



**PCT** WELTORGANISATION FÜR GEISTIGES EIGENTUM  
Internationales Büro  
INTERNATIONALE ANMELDUNG VERÖFFENTLICHT NACH DEM VERTRAG ÜBER DIE  
INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS (PCT)

<p>(51) Internationale Patentklassifikation<sup>5</sup> : <b>G06F 9/45</b></p>	<b>A1</b>	<p>(11) Internationale Veröffentlichungsnummer: <b>WO 94/22079</b></p> <p>(43) Internationales Veröffentlichungsdatum: 29. September 1994 (29.09.94)</p>
<p>(21) Internationales Aktenzeichen: <b>PCT/DE94/00221</b></p> <p>(22) Internationales Anmeldedatum: 1. März 1994 (01.03.94)</p> <p>(30) Prioritätsdaten: P 43 08 173.8 15. März 1993 (15.03.93) DE</p> <p>(71) Anmelder (für alle Bestimmungsstaaten ausser US): SIEMENS AKTIENGESELLSCHAFT [DE/DE]; Wittelsbacherplatz 2, D-80333 München (DE).</p> <p>(72) Erfinder; und (75) Erfinder/Anmelder (nur für US): SCHEPERS, Jörg [DE/DE]; Hallsteinweg 5, D-81739 München (DE).</p>		<p>(81) Bestimmungsstaaten: JP, US, europäisches Patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).</p> <p><b>Veröffentlicht</b> <i>Mit internationalem Recherchenbericht.</i></p>
<p>(54) Title: PROCESS FOR THE MACHINE-GENERATION OF SECONDARILY PROCESSABLE COMMAND GROUPS FROM A PROGRAM FOR SUPER-SCALAR MICROPROCESSORS</p> <p>(54) Bezeichnung: VERFAHREN ZUR MASCHINELLEN ERZEUGUNG VON NEBENLÄUFIG BEARBEITBAREN BEFEHLSGRUPPEN AUS EINEM PROGRAMM FÜR SUPERSKALARE MIKROPROZESSOREN</p> <p>(57) Abstract</p> <p>For the fast running of a program on super-scalar processors, the individual commands of the program must be distributed in command groups processable by processing units in the processor in such a way that the commands can be processed in parallel. To this end, account must be taken of data and control flow factors and pipeline conflicts. Therefore are first selected the commands the preceding commands of which have been already processed and these commands are examined to see whether a minimum number of delay cycles is necessary before their execution and the commands with a minimum number of delay cycles are stored in a list. One of these commands is selected by a heuristic selection process and arranged in a command group in which the command can be processed during the earliest possible execution cycle.</p> <p>(57) Zusammenfassung</p> <p>Um eine schnelle Abarbeitung eines Programms auf superskalaren Prozessoren ausführen zu können, müssen die einzelnen Befehle dieses Programms in von Verarbeitungseinheiten des Prozessors bearbeitbare Befehlsgruppen so aufgeteilt werden, daß die Befehle parallel bearbeitbar sind. Dabei müssen Daten- und Kontrollflußabhängigkeiten berücksichtigt werden sowie Pipelinekonflikte. Dazu werden zunächst die Befehle ausgewählt, deren Vorgängerbefehle bereits bearbeitet worden sind und diese Befehle dahingehend untersucht, ob vor deren Ausführung eine minimale Anzahl von Verzögerungszyklen erforderlich sind und die Befehle mit minimaler Anzahl in einer Liste abgespeichert. Von diesen Befehlen werden nach einem heuristischen Auswahlverfahren einer ausgewählt und dieser in eine Befehlsgruppe eingeordnet, bei der der Befehl im frühest möglichen Ausführungszyklus bearbeitbar ist.</p>		

**LEDIGLICH ZUR INFORMATION**

Codes zur Identifizierung von PCT-Vertragsstaaten auf den Kopfbögen der Schriften, die internationale Anmeldungen gemäss dem PCT veröffentlichen.

AT	Österreich	GA	Gabon	MR	Mauretanien
AU	Australien	GB	Vereinigtes Königreich	MW	Malawi
BB	Barbados	GE	Georgien	NE	Niger
BE	Belgien	GN	Guinea	NL	Niederlande
BF	Burkina Faso	GR	Griechenland	NO	Norwegen
BG	Bulgarien	HU	Ungarn	NZ	Neuseeland
BJ	Benin	IE	Irland	PL	Polen
BR	Brasilien	IT	Italien	PT	Portugal
BY	Belarus	JP	Japan	RO	Rumänien
CA	Kanada	KE	Kenya	RU	Russische Föderation
CF	Zentrale Afrikanische Republik	KG	Kirgisistan	SD	Sudan
CG	Kongo	KP	Demokratische Volksrepublik Korea	SE	Schweden
CH	Schweiz	KR	Republik Korea	SI	Slowenien
CI	Côte d'Ivoire	KZ	Kasachstan	SK	Slowakei
CM	Kamerun	LI	Liechtenstein	SN	Senegal
CN	China	LK	Sri Lanka	TD	Tschad
CS	Tschechoslowakei	LU	Luxemburg	TG	Togo
CZ	Tschechische Republik	LV	Lettland	TJ	Tadschikistan
DE	Deutschland	MC	Monaco	TT	Trinidad und Tobago
DK	Dänemark	MD	Republik Moldau	UA	Ukraine
ES	Spanien	MG	Madagaskar	US	Vereinigte Staaten von Amerika
FI	Finnland	ML	Mali	UZ	Usbekistan
FR	Frankreich	MN	Mongolei	VN	Vietnam

Verfahren zur maschinellen Erzeugung von nebenläufig bearbeitbaren Befehlsgruppen aus einem Programm für superskalare Mikroprozessoren

