

Inhalt

Vorwort	1
1. Einführung	7
1.1 Grundbegriffe	7
1.2 Berechenbarkeit	12
1.2.1 Turingmaschinen	12
1.2.2 Die Church-Turing-These	16
1.2.3 Turing-Vollständigkeit, algorithmische Vollständigkeit	17
1.3 Der Universalrechenautomat	17
1.3.1 Turingmaschinen und Rechenmaschinen	17
1.3.2 Berechenbarkeit und praktisches Rechnen	18
1.3.3 Funktionseinheiten	20
2. Minimalprinzipien der Rechentechnik	23
2.1 Minimalprinzipien der Turing-Vollständigkeit	23
2.2 Minimalmaschinen der Theorie	26
2.2.1 Das algorithmische Schema von Post als Grundlage des Universalrechners	26
2.2.2 Eine erste fiktive Maschine	28
2.2.3 Maschinen der mathematischen Grundlagenforschung	29
2.2.4 Rechnende Universalmaschinen	33
2.2.5 Die Minimalmaschine nach van der Poel	38
2.2.6 Muß man überhaupt verzweigen? – Die bedingte Befehlsausführung	40
2.2.7 Muß man überhaupt rechnen?	43
2.2.8 Universalrechner verbessern?	47
2.3 Minimalmaschinen der Praxis	49
2.3.1 Einfachmaschinen	49
2.3.2 Möglichst wenige Befehle	51
2.3.3 Einfachste Befehlsformate	63
3. Turingmaschinen	73
3.1 Hypothetische und fiktive Turingmaschinen	73
3.2 Turingmaschinen in der Grundlagenforschung	77
3.3 Spezielle Turingmaschinen	80
3.4 Universelle Turingmaschinen	84
4. Zur Begründung des Universalrechners	95
4.1 Entwicklungswege	95
4.2 Fiktive Turingmaschinen	98
4.2.1 Die Turing-Vollständigkeit nachweisen	98
4.2.2 Eine Definition der Turing-Vollständigkeit	99
4.2.3 Den Steuerautomaten implementieren	100
4.2.4 Effektive Turingmaschinen	103

4.2.5 Die universelle Turingmaschine abwandeln	106
4.2.6 Turingmaschinen mit adressierbarem Speicher	109
4.2.7 Turingmaschinen mit Assoziativspeicher	112
4.2.8 Eine Turingmaschine mit Einadreßbefehlen	114
4.3 Die Turingmaschine als Grundlage des praxisbrauchbaren Universalrechners	125
4.3.1 Wie die Turingmaschine rechnet, obwohl sie es nicht kann	125
4.3.2 Turingmaschinen, die rechnen können	130
4.3.3 Brauchbare Operationen	138
4.3.4 Der Übergang zum Einadreß-Universalrechner	140
4.3.5 Von der Turingmaschine zum Universalrechner	148
4.4 Vom Funktionszuordner zum Universalrechner	152
4.4.1 Der Funktionszuordner als alternative Modellvorstellung	152
4.4.2 Abschnittsweise Zuordnung	154
4.4.3 Wählbare Operationen	154
4.4.4 Gemeinsame Speicher – die Registermaschine	156
4.4.5 Die Universalmaschine	159
4.5 Turingmaschine und Funktionszuordner	160
5. Einfachmaschinen	163
5.1 Grundlagen	163
5.1.1 Entwicklungsziele	163
5.1.2 Wortlänge und Verarbeitungsbreite	164
5.1.3 Das Rechenwerk (die ALU)	165
5.1.4 Register in der Architektur, Register in der Schaltung	167
5.2 Einfache Einadreßmaschinen	169
5.2.1 Eine fiktive Einadreßmaschine	169
5.2.2 Die Befehlsformate der Einadreßmaschinen	170
5.2.3 Binäre und analytische Codierung	172
5.2.4 Befehlsabläufe	173
5.2.5 Die Einadreßmaschine mit Akkumulator	178
5.2.6 Akkumulator oder Arbeitsregister?	180
5.3 Stackmaschinen	181
5.3.1 Der Stack	181
5.3.2 Operationsbefehle	182
5.3.3 Die umgekehrte polnische Notation (RPN)	183
5.3.4 Stacks in der Rechnerarchitektur	186
5.3.5 Stacks implementieren	187
5.3.6 Einfache und komplexe Stackmaschinen	189
5.4 Die Einfachmaschine wird praxisbrauchbar	195
5.4.1 Adreßrechnung	195
5.4.2 Befehlsadressierung	209

