



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) DE 696 30 388 T2 2004.08.19

(12)

Übersetzung der europäischen Patentschrift

(97) EP 0 839 354 B1

(51) Int Cl.⁷: G06F 12/04
G06F 12/06

(21) Deutsches Aktenzeichen: 696 30 388.4

(86) PCT-Aktenzeichen: PCT/SE96/00748

(96) Europäisches Aktenzeichen: 96 917 795.5

(87) PCT-Veröffentlichungs-Nr.: WO 96/042055

(86) PCT-Anmeldetag: 05.06.1996

(87) Veröffentlichungstag
der PCT-Anmeldung: 27.12.1996

(97) Erstveröffentlichung durch das EPA: 06.05.1998

(97) Veröffentlichungstag
der Patenterteilung beim EPA: 15.10.2003

(47) Veröffentlichungstag im Patentblatt: 19.08.2004

(30) Unionspriorität:

9502113 09.06.1995 SE

(84) Benannte Vertragsstaaten:

AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LI,
LU, MC, NL, PT, SE

(73) Patentinhaber:

Saab AB, Linköping, SE

(72) Erfinder:

SÖDERQUIST, Ingemar, S-583 37 Linköping, SE

(74) Vertreter:

Prietsch, R., Dipl.-Ing., Pat.-Anw., 80687 München

(54) Bezeichnung: SPEICHERSTRUKTUR

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 eingelebt, 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

Datenspeicherstruktur

[0001] Die vorliegende Erfindung bezieht sich auf eine Datenspeicherstruktur der im Oberbegriff des Anspruchs 1 angegebenen Art.

Hintergrund der Erfindung

[0002] Die Entwicklung von Datenspeichern ist gleichzeitig mit der Entwicklung von Prozessoren für Rechner eingetreten. Damit sei gesagt, dass die Entwicklung der Prozessoren die Entwicklung der Speicher gelenkt hat. Aus diesem Grund war es seither Tradition, dass die Information in einem Speicher willkürlich in Bitgruppen (im Weiteren Bytes genannt) aufgeteilt wird, mit einer festen Anzahl Bits, die meistens unabhängig von früher selektierter Information adressiert werden. Daraus ergibt sich, dass der Adressendecoder fähig sein muss, innerhalb einer bestimmten Zugriffszeit eine willkürliche Stelle im Speicher zu erreichen, um dort Information auszulesen oder zu schreiben. Die Datenübertragungsgeschwindigkeit ist umgekehrt proportional der Zugriffszeit. Die Zugriffszeit ist natürlich davon abhängig, nach welcher Fertigungstechnologie der Speicher hergestellt worden ist. Häufig kommt die VLSI-Technologie zur Anwendung, und man bemüht sich, die Größe und Komplexität der Schaltkreise zu minimieren. Bei einer vorgegebenen Technologie wird die Größe des Speichers immer durch die Ausbreitungsgeschwindigkeit der Signale begrenzt. Gleichzeitig besteht auch die Beziehung, dass bei traditioneller Architektur ein größerer Speicherbereich zwangsläufig zu längerer Zugriffszeit führt, da die Komplexität des Adressendecoders mit zunehmender Speichergröße zunimmt. Der Bedarf an Datenverarbeitungsgeschwindigkeit wird ständig größer. Da die Komplexität des Adressendecoders zunimmt und die Ausbreitungsgeschwindigkeit konstant ist, wird die Gesamtspeicherfläche, das heißt die adressierbare Fläche, auf der Information gespeichert werden kann, mit zunehmender Rechnergeschwindigkeit immer kleiner.

[0003] Auf vielen Anwendungsgebieten, die in den letzten Jahren schlagartig an Bedeutung gewonnen haben, wie Telekommunikation, Radar, Massenspeicher für Rechner usw., wird die Information nicht in der Form von Bytes verarbeitet, sondern als Bytefolgen, die Vektoren genannt werden. In modernen digitalen Telefon-systemen und Datenübertragungssystemen werden Informationspakete, ATM, benutzt. Bei Radarsystemen hat die Information die Form von Impulsen. Nach der Digitalisierung kann der Impuls als ein großer Vektor betrachtet werden, der durch seinen Ausgangspunkt und seine Länge definiert wird.

[0004] Bei vielen Anwendungen werden große Blöcke miteinander verbundener Bytes auf die gleichen Weise verarbeitet, zum Beispiel ein Textblock in einer Textverarbeitungsanlage. Ein anderes Beispiel ist der Swap-speicher (virtueller Speicher) in einem Rechner. Das gemeinsame Merkmal ist, dass große Mengen von Information (Vektor) in ihrer Gesamtheit verarbeitet werden. Ein Adressieren auf Byteebene wird nicht angewendet.

[0005] In US, A, 4 747 070 wird eine Speicherstruktur gezeigt, die für eine Speicherung von Vektoren gestaltet ist. Die gezeigte Speicherstruktur ist für M parallele Speicherbänke vorgesehen. Jede Speicherbank ist N Wörter tief. M Wörter können jeweils gleichzeitig aus dem Speicher gelesen werden. Die parallele Speicherstruktur gestattet eine gesteigerte Übertragungsrate zwischen dem Systemspeicher und dem Prozessor. Jede Speicherbank wird separat adressiert.

[0006] In US, A, 4 858 107 wird eine Speicherstruktur gezeigt, welche die Einführung eines Prioritätsschemas zwischen einer Zentraleinheit und einem Videodisplayprozessor gestattet. Der Speicher ist in Bänke oder Bitebenen organisiert, jede mit einem temporären Flipflop, das in das jeweilige Schieberegister lädt. Jede Speicherbank oder Bitebene ist mit einer separaten Adresse versehen.

Zweck der Erfindung

[0007] Ein Zweck der Erfindung besteht darin, eine Speicherstruktur zu erhalten, die einer Speicherung von Vektoren angepasst ist.

[0008] Ein anderer Zweck der Erfindung besteht darin, eine Speicherstruktur zu erhalten, die einen schnellen und kontinuierlichen Zugriff zu Speicherplätzen zum Lesen und/oder Schreiben von Vektoren gestattet.

[0009] Ein weiterer Zweck der Erfindung besteht darin, eine Struktur zu erhalten, bei der das Schreiben eines Vektors und, unabhängig davon, zum gleichen Zeitpunkt das Lesen eines Vektors stattfinden kann.

[0010] Noch ein weiterer Zweck der Erfindung besteht darin, eine Speicherstruktur zu erhalten, bei der das Schreiben eines Vektors und praktisch gleichzeitig Lesen desselben stattfinden kann.

[0011] Noch ein anderer Zweck der Erfindung besteht darin, eine Speicherstruktur zu erhalten, bei der die Zeit zur Ausgabe geschriebener Daten bis herunter auf das Taktintervall gesteuert werden kann.

[0012] Noch ein anderer Zweck der Erfindung besteht darin, eine Speicherstruktur zu erhalten, die als Ergänzung zum gegenwärtigen Typ von Speicherstruktur für einen Rechner benutzt werden kann.

[0013] Ein weiterer Zweck der Erfindung besteht darin, eine Speicherstruktur zu erhalten, die eine Verarbei-

tung eines gespeicherten Vektors parallel von einer Vielzahl von Speicherstellen für Vektoren zu gestatten, um entweder den verarbeiteten Vektor neu zu positionieren oder ein Verarbeitungsergebnis zu erhalten.

Kurze Zusammenfassung der Erfindung

[0014] Die oben genannten Zwecke werden durch eine Speicherstruktur erfüllt, die die im kennzeichnenden Teil des Anspruchs 1 genannten Merkmale aufweist. Weitere Merkmale und weitere Entwicklungen der Erfindung gehen aus den anderen Ansprüchen hervor.

[0015] Gemäß der Erfindung hat nach der Konfiguration jede der Speicherstellen im Speicher eine Länge, die der Länge der großen Vektoren angepasst ist, und die parallel angeordnet sind, ausgehend von einer Informationseingabe und tiefer in den Speicher, so dass jeder Vektor ungeteilt aufeinander folgend mit Beginn des Vektors bei der Eingabe in den Speicher gespeichert wird, wobei die Adressierung so ausgeführt ist, dass sie bei der Eingabe in den Speicher stattfindet. Mittel, die als Schieberegister für die Eingabe von Information in ununterbrochener Folge zu den Speicherstellen im Speicher funktionieren.

[0016] Der Speicher ist vorzugsweise quer in Speicherblöcke unterteilt, so dass jede der Speicherstellen einen Teil in einem Speicherblock hat. Die Speicherstellen im Speicher können durch Verbinden eines oder mehrerer Speicherblöcke verlängert werden. Jeder Speicherblock kann folgendes umfassen:
als Mittel, die als Schieberegister funktionieren:

- einen Eingabepuffer mit serieller Eingabe von Daten analog zu einem Schieberegister;

als Speicherstellen:

- eine Vielzahl Speicherfelder, in welche Information vom Eingabepuffer sequentiell eingegeben werden kann.

[0017] In diesem Zusammenhang ist in jedem Speicherblock eine Steuerlogik angeordnet, welche Steuerlogik die Eingabe von Information in Speicherfelder über den Eingabepuffer steuert, wobei die Speicherblöcke durch die in diesen vorhandenen, kaskadenförmig aneinander gekoppelten Eingabepuffer kaskadenförmig aneinander gekoppelt sind, so dass Information seriell stufenweise durch eine Anzahl Puffer in den einzelnen aufeinander folgenden Speicherblöcken eingebbar ist, und die Steuerlogik in jedem Speicherblock ist so angeordnet, dass sie die Steuerlogik des nächsten Speicherblocks informiert, wenn die Speicherfelder im eigenen Speicherblock mit Information gefüllt sind oder alle Information aus diesen ausgelesen ist, so dass die Steuerlogik im nächsten Speicherblock die Aufgabe, Information in diesen Speicherblock zu schreiben, übernehmen kann. Für die Ausgabe von Information in ununterbrochener Folge aus den Speicherstellen im Speicher können wie Schieberegister funktionierende Mittel vorhanden sein. Außerdem können Mittel für eine Verarbeitung von parallel in den verschiedenen Speicherblöcken gespeicherten Daten vorhanden sein.

Vorteile der Erfindung

[0018] Die Speicherstruktur gemäß der Erfindung gestattet die Ausgestaltung eines großen und erweiterbaren Speicherraumes für Information, kontinuierliche Datenübertragung und gleichzeitig extrem hohe Übertragungsgeschwindigkeiten. Ein Erweitern des Speicherraums für einen Vektor und dabei der Größe des Speichers kann ohne Einfluss auf die Datenübertragungsgeschwindigkeit durchgeführt werden, da die Schnittstelle zum Speicher von außen gesehen unverändert ist.