Moderne Mikroprozessoren bestehen zumeist aus mehreren funktionalen Einheiten, im folgenden Verarbeitungseinheiten genannt, die Maschinenbefehle parallel ausführen können. Diese Eigenschaft wird als superskalar bezeichnet. Um alle Verarbeitungseinheiten auszulasten, müßten superskalare Prozessoren in einem Taktzyklus mehrere Befehle als Gruppe zur Ausführung bereitstellen. Da Verarbeitungseinheiten jedoch meist nur bestimmte Typen von Maschinenbefehlen, wie z.B. Load/-Store-oder Gleitpunktbefehle, ausführen können, muß bei der Befehlsbereitstellung bereits darauf geachtet werden, daß alle Befehle einer Gruppe auch auf entsprechenden Verarbeitungseinheiten ausführbar sind. Bei der Befehlsgruppierung müssen zudem Daten- und Kontrollflußabhängigkeiten zwischen den Befehlen berücksichtigt werden, um der Semantik des Programms zu genügen. Ferner gilt es Pipelinekonflikte innerhalb den Verarbeitungseinheiten zu minimieren, um für jede einzelne Verarbeitungseinheit einen möglichst hohen Durchsatz zu erzielen.

Die Befehlsgruppierung spielt somit eine wesentliche Rolle für die effektive Leistung eines superskalaren Mikroprozessors. Es gilt die Befehle innerhalb einer Befehlssequenz so anzuordnen, daß möglichst viele Befehle parallel ausgeführt werden können. Dazu sind bereits mehrere Lösungsansätze entwickelt und veröffentlicht worden, die sich in statische und dynamische Schedulingverfahren unterteilen lassen.

Bei dynamischen Schedulingverfahren werden Befehle während der Ausführung umgruppiert. Dies geschieht meist mit Hilfe spezieller Hardware, wie z.B. 'score boards' oder 'reservation tables'. Entsprechende Techniken sind u.a. in Tomasulo, R.M. An Efficient Algorithm for Exploring Multiple Arithmetic Units, IBM Journal of Research and Development, Jan. 1967, S. 25 bis 33 beschrieben. Sie erfordern teilweise erheblichen zusätzlichen Hardwareaufwand und operieren zumeist nur auf einer fest vordefinierten Anzahl von Befehlen (Befehlsfenster).

10

Statische Scheduling-Verfahren versuchen schon zur Compilezeit eine Befehlssequenz zu erzeugen, in der Pipelinekonflikte minimiert sind und möglichst viele Instruktionen parallel ausgeführt werden können. Verschiedene Algorithmen hierfür wurden insbesondere für Prozessoren mit Befehlspipeline sowie im Bereich der Mikroprogrammierung und für VLIW-Architekturen entwickelt. Ein Überblick über die Algorithmen für Super-Pipelinearchitekturen ist z.B. in Krishnamurthy, S.M., A Brief Survey of Papers on Scheduling for Pipelined Processors; SIGPLAN Notices, Vol. 25, No. 7, 1990, pp. 97-106 gegeben. Diese Algorithmen erzeugen heuristisch eine Befehlssequenz mit möglichst wenig Pipelinekonflikten, wobei jedoch Konflikte zwischen daten-unabhängigen Instruktionen meistens außer acht gelassen werden.

25

Die Problematik der Befehlsgruppierung, wie sie bei superskalaren Prozessoren vorkommt, wird ansatzweise in Algorithmen zur Codeerzeugung für VLIW-Architekturen behandelt. Insbesondere kommen dabei globale Scheduling-Verfahren wie Trace- oder Percolation-Scheduling zum Einsatz, die Befehle über Basisblockgrenzen hinweg verschieben können (z.B. Fisher, J.A., Trace Scheduling: A Technique for Global Microcode Compaction, IEEE Trans. on Comp., July 1981, S. 478 - 490). Im Unterschied zu superskalaren Architekturen müssen bei VLIW-Architekturen

30

jedoch bereits statisch breite Befehls Worte erzeugt werden, d.h. die einzelnen Befehle werden bereits zur Compilezeit den verschiedenen Verarbeitungseinheiten zugeordnet und die Instruktionen eines Befehls Wortes werden synchron ausgeführt.

5 Superskalare Architekturen bieten diesbezüglich mehr Freiräume, da Befehlsgruppierungen letztendlich erst zur Laufzeit vorgenommen werden, und nicht alle Befehle einer Gruppe synchron ausgeführt werden.

10 Die der Erfindung zugrundeliegende Aufgabe besteht darin, ein weiteres Verfahren anzugeben, nach dem auch für superskalare Prozessoren aus einem Programm parallel bearbeitbare Befehlsgruppen erzeugt werden, wobei sichergestellt ist, daß alle Befehle einer Befehlsgruppe ohne zusätzliche Verzögerungszyklen ausführbar sind. Derartige Verzögerungszyklen können bei einem  
15 Programm ohne Gruppierung dadurch bedingt sein, daß Datenabhängigkeit zwischen den Befehlen besteht, d.h. ein Befehl ausgeführt sein muß, bevor der datenabhängige folgende Befehl ausgeführt werden kann bzw. in der Pipelinestruktur Pipelinekonflikte bestehen, die sog. Interlock-Zyklen bedingen.  
20

Die Aufgabe wird mit einem Verfahren gemäß den Merkmalen des Anspruchs 1 gelöst.

25 Ein besonders vorteilhaftes Verfahren zur Erzeugung der Befehlsgruppen ergibt sich aus Anspruch 2.

Das erfindungsgemäße Verfahren verzichtet somit auf ein dynamisches Scheduling-Verfahren, so daß keine zusätzliche Hardware erforderlich ist. Die Einordnung der Befehle in die Befehlsgruppen erfolgt vielmehr bereits bei der Kompilierung d.h. bevor die Befehlsbearbeitung durchgeführt wird.  
30

Anhand eines Ausführungsbeispielles wird die Erfindung weiter erläutert.

