

Dipl.-Ing. Johannes Kneip, Barsinghausen

**Objektorientierte
Cache-Speicher für
programmierbare
monolithische
Multiprozessoren in der
digitalen Bildverarbeitung**

Reihe **9**: Elektronik

Nr. **267**

Inhalt

1	Einleitung	1
1.1	Trends in der digitalen Bildverarbeitung	1
1.2	Bedeutung der Speicherarchitektur für den Architekturentwurf	3
1.3	Ziele und Aufbau der Arbeit	5
2	Cachespeicher: Prinzipien, Realisierung und Bewertung	7
2.1	Motivation des Cache-Grundgedankens	7
2.2	Cachearchitekturen	10
2.3	Leistungsbewertung und Optimierung von Cache-Architekturen	13
2.3.1	Designparameter einer Cachearchitektur	13
2.3.2	Optimierungsziele	14
2.3.3	Leistungsmessung	15
2.4	Zusammenfassung Cachegrundlagen	17
3	Bildverarbeitungsalgorithmen und deren parallele Implementierung	18
3.1	Charakteristische Eigenschaften von Bildverarbeitungsalgorithmen	19
3.1.1	Klassifizierung von Bildverarbeitungsalgorithmen	19
3.1.2	Datenzugriffe bei Low-Level-Algorithmen	20
3.1.3	Datenzugriffe bei Medium-Level-Algorithmen	21
3.1.4	Datenzugriffe bei High-Level-Algorithmen	22
3.2	Parallele Implementierung von Bildverarbeitungsalgorithmen	23
3.2.1	Parallelitätsarten und Implementierungsebenen	23
3.2.2	Parallelität auf Funktionsebene	24
3.2.3	Parallelität auf Chipebene	25
3.2.4	Parallelität auf Systemebene	26
3.2.5	Ergebnisse	26
3.3	Parallelisierungsstrategien und resultierende Zugriffsmuster	27
3.3.1	Parallelisierungsstrategien auf Datenebene	27
3.4	Mathematische Beschreibung der Lokalitätseigenschaften	30
3.4.1	Zeitliche und örtliche Lokalität bei skalarem Zugriff	31
3.4.2	Zeitliche und örtliche Lokalität bei parallelem Zugriff	35
3.5	Zusammenfassung Algorithmeigenschaften und Parallelverarbeitung	36
4	Speicherkonzepte existierender Bildverarbeitungsprozessoren	38
4.1	Grundsätzliche On-Chip-Speicherkonzepte	39
4.1.1	Direktgekoppelte Prozessorarchitekturen mit lokalen Speichern	40
4.1.2	Speichergekoppelte Prozessorarchitekturen	40
4.2	Wechselwirkung Prozessor-Controlling und Speicherarchitektur	42
4.2.1	Cachekohärenz	43
4.2.2	Verhinderung und Auflösung von Konflikten	46
4.3	Vorstellung und Klassifikation bekannter Architekturen	47
4.3.1	Texas TMS320C80: Konfliktauflösende MIMD-Architektur	49

4.3.1.1	Überblick	49
4.3.1.2	Adreßrechner und Zugriffskonfliktauflösung	50
4.3.1.3	Speicherverwaltung	50
4.3.2	HiPAR–DSP: Konfliktvermeidende SIMD–Architektur	51
4.3.2.1	Überblick	51
4.3.2.2	Adreßrechner, Adreßtransformation und Konfliktvermeidung	53
4.3.2.3	Speicherverwaltung	54
4.4	Diskussion und Bewertung existierender Speicherarchitekturen	57
4.4.1	Erfüllung der Algorithmenanforderungen	57
4.4.2	Architektur Aspekte	57
4.4.3	Benutzer– und Compilerfreundlichkeit	59
4.5	Zusammenfassung der Anforderungen an einen Cache mit parallelem Zugriff	60
4.5.1	Anforderungen hinsichtlich der Algorithmeigenschaften	60
4.5.2	Anforderungen aus Prozessor– und Speicherarchitektur	60
4.5.3	Anforderungen von Programmierer und Compiler	61
4.5.4	Einschränkungen und Abgrenzungen des Speicherkonzepts	61
5	Objektorientierte Speicherarchitektur	62
5.1	Konzept des objektorientierten Caches	62
5.2	Objektorientierte Caches im Hardware–Software System	65
5.2.1	Parallelen zum objektorientierten Softwaredesign	65
5.2.2	Betrachtung aus Sicht der Prozessorarchitekturentwicklung	67
5.2.3	Vergleich mit dem klassischen Prinzip des virtuellen Speichers	68
5.3	Datenverteilung und Cache–Organisation	70
5.3.1	Nomenklatur	71
5.3.2	Logische Gliederung der internen Speicherbänke	72
5.3.3	Datenverteilung auf die Speicherbänke	73
5.3.3.1	Matrixzugriff im zweidimensionalen Adreßraum	73
5.3.3.2	Vektorzugriff im zweidimensionalen Adreßraum	76
5.3.3.3	Vektorzugriff im eindimensionalen Adreßraum	78
5.3.3.4	Wahlfreier, unabhängiger Zugriff der Verarbeitungseinheiten auf private Daten	79
5.3.4	Zusammenfassung Datenverteilung und Cacheorganisation	80
5.4	Adreßrechnung	80
5.4.1	Anforderungen an den Adreßrechner	81
5.4.2	Realisierungsalternativen und deren Bewertung	82
5.4.3	Praktische Realisierung des Adreßrechners	83
5.4.4	Adreßwortbreiten	85
5.4.5	Zentraler Adreßrechner mit Post–Modify	86
5.4.6	Ergebnisse	87
5.4.7	Zusammenfassung Adreßberechnung	87
5.5	Blockauswahl	88
5.5.1	Blockzuordnungsstrategien im objektorientierten Cache	88
5.5.2	Zusammenfassung Blockauswahl	91
5.6	Hit–/Misserkennung	92
5.6.1	Problematik der Hit–/Misserkennung bei einem Speicher mit Parallelzugriff	92