[0019] Bei der Speicherstruktur gemäß der Erfindung kann eine kontinuierliche Datenübertragung mit ungefähr 0,5–1 GBytes/s stattfinden, was mit dem gegenwärtigen Wert von 30–100 MBytes/s zu vergleichen ist. Daraus ergibt sich eine um das zehnfache schnellere Übertragung bei Anwendung der entsprechenden grundlegenden Technologie. Die Speicherstrukturen selbst sind nicht von der angewandten Technologie abhängig, sondern können bei beliebigen Speichertechnologien (RAM, Magnetspeicher, optische Speicher usw.) zur Anwendung kommen.

Kurze Beschreibung der Zeichnungen

[0020] Nachstehend folgt eine Beschreibung der Erfindung unter Bezugnahme auf die beigefügten Zeichnungen, worin:

- [0021] **Fig. 1** in übersichtlicher Darstellung das Prinzip einer Speicherstruktur gemäß der Erfindung zeigt;
- [0022] **Fig. 2** eine erste Ausführungsform einer Speicherstruktur gemäß der Erfindung zeigt;
- [0023] **Fig. 3** die innere Struktur eines Speicherblocks in ausführlicherer Darstellung als in **Fig. 2** zeigt;
- [0024] **Fig. 4** ein Ablaufdiagramm für die gesamte Steuerlogik für den Speicher zeigt;
- [0025] **Fig. 5** ein Ablaufdiagramm für die Steuerlogik in einem Speicherblock zeigt;

- [0026] **Fig.** 6 eine zweite Ausführungsform einer Speicherstruktur gemäß der Erfindung zeigt;
 [0027] **Fig.** 7 eine dritte Ausführungsform einer Speicherstruktur gemäß der Erfindung zeigt.

Beschreibung der Ausführungsformen

Definitionen

Gespeicherte Information

Bezeichnet eine Anzahl Vektoren, die jeder, von außen gesehen, kontinuierlich und in ungeteilter Form bearbeitet werden.

Vektor

Bezeichnet die gesamte kontinuierliche Menge Information, welche in einem Stück verarbeitet werden soll. Ein Vektor wird von außen mit Hilfe eines Ausgangspunkts, oder einer Identität, und der Länge definiert.

Bytegruppe

Der Größe eines Speicherfeldes angepasster Teil eines Vektors. Ein Vektor lässt sich als eine Summe von Bytegruppen beschreiben.

Speicherstruktur

Bezeichnet die Gesamtheit des Systems gem. Fig. 2. Die Speicherstruktur besteht aus einem oder mehreren Speicherblöcken, die Umsetzereinheiten 5, 6 und die Steuerlogik 4.

Speicherblock

Bezeichnet eine Einheit mit deutlich definierter Funktion, die nur mit den nächsten Nachbarn kommuniziert, siehe Fig. 2. Die Speicherstruktur wird immer um eine ganze Zahl von Speicherblöcken erweitert.

Speicherstelle

Bezeichnet den Teil der gesamten Speicherfläche einer Speicherstruktur, der nach der Konfiguration der Länge des gegenwärtigen Vektors angepasst ist. Die Speicherstellen werden beginnend bei Speicherblock #1 und nach innen verteilt, siehe Fig. 1.

Speicherebene

Der Teil eines Speicherblocks, dem nach der Konfiguration einem bestimmten Vektor (die Identität) zugeordnet ist.

Speicherfeld

Der Teil einer Speicherebene, der in einem Stück verarbeitet wird. Die Größe entspricht der der Eingabe/Ausgabepuffer.

Beschreibung der Zeichnungen

[0028] **Fig.** 1 zeigt eine Anzahl Speicherblöcke #1, #2 ... #M. die in Kaskadenschaltung aneinander gekoppelt sind. Jeder Speicherblock hat einen Informationsteil **1I**, **2I** ... **MI**, und Eingabepuffer **1IB** bzw. **2IB** ... bzw. **MIB**, einen Ausgabepuffer **1UB** bzw. **2UB** ... bzw. **MUB** und eine Steuerlogik **1S** bzw. **2S** ... bzw. **MS**. Jeder Speicherblock kann auf einem separaten Chip angeordnet sein, und eine selektierte Anzahl Chips kann miteinander gekoppelt werden, um einen gewählt großen Speicher zu bilden. Es ist auch möglich und angebracht, auf einem Chip mehrere Speicherblöcke einzurichten, wobei mehrere Multiblock-Chips in Kaskadenschaltung aneinander gekoppelt sein können. Jede der Speicherstellen im Speicher hat eine Länge, die der Länge großer Vektoren angepasst ist. Die Blöcke sind parallel zueinander angeordnet und erstrecken sich beginnend bei einer Ein-/Ausgabe für Information und tiefer in den Speicher, so dass jeder Vektor so angeordnet ist, dass er ungeteilt in Folge über die Eingabepuffer gespeichert wird, und zwar beginnend mit dem Vektor am Ein-/Ausgang des Speichers, und ähnlich dass er in Folge über die Ausgabepuffer ausgegeben wird, wenn dieser Bedarf vorliegt (siehe die Ausführungsform in **Fig.** 7). Die Adressierung geschieht an den Ein-/Ausgang des Speichers.

[0029] Bei der in **Fig.** 2 gezeigten Ausführungsform umfasst die Speicherstruktur eine Anzahl in Kaskadenschaltung gekoppelter Einheiten oder Speicherblöcke #1-#M, Steuerlogik **4** und gegebenenfalls Umsetzereinheiten **5**, **6**. Die hier gezeigten Speicherblöcke haben jeder einen Teil derselben darstellenden Block mit mehreren Speicherebenen #1I-#1N bzw. „M1-#MN, die hintereinander liegen, wobei jeder eine Anzahl Speicherfelder F1-FP umfasst. Es sei darauf hingewiesen, dass im Weiteren Teil der Beschreibung ein spezielles Speicherfeld im Speicher ein Bezugssymbol hat, das seine Zuordnung zum jeweiligen Speicherblock und zur jeweiligen Speicherebene angibt, z. B. #m, Qi, Fj bezeichnet ein Speicherfeld Fj in der Speicherebene Qi im

Speicherblock #m. Dieses Bezugssystem wird durchgehend angewandt.

[0030] Es ist jedoch offenbar, dass die Speicherblöcke in der Praxis eine Oberflächenstruktur aufweisen können, bei der die Ebenen auf verschiedene Stellen auf der Oberfläche verteilt oder als Speicherfelder in Folge angeordnet sind, so dass zum Beispiel im Falle eines Chips mit mehreren Blöcken die Speicherfelder in Speicherblöcken #2/Speicherebene #Qi in direkter Fortsetzung der Speicherfelder in Speicherblock #1/Speicherebene #Q1 usw. angeordnet sind.

[0031] Wie aus **Fig. 3** hervorgeht, hat die Speicherstruktur mehrere adressierbare Speicherstellen oder Speicherebenen Q1 ... Qi ... QN (i = 1 ... N). Jeder Speicherblock #m hat eine Anzahl Speicherfelder F1 bis FP, die jeder adressierbaren Speicherebene Q1 zugeordnet sind.

[0032] Jeder Speicherblock #m ist physisch in W_m Zeilen (entsprechend $N \times P$ in **Fig. 3**) und $a \times b$ Spalten organisiert. Jede Zeile bildet ein Speicherfeld und enthält eine Anzahl Bytegruppen oder Wörter (a), jeweils mit einer geeigneten Anzahl Bits (b). Der Speicherblock #m ist deshalb $W_m \times a \times b$ Bit groß. Die Speicherstruktur hat eine Gesamtgröße von $(W_1 + W_2 + \dots + W_M) \times a \times b$ Bit. Es ist auch der Sonderfall denkbar, dass a, b und W von Speicherblock zu Speicherblock variieren, aber dieser Fall wird hier nicht behandelt.

[0033] Ein Vektor V_i mit der Länge L_i Wörter ist als L_i/a Teilvektoren in den Speicherfeldern M1, F1, ..., M1, FP; M2 F1, ..., M2 FP; ..., MN F1, ..., NM FP gespeichert. Nach der Konfiguration gilt, dass $L_i = a \times (O_{i1} + O_{i2} + \dots + O_{im})$ Wörter, worin O_{im} den Inhalt von $M_i F_m$ ($i = 1 - N, m = 1 - P$) bezeichnet. Die Größe des Vektors ist demzufolge $a \times b \times (O_{i1} + O_{i2} + \dots + O_{im})$ Bit. Für jeden Speicherblock gilt auch, dass die Anzahl Zeilen immer der Summe der Anzahl Speicherfelder entspricht. $W_m = O_{1m} + O_{2m} + \dots + O_{Nm}$.

[0034] Vektoren verschiedener Längen können gleichzeitig in der Speicherstruktur gespeichert werden.

[0035] Die Speicherstruktur kann auf verschiedene Arten konfiguriert werden. Von außen gesehen werden die Vektoren nur vom Ausgangspunkt (der Identität) #1 Mi F1 aus adressiert, wo i jede beliebige Zahl zwischen 1 und N darstellen kann und F1 das erste Speicherfeld im Speicherblock #1 in der Speicherebene Mi bezeichnet. Die Länge der Vektoren Li wird durch die Konfiguration vorgegeben. Die Konfiguration kann statisch oder dynamisch sein.

[0036] Bei statischer Konfiguration sind alle O_{im} vorgegeben. Die Steuerlogik 4 besitzt Information über die Anzahl N Speicherstellen M_i ebenso wie über die Länge der jeweiligen Speicherstellen L_i . Die Steuerlogik 1S-MS in den Speicherblöcken besitzt Information über den Ausgangspunkt #1 Qi F1 und die Länge O_{im} des jeweiligen Speicherblocks. Der Endpunkt #m Qi Fj wird indirekt durch den Ausgangspunkt und die Länge vorgegeben.

[0037] Bei dynamischer Konfiguration verändert sich die Länge der gegenwärtigen Speicherebene, oder die Speicherstelle M_i ändert sich während des Schreibens. Es wird vorausgesetzt, dass dem Informationsvektor Information über sowohl die Länge als auch andere globale Information folgt. Diese Information ist alternativ in der ersten Bytegruppe als ein Kopf des Vektors V_i zu finden. Die Steuerlogik kann dann diese Information lesen, anwenden und entsprechend ihrer weiteren Ausbreitung in die Speicherstruktur verändern. Nach dem Kopf folgt die Information sequentiell.

