

# Evolution des x86-Befehlssatzes und seiner Erweiterungen

**Peter Ebert**

Dresden, 15.07.2009

---

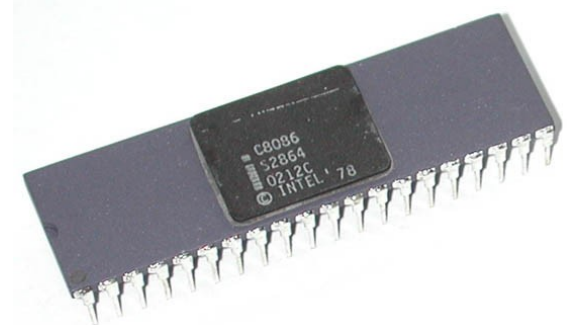
---

# Einführung

- Überblick & Historie
- Konkurrierende Befehlssatzarchitekturen
- Befehlsarten
- Registerstruktur
- x87

# Übersicht & Historie

- IBM 1981: erster PC
- x86-Architektur verwendet einen CISC-Befehlssatz
- alle Prozessoren seit dem Pentium Pro sind aber hybride CISC/RISC-Prozessoren



1978	1. Gen.: 8086		
1982	2. Gen.: 80286		
1985	3. Gen.: 80386	—————>	IA-32
1989	4. Gen.: 80486		
1993	5. Gen.: Pentium	—————>	MMX
1995	6. Gen.: P2, P3	—————>	3DNow!, SSE
1999	7. Gen.: Athlon (XP), P4	—————>	SSE2
2003	8. Gen.: Opteron	—————>	x86-64

# Konkurrierende Befehlssatzarchitekturen

- ARM (*Acorn Risc Machine*)

RISC-Architektur 1983 vom englischen Computerhersteller Acorn. Einsatz vor allem im eingebetteten Bereich

z.B.: Mobiltelefonen, PDAs, Routern, iPod, iPhone, Internet Tablets von Nokia und den neueren PDAs von ASUS, Konsolen wie der Nintendo DS, der GP2X und die Pandora.

- PowerPC (*Performance optimization with enhanced RISC Performance Chip*) 1991 durch ein Konsortium aus Apple, IBM und Motorola.

z.B.: Nintendo GameCube und Wii, Xbox 360 von Microsoft, Playstation 3 von Sony und in vielen eingebetteten Systemen. Auch benutzen PKW und Produkte in der Luft- und Raumfahrt

- SPARC (*Scalable Processor ARCHitecture*)

Von Sun Microsystems entwickelt ab 1985 und vermarktete ab 1987, offene Architektur, 1995 64-Bit-Erweiterung (UltraSparc)

# Befehlsarten

- **Transferbefehlen** werden Daten innerhalb des Systems bewegt. Die Daten werden dabei nur kopiert, d. h. bleiben an ihrem Quellort unverändert.

z.B. *move, load, transfer*

- **Arithmetische & logische Befehle**. Hier wird gerechnet, gezählt oder UND-verknüpft. Der Hauptoperand dabei ist sehr oft ein Prozessorregister (in den meisten Fällen der Akkumulator).

z.B. *ADD, AND*.

- **Sprungbefehle** dienen zur Verzweigung innerhalb eines Programmablaufs, zum Bilden von Schleifen, Reaktionen auf verschiedene Zahlenwerte und Aufruf von Unterprogrammen (Subroutinen).

z.B. *jump* oder *branch*

- **Stack-Befehle** speichern Daten auf den Stack oder holen Daten von dort zurück. Dabei wird das Stackpointer-Register automatisch aktualisiert.

z.B. *pop, push, pull*

- **Kombinierte Befehle** sind aus elementaren Aktionen der obigen Befehlsarten zusammengesetzt. → CISC

---

# Registerstruktur

- **Datenregister**

*AX (Akkumulator), BX (Basis), CX (Counter), DX (Daten)*

- **Zeiger- und Indexregister**

*SP (Stackpointer), BP (Basepointer), SI (Sourceindex), DI (Destinationindex)*

- **Segmentregister**

*CS (Codesegm.), DS (Datensegm.), SS (Stapelsegm.), ES (Extrasegm.)*  
Ab dem 386 zusätzlich FS und GS

