

Inhaltsverzeichnis

1	Einleitung	3
1.1	Stand der Technik	5
1.2	Aufbau der Arbeit	8
2	Entwicklungsumgebung für VLIW-Übersetzer	9
2.1	Prozessormodellierung	14
3	Scheduling	24
3.1	List-Scheduling	26
3.2	Code-Anordnung als Scheduling-Problem	28
4	Software-Pipelining	33
4.1	Pipeline-Schedules für azyklische Graphen	35
4.2	Pipeline-Schedules für zyklische Graphen	37
4.3	Anordnung von stark zusammenhängenden Teilgraphen	42
4.4	Berechnung der Mindestdistanzen	43
4.5	Vorbereitungen für das Software-Pipelining	47
4.6	Erzeugung des ausführbaren Schleifencodes	50
4.7	Software-Pipelining am Beispiel	53
5	Algorithmen zur Berechnung von Pipeline-Schedules	55
5.1	Ausgewählte Programme zur Bewertung der Verfahren	55
5.2	Das Grundverfahren	56
5.3	Pipeline-Schedules nach Rau/Gläser (RG)	58
5.4	Pipeline-Schedules nach Lam (LAM)	61
5.5	Pipeline-Schedules mit dem Prioritätsverfahren (PRIOR)	66
5.6	Ressource-Sensitives Software-Pipelining (RSSP)	71
5.7	Pipeline-Schedules nach Huff (SLACK)	74
5.8	Scheduling und Lineare Programmierung (ILP)	80
6	Vergleich der Scheduling-Algorithmen	84
6.1	Laufzeitgewinn durch Pipelining für unterschiedliche Prozessoren	85
6.2	Relative Güte der Schedules	87
6.3	Rechenaufwand der Scheduling-Verfahren	89

7	Zuteilung funktionaler Einheiten	94
7.1	Reihenfolge von Zuteilung und Anordnung	94
7.2	Datenfluß-orientierte Zuteilung	95
7.3	Zuteilung nach Ellis	96
7.4	Modulo-Zuteilung in Pipeline-Schedules	100
7.5	SCC-hierarchische Zuteilung	104
8	Weitere Aufgaben der VLIW-Code-Erzeugung	106
8.1	Registerzuteilung und Spillcode im VLIW-Code	106
8.2	Optimierung und VLIW-Code-Erzeugung	112
9	Zusammenfassung	118
10	Literatur	119