O
08h	IRQ0, system timer	18h	diskless boot , ROM BASIC
09h	IRQ1, keyboard data ready	19h	system bootstrap loader
0ah	IRQ2, lpt2/ega/vga/ nested IRQ9	1ah	timer I/O
0bh	IRQ3, com2	1bh	keyboard break
0ch	IRQ4, com1	1ch	system data (graphic)
0dh	IRQ5, harddisk, lpt2	1dh	system data (disc params)
...		...	

DOS: Funktionen

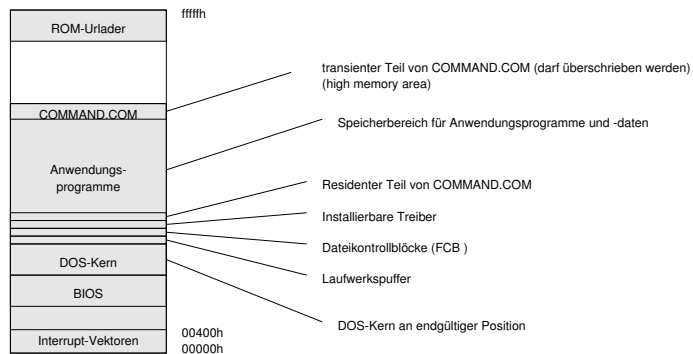
20h	terminate program
21h	misc. functions
22h	program termination address
23h	control-C / break handler
24h	critical error handler
25h	absolute disk read
26h	absolute disk write
...	
30h	far jmp instruction
33h	mouse
34h	floating point emulation
0bh	IRQ3, com2

PC: 8086 Reset und DOS-Boot



- Resetvektor ist ffff0h, dort System-ROM (BIOS)
- ROM-Urlader ermittelt Boot-Laufwerk, lädt Bootsektor ins RAM
- Bootcode lädt IO.SYS (BIOS) und MSDOS.SYS

PC: 8086 Reset und DOS-Boot



- DOS-Kern wird soweit nach unten verschoben wie möglich
- Dateipuffer und Treiber oberhalb des Kerns
- Anwendungsprogramme zwischen Puffern und Command.com

PC: Windows 2K Treiber

Figure 1.1 shows the major components of the Microsoft® Windows® 2000 operating system environment. [MS Win2K DDK]

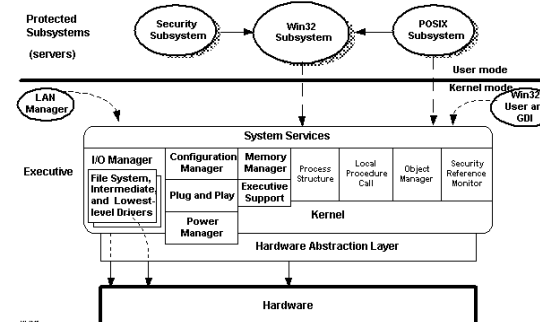


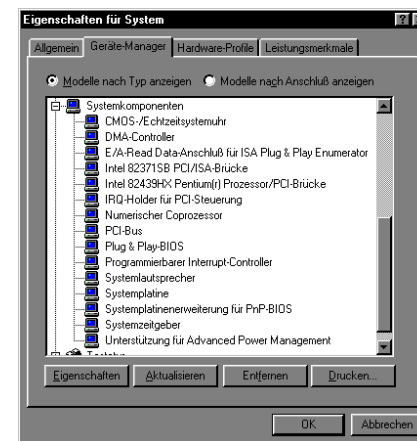
Figure 1.1 Windows 2000 Component Overview
 The Windows 2000 operating system environment includes some components that run in user mode and others that run in kernel mode. File system, intermediate, and lowest-level drivers are shown at the lower left, included with the kernel-mode I/O Manager. As Figure 1.1 shows, the Windows 2000 operating system includes a number of kernel-mode components with well-defined functionality isolated in each component. Those of most interest to kernel-mode driver writers are the Kernel, I/O Manager, Plug and Play Manager, Power Manager, Hardware Abstraction Layer (HAL), Configuration Manager, Memory Manager, Executive Support, and Process Structure components. Additional components of interest to some driver writers include the Object Manager and Security Reference Monitor. The Plug and Play (PnP) Manager and Power Manager are new components in Windows 2000. They support both Windows 2000-only drivers and WDM drivers. For more information about how Windows 2000 and WDM drivers use these new features of Microsoft operating systems, see the *Setup, Plug and Play, and Power Management Design Guide and Reference* in this DDK.

PC: Windows 9x Speicherbereiche

gemeinsam genutzter Systembereich	FFFFFFFh	1 GB
gemeinsam genutzt für Anwendungen	C0000000h	1 GB
privater Adreßbereich Anwendungen	80000000h	knapp 2 GB
ungenutzt	00400000h	4 MB
V86 Bereich	0010FFF0h 00000000h	1 MB inklusive "8086 A20 bug": real mode Bereich

- DOS-Bereich immer noch für Boot / Geräte (VGA) notwendig
- Kernel, Treiber, usw. im oberen 1 GB-Bereich

BIOS: Systemkomponenten



- Einstellung über PCI-Autokonfiguration bzw. die Treiber

zukünftige Entwicklungen!?

- Planarprozeß ist massiv parallel
 - Kosten fast unabhängig von der Anzahl einzelner Elemente
- => Moore's Law: exponentieller Anstieg des Integrationsgrades und damit exponentielles Wachstum von:
- CPU-Performance
 - Speicherkapazität (DRAM, Festplatten)
 - entsprechend komplexere Software

Moore's Law

- Planarprozeß ist massiv parallel
 - Kosten fast unabhängig von der Anzahl einzelner Elemente
- => Moore's Law: exponentieller Anstieg des Integrationsgrades
- mehr Funktionen bei gleichen Kosten (gleiche Chipfläche)
 - oder gleiche Funktion bei geringeren Kosten
 - rein wirtschaftlich bedingt
 - solange, bis Kapitalkosten für neue Technologie zu hoch

Verbesserungen durch: (relativer Anteil)

- feinere Lithographie (50%)
- verbesserte Transistoren / Strukturen (25%)
- bessere Rechnerarchitektur (25%)

Moore's Law: Lithographie, Hochintegration

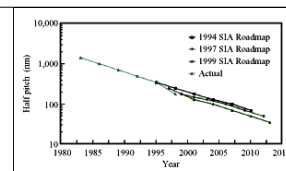


Figure 1
Historical and future trends of lithographic resolution capability. Here, half pitch is the minimum size of lithographic features on a chip. (SIA—Semiconductor Industry Association.)

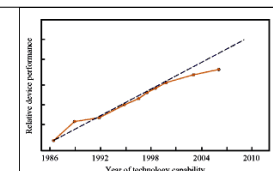


Figure 2
Comparison of performance for devices produced in successive technology generations vs. the year in which each technology generation first reached capability for volume production. Circles and the yellow curve represent historical and expected future behavior. The straight line represents an exponential growth rate as predicted from Moore's law. Circuit effects such as loading are not considered in this measurement of relative performance.

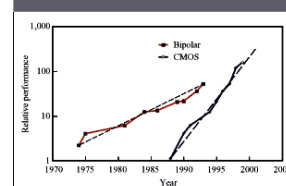


Figure 6
Historical and future server performance trends using bipolar and CMOS circuits. The straight lines represent the time-averaged exponential improvement in the performance of the technology.

- exponentielles Wachstum
- seit 1970, bis > 2015

CMOS vs. ECL:

- Hochintegration, langsame Xtors
- schnelle Xtors, Abwärmeproblem

[IBM JR&D 44-3, 2000]

Moore's Law: Transistor-Skalierung

[Intel µP-Forum 99]

As the technology scales...

Width = $W = 0.7$, Length = $L = 0.7$, $t_{ox} = 0.7$

- 1. Dimensions reduce 30%, this is good**
 $Area_{Cap} = C_s = \frac{0.7 \times 0.7}{0.7} = 0.7$
 $Fringing_{Cap} = C_f = 0.7$
 $Total_{Cap} \Rightarrow C = 0.7$
- 2. Capacitance on a node reduces by 30%, this is good**
 $Die_{Area} = X \times Y = 0.7 \times 0.7 = 0.7^2$
- 3. Transistor density (integration) doubles, this is good**
 $\frac{Cap}{Area} = \frac{0.7}{0.7 \times 0.7} = \frac{1}{0.7}$
- 4. Capacitance per unit area increases 43%, this is not good**

Moore's Law: bessere Transistoren

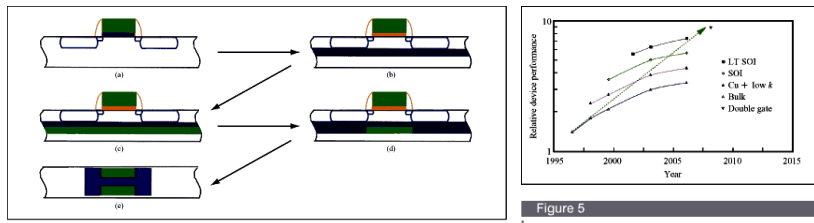


Figure 4
Plausible evolution in transistor structure toward a more symmetric structure that results in better control of the fields in the gate region, regulating device condition. The FETs pictured are: (a) bulk Si, (b) silicon-on-insulator (SOI), (c) ground plane, counter-electrode, (d) vertical double gate, and (e) fully symmetric double gate.

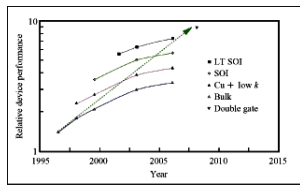


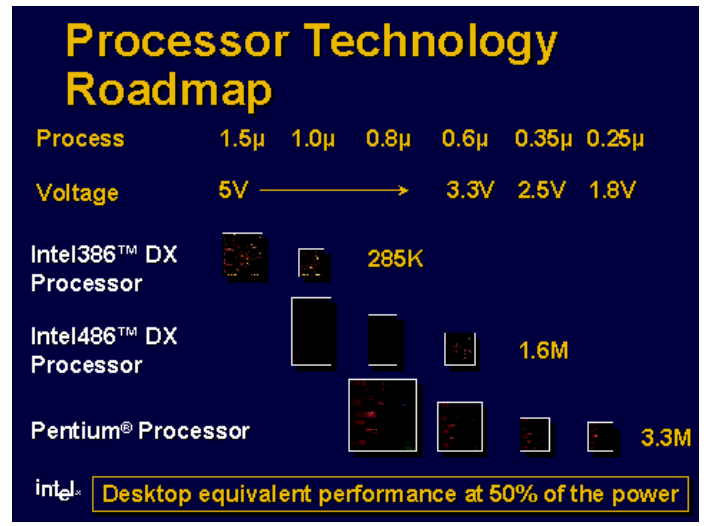
Figure 5
Application of new structures and materials to continue the trend (dashed line) of exponential improvement in device performance vs. time. The transistor structures indicated are bulk Si and double gate. The labels SOI and LT SOI refer to the use of silicon-on-insulator at room temperature or low temperature, while Cl+ low k refers to the use of copper metal interconnections with low-dielectric-constant insulators.

Evolution des MOS-Transistors:

- "bulk" Transistor direkt im Si-Substrat
- "silicon on insulator", SOI dünnes Substrat, darunter Isolationschicht
- "ground plane" Substrat, Isolator, leitende Schicht: Spiegelladungen
- "double gate" optimale, symmetrische Anordnung

[IBM JR&D 44-3, 2000]

x86: Halbleitertechnologien ...



[intel IDF98]

Moore's Law: bessere Verdrahtung

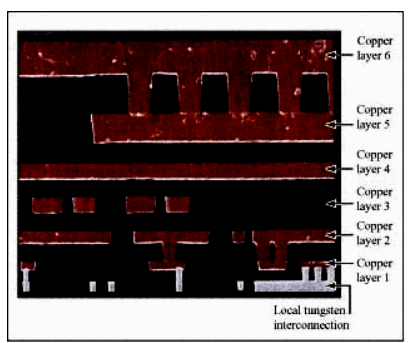


Figure 1
Cross-sectional scanning electron micrograph showing typical CMOS 7S interconnections with tungsten local interconnections and six levels of copper wiring. From [16], reproduced with permission of The Electrochemical Society, Inc.

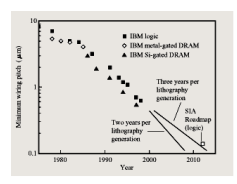


Figure 2
Minimum wire pitch used in IBM DRAM and CMOS logic technologies vs. year of introduction, and extrapolation of the current scaling trend into the future. Modified from [10], with permission of The Electrochemical Society, Inc.

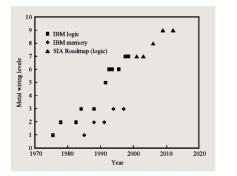
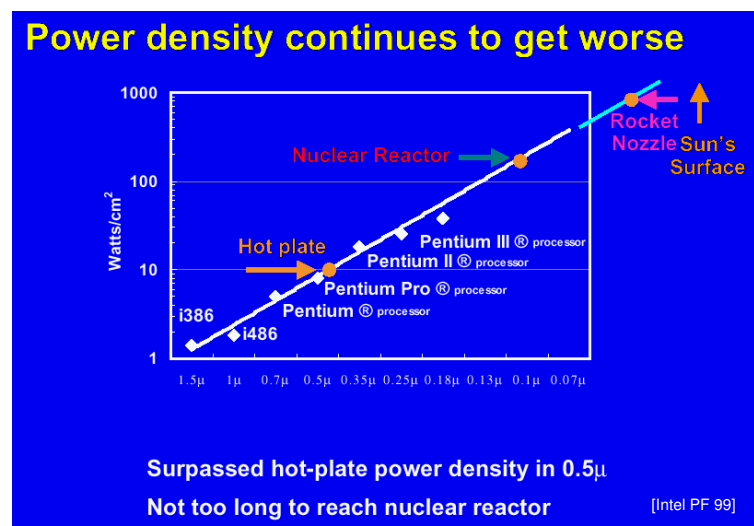


Figure 3
Number of wire levels used in IBM DRAM and CMOS logic technologies vs. year of introduction (includes tungsten local interconnections) and Semiconductor Industry Association (SIA) Roadmap values for future years. Modified from [10], with permission of The Electrochemical Society, Inc.

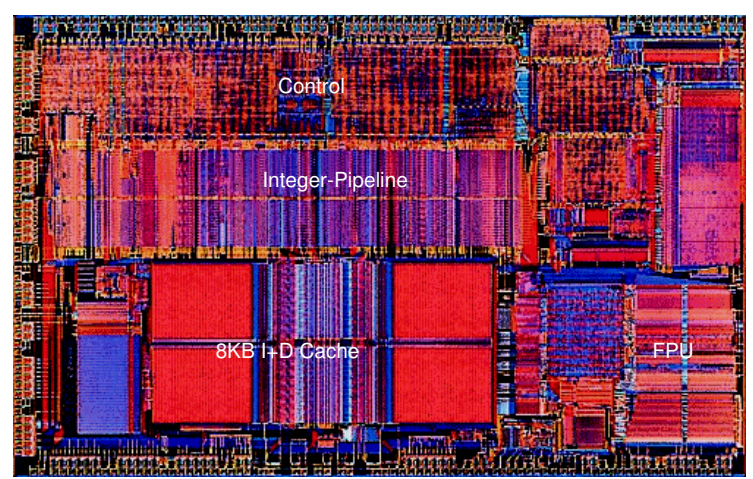
[IBM JR&D 44-3, 2000]

Moore's Law: Leistungsverbrauch



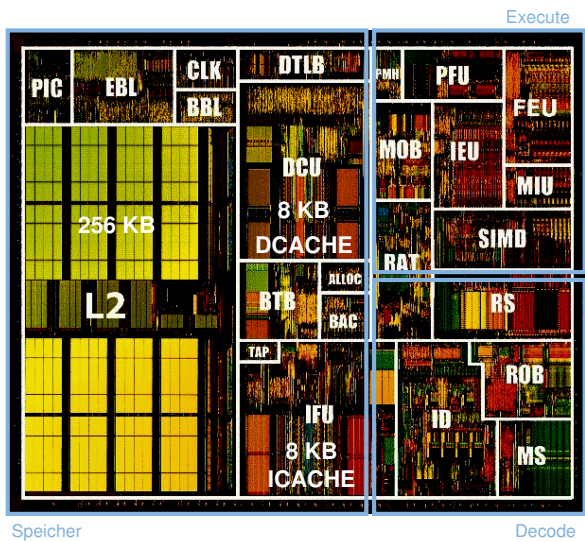
[Intel PF 99]

x86: Chiplayout 486DX



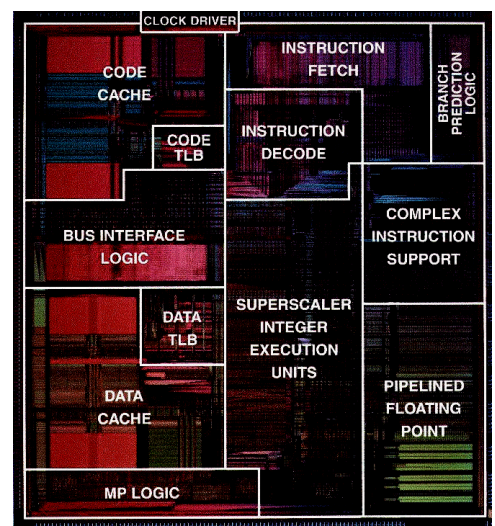
PC-Technologie | SS 2001 | 18.214

Pentium III: ChipLayout



PC-Technologie | SS 2001 | 18.214

x86: Chiplayout Pentium (P54C)

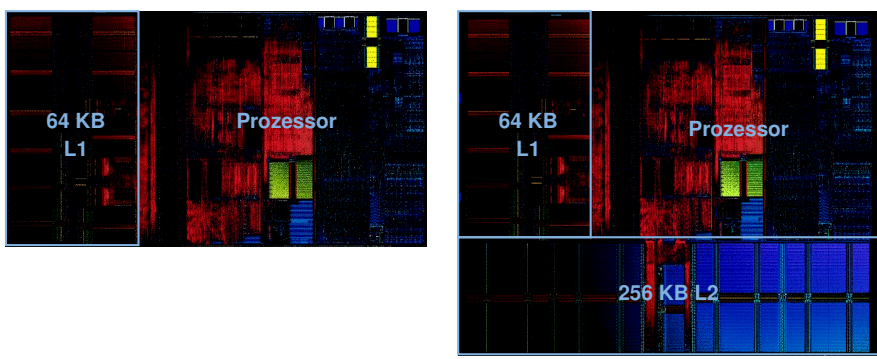


- ~ 40% Speicher
- ~ 60% Execute:
- ~ 15% FPU
- ~ 10% APIC/MP

[www.intel.com]

PC-Technologie | SS 2001 | 18.214

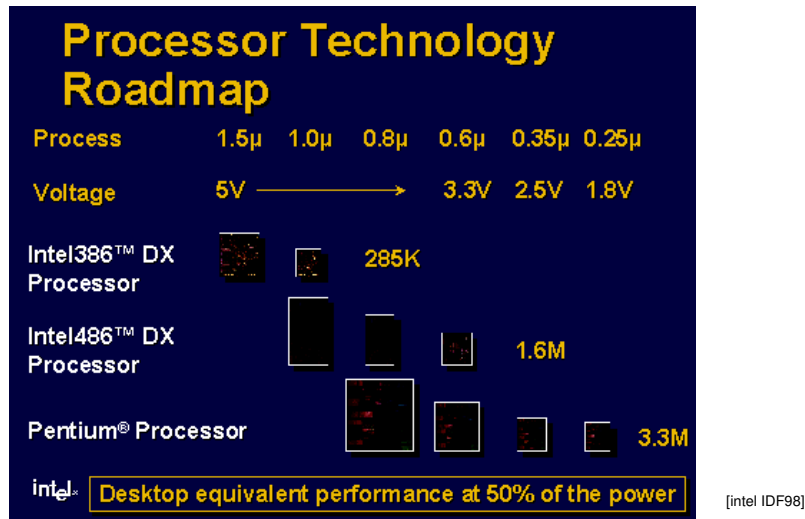
AMD K6: Layout K6-2 vs. K6-III



- gleicher Prozessorkern, 32K I\$, 32K D\$, 256K L2
- ca. 30% bzw. über 50% Chipfläche für Speicher

PC-Technologie | SS 2001 | 18.214

x86: Halbleitertechnologien ...



PC-Technologie | SS 2001 | 18.214

Literatur: Bücher

A.S.Tanenbaum	Computerarchitektur, 4. Auflage
J.L.Hennessy & D.A.Patterson	Computer Architecture, a Quantitative Approach, MKP 1996
H.-P. Messmer	PC-Hardwarebuch, 6. Aufl., Addison-Wesley 2000
S. Mueller	PC-Hardware Superbibel, Markt&Technik 1999
R. Hyde	http://webster.cs.ucr.edu/Page_asm/ArtofAssembly/pdf/AoAPDF.html
R. Duncan	MS-DOS für Fortgeschrittene, Vieweg 1987
S.P.Dandamudi	Introduction to Assembly Language Programming, Springer 1998
Intel	diverse Datenbücher (siehe developer.intel.com)
Commodore	PC-AT Service Manual (vollständige AT-Schaltpläne...)

PC-Technologie | SS 2001 | 18.214

Literatur: URLs

www.microsoft.com/hwdev
developer.intel.com

www.sandpile.org
bwrc.eecs.berkeley.edu/CIC
www.computerhistory.org
www.x86.org
www.tomshardware.com
www.pcodesguide.org
www.amd.com/swdev

www.usb.org
www.pcisig.org

PC-Technologie | SS 2001 | 18.214

Literatur: ausgewählte c't Artikel

A.S.Tanenbaum	Computerarchitektur, 4. Auflage
J.L.Hennessy & D.A.Patterson	Computer Architecture, a Quantitative Approach, MKP 1996
PCI-Bus, Interrupts	11/2000, 258ff
ACPI, Powermanagement	20/1998, 166ff
DVD Übersicht	20/1888, 101 ff, CSS: 08/2000, 221
Übersicht Intel/AMD Prozessoren	14/2000, 88ff
Speicher-Einmaleins	19/2000, 180ff
3D-Graphik	08/2000, 202ff, progressive Polygonmodelle: 16/1998, 166ff
USB, Firewire	02/1997, 284ff
Dateisysteme (FAT...)	06/2000, 116ff
LCD, Projektoren	12/2000, 170ff

PC-Technologie | SS 2001 | 18.214

x86 Prozessoren: Inhalt

Architektur der Intel x86-Familie:

- Historie 8008 -> Pentium III
- Register nur Übersicht
- Befehlssatz real / protected / virtual 8086 / ...
- Speichermodell
- Programmbeispiele

- RISC vs. CISC - Debatte
- Instruction Level Parallelism
- Aktuelle Implementation AMD Athlon

- Ausblick auf IA-64 und AMD x86-64

x86: Hardware-Evolution

- 8086, 80286, 80386: serielle Befehlsausführung
- 80486: Integer-Pipeline, interner 8KB-Cache
- Pentium: doppelte Pipeline, 8+8 KB Cache
- Pentium MMX: doppelte Pipeline, 16+16 KB, SIMD MMX

- K5, K6, 6x86, Athlon, superskalar, out-of-order, SIMD, ...
- Pentium Pro .. Pentium IV: 16 .. 64 KB L1, typ. 256 KB L2
- Duron: 64 KB L1, 64 KB non-exclusive L2

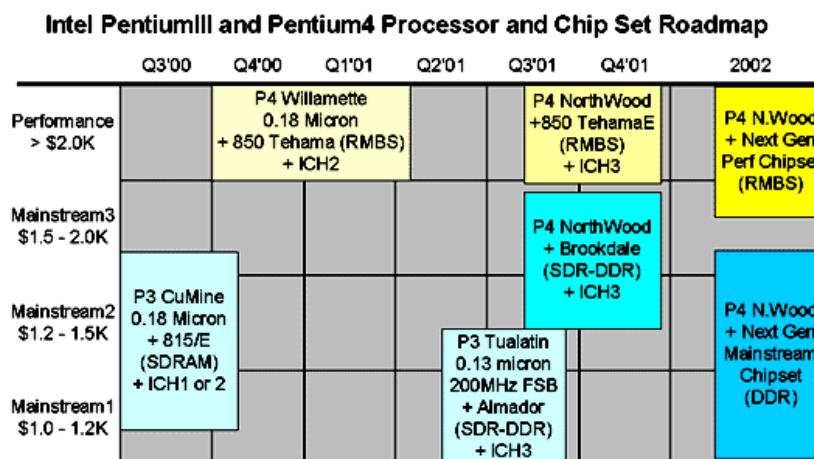
- (aufwendige) Befehlsdekodierung von x86 in "µOps"
- anschließend RISC-artige Rechenwerke/Pipeline
- kaum Nachteile gegenüber superskalaren RISC-Prozessoren

x86: Evolution ...

Intel Processor	Date of Product Introduction	Performance in MIPs ¹	Max. CPU Frequency at Introduction	No. of Transistors on the Die	Main CPU Register Size ²	Extern. Data Bus Size ²	Max. Extern. Addr. Space	Caches in CPU Package ³
8086	1978	0.8	8 MHz	29 K	16	16	1 MB	None
Intel 286	1982	2.7	12.5 MHz	134 K	16	16	16 MB	Note 3
Intel386™ DX	1985	6.0	20 MHz	275 K	32	32	4 GB	Note 3
Intel486™ DX	1989	20	25 MHz	1.2 M	32	32	4 GB	8KB L1
Pentium®	1993	100	60 MHz	3.1 M	32	64	4 GB	16KB L1
Pentium® Pro	1995	440	200 MHz	5.5 M	32	64	64 GB	16KB L1; 256KB or 512KB L2
Pentium II®	1997	466	266	7 M	32	64	64 GB	32KB L1; 256KB or 512KB L2
Pentium® III	1999	1000	500	8.2 M	32 GP 128 SIMD-FP	64	64 GB	32KB L1; 512KB L2

[Intel Pentium-III databook]

x86: Intel Roadmap Q3/00



x86: Pentium-Klasse

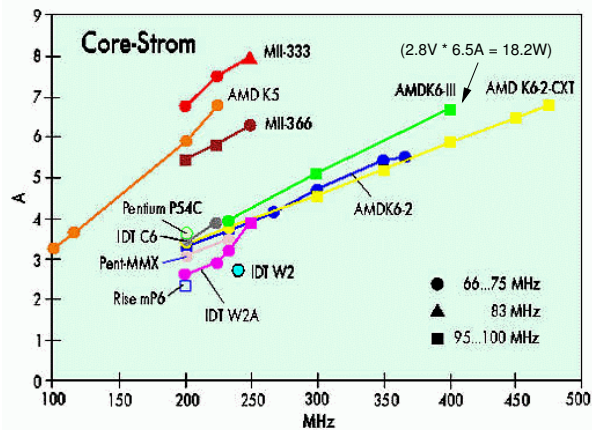
Moderne x86-CPU's im Vergleich					
Typ	Pentium	P6	K86	M1	Nx586
Hersteller	Intel	Intel	AMD	Cyrix	NexGen
Interner Takt	100 MHz	133 MHz	100 MHz	100 MHz	93 MHz
Daten-Cache	8 KByte	8 KByte	16 KByte	16 KByte	16 KByte
Befehls-Cache	8 KByte	8 KByte	8 KByte	unifrad	16 KByte
L2-Cache-Interface	-	ja	-	-	ja
L2-Cache	-	256 KByte	-	-	-
Dispatcher-Rate	2 Befehle	3 Befehle	2-3 Befehle	2 Befehle	2 Befehle
Parallele Einheiten	3 Einheiten	5 Einheiten	7 Einheiten	2 Einheiten	3 Einheiten
Out-of-Order	-	40 Befehle	16 Befehle	-	-
Renaming-Register	-	40 Register	16 Register	32 Register	-
IC-Prozeß	0,6 µ BICMOS	0,6 µ BICMOS	0,5 µ CMOS	0,65 µ CMOS	0,5 µ BICMOS
Metall-Layer	4	4	3	3	4
Logic Transistoren	2,4 Millionen	4,5 Millionen	2,4 Millionen	2,1 Millionen	2,4 Millionen
Transistoren f. L1	0,9 Millionen	1,0 Millionen	1,9 Millionen	0,9 Millionen	0,9 Millionen
Transistoren f. L2	-	15 Millionen	-	-	-
Alle Transistoren	3,3 Millionen	20,5 Mill.	4,3 Millionen	3,0 Millionen	3,3 Millionen
Fassungstyp	CPGA	CPGA	CPGA	CPGA	CPGA
Anzahl Pins	296 Pins	387 Pins	296 Pins	296 Pins	463 Pins
Die Size	163 mm ²	306 mm ² 202 mm ²	225 mm ²	394 mm ²	196 mm ²
Herstellungskosten	120 \$	350 \$ ¹	170 \$	340 \$	200 \$
Leistungsaufnahme	10 Watt	20 Watt	12 Watt	10 Watt	16 Watt
Verfügbarkeit	2QP4	3QP5	3QP5	3QP5	3QP4
SPECint92	113	200	130	120	110
SPECfp92	82	200	75	70	-

Quelle: Microprocessor Report
¹ Ohne L2-Cache
² Inclusive L2-Cache

- fünf Designs, vier Firmen
- alle superskalar
- dispatch 2-5X
- execute 3-7X
- Herstellungskosten (!)

[c't 05/95 122]

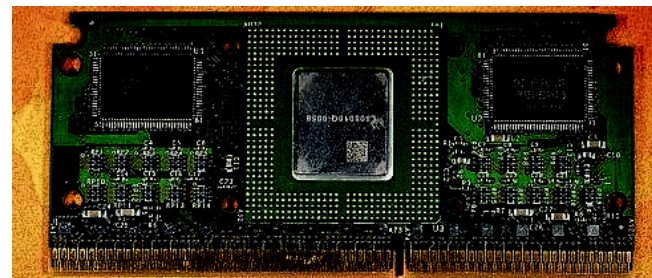
x86: Pentium-Klasse: Verlustleistung ...



- CMOS-Technologie: Leistung ~ (f/Hz) * (U/Volt) ²
- Kühltechnologie begrenzt auf < 50 W

[c't 10/99 176]

x86: Pentium-II/400 Package



Intel Verpackungstechnologie Q1/1999:

- CPU/FPU mit 16KB/16KB I+D Cache im Plastikgehäuse
- zwei externe SRAM-Chips für 512KB L2-Cache
- "Slot-1" Einsteckkarte (Busprotokoll patentiert)

x86: Performance 1999...

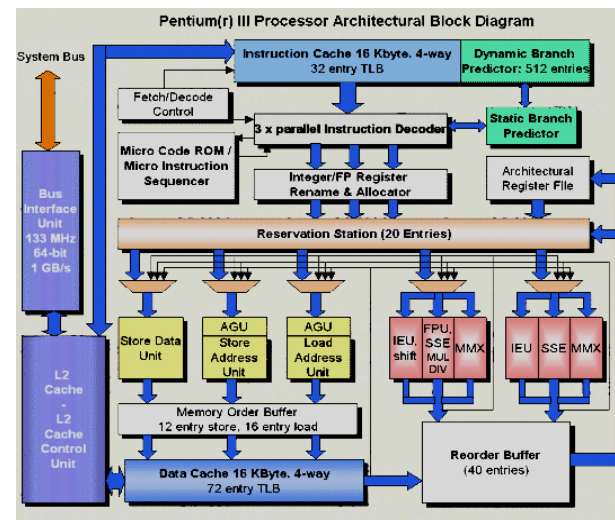
Leistungsfähigkeit der Prozessor-Familien				
Prozessor	BAPCo SYSmark 98	CPU3DMark99 ¹	FovRay 3.0 [s]	Unreal II [fps] ²
besser > < besser > < besser > < besser >				
bei 400 MHz, 64 MByte, Riva-TNT Grafik				
AMD K6-2	131	5588	57	25,2
AMD K6-III	151	6309	44	26,3
Intel Celeron	147	3681	42	27,7
Intel Pentium III	162	3903	44	30,6
bei 450 MHz, 64 MByte, Riva-TNT Grafik				
AMD K6-2	137	5999	52	24,7
AMD K6-III	164	7090	39	26,6
bei 500 MHz, 128 MByte, Riva-TNT2 Grafik				
Intel Celeron	178	4479	34	37,9
Intel Pentium III	192	7399	38	42,3
AMD Athlon	212	9343	27	44,1
bei 600 MHz, 128 MByte, Riva-TNT2 Grafik ³				
Intel Pentium III	221	9060	32	45,7
AMD Athlon	238	10015	22	47,3

¹ Futuremark's 3DMark99 Max; davon der CPU-Test, der von der Grafikkarte weitgehend unabhängig ist.
² Unreal 2.20, 800 x 600 Punkte, 16 Bit Farbtiefe
³ 700MHz-Werte stehen auf Seite 132

- Performance ~ Taktfrequenz, Architekturdifferenzen irrelevant (10%)
- K6-2 ohne L2-Cache, Celeron ohne ISSE/3Dnow!

[c't 10/99 176]

Pentium III



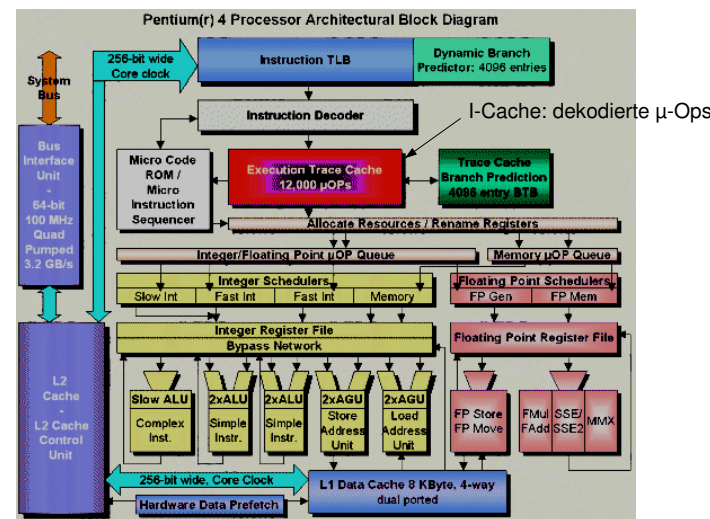
x86: Performance 2000...

Leistungsdaten aktueller AMD- und Intel-Systeme									
Prozessor	FSB [MHz]	Speicher	Board	Windows 98 SE - BAPCo SYSmark2000			FovRay 3.1g		Linux-Kernel
				SYSmark	Internet Content Creation	Office Productivity	PPS	sec	
besser > < besser > < besser > < besser > < besser >									
Fliegengewicht									
AMD K6-2/550	100	PC100-222	P5A	78	69	85	240	246	
Intel PIII 450 MHz	100	PC100-222	P3B-F	93	86	98	265	252	
AMD K6-III/450	100	PC100-222	P5A	88	75	98	257	212	
Intel Celeron 500	66	PC66-222	P3B-F	94	89	97	313	245	
Mittelgewicht									
Intel FC-PGA-Celeron 600	66	PC66-222	P3B-F	112	114	111	385	206	
Intel Pentium III 600 (Katmai)	100	PC100-222	P3B-F	124	124	124	353	202	
Intel FC-PGA-Celeron 700	66	PC100-222	CUV4X	123	126	120	433	184	
AMD Athlon-600	100	PC133-333	K7V	129	128	130	468	173	
AMD Duron-650	100	PC133-333	KT133	132	134	131	515	174	
AMD Duron-700	100	PC133-333	KT133	139	141	137	556	166	
Schwergewicht									
Intel Pentium III 800	133	PC133-333	D1184	167	167	167	556	122	
AMD Athlon-800	100	PC133-222	K7V	155	159	152	614	138	
Intel Pentium III 1000	133	PC133-333	CUV4X	185	189	182	698	102	
AMD Athlon-1000 (Thunderbird)	100	PC133-333	K7V	186	187	186	800	103	
Intel Pentium III 1000 (Rambus)	133	PC800-45	VC820	197	198	197	698	101	

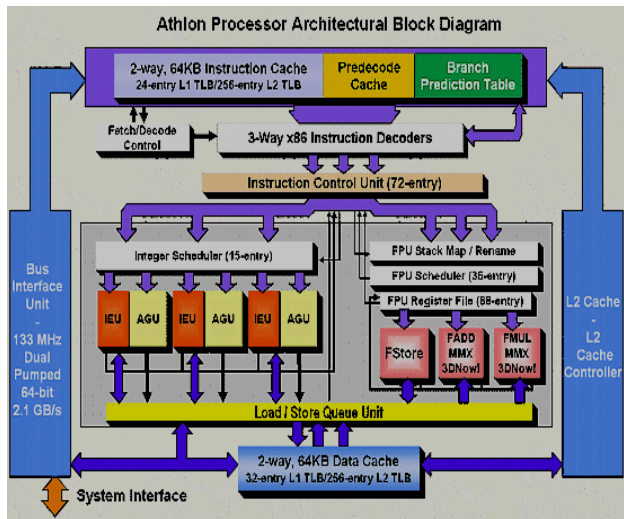
- alle Prozessoren mit integriertem L2-Cache (außer K6-2 und Athlon)
- Performance weitgehend proportional zum Takt
- keine signifikanten Vorteile für Intel oder AMD

[c't 14/00 098]

Pentium IV

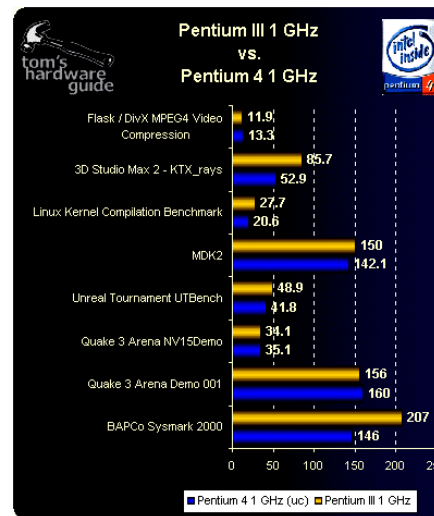


Athlon (Thunderbird)



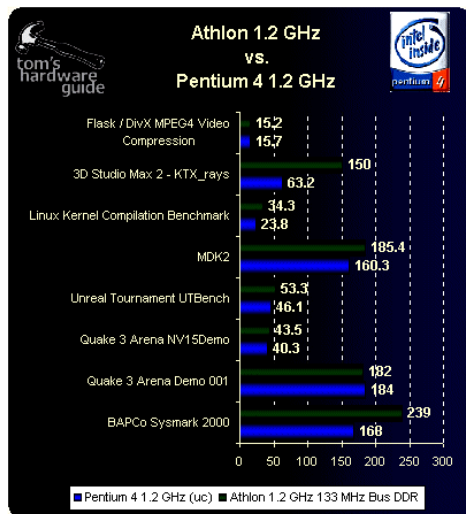
PC Technologie | SS 2001 | 18.214

Benchmarks: Pentium IV vs. Pentium III



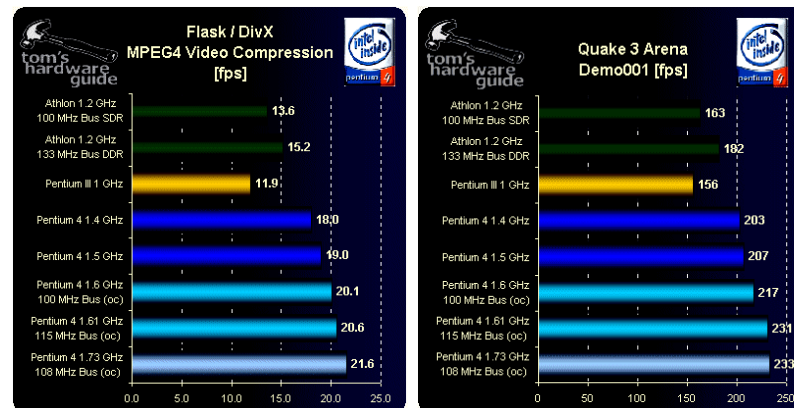
PC Technologie | SS 2001 | 18.214

Benchmarks: Pentium IV vs. Athlon



PC Technologie | SS 2001 | 18.214

Benchmarks: DivX / Quake



PC Technologie | SS 2001 | 18.214

x86: Probleme der x86-Architektur

"Insgesamt betrachtet, läßt sich die Lage der IA-32 mit dem Zustand der Himmelsmechanik kurz vor Kopernikus vergleichen. Die damalige Theorie, die Erde stünde fest verankert und bewegungslos im Raum, während die Planeten in Epizyklen um sie kreisen, beherrschte die Astronomie. Als jedoch die Beobachtungen immer besser wurden, kamen immer mehr Epizyklen dazu, bis das ganze Modell wegen seiner internen Komplexität in sich zusammenstürzte.

Intel befindet sich heute in einer ähnlichen Klemme..."
 [Tanenbaum 99]

Zukunft der x86-Architektur?!

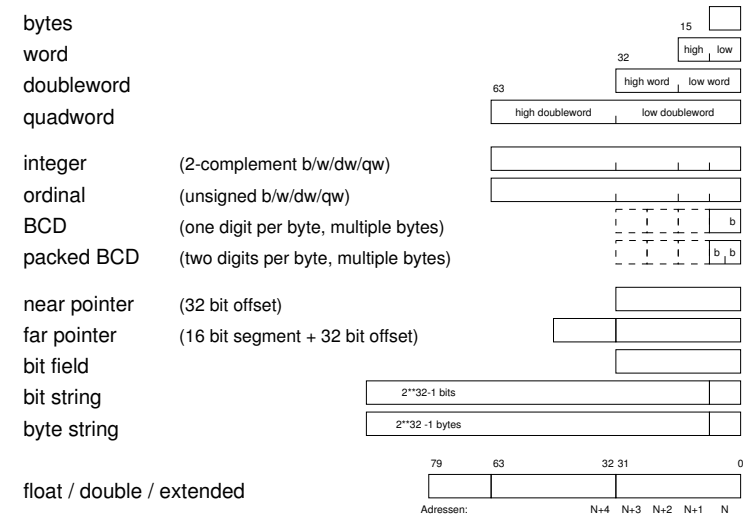
- => noch eine Erweiterung: AMD x86-64 Architektur
- => sauberer Neubeginn: Intel IA-64 Itanium

x86: Befehlssatz

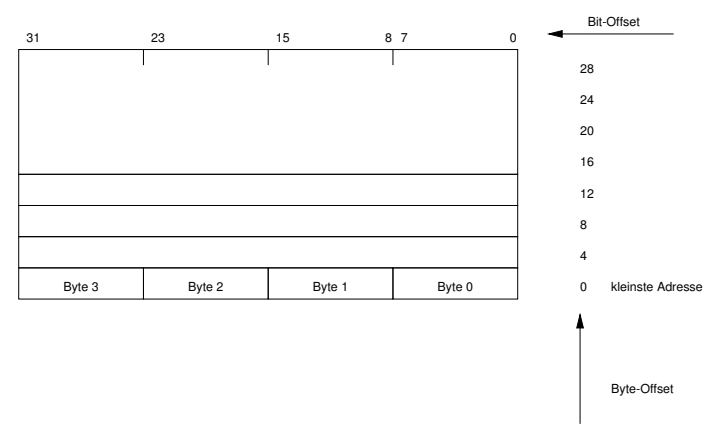
- Datenzugriff
 mov, xchg
 push, pusha, pop, popa
- Stack-Befehle
 cwd, cdq, cbw (byte->word), movsx, . . .
- Typumwandlung
 add, adc, inc, sub, sbb, dec, cmp, neg, . . .
- Binärarithmetik
 mul, imul, div, idiv,
- Dezimalarithmetik
 packed / unpacked BCD: daa, das, aaa, aas, . . .
- Logikoperationen
 and, or, xor, not, sal, shr, shr, . . .
- Sprungbefehle
 jmp, call, ret, int, iret, loop, loopne, . . .
- String-Operationen
 movs, cmps, scas, load, stos, . . .
- "high-level"
 enter (create stack frame), . . .
- diverser
 lahf (load AH from flags), . . .
- Segment-Register
 far call, far ret, lds (load data pointer)

=> CISC zusätzlich diverse Ausnahmen/Spezialfälle

x86: Datentypen: CISC . . .



x86: Byteorder



- "little endian": LSB eines Wortes bei der kleinsten Adresse

x86: Byteorder

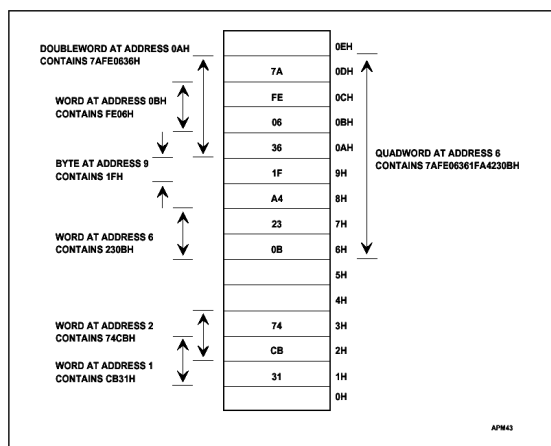


Figure 3-3. Bytes, Words, Doublewords and Quadwords in Memory

- Speicher ist voll byte-adressierbar

x86: Befehlsformate: CISC...

außergewöhnlich komplexes Befehlsformat:

- 1) prefix (repeat / segment override / etc.)
- 2) opcode (eigentlicher Befehl)
- 3) register specifier (Ziel / Quellregister)
- 4) address mode specifier (diverse Varianten)
- 5) scale-index-base (Speicheradressierung)
- 6) displacement (Offset)
- 7) immediate operand

- ausser dem Opcode alle Bestandteile optional
- unterschiedliche Länge der Befehle, von 1 .. 37 Byte

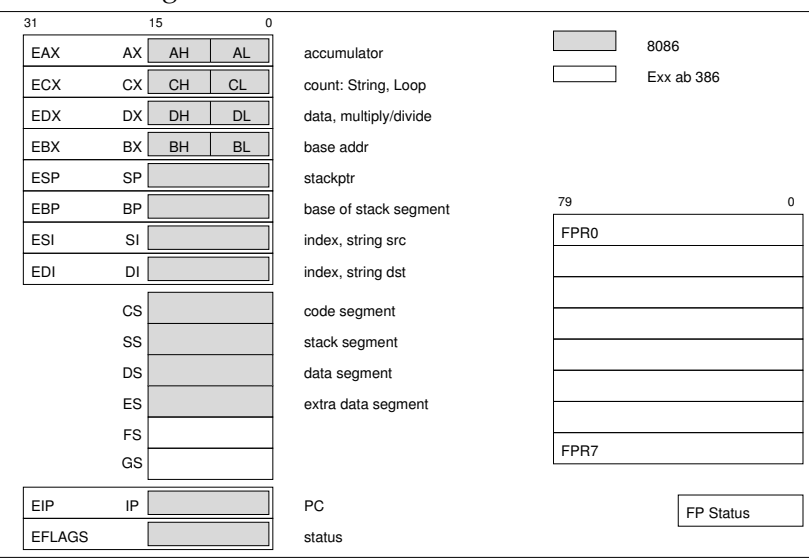
=> extrem aufwendige Dekodierung

x86: Modifier

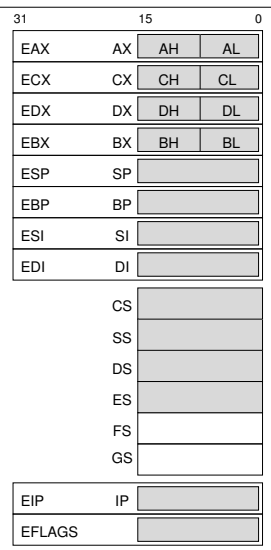
alle Befehle können mit "Modifiern" ergänzt werden:

- segment override Addr. aus angewähltem Segmentregister
- address size Umschaltung 16/32-bit
- operand size Umschaltung 16/32-bit
- repeat für Stringoperationen
Operation auf allen Elementen ausführen
- lock Speicherschutz für Multiprozessoren

x86: Register

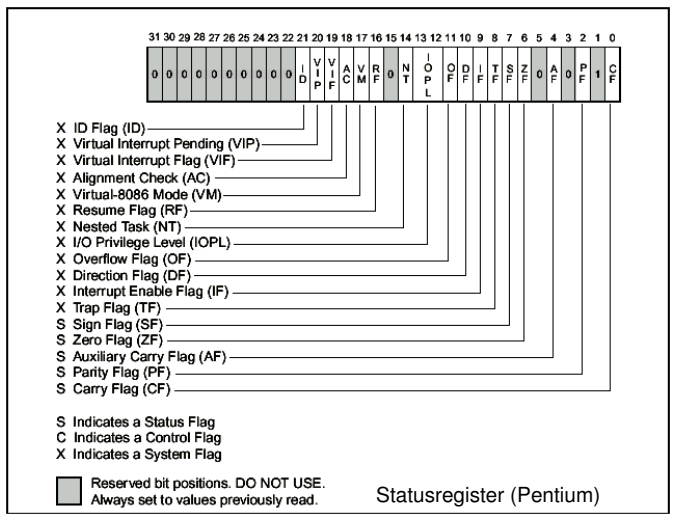


x86: Register



- sehr wenig Register
- alle Register haben Spezialaufgaben
- aber EAX .. EDI auch als GP Register
- viele Speicherzugriffe
- komplexe Segmentadressierung
- FP-Register als Stack organisiert
- schwer optimierbar

x86: EFLAGS Register



Leerseite

Leerseite

x86: CISC: Vergleichsbefehle

Table 4-3. Conditional Jump Instructions

Mnemonic	Flag States	Description
Unsigned Conditional Jumps		
JNBE	(CF or ZF)=0	Above/not below nor equal
JAE/JNB	CF=0	Above or equal/not below
JB/JNAE	CF=1	Below/not above nor equal
JBE/JNA	(CF or ZF)=1	Below or equal/not above
JC	CF=1	Carry
JE/JZ	ZF=1	Equal/zero
JNC	CF=0	Not carry
JNE/JNZ	ZF=0	Not equal/not zero
JNP/JPO	PF=0	Not parity/parity odd
JP/JPE	PF=1	Parity/parity even
Signed Conditional Jumps		
JG/JNLE	((SF xor OF) or ZF) =0	Greater/not less nor equal
JGE/JNL	(SF xor OF)=0	Greater or equal/not less
JL/JNGE	(SF xor OF)=1	Less/not greater nor equal
JLE/JNG	((SF xor OF) or ZF)=1	Less or equal/not greater
JNO	OF=0	Not overflow
JNS	SF=0	Not sign (non-negative)
JO	OF=1	Overflow
JS	SF=1	Sign (negative)

PC-Technologie | SS 2001 | 18.214

x86: CISC: "enter" instruction

The ENTER instruction can be used in two ways: nested and non-nested. If the lexical level is 0, the non-nested form is used. The non-nested form pushes the contents of the EBP register on the stack, copies the contents of the ESP register into the EBP register, and subtracts the first operand from the contents of the ESP register to allocate dynamic storage. The non-nested form differs from the nested form in that no stack frame pointers are copied. The nested form of the ENTER instruction occurs when the second parameter (lexical level) is not zero.

The following pseudo code shows the formal definition of the ENTER instruction. STORAGE is the number of bytes of dynamic storage to allocate for local variables, and LEVEL is the lexical nesting level.

```

PUSH EBP;
FRAME_PTR ← ESP;
IF LEVEL > 0
  THEN
    DO (LEVEL - 1) times
      EBP ← EBP - 4;
      PUSH Pointer(EBP); (* doubleword pointed to by EBP *)
    OD;
  PUSH FRAME_PTR;
FI;
EBP ← FRAME_PTR;
ESP ← ESP - STORAGE;

```

The main procedure (in which all other procedures are nested) operates at the highest lexical level, level 1. The first procedure it calls operates at the next deeper lexical level, level 2. A level 2 procedure can access the variables of the main program, which are at fixed locations specified by the compiler. In the case of level 1, the ENTER instruction allocates only the requested dynamic storage on the stack because there is no previous display to copy.

- volle Stackverwaltung für geschachtelte Funktionsaufrufe :-)

PC-Technologie | SS 2001 | 18.214

x86: Assembler-Beispiel

```

addr opcode assembler c quellcode
-----
                                .file "hello.c"
                                .text
0000 48656C6C .string "Hello x86!\n"
                                6F207838
                                36210A00
                                .text
                                print:
0000 55          pushl %ebp | void print( char* s ) {
0001 89E5        movl %esp,%ebp
0003 53          pushl %ebx
0004 8B5D08      movl 8(%ebp),%ebx
0007 803B00      cmpb $0,(%ebx) | while( *s != 0 ) {
000a 7418        je .L18
                                .align 4
                                .L19:
000c A100000000 movl stdout,%eax | putc( *s, stdout );
0011 50          pushl %eax
0012 0FBE03      movsbl (%ebx),%eax
0015 50          pushl %eax
0016 E8FCFFFF   call _IO_putc
                                FF
001b 43          incl %ebx | s++;
001c 83C408      addl $8,%esp | }
001f 803B00      cmpb $0,(%ebx)
0022 75E8        jne .L19
                                .L18:
0024 8B5DFC      movl -4(%ebp),%ebx | }
0027 89EC        movl %ebp,%esp
0029 5D          popl %ebp
002a C3          ret

```

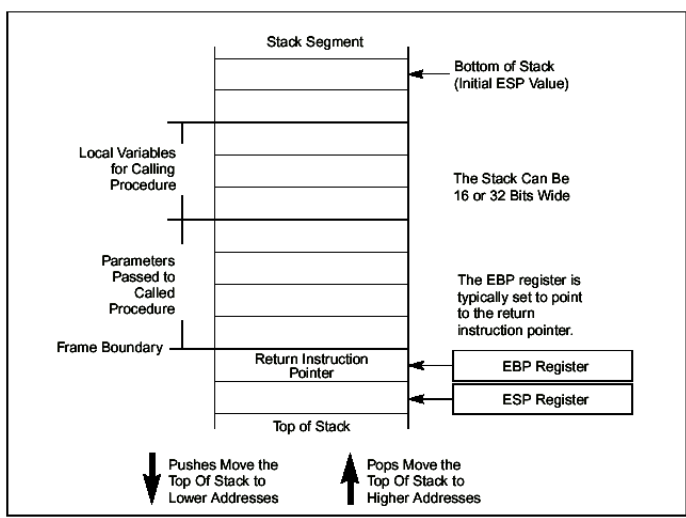
x86: Assembler-Beispiel (2)

```

addr opcode assembler c quellcode
-----
                                .Lf1:
                                .Lscope0:
002b 908D7426   .align 16
                                00
                                main:
0030 55          pushl %ebp | int main( int argc, char** argv ) {
0031 89E5        movl %esp,%ebp
0033 53          pushl %ebx
                                0034 BB00000000 movl $.LC0,%ebx | print( "Hello x86!\n" );
                                0039 803D0000      cmpb $0,.LC0
                                000000
0040 741A        je .L26
0042 89F6        .align 4
                                .L24:
0044 A100000000 movl stdout,%eax
0049 50          pushl %eax
004a 0FBE03      movsbl (%ebx),%eax
004d 50          pushl %eax
004e E8FCFFFFFF   call _IO_putc
0053 43          incl %ebx
0054 83C408      addl $8,%esp
0057 803B00      cmpb $0,(%ebx)
005a 75E8        jne .L24
                                .L26:
005c 31C0        xorl %eax,%eax | return 0;
005e 8B5DFC      movl -4(%ebp),%ebx | }
0061 89EC        movl %ebp,%esp
0063 5D          popl %ebp
0064 C3          ret

```

x86: Stack-Layout



x86: Stack-Verwaltung bei Interrupts

If no stack switch occurs, the processor does the following when calling an interrupt or exception handler (refer to Figure 4-5):

1. Pushes the current contents of the EFLAGS, CS, and EIP registers (in that order) on the stack.
2. Pushes an error code (if appropriate) on the stack.
3. Loads the segment selector for the new code segment and the new instruction pointer (from the interrupt gate or trap gate) into the CS and EIP registers, respectively.
4. If the call is through an interrupt gate, clears the IF flag in the EFLAGS register.
5. Begins execution of the handler procedure at the new privilege level.

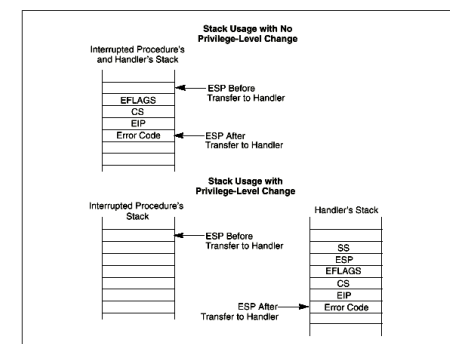
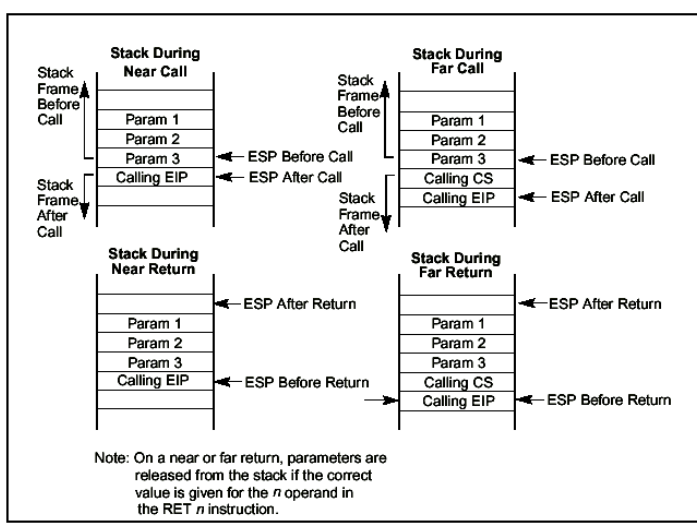


Figure 4-5. Stack Usage on Transfers to Interrupt and Exception Handling Routines

x86: Stack: near/far Calls



x86: Adressierungsarten

- displacement
- base
- base + displacement
- (index*scale) + displacement
- base + index + displacement
- base + (index*scale) + displacement
- immediate

$$\begin{pmatrix} \text{CS} \\ \text{SS} \\ \text{DS} \\ \text{ES} \\ \text{FS} \\ \text{GS} \end{pmatrix} + \begin{pmatrix} \text{EAX} \\ \text{ECX} \\ \text{EDX} \\ \dots \\ \text{ESP} \\ \text{EDI} \end{pmatrix} + \begin{pmatrix} 1 \\ 2 \\ 4 \\ 8 \end{pmatrix} + \begin{pmatrix} \text{Displacement} \\ \text{none} \\ \text{8-bit offset} \\ \text{32-bit offset} \end{pmatrix}$$

x86: Modi

real mode: 8086+

- segmentierte Adressierung, kein Speicherschutz
- direkte Hardwarezugriffe, z.B. Interrupt-Vektoren

protected mode: 80286+

- Segmentdeskriptoren, Speicherschutz: Ring 0 .. 3
- Hardwareunterstützung für Multitasking, Call Gates, ...

enhanced mode: 80386+

- 32-bit Register und Operanden
- Segmentierung und Paging, MMU, ...

virtual 8086 mode: 80386+

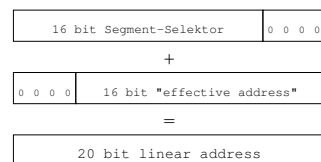
- Adressierung / Zugriff wie 8086, aber anschliessend Paging

PC-Technologie | SS 2001 | 18.214

x86: real mode

real mode := Speicherkonzept des 8086 Prozessors

- 20-bit Adressen, aber nur 16-bit Register



Adressüberlauf möglich, wenn Segmentselektor zu groß:

- nur 20 Adressleitungen am 8086: wrap around
- ab 80286 Problem mit Adressen 100000h - 10FFFEh (A20 Gate)

PC-Technologie | SS 2001 | 18.214

x86: protected mode

protected mode := Speicherzugriff mit Gültigkeitsprüfung

- Segment-Adressierung (ab 286)
 - vier (sechs) Segmentregister
 - Adresse = Segment-Basisadresse + Offset im Segment
 - Überprüfung von Segmentgrenzen und -rechten
-
- Paging (ab 386)
 - extrem flexibles Konzept für virtuellen Speicher
 - Paging ist mit Segmentierung kombinierbar

PC-Technologie | SS 2001 | 18.214

x86: Segment-Adressierung

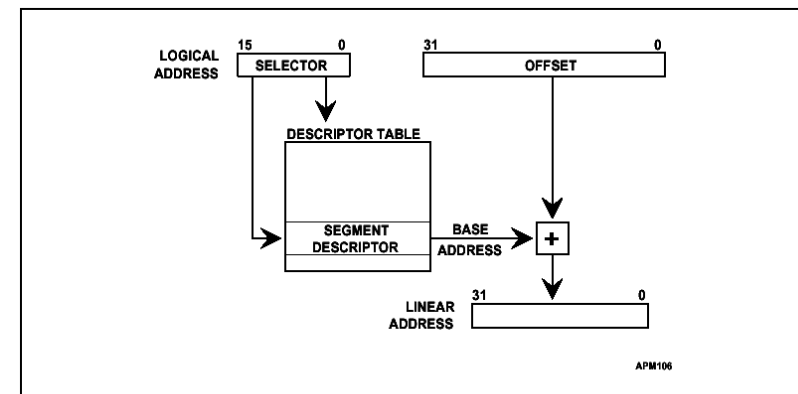


Figure 11-5. Segment Translation

- "far pointer": 16 bit Segment-Selektor, 32 bit Offset
- Deskriptortabelle enthält Basisadresse und Zugriffsrechte

PC-Technologie | SS 2001 | 18.214

x86: Segment-Adressierung

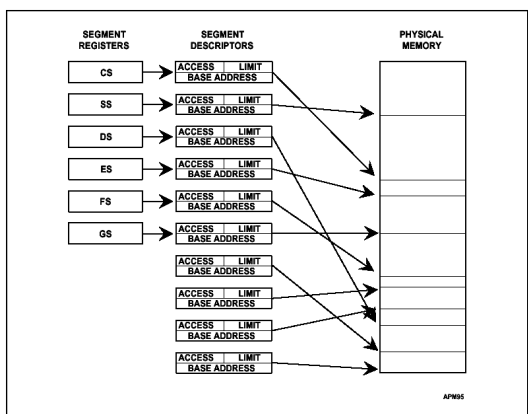


Figure 11-3. Multisegment Model

- sechs Segmentregister, bis zu 16383 Segmente a 4 GByte
- individuell einstellbare Zugriffsrechte pro Segment

x86: "flat addressing"

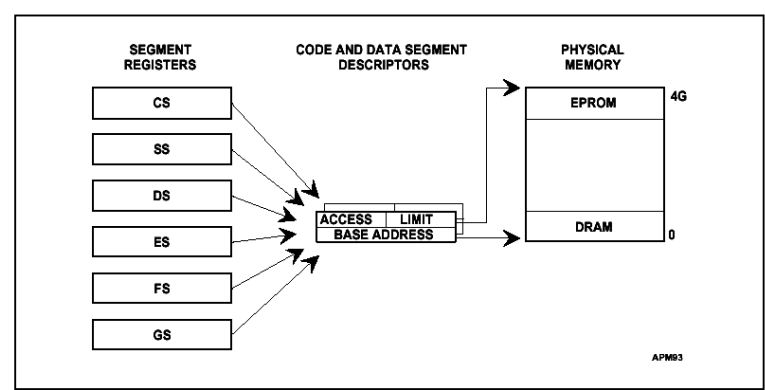


Figure 11-1. Flat Model

- alle Segmentregister enthalten dieselben Werte
- flacher 32-bit Adressraum (mit Range-Checks)

x86: Segmentdeskriptor

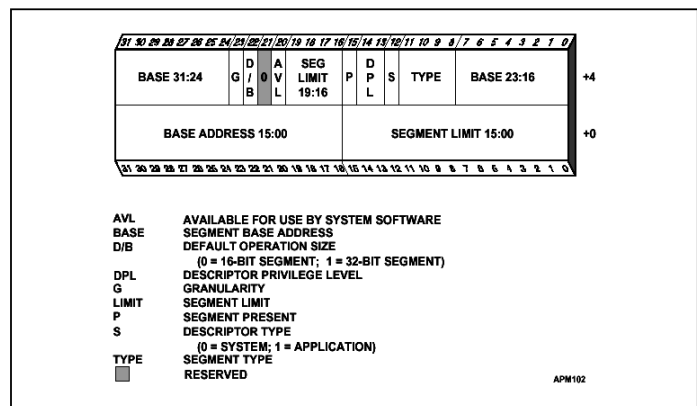


Figure 11-8. Segment Descriptors

- 32 bit Basisadresse und 20 bit Segmentlänge
- diverse Flags für Zugriffsrechte usw.

x86: "protected flat addressing"

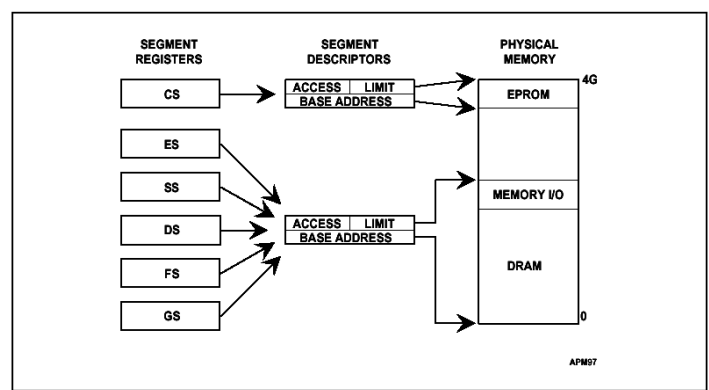


Figure 11-2. Protected Flat Model

- separates Code-Segment (evtl. mit Wraparound)
- kein Zusatzaufwand für Überprüfung von Speicher/Stackzugriffen

x86: 32 vs. 16 bit Code ...

8086 aufwärts: "16-bit Code":

- segmentierte Adressierung (real- oder protected mode)
- 16-bit Arithmetik, kein Zugriff auf die erweiterten Register
- Segmentgröße maximal 64 KB
- daher Probleme mit Code/Daten größer 64 KB, insbesondere Arrays
- ständiges Neuladen der Segmentregister

ab 80386: "32-bit Code":

- Zugriff auf die vollen 32-bit EAX .. EDI Register
- EAX .. EDI auch als 8 Universalregister nutzbar
- 32-bit Arithmetik

PC-Technologie | SS 2001 | 18.214

x86: 32 vs 16 bit Code: Addition

```

; 32 bit add in 16 bit Code
mov ax, word ptr a      ; niederwertiger Teil von a
mov dx, word ptr a+2    ; höherwertiger Teil von a
add ax, word ptr b      ; addiere b, setzt Carry
adc dx, word ptr b+2    ; addiere mit Carry,
mov word ptr c, ax      ; niederwertiger Teil der Summe
mov word ptr c+2, dx    ; höherwertiger Teil der Summe

```

```

mov eax, a              ; 32 bit add in 32 bit Code
add eax, b
mov c, eax

```

PC-Technologie | SS 2001 | 18.214

x86: 32 vs. 16 bit Code: Arrayzugriff

```

LONG countBlack( BYTE __huge *lpBits, LONG nbits ) {
    LONG result = 0; LONG i;
    for( i=0; i < nbits; ++i ) {
        if (!lpBits[i]) ++result;
    }
    return result;
}

```

; 32 bit Code für lpBits[i]:
 mov eax, DWORD PTR i
 mov ecx, DWORD PTR lpBits
 xor edx, edx
 mov dl, BYTE PTR [eax+ecx]

; 16 bit Code für lpBits[i]
 mov ax, WORD PTR i
 mov dx, WORD PTR i+2
 mov cx, WORD PTR lpBits
 mov bx, WORD PTR lpBits+2
 add ax, cx
 adc dx, 0
 mov cx, OFFSET __AHSHIFT ; Wert 3: (i/65536)*8
 shl dx, cl ; Siehe Oney, Win95 Prog., S.99
 add dx, bx
 mov bx, ax
 mov es, dx ; lädt Segmentregister ...
 mov al, BYTE PTR es:[bx]

PC-Technologie | SS 2001 | 18.214

Leerseite

PC-Technologie

x86: Segmente und Pages

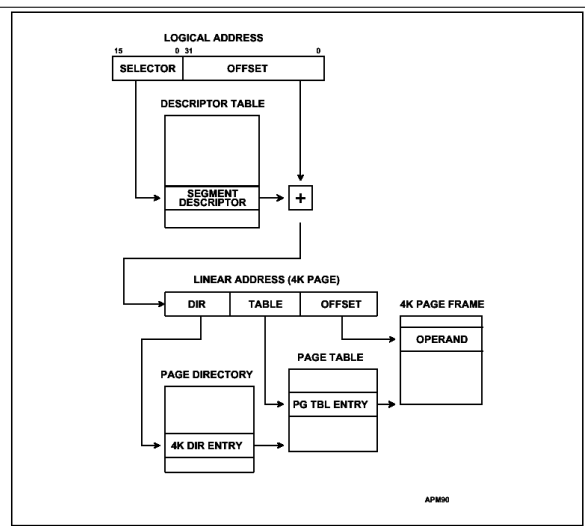


Figure 11-16. Combined Segment and Page Address Translation

x86: Page Tables

11.3.3. Page Tables

A page table is an array of 32-bit entries. A page table is itself a page, and contains 4096 bytes of data or at most 1K 32-bit entries. Four kilobyte pages, including page directories and page tables, are aligned to 4K-byte boundaries. Two levels of tables are used to address a page of memory. At the highest level is a page directory. A page directory holds up to 1K entries that address page tables of the second level. A page table of the second level addresses up to 1K pages in physical memory. All the tables addressed by one page directory, therefore, can address 1M (2^{20}) four-Kbyte pages. If each page contains 4K (2^{12}) bytes, the tables of one page directory can span a linear address space of four gigabytes ($2^{20} \times 2^{12} = 2^{32}$). For information on support of page sizes larger than 4K, see Appendix H.

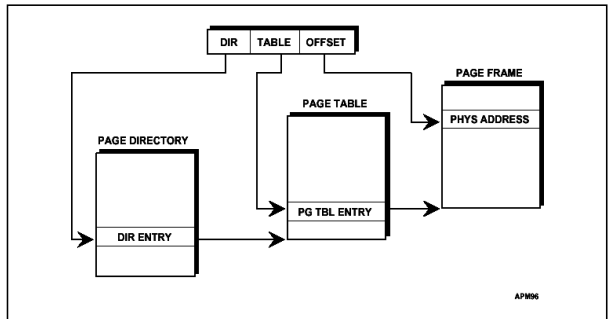
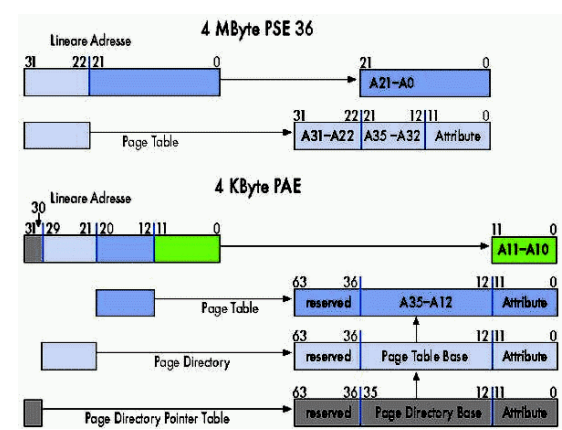


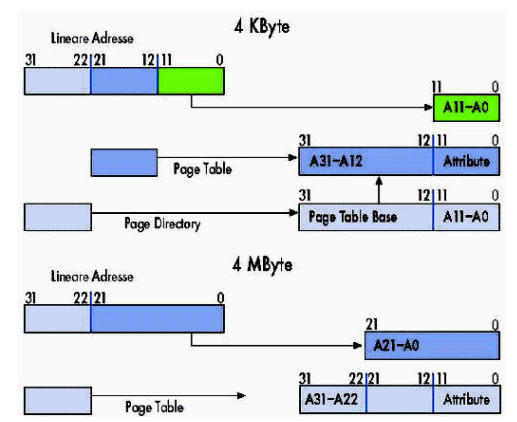
Figure 11-13. Page Translation

x86: erweiterter Adressraum



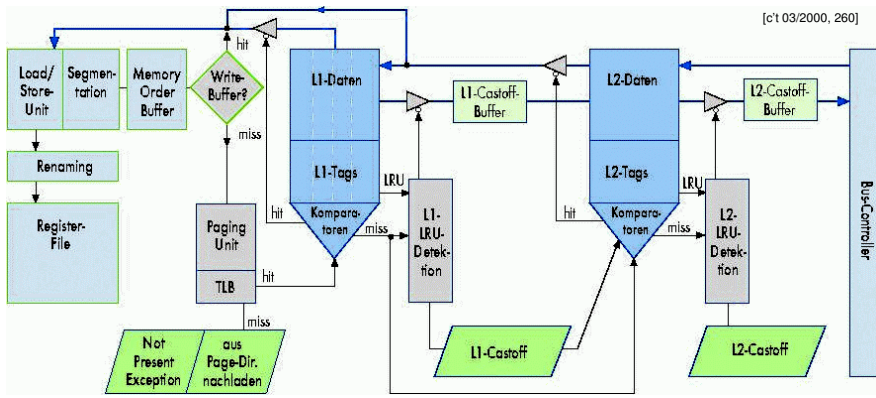
- Erweiterung auf 64 bit physikalische Adressen
- weiterhin < 4 GB pro Task

x86: Pagesizes: 4 KB vs. 4 MB



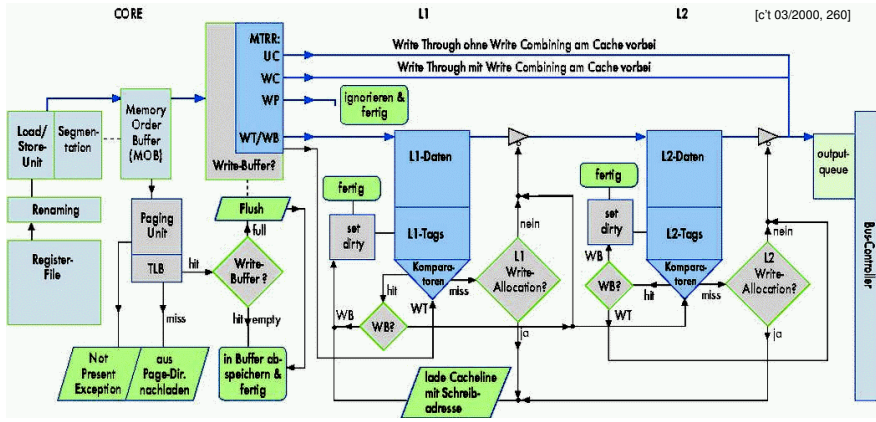
- kleine Pages erlauben feine Granularität
- aber evtl. viele TLB-Misses
- große Pages günstig für Betriebssystem / Framebuffer

x86: Pentium II Lesezugriff...



- Cachezugriffe: L1 typ. 1..2 Takte, L2 typ. 2..10 Takte
- Speicherzugriffe: ca. 100 Takte

x86: Pentium II Schreibzugriff...



- MemoryTypeRangeRegister: schnelle I/O, z.B. Graphikkarte
- weitere Stufen (z.B. AGP GART) im Chipsatz ...

x86: Deskriptor-Tabellen

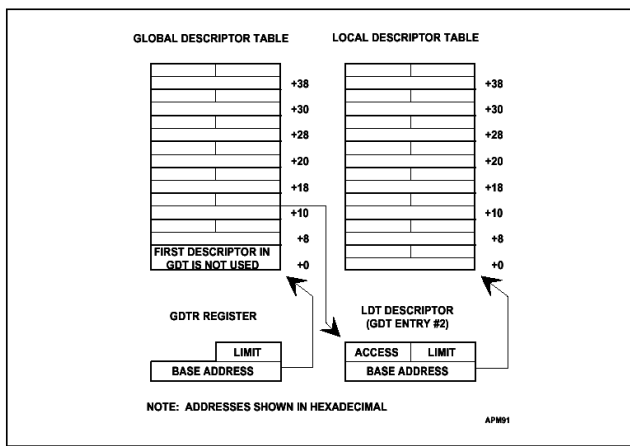
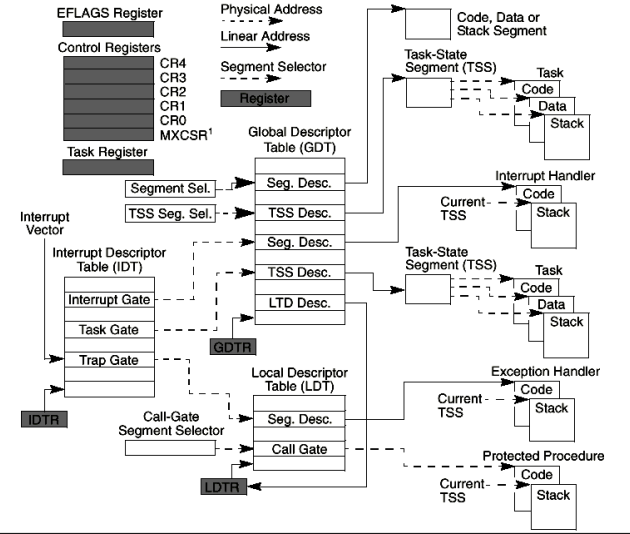


Figure 11-10. Descriptor Tables

- Prüfung von Zugriffsrechten und Adressgrenzen

x86: Pentium Datenstrukturen



x86: Protection Rings

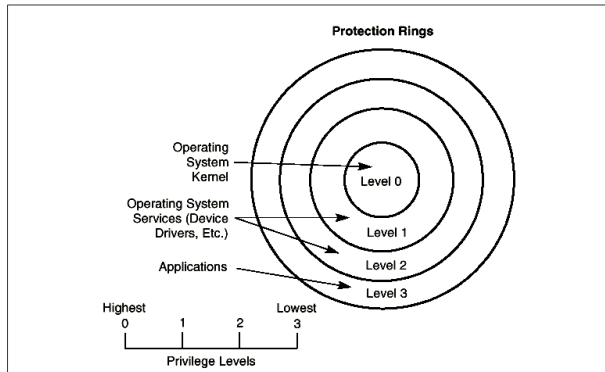


Figure 4-3. Protection Rings

- x86 unterstützt vier (!) getrennte Prioritätsstufen
- sehr feine Steuerung von Zugriffsrechten möglich

x86: I/O permission mask

10.5.2. I/O Permission Bit Map

The I/O permission bit map is a device for permitting limited access to I/O ports by less privileged programs or tasks and for tasks operating in virtual-8086 mode. The I/O permission bit map is located in the TSS (refer to Figure 10-2) for the currently running task or program. The address of the first byte of the I/O permission bit map is given in the I/O map base address field of the TSS. The size of the I/O permission bit map and its location in the TSS are variable.

