
Mikro- programmierung und Rechnerentwurf

von
Dr. Gerhard Chroust
IBM Laboratorium Wien

Inhalt

Vorwort der Herausgeber	9
Vorwort des Autors	11

A Einführung

1. Geschichtliche Entwicklung	17
1.1 Das klassische Steuerwerk	17
1.2 Systematischer Entwurf (Wilkes)	18
1.3 Das Heranwachsen der Mikroprogrammierung: Firmware	19
1.4 Schreibbare Speicher	21
1.5 Vertikale Verlagerung	22
1.6 Verwandte Gebiete	23
1.6.1 Mikroprozessoren und Mikroprogrammierung	23
1.6.2 Bit-Slice Architekturen	24
1.6.3 RISC Architekturen	25
2. Zusammenfassung	27
2.1 Schwerpunkte der Darstellung	27
2.2 Anwendungsmöglichkeiten	28
2.3 Vorausgesetztes Wissen	29

B Mikroprogrammierung und Rechnerentwurf

1. Die Idee der Mikroprogrammierung	33
1.1 Steuerwerk und Operationswerk	33
1.2 Das klassische Steuerwerk	39
1.3 Systematischer Entwurf: Mikroprogrammierung	42
1.4 Funktionale Bit	46
1.5 Firmware	48
2. Konzepte der Mikroprogrammierung	53
2.1 Kompatibilität	53
2.2 Interpretation und Übersetzung	59
2.3 Rechnerarchitektur	61
2.3.1 Der Begriff der Rechnerarchitektur	61
2.3.2 Gute Rechnerarchitektur	65
2.3.3 Komponenten von Rechnerarchitekturen	68
2.4 Mehrschichtige Rechnersysteme	74
2.4.1 Hierarchischer Aufbau von Rechnersystemen	74
2.4.2 Mehrfache Zwischenebenen	76
2.4.3 Existierende quellen-orientierte Zwischenebenen	83

2.5	Simulator und Emulator	85
2.6	Definitionen der Mikroprogrammierung	88
2.6.1	Existierende Definitionen	88
2.6.2	Integration und Relativierung der Definitionen	90
3.	Die Bewertung der Firmware	95
3.1	Vergleich mit Hardware-Lösungen	95
3.1.1	Vorteile im Entwicklungsprozeß	95
3.1.2	Vorteile im Betrieb	100
3.1.3	Vorteile in der Forschung	102
3.1.4	Vorteile in der Lehre	103
3.1.5	Nachteile gegenüber Hardware-Lösungen	106
3.2	Vergleich mit Software-Lösungen	107
3.2.1	Vorteile gegenüber Software-Lösungen	107
3.2.2	Nachteile gegenüber Software-Lösungen	108
3.3	Mikroprogrammierung durch den Benutzer	109
3.4	Benutzersichten der Mikroprogrammierung	112
4.	Verlagerung in die Firmware	115
4.1	Begriffsbestimmung	115
4.2	Vor- und Nachteile der Vertikalen Verlagerung	117
4.2.1	Vorteile der Vertikalen Verlagerung	117
4.2.2	Geschwindigkeitserhöhung	117
4.2.3	Nachteile und Probleme der Vertikalen Verlagerung	121
4.3	Grundmuster der Vertikalen Verlagerung	125
4.4	Identifizierung von verlagerungsfähigen Funktionen	127
4.4.1	Klassifizierung von verlagerungsfähigen Funktionen	127
4.4.2	Auswahl von verlagerungsfähigen abstrahierten Funktionen	128
4.4.3	Auswahl von verlagerungsfähigen de-facto-Funktionen	129
4.4.4	Ausschließungsgründe	130
4.5	Implementierung verlagerter Funktionen	131
4.6	Aufruf einer verlagerten Funktion	134
4.7	Laden der verlagerten Funktion	136
4.8	Verlagerung in verschiedenen Bereichen	136
4.8.1	Unterstützung von Betriebssystemfunktionen	137
4.8.2	Verlagerung in speziellen Anwendungsbereichen	144
4.8.3	Unterstützung höherer Programmiersprachen	147
4.9	Orthogonale Verlagerungen	149
4.9.1	Monitoring	150
4.9.2	Mikrodiagnose	156
5.	Die Mikroarchitektur des Steuerwerkes	161
5.1	Zeitliche Abarbeitung einer Mikroinstruktion	161
5.1.1	Das Steuerwortproblem	161

5.1.2	Der Abarbeitungszyklus einer Mikroinstruktion (DIL-Schleife)	164
5.1.3	Pipelining des Abarbeitungszyklusses	167
5.2	Adressierung der Mikroinstruktionen	169
5.3	Codierung des Operationsteiles	174
5.3.1	Codierungstechniken	176
5.3.2	Einsparung der Bitbreite bei fixem Programm	182
5.3.3	Horizontale und Vertikale Mikroprogrammierung	184
5.3.4	Quantitative Untersuchungen von Mikrocode	188
5.3.5	Nanoprogrammierung	191
5.4	Operandenadressierung	195
5.5	Wünschenswerte Eigenschaften von Wirtsarchitekturen	198
5.6	Speichertechnologie	199
6.	Emulationstechniken	205
6.1	Entwurf eines Emulators	205
6.2	Struktur eines Emulators	209
6.2.1	Speicherabbildung	209
6.2.2	Emulation von Maschinenbefehlen	212
6.2.3	Emulation von Systemunterbrechungen	215
6.2.4	Emulation der Eingabe/Ausgabe	215
6.3	Emulation mehrerer Rechnerarchitekturen	217
6.3.1	Rechnereigener Modus und Emulations-Modus	217
6.3.2	Klassifizierung von Emulatoren	217
6.3.3	Cross-System Emulatoren	219
6.3.4	Asymmetrischer Wirt und Universeller Wirt	219
6.3.5	Alleinstehende und Integrierte Emulation	220
6.3.6	Dynamische Mikroprogrammierung	221
6.4	Direkte Ausführung höherer Programmiersprachen	222
7.	Firmware Engineering	225
7.1	Firmware Engineering - eine neue Disziplin?	225
7.1.1	Abgrenzung	225
7.1.2	Die Komplexität der Mikroprogrammierung	228
7.1.3	Mikroprogrammierung durch den Benutzer	230
7.1.4	Firmware-Lebenszyklus und Entwicklungspfade	231
7.2	Entwicklung eines Mikroprogrammes	231
7.2.1	Anforderungsanalyse	232
7.2.2	Spezifikation	233
7.2.3	Entwurf und Implementierung	235
7.2.4	Übersetzung von Mikroprogrammiersprachen	246
7.3	Die Entwicklung der Dokumentation	255
7.4	Die Qualitätssicherung der Firmware	256
7.4.1	Statische Analyse	257

7.4.2	Statische Qualitätsmessungen	257
7.4.3	Formale Korrektheitsbeweise	259
7.4.4	Empirische Testverfahren	262
7.4.5	Abstrakte Programmausführung	265
7.4.6	Fehlertolerante Systeme	266
7.5	Das Umfeld eines Produktes	267
7.6	Entwicklungssysteme	267
8.	Entwicklungstendenzen	273
8.1	Heutiger Stand der Technik	273
8.2	Trends	274

C Anhang

1.	Literaturempfehlungen	281
1.1	Buchempfehlungen	281
1.2	Glossare	286
1.3	Periodische Fachzeitschriften und Konferenzen	286
1.4	Fachvereinigungen	287
2.	Literaturverzeichnis	288
3.	Register	338