



(19)  
Bundesrepublik Deutschland  
Deutsches Patent- und Markenamt

(10) **DE 697 37 304 T2** 2007.07.19

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 0 802 681 B1**

(21) Deutsches Aktenzeichen: **697 37 304.5**

(96) Europäisches Aktenzeichen: **97 301 685.0**

(96) Europäischer Anmeldetag: **13.03.1997**

(97) Erstveröffentlichung durch das EPA: **22.10.1997**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **31.01.2007**

(47) Veröffentlichungstag im Patentblatt: **19.07.2007**

(51) Int Cl.<sup>8</sup>: **H04N 7/26 (2006.01)**  
**H03M 7/42 (2006.01)**

(30) Unionspriorität:  
**9608150            25.03.1996        KR**

(73) Patentinhaber:  
**LG Electronics Inc., Seoul, KR**

(74) Vertreter:  
**Meissner, Bolte & Partner GbR, 80538 München**

(84) Benannte Vertragsstaaten:  
**DE, FR, GB**

(72) Erfinder:  
**Kim, Su Hwan, Seoul, KR; Bae, Seong Ok,  
Uiwang-si, Kyungki-do, KR**

(54) Bezeichnung: **Dekoder für Codes variabler Länge**

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**

## HINTERGRUND DER ERFINDUNG

## Gebiet der Erfindung

**[0001]** Die vorliegende Erfindung bezieht sich auf einen VLC-Decoder (VLC – variabel langer Code).

## Erläuterung des verwandten Stands der Technik

**[0002]** Im Allgemeinen ist der VLC (variabel langer Code), der manchmal auch als Hoffmann-Code bezeichnet wird, ein optimaler Code, der in einem Hochgeschwindigkeitsübertragungssystem verwendet wird, und auf den statistische Kennwerte eines Datenelements auf eine Weise angewendet werden, dass, sobald die statistischen Kennwerte eines Datenelements bekannt sind, das kürzere Codewort bestimmt wird, je mehr Videodaten generiert werden, und das längere Codewort bestimmt wird, je weniger Videodaten generiert werden. Wenn durch ein solches Verfahren geeignete Codewörter für alle möglichen Videodaten bestimmt werden, kann eine durchschnittliche Bitlänge von Codewörtern auf ein kürzeres Maß als eine ursprünglich beabsichtigte, durchschnittliche Bitlänge komprimiert werden. Deshalb handelt es sich bei diesem VLC um ein verlustfreies Codierverfahren, das häufig bei Daten verwendet wird, deren statistische Kennwerte hinlänglich bekannt sind.

**[0003]** Dieser VLC wird momentan in Systemen verwendet, die eines Verfahrens zum Komprimieren einer Videonutzinformation zur dichteren Komprimierung der Daten bedürfen, wie etwa MPEG 1 (Moving Picture Expert Group 1), MPEG 2 (Moving Picture Expert Group 2) und HDTV (High Definition Television, also hochauflösendes Fernsehen). Da im Laufe der Zeit nach immer schnelleren Decodern für Codes mit variabler Länge verlangt wurde, wurden Strukturen vorgeschlagen, die eine Architektur auf PLA-Basis (Programmable Logic Array – programmierbare Logikanordnung) und nicht auf Baubasis verwenden, wovon sich die meisten auf eine Verkürzung der längsten Verzögerungszeit des Decoders für Codes mit variabler Länge konzentrieren. Falls jedoch der komplizierte Decoder für Codes mit variabler Länge zusammen mit Schaltungen, die um ihn herum in einem System wie MPEG 1, MPEG 2 oder HDTV, an ihn angeschlossen sind, einen Funktionsblock bilden, hängt die Geschwindigkeit dieses Funktionsblocks von der längsten Verzögerungszeit ab, die in Teilen auftritt, die den Decoder für Codes mit variabler Länge und die ihn umgebenden Schaltungen verbinden. Deshalb sollte bei der Bereitstellung eines Decoders für Codes mit variabler Länge, die einen Funktionsblock bilden, der mit ihm umgebenden Schaltungen in einem System wie MPEG 1, MPEG 2 oder HDTV verbunden ist, nicht nur die längste Verzögerungszeit