- **Kontrollregister**

*IP (Instuction Pointer), PSW (Processor Status Word)*

---

## x87

- seit 8087 als math. Co-Prozessor
- seit der 4. Generation in der CPU integriert
- vor allem FP Verarbeitung (Sinus, Kosinus)
- 8x80bit-Register (FPR0-FPR7) als Stack + FP-Status-Register

---

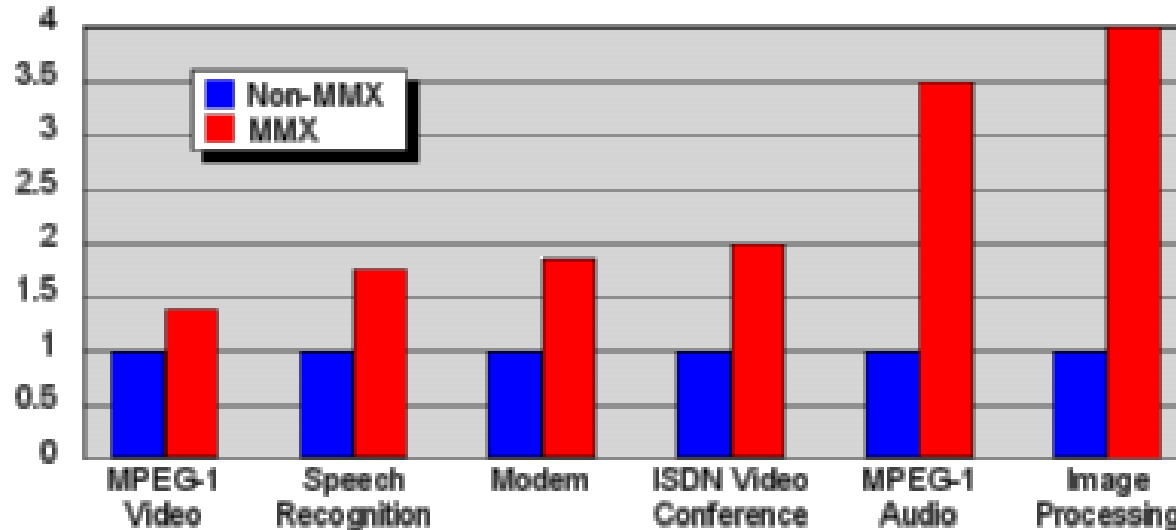
# Erweiterungen

- MMX
- 3DNow!
- SSE
- SSE2
- SSE3
- SSE4



# MMX

- Anfang 1997 von Intel eingeführt
- Grundideen: keine neue Hardware, größere Datenmengen parallelisiert zu verarbeiten → durch die SIMD-Architektur
- Leistungsgewinn für Audio- und Videoverarbeitung (lt. Intel: 10-20% für herkömmliche Software, bis 60% für optimierte Software)



### Überblick über MMX-Befehle

- Einführung von 56 Befehlen und 4 neuen Datentypen (Integer-Pakete)
- Nutzung des 64bit Mantissenbereichs der FP-Register als Register MM0-MM7
- Aufwendiger Wechsel zw. FPU- und MMX-Mode
- Hauptvorteil: Saturation-Mode und Wrap-around-Mode

Kategorie		Wrap-around	Saturation m. Vorzeichen	Saturation o. Vorzeichen
Arithmetik	Addition	PADDB, PADDW, PADDQ	PADDSB, PADDSW	PADDUSB, PADDUSW
	Subtraction	PSUBB, PSUBW, PSUBD	PSUBSB, PSUBSW	PSUBUSB, PSUBUSW
	Multiplication Multiply and Add	PMULL, PMULH PMADD		
Vergleich	Compare for Equal	PCMPEQB, PCMPEQW, PCMPEQD		
	Compare for Greater Than	PCMPGTPB, PCMPGTPW, PCMPGTPD		
Konversion	Pack		PACKSSWB, PACKSSDW	PACKUSWB
	Unpack High	PUNPCKHBW, PUNPCKHWD, PUNPCKHDQ		
	Unpack Low	PUNPCKLBW, PUNPCKLWD, PUNPCKLDQ		
Logische Operationen	And And Not Or Exclusive OR	Packed		Quadword
			PAND PANDN POR PXOR	
Shift	Shift Left Logical	PSLLW, PSLLD		PSLLQ
	Shift Right Logical	PSRLW, PSRLD		PSRLQ
	Shift Right Arithmetic	PSRAW, PSRAD		
Transfer	Register to Register Load from Memory Store to Memory	Doubleword		Quadword
		MOVB		MOVQB
		MOVD MOVQ		MOVQ MOVQB
MMX-Status		EMMS		

