

Trutz Eyke Podschun

# Das Assembler-Buch

Grundlagen, Einführung und  
Hochsprachenoptimierung



**ADDISON-WESLEY**

---

An imprint of Pearson Education Deutschland GmbH

München • Boston • San Francisco • Harlow, England  
Don Mills, Ontario • Sydney • Mexico City  
Madrid • Amsterdam

# Inhaltsverzeichnis

<b>Vorwort</b>	11
<b>Einleitung</b>	21
<b>Teil 1: Einführung in die Assembler- Programmierung</b>	27
<b>1 Assembler-Befehle – Oder: was macht ein Compiler mit »I := 0«?</b>	29
1.1 CPU-Operationen	30
1.1.1 Arithmetische Operationen	45
1.1.2 Logische Operationen	63
1.1.3 Operationen zum Datenvergleich	69
1.1.4 Bitorientierte Operationen	74
1.1.5 Operationen zum Datenaustausch	86
1.1.6 Operationen zur Datenkonvertierung	99
1.1.7 Verzweigungen im Programmablauf: Sprungbefehle	101
1.1.8 Andere bedingte Operationen	117
1.1.9 Programmunterbrechungen durch Interrupts/Exceptions	120
1.1.10 Instruktionen zur gezielten Veränderung des Flagregisters	125
1.1.11 Operationen mit »Strings«	127
1.1.12 Präfixe	134
1.1.13 Adressierungs-Befehle	141
1.1.14 Spezielle Befehle	143
1.1.15 Verwaltungs-(System-)Befehle	166
1.1.16 Obsolete Befehle	180
1.1.17 Privilegierte Befehle	182
1.1.18 CPU-Exceptions	183
1.2 FPU-Operationen	187
1.2.1 Grundlegende arithmetische Operationen	205

1.2.2	Trigonometrische Operationen	218
1.2.3	Andere transzendente Operationen	224
1.2.4	Operationen zum Datenvergleich und Datenklassifizierung	230
1.2.5	Operationen zum Datenaustausch	238
1.2.6	Operationen zur Datenkonversion	250
1.2.7	Verwaltungsbefehle	253
1.2.8	Obsoletere Operationen	267
1.2.9	FPU-Exceptions	268
1.2.10	FPU-Emulation	271
1.3	SIMD-Operationen	272
1.3.1	SIMD, die Erste: MMX	274
1.3.2	MMX-Exceptions	306
1.3.3	MMX-Emulation	306
1.3.4	SIMD, die Zweite: SSE	307
1.3.5	SIMD, die Dritte: SSE2	343
1.3.6	Exceptions unter SSE/SSE2	360
1.3.7	Sind die SIMD verfügbar?	365
1.3.8	3DNow!, die Erste: das AMD-SSE	368
1.3.9	3DNow!, die Zweite: das AMD-SSE2	379
1.3.10	3DNow!, die Dritte: das Intel-SSE	382
1.3.11	Exceptions unter 3DNow!, 3DNow!-X und 3DNow! Professional	382
1.3.12	Ist 3DNow! verfügbar?	382
<b>2</b>	<b>Hintergründe und Zusammenhänge</b>	<b>385</b>
2.1	Stack	385
2.1.1	Der Stack – ein Stapel Daten	386
2.1.2	Stack frames – Verwaltung eines Stapels	389
2.1.3	Stack Switching	393
2.2	Speicherverwaltung	394
2.2.1	Speicherorganisation	394
2.2.2	Segmente	395
2.2.3	Die Betriebsmodi des Prozessors	399
2.2.4	Segmenttypen, Gates und ihre Deskriptoren	407
2.2.5	Deskriptorentabellen	427
2.2.6	Selektoren	429
2.2.7	Hardwareunterstützung für Deskriptoren und Deskriptortabellen	431

2.2.8	Zugriffe auf den Speicher: Von Adressen und Adressräumen	434
2.2.9	Beziehungskisten: Von der effektiven zur logischen Adresse	435
2.2.10	Speichersegmentierung: Von der logischen zur virtuellen Adresse	438
2.2.11	Paging: Von der virtuellen zur physikalischen Adresse	441
2.2.12	Auslagerungsdatei	457
2.2.13	Das 32-Bit-Betriebssystem Windows	458
2.3	Multitasking	462
2.4	Schutzmechanismen	467
2.4.1	Schutzmechanismen im Rahmen der Speichersegmentierung	468
2.4.2	Schutzmechanismen bei Zugriff auf die Peripherie	483
2.5	Exceptions und Interrupts	486
2.5.1	Interrupts	486
2.5.2	Exceptions	489
2.5.3	Interrupt-Behandlung	489
2.5.4	Emulation von Exceptions	498
2.5.5	CPU-Exceptions	499
2.5.6	FPU-Exceptions	529
2.5.7	SIMD-Realzahl-Exceptions	542
2.5.8	Interrupts und Exceptions im Real und Virtual 8086 Mode	552