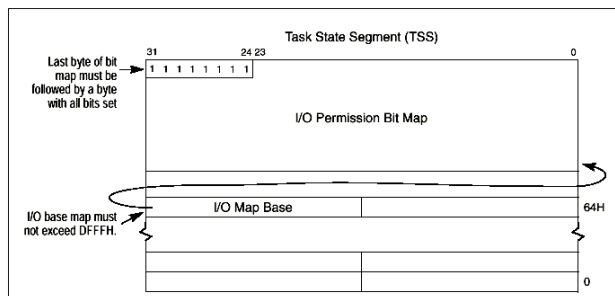


Figure 10-2. I/O Permission Bit Map

- Kontrolle der Zugriffsrechte für jede einzelne I/O-Adresse

x86: INTn INT3 BOUND

4.4.4. INT n, INTO, INT 3, and BOUND Instructions

The INT *n*, INTO, INT 3, and BOUND instructions allow a program or task to explicitly call an interrupt or exception handler. The INT *n* instruction uses an interrupt vector as an argument, which allows a program to call any interrupt handler.

The INTO instruction explicitly calls the overflow exception (#OF) handler if the overflow flag (OF) in the EFLAGS register is set. The OF flag indicates overflow on arithmetic instructions, but it does not automatically raise an overflow exception. An overflow exception can only be raised explicitly in either of the following ways:

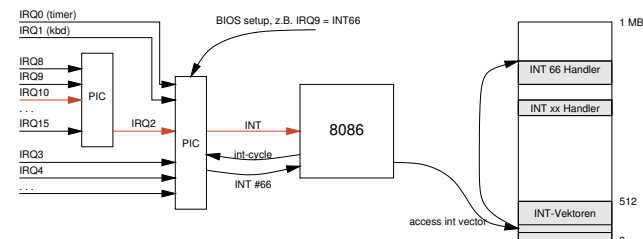
- Execute the INTO instruction.
- Test the OF flag and execute the INT *n* instruction with an argument of 4 (the vector number of the overflow exception) if the flag is set.

Both the methods of dealing with overflow conditions allow a program to test for overflow at specific places in the instruction stream.

The INT 3 instruction explicitly calls the breakpoint exception (#BP) handler.

- wichtigster Mechanismus zum Aufruf von OS-Funktionen

x86: Interrupts im real-mode



```
void INT_66_handler() {
    save_registers_to_stack();
    read_master_PIC();
    if (master_PIC_active) { // hardware interrupt
        read_slave_PIC(); // but which one?
        switch( slave_PIC ) {
            case slave_IRQ8: // handle RTC interrupt
            case slave_IRQ9: // handle s/w int a0h
            case slave_IRQ10: // free
        }
        reset_slave_PIC();
        reset_master_PIC();
    } else { // software interrupt 66
    }
    restore_registers;
    IRET;
}
```

- BIOS programmiert den PIC 8259
- Umsetzung IRQ auf INT-Nummern
- INT-Vektoren ab Adresse 0

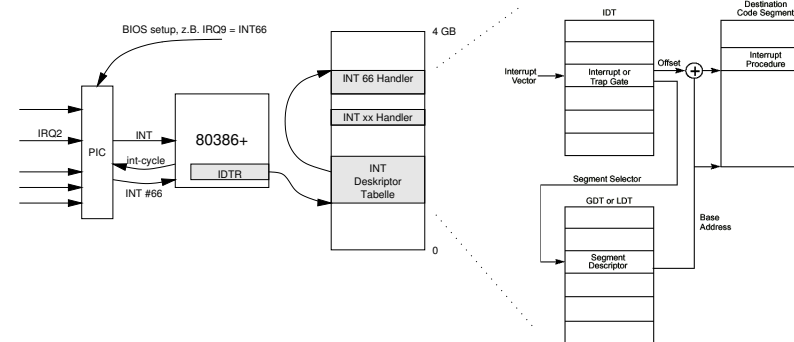
x86: Interrupt/exception vectors

Table 4-1. Exceptions and Interrupts

Vector No.	Mnemonic	Description	Source
0	#DE	Divide Error	DIV and IDIV instructions.
1	#DB	Debug	Any code or data reference.
2		NMI Interrupt	Non-maskable external interrupt.
3	#BP	Breakpoint	INT 3 instruction.
4	#OF	Overflow	INTO instruction.
5	#BR	BOUND Range Exceeded	BOUND instruction.
6	#UD	Invalid Opcode (UnDefined Opcode)	UD2 instruction or reserved opcode. ¹
7	#NM	Device Not Available (No Math Coprocessor)	Floating-point or WAIT/FWAIT instruction.
8	#DF	Double Fault	Any instruction that can generate an exception, an NMI, or an INTR.
9		CoProcessor Segment Overrun (reserved)	Floating-point instruction. ²
10	#TS	Invalid TSS	Task switch or TSS access.
11	#NP	Segment Not Present	Loading segment registers or accessing system segments.
12	#SS	Stack Segment Fault	Stack operations and SS register loads.
13	#GP	General Protection	Any memory reference and other protection checks.
14	#PF	Page Fault	Any memory reference.
15		(Intel reserved. Do not use.)	
16	#MF	Floating-Point Error (Math Fault)	Floating-point or WAIT/FWAIT instruction.
17	#AC	Alignment Check	Any data reference in memory. ³
18	#MC	Machine Check	Error codes (if any) and source are model dependent. ⁴
19	#XF	Streaming SIMD Extensions	SIMD floating-point numeric exceptions. ⁵
20-31		(Intel reserved. Do not use.)	
32-255		Maskable Interrupts	External interrupt from INTR pin or INT <i>n</i> instruction.

1. The UD2 instruction was introduced in the Pentium® Pro processor.
 2. IA processors after the Intel386™ processor do not generate this exception.
 3. This exception was introduced in the Intel486™ processor.
 4. This exception was introduced in the Pentium® processor and enhanced in the Pentium® Pro processor.
 5. This exception was introduced in the Pentium® III processor.

x86: Interrupts im enhanced mode



- Register IDTR: Basisadresse + Größe der Deskriptortabelle
- spezielle Befehle LIDT / SIDT
- Interrupt Deskriptortabelle irgendwo im Hauptspeicher
- mehrere Tabellen möglich: Umladen von IDTR

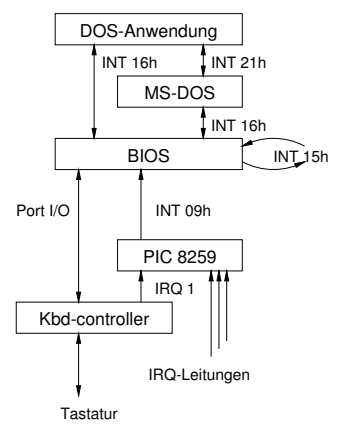
DOS: Tastaturzugriff

direkter Zugriff auf alle Geräte:

- real-mode Adressierung
- kein Speicherschutz
- direkte I/O Portzugriffe
- Interruptcontroller 8259
- DMA-Controller, ...

direkte BIOS-Aufrufe

- => Multitasking sehr problematisch
- => "Virtual 8086 Mode"

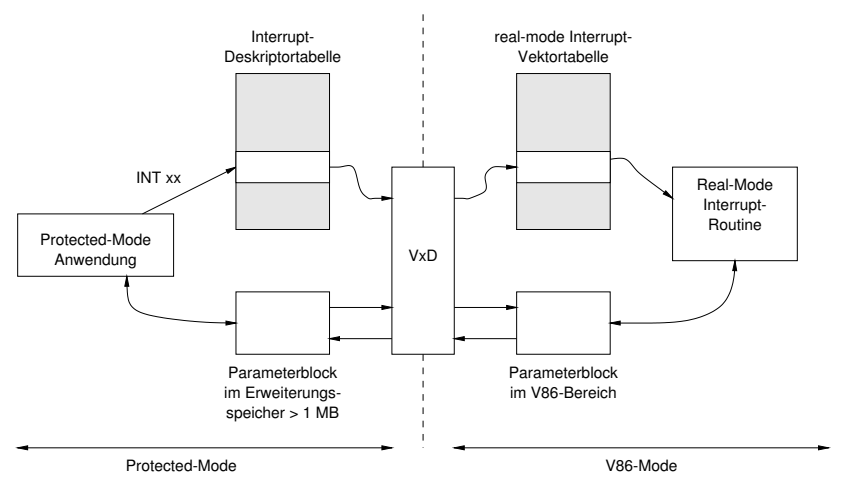


Win9X: "virtuelle Hardware"

virtuelle Maschinen für

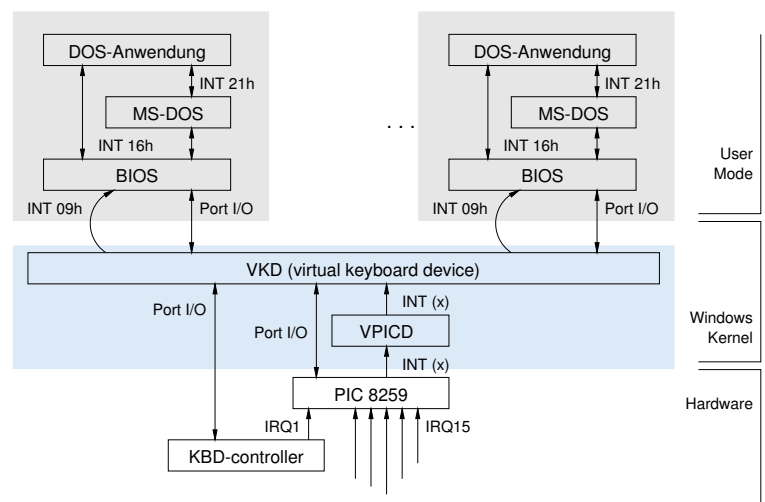
- DOS-Anwendungen
- (veraltete) real-mode Treiber
- Abfangen aller direkten I/O-Hardwarezugriffe
- Überprüfung, ob Zugriff zulässig
- wenn ja, Zugriff ausführen
- nutzt 386+ Virtual 8086 Mode
- real-mode Adressierung (20 bit) plus Paging (32 bit)

Win9x: "Software-Virtualisierung"



- Interrupt abfangen, Register anpassen, Mode umschalten, ...

Win9X: virtueller Tastaturzugriff



x86: CUID, Beispiel AMD K6-III)

Instruction	Returns	Value
Processor Speed Test	399 Mhz	"AuthenticAMD"
CPU ID (0)	Vendor ID	"AuthenticAMD"
CPU ID (1)	Processor & Features	
CPU ID (80000000)	Extended Functions	
CPU ID (80000001)	Processor & Features	
EAX	Processor:	0x691
EDX	Feature Flags	0x8080266F
bit 0	Floating-Point Unit	yes
bit 1	Virtual Mode Ext	yes
bit 2	Debugging Ext	yes
bit 3	Page Size Ext	yes
bit 4	Time Stamp Counter	yes
bit 5	Model Specific Regs	yes
bit 7	Machine Check Ext	yes
bit 8	CMFX/CHG8 Instruct	yes
bit 11	Fast System Call	yes
bit 13	Global Paging Ext	yes
bit 23	MMX Technology	yes
bit 31	3D Now!	yes
CPU ID (80000002..4)	Processor Name	"AMD-K6(tm) 3D+ Processor"
CPU ID (80000005)	L1 Cache information	
EDX	L1B Info	0x02800140
EDX	L1B Data Cache	0x20020220
bits 31-24	size	0x20 (32 Kbytes)
bits 23-16	associativity	0x02
bits 15-8	lines/tag	0x02
bits 7-0	line size	0x20 (32 bytes)
EDX	L1 Instr Cache	0x20020220
CPU ID (80000006)	L2 Cache information	
EDX	L2 Cache information	0x01004220
bits 31-16	size	0x0100 (256 Kbytes)
bits 15-12	associativity	0x04
bits 11-8	lines/tag	0x02
bits 7-0	line size	0x20

x86: Appendix H

APPENDIX H ADVANCED FEATURES

Some non-essential information regarding the Pentium processor are considered Intel confidential and proprietary and have not been documented in this publication. This information is provided in the *Supplement to the Pentium® Processor Developer's Manual* and is available with the appropriate non-disclosure agreements in place. Please contact Intel Corporation for details.

The *Supplement to the Pentium® Processor Developer's Manual* contains Intel confidential information on architecture extensions to the Pentium processor which are non-essential for standard applications. This includes low-level registers that provide access to such features as page size extensions, virtual mode extensions, testing and performance monitoring.

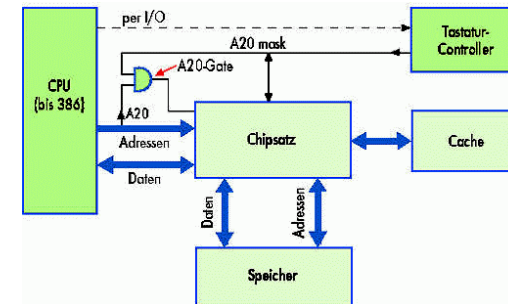
This information is specifically targeted at writers of the following types of software:

- Operating system kernels
- Virtual memory managers
- BIOS software

If you are writing software that does not fall into one of these categories, this information is non-essential and all required programming details are contained in the publicly available *Pentium® Processor Developer's Manual*, three-volume set.

- Details zum Pentium: nur per non-disclosure agreement (NDA)

x86: A20-Gate ...



- 8086 Bug: Überlauf der real-mode Adressen möglich, z.B.: 0ffffh + ffff0h = 10ffffh, aber nur 20 Adressleitungen: x0ffffh
- von Microsoft für DOS-Funktionen benutzt :-)
- Spezialbehandlung des Adressbereichs in Chipsatz oder Prozessor
- Umschaltung mühsam über Tastaturcontroller-Interrupt. . .

RISC vs. CISC: Motivation

- Rechner möglichst schnell, klein, sparsam, ..., aber billig
- sehr vielfältige Lösungen möglich
- Modeerscheinungen - z.B. "high-level instruction sets"

=> Rechnerarchitektur ist eine "Kunst"

=> gute Lösungen abhängig von HW/SW-Technologie

Ausgangsbasis für CISC: VAX, x86, 68000, ...

- Assemblerprogrammierung, schlechte Compiler
- Microcode schneller als Hauptspeicher
- Hardware für Rechenwerke vergleichsweise teuer

=> viele spezielle Maschinenbefehle

=> einmal eingeführte Befehle müssen später mitgeschleppt werden

PC-Technologie | SS 2001 | 18.214

RISC: die IBM 801

John Cocke et.al., IBM, 1975:

- warum CISC? Cache-Zugriffe genauso schnell wie Microcode...

=> Compiler-geeignete Rechnerarchitektur

=> ausschließlich Hochsprache "PL.8": auch für OS und Treiber

=> nur wenige, reguläre Maschinenbefehle

=> aber diese schnell: Pipeline, CPI \approx 1

=> separate I/D-Caches

=> 32 Universal-Register

- 801 vs. S/370: 801 in allen Aspekten besser
- sehr guter Compiler
- wenig publiziert

PC-Technologie | SS 2001 | 18.214

RISC: RISC-I und Mips

ca 1980: 801-Nachfolgeprojekte:

- Berkeley RISC-I "reduced instruction set computer"
- Stanford MIPS "microprocessor w/o interlocked pipeline stages"
- Compiler-gerechte Architektur
- single-Chip VLSI-Implementierung

bessere Performance als 8086/68000:

- sauberer Befehlssatz RISC
- "hardwired" Controller statt Microcode
- Pipeline
- viele Register, weniger Speicherzugriffe auch für CISC möglich!
- gut optimierende Compiler
- Caches, insbesondere I-Cache

PC-Technologie | SS 2001 | 18.214

RISC: Designphilosophie

- minimaler, regulärer Befehlssatz
- optimale VLSI-Implementierung
- Compiler erledigt den Rest
- Berücksichtigung von Amdahl's Gesetz
- umfangreiche Performance-Simulationen (Benchmarks)

ursprüngliche RISC Entwurfsentscheidungen:

- + 32-bit Prozessor, 4 GByte Adressraum
- + 32 Universalregister (ausser RISC/SPARC)
- + 32-bit Befehlsworte, wenig Formate
- Pipeline-Abhängigkeiten (delayed branches)
- Spezialregister (MIPS mult/div)

PC-Technologie | SS 2001 | 18.214

Loop: Instruction Scheduling

HW/SW-Interaktion auf Prozessoren mit Pipeline:

- Daten/Kontrollabhängigkeiten
- Wartezyklen (stalls), bis Vorgängerstufen fertig

=> sinnvolle Anordnung der Befehle notwendig

=> große Bedeutung optimierender Compiler

- Beispiel-Latenzen: DLX single-issue RISC aus [H&P]

instruction producing	instruction using result	latency [clocks]
FP ALU op.	FP ALU op.	3
FP ALU op.	FP STORE	2
FP LOAD	FP ALU op.	1
FP LOAD	FP STORE	0

```
ld R3, R2(0)
; wait 1
add R4, R4, R3
; wait 3
add R5, R5, R4
```

Loop: Vektor = Vektor + Skalar

- typisches Programmbeispiel: Vektor = Vektor + Skalar

```
int i; double s, x[];
...
for( i=1; i<=1000; i++) {
    x[i] = x[i] + s;
}
...
```

- nicht optimierter Code (am Beispiel DLX):

```
Loop: LD    F0, 0(R1)    ; F0 = array element
      ADDD  F4,F0,F2    ; add scalar in F2
      SD    0(R1),F4    ; store result
      SUBI  R1,R1,8     ; decrement pointer
      BNEZ  R1, Loop   ; branch R1!=zero
```

Loop: ohne Scheduling

```
Loop: LD    F0,0(R1)    ; F0 = array element
      ADDD  F4,F0,F2    ; add scalar in F2
      SD    0(R1),F4    ; store result
      SUBI  R1,R1,#8    ; decrement pointer
      BNEZ  R1, Loop   ; branch R1!=zero
```

- Ausführung auf der Pipeline:

```
Loop: LD    F0, 0(R1)    ; 1 (F0 laden)
      stall ; 2
      ADDD  F4,F0,F2    ; 3 (F0 geladen)
      stall ; 4
      stall ; 5
      SD    0(R1),F4    ; 6 (F4 fertig)
      SUBI  R1,R1,#8    ; 7
      BNEZ  R1, Loop   ; 8
      stall ; 9
```

- 9 Takte / Iteration

```
int i; double s, x[];
...
for( i=1; i<=1000; i++) {
    x[i] = x[i] + s;
}
...
```

Loop: mit Scheduling

```
Loop: LD    F0, 0(R1)    ; 1
      stall ; 2
      ADDD  F4,F0,F2    ; 3
      stall ; 4
      stall ; 5
      SD    0(R1),F4    ; 6
      SUBI  R1,R1,#8    ; 7
      BNEZ  R1, Loop   ; 8
      stall ; 9
```

- Ausnutzen des "branch delay slot": 6 Takte / Iteration

```
Loop: LD    F0, 0(R1)    ; 1
      stall ; 2
      ADDD  F4,F0,F2    ; 3
      SUBI  R1,R1,#8    ; 4
      BNEZ  R1, Loop   ; 5
      SD    8(R1),F4    ; 6
      /offset geändert/
```

```
int i; double s, x[];
...
for( i=1; i<=1000; i++) {
    x[i] = x[i] + s;
}
...
```

Loop: Unrolling

```

; solange R1 >= 3:
;
Loop: LD    F0, 0(R1)    ; element 0
      ADDD  F4, F0, F2  ;
      SD    0(R1), F4  ;

      LD    F6, -8(R1)  ; element 1
      ADDD  F8, F6, F2  ;
      SD    -8(R1), F8  ;

      LD    F10, -16(R1) ; element 2
      ADDD  F12, F10, F2 ;
      SD    -16(R1), F12 ;

      LD    F14, -24(R1) ; element 3
      ADD   F16, F14, F2 ;
      SD    -24(R1), F16 ;

      SUBI  R1, R1, #32 ;
      BNEZ  R1, Loop   ;
    
```

- noch kein Scheduling
- 6.8 Takte / Iteration

```

int i; double s, x[];
...
for( i=1; i<=1000; i++) {
  x[i] = x[i] + s;
}
...
    
```

Loop: Unrolling, mit Scheduling

```

; solange R1 >= 3:
;
Loop: LD    F0, 0(R1)    ; element 0
      LD    F6, -8(R1)  ; element 1
      LD    F10, -16(R1) ; element 2
      LD    F14, -24(R1) ; element 3

      ADDD  F4, F0, F2
      ADDD  F8, F6, F2
      ADDD  F12, F10, F2
      ADDD  F16, F14, F2

      SD    0(R1), F4
      SD    -8(R1), F8
      SD    -16(R1), F12
      SUBI  R1, R1, #32 ;
      BNEZ  R1, Loop   ;
      SD    8(R1), F16 ; 8-32 = -24
    
```



- 3.5 Takte / Iteration
- dreimal schneller als "triviale" Version!

```

int i; double s, x[];
...
for( i=1; i<=1000; i++) {
  x[i] = x[i] + s;
}
...
    
```

Loop: Diskussion

- optimierte Loop 3X schneller
- guter Compiler essentiell

aber:

- Optimierungen/Compiler nicht trivial
- maschinenspezifisch wegen Latenzen/Abhängigkeiten
- Loop-Unrolling erfordert viele Register
- erst recht für superskalare Maschinen

x86 hat zu wenig Register:

- => Compiler kann nicht optimieren
- => Register-Renaming / Tomasulo's Algorithmus

Loop: Register Renaming

```

; solange R1 >= 3:
;
Loop: LD    F0, 0(R1)    ; nur F0, F2, F4 verfügbar:
      ADDD  F4, F0, F2  ;
      SD    0(R1), F4   ; => zusätzliche Abhängigkeiten

      LD    F0, -8(R1)  ;
      ADDD  F4, F0, F2  ;
      SD    -8(R1), F4  ;

      LD    F0, -16(R1) ;
      ADDD  F4, F0, F2  ;
      SD    -16(R1), F4 ;

      LD    F0, -24(R1) ;
      ADDD  F4, F0, F2  ;
      SD    24(R1), F4  ;
    
```

- x86-Compiler hat nicht genug Register zur Auswahl
- viele zusätzliche "Name-Dependencies"

- => "Register Renaming" zur Laufzeit im Prozessor (!)
- => ~100 Register mit "Scoreboard" zur Kontrolle der Abhängigkeiten
- Athlon: bis zu 72 Befehle aktiv ...

superskalar: Register Renaming

```

...
R3 = R0 * R1
R4 = R0 + R2
R5 = R0 + R1
R6 = R1 + R4
R7 = R1 * R2
R1 = R0 - R2
R3 = R3 * R1
R1 = R4 + R4
...

...
R7 = R1 * R2
S1 = R0 - R2
R3 = R3 * S1
R1 = R4 + R4
...

...
R7 = R1 * R2
R1 = R4 + R4
...

S1 = R0 - R2
R3 = R3 * S1
...

```

- Compiler darf nur "definierte" Register verwenden
- auch für Zwischenergebnisse
- dadurch zusätzliche, unnötige RAW/WAR/WAW-Konflikte
- Auflösen der Konflikte durch Einsatz "interner" Register
- Verwaltung automatisch durch den Prozessor: Scoreboard

superskalare Prozessoren: Scoreboard

Zy	#	Dekodiert	Iss	Ret	Gelesene Register								Beschriebene Register										
					0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7			
1	1	R3=R0 * R1	1		1	1																	
2	2	R4=R0 + R2	2		2	1	1																
3	3	R5=R0 + R1	3		3	2	1																
4	4	R6=R1 + R4			3	2	1																
5					3	2	1																
6					1	2	1	1															
7					2	1	1																
8					3																		
9	5	R7=R1 * R2	4			1		1															1
10	6	R1=R0 - R2				2	1	1															1
11	7				4	1	1																1
12					5																		
13	8	R1=R4 + R4	8						2														1
14									2														1
15					8																		

Abb. 4.43: Operation einer superskalaren CPU mit Ausgabe und Fertigstellung von Instruktionen entsprechend ihrer Reihenfolge [Tanenbaum 99]

Befehlsausführung
superskalar,
in-order execution
(15 Takte)

superskalare Prozessoren: Scoreboard

Zy	#	Dekodiert	Iss	Ret	Gelesene Register								Beschriebene Register										
					0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7			
1	1	R3=R0 * R1	1		1	1																	
2	2	R4=R0 + R2	2		2	1	1																
3	3	R5=R0 + R1	3		3	2	1																
4	4	R6=R1 + R4			3	2	1																
5	5	R7=R1 * R2	5		3	3	2																1
6	6	S1=R0 - R2	6		4	3	3																1
7					2	3	3	2															1
8	7	R3=R3 * S1	4		3	4	2	1															1
9	8	S2=R4 + R4			3	4	2	3															1
10					1	2	3	2	3														1
11					3	1	2	2	3														1
12					6	2	1	3															1
13					7	2	1	1	3														1
14					4	1	1	2															
15					5	1	1	2															
16					8			1	2														
17						1		1															
18																							
19																							

Abb. 4.44: Operation einer superskalaren CPU mit Ausgabe und Fertigstellung von Instruktionen außer der Reihe [Tanenbaum 99]

Befehlsausführung
superskalar,
out-of-order execution
(9 Takte)

superskalare Prozessoren

VLSI-Technologie erlaubt immer mehr Transistoren/Chip

- größere Caches?
 - komplexere Prozessoren?
- | | | |
|---------------------|-------------------------|---------|
| • klassischer CISC: | serieller Befehlszyklus | CPI |
| • einfacher RISC: | Pipeline, 1 Befehl/Takt | 5 .. 15 |
| • superskalar: | mehrere Befehle/Takt | ~ 1 |
| | | < 1 |

- => I-Cache muss mehrere Befehle pro Takt liefern
- => Daten/Kontrollabhängigkeiten berücksichtigen
- => Ressourcen-Konflikte, Scoreboarding
- => extreme Komplexität

=> Speicherzugriffe sind das Nadelöhr:
1 GHz, 4 Befehle/Takt, 100 ns Latenz: 400 Befehle idle

superskalare Prozessoren

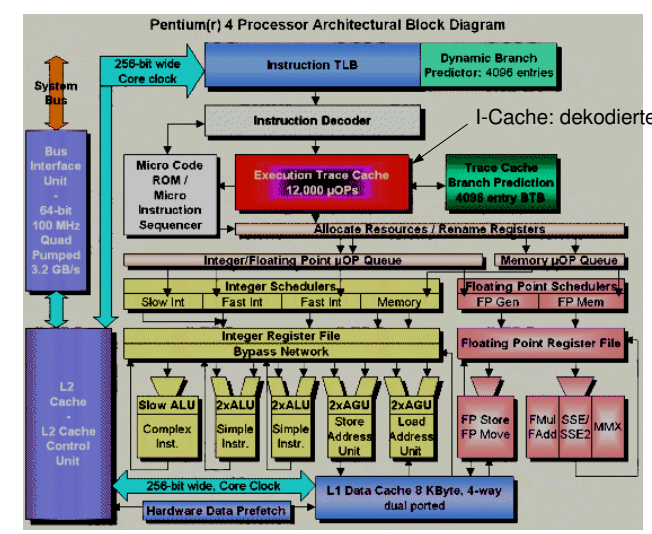
RISC vs. CISC für superskalare Prozessoren: RISC CISC

- | | | |
|--|---|---|
| komplexe Befehlsdekodierung | | • |
| mehrfache Funktionseinheiten | • | • |
| komplexes Steuerwerk (Scoreboard etc.) | • | • |
| out-of-order execution | • | • |
| große on-chip Caches | • | • |
| Speicherzugriffe sind das Nadelöhr | • | • |

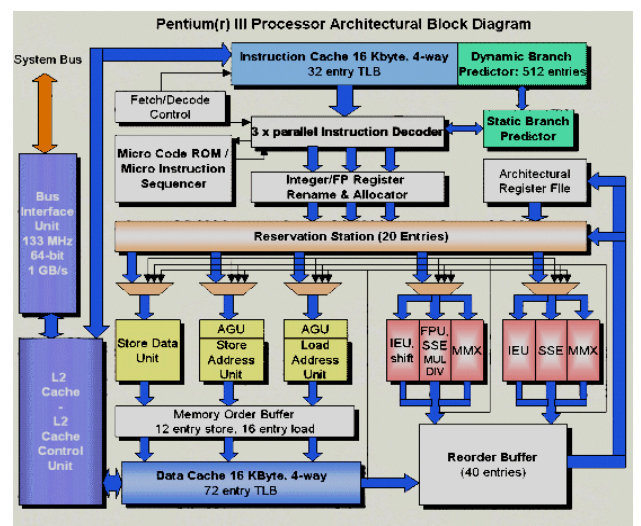
=> extreme Komplexität für RISC und CISC

- Marktbedeutung der IA-32 erlaubt große Investitionen
- bessere Chiptechnologie zuerst für x86 (Intel, AMD)
- alle x86-Prozessoren seit Pentium sind superskalar
- vgl. AMD K7 Präsentation (extern)
- K7 verwaltet bis zu 72 "instructions in flight"

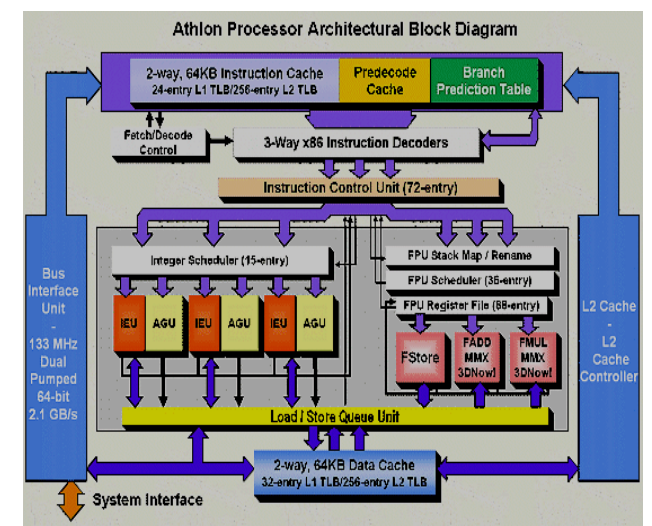
Pentium IV



Pentium III



Athlon (Thunderbird)

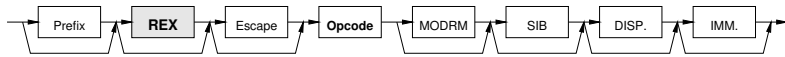


x86: AMD x86-64

- 64-bit Erweiterung der IA-32 [www.amd.com]
- voll abwärtskompatibel
- gute Performance für 32-bit Applikationen

- 64-bit Register und Programmzähler
- flacher 64-bit Adressraum
- 8 zusätzliche Universalregister
- 8 zusätzliche ISSE-Register, ISSE2 Funktionen

- diverse Betriebsmodi (64-bit / compatibility-64 / legacy-32)
- Trick: neuer Befehlsprefix "REX" für die 64-bit Befehle



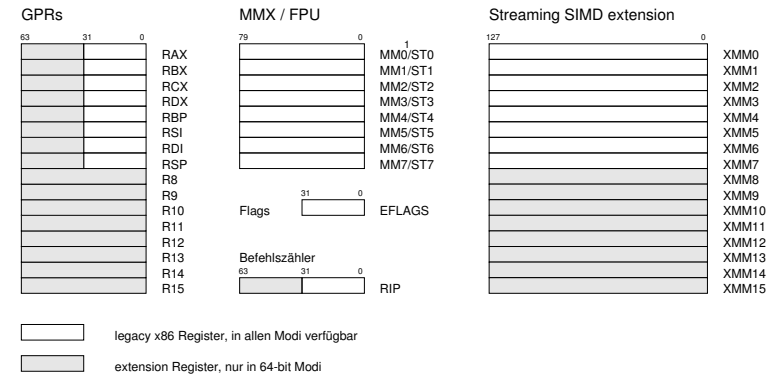
x86: AMD x86-64: Modi

- 64-bit mode: alle 64-bit Erweiterungen
- compatibility mode: für 16/32-bit Applikationen unter 64-bit OS
- legacy mode: Pentium-kompatibel

Mode		OS required	App. rcompile	#address bits	operand size	register extensions	GPR width
long mode	64-bit mode	new 64-bit OS	yes	64	32	yes	64
	compatibility mode		no	32 16	32	no	32
legacy mode		legacy 32/16-bit OS	no	32 16	32 16	no	32

x86: AMD x86-64

- fast doppelte Anzahl der Register gegenüber IA-32
- erstmals wirkliche Universalregister



x86: Intel IA-64

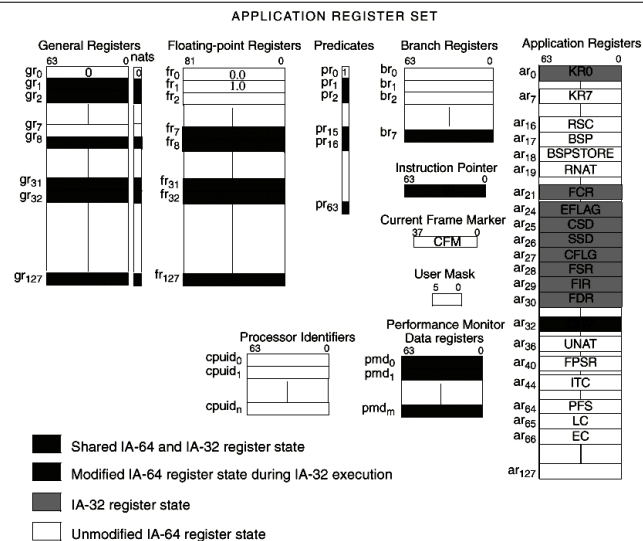
- völlig neue 64-bit Architektur
- basiert auf 64-bit RISC, vor allem HP PA-RISC

- Load/Store Architektur, 64-bit Register, 64-bit Adressen
- ein Befehlsformat: 41 bit mit Opcode und 3-Registeradressen
- viele parallele Funktionseinheiten
- sehr viele Register (mehr als 300 im Merced)

- Bündelung: je drei Befehle in zwei Speicherworten, "Maske"
- parallele Ausführung der Befehlsbündel (sofern möglich)
- Compiler verantwortlich für effiziente Bündelung
- "explicitly parallel instruction computing", EPIC
- Prädikation

- zusätzlicher Emulations-Modus für x86-Programme

IA64: Registersatz (und IA32-Modus)



PC-Technologie | SS 2001 | 18.214

x86: IA-64 "predication"

- bedingte Sprünge behindern das Pipelining
- effiziente Sprungvorhersage notwendig (BTB, BTC, ...)
- oft genügt "bedingte Ausführung" statt eines Sprunges:

```

if (R1 == 0) {
    R2 = R3;
}
    CMP R1, 0
    BNE L1
    MOVZ R2, R3, R1
    ; conditional move
    NOV R2, R3
L1: ...
    
```

- allgemein: Bedingung setzt Flag in einem Register
- then-Zweig arbeitet mit CMOV, else-Zweig mit CMOVN
- keine Sprungbefehle, Pipeline wird nicht behindert

```

if (R1 == 0) {
    R2 = R3;
    R4 = R5;
} else {
    R6 = R7;
    R8 = R9;
}
    CMP R1, 0
    BNE L1
    MOV R2, R3
    MOV R4, R5
    MOV R4, R5
    BR L2
L1: MOV R6, R7
    MOV R7, R8
L2:
    
```

PC-Technologie | SS 2001 | 18.214

x86: IA-64 load

- Speicherzugriffe sind langsam
- erst recht bei Cache-Misses oder Multiprozessorsystemen
- precise exceptions für Speicherzugriffe sind problematisch

"spekulatives Laden":

- Compiler verschiebt Leseoperationen möglichst nach vorne
- Zielregister der Ladeoperation wird als "dirty" markiert
- Speicher/Cachezugriffe werden "auf Probe" durchgeführt
- Resultat steht (hoffentlich) rechtzeitig zur Verfügung

wenn nicht:

- Compiler erzeugt CHECK-Befehle vor Lesen des Zielregisters
- Wartezyklen / Exception nur dann, wenn "dirty" noch gesetzt

PC-Technologie | SS 2001 | 18.214

x86: IA-64 vs. AMD x86-64

Marktbedeutung der IA-64 kaum abschätzbar:

- Unterstützung durch alle grossen Firmen angekündigt
- erste Versionen von Compiler und Tools verfügbar
- aber Hardware ("Merced") verspätet und zu langsam
- Rolle der AMD x86-x64 noch unsicherer:
- Notlösung, falls IA-64 "durchfällt" ?!
- erstmal neue IA-32 Prozessoren: Pentium-IV, K7-Ultra, ...
- siehe Intel IA-64 Präsentation (EPIC-Konzept, Merced, Tools)
- siehe AMD Roadmap und x86-64 Präsentation

PC-Technologie | SS 2001 | 18.214

SIMD: Media processing

"Media processing" mit dem PC ?!

- steigende Anforderungen für Audio, Video, Image, 3D
- grosse Datenmengen
- aber oft mit geringer Genauigkeit (8 bit .. 16 bit, 32 bit FP)
- x86-FPU ausgereizt

=> Trick: vorhandene ALUs/Datenpfade für SIMD verwenden

Befehlssatzerweiterungen:

• MMX	"multimedia extension"	1996
• 3Dnow!		1998
• ISSE	"internet SIMD streaming extension"	1999
• AltiVec	(PowerPC G4, Macintosh)	1999
• ISSE2		2000

PC-Technologie | SS 2001 | 18.214

SIMD: Flynn-Klassifikation

SISD "single instruction, single data"

=> jeder klassische PC

SIMD "single instruction, multiple data"

=> Feldrechner/Parallelrechner

=> z.B. Connection-Machine 2: 64K Prozessoren

=> eingeschränkt: MMX&Co: 2-8 fach parallel

MIMD "multiple instruction, multiple data"

=> Multiprozessormaschinen

=> z.B. vierfach PentiumPro-Server

MISD :-)

PC-Technologie | SS 2001 | 18.214

SIMD: Literatur

MMX: "The MMX technology page has been removed"

- developer.intel.com/drg/mmx/manuals/
- developer.intel.com/drg/mmx/appnotes/
- Linux "parallel-processing-HOWTO"
- IEEE Micro 8/96 S.42, c't 01/97 S.228ff

ISSE: Intel website:

- developer.intel.com/software/idap/resources/technical_collateral/pentiumiii/
- c't 04/00 S.314 (ISSE/3Dnow/AltiVec)

3D Now! AMD website:

- www.amd.com/K6/K6docs/, www.amd.com/swdev/
- c't 15/98 S.186 ff
- IEEE Micro 3/4-99 S.37ff

PC-Technologie | SS 2001 | 18.214

Amdahl's Gesetz

"Speedup" durch Parallelisierung?

[Gene Amdahl, 1967]

System 1: berechnet Funktion X, zeitlicher Anteil $0 < F < 1$

System 2: Funktion X' ist schneller als X mit "speedup" SX:

$$SX = \text{Zeitbedarf}(X) / \text{Zeitbedarf}(X')$$

Amdahl's Gesetz:
$$S_{\text{gesamt}} = \frac{1}{(1-F) + F/SX}$$

=> Optimierung lohnt nur für häufige Operationen !!

=> Beispiele:

$$SX = 10, F = 0.1, S_{\text{gesamt}} = 1 / (0.9 + 0.01) = 1.09$$

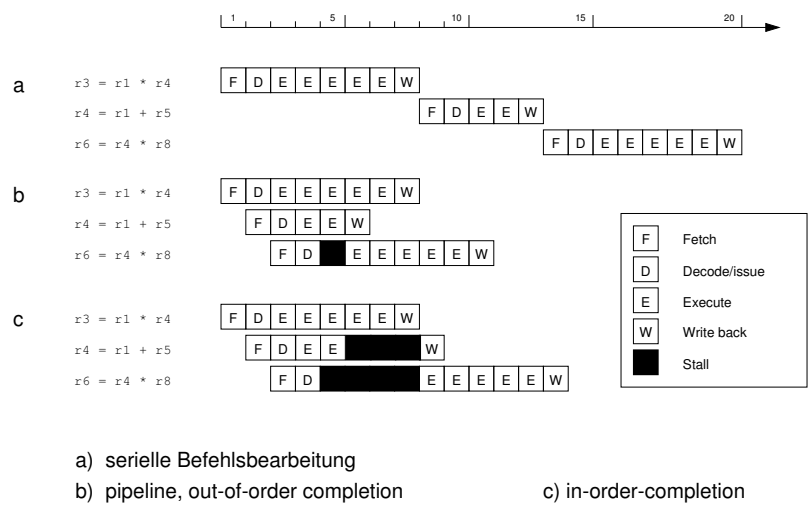
$$SX = 2, F = 0.5, S_{\text{gesamt}} = 1 / (0.5 + 0.25) = 1.33$$

$$SX = 2, F = 0.9, S_{\text{gesamt}} = 1 / (0.1 + 0.45) = 1.82$$

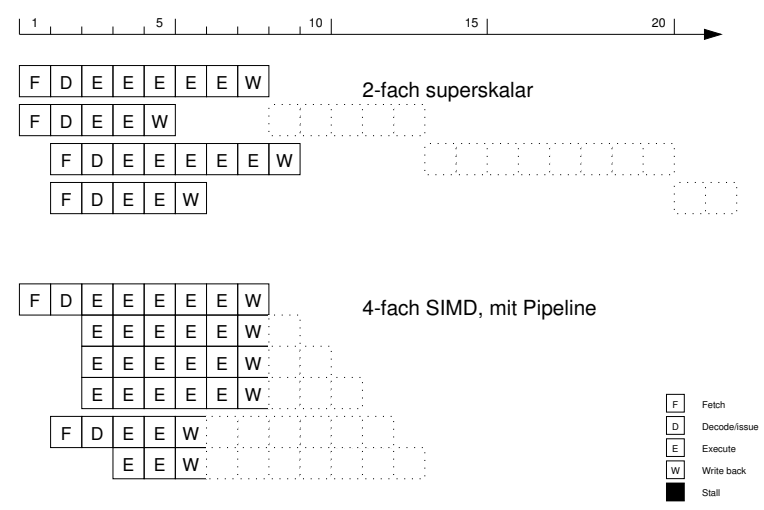
$$SX = 1.1, F = 0.98, S_{\text{gesamt}} = 1 / (0.02 + 0.89) = 1.10$$

PC-Technologie | SS 2001 | 18.214

Befehlspipeline: in order / out of order

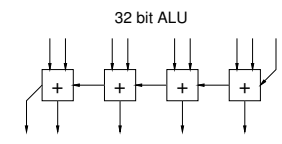


Superskalar, SIMD

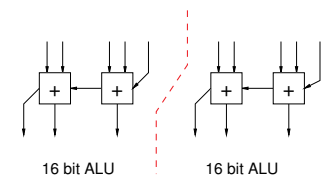


MMX: Grundidee

- 32/64-bit Datenpfade sind "overkill"
- ALUs aber leicht parallel nutzbar:
- carry-chain auftrennen



- => SIMD leicht implementierbar
- ~10% area on Pentium/MMX
- => Performance 2x .. 8x für MMX Ops
- => Performance 1.5x .. 2x für Apps



- MMX press release 03.05.1996
- Pentium-MMX zusätzlich mit größeren Caches als Pentium
- MMX nur in wenigen Applikationen wirklich genutzt

x86: Register

31	15	0		
EAX	AX	AH AL	accumulator	8086
ECX	CX	CH CL	count: String, Loop	Exx ab 386
EDX	DX	DH DL	data, multiply/divide	
EBX	BX	BH BL	base addr	
ESP	SP		stackptr	
EBP	BP		base of stack segment	
ESI	SI		index, string src	
EDI	DI		index, string dst	
	CS		code segment	
	SS		stack segment	
	DS		data segment	
	ES		extra data segment	
	FS			
	GS			
EIP	IP		PC	
EFLAGS			status	

79	0
FPR0	
FPR7	

FP Status

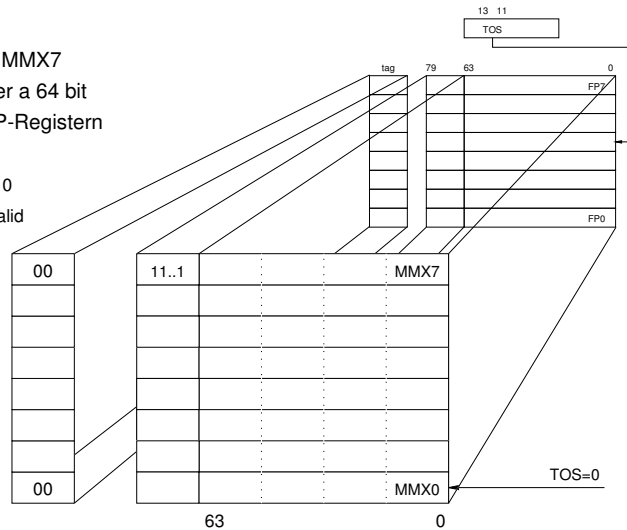
MMX: Entwurfsentscheidungen

Kompatibilität zu alten Betriebssystemen / Apps:

- keine neuen Register möglich => FP-Register nutzen
- keine neuen Exceptions => Überlauf ignorieren
- bestehende Datenpfade nutzen => saturation arithmetic
- möglichst wenig neue Opcodes => 64 bit
- alte Prozessoren und neue Software => Code doppelt
- => MMX DLLs
- Test-Applikationen: => 16 bit dominiert
- (audio/image/MPEG-1/3D-Graphik/...) => MMX DLLs
- keine Tools => Assembler
- optimierte Libraries verfügbar

MMX: Register

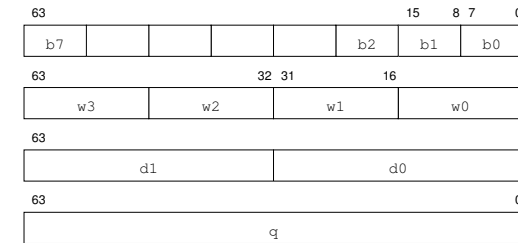
- MMX0 .. MMX7
- 8 Register a 64 bit
- in den FP-Registern
- FP NaN
- FP TOS = 0
- tag 00 = valid



MMX: Datenformate

64-bit Register, 4 Datentypen:

- packed byte *8 / packed word *4 / packed doubleword *2 / quadword
- Zugriff abhängig vom Befehl

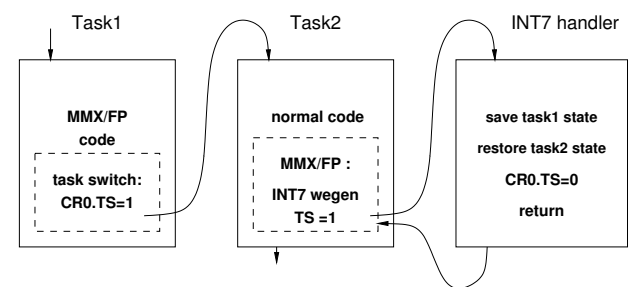


MMX: Befehlssatz

EMMS (FSAV / FRESTOR)		clear MMX state (handle FP regs)
MOVQ	mm1, mm2/mem32	move 32 bit data
MOVQ	mm1, mm2/mem64	move 64 bit data
PACKSSWB	mm1, mm2/mem64	pack 8*16 into 8*8 signed saturate
PUNPCKH	mm1, mm2/mem64	fancy unpacking (see below)
PACKSSDW	mm1, mm2/mem64	pack 4*32 into 4*16 signed saturate
PAND	mm1, mm2/mem64	mm1 AND mm2/mem64 / auch OR/XOR/NAND
PCMPEQB	mm1, mm2/mem64	8*a==b, create bit mask / auch GT
PADDB	mm1, mm2/mem64	8*add 8 bit data
PSUBD	mm1, mm2/mem64	2*sub 32 bit data / signed wrap
PSUBUSD	mm1, mm2/mem64	2*sub 32 bit data / unsigned saturate
PSLL	mm1, mm2/mem64/imm8	shift left mm1 / auch PSRA/PSRL
PMULL/HW	mm1, mm2/mem64	4*mul 16*16 store low/high 16 bits
PMADDWD	mm1, mm2/mem64	MAC 4*16 -> 2*32
insgesamt 57 Befehle		(Varianten B/W/D S/US)

MMX: Multitasking ...

Interaktion mit Betriebssystem / Taskwechsel:

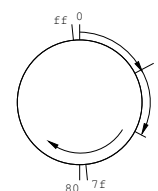


- FP-Register nur bei Bedarf sichern
- vorhandene FP INT7 Routine funktioniert auch für MMX
- keine Anpassung des Betriebssystems notwendig

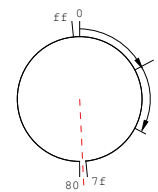
MMX: "Saturation Arithmetic"

was soll bei einem Überlauf passieren?

- wrap-around
..., 125, 126, 127, -128, -127, ...



- saturation
..., 125, 126, 127, 127, 127, ...
- Zahlenkreis "aufgeschnitten"
- gut für DSP-Anwendungen



paddw (wrap around):

a3	a2	a1	7FFFh
+	+	+	+
b3	b2	b1	0004h
a3+b3	a2+b2	a1+b1	8003h

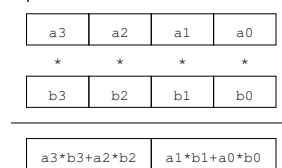
paddsw (saturating):

a3	a2	a1	7FFFh
+	+	+	+
b3	b2	b1	0003h
a3+b3	a2+b2	a1+b1	7FFFh

MMX: "packed multiply add word"

für Skalarprodukte:

pmaddwd



```

vector_x_matrix_4x4( MMX64* v, MMX64 *m ) {
    MMX64 v0101, v2323, t0, t1, t2, t3;

    v0101 = punpckldq( v, v ); // unpack v0/v1
    v2323 = punpckhdq( v, v ); // unpack v2/v3

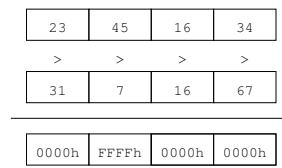
    t0 = pmaddwd( v0101, m[0] ); // v0|v1 * first 2 rows
    t1 = pmaddwd( v2323, m[1] ); // v2|v3 * first 2 rows
    t2 = pmaddwd( v0101, m[2] ); // v0|v1 * last 2 rows
    t3 = pmaddwd( v2323, m[3] ); // v2|v3 * last 2 rows

    t0 = padd( t0, t1 ); // add
    t2 = padd( t2, t3 ); //
    v = packssdw( t0, t2 ); // pack 32->16, saturate
}
    
```

MMX: "packed compare"

Vergleiche / Sprungbefehle:

pcmpgtw:



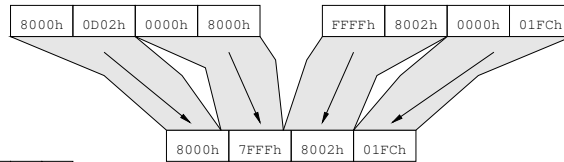
- schlecht parallelisierbar
- Pipeline-Abhängigkeiten

=> keine Sprungbefehle in MMX
=> compare-Operationen setzen Bitmasken

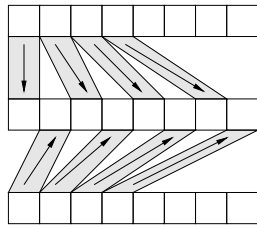
- Bitmasken für logische Ops verwendbar
- Beispiel: chroma-keying

MMX: packssdw / punpckhbw

packssdw: pack with saturation 32 -> 16 signed data:



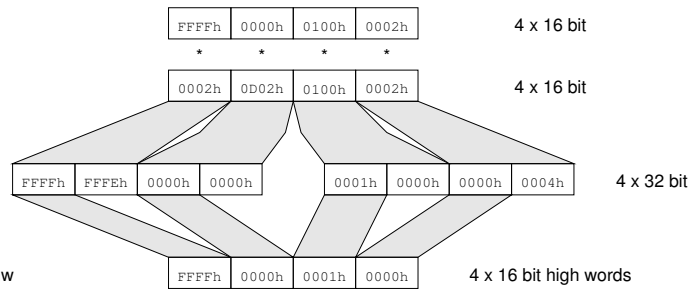
punpckhbw:



punpcklbw: lower 32 bits

MMX: pmullw / pmullhw

pmull[hw]: multiply 4 words, write low/high byte of results:



pmullhw

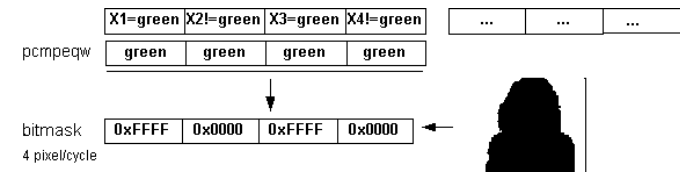
pmullw

mit Packbefehlen kombinieren, wenn 32-bit Resultate gewünscht

MMX: Chroma Keying (1)

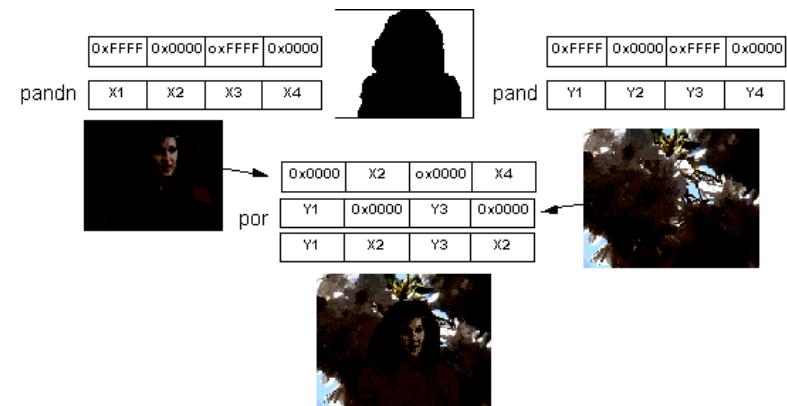
"Wetterbericht":

- MMX berechnet 4 Pixel / Takt
- keine Branch-Befehle
- Schritt 1: Maske erstellen (high-color: 16 bit/pixel)



[Intel MMX appnote]

MMX: Chroma Keying (2)



MMX: Zufallszahlen

```
x(t) = (x(t-1) * 47989) & 0xFFFF;
```

```
QuadWord DithMultVal = 0x4f314f314f314f31;
QuadWord DithRegInit = 0x4f31994d2379bb75;
```

```
Init:
MOVQ mm0, DithRegInit;
```

```
Loop:
PMULLW mm0, DithMultVal // x(t) -> x(t+1)
MOVQ [result64], mm0 // 1 clocks
```

- PMULLW latency 3, throughput 1 (on Pentium)
- bis zu vier Zufallszahlen pro Takt (U/V pipelines genutzt)

3Dnow! Motivation

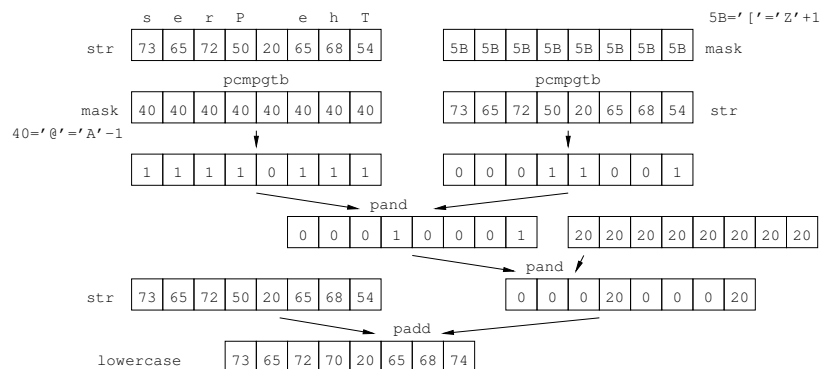
- stark wachsende Bedeutung von 3D-Spielen
- 32-bit Gleitkommaoperationen nötig für Geometrie-Transformationen
- FPU im AMD K6 vergleichsweise langsam
- MMX unterstützt nur Integer-Datentypen

=> SIMD-Befehle für 32-bit float Datentypen

- schnelle Add/Mult/MAC/Sqrt-Befehle
- muß ohne OS-Unterstützung nutzbar sein
- MMX-Register verwenden
- MMX zwei-Operanden Adressierung
- je zwei float-Datenwerte pro MMX-Register

=> 3Dnow! Spezifikation
(vergleiche Motorola AltiVec / Intel SSE)

MMX: toLowerCase()



aber: Probleme mit Umlauten...

[aus Intel MMX appnote]

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0																
1																
2	!	"	#	\$	%	&	'	()	+	,	-	.	/		
3	0	1	2	3	4	5	6	7	8	9	:	<	=	>	?	
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

3Dnow! Entscheidungen

SIMD-Befehle für 32-bit float Datentypen:

- MMX-Register verwenden, zwei Datenwerte pro Register
- zwei-Adress-Befehle
- keine Status-Flags, keine Exceptions
- MMX-Befehle nutzbar (logische, Vergleiche, ...)
- belegt nur einen einzigen x86 Opcode (0F0F ... subopcode)

möglichst wenig Chipfläche:

- keine Unterstützung für NaN/INF/...
- nur round-to-nearest-even Modus, +- 1LSB
- Saturation-Arithmetik statt Überlauf
- Approximation für Division und Quadratwurzel

3Dnow! Befehlssatz

Table 2. 3DNow!™ Floating-Point Instructions

Operation	Function	Opcode Suffix
PAVGUSB	Packed 8-bit Unsigned Integer Averaging	BFh
PFADD	Packed Floating-Point Addition	9Eh
PFSUB	Packed Floating-Point Subtraction	9Ah
PFSUBR	Packed Floating-Point Reverse Subtraction	AAh
PFACC	Packed Floating-Point Accumulate	AEh
PFCMPGE	Packed Floating-Point Comparison, Greater or Equal	90h
PFCMPGT	Packed Floating-Point Comparison, Greater	A0h
PFCMPEQ	Packed Floating-Point Comparison, Equal	80h
PFMIN	Packed Floating-Point Minimum	94h
PFMAX	Packed Floating-Point Maximum	A4h
PI2FD	Packed 32-bit Integer to Floating-Point Conversion	0Dh
PF2ID	Packed Floating-Point to 32-bit Integer	1Dh
PFRCP	Packed Floating-Point Reciprocal Approximation	96h
PFRSQRT	Packed Floating-Point Reciprocal Square Root Approximation	97h
PFMUL	Packed Floating-Point Multiplication	84h
PFRCPT1	Packed Floating-Point Reciprocal First Iteration Step	A6h
PFRSQIT1	Packed Floating-Point Reciprocal Square Root First Iteration Step	A7h
PFRCPT2	Packed Floating-Point Reciprocal/Reciprocal Square Root Second Iteration Step	B6h
PMULHRW	Packed 16-bit Integer Multiply with rounding	B7h

PC-Technologie | SS 2001 | 18.214

3Dnow! Prefetch

Speicherzugriffe in Multimedia-Applikationen:

- reguläre Speicherzugriffsmuster
- ungewöhnliche Lokalität
- viele Daten werden (pro Frame) nur einmal benötigt
- aber regelmässig (in jedem Frame)
- Performance stark von optimaler Cache-Ausnutzung abhängig

=> prefetch-Befehl

- quasi normaler Ladebefehl, aber ohne Zielregister
- gewünschte Daten werden in L1/L2-Cache geladen
- löst keine Exceptions / Page Faults aus

=> "memory streaming"

=> auch für andere Anwendung gut nutzbar (etwa Numerik)

PC-Technologie | SS 2001 | 18.214

3Dnow! Division / Quadratwurzel

- Rechenwerk für Division / Sqrt ist sehr aufwendig
- möglichst wenig Chipfläche für 3Dnow!
- teilweise nur geringe Genauigkeit benötigt
- etwa Shading/Beleuchtungsberechnung für 3D-Graphik

=> Division und Quadratwurzel per Approximation

- erster Befehl liefert 14/15 bit Approximation
- aus Lookup-Table und Interpolation
- mit vollem Takt
- zusätzliche Befehle für Newton-Iteration
- quadratische Konvergenz: zwei Iterationsschritte für volle Genauigkeit
- wenig Hardwareaufwand
- voll in Pipeline integriert, maximaler Durchsatz

PC-Technologie | SS 2001 | 18.214

3Dnow! Division / Quadratwurzel

- Hardware-Dividerer ist sehr aufwendig
- oft wird nicht die volle Genauigkeit benötigt, z.B. Beleuchtungsberechnung bei 3D-Graphik
- Iteration zur Berechnung von $1/b$:

$$x' = x * (2 - b*x)$$

(14-Bit Precision)	MOVD	MM0, [mem]	:	0		w	
	PFRCP	MM0, MM0	:	1/w		1/w	(approx.)
	MOVQ	MM2, [mem]	:	y		x	
	PFMUL	MM2, MM0	:	y/w		x/w	
(24-Bit Precision)	MOVD	MM0, [mem]	:	0		w	
	PFRCP	MM1, MM0	:	1/w		1/w	(approx.)
	PUNPCKLDQ	MM0, MM0	:	w		w	(MMX instruction)
	PFRCPT1	MM0, MM1	:	1/w		1/w	(intermed.)
	MOVQ	MM2, [mem]	:	y		x	
	PFRCPT2	MM0, MM1	:	1/w		1/w	(full prec.)
PFMUL	MM2, MM0	:	y/w		x/w		

PC-Technologie | SS 2001 | 18.214

3Dnow! Quadratwurzel

- Iteration zur Berechnung von $1/\sqrt{b}$:

$$x' = 0.5 * x * (3 - b*x^2)$$

- separate Befehle zur ersten Schätzung (14 bit)
- zwei weitere Befehle zur ersten und zweiten Iteration
- Resultat mit 24-bit Genauigkeit
- abschließende Multiplikation für \sqrt{x} statt $1/\sqrt{x}$

(24-Bit Precision)

```

MOVD    MM0, [mem]    ;      0 | a
PFRSQRT MM1, MM0     ; 1/(sqrt a) | 1/(sqrt a) (approx.)
MOVQ    MM2, MM1     ;      X_0 - 1/(sqrt a) (approx.)
PFMUL   MM1, MM1     ; X_0 * X_0 | X_0 * X_0 step 1
PUNPCKLDQ MM0, MM0  ;      a | a (MMX instr.)
PFRSQRT MM1, MM0     ; (intermediate) step 2
PFRCPIT2 MM1, MM2    ; 1/(sqrt a) (full prec.) step 3
PFMUL   MM0, MM1     ; (sqrt a) | (sqrt a)
    
```

3Dnow! Div/Sqrt-Applet

- Eingabeparameter:
Argument x
Startwert x0
- Ausgabe:
Iterationswerte x0,x1,...
exakte Lösung
- quadratische Konvergenz
- stark abhängig vom Startwert

3D Now! Apfelmännchen

```

Function IterPasD
  (I,R :Double; Grenze, Tiefe :Paratyp):Paratyp;
  var A,B,C:double;
Begin
  Count:= 0;
  A:=0; B:=0;
  Repeat
    C:= SQR(A) - SQR(B) + R;
    B:= 2*A*B + I;
    A:= C;
    INC (Count);
  Until (abs (A) >Grenze) or (Abs (B) > Grenze)
    or (Count=Tiefe);
IterpasD:=Count;
End;
    
```

3D Now! Apfelmännchen

```

; Quadriere (A + jB)**2 = A**2 - B**2 + j 2*A*B
; Entry MM0 ;A | B
; MM1 ;1 | -1
; MM2 ;R | I
;
loop:
  MOVQ MM3,MM0 ;MM3=A | B
  MOVQ MM4,MM0 ; oh weh
  PSLLQ MM3,32 ; das Vertauschen ist
  PSRLQ MM4,32 ; sehr mühsam ...
  POR MM3,MM4 ;MM3=B | A

  PFMUL MM3,MM0 ;MM3= A*B | A*B
  PFMUL MM0,MM0 ;MM0= A**2 | B**2
  PFMUL MM0,MM1 ;MM0= A**2 | -B**2
  PFACC MM0,MM3 ;MM0= A**2 - B**2 | A*B+A*B
  PFADD MM0,MM2 ;MM0= A**2 - B**2 + R | 2*A*B+I
  ; = A(n+1) | = B(n+1)
  ;iA = INT(A) | iB = Int(B)

  PF2ID MM4,MM0
  MOVQ iA,MM4
; Sieh nach, ob A oder B > GRENZE ist
...
dec CX ; iteration counter
jnz loop
    
```

ISSE: Entwurfsentscheidungen

- Markt fordert 3D
- mindestens doppelte FP-Performance notwendig

2-fach oder 4-fach SIMD?

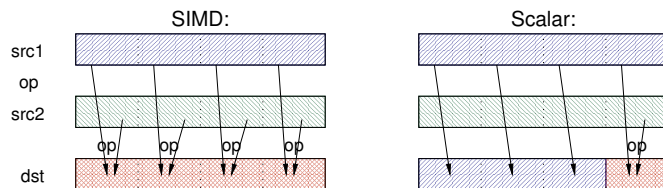
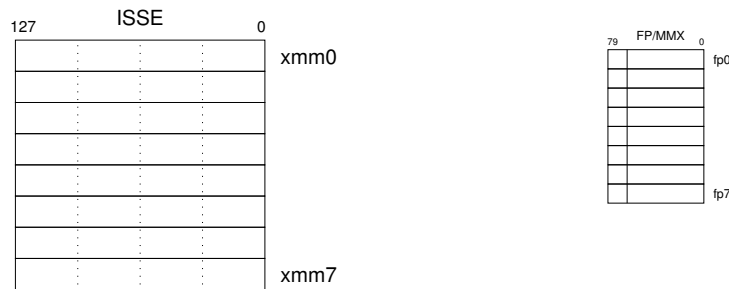
- 128-bit machbar (FP bereits 80-bit)
 - bereits 2 64-bit ALUs auf dem Prozessor
- => 4-fach SIMD

"already register-starved IA32 architecture"

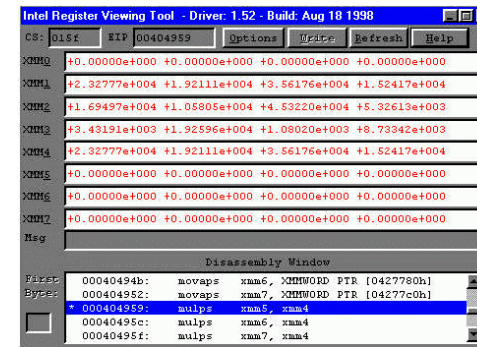
- => neue Register, 128-bit
- => erfordert OS-Unterstützung

- 70 neue Befehle
- sowohl "packed" als auch "scalar ISSE instructions"

ISSE: Register



ISSE: Register Viewing Tool



Softwareentwicklung für MMX / ISSE / 3Dnow:

- nur rudimentäre Compiler- und Tool-Unterstützung
- oft handoptimierter Assembler wg. bester Performance

ISSE: "Streaming"

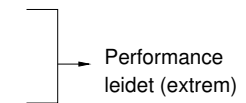
typisch für Medienverarbeitung:

- hohe Datenmenge / Datenrate
- geringe Lokalität: viele Daten (Pixel) werden nur 1x benötigt

=> Cache-"Pollution"

=> herkömmliche Cache-Strategien nutzlos

=> ALUs müssen auf die Daten warten



- 1GHz, 8x SIMD, 100 nsec Speicher: 800 OPs / 1 Zugriff

Streaming:

- Cache-Nutzung anpassen
- Prefetch: Daten rechtzeitig anfordern
- Speicherlatenz fast perfekt versteckt

(für Media-Apps.)

ISSE: Prefetch-Befehl

PREFETCHh—Prefetch Data Into Caches

Opcode	Instruction	Description
0F 18 11	PREFETCHT0 m8	Move data from m8 closer to the processor using T0 hint.
0F 18 12	PREFETCHT1 m8	Move data from m8 closer to the processor using T1 hint.
0F 18 13	PREFETCHT2 m8	Move data from m8 closer to the processor using T2 hint.
0F 18 10	PREFETCHNTA m8	Move data from m8 closer to the processor using NTA hint.

Description

Fetches the line of data from memory that contains the byte specified with the source operand to a location in the cache hierarchy specified by a locality hint:

- T0 (temporal data)—prefetch data into all cache levels.
- T1 (temporal data with respect to first level cache)—prefetch data in all cache levels except 0th cache level
- T2 (temporal data with respect to second level cache)—prefetch data in all cache levels, except 0th and 1st cache levels.
- NTA (non-temporal data with respect to all cache levels)—prefetch data into non-temporal cache structure. (This hint can be used to minimize pollution of caches.)

The source operand is a byte memory location. (The locality hints are encoded into the machine level instruction using bits 3 through 5 of the ModR/M byte. Use of any ModR/M value other than the specified ones will lead to unpredictable behavior.)

If the line selected is already present in the cache hierarchy at a level closer to the processor, no data movement occurs. Prefetches from uncacheable or WC memory are ignored.

The PREFETCHh instruction is merely a hint and does not affect program behavior. If executed, this instruction moves data closer to the processor in anticipation of future use.

ISSE: Programmierung

Intel VTune Performance Enhancement Environment:

- optimierender Compiler mit ISSE-Unterstützung:
 - Intrinsics
 - Vector Class Library
 - Vectorization
 - Intel Performance Library Suite
- C-Funktionen, Compiler inlining
- Klassen, inlining durch Compiler
- optimierender Compiler
- erfordert 16-Byte Alignment aller Datentypen
- umfangreiche Profiling-Tools
- sehr teuer

ISSE: Programmierung mit "Intrinsics"

```
float xa[SIZE], xb[SIZE], xc[SIZE];
float q;

void do_c_triad() {
    for( int j=0; j < SIZE; j++ ) {
        xa[j] = xb[j] + q*xc[j];
    }
}
```

ISSE-Programmierung mit "Intrinsics" und VTUNE:

```
#define VECTOR_SIZE 4
__declspec(align(16)) float xa[SIZE], xb[SIZE], xc[SIZE];
float q;

void do_intrin_triad() {
    __m128 tmp0, tmp1;

    tmp1 = _mm_set_ps1(q);
    for( int j=0; j < SIZE; j+= VECTOR_SIZE ) {
        tmp0 = _mm_mul_ps( *((__m128 *) &xc[j]), tmp1 );
        *((__m128 *) &xa[j]) =
            _mm_add_ps(tmp0, *((__m128 *) &xb[j]);
    }
}
```

ISSE: AoS / SoA

Array of Structures:

```
struct
{
    float A, B, C;
} AoS_data[1000];
```

- Daten lokal
- Anordnung schlecht für SIMD

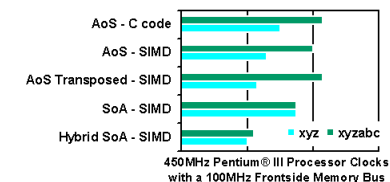
Structure of Arrays:

```
struct
{
    float A[1000], B[1000], C[1000];
} SoA_data;
```

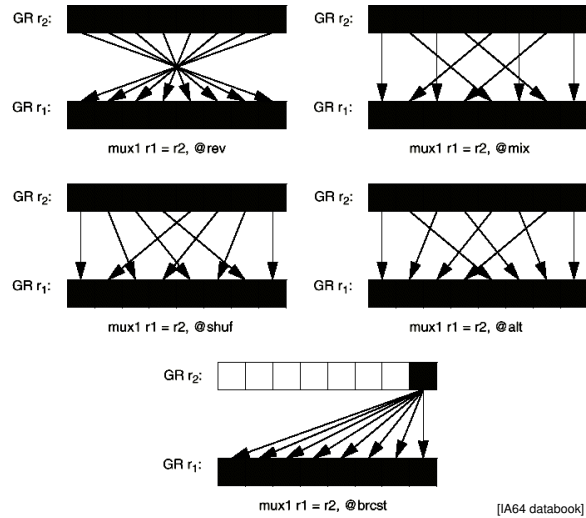
- Anordnung optimal für SIMD
- aber im Speicher "verstreut"

=> Hybrid SoA - SIMD

```
struct
{
    float A[8], B[8], C[8];
} Hybrid_data[125];
```



ISSE2: mux1-Befehl (IA64)



PC-Technologie | SS 2001 | 18.214

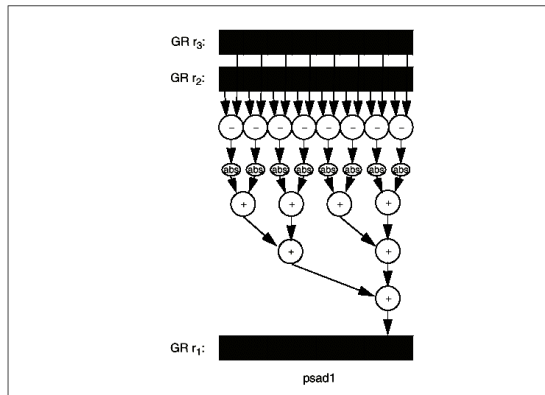
ISSE2: psad1-Befehl (IA64)

Parallel Sum of Absolute Difference

Format: (qp) psad1 r1 = r2, r3

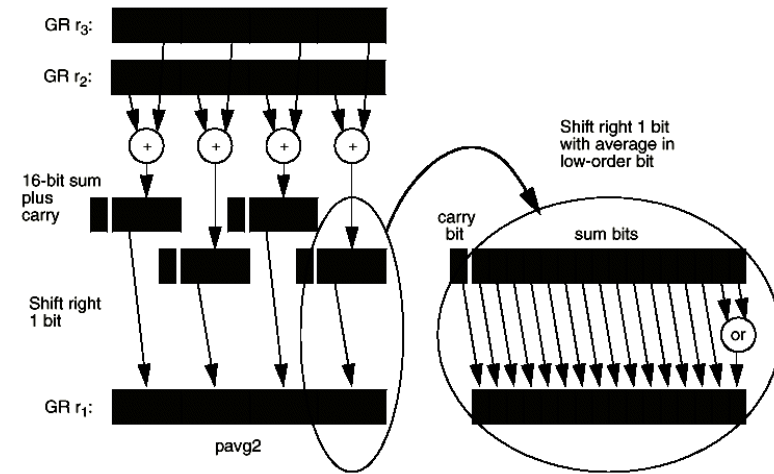
Description: The unsigned 8-bit elements of GR r2 are subtracted from the unsigned 8-bit elements of GR r3. The absolute value of each difference is accumulated across the elements and placed in GR r1.

Figure 7-36. Parallel Sum of Absolute Difference Example



PC-Technologie | SS 2001 | 18.214

ISSE2: pavg2-Befehl (IA64)



PC-Technologie | SS 2001 | 18.214

ISSE: FIR-Filter

Nutzen von MMX/ISSE für Filter?

- MMX und ISSE für 16-bit Integer
- ISSE für 32-bit Gleitkommawerte
- maximal vierfache Leistung gegenüber skalarem Code

aber:

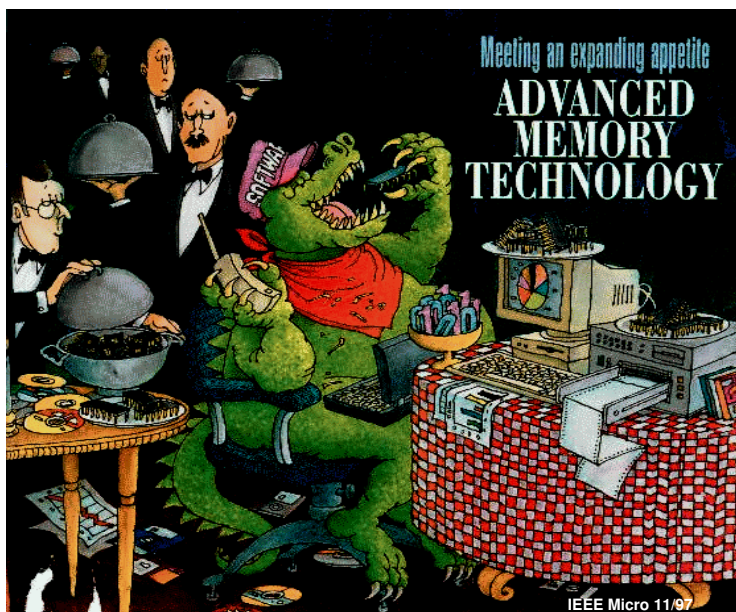
- erfordert Operanden-Alignment (16-Byte Grenzen)
- z.B. durch Duplizierung der Daten/Koeffizienten-Arrays
- Arraygrößen Vielfache von 4
- Multiplikation parallel, aber Akkumulation schwierig

=> siehe Intel Appnote
"32-bit FP FIR Filter implemented using SSE"

PC-Technologie | SS 2001 | 18.214

Speicher: Übersicht

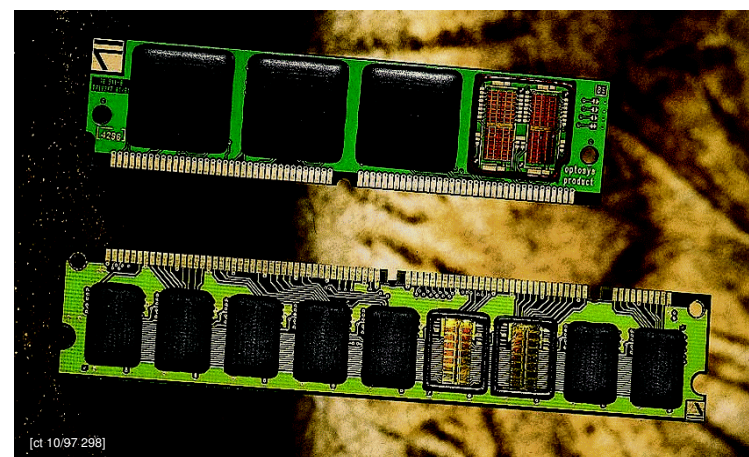
- Motivation: "performance gap" zwischen CPU und Speicher
- DRAM Grundlagen
- Speicherhierarchie, Cache
- SDRAM, Rambus
- IRAM



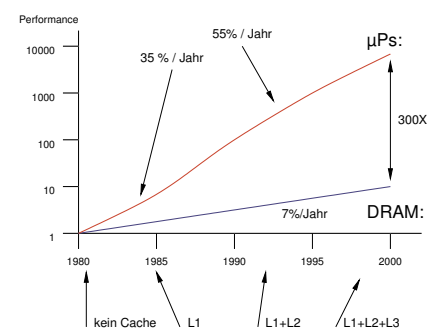
Speicher: Literatur

[IEEE Micro 3/97]	IRAM
[IEEE Micro 11/97]	Advanced Memory Technology: Übersicht, RAMBUS, SDRAM
[Hennessy & Patterson]	Kapitel 5, Speicherhierarchie
[c't 07/96 p.158]	"SIMMsalabim"
[c't 10/97 p.298]	"Schnelle Speicherkäfer"
[c't 96-2000]	diverse Testberichte
www.rambus.com	alle RAMBUS Docs
www.jedec.org	Standards
developer.intel.com	Memory homepage, Chipsätze
[[Cvetanovic/Bhandarkar ISCA 96]	Performance-Analyse Alpha-21164

SIMM / DIMM: 72/168 polig 32/64 bit



DRAM: Performance Gap



- DRAM-Kapazität: 60% / Jahr, Latenz: 7% / Jahr
- Prozessor-Performance: 55% / Jahr
- Kluft vergrößert sich ständig
- => Speicherhierarchie mit Caches notwendig

DRAM: Performance gap: Beispiel

- Zeit für L2-Cache-Miss (# idle instructions):

Alpha 21064 (7000):	340ns / 5.0ns	68 clocks x 2 = 136
Alpha 21164 (8400):	266ns / 3.3ns	80 clocks x 4 = 320
Alpha 21264 (est.):	180ns / 1.7ns	108 clocks x 6 = 648
...		