- keine Unterstützung durch das Betriebssystem erforderlich
- Zögerliche Unterstützung in der SW-Industrie
- wurde schnell von 3DNow! Und SSE überholt
- keine FP-Verarbeitung
- Belegung der FP-Register, langsamer Context-Switch (50Takte)

### Erste Prozessoren

- AMD: ab AMD K6
- Centaur Technology: ab IDT WinChip C6
- Cyrix: ab Cyrix 6x86MX
- Intel: ab Intel Pentium MMX
- Rise Technology: Rise mP6
- Transmeta: ab Transmeta Crusoe



# 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

- von AMD, Centaur und Cyrix erarbeitete Multimediaerweiterung

- 1997/1998 eingeführt mit dem AMD K6-2
- Revisionen: A bis G (März 2000)
- nutzt wie MMX die FP-Register
- 21 Befehle, davon 19 für FP-Verarbeitung
- +45 Befehle in späteren Prozessoren



---

# SSE

- *Streaming SIMD Extensions*, (ISSE)
- 1999 von Intel mit Pentium-III-(Katmai) vorgestellt und trug deshalb anfangs den Namen *Katmai New Instructions* (KNI)
- 70 Befehle, vor allem für FP-Parallelität
- 128bit Befehle mit Einführung von 8x128bit Registern (XMM0-XMM7) im P3.
- Unterstützung durch das Betriebssystem notwendig

## Erste Prozessoren

- AMD: ab Athlon XP, beim Duron mit Prozessorkern *Morgan*
- Centaur Technology: beim C3 mit Prozessorkern *Nehemiah*
- Intel: ab Pentium III, beim Celeron mit Prozessorkern *Coppermine*
- Transmeta: ab Efficeon

---

# SSE2

- 2001 mit dem P4
- erweitert SSE und ersetzt MMX
- ermöglicht die Verarbeitung von FP mit doppelter Genauigkeit sowie von Integer in XMM-Register → alle Datentypen in SIMD verarbeitbar
- 144 Befehle
- AMD nutzt weiter 8x128bit Register (XMM8-XMM15) im 64bit Modus
- Leistungsgewinn: 10-20% Integer, 0% FP

## SSE2-unterstützende Prozessoren

- AMD ab K8 CPUs (Athlon 64, Sempron 64, Turion 64, etc.)
- Intel ab Pentium 4, Xeon, Celeron
- Transmeta Efficeon
- VIA C7 und VIA Nano

---

# SSE3

- 2004 Pentium 4 Prescott, *Prescott New Instructions* (PNI)
- 13 Befehle  
(ADDSUBPD, ADDSUBPS, HADDPD, HADDPS, HSUBPD, HSUBPS, LDDQU, MOVDDUP, MOVSHDUP, MOVSLDUP, FISTTP, MONITOR, MWAIT)
- vor allem für horizontale Rechnung im Register und für Intel's Hyper-Threading
- AMD übernahm 11 der 13 Befehle

## SSE3-unterstützende Prozessoren

- AMD ab K8 CPUs (Athlon 64, Sempron 64, Turion 64, etc.)
- Intel ab Pentium 4, Xeon, Celeron
- Transmeta Efficeon
- VIA C7 und VIA Nano

---

# SSSE3

- *Supplemental Streaming SIMD Extension 3 [Tejas New Instructions (TNI) oder Merom New Instructions (MNI)]*
- von Intel, AMD ohne Unterstützung
- 16 neue Befehle (6 für horizontale Registerrechnung)

## Intel

- Xeon 5100 Series, Xeon 5300 Series, Xeon 3000 Series
- Core 2 Duo
- Core 2 Extreme
- Core 2 Quad
- Core i7
- Pentium Dual Core
- Celeron 4xx Sequence Conroe-L
- Celeron Dual Core E1200
- Celeron M 500 series
- Atom