5.4.3	Adreßverlängerung	210
5.4.4	Bankadressierung	213
5.4.5	Zugriffstabellen	222
5.4.6	Adreßverlängerung durch Erweiterung der Maschinenarchitektur	225
5.4.7	Adreßverlängerung außerhalb der Maschinenarchitektur	229
5.4.8	Unterprogramme	232
5.4.9	Unterbrechungen (Interrupts)	243
5.4.10	Ein- und Ausgabe	252
5.4.11	Weiterentwickelte Akkumulatormaschinen	257
6.	Architekturbeispiele	263
6.1	Maschinen aus Princeton	263
6.2	TX0	268
6.3	CDC 160	271
6.4	PDP-8	275
6.5	Data General Nova	281
6.6	CDC 1700	287
6.7	Mikroprogrammsteuerwerke und Mikrobefehle	293
6.7.1	Mikroprogrammsteuerwerke	293
6.7.2	Die Mikrobefehlsformate der Beispiele	297
6.7.3	Microdata Micro 800	299
6.7.4	IBM 360/25	305
6.7.5	Burroughs B1700	309
6.7.6	International Meta Systems MAX 2	319
7.	Neue Einfachmaschinen	329
7.1	Zeitgemäße Wirkprinzipien	329
7.1.1	Datenstrukturen	330
7.1.2	Befehlswirkungen	331
7.1.3	Adressierung und Adreßrechnung	332
7.2	Einfachmaschinen systematisch entwickeln	351
7.2.1	Mit der Turingmaschine beginnen	352
7.2.2	Ressourcenvektormaschinen	353
7.2.3	Ressourcenvektormaschinen bauen	358
7.2.4	Lange und kurze Befehle	364
7.3	Einfache Ressourcenvektormaschinen	373
7.3.1	Grundsätze der Architekturgestaltung	373
7.3.2	Informationsstrukturen	379
7.3.3	Der Ressourcenvektor	382
7.3.4	Praxisverbesserungen	385
7.3.5	Befehlsformate und Befehlswirkungen	386

7.4 Maschinen mit kurzen und langen Befehlen	400
7.4.1 Ganz kurze Befehle	400
7.4.2 Kurze und lange Befehle	405
7.4.3 Ganz lange Befehle	406
7.4.4 Kurze Befehle besser ausnutzen	407
7.4.5 Kurze Adressen besser ausnutzen	410
7.5 Maschinenbefehle und Mikrobefehle	412
7.5.1 Mikrobefehle in der Anwendungsprogrammierung	412
7.5.2 Mikrobefehle und RISC-Befehle	413
7.5.3 Vom Mikroprogrammsteuerwerk zum Universalrechner	414
7.5.4 Mikrobefehle als Maschinenbefehle	422
Anhang 1 Das C/Unix-Stackmodell	431
Anhang 2 Das x86-Registermodell	436
Anhang 3 Das Codierungsvermögen einfacher Befehlsformate	437
Anhang 4 Befehlswirkungen	440
4.1 Operationsbefehle mit zwei Operanden	440
4.2 Operationsbefehle mit einem Operanden	441
4.3 Verschieben und Rotieren im A-Register.....	441
4.4 Verschieben und Rotieren in A und MQ (64 Bits).....	441
4.5 Verschieben und Rotieren in A, AUX und MQ (96 Bits)	442
4.6 Verschieben und Rotieren des Operanden	442
4.7 Bitbefehle mit zwei Operanden	443
4.8 Bitbefehle mit einem Operanden.....	444
4.9 Spezielle Bitbefehle.....	445
4.10 Transportbefehle.....	446
4.11 Registerrechenbefehle	446
4.12 Verzweigen, Unterprogrammruf, Rückkehr.....	447
4.13 Steuerbefehle (Auswahl)	448
4.14 E-A-Befehle.....	449
4.15 16-Bit-Befehle mit 6-Bit-Operationscode (Kurzbefehle).....	449
4.16 Der Stackmechanismus	450
Literatur- und Quellenverzeichnis	453
Index.....	465