- Caches essentiell notwendig, um DRAM-Latenz zu verstecken
- Problem wird mit jeder Prozessorgeneration schlimmer
- Beispiel: Analyse für Alpha 21164 [ISCA'96]
- CPU mit idealem Speicher:
Performance durch Verlustleistung limitiert (ca. 50Watt)

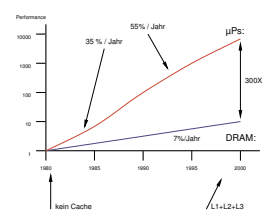
DRAM: Performance Gap: Was tun?

schnellerer Speicher notwendig ...

- aber DRAM inhärent langsam
- SRAM sehr teuer
- => DRAM besser ausnutzen
- SDRAM, SDRAM-DDR
- RAMBUS, SLDRAM

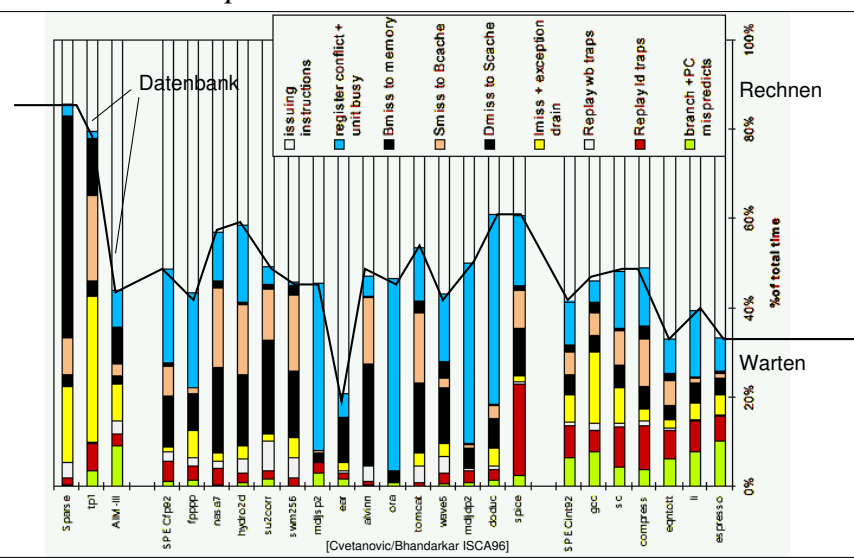
- => Speicherhierarchie
- größere, schnellere Caches
- bessere Cache-Organisation
- Prefetch-Optimierungen

- => neue Konzepte?
- IRAM

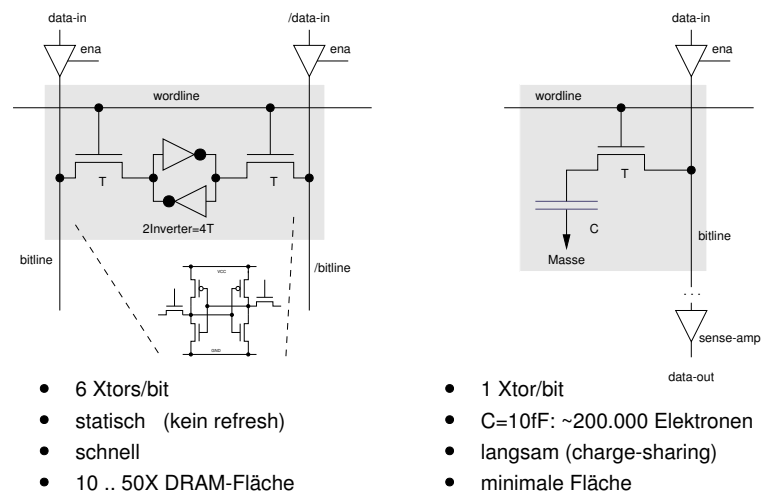


"Cache: a safe place for hiding or storing things"
Websters dictionary

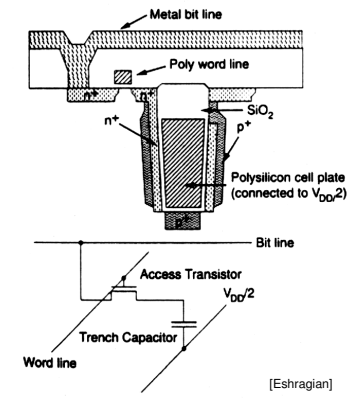
DRAM: Alpha 21164



DRAM vs. SRAM



DRAM: Trench-Kondensator



- Bauform Trench: Platten vertikal am Rand eines Grabens
- Bauform Stack: mehrere horizontale Schichten

DRAM: Stack / Trench-Kondensator

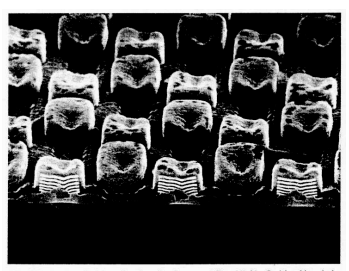
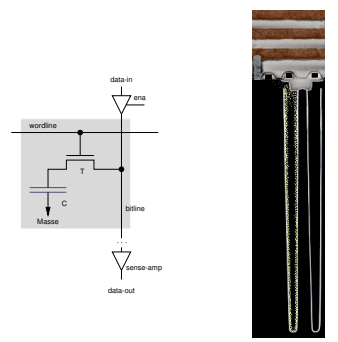
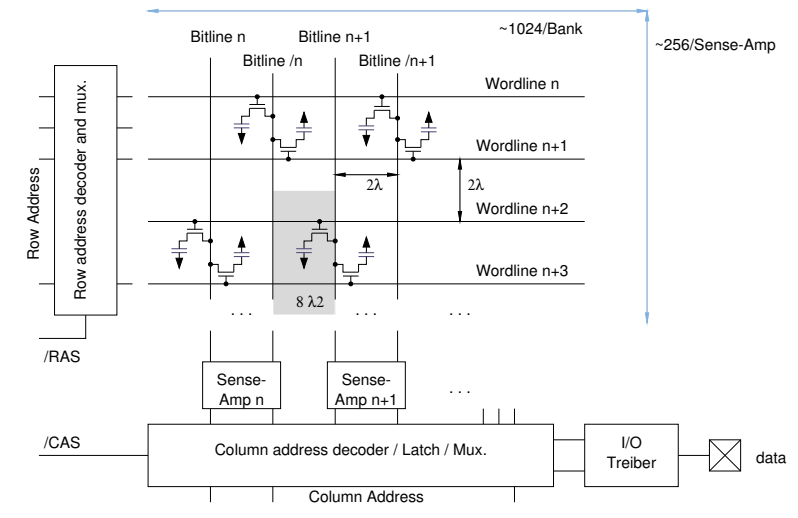


Abb. 7: Prototypen von Speicherzellen (Stacked-Kondensatoren) für zukünftige Speicherchips wie den Ein-Gigabit-Chip. Da für DRAM-Chips eine minimale Speicherkapazität von 25 fF notwendig ist, bringt es erhebliche Platzvorteile, die Kondensatorelemente vertikal übereinander zu stapeln. Die Dicke der Schichten beträgt etwa 50 nm. (Foto: Siemens)

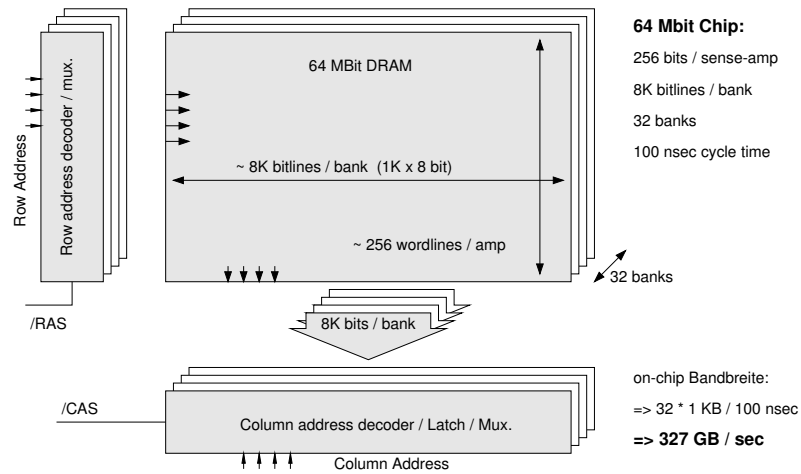


- "stacked capacitors" [Siemens 1Gb DRAM Prototyp 96]
- C=10fF: ~200.000 Elektronen
- "trench capacitors" [IBM CMOS-6X embedded DRAM]

DRAM: Layout



DRAM: Organisation / Bandbreite



PC-Technologie | SS 2001 | 18.214

DRAM: Funktion

Read:

- /RAS = 0: Auswahl der Wordline, Aktivierung der Bitlines
Auslesen und Auswertung der selektierten Zellen
- /CAS = 0: Auswahl der Bitline, Ausgabe der Daten
Zurückschreiben der gelesenen Daten (!)
- /RAS = 1: Precharge der Bitlines

Write:

- /CAS = 0: Zurückschreiben der gelesenen + neuer Daten

SDRAM:

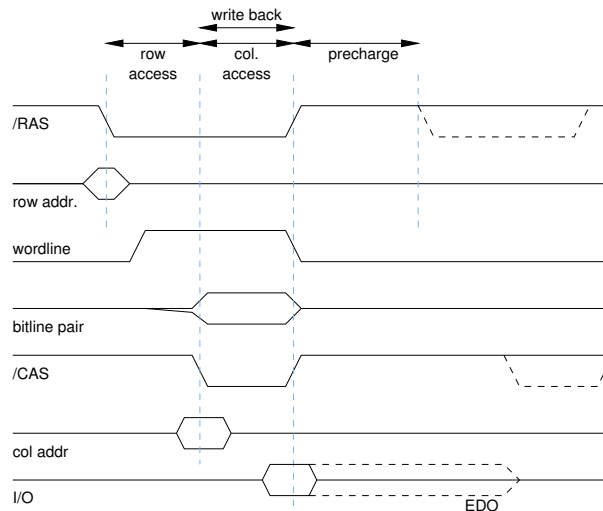
zusätzliche Register, diverse Burst-Modi

Refresh:

alle 16 .. 32 ms notwendig

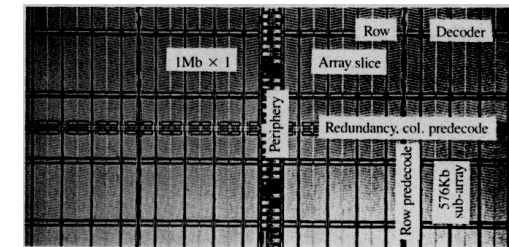
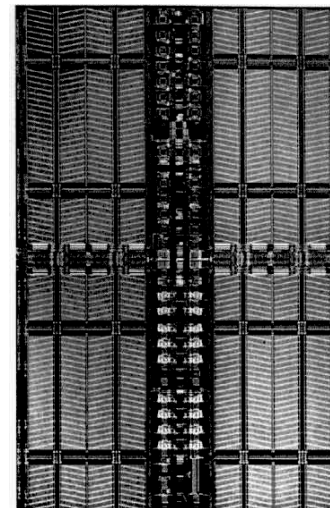
PC-Technologie | SS 2001 | 18.214

DRAM: Ansteuerung (asynchron)



PC-Technologie | SS 2001 | 18.214

DRAM: Floorplan (IBM 4Mbit)



- Größenvergleich zwischen I/O, Col/Row-Decoder, Array
- Konfiguration nach Marktlage
links: 4 Mbit, oben: 16 Mbit
- Redundanz für besseren Yield:
links: 4.0/4.5 Mbit Kapazität/brutto
[IBM JR&D 1995]

PC-Technologie | SS 2001 | 18.214

DRAM: Trend und Dilemma

- Preisverfall: 16Mb: 50\$ @ 1/96 -> 10\$ @ 12/96 -> 4\$ @ 12/97
- Anzahl DRAMs / Computer sinkt:
 - Kapazität steigt mit 50% - 60% / Jahr
 - Software benötigt 33% / Jahr
 - Mindestanzahl gegeben durch Busbreite vs. DRAM-Breite (4bit)
- überhaupt ein Markt für große DRAMs? (256Mb, 1Gb, ...)

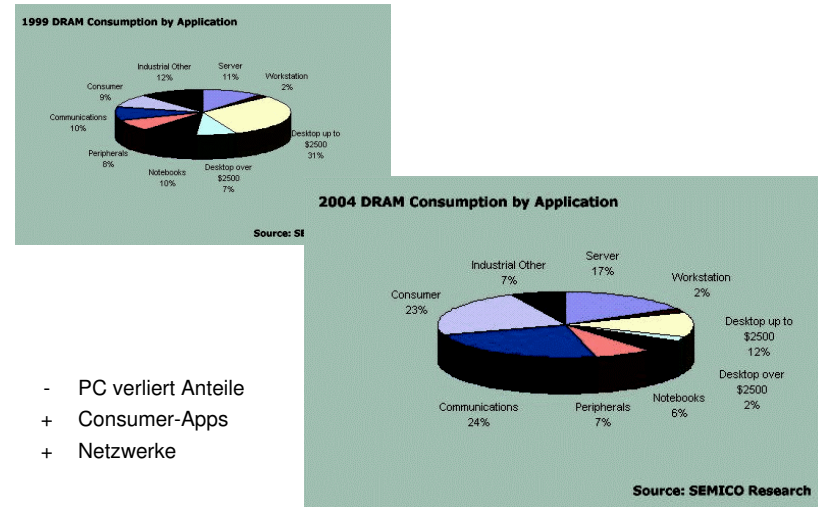
# Chips	' 86	' 89	' 92	' 96	' 99	' 02
	1Mb	4Mb	16Mb	64Mb	256Mb	1Gb
4 MB	32	→ 8				60% / Jahr →
8 MB		16	→ 4			
16 MB			8	→ 2		
32 MB			16	→ 4	→ 1	
64 MB				8	→ 2	
128 MB		33% / Jahr ↓			4	→ 1
256 MB					8	→ 2

DRAM: der Halbleitermarkt

DRAM als Standardbauteile: erfordert standardisierte Schnittstelle

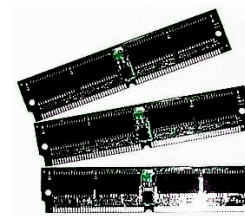
- Markt (1995): DRAMs 37 Mrd. \$, μ Ps 20 Mrd. \$
- hohe Stückzahlen, viele Lieferanten, wenig Profit
- 'quadratische' Speichermatrix mit N*N Bits, extern 1/4/16 Bits
- Architekturverbesserungen minimal: PM, EDO, SDRAM, DDR, ...
- Generationen: 64 Kb, 256 Kb, 1Mb, 4Mb, 16Mb, 256Mb, ... (1 Gb)
- "kleine" Anwendungen müssen bestehendes Angebot nutzen
- spezielle Varianten bei entsprechender Stückzahl (z.B. N64, PSX2)
- PC-Markt bestimmt die Marschrichtung
- Integration von DRAM und Logik zunehmend aktuell (IRAM & Co)

DRAM: Halbleitermarkt

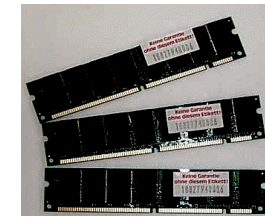


- PC verliert Anteile
- + Consumer-Apps
- + Netzwerke

DRAM: Bauformen SIMM / DIMM / RIMM



EDO-SIMM 60ns, 72p.



SDRAM-100 DIMM 168p.

RAMBUS-PC800 RIMM 168p



SDRAM:

SDRAM: synchrone Ansteuerung für bessere Performance:

- interner Aufbau wie asynchrone DRAMs
- getaktete I/O-Register
- Wertekombination auf CD/nRD/nWE/... wird als Befehl interpretiert
- mehrere Burst Read/Write Modi
- Mode-Register, etwa Auswahl Burstlength 1/2/4/8
- übliche Taktraten 66 MHz / 100 MHz / 133 MHz
- PC-66 / und PC-100 Spezifikationen von Intel
- PC-133 Spezifikation zuerst von VIA / von Intel übernommen
- diverse Varianten (SGRAM / double data rate "DDR" / ...)
- Marktbedeutung <=> Patentstreitigkeiten (u.a. mit Rambus, Inc.)

[developer.intel.com/memory]

SDRAM: Commands

Command Truth Table

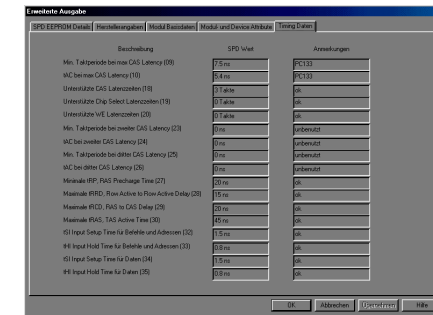
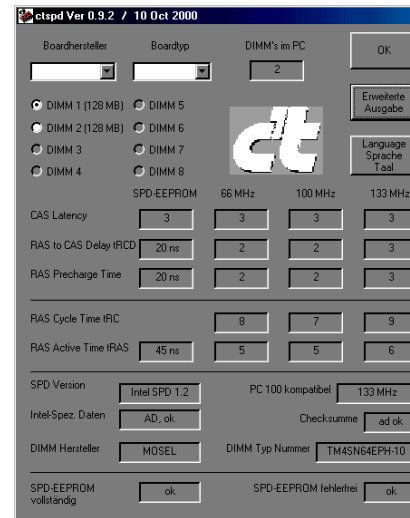
Function	Symbol	CKE n-1	CKE n	CS#	RAS#	CAS#	WE#	A11	A10	BA(0:1)	A9-A0
Device deselect	DSEL	H	X	H	X	X	X	X	X	X	X
No Operation	NOP	H	X	L	H	H	H	X	X	X	X
Read	READ	H	X	L	H	L	H	V	L	V	V
Read w/ auto precharge	READAP	H	X	L	H	L	H	V	H	V	V
Write	WRIT	H	X	L	H	L	L	V	L	V	V
Write w/ auto precharge	WRITEAP	H	X	L	H	L	L	V	H	V	V
Bank Activate	ACT	H	X	L	L	H	H	V	V	V	V
Precharge select bank	PRE	H	X	L	L	H	L	V	L	V	X
Precharge all banks	PALL	H	X	L	L	H	L	X	H	X	X
Auto refresh	CBR	H	H	L	L	L	H	X	X	X	X
Self refresh entry from IDLE	SLFRSH	H	L	L	L	L	H	X	X	X	X
Self refresh exit	SLFRSHX	L	H	H	X	X	X	X	X	X	X
Power Down entry from IDLE	PWRDN	H	L	X	X	X	X	X	X	X	X
Power Down exit	PWRDNX	L	H	H	X	X	X	X	X	X	X
Mode register set	MRS	H	X	L	L	L	L	L	L	V	V

steigende Taktflanke:

- nCS
- nRAS
- nCAS
- nWE

=> SDRAM-Befehl

SDRAM: SPD EEPROM Daten



"serial presence detect":

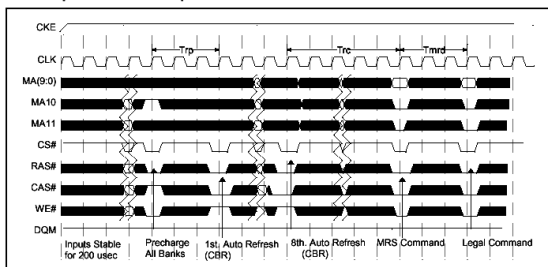
- EEPROM mit allen Timing-Daten
- volle Autokonfiguration
- typ. Zeiten 20 .. 50 nsec.

SDRAM: Initialisierung

The initialization sequence can be issued at *anytime*. Following the initialization sequence, the device must be ready for full functionality. SDRAM devices are initialized by the following sequence:

1. At least one NOP cycle will be issued after the 1msec device deselect.
2. A minimum pause of 200µsec will be provided after the NOP.
3. A precharge all (PALL) will be issued to the SDRAM.
4. 8 Auto refresh (CBR) refresh cycles will be provided.
5. A mode register set (MRS) cycle will be issued to program the SDRAM parameters (e.g., Burst length, CAS# latency etc.).
6. After MRS the device should be ready for full functionality within 3 clocks after T_{mrd} is met.

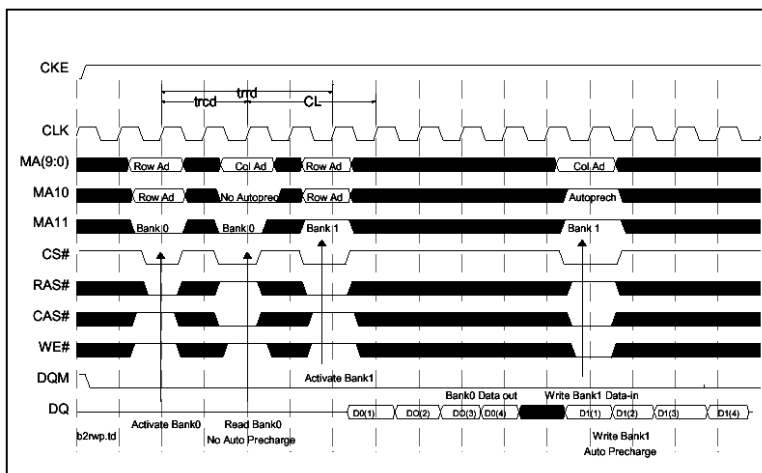
Power Up Initialization Sequence



PC-Technologie | SS 2001 | 18.214

SDRAM: Read / Write Bursts

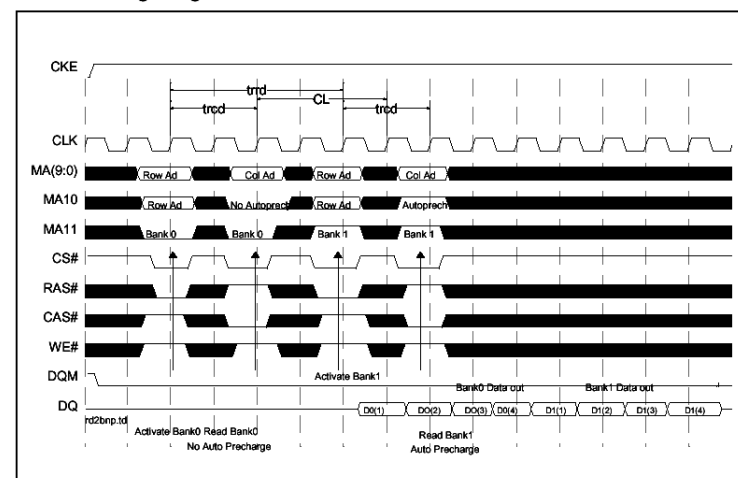
Read and Write Commands (Burst Length 4 Shown)



PC-Technologie | SS 2001 | 18.214

SDRAM: "Ping Pong Read"

Two Bank Ping Pong Read



PC-Technologie | SS 2001 | 18.214

Leersseite

PC-Technologie

SDRAM: DDR Read

Burst Read Operation

The burst read operation in DDR SDRAM is done in the same manner as the current SDRAM. The burst read command is issued by asserting CS and CAS Low while holding RAS and WE High at the rising edge of the clock (CLK) after T_{RCD} from the bank activation. The address inputs determine the starting address for the burst. The mode register sets the type of burst (sequential or interleave) and the burst length (2, 4, 8). The first output data is available after the CAS latency from the read command, and the consecutive data are presented on the falling and rising edge of DQS until the burst length is completed.

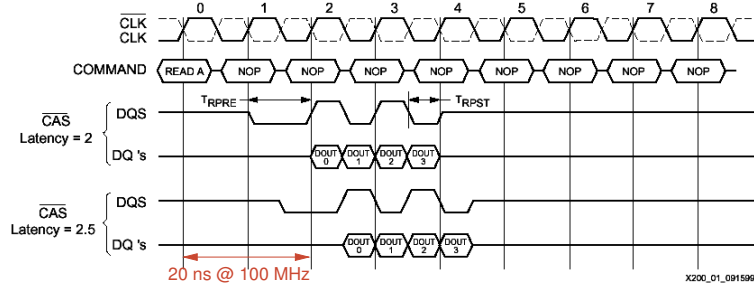


Figure 1: Burst Read Operation Timing

[Xilinx appnote]

SDRAM: DDR Controller

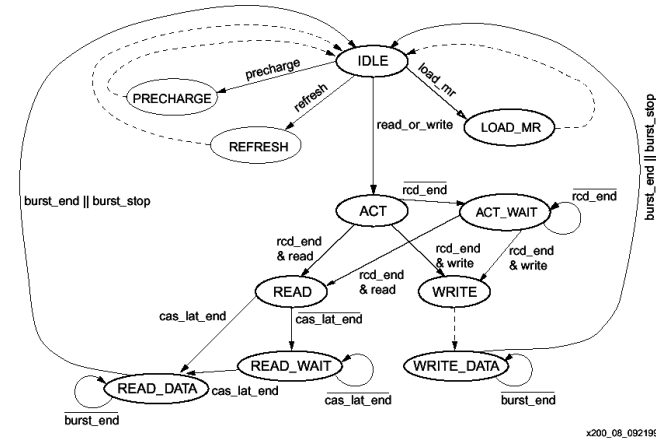


Figure 8: State Machine Diagram

An overview of the State Machine is shown in Figure 8. The dashed lines indicate an automatic sequence.

[Xilinx appnote]

SDRAM: DDR Write

Burst Write Operation

The burst write command is issued by having CS, CAS and WE Low while holding RAS High at the rising edge of the clock (CLK). The address inputs determine the starting column address. There is no write latency relative to DQS required for burst write cycle. The first data of a burst write cycle must be applied on the DQ pins T_{DS} (data-in setup time) prior to data strobe edge. The data strobe signal is enabled after T_{DQSS} from the rising edge of CLK issued by the WRITE command. The remaining data inputs must be supplied on each subsequent falling and rising edge of Data Strobe until the burst length is completed. When the burst has been finished, any additional data supplied to the DQ pins will be ignored.

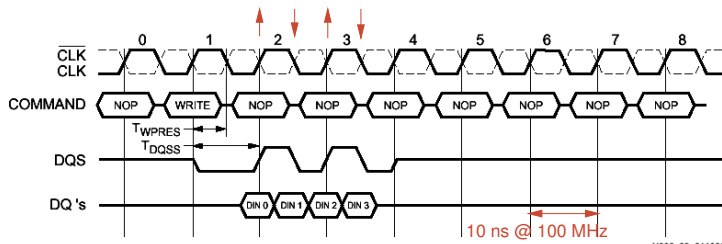
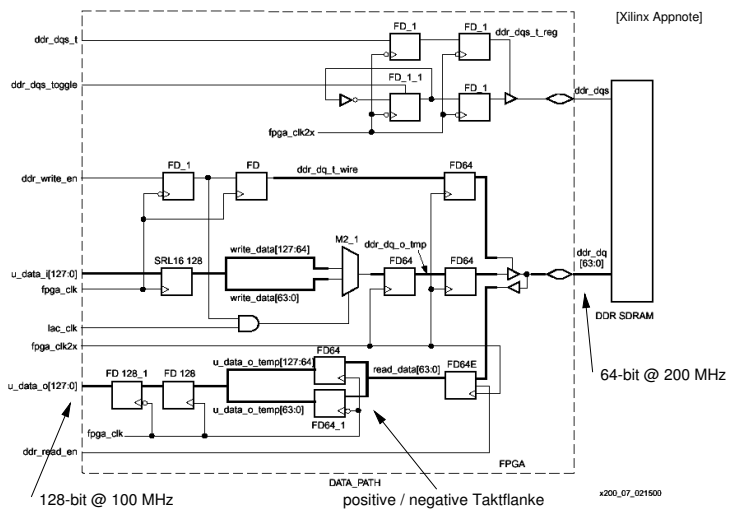


Figure 2: Burst Write Operation Timing

[Xilinx appnote]

SDRAM: DDR Datenpfad

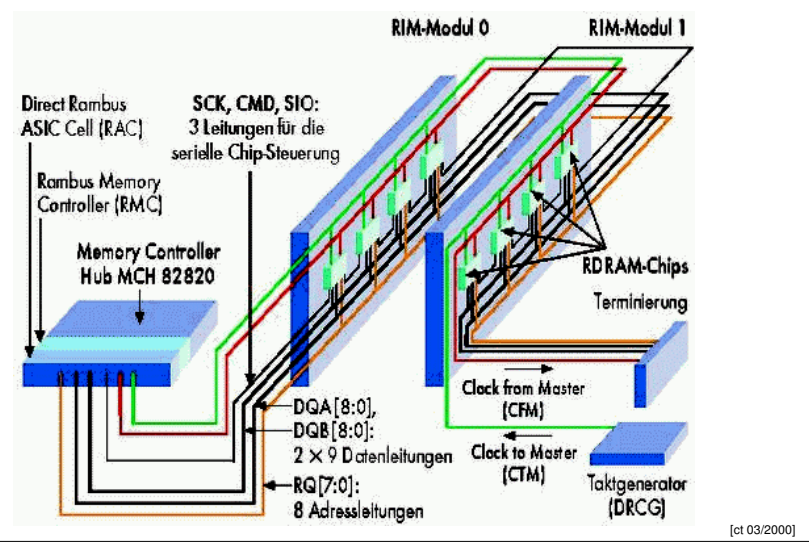


[Xilinx Appnote]

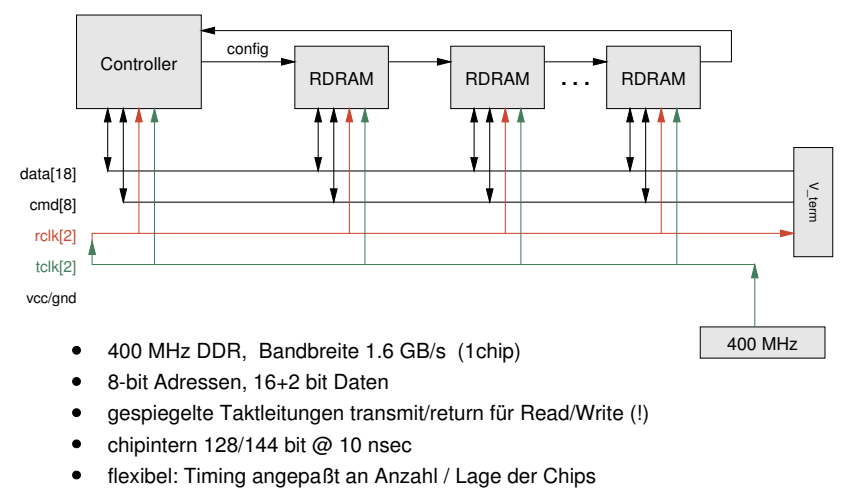
RAMBUS: Motivation

- steigende Anforderungen (etwa für 3D-Apps.)
 - immer mehr Speicherbandbreite erforderlich
 - sinkende Anzahl einzelner DRAM-Chips
-
- Bustakt (133 MHz) kaum weiter zu steigern
 - breitere Busse als 64 bit sehr teuer
 - Boards sollen minimale/maximale Bestückung vertragen
 - DDR problematisch, da Verzögerungen bereits ausgereizt
- => konventionelle Speichertechnik "am Anschlag"
- => RAMBUS
- timing-optimierter Bus (266 .. 400 MHz DDR)
 - wenige Leitungen (18 data + 8 cmd + 4 clock + vcc + gnd)
 - flexible Bestückung (N64/PSX2: nur je 2 Chips)

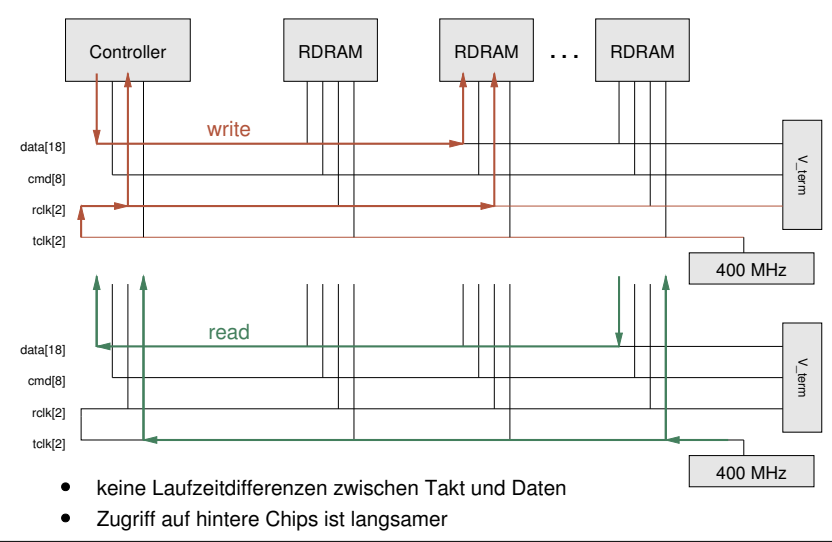
Rambus: Prinzip



RAMBUS: Konzept



RAMBUS: Read/Write



RAMBUS: signal delay matching

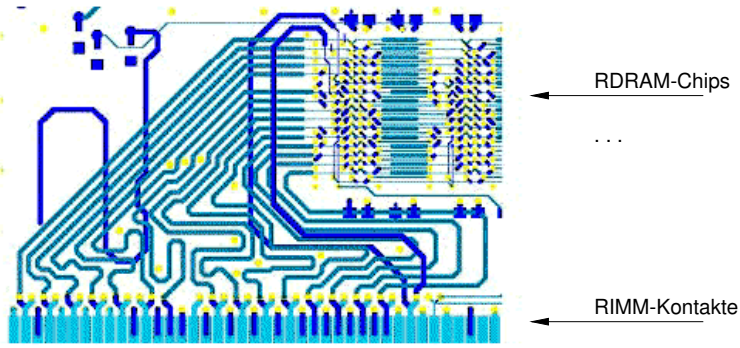
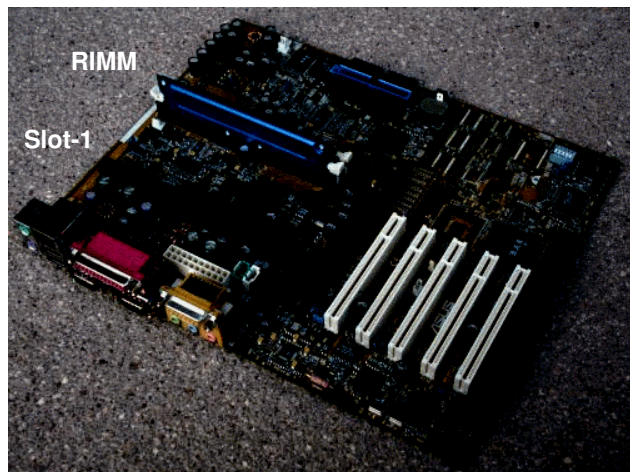


Figure 5-10: Delay Matching (Left Side)

- Leitungslängen angepasst für einheitliche Laufzeiten
- 800 MHz / 1.25 nsec / ~ 18 cm 0.5 nsec / ~ 7 cm

RAMBUS: Asus P3C (i820)



RAMBUS: basic read / write transactions

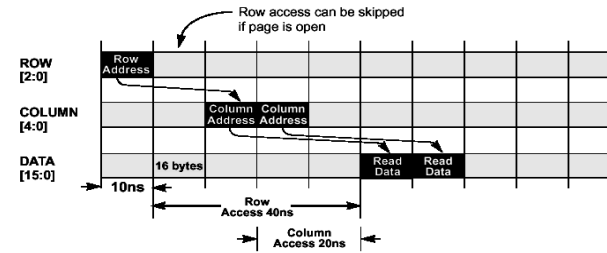


Figure 6: Read Transaction

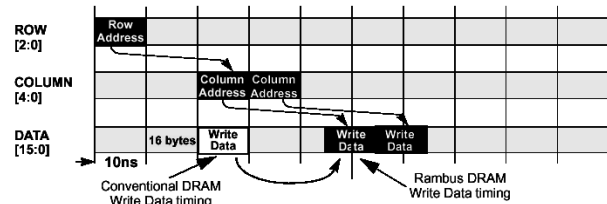


Figure 7: Write Transaction

RAMBUS: basic read / write transactions

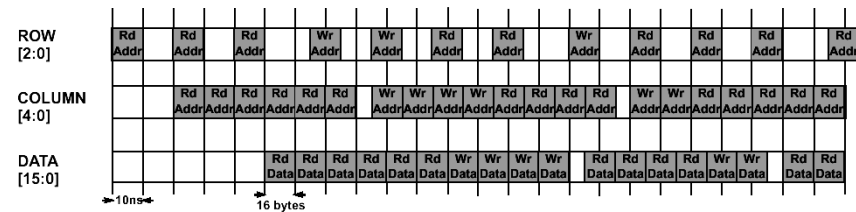


Figure 8: Simultaneous Pipelined Transaction

- separate Steuerleitungen für Row / Column-Select
- ermöglicht Pipelining von Lese- und Schreibzugriffen
- Datenleitungen im Idealfall fast 100 % ausgelastet
- aber nur mit geeigneten Zugriffen (32-Byte Ausrichtung)
- Performance Compiler-abhängig

RAMBUS: Read

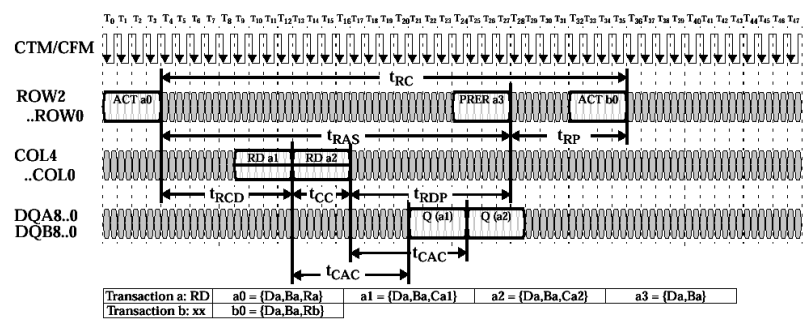


Figure 15: Read Transaction Example

- $t_{RCD} / t_{CC} / t_{CAC}$ abhängig vom Modul (PC-800 / PC-700 / usw.)
- zusätzliche Latenztakts für "hintere" Module
- zusätzliche Latenztakts zur Temperaturregelung
- (1 Chip reicht für volle Datenrate => höhere Belastung als bei SDRAM)

RAMBUS: Write

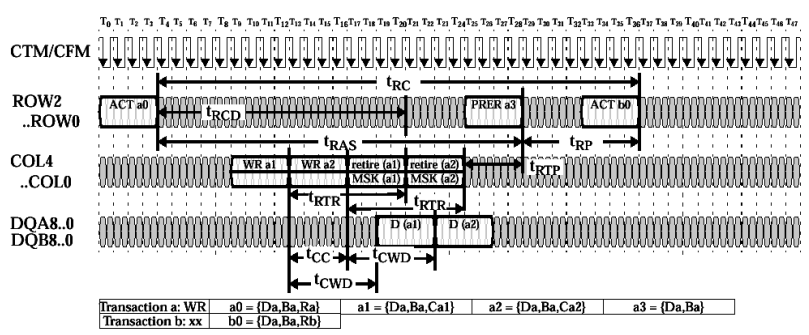


Figure 16: Write Transaction Example

- $t_{RCD} / t_{CC} / t_{CAC}$ abhängig vom Modul (PC-800 / PC-700 / usw.)
- zusätzliche Latenztakts für "hintere" Module
- zusätzliche Latenztakts zur Temperaturregelung
- (1 Chip reicht für volle Datenrate => höhere Belastung als bei SDRAM)

RAMBUS: Interleaved Write

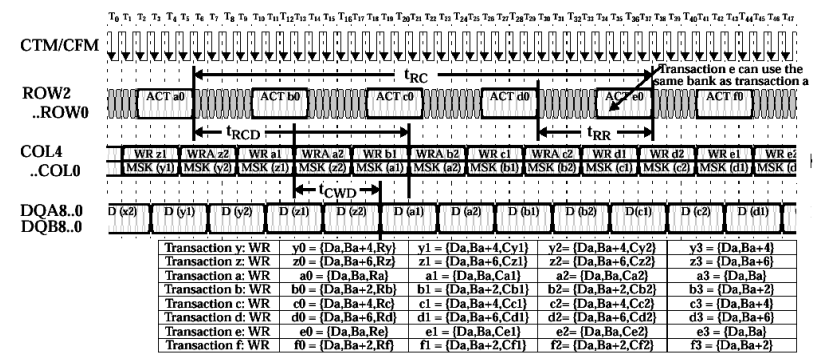
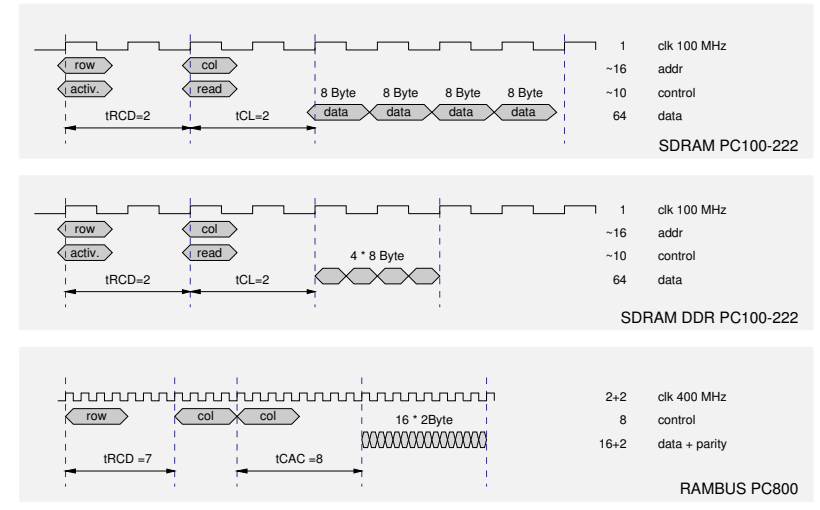


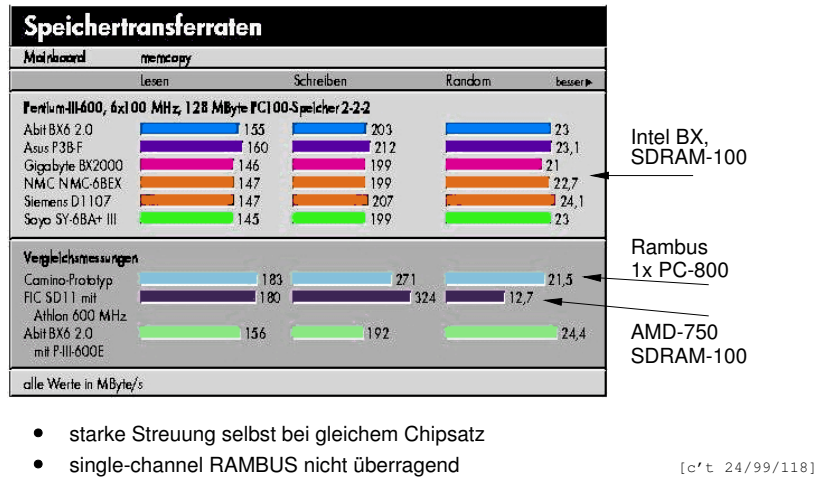
Figure 20: Interleaved Write Transaction with Two Dualoct Data Length

- entsprechend komplexe Zyklen auch für Read
- zusätzliche Buszyklen für Refresh / Powermanagement / usw.

RAMBUS: vs. SDRAM / SDRAM-DDR

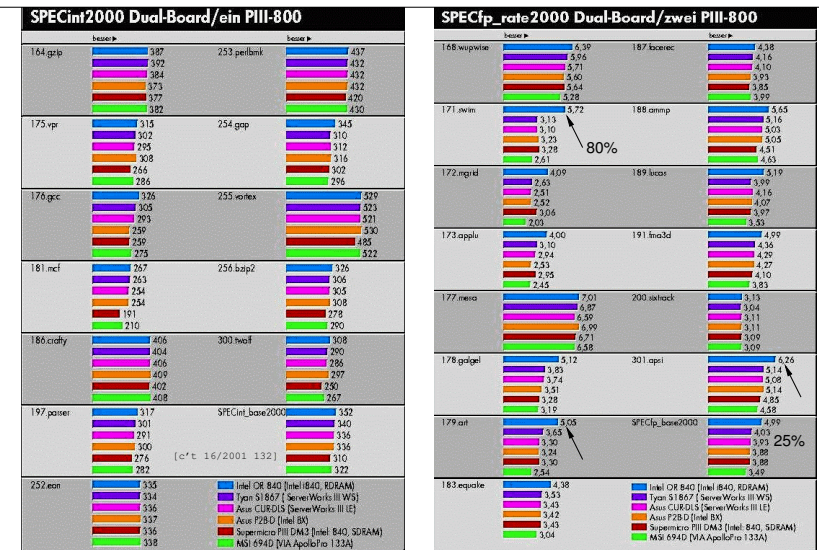


RAMBUS: c't memcopy

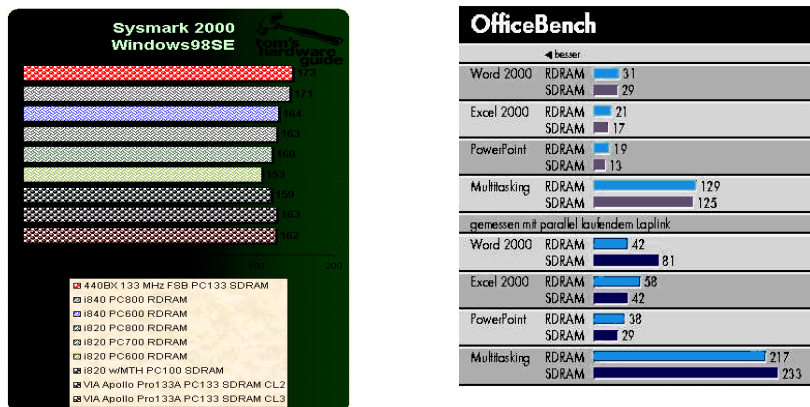


- starke Streuung selbst bei gleichem Chipsatz
- single-channel RAMBUS nicht überragend

RAMBUS: SPECint 2000 / SPECfp 2000



RAMBUS: Office Benchmark



- Speicher überhaupt gefordert ?!
- SDRAM 133 besser als RDRAM ?!

[www.tomshardware.com]
[c't / www.dell.com]

RAMBUS: Fazit ...

lohnt die neue, teure Technik?

- interessantes und flexibles Konzept
- ein Chip reicht für volle Datenrate: geeignet für 1 Gbit Generation
- volle Autokonfiguration und adaptives Bustiming
- widersprüchliche Benchmark-Ergebnisse
- single-channel RDRAM-800 kaum besser als SDRAM-133
- dual-channel RDRAM-800 teuer aber gut
- Rolle von SDRAM-DDR ?!
- derzeit nur Markenmodule, keine "no name" Billigware
- Preise bisher nicht konkurrenzfähig
- neueste Intel-Roadmap "unklar" (developer forum, Feb'00)
RDRAM (desktop) + SDRAM (mobile, server) + advanced DRAM

Cache: Motivation

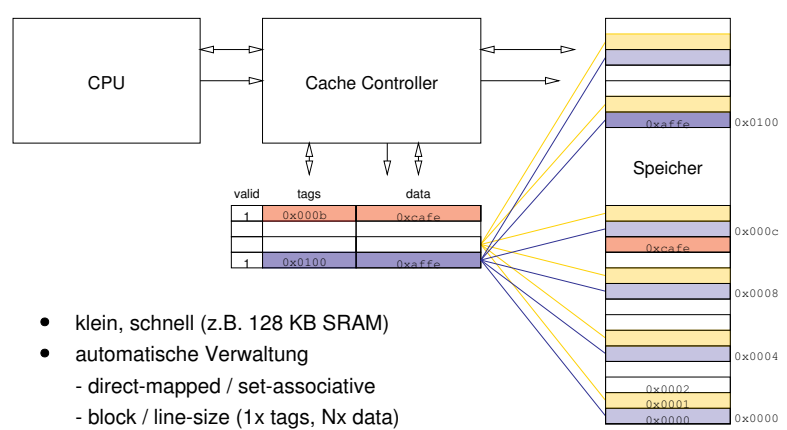
- DRAM langsam, SRAM teuer
- Lokalität: Daten mehrfach genutzt / benachbarte Daten genutzt

=> "Cache"

- kleiner SRAM-Zwischenspeicher
- Cache-Treffer laufen mit SRAM-Performance
- aber Overhead: Misses langsamer als ohne Cache

Parameter:	Beispiel
• Grösse	64 KByte
• hit-time	1 clk
• miss-time	50 clk
• miss-rate	1%
• Organisation	voll-assoziativ

Cache: Prinzip



- Klein, schnell (z.B. 128 KB SRAM)
- automatische Verwaltung
 - direct-mapped / set-associative
 - block / line-size (1x tags, Nx data)
 - write-through / -back / -allocate
- Vergleich der Tags, abhängig davon Cache- oder Speicherzugriff

Cache: Parameter . . .

- Gesamtgröße, Blockgröße, Zugriffszeit, Miss-Zeit, ...
 - Organisation: wo kann ein Block platziert werden? (direct-mapped)
 - Zugriff: wie wird ein Block gefunden? (tags, valid bit)
 - Ersetzung: welcher Block wird beim Miss ersetzt? (random, LRU)
 - Schreib-Strategie: write back / write through / ... (dirty bit, ...)
 - Architektur: separate I/D oder unified Cache?
-
- größere Blocks (weniger Verdrängung, aber geringere Kapazität)
 - höhere Assoziativität (aber komplexere Verwaltung)
 - Victim-Caches (billig und effizient)
 - HW-Prefetching (z.B. instruction prefetch / branch prediction)
 - Compiler-Prefetching (bei bekannten Zugriffsmustern)
-
- critical word first (x86: von Intel patentiert)
 - write buffer (alle aktuellen Prozessoren)
 - nonblocking caches (effizient, aber komplex)

Cache: AlphaServer 8200 (300MHz 21164)

memory type	size	location	latency		bandwidth
	KB		ns	cycles	
I-Cache	8K	on chip	6.6	2	4800
D-Cache	8K	on chip	6.6	2	4800
L2-Cache	96K	on chip	20.0	6	4800
L3-Cache	~ 4M	off chip	26.0	8	960
main memory	64M .. 4G	off chip	253.0	76	1200
single DRAM	16M	off chip	~ 60.0	18	30..100

Program	CPI	misses / 1000 instr.					% time spent in				
		I	D	L2	L3	μP	I	D	L2	L3	
SPECint92	1.2	7	25	11	0	0.78	0.03	0.13	0.05	0.00	
SPECfp92	1.2	2	47	12	0	0.68	0.01	0.23	0.06	0.02	
database	3.6	97	82	119	13	0.23	0.16	0.14	0.20	0.27	
sparse	3.0	0	38	36	23	0.27	0.00	0.08	0.07	0.58	

[Patterson 97]

Cache: Missrate

Missrate Beispiele: (SPEC 92, R2000, direct-mapped, 32-byte blocks)

Size	Instruction	Data	Unified
1K	3.06%	24.61%	13.34%
2K	2.26%	20.57%	9,78%
4K	1.78%	15.94%	7.24%
8K	1.10%	10.19%	4.57%
16K	0.64%	6.47%	2.87%
32K	0.39%	4.82%	1.99%
64K	0.15%	3.77%	1.35%
128K	0.02%	2.88%	0.95%

- Werte sehr stark programmabhängig
- CPU / Multiuser-Auslastung / Messzeit / ...

[H&P p.384]

PC-Technologie | SS 2001 | 18.214

Cache: Missrate: Beispiel

- Speicherzugriffe: 75% Instruction, 25% Data
- avg. memory access time
= hit time + (miss rate * miss penalty)
- cache hit: 1 clock
- cache miss: 50 clocks

Size	Instruction	Data	Unified
1K	3.06%	24.61%	13.34%
2K	2.26%	20.57%	9,78%
4K	1.78%	15.94%	7.24%
8K	1.10%	10.19%	4.57%
16K	0.64%	6.47%	2.87%
32K	0.39%	4.82%	1.99%
64K	0.15%	3.77%	1.35%
128K	0.02%	2.88%	0.95%

16K I + 16K D Cache:

- miss rate: $(75\% * 0.64\%) + (25\% * 6.47\%) = 2.10\%$
- tmac: $75\% * (1 + 0.64\% * 50) + 25\% * (1 + 6.47\% * 50) = (75\% * 1.32) + (25\% * 4.235) = 2.05$

32K unified Cache: load/store hit: 1 extra cycle (one port only)

- miss rate: 1.99%
- tmac: $75\% * (1 + 1.99\% * 50) + 25\% * (1 + 1 + 1.99\% * 50) = (75\% * 1.995) + (25\% * 2.995) = 2.24$

[H&P p.385]

=> split I/D Cache ist schneller (für dieses Beispiel)

PC-Technologie | SS 2001 | 18.214

Cache: Compulsory / Capacity / Conflict

3 Arten Cache-Misses:

- compulsory (cold start / first reference)
erster Zugriff auf einen Block
- capacity Cache zu klein für alle benötigten Blöcke;
Blöcke müssen ausgetauscht werden
=> Cache vergrößern
- conflict (collision misses / interference misses)
bei direct mapped / set associative Caches:
mehrere Blöcke im gleichen Set benötigt
=> Organisation verbessern, etwa 4fach assoz.
=> victim buffers

PC-Technologie | SS 2001 | 18.214

Cache: direct-mapped conflict misses

```
static void filterF(char* in1, char* out1)
{
    register int i0,i1,i2;
    register int x, int y;
    register char *in,*out;
    in = in1;
    out = out1;
    for( y=0; y < YRES; y++ ) {
        i0 = (int)in[0];
        i1 = (int)in[1];
        /* ignore boundary pixels, over/underflow for this benchmark */
        for( x=1; x < XRES-1; x++ ) {
            i2 = (int)in[x+1];
            out[x] = (char)( (i0 + (2*i1) + i2) / 4 );
            i0 = i1; i1 = i2;
        }
        in += XRES;
        out += XRES;
    }
} /*filterF*/
```

execution time via array size: [comp.arch posting]

SYS	511	512	513	1023	1024	1025	2047	2048	2049
CRIM	0.2	0.3	0.2	0.8	7.3*	0.9	3.7	33.4*	3.4 D
INDIGO4K	0.2	0.3	0.2	0.8	9.4*	0.8	3.2	37.9*	3.2 D
IN4K-fix	0.2	0.2	0.2	0.8	0.8	0.8	3.3	3.2	3.2 D
HP 720	0.3	0.7	0.3	1.1	2.7*	1.0	4.2	10.8*	4.2 D
HP 735	0.1	0.6*	0.1	0.6	2.7*	0.6	2.4	11.1*	2.6 D
HP 735	0.1	0.7*	0.1	0.6	2.7*	0.6	2.2	10.8*	2.2 D
Gwy486-66	0.3	0.3	0.3	1.3	1.4	1.3	5.5	5.5	5.5 SA?

PC-Technologie | SS 2001 | 18.214

x86: Pentium III Caches. . .

Cache or Buffer	Characteristics
L1 Instruction Cache ¹	- P6 family and Pentium [®] processors: 8 or 16 KBytes, 4-way set associative, 32-byte cache line size; 2-way set associative for earlier Pentium [®] processors. - Intel486™ processor: 8 or 16 KBytes, 4-way set associative, 16-byte cache line size, instruction and data cache combined.
L1 Data Cache ¹	- P6 family processors: 16 KBytes, 4-way set associative, 32-byte cache line size; 8 KBytes, 2-way set associative for earlier P6 family processors. - Pentium [®] processors: 16 KBytes, 4-way set associative, 32-byte cache line size; 8 KBytes, 2-way set associative for earlier Pentium [®] processors. - Intel486™ processor: (see L1 instruction cache).
L2 Unified Cache ^{2,3}	- P6 family processors: 128 KBytes, 256 KBytes, 512 KBytes, 1 MByte, or 2 MByte, 4-way set associative, 32-byte cache line size. - Pentium [®] processor: System specific, typically 256 or 512 KBytes, 4-way set associative, 32-byte cache line size. - Intel486™ processor: System specific.
Instruction TLB (4-KByte Pages) ¹	- P6 family processors: 32 entries, 4-way set associative. - Pentium [®] processor: 32 entries, 4-way set associative; fully set associative for Pentium [®] processors with MMX™ technology. - Intel486™ processor: 32 entries, 4-way set associative, instruction and data TLB combined.
Data TLB (4-KByte Pages) ¹	- Pentium [®] and P6 family processors: 64 entries, 4-way set associative; fully set associative for Pentium [®] processors with MMX™ technology. - Intel486™ processor: (see Instruction TLB).
Instruction TLB (Large Pages)	- P6 family processors: 2 entries, fully associative - Pentium [®] processor: Uses same TLB as used for 4-KByte pages. - Intel486™ processor: None (large pages not supported).
Data TLB (Large Pages)	- P6 family processors: 8 entries, 4-way set associative. - Pentium [®] processor: 8 entries, 4-way set associative; uses same TLB as used for 4-KByte pages in Pentium [®] processors with MMX™ technology. - Intel486™ processor: None (large pages not supported).
Write Buffer	- P6 family processors: 12 entries. - Pentium [®] processor: 2 buffers, 1 entry each (Pentium [®] processors with MMX™ technology have 4 buffers for 4 entries). - Intel486™ processor: 4 entries.

x86: Pentium III Cache-Modi

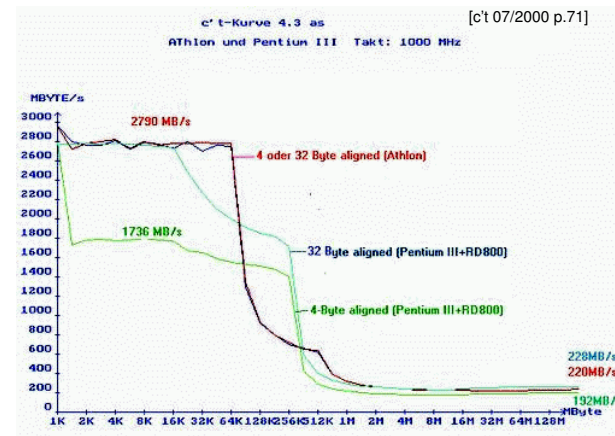
Caching Method	P6 Family Processors	Pentium [®] Processor	Intel486™ Processor
Uncacheable (UC)	Yes	Yes	Yes
Write Combining (WC)	Yes ¹	No	No
Write Through (WT)	Yes	Yes ²	Yes ²
Write Back (WB)	Yes	Yes ²	No
Write Protected (WP)	Yes ¹	No	No

NOTES:

- Requires programming of MTRRs to implement.
- Speculative reads not supported.

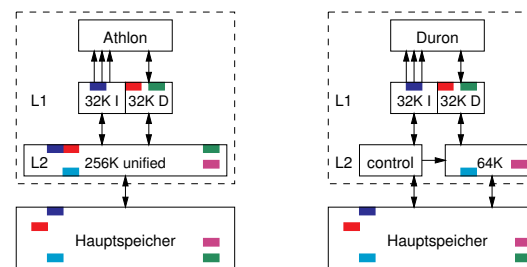
- Write Combining (WC)—System memory locations are not cached (as with uncacheable memory) and coherency is not enforced by the processor's bus coherency protocol. Speculative reads are allowed. Writes may be delayed and combined in the write buffer to reduce memory accesses. The writes may be delayed until the next occurrence of a buffer or processor serialization event, e.g., CPUID execution, a read or write to uncached memory, interrupt occurrence, LOCKed instruction execution, etc. if the WC buffer is partially filled. This type of cache-control is appropriate for video frame buffers, where the order of writes is unimportant as long as the writes update memory so they can be seen on the graphics display. See Section 9.3.1., "Buffering of Write Combining Memory Locations", for more information about caching the WC memory type. The preferred method is to use the new SFENCE (store fence) instruction introduced in the Pentium[®] III processor. The SFENCE instruction ensures weakly ordered writes are written to memory in order, i.e., it serializes only the store operations.

x86: ctkurve



- Messung der Cache-Transferrate vs. Blockgröße (random)
- Caches deutlich sichtbar: Pentium 16K/256K, Athlon 64K/512K

x86: AMD Duron: Cache

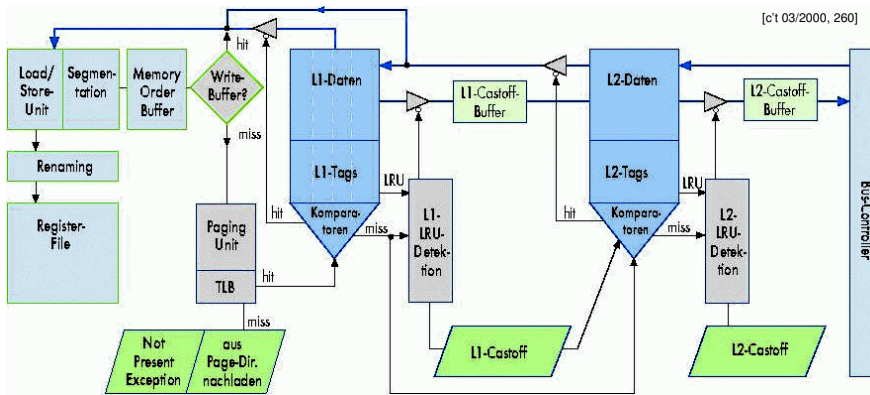


AMD Duron "exclusive" L2-Cache:

=> vgl. "victim buffer"

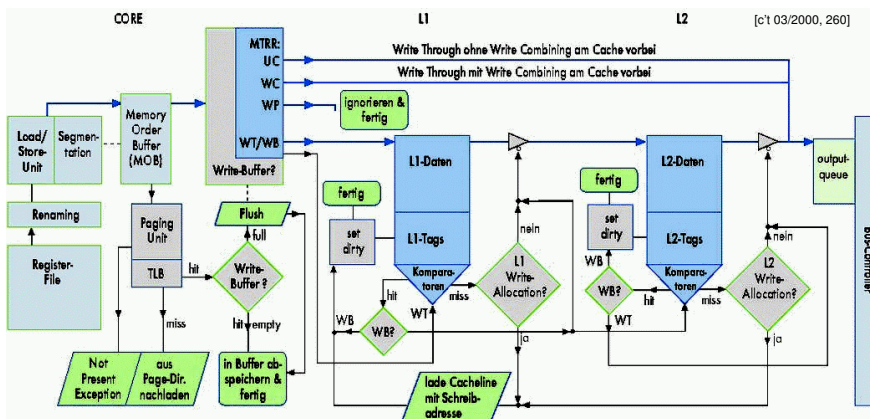
- L1-Cache: wie im Athlon (32KB + 32KB)
- L2-Cache: nur 64 KB statt 256 KB
- wäre bei herkömmlicher Verwaltung sinnlos (alle Daten doppelt)
- daher: L2-Cache speichert nur Daten, die nicht im L1 sind
- nur ca. 10% Performanceverlust

x86: Pentium II Lesezugriff...



- Cachezugriffe: L1 typ. 1..2 Takte, L2 typ. 2..10 Takte
- Speicherzugriffe: ca. 100 Takte

x86: Pentium II Schreibzugriff...



- MemoryTypeRangeRegister: schnelle I/O, z.B. Graphikkarte
- weitere Stufen (z.B. AGP GART) im Chipsatz ...

Speicherhierarchie: Übersicht, typ. Werte

	TLB	L1-Cache	L2-Cache	Virtueller Speicher
size / byte	32 .. 8K	1 .. 128K	256K .. 16M	16M .. 8G
block size / byte	4..8	4..32	32..256	4K..16K
hit time / clk	1	1..2	6..15	10..100
miss penalty / clk	10..30	8..66	30..200	700K..6M
miss rate / %	0.1- 2	0.5 .. 20	15 .. 30	0.000001 .. 0.001
backup	L1	L2	DRAM	Disks
block placement	FA	DM	DM / SA	FA
block identification	tags	tags	tags	table
block replacement	random	-	random	~ LRU
write strategy	flush	WT / WB	WB	WB

FA/SA/DM = full/set associative/direct mapped
WB/WT = write back/write through

Speicherhierarchie: Fazit

- performance gap wächst und wächst
- DRAM inhärent langsam

=> Speicherhierarchie wird immer wichtiger

- größere, tiefere Caches
- komplexere Caches: voll assoziativ, non-blocking, etc.
- aber Nutzen nur für "einfache" Anwendungen

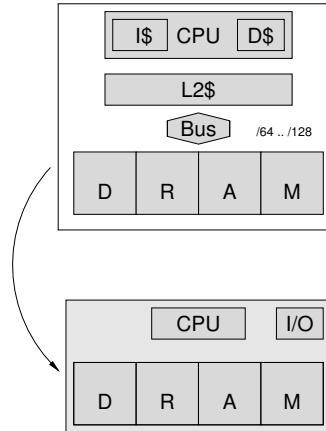
=> wichtige Forschungsaufgaben:

- intelligenteres Cache-Management
- Prefetching
- computational RAM / IRAM / ...

IRAM: Konzept / "Vision"

IRAM := $\mu P + DRAM + I/O$ auf einem Chip

- Performance gap CPU/Speicher schließen:
 Latenz 5-10x 20ns statt 200ns
 Bandbreite 100x TB/s
- Speicherorganisation anpassen:
 beliebig wählbar: #bits, Busbreite, ...
- Energieverbrauch senken:
 kein DRAM-Bus: 2-4x
- Platzverbrauch senken:
 CPU passt auf DRAM: 2-4x



IRAM: Motivation

mehrfache Motivation:

- Stromverbrauch, Platzbedarf - insbesondere für mobile Geräte
- Anpassung von Speicherbedarf und -organisation
- Performance gap zwischen Prozessor und DRAM schließen, minimale Latenz, maximale On-chip Bandbreite
- neue Marktstrategie für DRAM-Produzenten (wegen Preisverfall)

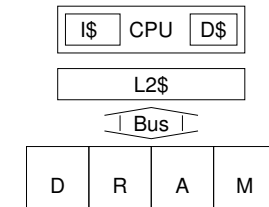
Alternativen?

- neue, revolutionäre Chip-Packaging Technologien - unwahrscheinlich
- komplexere CPUs (out-of-order, multiple-issue, ...) - schwierig
- neue DRAM Standards (SDRAM, RAMBUS, ...) - nicht in Sicht

IRAM: performance gap "Tax"

Caches: kein Wert an sich, nur zum Schließen des performance gap

Beispiele:	%Fläche (~Kosten)	%Transistoren (~Leistung)
• Alpha 21164	37%	77%
• ARM SA110	61%	94%
• Pentium Pro	64%	88%



Patterson: performance gap "tax"

IRAM: Architektur?

1G Transistoren möglich, aber welche Rechnerarchitektur?

- ein Prozessor + DRAM:
 - Nutzen fraglich, evtl. langsamer als optimierte CPU + Cache + DRAM
 - verschenkt hohe on-chip Bandbreite, da #issues < 8
 - wenig innovativ
- SIMD oder MIMD Parallelrechner?
 - viele Prozessoren, aber nur wenig RAM / Prozessor
 - Programmierung ist ungelöstes Problem
 - alle bisherigen Varianten gescheitert
- Graphikprozessoren - bereits am Markt und etabliert
- I-VRAM := DRAM + RISC + Vektorrechner

[Berkeley IRAM group]

IRAM: "vanilla" approach?!

vorhandenen Rechner (Alpha 21164) in DRAM Technologie implementieren

- gleiche Architektur: gleiche Caches, einfaches DRAM, ...
- übliche Benchmarks simulieren

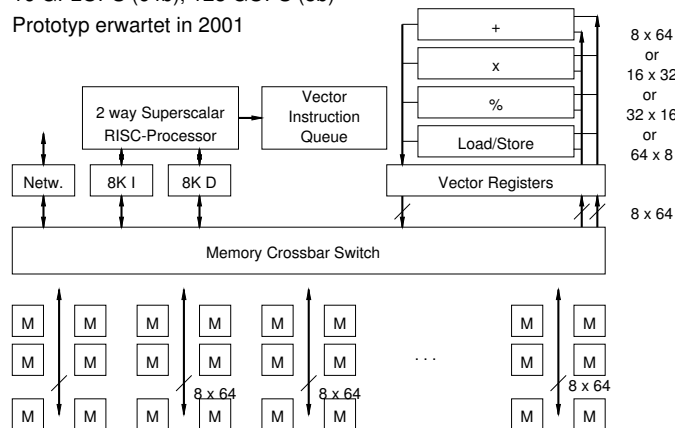
Logik in DRAM Prozeß? Faktoren: (optimistisch - pessimistisch)

- | | | |
|---------------------------|------------|------------|
| • Logik langsamer | 1.3 - 2.0 | |
| • SRAM (Caches) langsamer | 1.1 - 1.3 | |
| • DRAM schneller | 10.0 - 5.0 | |
| • SPEC92 | 0.8 - 0.6 | langsamer! |
| • Database | 1.1 - 0.9 | gleich |
| • Sparse matrix | 1.8 - 1.2 | schneller |

Performance nicht überzeugend, aber Leistung/Platzbedarf/Kosten besser

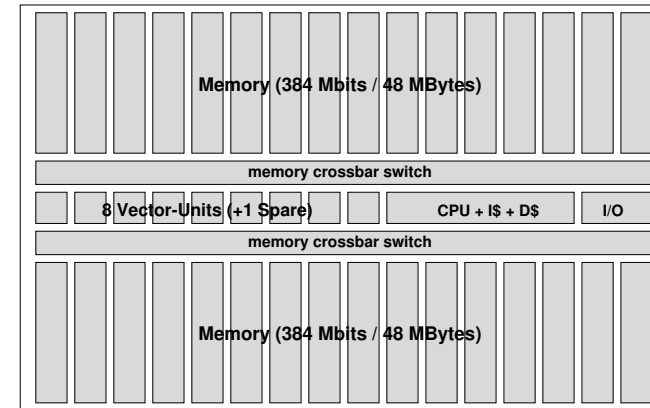
IRAM: V-IRAM 2

- 0.18 µm, fast logic, 1 GHz, 96 MByte DRAM
- 16 GFLOPS (64b), 128 GOPS (8b)
- Prototyp erwartet in 2001



IRAM: V-IRAM 2 Floorplan

- 0.18 µm, 1G Transistoren: 80% DRAM, 4% Vector, 3% CPU
- Größe und Redundanz wie 1Gb DRAM



IRAM: Zusammenfassung

Moore's Law: 1% / Woche

- Engpaß ist Performance gap zwischen CPU und DRAM
- radikal neue Speichertechnologien zunächst unwahrscheinlich
- Technologie ermöglicht CPU und DRAM auf einem Chip ab 1998/1999

IRAM Potential

- Bandbreite 100x, Latenz 5-10x, Leistung 2-4x
- V-IRAM als Technologiedemo? (Graphikchips bereits lieferbar!)
- V-IRAM: 25-100MB Speicher @ 20ns, 4-16 GFLOPS, serielle I/O
- V-IRAM: 1 TB/s Bandbreite, Smart-SIMMs = TFLOPS

dramatische Auswirkungen auf Halbleiter-Markt

- wer liefert DRAM, wer liefert Mikroprozessoren?

Parallelrechner: Motivation

immer höhere Performance gefordert

=> schnellere Einzelprozessoren
aber Takte oberhalb von 10 GHz unrealistisch

=> mehrere Prozessoren

- diverse Architekturkonzepte
- shared-memory vs. message-passing
- Overhead durch Kommunikation
- Programmierung ist ungelöstes Problem

derzeit beliebtester Kompromiss:

- bus-basierte SMPs mit 2-16 Prozessoren

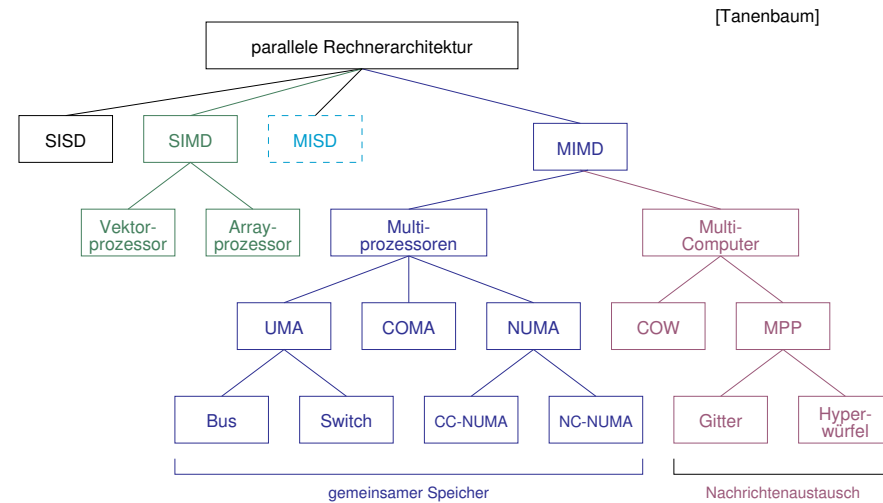
Parallelrechner: Literatur

Tanenbaum, Computerarchitektur (4. Auflage)
Hennessy & Patterson, computer architecture

Messmer, PC-Hardwarebuch
Intel Pentium Manual

Intel ITJ (ASCI red)
diverse c't-Artikel, insbesondere Benchmarks

Parallelrechner: Klassifikation

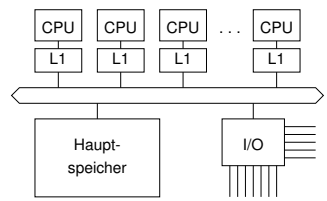


Parallelrechner:

- Programmierung ist ungelöstes Problem
- Aufteilung der Programme auf CPUs/nodes?
- insbesondere bei komplexen Kommunikationsnetzwerken
- Parallelität typischer Programme (gcc, spice, ...): kleiner 8
- massiv parallele Rechner sind dann Verschwendung
- aber SMP-Lösungen mit 4..16 Prozessoren attraktiv
- Datenbankanwendungen oft gut parallelisierbar
- z.B. je ein Thread/Prozeß pro Anfrage
- Vektor/Feld-Rechner für Numerik, Simulation
- Supercomputer derzeit nur für Numerik / Militär
- ansonsten "kleine" SMP-basierte Rechner

SMP: "Symmetric multiprocessing"

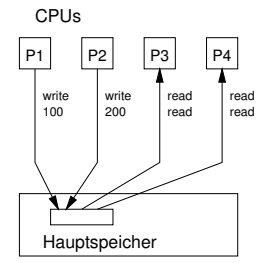
- mehrere Prozessoren teilen gemeinsamen Hauptspeicher
- Zugriff über Verbindungsnetzwerk oder Bus
- geringer Kommunikationsoverhead
- bus-basierte Systeme sind sehr kostengünstig
- aber schlecht skalierbar (Bus wird Flaschenhals)
- lokale Caches für gute Performance notwendig
- MESI-Protokoll und Snooping für Cache-Kohärenz



SMP: Eigenschaften ...

