

Betriebssysteme

Hardwaregrundlagen

G. Richter

9. April 2017

Inhalt

1 Einführung

- Geschichte der Hardware
- Geschichte der Betriebssysteme

2 Hardware

Geschichte

1940: Zuse Z3

- 2000 Relais
- 2 Register
- 64 Worte mit 22 Bit
- Gleitkommaarithmetik



Quelle : Wikipedia

Geschichte

1945: ENIAC

- 17468 Elektronenröhren
- 1500 Relais
- 20 Akkumulatoren
- 1 Multiplizierer
- 174 kW Stromverbrauch



Quelle : Wikipedia

Geschichte

1959: IBM 7090

- 50000 Transistoren
- 32 k Worte (36 Bit) Speicher
- 0,0005 GHz Takt
- Data-Channel (Vorläufer von DMA)
- IBM 7040 als Ein-Ausgaberechner

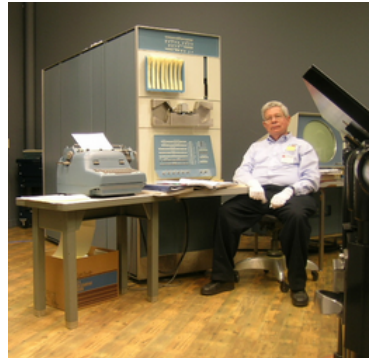


Quelle : Wikipedia

Geschichte

1961 PDP-1 (DEC)

- Magnetischer Kernspeicher für
- 4096 18-Bit Worte
- 200 kHz Takt
- CRT, 512 x 512 Pixel Grafik
- Erster Minicomputer



Quelle : Wikipedia

Geschichte

1970 PDP-11

- 16 Bit
- 600000 Mal verkauft
- integrierte Schaltkreise
- Unix und „C“



Quelle : Wikipedia

Geschichte

1981 IBM PC-XT

- 16 Bit mit 8-Bit Datenbus
- Intel 8088 Mikroprozessor
- 1 MB Adressraum
- 64 kB RAM
- PC-DOS
- 2 8" 128 kB Floppy-Disk Laufwerke
- BASIC



Quelle: Wikipedia

Anfänge der Betriebssysteme

ENIAC kein Betriebssystem, nur ein Anwendungsprogramm kann ausgeführt werden. Wechsel des Programms durch Umverdrahtung

IBM 7090 IBSYS Vorbereitete Programme auf Lochkarten werden vom Vorrechner auf Magnetband überspielt, das dann im Batchbetrieb vom Hauptrechner komplett eingelesen und ausgeführt wird. Ein Monitor lädt das Programm vom Band und führt es aus.

DEC PDP11 RSX Echtzeitbetriebssystem, Vorbild für Windows NT, Unix. Multiuserfähig

IBM PC-XT PC-DOS, Vorläufer von MS-DOS, Orientierung an Unix, Single User, Single Tasking

Intel-Prozessoren 1970 - 1980

1971: 4004

- 2300 Transistoren,
- ~100 kHz Takt
- 4 Bit

1975: 8080

- 8 Bit
- 2 MHz
- 29000 Transistoren

1980: 8088

- 16 Bit
- 8 Bit Datenbus
- 4 MHz

Intel-Prozessoren 1980 - heute

1985: 80386

- 32 Bit
- 275000 Transistoren
- Protected Mode
- Virtuelle Adressierung
- 30 Mhz

1993: 80586, dann Pentium (wegen Namensrecht)

- über 3 Mio. Transistoren
- legendärer Rechenfehler
- bis 100 MHz
- integrierte FPU
- integrierter Cache

Intel-Prozessoren

1999: Pentium III

- 500 Mhz
- 9 Mio. Transistoren
- L2 Cache
- SIMD (Single Instruction multiple Data, SSE)

