

Inhaltsverzeichnis

Kapitel 0	Einleitung	1
0.1	Verschiedene Ansätze beim Vergleich von Architekturen	1
0.2	Ein Ansatz zum Vergleich von Architekturkonzepten	4
Kapitel 1	Modellbildung	6
1.1	Das Architekturmodell	6
1.1.1	Maschinensprache, Hochsprachen und Compiler	
1.1.2	Die Hardware	
1.2	Dhrystone, ein Modell für Workload	25
1.2.1	Eine allgemeine Analyse des Dhrystone-Benchmarks	
1.2.2	Analysen für RISC-Architekturen	
1.2.3	Analysen für CISC-Architekturen	
1.3	Das Gütemaß	38
1.4	Einige wichtige Schaltkomponenten	39
1.4.1	Multiplexer	
1.4.2	Verschiedene n-bit Inkrementierer	
1.4.3	Verschiedene n-bit Addierer	
1.4.4	Der Wallace-Tree Multiplizierer	
1.4.5	Merges	
1.4.6	Die Shifter	
Kapitel 2	Entwicklung eines guten Vertreters für RISC-Architekturen (Von-Neumann-Bauart)	49
2.1	Eine sehr einfache RISC-Architektur (R-1.0)	49
2.1.1	Spezifikation der Hardware und der Maschinensprache	
2.1.2	Spezifikation der Compilerstrategien	
2.1.3	Bestimmung der Laufzeit	
2.1.4	Auswertung der Architektur	
2.2	Übergang zu general-purpose Registern (R-2.0)	62
2.2.1	Maschinensprache und die sie realisierende Hardware	
2.2.2	Änderungen an den Compilerstrategien	
2.2.3	Vergleich der beiden Architekturen R-1.0 und R-2.0	
2.3	Ein CPU-Register für den Frame-Pointer (R-2.1)	68
2.3.1	Auswirkungen auf den Compiler	

2.3.2	Auswirkungen der Compileränderungen auf die Güte	
2.4	Verbesserungen der Stack-Zugriffe (R-2.2)	71
2.4.1	Auswirkungen auf den Compiler	
2.4.2	Auswirkungen der Compileränderungen auf die Güte	
2.5	Optimierung der Anzahl der CPU-Register (R-2.3, R-2.4)	75
2.5.1	Änderungen an der Hardware	
2.5.2	Änderungen am Compiler	
2.5.3	Vergleich der verschiedenen Architektur-Varianten	
2.6	Auswirkungen von Nonalignment	89
2.6.1	Komprimierung der Strings	
2.6.2	Daten-Nonalignment	
2.6.3	Nonalignment auf den Instruktionen	
2.7	Zusammenfassung der Ergebnisse	96
Kapitel 3	Entwicklung eines guten Vertreters für CISC-Architekturen	
	(Von-Neumann Bauart)	102
3.1	Mikroprogrammierte oder nanoprogrammierte Kontrolle	102
	(C-1.0, C-2.0)	
3.1.1	Spezifikation der Maschinsprache und der Datenpfade	
3.1.2	Die mikroprogrammierte Kontrolllogik	
3.1.3	Die nanoprogrammierte Kontrolllogik	
3.1.4	Auswertung und Vergleich der beiden Architekturen	
3.2	Einschränkung des Instruktionssatzes (C-1.1, C-2.1)	125
3.2.1	Die eingeschränkte Maschinsprache	
3.2.2	Auswirkungen auf die Kontrolllogik	
3.2.3	Auswirkungen auf die Güte	
3.3	Einschränkung der Registeranzahl (C-1.2, C-2.2)	133
3.3.1	Änderungen am Compiler	
3.3.2	Auswirkungen auf das Instruktionsformat und die Hardware	
3.3.3	Auswirkungen auf die Güte	
3.4	Änderungen in den Datenpfaden	136
3.4.1	Entfernen des Shifters der Addierer-Erweiterung	
3.4.2	Einschränken der 32-Bit Shifter	
3.4.3	Entfernen der Addierer-Erweiterung	
3.4.4	Einschränken der Hilfsregister	
3.5	Zusammenfassung der Ergebnisse	144
3.5.1	Die Wahl der Addierer, Inkrementierer und der Realisierung der Multiplikation	
3.5.2	Gute Vertreter für die CISC-Architekturen	
3.5.3	Überblick über die Gewinne der CISC-Architekturen	