[0038] Bei statischer Konfiguration wird vorausgesetzt, dass der Informationsvektor, der aus einer oder in eine Speicherstelle Qi gelesen bzw. geschrieben wird, hat eine Länge, die der Konfiguration entspricht. Bei der statischen Konfiguration kann der Kopf ausgelassen werden.

[0039] Jeder Speicherblock #1 hat einen Eingabepuffer iLB mit einem Pufferspeicher, der eine Anzahl Speicherbit für jedes Speicherfeld in der Speicherebene für ankommende Information und einen Ausgabepuffer iUB mit der gleichen Pufferspeichergröße für ausgehende Information umfasst. Darüber hinaus vorhanden ist eine Steuerlogik iS, die das Schreiben und Lesen von Information in der Speicherebene und dabei in welchem Speicherfeld oder welchen Speicherfeldern derselben steuert.

[0040] Die Rechnereinheit, die über die Schnittstelle 5, 6 (siehe **Fig. 2**) an die Speicherstruktur gemäß der Erfindung angeschlossen ist, ladet die ankommende Information durch den Informationseingang und gibt Information aus über den Informationsausgang in Form von Vektoren, beispielsweise Datenpaketen. Jeder Vektor kann mit einem Identität, Länge usw. enthaltenden Kopf ausgestattet werden. Alternativ kann die erforderliche Information, was für den Fachmann einsichtig ist, auf beliebige andere Art der Schnittstelle zugeführt werden. Eine Steuerlogik 4 ist an die Schnittstelle angeschlossen, um den Speicher in Übereinstimmung mit der bei diesem ausgelesenen Information zu steuern. Die Steuerlogik 4 kann darüber hinaus mit einem (nicht dargestellten) Internspeicher ausgestattet sein, in dem Daten zu den gespeicherten Vektoren, einschl. wo diese im Speicher gespeichert sind, und Information darüber, wo neue Vektoren gespeichert werden können. Somit findet die Adressierung an den Speicher unter dem Einfluss der Steuerlogik 4 statt. Dies ist in **Fig. 4A** erkennbar.

[0041] Die **Fig. 4A–4C** zeigten ein Ablaufdiagramm einer Ausführungsform für die Arbeitsweise der Steuerlogik, teils während einer Aktivierung (**Fig. 4A**), teils während des Schreibens in den Speicher (**Fig. 4B**) und teils während des Auslesens aus dem Speicher (**Fig. 4C**). Die Ablaufdiagramme sind so erläuternd, dass sich eine Beschreibung der einzelnen Stufen erübriggt.

[0042] Wie aus der nachfolgenden, ausführlicheren Beschreibung hervorgeht, ist die in den Speicher einge-

gebene Information zweckmäßig vom Paralleltyp, d. h. in Form von parallelen Bytes von geeigneter Länge, zum Beispiel 8, 16, 32 Bit. Wenn rein serielle Information eingegeben wird, führt die Umsetzereinheit 5 eine Umsetzung von der Bitform in die Parallelbyteform durch. Auf gleiche Weise sollte gegebenenfalls eine Umsetzung von Parallelbytes in Bit in der Umsetzereinheit **6** stattfinden. Es sei darauf hingewiesen, dass kein Hindernis besteht, dass Eingabeinformation am Informationseingang serielle Form und die Ausgabeinformation am Informationsausgang Parallelform oder umgekehrt hat, wobei dann nur eine der Umsetzereinheiten **5** oder **6** benötigt wird.

[0043] Während der dynamischen Konfiguration (siehe **Fig. 4B**) wird der Kopf jedes zu schreibenden Informationsvektors gelesen. Die darin enthaltene Information wird teilweise benutzt, um festzustellen, wie viel Speicherplatz der Vektor in Anspruch nehmen wird, und teilweise dazu, die Steuerlogik **1S** im ersten Speicherblock #1 zu aktivieren und ihr Information darüber, wie viele der nachfolgenden Steuerlogiken iS in den anderen Speicherblöcken #1 aktiviert werden müssen.

[0044] In der Praxis wird jeder gespeicherte Vektor seinen Anfang im Speicherfeld F1 in einer der Ebenen Qi im Speicherblock #1 haben und sich von dort aus tiefer in den Speicher erstrecken. Man kann erwägen, die Speicherung eines Vektors als eine einzelne kontinuierliche Folge auszuführen. Auf diese Weise werden die Vektoren parallel arrangiert, und deren Adressierung erfolgt am Anfang der einzelnen Ebenen im Speicherblock #1. Viele der Speicherfelder F1–FP werden wirklich in jedem Speicherblock für die Speicherung benutzt, so dass die Speicherung als gefaltet betrachtet werden kann, aber in Wirklichkeit liegt der gesamte Vektor in einer ununterbrochenen Folge, vom Anfang bis zum Ende von Speicherblock zu Speicherblock.

[0045] Jede Steuerlogik in den Speicherblöcken enthält im Inneren einen Speicher mit Information darüber, welche Speicherebenen und welche Felder darin leer oder bereits gefüllt sind. Wenn Vektoren, die gespeichert werden sollen, ankommen, findet zuerst eine Suche statt, entweder in der Steuerlogik **4** oder in der Steuerlogik **1S**, um eine verfügbare Speicherebene zu finden.

[0046] Wie aus **Fig. 5**, in der eine Ausführung des Ablaufdiagramms für jede Steuerlogik **1S–MS** in den Speicherblöcken, hervorgeht, beginnt das Schreiben, wenn die Information, der Vektor, durch die Schnittstelle geht (siehe **Fig. 2**). Die Konfiguration muss nicht vollständig für alle Speicherblöcke erfolgen, denn dies findet nach und nach entsprechend der Ausbreitungsgeschwindigkeit der Information in den Speicher hinein statt. Angenommen, die Ebene Qi, wobei i eine Zahl zwischen 1 und N darstellt, erweist sich als leer. Es ist zu beachten, dass ein Vektor immer in der gleichen Ebene, zum Beispiel Qx, wobei x eine Zahl zwischen 1 und N darstellt, in all den einzelnen Blöcken, d. h. in den Ebenen #1, Qx; #2 Qx; #2 Qx usw., gespeichert wird. Der ankommende Vektor beginnt, in den Eingabepuffer 1IB eingegeben zu werden. Sobald er mit einer Bytegruppe gefüllt ist, erfolgt eine Übertragung desselben in das erste Speicherfeld F1 in der Speicherebene #1 Qx. Der Eingabepuffer ist auf entsprechende Weise mit einem Schieberegister gefüllt, aber die Ausgabe von den Speicherfeldern kann entweder seriell oder parallel erfolgen, vorzugsweise parallel, damit der gesamte Speicherinhalt des Eingabepuffers entleert wird.

[0047] Auf gleiche Weise findet beim Schieberegister ein direktes Füllen statt, wonach der gesamte Inhalt des Speichers in das nächste Speicherfeld F2 in der Ebene #1 Qx usw. entleert wird, bis das letzte Speicherfeld FP in der Ebene #1 Qx gefüllt ist. Die Steuerlogik hat in diesem Zusammenhang einen (nicht abgebildeten) Eingabe-Feldpointer, der das gegenwärtig für Schreiben benutzte Speicherfeld ansteuert.

[0048] Daraufhin wird der Eingabepuffer **1IB** angesteuert, so dass er mit dem Füllen des Eingabepuffers **2IB** beginnt, auf gleiche Weise wie bei einem Schieberegister, und zwar seriell, d. h. Parallelbyte auf Parallelbyte, bis er gefüllt ist. Die gesamte serielle Länge des als Puffer benutzten Schieberegisters wird ab jetzt serielles Byte genannt, denn es repräsentiert die für ein Füllen des Puffers erforderliche Anzahl Taktsignale, d. h. es repräsentiert die serielle Länge einer Parallelbytegruppe.

[0049] Wenn der Eingabepuffer gefüllt ist, füllt er das erste Speicherfeld F1 in Ebene #2 Qx, und die verschiedenen folgenden Speicherfelder werden danach auf die gleiche Weise gefüllt wie das Speicherfeld in Speicherebene #1 Qx, d. h. gesteuert durch den Feldpointer von der Steuerlogik. Wenn die Speicherebene #1 Qx gefüllt ist, wird die Übertragung von Information in den Eingabepuffer 2IB des Speicherblocks #2 weiter übertragen an den Eingabepuffer des nächsten Speicherblocks und danach von diesem an die Speicherfelder in der nächsten Speicherebene #3Qx in diesem Speicherblock usw. Es sei darauf hingewiesen, dass die Möglichkeit besteht, mit vielen Eingabe- und Ausgabepuffern zu arbeiten, zum Beispiel zur gleichzeitigen Versorgung mehrerer Speicherebenen, obwohl dies in keiner der Figuren gezeigt ist. Auch die Steuerlogik kann mehrfach vorkommen.

[0050] Es sei darauf hingewiesen, dass Information in Form von sequentiellen Parallelbytes in die Eingabepuffer eingegeben werden kann, wobei jeder Puffer de facto so viele Parallelschieberegister enthält, wie Bit in jedem Byte enthalten sind. Der Ausdruck Parallelbytegruppe bezeichnet die Anzahl Bytes, die einen Eingabe-/Ausgabepuffer füllen. Wenn ein Eingabepuffer in ein Speicherfeld in einer Ebene entleert wird, muss eine sehr große Datenmenge übertragen werden. Deshalb umfasst jedes Speicherfeld eine extrem große Anzahl von Parallelbytegruppen.

[0051] Daten können unmittelbar nach dem Schreiben wieder aus einem Speicherfeld gelesen oder ausge-

geben werden.

[0052] Wie im Ablaufdiagramm in **Fig. 5** deutlich erkennbar, wird der Inhalt eines Speicherfeldes während des Lesens zu einem Zeitpunkt parallel mit der gesamten Parallelbytegruppe im ersten Feld in der in Rede stehenden Ebene **1i** an den Ausgabepuffer **1UB** übertragen und seriell ausgegeben, d. h. Parallelbyte nach Parallelbyte.

[0053] Wenn die Übertragung an den Ausgabepuffer des vorletzten Speicherfelds in Ebene #1 Qi stattgefunden hat, aktiviert die Steuerlogik **1S** die Steuerlogik **2s** zur Übertragung des Inhalts des ersten Speicherfelds #2 Qi an den Ausgabepuffer **2UB**.