2001: Itanium

- 64-Bit
- 800 MHz

2005: Pentium-D

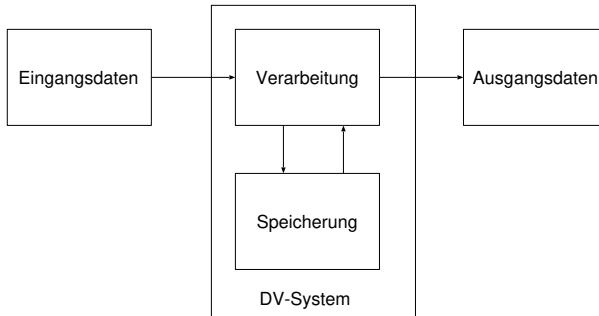
- erster Mehrkernprozessor
- 3 Ghz
- Hardware-Virtualisierung

Intel-Prozessoren

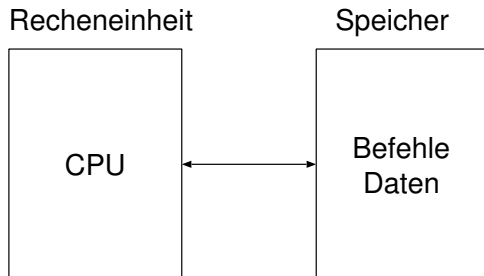
2017: Core-iX

- bis zu 4 Kerne, 8 Threads
- 3-Level-Cache
- 14 nm Strukturbreite
- integrierte Grafik
- integrierte Speicheransteuerung

Informationsverarbeitung



Ein einfacher Rechner



Dies ist eine sog. Von-Neumann Maschine, d.h. Befehle und Daten stehen in demselben Speicher

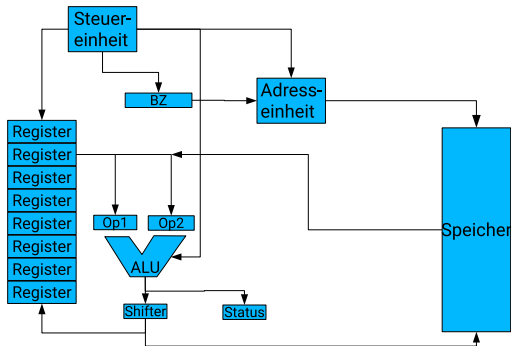
Aufbau einer einfachen CPU

BZ: Befehlszähler

ALU: Arithmetisch-logische

Einheit

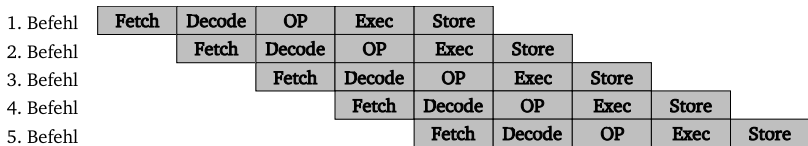
OP1/2: Operanden



Befehlsbearbeitung

- 1 Steuereinheit gibt Adresseinheit den Inhalt des BZ, um einen Befehl zu holen
- 2 der geholte Befehl wird dekodiert
- 3 Die Operanden werden in die Operandenspeicher geholt
- 4 der Befehl wird mit der ALU ausgeführt
- 5 das Ergebnis aus der ALU wird abgespeichert (Register, Speicher oder nur im Status)

