(19) **Bundesrepublik Deutschland**
Deutsches Patent- und Markenamt

(10) **DE 695 33 020 T2** 2005.04.07

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 0 715 264 B1**
(21) Deutsches Aktenzeichen: **695 33 020.9**
(86) PCT-Aktenzeichen: **PCT/JP95/00552**
(96) Europäisches Aktenzeichen: **95 913 338.0**
(87) PCT-Veröffentlichungs-Nr.: **WO 95/026530**
(86) PCT-Anmeldetag: **24.03.1995**
(87) Veröffentlichungstag der PCT-Anmeldung: **05.10.1995**
(97) Erstveröffentlichung durch das EPA: **05.06.1996**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **12.05.2004**
(47) Veröffentlichungstag im Patentblatt: **07.04.2005**

(51) Int Cl.[7]: **G06F 15/16**
G06F 9/44

(30) Unionspriorität:

5714194	**28.03.1994**	**JP**
5714294	**28.03.1994**	**JP**

(84) Benannte Vertragsstaaten:
DE, FR, GB

(73) Patentinhaber:
Sony Corp., Tokio/Tokyo, JP

(72) Erfinder:
KATSUO, Satoshi, Tokyo 141, JP; SHIGATA, Taro, Tokyyo 141, JP

(74) Vertreter:
Mitscherlich & Partner, Patent- und Rechtsanwälte, 80331 München

(54) Bezeichnung: **VERFAHREN UND APPARAT ZUR KOMPILIERUNG PARALLELER BILDVERARBEITUNGSPROGRAMME**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung

[0001] Die Erfindung betrifft ein Kompilierungsverfahren für ein Parallelverarbeitungsprogramm, das Ausführungscodes eines unter Verwendung einer Programmiersprache aufgestellten Programms generiert, damit mehrere arithmetische Prozessoren eine parallele Bildverarbeitung ausführen können, ferner ein Bildverarbeitungsgerät für die Durchführung einer Bildverarbeitung durch Parallelverarbeitung mehrerer Operationen, sowie ein Bildverarbeitungsverfahren, das den Betrieb des Bildverarbeitungsgeräts ermöglicht.

[0002] In dem Parallelverarbeitungssystem führt der Programmentwickler bei der von dem Benutzer definierten Prozedur der Verteilung (Verarbeitung) auf mehrere Prozessoren, die diesen eine Parallelverarbeitung ermöglicht, die Programmbeschreibung und -erstellung im allgemeinen so durch, daß die verteilten Prozeduren von den jeweiligen betroffenen Prozessoren ausgeführt werden.

[0003] Wenn man davon ausgeht, daß die von dem Benutzer definierte Prozedur in das Parallelverarbeitungssystem geladen wird, erstellt der Programmentwickler ein Source-Programm in der Weise, daß die Verteilung der Last auf die jeweiligen Prozessoren in einem Parallelverarbeitungssystem im voraus in geeigneter Weise erfolgt, wie dies oben beschrieben wurde.

[0004] Deshalb muß der Programmentwickler das Source-Programm so beschreiben, daß die Last zweckmäßig auf die einzelnen Prozessoren verteilt wird, damit die Verarbeitung mit hoher Effizienz erfolgt.

[0005] Bei der Erstellung des Programms muß der Programmentwickler sich bei der Verteilung der Last auf die einzelnen Prozessoren jedoch der Zahl der Prozessoren bewußt sein, auf die die Last verteilt wird. Dies erschwert die Arbeit des Programmentwicklers.

[0006] Bei der Bildverarbeitung mit Hilfe eines Parallelverarbeitungssystems treten bei der Bildaufteilung Fälle auf, in denen die Zahl kleiner Bereiche, die durch das Teilen entstehen, größer ist als die Zahl der Prozessoren und die Zahl der kleinen Bereiche nicht durch die Zahl der Prozessoren geteilt werden kann (unteilbar ist), so daß sich in dem Bild ein Restbereich, d.h. ein Bruchteilbereich, ergibt. Wenn ein solcher Bruchteilbereich auftritt, wird veranlaßt, daß einer der Prozessoren eine Bruchteilverarbeitung für den Bruchteilbereich durchführt.

[0007] Wenn für den Fall, daß die Last verteilt wird, einem Teil der Prozessoren die Verarbeitung für die integrale Durchführung der Berechnung von Ergebnissen der Bildverarbeitung in kleinen Bereichen auferlegt wird, die durch das Unterteilen eines Bilds gewonnen werden, muß dieser Teil der Prozessoren über zwei Arten von Ausführungscodes verfügen, nämlich den über den Ausführungscode für die arithmetische Verarbeitung der kleinen Bereiche und über den oben beschriebenen Ausführungscode für die Integrationsverarbeitung für die integrale Durchführung der Operation. Da die Erstellung eines solchen Programms für den Programmentwickler zur Folge hat, ein Programm erstellen zu müssen, bei dem er auch die oben erwähnte Integrationsverarbeitung in Rechnung stellen muß, ist seine Arbeitsbelstung noch größer als die oben erwähnte Arbeit.

[0008] Darüber hinaus werden dem Prozessor zwei Arten von Ausführungscodes zugeführt, so daß die Menge der Ausführungscodes für den Prozessor größer wird als die Menge der Ausführungscodes für die arithmetische Verarbeitung von kleinen Bereichen. Außerdem erfolgt die Integrationsverarbeitung zusätzlich, so daß die benutzte Speicherkapazität im wesentlichen zweimal aufgewendet wird.

[0009] Die vorliegende Erfindung entstand im Hinblick auf die oben beschriebenen realen Umstände, und es ist ihr Ziel, ein Kompilierungsverfahren für ein Parallelverarbeitungsprogramm zur Verfügung zu stellen, das ein Anwachsen der zu generierenden Ausführungscodes verhindert, ohne daß sie eine große Arbeitsbelastung für den Programmentwickler zur Folge hat.

[0010] Es ist ein weiteres Ziel der Erfindung, ein Bildverarbeitungsgerät und ein Bildverarbeitungsverfahren zu schaffen, die sich für die Generierung von Ausführungscodes, z.B. in Abhängigkeit von dem benutzten Anwendungsprogramm usw., eignen und so eine Bildverarbeitung ermöglichen, die effizienter ist als beim Stand der Technik.

[0011] Aspekte der vorliegenden Erfindung sind in den anliegenden Ansprüchen angegeben.

[0012] Im Hinblick auf die oben beschriebenen Probleme ist ein Kompilierverfahren für ein Parallelverarbeitungsprogramm gemäß einem Ausführungsbeispiel der Erfindung auf ein Kompilierungsverfahren für ein Par-

allelverarbeitungsprogramm gerichtet, das ein unter Verwendung einer Programmiersprache aufgestelltes Programm kompiliert, um es einer Mehrzahl von arithmetischen Prozessoren zu ermöglichen, eine parallele Bildverarbeitung auszuführen, um Ausführungscodes zu generieren, wobei das Verfahren umfaßt: einen Analysierschritt zur Durchführung einer Identifizierung und Analyse einer Steueranweisung aus dem aufgestellten Programm, an die ein Identifizierer angehängt ist, der Parallelverarbeitung bei der Bildverarbeitung anzeigt, und einen Ausführungscode-Generierschritt zum Generieren von Parallelverarbeitungs-Ausführungscodes für die betreffenden arithmetischen Prozessoren nach Maßgabe des in dem Analysierschritt gewonnenen Analyseergebnisses und der Zahl der mehreren arithmetischen Prozessoren.

[0013] Das Bildverarbeitungsverfahren gemäß einem Ausführungsbeispiel der Erfindung ist auf ein Bildverarbeitungsverfahren zur Durchführung einer Bildverarbeitung nach Maßgabe von Ausführungscodes eines unter Verwendung einer Programmiersprache aufgestellten Programms gerichtet, das es einer Mehrzahl von Prozessoren ermöglicht, die Bildverarbeitung parallel durchzuführen, wobei das Verfahren umfaßt: einen Analysierschritt zur Durchführung einer Identifizierung und Analyse einer Steueranweisung aus dem aufgestellten Programm, an die ein Identifizierer angehängt ist, der Parallelverarbeitung bei der Bildverarbeitung anzeigt, einen Ausführungscode-Generierschritt zum Generieren von Parallelverarbeitungs-Ausführungscodes für die betreffenden arithmetischen Prozessoren nach Maßgabe des in dem Analysierschritt gewonnenen Analyseergebnisses und der Zahl der mehreren arithmetischen Prozessoren und einen Bildverarbeitungsschritt, in dem die arithmetischen Prozessoren jeweils eine Bildverarbeitung auf der Basis der erstellten Ausführungscodes durchführen.

[0014] Ein Bildverarbeitungsgerät nach einem Ausführungsbeispiel der Erfindung ist auf ein Bildverarbeitungsgerät gerichtet, das sich für die Durchführung der Bildverarbeitung durch Parallelverarbeitung mehrerer Operationen eignet, wobei das Gerät aufweist: eine Mehrzahl von arithmetischen Mitteln jeweils zur Durchführung paralleler Operationen der Bildverarbeitung für entsprechende kleine Bereiche, die durch Unterteilen eines Bilds eines zugeführten Bildsignals gewonnen werden, sowie eine Steuereinrichtung für die Datenzuführung zur Durchführung einer integralen Operation auf der Basis der jeweiligen Operationsergebnisse aus den arithmetischen Mitteln.

[0015] **Fig.** 1 zeigt ein Blockdiagramm, aus dem die Konfiguration eines Bildverarbeitungsgeräts nach einem Ausführungsbeispiel der Erfindung hervorgeht,

[0016] **Fig.** 2 zeigt ein Hauptflußdiagramm zur Erläuterung der Arbeitsweise des Bildverarbeitungsgeräts,

[0017] **Fig.** 3 zeigt ein Flußdiagramm zur Erläuterung der Verarbeitungsprozedur der Subroutine 1 in dem Hauptflußdiagramm,

[0018] **Fig.** 4 zeigt eine Modellansicht zur Erläuterung der Beziehung zwischen der Bildkonfiguration des Bildverarbeitungsgeräts und kleiner unterteilter Bereiche sowie die Art der Bewegung des kleinen unterteilten Bereichs,

[0019] **Fig.** 5 zeigt ein Flußdiagramm zur Erläuterung der Verarbeitungsprozedur der Subroutine 2 in dem Hauptflußdiagramm,

[0020] **Fig.** 6 zeigt eine Modellansicht eines Bildbeispiels für den Fall, daß in der Subroutine 2 ein Parallelverarbeitungs-Flag gesetzt ist,

[0021] **Fig.** 7 zeigt ein Flußdiagramm zur Erläuterung der Verarbeitungsprozedur der Subroutine 3 in dem Hauptflußdiagramm.

[0022] Anhand der anliegenden Zeichnungen werden nun Ausführungsbeispiele eines Kompilierverfahrens für ein Parallelverarbeitungsprogramm, ein Bildverarbeitungsgerät und ein Bildverarbeitungsverfahren gemäß der Erfindung beschrieben.

[0023] Das Kompilierverfahren für ein Parallelverarbeitungsprogramm gemäß der Erfindung ist ein Verfahren, das folgende Schritte umfaßt: In einem Analyseschritt wird ein erstelltes Programm analysiert, in einem Ausführungscode-Generierungsschritt wird zur Generierung von Ausführungscodes zwischen dem Programm der arithmetischen Prozessoren und Programmen mit Ausnahme des obigen auf der Basis des Analyseergebnisses aus dem Analyseschritt unterschieden, um dadurch Ausführungscodes der Prozessoren zu generieren, die die jeweiligen Funktionen ausführen.

[0024] In diesem Ausführungsbeispiel wird ein Beispiel eines Bildverarbeitungsgeräts (Einheit) beschrieben, bei dem das oben erwähnte Verfahren angewendet wird.