Kapitel 4	Vergleich der RISC- und CISC-Architekturen (Von-Neumann Bauart)	151
4.1	Vergleich und Analyse	151
4.1.1	Gütevergleich	
4.1.2	Herausstellen der Unterschiede	
4.1.3	Analysieren der Unterschiede	
4.2	Synthese einer RCISC-Architektur (RC-1.0)	167
4.2.1	Spezifikation der Maschinsprache und der Datenpfade	
4.2.2	Steuerung der Datenpfade und des Speichersystems	
4.2.3	Die programmierte Kontrolllogik	
4.2.4	Compiler und Bilanz	
Kapitel 5	Optimierung beim Prozedurhandling	180
5.1	Code-Inlining bei Prozeduren und Funktionen	181
5.1.1	Möglichkeiten des Inlinings bei Dhrystone	
5.1.2	Eine Strategie für die Codeanpassung	
5.1.3	Auswirkungen auf die Compilation	
5.1.4	Auswirkungen auf die Güte der drei Architekturen	
5.2	Zusätzliche Register für die Parameter	192
5.2.1	Die Anzahl der Register für die Parameter	
5.2.2	Änderungen an der Hardware	
5.2.3	Auswirkungen auf die Compilation	
5.2.4	Auswirkungen auf die Güte	
Kapitel 6	Übergang zu Harvard-Architekturen	210
6.1	Delayed Branch mit NOP's	211
6.1.1	Änderungen der Datenpfade	
6.1.2	Änderungen in der Kontrolllogik	
6.1.3	Laufzeitänderungen	
6.1.4	Auswirkungen auf die Güte	
6.2	Delayed Branch mit Optimierung	221
6.2.1	Eine Optimierungsstrategie	
6.2.2	Optimierungen bei der RCISC-Architektur	
6.2.3	Optimierungen bei den Architekturen CISC und RCISC	
6.2.4	Auswirkungen der Optimierung auf die Güte	
6.3	Branch-Target Cache	227
6.3.1	Funktionsweise und Aufbau des Branch-Target Caches	
6.3.2	Geänderte Instruktionsformate	
6.3.3	Die neue Laufzeit des Benchmarks	
6.3.4	Auswirkungen auf die Güte	

Kapitel 7	Zusammenfassung und Ausblick	238
7.1	Zusammenfassung der Ergebnisse	238
7.1.1	Die Auswirkungen der drei wichtigsten Parameter auf die Architekturen	
7.1.2	Ergebnisse beim Analysieren einiger Designtricks	
7.2	Die Auswirkungen unterschiedlicher Analysemethoden	241
7.3	Ansatzpunkte für weitere Forschungen	243

Anhänge

Anhang A	Ergänzungen zur Dhrystone-Analyse	250
A.1	Codeerzeugung für RISC-Architekturen	250
A.2	Codeerzeugung für CISC-Architekturen	253
Anhang B	Ergänzungen zu den RISC-Architekturen aus Kapitel 2	255
B.1	Angaben zur Architektur R-1.0	255
B.1.1	Die Gleichungen der Kontrollsignale	
B.1.2	Timinganalyse für die Maschineninstruktionen	
B.1.3	Code für die kritischen C-Instruktionen	
B.1.4	Code für die speziellen Compiler-Sequenzen	
B.2	Angaben zur Architektur R-2.0	261
B.2.1	Die Gleichungen der Kontrollsignale	
B.2.2	Timinganalyse der Maschineninstruktionen	
B.2.3	Code für die kritischen C-Instruktionen	
B.2.4	Code für die speziellen Compilersequenzen	
B.3	Angaben zu den Architekturen R-2.1 und R-2.2	267
B.3.1	Codeänderungen für die Architektur R-2.1	
B.3.2	Codeänderungen für die Architektur R-2.2	
B.4	Angaben zu den Architekturen R-2.3 und R-2.4	269
B.4.1	Kodierung der Maschinsprache und Steuerung der Hardware	
B.4.2	Timinganalyse für die CPU mit 3-Adreß-Registerfile	
B.4.3	Codeänderungen für die Architektur R-2.3	
B.4.4	Codeänderungen für die Architektur R-2.4	