[0054] Sofort nach der Übertragung des letzten Speicherfelde FP an den Ausgabepuffer **1UB** und wenn dieser seinen Inhalt an Parallelbytes seriell ausgegeben hat, gibt ein Ausgabepuffer **2UB** seinen Inhalt Parallelbyte nach Parallelbyte seriell an den Ausgabepuffer **1UB** aus, so dass dieser direkt auf den früheren Inhalt des Puffers **1UB** folgt.

[0055] Auf diese Weise gibt es keine Unterbrechung in der Informationsausgabe. Die Information in den Speicherfeldern in Ebene #2 Qi wird schrittweise ausgegeben an den Ausgabepuffer **2UB**, gesteuert von einem (nicht dargestellten) Ausgabefeldpointer durch die Steuerlogik **2S**, und von diesem seriell über Puffer **1UB** ausgegeben. Wenn das vorletzte Speicherfeld FP in Ebene **2i** an den Ausgabepuffer **2UB** übertragen wird, wird die Steuerlogik für den nächsten Speicherblock aktiviert, damit die Information im Speicherfeld an den Ausgabepuffer des Speicherblocks übertragen wird usw.

[0056] Bei der oben beschriebenen Ausführungsform gibt jeder Ausgabepuffer in den einzelnen Blöcke seinen Inhalt, eine Parallelbytegruppe, Parallelbyte nach Parallelbyte seriell an den nächsten Ausgabepuffer. Dies bedeutet, dass die Steuerung sowohl bei der Eingabe als auch der Ausgabe Parallelbyte nach Parallelbyte stattfindet. Jede Parallelbytegruppe kann in diesem Fall eine relativ große Anzahl Parallelbytes enthalten, d. h. der Start der Ausgabe eines Vektors mit Daten kann nach dem relativ langen Zeitintervall von einer Anzahl Taktabständen stattfinden, welche durch die Länge der Parallelbytegruppen vorgegeben sind.

[0057] Es gibt jedoch Anwendungen, bei denen die Steuerung der Datenausgabe auf einem so niedrigen Wert liegt wie der Taktabstand des Systems, in welchem der Speicher gemäß der Erfindung einen Teil darstellt. Die Kopflogik des Speichers besitzt Information darüber, in welcher Parallelbytegruppe die Ausgabe stattfinden soll und auch über die Phasenverschiebung, die innerhalb der Zeitlänge der Parallelbytegruppe erforderlich ist, damit die Ausgabe genau zum erwünschten Zeitpunkt stattfindet.

[0058] Eine Ausführungsform der Erfindung mit einer Steuerlogik, mit der dies erzielt werden kann, ist in **Fig. 6** dargestellt. Es besteht die Möglichkeit einer Ausgabe des gleichen Speicherinhalts an verschiedene Teile des Systems zu voneinander unabhängigen, steuerbaren Zeitpunkten, und in diesem Fall ist es möglich, viele Ausgabepuffer an die Speicherebenen angeschlossen zu haben. Allerdings ist in den Figuren nur ein Ausgabepuffer OUTBUFF dargestellt. Jeder Ausgabepuffer umfasst einen Zwischenpuffer MBUFF, in dem während der Steuerung der Inhalt im Schieberegister sk übertragbar ist. Natürlich weisen die einzelnen Ausgabepuffer die gleiche Arbeitsweise auf, und demzufolge trifft auch für alle OUTBUFF-Ausgabepuffer die gleiche Beschreibung zu.

[0059] Wenn die örtliche Steueranordnung in einem Speicherblock eine Indikation dafür erhält, dass der Speicherinhalt in einer Speicherebene über einen Ausgabepuffer OUTPUT ausgegeben werden soll, wird das erste Speicherfeld in dieser Speicherebene mit Hilfe eines hierfür vorhandenen Steuersignals C-1 in den Zwischenpuffer MBUFF heruntergeladen. Beim entsprechenden Taktsignal für die aktuelle Ausgabe erfolgt eine Übertragung des Inhalts des Zwischenpuffers mit Hilfe eines Steuersignals C-2 in das Schieberegister sk für die getaktete Ausgabe direkt mit Steuerung am Taktgebereingang CLK des Schieberegisters. Die Übertragung des Inhalts des nächsten Speicherfeldes an den Zwischenpuffer OUTBUFF erfolgt entweder direkt auf die Übertragung aus dem Zwischenpuffer an das Schieberegister oder vorzugsweise zu Beginn der nächsten ganzen Parallelbytegruppenperiode. Allerdings steht nichts einer Übertragung in den Zwischenpuffer zu jedem beliebigen Zeitpunkt im Intervall vor der Übertragung in das Schieberegister sk entgegen, beispielsweise während der Taktperiode direkt vor der Übertragung aus dem Zwischenpuffer in das Schieberegister.

[0060] Wenn dann der gesamte Speicherblock leer ist, so wie vorstehend beschrieben, wird Eingabe in den Ausgabepuffer zwecks weiterer Übertragung an den Ausgang fortgesetzt. Diese Daten werden wegen der gleichen Steuerung für den nächsten folgenden Block direkt als eine zeitlich korrekt fortgesetzte serielle Bytesgruppe in kontinuierlicher Reihenfolge in das Schieberegister sk an dessen serielllem Eingang eingegeben, der mit dem Schieberegister für den folgenden Ausgabepuffer verbunden ist, um durch das Schieberegister an den Ausgang des Speichers ausgegeben zu werden.

[0061] Ein Beispiel für eine Steuereinrichtung für den ersten Ausgabepuffer OUTBUFF für den Speicherblock für die Ausgabe der jeweiligen Übertragungen von den jeweiligen Speicherfeldern an den Zwischenpuffer MBUFF und danach vom Speicherpuffer an das Schieberegister sk ist unten in **Fig. 6** dargestellt. Bei der Überprüfung, ob die Ausgabe von in den Speicherebenen auszugebenden Daten zu einem bestimmten Zeitpunkt an Logikschaltung **10** geht, diese zeigt zuerst mit einem Ausgabefeldpointer auf das erste Speicherfeld in den Speicherebenen nach der Passage einer kleinen Gitterschaltung **11**, welche den Durchgang von Signalen von

der Logikschaltung in Anwesenheit eines Steuersignals CKL/b gestattet, wobei b die von einem Zahler **12** gezählte Anzahl Parallelbyteschritte im Schieberegister SK bezeichnet. Der Ausgabefeldpointer bewegt sich zum nächsten Speicherfeld beim nächsten Steuersignal CLK/b und zum nächsten Speicherfeld beim nächsten Steuersignal usw. Dies ist durch das Oval **13** dargestellt. Darüber hinaus ist die Phasenverschiebung bezüglich der Phase des Ausgabefeldpointers im oberen Teil der Figur für die einzelnen Positionen P1–Pa dargestellt, wo a die Positionsänge eines Speicherfeldes in den einzelnen Feldern Fi und F(i + 1) angibt, wobei C-1 bei dieser Ausführungsform zu Beginn des Ausgabefeldpointers kommt und C-2 steuerbar zur gewünschten Position, z. B. P2, versetzt wird.

[0062] Der Ausgang vom Zähler **12** wird zusammen mit einem Ausgang von der Logikschaltung **11** in eine zweite Logikschaltung **14** eingegeben, die für den Feldwechsel vorhanden ist. Die Logikschaltung **14** gibt dann, gemäß der dargestellten Ausführungsform, das Signal C-1 an ihrem Ausgang aus, welches Signal im gezeigten Fall die Datenübertragung vom selektierten Speicherfeld über eine kleine Gitterschaltung **15** an den Zwischenpuffer MBUFF ausgibt, wobei die Gitterschaltung das Signal beim nächsten Taktsignal CLK ausgibt. Das Signal C-1 wird auch dem Eingang einer Verzögerungseinheit **16** zugeleitet, welche an einem Steuereingang ein Signal **15** von der Logikschaltung **10** hat, welches Signal vorzugsweise in digitaler Form ansteht und die gegenwärtige Verzögerung für Daten im Zwischenpuffer vor der Übertragung an das Schieberegister SK1 feststellt. Dieser Vorgang setzt sich fort, bis der Ausgabefeldpointer beim letzten Feld in der auszugebenden Speicherebene angelangt ist. Das Signal von der Logikschaltung **14** von Logikschaltung **10** steht nicht mehr an. Statt dessen bietet die Logikschaltung **10** bei der vorletzten Ausgabefeldpointerposition einen Ausgang an den nächsten Block, um dort die Steuerlogik zu aktivieren.

[0063] Obwohl dies nicht dargestellt ist, lassen sich viele steuerbare Verzögerungen separat von der Logikschaltung **10** anordnen, entweder um bei Bedarf das Signal C-1 mit der gewünschten Verzögerung auszugeben und/oder unterschiedliche Verzögerungen bei verschiedenen Ausgabepuffern o. dgl. zu erhalten.

[0064] Bei allen vorstehend beschriebenen Ausführungsformen des Speichers sind die Eingabe- und Ausgabedaten auf der gleichen Seite des Speicherblocks eingegeben bzw. ausgegeben worden. Bei der **Fig. 7** gezeigten Ausführungsform können ganze gespeicherte Vektoren entweder gemäß der vorstehend beschriebenen Ausführungsform ausgegeben werden, und es ist auch oder alternativ möglich, die Signalverarbeitung an Daten auszuführen, die gleichzeitig in verschiedenen Speicherblöcken gespeichert sind, d. h. parallel.

[0065] Die Signalverarbeitung kann zum Beispiel das durchgehende Ersetzen von Sonderzeichen in einer gespeicherten Textkette für spätere Ausgabe des gesamten gespeicherten Vektors durch die mit gestrichelten Linien dargestellten Ausgabepuffer betreffen. In einem solchen Fall erfolgt eine Rückübertragung der Daten nach der Verarbeitung an das Speicherfeld in der Speicherebene, aus der sie entnommen wurden. Alternativ oder ergänzend kann eine aufwendigere Datenverarbeitung durchgeführt werden. Ein Vorteil dieser Ausführungsform besteht darin, dass die Signalverarbeitung einer großer Menge gespeicherter Daten schnell durchgeführt werden kann.

[0066] Es kann auch möglich sein, irgendeine Art von Behandlung der gespeicherten Daten durchzuführen, um eine besondere Eigenschaft zu untersuchen oder ein Ergebnis zu erhalten. In einem solchen Fall besteht vielleicht überhaupt kein Interesse daran, einen eingegebenen Vektor auszugeben, und dann werden die mit gestrichelten Linien dargestellten Ausgabepuffer nicht benötigt. Diese sind deshalb gemäß der Erfindung überhaupt nicht erforderlich.