[0025] Die Bildverarbeitungseinheit umfaßt, wie in **Fig.** 1 dargestellt, z.B. einen Einheits-Steuerteil **1** zum Steuern der Gesamteinheit, einen Datenfluß-Steuerteil **2** zum Steuern des Datenflusses, einen Arithmetikteil **3** mit einer Mehrzahl von arithmetischen Prozessoren, einen Programmpuffer **4** und einen Eingabe-/Ausgabeteil **5**.

[0026] Der Einheits-Steuerteil **1** steuert den Programmpuffer **5** für das Laden von Programmen entsprechend dem Datenfluß-Steuerteil **2** und dem Arithmetikteil **3** in den Programmpuffer **4** in dem Programmspeicher innerhalb des Prozessors des Datenfluß-Steuerteils **2** und das Speichern von Programmen in den jeweiligen arithmetischen Prozessoren des Arithmetikteils **3** über einen Speicherbus **4**. Darüber hinaus führt der Einheits-Steuerteil **3** eine Ausführungssteuerung der jeweiligen arithmetischen Prozessoren des Datenfluß-Steuerteils **2** und des Arithmetikteils **3** durch und die Taktzuführung an diese.

[0027] Der Datenfluß-Steuerteil **2** führt auf der Basis der jeweiligen Operationsergebnisse aus dem Arithmetikteil **3** eine integrale Operation durch und verteilt und liefert das Ergebnis dieser Operation nach Bedarf an den Arithmetikteil **3**. Dies wird nun praxisnäher erläutert. Der Datenfluß-Steuerteil **2** liefert ein Steuersignal an einen (nicht dargestellten) gemeinsam genutzten Speicher innerhalb des Eingabe-/Ausgabeteils **5**. Auf diese Weise führt der Datenfluß-Steuerteil **2** eine Steuerung durch, durch die Daten aus dem gemeinsam genutzten Speicher über einen Datenbus **7** auf die (nicht dargestellten) verteilten Speicher in den jeweiligen arithmetischen Prozessoren verteilt werden. Darüber hinaus führt der Datenfluß-Steuerteil **2** das Senden und den Empfang von Steuersignalen für die jeweiligen arithmetischen Prozessoren des Arithmetikteils **3** durch, um das Verarbeitungsergebnis aus dem arithmetischen Prozessor zu sammeln, in dem die arithmetische Verarbeitung bezüglich der gelieferten Daten abgeschlossen wurde. Dann führt der Datenfluß-Steuerteil **2** eine Steuerung durch, um Daten des Verarbeitungsergebnisses über den Datenbus **7** an den gemeinsam genutzten Speicher in dem Eingabe-/Ausgabeteil **5** zu liefern. Der Datenfluß-Steuerteil **2** führt eine Steuerung durch, um die nächsten Daten über den Datenbus **7** aus dem gemeinsam genutzten Speicher für den arithmetischen Prozessor zu verteilen, in dem die arithmetische Verarbeitung abgeschlossen wurde. Der Datenfluß-Steuerteil **2** wiederholt das Verteilen und Sammeln von Daten in der oben beschriebenen Weise, um eine Steuerung durchzuführen, bis das Programm des Datenfluß-Steuerteils **2** abgeschlossen wurde.

[0028] Wie oben beschrieben wurde, ist die Bildverarbeitungseinheit so ausgebildet, daß ein angeliefertes Bildsignal einer Parallelverarbeitung unterzogen wird, so daß die Geschwindigkeit der Bildverarbeitung gesteigert wird. Darüber hinaus ist die Bildverarbeitungseinheit so ausgebildet, daß die jeweiligen arithmetischen Prozessoren an einer Vielzahl von kleinen Bereichen, die durch Unterteilen der Bilddaten gewonnen werden, das gleiche Programm ausführen. Das heißt, die Bildverarbeitungseinheit enthält mehrere arithmetischen Prozessoren zur Ausführung des gleichen Programms und führt so eine parallele Bildverarbeitung (Bilddarstellung) mit hoher Geschwindigkeit durch mittels einer Architektur, die ein einzelnes Bild in kleine Bereiche unterteilt, um eine Parallelverarbeitung durchzuführen.

[0029] Für die Verarbeitung der Bilder ist eine Bildverarbeitung vorgesehen, die nicht innerhalb kleiner unterteilter Bereiche abgeschlossen wird, um so dem Modell zu entsprechen, das dieser Architektur genügt. Als Bildverarbeitung dieser Art ist z.B. der Fall vorgesehen, bei dem der mittlere Dichtewert aller Bilder bestimmt wird oder dgl.

[0030] Wenn die Last der Bildverarbeitung in der Bildverarbeitungseinheit auf mehrere arithmetische Prozessoren verteilt wird, so daß ein Teil der arithmetischen Prozessoren eine Integrationsverarbeitung ausführen kann, bei der die Ergebnisse der betreffenden Bildverarbeitung der Lastverteilung gesammelt werden, benötigt der Arithmetikteil **3** zwei Arten von Ausführungscodes, nämlich einen Ausführungscode für den arithmetischen Prozessor zur Verarbeitung von unterteilten Bereichen, und einen Ausführungscode für den arithmetischen Prozessor zur Integrationsverarbeitung. Die auf diese Weise generierten Ausführungscodes führen nicht zu einem Modell, das der oben beschriebene Architektur genügt, und die Menge der generierten Ausführungscodes vergrößert sich.

[0031] Im Hinblick hierauf ermöglicht die Bildverarbeitungseinheit, bei der die Erfindung angewendet wird, den jeweiligen arithmetischen Prozessoren des Arithmetikteils **3**, ein geteiltes Programm auf der Basis eines Modells auszuführen, das der Architektur genügt, um die betreffenden arithmetischen Ergebnisse in den Datenfluß-Steuerteil **3** zu übernehmen und eine Integrationsverarbeitung durchzuführen und ferner eine Verarbeitung durchzuführen, durch die dieses Integrationsverarbeitungsergebnis an die betreffenden arithmetischen

Prozessoren des Arithmetikteils **2** ausgeliefert wird, so daß diese eine Operation ausführen können usw.

[0032] Für den Fall, daß die Zahl der zu verarbeitenden Bilder groß ist, wurde auch ein Verfahren vorgeschlagen, bei der Bilder den jeweiligen arithmetischen Prozessoren Stück für Stück zugeteilt werden. Falls die Zahl der zu verarbeitenden Bilder durch die Zahl der arithmetischen Prozessoren teilbar ist, gibt es keinen arithmetischen Prozessor, der während der Parallelverarbeitung leerläuft, so daß der Grad der Parallelverarbeitung maximal verbessert wird. Darüber hinaus wird auch eine Kommunikation zwischen dem Datenfluß-Steuerteil **2** und den arithmetischen Prozessoren des Arithmetikteils **3** überflüssig, wie sie bei der oben beschriebenen Bildteilungsverarbeitung erforderlich war. Für den Fall, daß die Zahl der zu verarbeitenden Bilder durch die Zahl der arithmetischen Prozessoren teilbar ist, ist aus diesem Grund die Verarbeitungsgeschwindigkeit der parallelen Bildverarbeitung, die den jeweiligen arithmetischen Prozessoren jeweils in Einzelbildern zugeteilt wird, höher.

[0033] In der Konfiguration der Bildverarbeitungseinheit und bei der quantitativen Beziehung des Bildsignals gibt es jedoch Fälle, in denen dann, wenn die Zahl der Bilder eines zugeführten Bildsignals durch die Zahl der arithmetischen Prozessoren des Arithmetikteils **3** geteilt wird, die Zahl der Bilder nicht durch die Zahl der arithmetischen Prozessoren teilbar ist, so daß ein Rest bleibt.

[0034] Wenn sich der Zustand ergibt, in dem eine solche Beziehung gilt, gibt es Fälle, in denen in der Bildverarbeitungseinheit der Parallelismus (der Grad der Parallelverarbeitung) extrem verringert wird, so daß die Effizienz der Bildverarbeitung beträchtlich herabgesetzt wird.

[0035] Als praxisnahes Beispiel werde nun der Fall betrachtet, daß Bilder (Bilddarstellungen) von 17 Rahmen in einer Konfiguration verarbeitet werden, bei der der Arithmetikteil **3** z.B. 16 arithmetische Prozessoren enthält. Dann wird bei der Verarbeitung des 17. Rahmens nur einer der arithmetischen Prozessoren in dem Arithmetikteil **3** tätig, während 15 arithmetische Prozessoren inaktiv sind. Deshalb befinden sich diese arithmetischen Prozessoren im Leerlauf. Dies hat zur Folge, daß die Effizienz der Bildverarbeitung verringert wird.

[0036] Im Hinblick darauf verfügt die vorliegende Bildverarbeitungseinheit über Verarbeitungsmodi in Verbindung sowohl mit dem Fall, daß ein Bild in kleine Bereiche unterteilt ist, damit die arithmetischen Prozessoren in Übereinstimmung mit dem Modell, das der oben beschriebene Architektur genügt, in jedem dieser kleinen Bereiche tätig werden können, als auch in Verbindung mit dem Fall, daß Bilder jeweils als Einzelbilder den entsprechenden arithmetischen Prozessoren zugeordnet werden, damit diese aktiv werden können, wobei dieser Fall nicht mit dem Modell konform ist, das der Architektur genügt.

[0037] Die Verarbeitung für die verschiedenen Fälle wird z.B. von einem Host-Computer **10** durchgeführt, der, wie in **Fig.** 1 dargestellt, die Kompilierung vornimmt. Darüber hinaus führt der Host-Computer **10** auch die Verarbeitung zu verschiedenen Fällen der Kompilierung durch, um die Bildteilungsverarbeitung, einschließlich der Integrationsverarbeitung, auf Basis des Modells der oben beschriebenen Architektur auszuführen, sowie eine Kompilierung für das Zuordnen der Verarbeitung aller jeweiligen Bilder zu arithmetischen Prozessoren, um dadurch eine Verarbeitungseffizienz zu erreichen, die höher ist als beim Stand der Technik.

[0038] Es folgt nun eine Erläuterung in Verbindung mit einem Bildverarbeitungsverfahren, die es der oben erwähnten Bildverarbeitungseinheit ermöglicht, eine Hochgeschwindigkeits-Bildverarbeitung durch Parallelverarbeitung durchzuführen, wobei auf die Flußdiagramme von **Fig.** 2, 3 und 5 Bezug genommen wird, sowie eine Erläuterung in Verbindung mit von der Bildverarbeitungseinheit verarbeiteten Pixeln und deren Blockkonfiguration, wobei auf **Fig.** 4 Bezug genommen wird. Wie oben beschrieben wurde, werden Ausführungscodes eines Parallelverarbeitungsprogramms, die nach Kompilierung an die Bildverarbeitungseinheit geliefert werden, von dem Host-Computer **10** generiert, der außerhalb der Bildverarbeitungseinheit angeordnet ist.

[0039] Der Host-Computer **10** beginnt mit der Kompilierung des Parallelverarbeitungsprogramms, wie dies z.B. in **Fig.** 2 dargestellt ist. Dann geht die Verarbeitungsprozedur weiter zu der Subroutine SUB1.

[0040] In dieser Subroutine SUB1 führt der Host-Computer **10** in dem Bildverarbeitungs-Source-Programm, in dem die Prozedur des Parallelverarbeitungsprogramms beschrieben ist, eine Identifizierung und Analyse der Steueranweisung durch, an die ein Identifizierer angehängt ist, der Parallelverarbeitung anzeigt. Die Verarbeitungsprozedur geht dann weiter zu der Subroutine SUB2.

[0041] In der Subroutine SUB2 unterscheidet der Host-Computer anhand der Beziehung zwischen der Zahl der Bilder, die der Bildverarbeitung unterzogen werden, und der Zahl der arithmetischen Prozessoren des

Arithmetikteils **3** und des Analyseergebnisses der Subroutine SUB1, ob die Bildverarbeitung durch Bildteilungsverarbeitung oder durch parallele Bildverarbeitung ausgeführt wird. Die Verarbeitungsprozedur geht dann weiter zu der Subroutine SUB3.