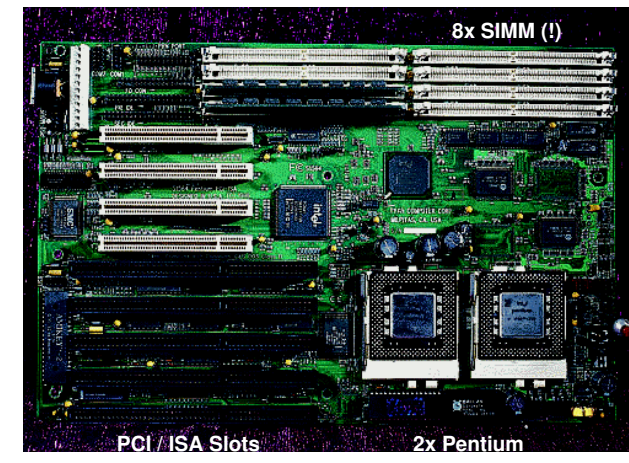
"symmetric multiprocessing":

- alle CPUs gleichrangig, Zugriff auf Speicher und I/O
- gleichzeitiger Zugriff auf eine Speicheradresse?
- strikte / sequentielle / Prozessor- / schwache Konsistenz

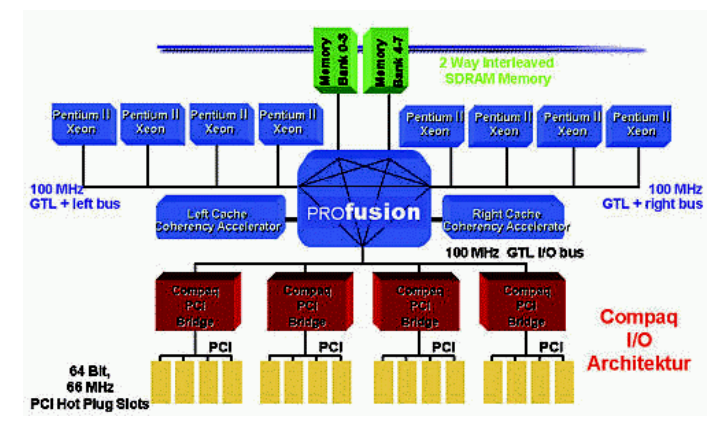


W1 100	W1 100	W2 200
W2 200	R3 = 100	R4 = 200
R3 = 200	W2 200	W1 100
R3 = 200	R3 = 200	R3 = 100
R4 = 200	R4 = 200	R4 = 100
R4 = 200	R4 = 200	R4 = 100

SMP: dual Pentium-Board (1998)



SMP: Pentium II (Compaq Profusion)



SMP: Cache-Kohärenz

aus Performancegründen:

- jeder Prozessor hat seinen eigenen Cache (L1, L2, ...)
- aber gemeinsamer Hauptspeicher

=> Problem: "Cache-Kohärenz"

Prozessor X greift auf Daten zu, die im Cache von Y liegen

- 1) Lesezugriff von X: Y muß seinen Wert liefern
- 2) Schreibzugriff von X: Y muß Wert von X übernehmen
- 3) was soll bei gleichzeitigem Zugriff passieren?!
(vgl. Java synchronized Konzept)

=> MESI-Protokoll mit Snooping

- Caches enthalten Wert, Tag, und 2-bit MESI-Zustand

SMP: MESI Konzept

MESI := modified, exclusive, shared, invalid

- jede Cache-Speicherstelle wird um 2 Statusbits erweitert
- alle Prozessoren überwachen die Zugriffe anderer Prozessoren
- entsprechende Aktualisierung der Statusbits

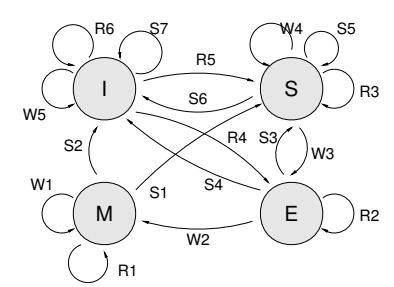
Zustand:	Bedeutung (grob):
invalid	Wert ist ungültig (z.B. noch nie geladen)
exclusive	gültiger Wert, nur in diesem Cache vorhanden
modified	gültiger Wert, nur in diesem Cache vorhanden, gegenüber Hauptspeicher-Wert verändert
shared	gültiger Wert, in mehreren Caches vorhanden

SMP: MESI Zustände

MESI-Zustand	Cache-Eintrag gültig?	Wert im Speicher gültig?	Kopien in anderen Caches?	Zugriff betrifft
M	ja	nein	nein	Cache
E	ja	ja	nein	Cache
S	ja	ja	möglich	Speicher
I	nein	unbekannt	möglich	Speicher

- Cache-Strategie: write-back, kein write-allocate
- Schreibzugriffe auf M führen nicht zu Bus-Transaktionen
- Werte in E stimmen mit Hauptspeicherwerten überein
- Werte in S sind aktuell, Lesezugriff ohne Bus-Transaktion
- Schreibzugriff auf S: lokal S, fremde auf I, Wert abspeichern
- mit write-through Caches: Zustände S/I, kein M/E

SMP: MESI Übergänge



- Lesezugriffe:
- M-M R1 Cache-Hit, CPU bekommt Daten
 - E-E R2 Cache-Hit, CPU bekommt Daten
 - S-S R3 Cache-Hit, CPU bekommt Daten
 - I-E R4 Miss, Speicher liefert Daten
 - I-S R5 Miss, externer Cache liefert Daten
 - I-I R6 Miss, Adresse nicht cacheable

- Schreibzugriffe:
- M-M W1 Hit, CPU aktualisiert Cache
 - E-M W2 Hit, CPU aktualisiert Cache
 - S-E W3 Hit (write-back): Cache aktualisiert, Buszyklus markiert fremde Kopien als invalid
 - S-S W4 Hit (write-through): Caches und Speicher aktualisiert
 - I-I W5 Miss, Speicher schreiben, aber kein write-allocate

- Snoop-Zyklen
- M-S S1 Hit, Speicher schreiben
 - M-I S2 Hit, Speicher schreiben
 - E-S S3 Hit, aber nicht modifiziert
 - usw.

SMP: MESI Snooping

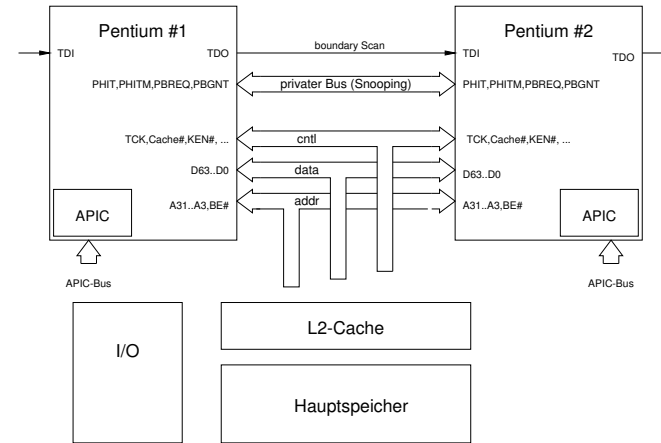
Snooping := "schnüffeln"

- alle Prozessoren überwachen alle Bus-Transaktionen
- Zugriffe auf "modified"-Werte werden erkannt:
 1. fremde Bus-Transaktion unterbrechen
 2. eigenen (=modified) Wert zurückschreiben
 3. Status auf shared ändern
 4. unterbrochene Bus-Transaktion neu starten
- erfordert spezielle Snoop-Logik im Prozessor
- garantiert Cache-Kohärenz aller Prozessoren
- optimale Performance

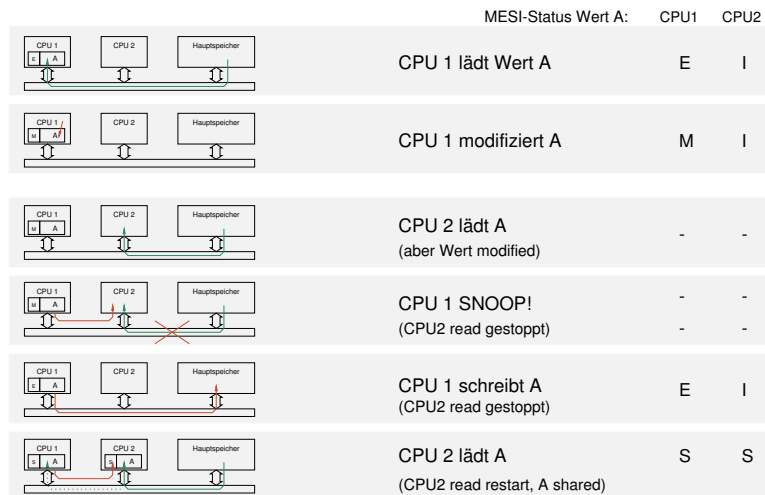
Beispiel: siehe nächste Folie

[PC-Hardwarebuch]

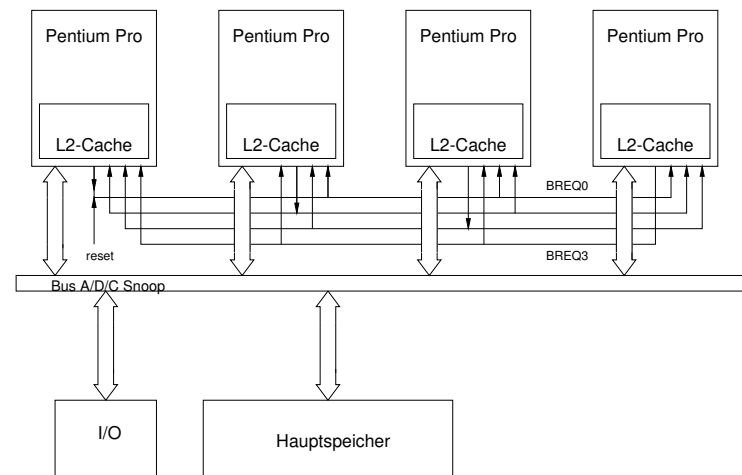
SMP: Pentium



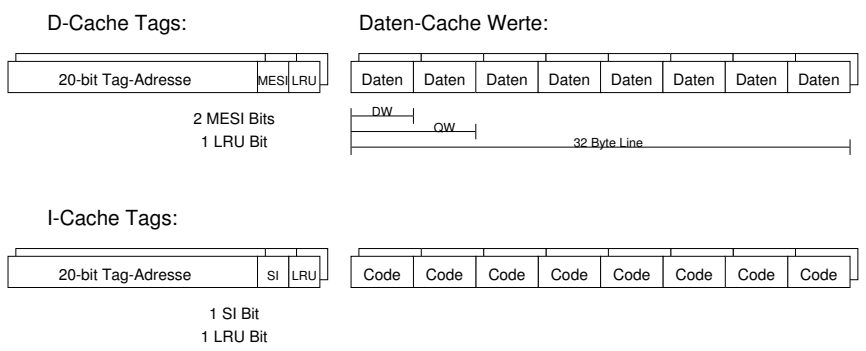
SMP: MESI Snooping: Beispiel



SMP: Pentium Pro

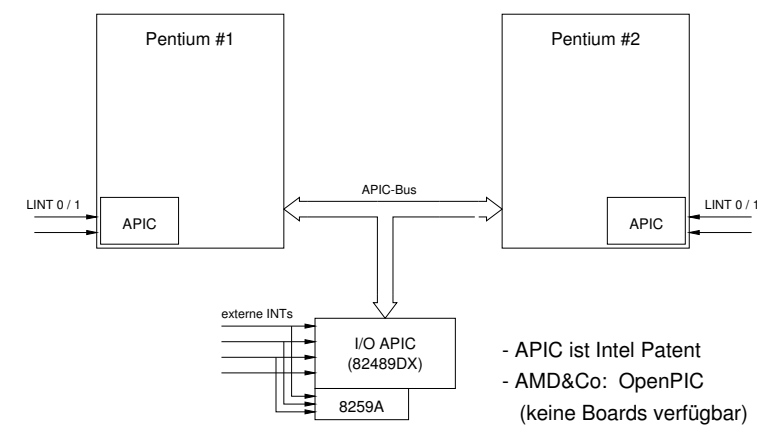


SMP: MESI Pentium



- 32-Byte Cache-Lines
- D-Cache unterstützt MESI, I-Cache nur SI
- externe Signale zeigen MESI-Übergänge an [PC-Hardwarebuch]

SMP: Pentium APIC



- APIC ist Intel Patent
- AMD&Co: OpenPIC (keine Boards verfügbar)

SMP: Interrupts

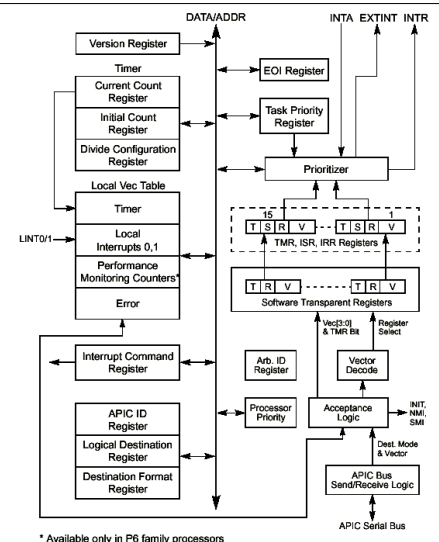
spezielle Interrupt-Behandlung in SMP-Rechner notwendig:

- welcher Prozessor soll einen Interrupt bearbeiten?
- statisch, z.B. immer der erste Prozessor
- der am wenigsten ausgelastete
- round-robin, oder ähnliche Strategien

Interrupt-Maskierung

- externe Folie

SMP: Pentium APIC



* Available only in P6 family processors

x86: locked atomic operations

7.1. LOCKED ATOMIC OPERATIONS

The 32-bit Intel Architecture processors support locked atomic operations on locations in system memory. These operations are typically used to manage shared data structures (such as semaphores, segment descriptors, system segments, or page tables) in which two or more processors may try simultaneously to modify the same field or flag. The processor uses three interdependent mechanisms for carrying out locked atomic operations:

- Guaranteed atomic operations.
- Bus locking, using the LOCK# signal and the LOCK instruction prefix.
- Cache coherency protocols that insure that atomic operations can be carried out on cached data structures (cache lock). This mechanism is present in the P6 family processors.

These mechanisms are interdependent in the following ways. Certain basic memory transactions (such as reading or writing a byte in system memory) are always guaranteed to be handled atomically. That is, once started, the processor guarantees that the operation will be completed before another processor or bus agent is allowed access to the memory location. The processor also supports bus locking for performing selected memory operations (such as a read-modify-write operation in a shared area of memory) that typically need to be handled atomically, but are not automatically handled this way. Because frequently used memory locations are often cached in a processor's L1 or L2 caches, atomic operations can often be carried out inside a processor's caches without asserting the bus lock. Here the processor's cache coherency protocols insure that other processors that are caching the same memory locations are managed properly while atomic operations are performed on cached memory locations.

Note that the mechanisms for handling locked atomic operations have evolved as the complexity of Intel Architecture processors has evolved. As such, more recent Intel Architecture processors (such as the P6 family processors) provide a more refined locking mechanism than earlier Intel Architecture processors, as is described in the following sections.

- notwendig für Multiprozessorsysteme

SMP: x86 Memory Type Range Registers

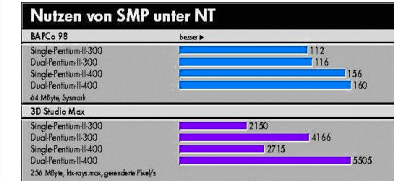
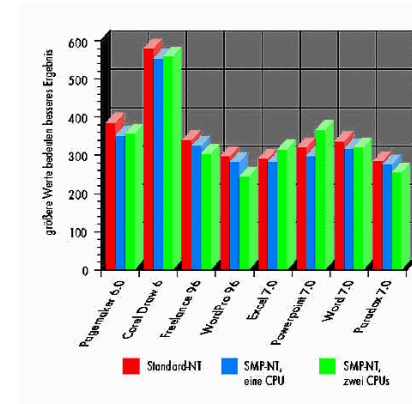
Table 9-6. MTRR Memory Types and Their Properties

Mnemonic	Encoding in MTRR	Cacheable in L1 and L2 Caches	Writeback Cacheable	Allows Speculative Reads	Memory Ordering Model
Uncacheable (UC)	0	No	No	No	Strong Ordering
Write Combining (WC)	1	No	No	Yes	Weak Ordering
Write-through (WT)	4	Yes	No	Yes	Speculative Processor Ordering
Write-protected (WP)	5	Yes for reads, no for writes	No	Yes	Speculative Processor Ordering
Writeback (WB)	6	Yes	Yes	Yes	Speculative Processor Ordering
Reserved Encodings*	2, 3, 7 through 255				

NOTE:

- * Using these encoding result in a general-protection exception (#GP) being generated.
- Register (Pentium+) zur Einstellung des Cache-Verhaltens
- Vorsicht mit aggressiven Optimierungen. . .

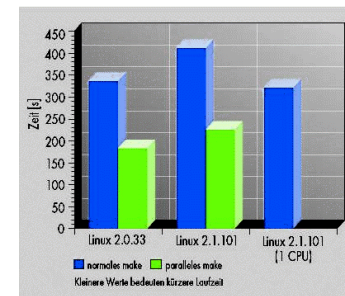
SMP: Windows NT Benchmarks



- fast kein Gewinn für die BAPCo
- 3D-Studio Max doppelte Perf.
- Verwaltungsoverhead ~ 10%

SMP: Quake3, Linux make

Mainboard	Prozessoren	Speicher	Betriebssystem, Treiber	Großkarte TNT2-Ultra, 32 MByte SDR-SDRAM		Großkarte GeForce, 32 MByte DDR-SDRAM	
				Demo001, fastest		Demo001, high quality	
				besser #	besser #	besser #	besser #
Asus P2B-D	1 x 800 MHz	1 x 128 MByte DIMM	Win 2000, 5.22	98,3	45,2	119,5	94,4
Asus P2B-D	2 x 800 MHz	1 x 128 MByte DIMM	Win 2000, 5.22	139,3	45,1	130,7	78,4
Intel ORB40	1 x 800 MHz	2 x 128 MByte RDRAM	Win 98 SE, 3.68	104,5	48,9	128,4	98,1
Intel ORB40	1 x 800 MHz	2 x 128 MByte RDRAM	Win 2000, 5.22	106,7	45,3	131,4	98,7
Intel ORB40	2 x 800 MHz	2 x 128 MByte RDRAM	Win 2000, 5.22	161,2	43,1	149,8	81,4



- Nutzen nur für geeignete Apps.
- evtl. seltsame Effekte (Quake)
- beträchtlicher OS-Overhead (in Win2K, Linux 2.4 besser)
- gut für Server-Aufgaben siehe Compaq "Piranha"

ASCI: Motivation

"Accelerated Strategic Computing Initiative", DOE seit ~1996

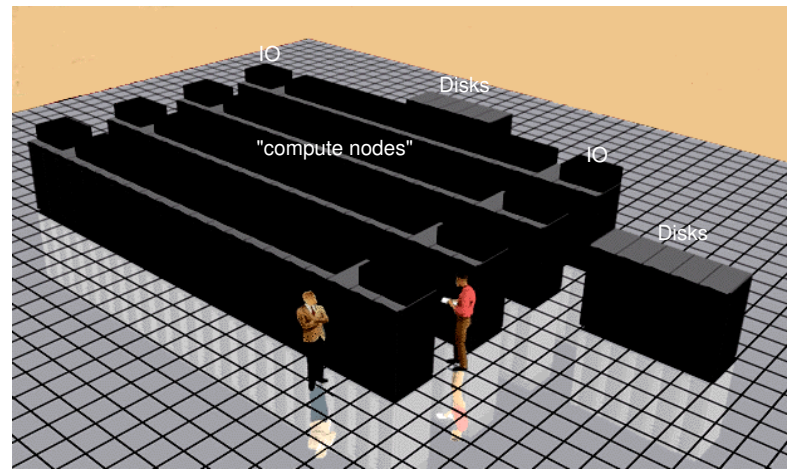
- Überalterung der Kernwaffenbestände
- Simulation notwendig wegen Teststopp-Verträgen ...
- und außerdem die "grand challenge" Anwendungen (QM, Wettervorhersage, finite-elements, ...)

=> Realisierung mehrerer Prototypen-Rechner für 1 TFlop
 => Bau eines 100 TFlops Rechners bis ca. 2002

- "option red" Intel, Sandia NL
9400 Prozessoren (PentiumPro/200), PC-Standardkomponenten
- "pacific blue" IBM, LLNL
- "mountain blue" SGI, Los Alamos NL

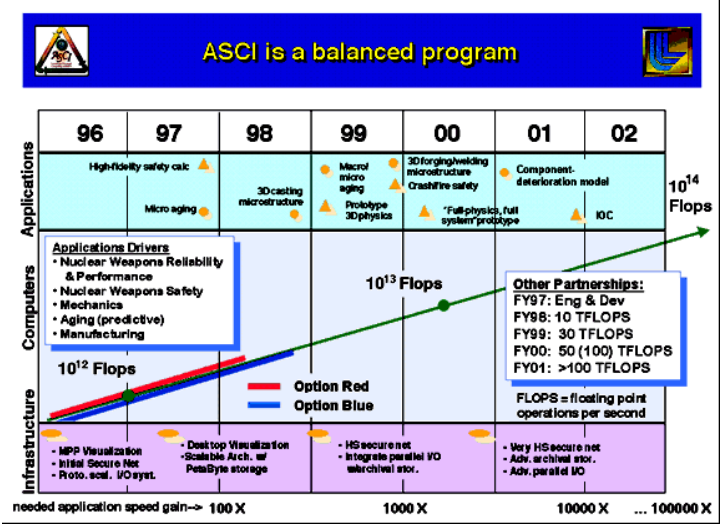
[www.sandia.gov/ASCI/]

ASCI red: (Intel 1997)



9216 P6-CPU's 594 GB RAM 1 TB Disk 1.0GB/s I/O 1.8 TFLOPS

ASCI: Roadmap



ASCI red: Photo



ASCII red: Architektur

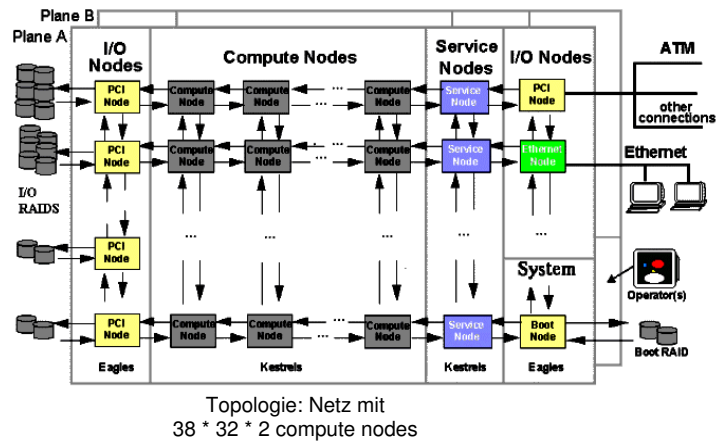


Figure 5: Logical System Block Diagram for the ASCII Option Red Supercomputer. This system uses a split-plane mesh topology and has 4 partitions: System, Service, I/O and Compute. Two different kinds of node boards are used and described in the text: the Eagle node and the Kestrel node. The operators console (the SPC station) is connected to an independent ethernet network that ties together patch support boards on each card cage.

ASCII red: "compute node"

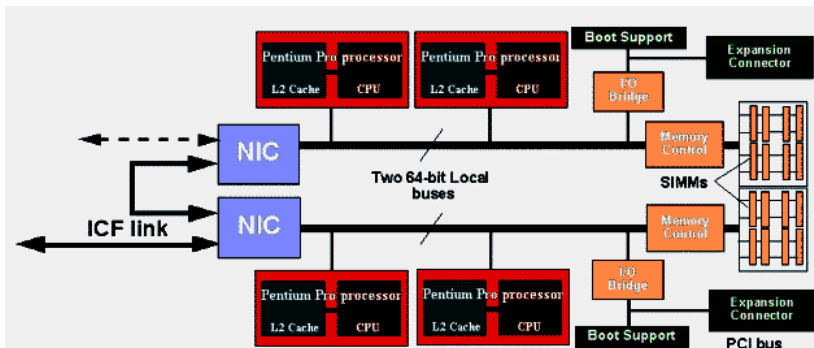


Figure 3: The ASCII Option Red supercomputer Kestrel Board. This board includes two compute nodes chained together through their NIC's. One of the NIC's connects to the MRC on the backplane through the ICF Link.

ASCII red: I/O-Node

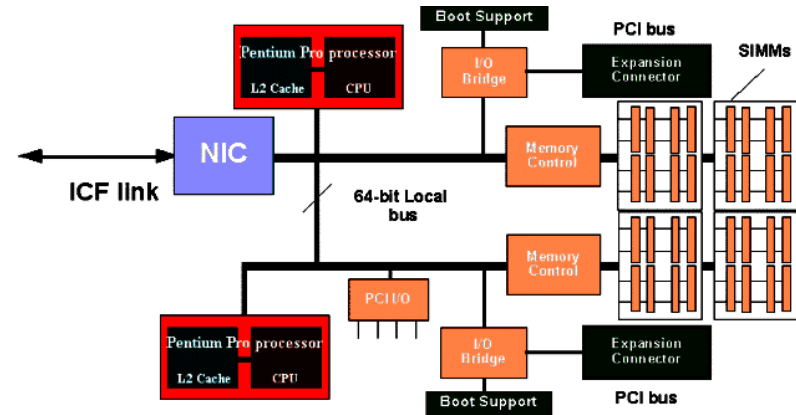


Figure 2: The ASCII Option Red Supercomputer I/O and system Node (Eagle Board). The NIC connects to the MRC on the backplane through the ICF Link.

ASCII red: "interconnection node"

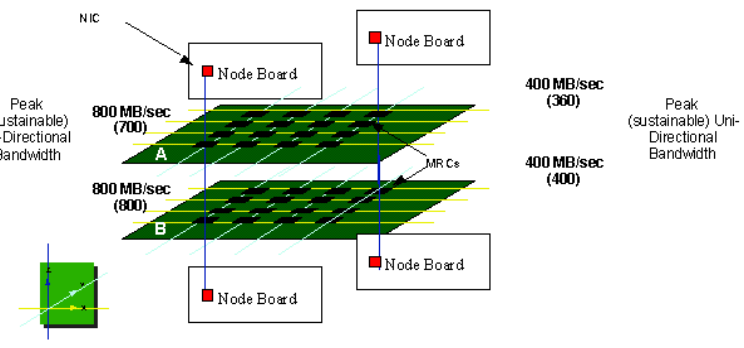


Figure 4: ASCII Option Red Supercomputer 2 Plane Interconnection Facility (ICF). Bandwidth figures are given for NIC-MRC and MRC-MRC communication. Bi-directional bandwidths are given on the left side of the figure while directional bandwidths are given on the right side. In both cases, sustainable (as opposed to peak) numbers are given in parentheses.

ASCI red: Performance

- 200 MHz PentiumPro: 200 MFLOPs peak
- 9200x: 1.8 TFLOPs peak

- Weltrekord am 07.12.1996: 1 TFLOP erreicht
 - handoptimierter Assemblercode
 - handoptimierter Algorithmus (LRU blocked, pivoting)
 - Maschine 80% vollständig => ca. 140 MFLOPs/node

 - 75% der Maximalleistung erreicht (!)

- Speicherlimitierte Programme < 20 MFLOPs / node
- Compilierte Programme 20 .. 80 MFLOPs / node

- 640 Disks, 1540 Netzteile, 616 ICF-Backplanes . . .
- MTBF > 50 hours (bzw. 97% nodes aktiv für > 4 Wochen)

[Intel ITJ Q1/98]

Bus: Agenda

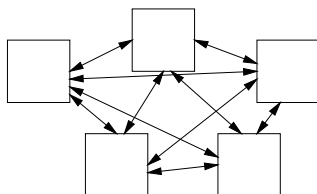
- Motivation für Busse
- ISA-Bus im PC/XT und PC/AT
- ISA Plug and Play
- EISA, MCA, VLB
- PCI-Bus
- AGP

PC-Technologie | SS 2001 | 18.214

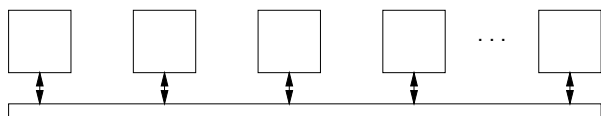
Bus: Motivation

Komponenten direkt verbinden:

- viele Signale
- irreguläre Struktur



Bus:



- n-Signale plus Steuersignale
- erfordert Arbitrierung

PC-Technologie | SS 2001 | 18.214

Bus: Literatur

- allgemein, ISA:

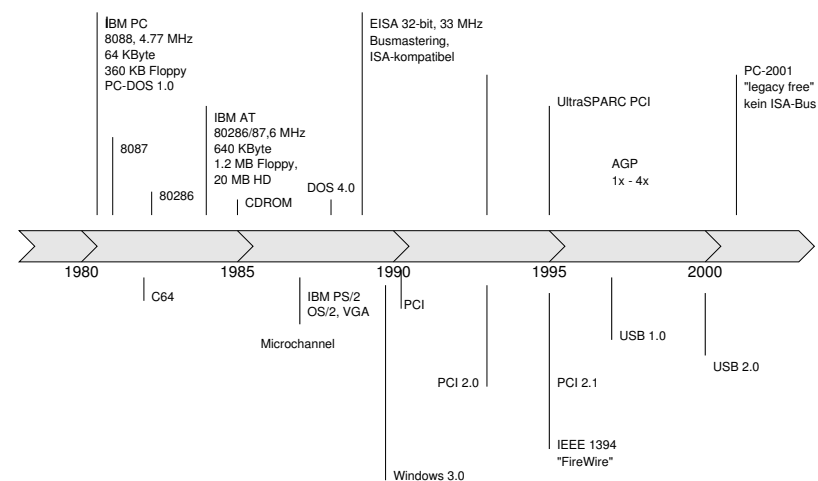
Tanenbaum, Computerarchitektur
 Hans Peter Messmer, PC-Hardwarebuch
www.ieee.org IEEE Standards, z.B. VME-Bus, ISA-Bus, ...

- PCI und AGP:

developer.intel.com
www.pcisig.org
www.webopedia.com/TERM/P/PCI.html
www.techfest.com/hardware/
www.pcguides.com/ref/mbsys/buses/

PC-Technologie | SS 2001 | 18.214

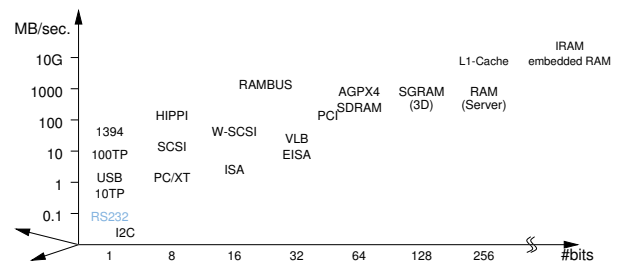
Bus: Timeline



PC-Technologie | SS 2001 | 18.214

Bus: Taxonomie

- Protokoll
 - Datenübertragung
 - Datenrate, Bandbreite
 - Anzahl der Geräte
 - Arbitrierung
- Speicherzugriff vs. messages
seriell / parallel (#bits)
bit / sec.
master, # slave
single / multi master
round-robin, ...



Bus: Peripherie

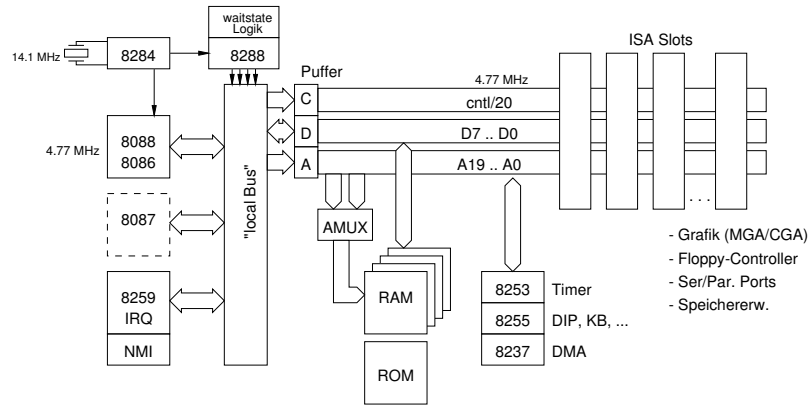
Anforderungen:	Bytes / sec.	Anschluss an:
• Speicher (PC800)	1.600.000.000	separat
• 3D (AGPx4)	1.000.000.000	AGP / PCI
• Video (800x600x16bx50)	48.000.000	AGP / PCI
• Festplatte	20.000.000	PCI (IDE/SCSI)
• Netzwerk (100TP)	10.000.000	PCI
• Audio (Synth 64x48Kx24b)	9.000.000	PCI
• Audio (CD: 2x44Kx16b)	176.000	PCI / ISA
• Floppy	100.000	ISA
• Modem	> 10.000	ISA / USB
• Terminal (25x80 Zeichen)	2.000	USB / ser.
• Maus	150	USB / ser.
• Tastatur	< 20	USB / ser.

=> Hierarchie von Bussen

Leerseite

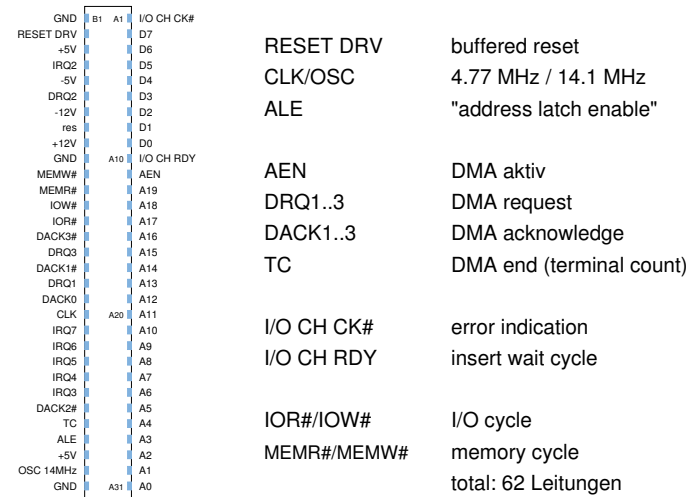
Leerseite

ISA: PC/XT



- x86 mit diversen Support-Chips
- ein gemeinsamer Bus für alle Komponenten

ISA: PC/XT Slot / Signale

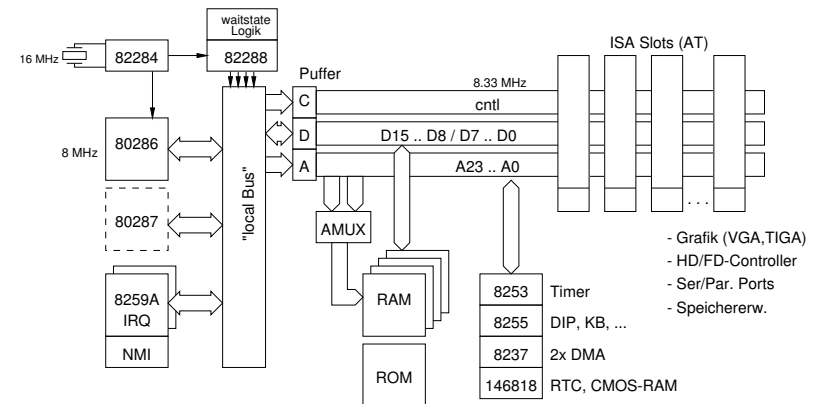


ISA: PC/XT Eigenschaften

Original-IBM PC:

- Intel 8088, 4.77 MHz (Turbo-Versionen bis 10 MHz)
- real-mode, 1 MB Adressraum
- nutzt alle verfügbaren Support-Chips
- ein gemeinsamer Bus
- 20 bit Adressen (1MB), 8 bit Daten, diverse Steuerleitungen
- RAM / ROM mit am zentralen Bus
- RAM-Refresh über Timer und DMA
- 8 Interrupt-Quellen, 3 DMA-Kanäle frei
- weitere Peripherie (Grafik!) über Slots
- nur CPU und DMA als Busmaster

ISA: PC/AT

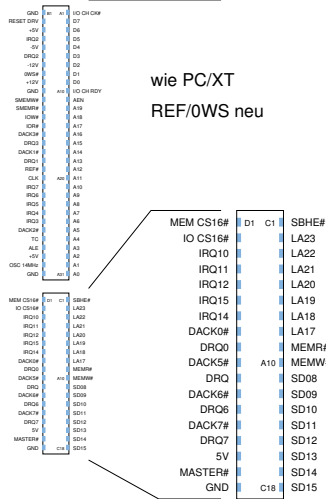


- 286 mit passenden Support-Chips
- gemeinsamer Bus, 8/16-bit Transfers

ISA: PC/AT Eigenschaften

- 80286/80287-Prozessor, plus passende Support-Chips
- 16-bit Daten, 24-bit Adressen
- real-mode oder protected-mode
- neue Slots, abwärtskompatibel für 8-bit XT-Karten
- eingeschränktes Busmastering möglich
- max. Bustakt 8.33 MHz ("ISA Standard")
- 15 Interrupt-Kanäle
- insgesamt 7 DMA-Kanäle, davon 4x 8-bit, 3x 16-bit

ISA: PC/AT Slot / Signale

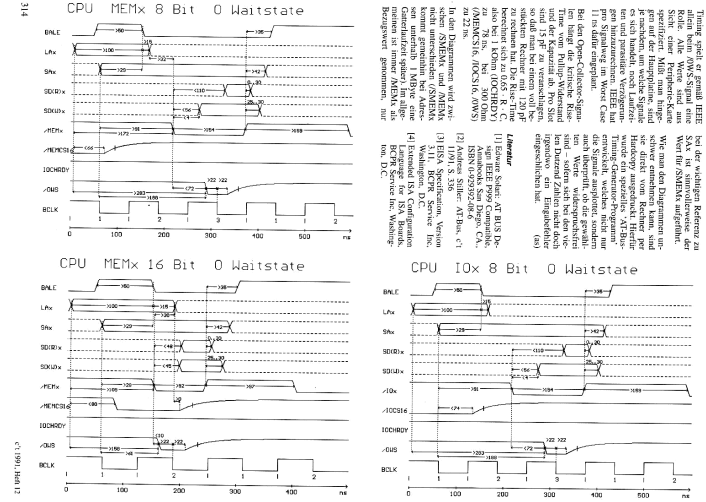


AT-Bus: XT-Bus + 36 neue Kontakte
XT Karten passen weiterhin

obere 8 Datenleitungen SD08..SD15
weitere Adressleitungen
neue DMA/IRQ-Leitungen

- OVS (B8): 0 Wartezyklen nötig
- REF (B19): Speicherrefresh (dack0)
- SBHE: system bus high enable
- IRQ1x: Interrupt-Inputs
- DRQ/DACK: DMA request / acknowledge
- Master: Busmaster-Anforderung
- SMEM/MEM: Zugriffe 0..1M, 1M ..16M

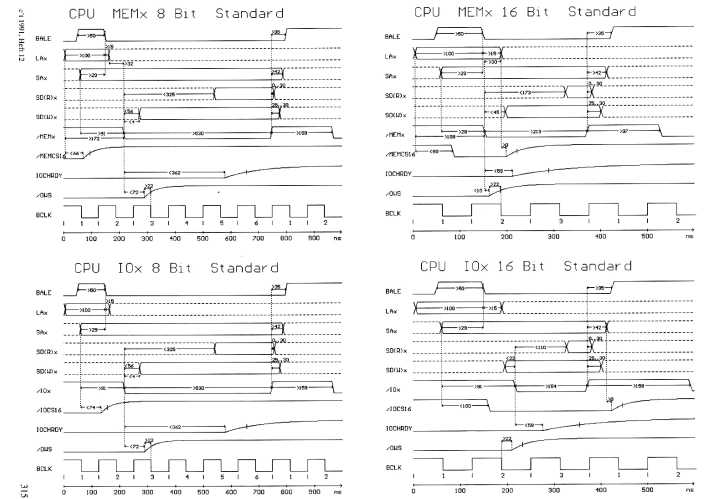
ISA: PC/AT read/write, 0 wait



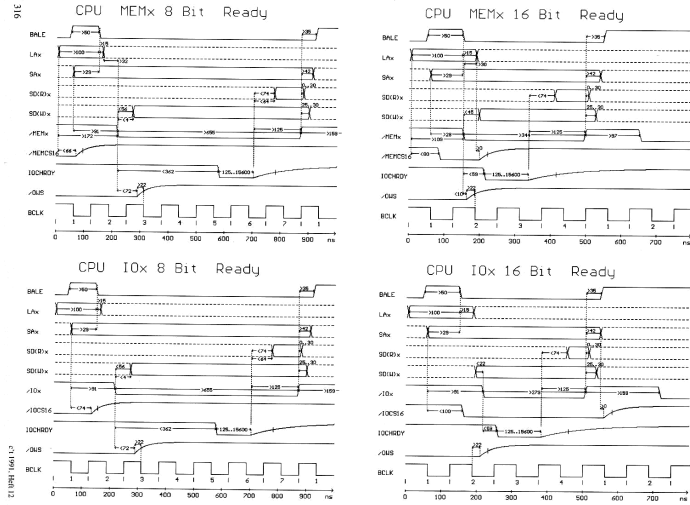
Timing: 1ns, 10ns, 100ns, 1µs, 10µs, 100µs, 1ms

Legend:
 [1] Editor-Symbol: AT-BUS-IOx-8-Bit-0-Waitstate
 [2] Editor-Symbol: AT-BUS-IOx-16-Bit-0-Waitstate
 [3] Editor-Symbol: AT-BUS-IOx-8-Bit-Standard
 [4] Editor-Symbol: AT-BUS-IOx-16-Bit-Standard

ISA: PC/AT read/write

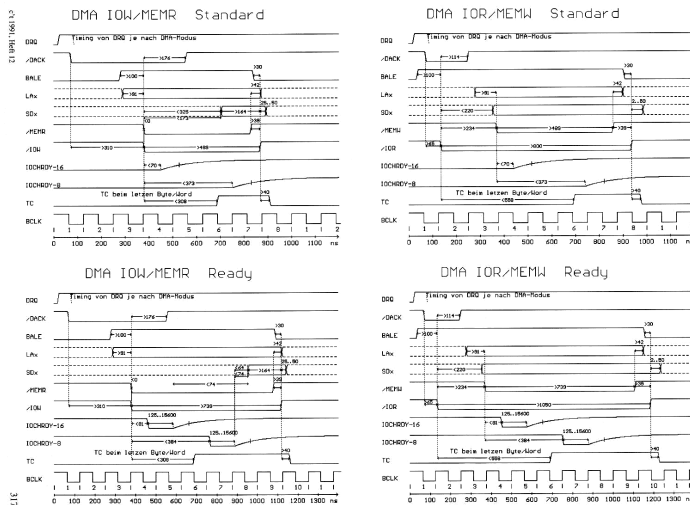


ISA: PC/AT read/write (ready)



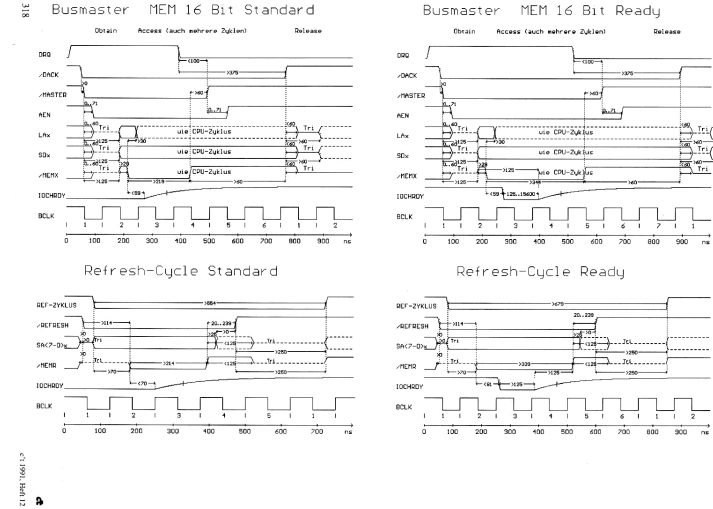
PC-Technologie | SS 2001 | 18.214

ISA: PC/AT DMA-Zyklen



PC-Technologie | SS 2001 | 18.214

ISA: PC/AT Busmaster-Zyklen



PC-Technologie | SS 2001 | 18.214

Leersseite

PC-Technologie

ISA: Plug and Play

- ISA-Karten benutzerkonfiguriert: Jumper
- viele Hersteller, keine einheitliche Konfiguration
- beschränkte Ressourcen (I/O, IRQ, DMA)

=> häufig Probleme durch Konflikte
=> Abhilfe durch Autokonfiguration der Karten

Plug and Play ISA Specification:

- automatische Erkennung von PnP-Karten
- Konfiguration durch PnP-BIOS und PnP-OS (=Windows 9x)
- entwickelt von Microsoft und Intel, 1993-1994

www.roestock.demon.co.uk/isapnptools/
www.microsoft.com/hwdev/respec/pnpspecs.htm

ISA: Plug and Play

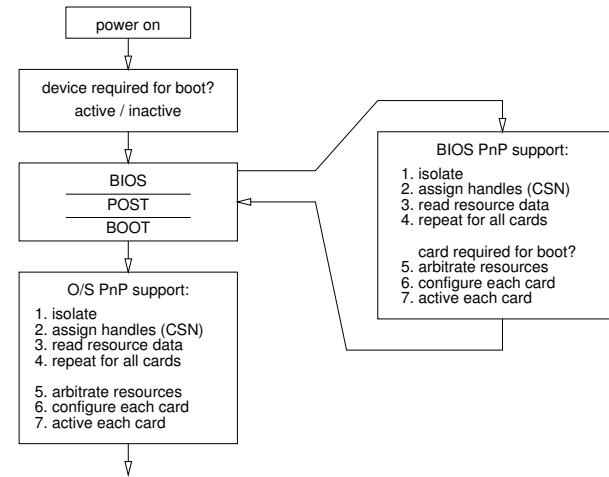
essentielle Funktionen des ISA-PnP:

- Erkennung der PnP-Karten (isolate)
- Auslesen der Konfigurationsdaten (identify)
- Setzen gültiger Konfigurationsdaten
- Laden geeigneter Kartentreiber

Vorteile:

- PnP-Karten sind kompatibel zu alten ISA-Karten
- keine Änderung des PCs / Motherboards notwendig
- System voll autokonfigurierbar, solange nur PnP-Karten
- "ease of use for the end user"

ISA: Plug and Play: Sequenz



PnP: I/O-Ports

Three 8-bit ports are used by the software to access the configuration space on each Plug and Play ISA card. The ports are listed in table 1. The configuration space is implemented as a set of 8-bit registers. These registers are used by the Plug and Play software to issue commands, check status, access the resource data information, and configure the Plug and Play hardware.

The ports have been chosen so as to avoid conflicts in the installed base of ISA functions, while at the same time minimizing the number of ports needed in the ISA I/O space.

Table 1. Auto-configuration Ports

Port Name	Location	Type
ADDRESS	0x0279 (Printer status port)	Write-only
WRITE_DATA	0x0A79 (Printer status port + 0x0800)	Write-only
READ_DATA	Relocatable in range 0x0203 to 0x03FF	Read-only

The ADDRESS and WRITE_DATA ports are located at fixed addresses. The WRITE_DATA port is located at an address alias of the ADDRESS port. All three auto-configuration ports use a 12-bit ISA address decode.

The READ_DATA port is relocatable within the I/O range from 0x0203h to 0x03FFh. This is the only readable auto-configuration port.

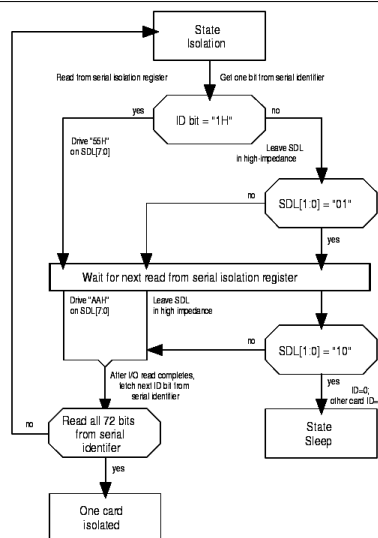
ISA: Plug and Play Isolate

Isolate: Erkennen der PnP-Karten auf dem ISA-Bus

- drei I/O-Adressen:

ADDRESS	0x0279	write
WRITE_DATA	0x0A79	write
READ_DATA	0x0203 .. 0x03FF	read
- initiation:
 - write 0x00, 0x00
 - write initiation key (LFSR)
- isolation:
 - repeat
 - isolate one card
 - if (ISA conflict) relocate READ_DATA
 - until (no more cards found)
- identify

PnP: Isolation-Protocol



findet ein pnp-Gerät pro Iteration
(höchste ID zuerst)
entsprechend oft wiederholen

erkennt Konflikte mit nicht-pnp Karten
bei Konflikt den Read-Port wechseln

PnP: Hardware-Protocol

3.3.1. Hardware Protocol

The isolation protocol can be invoked by the Plug and Play software at any time. The initiation key, described earlier, puts all cards into configuration mode. The hardware on each card expects 72 pairs of I/O read accesses to the READ_DATA port. The card's response to these reads depends on the value of each bit of the serial identifier which is being examined one bit at a time, in the sequence shown in figure 5.

If the current bit of the serial identifier is a „1“, then the card will drive the data bus to 0x55 to complete the first I/O read cycle. If the bit is „0“, then the card puts its data bus driver into high impedance. All cards in high impedance will check the data bus during the I/O read cycle to sense if another card is driving D[1:0] to „01.“ During the second I/O read, the card(s) that drove the 0x55, will now drive a 0xAA. All high impedance cards will check the data bus to sense if another card is driving D[1:0] to „10.“

If a high impedance card sensed another card driving the data bus with the appropriate data during both cycles, then that card ceases to participate in the current iteration of card isolation. Such cards, which lose out, will participate in future iterations of the isolation protocol.

NOTE: *During each read cycle, the Plug and Play hardware drives the entire 8-bit data bus, but only checks the lower 2 bits.*

If a card was driving the bus or if the card was in high impedance and did not sense another card driving the bus, then it should prepare for the next pair of I/O reads. The card shifts the serial identifier by one bit and uses the shifted bit to decide its response.

The above sequence is repeated for the entire 72-bit serial identifier.

At the end of this process, one card remains. This card is assigned a handle referred to as the *Card Select Number* (CSN) that will be used later to select the card. Cards which have been assigned a CSN will not participate in subsequent iterations of the isolation protocol. Cards must be assigned a CSN before they will respond to the other commands defined in the specification.

PnP: Software-Protocol

3.3.2. Software Protocol

The Plug and Play software sends the initiation key to all Plug and Play cards to place them into configuration mode. The software is then ready to perform the isolation protocol.

The Plug and Play software generates 72 pairs of I/O read cycles from the READ_DATA port. The software checks the data returned from each pair of I/O reads for the 0x55 and 0xAA driven by the hardware. If both 0x55 and 0xAA are read back, then the software assumes that the hardware had a „1“ bit in that position. All other results are assumed to be a „0.“

During the first 64 bits, software generates a checksum using the received data. The checksum is compared with the checksum read back in the last 8 bits of the sequence.

There are two other special considerations for the software protocol. During an iteration, it is possible that the 0x55 and 0xAA combination is never detected. It is also possible that the checksum does not match. If either of these cases occur on the first iteration, it must be assumed that the READ_DATA port is in conflict. If a conflict is detected, then the READ_DATA port is relocated. The above process is repeated until a non-conflicting location for the READ_DATA port is found. The entire range between 0x200 and 0x3FF is available, however in practice it is expected that only a few locations will be tried before software determines that no Plug and Play cards are present.

During subsequent iterations, the occurrence of either of these two special cases should be interpreted as the absence of any further Plug and Play cards (i.e. the last card was found in the previous iteration). This terminates the isolation protocol.

PnP: Architektur der Steckkarten

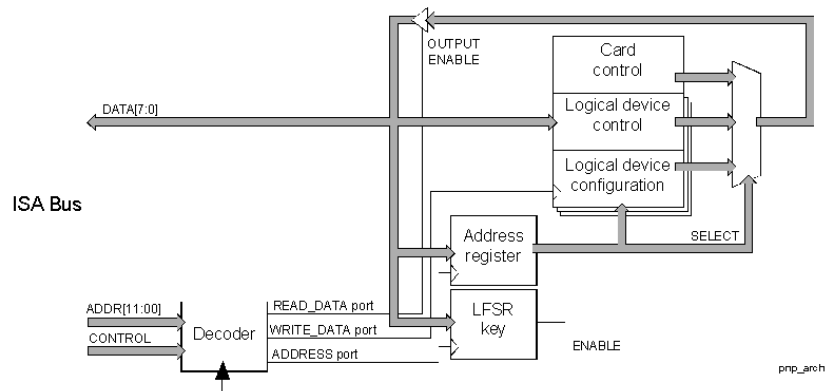


Figure 3. Logic Flow for Auto-configuration

- diverse Konfigurationsregister, spezielles LFSR
- ansonsten "normale" ISA-Karte

PnP: Configuration space

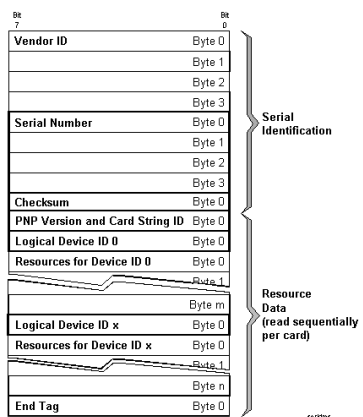


Figure 7. Serial Identifier and Resource Data

- Details siehe ISAPNP-Spezifikation

PCI: Motivation

ISA-Bus:

- zu langsam für Video und 3D
- Autokonfiguration problematisch
- kein effizientes Busmastering

=> neuer Bus erforderlich

=> aber Platzbedarf und Kosten ähnlich wie ISA

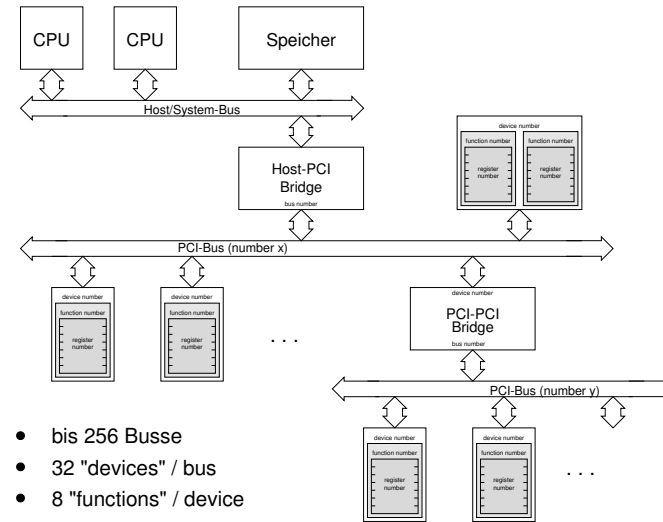
PCI: "Peripheral Component Interconnect", 1992:

- 32-bit Bus mit Option auf 64-bit
- Bandbreite 132 MB/sec @ 33 MHz
- weitgehende Autokonfiguration
- entwickelt, patentiert, freigegeben von Intel
- PCI Special Interest Group, www.pcisig.org

PCI: Übersicht

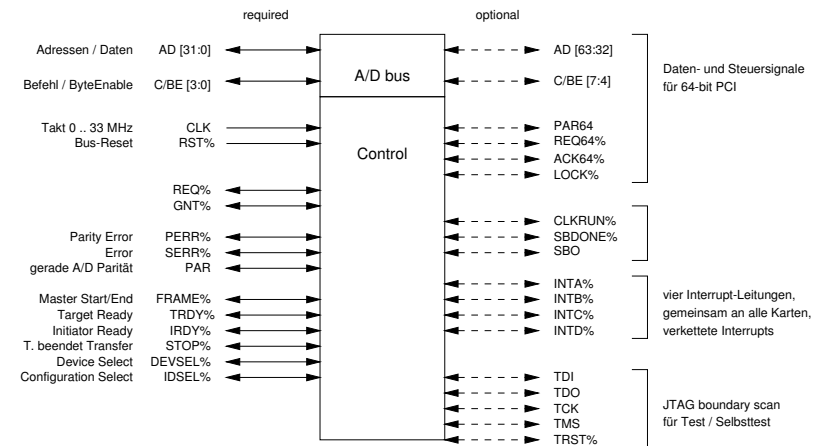
- universelles Bussystem
- ausreichend schnell für fast alle Anwendungen
- diverse Varianten (33/66 MHz, 32/64 Bit, 3.3/5 V, PCI-X)
- volle Autokonfiguration aller Devices
- flexibles Busmaster- und Interrupt-Konzept
- Burst-Transfers für hohe Bandbreite
- kaskadierbar über PCI-PCI-Bridges
- seit ca. 1994 in allen PC-Chipsätzen integriert
- teilweise direkt als Prozessorkomponente, etwa UltraSPARC Ili
- erstaunlicher Markterfolg, über 800 Hersteller
- hat fast alle proprietären Busse verdrängt
- eingeführt 1992, Version 2.0 seit 1993, derzeit 2.1

PCI: Hierarchie mit mehreren Bussen



- bis 256 Busse
- 32 "devices" / bus
- 8 "functions" / device

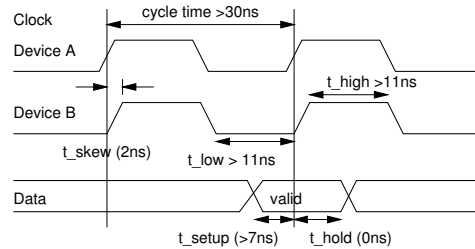
PCI: Signale



- A/D multiplex, insgesamt 120 / 184 Pins

PCI: Takt

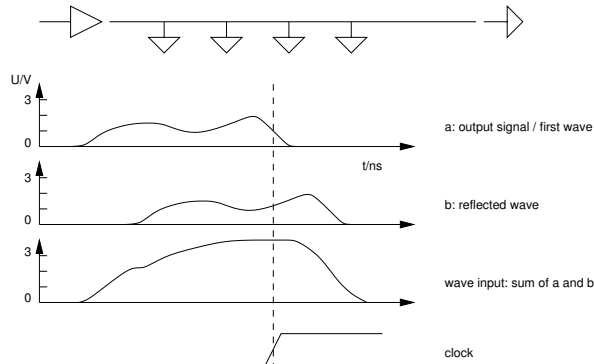
- Taktfrequenz: 33 MHz (Standard) / 66 MHz (Option)



- jeweils ein Takt für Adress/Daten/Wait-Zyklus
- niedrigere Frequenzen sind erlaubt
- PC-Motherboards: PCI-Takt meistens an Systemtakt gekoppelt
- Übertakten problematisch

PC-Technologie | SS 2001 | 18.214

PCI: "reflected waves"



- nutzt Reflektionen am Ende des Busses ...
- Platinenlayout kritisch, kurze Leitungen
- besondere Eingangsschaltungen in den PCI-Geräten
- kein statischer Stromverbrauch (anders als SCSI)

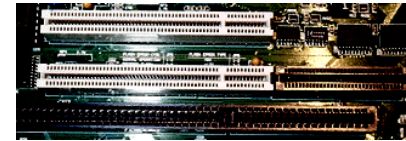
PC-Technologie | SS 2001 | 18.214

PCI: Stecker

- Varianten:

3.3V / 5V Signale
33 MHz / 66 MHz
32 Bit / 64 Bit

Position des Stegs im Stecker
Signal M66EN auf GND/VCC
einfacher / langer Stecker

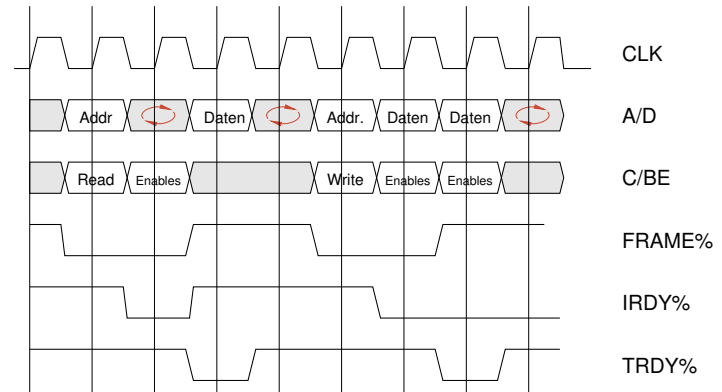


PCI, 5V, 32 bit
PCI, 5V, 64 bit
ISA, 16 bit

- 66 MHz nur dann, wenn alle Karten es erlauben
- narrensicher

PC-Technologie | SS 2001 | 18.214

PCI: Transfer



- FRAME signaliert Start und Ende eines Transfers
- je ein Warte-Takt (\$) zur Umschaltung Read/Write
- Transfer beendet wenn TRDY & IRDY

PC-Technologie | SS 2001 | 18.214

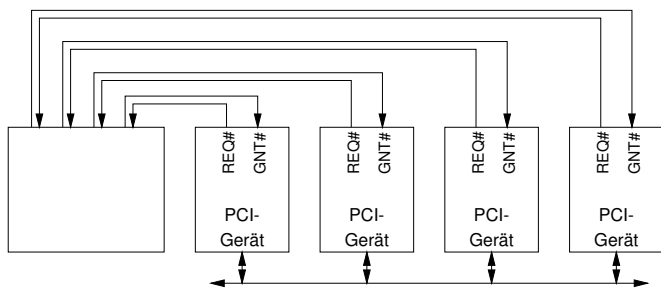
PCI: Befehle / Transfers

C/BE	3	2	1	0	
0	0	0	0	0	Interrupt Acknowledge
0	0	0	0	1	Special Cycle (broadcast, data contains message type)
0	0	1	0	0	I/O Read
0	0	1	1	1	I/O Write
0	1	0	0	0	reserved
0	1	0	1	1	reserved
0	1	1	0	0	Memory Read
0	1	1	1	1	Memory Write
1	0	0	0	0	reserved
1	0	0	1	1	reserved
1	0	1	0	0	Configuration Read
1	0	1	1	1	Configuration Write
1	1	0	0	0	Memory Read Multiple
1	1	0	1	1	Dual Address Cycle
1	1	1	0	0	Memory Read Line
1	1	1	1	1	Memory Write And Invalidate

- I/O: Byte-Adressen, keine Bedingungen (Nebenwirkungen)
- Memory: 4-Byte Adressen, Lesen liefert zuletzt geschriebenen Wert

PC-Technologie | SS 2001 | 18.214

PCI: Arbitrierung



- ein zentraler Arbitrier (normalerweise im Chipsatz)
- Anforderung mit REQ, Erlaubnis mit GNT
- gilt für eine Transaktion beliebiger Burst-Länge
- Arbitrier kann GNT entziehen, um Transaktion zu beenden

PC-Technologie | SS 2001 | 18.214

PCI: PCI-BIOS

- PCI definiert auch Standard-Programmierschnittstelle
- als Menge von x86 Software-Interrupts
- als "Gast" (Erweiterung) des Uhren-Interrupts 1Ah
- PCI 2.1 definiert 12 Funktionen, Übergabe per AL-Register

```
boolean PCI_BIOS_Present() {
    int AX = 0xB101; // "magic" PCI 2.1 defined function number
    interrupt( 0x1A );
    boolean pci_bios_present = (AH==0) && (DX==0x4350);
    int number_of_busses = CL;
    int pci_version = BX;
    int config_version = AL;

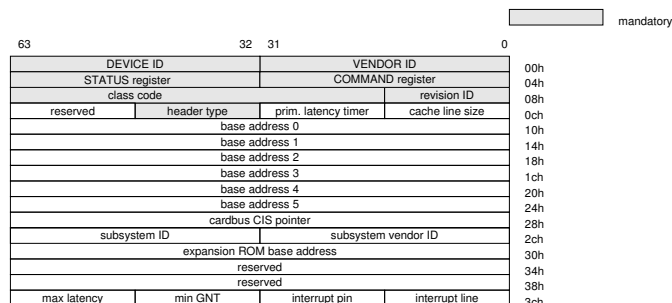
    return pci_bios_present;
}
```

PCI: PCI-BIOS Funktionen

- PCI definiert auch Standard-Programmierschnittstelle
- Beispielprogramme (Turbo-Pascal) z.B. in [ct 02-96-266ff]

```
PCI_BIOS_present
find_PCI_device
find_PCI_class_code
generate_special_cycle (shutdown, halt, x86 )
read_configuration_xxx
write_configuration_xxx
get_PCI_interrupt_routing_options
set_PCI_IRQ
. . .
```

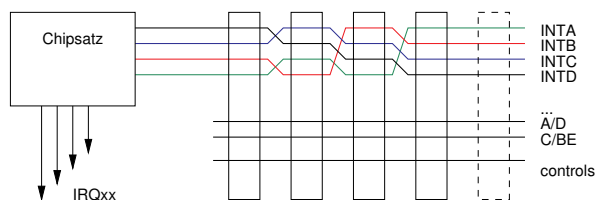

PCI: Configuration Space



256 Byte (64 DWORD) Configuration Space pro PCI Gerät:

- eindeutige Hersteller / Geräte-ID
- Status- und Befehlsregister
- Identifikation der Karten erlaubt Autokonfiguration

PCI: Interrupts



- Chipsatz setzt INTA .. INTD auf IRQxx um
- jede Karte darf alle Interrupts benutzen

Interrupt-Handler "chaining"

- Handler sucht nach auslösendem Gerät
- nicht immer perfekt implementiert :-)
- oft nur INTA benutzt: "krumme" Verdrahtung = weniger Konflikte
- manchmal hilft "Karten umstecken"

PCI: Interrupt Chaining

```

/* irq handler with irq chaining - from AMCC S5933 databook */
void interrupt_handler( void )
{
    byte status;

    /* read AMCC S5933 chip status */
    status = inportb( REG_BASE_ADDR + AMCC_REG_INTCSR + 2 );

    if ((status & ANY_S5933_INT) != 0) { /* handle interrupt */
        /* disable bus mastering */
        outportb( REG_BASE_ADDR + AMCC_REG_MCSR+1, 0x11 );

        /* identify interrupt source(s) */
        if ((status & READ_TC_INT) != 0)
            /* read TC interrupt code goes here */
        if ((status & WRITE_TC_INT) != 0)
            /* write TC interrupt code goes here */
        ...

        /* clear interrupt enables */
        outportb( REG_BASE_ADDR + AMCC_REG_INTCSR+1, 0 );
        outportb( REG_BASE_ADDR + AMCC_REG_INTCSR+2, status );
    }
    else { /* not an S5933 interrupt, dispatch to next handler */
        chain_intr( oldhandler );
    }

    /* end of interrupt handler: clear in-service bits */
    ...
    outportb( 0x20, 0x60 | (interrupt_line & 0x07) );
}

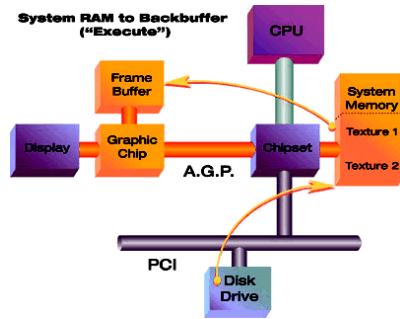
```

AGP:

"Accelerated Graphics Port"

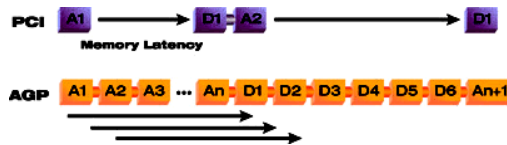
- basiert auf der 66 MHz PCI-Spezifikation
- nur zwei Geräte, Punkt-zu-Punkt Verbindung statt Bus
- pipeline-read/write Zugriffe
- daher keine Blockierung wegen Speicherlatenz
- spez. Transfermodi (Direktzugriff auf Hauptspeicher)
- separate Daten/Adressleitungen (sideband addressing)
- AGP 1x Transfers wie PCI-66
- AGP 2x "double data rate" Transfers
- AGP 4x niedrigere Spannungen, vierfache Datenrate
Bandbreite bis 1 GB/s
- developer.intel.com

AGP: Konzept



- Chipsatz unterstützt sowohl AGP als auch PCI
- spezielle AGP-Transfers für Direktzugriff auf Hauptspeicher
- dazu Adressumsetzung im Chipsatz
- sehr komplizierte Details (siehe AGP specification)

AGP: vs. PCI



- PCI-Bus erlaubt keine "split-transactions":
- jede Transaktion muß Latenzzeiten voll abwarten
- AGP definiert Dutzende spezieller Transfermodi
- Hostrechner oder Graphikkarte als Initiator
- split-transactions zum Verdecken der Speicher-Latenzzeiten

AGP: AGP-4x Transfer

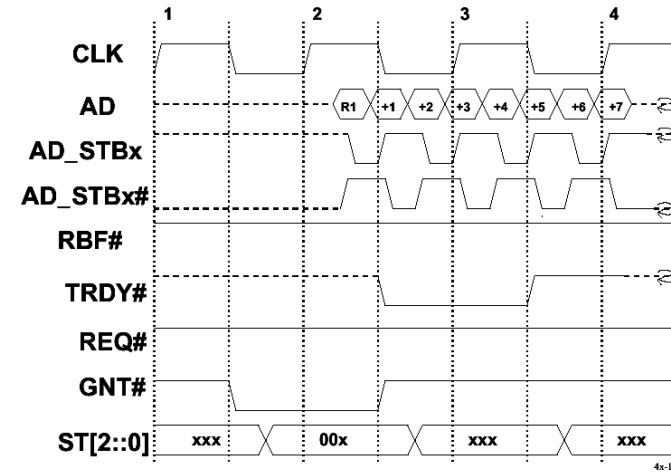


Figure 3-31: 4x Read Data - No Delay

AGP: Performance, vs. PCI

Vergleich PCI- und AGP-Grafikkarten								
	3D Mark99 MAX, Racing	3D Mark99 MAX, 1st Person	Expendable/16	Expendable/32	Quake II, Crusher /16	Quake II, Crusher /32	Quake 3, q3demo1 /16	Quake 3, q3demo1 /32
TNT-PCI-Grafik	38	42	35	20	27	21	18	14
TNT-AGP-Grafik	42	46	35	23	31	23	21	14
TNT2-PCI-Grafik	30	45	45	39	36	29	33	25
TNT2-AGP-Grafik ¹	30	44	45	39	36	- ²	- ²	- ²

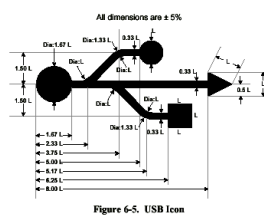
Alle Tests liefen unter DirectX6.1, bei 1024 x 768 Bildpunkten, 75 Hz Bildwiederholrate, "Wait for VSync"-on, Audio ein, alle Werte in Frames per Second (fps), /16: HiColor 16 Bit, /32: TrueColor 32 Bit

¹ Vergleichswerte Asus V3800 TVR mit VSync = Off aus [7], ² keine Vergleichswerte verfügbar

- achtfache Bandbreite von AGP4x gegenüber Standard-PCI
- Texturdaten im Hauptspeicher möglich
- wird aber von aktuellen Apps/Spielen noch nicht genutzt
- Vorteile nur bei Benchmarks / großen Texturen

Serielle Busse: Agenda

- Motivation für serielle Busse
- USB - Architektur
- USB - Treiber
- IEEE 1394 (FireWire)



Serielle Busse: Motivation

warum neue Busse?

- Schnittstellengewirr des "legacy"-PC
- Überwinden der lästigen IRQ/DMA/IO Ressourcenkonflikte
- Autokonfiguration
- Hot-Plugging
- Plattformunabhängige Standards (PC/Audio/Video/...)

warum serielle Busse?

- so billig wie möglich, nicht schneller als nötig
 - möglichst lange Kabel
 - Probleme mit Übersprechen/Skew bei paralleler Übertragung
 - Chips sind billig, Kupfer ist teuer
- NE2000-Karte ca. DM 20.00, U2W-SCSI Kabel DM 300.00

Serielle Busse: Literatur

www.usb.org
developer.intel.com
www.microsoft.com/hwdev/usb/
 Kelm (Hrsg.) USB, Universal Serial Bus, Franzis Verlag, 1999
usb.sourceforge.net (USB for Linux homepage)
www.emsys.de/usb (Hardware-Übersicht)
 Don Anderson Universal Serial Bus System Architecture, MindShare, 1997
www.mindshare.com

www.1394.org
www.microsoft.com/hwdev/1394
www.firewire.org (=www.apple.com) www.apple.com/firewire/firewireproducts.html
 Don Anderson FireWire System Architecture, MindShare, 1999
1394.sourceforge.net (FireWire for Linux)
 Mike Teener, 1394 overview www.zayante.com

Serielle Busse: USB vs. 1394

welcher Bus für welche Anwendung?

(Microsoft unterstützt beide)

- USB: ausgerichtet auf PC und PC-Peripherie
Single-Master-Architektur
- FireWire: eigenständige Multimediageräte
Audio/Video-Protokolle etabliert
kein PC als Master notwendig

Bus	Geschwindigkeit	Host-Komplexität	Peripherie-Komp.
IEEE 1394	400 Mb/sec.	12000-20000 gates	5000 - 7000 gates
USB 1.1	12 Mb/sec	10000 gates	2500 - 3000 gates

USB:

- entwickelt seit ~1990 von
 - Compaq DEC IBM Intel Microsoft NEC Northern Telecom
 - große Ähnlichkeit zu Apples FireWire/IEEE 1394
 - aber Ausrichtung auf PC und PC-Peripherie
 - serieller, billiger Bus, bis zu 127 Geräte
 - low/full speed mit 1.5 Mb/s und 12 Mb/s
 - volle Autokonfiguration mit Hot-Plugging
 - Geräte identifizieren sich für das Betriebssystem
 - isochrone Transfers für Media-Processing
- | | | |
|-----------|-----------|------------|
| • USB 1.0 | Q1 / 1996 | |
| • USB 1.1 | Q3 / 1998 | |
| • USB 2.0 | Q1 / 2000 | (480 Mb/s) |

PC-Technologie | SS 2001 | 18.214

USB: Intel IDF Demo



- 1996 nur wenige Geräte verfügbar
- seit HX/PIIX3 in jedem Intel Chipsatz
- mittlerweile etabliert
- Gerätevielfalt

PC-Technologie | SS 2001 | 18.214

USB: Ziele

- preisgünstig (Stecker/Kabel/ASICs)
- einheitliche Stecker und Kabel (4 Pins, Typ A/B, max. 5m)
- unverwechselbare Stecker (Hub Typ A, Client Typ B)
- sehr viele Geräte (max. 127 Geräte)
- Hot-Plugging (Anstecken im Betrieb)
- zwei Geschwindigkeiten (1.5 Mb/s und 12 Mb/s)
- flexible Datenübertragung (4 Transferarten)
- Stromversorgung über das Kabel (5V, max. 500 mA)
- benötigt keine ISA-Ressourcen (IRQ/DMA/IO-Ports)
- Schlafzustand der Geräte (suspend nach 3 msec.)
- USB 2.0 deutlich schneller (bis 480 Mb/s)

PC-Technologie | SS 2001 | 18.214

USB: Gerätespektrum . . .