[0067] Jeder Speicherblock, zum Beispiel #m, hat einen an diesen angeschlossenen Verarbeitungspuffer mB_m, an den die Steuerlogik #mL den Inhalt in einer Speicherebene, Speicherfeld für Speicherfeld, auf gleiche Weise wie an ihren Ausgabepuffer senden kann. Der Inhalt des Speichers #m,B_m wird danach an eine örtliche Signalverarbeitungseinheit #m,LS_m übertragen, in der die besondere Signalverarbeitung ausgeführt wird. Anschließend können die verarbeiteten Daten entweder, wie vorstehend angegeben und mit gestrichelten Linien in der Figur gezeigt, zurück zum Speicher oder zu den Speicherfeldern geleitet werden, dem bzw. denen sie entnommen worden sind, oder sie können an eine für alle Blöcke gemeinsam angeordneten Verarbeitungseinheit GS übertragen werden, welche ein Ergebnis extrahiert und dieses ausgibt, oder es kann beides stattfinden. Es sei darauf hingewiesen, dass, abhängig vom Inhalt der einzelnen Speicherblöcke, diese Extraktion eines Ergebnisses auf separaten Speicherebenen, in Speicherfeldern in einer Speicherebene oder im gesamten Speicherblock durchgeführt werden kann. Da die Art von Signalverarbeitung, die angewandt wird, keine Bedeutung für das Konzept der Erfindung hat, ist nachstehend nur ein Beispiel dafür angeführt, nämlich die von gesamten Speicherblöcken oder Speicherebenen in Speicherblöcken.

[0068] Es ist auch möglich, die Verarbeitung in allen Speicherebenen im gleichen Speicherblock auf die oben angegebene Art zu durchzuführen. Eine Situation, in der dies vorkommen könnte, ist die Speicherung von Videosignalen für Bilderzeugung, bei der jede Zeile im Bild in unterschiedlichen Speicherebenen in den Speicherblöcken gespeichert werden kann. Die vorliegende Erfindung gestattet einen schnellen und einfachen Zugriff zu Teilen des Bildes und eine Bildverarbeitung an diesen Teilen, wobei die Zeilen in einem speziellen Speicherblock oder in mehreren aneinander grenzenden Speicherblöcken gesammelt sind. Dadurch wird eine schnell Bildverarbeitung bei jedem beliebigen Typ von Bildern möglich.

Anwendungsvorteile

[0069] Beispiele für Vektoren, die sich besonders für eine Speicherung nach der erfindungsgemäßen Methode eignen, sind Textmengen von einer Texiverarbeitungsanlage. Die Textmenge kann als Startpunkt eines Vektors und dessen Länge definiert werden. Es muss jedoch nicht genannt werden, wo sich dieser befindet. Ein Vorteil bei der Anwendung der Erfindung besteht in einer erheblichen Steigerung der Rechnergeschwindigkeit.

[0070] Traditionell werden Textmengen aufgeteilt auf verschiedene Stellen auf z. B. einer Festplatte gespeichert. Um an die gespeicherte Information kommen zu können, wird zuerst ein Sektor adressiert und dort Information abgegriffen, der nächste Sektor adressiert und dort Information abgegriffen usw. Daraus ergeben sich eine große Anzahl Adressierungsvorgänge und viel Warten auf Information.

[0071] Traditionell läuft auch das Abgreifen von Information aus einer nahe gelegenen Position relativ schnell ab. Je mehr Speicher durchsucht werden muss, desto länger dauert der Vorgang. Auf diese Weise verringert sich die Datenübertragungsgeschwindigkeit proportional der Größe des Speichers und der Streuung der Information.

[0072] Gemäß der Erfindung befindet sich der Startpunkt der Information in einem Vektor am Ein-/Ausgang des Speichers und ist eine Kette, d. h. in den verschiedenen Ebenen # m_i, Q_j , wobei j eine beliebige Zahl zwischen 1 und N ist, wobei sich die Kette vom Ein-/Ausgang tiefer in den Speicher hinein erstreckt. Ein neuer Vektor kann ohne Unterbrechung für eine Suche nach geeigneten Stellen in den Speicher eingegeben werden. Auf gleiche Weise ist die Ausgabe eines gespeicherten Vektors ohne Unterbrechung möglich bei den Ausführungsformen, bei denen eine Ausgabe erforderlich ist. Daraus ergibt sich, dass die Menge der Information unendlich groß werden darf, ohne dass dies in irgendeiner Weise die Zugriffszeit beeinträchtigt. Der Endpunkt eines Vektors wird für das Speicherfeld in den letzten Speicherblöcken markiert, wo der Vektor geschrieben und durch die Steuerlogik in dem Speicherblock gelesen werden kann, welcher diese Information in seinem internen Speicher speichert. Ein gleichzeitiges Lesen und Schreiben des Speichers kann stattfinden. Diese Steuerlogik kennt demzufolge den Endpunkt für sowohl die Eingabe als auch die Ausgabe und signalisiert dies den anderen Steuerlogiken, insbesondere während der Ausgabe des Ende des Vektor, durch die jeweiligen Ausgabepuffer.

[0073] Jede Steuerlogik iS umfasst deshalb eine Liste für jeden Informationsvektor mit Angaben darüber, um welche Informationsart es sich handelt, und über den Start- und Endpunkt des Vektoteils, der in seinem zugeordneten Speicherblock mit Verknüpfungen geschrieben werden kann. Dieser Teil der Steuerlogik ist zweckmäßigerweise in der Hardware ausgeführt, was effizienter ist als in einem dynamischen Speicher. Eine Ausführungsform der Steuerlogik für die Ausführungsformen gemäß **Fig. 1–5** kann entsprechend **Fig. 6** ohne Verzögerung **16** ausgeführt werden. Von außen wird der Speicher wie ein normaler Speicher aussehen. Die einzige Begrenzung bei einem Speicher gemäß der Erfindung besteht darin, dass er nur bei der Behandlung von Vektoren effizient ist. Der Speicher kann als Teil des normalen Speichers eines Computers eingeschlossen werden.

[0074] Das neue System gestattet sowohl ein schnelles Auffinden von Information als auch eine schnelle Übertragung von Daten. Ein Speicherblock kann ein Teil eines Chips darstellen, d. h. mehrere Speicherblöcke können auf dem gleichen Chip oder die Speicherblöcke können auf separaten Chips angeordnet werden. Alternativ kann jeder Speicherblock aus mehreren Chips oder aus Standardkomponenten aufgebaut sein. All diese Möglichkeit können zusammen auf dem Markt vorhanden sein, was eine flexible Anwendung mit einer Kombination verschiedener Einheiten mit sich führt.

[0075] In dem Fall, dass die Vektoren für eine Anwendung tendieren, so lang zu werden, dass sie in mehrere Vektoren unterteilt werden müssen, damit eine Speicherung im Speicher stattfinden kann, können zum Beispiel auf einfache Weise zusätzliche Speicherblöcke (Chips) nach dem letzten angeschlossen werden, um auf diese Weise den Specherraum für Vektoren zu vergrößern, die dann gespeichert werden können.

[0076] Ein anderes Anwendungsgebiet findet sich in der Telekommunikation, wo Daten in serieller Bitform ankommen können, in dem Fall, dass geeignete Umsetzereinheiten **5** für vom ersten Eingabepuffer kommende Information vorhanden sind, die die Umsetzung von Bit in Byte bewirken.

[0077] Die Eingabepuffer können auch so organisiert sein, dass sie serielle Schieberegister darstellen (Sonderfall). Es besteht auch die Möglichkeit, an den Eingabepuffern serielle und an den Ausgabepuffern parallele Schieberegister oder umgekehrt vorzusehen. Ob gewählt wird, zuerst eine Parallelisierung der Daten stattfinden soll, ist eine Frage der Implementierung. Dies ist sehr wahrscheinlich, wenn eine Anwendung in der zukünftigen Telekommunikation in Erwägung gezogen wird. Dort steht die Eingangsinformation in serieller Form an. Dann wird zuerst eine Parallelumsetzung in der Komponente **5** außerhalb des Speichers durchgeführt.

[0078] Jede Steuerlogik **4** und **mS** kann als einfaches Datenchip mit einem festen Programm ausgeführt sein. Alternativ kann die Steuerlogik **mS** in den Speicherebenen integriert sein. In **Fig. 4** und **5** ist die Gestaltung von Ablaufdiagrammen für die Steuerlogik dargestellt. Die Funktionen, die in den Ablaufdiagrammen erzielt werden, sind vorstehend beschrieben worden, weshalb die Ablaufdiagramme selbst keiner weiteren Erläuterungen bedürften, sondern vom Fachmann sofort verstanden werden.

[0079] Viele denkbare Modifikationen im Rahmen des Umfangs der Erfindung gehen aus den beiliegenden Patentansprüchen hervor.