[0042] In der Subroutine SUB3 generiert der Host-Computer **10** Ausführungscodes auf der Basis des Ergebnisses der Subroutine SUB2.

[0043] Der Host-Computer **10** führt die Verarbeitung in der Reihenfolge dieser drei Subroutinen SUB1 bis SUB3 durch, um dadurch das Kompilieren des Parallelverarbeitungsprogramms abzuschließen. Dann geht die Verarbeitungsprozedur weiter zu dem Schritt S1.

[0044] In dem Schritt S1 liefert der Host-Computer **10** die in den Subroutinen SUB1 bis SUB3 kompilierten Ausführungscodes an den Einheits-Steuerteil **1** der Bildverarbeitungseinheit. Der Einheits-Steuerteil **1** speichert dann (temporär) einmal die Ausführungscodes aus dem Host-Computer **10** in den Programmpuffer **4**, um anschließend eine Steuerung durchzuführen, durch die die Ausführungscodes eines Programms, das dem Datenfluß-Steuerteil **2** und dem Arithmetikteil **3** in dem Programmpuffer **4** entspricht, über den Speicherbus **6** in den Programmspeicher in dem Prozessor des Datenfluß-Steuerteils **2** und den entsprechenden arithmetischen Prozessoren des Arithmetikteils **3** geladen werden.

[0045] In dem Schritt S2 verarbeiten die arithmetischen Prozessoren des Arithmetikteils **3** zweidimensionale Daten, die nach Maßgabe der Ausführungscodes über den Eingabe-/Ausgabeteil **5** in Einheiten kleiner Bereiche, die durch Teilen eines Bilds gewonnen werden, oder in Einheiten aus jeweils einem Rahmen angeliefert werden. Der arithmetische Prozessor des Datenfluß-Steuerteils **2** führt nach Maßgabe der Ausführungscodes das Zuführen/Sammeln von Daten und/oder die Integrationsverarbeitung von Operationsergebnissen des Arithmetikteils **3** durch. Durch eine solche Serie von arithmetischen Verarbeitungen wird die Bildverarbeitung mit hoher Geschwindigkeit durchgeführt.

[0046] Im folgenden wird die Prozedur der oben erwähnten Subroutinen SUB1 bis SUB3 beschrieben.

[0047] Im vorliegenden Beispiel wird in dem Source-Programm des Parallelverarbeitungsprogramms eine mit der sog. C-Sprache konforme Sprache benutzt, die eine höhere Sprache ist.

[0048] In der Subroutine SUB1 analysiert das aufgestellte Source-Programm, ob es innerhalb der "Funktion" eine integrale arithmetische Verarbeitung gibt oder nicht, die eine arithmetische Verarbeitung darstellt, bei der die in mehreren kleinen Bereichen durch die Benutzung geteilter zweidimensionalen Daten gewonnenen Operationsergebnisse zur Durchführung einer integralen Operation verwendet werden, und ob es eine Verteilungsverarbeitung gibt oder nicht, bei der das Ergebnis der oben erwähnten integralen Operation auf die betreffenden arithmetischen Prozessoren verteilt wird usw..

[0049] In dem Schritt S10 des Flußdiagramms von **Fig.** 3 analysiert der Host-Computer z.B. die Konfigurationsdatei (im folgenden kurz als CONFIG.FILE bezeichnet) für die Bildverarbeitungseinheit. In der CONFIG.FILE ist als Header-Information z.B. eine Information wie die Zahl der arithmetischen Prozessoren in dem Arithmetikteil **3** der Bildverarbeitungseinheit **2** angegeben. Dann wird die Gesamtzahl der arithmetischen Prozessoren in der Bildverarbeitungseinheit unter Verwendung dieser Information geprüft.

[0050] In dem Schritt S11 analysiert der Host-Computer dann das Source-Programm des Parallelverarbeitungsprogramms.

[0051] In der Sprache zur Beschreibung des Source-Codes des Parallelverarbeitungsprogramms werden vfor und hfor für neue Parallelverarbeitung benutzt, wobei "v" und "h" als Identifizierer vorgestellt werden, welche als Schlüsselwörter dienen, die die Durchführung paralleler Verarbeitung bei der Bildverarbeitung anregen und Wiederholung in vertikaler Richtung bzw. Wiederholung in horizontaler Richtung bezeichnen, z.B. für eine Anweisung der C-Sprache, die wiederholte Verarbeitung bezeichnet. Parameter innerhalb der Klammern der vfor-Anweisung und der hfor-Anweisung sind die gleichen wie für die Anweisung der C-Sprache.

[0052] In ähnlicher Weise sind die Parameter (Variablen) Vinit und Hinit für eine neue Parallelverarbeitung, die sich auf durch Teilen eines Bilds gewonnene kleine Bereiche bezieht, Parameter, die Koordinaten in vertikaler Richtung bzw. in horizontaler Richtung in der linken oberen Position des kleinen Bereichs bezeichnen.

[0053] Als ein Beispiel des Source-Programms wird nun das im folgenden beschriebene Formatbeispiel be-

trachtet.

```
.....
 vfor(Vinit=0; Vinit<512; Vinit+=8){                    ...  (1-1-1)
  hfor(Hinit=0 ; Hinit<768; Hinit+=8){                  ...  (1-1-2)
       read(x, 8, 8);                                   ...  (1-1-3)
       .....
       tmp=x[0]+ ....                                   ...  (1-1-4)
       .....
       write(y, 8, 8);                                  ...  (1-1-5)
       .....
        total=total+tmp;                                ...  (1-1-6)
   }                                                    ...  (1-1-7)
 }                                                      ...  (1-1-8)
.....
 vfor(Vinit=0; Vinit<512; Vinit+=8){                    ...  (1-2-1)
  hfor(Hinit=0; Hinit<768; Hinit+=8){                   ...  (1-2-2)
       read(x, 8, 8);                                   ...  (1-2-3)
       .....
       tmp=total+ .....                                 ...  (1-2-4)
       .....
       write(y, 8, 8);                                  ...  (1-2-5)
       .....
 }                                                      ...  (1-2-6)
}                                                       ...  (1-2-7)
```

[0054] Parameter innerhalb der Klammern der vfor-Anweisung und der hfor-Anweisung des Source-Programms, d.h. Vinit=0; Vinit<512 und Hinit=0; Hinit<768 zeigen an, daß das Bild, wie in **Fig.** 4(a) dargestellt, aus 512×768 Pixeln besteht. Vinit+=8 und Hinit+=8 zeigen an, daß die kleinen Bereiche, die durch Unterteilen des Bilds gewonnen werden, aus Blöcken der Größe 8×8 Pixeln bestehen. Vinit=0 und Hinit=0 zeigen an, daß das Lesen oder Schreiben aus/in kleinen Bereichen durch die hfor-Anweisung zuerst von dem linken oberen zu dem rechten Ende des Bilds ausgeführt wird und dann durch die vfor-Anweisung das Lesen oder Schreiben aus/in dem kleinen Bereich an der linken Endposition ausgeführt wird, die der nächsten Zeile entspricht.

[0055] Für das Lesen oder Schreiben des kleinen Bereichs werden die Lese-Anweisung und die Schreib-Anweisung benutzt. Das erste Argument x der Lese-Anweisung und der Schreib-Anweisung ist der Puffer-(Anordnungs)-Name zum Speichern des kleinen Bereichs, das zweite Argument ist die Festlegung der Größe in horizontaler Richtung, und das dritte Argument ist die Festlegung der Größe in vertikaler Richtung.

[0056] Wenn die Aufmerksamkeit auf die vfor-Anweisung der Anweisungen (1–1–1), (1–2–1), die hfor-Anweisung der Anweisungen (1–1–2), (1–2–2), die Lese-Anweisung der Anweisungen (1-1–3), (1–2–3) und die Schreib-Anweisung der Anweisungen (1–1–5), (1–2–5) gerichtet wird, kann der Host-Computer **10** in dem Schritt S11 durch Analyse des Source-Programms die Parameter des kleinen Bereichs erkennen, die für die Generierung der Ausführungscodes des Datenfluß-Steuerteils **2** erforderlich sind. Das heißt, der Host-Computer **10** kann durch Analyse des Source-Programms die Größen in horizontaler/vertikaler Richtung, die Anfangsposition und die Art der Bewegung eines kleinen Bereichs erkennen. Die Gesamtzahl der zu verarbeitenden kleinen Bereiche des Gesamtbilds ist aus der vfor-Anweisung und der hfor-Anweisung ersichtlich.

[0057] In dem Schritt S12 erstellt der Host-Computer **10** dann auf der Basis des Analyseergebnisses von Schritt S11 eine Informationstabelle des Datenfluß-Steuerteils **2**. Dann geht die Verarbeitungsprozedur weiter

zu dem Schritt S13.

[0058] In dem Schritt S13 erstellt der Host-Computer **10** eine Informationstabelle für die Verarbeitung des arithmetischen Prozessors innerhalb des Source-Programms. Die Informationstabelle für die Verarbeitung des arithmetischen Prozessors ist eine Tabelle, in der Informationen gesammelt werden, die in dem Source-Programm für die Verarbeitung der betreffenden kleinen Bereiche durch Analyse, z.B. des von dem Syntax-Schlüsselwort vfor der Anweisung (1–2–1) oder hfor der Anweisung (1–2–1) usw. umschlossenen Source-Programm gewonnen werden.

[0059] In dem Schritt S14 entnimmt der Host-Computer **10** dann die Information des ersten arithmetischen Blocks. Das heißt, der Host-Computer **10** liest eine Information ein, die die arithmetische Verarbeitung angibt, die von dem arithmetischen Prozessor bezüglich kleiner Bereiche ausgeführt wird, z.B. die Anweisung (1–1–4) zwischen der Lese-Anweisung, der Anweisung (1–1–3), und der Schreib-Anweisung der Anweisung (1–1–5), oder die Anweisung (1–2–4) zwischen der Lese-Anweisung der Anweisung (1–2–3), und der Schreib-Anweisung der Anweisung (1–2–5). Der arithmetische Block ist hier ein Block, der von vfor und hfor eingeschlossen ist und den Operationsinhalt beschreibt, den die arithmetischen Prozessoren ausführen.

[0060] In dem Schritt S15 prüft der Host-Computer **10** dann, ob die Berechnungsformel zur Durchführung der Integrationsverarbeitung in dem arithmetischen Block, der in dem Schritt S14 ausgelesen wurde, enthalten ist oder nicht. Für den Fall, daß keine Berechnungsformel, z.B. die Anweisung (1–1–6), zur Durchführung der Integrationsverarbeitung bezüglich der Bildverarbeitung des kleinen Bereichs enthalten ist (NEIN), geht die Verarbeitungsprozedur weiter zu dem Schritt S17. Für den Fall hingegen, daß eine Berechnungsformel, wie z.B. die Anweisung (1–1–6) enthalten ist (JA), geht die Verarbeitungsprozedur weiter zu dem Schritt S16.

[0061] In dem Schritt S16 stellt der Host-Computer **10** z.B. das Integrations-Flag auf (setzt es auf "1"), so daß der arithmetische Prozessor in dem Datenfluß-Steuerteil **2** eine Operation ausführen kann, die sich auf die Berechnungsformel bezieht. Dann geht die Verarbeitungsprozedur weiter zu dem Schritt S17.

[0062] Es ist zu beachten, daß der Host-Computer **10** in dem Schritt S15 auch das Vorhandensein oder Nichtvorhandensein der Verteilung des Ergebnisses der integralen Operation prüfen kann. Das heißt, für den Fall, daß keine Berechnungsformel vorliegt, wie z.B. die Anweisung (1–2–4) (NEIN), geht die Verarbeitungsprozedur des Host-Computers **10** weiter zu dem Schritt S17. Wenn hingegen eine Berechnungsformel vorliegt, wie die Anweisung (1–2–4) (JA), kann die Verarbeitungsprozedur des Host-Computers **10** zu dem Schritt S16 weitergehen, um in dem Schritt S16 das Verteilungs-Flag aufzustellen (zu setzen) und zu dem Schritt S17 überzugehen.

