

Inhaltsverzeichnis

Computertechnik 1	2	Default-Segmentregister und Segment-Override-Präfix	3	Rotate durch Carry-Flag	4	Minimaldeklaration: PROC, ENDP und RET	6	Prinzip der Pointer	7
Zahlensysteme.....	2	Opcode-Aufbau der Move-Befehle	3	Rotate ohne Carry-Flag.....	4	Far- und Near-Prozedur-Deklaration.....	6	Direkte Adressierung Variablenname → Wert.....	7
Zahlenumwandlungen.....	2	Allgemeiner Move-Befehl.....	3	Befehle zur Veränderung des Carry-Flag.....	4	Prozedurdeklaration mit Register-Save und -Restore	6	Speicher-indirekte Adressierung Variablenname → Adresse → Wert	7
Dezimal zu Binär.....	2	Immediate Move-Befehl.....	3	Shift-Befehle.....	4	Aufruf von Unterprogrammen.....	6	Unterrichtsnotizen	8
Hexadezimal zu Dezimal	2	Move-Befehl mit Akkumulator	3	Shift Left (SHL) und Shift Arithmetic Left (SAL)	4	Parameterübergabe an Unterprogramme	6	Arbeitsregister	8
Dezimal zu Hexadezimal	2	Move-Befehl mit segmentregistern	3	Shift Right (SHR)	4	Arten an Parameterübergaben.....	6	Memory Interface	8
Darstellung negativer Zahlen.....	2	Move-Befehl mit allgemeiner Adressierung	3	Shift Arithmetic Right (SAR)	4	Realisierung in der Assembler Programmierung	6	I/O Interface	8
Einerkomplement.....	2	Move-Befehl mit Immediate-Adressierung.....	3	Vergleichs- und Sprungbefehle	4	Interrupt	6	Datentransfer	8
Zweierkomplement.....	2	Akkumulator-Move-Befehle mit direkter Adressierung	3	Einteilung der Sprungbefehle.....	4	Interrupt Typen	6	Register für indirekte Adressierung.....	8
Rechnen mit negativen Zahlen	2	Move-Befehle für Segmentregister	3	Begriffe.....	4	Synchron zur Programmausführung (SW Interrupts)	6	Physikalische Adresse	8
Addition und Subtraktion	2	Opcode-Aufbau des Exchange-Befehls.....	3	Bedingte ←→ Unbedingte Sprungbefehle	4	Asynchron zur Programmausführung	6	Probleme bei der Adressierung.....	8
Multiplikation.....	2	Exchange-Befehl mit allgemeiner Adressierung.....	3	Short ←→ Near ←→ Far-Sprungbefehle.....	4	Traps → Instruktion INT	6	Opcodes Beispiel.....	8
Einführung in Microcomputersysteme.....	2	Exchange-Befehl für Akkumulator und 16-Bit-Register	3	Direkte ←→ Indirekte Sprungbefehle.....	4	Traps (INT n) vs. Prozeduraufrufe.....	6	Bitmasken	8
Microcomputersystem	2	Opcode-Aufbau der IN-/OUT-Befehle	3	Absolute ←→ Relative Sprungbefehle.....	4	Unterschiede auf dem Stack	6	Rotate.....	8
Microcomputer	2	Indirekte Portadressierung	3	Intra- und Intersegment-Sprungbefehle	4	Interrupt-Vektor-Tabelle.....	6	Rotate through Carry Left.....	8
Aufbau des Bussystems	2	Direkte Portadressierung.....	3	Intrasegment-Sprung.....	4	Initialisierung Interrupt-Vektor-Tabelle	6	Vergleiche.....	8
Speicher (Memory).....	2	Befehlsgruppen: Immed, Shift, Grp1 und Grup2.....	3	Intersegment-Sprung.....	4	Exceptions	6	Register	8
Die zentralen Speicher.....	2	Opcode der 80186-Befehle.....	3	Unbedingte Sprungbefehle	4	Hardware Interrupts	6	Stack Frames (int: 16-Bit)	8
Die peripheren Speicher	2	Arithmetische Operationen.....	3	Bedingte Sprungbefehle	5	Ablauf externer (HW) Interrupt	6	Prüfung 1: Aufgaben	8
Speicherung von Daten in zentralen Speichern.....	2	Einführung	3	Arithmetische Sprungbefehle.....	5	Nested Interrupts (extern), Register sichern.....	6	Binäre Zahlen.....	8
Adressierungsarten.....	2	Übersicht der arithmetischen Befehlen.....	3	Arithmetisch "unsigned" (below or above).....	5	Stack-Aufruf und Rücksprung.....	7	Binär Rechnen.....	8
Immediate-Adressierung	2	Binäre arithmetische Operationen.....	3	Arithmetisch "signed" (greater or less).....	5	Interrupt-System des 80186.....	7	Assembler	8
Direkte Adressierung.....	2	Unäre arithmetische Operationen	3	Flag-orientierte Sprungbefehle	5	Initialisieren von externen Interrupts (HW)	7	Opcode	8