## **Teil 2: Erzeugung und Verwendung von Assemblermodulen**

555

3	Der Stand-Alone-Assembler	557
3.1	Vorbemerkungen	558
3.1.1	Datenbezeichnungen	558
3.1.2	Symbole	558
3.1.3	Expression	559
3.1.4	Qualifizierte Typen	560
3.1.5	Beispiele	560
3.2	Direktiven	561
3.2.1	Direktiven zur Datendeklaration	561
3.2.2	Direktiven zur Typ-Deklaration	570
3.2.3	Direktiven zur Symboldeklaration	598
3.2.4	Direktiven zur Daten- und Codeausrichtung	604

3.2.5	Direktiven zur Deklaration und Nutzung von Prozeduren	610
3.2.6	Direktiven zu Scope und Sichtbarkeit	621
3.2.7	Vollständige Segmentkontrolle	627
3.2.8	Vereinfachte Segmentkontrolle	639
3.2.9	Direktiven zur bedingten Steuerung des Programmablaufs	652
3.2.10	Makros	655
3.2.11	Bedingte Assemblierung	662
3.2.12	Direktiven zur Steuerung von Listings	666
3.2.13	Direktiven zur Anwahl des Befehlssatzes	672
3.2.14	Interaktion mit dem Programmierer	675
3.2.15	Assembler-Einstellungen	679
3.2.16	Verschiedenes	689
3.3	Operatoren	690
3.3.1	Operatoren in Ausdrücken	690
3.3.2	Operatoren für Strings	704
3.3.3	Run-Time-Operatoren	704
3.3.4	Operatoren in Makros	706
3.4	Vordefinierte Symbole	709
3.4.1	Vordefinierte String-Symbole (Textmakros)	709
3.4.2	Vordefinierte Symbole (Numerische Makros)	710
3.4.3	Makros zur Verwaltung von Strings	713
3.4.4	TASM-Symbole für OOP	714
3.5	Assemblermodule in Hochsprachen	714
3.5.1	Erzeugung des Assembler-Quelltextes	715
3.5.2	Assemblierung zum OBJ-File	719
3.5.3	Einbindung in Hochsprachen	720
3.5.4	Aufrufkonventionen	723
3.5.5	Übergabekonventionen	725
3.5.6	FAR und NEAR – eine Frage des Standpunktes	726
3.5.7	Tabus	726
3.6	Assembler und die strukturierte Ausnahmebehandlung (SEH)	729
4	<b>Der Integrierte Assembler</b>	745
4.1	Programmierung mit dem Inline-Assembler	745
4.2	Inline-Assembler und die strukturierte Ausnahmebehandlung (SEH)	760

<b>Teil 3: Anhang</b>	761
5 Anhang	763
5.1 Definitionen und Erläuterungen	763
5.1.1 Befehlssemantik	763
5.1.2 Adress- und Operandengrößen	765
5.1.3 Mnemonics, Befehlssequenzen, Opcodes und Microcode	768
5.1.4 Anwendungen, Programme, Module, Tasks, Prozesse und Threads	772
5.1.5 »Unschärfen« und Ungenauigkeiten in diesem Buch	776
5.2 Datenformate	778
5.2.1 »Little-Endian«- und »Big-Endian«-Format	781
5.2.2 Binäre Zahlendarstellung und Hexadezimalsystem	782
5.2.3 Elementardaten	788
5.2.4 Gepackte Daten	811
5.2.5 Erweiterte Elementardaten	814
5.2.6 Gegenüberstellung der verschiedenen Datenbezeichnungen	816
5.3 Speicheradressierung	816
5.4 Ports	827
5.5 Befehls-Decodierung	832
5.5.1 Decodierung des/der Präfixe(s)	832
5.5.2 Decodierung des Opcodes	832
5.5.3 Decodierung eines ModR/M- und ggf. eines SIB-Byte	833
5.5.4 Decodierung einer Adresse oder Konstanten	833
5.6 Tabellen zur Single-Instruction-Multiple-Data-Technologie (SIMD)	844
5.6.1 Unter SIMD auf Intel-Prozessoren verfügbare Datenformate	844
5.6.2 Unter SIMD auf Intel-Prozessoren verfügbare Instruktionen	845
5.6.3 Unter SIMD auf AMD-Prozessoren verfügbare Datenformate	850
5.6.4 Unter SIMD auf AMD-Prozessoren verfügbare Instruktionen	851
5.6.5 Entsprechungen und Unterschiede der Intel- und AMD-SIMD-Befehle	855
5.7 Weitere Register der CPU	856
5.7.1 Kontroll-Register	856
5.7.2 Debug-Register	863

5.7.3	Modellspezifische Register (MSRs)	867
5.8	FPU-, MMX- und XMM-Umgebung	868
5.9	Historie	874
5.9.1	Pentium 4	874
5.9.2	Pentium III, Xeon	874
5.9.3	Pentium II, Pentium II Xeon, Celeron	875
5.9.4	Pentium Pro	875
5.9.5	Pentium	877
5.9.6	80486	879
5.9.7	80386 / 80387	881
5.9.8	80286 / 80287	890
5.9.9	80186/80188	895
5.9.10	8086 / 8087	896
5.9.11	16-Bit-Protected-Mode	898
5.10	Verzeichnis der Abbildungen und Tabellen	900
5.10.1	Abbildungen	900
5.10.2	Tabellen	906
5.11	ASCII- und ANSI-Tabelle	911
	<b>Stichwortverzeichnis</b>	<b>913</b>