[0063] In dem Schritt S17 prüft der Host-Computer **10**, ob ein Bruchteil erzeugt wird oder nicht, wenn die arithmetischen Prozessoren kleinen Bereichen des Bilds entsprechen können. Das heißt dieser Bruchteil zeigt den Rest an, der entsteht, wenn bei der Teilung des Bilds in kleine Bereiche die Gesamtzahl der kleinen Bereiche nicht durch die Gesamtzahl der arithmetischen Prozessoren teilbar ist. Die Bruchteilzahl für den kleinen Bereich bedeutet in dem Source-Programm für die Verarbeitung der betreffenden kleinen Bereiche, daß sie kleiner ist als die Zahl der arithmetischen Prozessoren.

[0064] Die Verarbeitungsprozedur des Host-Computers **10** geht dann weiter zu dem Schritt S18, wenn der Bruchteil existiert (JA) bzw. zu dem Schritt S19, wenn kein Bruchteil existiert (NEIN).

[0065] In dem Schritt S18 setzt der Host-Computer **10** das Bruchteil-Flag. Die Verarbeitungsprozedur geht so weiter zu dem Schritt S19.

[0066] In dem Schritt S19 prüft der Host-Computer **10**, ob die Analyseverarbeitung für alle arithmetischen Blöcke abgeschlossen wurde. Wenn die Analyseverarbeitung noch nicht für alle arithmetischen Blöcke abgeschlossen wurde (NEIN), geht die Verarbeitung weiter zu dem Schritt S19N.

[0067] In dem Schritt S19N liest der Host-Computer **10** die Information des nächsten arithmetischen Blocks ein, um die Analyse fortzusetzen. Nachdem der Host-Computer **10** die Information des nächsten arithmetischen Blocks eingelesen hat, geht die Verarbeitungsprozedur weiter zu dem Schritt S15. Wenn die Analyseverarbeitung in dem Schritt S19 hingegen für alle arithmetischen Blöcke abgeschlossen wurde (JA), geht die Verarbeitungsprozedur des Computers **10** auf Rücklauf, um die Subroutine SUB1 abzuschließen.

[0068] Es folgt eine Erläuterung in Verbindung mit der Parallelverarbeitung für ein Verarbeitungsprogramm,

einschließlich der Integrationsverarbeitung, z.B. für vier unten angegebene Bilder. Dieses Verarbeitungsprogramm hat z.B. das im folgenden beschriebene Format.

```
 ..........
  vfor(Vinit=0; Vinit<512*2; Vinit+=512){              ···  (2-1)
   hfor(Hinit=0; Hinit<768*2; Hinit+=768){             ···  (2-2)
    vfor(Vinit=0; Vinit<512; Vinit+=8){                ···  (2-3)
     hfor(Hinit=0; Hinit<768; Hinit+=8){               ···  (2-4)
         read(x, 8, 8);                                ···  (2-5)
         ..........
         tmp=x[0]+ ····                                ···  (2-6)
         ..........
         write(y, 8, 8);                               ···  (2-7)
         ..........
         total=total+tmp;                              ···  (2-8)
      }                                                ···  (2-9)
     }                                                 ···  (2-10)
    }                                                  ···  (2-11)
   }                                                   ···  (2-12)
...........
```

[0069] In diesem Verarbeitungsprogramm ist der Fall dargestellt, daß die Bildverarbeitung für vier Bilder durchgeführt wird, die jeweils z.B. aus 512×768 Pixeln bestehen, wie dies in **Fig.** 6 dargestellt ist, wenn in dem Arithmetikteil **3** z.B. vier arithmetische Prozessoren vorhanden sind. Bei einer solchen Operation, bei der alle Bilder als Einheit benutzt werden, gibt es Fälle, in denen eine Bildverarbeitung durchgeführt wird, um z.B. die mittlere Dichte aller Bilder usw. zu bestimmen. Um eine solche Bildverarbeitung auf der Basis des der oben beschriebenen Architektur entsprechenden Modells durchzuführen, unterscheidet der Host-Computer **10** in der Subroutine SUB2 zwischen Bildteilungsverarbeitung, einschließlich Integrationsverarbeitung, die in der SUB1 analysiert wird, und paralleler Bildverarbeitung, um die Bildoperation (Bilddarstellung) durch entsprechende Prozessoren für jeweils eine Rahmeneinheit parallel durchzuführen.

[0070] Wie **Fig.** 5 zeigt, entnimmt der Host-Computer **10** in dem Schritt S20 des Flußdiagramms der Subroutine SUB2 die Information des ersten arithmetischen Blocks. Dann geht die Verarbeitungsprozedur weiter zu dem Schritt S21.

[0071] In dem Schritt S21 unterscheidet der Host-Computer **10** anhand des Vorhandenseins oder Nichtvorhandenseins des Integrationsverarbeitungs-Flags, ob Integrationsverarbeitung vorliegt oder nicht. Wenn Integrationsverarbeitung vorliegt, wie dies z.B. durch die Anweisung (2–8) angezeigt wird, da das Integrationsverarbeitungs-Flag gesetzt sein sollte (JA), geht die Verarbeitungsprozedur des Host-Computers **10** weiter zu dem Schritt S22. Wenn keine Integrationsverarbeitung vorliegt (NEIN), geht die Verarbeitungsprozedur weiter zu dem Schritt S25.

[0072] In dem Schritt S22 prüft der Host-Computer **10**, ob die Zahl der zu verarbeitenden Bilder gleich 1 ist oder nicht. Wenn die Zahl der zu verarbeitenden Bilder gleich 1 ist (JA), geht die Verarbeitungsprozedur des Host-Computers **10** weiter zu dem Schritt S25. Wenn die Zahl der zu verarbeitenden Bilder gleich 2 oder größer ist (NEIN), d.h. bei einem Programm, wie es z.B. durch die Anweisung (2–2) bis zu der Anweisung (2–5) angegeben ist, geht die Verarbeitungsprozedur weiter zu dem Schritt S23.

[0073] In dem Schritt S23 prüft der Host-Computer **10**, ob die Zahl der zu verarbeitenden Bilder durch die Zahl der arithmetischen Prozessoren in dem Arithmetikteil **3** teilbar ist oder nicht. Wenn die Zahl der zu verarbeitenden Bilder durch die Zahl der arithmetischen Prozessoren teilbar ist (JA), geht die Verarbeitungsprozedur des Host-Computers **10** weiter zu dem Schritt S24. Wenn die Zahl der zu verarbeitenden Bilder nicht durch die Zahl der arithmetischen Prozessoren in dem Arithmetikteil **3** geteilt werden kann (NEIN), geht die Verarbeitungspro-

zedur weiter zu dem Schritt S25.

[0074] In dem Schritt S24 setzt der Host-Computer **10** das Parallelverarbeitungs-Flag. Dann geht die Verarbeitungsprozedur weiter zu dem Schritt S25.

[0075] In dem Schritt S25 prüft der Host-Computer **10**, ob die Analyse für alle arithmetischen Blöcke abgeschlossen wurde oder nicht. Wenn die Analyse des arithmetischen Verarbeitungsverfahrens noch nicht für alle arithmetischen Blöcke abgeschlossen wurde (NEIN), geht die Verarbeitungsprozedur weiter zu dem Schritt S26.

[0076] In dem Schritt S26 liest der Host-Computer **10** die Information des nächsten arithmetischen Blocks ein, um die Analyse fortzusetzen. Nachdem der Host-Computer **10** die Information des nächsten arithmetischen Blocks eingelesen hat, geht die Verarbeitungsprozedur weiter zu dem Schritt S22. Wenn hingegen die Analyseverarbeitung für alle arithmetischen Blöcke in dem Schritt S25 abgeschlossen wurde (JA), geht die Verarbeitungsprozedur des Host-Computers **10** weiter auf Rücklauf. Damit ist die Subroutine SUB2 abgeschlossen.

[0077] Wenn in dieser Weise eine Unterscheidung zwischen der parallelen Bildverarbeitung und der Bildteilungsverarbeitung durchgeführt wird, sind Maßnahmen dagegen getroffen, daß das Modell, das z.B. auf der oben beschriebenen Architektur beruht, nicht verletzt wird, wenn in der nachfolgenden Subroutine SUB3 die Ausführungscode-Generierung für das Programm durchgeführt wird. Die dem Analyseergebnis entsprechende Kompilierung und die Steuerung der Verarbeitung des in dieser Weise ausgeführten Source-Programms erfolgt nach dem in **Fig.** 7 dargestellten Flußdiagramm der Subroutine SUB3.

[0078] In dem Schritt S30 prüft der Host-Computer **10**, ob das für die Implementierung der Kompilierung gelieferte Source-Programm ein Programm für den Datenfluß-Steuerteil **2** ist oder nicht. Falls das Source-Programm ein Programm für den Datenfluß-Steuerteil **2** ist (JA), geht die Verarbeitungsprozedur des Host-Computers **10** weiter zu dem Schritt S31. Für den Fall, daß das Source-Programm kein Programm für den Datenfluß-Steuerteil **2** ist (NEIN), wird es als Programm für einen arithmetischen Prozessor betrachtet. Dann geht die Verarbeitungsprozedur weiter zu dem Schritt S40.

[0079] In dem Schritt S31 prüft der Host-Computer **10**, ob das Flag für parallele Bildverarbeitung gesetzt ist oder nicht. Wenn das Flag für parallele Bildverarbeitung gesetzt ist (JA), prüft der Host-Computer **10**, ob angegeben ist, daß die Zahl der Bilder durch die Zahl der arithmetischen Prozessoren teilbar war. Dann geht die Verarbeitungsprozedur weiter zu dem Schritt S32. Wenn das Flag für parallele Bildverarbeitung nicht gesetzt ist (NEIN), geht die Verarbeitungsprozedur weiter zu dem Schritt S33.

[0080] In dem Schritt S32 liefert der Host-Computer **10** alle Daten eines Bildes an das entsprechende Exemplar der arithmetischen Prozessoren, um Ausführungscodes des Datenfluß-Steuerteils **2** zu generieren, der die Bildverarbeitung für mehrere Bildern parallel ausführt. Anschließend geht die Verarbeitungsprozedur weiter zu dem Schritt S43. Die Verarbeitung in diesem Schritt ist eine Verarbeitung, in der z.B. der mittlere Dichtewert eines Bilds oder dgl. als praxisnäherer Bildverarbeitung erfolgt. So werden als Ausführungscodes, die für den Datenfluß-Steuerteil **2** generiert werden, z.B. Ausführungscodes generiert, um die Daten eines Bilds zu senden, die für die Verarbeitung von einem arithmetischen Prozessor in dem Arithmetikteil **3** benötigt werden.

[0081] In dem Schritt S33 prüft der Host-Computer **10**, ob das Integrations-Flag gesetzt ist oder nicht. Wenn das Integrations-Flag nicht gesetzt ist (NEIN), geht die Verarbeitungsprozedur weiter zu dem Schritt S37. Wenn das Integrations-Flag gesetzt ist (JA), geht die Verarbeitungsprozedur weiter zu dem Schritt S34.

[0082] In dem Schritt S34 prüft der Host-Computer **10**, ob das Bruchteil-Flag gesetzt ist oder nicht. Wenn das Bruchteil-Flag gesetzt ist (JA), geht die Verarbeitungsprozedur weiter zu dem Schritt S35. Wenn kein Bruchteil-Flag gesetzt ist (NEIN), geht die Verarbeitungsprozedur weiter zu dem Schritt S36.