Das Verfahren geht aus von einem Scheduling-Verfahren, das in  
 5 Gibbons, P.B., Muchnick, S.S., Efficient Instruction Scheduling  
 for a Pipelined Architecture, Proc. 86, Sigplan Notices, Vol  
 21, Nr. 7, Seiten 11 bis 16 beschrieben ist. Zur Erläuterung  
 wird das im folgenden angegebene Programmstück verwendet, das  
 in Assemblercode geschrieben ist:

10

```

1:      lw      r3,      0(r4)      #load word
2:      add     r3,      r3,1      #add
3:      sl     r4,      1          #shift left
4:      sub     r3,      r3,r4      #subtract'
5:      l.s    f2,      0(r5)      #load float
6:      add.s   f1,      f2,f1      #add float
7:      abs.s   f3,      f2          #float ab-
                                         solute
8:      s.s    f3,      0(r5)      #store
                                         float
9:      bgtz   r3,      L1          #branch
                                         greater
                                         zero

```

Die einzelnen Befehle sind mit Hilfe der neben den Befehlen an-  
 gegebenen Erläuterungen leicht verständlich. Zum Beispiel be-  
 sagt der Befehl 1, daß ein Wort in das Register r3 geladen wer-  
 15 den soll und zwar von einem Speicherplatz, dessen Adresse im  
 Register r4 steht. Im zweiten Befehl soll zum Inhalt des Regi-  
 sters r3 eine 1 addiert werden und das Ergebnis in das Register  
 r3 abgespeichert werden. Der Befehl 3 gibt an, daß das im Regi-  
 20 ster r4 stehende Datum um eine Stelle nach links verschoben  
 werden soll. Befehl 4 ist ein Substraktionsbefehl, und zwar  
 soll der Inhalt von Register r4 vom Inhalt von Register r3 sub-

trahiert werden und das Ergebnis in das Register r3 abgespeichert werden. Die Befehle 1 bis 4 sind sog. integer-Befehle. Die Befehle 5 bis 9 dagegen sind Gleitpunktbefehle. Befehl 5 gibt an, daß eine Gleitpunktzahl in das Gleitpunktregister f2 geladen werden soll, deren Adresse im Register r5 steht. Nach Befehl 6 wird der Inhalt des Gleitpunktregisters f1 mit dem Inhalt des Gleitpunktregister f2 addiert und das Ergebnis im Gleitpunktregister f1 abgespeichert. Im Befehl 7 wird der Absolutwert der Gleitpunktzahl gebildet, und zwar der Zahl, die im Register f2 steht und das Ergebnis in das Gleitpunktregister f3 abgespeichert. Der Inhalt des Gleitpunktregisters f3 wird im Speicher unter einer Adresse abgespeichert, die im Register r5 steht und schließlich und endlich wird im Befehl 9 ein Verzweigungsbefehl ausgeführt, bei dem zur Adresse L1 gesprungen wird, wenn der Inhalt des Register r3 größer Null ist.

Mit Hilfe der graphischen Darstellung der Figur 1 werden die Datenabhängigkeiten der einzelnen Befehle gezeigt und dabei die Probleme, die bei einer parallelen Bearbeitung auftreten, besser erkenntlich. Die Darstellung besteht aus Knoten und Kanten, wobei an den Kanten jeweils eine Verzögerungszyklentabelle V-Tab angeordnet ist, die angibt, ob ein Verzögerungszyklus zwischen zwei aufeinanderfolgenden Befehlen erforderlich ist. Die erste Zahl zeigt die Verzögerungszyklen, die durch explizite NO-OPERATION (noop) Instruktionen in der Befehlssequenz sichergestellt werden müssen, die zweite Zahl die durch Pipelinekonflikte bedingte Anzahl von Verzögerungszyklen, die im Programm mit Hilfe von s-noop-Befehlen symbolisiert werden. Zum Beispiel ist bei den integer-Befehlen ein Verzögerungszyklus erforderlich, wenn der Befehl 2 direkt nach Befehl 1 ausgeführt wird. Dagegen ist der Befehl 3 unabhängig von der Ausführung des Befehles 2, so daß diese Befehle parallel ausführbar sind. Er kann aber aufgrund der Datenabhängigkeit nur nach dem Befehl 1 ausgeführt werden, da sonst die Speicheradresse in

Register r4 falsch wäre. Entsprechendes gilt auch für die Gleitpunktbefehle, die auf der rechten Seite des Diagramms dargestellt sind. Hier sind in V-Tab Verzögerungszyklen aufgrund von Datenabhängigkeit an der ersten Position angegeben; Verzögerungszyklen, die aufgrund von Pipelinekonflikten vorliegen, an der zweiten Position. Das bedeutet z.B. daß der Befehl 8 "s.s" erst dann ohne zusätzliche Verzögerungszyklen ausgeführt werden kann, wenn nach Ausführung des Befehles 7 "abs s".zwei weitere Taktzyklen verstrichen sind.

10

Weiterhin ist in Figur 1 die Abhängigkeit zwischen den Befehlen "add.s" und "abs.s" angegeben. Z.B. kann "add.s" durch dieselbe Verarbeitungseinheit nach "abs.s" erst nach drei Verzögerungszyklen ausgeführt worden. Umgekehrt kann "abs.s" erst zwei Verzögerungszyklen nach "add.s" ausgeführt werden. Bei Bearbeitung auf verschiedenen Einheiten besteht diese Abhängigkeit nicht.

15

Bevor das erfindungsgemäße Verfahren im einzelnen erläutert wird, sind noch einige Begriffe zu definieren, die im Verfahren verwendet werden:

20

1. Blockierungsstelle L, diese ist gesetzt, wenn ein Befehl mindestens einen Verzögerungszyklus zu einem der direkt nachfolgenden Befehle verursacht.