- | | |
|--|---|
| • Tastaturen, Mäuse | bisherige Schnittstelle:
(PS/2, seriell) |
| • Joysticks, Gamepads (force feedback) | (gameport) |
| • Monitore mit integrierten Hubs | |
| • Drucker, Scanner | (SCSI/parallel) |
| • Digitalkameras | (seriell) |
| • Modems, ISDN-Adapter, USB-Netze | (seriell) |
| • Wechselplatten, CDROM, ... | (SCSI/parallel) |
| • Audio-Geräte | (analog audio in/out) |
| • MIDI-Geräte (Tonstudios) | (seriell) |
| • Kopierschutz-Adapter | (seriell/parallel) |
| • . . . | |

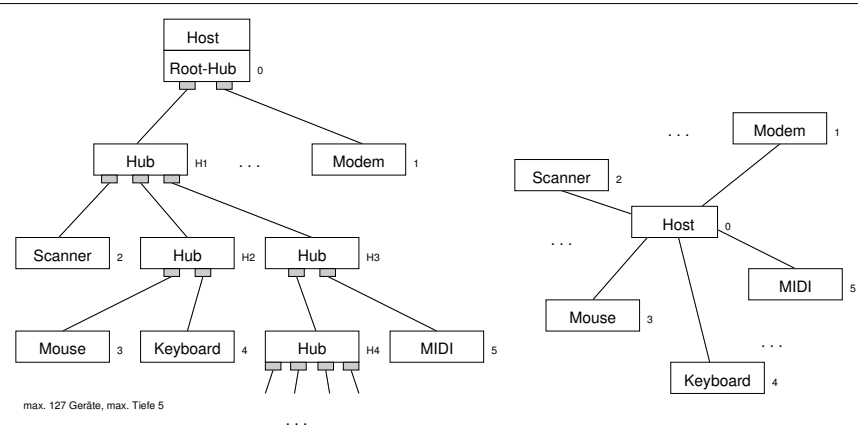
PC-Technologie | SS 2001 | 18.214

USB: Aufgaben der Host-SW:

- detect attachment and removal of USB devices
- managing control flow between host and USB devices
- managing data flow between host and USB devices
- collecting status and activity statistics
- providing power to attached USB devices

- device enumeration and configuration
- isochronous data transfers
- asynchronous data transfers
- power management
- device and bus management information

USB: Architektur



max. 127 Geräte, max. Tiefe 5

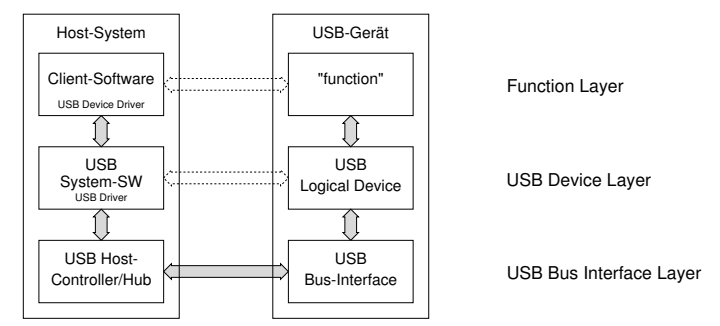
physikalische Struktur:

- Baum, Host an der Wurzel

logische Topologie:

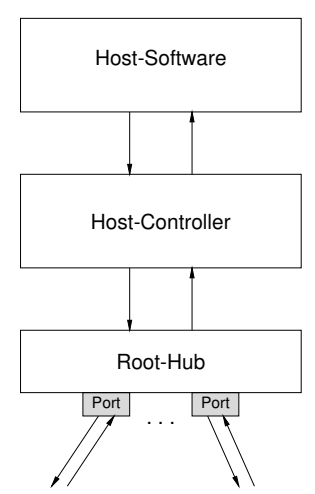
- Stern, Host steuert alles

USB: Software-Architektur



- USB-Device-Driver Client Requests -> IO request packets
- USB-Driver IRPs -> Transactions
- Host-Controller-Driver Frame List, Scheduling

USB: Host-Controller und Root-Hub



- alle USB-Transfers unter Kontrolle der Host-Software

- Varianten: Open/Universal-HC
- Erzeugen der Bus-Transaktionen
- Abfolge der Transaktionen (Scheduling)
- De-/Serialisierung der Daten

- Steuerung der Stromzufuhr
- Freischalten/Sperrern der Ports
- Dis-/Connect-Erkennung
- Status-Verwaltung der Ports

USB: Transferarten

vier separate Datentransfers:

- **Control-Transfer** Host sendet Befehle an die Clients
10% Bandbreite garantiert
 - **Interrupt-Transfer** Host fragt Clients nacheinander per Polling ob "Interrupt"-Ereignisse vorliegen
 - **Bulk-Transfer** eigentliche Datenübertragung ohne Latenz / Echtzeitanforderungen
 - **Isochronous-Transfer** Datenübertragung mit garantierter Latenz / Bandbreite, etwa für Audiodaten.
- 90% Bandbreite für Interrupt/Isochron.
Bulk-Transfers nur wenn Bus frei

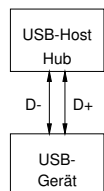
PC-Technologie | SS 2001 | 18.214

USB: Pipe / Endpoint

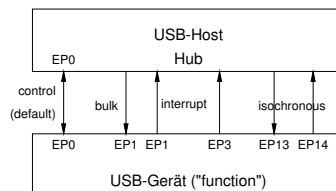
USB-"Pipe" :=

- einzelner logischer Datenkanal
- als Verbindung zwischen Endpoints an Root und "Function"
- bis zu 16 Endpoints pro Function
- Adressierung: 7-bit Function-ID, 4-bit EP-ID, Richtung
- Endpoints werden im Device-Deskriptor definiert
- EP0 ist immer Control

phyikalisch:

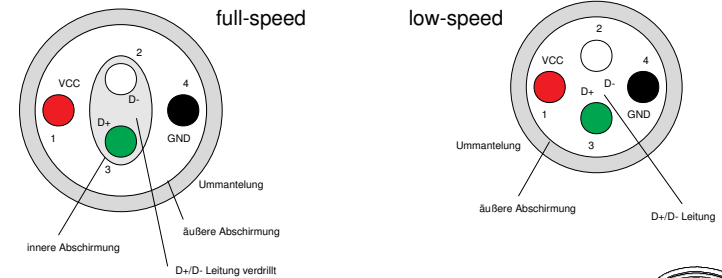


logisch:

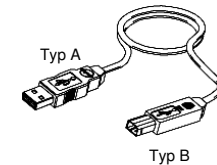


PC-Technologie | SS 2001 | 18.214

USB: Kabel und Stecker

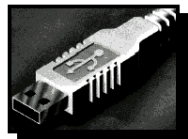

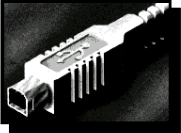
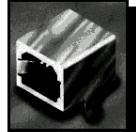


- möglichst billige Kabel / mit Stromversorgung
- max. 30 nsec. Laufzeit pro Kabelsegment
- Kabellängen von 0.8 .. 5.0 m je nach Querschnitt
- Typ A/B-Stecker verhindert Zyklen und Kurzschlüsse
- low-speed Geräte haben festes Kabel (< 3m)



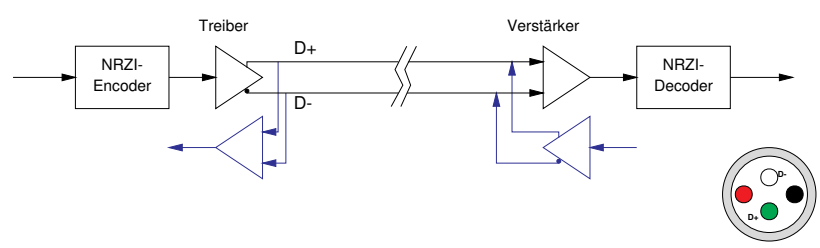
PC-Technologie | SS 2001 | 18.214

USB: Stecker

Series "A" Connectors	Series "B" Connectors
<p>◆ Series "A" plugs are always oriented upstream towards the <i>Host System</i></p>  <p>"A" Plugs (From the USB Device)</p>  <p>"A" Receptacles (Downstream Output from the USB Host or Hub)</p>	<p>◆ Series "B" plugs are always oriented downstream towards the <i>USB Device</i></p>  <p>"B" Plugs (From the Host System)</p>  <p>"B" Receptacles (Upstream Input to the USB Device or Hub)</p>

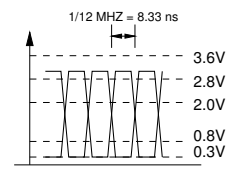
PC-Technologie | SS 2001 | 18.214

USB: Pegel / Kodierung



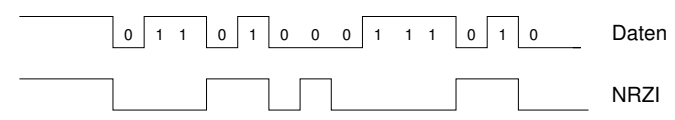
- Hin- und Rückrichtung über dieselben Leitungen
- Störsicherheit durch differentielle Signale
- NRZI-Kodierung und Bit-Stuffing

Differential "1": (D+) - (D-) > 200 mV
 Differential "0": (D+) - (D-) < -200 mV

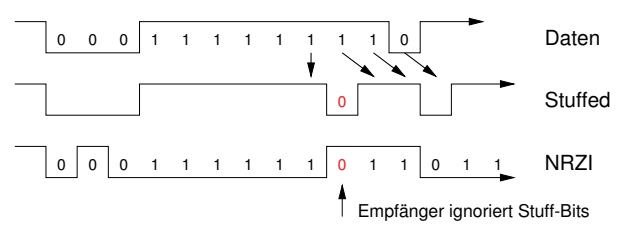


USB: NRZI und Bitstuffing

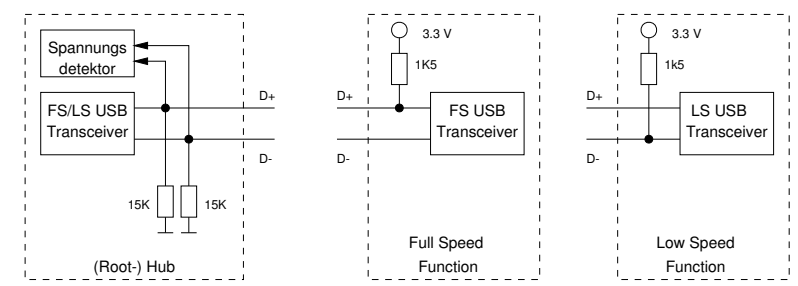
- NRZI-Kodierung ("non return to zero inverted")
- 0-Datenbit: Signalwechsel, 1-Datenbit: kein Signalwechsel



- Bit-Stuffing nach 6 Einsen zur Taktrückgewinnung:



USB: Connect / Disconnect

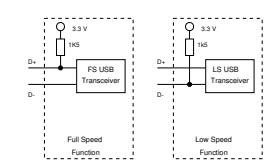


- kein Gerät am Hub: (D+) 0.0 V, (D-) 0.0 V (2x pulldown)
- FS-Gerät am Hub: (D+) 3.3 V, (D-) 0.0 V
- LS-Gerät am Hub: (D+) 0.0 V, (D-) 3.3 V
- Hub misst (D+)/(D-) Connect/Disconnect wird erkannt

USB: J/K-Zustände

Bus inaktiv: (D+)/(D-) nicht aktiv angesteuert:

- FS/LS-Erkennung über die Pullups
- aber unterschiedliche Spannungen



Bus aktiv: Hub oder Function treibt (D+)/(D-)

- differentielle Kodierung, kein Unterschied zwischen FS/LS

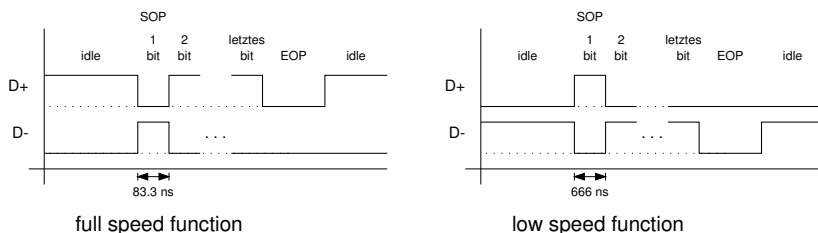
	Zustand	D+	D-	differenziell
low-speed	J (idle)	low	high	0
	K (resume)	high	low	1
full-speed	J (idle)	high	low	1
	K (resume)	low	high	0

USB: Synchronisation

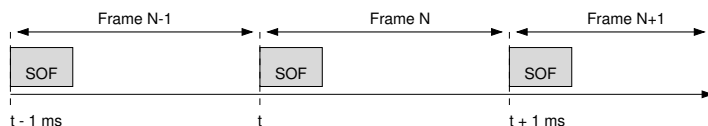
Synchronisation von Hub und Functions:

- inaktiver Bus wird nicht getrieben:
- SOP (start of packet)
- EOP (end of packet)
- RESET

J-Zustand
Wechsel in K-Zustand
(D+)/(D-) beide 0, < 2.5 µs
(D+)/(D-) beide 0, > 2.5 µs



USB: Frames und Packets

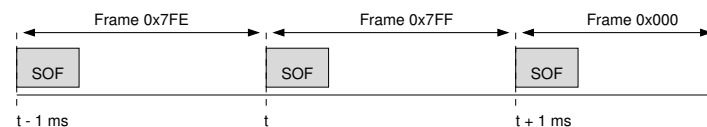


- 1-ms Zeitabschnitte (=: Frames) (12.000 bit/frame FS)
- Frame beginnt mit SOF-Token inkl. 11-bit Frame-Nummer
- Suspend-Mode, wenn keine SOF-Token
- besondere Verwaltung von low-speed Geräten
- SOF-Token:

SYNC	SOF	Frame #	CRC5	EOP	idle
00000001	0xA5	0x7EE	0x1D	*0*	11964 Takte

NRZI: sieben Wertewechsel Packet Identifier: SOF (D+)/(D-) beide Null Bus nicht getrieben

USB: leerer full-speed Bus



SYNC	SOF	Frame #	CRC5	EOP	idle
00000001	0xA5	0x7FD	0x1F	*0*	11965 Takte
SYNC	SOF	Frame #	CRC5	EOP	idle
00000001	0xA5	0x7FE	0x1D	*0*	11964 Takte
SYNC	SOF	Frame #	CRC5	EOP	idle
00000001	0xA5	0x7FF	0x02	*0*	11964 Takte
SYNC	SOF	Frame #	CRC5	EOP	idle
00000001	0xA5	0x000	0x08	*0*	11964 Takte

USB: Paket-Typen

PID-Name	PID (3..0)	PID (hex)	Gruppe
SOF	0101b	0xA5	Token
SETUP	1101b	0x2D	Token
IN	1001b	0x69	Token
OUT	0001b	0xE1	Token
DATA0	0011b	0xC3	Daten
DATA1	1011b	0x4B	Daten
ACK	0010b	0xD2	Handshake
NAK	1010b	0x5A	Handshake
STALL	1110b	0x1D	Handshake
PRE	1100b	0x39	Special

- 10 verschiedene USB Paket-Typen
- PID 7..4 = NOT (PID 3..0)
- Auswahl der Target-Function über Adresse und Endpoint (abhängig vom PID)

USB: Paket-Beispiele

SYNC	SOF	Frame #	CRC5	EOP
00000001	0xA5	0x7FD	0x1F	*0*

Start-of-Frame

SYNC	SETUP	ADDR	EP	CRC5	EOP
00000001	0x2D	0x01	0x00	0x17	*0*

Setup-Token

SYNC	IN	ADDR	EP	CRC5	EOP
00000001	0x69	0x03	0x01	0x07	*0*

In-Token

SYNC	OUT	ADDR	EP	CRC5	EOP
00000001	0xE1	0x02	0x02	0x01	*0*

Out-Token

SYNC	DATA0	data	CRC16	EOP
00000001	0xC3	00 11 22 33 44 55 66 77	0xCBA8	*0*

USB: 3-Phasen Datentransfer

Token	Sender Token-Phase	Sender Daten-Phase	Sender Handshake-Phase
SETUP	Host	Host	Function
OUT	Host	Host	Function
IN	Host	Function	Host

Beispiel: Host sendet Daten, Function quittiert (ACK oder NAK)

PC	idle	SYNC	OUT	ADDR	EP	CRC5	EOP
		00000001	0xE1	0x02	0x02	0x01	**

PC	idle	SYNC	DATA0	data	CRC16	EOP
		00000001	0xC3	00 11 22 33 44 55 66 77	0xCBA8	**

Maus	idle	SYNC	ACK	EOP
		00000001	0xD2	**

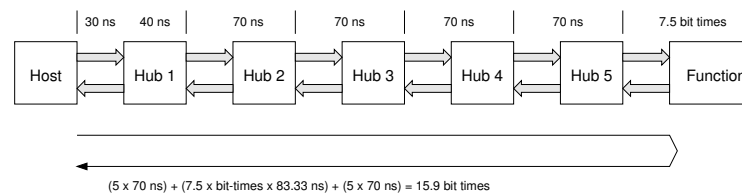
USB: Datentransfer

PC	idle	SYNC	SOF	Frame #	CRC5	EOP	
	11966	00000001	0xA5	0x002	0x15	**	
PC	idle	SYNC	IN	ADDR	EP	CRC5	EOP
	4	00000001	0x69	0x02	0x02	0x01	**
Maus	idle	SYNC	DATA0	data	CRC16	EOP	
	5	00000001	0xC3	00 11 22 33 44 55 66 77	0xCBA8	**	
PC	idle	SYNC	ACK	EOP			
	3	00000001	0xD2	**			
PC	idle	SYNC	SOF	Frame #	CRC5	EOP	
	11804	00000001	0xA5	0x003	0x0A	**	
PC	idle	SYNC	IN	ADDR	EP	CRC5	EOP
	4	00000001	0x69	0x02	0x02	0x01	**
Maus	idle	SYNC	DATA1	data	CRC16	EOP	
	5	00000001	0x4B	88 99 AA BB CC DD EE FF	0x8705	**	
PC	idle	SYNC	ACK	EOP			
	3	00000001	0xD2	**			

USB: Timeout

Datenübertragung zwischen Sender und Empfänger

- Mindestzeit bis zur Antwort / Handshake des Empfängers ?
- Bus-Turn-Around-Zeit:



- Empfänger hat 7 Takte Zeit für Antwort / Handshake: max. 16 Takte
- Timeout nach 18 Takten

USB: Deskriptoren

Plug and Play:

- Host muß alle USB-Functions identifizieren können
- wie findet der Host die benötigten Treiber?

=> Deskriptoren:	Code	
• Device Descriptor	0x01	
• Configuration Descriptor	0x02	(jeweils eine aktiv)
• String Descriptor	0x03	(optional)
• Interface Descriptor	0x04	
• Endpoint Descriptor	0x05	(FIFO-Größe etc.)
• Protokoll definiert Transfers zum Auslesen der Deskriptoren		

PC-Technologie | SS 2001 | 18.214

USB: Device Descriptor

Name	Offset	Länge	Beschreibung	Beispielwert
bLength	0	1	Größe in Bytes	0x12
bDescriptorType	1	1	= Device-Descriptor	0x01
bcdUSB	2	2	USB-Version (1.0,1,1.2.0)	0x0001
bDeviceClass	4	1	Klassen-Code	0x00
bDeviceSubClass	5	1	Subklassen-Code	0x00
bDeviceProtocol	6	1	Protokoll-Code	0x00
bMaxPacketSize0	7	1	EP0-FIFO in Byte	0x08
idVendor	8	2	Vendor-ID	0x3C05
idProduct	10	2	Produkt-ID (etwa 0x9084)	0x9084
bcdDevice	12	2	Versions-Nr. (1.02)	0x0102
iManufacturer	14	1	String-Index für	0x01
iProduct	15	1	Hersteller / Produkt	0x02
iSerialNumber	16	1	Seriennummer	0x03
bNumConfigurations	17	1	# Konfigurationen	0x01

- Vendor-ID wird vom USB Implementers Forum vergeben

PC-Technologie | SS 2001 | 18.214

USB: Configuration Descriptor

Name	Offset	Länge	Beschreibung	Beispielwert
bLength	0	1	Größe in Bytes	0x09
bDescriptorType	1	1	= Config-Descriptor	0x02
wTotalLength	2	2	Länge aller Desr. (zB 34)	0x0022
bNumInterfaces	4	1	Interface pro Konfiguration	0x01
bConfigurationValue	5	1	Nummer der Konfiguration	0x01
iConfiguration	6	1	String-Index	0x04
bmAttributes	7	1	z.B. Bus-powered, wakeup	0xA0
maxPower	8	1	in 2 mA Schritten	0x32

- Host liest Configuration-D. mitsamt allen Interface/Endpoint-D.
- SetConfiguration(0) deaktiviert das Gerät ("unkonfiguriert")
- weitere Datendefinitionen für Interfaces / Endpoints / Strings

PC-Technologie | SS 2001 | 18.214

USB: Device-States

- - nicht mit dem Bus verbunden
- attached angeschlossen, Hub benachrichtigt Host
- powered Hub aktiviert Versorgung, noch kein Reset
- default nach Reset, Device reagiert an Adresse 0
- address nach SetAddress()
- configured nach SetConfiguration() etc.
das Gerät kann jetzt benutzt werden
- suspended Stromsparen, wenn 3 ms keine Busaktivität

PC-Technologie | SS 2001 | 18.214

USB: Request-Codes

bRequest	Code
GET_STATUS	0
CLEAR_FEATURE	1
reserved	2
SET_FEATURE	3
reserved	4
SET_ADDRESS	5
GET_DESCRIPTOR	6
SET_DESCRIPTOR	7
GET_CONFIGURATION	8
SET_CONFIGURATION	9
GET_INTERFACE	10
SET_INTERFACE	11
SYNC_FRAME	12

- Control-Transfers
- auf EP0 Control-Pipe

USB: Enumeration

Plug and Play erfordert automatische Erkennung aller Geräte:

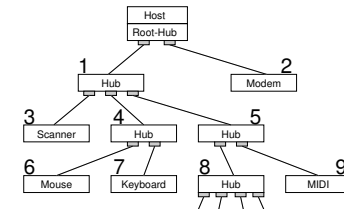
- 0 Hub erkennt Connect/Disconnect elektrisch
- 1 meldet dies beim nächsten Interrupt-Transfer
- 2 Host sendet Standard-Device-Request auf Adresse 0
- 3 Function antwortet mit seinem Device-Descriptor
- 4 Host sendet Reset an das Gerät
- 5 Host sendet SetAddress an das Gerät (noch auf Adresse 0)
- 6 Host fragt vollen Device-Descriptor ab
- 7 Host sucht geeigneten Treiber für Vendor-ID und Product-ID
- 8 Host fragt nach Configuration- und Device-Descriptor
- 9 Host konfiguriert das Gerät

[siehe Kelm: USB S94ff]

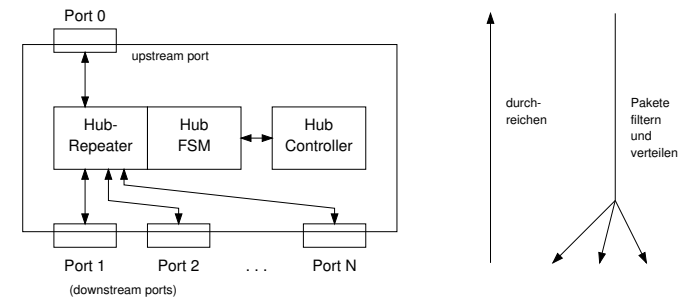
USB: Enumeration-Init

Erkennung aller Geräte beim Einschalten:

- alle Hub-Downstream-Ports zunächst deaktiviert
- Host sieht nur je ein Gerät pro Root-Hub Port
- Host kann oberstes Gerät konfigurieren
- Downstream-Ports aller Hubs werden nacheinander konfiguriert
- wahlweise Tiefen/Breitensuche



USB: Hub



- Hub-Controller ist eigenständige USB-Funktion
- Diagramm Downstream-Port-FSM
- Funktionalität zur Ansteuerung von low-speed functions

USB: Device-Klassen

USB-Klasse :=

- Gruppe von Geräten/Interfaces mit gemeinsamen Eigenschaften
- z.B. gleiche Datenformate, Kommunikationsverhalten, ...
 - Human-Interface-Device
 - Audio-Device
 - Communication-Device
 - Printer-Device
 - Monitor-Device
 - Power-Device
 - Mass-Storage-Device
 - ...
- Betriebssystem benötigt nur einen Treiber pro Klasse
- Treibersuche über Device/Config-Descriptor-Infos

PC-Technologie | SS 2001 | 18.214

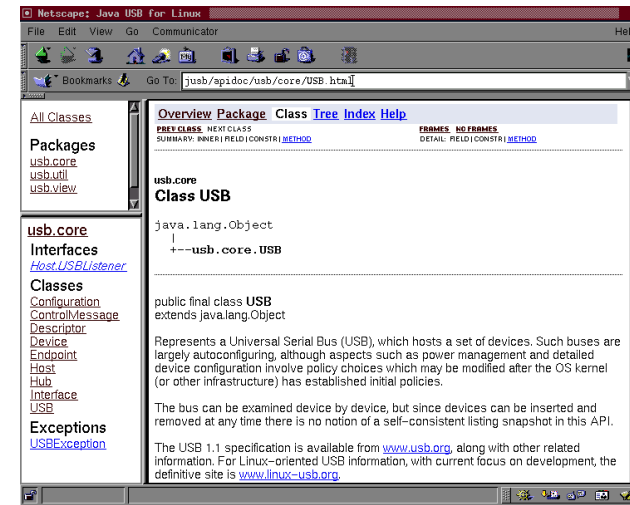
USB: Human Interface Device Class

Human-Interface-Device Class:

- Tastatur, Maus
- Schalter und Regler am PC
- Spielkomponenten wie Joystick, Datenhandschuh, Pedale, ...
- Geräte ohne menschliche Interaktion, aber mit ähnlichen Datenformaten: Barcode-Leser, Thermometer, Voltmeter, ...
- Definition eines universellen Protokollformats
- Report- und Physical-Descriptor
www.usb.org/developers/data/hidpage.htm
- besondere Behandlung von Tastatur und Maus (für Systemboot)

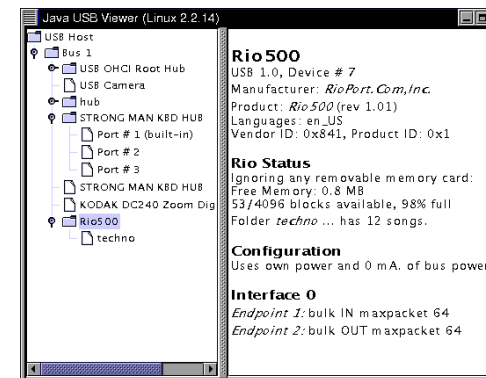
PC-Technologie | SS 2001 | 18.214

USB: Java USB: API



PC-Technologie | SS 2001 | 18.214

USB: Java USB Browser



- USB-Wrapper in Java, für Linux 2.4
- alpha-Status, fast keine Gerätetreiber

jusb.sourceforge.net

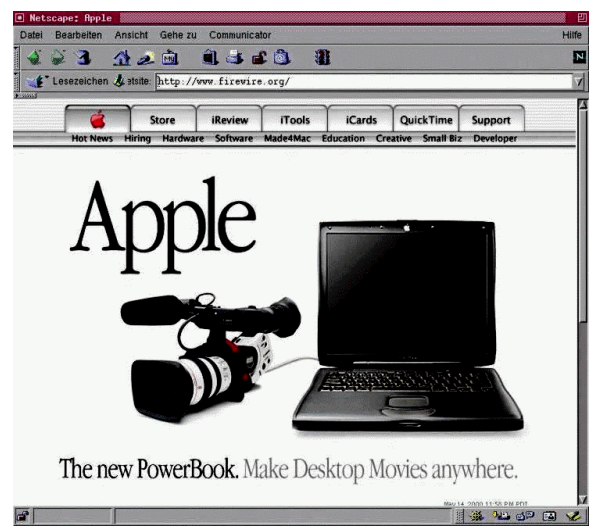
PC-Technologie | SS 2001 | 18.214

FireWire:

- entwickelt von Apple seit 1986
- IEEE Standard 1394-1995
- komplexer und schneller als USB: 100/200/400 Mb/s
- volle Autokonfiguration, kein zentraler Host
insbesondere: kein PC notwendig
- derzeit ca. 40 Geräte (www.apple.com)
- etabliert für digitales Video (8 Kanäle MPEG2: <120 Mb/s)
- diverse Erweiterungen / Zusatzprotokolle, zum Beispiel:
 - Audio and Music Transmission Protocol
 - IP over 1394
- Lizenzprobleme (derzeit 0.25\$ / Gerät)
- Microsoft: "IP-Landmines"



FireWire: Homepage



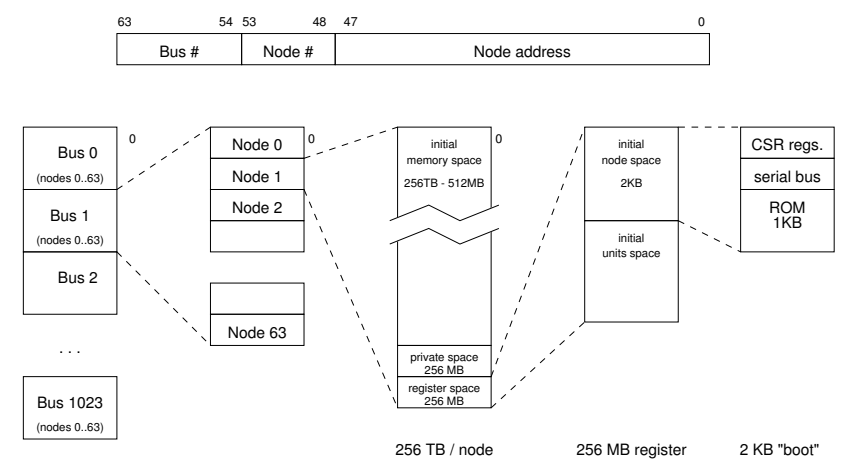
FireWire: Architektur

FireWire-System:

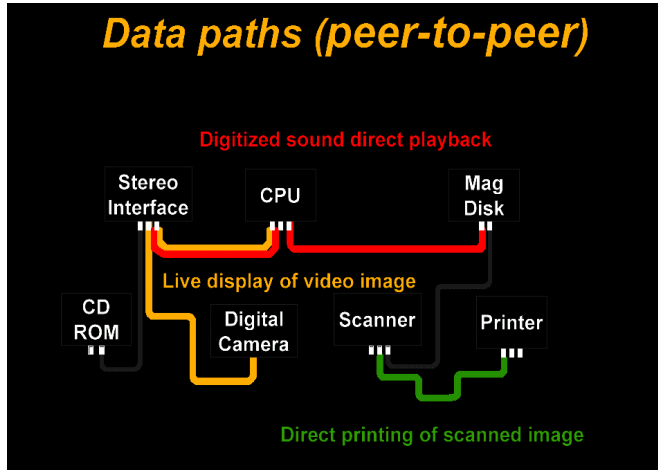
- bis 1023 separate Busse
- bis 63 Geräte pro Bus (bis 64.449 Geräte)
- 64-bit Adressraum (10 Bus, 6 Gerät, 48 bit pro Gerät)
- volle Autokonfiguration, keine Jumper/Terminatoren/...
- asynchrone und isochrone Nachrichten (@8KHz)
- bis 16 Kabelverbindungen (Hops) pro Bus
- Kabellängen bis 500m (gesamt)
- Kabel 6-adrig, 2x Daten differentiell, Versorgungsspannung
- 400 Mb/s bis 10m Kabellänge
- Peer-to-Peer Verbindungen (statt Master-Slave)
- Beispiel: direktes Drucken von Digitalkamera auf 1394-Drucker



FireWire: Adressraum

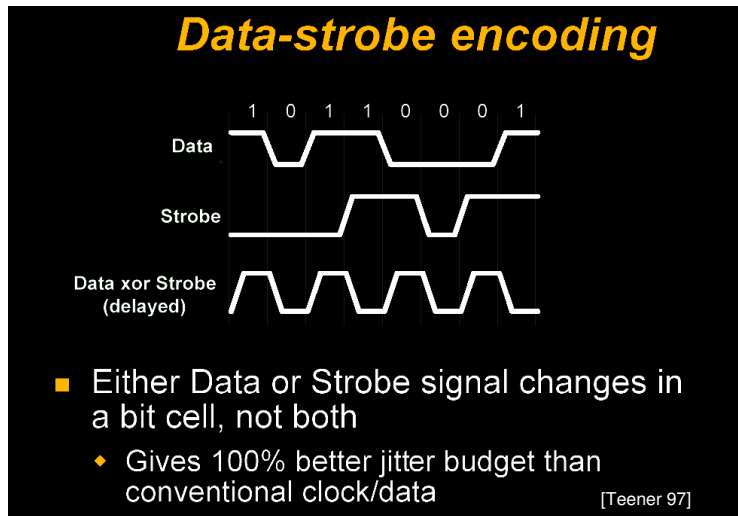


1394: Peer-to-peer transfers

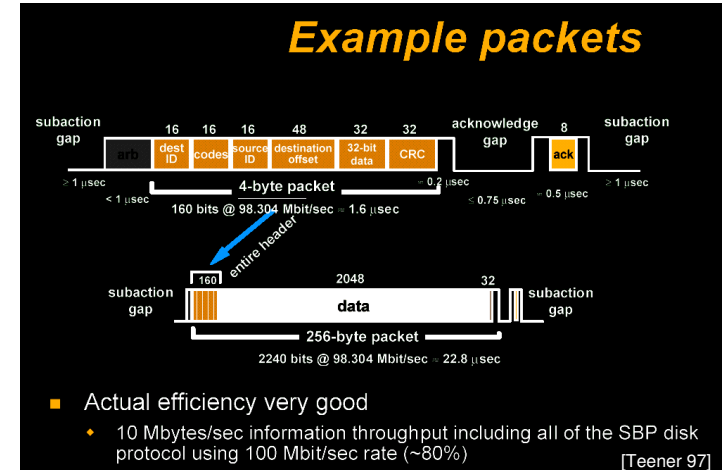


- andere Busse: fast immer zentrale Steuerung via Master

1394: Daten- und Taktsignal



1394: Beispiel Datenpaket



- aber geringe Effizienz für sehr kurze Pakete

FireWire: Arbitrierung

- kein zentraler Bus-Master
- direkte Kommunikation aller Geräte
- verteilte Arbitrierung, Fairness-Garantien



- nach jedem Anstecken/Abnehmen eines Geräts:
 - Bus-Reset (80 .. 300 ms)
 - Tree-Identification (10 .. 166 µs)
 - Self-Identification (70 µs)
 - ermittelt neuen Root-Node
 - sofern zyklensfrei
- Bus automatisch wieder betriebsbereit

1394: Tree Identification (1)

Tree identification #1

- After reset, each node only knows if it is a leaf (one connected port) or a branch (more than one connected port) [Teener 97]

1394: Arbitration (1)

Normal arbitration #1

- Suppose nodes #0 and #2 start to arbitrate at the same time, they both send a request to their parent ...

1394: Tree Identification (2)

Tree identification #2

- After Tree ID process, the Root node is determined and each port is labeled as pointing to a child or a parent
 - ◆ Root assignment is "sticky", will normally persist across a bus reset.

1394: Arbitration (2)

Normal arbitration #2

- The parents forward the request to their parent and deny access to their other children ...

1394: Arbitration (4)

Normal arbitration #4

The diagram shows a tree structure with a root node #4 (ch ch) and three children: leaf node #0 (p), branch node #3 (p, ch ch), and leaf node #2 (p). A green arrow labeled 'grant' points from the root to leaf node #0. Red arrows labeled 'deny' point from the root to branch node #3, and from branch node #3 to leaf node #2. A red arrow labeled 'data prefix' points from leaf node #0 back to the root.

- The winning node #0 changes its request to a data transfer prefix, while the losing node #2 withdraws its request ...

PC-Technologie | SS 2001 | 18.214

1394: Fairness

Fairness interval

The diagram shows a sequence of events on a timeline. It starts with 'fairness interval N-1' and 'fairness interval N+1'. In the middle is 'fairness interval N'. Within this interval, there are three owners: 'owner A', 'owner B', and 'owner M'. Each owner has a sequence of 'arb' (arbitration) and 'data' blocks. 'arb' blocks are separated by 'subaction gaps'. 'data' blocks are separated by 'subaction gaps'. The entire sequence is bounded by 'arbitration reset gaps'.

- Fairness Interval is bounded by "arbitration reset gaps"
- Reset gaps are longer than normal subaction gaps

- andere Busse: fast immer zentrale Steuerung via Master

PC-Technologie | SS 2001 | 18.214

1394: Arbitration (5)

Normal arbitration #5

The diagram shows a tree structure with a root node #4 (ch ch) and three children: leaf node #0 (p), branch node #3 (p, ch ch), and leaf node #2 (p). Red arrows labeled 'data prefix' point from leaf node #0 to the root, from branch node #3 to the root, and from leaf node #2 to branch node #3.

- The parent of node 1 sees the data prefix and withdraws the grant, and now all nodes are correctly oriented to repeat the packet data (a "deny" is a "data prefix!") ...

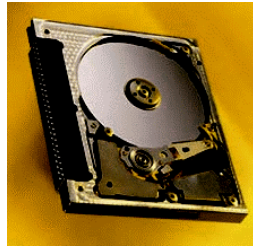
PC-Technologie | SS 2001 | 18.214

Leerseite

PC-Technologie

Disks: Agenda

- Festplatten
- IDE - Schnittstelle
- SCSI
- RAID
- Filecache/ OS-Strategien



PC-Technologie | SS 2001 | 18.214

Disks: "iron oxide valley"

"I think Silicon Valley was misnamed. If you look back at the dollars shipped in products in the last decade there has been more revenue from magnetic disks than from silicon. They ought to rename the place Iron Oxide Valley"

Al Hoagland, One of the Pioneers of Magnetic Disks (1982)
[Hennessy & Patterson, Computer Architecture, 6.2]

PC-Technologie | SS 2001 | 18.214

Disks: IBM Microdrive



- 340 MB, kleiner als PCMCIA-II Karte, 16 Gramm

PC-Technologie | SS 2001 | 18.214

Disks: Literatur

Friedhelm Schmidt:	SCSI-Bus und IDE-Schnittstelle, Addison-Wesley 93
H.-P. Messmer	PC-Hardwarebuch, Addison-Wesley 97

c't Plattenkarussell
c't SCSI-Einführung, Hefte 17/98/184, 18/98/144, 19/98/264

ATA-1 bis ATAPI-5 Spezifikationen
SCSI-1 bis SCSI-3 Spezifikationen
SCSI-3 MMC Spezifikation

www.seagate.com, www.quantum.com, www.storage.ibm.com

PC-Technologie | SS 2001 | 18.214

Disks: Plattenkarussell

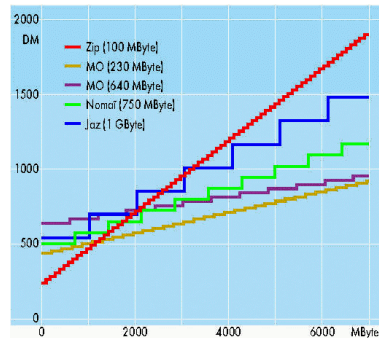
Festplatten mit EIDE-Schnittstelle												
Spezifikation	Kapazität	Drehzahl	Cache	Bauform	Lesebreite	Datenüberferanz			Gewicht/Mittelwert	Inter-face	Leistungsbuch	
						Access	Lesen	Schreiben			[MByte/s]	Leise
	[MByte]	[U/min]	[KByte]	[Zoll]	[ms]	min/mittel/max	min/mittel/max		[MByte/s]		[dB(A/Sone)]	[dB(A/Sone)]
besten >												
ST31621A ³³	1548	3600	64	3,5"	23,0/18,8	1,47/2,89/4,01	1,47/2,89/3,97	2,22	7,88	4	33,5/1,5	47,8/4,7
ST317242A Mediatek 17242	16447 ¹²	5400	512	3,5"	13,2/8,1	8,93/12,4/15,5	8,88/12,4/15,5		6,63	4	26,1/1,7	42,0/2,1
ST32122A Mediatek ¹²	2014	4500	128	3,5"	19,9/15,7	3,54/5,35/6,62	3,54/5,35/6,62		5,32	4	37,7/2,0	47,7/4,7
ST32140A ³⁰	2015	5400	128	3,5"	19,1/14,6	2,56/3,81/4,86	2,52/3,75/4,86	2,81	5,40	4	41,8/2,8	47,5/4,6
ST34321A Mediatek 4321 ¹⁷	4103	5400	128	3,5"	15,0/11,2	5,58/7,79/9,63	5,19/7,36/9,38		9,88	4	33,8/1,3	38,9/2,4
ST34342A Mediatek ¹⁹	4103	4500	128	3,5"	20,7/15,5	3,26/5,40/6,82	3,21/5,38/6,82		4,64	4	39,1/2,7	49,0/5,6
ST36450A Mediatek Pro ¹⁴	6149	5400	448	3,5"	16,7/11,1	5,08/6,90/8,65	4,69/6,72/8,65	4,68	7,23	4	38,4/2,1	45,5/4,1
ST36451A Mediatek Pro 6451 ¹⁴	6149	5400	448	3,5"	16,5/11,2	5,04/6,85/8,59	5,04/6,85/8,59	4,91	7,62	4	36,0/1,9	46,2/4,3
ST36530A Mediatek Pro 6530 ¹¹	6208	7200	448	3,5"	14,3/9,2	4,51/7,18/14,0	4,50/7,18/14,0	7,23	1,51	4	35,8/1,9	47,9/4,3
ST36531A Mediatek 6531 ¹⁷	6204	5400	128	3,5"	14,7/10,6	5,00/7,77/9,67	4,83/7,70/9,62	5,43	7,47	4	34,0/1,5	43,1/2,4
ST3666A [2EM HP] ¹	520	3800	120	3,5"	24,9/24,7	1,11/1,79/2,22	1,17/1,79/2,22	1,51	7,07	3	40,7/2,8	48,3/5,2
ST38421A U4 8421	8056 ¹²	5400	256	3,5"	13,6/9,4	8,89/12,9/16,0	8,80/12,9/16,0	7,47	5,40	4	30,5/1,1	42,3/2,4
ST38641A Mediatek 8641 ¹⁷	8207 ¹²	5400	128	3,5"	15,3/10,7	5,07/7,83/9,59	4,96/7,72/9,59	5,40	7,62	4	35,1/1,6	42,7/3,3
ST39140A Mediatek Pro 9140 ¹¹	8693 ¹²	7200	448	3,5"	14,4/8,9	3,40/11,3/14,0	3,59/11,4/14,0		7,61	4	36,8/2,1	50,7/5,1
ST51270A ³⁰	1223	5400	128	3,5"/0,75	19,7/16,4	2,34/3,65/4,62	2,26/3,58/4,61		4,64	4	38,6/2,5	43,1/3,3
ST52520A Mediatek Pro 2 5 ²	2446	5400	112	3,5"/0,75	16,0/12,0	4,45/6,74/8,56	4,83/6,74/8,56		4,72	4	35,6/1,8	46,1/3,9
Wechselspeicher												
AC11200i Caviar ⁹	1222	5200	256	3,5"	18,4/15,3	4,79/7,06/9,11	4,10/7,03/9,13	4,72	3,48	4	34,1/1,3	40,5/2,5
AC21200i Caviar ³⁰	1222	5200	128	3,5"	18,4/15,1	3,15/4,78/6,03	2,72/4,72/6,03	3,15	3,30 ³¹	4	37,4/2,1	48,5/4,1
AC21600i Caviar ¹⁴	1549	5200	128	3,5"	18,3/14,8	4,01/5,68/7,22	2,85/4,79/6,79	3,48	4,64	4	36,8/1,9	49,4/4,0
AC22100i Caviar ²¹	2014	5200	128	3,5"	18,4/13,8	4,10/5,19/7,90	3,49/5,71/7,89	3,48	4,64	4	38,1/2,2	49,4/4,4
AC22500i Caviar ⁹	2441	5200	256	3,5"	18,5/13,6	4,85/7,15/9,23	4,62/7,08/9,24	4,64	4,64	4	38,8/1,6	48,9/3,6
AC23200i Caviar ⁹	3098	5400	256	3,5"	16,7/11,9	5,70/8,26/9,85	5,71/8,18/9,85	4,56	4,64	4	36,6/1,4	48,7/4,0
AC24300i Caviar ⁹	4112	5400	256	3,5"	16,6/11,3	5,70/8,27/9,84	3,75/4,75/5,57	4,24	4,64	4	37,2/2,0	42,9/3,2
AC24600i Caviar ⁹	6149	5400	512	3,5"	16,4/10,5	7,57/10,5/12,5	7,57/10,5/12,5	6,49	10,6	4	32,5/1,1	45,7/3,4
AC28600i Caviar ²¹	8244 ¹²	5400	512	3,5"	14,9/10,3	8,14/11,3/13,1	8,14/11,3/13,1	6,04	7,47	4	31,5/1,1	42,7/3,7
AC291000i Expert ¹²	8693 ¹²	7200	1866	3,5"	13,3/9,1	10,3/14,8/17,0	10,1/14,3/17,0	10,6	7,29	4	41,0/2,5	44,1/2,5
AC310100i Caviar ⁹	9671 ¹²	5400	512	3,5"	16,4/10,3	7,57/10,8/12,7	7,57/10,8/12,7	4,73	4,64	4	33,8/1,3	48,9/4,1
AC313000i Caviar ²²	12417 ¹²	5400	512	3,5"	15,0/10,0	7,84/11,2/13,1	7,84/11,2/13,1	7,29	4,64	4	33,6/1,5	48,2/3,5

PC-Technologie | SS 2001 | 18.214

Leersseite

PC-Technologie

Disks: einige Wechselsplatten



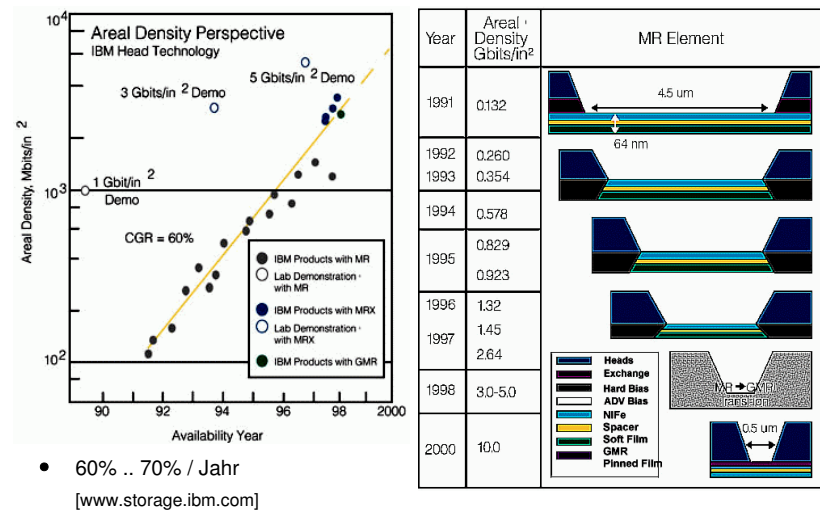
- diverse aktuelle Wechselsplatten, magnetisch/magnetooptisch
- Kapazität vs. Performance vs. Kosten vs. Kosten/MB
- MO bietet extreme Datensicherheit, aber schlechtere Performance

PC-Technologie | SS 2001 | 18.214

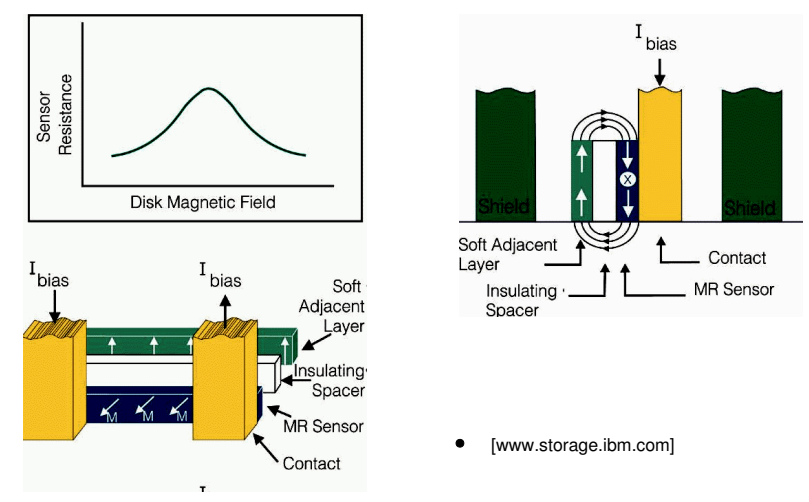
Leersseite

PC-Technologie

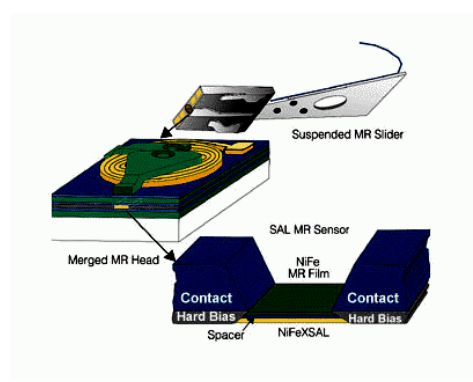
Disks: Trend



Disks: MR-Lesekopf: Prinzip

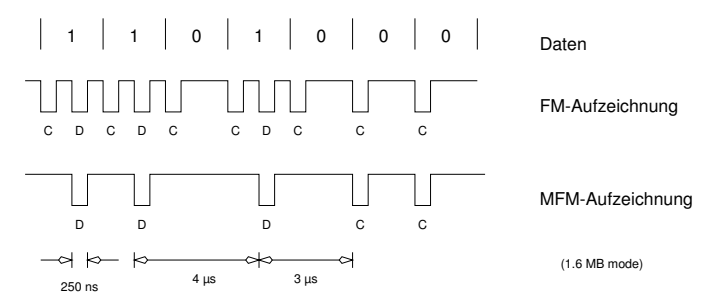


Disks: MR-Lesekopf: Aufbau



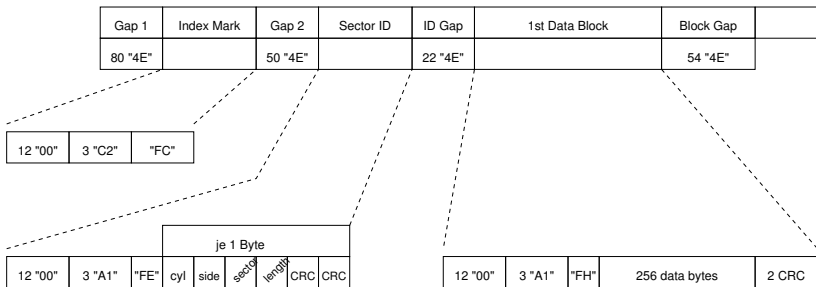
- Schreiben mit "normaler" Spule
- magnetoresistiver Lesekopf (MR)

Disks: FM/MFM Aufzeichnung (Floppy)



- Flusswechsel möglichst eng für hohe Speicherkapazität
- begrenzt durch Material, Lesekopf, oder Elektronik
- Frequenzmodulation verwendet Takt- und Datenimpulse
- MFM doppelte Kapazität
- Festplatten: Lauflängenkodierung (RLL) für höhere Kapazität

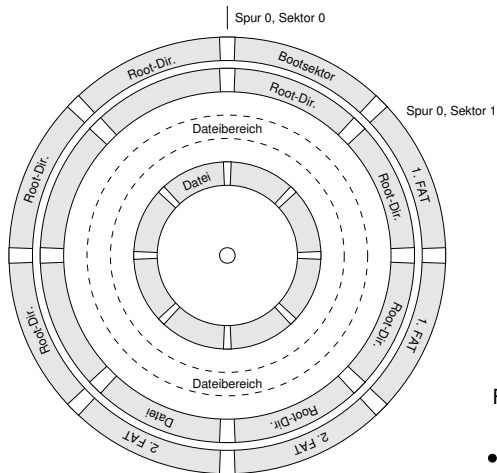
Disks: MFM Sektorformat



- keine separate Taktspur: selbsttaktend, spurführend
- muß Drehzahlschwankungen ausgleichen
- hohe Redundanz, spez. Taktmuster, CRC-Fehlerkorrektur
- Index-Markierungen für Spur/Sektornummer
- wird beim Formatieren erzeugt (nur Floppy)

PC-Technologie | SS 2001 | 18.214

Disks: Floppy-Sektorlayout



Floppy:

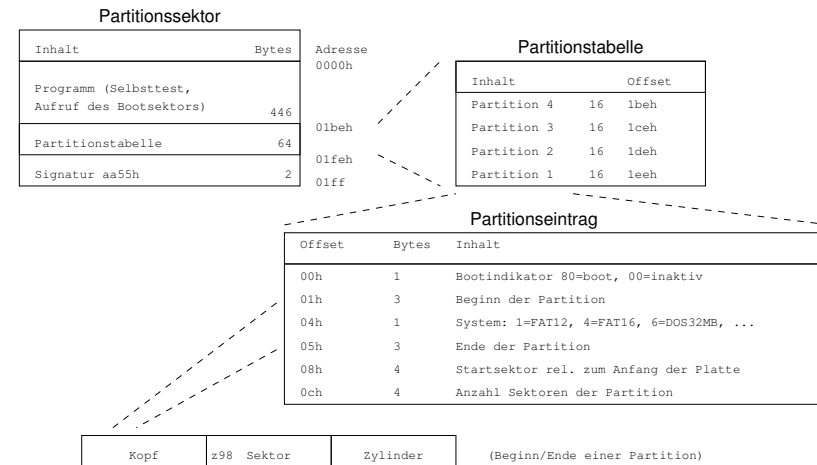
- Spur 0 außen:
- Bootsektor
- FATs
- Stammverzeichnis
- Dateien ab Spur 1

Festplatten:

- entsprechender Aufbau
- bad-sector management
- genaue Lage CHS unbekannt

PC-Technologie | SS 2001 | 18.214

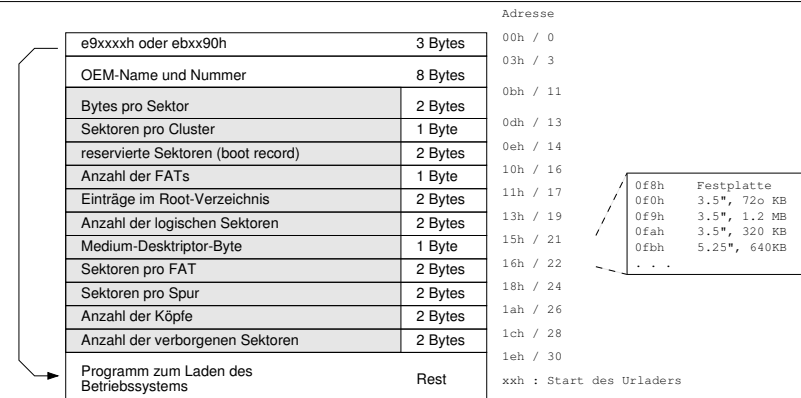
Disks: Partitionssektor



Selbsttest ab Adresse 0000, verzweigt zum Bootsektor

PC-Technologie | SS 2001 | 18.214

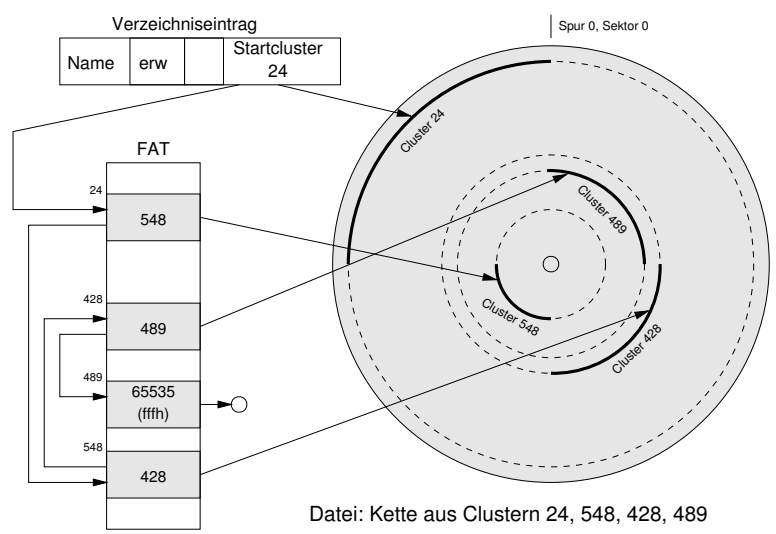
Disks: Bootsektor



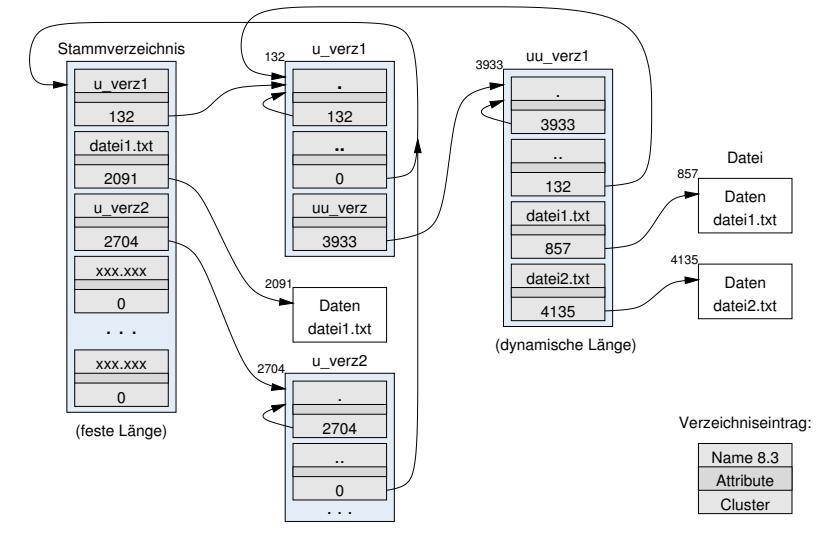
- erster Sektor der Partition (Kopf 0, Spur 0, Sektor 1)
- "Medium Descriptor Table" von 0bh .. 1eh
- ebxxxx: near jump xxxx / e9xx90: short jump xx nop

PC-Technologie | SS 2001 | 18.214

Disks: File Allocation Table (FAT)



Disks: DOS-Verzeichnisstruktur



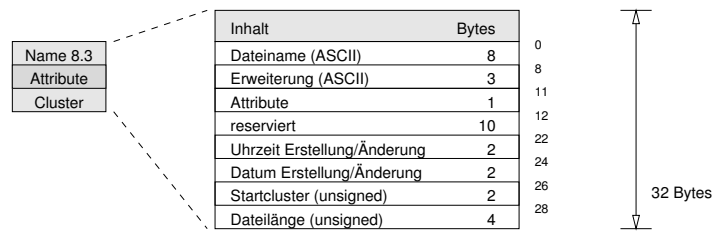
Disks: File Allocation Table

FAT-12	FAT-16	FAT-32	Bedeutung
000h	0000h	0000 0000h	frei
ff0h..ff6h	fff0h..fff6h	0fff fff0h..0fff fff6h	reserviert
ff7h	fff7h	0fff fff7h	defekter Sektor
ff8h..fffh	fff8h..ffffh	0fff fff8h..0fff ffffh	Ende der Clusterkette
xxxh	xxxxh	0xxx xxxh	nächster Cluster der Datei
4077	65517	2 ²⁸ = 256M	max. Anzahl der Cluster

- geringe Anzahl der Cluster in FAT-16 führt zu riesigen Clustern:
- ungeeignet für große Platten / Vielzahl von kleinen Dateien

Kapazität	Clustergröße (FAT-16)
16..128 MB	2 KB (4 Sektoren)
128..256 MB	4 KB (8 Sektoren)
256..512 MB	8 KB (16 Sektoren)
512..1024 MB	16 KB (32 Sektoren)
1024..2048 MB	32 KB (64 Sektoren)

Disks: DOS-Verzeichniseintrag

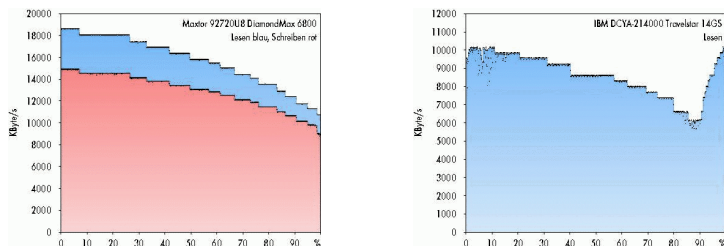


Dateiname:

- 8 Zeichen ASCII, 3 Zeichen Erweiterung
- Name '2eh' bzw. '.' bedeutet Verzeichnis, '..' das Stammverzeichnis
- Name 'e5h' bedeutet "gelöscht"

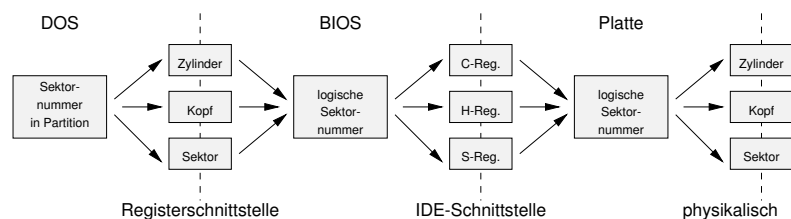
Disks: Zonenmessung

Anordnung der logischen Blöcke auf der Platte?



- c't-Messung: R/W-Transferraten als Funktion der Blockadresse
- viele Varianten möglich
- schnellste Zone (außen) meistens bei Adresse 0
- gibt es ein "optimales" Mapping?

Disks: BIOS/CHS/LBA-Adressierung



Adressierung von Daten auf einer Platte:

- CHS: Cylinder, Head, Sektor
- LBA, logical block addressing (fortlaufend ab 0)
- DOS/BIOS zu wenig Bits: Probleme bei 504M, 2G, 8G, ...
- herstellerspezifisches Mapping LBA - Sektor der Platte
- "Zonenmessung"

BIOS: 528 MByte Grenze (int13h)

3.2 The 528-megabyte barrier

BIOSs provide Int 13h services for accessing ATA drives from DOS. For conventional Int 13h the Cylinder-Head-Sector (CHS) values supplied to the Int 13h interface were passed to the drive without modification. This method of access allows "ill-behaved" applications to successfully access the drive, bypassing the BIOS's Int 13h interface. ATA drives support more than 1024 cylinders but the Int 13h interface is limited to 1024, this prevents the BIOS from accessing the full media by passing CHS values directly to the drive. Table 1 illustrates the limitations caused by the differences between the Int 13h and ATA maximum geometries.

Table 1 – Disk drive min/max

	BIOS	ATA	Limit
Max sectors/track	63	255	63
Max heads	256	16	16
Max cylinders	1024	65536	1024
Capacity	8.4 GB	136.9 GB	528 MB

This table illustrates how the conventional Int 13h interface with an 8.4 GB limit is restricted to 528 MB (63 * 16 * 1024 * 512). One solution to this problem is to address the drive using the Int 13h Extensions described in this technical report. Another solution is to create a false geometry that "fits" within Int 13h limitations, and also uses the full capacity of the drive. This capability is called geometric or drive translation. The translated geometry is applied in a manner that causes all sectors to maintain the same physical location on the media as when the drive is used in an untranslated environment. The Int 13h interface only has 10 bits for the cylinder, therefore Int 13h Fn 08h always returns the altered geometry information. This allows all DOS applications to function normally. Windows™ 3.11 and below functions normally when 32-bit disk access mode is disabled. A Windows™ driver which supports the geometry reported by Int 13h Fn 08h is required for 32-bit protected disk access mode.

BIOS: bit shifting

A simple bit-shift mapping scheme may create altered drive geometries. This method has the advantage of working with all ATA drives, including those drives which do not support LBA. A second advantage is that operation is fast and the code is small. The disadvantage of this method is that it lacks the flexibility to translate all geometries reported by a drive with a capacity less than 8.4 GB. However, drives which are ATA-2 (X3.279-1996) and above compatible will report geometries that may be translated. Annex D of ATA-2 or Annex B of ATA-3 and ATA/ATAPI-4 place limits on geometries for drives with less than an 8.4 GB capacity. The bit-shift method of translation manipulates the head and cylinder part of the geometry, but not the sectors per track. Table 2 describes the bit-shift translation capability:

Table 2 – Bit Shift Translation

Actual cylinders	Actual heads	Altered cylinder	Altered heads (see note)	Approx. size
1<C≤1024	1<H≤16	C=C	H=H	528 MB
1024<C≤2048	1<H≤16	C=C/2	H=H*2	1 GB
2048<C≤4096	1<H≤16	C=C/4	H=H*4	2.1 GB
4096<C≤8192	1<H≤16	C=C/8	H=H*8	4.2 GB
8192<C≤16384	1<H≤16	C=C/16	H=H*16	8.4 GB
16384<C≤32768	1<H≤8	C=C/32	H=H*32	8.4 GB
32768<C≤65536	1<H≤4	C=C/64	H=H*64	8.4 GB
NOTE – Value can not be greater than 255 in some Operating Systems.				

BIOS: LBA translation

Table 3 – LBA assist translation

Range	Sectors	Heads	Cylinders
1 < X ≤ 1,032,192	63	16	X/(1,008)
1,032,192 < X ≤ 2,064,384	63	32	X/(2,016)
2,064,384 < X ≤ 4,128,768	63	64	X/(4,032)
4,128,768 < X ≤ 8,257,536	63	128	X/(8,064)
8,257,536 < X ≤ 16,450,560	63	255	X/(16,065)

NOTE – X is the capacity of the drive, calculated by multiplying words 1, 3, and 6 of the IDENTIFY DEVICE data. This number may be different than the drive size reported by IDENTIFY DEVICE words 60 and 61.

These two translation methods yield similar geometries in many cases. The difference between the two translations methods becomes apparent when a drive reports less than 63 sectors per track. The LBA assisted method always assigns a geometry with 63 sectors per track. The bit-shift method uses the sectors returned by

- evtl. andere Resultate als "bit-shifting"-Technik
- beide Varianten: bis max. 16 GByte
- beide Varianten: Platte nach BIOS-Wechsel evtl. nicht mehr lesbar

BIOS: extended BIOS translation

Table 8 – Device address packet

Offset	Type	Description
0	Byte	Packet size in bytes. Shall be 16 (10h) or greater. If the packet size is less than 16 the request is rejected with CF=1h and AH=01h. Packet sizes greater than 16 are not rejected, the additional bytes beyond 16 shall be ignored.
1	Byte	Reserved, must be 0
2	Byte	Number of blocks to transfer. This field has a maximum value of 127 (7Fh). A block count of 0 means no data is transferred. If a value greater than 127 is supplied the request is rejected with CF=1 and AH=01.
3	Byte	Reserved, must be 0
4	Double word	Address of transfer buffer. This is the buffer which Read/Write operations will use to transfer the data. This is a 32-bit address of the form Seg:Offset.
8	Quad word	Starting logical block address, on the target device, of the data to be transferred. This is a 64 bit unsigned linear address. If the device supports LBA addressing this value should be passed unmodified. If the device does not support LBA addressing the following formula holds true when the address is converted to a CHS value: $LBA = (C_1 * H_0 + H_1) * S_0 + S_1 - 1$ Where: C ₁ = Selected Cylinder Number H ₀ = Number of Heads (Maximum Head Number + 1) H ₁ = Selected Head Number S ₀ = Maximum Sector Number S ₁ = Selected Sector Number For ATA compatible drives, with less than or equal to 15,482,880 logical sectors, the H ₀ and S ₀ values are supplied by WORDS 3 and 6 of the IDENTIFY DEVICE command.

- lineare 64-bit LBA-Adressierung

BIOS: extended read/write commands

4.2.2 Extended read

Entry:
 AH - 42h
 DL - Drive number
 DS:SI - Disk address packet

Exit:
 carry clear
 AH - 0
 carry set
 AH - error code

This function transfer sectors from the device to memory. In the event of an error, the block count field of the disk address packet contains the number of good blocks read before the error occurred.

4.2.3 Extended write

Entry:
 AH - 43h
 AL - 0 or 1, write with verify off
 2, write with verify on
 DL - Drive number
 DS:SI - Disk address packet

Exit:
 carry clear
 AH - 0
 carry set
 AH - error code

This function transfer sectors from memory to the device. If write with verify is not supported, this function rejects the request with AH=01h, CF=1. Function 48h is used to detect if write with verify is supported. In the event of an error, the block count field of the disk address packet contains the number of blocks written before the error occurred. AL also contains the values 0, 1, or 2. This function rejects all other values with AH=01h, CF=1

BIOS: extended BIOS detection

4.2.1 Check extensions present

Entry:
 AH - 41h
 BX - 55AAh
 DL - Drive number

Exit:
 carry clear
 AH - Version of extensions
 AL - Internal use only
 BX - AA55h
 CX - Interface support bit map (see Table 9)
 carry set
 AH - error code (01h, Invalid Command)

Table 9 – Extension result buffer

Bit	Description
0	1 - Fixed disk access subset
1	1 - Drive locking and ejecting subset
2	1 - Enhanced disk drive support subset
3-15	Reserved, must be 0

This function is used to check for the presence of Int 13h extensions. If the carry flag is returned set, the extensions are not supported for the requested drive. If the carry flag is returned cleared, BX shall be checked for the value AA55h to confirm that the extensions are present. If BX is AA55h, the value of CX is checked to determine what subsets of this interface are supported for the requested drive. At least one subset must be supported. The version of the extensions is 21h. This indicates that the Int 13h extensions are compliant with this technical report.

- lineare 64-bit LBA-Adressierung

BIOS: extended BIOS device parameters

Table 4 – Standard device parameter table

Byte	Type	Description
0-1	Word	Physical number of cylinders
2	Byte	Physical number of heads
3	Byte	Not Axh signature, indicates untranslated table
4	Byte	Reserved
5-6	Word	Precompensation (obsolete)
7	Byte	Reserved
8	Byte	Drive control byte
9-10	Word	Reserved
11	Byte	Reserved
12-13	Word	Landing zone (obsolete)
14	Byte	Sectors per track
15	Byte	Reserved

Table 5 – Translated device parameter table

Byte	Type	Description
0-1	Word	Logical cylinders, limit 1024
2	Byte	Logical heads, limit 256 (see note)
3	Byte	Axh signature, indicates translated table
4	Byte	Physical sectors per track, limit 63
5-6	Word	Precompensation (obsolete)
7	Byte	Reserved
8	Byte	Drive control byte
9-10	Word	Physical cylinders, limit 65536 (see note)
11	Byte	Physical heads, limit 16 (see note)
12-13	Word	Landing zone (obsolete)
14	Byte	Logical sectors per track, limit 63
15	Byte	Checksum, 2's complement of the 8 bit unsigned sum of bytes 0-14

NOTE – 0 indicates the maximum value. See table 2.

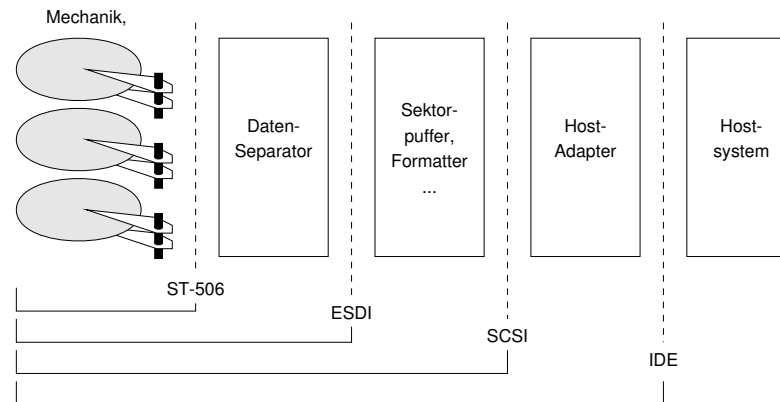
- siehe "extended BIOS" specification

Disks: IDE-Schnittstelle

IDE	"integrated drive electronics"
EIDE	"enhanced IDE"
ATA	"AT attachment"
ATAPI	AT attachment packet interface

- Anschluss von Festplatten an den AT-Bus
- minimaler Hardwareaufwand des Interfaces (=billig)
- registerkompatible Variante eines WD ST506 Controllers
- vollständiger ST506-Controller in der Platte integriert (=IDE)
- mittlerweile standardisiert (ATA-1, 2, 3, 4, ATAPI, MMC, ...)
- Anschluss für Fest- und Wechsellplatten, CD-Laufwerke, usw.
- derzeit fast immer im PC-Chipsatz integriert
- siehe ATAPI-5 Spezifikation

Disks: ST506 vs. SCSI vs. ATAPI



- IDE/ATA-Platte enthält kompletten Controller und Adapter

ATAPI: Signale

Table A.3 – 40-pin I/O connector interface signals

Signal name	Connector contact	Conductor	Connector contact	Signal name
RESET-	1	1	2	Ground
DD7	3	3	4	DD8
DD6	5	5	6	DD9
DD5	7	7	8	DD10
DD4	9	9	10	DD11
DD3	11	11	12	DD12
DD2	13	13	14	DD13
DD1	15	15	16	DD14
DD0	17	17	18	DD15
Ground	19	19	20	(keypin)
DMARQ	21	21	22	Ground
DIOW-STOP	23	23	24	Ground
DIOR-HDMARDY-HSTROBE	25	25	26	Ground
IORDYDDMARDY-IDSTROBE	27	27	28	CSEL
DMACK-	29	29	30	Ground
INTRQ	31	31	32	Obsolete (see note)
DA1	33	33	34	FDIACS-OBUID-
DA0	35	35	36	DA2
CS0-	37	37	38	CS1-
DASP-	39	39	40	Ground

NOTE – Pin 32 was defined as IOCS16 in ATA-2, ANSI X3.279-1996.

- billiges 40-pol. Flachbandkabel
- Signale praktisch identisch mit den ISA-Bus Signalen
- seit kurzem auch 80-pol. Kabel für Ultra-DMA Modi

[ATAPI-5 Spec.]

ATAPI: ATA-Register

Table F.4 – Register functions and selection addresses except PACKET and SERVICE commands

Addresses					Functions	
CS0-	CS1-	DA2	DA1	DA0	Read (DIOR-)	Write (DIOW-)
N	N	x	x	x	Released	Not used
Control block registers						
N	A	N	x	x	Released	Not used
N	A	A	N	x	Released	Not used
N	A	A	A	N	Alternate Status	Device Control
N	A	A	A	A	Obsolete(see note)	Not used
Command block registers						
A	N	N	N	N	Data	Data
A	N	N	N	A	Error	Features
A	N	N	A	N	Sector Count	Sector Count
A	N	N	A	A	Sector Number	Sector Number
A	N	A	N	N	Cylinder Low	Cylinder Low
A	N	A	N	A	Cylinder High	Cylinder High
A	N	A	A	N	Device/Head	Device/Head
A	N	A	A	A	Status	Command
A	A	x	x	x	Released	Not used

Key:
 A = signal asserted N = signal negated x = don't care
 NOTE – This register is obsolete. It is recommended that a device not respond to a read of this address.

0
4
6
7
0
1
2
3
4
5
6
7

[ATAPI-5 Spec.]

- Host schreibt Parameter in Register 1-6
- Befehl starten durch Schreiben auf Register 7
- Datenübergabe nacheinander über das Data-Register 0

ATAPI: Befehle (Ausschnitt)

Command Name	Op Code	Type	Sub-clause
BLANK	A1h		6.1.1.
CLOSE TRACK/SESSION	5Bh		6.1.2.
FORMAT UNIT	04h		6.1.3.
INQUIRY	12h	M	SPC
LOAD/UNLOAD C/DVD	A6h	O	6.1.5.
MECHANISM STATUS	BDh	M	6.1.6.
MODE SELECT (6)	15h	M	SPC
MODE SENSE (10)	5Ah	M	SPC
MODE SENSE (6)	1Ah	M	SPC
PAUSE/RESUME	4Bh	A	6.1.7.
PLAY AUDIO (10)	45h	A	6.1.8.
PLAY AUDIO (12)	A5h	A	6.1.9.
PLAY AUDIO MSF	47h	A	6.1.10.
PLAY C/DVD	BCh	O	6.1.11.
PREVENT/ALLOW MEDIUM REMOVAL	1Eh	M	SPC
READ (10)	28h	M	SPC
READ BUFFER CAPACITY	5Ch		6.1.12.
READ C/DVD	BEh	O	6.1.13.
READ C/DVD MSF	B9h	O	6.1.14.
READ C/DVD RECORDED CAPACITY	25h	M	6.1.15.
READ DISC INFORMATION	51h		6.1.16.
READ DVD STRUCTURE	ADh		6.1.17.
READ HEADER	44h	M	6.1.18.
READ MASTER CUE	59h		6.1.19.
READ SUB-CHANNEL	42h	M	6.1.21.

ATAPI: Register für Packet-Command

Table F.5 – Register functions and selection addresses for PACKET and SERVICE commands

Addresses					Functions	
CS0-	CS1-	DA2	DA1	DA0	Read (DIOR-)	Write (DIOW-)
N	N	x	x	x	Released	Not used
Control block registers						
N	A	N	x	x	Released	Not used
N	A	A	N	x	Released	Not used
N	A	A	A	N	Alternate Status	Device Control
N	A	A	A	A	Obsolete(see note)	Not used
Command block registers						
A	N	N	N	N	Data	Data
A	N	N	N	A	Error	Features
A	N	N	A	N	Interrupt reason	
A	N	N	A	A		
A	N	A	N	N	Byte count low	Byte count low
A	N	A	N	A	Byte count high	Byte count high
A	N	A	A	N	Device select	Device select
A	N	A	A	A	Status	Command
A	A	x	x	x	Released	Not used

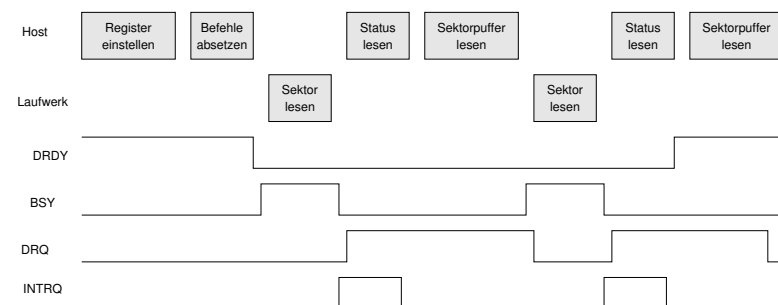
Key:
 A = signal asserted N = signal negated x = don't care
 NOTE – This register is obsolete. A device should not respond to a read of this address.

0
4
6
7
0
1
2
3
4
5
6
7

[ATAPI-5 Spec.]

- CD/CDR/DVD haben andere Organisation als Festplatten
- Packet-Command definiert neue Bedeutung der Register
- Datentransfer wie bei normalen ATA-Befehlen

ATAPI: Prinzip PIO-Lesezugriff



- Laufwerk liest/schreibt jeweils ganzen Sektor
- PIO Host liest/schreibt jedes Datenwort einzeln
- DMA Datentransfer via DMA mit vollem Handshake
- Ultra-DMA DMA ohne Handshake, aber mit CRC

ATAPI: PIO-Modi 0 .. 4

Table 49 – PIO data transfer to/from device

PIO timing parameters	Mode 0 ns	Mode 1 ns	Mode 2 ns	Mode 3 ns	Mode 4 ns	Note
t_0 Cycle time (min)	600	383	240	180	120	1,4
t_1 Address valid to DIOR-/DIOW- setup (min)	70	50	30	30	25	
t_2 DIOR-/DIOW- (min)	165	125	100	80	70	1
t_3 DIOR-/DIOW- recovery time (min)	-	-	-	70	25	1
t_4 DIOW- data setup (min)	60	45	30	30	20	
t_5 DIOW- data hold (min)	30	20	15	10	10	
t_6 DIOR- data setup (min)	80	35	20	20	20	
t_7 DIOR- data hold (min)	5	5	5	5	5	
t_8 DIOR- data tristate (max)	30	30	30	30	30	2
t_9 DIOR-/DIOW- to address valid hold (min)	20	15	10	10	10	
t_{RD} Read Data Valid to IORDY active (if IORDY initially low after t_4) (min)	0	0	0	0	0	
t_A IORDY Setup time	35	35	35	35	35	3
t_B IORDY Pulse Width (max)	1250	1250	1250	1250	1250	
t_C IORDY assertion to release (max)	5	5	5	5	5	

[ATAPI-5 Spec.]

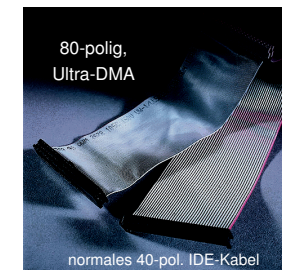
NOTES –
 1 t_0 is the minimum total cycle time, t_2 is the minimum DIOR-/DIOW- assertion time, and t_3 is the minimum DIOR-/DIOW- negation time. A host implementation shall lengthen t_1 and/or t_4 to ensure that t_0 is equal to or greater than the value reported in the device's IDENTIFY DEVICE data. A device implementation shall support any legal host implementation.
 2 This parameter specifies the time from the negation edge of DIOR- to the time that the data bus is released by the device.
 3 The delay from the activation of DIOR- or DIOW- until the state of IORDY is first sampled. If IORDY is inactive then the host shall wait until IORDY is active before the PIO cycle is completed. If the device is not driving IORDY negated at the t_4 after the activation of DIOR- or DIOW-, then t_4 shall be met and t_0 is not applicable. If the device is driving IORDY negated at the time t_4 after the activation of DIOR- or DIOW-, then t_{RD} shall be met and t_4 is not applicable.
 4 Mode may be selected at the highest mode for the device if CS(1,0) and AD(2,0) do not change between read or write cycles or selected at the highest mode supported by the slowest device if CS(1,0) or AD(2,0) do change between read or write cycles.