[0083] In dem Schritt S35 generiert der Host-Computer **10** Ausführungscodes des Datenfluß-Steuerteils **2**, der eine Bildteilungsverarbeitung, einschließlich einer Integrationsverarbeitung und einer Bruchteilverarbeitung vornimmt. Somit geht die Verarbeitung weiter zu dem Schritt S43.

[0084] Die Bildteilungsverarbeitung umfaßt eine Verarbeitung, um Daten, wie das durch die Durchführung einer integralen Verarbeitungsprozedur gewonnene Ergebnis oder dgl., als numerische Werte an die jeweiligen arithmetischen Prozessoren in dem Arithmetikteil **3** zu senden, wie dies die Gelegenheit erfordert, so daß die betreffenden arithmetischen Prozessoren eine Operation auf der Basis der ihnen zugeführten numerischen

Werte ausführen.

[0085] Die Ausführungscodes der Bildteilungsverarbeitung werden aus einem Source-Programm generiert, das so ausgebildet ist, daß es wiederholt eine Steuerung durchführt, bei der der Prozessor des Datenfluß-Steuerteils **2** die Daten kleiner Bereiche, die der Zahl der arithmetischen Prozessoren in dem Arithmetikteil **3** entsprechen, aus dem gemeinsam genutzten Speicher in dem Eingabe-/Ausgabeteil **5** ausliest, um sie an die verteilten Speicher in den betreffenden arithmetischen Prozessoren zu liefern und die entsprechenden verarbeiteten Daten in dem gemeinsam genutzten Speicher in dem Eingabe-/Ausgabeteil **5** zu sammeln, nachdem die Operationen der arithmetischen Prozessoren abgeschlossen wurden.

[0086] Der Ausführungscode der Bruchteilverarbeitung ist ein Ausführungscode, durch den die Zahl der wiederholt ausgeführten Verarbeitung, die für Bilddaten des Restbereichs durchgeführt wird, um 1 erhöht wird, so daß sie die bedingte Anweisung enthält, den Schreibzugriff für den gemeinsam genutzten Speicher des Eingabe-/Ausgabeteils **5** für das Datenverarbeitungsergebnis aus dem Abschnitt zu sperren, der über der Gesamtzahl der kleinen Bereiche liegt. Die Bruchteilverarbeitung bedeutet nämlich eine Verarbeitung, bei der der Datenfluß-Steuerteil **2** nur Operationsergebnisse bezüglich aller kleinen Bereiche in den gemeinsam genutzten Speicher des Eingabe-/Ausgabeteils **5** einschreibt und mit Ausnahme der obigen keine Ergebnisse in den gemeinsam genutzten Speicher des Eingabe-/Ausgabeteils 5 einschreibt.

[0087] In dem Schritt S36 generiert der Host-Computer **10** dann Ausführungscodes des Datenfluß-Steuerteils **2**, der eine Bildteilungsverarbeitung ausführt, einschließlich einer Integrationsverarbeitung. Dann geht die Verarbeitungsprozedur weiter zu dem Schritt S43.

[0088] Wenn hingegen die Verarbeitungsprozedur aufgrund des Entscheidungsergebnisses (NEIN) in dem Schritt S33 zu dem Schritt S37 übergeht, prüft der Host-Computer **10** auch in diesem Schritt S37, ähnlich wie in dem Schritt S34, ob das Bruchteil-Flag gesetzt ist oder nicht. Wenn das Bruchteil-Flag gesetzt ist (JA), geht die Verarbeitungsprozedur des Host-Computers **10** weiter zu dem Schritt S38. Wenn kein Bruchteil-Flag gesetzt ist (NEIN), geht die Verarbeitungsprozedur weiter zu dem Schritt S39.

[0089] In dem Schritt S38 generiert der Host-Computer **10** Ausführungscodes des Datenfluß-Steuerteils **2**, der eine Bildteilungsverarbeitung, einschließlich einer Bruchteilverarbeitung, ausführt. Dann geht die Verarbeitungsprozedur weiter zu dem Schritt S43.

[0090] In dem Schritt S39 generiert der Host-Computer **10** in üblicher Weise Ausführungscodes des Datenfluß-Steuerteils **2**. Dann geht die Verarbeitungsprozedur weiter zu dem Schritt S43.

[0091] In dem Schritt S40 prüft der Host-Computer **10** hingegen, ob ein Integrationsverarbeitungs-Flag gesetzt ist oder nicht. Wenn das Integrationsverarbeitungs-Flag gesetzt ist (JA), geht die Verarbeitungsprozedur des Host-Computers **10** weiter zu dem Schritt S41. Wenn kein Integrationsverarbeitungs-Flag gesetzt ist (NEIN), geht die Verarbeitungsprozedur weiter zu dem Schritt S42.

[0092] In dem Schritt S41 generiert der Host-Computer **10** Ausführungscodes für die arithmetischen Prozessoren des Datenfluß-Steuerteils **2** und des Arithmetikteils **3**, einschließlich der Integrationsverarbeitung. Dann geht die Verarbeitungsprozedur weiter zu dem Schritt S43.

[0093] In dem Schritt S42 generiert der Host-Computer **10** in üblicher Weise Ausführungscodes für die arithmetischen Prozessoren des Arithmetikteils **3**. Dann geht die Verarbeitungsprozedur weiter zu dem Schritt S43.

[0094] In dem Schritt S43 prüft der Host-Computer **10**, ob die Generierung von Ausführungscodes für die arithmetischen Prozessoren des Datenfluß-Steuerteils **2** und des Arithmetikteils **3** für alle arithmetischen Blöcken abgeschlossen wurde oder nicht. Wenn die Kompilierung noch nicht abgeschlossen wurde (NEIN), kehrt die Verarbeitungsprozedur des Host-Computers **10** zu dem Schritt S30 zurück, um die oben beschriebene Prozedur zu wiederholen. Wenn die Kompilierung hingegen abgeschlossen wurde (JA), geht die Verarbeitungsprozedur des Host-Computers **10** auf Rücklauf und beendet so die Subroutine SUB3, in der diese Verarbeitung zur Generierung der Ausführungscodes durchgeführt wird.

[0095] Wie oben festgestellt wurde, wird ein Lösungsweg beschritten, bei dem die Verarbeitung in Abhängigkeit von der Zahl der angelieferten Bilder und der Zahl der arithmetischen Prozessoren umgeschaltet wird, um die arithmetische Verarbeitung von Einheiten kleiner Bereiche oder eines Rahmens in den entsprechenden arithmetischen Prozessoren des Arithmetikteils **3** durchzuführen und eine integrale arithmetische Verarbeitung

durch den arithmetischen Prozessor des Datenfluß-Steuerteils **2** durchzuführen, um das Ergebnis der integralen arithmetischen Verarbeitung so, wie es die Gelegenheit erfordert, an die jeweiligen arithmetischen Prozessoren des Arithmetikteils **3** auszugeben und dadurch die betroffenen arithmetischen Prozessoren in die Lage zu versetzen, sich der Bildverarbeitung für den gegebenen Bereich zu widmen und so die Art der zu generierenden Ausführungscodes beschränken zu können, indem automatisch eine Methode mit höherer Effizienz gewählt wird, ohne das der oben beschriebene Architektur entsprechende Modell zu verletzen. Auf diese Weise kann die Menge der generierten Ausführungscodes so klein wie möglich gehalten werden, und außerdem ist eine Programmoptimierung möglich.

[0096] Darüber hinaus wird ein Lösungsweg beschritten, bei dem ein Identifizierer detektiert wird, der Parallelverarbeitung in dem Source-Programm anzeigt, um auf der Basis des Identifizierers eine Unterscheidung zu treffen zwischen der Verarbeitung des Datenfluß-Steuerteils **2** und derjenigen des Arithmetikteils **3**, und Ausführungscodes für eine Vielzahl von arithmetischen Prozessoren des Arithmetikteils **3** zu generieren und damit dem Programmentwickler die Möglichkeit zu verschaffen, das Programm zu schreiben, ohne daß er sich der Rollenverteilung für die jeweiligen arithmetischen Prozessoren bewußt sein muß, so daß die Zeit für die Programmentwicklung verkürzt werden kann. Dadurch können die Entwicklungskosten niedrig gehalten werden.

[0097] Es wird außerdem ein Lösungsweg beschritten, um die oben beschriebene Bruchteilverarbeitung entsprechend dem Bruchteil (Rest) durchzuführen, der in Bezug auf die Zahl der arithmetischen Prozessoren und der Zahl der unterteilten kleinen Bereiche festgelegt wird, so daß eine effiziente Verarbeitung mit einer Anzahl von Ausführungscodes möglich ist, die im Vergleich zu der herkömmlichen Menge an Ausführungscodes sehr niedrig gehalten wird.

[0098] Bei dem Bildverarbeitungsverfahren wird ein Lösungsweg beschritten, um nach einer Prozedur vorzugehen, bei der die Operationsergebnisse in den arithmetischen Prozessoren gesammelt werden, deren Rollen in Abhängigkeit von der Zahl der unterteilen Bilder verteilt werden, um eine Integrationsverarbeitung derselben durchzuführen und dadurch die Menge der generierten Ausführungscodes in dem Bildverarbeitungsverfahren stark reduzieren zu können, so daß die Effizienz der Bildverarbeitung verbessert werden kann. Auf diese Weise ist eine Entwicklung in kurzer Zeit möglich, und die Entwicklungskosten können niedrig gehalten werden.

[0099] Darüber hinaus wird bei dem Bildverarbeitungsverfahren ein Lösungsweg beschritten, bei dem die Zahl der zu verarbeitenden Bilder und die Zahl der arithmetischen Prozessoren entsprechend einer ausgeführten Verarbeitung, z.B. einer Verarbeitung der mittleren Dichte usw., geprüft wird, um ein Verarbeitungsverfahren zur Generierung der Ausführungscodes auszuwählen, so daß die Verarbeitungsgeschwindigkeit höher sein kann als bei der herkömmlichen Bildverarbeitung. Auf diese Weise wird z.B. auch in dem Anwendungsprogramm die Bildverarbeitung durchgeführt, ohne das Modell zu verletzen, das der Architektur genügt, die als Architektur für die Parallelverarbeitung benötigt wird, so daß die Verarbeitungsgeschwindigkeit höher sein kann als bei der herkömmlichen Bildverarbeitung.

[0100] Das Bildverarbeitungsgerät ist so ausgebildet, daß in den betreffenden arithmetischen Prozessoren des Arithmetikteils eine arithmetische Verarbeitung von Daten durchgeführt wird, die von dem Datenfluß-Steuerteil angeliefert wurden, um eine integrale Operation durchzuführen, indem die Verarbeitungsergebnisse der betreffenden arithmetischen Prozessoren benutzt werden, die in dem Datenfluß-Steuerteil gesammelt werden, um sie ein zweites Mal auf die betreffenden arithmetischen Prozessoren zu verteilen, wenn die Gelegenheit dies erfordert, so daß die betreffenden arithmetischen Prozessoren des Arithmetikteils sich der Verarbeitung für die Berechnung von gegebenen kleinen Bereichen widmen können, um die Bildverarbeitung durch Parallelverarbeitung der angelieferten Bildsignale zu implementieren und so eine höherwetige Bildverarbeitung durchführen zu können, als dies beim Stand der Technik möglich ist.

[0101] Bei der Kompilierung eines unter Verwendung einer Programmiersprache aufgestellten Programms, das es einer Mehrzahl von arithmetischen Prozessoren ermöglicht, die Bildverarbeitung parallel durchzuführen, um Ausführungscodes zu generieren, werden erfindungsgemäß die Identifizierung und die Analyse der Steueranweisung, an die ein Identifizierer angehängt ist, der Parallelverarbeitung bei der Bildverarbeitung anzeigt, von dem erstellen Programm ausgeführt, so daß Ausführungscodes für die Parallelverarbeitung der betreffenden arithmetischen Prozessoren generiert werden, die dem Analyseergebnis und der Zahl der arithmetischen Prozessoren entsprechen. Die jeweiligen arithmetischen Prozessoren führen dann die Bildverarbeitung auf der Basis der generierten Ausführungscodes durch, so daß die Operationsergebnisse der betroffenen arithmetischen Prozessoren einer Integrationsverarbeitung unterzogen werden können, so daß der Programmentwickler das Programm beschreiben (schreiben) kann, ohne daß er sich der Rollenverteilung der jeweiligen Prozessoren bewußt sein muß. Dadurch kann die Programmentwicklungszeit verkürzt werden. So kann

die Arbeitslast des Programmentwicklers verringert werden, so daß die Entwicklungskosten niedriger gehalten werden können.