25

2. Nachfolgezahl, diese Zahl gibt an, wieviel datenabhängige Nachfolgebefehle auf einen Befehl folgen. Je mehr Nachfolger ein Befehl hat, desto höher ist seine Priorität bei der Einordnung der Befehle in die Befehlsgruppen.

30

3. Distanzwert, er gibt die maximale Entfernung in Taktzyklen zwischen dem betrachteten Befehl und dem letzten Befehl im Diagramm an. Dabei sind auch die Verzögerungszyklen zu berücksichtigen. Es sei:

$d(n, n')$  die maximale Anzahl von Verzögerungszyklen zwischen einem Befehl  $n$  und einem direkt datenabhängigen Nachfolgebefehl  $n'$  und  $h(n')$  der Distanzwert von  $n'$ , dann ergibt sich der

5 Distanzwert von  $n$  aus

$h(n) := 0$ , wenn die Anzahl der Nachfolger gleich 0 ist,  
oder

$h(n) = h(n') + d(n, n') + 1$ , wobei bei mehreren direkten Nachfolgern  $n'$  der maximale Wert genommen wird.

10

Diese Werte sind in Wertetabellen W-Tab eingefügt, die zur Verdeutlichung an den einzelnen Knoten des Diagramms der Figur 1 angegeben sind. Von links ausgehend ist an der ersten Stelle die Blockierungsstelle  $L$ , an der zweiten Stelle die Nachfolgezahl, an der dritten Stelle der Distanzwert angegeben. Zum Beispiel ist bei dem Befehl 1 "lw" die Blockierungsstelle  $L$  gesetzt, da ein Verzögerungszyklus folgt; die Nachfolgezahl ist 2, da zwei datenabhängige Befehle direkt folgen; der Distanzwert ist 4 zwischen dem ersten Befehl und dem Befehl 9. Entsprechend können die anderen Wertetabellen, die den anderen

15  
20

Knoten zugeordnet sind, interpretiert werden.

Das erfindungsgemäße Verfahren wird nunmehr mit Hilfe des Diagramms der Figur 1 und unter Verwendung von Tabelle 1 und Tabelle 2 weiter erläutert. Tabelle 1 zeigt, wie nacheinander die einzelnen Befehle des Programmstückes behandelt werden und dann entsprechend Tabelle 2 zu Befehlsgruppen zusammengefügt werden, die von den Verarbeitungseinheiten des superskalaren Prozesses parallel ausführbar sind. Dabei ist hier als Beispiel

25  
30

vorausgesetzt, daß der superskalare Prozessor Datenwrote mit einer Breite von 3 Befehlen laden kann.

Die Untersuchung des Programmstückes erfolgt immer mit den in Diagramm nach Figur 1 ersten Befehlen, die keine Vorgängerbe-

fehle haben, von denen sie datenabhängig sind. Anschließend erfolgt die Untersuchung der diesen Befehlen nachfolgenden Befehle und es wird immer untersucht, in welche Befehlsgruppe ein Befehl dann einzuordnen ist. Ziel ist es also, die Befehle in die Befehlsgruppen so einzuordnen, daß die Anzahl der erforderlichen Verzögerungszyklen minimal wird. Dieses Ziel wird mit folgenden Verfahrensschritten erreicht:

Im ersten Schritt wird für jeden Befehl die Blockierungsstelle, die Nachfolgezahl, der Distanzwert berechnet und in die Wertetabelle W-Tab eingespeichert. In der V-Tab stehen die Verzögerungszyklen.

Im zweiten Schritt wird dafür gesorgt, daß alle Befehle unmarkiert sind.

Im dritten Schritt wird eine erste Liste CS gebildet, in die die Befehle aufgenommen werden, die keinen unmarkierten Vorgängerbefehl haben.

Im vierten Schritt wird überprüft, ob die erste Liste CS leer ist oder nicht und für den Fall, daß die erste Liste leer ist, das Verfahren beendet.

Im fünften Schritt werden diejenigen unmarkierten Befehle aus der ersten Liste bestimmt, die ausführbar werden nach einer minimalen Anzahl von Verzögerungszyklen und aus diesen Befehlen wird eine zweite Liste RS gebildet.

Im sechsten Schritt wird nach einem heuristischen Auswahlverfahren einer der in der zweiten Liste enthaltenen Befehle ausgewählt und für die Einordnung in die Befehlsgruppen vorgesehen.

Im siebten Schritt wird der ausgewählte Befehl in eine der Befehlsgruppen in einer Komponente eingefügt. Dazu wird zunächst der frühest mögliche Zyklus bestimmt, in dem der in Schritt 6 selektierte Befehl ausgeführt werden kann. Danach wird die Befehlsgruppe gewählt, die in diesem Taktzyklus oder aber möglichst bald danach ausgeführt wird, und in die der Befehl unter Berücksichtigung ggf. vorhandener, durch die Architektur des Prozessors bedingter, Gruppierungsrestriktionen plaziert werden kann. Leere Stellen innerhalb der Befehlsgruppe vor dem angeordneten letzten Befehl müssen mit s-noop Befehlen aufgefüllt werden.

Im Schritt 8 wird der eingefügte Befehl markiert und aus der Liste CS entfernt. Es wird mit Schritt 3 weiter verfahren.

Das heuristische Auswahlverfahren (Schritt 6) kann z.B. in folgender Weise erfolgen:

-Zunächst werden die Befehle in der zweiten Liste RS ausgewählt, deren Blockierungsstelle L gesetzt ist. Gibt es nur einen derartigen Befehl, dann ist dieser Befehl ausgewählt. Gibt es mehrere derartige Befehle oder gibt es keinen Befehl mit einer gesetzten Blockierungsstelle, dann wird folgendermaßen weiter vorgegangen:

- Es werden aus der zweiten Liste Befehle ausgewählt, die den größten Distanzwert haben. Gibt es mehrere Befehle mit größten Distanzwert, dann werden aus diesen die Befehle ausgewählt, die die größte Nachfolgezahl haben.