- Protokoll/Handshake immer gleich, unterschiedliche Wartezeiten

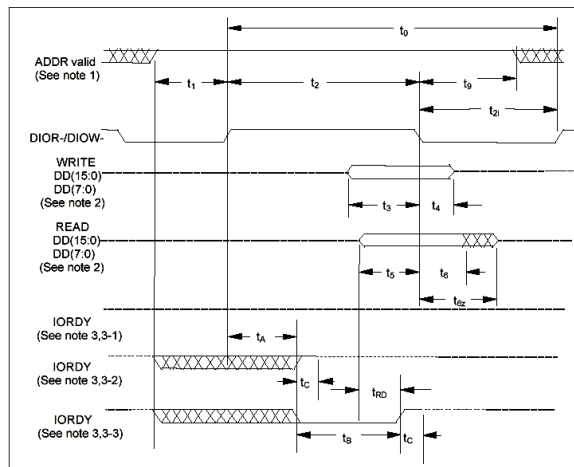
ATAPI: Ultra-DMA

- aktuelles, derzeit schnellstes Übertragungsverfahren
- Ultra-DMA/66 bis 66 MB/s

- Sender (Host/Platte) schickt Daten und Strobe-Impulse
- reduziertes Handshake
- dafür CRC-Fehlerkorrektur
- erfordert neues 80-pol. Kabel
- Anordnung abwechselnd Daten/Masseleitung



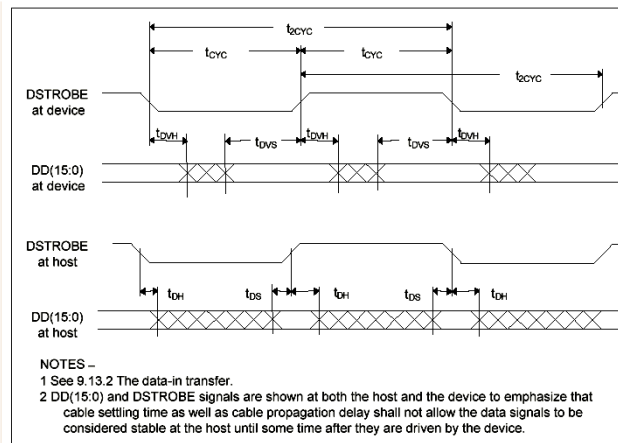
ATAPI: PIO Waveforms



[ATAPI-5 Spec.]

- Host kontrolliert und initiiert alle Transfers

ATAPI: Ultra-DMA Waveforms



[ATAPI-5 Spec.]

NOTES –
 1 See 9.13.2 The data-in transfer.
 2 DD(15:0) and DStrobe signals are shown at both the host and the device to emphasize that cable settling time as well as cable propagation delay shall not allow the data signals to be considered stable at the host until some time after they are driven by the device.

Figure 50 – Sustained Ultra DMA data-in burst

- kein Handshake, jeweiliger Sender steuert Daten und Strobe

ATA: Marktbedeutung

**126 Million Units and 87%
ATA must be doing something right!**

Mobile+Desktop represent
126 MU in '98 and 87% of
shipments. Category
dominated by ATA.

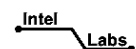
Projections do not
forecast any substantial
change in the mix

Disk Drive Unit Shipments** (in thousands)

	Shipments		Forecast		
	1998	1999	2000	2001	2002
Mobile Drives	17846	20990	24340	28215	32600
Desktop Drives	108628	125646	143780	163180	184200
Server Drives	18493	21718	25700	30550	36130
Total	144967	168354	193820	221945	252930



**1999 Disk/Trend report at IIST Lk. Arrowhead



ATA: Serial-ATA

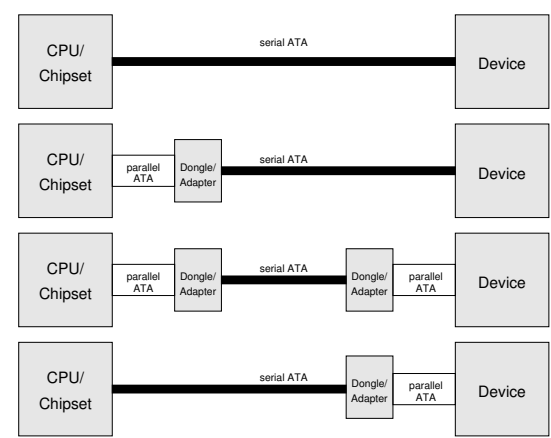
parallele Datenübertragung problematisch:

- teure Kabel
- Skew-Probleme
- höhere Taktraten als 100 MHz schwierig

=> Umstellung auf serielle Datenübertragung
"Serial-ATA"

- Beibehalten des ATAPI-Befehlssatzes
- volle Kompatibilität
- Unterstützung durch alle großen Hersteller
- bei Bedarf "Dongles" zur Parallel/Seriell-Umwandlung

ATA: Serial-ATA Dongles

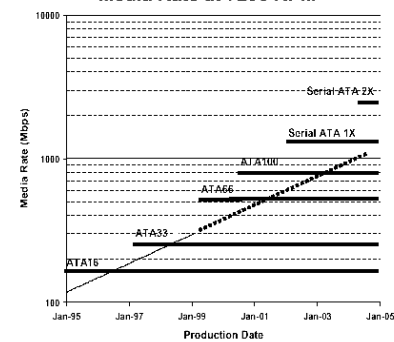


- bei Bedarf "Dongles" zur Parallel/Seriell-Umwandlung
- alte Hardware kann weiter genutzt werden, einfache Migration

ATA: Serial-ATA Roadmap

Another Driving Factor

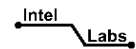
Media Rate at 7200 RPM



Interface rate driven to change with media rate
– Minimizes buffering problems

An ATA133 transition seems unnecessary

Interface takes a couple years to develop & deploy so some degree of developing in anticipation of the need is prudent



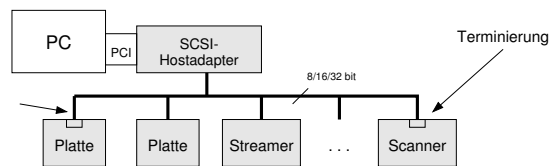
SCSI: Übersicht

SCSI := Small Computer Systems Interface

- hervorgegangen aus "Shugart Associates SI"
- standardisiert als SCSI-I, SCSI-II, SCSI-III
- Einsatz in PCs (Server), Mac, Workstations
- keine reine Festplattenschnittstelle
- sondern universeller Bus für Peripheriegeräte ("Targets")
- z.B. Bandlaufwerke, Scanner, Musiksynthesizer, ...
- 8-bit parallel (wide-SCSI mit 16-/32-bit)
- "Hostadapter" steuert den Bus
- komplexe Befehle und Arbitrierung
- flexibler, aber auch teuer und komplexer als EIDE/ATAPI
- Praxistips in der Artikelserie in ct 17-19/98

PC-Technologie | SS 2001 | 18.214

SCSI: Grundlagen

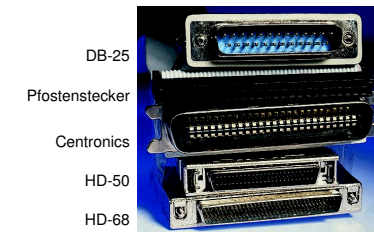
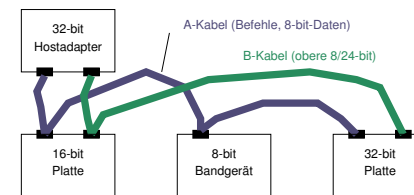


- Bus mit 8 Geräten (LUN 0..7), inklusive Controller
- Gerätenummer per Schalter eingestellt (nicht automatisch!)
- komplexe Regeln zur Verkabelung (Terminierung, Abstände)
- parallele Datenübertragung, 8-bit oder (wide) 16/32-bit
- aufwendiges Busprotokoll mit Arbitrierung und split-transactions
- Geräte handeln die jeweils bestmögliche Geschwindigkeit aus
- langsame Geräte stören schnelle Geräte nicht

PC-Technologie | SS 2001 | 18.214

SCSI: Varianten

- Befehlssätze: SCSI-1, SCSI-2, SCSI-3
- Busbreite: normal 8-bit, wide-SCSI 16-bit und 32-bit
- Bustiming: SCSI-1 bis 5 MB/s, Fast 10 MB/s, Ultra 20 MB/s
- alle Kombinationen, z.B. U2W = Ultra-Wide SCSI-2
- alle Gerätevarianten miteinander kombinierbar
- insbesondere auch normale und wide-SCSI Geräte



PC-Technologie | SS 2001 | 18.214

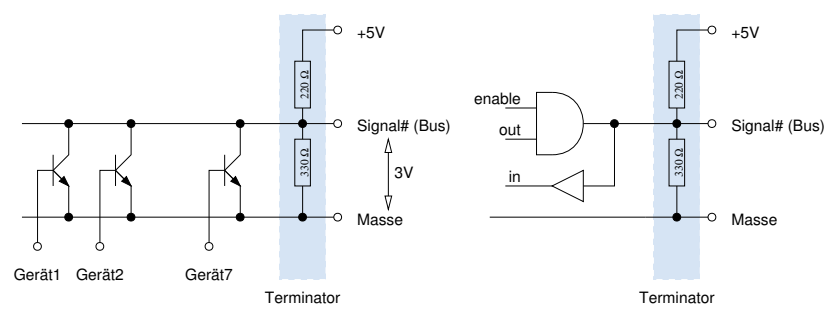
SCSI: Signale

Signal name	Connector contact number		Cable conductor number	Connector contact number		Signal name
	Set 2	Set 1		Set 1	Set 2	
GROUND	1	1	1	2	2	-DB(0)
GROUND	2	3	3	4	4	-DB(1)
GROUND	3	5	5	6	6	-DB(2)
GROUND	4	7	7	8	8	-DB(3)
GROUND	5	9	9	10	10	-DB(4)
GROUND	6	11	11	12	12	-DB(5)
GROUND	7	13	13	14	14	-DB(6)
GROUND	8	15	15	16	16	-DB(7)
GROUND	9	17	17	18	18	-DB(P)
GROUND	10	19	19	20	20	GROUND
GROUND	11	21	21	22	22	GROUND
RESERVED	12	23	23	24	24	RESERVED
OPEN	13	25	25	26	26	TERMPWR
RESERVED	14	27	27	28	28	RESERVED
GROUND	15	29	29	30	30	GROUND
GROUND	16	31	31	32	32	-ATN
GROUND	17	33	33	34	34	GROUND
GROUND	18	35	35	36	36	-BSY
GROUND	19	37	37	38	38	-ACK
GROUND	20	39	39	40	40	-RST
GROUND	21	41	41	42	42	-MSG
GROUND	22	43	43	44	44	-SEL
GROUND	23	45	45	46	46	-C/D
GROUND	24	47	47	48	48	-REQ
GROUND	25	49	49	50	50	-I/O

- 8-bit SCSI, entsprechend mehr Datenleitungen für Wide-SCSI

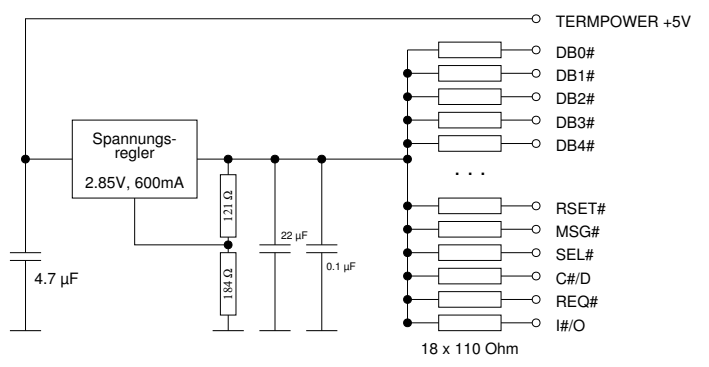
PC-Technologie | SS 2001 | 18.214

SCSI: Signale und Terminierung



- 8-bit SCSI hat 18 Signale auf 50-poligem Kabel
- Signale active-low mit open-Collector Schaltung: kurzschlußfest (!)
- ausgeschaltete Geräte stören den Bus nicht (!)
- Terminator zieht die Leitung auf "inaktiven" high-Pegel
- Terminierung nur an den beiden Endes des Busses

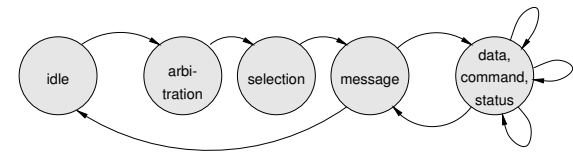
SCSI: aktive Terminierung



- höhere Übertragungsrate erfordert Unterdrückung von Reflexionen
- geforderte Leitungsimpedanz 100..132 Ohm
- mit Spannungsregler / Konstantstromquelle

SCSI: Protokoll

- kompliziertes Mehrphasen-Busprotokoll:



- jeder Datentransfer erfordert die Arbitration-Phasen
- Datenphase erlaubt effiziente Burst-Transfers
- trotzdem beträchtlicher Overhead (H&P: 1 ms pro Transfer)
- mit gleicher Platte langsamer als ATAPI (aber flexibler)
- Arbitrierung/Befehlsübertragung immer 8-bit, asynchron
- Details siehe SCSI Spezifikation

SCSI: SCSI-Befehlssatz

Table N.3 - Commands Common to all SCSI Devices

Command Name	Operation Code	SCSI-3	
		Type	Ref Sid
CHANGE DEFINITION	40h	O	
COMPARE	39h	O	
COPY	18h	O	
COPY AND VERIFY	3Ah	O	
INQUIRY	12h	M	
LOCK/UNLOCK CACHE	36h	O	
LOG SELECT	4Ch	O	
LOG SENSE	4Dh	O	
MODE SELECT (10)	55h	O	
MODE SELECT (6)	15h	M	
MODE SENSE (10)	5Ah	M	
MODE SENSE (6)	1Ah	M	
PREFETCH	3Ah	O	
PREVENT/ALLOW MEDIUM REMOVAL	1Eh	M	
READ (10)	28h	M	
READ (12)	A8h	O	
READ (6)	08h	O	
READ BUFFER	3Ch	O	
READ LONG	3Bh	O	
RECEIVE DIAGNOSTIC RESULTS	1Ch	O	
RELEASE (10)	57h	M	
RELEASE(6)	17h	O	
REQUEST SENSE	03h	M	
RESERVE(10)	56h	M	
RESERVE(6)	16h	O	
SEEK (10)	2Bh	M	
SEEK (6)	0Bh	M	
SEND DIAGNOSTIC	1Dh	M	
SET LIMITS (10)	33h	O	
SET LIMITS (12)	B3h	O	
START/STOP UNIT	1Bh	M	

Key: M = command implementation is mandatory
O = command implementation is optional

- für alle SCSI-Geräte
- zusätzliche Erweiterungen für Platten, Scanner, ...
- Standard: SCSI-3 MMC "multi media commands"

[SCSI-3 MMC spec]

SCSI: SCSI-3 MMC

"MultiMedia Command Set"

- standardisierte Befehlssatzerweiterung für SCSI
- insbesondere für CD/CDR/DVD/DVDR-Geräte:
 - digitales Auslesen von Audio-Tracks ("grabbing")
 - Ansteuerung von digitalen Audio-Ausgängen
 - Ansteuerung / Kalibrierung von CDR/DVD-Brennern
 - Unterstützung für das CSS-Kryptverfahren auf DVDs
- MMC-Befehle auch für ATAPI-Geräte definiert
- erlaubt gemeinsame Treiber für SCSI- und ATAPI-Geräte
- in aktuellen Geräten (etwa CD-Brenner) implementiert

PC-Technologie | SS 2001 | 18.214

SCSI: MMC-Befehlssatz

Command Name	Operation Code	MMC Type	Sub-clause
BLANK Command	A1h	O	6.1.1.
CLOSE TRACK/SESSION	5Bh	M	6.1.2.
FORMAT UNIT	04h	O	6.1.3.
LOAD/UNLOAD CD	A6h	O	6.1.5.
MECHANISM STATUS	BDh	M	6.1.6.
PAUSE/RESUME	4Bh	O	6.1.7.
PLAY AUDIO (10)	45h	A	6.1.8.
PLAY AUDIO (12)	A5h	A	6.1.9.
PLAY AUDIO MSF	47h	A	6.1.10.
READ BUFFER CAPACITY	5Ch	O	6.1.12.
READ CD	BEh	O	6.1.13.
READ CD MSF	B9h	M	6.1.14.
READ CD RECORDED CAPACITY	25h	M	6.1.15.
READ DISC INFORMATION	51h	M	6.1.16.
READ HEADER	44h	M	6.1.18.
READ MASTER CUE	59h	O	6.1.19.
READ SUB-CHANNEL	42h	M	6.1.21.
READ TOC/PA/ATIP	43h	M	6.1.22.
READ TRACK INFORMATION	52h	O	6.1.23.
REPAIR TRACK	58h	O	
RESERVE TRACK	53h	M	6.1.28.
SCAN	BAh	O	6.1.30.
SEEK	2Bh	M	
SEND CUE SHEET	5Dh	O	6.1.31.
SEND OPC INFORMATION	54h	O	6.1.33.
SET CD SPEED	BBh	M	6.1.34.
STOP PLAY/SCAN	4Eh	O	
SYNCHRONIZE CACHE	35h	M	
WRITE (10)	2Ah	O	6.1.38.

- CD-Befehle:
Load/Unload CD
Play Audio (analog/dig.)
Read CD (grabbing)
Read Sub-Channel
Read TOC / ...

PC-Technologie | SS 2001 | 18.214

SCSI: MMC vs. ATAPI

Annex B ATAPI Compliance (normative)

B.1. Introduction

This section describes the implementation of the MultiMedia Commands in ATAPI devices. The intent is to make the command sets highly compatible. It is desired that a common driver may be written to control both SCSI and ATAPI devices.

B.2. General

ATAPI devices implement a subset of SCSI behavior. Certain errors and conditions that exist in SCSI don't exist in ATAPI. In addition, certain terms are used in ATAPI instead of related SCSI terms. The mechanisms for transporting the commands, data, and status are unique to each transport. Addressing of units is also unique to each transport. MMC does not directly specify any of these mechanisms; the command and data layer definition may be layered on either transport.

B.2.1. Terms

B.2.1.1. Host - the ATAPI equivalent for the SCSI term "Initiator."

B.2.1.2. Device - the ATAPI equivalent for the SCSI term "Target" or "Logical Unit."

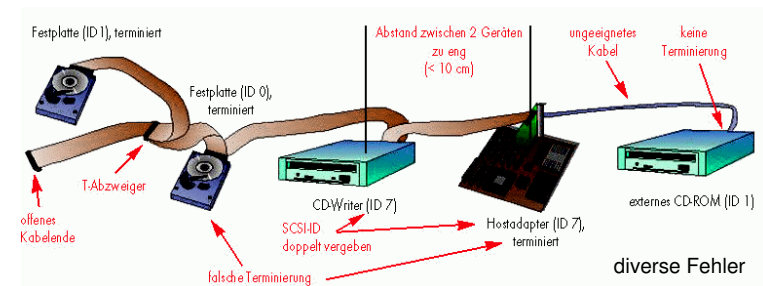
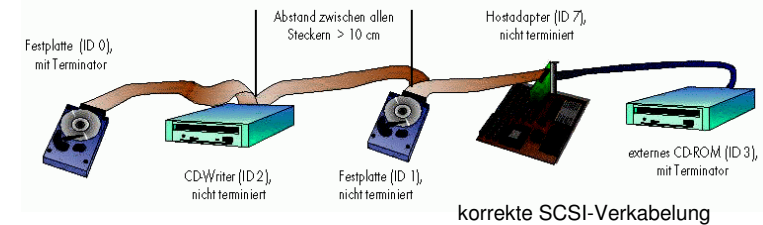
B.2.1.3. Command Packet - the ATAPI equivalent for the SCSI term "Command Descriptor Block."

B.2.2. Supported Block Sizes

ATAPI does not use the block size in the mode select block descriptor. Instead, the block size shall be determined by the command. The READ command shall return 2048 bytes per block. The WRITE command shall send the number of bytes per block as determined by the WRITE PARAMETERS mode page. The READ CD and READ CD MSF commands shall return the number of bytes per block as specified by the command.

PC-Technologie | SS 2001 | 18.214

SCSI: Verkabelung



PC-Technologie | SS 2001 | 18.214

Server: dimensionieren . . .

Ausgangslage und Aufgabe:

[H&P, 530ff]

- Prozessor mit 500 MIPS, kostet \$30.000
- Speicher, Busbreite 16 Byte, 100 ns Zykluszeit
- I/O-Bus mit 200 MB/s Bandbreite, Platz für 20 SCSI-2 Controller
- Betriebssystem benötigt 10.000 CPU-Befehle pro Platten-I/O
- SCSI-2 Busse, jeweils bis 20 MB/s, bis 15 Geräte (= "SCSI String")
- SCSI-2 Controller a \$1.500, mit 1 ms Latenzzeit pro I/O-Transfer
- Platten mit 2 GB oder 8 GB, Preis jeweils \$0.25 pro MB
- beide Platten jeweils 7.200 rpm, 8 ms access time, 6 MB/s Transfer
- geforderte Speicherkapazität 200 GB
- mittlere Blockgröße für I/O-Transfers ist 16 KB

=> Performance mit kleinen/großen Platten? Kosten pro I/O-Transfer? günstigste Konfiguration? wie viele Controller, welche Platten, usw.

PC-Technologie | SS 2001 | 18.214

Server: Grenzen durch CPU, Speicher, Bus

- IOPS = Anzahl I/O-Transfers pro Sekunde

$$\text{IOPS}_{\text{CPU}} = \frac{500 \text{ MIPS}}{10.000 \text{ Befehle pro I/O}} = 50.000$$

$$\text{IOPS}_{\text{Speicher}} = \frac{(1/100 \text{ ns}) \times 16 \text{ Byte}}{16 \text{ KB pro I/O-Transfer}} \sim 10.000$$

$$\text{IOPS}_{\text{Bus}} = \frac{200 \text{ MB / s}}{16 \text{ KB pro I/O-Transfer}} \sim 12.500$$

=> Speicher limitiert auf maximal 10.000 IOPS

PC-Technologie | SS 2001 | 18.214

Server: Grenzen durch Controller und Platten

- Dauer eines SCSI-2 Transfers für 16 KB Daten:
- aber Controller benötigt 1 ms Overhead für den Transfer, also

$$t_{16\text{KB}} = \frac{16 \text{ KB}}{20 \text{ MB / s}} = 0.8 \text{ ms}$$

$$\text{IOPS}_{\text{controller}} = \frac{1}{(0.8 \text{ ms} + 1.0 \text{ ms})} \sim 556 \text{ IOPS}$$

- mittlere Dauer für Platten-I/O mit 16 KB Daten (zufällige Zugriffe):

$$t_{\text{disk}} = 8 \text{ ms} + \frac{0.5}{7200 \text{ rpm}} + \frac{16 \text{ KB}}{6 \text{ MB / s}} = 8 + 4.2 + 2.7 = 14.9 \text{ ms}$$

$$\text{IOPS}_{\text{disk}} = \frac{1}{14.9 \text{ ms}} \sim 67 \text{ IOPS}$$

PC-Technologie | SS 2001 | 18.214

Server: kleine oder große Platten

- 200 GB Kapazität: 25 8-GB Platten oder 100 2-GB Platten
- entsprechende Anzahl der IOPS:

$$\text{IOPS}_{2\text{GB}} = 100 \times 67 = 6700$$

$$\text{IOPS}_{8\text{GB}} = 25 \times 67 = 1675$$

- Mindestanzahl der Controller bei 15 Platten pro String

$$\text{Strings}_{2\text{GB}} = (100 / 15) = 7$$

$$\text{Strings}_{8\text{GB}} = (25 / 15) = 2$$

- Mindestanzahl der Controller, damit diese nicht der Flaschenhals?

$$\text{Disks/String} < 557 / 67 < 8$$

$$\text{Strings}_{2\text{GB}} = (100 / 8) = 12.5 = 13 \quad (\text{aufrunden})$$

$$\text{Strings}_{8\text{GB}} = (25 / 8) = 3.1 = 4 \quad (\text{aufrunden})$$

PC-Technologie | SS 2001 | 18.214

Server: Performance

Architekturen:

Typ	#Platten	#Controller
2 GB	100	7 (min) 13 (opt)
8 GB	25	2 (min) 4 (opt)

Performance:

Platte	#SCSI	CPU	Speicher	Bus	Disks	Strings	IOPS	Kosten
8 GB	2	50.000	10.000	12.500	1675	1112	1112	\$82.200
8 GB	4	50.000	10.000	12.500	1675	2224	1675	\$87.200
2 GB	7	50.000	10.000	12.500	6700	3892	3892	\$91.700
2 GB	13	50.000	10.000	12.500	6700	7228	6700	\$100.700

- Server-Performance wird durch die Platten bzw. Controller limitiert (!)
- beste Performance mit vielen kleinen Platten und Controllern
- außerdem bestes Preis/IOPS-Verhältnis (\$76, \$52, \$24, \$15 pro IOPS)
- aber geringere Zuverlässigkeit (siehe RAID)

PC-Technologie | SS 2001 | 18.214

RAID: Motivation

Amdahl's Gesetz:

langsamste Komponente behindert Leistungssteigerungen

- => ausgewogenes Verhältnis CPU - Speicher - I/O nötig
- => CPU und Speicher skalieren mit der Halbleitertechnologie
- => aber wie kann die I/O-Leistung gesteigert werden?

RAID, "redundant array of inexpensive disks":

- Grundidee: viele kleine PC-Festplatten statt einer großen
- bedingt in damaliger (1985er) Festplattentechnologie: Großrechner-Festplatten vs. PC-Festplatten
- Zuverlässigkeit durch redundante Platten
- Wiederherstellung der Daten nach Plattenausfall
- ursprünglich: "independent disks"

PC-Technologie | SS 2001 | 18.214

Disks: RAID

"redundant array of inexpensive disks"

- bahnbrechende Untersuchung von Festplatten-Performance
- ursprünglich Analyse von Großrechner- und PC-Festplatten
- Ersetzen weniger großer durch viele kleine Festplatten
- Zuverlässigkeit des Gesamtsystems?

- diverse RAID-Varianten (=level)
- unterschiedliche Anzahl von Platten
- Strategien zur Verwendung von Nutz- und Reserveplatten
- Ausfallsicherheit, Hot-Plugging
- Optimierung auf Schreib- und/oder Leseperformance
- vielfache Anwendungen

- möglichst das Original lesen!

[Patterson, Gibson, Katz: UCB report CS-98-391]

PC-Technologie | SS 2001 | 18.214

RAID: Ausgangsbasis (1987)

Characteristics	IBM 3380	Fujitsu M2361A	Conners CP3100	3380 v. CP3100	2361 v. CP3100
					(>1 means 3100 better)
Disk diameter (inches)	14	10.5	3.5	4	3
Formatted Data Capacity (MB)	7500	600	100	.01	.2
Price/MB(controller incl.)	\$18-\$10	\$20-\$17	\$10-\$7	1-2.5	1.7-3
MTTF Rated (hours)	30,000	20,000	30,000	1	1.5
MTTF in practice (hours)	100,000	?	?	?	?
No. Actuators	4	1	1	.2	1
Maximum I/O's/second/Actuator	50	40	30	.6	.8
Typical I/O's/second/Actuator	30	24	20	.7	.8
Maximum I/O's/second/box	200	40	30	.2	.8
Typical I/O's/second/box	120	24	20	.2	.8
Transfer Rate (MB/sec)	3	2.5	1	.3	.4
Power/box (W)	6,600	640	10 [†]	660	64
Volume (cu. ft.)	24	3.4	.03	800	11

Table I. Comparison of IBM 3380 disk model AK4 for mainframe computers, the Fujitsu M2361A "Super Eagle" disk for minicomputers, and the Conners Peripherals CP 3100 disk for personal computers. By "Maximum I/O's/second" we mean the maximum number of average seeks and average rotates for a single sector access. Cost and reliability information on the 3380 comes from widespread experience [IBM 87] [Gawlick87] and the information on the Fujitsu from the manual [Fujitsu 87], while some numbers on the new CP3100 are based on speculation. The price per megabyte is given as a range to allow for different prices for volume discount and different mark-up practices of the vendors.

[†]The 8 watt maximum power of the CP3100 was increased to 10 watts to allow for the inefficiency of an external power supply (since the other drives contain their own power supplies).

PC-Technologie | SS 2001 | 18.214

RAID: Kriterien

- Gesamtkapazität der Festplatte(n) MByte
- maximale und typische Bandbreite MByte/s
- maximale und typische Latenzzeiten s
- Kosten, Volumen, Energieverbrauch \$, m³, W
- Zuverlässigkeit
 - MTTF, "mean time to failure" s
 - MTTR, "mean time to repair" s
 - $MTTF_{total} = (MTTF_{single} / \text{number_of_disks})$

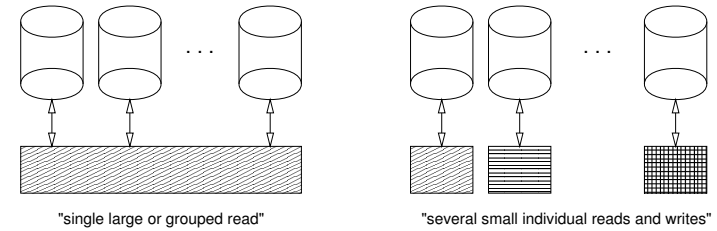
RAID-Konzept: viele parallele Platten

- höhere Gesamtkapazität, höhere Bandbreite
- Redundanz erhöht (!) die Zuverlässigkeit
- damalige Annahme: ca. 100 Platten, heute: typ. 5-10

RAID: Glossar

D	Gesamtanzahl der Platten
G	Anzahl der Daten- (=nutz) Platten pro Gruppe
NG	Anzahl der Gruppen
C	Anzahl der redundaten Check-Platten
rc	Verhältnis C/G
s	slowdown, typ. $1 < s < 2$

RAID: Szenarien



welche Anwendungen benötigen hohe I/O-Leistung?

"scientific":	wenige, aber große Transfers
"database"	sehr viele kleine Transfers

RAID: Statistik

Annahmen zur Zuverlässigkeit der Platten:

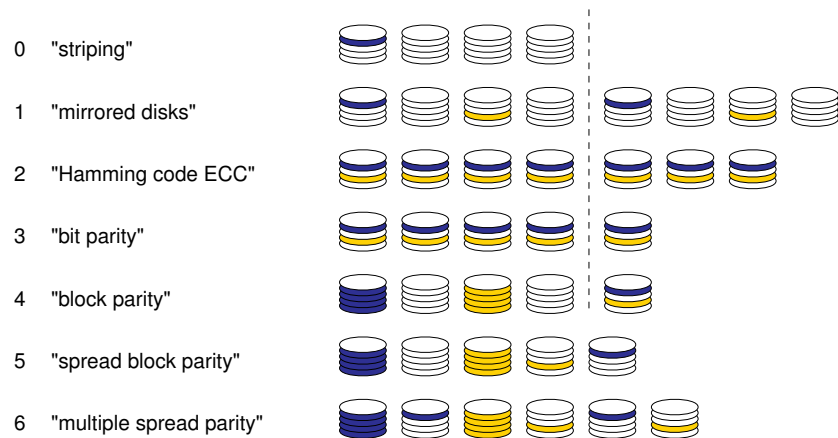
- Ausfälle sind zufällig, unabhängig, exponentialverteilt
- äußere Einflüsse (Sabotage, Stromausfall, ...) nicht berücksichtigt
- Controller ist robust

$$MTTF_{group} = \frac{MTTF_{disk}}{G + C} = \frac{1}{\text{probability of failure before repairing the dead disk}}$$

$$P_{second_failure} = \frac{MTTR}{MTTF_{disk} / (G+C-1)}$$

$$MTTF_{raid} = \frac{(MTTF_{disk})^2}{(D+C*NG) * (G+C-1)*MTTR}$$

RAID: Level-Übersicht



PC-Technologie | SS 2001 | 18.214

RAID-0: Striping

- Aufteilen jedes (großen) Zugriffs in "Streifen"
 $D = G, C = 0$
- jede Platte verarbeitet Anteil $1/D$
- jeder Zugriff benutzt alle Platten
- theoretisch D -fache Bandbreite für Lesen und Schreiben
- nur für genügend große Zugriffe
- aber keine Fehlertoleranz
- Zuverlässigkeit sinkt auf $1/D$
- Einsatz nur für geringe Anzahl von Platten
- nicht im originalen "RAID paper" enthalten
- marktübliche "RAID-0" Controller verwalten zwei Platten

PC-Technologie | SS 2001 | 18.214

RAID-1: Mirroring

- Daten werden auf je zwei Platten "gespiegelt"
 $G = 1, C = 1$
- nutzt nur 50% der Gesamtkapazität der Platten
- jeder Schreibzugriff geht auf zwei Platten
- Schreibzugriff muß auf die jeweils langsamere Platten warten
- optimierte Version benutzt doppelten Controller
- erlaubt doppelte Bandbreite beim Lesen
- kein komplexer Controller notwendig
- ineffizient, aber sehr zuverlässig
z.B. 500 Jahre MTTF
- keine besondere Marktbedeutung

PC-Technologie | SS 2001 | 18.214

RAID-2: Hamming Code ECC

- Hamming-Code zur Fehlerkorrektur jeder Gruppe von Platten
z.B. ($G=10, C=4$) oder ($G=25, C=5$) usw.
- analog zur ECC-Fehlerkorrektur bei DRAMs
- Controller muß ECC berechnen und auswerten
- Aufteilung in Daten- und Check-Platten
- "große" Zugriffe laufen auf alle Platten einer Gruppe
- dabei volle Performance beim Lesen und Schreiben
- "kleine" Zugriffe kompliziert: gesamten Block lesen, ECC mit neuen Daten berechnen, gesamten Block schreiben
- daher sehr schlechte Gesamtperformance
- CRC-Code der einzelnen Platten unnötig
- sehr hohe Zuverlässigkeit, z.B. 50 Jahre MTTF mit $G=10$

PC-Technologie | SS 2001 | 18.214

RAID-3: Bit-Parität

- eine Platte mit Paritätscode pro Gruppe
C=1
- Hamming-Code ermittelt, welche Platte Fehler aufweist
- dies liefert aber bereits der CRC jeder einzelnen Platte
- Paritätscode reicht aus, um den Fehler zu korrigieren
- weniger Checkdisks als RAID-2
- aber gleiches Performanceproblem für "kleine" Zugriffe
- jeder Schreibzugriff benutzt die Paritätsplatte
- weniger Platten als RAID-2, daher Preis/Leistung besser
- sehr hohe Zuverlässigkeit, z.B. 50 Jahre MTTF mit G=10

RAID-3: Vergleich Level 2 / 3

MTTF	Exceeds Useful Lifetime		
	G=10 (820,000 hrs or >90 years)	G=25 (346,000 hrs or 40 years)	
Total Number of Disks	1.10D	1.04D	
Overhead Cost	10%	4%	
Useable Storage Capacity	91%	96%	
I/Os/Sec (vs. Single Disk)	Full RAID	Per Disk L3 L3/L2	Per Disk L3 L3/L2
Large Reads/sec	D/S	.91/S 127%	.96/S 112%
Large Writes/sec	D/S	.91/S 127%	.96/S 112%
Large R-M-W/sec	D/2S	.45/S 127%	.48/S 112%
Small Reads/sec	D/SG	.09/S 127%	.04/S 112%
Small Writes/sec	D/2SG	.05/S 127%	.02/S 112%
Small R-M-W/sec	D/2SG	.05/S 127%	.02/S 112%

Table IV. Characteristics of a Level 3 RAID. The L3/L2 column gives the % performance of L3 in terms of L2 (>100% means L3 is faster). The performance for the full systems is the same in RAID levels 2 and 3, but since there are fewer check disks the performance per disk improves. Once again if the disks in a group are synchronized, then S = 1.

RAID-4: Block-Parität

- eine Platte mit Paritätscode pro Gruppe
C=1
- einzelner Datenblock wird auf eine einzelne Platte geschrieben
- Parität des Blocks auf die Paritätsplatte
- Paritätscode reicht aus, um den Fehler zu korrigieren
- gleiche Anzahl Platten wie RAID-3
- aber andere Organisation
- Lesezugriffe parallel ausführbar
- Schreibzugriffe parallel auf Datenplatten ausführbar
- aber Flaschenhals Paritätsplatte
- sehr hohe Zuverlässigkeit, z.B. 50 Jahre MTTF mit G=10

RAID-5: verteilte Parität

- Paritätscode auf alle Platten einer Gruppe verteilt
C=1
- einzelner Datenblock wird auf eine einzelne Platte geschrieben
- Parität des Blocks auf die zugehörige Paritätsplatte
- Paritätscode reicht aus, um den Fehler zu korrigieren
- gleiche Anzahl Platten wie RAID-3
- aber effizienteste Organisation:
- Lesezugriffe parallel ausführbar
- Schreibzugriffe weitgehend parallel ausführbar
- attraktivste Variante, erfordert aber komplexen Controller
- hohe Zuverlässigkeit, z.B. 50 Jahre MTTF mit G=10

RAID-6: unabhängige, verteilte Parität

- mehrfacher Paritätscode auf alle Platten einer Gruppe verteilt
C=2, 3, ...
- einzelner Datenblock wird auf eine einzelne Platte geschrieben
- Parität des Blocks auf die zugehörigen Paritätsplatten
- diverse Code-Varianten möglich
- ähnlich wie RAID-5
- aber bessere Fehlererkennung/korrektur
- Lesezugriffe parallel ausführbar
- Schreibzugriffe weitgehend parallel ausführbar
- noch komplexerer Controller als RAID-5
- nicht im originalen RAID-Paper erwähnt

RAID: Vergleich (1987)

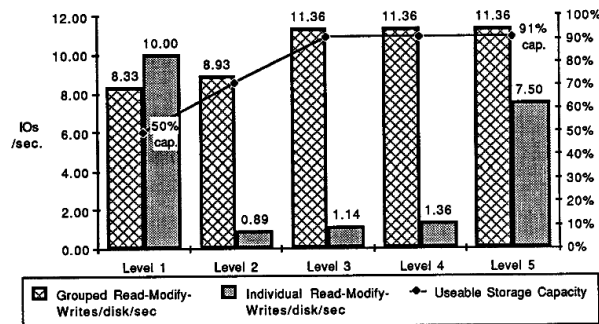


Figure 5. Plot of Large (Grouped) and Small (Individual) Read-Modify-Writes per second per disk and useable storage capacity for all five levels of RAID (D=100, G=10, I/O=30/sec, S=1.2). To scale performance to other speed disks, simply multiply these numbers by the ratio to 30 I/O's/sec.

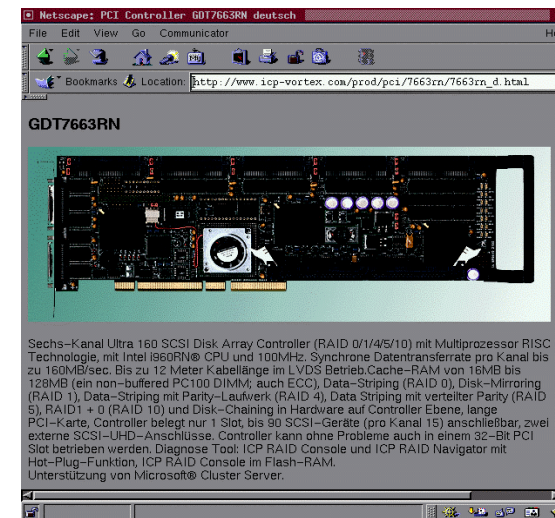
- Level-1 schnell, sicher, teuer
- Level-5 der beste Kompromiss

RAID: vs. single disks (1987)

Characteristics	RAID 5L (100,10) (CP3100)	SLED (IBM 3380)	RAID v. SLED (>1 better for RAID)	RAID 5L (10,10) (CP3100)	SLED (Fujitsu M2361A)	RAID v. SLED (>1 better for RAID)
Formatted Data Capacity (MB)	10,000	7,500	1.33	1,000	600	1.67
Price/MB (controller incl.)	\$11-\$8	\$18-\$10	2.2-.9	\$11-\$8	\$20-\$17	2.5-1.5
Rated MTTF (hours)	820,000	30,000	27.3	8,200,000	20,000	410
MTTF in practice (hours)	?	100,000	?	?	?	?
No. Actuators	110	4	22.5	11	1	11
Max I/O's/Actuator	30	50	.6	30	40	.8
Max Grouped RMW/box	1250	100	12.5	125	20	6.2
Max Individual RMW/box	825	100	8.2	83	20	4.2
Typ I/O's/Actuator	20	30	.7	20	24	.8
Typ Grouped RMW/box	833	60	13.9	83	12	6.9
Typ Individual RMW/box	550	60	9.2	55	12	4.6
Volume/Box (cubic feet)	10	24	2.4	1	3.4	3.4
Power/box (W)	1100	6,600	6.0	110	640	5.8
Minimum Expansion Size (MB)	100-1000	7,500	7.5-75	100-1000	600	0.6-6

Table VII. Comparison of IBM 3380 disk model AK4 to Level 5 RAID using 100 Conners & Associates CP 3100s disks and a group size of 10 and a comparison of the Fujitsu M2361A "Super Eagle" to a level 5 RAID using 10 inexpensive data disks with a group size of 10. Numbers greater than 1 in the comparison columns favor the RAID.

RAID: Beispiel für einen Controller



Disks: Filecache

"Filecache"

- Plattenzugriffe deutlich langsamer als Speicherzugriffe
 - häufig benutzte Daten (Dateien) im Hauptspeicher halten
- => Teil des Hauptspeichers als Filecache reservieren
- aber Filecache reduziert nutzbaren Hauptspeicher
 - wo liegt das Optimum?
- nutzungsabhängig, single/multi user, workstation/server
 - verschiedene Betriebssystemstrategien
 - z.B. Windows 95 vs. Windows NT
- im folgenden einige Beispiele aus H&P

Disks: Filecache

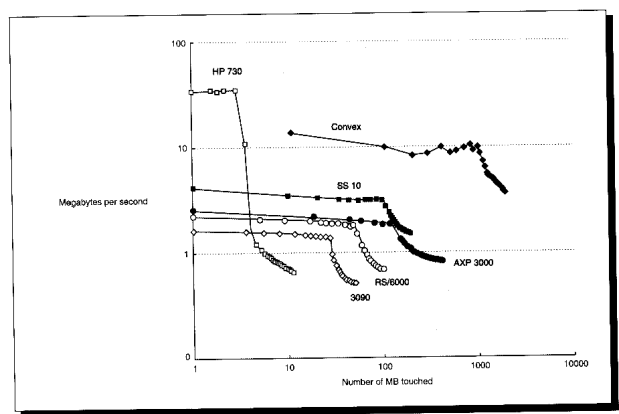


FIGURE 5.26 Performance versus megabytes touched for several workstations and mainframes (see section 5.8). Note the log-log scale. These results use the nominal values selected by the self-scaling benchmark. For example, 50% of accesses are reads and 50% are writes. The primary difference between the systems is the average access size of 120 KB for the Convex; adjusting for a common access size would halve Convex performance but make little change to the other lines in this plot.

[Hennessy & Patterson]

Disks: Filecache: Performance

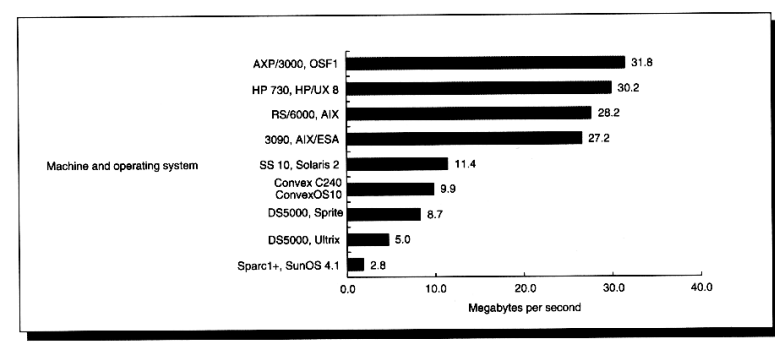


FIGURE 6.38 File cache performance for machines in 6.35. This plot is for 32-KB reads with the number of bytes touched limited to fit within the file cache of each system. Figure 6.36 (page 541) shows the size of the file caches that achieve this performance. (See the caption of Figure 6.36 for details on measurements.)

Disks: Filecache: Size

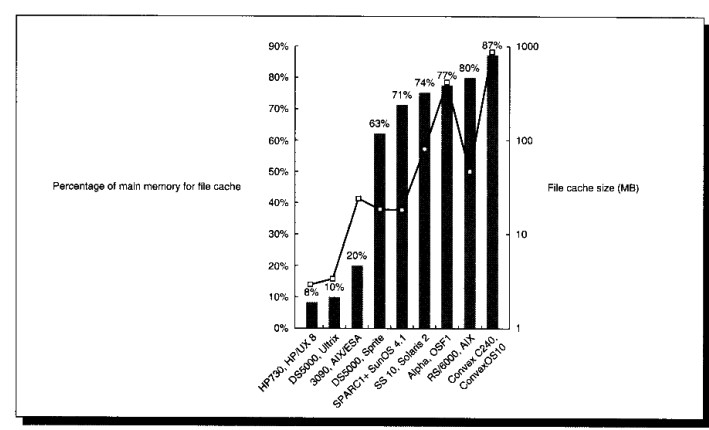


FIGURE 6.39 File cache size. The bar graph shows the maximum percentage of main memory for the file cache, while the line graph shows the maximum size in megabytes, using the log scale on the right. Thus the HP 730 HP/UX version 8 uses only 8% of its 32-MB main memory for its file cache, or just 2.7 MB, and the Convex C240 uses 87% of its 1024-MB main memory, or 890 MB, for its file cache.

Disks: Filecache: Read vs. Write

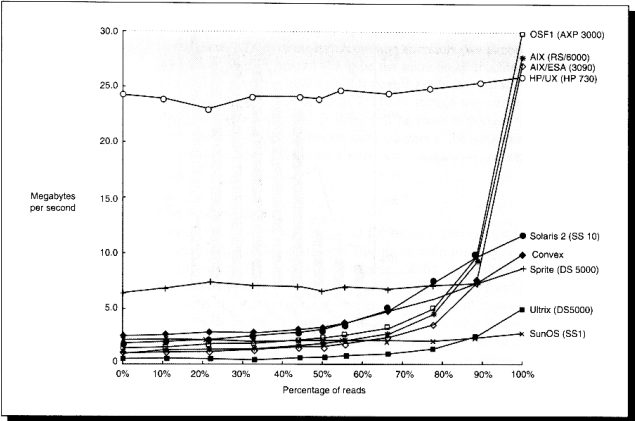


FIGURE 6.40 File cache performance versus read percentage. 0% reads means 100% writes. These accesses all fit within the file caches of the respective machines. Note that the high performance of the file caches of the AXP/3000, RS/6000, and 3090 are only evident for workloads with ≥ 90% reads. Access sizes are 32 KB. (See the caption of Figure 6.36 for details on measurements.)

CD, DVD: Agenda

- Grundlagen der CD-Technik
- CD-ROM
- CD-R, CD-RW
- DVD und Video
- DVD-R, DVD-RAM
- ISO-9660 Dateisystem
- UDF / Packet-Writing



PC-Technologie | SS 2001 | 18.214

CD/DVD: Literatur

- | | |
|--|---|
| www.disctrionics.co.uk/ | (übersichtliche Kurzbeschreibungen zu CD/DVD) |
| www.fadden.com/cdrfaq | (alles rund ums Thema CD-R und CD-RW) |
| www.dvddemystified.com/dvdfaq.html | (DVD-FAQ, viele mirrors weltweit) |
| www.ping.be/~pin11466/formtxt.html | (schöne Übersicht) |
| www.unik.no/~robert/hifi/dvd/ | (umfangreiche Link-Sammlung zu DVD) |
| www.phoenix.com/techs/specs.html | (El Torito Format für bootfähige CDs) |

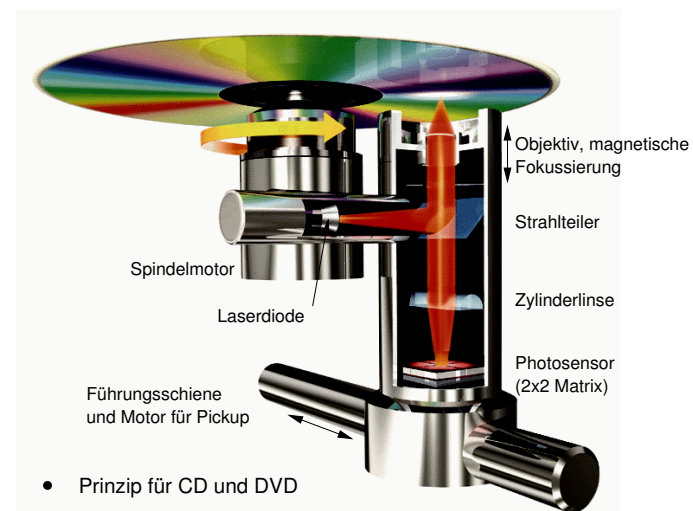
diverse c't Artikel:
 02/93 178f CD "color books" und Dateiformate
 DVD: 23/99 S.100f

diverse Standards, insbesondere ECMA-xxx (frei verfügbar), www.ecma.ch

- | | |
|------------|-----------------------------|
| ECMA-119 | ISO-9660 Dateisystem |
| ECMA-267 | DVD-ROM Spezifikation |
| SCSI-3 MMC | SCSI MultiMedia Command Set |

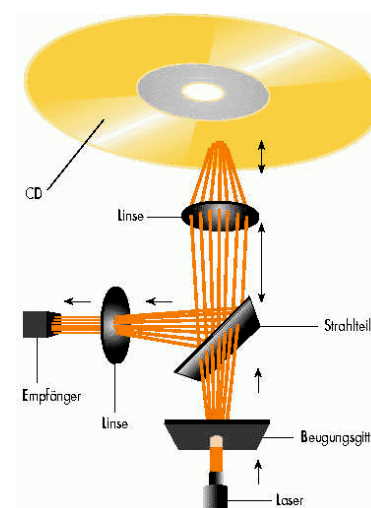
PC-Technologie | SS 2001 | 18.214

CD: Aufbau eines Players



PC-Technologie | SS 2001 | 18.214

CD: Multibeam-Technik

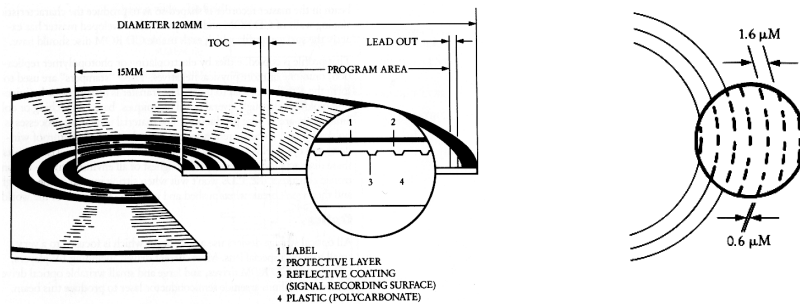


- Strahlteiler
- komplexer Empfänger mit mehreren Photodioden
- liest mehrere Spuren gleichzeitig
- statt höherer Drehzahl

Testbericht in [ct 08/99 74]

PC-Technologie | SS 2001 | 18.214

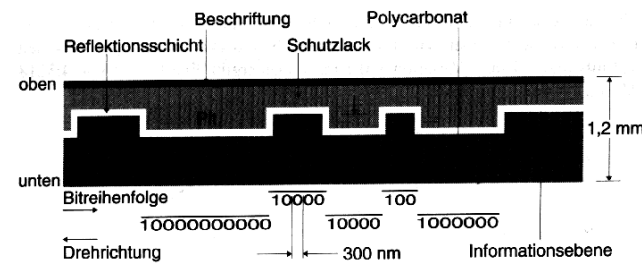
CD: Prinzip



- Polycarbonaträger, 12cm Durchmesser
- eingeprägte Vertiefungen ("pits") bilden die Daten
- spiralförmige Datenspur, 1.6µm Abstand, ca. 16000 Windungen
- Fertigungsmängel fest eingeplant => leistungsfähige Fehlerkorrektur

[CD-ROM - The new Papyrus]

CD: Schichtaufbau



- Polycarbonatschicht ~ 1.2 mm
- Größe der Pits / Lands ~ 1.0 µm
- Interpretation: Land = 0, Pit = 1, Wechsel Land/Pit = 1
- Achtung: Kratzer oben zerstören die Daten

CD: Reflexion

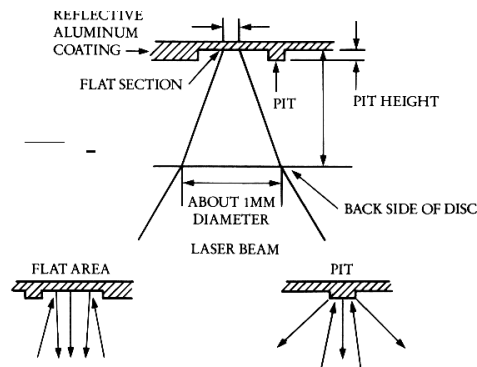
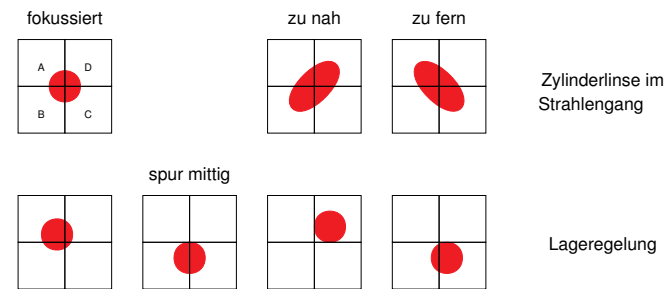


Figure 4. Relation between pits and photodetector output.

- Lands reflektieren das Laserlicht
- Pits streuen das Laserlicht

[CD-ROM - The new Papyrus]

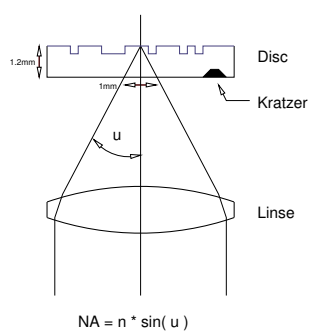
CD: Fokussierung, Spurregelung, ...



Sensorfeld mit 2x2 Photodioden zur Regelung:

- Fokussierung: aus Differenzsignal $(A+C) - (B+D)$
- Spurregelung: aus Differenzsignal $(A+B) - (C+D)$
- Nutzsignal: Land/Pit-Übergänge aus Summensignal
- Drehzahl: aus Taktfrequenz des Nutzsignals

CD: numerische Apertur



Brechungsindex 'n' eines Materials:

- Vakuum = 1
- Luft ~ 1
- Diamant = 2.4

NA := Maß für Auflösungsvermögen des Objektivs

	NA	Öffnungswinkel
CD	0.45	24 .. Grad
DVD	0.5 .. 0.6	.. 37 Grad

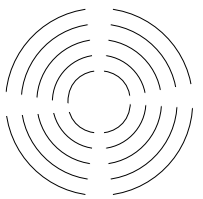
$$NA = n \cdot \sin(u)$$

- Auflösungsvermögen wie bei Mikroskopen (!)
- asphärische Linsen notwendig
- Kratzer/Staub auf der Oberfläche stören kaum

CD: CAV vs. CLV

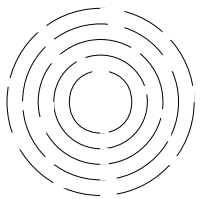
Constant Angular Velocity

(Floppy, aktuelle CD-ROM Laufwerke)



Constant Linear Velocity

(Audio/Video CD, DVD)



- Audiodaten: konstante Datenrate sinnvoll: CLV
- Drehzahl: innen hoch, außen langsam
- Angabe "48x"-Laufwerk: 48x Datenrate (CLV) der Audio-CD
- CAV erlaubt Spurwechsel ohne Drehzahländerung
- aktuelle CD-ROMs: CAV soweit per Daten/Fehlerrate möglich

Programme "CD-Bremse", "CD-Bänschmaak": home.t-online.de/home/Joern.Fiebelkorn/

CD: EFM

Eight-to-Fourteen Modulation:

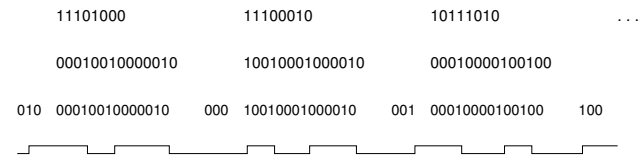
- selbsttaktende Aufzeichnung, NRZI
- minimal 2 Nullen, maximal 11 Nullen zwischen Einsen

data bits	channel bits
0000 0000	01001000100000
0000 0001	10000100000000
0000 0010	10010000100000
0000 0011	10001000100000
0000 0100	01000100000000
0000 0101	00000100010000
...	... via lookup table

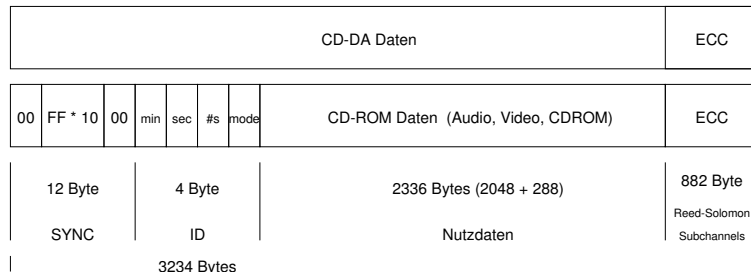
- zusätzlich 3 "Merge-Bits" zwischen zwei Codeworten einfügen
- eigentlich 8-17 Modulation
- DVD verwendet verbessertes 8-16 Verfahren

CD: Kodierung der Daten:

- Nutzdaten
- Nutzdaten in Frames einteilen
- Reed-Solomon Checksumme an Frames anfügen
- 14-bit EFM-Daten aus 8-Bit Nutzdaten
- 17-bit EFM mit Merge-Bits
- 17-bit EFM, Sync-Pattern anfügen
- Pits and Lands

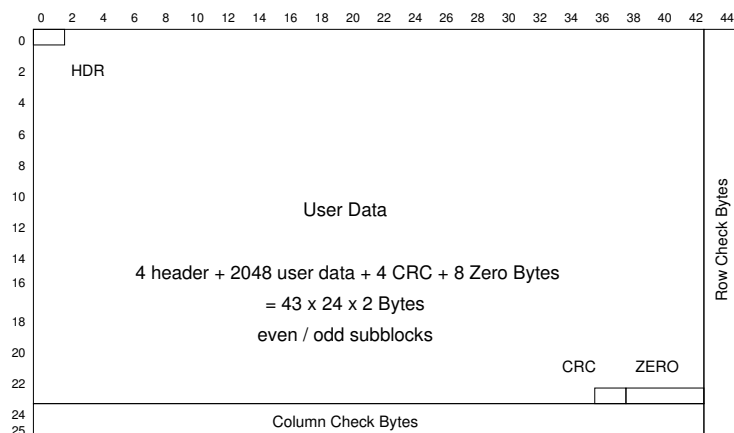


CD: Sektoren (Frames)



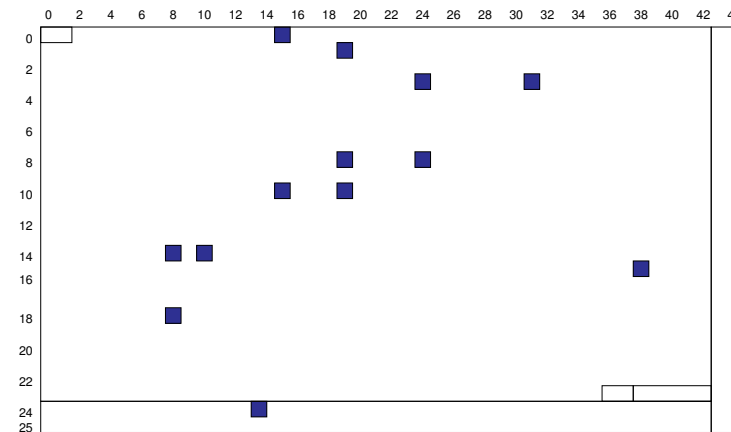
- 75 Sektoren pro Sekunde, 0 .. 74
- Numerierung per (minute, second, sector)
- 60 Minuten: 270.000 Frames (553 MB)
- 74 Minuten: 333.000 Frames (682 MB)

CD-ROM LEC Reed-Solomon Code



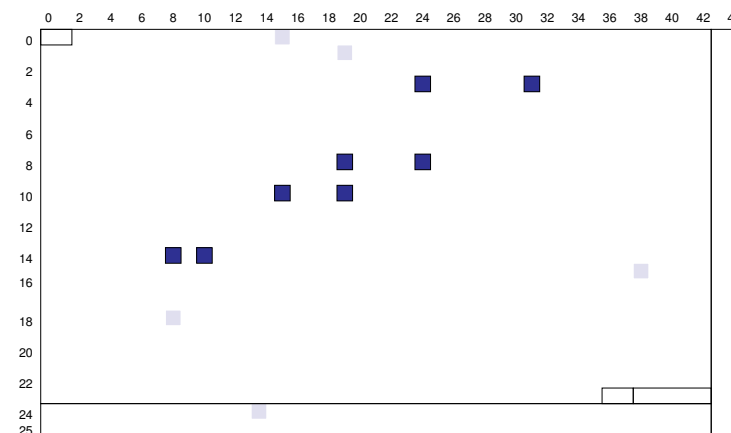
- Layered Error Correction (CD-ROM Mode 1)

CD: Reed-Solomon Code



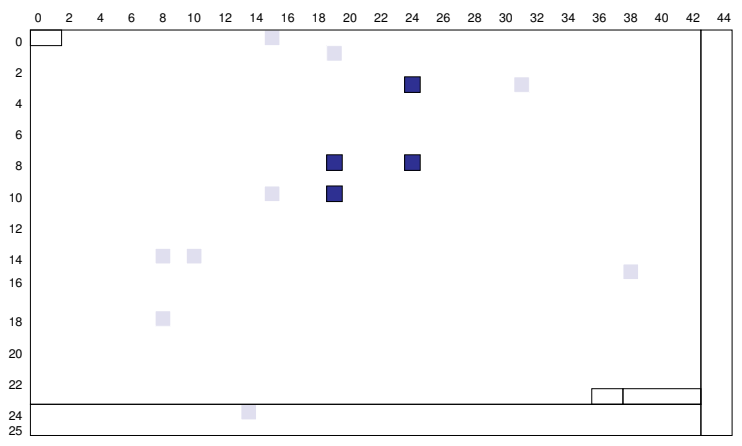
- Beispiel: Fehler vor der Korrektur

CD: Reed-Solomon Code



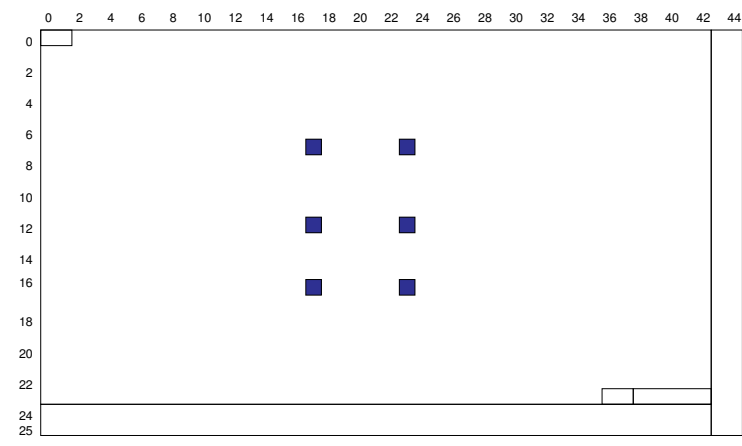
- erster Schritt: alle Einzelfehler in Zeilen korrigiert

CD: Reed-Solomon Code



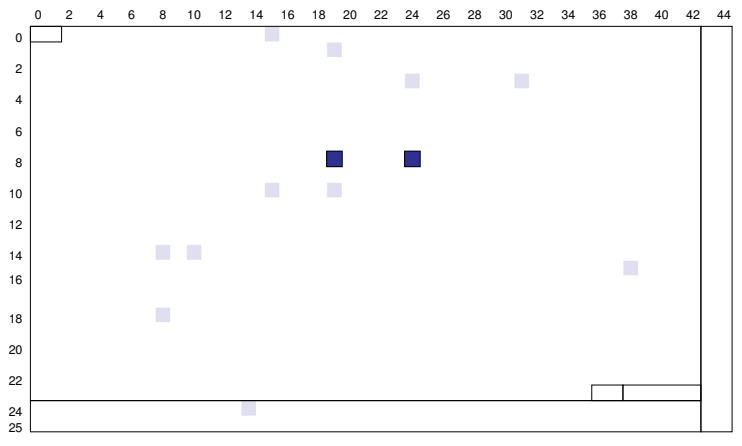
- zweiter Schritt: alle Einzelfehler in Spalten korrigiert

CD: Reed-Solomon Code



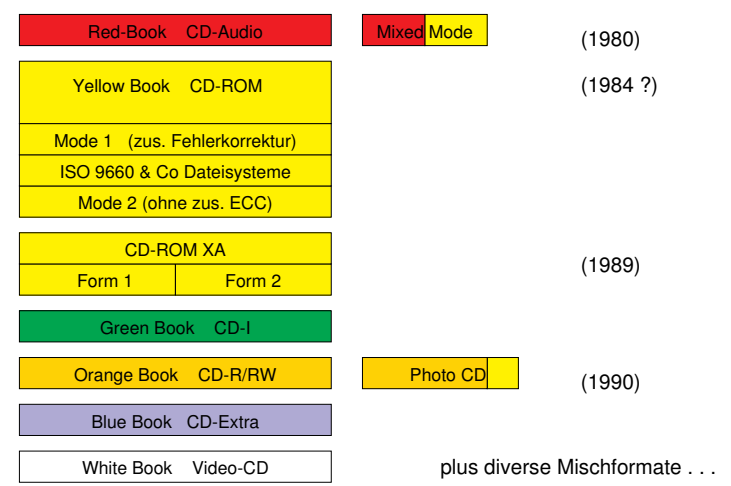
- keine Einzelfehler in Zeilen/Spalten, trotzdem korrigierbar
- Bitfehlerrate CD-ROM besser als 1E-13

CD: Reed-Solomon Code

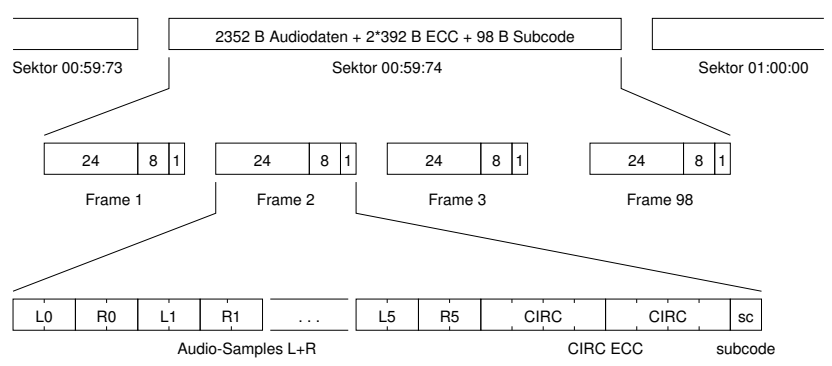


- dritter Schritt: wieder die Zeilen korrigiert, dann wieder die Spalten

CD: "colors"

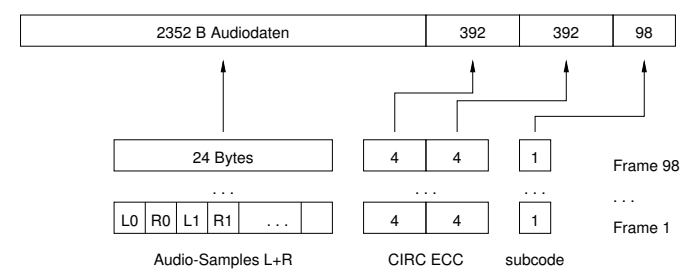


CD: Audioformat, Sektoren, Frames



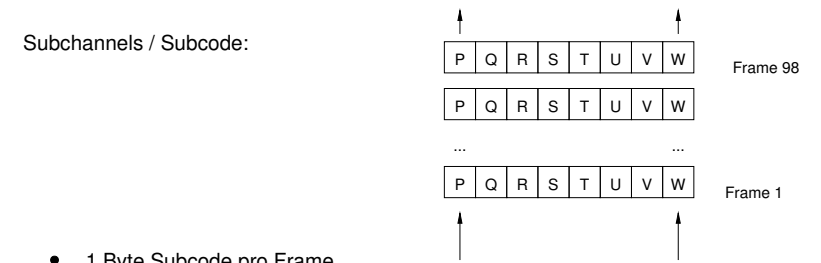
- 75 Sektoren pro Sekunde
- 98 Frames a 24 Bytes (+ECC) pro Sektor
- je 1 Byte Subcode pro Sektor

CD: Audioformat, konzeptionell



- 98 Frames a 24 Bytes pro Sektor
- 24 Bytes: je 6 Samples linker/rechter Kanal
- ein Byte Subcode pro Frame
- $75/s * 2352B = 44100/s * 16b * 2 / 8 = 176 \text{ KB/s}$
- ECC korrigiert Bursts bis zu 7000 fehlenden Bits

CD: Subchannels

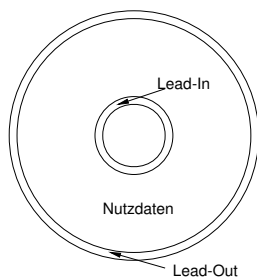


- 1 Byte Subcode pro Frame
- P markiert Start eines Tracks
- Q für Inhaltsverzeichnis der CD (TOC)
- R..W abhängig vom Format
z.B. konstant Null bei CD-ROM
Nutzung für CD-Text
Datenrate: $75 * 98 \text{ bit/s} = 918 \text{ B/s}$

CD: Lead In / Lead Out

Lead-In:

- spezieller Bereich am Anfang (innen) der CD
- Nutzdaten konstant Null
- Q-Subcode enthält das Inhaltsverzeichnis
- bis zu 99 Tracks erlaubt: ca. 9 MByte
- Lead-Out kennzeichnet Ende der CD
- Nutzdaten und Subcodes konstant Null
- Multisession-CDs:
je 1 Lead-In/Out Bereich pro Session
plus Master-Lead In / Lead-Out



PC-Technologie | SS 2001 | 18.214

CD: Photo-CD

- Kodak / Philips 1993
- basiert auf CD-ROM/XA
- Filme (Kleinbild) werden mit 2000 dpi gescannt
- Auflösung 3072x2048 Pixel (optional 6144x4096)
- bis ca. 100 Photos
- multisession-Format (erlaubt mehrere Filme)
- proprietäres Datenformat
- mehrere Auflösungen: 192x128 bis 3072x2048 Pixel
- vergleichsweise hohe Kosten
- Markterfolg nur im Profi-Bereich
- neuer Versuch als "Picture-CD" (mit Intel/Adobe Software)
1024x1536 Pixel, JPEG-Format



[www.kodak.com]

PC-Technologie | SS 2001 | 18.214

CD: Datenformate Daten / Audio

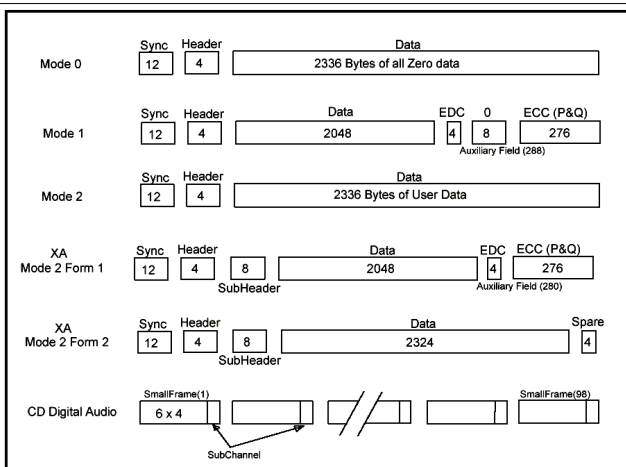


Figure 2 - CD-ROM Sector Formats

- Kapazität vs. Fehlerkorrektor

PC-Technologie | SS 2001 | 18.214

Leerseite

PC-Technologie

CD: Mixed Mode CD

- kombiniert Audio-Tracks mit Daten/Video-Tracks
- Audio CD-Player erwarten nur Audio-Tracks:
 - ältere Player lesen Daten als Audio
 - Knacksen (Vorsicht: extreme Lautstärke)
 - neuere Player schalten den Track stumm
 - alternative Anordnung: Datentrack(s) ganz hinten
- "kranke" Block- bzw. Min/Sek/Frame-Adressierung
- (siehe Beispiel auf nächster Folie)
- wird von allen aktuellen PC-Laufwerken unterstützt

CD: Mixed Mode CD (Beispiel)

Block Description	Logical Address (Decimal)	Track Relative logical address	Absolute M/S/F Address ¹	Track / Index	Track Relative M/S/F Address	Sector Contains Info or Pause	Mode Audio or Data	CD Data Mode ²
Lead-in Area	---	---	---	0/-	---	---	Audio	---
Pre-gap	---	---	00/00/00	1/0	00/02/00 ²	Pause	Data	Null
1st Track data	0000 ⁴	0	00/02/00 ⁵	1/1	00/00/00	Info	Data	L-EC
2nd track data	6000 ⁴	0	01/22/00 ⁵	2/1	00/00/00	Info	Data	L-EC
	7500	1500	01/42/00	2/2	00/20/00	Info	Data	L-EC
Post gap	9000	3000	02/02/00	2/3	00/40/00	Pause	Data	Null
Pause-silence	9150	-150 ⁶	02/04/00	3/0	00/02/00 ²	Pause	Audio	---
3rd track audio	9300 ⁸	0	02/04/00 ⁹	3/1	00/00/00	Info	Audio	---
	1400	2250	02/34/00	3/2	00/03/00	Info	Audio	---
4th track audio	21975 ⁸	0	04/53/00 ⁹	4/1	00/00/00	Info	Audio	---
Pre-gap part 1	30000	-225 ⁶	06/40/00	5/0	00/03/00 ²	Pause	Audio	---
Pre-gap part 2	300075	-150	06/41/00	5/0	00/02/00 ²	Pause	Data	Null
5th track data	30225	0	06/43/00	5/1	00/00/00	Info	Data	L-EC
Last Information	263999 ¹⁰	233 774	58/39/74	5/1	51/56/74	Info	Data	L-EC
Post-gap	---	233 775	58/40/00	5/2	51/57/00	Pause	Data	Null
Lead-out area	264000 ¹¹	0	58/42/00	AA/ ¹³	00/00/00	Pause	Audio	---

CD: LBA/MSF Umrechnung

Table 207- LBA to MSF translation

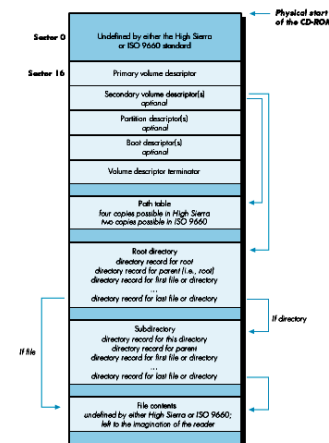
Condition	Formulac
$-150 \leq LBA \leq 404849$	$M = IP\left(\frac{LBA + 150}{60 \cdot 75}\right)$ $S = IP\left(\frac{LBA + 150 - M \cdot 60 \cdot 75}{75}\right)$ $F = IP(LBA + 150 - M \cdot 60 \cdot 75 - S \cdot 75)$
$-45150 \leq LBA \leq -151$	$M = IP\left(\frac{LBA + 450150}{60 \cdot 75}\right)$ $S = IP\left(\frac{LBA + 450150 - M \cdot 60 \cdot 75}{75}\right)$ $F = IP(LBA + 450150 - M \cdot 60 \cdot 75 - S \cdot 75)$
$00/00/00 \leq MSF \leq 89/59/74$	$LBA = (M \cdot 60 + S) \cdot 75 + F - 150$
$90/00/00 \leq MSF \leq 99/59/74$	$LBA = (M \cdot 60 + S) \cdot 75 + F - 450150$

- logische Blockadresse vs. Minute/Sekunde/Frame

CD-ROM: ISO 9660

Standard-Dateiformat für CD-ROMs

- Daten starten in Sektor 16 (00:02:16)
- DOS-kompatibel (FAT)
- Dateinamen mit 8+3 Zeichen
- bis zu 8 Verzeichnisebenen
- Level-2 erlaubt Namen bis 32 Zeichen
- plattformunabhängig
- Dateien müssen linear vorliegen
- keine späteren Änderungen möglich
- Level-3 erlaubt fragmentierte Dateien



CD-ROM: Joliet

- Microsoft-Erweiterung von ISO-9660
- erlaubt Windows95-Dateinamen
- Namen bis 64 Zeichen, inklusive Sonderzeichen
- integriert in Windows 9x/2K
- ebenfalls in neueren Linux-Versionen

www-plateau.cs.berkeley.edu/people/chaffee/joliet.html

CD-ROM: Rock-Ridge und andere

Rock-Ridge:

- Erweiterung von ISO-9660 für Unix-Systeme
- erlaubt lange Dateinamen
- Unix-style Datei-Attribute (owner, permissions)
- symbolische Links
- abwärtskompatibel (ISO-9660 Systeme sehen die 8+3 Daten)
- ftp.yimi.com/pub/rockridge/

Macintosh HFS:

- CD-ROM Format mit Apple's hierarchical file system
- völlig inkompatibel mit ISO-9660
- benötigt entsprechende Treiber

CD-ROM: El Torito

bootfähige CD-ROMs?

