

Dipl.-Ing. Winfried Gehrke, Neu Wulmstorf

**Assoziatives Controlling
von programmierbaren
Parallelprozessoren für die
Videosignalverarbeitung**

Reihe **10**: Informatik/
Kommunikationstechnik Nr. **430**

Inhalt

1	Einleitung	1
1.1	Bisherige Architekturansätze für die Videosignalverarbeitung	2
1.2	Ziele und Aufbau der Arbeit	4
2	Modellierung von Verfahren der digitalen Videosignalverarbeitung	6
2.1	Klassifikation der Algorithmen der Videosignalverarbeitung	6
2.1.1	Low-Level-Algorithmen	7
2.1.2	Medium-Level-Algorithmen	7
2.1.3	High-Level-Algorithmen	8
2.2	Eigenschaften von Verfahren der Videosignalverarbeitung	8
2.3	Algorithmenmodell für Verfahren der digitalen Videosignalverarbeitung	10
2.3.1	Benchmark FE, "Fahrbahnranderkennung"	12
2.3.2	Benchmark HC, "Hybridcodierung"	13
2.3.3	Benchmarks LL und HL für low-level-orientierte und high-level-orientierte Verfahren	15
3	Klassifikation von parallelen VLSI-Architekturen für die digitale Bildverarbeitung	18
3.1	SIMD-Architekturen	21
3.1.1	Architekturbeispiel: CAP III	22
3.1.2	Architekturbeispiel: HiPAR-DSP	23
3.2	MIMD-Architekturen	24
3.2.1	Architekturbeispiel: MVP	26
3.3	MSIMD-Architekturen	27
3.3.1	Architekturbeispiel: AxPe640V	28
3.4	SPMD-Architekturen	29
3.4.1	Architekturbeispiel: ISMP	31
3.5	Modellbasierte Bewertung der Architekturalternativen	32
3.5.1	Abschätzung der Rechenzeit	33
3.5.1.1	Rechenleistung der Controlling-Alternativen für die Verarbeitung eines Algorithmus	34
3.5.1.2	Zeitgleiche Verarbeitung unterschiedlicher Algorithmen	35
3.5.1.3	Pipeline-Verluste bei datenabhängigen Verzweigungen	37
3.5.2	Abschätzung des schaltungstechnischen Aufwandes	40
3.5.3	Effizienzanalyse	42
3.6	Schlußfolgerungen	45
4	Assoziatives Controlling von Parallelprozessorarchitekturen	48
4.1	Grundkonzept des Assoziativen Controllings	48
4.2	Assoziatives Controlling nach Potter	51
4.3	Verallgemeinertes Architekturmodell eines assoziativ gesteuerten Parallelprozessors	52
4.4	Zuordnung der Instruktionsströme zu den Datenpfaden	54
4.4.1	Explizite Zuordnung	54
4.4.1.1	Ausnutzung des Instruktionsspeichers	56
4.4.1.2	Verzögerungen bei datenabhängigen Verzweigungen	57
4.4.2	Implizite Zuordnung	60
4.5	Verarbeitung von Cache-Misses	62
4.6	Architekturerweiterung der Datenpfade zur Unterstützung der zeitgleichen Verarbeitung unterschiedlicher Algorithmen	64
4.7	Adressierung der Instruktionsströme	65
4.7.1	Adressierung der Instruktionsströme durch separate Adressierungseinheiten	66
4.7.2	Adressierung der Instruktionsströme durch eine singuläre Adressierungseinheit in Verbindung mit einem Umordnungsnetzwerk	66

4.7.3	Vergleich der Alternativen	68
5	Dynamisches Assoziatives Controlling	70
5.1	Motivation und Grundprinzip des Ansatzes	70
5.2	Basisarchitektur eines dynamisch assoziativ gesteuerten Prozessors	74
5.3	Verarbeitung von grundlegenden Programm–Konstrukten	76
5.3.1	Sequentieller Programmcode	76
5.3.2	Bearbeitung von datenunabhängigen Sprüngen	76
5.3.3	Datenabhängige Aufspaltung des Kontrollflusses	76
5.3.4	Bearbeitung von Unterprogrammaufrufen	77
5.3.5	Programmschleifen	77
5.3.6	Explizite Synchronisation von Instruktionsströmen	80
5.4	Aufbau der Abhängigkeitsliste	81
5.5	Verarbeitung der Abhängigkeitsliste durch die Steuereinheit	85
5.5.1	Architektur der Steuereinheit eines dynamisch assoziativ gesteuerten Prozessors	85
5.5.2	Abarbeitung der Abhängigkeitsliste	88
5.5.3	Start von Nachfolger–Blöcken	89
5.5.4	Verarbeitung von Daten–Cache–Misses	91
5.5.5	Abbruch der Verarbeitung nicht ausgeführter Basic–Blöcke durch den Scheduler	92
5.6	Architektur des Schedulers	93
5.6.1	Bestimmung der erforderlichen Größe des Nachfolger–FIFOs	96
5.6.2	Bestimmung der erforderlichen Größe des Cache–Miss–FIFOs	97
5.6.3	Abschätzung des schaltungstechnischen Aufwands für die Implementierung des Schedulers	97
6	Modellbasierte Effizienzanalyse des Assoziativen Controllings	99
6.1	Modellierung des Statischen Assoziativen Controllings	99
6.1.1	Modellierung des schaltungstechnischen Aufwandes	99
6.1.2	Modellierung der Rechenleistung	100
6.1.3	Vergleich des Statischen Assoziativen Controllings mit bisherigen Controlling–Ansätzen	101
6.2	Modellierung der Rechenleistung unter Berücksichtigung von Daten–Cache–Misses	105
6.2.1	Verfahrensmodell zur Rechenleistungsanalyse	106
6.2.2	Modellierung der Rechenzeit unter Berücksichtigung von Cache–Misses	107
6.2.3	Effizienz des Assoziativen Controllings unter Berücksichtigung von Cache–Misses	111
6.3	Zusammenfassende Bewertung	117
7	Ausblick	119
7.1	Assoziatives Controlling für heterogene Parallelprozessorarchitekturen	119
7.2	Unterstützung von schaltungstechnischer Redundanz	120
8	Zusammenfassung	122
Anhang		125
A	Modellierung der Rechenzeit bei Einsatz der Split–And–Merge–Strategie	125
B	Bestimmung der Modellparameter für die verwendeten Benchmarks	127
B.1	Architekturmodell für die Ableitung der Modellparameter	127
B.2	Bestimmung der Modellparameter für den Benchmark FE	129
B.2.1	Gradientenfilterung	129
B.2.2	Gradientenbetragsbildung	130
B.2.3	Binarisierung	131
B.2.4	Konturverdünnung	132

B.2.5	Konturverifikation	132
B.2.6	Hough-Transformation	133
B.2.7	Geradenextraktion	135
B.3	Bestimmung der Modellparameter für den Benchmark HC	135
B.3.1	Bewegungsschätzung (Blockmatching)	136
B.3.2	Bestimmung des Prädiktionsfehlers und Rekonstruktion	138
B.3.3	Diskrete Cosinus Transformation	138
B.3.4	Quantisierung	140
B.3.5	Inverse Quantisierung	141
B.3.6	Variable-Längen-Codierung	141
B.3.7	Variable-Längen-Decodierung	143
C	Erweiterte Modellierung der Architektureffizienz	145
C.1	Ergebnisse der Effizienzanalyse ohne Berücksichtigung von Cache-Misses	145
C.2	Ergebnisse der Effizienzanalyse unter Berücksichtigung von Cache-Misses	153
	Literaturverzeichnis	162