- Immer dann, wenn mehrere gleichwertige Befehle ermittelt worden sind, kann einer dieser Befehle frei gewählt werden und in die Befehlsgruppe entsprechend Schritt 7 eingefügt werden. Das heuristische Auswahlverfahren nach Schritt 6 kann selbstverständlich auch auf andere Weise erfolgen, z.B. können die einzelnen dargestellten Schritte vertauscht werden, was aber unter Umständen zu schlechteren Lösungen führt.

Tabelle 1

	CS	RS	Befehl (Zyklus)
1	lw;l.s	lw;l.s	l.s (1)
2	lw,add.s;abs.s	lw	lw (1)
3	add;sl;add.s;abs.s	sl; add.s; abs.s	abs.s (3)
4	sl s.s;add;add.s	add; sl;	sl (2)
5	add.s; s.s;add	add	add (3)
6	sub;add.s;s.s	sub; s.s; add.s	add.s (3)
7	sub;s.s	sub; s.s	sub (4)
8	s.s..	s.s	s.s (5)
9	bgzt	bgzt	bgzt (6)

Tabelle 2

## 5 Programm Array

Zyklus	KP 1	KP 2	KP 3
1	l.s	lw	s-noop BG1
2	sl	s-noop	s-noop BG2
3	abs.s	add	add.s BG3
4	sub	s-noop	s-noop BG4
5	s-noop	s-noop	s-noop BG5
6	s.s	bgzt	s-noop BG6

10 In Tabelle 1 ist in der ersten Spalte die erste Liste als CS bezeichnet, die zweite Liste als RS und die dritte Spalte gibt an, welcher Befehl ausgewählt worden ist. In der Tabelle 2 sind die einzelnen Befehlsgruppen BG gezeigt, die nach dem Verfahren entstehen. In der ersten Spalte ist der Verarbeitungszyklus angegeben, in dem die Befehlsgruppe ausgeführt wird und in einer

15 zweiten Spalte eine erste Komponente KP1 der Befehlsgruppen BG, in einer dritten Spalte eine zweite Komponente KP2 der Befehlsgruppen BG und in einer vierten Spalte eine vierte Komponente

KP3 der Befehlsgruppen BG. Innerhalb der einzelnen Befehlsgruppen sind Verzögerungszyklen mit s-noop angegeben.

Zunächst sind die beiden Tabellen 1 und 2 leer. Es wird angenommen, daß die Schritte 1 und 2 bereits durchgeführt sind. Be-  
5 begonnen wird somit mit Schritt 3 des Verfahrens. Zunächst werden die Knoten oder Befehle gesucht, die keinen unmarkierten Vorgänger haben. Dies sind entsprechend der Figur 1 der Befehl "lw" und der Befehl "l.s". Diese beiden Befehle werden in die  
10 erste Zeile in der Tabelle 1 in der Spalte CS eingetragen. Dann wird im Schritt 5 untersucht, welcher dieser beiden Befehle eine minimale Anzahl von Verzögerungszyklen erfordern. Beide Befehle können ohne zusätzliche Verzögerungszyklen ausgeführt werden und dementsprechend werden die Befehle in die Spalte RS  
15 eingefügt. Mit Schritt 6 wird einer der beiden Befehle ausgewählt. Da beide Befehle eine gesetzte Blockierungsstelle haben, wird der Befehl ausgewählt, der den größten Distanzwert hat und das ist der Befehl "l.s.". Dieser Befehl wird nun in der  
20 Tabelle 2 in eine Befehlsgruppe eingefügt. Unter Vereinbarung, daß die Einfügung in der Befehlsgruppe mit dem frühest zulässigen Ausführungszyklus geschieht, und das z.B. an der am weitesten links stehenden zulässigen Spalte (Komponente), dann wird der Befehl der Befehlsgruppe im Zyklus 1 der Komponente KP1 zugeordnet. Der Befehl "l.s" wird markiert und es beginnt  
25 wiederum mit Schritt 3 des Verfahrens.

Nach Schritt 3 werden die Befehle ausgewählt, die keinen unmarkierten unmittelbaren Vorgänger haben. Dies sind nun neben "lw" auch die Befehle "add.s" und "abs.s". Diese werden in die Spalte  
30 CS eingefügt. Im Schritt 5 wird nun untersucht, welche dieser Befehle nach einer minimalen Anzahl von Verzögerungszyklen bearbeitet werden können und diese in die Spalte RS eingefügt. RS enthält nur den Befehl lw, so daß dieser selektiert und

ebenfalls, gemäß Schritt 7, im ersten Zyklus an zweiter Position KP2 eingefügt wird.

5 Es soll noch ein dritter Durchgang erläutert werden. Wieder wird mit Schritt 3 begonnen und untersucht, welche zusätzlichen Befehle keine unmarkierten Vorgängerbefehle aufweisen. Dies sind die Befehle "add" und "sl". Für alle Befehle aus CS wird Schritt 5 durchgeführt und festgestellt, welcher Befehl nach einer minimalen Anzahl von Verzögerungszyklen verarbeitungs-

10 bereit wird. Dies sind die Befehle "sl", "add.s" und "abs.s", die dann in die zweite Spalte RS eingefügt werden. Für die drei in der Spalte RS enthaltenen Befehle wird nun der Schritt 6 durchgeführt. Von denen sind nur die Befehle "abs.s" und "add.s" mit einer gesetzten Blockierungsstelle versehen, so daß

15 derjenige mit dem größten Distanzwert ("abs.s") ausgewählt wird. Dieser muß nun in der Tabelle in die Befehlsgruppen eingefügt werden. Dabei muß berücksichtigt werden, daß "abs.s" erst einen Verzögerungszyklus nach "l.s" ausführbar ist. Er wird somit im Zyklus 3 entsprechend Schritt 7 eingefügt.