- "El-Torito" Spezifikation von Phoenix und IBM (1994)
- Grundidee: BIOS ersetzt Laufwerk A: durch die CD-ROM
- basiert auf ISO-9660 Dateisystem
- Sektor 16 wie gehabt für Primary Volume Descriptor
- Sektor 17 als Boot Record Volume Descriptor
- erlaubt mehrere Boot-Sektoren pro CD
- Zugriff über BIOS/DOS INT-13 Schnittstelle
- CD-ROM kann als Live-Filesystem genutzt werden

CD-ROM: El Torito

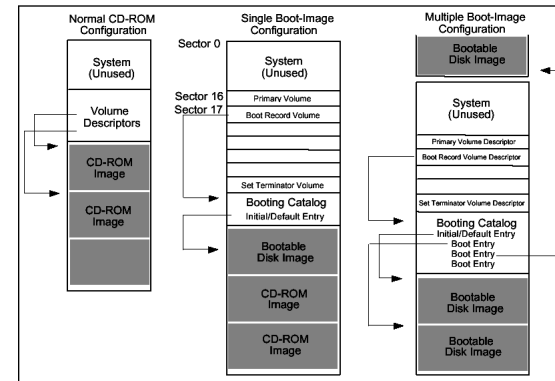
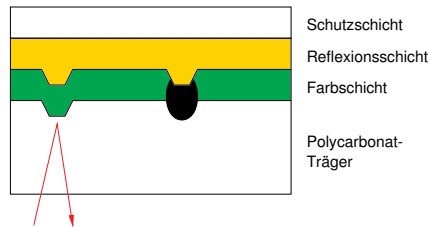


Figure 1. Three types of CD-ROM configuration:

1. The Normal CD-ROM configuration is not bootable, uses Root Directory and CD-ROM drivers to access CD-ROM images.
2. A BIOS with Single Boot-Image capability accesses the Initial/Default Entry to access a single bootable disk image. After loading the operating system, the system can revert to standard CD-ROM drivers and the Root Directory to access CD-ROM images.
3. A BIOS with Multiple Boot-Image capability can access any one of a number of Bootable Disk Images listed in the Booting Catalog. After loading the operating system, the system can access other items in the disk image with standard INT 13 calls or return to normal access of CD-ROM images using CD-ROM drivers and the Root Directory.

CD-R: Prinzip



- mechanische Prägung (Pits/Lands) nicht praktikabel
- statt dessen: Farbstoff durch Laserimpuls zerstören
- etwas andere Reflexionsdaten als CD
- Spurführung des Pickups erfordert Daten:
=> Rohlinge enthalten vorbereitete Spiralspur (siehe DVD pre-groove)

PC-Technologie | SS 2001 | 18.214

CD-R: erweitertes Lead-In

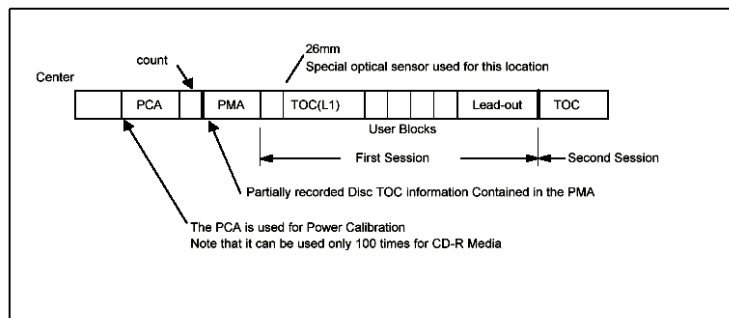
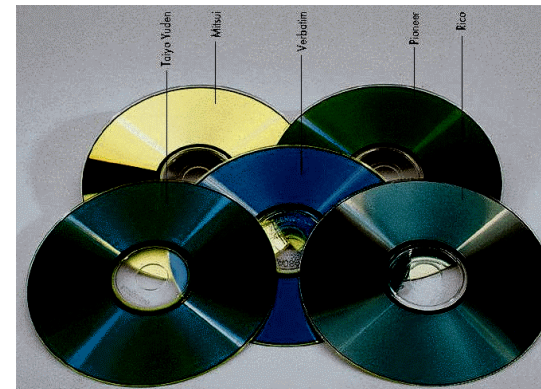


Figure 3 - CD-R/RW Disc Layout

- erweiterte Lead-In Zone (weiter innen als normale CD)
- u.a. Kalibrierung der Laserintensität beim Schreiben
- Audio/Datenformat unverändert

PC-Technologie | SS 2001 | 18.214

CD-R: Rohlinge, Farbstoffe

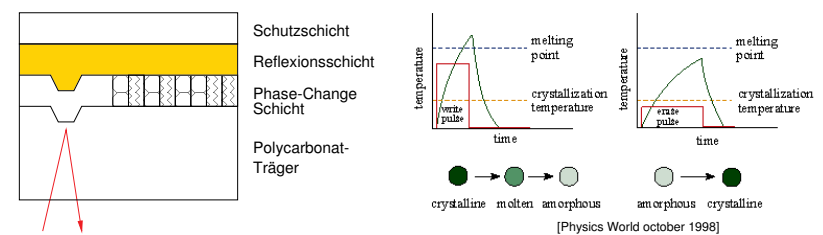


Cyanin Phthalocyanin Metallkomplex-Azo-Farbstoff

- diverse Farbstoffe, aber Haltbarkeit, Schreibeigenschaften ähnlich

PC-Technologie | SS 2001 | 18.214

CD-RW: Prinzip

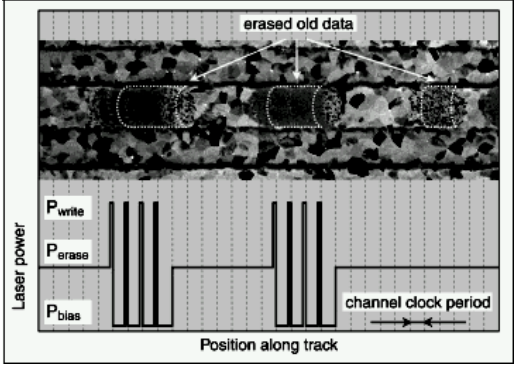


- Phase-Change Verfahren für wiederbeschreibbare CDs
- Material mit kristalliner / amorpher Struktur
- deutlich kleinere Reflexionsänderung als bei CD/CDR
- Umschalten durch schwache/starke Laserimpulse
- schnelle Abkühlung: amorph, langsame Abkühlung: kristallin
- bis zu 100.000 Mal wiederbeschreibbar (theoretisch)

PC-Technologie | SS 2001 | 18.214

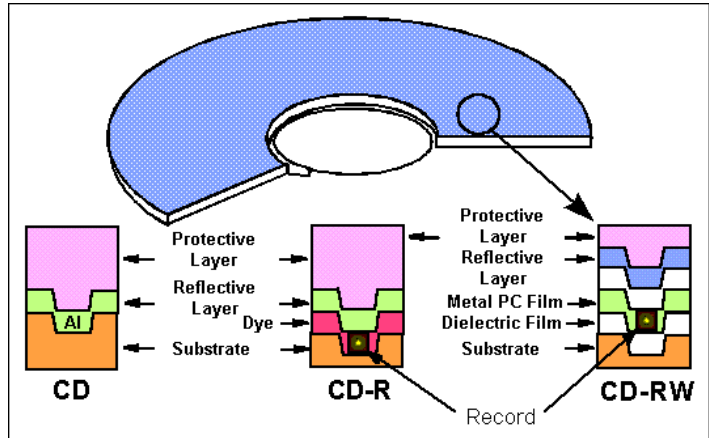
CD-RW: amorph / polykristallin

Figure 2. A phase change disc viewed through an electron microscope. The grooved structure is required for tracking during recording and reading. The amorphous marks show up as gray regions without a visible microstructure. The marks are surrounded by polycrystalline material consisting of a large number of small randomly oriented crystallites whose facets show up as sharp boundaries between the crystallites. Direct overwrite is done by adjusting the laser power to an erase level; erased marks show up as regions with smaller crystallite size.



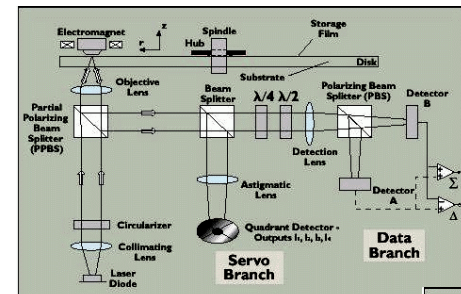
[CAOM 43-11]

CD-RW: Aufbau CD / CDR / CDRW



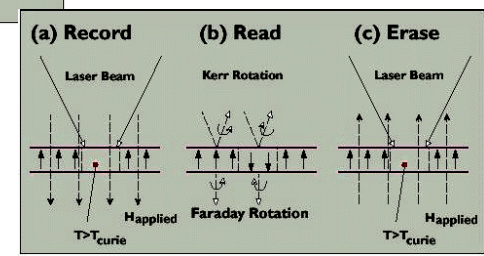
- CD-Pressung "parallel"
- CD-R / CD-RW Schreiben sequentiell, entlang der Rohspur

magneto-optische Verfahren (MO)

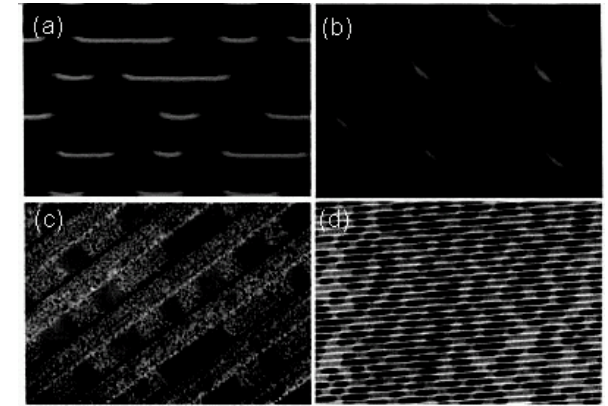


[CAOM 43-11]

- Mechanik wie bei CD
- zusätzlich Magnet (gegenüber Pickup)
- Intensitätsdifferenz durch Polarisation



Pits: CD, CD-R, CD-RW, MO



- a) CD (Pits gepreßt)
- b) CD-R (Pits gebrannt)
- c) CD-RW (amorph/kristallin)
- d) MO (Kerr-Effekt)

[PhysicsWorld October 1998]

CD-R: "Überbrennen"

- Länge der Rohspur definiert die Kapazität der CD-R/RW
- spezielle Rohlinge (80 Min) mit engerer Rohspur

"Überbrennen":

- angegebene Kapazität enthält >90 Sekunden Lead-Out
- plus einige Sekunden Reserve
- verkürztes Lead-Out erlaubt mehr Daten
- muß von Brenner und Software unterstützt werden (zB. www.feurio.com)
- evtl. Probleme mit älteren / abgenutzten Playern
- alternativ für Audio: Daten minimal stauchen

PC-Technologie | SS 2001 | 18.214

CD-R: Buffer-Underrun

- sequentielles Schreiben der CD-R:
- => Brenner benötigt kontinuierlichen Datenstrom
typische Puffergröße 2..4 MB

Problem Buffer-Underrun:

- CD-R entspricht nicht mehr den Normen
- Rohling defekt (CD-R) / neu formatieren (CD-RW)

www.burn-proof.com:

- Brenner rechtzeitig (kontrolliert) stoppen
- Position auf der CD-R merken (Spur, Position 100 µm)
- neu aufsetzen, sobald Daten verfügbar
- Fehlerkorrektur beseitigt die Lücke ("burst error")
- wird von einigen neuen Brennern unterstützt

PC-Technologie | SS 2001 | 18.214

CD-R: Audio-Grabbing

digitales Auslesen von CD-DA:

- optimal mit Audio-Playern (Digitalausgang, 1X Speed)
- Digitalausgänge an CD-ROMs selten / oft fehlerhaft

"Packet"-Interface problematisch:

- in alten Laufwerken schlecht implementiert
- Audio-Format hat keine fortlaufenden Sektor-IDs
- mm:ss:ff-Marken: ff-Werte fehlen manchmal
- Packet vs. Streaming: Probleme beim Wiederaufsetzen
- nur einfache Fehlerkorrektur, kein LEC

=> gutes Laufwerk notwendig

=> mehrfaches Lesen / Korrelation der Daten (cdparanoia)

PC-Technologie | SS 2001 | 18.214

CD: Audio Grabbing via SCSI3 MMC

Table 95 - CD-DA (Digital Audio) Data Block Format

Bit	7	6	5	4	3	2	1	0
Byte								
0	Left Channel (Lower Byte)							(LSB)
1	(MSB)	Left Channel (Upper Byte)						
2	Right Channel (Lower Byte)							(LSB)
3	(MSB)	Right Channel (Upper Byte)						
2348	Left Channel (Lower Byte)							(LSB)
2349	(MSB)	Left Channel (Upper Byte)						
2350	Right Channel (Lower Byte)							(LSB)
2351	(MSB)	Right Channel (Upper Byte)						

If the CD Drive does not support the CD-DA Stream-Is-Accurate capability, See Table 230 - CD Capabilities and Mechanical Status Page, then the digital audio data must be read as a continuous stream. If while streaming the drive must stop, there will be a non recoverable error generated READ ERROR - LOSS OF STREAMING. This is due to the 1 second uncertainty of the address. (i.e. there is no header in CD-DA data). Reissuing the command may not return exactly the same data as the previous try. When the drive supports the stream accurate capability, there will be no error, only some time delay for rotational latency.

PC-Technologie | SS 2001 | 18.214

UDF: Dateisystem

- CDR Medien sind nur einmal beschreibbar
- ISO-9660 erwartet TOC und Directories an fester Position
- => spätere Änderungen unmöglich

UDF-Dateisystem: "universal disk filesystem"

- basiert auf ISO 9660
- aber erweitertes, flexibleres Dateisystem
- "virtual allocation tables"
- gültiges Directory jeweils im letzten geschriebenen Block
- dort Verweise auf Dateien und ältere Directory-Blöcke
- keine Beschränkung der Verzeichnis-Schachtelungstiefe
- Finalisieren der CD erzeugt volles ISO 9660 Dateisystem
- www.osta.org

PC-Technologie | SS 2001 | 18.214

UDF: Packet Writing

- CDR Medien sind nur einmal beschreibbar

UDF-Packet Writing:

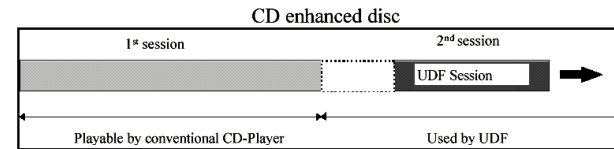
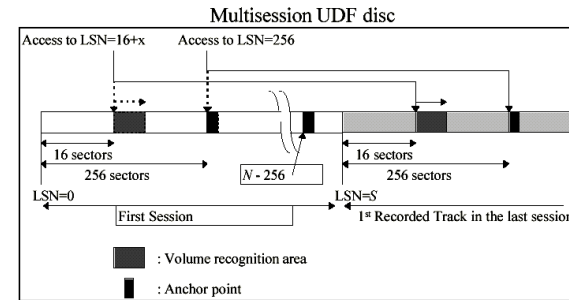
- Dateien in einzelnen kleinen Paketen schreiben
- zunächst ohne TOC im Lead-In

"virtual allocation tables":

- gültiges Directory jeweils im letzten geschriebenen Block
- dort Verweise auf Dateien und ältere Directory-Blöcke
- Dateien können immer noch nicht gelöscht werden
- neues Directory ohne Verweis auf gelöschte Datei schreiben
- Datei modifiziert:
- neue Datei schreiben, neues Directory schreiben

PC-Technologie | SS 2001 | 18.214

UDF: Multisession / enhanced disks



PC-Technologie | SS 2001 | 18.214

Leerseite

PC-Technologie

DVD: Konzept und Anforderungen

Anforderungen für DVD-Video:

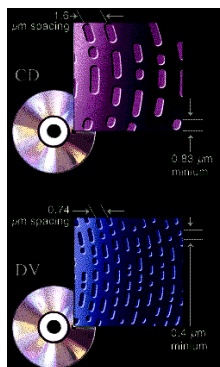
- 135 Minuten Spieldauer pro Seite
- bessere Auflösung als die Laserdisc
- Surround-Audio in CD-Qualität
- Audiospuren für 3-5 Sprachen
- Untertitel in mehreren Sprachen
- diverse Bildformate (Letterbox, Pan, Widescreen)
- Interaktion wie bei Video-CDs
- Jugendschutz
- Kopierschutz
- CD-kompatibel
- Herstellungskosten ähnlich wie CDs

MPEG-2, AC3:
=> ca. 6 Mb/s
=> 4-5 GB / Seite

DVD: Verbesserungen

höhere Kapazität der DVD gegenüber der CD:

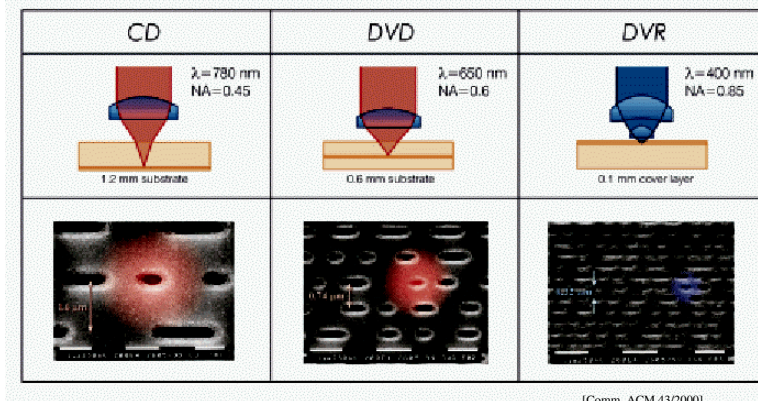
- kleinere Pits, kleinerer Spurbestand
- veränderte Header-Strukturen
- weniger Parity-Bits
- Weglassen der Subcodes
- 2048-Byte Sektoren
- kleinerer nicht-genutzten Innenteil (Radius CD 25 mm, DVD 24 mm)



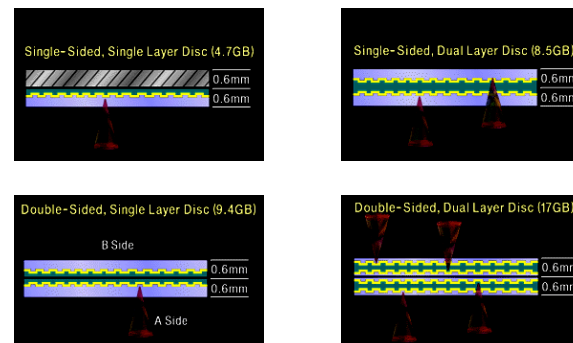
=>	DVD-5	single layer	4.4 GB	6.7x
	DVD-9	double layer	8.0 GB	12.3x
	DVD-10	double side	8.8 GB	13.5x
	DVD-18	DS / DL	15.9 GB	24.4x CD-ROM

DVD: Vergleich mit CD und DVR

Figure 3. Three generations of optical disc systems. Progress in "areal density," or bit count per unit area, takes big steps; a CD holds 650MB, a DVD 4.7GB, and a DVR 22GB) by reducing the spot size through a shorter wavelength and stronger objective lens (with higher numerical aperture). The electron micrographs show read-only discs with replicated pit patterns.



DVD: 4 Formate



- single/double side
- single/dual layer (äußere Schicht halbdurchlässig)

DVD: Datenformat (Sektorformat)

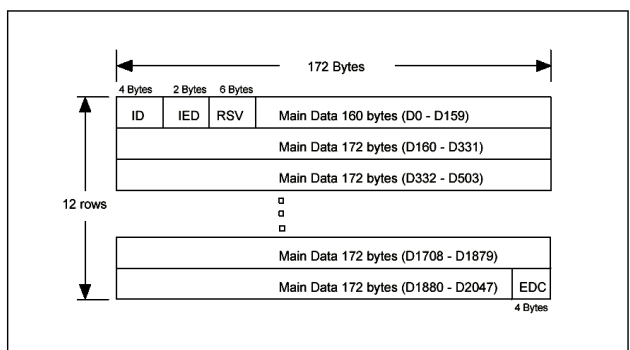


Figure 13 - Sector Layout

- eindeutige Block-ID, 4-Byte layered ECC
- vergleiche CDROM

DVD: Datenformat (Header)

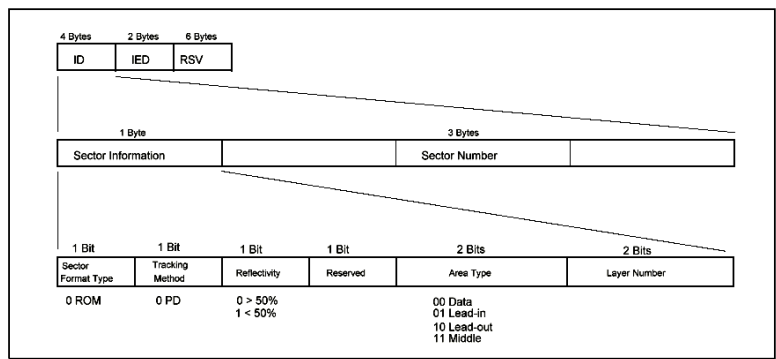


Figure 11 - Header Layout

- vollständige Information für jeden einzelnen Block
- 2-bit layer number: Seite 1/2, außen/innen
- vergleiche CDROM

DVD: Sektoranordnung dual-layer

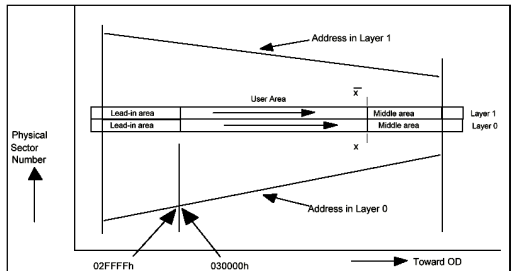


Figure 10 - Opposite Track Path Description

Layer-Umschaltung durch Fokussierung (schnell)

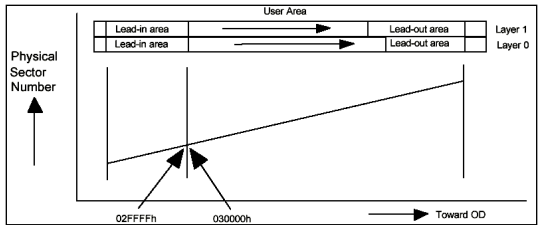
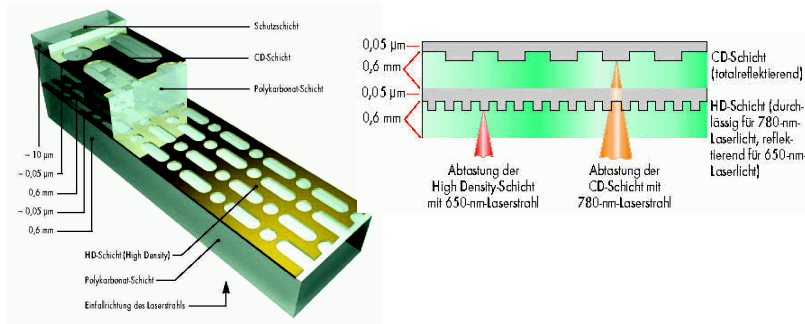


Figure 9 - Parallel Track Path Description

Layer-Umschaltung erfordert Kopfneupositionierung

DVD: Super Audio CD



- Kompatibilität mit Audio-CDs
- zweite (DVD)-Schicht mit Stereo "bitstream", 2.8 Mb/s
- vgl. DVD-Audio

[ct 21/98 242]

PC-Technologie | SS 2001 | 18.214

DVD-Audio

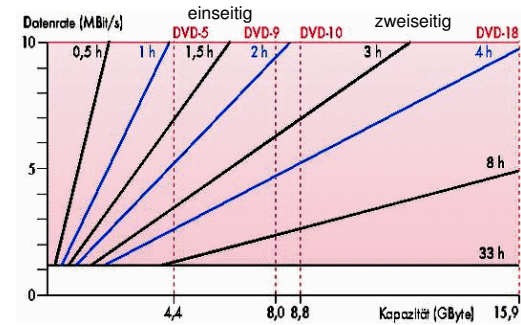
- Spezifikation für DVD-Audio seit Q1/1999
- nutzt die DVD-5 (4.7 GB)

diverse Audioformate werden unterstützt:

- Abtastraten 44.1 / 48 / 88.2 / 96 KHz
- Quantisierung mit 12 / 20 / 24 bit
- mindestens 74 min. Spieldauer für alle Modi
- 16 bit, 44.1 Stereo, 7 Stunden Spieldauer
- 24 bit, 96 KHz, 2-6 Kanäle Surround
- 24 bit, 192 KHz Stereo
- Dolby Digital, DTS, MPEG-AAC, ...
- plus Standbilder und Textinformationen
- bisher kaum erhältlich

PC-Technologie | SS 2001 | 18.214

DVD: Video, Datenrate vs. Spieldauer



- Formate: 720x576x25 PAL / 720x480x29.97 NTSC
- 2 Stunden Spieldauer gewünscht, bei 5 GB
- typische Datenrate für MPEG-2 mit AC3-Audio
- Digitales Fernsehen: DF1 sendet MPEG-2 mit

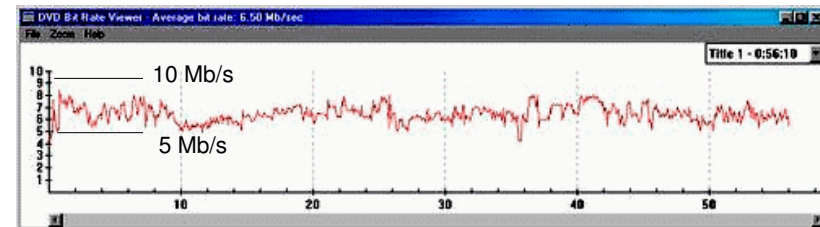
5.5 Mb/s

1.5 .. 9.8 Mb/s

6.8 Mb/s

PC-Technologie | SS 2001 | 18.214

DVD: Datenrate MPEG-2

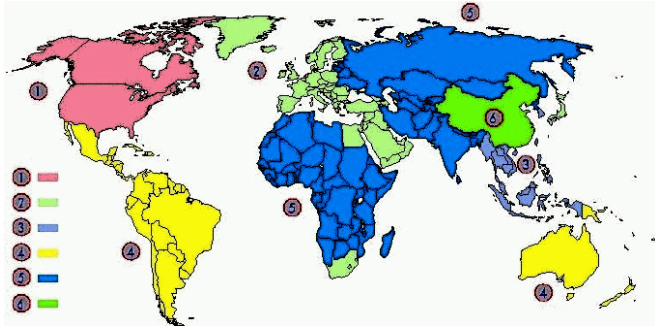


[ct 20/99, Sesamestreet, Region 0]

- typische Datenrate der Video-DVD ist 5 .. 10 Mb/s
- andere Datenformate (MPEG-4) bisher nicht verwendet
- siehe MPEG Standards

PC-Technologie | SS 2001 | 18.214

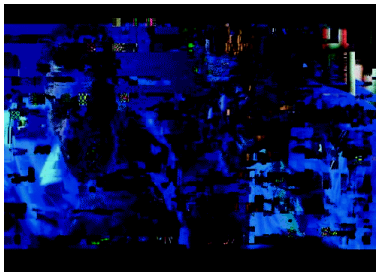
DVD: Region Codes



DVD-Video spezifiziert Region-Codes

- zeitversetzte Veröffentlichungen zwischen USA / EU / Japan
- Sicherung des Kino-Marktes
- Region 0 ist universell nutzbar
- Code in Laufwerks-Firmware, typisch höchstens 5x wechselbar

DVD: CSS



- direkte Kopie einer DVD-Video
- Daten großteils unlesbar
- nicht alle DVDs sind verschlüsselt

"Content Scrambling System":

- Schutz vor digitalen (=perfekten) Raubkopien
- verschlüsselte Übertragung zwischen Laufwerk und Decoder (HW/SW)
- komplexes Challenge-Response-Protokoll zur Authentifizierung
- Codes im Lead-In der DVD gespeichert, dort nicht zugreifbar
- Verfahren nicht publiziert, nur für Hersteller zugänglich

DVD: DeCSS

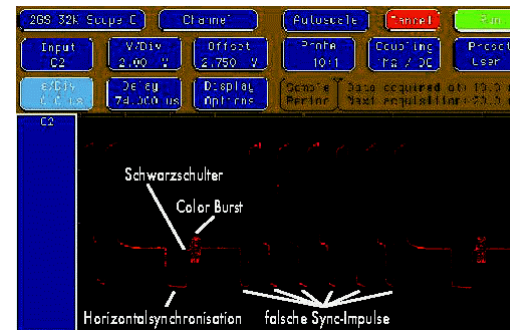
mittlerweile ist CSS geackert:

- diverse Angriffspunkte in den Windows Treiberschichten
- Screenshots -> AVI
- Software-Player cracken
- algorithmische Angriffe möglich wegen 40-bit Schlüssel

DeCSS:

- Windows-GUI
 - dekodiert DVD-Daten auf die Festplatte
 - verwendet Player-Key aus Xing Software-Player
 - Verbreitung via Internet / Abmahnungen durch DVD-Anwälte
- => ermöglicht DVD-Player für Linux
- => rechtliche Situation unklar

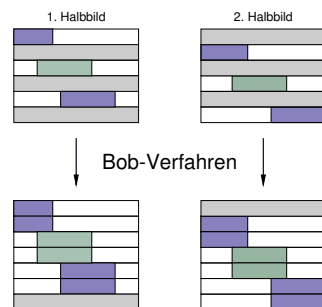
DVD: Macrovision



Schutz gegen analoge DVD-Kopien auf Videorekorder:

- zusätzliche Synchronimpulse
- AGC: wechselnde Schwarzschulter-Werte
- im unsichtbaren Bereich: Fernseher ignoriert das Signal

DVD: Interlace-Probleme



De-Interlacing:

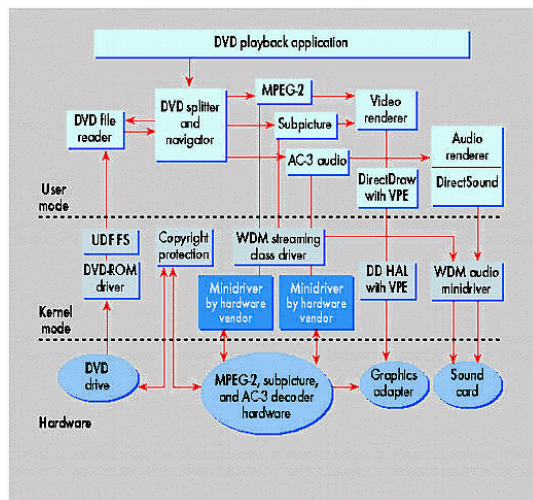
- Kino: Vollbilder (24 Bilder/s)
- Fernsehen: Halbbilder:

PAL, Secam: 25 / 50
NTSC: 30 / 60

- Monitore: Vollbilder bei hoher Wiederholrate

- => komplexe Umrechnung notwendig
- => sonst schlechte Bildqualität (Fransen, Kammefekte)
- => Kinofilme laufen um Faktor 25/24 zu schnell . . .

DVD: Windows-Treiber ...



DVD: Laufwerke Q4/99

Benchmark-Ergebnisse DVD-ROM-Laufwerke

DVD-ROM-Modell	maximale Datenrate DVD-9	abspielbar von Fehler-DVD	Audio-Grabben	mittlere Zugfrequentz	mittlere Datenrate CDR	mittlere Datenrate CD-RW
	MB/s/s. besser >	Prozent/ besser >	Datenrate Kiste besser >	Hz/ besser >	MB/s/s. besser >	MB/s/s. besser >
DVD-ROM ATAPI						
Alteey DVD-DD-3206E	6,3	44	3,8	127	2,9	0,6
AOpen DVD9432	3,4	99	10	97	3,5	2,1
Creative L. Encore 6X	8,1	79	9	134	2,7	1,6
Guillemot Thecker 6X	8,0	89	6	88	3,7	2,1
Hirsch GD-2500 6X	8,1	76	13	131	2,7	1,6
NEC DV-S900	6,2	83	6	85	4,1	0,9
Toscano SF 8583	6,8	48	11	132	3,7	0,9
Philips DVD932	8,1	44	6	93	3,7	2,1
Power DVD-4035	3,4	100	12	92	3,4	3,4
Sony DD1220E6	6,6	90	11	146	2,0	1,1
DVD-ROM SCSI						
Power DVD-L035	3,4	34	10	94	3,5	3,2
Toshiba SD-M1201	6,3	90	6	124	3,7	2,2
DVD-LAM						
AOpen DVD-S205	2,7	94	9	117	1,7	1,7
Hirsch GF 1000	2,7	36	10	265	1,4	1,7

In **abspielbar von Fehler-DVD** steht die Menge des von der Fehler-DVD abspielbaren Materials in Prozent. Als DVD-Player diente PowerDVD 1.40 unter Windows 98. Von jedem Clip wurde die roheffektive Spielzeit bis zum Hängenbleiben gezählt. Störende Passagen wickelten wir nur mit der Hälfte ihrer Dauer. Wenn ein ganzer Clip nicht rochelte, erzielte er nur 1 bis 5 %. Bei ein oder zwei kleinen Aussetzern in einem Clip wurde dieser mit 98 bis 99 % gewertet. Das Ergebnis ist der Mittelwert der Clips von 2 bis 8.

Die in der Spalte **Audio-Grabben** angegebenen maximalen Datenraten gelten grundsätzlich nur für fehlerlos gelesene Tracks. Im Unterschied zum Lesen von CD-ROM arbeiten die Laufwerke beim Grabben verteilend mit konstanter Datenrate [CV = Constant Linear Velocity]. Zum Audio-Grabben dienten die beiden Program...

me WinDAC32 in der Version 1.48 [www.win-dac.de] sowie als Gegencheck CD Speed 99 Version 0.41 Beta [http://come.to/catspeed].

Die **mittlere Datenrate** versteht die Messungen von fünf CD-Recordables mit unterschiedlichen Farb- und Reflexionsstufen (Cyan/Gold, Cyan/Silber usw.) zu einem Durchschnittswert.

Zum Benchmark und zum Abspielen der DVD-Videos mit Softwarecodern stand ein Asus P2B mit Pentium-II-333 bei 66 MHz Buszeit unter Windows 98 zur Verfügung. Die DVD-Laufwerke waren am sekundären EIDE-Kanal als Master beziehungsweise an einem Wide-Ultra-SCSI-Adapter SYMBIOS 10 von Symbios Logic angeschlossen. Zur Anzeige diente eine Diamond Viper V550 mit 16 MB/640x480 RAM. Die Geschwindigkeitsmessungen wurden mit einer speziellen, nicht öffentlichen Version von H2Bench unter Windows 98 durchgeführt.

- DVD 6x, Tendenz steigend

[ct 20/99]

DVD: DVD-R

DVD-Recordable:

- voll kompatibel mit DVD-Video, DVD-Audio, DVD-ROM
- kann auf jedem DVD-Player abgelesen werden
- Recorder sind noch extrem teuer

	Ver 1.0	Ver 1.9 / 2.0
Seiten	1 / 2	1 / 2
Kapazität	3.95 GB	4.7 GB
verfügbar	1997	1999
Pit µm	0.44 x 0.80	0.40 x 0.74
Verfahren	Farbstoffe wie CDR, 635 nm Laser	

DVD: DVD-RAM

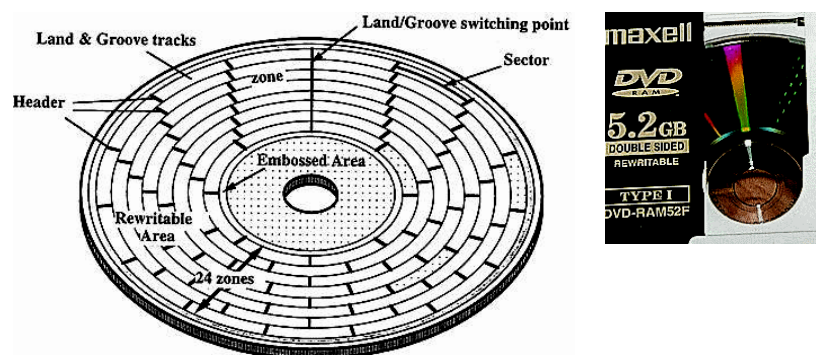
wiederbeschreibbare DVDs:

- mehrere, untereinander inkompatible Verfahren
- nicht mit DVD-R kompatibel
- Phase-Change-Technik wie CD-RW
- zoned CLV, wobbled pre-groove, usw.

	Ver 1.0	Ver 1.9 / 2.0
Seiten	1 / 2	1 / 2
Kapazität	2.6 GB	4.7 GB
verfügbar	1997	1999
Pit μm	0.41 x 0.74	0.28 x 0.615
Verfahren	phase change wie CD-RW	

PC-Technologie | SS 2001 | 18.214

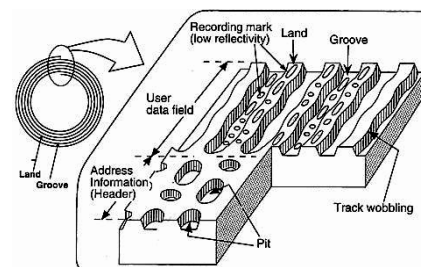
DVD: DVD-RAM Sektoren



- Disk ist in 24 Zonen eingeteilt
- innerhalb einer Zone konstante Umdrehungsgeschwindigkeit

PC-Technologie | SS 2001 | 18.214

DVD: DVD-RAM Pregroove

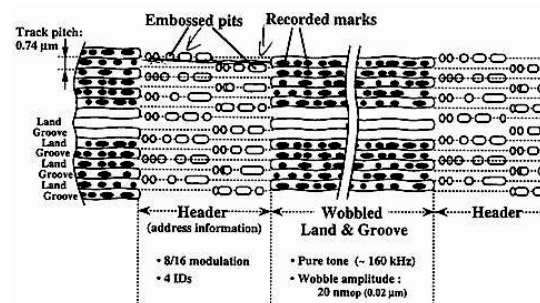


CD/DVD-Rohlinge enthalten eine Roh-Spur:

- Roh-Spur für Spurführung des Schreib/Lese-Kopfes
- Track-Wobbling für Drehzahlregelung
- regelmässig Header-Sektoren
- DVD-RAM: Daten abwechselnd in Lands und Grooves

PC-Technologie | SS 2001 | 18.214

DVD: DVD-RAM Datenaufzeichnung



- Rohlinge enthalten fertige Header-Zonen
- DVD-RAM Typ, Kapazität, Schreibgeschwindigkeit, usw.
- Aufzeichnung nur in die Datenbereiche
- UDF: max 1 Partition a 2.3 GB / Seite
- FAT 16: mehrere Partitionen a 2 GB / Seite

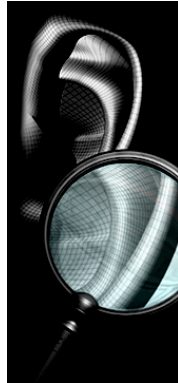
PC-Technologie | SS 2001 | 18.214

Audio: Agenda

- Einführung
- digitale Signalverarbeitung

- Audiowiedergabe
- AC97 / AMR
- virtuelle Studios

- DirectSound | ALSA
- 3D-Audio



PC-Technologie | SS 2001 | 18.214

Leerseite

PC-Technologie

Leerseite

PC-Technologie

Leerseite

PC-Technologie

Audio: Anwendungen

Wozu PC-Audio?

- | | |
|---|-----------------------------|
| <ul style="list-style-type: none"> • Musik/Videos abspielen • Sprachausgabe / -eingabe | CD, DVD MP3, AC3, MD, ... |
| <ul style="list-style-type: none"> • Streaming-Media, Telephonie • Modem-Funktionen | RA & Co
AC97, AMR, CMR |
| <ul style="list-style-type: none"> • Musikaufnahme / -produktion • Musikinstrument, Synthesizer | virtuelle Studios |
| <ul style="list-style-type: none"> • 3D-Audio für virtual reality | Spiele, VR-Anwendungen |
- => höchst unterschiedliche Anforderungen
Bandbreite/Rechenleistungen von KB/s bis GB/s

PC-Technologie | SS 2001 | 18.214

Audio: Trends

- Moore's Law: Rechenleistung steigt 50% / Jahr:
 - immer bessere DSP-Algorithmen
- => Ersetzen externer HW-Geräte durch SW
AC97-Codec statt Soundkarte
- => Highend-Soundkarte statt Tonstudio
low-cost HD-Recording
virtuelle Tonstudios, volle Audio/MIDI-Integration
- => völlig neue Möglichkeiten
bessere Tonqualität (24 bit, 96 KHz), Surround 5.1, etc.
3D-Audio statt Stereo oder 5.1
Echtzeitmanipulation von Audiodaten
(z.B. Autotune zur Gesangskorrektur)



PC-Technologie | SS 2001 | 18.214

Sie sehen gut aus, aber Ihr Gesang ist lausig?

ANTARES AutoTune und ATR-1: Perfekter Gesang aus der Box

Sie sehen gut aus, aber Ihr Gesang ist lausig? Kein Problem, denn mit ANTARES Auto Tune als Software oder dem ATR-1 Hardwarerack wird Ihre Aufnahme trotzdem perfekt!

- 19" Gehäuse
- Datenformat 20 bit linear, 56 bit intern
- Samplingfrequenz 46,875 kHz
- AD-Wandlung 20 bit (103 dB Dynamic Range)
- DA-Wandlung 24 bit (105 dB Dynamic Range)
- LiveTauglich durch MIDI-Steuerung
- Inputs und Outputs:
XLR symmetrisch,
Klinke symmetrisch/unsymmetrisch
- Display: 2 x 20 Zeichen LCD,
Korrektur-Indikator 10 x LED,
Input-Level 6 x LED

Softwareversionen erhältlich für:
• Stand alone, TDM und VST für Mac
• DirectX für IBM-kompatible PCs

DM 1.998,-
unverändliche Preissampflösung

hy per act ive

**Hyperactive
Audioteknik
GmbH**
Silberbachstraße 9
85272 Tumbasheim
Tel. (0 81 28) 98 23 27
Fax (0 81 28) 98 23 28
hyperactive@t-online.de
Mitglied im VVM e.V.
Fördermitglied des VOT

[Echtzeitkorrektur von Gesang / Instrumenten, seit Q1/1999]

PC-Technologie | SS 2001 | 18.214

Audio: Literatur

- | | |
|--|---|
| developer.creative.com | (Soundblaster Infos, EAX Specs und SDK) |
| developer.intel.com/design/idf/ | (Intel Developer Forum 2000, AC97/3D Audio) |
| developer.intel.com/ial/scalableplatforms/audio | (Intel Audio roadmap, AC97 und AMR specs) |
| www.microsoft.com/directx | (Microsoft DirectX homepage und download) |
| www.opensound.com | (Linux Opensound Treiber) |
| www.alsa-project.org | (Advanced Linux Sound Architecture) |
| www.sensaura.com | (Sensaura 3D Audio) |
| www.dolby.com | (Dolby Labs, AC3 specs usw.) |
- diverse c't Artikel
- Bargen, Donnelly: Inside DirectX, Microsoft Press, 1998
- Savell: EMU10K1 digital audio processor, IEEE Micro 02/1999
- Zeitschriften Keyboards, Keys, ...

PC-Technologie | SS 2001 | 18.214

Audio: typische Datenformate

- Beispiele für verbreitete Formate (ohne Kompression):

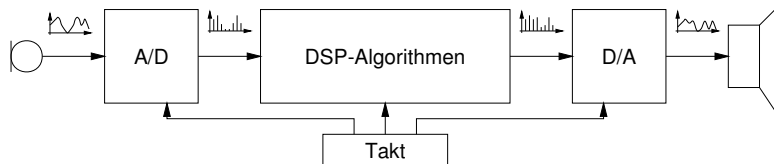
Sun .au:	8 KHz, 8 bit, mono	8 KB/s
CD-DA:	44.1 KHz, 16 bit, stereo	176 KB/s
ADAT:	48 KHz, 16 bit, 8 Kanäle (4 stereo)	768 KB/s

- für DVDs:

	PCM	Dolby Digital	MPEG-Audio	DTS	SDDS
Verfahren	linear	AC3	MP1L3, MP2L2	APT	ATRAC
Sample-frequenz	44.1K, 48K, 96K	48K	48K	48K	48K
Datenrate bis	6 Mb/s	448 Kb/s	640 Kb/s	768 Kb/s	1.28 Mb/s
Kanäle	8	5.1	7.1	7.1	5.1
DVD-Player					
PAL	ja	ja	MPEG1	optional	optional
NTSC	ja	ja	optional	optional	optional

PC-Technologie | SS 2001 | 18.214

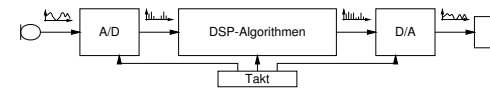
Prinzip digitaler Signalverarbeitung



- analoge Eingangssignale (zeit- / wert-kontinuierlich)
 - analog/digital-Wandlung (zeit- / wert-diskret)
 - digitale, diskrete Verarbeitung
 - digital/analog-Wandlung (Tiefpaßfilter)
 - analoge Ausgangssignale (zeit- / wert-kontinuierlich)
- Nyquist-Theorem für Abtastrate
 - Hörbereich bis ca. 20 KHz (=> >40 KHz Abtastrate)

PC-Technologie | SS 2001 | 18.214

DSP: Verstärker, Mixer



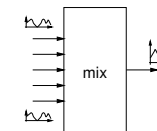
- digitaler Verstärker

```
output[t] = input[t] * gain;
```



- digitaler Mixer:

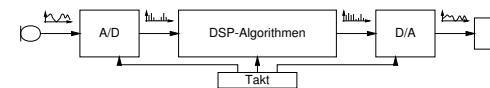
```
output[t] = 0.0;
for( int i=0; i < n_inputs; i++) {
    output[t] += input[i][t] * gain[i];
}
```



- viele MAC-Operationen (multiply-accumulate)
- Overflow beachten, saturation arithmetic

PC-Technologie | SS 2001 | 18.214

DSP: Echo, Hall, und mehr



- Verzögerung: auf alte Abtastwerte zurückgreifen

```
delay[t] = input[t]*gain + input [t-t2]*gain2;
```

- Rekursion möglich: Echo, Hall

```
hall[t] = input[t]*gain + hall[t-t2]*gain2;
```

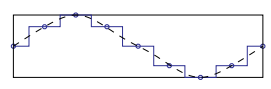
- Algorithmen für viele Anwendungen: Verzögerung, Echo, Hall, Overdrive, Verzerrung, ... Filter, Formatfilter, Tonhöhenänderung, Tempoänderung, ...

PC-Technologie | SS 2001 | 18.214

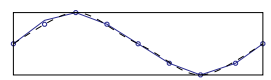
DSP: Sampling

- Sampling: Abtastwerte abspeichern

```
sample[t] = input[t];
```

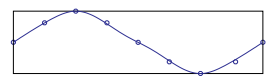


- Samples direkt abspielbar (CD, Spiele)



- "Wavetable"-Synthesizer:

```
output[t] = sample[t*pitch]
            interpolate( sample[] )
            filter( interpolate( sample[] ) )
            effects( filter( interpolate( sample[] ) ) )
```



Interpolation: nearest / linear / splines / ...

Audio: Soundblaster

- erste verbreitete Soundkarte, ISA-Bus, 8-bit D/A
- spätere Erweiterungen: 16-bit, OPL3, MPU401, ...
- keine API, direkte Registerzugriffe
- belegt sehr viele ISA-Ressourcen (bis 2 IRQ, 2 DMA, Ports)
- aber de-facto Standard
- Kompatibilität wird auch in AC97 Spec noch gefordert
- stirbt (mit DOS-Spielen) langsam aus

(leider bisher keine Abbildung gefunden)

Audio: AC 97

AC97: Intel Vorschlag für PC-Audio

- zwei Chips: Controller (digital) und Codec (analog)
- Analog-Codec klein und billig, Gehäuse definiert
- AC-Link Interface zwischen Controller und bis zu vier Codecs
- 16-bit stereo, full duplex, 48 KHz Abtastrate
- Rauschabstand: S/N besser als 90 dB
- vier Stereoeingänge (CD, Video, Line In, Aux)
- zwei Mikrophoneingänge
- Stereoausgang, zusätzlich Monoausgang für PC-Lautsprecher
- Power-Management
- optional bis 20 bit Auflösung
- optional Klangregelung, Loudness, 3D-Basisverbreiterung
- optional Kopfhörerausgang

Audio: AC 97 Architektur

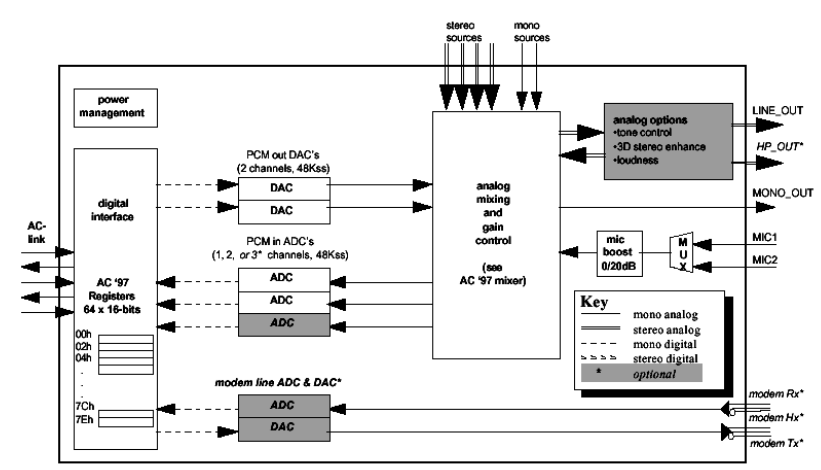
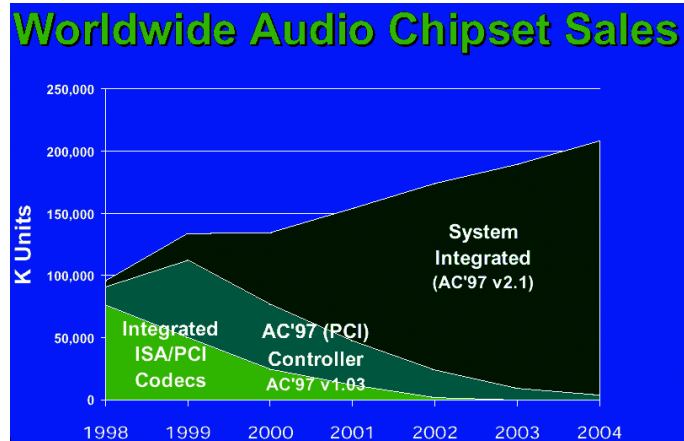


Figure 1. AC '97 1.0 Block Diagram

Audio: AC 97



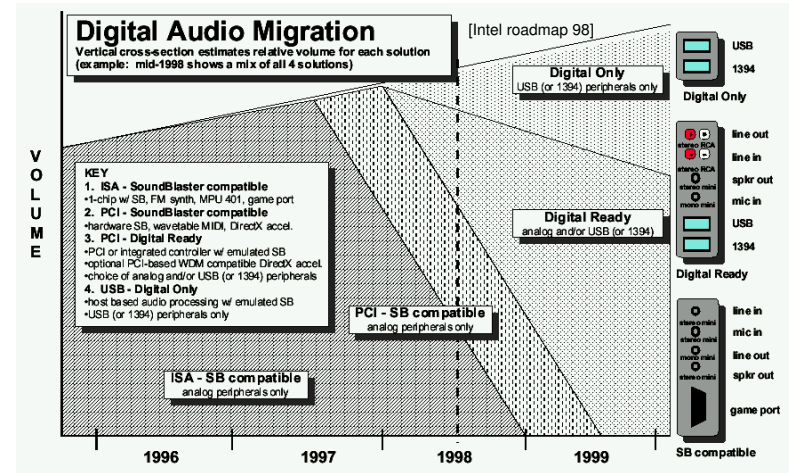
- externe Soundkarten sterben aus [IDF 2000]

Audio: Intel Roadmap

Audio '98 Roadmap:

- Hardware:
Audio-Controller Funktionen im Chipsatz
analoger Codec durch USB/1394 Geräte ersetzt
- Software:
weitere DirectSound Funktionen
Hardwarebeschleunigung für Mixer, Synth, 3D
Verzicht auf Soundblaster-Kompatibilität

Audio: Digital Audio Migration



- Verspätung: bisher (Q2/2000) kaum rein digitale Systeme

virtuelles Studio: Konzept

virtuelles Studio

- Mikrophone
- AD-Wandler mit Verstärker
- SW-Sampler/Synthesizer
- SW-Mischpult
- SW-"Plugins"
- Festplatten (HD-Recording)
- Einstellungen:
SW-Patchfelder / MIDI
reproduzierbar (total recall)
speicherbar und automatisierbar
- Cubase / Logic / ProTools / . . .

herkömmliches Studio

- Mikrophone
- Vorverstärker
- Sampler/Synthesizer
- Mischpult
- Effektgeräte
- Tonbandmaschine
- Patchkabel / via MIDI
- nicht reproduzierbar
- nicht speicherbar

virtuelles Studio: *Emagic Werbung*



PC-Technologie | SS 2001 | 18.214

virtuelles Studio: *HD-Recording*

Tonaufnahme direkt auf Festplatte:

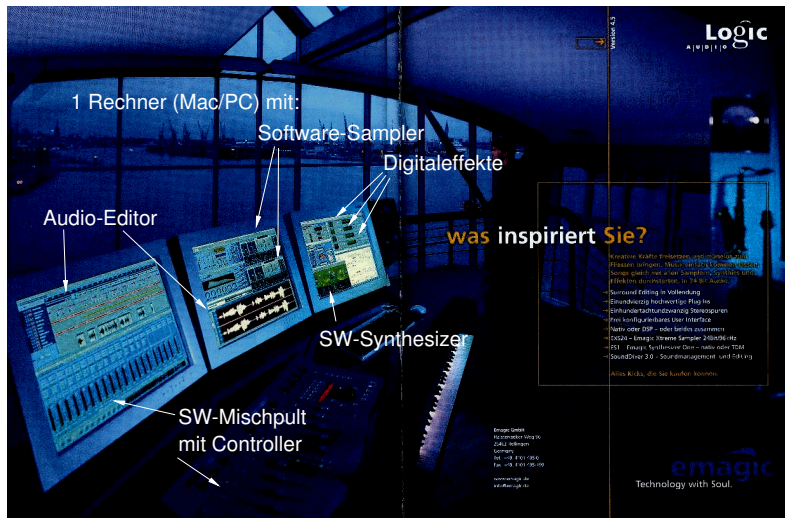
- 16 bit, 48 KHz: ~ 100 KB/s pro Spur
- HD-Dauertransferrate ~ 5 .. 10 MB/s
- HD-Kapazität ~ 10 .. 20 GB

- => 50 .. 100 Tonspuren pro Platte
- => Aufzeichnungsdauer ~ 4000 Sekunden (50 Spuren)
- ~ 5 Stunden (10 Spuren)

- geringere Kosten als Magnetbänder (!)
- direkter Zugriff, kein Umspulen
- nichtlineare Aufzeichnung, einfaches Editieren
- Mixdown auf DAT / direkt auf CDR

PC-Technologie | SS 2001 | 18.214

virtuelles Studio: *Emagic Werbung*



PC-Technologie | SS 2001 | 18.214

virtuelles Studio: *Steinberg VST-2*

Steinberg VST2:

[www.steinberg.de]

- Schnittstelle zwischen virtuellen Geräten
- Audio- und MIDI-Funktionen, samplegenau
- Integration in Cubase / andere Hostapplikationen
- realisiert als C++ Basisklasse, implementiert für PC/Mac/SGI
- 32-bit Gleitkomma für alle Datenoperationen
- ISSE-Optimierung
- zusätzliche GUI-Wrapper für Oberfläche der Plugins
- minimaler Overhead, optimale Performance
- als Standard etabliert

siehe Beispiel:

PC-Technologie | SS 2001 | 18.214

virtuelles Studio: VST-2 Plugin

```
#include "AGain.hh"

AGain::AGain( audioMasterCallback audioMaster )
    : AudioEffectX( audioMaster, 1, 1 ) // 1 program, 1 parameter
{
    fGain = 1.0;           // default gain 0 dB
    setNumInputs( 2 );    // stereo in
    setNumOutputs( 2 );  // stereo out
    setUniqueID( "AGain" ); // unique name for this plugin
    canMono();           // ok to feed with input with same values
    canProcessReplacing(); // accumulate / overwrite
    strcpy( programName, "default" );
}

void AGain::setParameter( long index, float value )
{
    fGain = value;
}

...

void AGain::process( float **inputs, float **outputs, long n_samples )
{
    float *in1 = inputs[0]; float *out1 = outputs[0];
    float *in2 = inputs[1]; float *out2 = outputs[1];

    while( --n_samples >= 0 ) {
        (*out1++) += (*in1++) * fGain; // accumulating: Mixer
        (*out2++) += (*in2++) * fGain; // should use ISSE/3Dnow
    }
}

...
```

PC-Technologie | SS 2001 | 18.214

Audio: Soundblaster Live

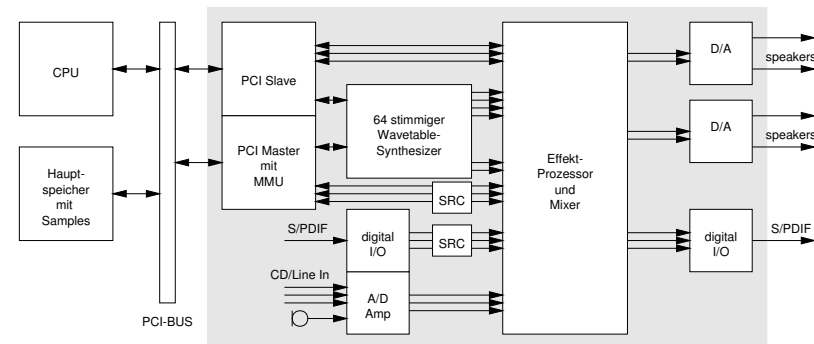
Beispiel Soundblaster-Live:

www.sblive.com, www.emu.com

- state-of-the-art PC-Soundkarte
- Wiedergabe und Mixer mit 16 bit, 48 KHz
- 6 analoge Eingänge, 4 analoge Ausgänge
- S/PDIF Digitaleingang und -ausgang
- 64-stimmiger Synthesizer/Sampler (3 MIDI-Kanäle)
- reserviert (max.) halben PC-Hauptspeicher für Samples
- hochwertige Digitaleffekte
- Hardwareunterstützung für 3D-Audio
- unterstützt alle aktuellen Softwareschnittstellen
- digitaler Signalprozessor EMU10K1
- zusätzlich einige Analogbauelemente
- ca. DM 100,00 - Profi-Variante DM 1.000,00 (bessere Wandler)

PC-Technologie | SS 2001 | 18.214

Audio: Emu 10K1 Blockschaltbild



- PCI-Busmaster mit eigener MMU
- bis zu 32 MByte Samples im Hauptspeicher
- vier analoge Ausgänge, digital S/PDIF inklusive 5.1

PC-Technologie | SS 2001 | 18.214

Audio: FM-Synthese

Frequenzmodulation als Syntheseverfahren

- X.Y (Stanford, 19XX)
- "typische" Klänge, z.B. "glockige" E-Pianos
- berühmt durch Yamaha DX-7 Synthesizer
- Yamaha OPL3-Chip auf vielen Soundkarten
- oft als GM Tonerzeugung mißbraucht
- PC I/O-Mapping:

0x388	Indexregister
0x389	Datenregister

$$FM(t) = A1 * \sin(f1*t + A2 * \sin(f2*t))$$

PC-Technologie | SS 2001 | 18.214

Direct Sound: Übersicht

DirectSound:

- Microsoft API zur Audioverarbeitung
- als Ersatz für direkte SB16 Registerprogrammierung
- Abspielen und Mischen von beliebig vielen .wav Quellen
- automatische Verwaltung der benötigten Puffer
- erkennt und benutzt vorhandene Hardware-Funktionen
- Hardware-Emulation in Software, wenn notwendig
- Sample rate conversion
- 3D-Funktionen inklusive HRTF
- (bisher nur) ein Stereo-Ausgang
- nicht für "Profi"-Applikationen geeignet

PC-Technologie | SS 2001 | 18.214

Direct Sound: Prinzip

- basiert auf Microsoft's COM Objektmodell
- Objektbasiert, aber Zugriff über "nacktes" C
- Gerätehersteller liefert die notwendigen low-level Treiber
- DirectSound-Applikation:
 - DirectSound-Objekt anlegen
 - gewünschte Hardwaregeräte auswählen
 - Lautsprecherkonfiguration auswählen (Aufstellwinkel)
 - benötigte Soundpuffer anlegen
 - WAV-Daten in die Soundpuffer schreiben
- Mischen und Ausgabe wird von DirectSound erledigt

PC-Technologie | SS 2001 | 18.214

Direct Sound: Features

externe Dokumentation:

- DirectSound Dokumentation (.doc)
- DirectSound Headerdateien
- Quelltext für DirectSound Beispielprogramme

PC-Technologie | SS 2001 | 18.214

Direct Sound: 3D-Audio

- internes Koordinatensystem
- DirectSound3DListener Position und Ausrichtung des Hörers
- DirectSound3DPuffer eine 3D-Schallquelle
- Abschwächen der Lautstärke als Funktion der Entfernung
- Balance links/rechts abhängig von der Richtung
- Zeitverzögerung zwischen linkem/rechtem Ohr
- Ausrichtung der Schallquellen (Ausbreitungstrichter)
- Dämpfung von Schallquellen hinter dem Hörer
- Doppler-Effekt für bewegte Schallquellen
- generische HRTF
- vollautomatisch

PC-Technologie | SS 2001 | 18.214

3D-Audio: Motivation

Stereo ist Anachronismus:

- Stereoaufnahmen seit ca. 1940
- Schallplatten erlauben höchstens zwei Spuren
- Mehrspur-Magnetbänder sehr teuer
- Audio-CD "nur" stereo wegen Kompatibilität
- Kino mit Surround, aber nicht individuell

Verbesserungen:

- optimale Klangqualität inklusive Raum"staffelung"
- virtual reality, Spiele

=> 3D-Modelle der Gehörwahrnehmung notwendig

PC-Technologie | SS 2001 | 18.214

3D-Audio: Physiologie

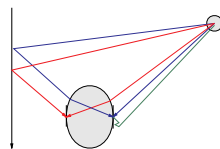
Ortung von Schallquellen:

- Lautstärkedifferenz linkes/rechtes Ohr
- Ankunftszeit linkes/rechtes Ohr
- Differenz direkter / gebrochener Schall
- Reflektion / Beugung im Außenohr
- Ortserwartungen (Hubschrauber oben vs. Hundegebell am Boden)
- Lautstärkeerwartung (tickende Uhr vs. Preßlufthammer)

=> Effekte individuell unterschiedlich

- HRTF: "Head related transfer function"
- jahrelanges Training
- Messung aufwendig: EAX/DirectSound/etc: gemittelte HRTF

=> Online-Training?! => Studien-/Diplomarbeit



PC-Technologie | SS 2001 | 18.214

3D-Audio: Literatur

Details zu 3D-Audio:

- www.sensaura.com
- AD / Sensaura Präsentation Intel Developer Forum 2000
=> tech-www

PC-Technologie | SS 2001 | 18.214

Leerseite

PC-Technologie

Linux:

- Linux OSS (open sound system), www.opensound.com
- Advanced Linux Sound Architecture, www.alsa-project.org
- low-level (Audio-) Treiber für viele Soundkarten
- rudimentäre Unterstützung weiterer Funktionen
- leider kaum gute Audiosoftware
- inhärente Latenzprobleme im Unix-Kernel (!)
- Streaming problematisch
- neue Betriebssysteme notwendig?! (BeOS, MidiShare: www.grame.fr)

```
"Because of non real-time character of a time-shared system
like Linux the driver offers a queue in the kernel
which is needed to prevent events to be scheduled too late.
This queue introduces big latency in event processing.
```

```
This [...] issue restricts building midi oriented applications
that can perform on-par with applications on Apple Macintoshes
and Atari ST's regarding real-time response." [ALSA Docs]
```

PC-Technologie | SS 2001 | 18.214

Linux: ALSA Interfaces

ALSA-Architektur:

- Audio-Treiber als Kernel-Module
 - zentrale low-level ALSA Kernel-API
 - anwenderfreundlichere ALSA Library-API
-
- Information Interface /proc/asound
 - Control Interface /dev/snd/controlCX
 - Mixer Interface /dev/snd/mixerCXDX
 - PCM Interface /dev/snd/pcmCXDX
 - Raw MIDI Interface /dev/snd/midiCXDX
 - Sequencer Interface /dev/snd/seq
 - Timer Interface /dev/snd/timer

PC-Technologie | SS 2001 | 18.214

Leerseite

PC-Technologie

Leerseite

PC-Technologie

Graphik: Agenda

- Einführung
- Monitor/Display-Technik
- Anforderungen für 2D-Graphik

- 3D-Graphik
- Polygonbasierte Modellierung
- Texturen
- aktuelle Trends

- Voxel-basierte Graphik

Graphik: Literatur

www.nvidia.com , www.ati.com , www.3dfx.com	(GeForce / Radeon / Voodoo Graphikchips)
www.videologic.com	(PowerVR)
www.madonion.com	(3DMark Benchmark)
www.microsoft.com/directx	(Microsoft DirectX homepage und download)
www.sharkyextreme.com , www.3dconcept.ch	(aktuelle Infos und Tests)
www.tomshardware.com	(diverse Tests)
www.vesa.org	(Video Electronics Standards Association)

diverse c't Artikel, u.a. 19/99-248 (Direct3D/OpenGL), zuletzt 08/2000 (GeForce Test)

c't Artikelserie "3D a la carte", 4/89 bis 9/89 (volle Renderpipeline mit Pascal-Code)

www.flipcode.com/voxtut (Tutorial Voxel-Graphik)

Foley, van Dam, Fundamentals of Interactive Computer Graphics

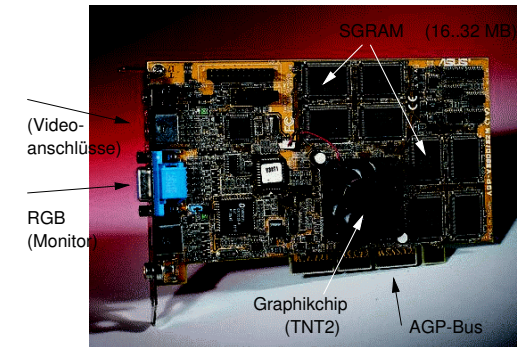
Bargen, Donnelly: Inside DirectX, Microsoft Press, 1998

Andre LaMothe: Tricks of the Windows Game Programming Gurus, Sams 1999

Alan Watt, 3D Computer Graphics, Addison-Wesley 1993

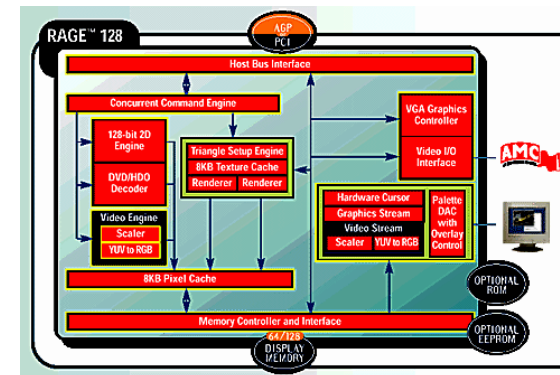


Graphik: typische Graphikkarte



- ASUS V3880 mit TNT2-Chip, 32 MByte SGRAM

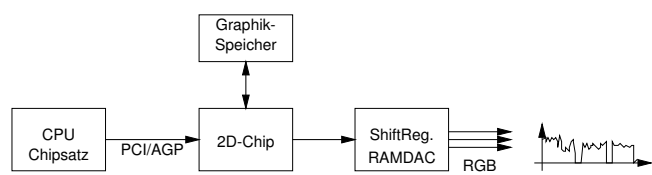
3D: Architektur ATI Rage 128



- separate Einheiten für 2D (VGA), 3D, Video
- diverse On-Chip Caches
- RGB-Ausgang zum Monitor, Video Ein/Ausgänge

[www.ati.com]

Graphik: Grundfunktionen



Bilddarstellung

- Auslesen des Bildspeichers (gewählte Auflösung / Wiederholrate)
- Color-Lookup-Table ("RAMDAC")
- serielles RGB/Videosignal mit Sync-Impulsen

2D-Funktionen

- Linien (Bresenham-Algorithmus)
- Rechtecke füllen / kopieren (BitBlit)
- Schrift

Graphik: Bresenham-Algorithmus

```

{ line slope 0 < slope < 1 }
procedure BRESENHAM( x1, y1, x2, y2, color: integer );
var dx, dy, incr1, incr2, d, x, y, xend: integer;
begin
  dx := ABS( x2-x1 );
  dy := ABS( y2-y1 );
  d := 2 * dy - dx;
  incr1 := 2 * dy;           { used for increment if d < 0 }
  incr2 := 2 * (dy-dx);     { used for increment if d >= 0 }

  if (x1 > x2) then begin
    x := x2;
    y := y2;
    xend := x1;
  else begin
    x := x1;
    y := y1;
    xend := x2;
  end;
  WRITE_PIXEL( x, y, color );      { first point on line }

  while (x < xend) do begin
    x := x + 1;
    if (d < 0) then
      d := d + incr1;
    else begin
      y := y + 1;
      d := d + incr2;
    end;
    WRITE_PIXEL( x, y, color ); { selected point near line }
  end;
end {BRESENHAM}

```

Graphik: Speicherbandbreite

Bandbreitenbedarf für 2D-Graphik:

	Auflösung	Farbtiefe	f/Hz	MB / s
CGA	24x80 (x8)	4	75	0.576
ATARI ST	640x400	1	72	2.3
EGA	480x350	4	60	5
VGA	640x480	8	75	23
XGA	1024x768	16	75	118
UGA	1600x1200	32	100	768
BX-SDRAM-100				250
RAMBUS PC800				< 1600

- ohne Koordinatentransformation
- ohne Z-Buffer, Double-Buffering, Texturen

Graphik: Speicherbandbreite 3D

Speicherbedarf bei: 1024x768 Pixel, 16-bit Farben, 30 fps

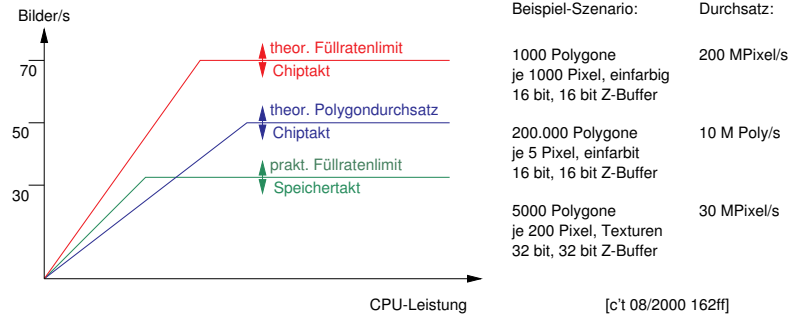
- Framebuffer: 1024x768x2 Byte 1.5 MB
- double buffering: 1.5 MB
- Z-Buffer 1.5 MB
- Rest frei für Texturen

Bandbreite, Annahme: im Mittel 3 überlappende Polygone / Pixel

- Bildwiedergabe (75 Hz): 118 MB/s
- double-buffer zeichnen (30fps) 47 .. 141 MB/s
- Z-Buffer lesen und schreiben (30fps *(3 + 1)) 188 MB/s
- Texturen (4 Zugriffe a 16 bit pro Pixel, 30 fps * 3) 1132 MB/s

=> Texturmapping ist der Engpaß ("Füllrate")

Graphik: Füllrate / Polygondurchsatz



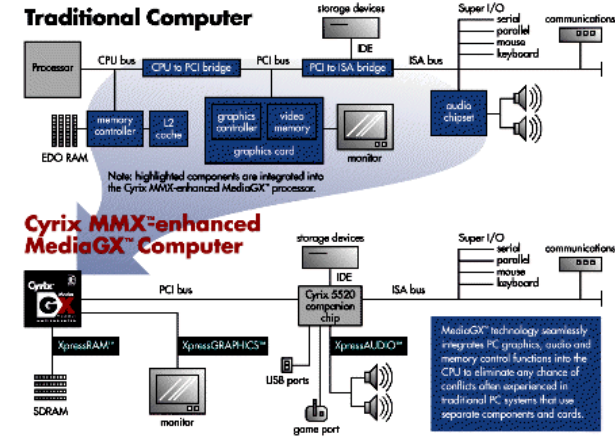
- drei zentrale Kenngrößen für Graphikchip-Bewertung
- extremer Einfluß der Textur-Algorithmen
- derzeit meistens durch Füllrate limitiert

Graphik: Unified memory architecture

UMA := gemeinsamer Haupt- und Graphikspeicher

- uraltes Prinzip (8-bit μ Ps, Atari ST, ...)
- doppelte Motivation:
 - minimale Kosten, da kein separater Graphikspeicher
 - CPU kann direkt die Graphikdaten manipulieren
 - aber Speicherbandbreite stark reduziert
 - Speicherkapazität genauso wie bei separatem Graphikspeicher
- interessant vor allem für low-cost Rechner
- Beispiele: Intel i810 Chipsatz, Cyrix MediaGX
- für aktuelle 3D-Anwendungen ungeeignet

Graphik: Cyrix MediaGX



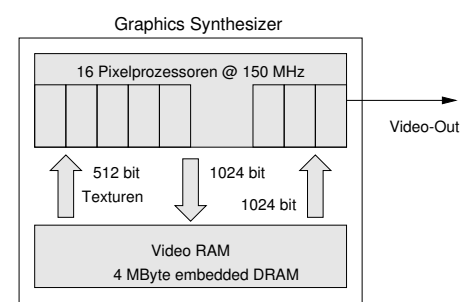
- PC mit nur drei Chips (CPU, Companion, Super IO) + DRAM

Graphik: Playstation2 Emotion Engine

UMA := gemeinsamer Haupt- und Graphikspeicher

- uraltes Prinzip (PC/AT CGA, Atari ST, ...)
- doppelte Motivation:
 - minimale Kosten, da kein separater Graphikspeicher
 - CPU kann direkt auf Graphikdaten zugreifen
 - aber Speicherbandbreite stark reduziert
 - Speicherkapazität genauso wie bei separatem Graphikspeicher
- interessant vor allem für low-cost Rechner
- Beispiele: Intel i810 Chipsatz, Cyrix MediaGX
- für aktuelle 3D-Anwendungen ungeeignet

Graphik: Playstation2-Architektur



Graphikspeicher als embedded-DRAM

- 2560 bit Speicherinterface (on chip)
- extrem hohe Bandbreite (vermutlich TB/s), Füllrate 48 GB/s
- vergleiche IRAM-Konzept
- technologiebedingt derzeit nur 4 MB Kapazität
- maximale Auflösung nur 640x480

Graphik: 3D-Transformationen

Verschieben um eine vorgegebene Länge

$$\begin{bmatrix} 1 & 0 & 0 & tx \\ 0 & 1 & 0 & ty \\ 0 & 0 & 1 & tz \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} vx \\ vy \\ vz \\ 1 \end{bmatrix} = \begin{bmatrix} vx+tx \\ vy+ty \\ vz+tz \\ 1 \end{bmatrix}$$

Vergößern um einen Faktor

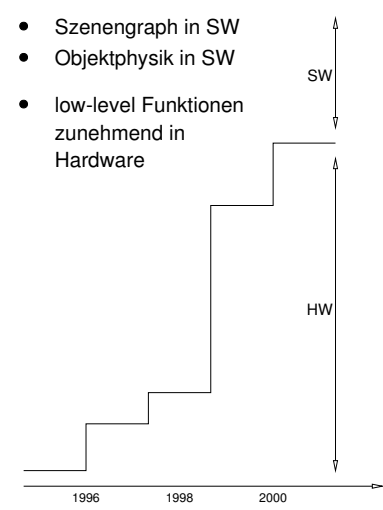
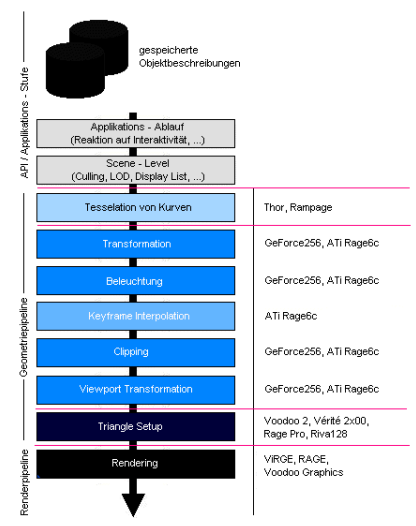
$$\begin{bmatrix} sx & 0 & 0 & 0 \\ 0 & sy & 0 & 0 \\ 0 & 0 & sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} vx \\ vy \\ vz \\ 1 \end{bmatrix} = \begin{bmatrix} sx \cdot vx \\ sy \cdot vy \\ sz \cdot vz \\ 1 \end{bmatrix}$$

Drehen um einen gegebenen Winkel

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) & 0 \\ 0 & \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} vx \\ vy \\ vz \\ 1 \end{bmatrix} = \begin{bmatrix} vx \\ \cos(\phi) \cdot vy - \sin(\phi) \cdot vz \\ \sin(\phi) \cdot vy + \cos(\phi) \cdot vz \\ 1 \end{bmatrix}$$

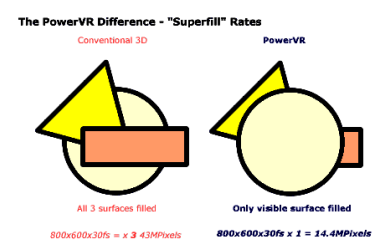
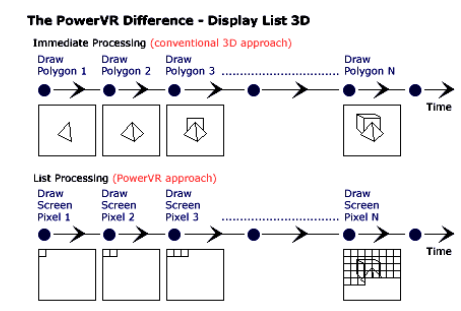
- homogene Koordinaten
- Translation
- Skalierung
- Rotation
- perspektivisch korrekt

Graphik: 3D-Pipeline



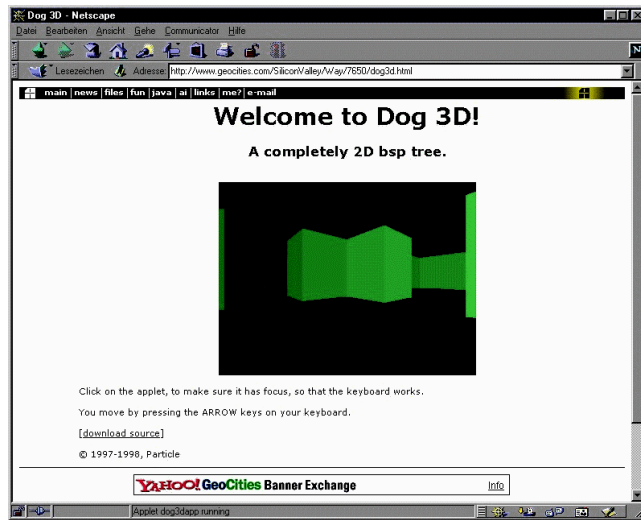
- Szenengraph in SW
- Objektphysik in SW
- low-level Funktionen zunehmend in Hardware

Graphik: PowerVR Konzept



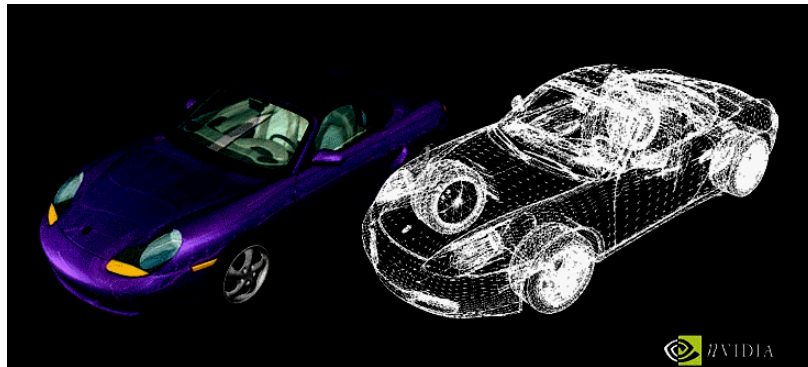
- interessantes Konzept
- erfordert angepasste Applikation
- oder Umsortieren im Treiber
- Dreamcast / SEGA "Naomi"

3D: Binary Space Partitioning



PC-Technologie | SS 2001 | 18.214

3D: Wireframe / Phong-Shading



- runde Formen erfordern sehr feines Netz
- oder bessere Modellierung (NURBS etc.)