Indirekte Adressierung	2	Datenfluss im Prozessor bei arithmetischen Befehlen.....	3	Vergleichsbefehle	5	Interrupt Vektor-Tabelle	7	Prüfung 2: Aufgaben	8
Indexierte Adressierung	2	Einschränkungen.....	3	Befehle zur Konstruktion von Zählschleifen	5	Interrupt Prioritäten	7	Interrupts	8
Architektur des Prozessors 8068	2	Die "arithmetischen" Flags des Prozessors 8086	3	Strukturierte Codierung in Assembler.....	5	Datenkonsistenz Problematik bei Interrupt: Beispiel	7	Datenübergabe und Stack	8
Blockschaltbild des Prozessors 8068.....	2	Das Carry-Flag	3	Strukturierte Codierung im Entwicklungsprozess	5	Zusammenfassung Interrupt	7	Assembler	8
Die BIU (Bus Interface Unit)	2	Das Overflow-Flag	3	Das Prinzip der strukturierten Codierung	5	Darstellung von Datentypen	7	Opcode.....	8
Die EU (Execution Unit)	2	Das Sign-Flag.....	3	Realisierung der Strukturelemente in Assembler.....	5	Einführung.....	7	Codebeispiele.....	9
Der Registersatz des Prozessors 8086	2	Das Zero-Flag.....	3	If/then/else	5	Entwicklung der Datentypen	7		
Die Adressbildung des 8086.....	2	Das Parity-Flag.....	3	While.....	5	Konzept von Datentypen.....	7		
Adressräume.....	2	Die arithmetischen Befehle im Detail.....	4	Repeat/until	5	Wertebereich	7		
Datentransfer-Befehle	2	Addition	4	Endless (Endlosschleife)	5	Operationen	7		
Assembler-Befehle.....	2	Addition mit Carry	4	For (universelle Zählschleife).....	5	Begriffe für die Implementation von Datentypen.....	7		
Assembler-Schreibweise und -Syntax.....	2	Erhöhen um eins (Increment)	4	Do (Zählschleife unter Verwendung des LOOP-Befehls)	5	Wichtige Informationen eines Datenelementes.....	7		
Symbolische Speicheradressen	2	Subtraktion	4	Switch-case.....	5	Übersicht Datentypen	7		
Adressierung von Datenoperanden	2	Subtraktion mit Borrow	4	Strukturelemente mit mehreren Bedingungen	5	Skalare Datentypen	7		
Adressierungsarten	2	Verminderung um eins (Decrement)	4	Zwei Bedingungen mit UND verknüpft	5	Strukturierte Datentypen.....	7		
Bildung der Offset-Adresse.....	2	Negieren einer vorzeichenbehafteten Zahl	4	Zwei Bedingungen mit ODER verknüpft.....	5	Zeigertypen	7		
Datentransfer-Befehle	2	Multiplikation.....	4	Unterprogramme und Stack	5	Darstellung skalarer Datentypen.....	7		
Der Move-Befehl: Kopieren von Datenwerten	2	Division	4	Einführung	5	Typ Character: Zeichen.....	7		
Der Exchange-Befehl: Austauschen von Datenwerten.....	2	Konvertierung der Operandengröße	4	Das Stack-Prinzip	5	Typ Integer: ganze Zahlen	7		
Die Input/Output-Befehle: Ein-/Ausgabe von/zu Ports	2	Logische und Shift-/Rotate Befehle	4	Queue	5	Typ Enumeration: Aufzählungstyp	7		
Ergänzungen zur Schreibweise von Speicheroperanden	2	Das Prinzip der logischen Operationen	4	Stack	5	Typ Boolean: logischer Datentyp.....	7		
Die Segmentpräfixe	2	Anwendung eines Bitmusters	4	Stack-Realisation beim 8086.....	6	Darstellung strukturierter Datentypen	7		
Maschinencode der 8086-Prozessoren.....	2	Logische Befehle	4	Stack-Segment und Stackpointer.....	6	Typ Record: Verbund	7		
Aufbau des 8086-Opcodes	2	Die logischen Befehle AND, OR und XOR	4	Funktionen der Stack-Operationen PUSH und POP	6	Typ Array: Felder.....	7		
Prinzipieller Aufbau	2	Der logische Befehl NOT	4	Stack-Befehle des 8086.....	6	Typ String: Zeichenketten.....	7		
Bedeutung der Bitgruppen	2	Shift- und Rotate-Befehle	4	Daten sichern / zurückerholen	6	Längenzähler	7		
Erstes Byte "opcode"	2	Das Prinzip der Shift- und Rotate-Befehle	4	Unterprogramme aufrufen / beenden.....	6	Terminator	7		
Zweites Byte "addr-mode" (address-mode).....	3	Rotate-Befehle	4	Deklaration von Unterprogrammen in Assembler	6	Typ Pointer: Zeiger auf beliebige Datenstrukturen.....	7		