Patentansprüche

1. Verfahren zum Kompilieren eines parallelen Verarbeitungsprogramms, wobei das parallele Verarbeitungsprogramm generierte Ausführungscodes aufweist, die so betrieben werden können, daß sie es einer Mehrzahl von arithmetischen Prozessoren ermöglichen, eine parallele Bildverarbeitung auszuführen, wobei das Verfahren die Verfahrensschritte aufweist:
(A) Analysieren einer Steueranweisung (SUB1) eines Quellprogramms, um festzustellen, ob daran ein Identifizierer für die parallele Bildverarbeitung angehängt ist, und
(B) wenn der Identifizierer für die parallele Bildverarbeitung an die Steueranweisung angehängt ist,
Generieren von Ausführungscodes (SUB3),
wobei der Verfahrensschritt (A) einen Verfahrensschritt umfaßt, um festzustellen, ob die Zahl der der Bildverarbeitung zu unterziehenden Bilder durch die Zahl der arithmetischen Prozessoren ohne Rest teilbar ist oder nicht, und wenn dies der Fall ist,
der Verfahrensschritt (B) den Verfahrensschritt des Generierens von Ausführungscodes für die arithmetischen Prozessoren umfaßt, um zu ermöglichen, daß die Bilder einer parallelen Verarbeitung durch entsprechende arithmetische Prozessoren unterzogen werden.

2. Verfahren nach Anspruch 1, bei dem die Bilder für die Verarbeitung in kleine Bereiche partitioniert werden und die Mehrzahl von arithmetischen Prozessoren so betrieben werden können, daß sie die Bildverarbeitung der kleinen Bereiche parallel ausführen und diese Bearbeitung wiederholen, bis das ganze Bild verarbeitet ist.

3. Verfahren nach Anspruch 2, bei dem der Verfahrensschritt (A) einen Verfahrensschritt umfaßt, um festzustellen (S 17), ob die Zahl der kleinen Bereiche, die der Bildverarbeitung unterzogen werden, durch die Zahl der arithmetischen Prozessoren ohne Rest teilbar ist, und falls nicht,
der Verfahrensschritt (B) den Verfahrensschritt des Generierens eines Ausführungscodes oder von Ausführungscodes umfaßt, um das Einschreiben eines Verarbeitungsergebnisses eines arithmetischen Prozessors oder von arithmetischen Prozessoren in einen Speicher zu sperren, bis der Rest verarbeitet wurde.

4. Verfahren nach Anspruch 1 oder 2 oder 3, mit dem weiteren Verfahrensschritt:
(C) Verarbeiten der Bilder (S2) auf der Basis der generierten Ausführungscodes.

5. Verfahren nach einem der vorhergehenden Ansprüche, bei dem der Verfahrensschritt (A) den Verfahrensschritt umfaßt, daß auf der Basis des Ergebnisses der an den kleinen Bereichen durchzuführenden Bildverarbeitung detektiert wird, ob eine integrale Operation vorliegt oder nicht, und einen Verfahrensschritt, in dem detektiert wird, ob an dem Ergebnis der integralen Operation eine Verteilungsoperation vorzunehmen ist oder nicht.

6. Gerät zum Kompilieren eines parallelen Verarbeitungsprogramms, wobei das parallele Verarbeitungsprogramm generierte Ausführungscodes aufweist, die so betrieben werden können, daß sie es einer Mehrzahl von arithmetischen Prozessoren ermöglichen, eine parallele Bildverarbeitung auszuführen, wobei das Gerät aufweist:
eine Einrichtung (**1**) zum Analysieren einer Steueranweisung eines Quellprogramms, um festzustellen, ob daran ein Identifizierer für die parallele Bildverarbeitung angehängt ist, und
eine Einrichtung (**1**, **4**) zum Generieren von Ausführungscodes, wenn der Identifizierer für die parallele Bildverarbeitung an die Steueranweisung angehängt ist, und wobei die Einrichtung zum Analysieren einer Steueranweisung so betrieben werden kann, daß sie feststellt, ob die Zahl der der Bildverarbeitung zu unterziehenden Bilder durch die Zahl der arithmetischen Prozessoren ohne Rest teilbar ist oder nicht, und wenn dies der Fall ist, die Einrichtung zum Generieren von Ausführungscodes so betrieben werden kann, daß sie Ausführungscodes für die arithmetischen Prozessoren generiert, die es ermöglichen, daß die Bilder einer parallelen Verarbeitung durch entsprechende arithmetische Prozessoren unterzogen werden.

7. Gerät nach Anspruch 6, bei dem die Bilder für die Verarbeitung in kleine Bereiche partitioniert werden und die Mehrzahl von arithmetischen Prozessoren so betrieben werden können, daß sie die Bildverarbeitung der kleinen Bereiche parallel ausführen und diese Bearbeitung wiederholen, bis das ganze Bild verarbeitet ist.

8. Gerät nach Anspruch 7, bei dem die Einrichtung zum Analysieren einer Steueranweisung so betrieben

werden kann, daß sie feststellt, ob der Zahl der der Bildverarbeitung zu unterziehenden kleinen Bereiche durch die Zahl der arithmetischen Prozessoren ohne Rest teilbar ist oder nicht, und falls nicht,
die Einrichtung zum Generieren von Ausführungscodes so betrieben werden kann, daß sie einen Ausführungscode oder Ausführungscodes generiert, um das Einschreibens eines Verarbeitungsergebnisses eines arithmetischen Prozessors oder von arithmetischen Prozessoren in einen Speicher zu sperren, bis der Rest verarbeitet wurde.

9. Gerät nach Anspruch 6 oder 7 oder 8, das ferner aufweist:
die genannten Mehrzahl von arithmetischen Prozessoren zur Verarbeitung der Bilder (3) auf der Basis der generierten Ausführungscodes,

10. Gerät nach einem der Ansprüche 6 bis 9, bei dem die Einrichtung zum Analysieren einer Steueranweisung so betrieben werden kann, daß sie auf der Basis des Ergebnisses der an den kleinen Bereichen durchzuführenden Bildverarbeitung feststellt, ob eine integrale Operation vorliegt oder nicht, und detektiert, ob an dem Ergebnis der integralen Operation eine Verteilungsoperation vorzunehmen ist oder nicht.

11. Gerät nach einem der vorhergehenden Ansprüche 6 bis 6, das ferner aufweist:
die genannten Mehrzahl von arithmetischen Prozessoren und
eine Einrichtung (**2**) für die Steuerung der Datenzuführung für die Ausführung einer integralen Operation auf der Basis der betreffenden Ergebnisse aus den arithmetischen Prozessoren.

12. Gerät nach Anspruch 11, bei dem die Mehrzahl von arithmetischen Prozessoren auf eine Steuereinrichtung ansprechen, die so betrieben werden kann, daß sie in Abhängigkeit von der Zahl der zu verarbeitenden Bilder eine selektive Umschaltung vornimmt, um eine der folgenden Tätigkeiten auszuführen:
eine erste parallele Bildverarbeitung in den mehreren arithmetischen Prozessoren in Bezug auf entsprechende geteilte kleine Bereiche eines Bilds eines zugeführten Bildsignals in Abhängigkeit von der Zahl der Bilder des Bildsignals, wobei die mehreren arithmetischen Prozessoren so betrieben werden können, daß sie auf der Basis der betreffenden Bildverarbeitungsergebnisse aus diesen arithmetischen Prozessoren eine integrale Operation ausführen, und
eine zweite parallele Bildverarbeitung in jedem arithmetischen Prozessor in Bezug auf eine Mehrzahl von Bildern und die Zahl der arithmetischen Prozessoren, wobei die Bildverarbeitung in Einheiten eines Bilds des zugeführten Bildsignals erfolgt.

13. Gerät nach Anspruch 12, bei dem die Steuereinrichtung eine solche Steuerung vornimmt, daß,
falls die Zahl der zu verarbeitenden Bilder gleich 1 ist oder falls der durch Teilen der Zahl von Bildern durch die Zahl von arithmetischen Prozessoren gewonnene Wert einen Rest aufweist die Steuereinrichtung veranlaßt, daß die erste Bildverarbeitung stattfindet, und
für den Fall, daß die Zahl der zu verarbeitenden Bilder durch die Zahl der arithmetischen Prozessoren geteilt werden kann, die Steuereinrichtung veranlaßt, daß die zweite Bildverarbeitung stattfindet.