Patentansprüche

1. Speicherstruktur zur Speicherung von Speichervektoren an zumindest einer Speicherstelle, wobei
 - der Speicher quer in Speicherblöcke (#1–#M) unterteilt ist;
 - jede der Speicherstellen im Speicher (#1,Qi–#M,Qi) parallel sich von einer Eingabe für Information und tiefer in den Speicher erstreckend angeordnet ist, so dass alle Speicherstellen je einen Teil in jedem Speicherblock haben;
 - Hilfsmittel (**1IB–MIB**) vorhanden sind, die wie Schieberegister für die Eingabe von Information in ununterbrochener Folge an die Speicherstellen im Speicher funktionieren; **dadurch gekennzeichnet**, dass
 - jeder Vektor für eine ungeteilte Speicherung in einer sequentiellen Reihenfolge mit dem Beginn des Vektors an der Eingabe des Speichers (Speicherfeld F1 in der Speicherebene #1,Qi) angeordnet ist;
 - die Adressierung so gestaltet ist, dass sie durch die Eingabe in den Speicher erfolgt;
 - jede der Speicherstellen im Speicher (#1,Qi–#M,Qi) nach der Konfiguration eine der Länge der großen Vektoren angepasste Länge aufweist und die Speicherstellen im Speicher durch Anschließen eines oder mehrerer Speicherblöcke erweitert werden können.
2. Speicherstruktur nach Anspruch 1, dadurch gekennzeichnet, dass jeder Speicherblock folgendes umfasst: als Mittel, die als Schieberegister funktionieren: einen Eingabepuffer (jIB) mit serieller Eingabe von Daten analog zu einem Schieberegister; als Speicherstellen: eine Vielzahl Speicherfelder (F1– FP), in welche Information vom Eingabepuffer sequentiell oder parallel eingegeben werden kann; und dass eine Steuerlogik (iS) in jedem Speicherblock angeordnet ist, welche Steuerlogik die Eingabe von Information in die Speicherfelder über die Eingabepuffer steuert; wobei die Speicherblöcke durch die in diesen vorhandenen, kaskadenförmig aneinander gekoppelten Eingabepuffer kaskadenförmig aneinander gekoppelt sind, so dass Information seriell durch mehrere der Puffer in aufeinander folgenden Speicherblöcken eingebbar ist; und dass die Steuerlogik im nächsten Speicherblock so angeordnet ist, dass, wenn die Speicherfelder im eigenen Speicherblock mit Information gefüllt sind oder Information aus diesen ausgelesen ist, damit die Steuerlogik im nächsten Speicherblock die Aufgabe, Information in den Speicherfeldern in diesem Speicherblock zu lesen, übernehmen kann.
3. Speicherstruktur nach einem der vorgenannten Ansprüche, dadurch gekennzeichnet, dass Mittel (**1UB–MUB**) vorhanden sind, die wie Schieberegister für die Ausgabe von Information in ungeteilter Folge von den Speicherplätzen im Speicher funktionieren.
4. Speicherstruktur nach Anspruch 1, dadurch gekennzeichnet, dass
 - ein Ausgabepuffer (jUB) mit serieller Ausgabe von Daten analog zu einem Schieberegister angeordnet ist;
 - Information in den Speicherfeldern (F1–FP) des Rechners während der Ausgabe von Daten sequentiell an den Ausgabepuffer übertragen wird; wobei die Steuerlogik (iS) in jedem Speicherblock so angeordnet ist, dass sie die Ausgabe von Information in den Speicherfeldern über die Ausgabepuffer steuert und die Steuerlogik im nächsten Speicherblock informiert, wenn der Speicher im eigenen Speicherblock von Information entleert ist, so dass diese die Aufgabe, Information in den Speicherfeldern in diesem Speicherblock zu lesen, übernehmen kann.
5. Speicherstruktur nach einem der vorgenannten Ansprüche, dadurch gekennzeichnet, dass die wie Schieberegister funktionierenden Mittel von solcher Art sind, dass sie eine serielle Zufuhr von Bytes handhaben können.
6. Speicherstruktur nach einem der Ansprüche 2–5, dadurch gekennzeichnet, dass die Übertragung zwischen Speicherfeldern und Puffern parallel stattfindet, während der gesamte Informationsgehalt gleichzeitig von der Steuerlogik in dem in Rede stehenden Speicherblock gesteuert wird, wenn die Übertragung stattfindet.
7. Speicherstruktur nach einem der Ansprüche 2–6, dadurch gekennzeichnet, dass die Information während der Ein- und/oder Ausgabe für eine kontinuierliche Übertragung an Eingabe- und/oder Ausgabepuffer in den verschiedenen Speicherblöcken ohne Handshake-Vorgänge zwischen diesen angeordnet ist.
8. Speicherstruktur nach einem der Ansprüche 2–7, dadurch gekennzeichnet, dass die Mittel (**1IB–MIB**, **1UB–MUB**), die wie Schieberegister funktionieren, parallel zueinander angeordnet sind und unabhängig von einander gesteuert werden können, was eine gleichzeitige Ein- und Ausgabe von Information in den/aus den Speicherstellen im Speicher gestattet.

9. Speicherstruktur nach einem der vorgenannten Ansprüche, dadurch gekennzeichnet, dass Mittel für eine Phasenverschiebung der Leseperiode relativ der Schreibperiode durch eine selektive Anzahl Taktpulse vorhanden sind.

10. Speicherstruktur nach Anspruch 9, dadurch gekennzeichnet, dass die Mittel für die Phasenverschiebung einen einstellbaren Relaisschaltkreis mit einer der gewünschten Phasenverzögerung einstellbaren Verzögerung umfasst.

11. Speicherstruktur nach einem der vorgenannten Ansprüche, dadurch gekennzeichnet, dass die in den einzelnen Speicherblöcken gespeicherte Information parallel zu jedem ihrer Verarbeitungsvorgänge gelesen werden kann.

12. Speicherstruktur nach Anspruch 11, dadurch gekennzeichnet, dass die Information im Speicherblock nach der Verarbeitung neu gespeichert werden kann.

13. Speicherstruktur nach Anspruch 11 oder 12, dadurch gekennzeichnet, dass die verarbeitete Information an eine Ergebniseeinheit übertragen werden kann, welche die verarbeitete Information von den Verarbeitungsvorgängen der Speicherblöcke zu einem Ergebnis zusammenstellt.