20

Zum Abschluß der Erläuterung des Verfahrens wird ein vierter Durchgang geschildert. Nach Markierung des Befehls "abs.s" entsprechend Schritt 8 wird in Schritt 3 der Befehl "s.s" in die Spalte CS eingefügt. In dieser Spalte stehen zusätzlich noch

25 die Befehle "add" "sl" und "add.s". Nur die Befehle "add" und "sl" sind ohne Verzögerungszyklus ausführbar, da "add.s" frühestens zwei Verzögerungszyklen nach abs.s ausgeführt werden darf. Zwischen "add und sl" wird ein Befehl nach Schritt 6 ausgewählt. Da beide gleiche Einträge in W-Tab haben, kann einer

30 frei gewählt werden, z.B. "sl". Der Befehl "sl" muß nun in eine der Befehlsgruppen eingefügt werden. Es ist erkennbar, daß der Befehl "sl" unmittelbar nach Ausführung des Befehles "lw" ausführbar ist, so daß er im zweiten Zyklus angeordnet werden kann. Entsprechend Schritt 7 wird er somit der ersten Kompo-

nente KP1 der zweiten Befehlsgruppe BG2 zugeordnet. Der Befehl "sl" wird markiert und dann anschließend mit Schritt 3 des Verfahrens wieder weiter gefahren.

5 Dies erfolgt solange, bis keine Befehle in die Spalte 1 der Tabelle 1 mehr enthalten sind. Als Ergebnis ergeben sich Befehlsfolgen BG1 bis BG6 entsprechend der Tabelle 2.

10 Es wird darauf hingewiesen, daß immer dann, wenn nach Schritt 6 mehrere gleichartige Befehle vorliegen, einer der Befehle freige wählt werden kann und dementsprechend die in einer Befehlsgruppe enthaltenen Befehle entsprechend dieser Wahl unterschiedlich sein können. Alle bisher nicht besetzten Stellen in den Befehlsgruppen BG können mit snoop-Befehlen ergänzt werden.  
15 Wie das Programm Array der Tabelle 2 zeigt, müssen nur zwei Verarbeitungseinheiten Gleitpunktbefehle verarbeiten können.

Die graphische Darstellung der Tabelle 2 zeigt, daß aus einem Programm eine zweidimensionale Darstellung entwickelt wird. Damit kann der superskalare Prozessor nicht nur einen Befehl pro Zyklus holen, sondern er kann eine Gruppe von Befehlen holen, die alle parallel ausgeführt werden können, ohne daß Konflikte entstehen. Das Ziel der Codeerzeugung für superskalare Prozessoren, das darin besteht, daß eine maximale Anzahl von Befehlen parallel verarbeitbar sind, ist dann erreicht. Superskalare Prozessoren erfordern also keine Befehlsfolgen, sondern Folgen von Befehlsgruppen. Dies kann sehr gut an der Tabelle 2 abgelesen werden. Die Anzahl der Komponenten in dem Programm array hängt ab von der Anzahl der Befehle, die der Prozessor zugleich laden kann.  
20  
25  
30

Es ist noch zu erwähnen, daß die Einfügung der snoop Befehle nur für die Erläuterung erforderlich war. Für die Codeerzeugung

14

werden die Zeilen des arrays hintereinander angeordnet, z.B.  
von links nach rechts und die snoop-Befehle gestrichen.

5

10

## Patentansprüche

1. Verfahren zur maschinellen Erzeugung von nebenläufig bearbeitbaren Befehlsgruppen aus einem Programm für superskalare Prozessoren,

- 5 a) bei dem für jeden Befehl des Programms in einer Wertetabelle (W-Tab)
- eine Blockierungsstelle gesetzt wird, wenn vor Ausführung eines direkt datenabhängigen Befehles ein Verzögerungszyklus einzuschieben ist,
  - 10 - eine Nachfolgezahl eingefügt wird, die angibt, wieviele datenabhängige Befehle direkt folgen,
  - ein Distanzwert eingetragen wird, der die maximale Anzahl von Taktzyklen zum letzten der datenabhängigen Befehle
  - 15 angibt,
- b) bei dem in einer Verzögerungszyklentabelle (V-Tab) für jeden Befehl angegeben ist, wieviel Verzögerungszyklen zwischen den Befehlen liegen,
- c) bei dem jeder der Befehle in Abhängigkeit seiner Blockierungsstelle, der Nachfolgezahl, des Inhalts der Verzögerungszyklentabelle und seinem Distanzwert in Befehlsgruppen
- 20 eingeordnet sind.

2. Verfahren nach Anspruch 1, bei dem

- 25 a) zunächst alle Befehle unmarkiert sind,
- b) alle Befehle, die keinen vorhergehenden datenabhängigen unmarkierten Befehl haben, in einer ersten Liste (CS) abgespeichert werden,
- d) diejenigen Befehle aus der ersten Liste (CS) ausgewählt
- 30 werden, die nach einer minimalen Anzahl von Verzögerungszyklen ausführbar sind und diese Befehle in einer zweiten Liste abgespeichert werden,
- d) nach einem heuristischen Auswahlverfahren ein Befehl ausgewählt wird,

- e) jeder ausgewählte Befehl in eine der aus Komponenten bestehenden Befehlsgruppe so eingeordnet wird, daß der Befehl der Befehlsgruppe mit dem frühest zulässigen Ausführungszyklus zugeordnet wird und das entsprechend einer vorgebbaren Reihenfolge in eine der Komponenten,
- f) der eingeordnete Befehl markiert wird,
- g) die Schritte a) bis f) solange durchgeführt werden bis alle Befehle des Programms bearbeitet sind.