Pipelining

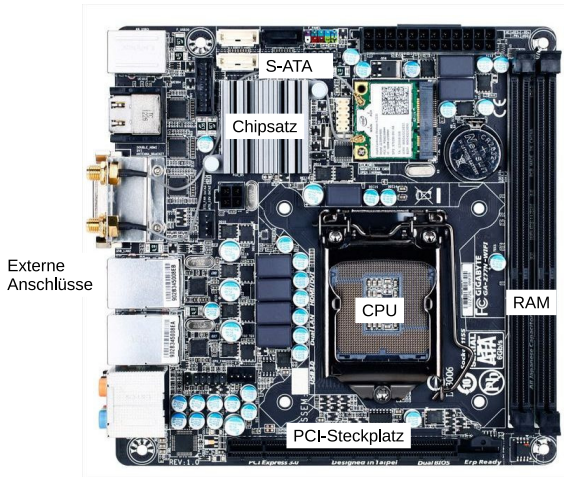


privilegierte Befehle

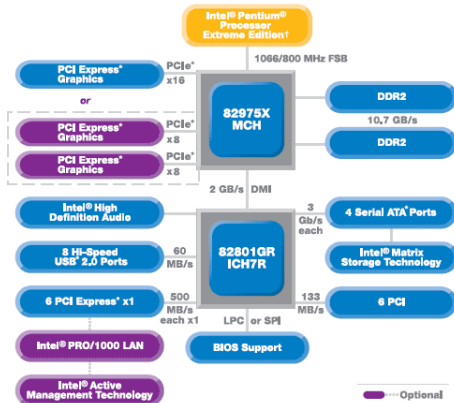
Moderne CPUs bieten sogenannte Privilegierungsstufen. Bei niedrigen Privilegien steht nur eine Untermenge des Befehlssatzes zur Verfügung. Beispiele für in niedrigen Stufen nicht erlaubte Befehle:

- Sperren des Interrupts
- Schreiben/Lesen im I/O-Adressraum
- Lesen/Schreiben von Steuerregistern der CPU

Motherboard 2014

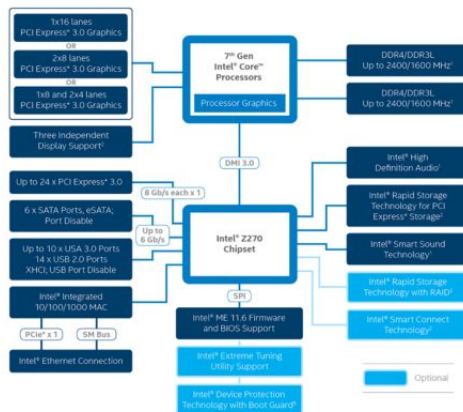


Chipsatz Northbridge/Southbridge



Quelle: Intel

Moderner Chipsatz nur Southbridge

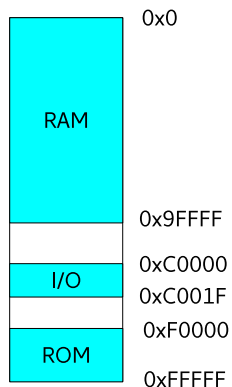


Quelle: Intel

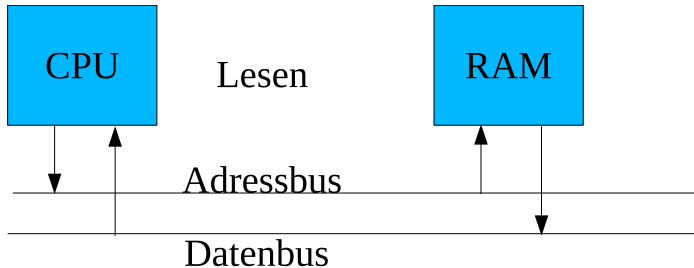
Adressraum

- Menge der Adressen, die ein Prozessor bilden kann
- 8-Bit Prozessor: $2^8 = 256$ Adressen
- 16-Bit Prozessor $2^{16} = 65536$ Adressen
- 32-Bit Prozessor $2^{32} = 4294967296$ Adressen
- 64-Bit Prozessor $2^{64} = 1,84467440737 \cdot 10^{19}$ Adressen

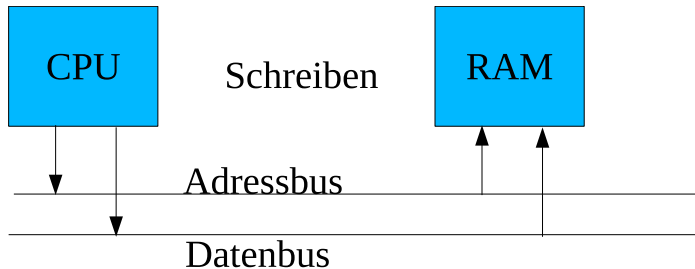
Adressraum des PC-XT: 20 Bit Adressbreite



