

Inhaltsverzeichnis

1	Einleitung	1
1.1	Überblick	1
1.2	Problembeschreibung und Vorgehensweise	3
2	RISC-Architekturmerkmale unter dem Aspekt zeitlicher Randbedingungen	6
2.1	Echtzeitdatenverarbeitung in embedded-Systemen	6
2.1.1	Allgemeine Eigenschaften von Echtzeitsystemen	6
2.1.2	Softwarestruktur in Echtzeitsystemen	9
2.2	Kennzeichen von embedded-Systemen und ihren Anwendungen	13
2.2.1	Spezifikation der Hardwarekomponenten	14
2.2.2	Analyse der Softwareeigenschaften	16
2.3	Architekturmerkmale von RISC-Prozessoren	20
2.3.1	Pipelining	23
2.3.1.1	Befehlspipelining bei RISC-Prozessoren	24
2.3.1.2	Maschinenzykluszeit bei Befehlspipelining	25
2.3.1.3	Kenngrößen des Befehlspipelining	27
2.3.2	Befehlssatz	27
2.3.2.1	Umfang des Befehlssatzes und Befehlsformate	28
2.3.2.2	Strategien zur Vermeidung befehlsbedingter suboptimaler Pipelineauslastung	29
2.3.3	Load/Store-Architektur	33
2.3.3.1	Vermeidung von Pipeline-Interlocks durch Speicherbefehle	34
2.3.3.1.1	Implementierung der Befehlspipeline	35
2.3.3.1.2	Struktur des Speichersystems	37
2.3.4	Registerstruktur	38
2.3.4.1	Registerverwaltung durch Software	39
2.3.4.2	Hardwareunterstützung für die Registerverwaltung	41
3	Die Kommunikation zwischen RISC-Prozessoren und ihrer Umgebung	44
3.1	Zeitkomponenten der Peripheriekommunikation	44
3.1.1	Zeitkomponenten bei der Datenauswahl	45
3.1.1.1	Auswahl von Operanden	45
3.1.1.2	Auswahl von Befehlsworten	50
3.1.2	Zeitkomponenten beim Datenzugriff	54
3.2	Speicherhierarchie in kleinen embedded-Systemen	57

3.3	Zugriffsverfahren auf I/O-Komponenten	61
3.3.1	Speicherbezogene Auswahl von I/O-Komponenten	62
3.3.2	Isolierte Auswahl von I/O-Komponenten	65
3.3.3	Vergleich von speicherbezogener und isolierter Auswahl in RISC-Prozessoren	67
4	Optimierung isolierter I/O-Kommunikation in RISC-Prozessoren	69
4.1	Individuelle I/O-Schnittstellen	70
4.1.1	Quasi-Load/Store	71
4.1.2	Impliziter Datentransfer	73
4.1.3	Integriertes I/O-System	74
4.2	Realisierung isolierter I/O-Auswahl in einer aktuellen RISC-Architektur	76
4.2.1	Implementierung einer Quasi-Load/Store-Schnittstelle	78
4.2.1.1	Adressierung über alternative Adreßräume	78
4.2.1.2	Implementierung über zusätzliche Befehle	82
4.2.2	Realisierung von Impliziter und Integrierter Schnittstelle	87
4.2.2.1	Registerüberlagerung	87
4.2.2.2	Modifikation der Befehlscodierung	95
5	Bewertung von Mechanismen zur I/O-Kommunikation bei RISC-Prozessoren	100
5.1	Methode zur Bewertung einer Optimierung von I/O-Zugriffen	103
5.1.1	Definition geeigneter Größen zur Bewertung der Effizienz	103
5.1.2	Verhaltensmodellierung der Hardware	106
5.1.3	Lastmodellierung	110
5.1.4	Simulationsverfahren	113
5.2	Bewertungsergebnisse	115
5.2.1	Einfluß des Datentyps der I/O-Operanden auf den Effizienzgewinn	116
5.2.2	Abhängigkeit der Effizienz von der Datenstruktur und vom zurückgelegten Datenweg der I/O-Operanden	128
5.3	Schlußfolgerungen aus den Bewertungsergebnissen	149
5.3.1	Maßnahmen zur Reduzierung von Datentyp- und Datenwegabhängigkeiten	149
5.3.2	Unterstützung modifizierter I/O-Schnittstellen durch Hochsprachencompiler	155
6	Schlußbetrachtung und Ausblick	166

Literaturverzeichnis	168
Anhang	175
A Liste der Benchmarkfunktionen zur Lastmodellierung	176
A.1 Grundfunktionen	176
A.2 Verbundfunktionen	177
A.3 Elementare Algorithmen	178
A.3.1 Digitale Signalverarbeitung	178
A.3.2 Statistische Funktionen	179
A.3.3 Funktionen zum Suchen und Sortieren	179
A.4 Komplexe Algorithmen	180
A.4.1 Echtzeit-Algorithmen für nichtlineare dynamische Systeme	180
A.4.2 Motorsteuerungen	181
A.4.3 Sprach- und Bildverarbeitung	181
A.4.4 Transformation von Roboterkoordinaten	182