5

10

### 3. Verfahren nach Anspruch 2,

bei dem das heuristische Verfahren darin besteht, daß derjenige Befehl aus der zweiten Liste (RS) ausgewählt wird,

- dessen Blockierungsstelle gesetzt ist,

15

- bei fehlender gesetzter Blockierungsstelle der Befehl mit maximalem Distanzwert und maximaler Nachfolgezahl,

- wobei bei mehreren ermittelten Befehlen ein beliebiger dieser Befehle ausgewählt wird.

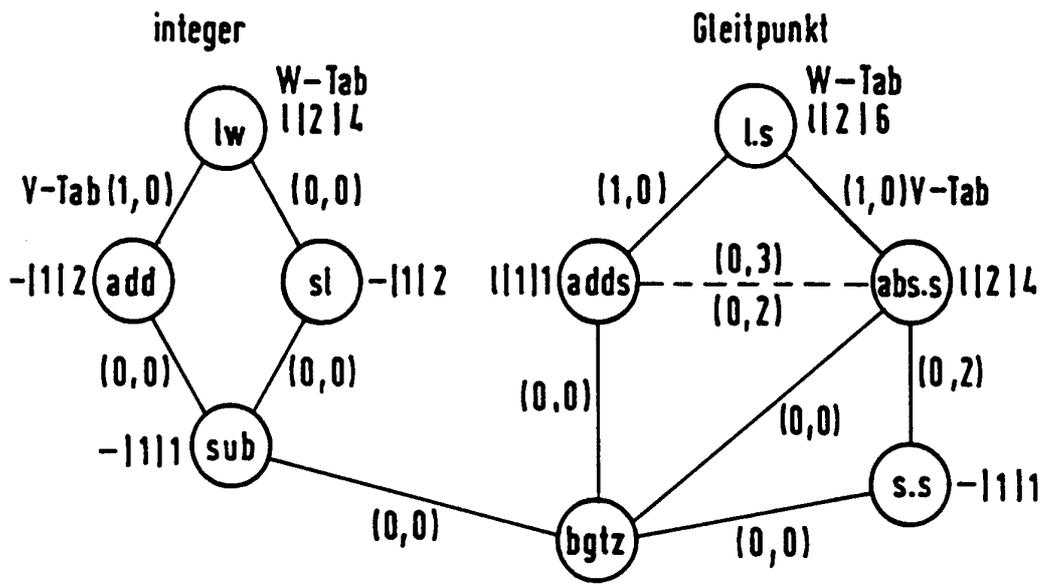
20

### 4. Verfahren nach Anspruch 2 oder 3,

bei dem die Anzahl der Komponenten pro Befehlsgruppe entsprechend der Anzahl der Befehle festgelegt wird, die ein Prozessor gleichzeitig laden kann.

25

30



**INTERNATIONAL SEARCH REPORT**

Intern. Application No  
PCT/DE 94/00221

**A. CLASSIFICATION OF SUBJECT MATTER**  
IPC 5 G06F9/45

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)  
IPC 5 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO,A,91 20031 (SUPERCOMPUTER SYSTEMS LTD.) 26 December 1991 see the whole document ---	1-4
A	EP,A,0 501 653 (SUN MICROSYSTEMS) 2 September 1992 see the whole document ---	1-3
A	SIGPLAN NOTICES vol. 21, no. 7 , July 1986 , PALO ALTO, US, pages 11 - 16 GIBBONS AND MUCHNICK 'Efficient instruction scheduling for a pipelined architecture' cited in the application see the whole document ---	1-3
	-/--	

Further documents are listed in the continuation of box C.       Patent family members are listed in annex.

\* Special categories of cited documents :

<p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier document but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p>	<p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.</p> <p>"&amp;" document member of the same patent family</p>
--	--

Date of the actual completion of the international search  14 June 1994	Date of mailing of the international search report
---	--

Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+ 31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+ 31-70) 340-3016	Authorized officer  Weinberg, L
--	---------------------------------------

INTERNATIONAL SEARCH REPORT

Intern. Application No  
PCT/DE 94/00221

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>IBM JOURNAL OF RESEARCH AND DEVELOPMENT vol. 34, no. 1 , January 1990 , NEW YORK US pages 85 - 92 WARREN 'Instruction scheduling for the IBM system/6000 processor' see the whole document ---</p>	1-3
A	<p>COMPUTER ARCHITECTURE NEWS vol. 21, no. 1 , March 1993 , NEW YORK US pages 4 - 11 INOUE AND TAKEDA 'Performance evaluation for various configurations of superscalar processors' see page 5, right column, paragraph 4 - page 7, left column, paragraph 1 -----</p>	1

# INTERNATIONAL SEARCH REPORT

Information on patent family members

Intern. Appl. Application No <b>PCT/DE 94/00221</b>
--

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO-A-9120031	26-12-91	US-A- 5179702	12-01-93
		AU-A- 8099891	07-01-92
		AU-A- 8201891	07-01-92
		EP-A- 0533805	31-03-93
		EP-A- 0535107	07-04-93
		EP-A- 0533813	31-03-93
		EP-A- 0537257	21-04-93
		JP-T- 5508494	25-11-93
		WO-A- 9120029	26-12-91
		WO-A- 9120030	26-12-91
		WO-A- 9120033	26-12-91
		WO-A- 9120036	26-12-91
		US-A- 5193192	09-03-93
		US-A- 5107418	21-04-92
		US-A- 5202988	13-04-93
		US-A- 5175856	29-12-92
		US-A- 5159678	27-10-92
		US-A- 5187789	16-02-93
		US-A- 5202975	13-04-93
		US-A- 5307478	26-04-94
EP-A-0501653	02-09-92	US-A- 5202993	13-04-93