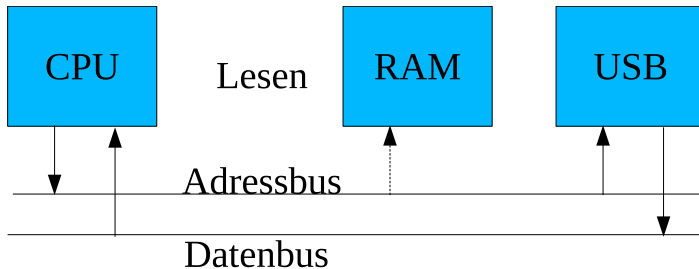
RAM Lesen



RAM Schreiben



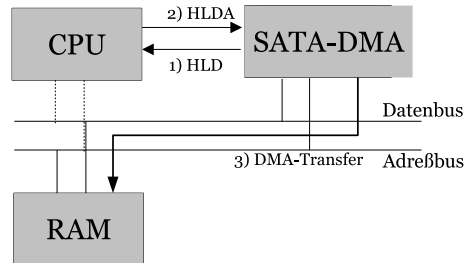
IO Lesen



DMA

Direct memory access

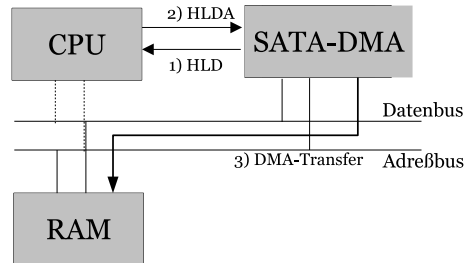
- CPU kann die Kontrolle über den Bus an einen andern Busteilnehmer übergeben



DMA

Direct memory access

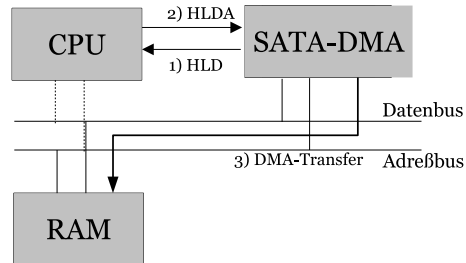
- CPU kann die Kontrolle über den Bus an einen andern Busteilnehmer übergeben
- DMA-Controller fordert den Bus an (HLD)



DMA

Direct memory access

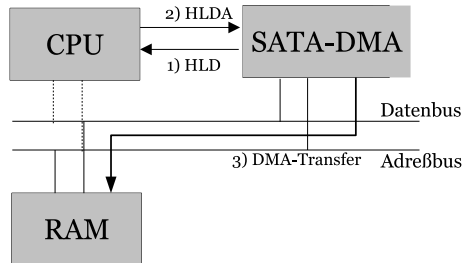
- CPU kann die Kontrolle über den Bus an einen andern Busteilnehmer übergeben
- DMA-Controller fordert den Bus an (HLD)
- CPU gibt Bus frei (HLDA)



DMA

Direct memory access

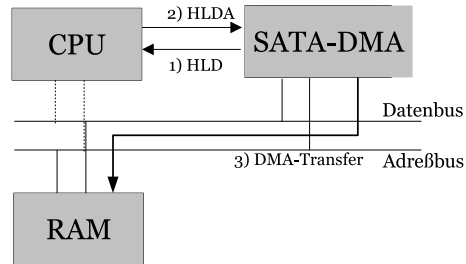
- CPU kann die Kontrolle über den Bus an einen andern Busteilnehmer übergeben
- DMA-Controller fordert den Bus an (HLD)
- CPU gibt Bus frei (HLDA)
- DMA-Controller führt den Transfer durch