5.6.2	Lösungsansatz und –alternativen	92
5.6.3	Sonderfall des wahlfreien Zugriffs	95
5.6.4	Größe der notwendigen Tags	96
5.6.5	Zusammenfassung Hit–/Misserkennung	96
5.7	Realisierung der Speichercrossbars	97
5.7.1	Komplexität und Timing der Daten– und Adreßcrossbars	99
5.7.2	Zusammenfassung Crossbars	100
5.8	Datentransfer zwischen Cache und externem Hauptspeicher	101
5.8.1	Anordnung der virtuellen Objekte im Hauptspeicher	101
5.8.2	Aufbau des I/O–Controllers	102
5.8.3	Beschleunigung des Datentransfers zwischen Cache und Hauptspeicher	103
5.8.4	Spekulatives Vorladen zur Verringerung der Miss–Rate	105
5.8.5	Zusammenfassung I/O–Controller	106
5.9	Integrationsfähigkeit in einen Hochsprachencompiler	106
5.10	Zusammenfassung: Realisierung der Cachearchitektur	107
6	Realisierungsbeispiel und Ergebnisse	109
6.1	Realisierungsbeispiel und technische Daten	110
6.2	Bewertung des Hardwareaufwands	110
6.2.1	Einfluß der geringeren RAM–Bankzahl	112
6.2.2	Vergleich des Logik– und Registeraufwands	112
6.3	Bewertung des Hit–/Missverhaltens	113
6.3.1	Hit–/Missverhalten für Low– und Medium–Level Algorithmen	114
6.3.2	Speichernutzungsgrad	118
6.3.3	Einfluß des spekulativen Vorladens	118
6.4	Bewertung anhand eines komplexen Bildverarbeitungsverfahrens	120
6.5	Gesamtbewertung	122
7	Ausblick	124
7.1	Erweiterung der möglichen Zugriffsmuster	124
7.1.1	Andere Zugriffsformen auf ein– oder zweidimensionale Adreßräume	124
7.1.1.1	Quasireguläre Zugriffe mit zeitlicher Konfliktauflösung	124
7.2	Verwendung des Caches in teilsynchronisierten Architekturen	125
7.3	Erweiterung auf Instruktionen	127
7.4	Nutzung zusätzlicher Objekteigenschaften	127
7.5	Objektorientierte Prozessorarchitektur	128
8	Zusammenfassung	130
	Anhang	133
A	Realisierung der Divisions– und Modulooperation	133
A.1	Standard–Dividierer für positive Festkommazahlen	133
A.2	Eigenschaften der Nachkommastellen bei Division durch $2^n + 1$	134
A.3	Parallele Division durch $2^n + 1$	136

B	Crossbar-Realisierung	143
C	Load/Store – Befehlssatz eines Prozessors mit objektorientiertem Cache	145
D	Listenbasierte Hough-Transformation als Beispiel eines komplexen Bildverarbeitungsverfahrens	147
D.1	Gradientenbasierte Hough-Transformation für Geraden	148
D.2	Listenbasierte Implementierung	148
D.3	Besonderheiten beim Einsatz des objektorientierten Caches	150
	Literaturverzeichnis	151