Anhang C Ergänzungen zu den CISC-Architekturen aus Kapitel 3	279
C.1 Angaben zur Architektur C-1.0	279
C.1.1 Der Mikrocode	
C.1.2 Die Laufzeiten der Maschineninstruktionen	
C.2 Angaben zur Architektur C-2.0	295
C.2.1 Mikrocode und Nanocode	
C.2.2 Die Laufzeiten der Maschineninstruktionen	
C.3 Angaben zu den Architekturen mit eingeschränktem Instruktionssatz	305
C.3.1 Die Architektur C-1.1	
C.3.2 Die Architektur C-2.1	
C.4 Angaben zu den Architekturen mit reduzierten Datenpfaden	308
C.4.1 Einschränkungen der Shift-Operationen der ALU	
C.4.2 Entfernen der Addierer-Erweiterung	
C.4.3 Entfernen des Hilfsregisters ZO	
Anhang D Ergänzungen zu den Architekturen aus Kapitel 4	314
D.1 Angaben zur CISC-Architektur C-ls	314
D.1.1 Die Mikro-Codes der Architektur C-ls	
D.1.2 Simulation einiger CISC-Instruktionen	
D.1.3 Die Laufzeiten der neuen Instruktionen	
D.2 Angaben zur CISC-Architektur C-rm	317
D.2.1 Neue Mikro- und Nano-Codes und ihre Laufzeiten	
D.2.2 Neue Codes für Array- und Strukturzugriffe	
D.3 Angaben zur RCISC-Architektur	319
D.3.1 Die Mikro- und Nano-Codes	
D.3.2 Änderungen bei der Übersetzung der C-Instruktionen	
D.3.3 Gleichungen der Kontrollsignale	
D.3.4 Laufzeiten der Maschineninstruktionen	
Anhang E Ergänzungen zu den Architekturen aus Kapitel 5	329
E.1 Angaben zum Code-Inlining bei Prozeduraufrufen	329
E.1.1 Änderungen am Programmtext	
E.1.2 Änderungen in den Bilanzen	
E.2 Angaben zu den Architekturen mit Parameterregistern	336
E.2.1 Laufzeitänderungen bei der RISC-Architektur durch überlappende Registerfenster	
E.2.2 Laufzeitänderungen bei CISC und RCISC durch überlappende Registerfenster	
E.2.3 Änderungen der Bilanzen durch Parameter-Register	

Anhang F Ergänzungen zu den Harvard-Architekturen	345
F.1 Änderungen in der Kontrolle und den Laufzeiten	345
F.1.1 Steuerung und Timinganalyse des RISC-Registerfiles	
F.1.2 Steuerung der RISC-Architektur	
F.1.3 Timinganalyse der Harvard-RISC	
F.1.4 Steuerung der CISC-Architektur	
F.1.5 Laufzeiten der CISC-Instruktionen	
F.1.6 Lange Instruktionsfetches bei CISC	
F.1.7 Steuerung der RCISC-Architektur	
F.1.8 Laufzeiten der RCISC-Instruktionen	
F.2 Optimierung der delay slots	356
F.2.1 Übersetzungsschemata	
F.2.2 Handkodierte Routinen und Anweisungen	
F.2.3 Delay slots mit Kontextabhängigkeiten	
Anhang G Der Programmtext des Dhrystone-Benchmarks	365
G.1 Der Quelltext des Dhrystone-Programmes (C / 1.1)	365
G.2 Der modifizierte Programmtext des Dhrystone-Benchmarks	371
Literaturverzeichnis	375