

# Inhaltsverzeichnis

<b>Vorwort</b>	XI
<b>1. Einführung in die Speicherorganisation</b>	1-1
1.1 Grundsätzliches	1-2
1.2 Linearer Speicher	1-2
1.3 Mapping des linearen Speichers	1-5
1.4 Einteilung des logischen und physikalischen Adressraums in Abschnitte	1-6
1.5 Zugriffsrechte auf die Programmabschnitte	1-7
1.6 Virtueller Speicher	1-7
1.7 Segmentierter Speicher	1-8
1.8 Mapping des segmentierten Speichers	1-10
1.9 Segmenttypen und Zugriffsrechte	1-11
1.10 Überwachung des Zugriffs	1-11
<b>2. Memory Management und virtuelle Adressierung</b>	2-1
2.1 Memory Management	2-2
2.2 Virtuelle Adressen	2-2
2.3 Segmente und Segment-Deskriptoren	2-6
2.4 Deskriptor-Tabellen	2-9
2.5 Privileg-Ebenen	2-18
2.6 Übersetzung von virtuellen in physikalische Adressen	2-20
2.7 Die Segment-Register im "Protected Mode"	2-21
2.8 Deskriptor-Typen	2-26
2.9 Alias-Deskriptoren	2-26
2.9.1 Programmierbeispiel: EPROM-Tabellen in RAM kopieren	2-31
2.10 Initialisierung der System-Segment-Register GDTR und IDTR	2-33
2.10.1 Programmierbeispiel: GDTR und IDTR laden	2-35
2.10.2 Programmierbeispiel: RAM-Tabellen dynamisch erweitern	2-37
2.11 Initialisierung des System-Segment-Registers LDTR	2-39
2.11.1 Programmierbeispiel: Laden des LDTR-Registers im "Protected Mode"	2-45
2.12 RAM-Tabellen als Daten-Segmente	2-46
2.12.1 Programmierbeispiel: Zugriff auf Alias-Deskriptoren	2-49
2.12.2 Programmierbeispiel: Alias-Segment	2-53
2.13 Starten eines 80286-Programms im "Real Address Mode"	2-55

<b>3.</b>	<b>Schutz</b>	<b>3-1</b>
3.1	Schutztypen	3-2
3.2	Schutz-Implementierung	3-2
3.3	Memory Management und Schutz	3-5
3.4	Privileg-Ebenen und Schutz	3-11
3.5	Struktur des Software-Systems	3-15
3.6	Daten-Segment/Executable-Segment-Deskriptoren	3-19
3.6.1	Gemeinsamkeiten der "Access Bytes"	3-21
3.7	Das "Access Byte" des Executable-Segment-Deskriptors	3-22
3.7.1	Das Readable-Bit (R)	3-22
3.7.2	Das Conforming-Bit (C)	3-23
3.8	Das "Access Byte" des Daten-Segment-Deskriptors	3-25
3.8.1	Das Writable-Bit (W)	3-25
3.8.2	Das Expansion-Direction-Bit (ED)	3-25
3.9	Ablaufsteuerung eines Programms ohne Privileg-Wechsel	3-27
3.9.1	Privileg-Ebene einer "Task"	3-27
3.9.2	Programmierbeispiel: Intra-Level-Steuerung	3-31
3.9.3	Requestor's Privilege Level (RPL)	3-33
3.10	Ablaufsteuerung eines Programms mit Privileg-Wechsel	3-35
3.10.1	Der "Call Gate"-Deskriptor	3-36
3.10.2	Programmierbeispiel: Privileg-Wechsel über einen "Call Gate"-Deskriptor	3-38
3.10.3	"Call Gate"-Deskriptor Privileg-Level (DPL)	3-40
3.10.4	Regeln für "Interlevel"-Programmabläufe	3-41
3.11	"Interlevel"-Rückkehr	3-43
3.12	"Interlevel"-Prozeduraufruf mit Parameterübergabe	3-45
3.12.1	Das "WORD COUNT"-Feld	3-45
3.12.2	Programmierbeispiel: Parameterübergabe	3-46
3.13	JMP-Befehl und "Call Gate"-Deskriptor	3-50
3.14	Daten-Zugriffe	3-51
3.15	Parameter-Gültigkeit	3-53
3.15.1	Programmierbeispiel: Privileg-Verletzung	3-54
3.15.2	Anwendung des ARPL-Befehls	3-58
3.15.3	Programmierbeispiel: Privileg-Anpassung	3-61
3.16	Genauere Überprüfung von Pointer-Parametern	3-64
3.16.1	Anwendung der Befehle VERR und VERW	3-65
3.16.2	Programmierbeispiel: Zugriffs-Überprüfung	3-68
3.16.3	Anwendung der Befehle LAR und LSL	3-70
3.16.4	Programmierbeispiel: Zugriffs- und Limit-Überprüfung	3-74
<b>4.</b>	<b>"Tasks" und Status-Wechsel</b>	<b>4-1</b>
4.1	Der Task-Status	4-3