PC-Technologie | SS 2001 | 18.214

3D: Backface-Culling

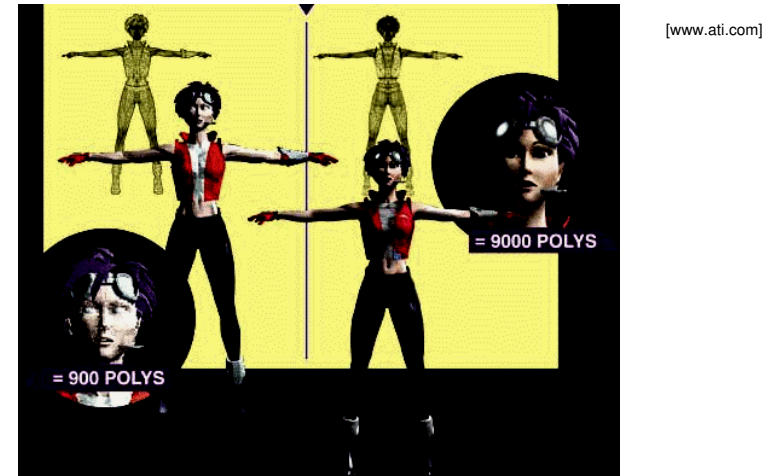


Backface-Culling: ("Rückseiten-Aussortierung")

- Flächen werden nur dargestellt, wenn Normale "nach vorne" zeigt
- Objektvorderfläche an Sichtpyramide abgeschnitten

PC-Technologie | SS 2001 | 18.214

3D: Polygonanzahl und Bildqualität



- Optimierung der Polygonanzahl ist interessantes Problem !

PC-Technologie | SS 2001 | 18.214

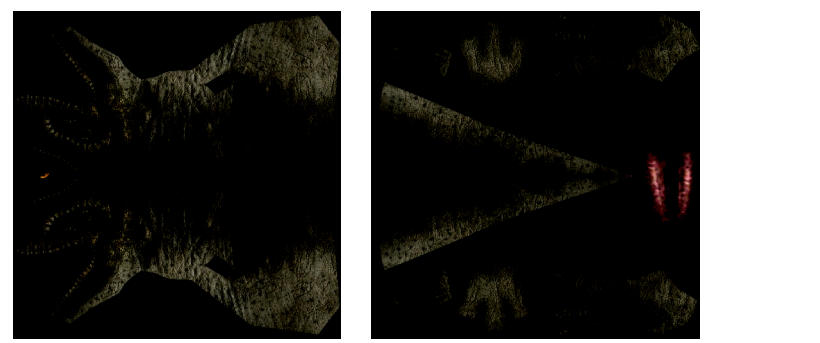
3D: Shading vs. Texturen



[c't 10/97 150]

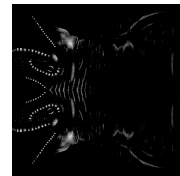
- MotoRacer mit und ohne Texturen (wegen Fehler der Graphikkarte)
- einfaches Shading nur für Plastik / Metall-Oberflächen geeignet

Texture Mapping :-)

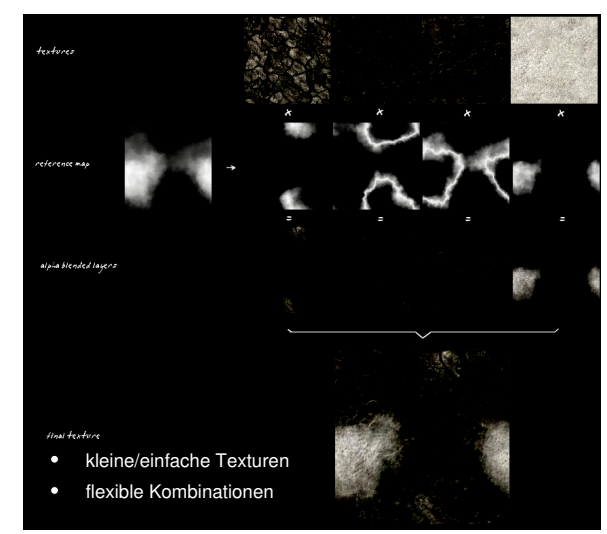


Tyrannosaurus aus Crytek X-Isle Demo

- JPEGs, jeweils 512x512x24
- separate Texturebene für Bump-Mapping



3D: überlagerte Texturen



- kleine/einfache Texturen
- flexible Kombinationen

[www.flipcode.com/voxtut]

Texturen: bilinear vs. ungefiltert



bilinear gefiltert | ohne Filter:

- wesentlich verringerte Pixelstrukturen
- Transparenz über Alpha-Blending:
- keine Glättung der Objektkanten:

vierfacher Rechenaufwand
 Fenster auch geglättet
 => Antialiasing notwendig

3D: Clipping-Fehler



[Tomb Raider 2, 640x480]

- zu geringe Genauigkeit der Kollisionserkennung / des Z-Buffers
- bekanntes Problem in TR2
- allgemein: "binary space partitioning" vs. transparente Objekte

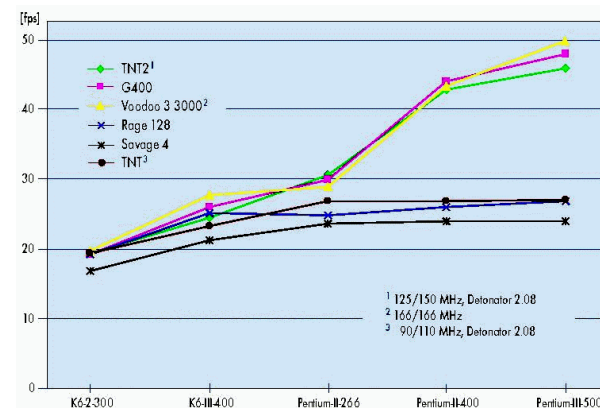
Graphik: Karten-Vergleich

Grafikkarten mit anderen Chips unter Direct3D 6.1 und OpenGL										
	3D Max 99 MAX, Racing		3D Max 99 MAX, 1st Person		Unreal 2.25 ¹		Expendable, 1024/16		Expendable, 1280/16	
	besten		besten		besten		besten		besten	
Siemens III S540	33	31	20	20	24	16	20	20	30	30
Winner II	33	31	20	19	13	17	17	17	30	30
Besatz 99	25	23	18	21	14	17	17	17	26	26
SR9	34	33	18	21	13	17	17	17	31	31
Rage 128	32	33	25	26	18	24	24	24	31	31
Millennium G400	34	36	27	26	37	41	41	41	32	32
Voodoo 3 3000	34	34	44	48	44	45	45	45	37	37
	47	36	48 ²	44	41	43	43	43	43	43
		45	32 ²	44	45	47	47	47	47	47

Alle Tests durchgeführt unter Windows 98 und Direct 6.1, sofern nicht anders angegeben bei 1024 x 768, 16 Bit, 75 Hz, Bildwiederholrate, "Wait for VSync", off. Audio ein, alle Werte in Frames pro Sekunde (fps).
 1024: 1024 x 768, 1280: 1280 x 1024, 16: HiColor 16 Bit, 32: TrueColor 32 Bit.
 Pentium II, 400 MHz Pentium III, 500 MHz ¹ Direct3D, shiny surfaces ein ² via GlideAPI

- Benchmark-Daten für Savage / Rage 128 / G400 / Voodoo 3
- Karten mit Riva TNT/TNT2 ähnlich
- alle Spiele in 1024x768x16 spielbar
- Voodoo3 Glide-API performanter als DirectX 6.1

Graphik: Karten-CPU-Vergleich



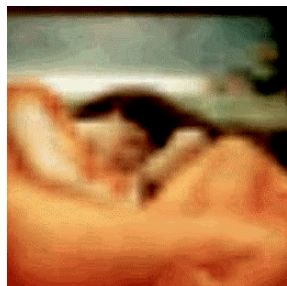
- Limitierung durch Füllrate / Polygondurchsatz gut sichtbar
- FPU des Pentium-III deutlich besser als K6
- Athlon/Duron etwa gleichwertig zum PIII

Trends: Fotorealismus

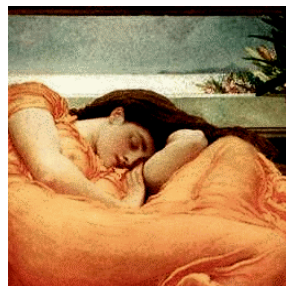


- viele Polygone, gute Texturen, Schatten, . . .

Trends: Texturkompression



256x256x16 Pixel = 128 KB



2048x2048x16 Pixel = 8 MB

hochauflösende Texturen wichtig für:

- Hintergrund / Himmel / usw.
- Nahansichten
- Kompression notwendig, um Kapazität/Bandbreite zu sparen
- diverse Verfahren, u.a. DirectX7

Trends: Bump mapping



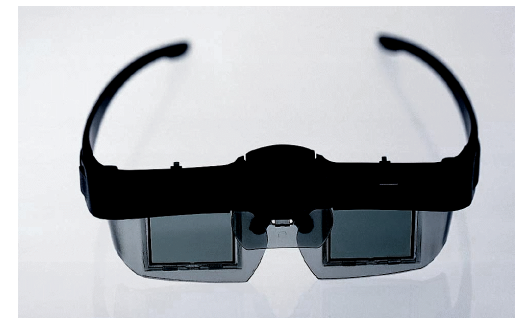
normal



bump mapping

- Überlagerung mehrerer Texturen
- für "rauhe" Oberflächen
- statt aufwendiger Polygonmodellierung

Trends: Stereo



Shutterbrille:

- abwechselndes Abdunkeln der Augen
- Anzeige des linken / rechten Stereobildes
- per Treiber: ohne Modifikation der Applikation
- erfordert doppelte Framerate

Trends: 3dfx FSAA-Demo



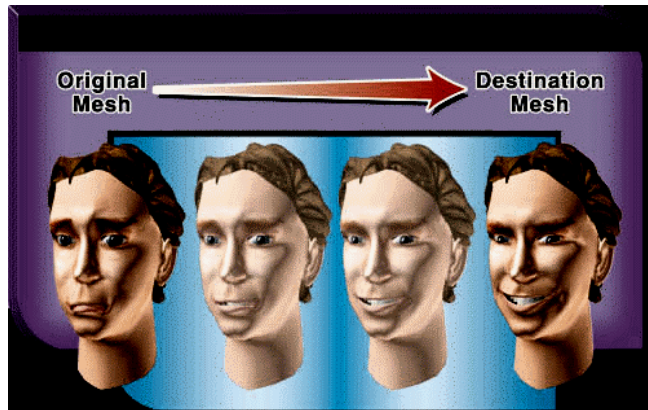
Voodoo4 full screen antialiasing eingeschaltet (4X)

(ohne)

deutlich bessere Bildqualität:

- glatte Objektkanten
- weniger Flimmern
- aber Schärfeverlust

Trends: "Keyframe-Interpolation"



- Interpolation zwischen Keyframes
- direkt auf der Graphikkarte
- für flüssigere Animationen

[www.ati.com]

PC-Technologie | SS 2001 | 18.214

Trends: Max Payne "bullet cam"

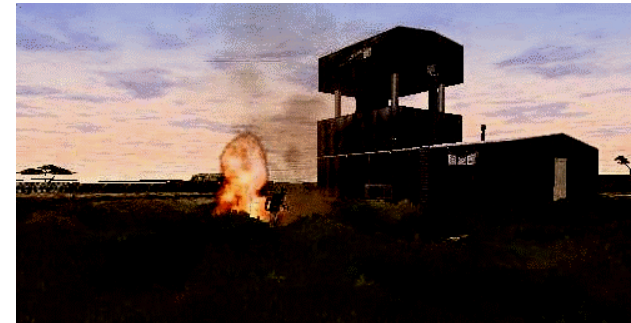


[Max Payne Demo]
[www.pcgames.de]

- Kinoeffekte in Echtzeit
- Zeitlupe, Replay, ...
- Problematik Gewaltverherrlichung ?!

PC-Technologie | SS 2001 | 18.214

Voxel: Beispiel DeltaForce2



- Voxel-basierte Modellierung der Umgebung
- ermöglicht kleine Objekte (=> Gras, Rauch)
- Häuser, Darsteller als Polygonmodelle

[www.deltaforce2hq.com]

PC-Technologie | SS 2001 | 18.214

Voxel: Beispiel DeltaForce2



- weitläufigere Landschaften als mit Polygonmodellierung

[www.deltaforce2hq.com]

PC-Technologie | SS 2001 | 18.214

DirectX: Agenda

- Einführung und Konzepte
- Vorführung DirectX7 SDK

PC-Technologie | SS 2001 | 18.214

DirectX: Literatur

www.microsoft.com/directx (DirectX portal page)
www.microsoft.com/msdn (MS developer network)
 DirectX SDK (Docs und Beispiele)

Bargen, Donnelly: Inside DirectX, Microsoft Press, 1998

Root, Boer, DirectX complete, McGraw-Hill, 1999

Sirotn, Debeloff, Urri: DirectX-Programmierung mit Visual C++, Addison-Wesley, 1999

Petzold, Programming Windows95, Microsoft Press, 1996

Stroustrup, C++ Programming Language, Addison-Wesley, 1997

c't Artikelserie zu Delphi mit DirectX, 1999 bis 2000

PC-Technologie | SS 2001 | 18.214

DirectX: Übersicht

Direct<X>: Multimedia-Schnittstellen für Windows

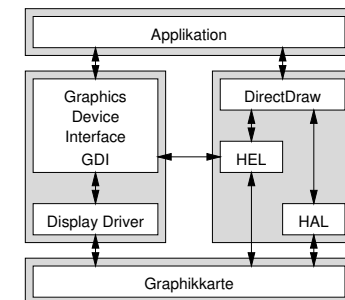
- DirectDraw high-performance 2D-Graphik
- Direct3D 3D-Graphik
- DirectShow Videowiedergabe
- DirectSound grundlegende Audiofunktionen (3D)
- DirectMusic Software-Synthesizer
- DirectInput schnelle Treiber für Tastatur/Maus/Joystick/...
- DirectSetup einfache und sichere Installation
- DirectPlay Multiplayer / Internet
- was fehlt? Sprachein-/ausgabe
...

PC-Technologie | SS 2001 | 18.214

DirectX: DirectDraw

DirectDraw: Grundfunktionen für 2D-Graphik

- direkte (low-level) Verwaltung des Bildspeichers
- front buffer / back buffer / surface flipping
- hardware overlays
- bit-blitting
- kein Ersatz für das GDI:
Linien, Fonts, etc. per GDI
- HEL: Hardware Emulation Layer
- HAL: Hardware Abstraction Layer
- DDI: GDI Device Driver Interface



PC-Technologie | SS 2001 | 18.214

DirectX: Direct3D

"world-class game and interactive 3D-graphics
on a computer running Microsoft windows"

[DirectX Overview]

- geräteunabhängiger Zugriff auf 3D-Hardware
- nutzt Hardwarebeschleunigung, soweit vorhanden
- oder Softwareemulation

- Transformation, Clipping
- Z-Buffer / W-Buffer
- Rendering mit Flat- / Gouraud-Shading
- Texturen, Mip-Mapping
- diverse Lichtquellen

- Funktionsumfang vergleichbar mit OpenGL (seit DirectX7)

PC-Technologie | SS 2001 | 18.214

DirectX: DirectShow

"streaming media architecture"

- Wiedergabe und Aufnahme von Datenströmen
- Formatkonvertierung
- komprimierte Audio/Videodaten
- verwendet Hardwarefunktionen, soweit vorhanden

- WAV, MP3, ...
- MPEG, AVI, ...

- basiert auf WDM-Treibern
- aber auch Unterstützung von "legacy"-Treibern (video for windows)

PC-Technologie | SS 2001 | 18.214

DirectX: DirectSound

Audiowiedergabe:

- flexibler Audiomixer
- nutzt Audiohardware soweit möglich, sonst Software
- schnell, geringe Latenz
- Sampleraten-/Formatkonvertierung

- 3D-Audiofunktionen
- Position und Richtung von Hörer und Schallquellen
- Doppler-Effekt, entfernungsabhängige Dämpfung
- HRTF-Funktionen

- siehe Audio-Folien

PC-Technologie | SS 2001 | 18.214

DirectX: DirectMusic

"message-based musical data"

- Noten, Akkorde, Phrasierung, ...
- MIDI-Schnittstelle
- Klangerzeugung über Software-Synthesizer
- und Wiedergabe über DirectSound
- oder Ansteuerung externer Synthesizer

- DLS (downloadable sounds / SoundFont2)
- DirectMusic Producer

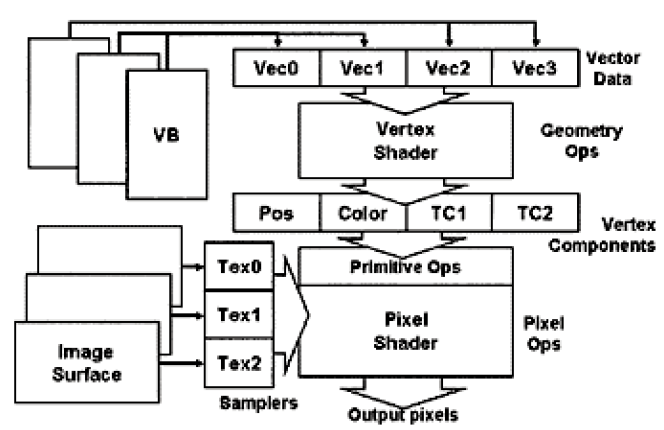
- Hardwareunterstützung nur in Windows98/2000

PC-Technologie | SS 2001 | 18.214

DirectX: DirectInput

- Treiberschicht zu Eingabe-Geräten
- Treiber für sehr viele Geräte
- umgeht die normalen Windows-Treiber
- minimale Latenz
- Unterstützung von force-feedback Geräten
- diverse Kraft

DirectX8: Graphics



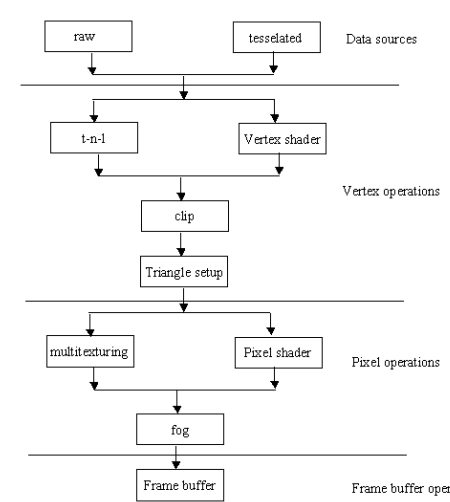
- kombiniert bisherige APIs: DirectDraw und Direct3D
- interne Programmiersprachen für "Vertex/Pixel-Shader"

DirectX 8: VertexShader



- morphing / twining animation
- matrix palette skinning
- user-defined lighting models
- general environment mapping
- procedural geometry
- developer defined algorithms

DirectX 8: Graphik-Pipeline

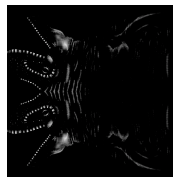


- kompatibel mit DirectX 7
- neue Fkt. nur auf "neuer" HW

DirectX 8: PixelShader

PixelShader: Programmiersprache für Pixel-Ops:

- direkter Zugriff auf die Graphik-Hardware,
- insb. für Textur-Operationen:
- beliebige Textur-Operationen
- per-pixel lighting
- bump-mapping
- per-pixel environment mapping
- other developer-defined algorithms



DirectX 8: VertexShader Beispiel...

- Folie noch nicht fertig, siehe MS DX8 Overview (PowerPoint)

DirectX 8: Multitexturing

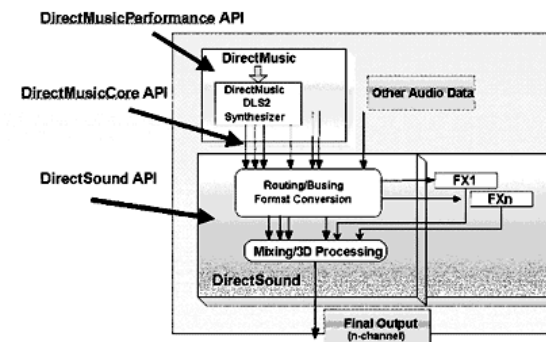


- Überlagerung mehrerer Texturen
- z.B. für light-maps (siehe oben)
- oder bump-mapping (rechts)

Ati Radeon: 2 Pipes, je 3 Texturen
 Geforce 2: 4 Pipes, je 2 Texturen



DirectX 8: Audio

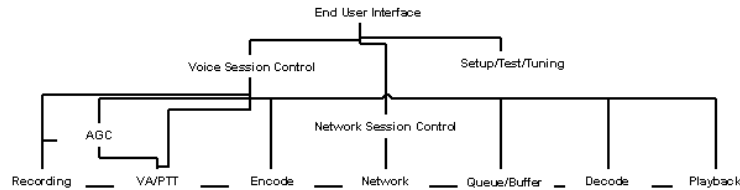


Integration von DirectSound und DirectMusic:

- 3D-Audio für alle Quellen (.wav, .asf, DLS, ...)
- DLS2 SW-Synthesizer (6-stage envelopes, effects, LFOs, ...)

DirectX 8: Voice Communication

DirectPlayVoice: Echtzeit-Sprachkommunikation in DirectPlay



- Verwaltung über den DirectPlay Server
- automatische Verwaltung jeder DirectPlay Voice Session
- automatische Kalibrierung der Bandbreite/Latenz

DirectX: COM

Component Object Model:

- Microsoft's Komponentenmodell
- entwickelt für OLE (object linking and embedding)
- Interaktion zwischen Software-"Objekten" (Komponenten)
- über Schnittstellen (Interfaces)
- als Tabelle mit Zeigern auf C-Funktionen
- Interface-Hierarchie, IUnknown
- Schnittstelle wird nach Definition nie mehr verändert (GUID)
- statt dessen Einführung neuer Schnittstellen
- Speicherverwaltung mit Referenzzählung

```
QueryInterface() / AddRef() / Release()
```

DirectX: vtable

COM-Funktionsaufrufe:

- vtable: Tabelle mit Funktionspointern
- COM-Referenz ist Pointer auf eine vtable
"long pointer to Direct Draw interface"

```
lpDD -> lpVtbl ->Release( lpDD );      /* C */
lpDD -> Release();                      // C++
```

- keine Ausnahmebehandlung
- Fehlercodes jeder Funktion prüfen!

```
if (FAILED( lpDDSBBack->GetDC( &hdc )) { ... }
if (SUCCEEDED( lpDDSBBack->GetDC( &hdc )) { ... }
```

SDL: "Simple DirectMedia Layer"



(www.libsdl.org)

Leerseite

SDL: Beispiel

```

/* Print out all the keysyms we have, just to verify them */

#include <stdio.h>
#include <ctype.h>

#include "SDL.h"

int main(int argc, char *argv[])
{
    SDLKey key;

    if ( SDL_Init(SDL_INIT_VIDEO) < 0 ) {
        fprintf(stderr, "Couldn't initialize SDL: %s\n",
            SDL_GetError());
        exit(1);
    }
    for ( key=SDLK_FIRST; key<SDLK_LAST; ++key ) {
        printf("Key #%d, \"%s\"\n", key, SDL_GetKeyName(key));
    }
    SDL_Quit();
    return(0);
}

```

Leerseite

Mobile PCs: Agenda

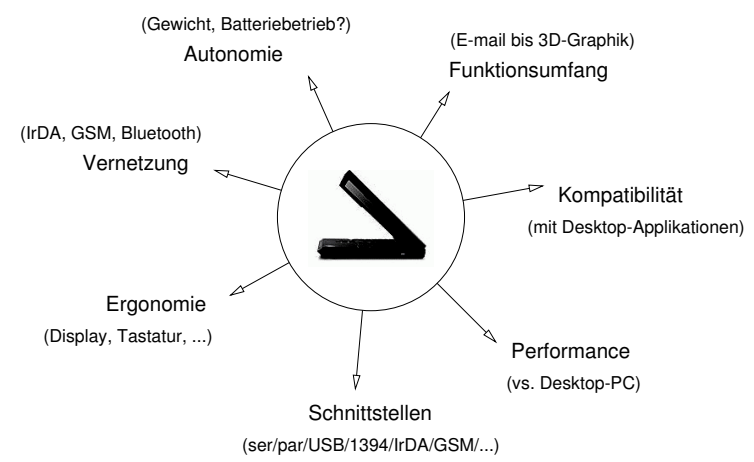
- Einführung
- Geräteklassen
- CMOS: Stromverbrauch
- Power-Management
- Displays: LCD .. eInk
- drahtlose Netzwerke:
- IrDA
- GPRS, Bluetooth & Co.



Mobile PCs: Literatur

developer.intel.com/mobile	(Intel "mobile processors", Power-Management)
www.transmeta.com	(Crusoe Prozessor und Whitepaper)
www.irda.org	(Infrarot Datenkommunikation)
www.cs.uit.no/linux-irda/	(Linux IrDA project homepage)
www.microsoft.com/ddk	(Device driver kit homepage)
www.bluetooth.com	(Bluetooth Konsortium)
www.dafu.de	(deutsche Datenfunk-Seite)
diverse c't Artikel und Tests, u.a.	
ct 1999/06/218-255	(Übersicht und Tests Funknetzwerke)
ct 2000/07/110-141	(Übersicht und Tests PDA und Subnotebooks)
ct 1999/25/114ff	(IrDA)
www.research.ibm.com/journal/	(IBM Journals R&D / system seit kurzem online!)

Mobile PCs: Anforderungen



- riesiges Gerätespektrum möglich!

Mobile PCs: Geräteklassen

- | | |
|--|--|
| <ul style="list-style-type: none"> • Portable ("Schlepptopp") • Notebook • Sub-Notebook • Ebook / Webpad • PDA • Smartphones | <p>PC im Kompaktgehäuse, PCI-Stots, ...
 vollwertiger PC, ca. 3kg, ca. 2h
 reduzierter PC (Windows CE?), < 2kg, 1-3h
 neue Geräteklasse, keine Tastatur
 Palm/Psion/CE&Co: mobil, aber keine PCs
 Nokia 9110, Ericsson RA380, ...</p> |
|--|--|



(1990)



(1998)

Mobile PCs: PC '98 / PC '99

Mobile Platform '98:

- Pentium-II Prozessor
- AGP 1X Graphik
- USB-Ports
- 1X DVD-ROM
- MPEG2-Wiedergabe
- mit AC3 Audio
- Anschluss für Digitalkamera

Mobile Platform '99:

- Pentium-II/III Prozessor
- AGP 2X Graphik
- mit 3D-Funktionen
- "Device Bay"
- 2X DVD-ROM
- MPEG2-Wiedergabe
- MPEG1-Encoding
- IEEE 1394 Anschluss

(developer.intel.com/mobile)

PC-Technologie | SS 2001 | 18.214

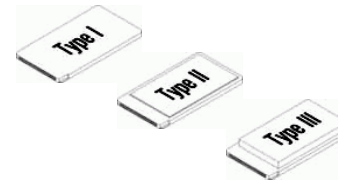
Mobile PCs: Komponenten

Notebook := "geschrumpfter" PC:

- "mobiler" Prozessor: Takt / Vdd reduziert
- normaler Chipsatz plus IO-Chip: AGP/PCI, USB, ser/par/...
- SDRAM (SO-DIMM)
- AGP/PCI Graphikcontroller, 2D/3D/Video, LCD-Treiber
- IDE Festplatte (2.5" Formfaktor, derzeit 6-30 GB)
- IDE CDROM / DVD Laufwerk
- PC-Card Interface (PCMCIA/Cardbus)
- USB / 1394 / legacy ports
- Handys, PDAs, usw: systemspezifische Hardware

PC-Technologie | SS 2001 | 18.214

PCMCIA

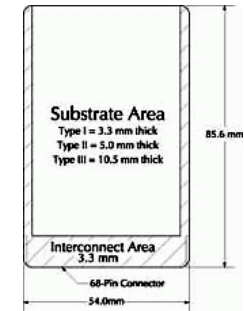


"Personal Computer Memory Card Intl. Association"

- Norm für Speicher- / Erweiterungskarten
- 68 Pins, 16-bit Daten, einfaches Protokoll
- "Card Information Structure" für PnP
- hot-swap

Cardbus:

- Größe und Stecker wie PCMCIA
- aber elektrisch vollständig anders
- weitgehend PCI-Bus kompatibel (32-bit)



(www.pcmcia.org)

PC-Technologie | SS 2001 | 18.214

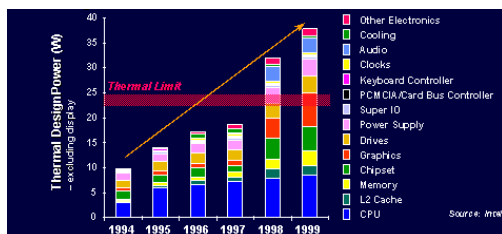
Speichermedien...



- MicroDrive / CompactFlash / SmartMediaCard / MemoryStick
- jeweils mit PCMCIA-Adapter
- völlig unterschiedlicher Konzepte
- "nackter" Speicher ... bis hin zum "magic gate" memorystick

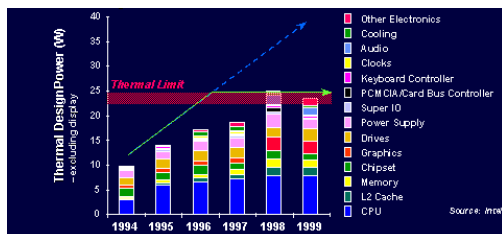
PC-Technologie | SS 2001 | 18.214

Leistungsverbrauch: Intel Roadmap



Stromverbrauch von Notebook-Komponenten [Intel IDF 98]

- thermisches Limit 25 W
- Batteriegewicht vs. Laufzeit
- Daten ohne Display, ca. 5..10 W zusätzlich
- Powermanagement
- bessere Batterien?



(developer.intel.com/mobile)

Powermanagement

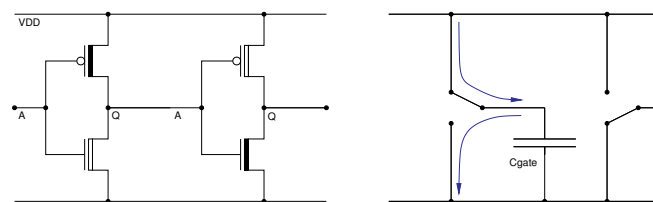
einige Schlagworte und Zeitpunkte:

- 1989 386SL: Abschalten von Funktionseinheiten
- 1991 Advanced Power Management (APM) Spez.
- 1995 Voltage Reduction Technologie (split voltage P5)
- 1997 Advanced Configuration and Power Interface (ACPI)
- 2000 Intel SpeedStep - zwei Stufen, z.B. 700/550 MHz
Transmeta "LongRun" - mehrere Stufen

- bisher nur unzureichend umgesetzt
- noch beträchtliches Einsparpotential

(Übersicht in c't 2000/07/216ff)

Leistungsverbrauch: CMOS



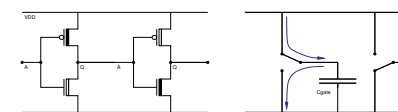
- (fast) kein statischer Stromverbrauch
- Kurzschluss-Strom beim Umschalten
- Umladen der Gate-Kapazität dominiert, also:

$$Q = C \cdot U$$

$$I = dQ/dt \sim f \cdot C_{gate} \cdot U$$

$$P = U \cdot I \sim f \cdot U^2$$

Leistungsverbrauch: CMOS



- kein Ruhestrom
- Stromaufnahme $P \sim f \cdot U^2$

=> Taktfrequenz reduzieren (Rechenleistung sinkt)
=> unbenutzte Rechenwerke abschalten (Powermanagement)

=> Versorgungsspannung reduzieren

- Rücksicht auf Systemumgebung (split voltage μ Ps)
- erfordert angepasste Si-Technologie (0.35 -> 0.25 -> ...)
- Signallaufzeiten spannungsabhängig
- reduzierte Betriebsfrequenz

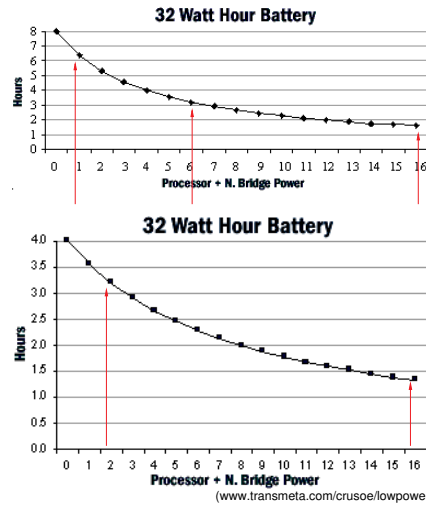
- Intel "SpeedStep", Transmeta "LongRun", AMD "PowerNow"

Leistungsverbrauch: Transmeta LongRun

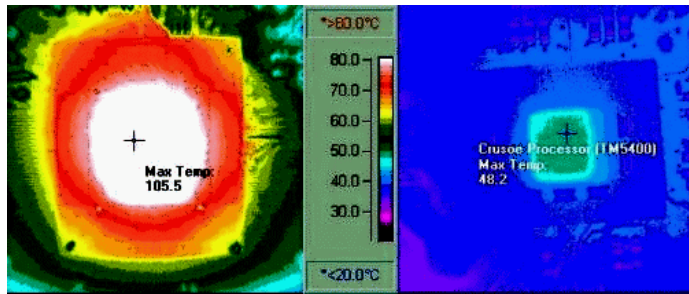
- "all day mobile computer" (Annahme: 4W System)
Prozessor plus Chipsatz:
- Crusoe: 8mW .. 1W (typ.)
- normal: 6 .. 16 W



- "mobile multimedia PC" (Annahme: 8W System)
- Crusoe: ~ 2W für DVD-dec.



Leistungsverbrauch: Transmeta LongRun



Pentium-II / K6 ?!

TMS 5400

- Temperaturprofil während DVD-Softwaredekodierung
- TM5400 braucht keinen Prozessorlüfter . . .
- und integriert die "northbridge" (Speicher/PCI-Interface)
- aber Marktchance?! (Performance vs. Laufzeit)

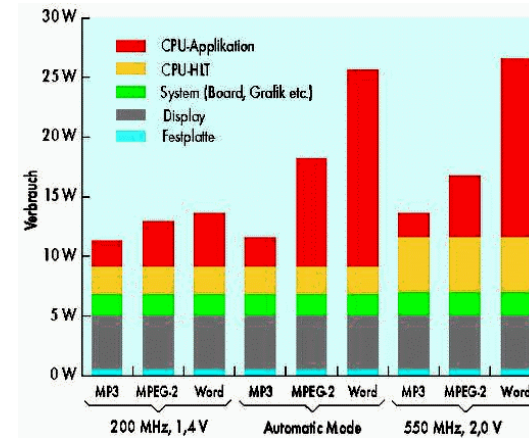
S3/Transmeta: WebPad

"designed for web surfing, portable multimedia, electronic book applications, streaming audio, video, and productivity"

- Crusoe Prozessor
- "text book size" (kleiner A4)
- 10.4" LCD
- Festplatte
- USB
- 8-10 h Laufzeit mit Batterien
- mobile Linux

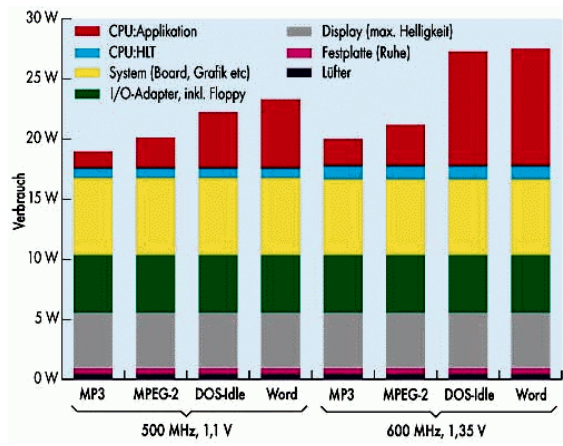


AMD: PowerNow!



- Stromverbrauch für K6-2+, 200..550 MHz Takt (0.25µm CMOS)
- Meßwerte im System, inkl. Spannungsregler&Co.

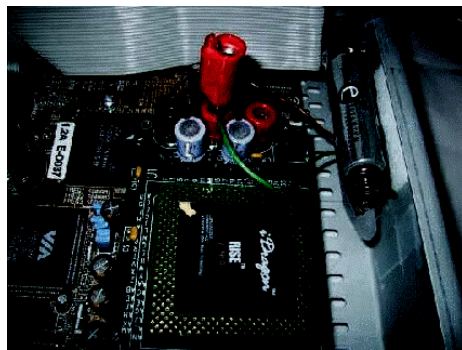
Intel: SpeedStep ...



- "mobile Pentium-III bei 500 / 600 MHz"
- interessantes Verhältnis: CPU - Chipsatz - I/O - Display

(c't 15/2000 p.026)

Rise: Dragon CPU



low-power x86 Prozessor:

- DVD-Dekodierung im Batteriebetrieb (eine Mignonzelle)
- Rest des Systems läuft mit Netzteil...

(tomshardware.com, Microprocessor Forum, 13/10/2000)

Crusoe: LongRun..., Benchmarks

Anwendungsleistung	BAFCo SYSmark 2000	ForView 3.1 chess2, new	3DMark 2000 CPU Marks
Crusoe IM5600			
300 MHz	31	124	33
600 MHz	50	257	56
LongRun	50	256	56
Pentium III			
500 MHz	86	347	78
600 MHz	92	417	81
AMD K6-2+			
200 MHz	41	[nicht gemessen]	[nicht gemessen]
550 MHz	77	[nicht gemessen]	[nicht gemessen]
PowerNow!	75	[nicht gemessen]	[nicht gemessen]

Verbrauch	Milliwatt Leistungsaufnahme (zum Beispiel PowerPoint, Excel, MPEG) [Watt]
Crusoe IM5600¹	
300 MHz, 1,2 V	2,8
600 MHz, 1,6 V	3,5
LongRun	3,2
Pentium III²	
500 MHz, 1,1 V	5,4
600 MHz, 1,35 V	9,75
AMD K6-2+	
200 MHz, 1,4 V	2,5
550 MHz, 2,0 V	14

Crusoe TMS5600, 600MHz:

- VLIW Prozessorkern
- "Code-Morphing" (=JIT)
- integrierte Northbridge
- (deutlich) schlechtere Performance
- bei geringerer Verlustleistung

(Sony Vaio CR1E, c't 22/2000, 112)

ATI, S3: mobile graphics...

Benchmarks	Expendable (800 x 600, 16 Bit) [fps]	Quake 2, Demo1 (1024 x 768, 16 Bit) [fps]	Quake 2, Demo 1 (800 x 600, 16 Bit) [fps]	Flight Simulator 98 [fps]	Transfer rate Video Speicher ² [MByte/s]	AGP-Texturspeicher ³ [MByte/s]	Spielzeit in Akku-betrieb ⁴ [Stunden]
ATI Rage Mobility (Fujiitsu Siemens LifeBook E-6530)	17	0	19	25	43	31	0,6 ⁵
S3 Savage/MX (Toshiba Tecra 8100)	36 ¹	20	29	40	134	47	1,7 ⁶
Nvidia GeForce256 (Dell Inspiron 1100)	37	118	138	90	—	33	—

- "einfache" Graphikkchips zu langsam für 3D-Spiele

Stand 06/2001:

- "Mobile"-Graphikkchips mit 16 .. 32 MByte Bildspeicher
- konkurrenzfähige Graphikperformance
- Stromverbrauch ähnlich wie CPUs ...
- aber "langsame" Displays

(c't 05/2000, p.144)

ACPI:

"Advanced Configuration and Power Interface"

- Intel / Microsoft Spezifikation für Power-Management
- umgesetzt in Windows 98/ME und 2000
- diverse Schlaf- / Stromsparszustände / Throttling
- z.B. "hibernate" (suspend-to-memory, suspend-to-disk)
- Hot-Swapping von Komponenten
- ACPI-Betriebssystem verwaltet alle HW-Komponenten direkt
- funktioniert nur, wenn alle Geräte und Treiber mitspielen
- bisher noch reichlich Probleme
- "embedded controller" Interface
- ASL/AML ("ACPI source/machine language") zur Beschreibung

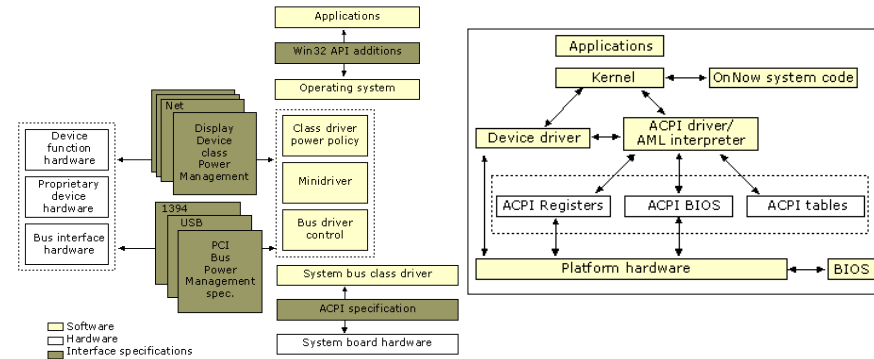
PC-Technologie | SS 2001 | 18.214

ACPI: Win2K GUI



PC-Technologie | SS 2001 | 18.214

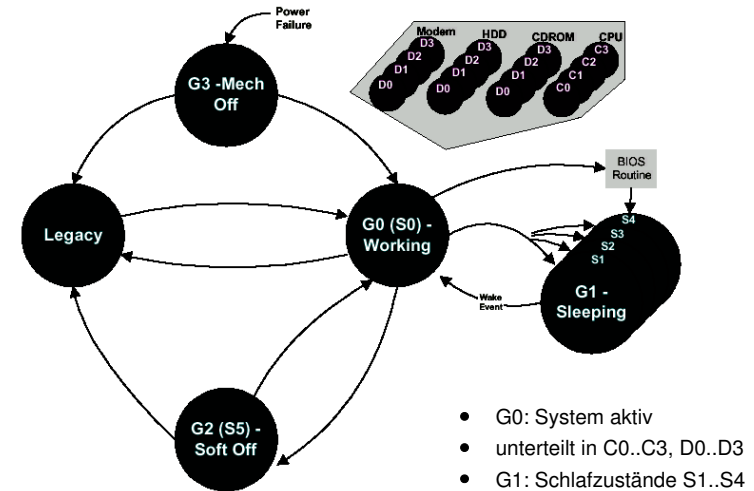
ACPI: Architektur



- Windows enthält Treiber für die Busse PCI/USB/1394/...
- und "generische" Treiber für wichtige Geräte

PC-Technologie | SS 2001 | 18.214

ACPI: Zustände



- G0: System aktiv
- unterteilt in C0..C3, D0..D3
- G1: Schlafzustände S1..S4

Figure 3-1 Global System Power States and Transitions

(ACPI spec.)

PC-Technologie | SS 2001 | 18.214

ACPI: Schlaf-Zustände

Sleeping states (Sx states) are types of sleeping states within the global sleeping state, G1. The Sx states are briefly defined below. For a detailed definition of the system behavior within each Sx state, see section 7.5.2. For a detailed definition of the transitions between each of the Sx states, see section 9.1.

S1 Sleeping State:
The S1 sleeping state is a low wake-up latency sleeping state. In this state, no system context is lost (CPU or chip set) and hardware maintains all system context.

S2 Sleeping State:
The S2 sleeping state is a low wake-up latency sleeping state. This state is similar to the S1 sleeping state except the CPU and system cache context is lost (the OS is responsible for maintaining the caches and CPU context). Control starts from the processor's reset vector after the wake-up event.

S3 Sleeping State:
The S3 sleeping state is a low wake-up latency sleeping state where all system context is lost except system memory. CPU, cache, and chip set context are lost in this state. Hardware maintains memory context and restores some CPU and L2 configuration context. Control starts from the processor's reset vector after the wake-up event.

S4 Sleeping State:
The S4 sleeping state is the lowest power, longest wake-up latency sleeping state supported by ACPI. In order to reduce power to a minimum, it is assumed that the hardware platform has powered off all devices. Platform context is maintained.

S5 Soft Off State:
The S5 state is similar to the S4 state except the OS does not save any context nor enable any devices to wake the system. The system is in the "soft" off state and requires a complete boot when awakened. Software uses a different state value to distinguish between the S5 state and the S4 state to allow for initial boot operations within the BIOS to distinguish whether or not the boot is going to wake from a saved memory image.

ACPI: Beispiel Modem

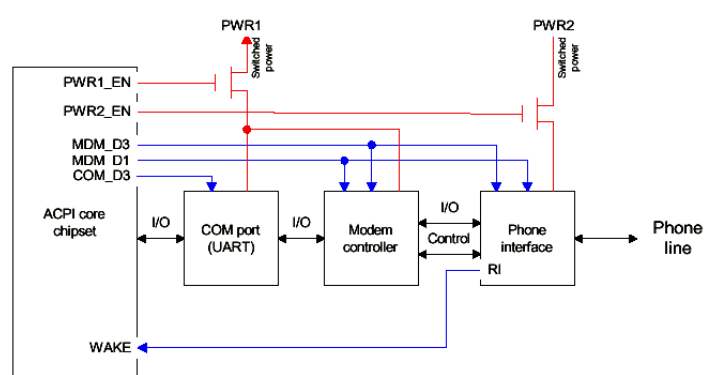


Figure 3-2 Example Modem and COM Port Hardware

- Hardware wird via Chipsatz ein-/ausgeschaltet
- INT-Eingang für wake-on-ring
- entsprechend für andere I/O-Komponenten

ACPI: Modem -Zustände

To illustrate how these power management methods function in ACPI, consider an integrated modem. (This example is greatly simplified for the purposes of this discussion). The power states of a modem are defined as follows (this is an excerpt from the Modem Device Class Power Management Specification):

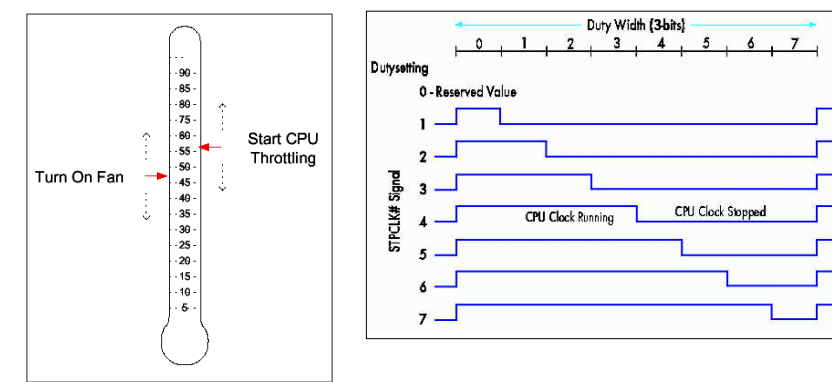
- D0 - Modem controller on
 - Phone interface on
 - Speaker on
 - Can be on hook or off hook
 - Can be waiting for answer
- D1 - Modem controller in low power mode (context retained by device)
 - Phone interface powered by phone line or in low power mode
 - Speaker off
 - Must be on hook
- D2 - Same as D3
- D3 - Modem controller off (context lost)
 - Phone interface powered by phone line or off
 - Speaker off
 - On hook

The power policy for the modem are defined as follows:

- D3 → D0 COM port opened
- D0,D1 → D3 COM port closed
- D0 → D1 Modem put in answer mode
- D1 → D0 Application requests dial or the phone rings while the modem is in answer mode

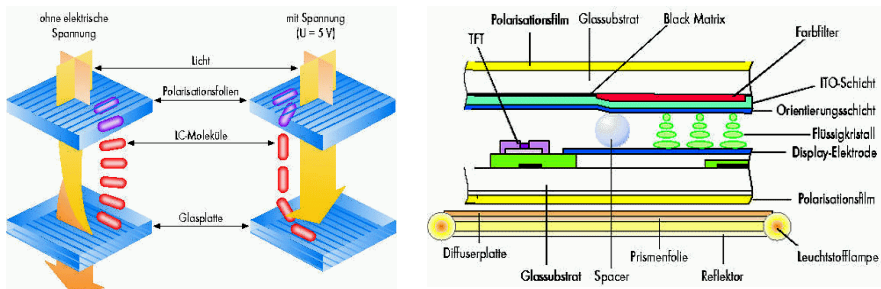
The wakeup policy for the modem is very simple: when the phone rings and wakeup is enabled, wake the machine.

ACPI: CPU Throttling



- Temperaturmessung der CPU (des Systems)
- automatische Regelung des Lüfters
- und Heruntertakten (1:1 bis 1:8) der CPU

LC-Displays:



- Flüssigkristalle zwischen zwei Polarisationsfiltern
- Matrixsteuerung (passiv oder TFT), Farbfilter
- Hintergrundbeleuchtung
- aufwendige Herstellung, geringer Yield: teuer
- geringe Effizienz

LCD: Display-Technologie vs. Auge

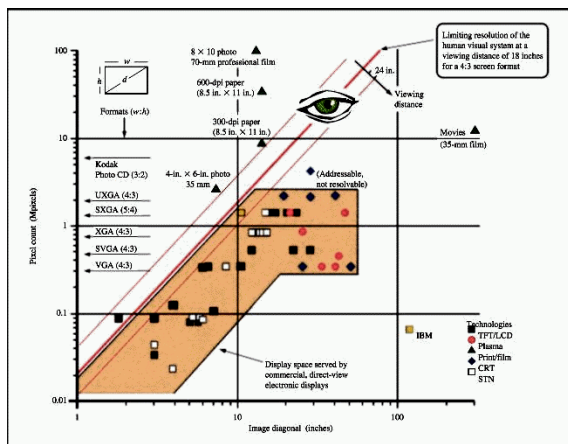


Figure 1
A view of display space showing existing electronic, paper, and film displays.

LCD: Evolution

- 1992: VGA (640x480)
- 1998: SXGA (1280x1024)
- monochrom bis true color
- 8" bis 15" Bild diagonale
- Auflösung bis 200 dpi (IBM SXGA Prototyp)
- Qualität wie Laserdrucker / Zeitschriften-Farbdruck
- weitere Steigerung nötig?

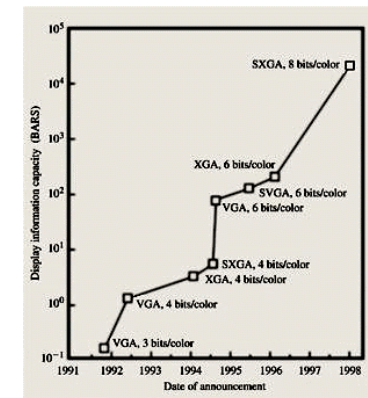


Figure 2

Information capacity increase of IBM TFT/LC displays with time. Information capacity is in units of pixel count times number of colors (billions of addressable retinal stimuli, or BARS).

LCD: 200dpi Prototyp



gedrucktes Cover des JR&D

200dpi LCD Prototyp



- vergleichbare Auflösung
- besserer Kontrast
- LCD subjektiv besser

LCD: 200dpi Prototyp

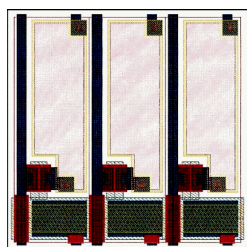


Figure 2
CAD layout of three color subpixels of dimensions $58 \mu\text{m} \times 154 \mu\text{m}$, arranged in a vertically striped mode.

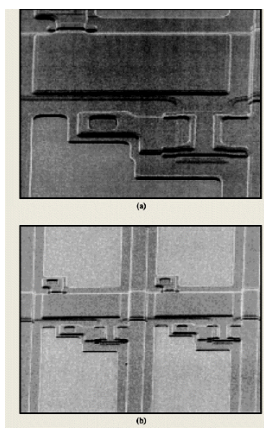
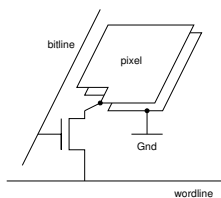


Figure 6
Oblique-angle scanning electron micrographs of the array depicted in Figure 5. The subpixels are $54 \mu\text{m}$ wide.

Prototyp am Limit der Technologie:

- Pixelgröße vs. Kapazität vs. Multiplexing
- Kontrastverhältnis (aktive/passive Fläche)
- keine Redundanz möglich (vgl. DRAM)
- 3 x 1.2M Pixel: Ausbeute problematisch

LCD: Kontakte . . .

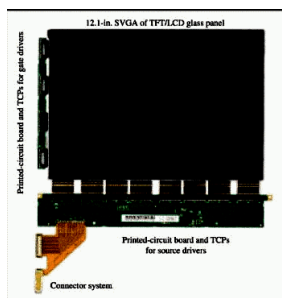


Figure 1
Part of a 12.1-in. SVGA TFT color LCD module for the IBM ThinkPad 560. Eight source-driver tape-carrier packages (TCPs), which provide a bending structure, and four gate-driver TCPs are attached by means of an anisotropic conductive film to the LCD glass panel. Printed-circuit boards are soldered to the input electrodes on the TCPs.

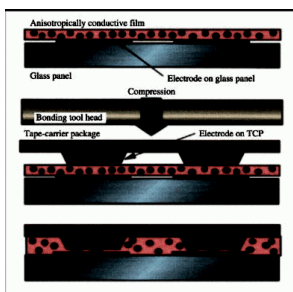


Figure 3
Bonding process for TCPs and LCD glass panels. An ACF is attached to the glass panel and a TCP is pre-attached, with accurate alignment. Final curing is then done. Adhesive fills the space between the electrodes, bonding the TCP and glass panel. Particles between the corresponding electrodes create an electrical contact, and insulation between adjacent electrodes is maintained.

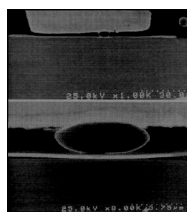
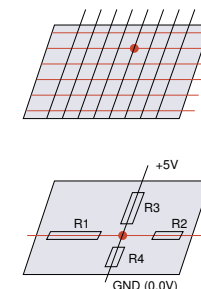
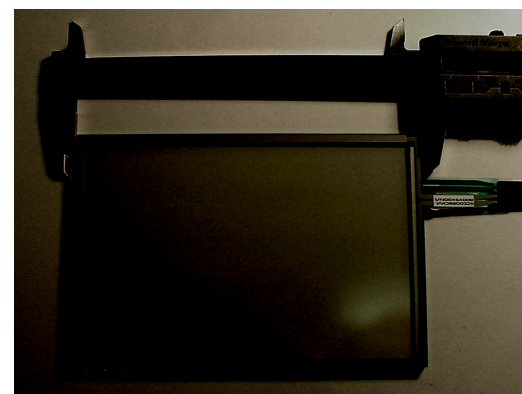


Figure 4
Cross-sectional view of interconnection by a $5\text{-}\mu\text{m}$ -diameter electrically conductive particle for the TCP sample. The particle is defocused between a copper lead electrode on the TCP and an aluminum/chromium electrode on the glass panel.

[IBM JR&D 1998]

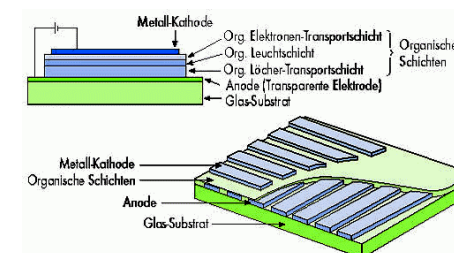
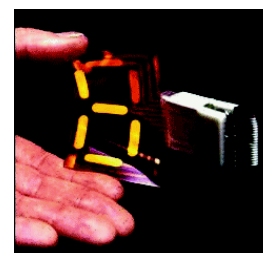
- Ansteuerung per Multiplexing, trotzdem (1280+1024) Anschlüsse
- großflächig, Glas: normales Chip-Bonding unmöglich

Touchscreen:



- kapazitiv: orthogonale Streifen-Elektroden, serielles Auslesen
- resistiv: Stiftbedienung, Kurzschlußposition wird gemessen (billiger, aber geringere Genauigkeit)

organische Displays:



- spezielle Farbstoffe
- Lichtemission bei angelegter Spannung
- im Prinzip beliebige Farbe und Helligkeit
- flexibel (Folie statt Glas als Träger)
- kein Lichtverlust durch Filter
- billige Herstellung: Farbstoffe per "Tintendrucker" aufbringen
- aber noch zu geringe Lebensdauer

organische Farbstoffe:

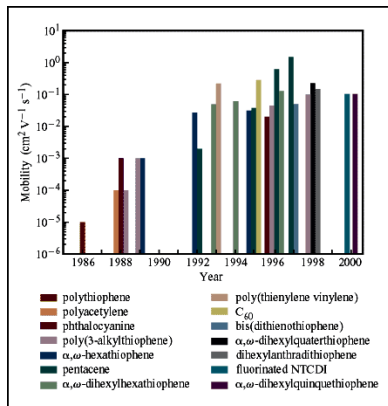
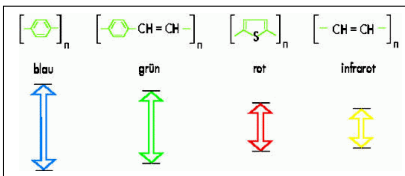


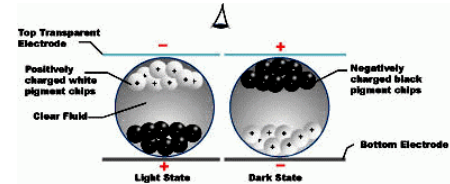
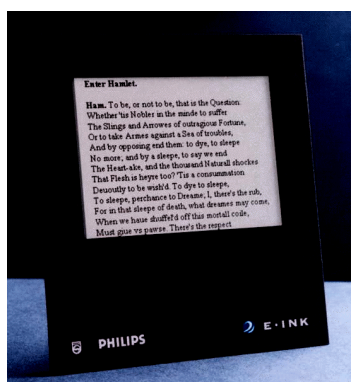
Figure 1
Semilogarithmic plot of the highest field-effect mobilities (μ) reported for OTFTs fabricated from the most promising polymeric and oligomeric semiconductors versus year from 1986 to 2000.



- Farbe
- Leuchtdichte
- Entdeckungsjahr
- von Farbstoffen für org. Polymerdisplays

(IBM J.R.D 45-1, 2001)

eInk:

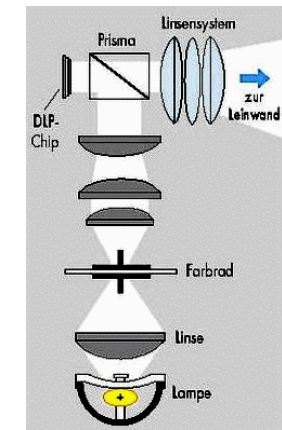
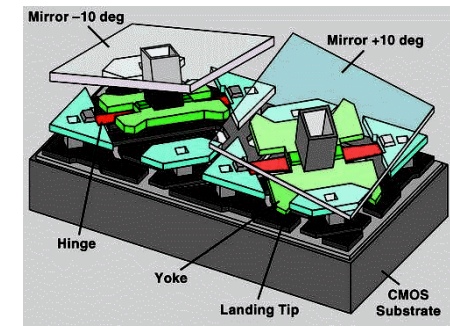


(Philips/eInk Prototyp, 80 dpi, Juni 2001)

- schwarz/weiß gefüllte/gefärbte Kugeln auf Trägermaterial
- Ansteuerung wie LCD (passiv oder TFT)
- aber metastabil: daher stromsparend

(www.gyriconmedia.com, www.eink.com)

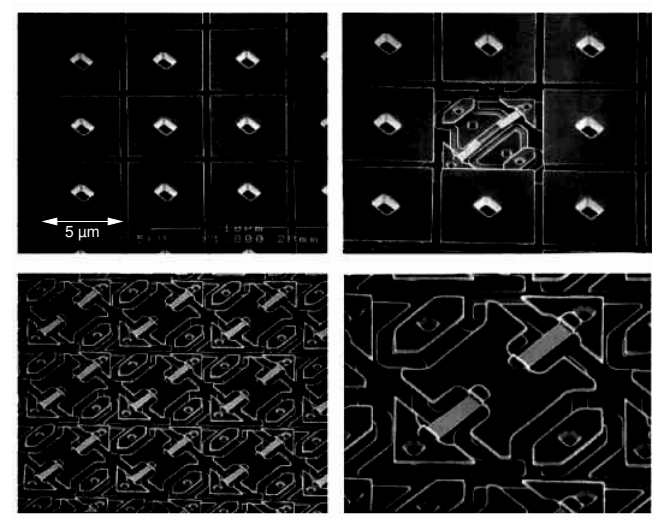
Mikrosysteme: DLP



"Digital Light Processing":

- Mikrospiegel über SRAM, z.B. 1024x768
- Lichtrichtung umschaltbar
- sehr schnell: Zeitmultiplex für Farbe&Helligkeit
- heller und kleiner als LCD-Projektoren

Mikrosysteme: DLP



Mobile PCs: Vernetzung

drahtlose Vernetzung wünschenswert

- Infrarot-Datenübertragung (IrDA und FastIrDA)
- Funknetze
 - GSM / GPRS / DECT
 - UTMS
 - Bluetooth
 - usw.

zusätzlich klassische Schnittstellen, integriert oder PCMCIA:

- Analog/ISDN-Modem
- Ethernet
- USB / 1394

(www.dafu.de, www.irda.org, www.bluetooth.org)

Mobile PCs: IrDA

Infrarot-Datenübertragung:

- ursprünglich von HP als Druckerschnittstelle eingeführt
- mittlerweile als vollwertige Schnittstelle realisiert
- diverse Protokollschichten
- Übertragungsgeschwindigkeit bis 4 Mb/s (FastIrDA)



- www.irda.org

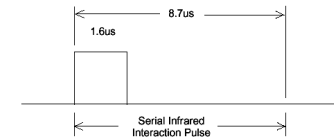
IrDA: Datenraten

Signaling Rate	Modulation	Rate Tolerance % of Rate	Pulse Duration Minimum	Pulse Duration Nominal	Pulse Duration Maximum
2.4 kb/s	RZI	+/- 0.87	1.41 us	78.13 us	88.55 us
9.6 kb/s	RZI	+/- 0.87	1.41 us	19.53 us	22.13 us
19.2 kb/s	RZI	+/- 0.87	1.41 us	9.77 us	11.07 us
38.4 kb/s	RZI	+/- 0.87	1.41 us	4.88 us	5.96 us
57.6 kb/s	RZI	+/- 0.87	1.41 us	3.26 us	4.34 us
115.2 kb/s	RZI	+/- 0.87	1.41 us	1.63 us	2.23 us
0.576 Mb/s	RZI	+/- 0.1	295.2 ns	434.0 ns	520.8 ns
1.152 Mb/s	RZI	+/- 0.1	147.6 ns	217.0 ns	260.4 ns
4.0 Mb/s (single pulse)	4PPM	+/- 0.01	115.0 ns	125.0 ns	135.0 ns
4.0 Mb/s (double pulse)	4PPM	+/- 0.01	240.0 ns	250.0 ns	260.0 ns

Table 2. Signaling Rate and Pulse Duration Specifications

5.2. Serial Infrared Interaction Pulses

In order to guarantee non-disruptive coexistence with slower (up to 115.2 kb/s) systems, once a higher speed (above 115.2 kb/s) connection has been established, the higher speed system must emit a **Serial Infrared Interaction Pulse (SIP)** at least once every 500 ms as long as the connection lasts to quiet slower systems that might interfere with the link (see Section 4.1). The pulse can be transmitted immediately after a packet has been transmitted. The pulse is shown below:



IrDA: Datenformat bis 1.152 Mbit/s

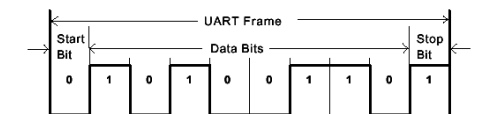
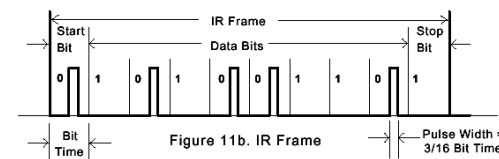
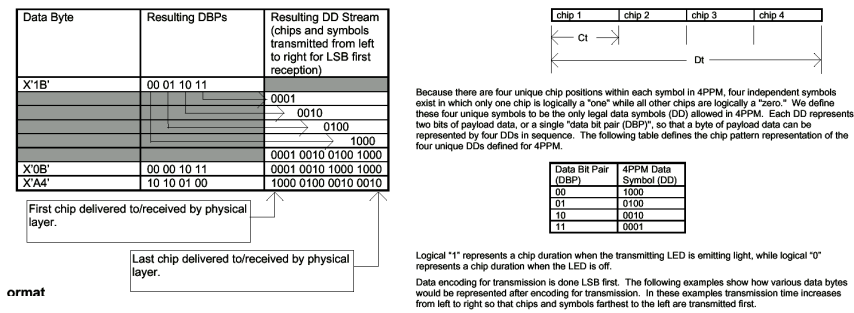


Figure 11a. UART Frame



- direkte Umsetzung von RS-232 in Einzelimpulse
- höhere Datenraten erfordern bessere Kodierung

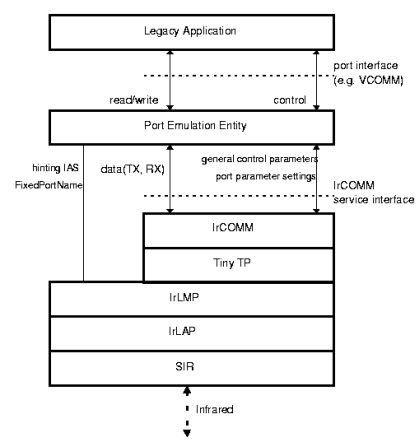
IrDA: Datenformat 4 Mb/s



ormat

- 4PPM: "Four Pulse Position Modulation"
- one-hot Bitmuster pro Zeitintervall Dt
- Lage der "1" kodiert die Daten

IrDA: Protokollstack



- Emulation von seriellen / parallelen Ports
- Netzwerkfunktionen inklusive TCP/IP

IrDA: Telecom Framework

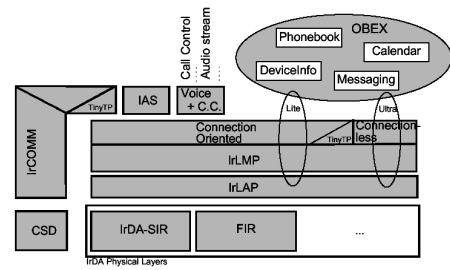
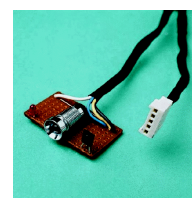
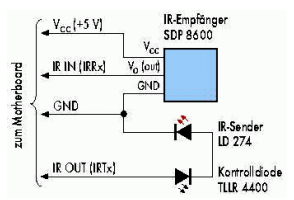


Figure 2-2 The IrDA Telecom Framework

- "OBEX": Objektkommunikation
- z.B. Austausch von Visitenkarten zwischen IrDA-Handys
- "Voice"-Kommunikation (Audiodaten plus Control)
- darunter die bekannten IrDA-Protokollschichten

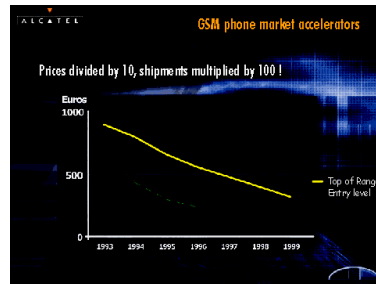
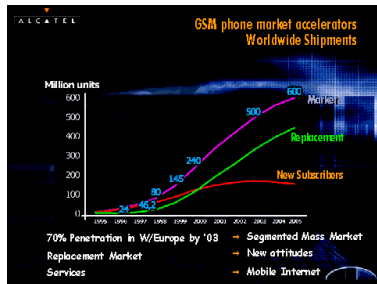
IrDA: Schaltungsvorschlag



Selbstbau eines IrDA-Transceivers:

- für direkten Anschluß an ein Motherboard
- erfordert entsprechende Hardware und BIOS-Unterstützung
- Empfängerbaustein evtl. schwer zu bekommen
- Schaltung läuft auch mit entsprechende Ersatztypen

Mobile PCs: Marktprognose GSM



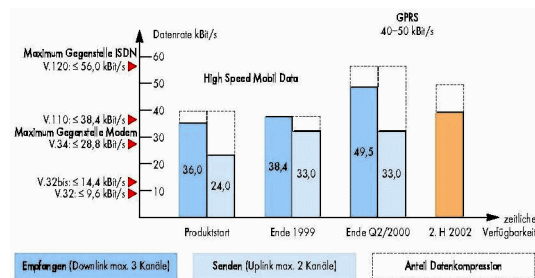
(Alcatel, www.alcatel.fr)

- Erwartung für 2004: 600M Handys, davon 400M Ersatz (!)
- Notebook-IrDA-Handy bzw. Notebook+GSM-Modem
- aber Bandbreite (9Kb/s) zu klein

GSM & Nachfolger...

Schneller funken	
besser ▶	
GSM	9,6
HSCSD	76,8
GPRS	160
ECSD	473,6
EGPRS	473,6
UMTS FDD	384
UMTS TDD	384

Stufenweise schneller: Aus GSM lässt sich mit geringem Aufwand doch ein schnelles Netz basteln, aber die GSM-Frequenzbänder sind für den künftigen Bedarf zu schmal, sodass man UMTS benötigt.



(c't 19/2000 190)

diverse Verbesserungen über GSM sind möglich...

- HSCSD: wie GSM, aber mit Kanalbündelung
- GPRS: paket-orientierte Vermittlung, wenn Kapazität frei
- EDGE: (enhanced data rates for GSM evolution)
- UMTS: deutlich komplizierter

Mobile PCs: GPRS

General Packet Radio Service:

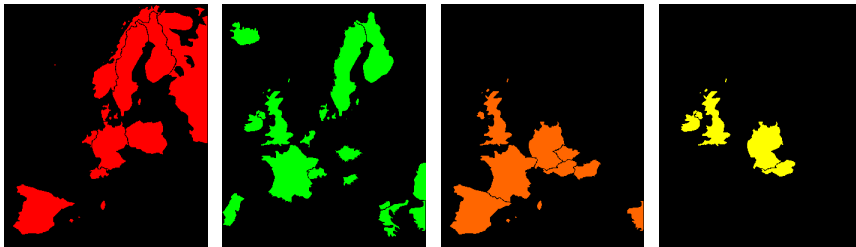
- basiert auf bestehender GSM -Technik
- Bündelung mehrerer Funkfrequenzen, sofern verfügbar
- paketbasierte Datenübertragung
- Datenrate 40-50 kb/s, geplant bis 100 kb/s
- volumenabhängige Abrechnung ?!
- Einführung seit Q2/2000, flächendeckend bis Q4/2000
- erste Handys vorgestellt
- Notebook-IrDA-Handy bzw. Notebook+GSM-Modem

Mobile PCs: UMTS

Universal Mobile Telecommunications System (UMTS/IMT-2000):

- Nachfolger der GSM900/1800 Netze
- als internationaler Standard vorgesehen
- paketbasierte Datenübertragung
- Datenrate bis 2 Mb/s (max.), 384 kb/s (flächendeckend)
- ausreichend für Videokonferenz usw.
- Frequenzen werden versteigert . . .
- Lizenzen laufen 20 Jahre
- Versorgungspflicht: 2003 25%, 2005 50% der Bevölkerung
- www.umts-forum.org

GPRS & 3G: "Claims"



Nokia Ericsson Motorola/Cisco Siemens

"GPRS & 3G Activities Maps / Europe"

- die Claims sind abgesteckt
- enorme (Markt-) Bedeutung drahtloser Netzwerke

[www.mobileapplicationsinitiative.com/GAA_upload/europe.html]

Mobile PCs: Bluetooth

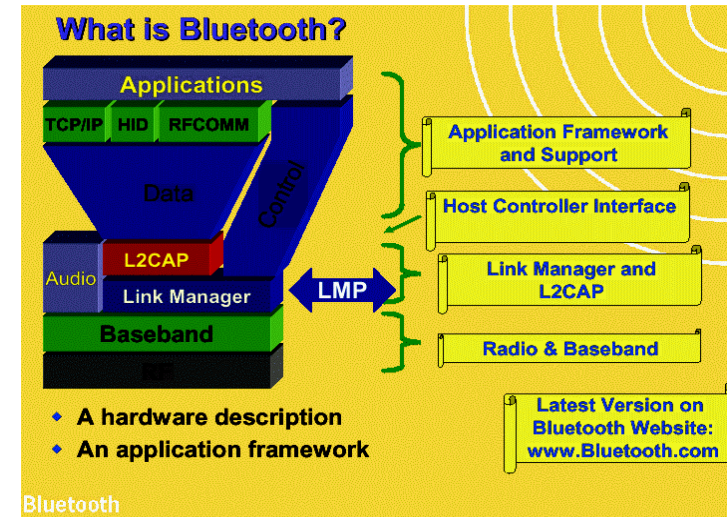
neuer Standard für Funknetze mit kleinen "Piconet"-Zellen:

- bis zu 8 Geräte bilden eine kleine Funkzelle
- Reichweite normal bis 10m, verstärkt bis 100m
- Überlappung mehrerer Funkzellen (Scatter)
- bis zu 1 Mb/s (brutto)
- sehr billige und kleine Hardware (<5\$ geplant)
- Integration in fast alle Geräte möglich
- als Ersatz für jede Art von Verkabelung
- Sicherheitsmechanismen definiert
- Standard seit Q3/1999
- www.bluetooth.com
- siehe Intel Tutorial



(Notebook-Adapter und Printerver)

Bluetooth:



Bluetooth: Cable Replacement . . .

	Bluetooth	Cable
Topology	Supports up to 7 simultaneous links	Each link requires another cable
Flexibility	Goes through walls, bodies, cloths...	Line of sight or modified environment
Data rate	1 MSPS, 720 Kbps	Varies with use and cost
Power	0.1 watts active power	0.05 watts active power or higher
Size/Weight	25 mm x 13 mm x 2 mm, several grams	Size is equal to range. Typically 1-2 meters. Weight varies with length (ounces to pounds)
Cost	Long-term \$5 per endpoint	~ \$3-\$100/meter (end user cost)
Range	10 meters or less Up to 100 meters with PA	Range equal to size. Typically 1-2 meters
Universal	Intended to work anywhere in the world	Cables vary with local customs
Security	Very, link layer security, SS radio	Secure (its a cable)

• **Cable Replacement**