Es folgen 7 Blatt Zeichnungen

Anhängende Zeichnungen

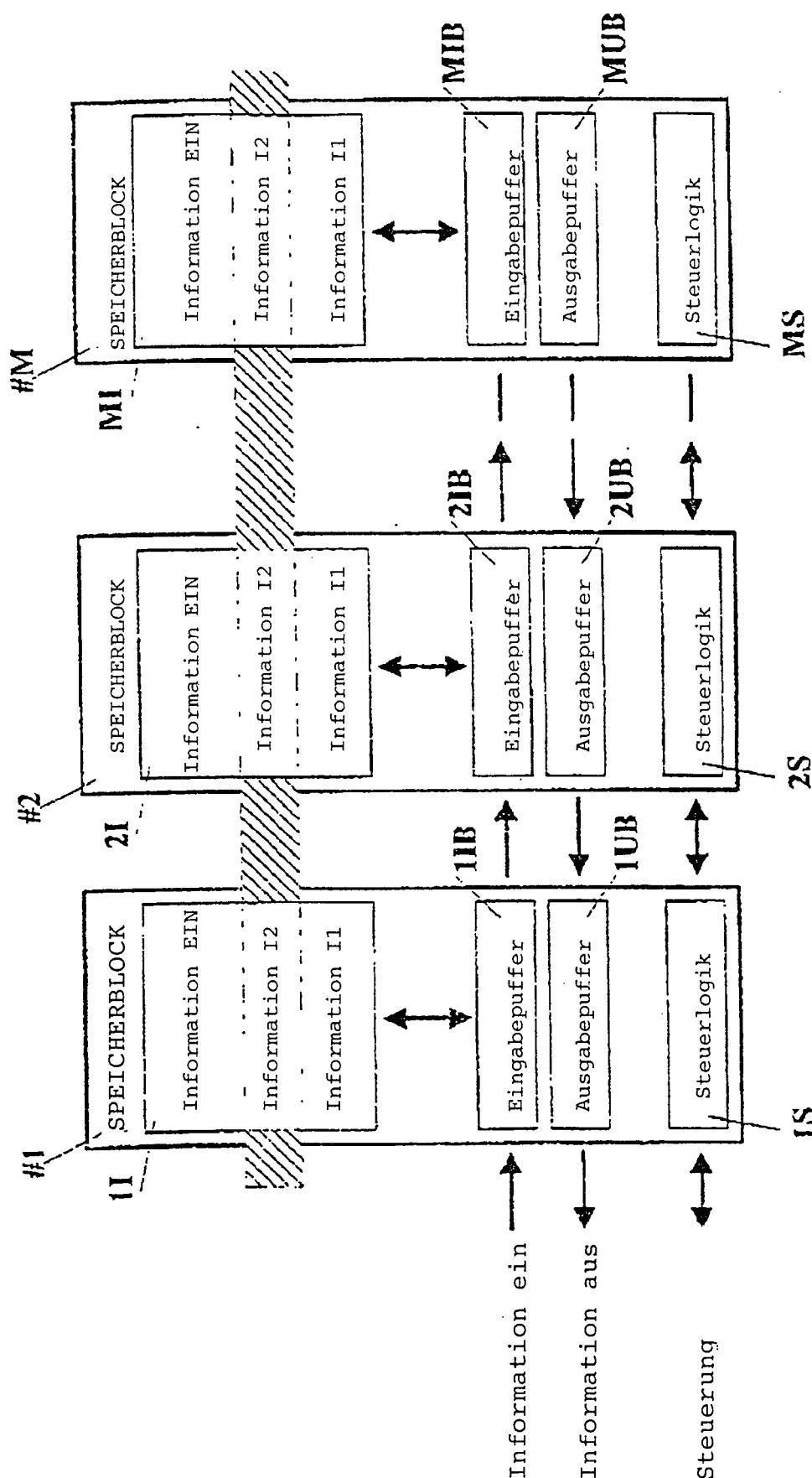


Fig. 1

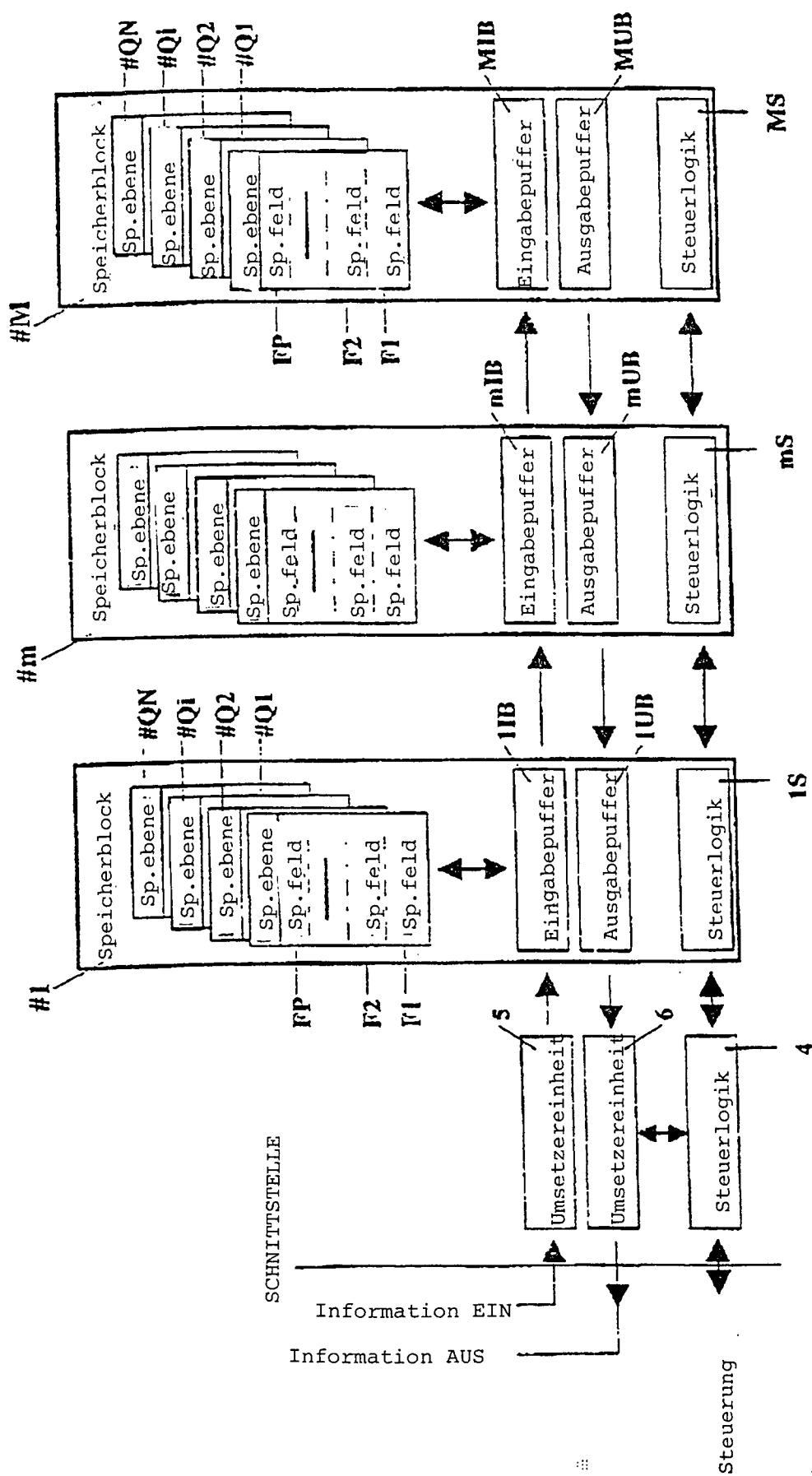


Fig. 2

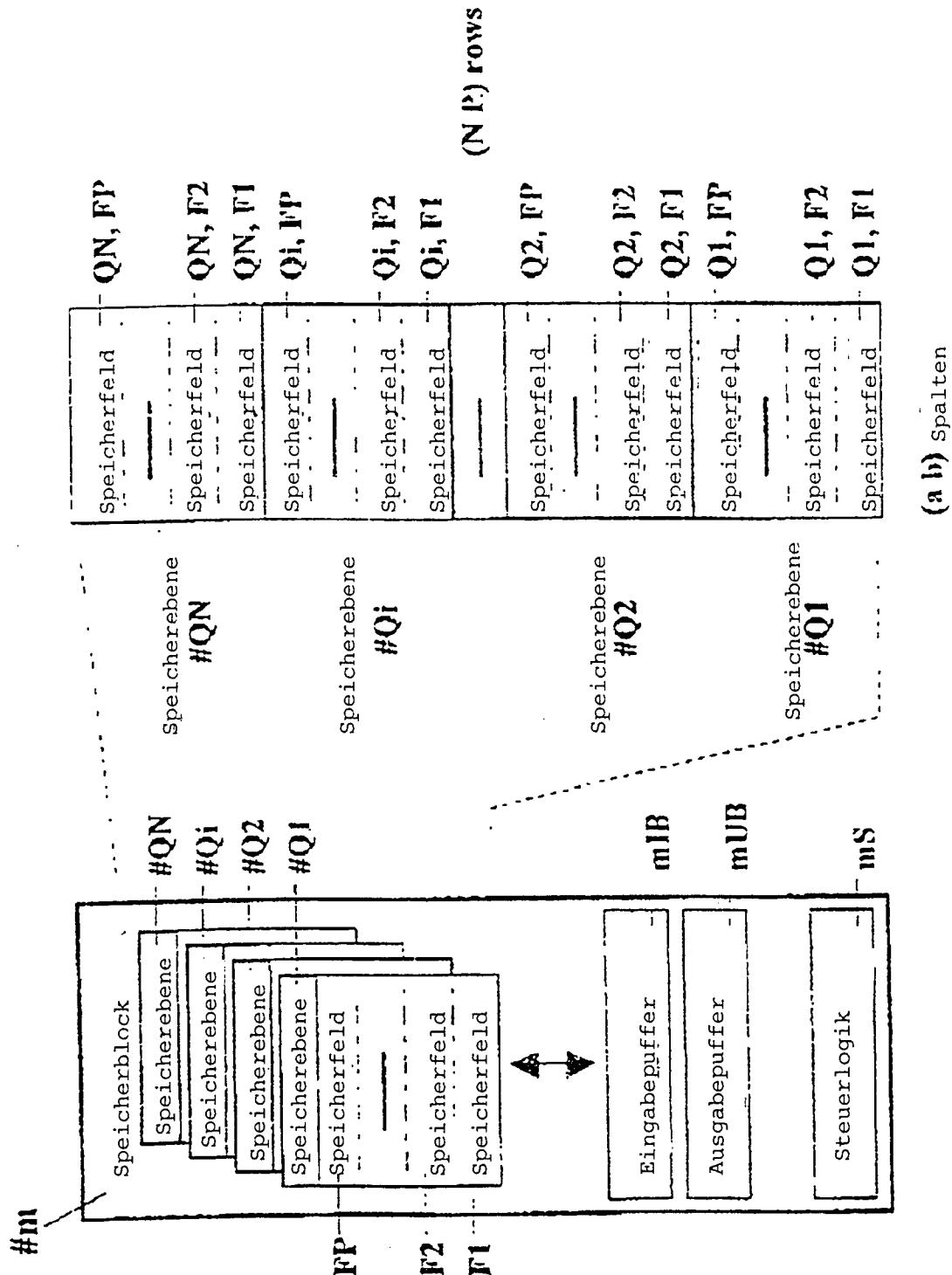


Fig. 3

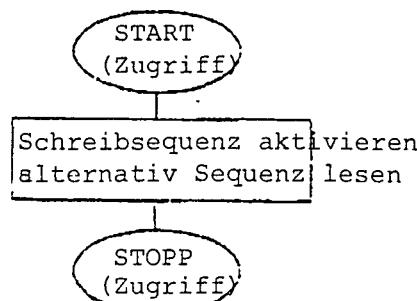