---

4.2	Task-Status-Segment-Deskriptoren	4-6
4.3	Initialisierung des Task-Registers	4-7
4.3.1	Programmierbeispiel: Task-Register laden	4-7
4.4	Starten einer "Task"	4-10
4.5	Task-LDT-Selektor und -Deskriptor	4-12
4.6	Gemeinsame LDT für mehrere "Tasks"	4-13
4.7	Privileg-Wechsel und Stack-Segment-Anfangspositionen	4-14
4.8	Task-Wechsel	4-17
4.9	Verkettung von "Tasks"	4-23
4.9.1	Task-Wechsel-Effekte durch JMP, CALL	4-25
4.10	Task Gates	4-26
4.10.1	Die Felder DESTINATION-SELEKTOR, TYPE, das Bit P (Present)	4-26
4.10.2	Task-Gate-Deskriptor Privilege-Level (DPL)	4-27
4.11	Starten der ersten "Task" ("Urtask")	4-29
4.11.1	Programmierbeispiel: "Urtask"	4-30
4.12	Task-Status-Alias-Segmente und -Deskriptoren	4-38
<b>5. Interrupts</b>		<b>5-1</b>
5.1	Interrupt-Deskriptor-Tabelle IDT	5-2
5.2	Interrupt Task	5-3
5.3	Interrupt-Prozedur	5-6
5.4	Interrupt Gate- und Trap Gate-Deskriptoren	5-7
5.5	Interrupt und Gate-DPL	5-8
5.6	Unterschied zwischen "Interrupt"-Deskriptoren und "Trap Gate"-Deskriptoren	5-12
5.7	Task-Wechsel-Effekte bei Interrupts	5-13
5.8	"Conforming"-Interrupt-Prozeduren	5-14
5.9	Stack-Eintragungen nach Interrupt	5-15
5.10	Verlassen des "Interrupt Handlers"	5-16
5.11	Task-Wechsel-Effekte bei IRET	5-18
5.12	Interrupt-Rückkehr innerhalb der gleichen "Task"	5-19
5.13	Externe Interrupts	5-22
5.13.1	Programmierbeispiel: Interrupt-Vektor-Nummern	5-24
5.14	Nicht-maskierbare externe Interrupts	5-26
<b>6. Ausnahmen und Interrupts</b>		<b>6-1</b>
6.1	Interaktionen zwischen Interrupt-Anforderungen und "Gate"-Deskriptoren	6-3
6.2	Der Fehler-Code	6-3
6.3	Ausnahmen und reservierte Vektoren	6-4
6.4	Ausnahme-Bedingungen und Fehlerbehandlung	6-6

---

6.4.1	Interrupt 0: Divisions-Fehler	6-6
6.4.2	Interrupt 1: Einzelschritt	6-6
6.4.3	Interrupt 3: Breakpoint	6-7
6.4.4	Interrupt 4: Overflow	6-7
6.4.5	Interrupt 5: Feldgrenze überschritten	6-7
6.4.6	Interrupt 6: undefinierter Operations-Code	6-8
6.4.7	Interrupt 7: Prozessor-Erweiterung nicht vorhanden	6-8
6.4.8	Interrupt 8: Doppelfehler	6-9
6.4.9	Interrupt 9: Segment-Überlauf der Prozessor-Erweiterung	6-9
6.4.10	Interrupt 10: Ungültiges Task-Status-Segment	6-9
6.4.11	Interrupt 11: Segment nicht vorhanden	6-11
6.4.12	Interrupt 12: Stack-Segment-Überlauf oder "nicht vorhanden"	6-11
6.4.13	Interrupt 13: Allgemeine Schutzverletzung	6-12
6.4.14	Interrupt 16: Fehler der Prozessor-Erweiterung	6-13
 <b>7. Eingabe/Ausgabe</b>		7-1
7.1	Ein-/Ausgabe und Schutz	7-2
7.2	Programmierbeispiel: IOPL-Anpassung	7-5
7.3	Interrupt-Flag und IOPL	7-7
 <b>8. Benutzung der Numerischen Prozessor-Erweiterung 80287</b>		8-1
8.1	Das "Emulation Mode Flag" EM	8-3
8.2	Das "Math Present Flag" MP	8-3
8.3	Das "Task Switched Flag" TS	8-4
8.4	Der WAIT-Befehl	8-4
8.5	Interpretation des MP- und EM-Flags	8-5
8.6	Der 80287-Task-Status	8-5
8.7	Programmierbeispiel: 80287-Initialisierung	8-6
8.8	80287-Context-Wechsel	8-10
8.9	Programmierbeispiel: Context-Wechsel	8-12
 <b>9. Einführung in den "System Builder" BLD286</b>		9-1
9.1	Entwurf eines statischen Systems	9-2
9.2	Die BLD286-Spezifikationen	9-3
9.2.1	Das "Build"-Programm	9-5
9.2.2	SEGMENT-Definition	9-5
9.2.2.1	Beispiele für Segment-Definitionen	9-9
9.2.3	TABELLEN-Definition	9-9
9.2.3.1	Beispiele für Tabellen-Definitionen	9-13

9.2.4	TASK-Definitionen	9-15
9.2.4.1	Programmierbeispiel: System-Komposition für das ASM286-Programm – Urtask	9-18
9.2.5	GATE-Definition	9-23
9.2.5.1	Programmierbeispiel: System-Komposition für ein ASM286-Programm mit Privileg- und Task-Wechsel über Gate-Deskriptoren	9-29
9.2.6	ALIAS-Definition	9-40
9.2.6.1	Programmierbeispiel: Alias-Task-Status-Segment	9-41