DMA

Direct memory access

- CPU kann die Kontrolle über den Bus an einen andern Busteilnehmer übergeben
- DMA-Controller fordert den Bus an (HLD)
- CPU gibt Bus frei (HLDA)
- DMA-Controller führt den Transfer durch
- DMA-Controller gibt den Bus wieder frei



Interrupt

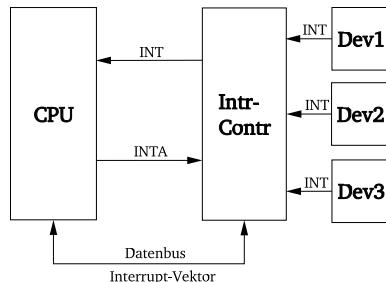
Eine CPU kann bei der Abarbeitung von Befehlen durch ein Hardware-signal unterbrochen werden. Dies nutzt man, um externe Ereignisse (Tastatureingabe, Eintreffen eines Datenblocks über LAN, uvm.) zu signalisieren.

Die Alternative wäre, diese Ereignisse permanent abzufragen (Polling). Dies würde die CPU extrem belasten und sogar von anderen Tätigkeiten abhalten.

Dies erfordert einen erhöhten Aufwand in Hardware, der aber durch den Performancegewinn mehr als aufgewogen wird.

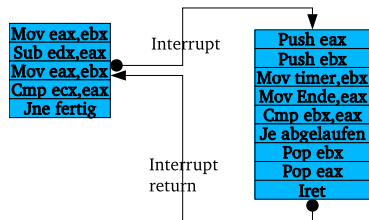
Ablauf eines Interrupts in der Hardware

- Ein Gerät (Devx) stellt die Interruptanforderung
- Der Interruptcontroller gibt das Signal an die CPU
- die CPU quittiert die Anforderung mit INTA und fordert den Interruptvektor an
- CPU unterbricht die Programmbearbeitung
- CPU führt Interruptreaktionsroutine aus



Interrupt in Software

- CPU unterbricht Bearbeitung des Anwendungsprogramms
- CPU holt mittels des Interruptvektors die Adresse der Reaktionsroutine
- CPU speichert Rücksprungadresse, Flags und Codesegment auf Stack
- CPU springt zur Reaktionsroutine
- mit dem Befehl Iret holt die CPU die Rücksprungadresse vom Stack und kehrt zur Anwendung zurück



Nutzung des Interrupt-Vektors

Beim Intel 808 wird die Einsprungsadresse für den Interrupt-Handler aus einer Tabelle entnommen. Dazu wird der Interrupt-Vektor, der beim INTACK-Zyklus übergeben wird, mit 4 (Größe eines Eintrags in der Tabelle) multipliziert. Dies ist die Adresse, an der die CPU dann Codesegment (CS) und Befehlszeiger (IP) des Handlers liest.

Ip	Interrupt 0
CS	
Ip	Interrupt 1
CS	
Ip	Interrupt 2
CS	
Ip	Interrupt 3
CS	
Ip	Interrupt 4
CS	
Ip	Interrupt 5
CS	

Nutzung des Interrupt-Vektors

- Beim Intel 80386 wird die Einsprungadresse für den Interrupt-Handler aus einer Struktur (Interrupt-Gate) entnommen.
- Dazu wird der Interrupt-Vektor mit 8 (Größe eines Eintrags in der Tabelle) multipliziert. Dies ist der Offset auf den Inhalt des CPU-Steuerregisters IDTR, an der die CPU dann das entsprechende Interrupt-Gate findet
- Darin sind wieder CS (Segm.Sel.), IP (Offset 31..0) enthalten
- Neu: darin ist auch ein DPL (Descriptor Privilege-Level) enthalten. Dieser bestimmt die Privilegstufe, auf der der Interrupt-Handler läuft.