Es folgen 7 Blatt Zeichnungen

Anhängende Zeichnungen

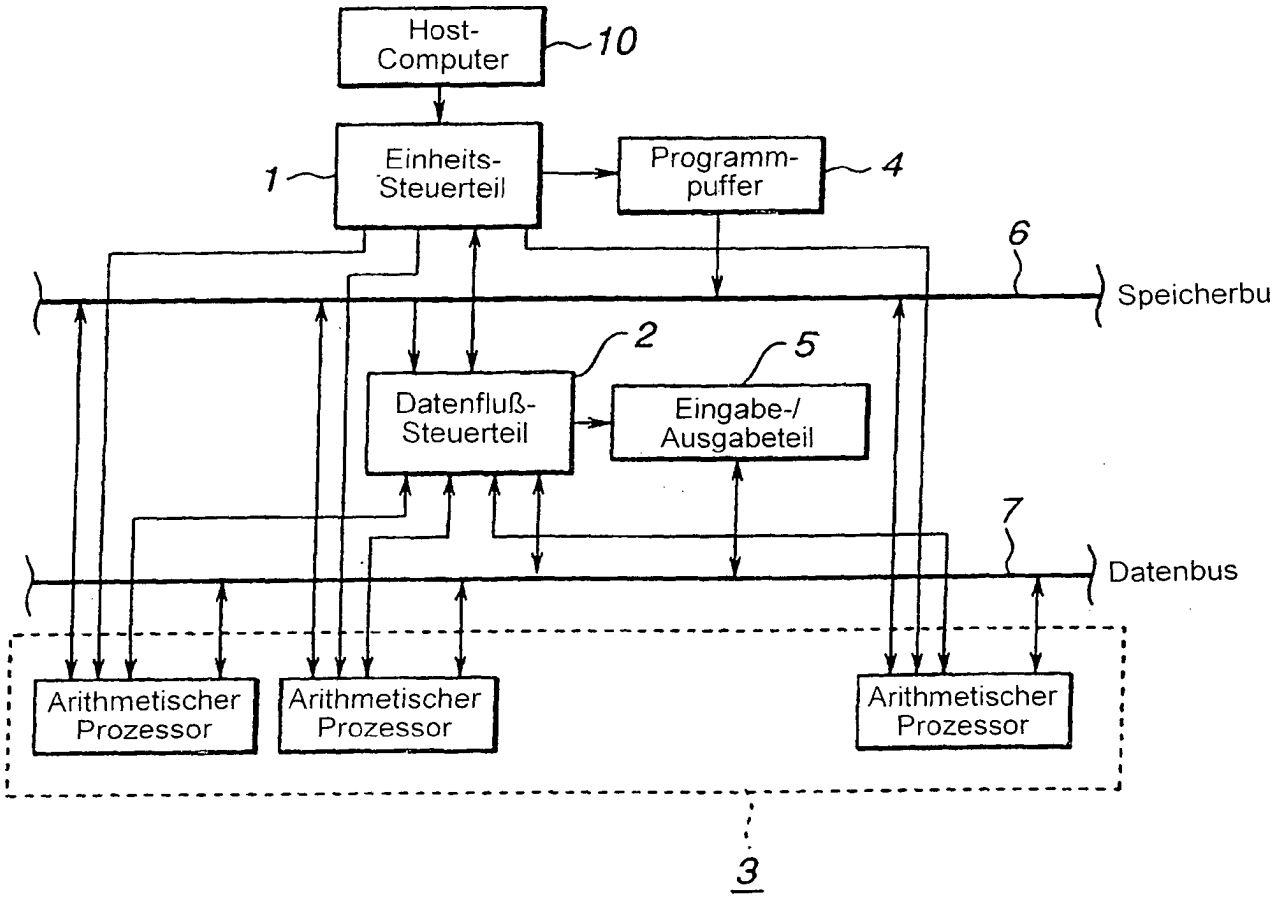


FIG.1

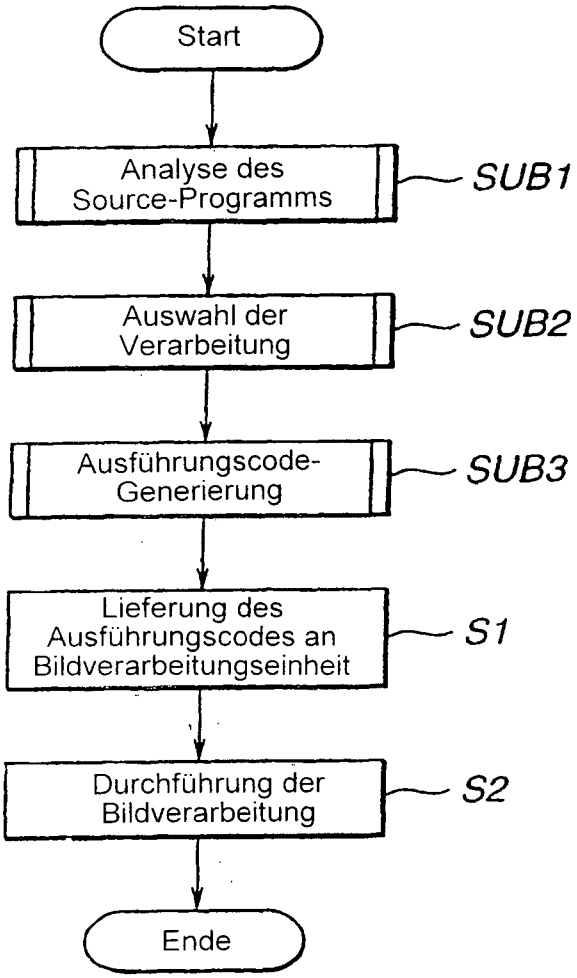

Start
Analyse des Source-Programms
SUB1
Auswahl der Verarbeitung
SUB2
Ausführungscode-Generierung
SUB3
Lieferung des Ausführungscodes an Bildverarbeitungseinheit
S1
Durchführung der Bildverarbeitung
S2
Ende

FIG.2

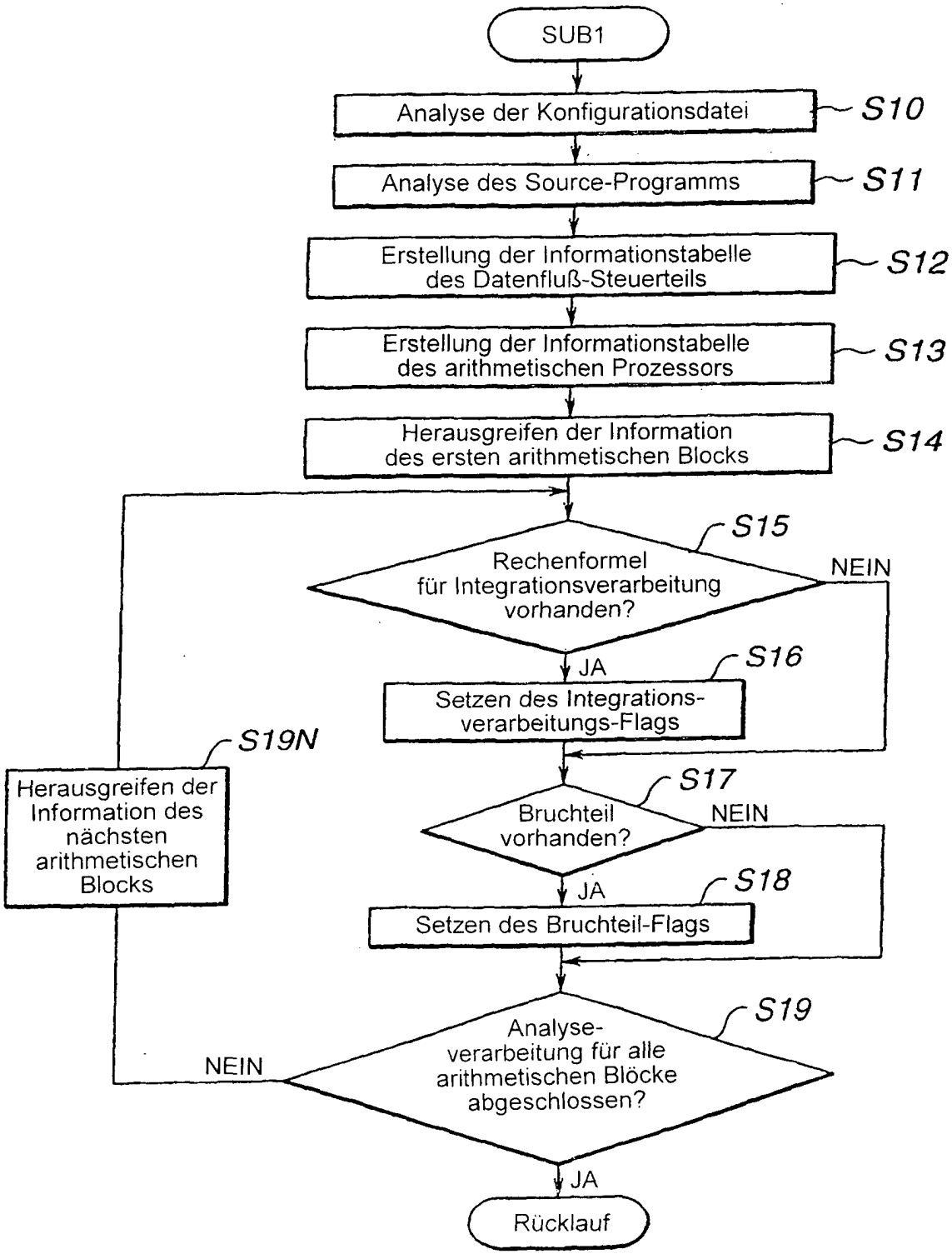
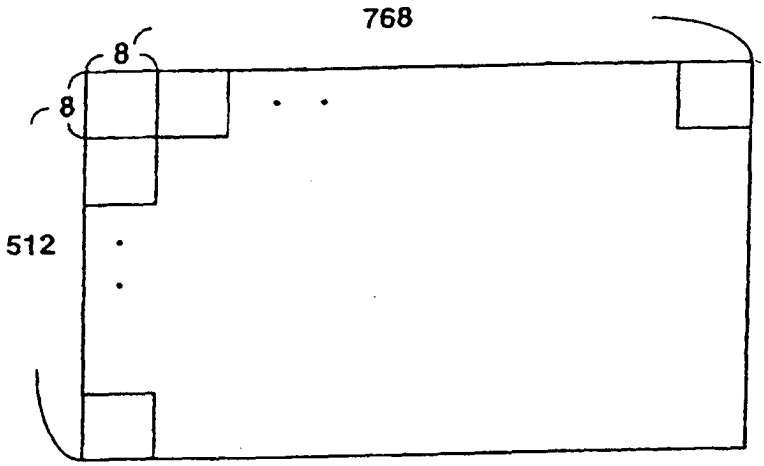

SUB1
Analyse der Konfigurationsdatei
S10
Analyse des Source-Programms
S11
Erstellung der Informationstabelle des Datenfluß-Steuerteils
S12
Erstellung der Informationstabelle des arithmetischen Prozessors
S13
Herausgreifen der Information des ersten arithmetischen Blocks
S14
S15
Rechenformel für Integrationsverarbeitung vorhanden?
NEIN
JA
S16
Setzen des Integrations-verarbeitungs-Flags
S17
Bruchteil vorhanden?
NEIN
JA
S18
Setzen des Bruchteil-Flags
S19
Analyse-verarbeitung für alle arithmetischen Blöcke abgeschlossen?
NEIN
JA
Rücklauf
S19N
Herausgreifen der Information des nächsten arithmetischen Blocks

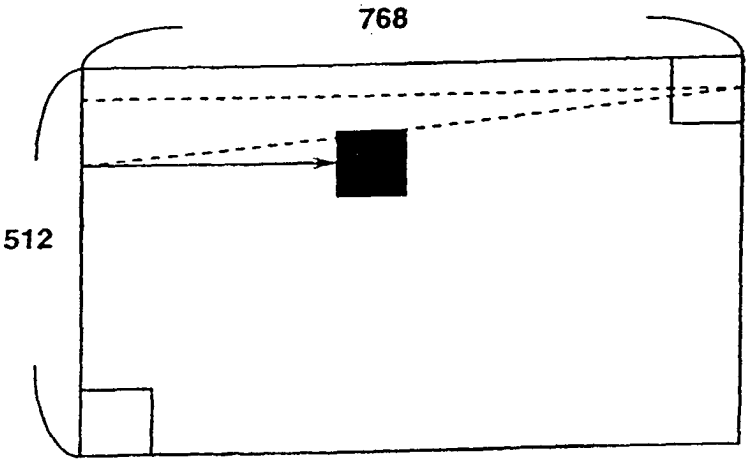
FIG.3

FIG.4(a)



Beziehung zwischen der Pixelzahl und dem Pixelblock eines Bilds

FIG.4(b)