## VIA

- Nano



---

# SSE4

- 2006/2007 von Intel
- 2 Standards mit 54 Befehlen  
SSE4.1: 47 Befehle ab Penryn-Kern  
SSE4.2: 7 Befehle ab Core i7
- Schwerpunkt ist die Beschleunigung von Zeichenketten-Operationen und eine bessere Vektorisierung
- AMD unterstützt 4 Befehle + 2 eigene → SSE4a

---

# Zukünftige Erweiterungen

- SSE5 (XOP, FMA4, CVT16)
- AVX
- AES & CLMUL

---

# SSE5

- 2007 von AMD als Konkurrenz angekündigt
- Details teilweise bekannt (170 Befehle, 256bit-Unterstützung)
- 2008: Intel veröffentlicht AVX-Pläne → Überarbeitung
- Mai 2009: Ankündigung von 3 Teilerweiterungen:
  - XOP: Großteil der Originalbefehle, kompatibilitäts-angepasst für AVX
  - FMA4: *Floating point vector multiply-accumulate* mit 4 Operanden (Intel vielleicht FMA3 mit 3 Operanden)
  - CVT16: Konvertierung zwischen FP-SP und FP-HP

## Erste Prozessoren

- AMD: 2011 mit dem Bulldozer Prozessor

---

# AVX

- *Advanced Vector Extensions*
- März 2008 von Intel
- Befehle im VEX-Kodierungs-Schema (Intel)
- 12 neue Befehle, vor allem für FP-Vektoren
- Unterstützt 16x256bit Register (YMM0-YMM15)
- Verbessert die Parallelisierung und den Pipelinedurchsatz
- 2 Operanden Befehle können ein separates Register als Ziel nutzen

## Erste Prozessoren

- Intel: 2010 mit dem Sandy Bridge Prozessor
- AMD: 2011 mit dem Bulldozer Prozessor

---

# AES & CLMUL

- *Advanced Encryption Standard*
- März 2008 von Intel
- 5 neue Befehle (AESENC, AESENCLAST, AESDEC, AESDECLAST, AESKEYGENASSIST)
- Beschleunigung der Codierung/Decodierung mittels AES
  
- *Carry-less Multiplication*
- März 2008 von Intel
- 1 neuer Befehle (PCLMULQDQ)
- Beschleunigung der Codierung mittels Block-Chiffre

## Erste Prozessoren

- Intel: 2010 mit den Sandy Bridge Prozessor
- AMD: 2011 mit dem Bulldozer Prozessor

---

# Allgemeine Aussagen

- Erfolg durch Prinzip der Abwärtskompatibilität, durch Zusammenarbeit mit Microsoft und gutem Marketing
- Existenz nicht gefährdet

## Trends

- CISC: von 135 Befehlen auf ~700
- SIMD: Vektorisierung aller Datentypen
- größere Register, längere Befehle

---

# Quellen

- <http://tams-www.informatik.uni-hamburg.de/lectures/2002ss/vorlesung/pc-technologie/02-x86.pdf>
- <http://tams-www.informatik.uni-hamburg.de/lectures/2002ss/vorlesung/pc-technologie/03-mmx.pdf>
- [http://tams-www.informatik.uni-hamburg.de/lectures/2005ws/vorlesung/rechnerstrukturen/RS\\_Zhang\\_02.pdf](http://tams-www.informatik.uni-hamburg.de/lectures/2005ws/vorlesung/rechnerstrukturen/RS_Zhang_02.pdf)
- <http://de.wikipedia.org/wiki/X86-Prozessor> und weiterführende Seiten
- <http://en.wikipedia.org/wiki/X86> und weiterführende Seiten
- <http://www.cpu-galaxy.at/CPU/Intel%20CPU/CPU%20Intel.htm>
- [http://www.sandpile.org/ia32/opc\\_enc.htm](http://www.sandpile.org/ia32/opc_enc.htm)
- <http://www.hs-augsburg.de/~ragun/Studio/mnp/rsys-2-6.pdf>
- <http://user.cs.tu-berlin.de/~nurgle/mmx/mmx.html>
- [http://www.thg.ru/cpu/20041228/images/intel\\_s7\\_pentium\\_233\\_mmx.jpg](http://www.thg.ru/cpu/20041228/images/intel_s7_pentium_233_mmx.jpg)
- <http://www.karbosguide.com/images/k6-2.gif>

---

Danke!

*“The x86 isn't all that complex –  
it just doesn't make a lot of sense.”*

M. Johnson, AMD, 1999