A. KLASSIFIZIERUNG DES ANMELDUNGSGEGENSTANDES IPK 5 G06F9/45		
Nach der Internationalen Patentklassifikation (IPK) oder nach der nationalen Klassifikation und der IPK.		
B. RECHERCHIERTE GEBIETE		
Recherchierter Mindestprüfstoff (Klassifikationssystem und Klassifikationssymbole) IPK 5 G06F		
Recherchierte aber nicht zum Mindestprüfstoff gehörende Veröffentlichungen, soweit diese unter die recherchierten Gebiete fallen		
Während der internationalen Recherche konsultierte elektronische Datenbank (Name der Datenbank und evtl. verwendete Suchbegriffe)		
C. ALS WESENTLICH ANGESEHENE UNTERLAGEN		
Kategorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
A	WO,A,91 20031 (SUPERCOMPUTER SYSTEMS LTD.) 26. Dezember 1991 siehe das ganze Dokument ---	1-4
A	EP,A,0 501 653 (SUN MICROSYSTEMS) 2. September 1992 siehe das ganze Dokument ---	1-3
A	SIGPLAN NOTICES Bd. 21, Nr. 7, Juli 1986, PALO ALTO, US, Seiten 11 - 16 GIBBONS AND MUCHNICK 'Efficient instruction scheduling for a pipelined architecture' in der Anmeldung erwähnt siehe das ganze Dokument ---	1-3
	-/--	
<input checked="" type="checkbox"/> Weitere Veröffentlichungen sind der Fortsetzung von Feld C zu entnehmen <input checked="" type="checkbox"/> Siehe Anhang Patentfamilie		
* Besondere Kategorien von angegebenen Veröffentlichungen : "A" Veröffentlichung, die den allgemeinen Stand der Technik definiert, aber nicht als besonders bedeutsam anzusehen ist "E" älteres Dokument, das jedoch erst am oder nach dem internationalen Anmeldedatum veröffentlicht worden ist "L" Veröffentlichung, die geeignet ist, einen Prioritätsanspruch zweifelhaft erscheinen zu lassen, oder durch die das Veröffentlichungsdatum einer anderen im Recherchenbericht genannten Veröffentlichung belegt werden soll oder die aus einem anderen besonderen Grund angegeben ist (wie ausgeführt) "O" Veröffentlichung, die sich auf eine mündliche Offenbarung, eine Benutzung, eine Ausstellung oder andere Maßnahmen bezieht "P" Veröffentlichung, die vor dem internationalen Anmeldedatum, aber nach dem beanspruchten Prioritätsdatum veröffentlicht worden ist "T" Spätere Veröffentlichung, die nach dem internationalen Anmeldedatum oder dem Prioritätsdatum veröffentlicht worden ist und mit der Anmeldung nicht kollidiert, sondern nur zum Verständnis des der Erfindung zugrundeliegenden Prinzips oder der ihr zugrundeliegenden Theorie angegeben ist "X" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann allein aufgrund dieser Veröffentlichung nicht als neu oder auf erfinderscher Tätigkeit beruhend betrachtet werden "Y" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann nicht als auf erfinderscher Tätigkeit beruhend betrachtet werden, wenn die Veröffentlichung mit einer oder mehreren anderen Veröffentlichungen dieser Kategorie in Verbindung gebracht wird und diese Verbindung für einen Fachmann naheliegend ist "&" Veröffentlichung, die Mitglied derselben Patentfamilie ist		
Datum des Abschlusses der internationalen Recherche  14. Juni 1994		Absenddatum des internationalen Recherchenberichts  1994
Name und Postanschrift der Internationale Recherchenbehörde Europäisches Patentamt, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+ 31-70) 340-2040, Tx. 31 651 epo nl, Fax (+ 31-70) 340-3016		Bevollmächtigter Bediensteter  Weinberg, L

## C.(Fortsetzung) ALS WESENTLICH ANGESEHENE UNTERLAGEN

Kategorie°	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
A	IBM JOURNAL OF RESEARCH AND DEVELOPMENT Bd. 34, Nr. 1 , Januar 1990 , NEW YORK US Seiten 85 - 92 WARREN 'Instruction scheduling for the IBM system/6000 processor' siehe das ganze Dokument ---	1-3
A	COMPUTER ARCHITECTURE NEWS Bd. 21, Nr. 1 , März 1993 , NEW YORK US Seiten 4 - 11 INOUE AND TAKEDA 'Performance evaluation for various configurations of superscalar processors' siehe Seite 5, rechte Spalte, Absatz 4 - Seite 7, linke Spalte, Absatz 1 -----	1

## INTERNATIONALER RECHERCHENBERICHT

Angaben zu Veröffentlichungen, die zur selben Patentfamilie gehören

Internationales Aktenzeichen

PCT/DE 94/00221

Im Recherchenbericht angeführtes Patentdokument	Datum der Veröffentlichung	Mitglied(er) der Patentfamilie	Datum der Veröffentlichung
WO-A-9120031	26-12-91	US-A- 5179702	12-01-93
		AU-A- 8099891	07-01-92
		AU-A- 8201891	07-01-92
		EP-A- 0533805	31-03-93
		EP-A- 0535107	07-04-93
		EP-A- 0533813	31-03-93
		EP-A- 0537257	21-04-93
		JP-T- 5508494	25-11-93
		WO-A- 9120029	26-12-91
		WO-A- 9120030	26-12-91
		WO-A- 9120033	26-12-91
		WO-A- 9120036	26-12-91
		US-A- 5193192	09-03-93
		US-A- 5107418	21-04-92
		US-A- 5202988	13-04-93
		US-A- 5175856	29-12-92
		US-A- 5159678	27-10-92
		US-A- 5187789	16-02-93
		US-A- 5202975	13-04-93
		US-A- 5307478	26-04-94
-----	-----	-----	-----
EP-A-0501653	02-09-92	US-A- 5202993	13-04-93
-----	-----	-----	-----