Ansicht zur Erläuterung der Datenzuführung des Pixelblocks

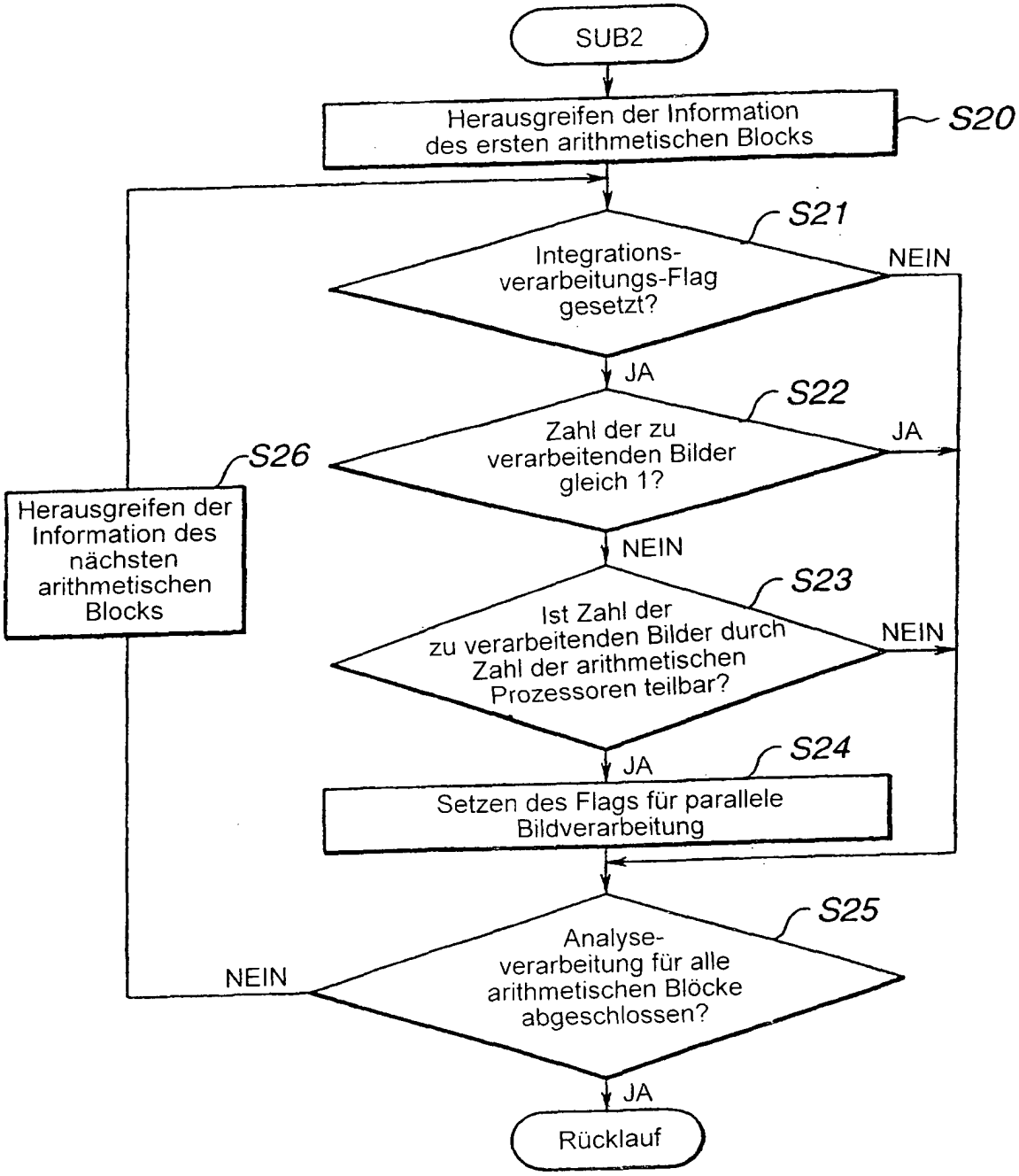

SUB2
Herausgreifen der Information des ersten arithmetischen Blocks
S20
S21
Integrations-verarbeitungs-Flag gesetzt?
NEIN
JA
S22
Zahl der zu verarbeitenden Bilder gleich 1?
JA
S26
Herausgreifen der Information des nächsten arithmetischen Blocks
NEIN
S23
Ist Zahl der zu verarbeitenden Bilder durch Zahl der arithmetischen Prozessoren teilbar?
NEIN
JA
S24
Setzen des Flags für parallele Bildverarbeitung
S25
Analyse-verarbeitung für alle arithmetischen Blöcke abgeschlossen?
NEIN
JA
Rücklauf

FIG.5

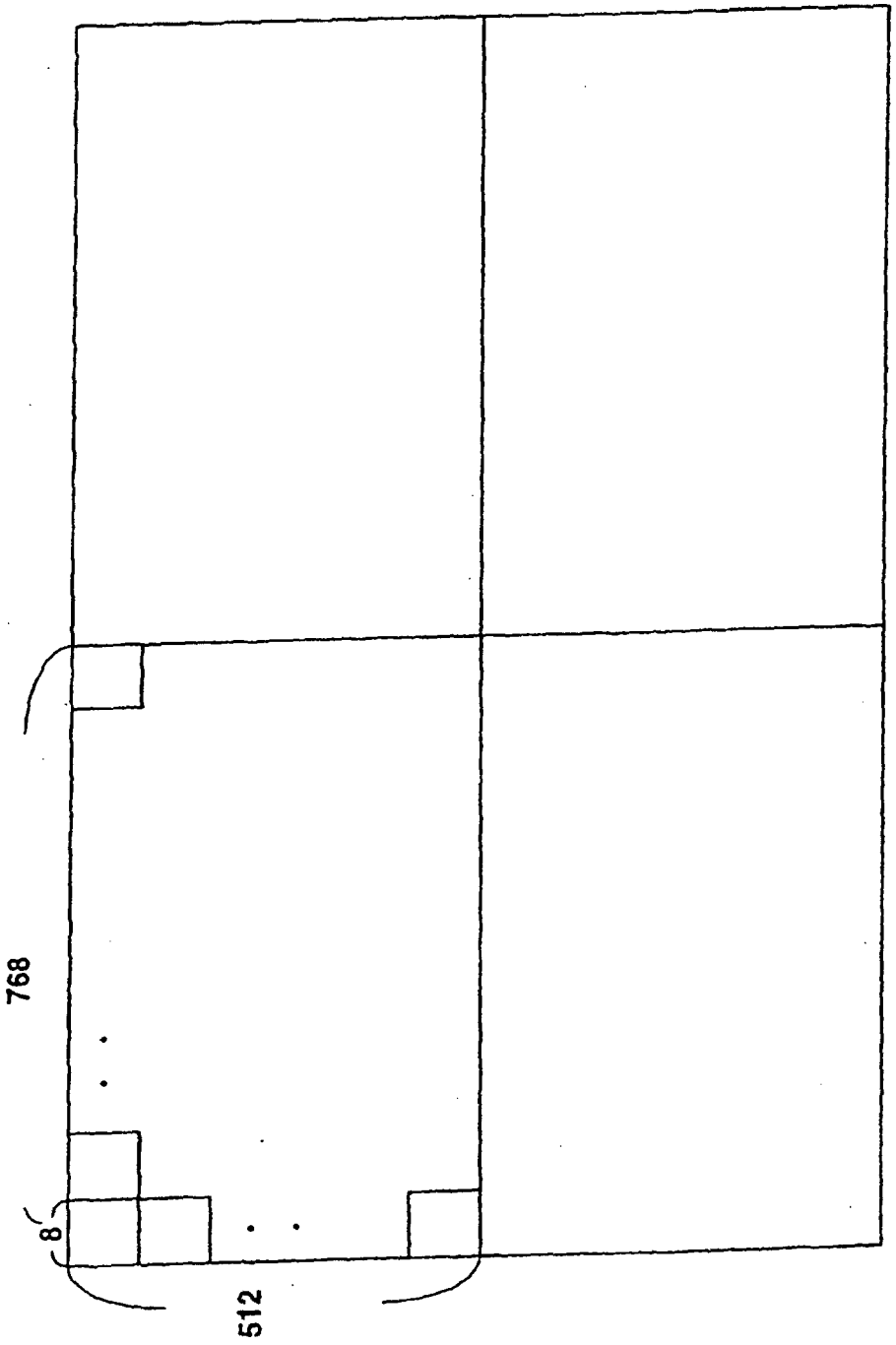


FIG.6

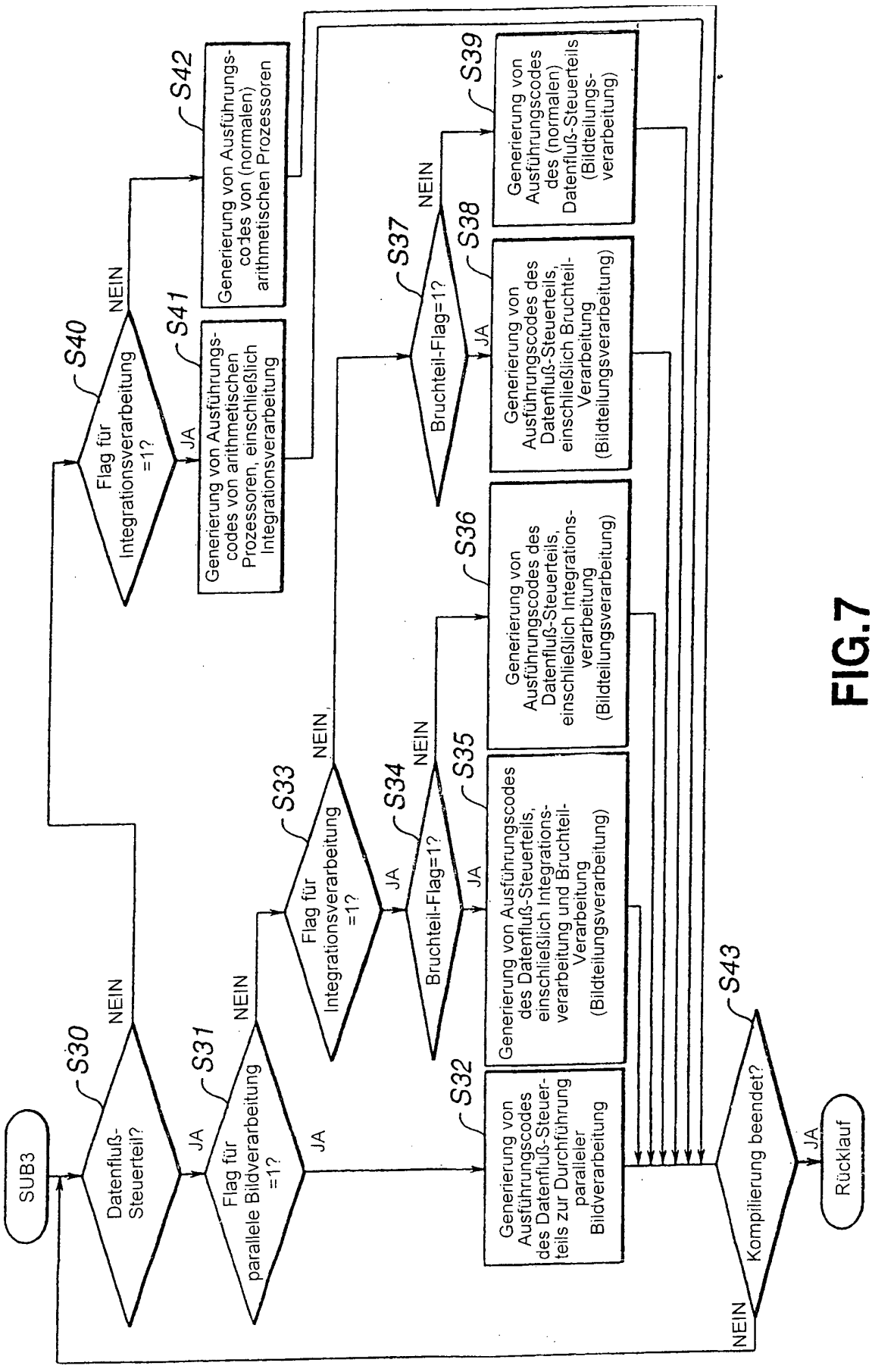

SUB3
S30
Datenfluß-Steuerteil?
NEIN
JA
S31
Flag für parallele Bildverarbeitung =1?
NEIN
JA
S32
Generierung von Ausführungscodes des Datenfluß-Steuerteils zur Durchführung paralleler Bildverarbeitung
S33
Flag für Integrationsverarbeitung =1?
NEIN
JA
S34
Bruchteil-Flag=1?
NEIN
JA
S35
Generierung von Ausführungscodes des Datenfluß-Steuerteils, einschließlich Integrationsverarbeitung und Bruchteil-Verarbeitung (Bildteilungsverarbeitung)
S36
Generierung von Ausführungscodes des Datenfluß-Steuerteils, einschließlich Integrationsverarbeitung (Bildteilungsverarbeitung)
S37
Bruchteil-Flag=1?
NEIN
JA
S38
Generierung von Ausführungscodes des Datenfluß-Steuerteils, einschließlich Bruchteil-Verarbeitung (Bildteilungsverarbeitung)
S39
Generierung von Ausführungscodes des (normalen) Datenfluß-Steuerteils (Bildteilungsverarbeitung)
S40
Flag für Integrationsverarbeitung =1?
NEIN
JA
S41
Generierung von Ausführungscodes von arithmetischen Prozessoren, einschließlich Integrationsverarbeitung
S42
Generierung von Ausführungscodes von (normalen) arithmetischen Prozessoren
S43
Kompilierung beendet?
NEIN
JA
Rücklauf

FIG.7