Fig. 4A

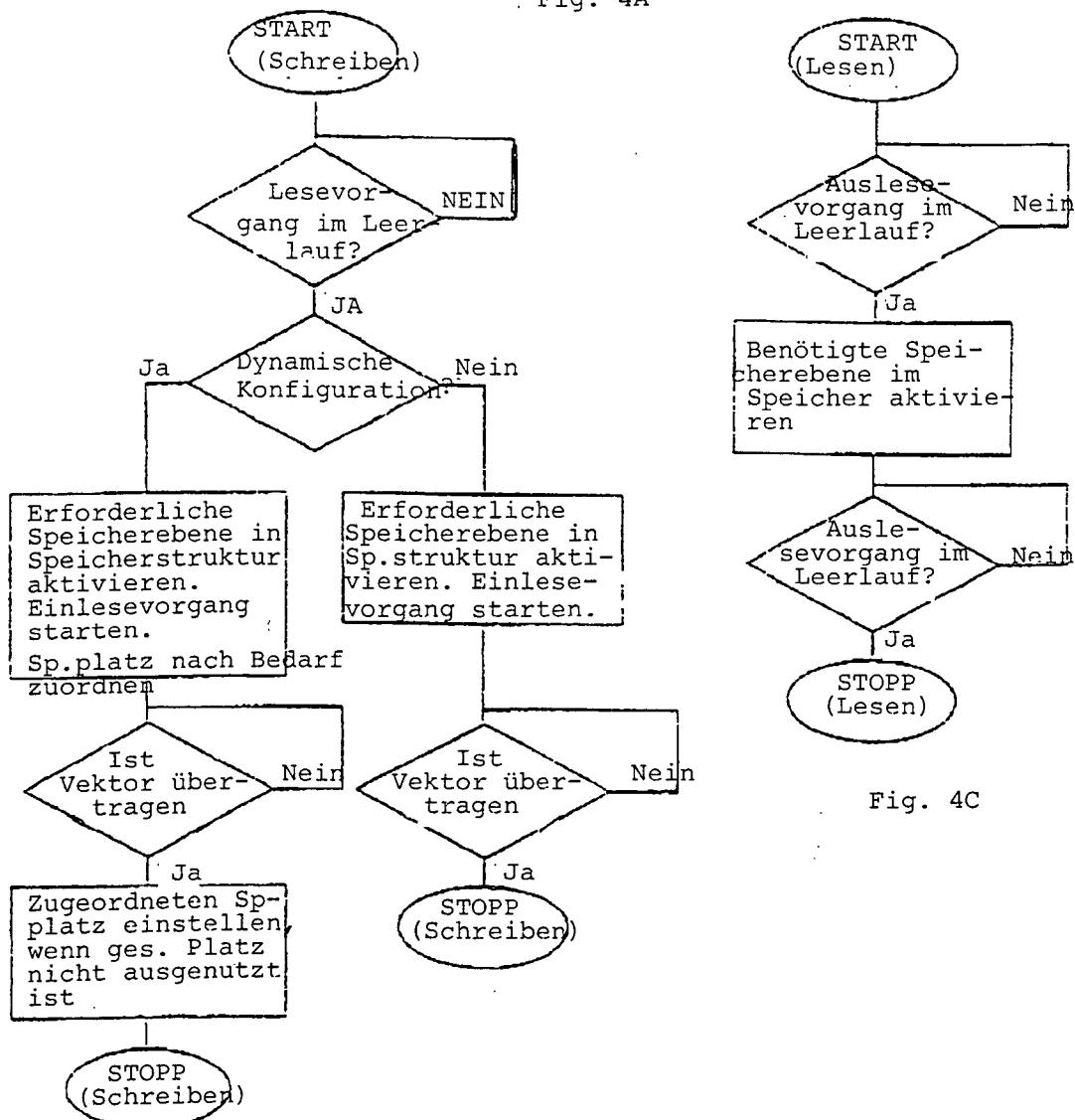


Fig. 4B

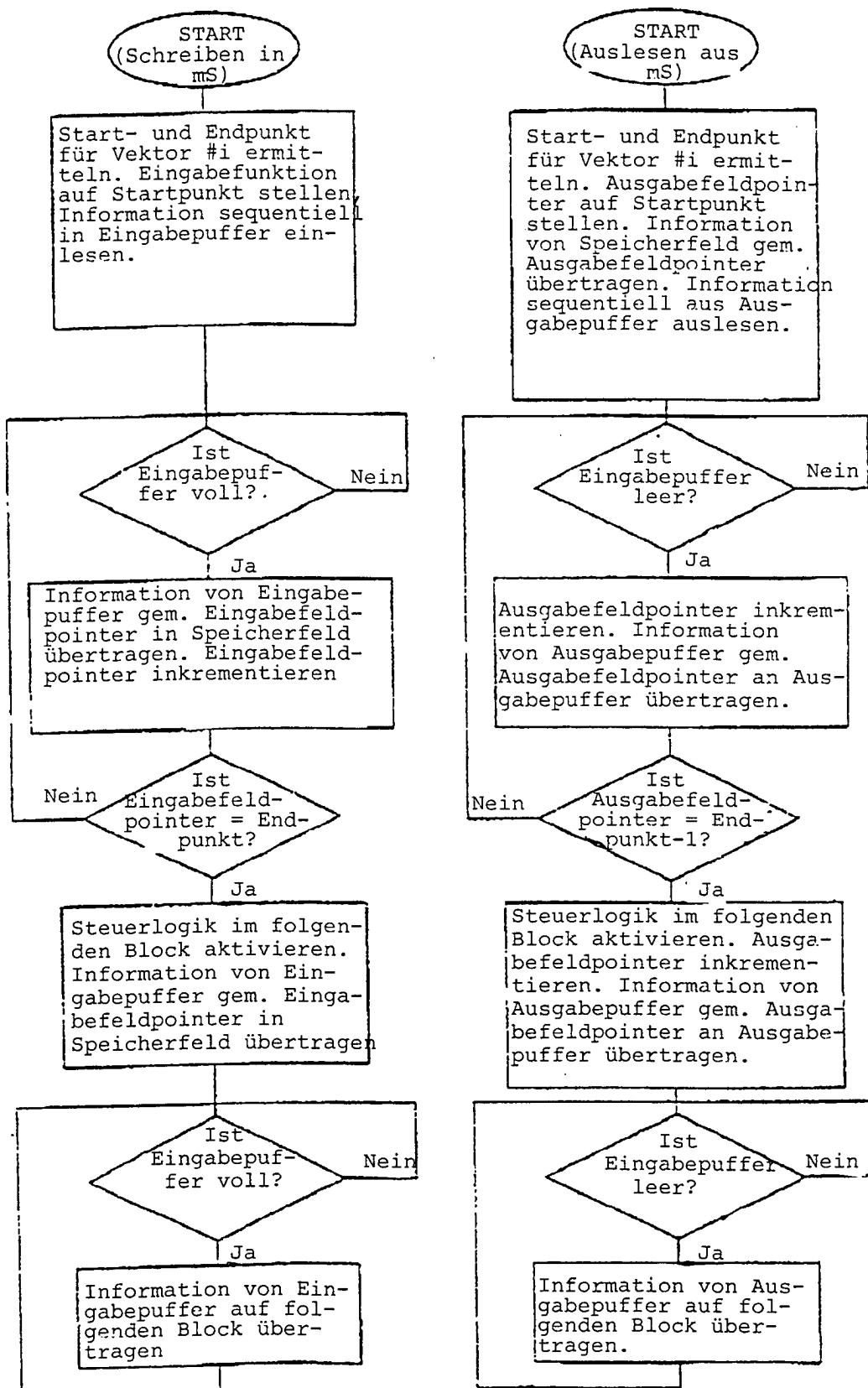


Fig. 5

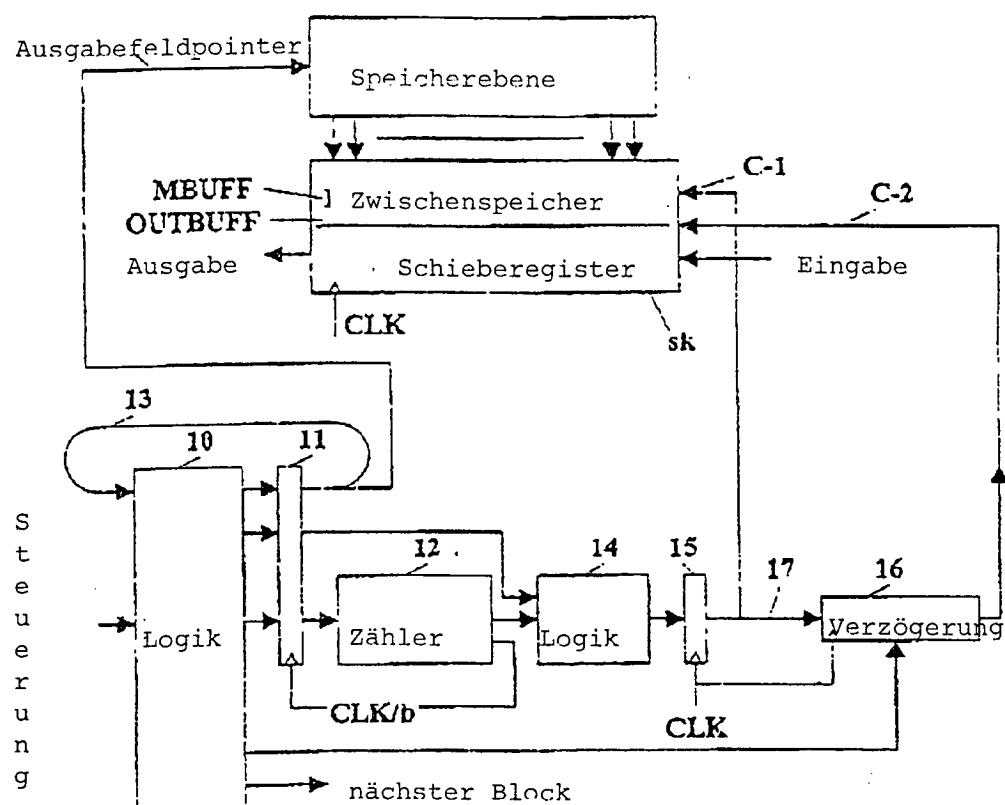
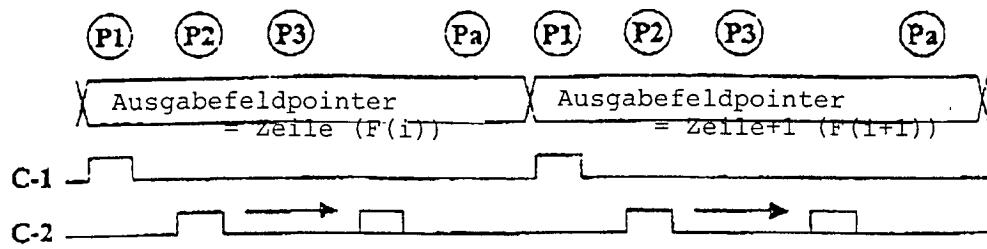


Fig. 6

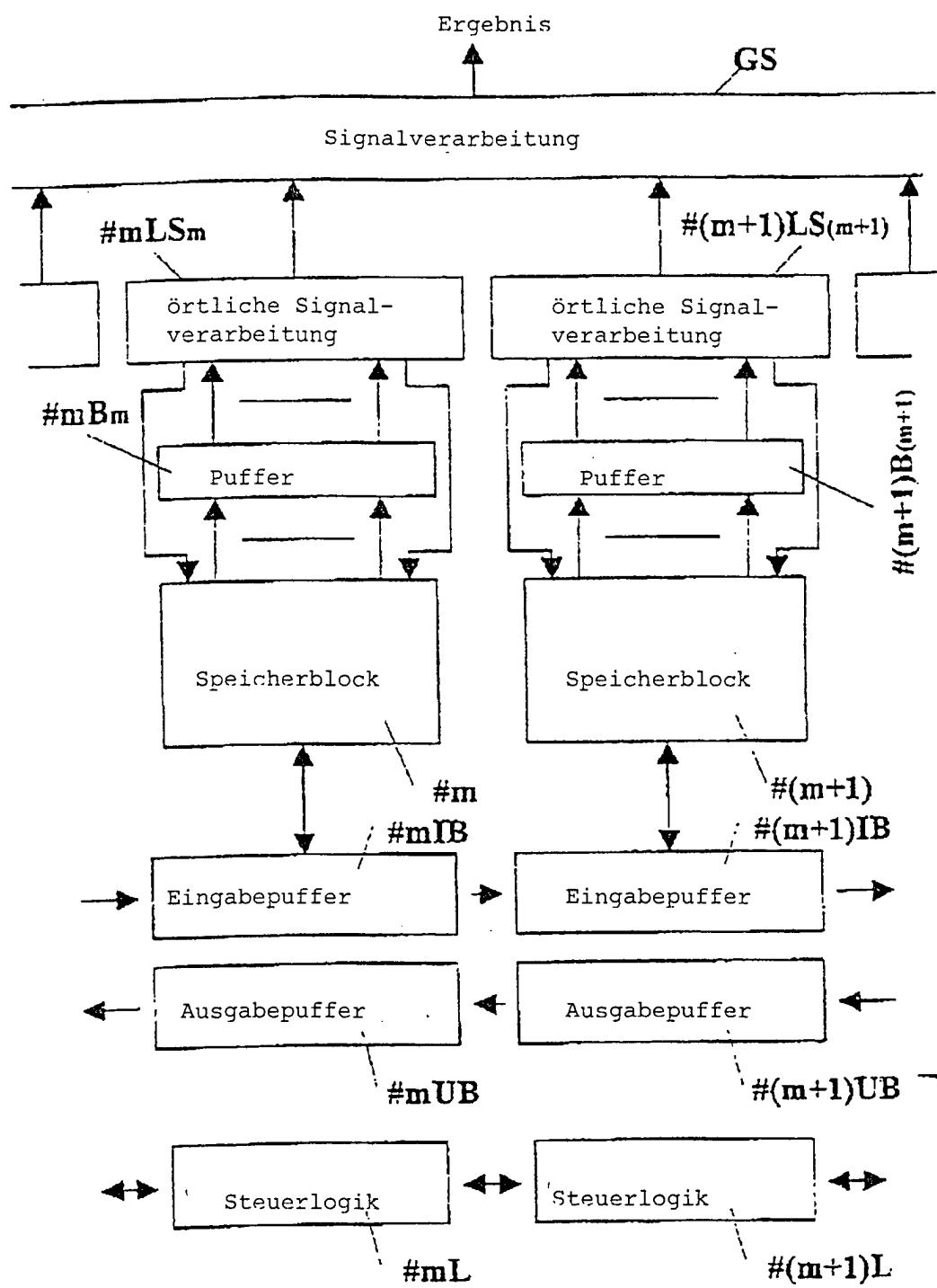


Fig. 7