des Decoders für Codes mit variabler Länge selbst berücksichtigt werden, sondern auch die Verzögerungszeit, die in den Teilen auftritt, die den Decoder für Codes mit variabler Länge und die ihn umgebenden Schaltungen verbinden. Wie in [Fig. 1](#) gezeigt ist, wird ein codiertes Wort (oder Datenelement), das an einem Eingabepuffer **2** eingeht, über einen Datenweg **3**, in dem sich ein Decoder für Codes mit variabler Länge befindet, unter der Steuerung eines Steuerlogikteils **1** decodiert und über einen Ausgabepuffer **4** ausgegeben. Wie aus Vorstehendem zu sehen ist, bildet ein Decoder für Codes mit variabler Länge (VLC) im Allgemeinen zusammen mit ihm umgebenden und mit ihm verbundenen Schaltungen einen Funktionsblock. Obwohl in diesem Falle eine Ausgabe der VLC-Tabelle direkt in einem Eingabepuffer in einem anderen Funktionsblock gespeichert werden kann, wird die Ausgabe in den meisten Fällen jedoch auf eine oder mehrere, ihn umgebende Schaltungen angelegt oder im Eingabepuffer eines anderen Funktionsblocks gespeichert, nachdem sie im anderen Funktionsblock in eine andere geeignete Form umgesetzt wurde.

**[0004]** Eine Vorrichtung aus dem Stand der Technik zum Decodieren eines variabel langen Codes nach einem Stand der Technik (US-Patent Nr. 5,245,338) wird nun mit Bezug auf [Fig. 2](#) erläutert. [Fig. 2](#) stellt ein Blockschema dar, das einen Decoder für Codes mit variabler Länge aus dem Stand der Technik zeigt.

**[0005]** Mit Bezug auf [Fig. 2](#) umfasst der Decoder für Codes mit variabler Länge aus dem Stand der Technik einen Eingabepuffer **201**, ein erstes Zwischenspeicherteil **202**, ein erstes Multibit-Schieberegister **203**, ein zweites Zwischenspeicherteil **204**, ein zweites Multibit-Schieberegister **205**, eine VLC-Tabelle **206**, einen ersten Zwischenspeicher **207**, ein Steuersignalgenerierungsteil **208** und einen Ausgabepuffer **209**. Das erste Zwischenspeicherteil **202** umfasst einen zweiten Zwischenspeicher **202a** und einen dritten Zwischenspeicher **202b**, das zweite Zwischenspeicherteil **204** umfasst einen vierten Zwischenspeicher **204a** und einen fünften Zwischenspeicher **204b**. Das Steuersignalgenerierungsteil umfasst einen Addierer **208a** und einen sechsten Zwischenspeicher **208b**.

**[0006]** Mit Bezug auf [Fig. 2](#) decodiert die VLC-Tabelle **206** mit Zufallslogik kontinuierlich aufeinanderfolgende Codewörter, um im Ansprechen auf ein Lesesignal oder eine feststehende Zeichentaktfrequenz decodierte Codewörter mit einer feststehenden Länge zu erzeugen. Das erste Zwischenspeicherteil **202** und das Multibit-Schieberegister **203** bilden geeigneter Weise die Übergabestelle zwischen dem Eingabepuffer **201** und dem zweiten Zwischenspeicherteil **204**. Der Eingabepuffer **201** speichert einen seriellen Bitstrom von Codewörtern mit variablen Längen, die aus einem Datenkanal her eingeht, in einem Daten-

segment mit einer feststehenden Länge ab. Das Steuersignalgenerierungsteil **208** stellt parallel 16 Bit große Daten im Ansprechen auf das Lesesignal aus dem sechsten Zwischenspeicher **208b** her. Die VLC-Tabelle **206** stellt eine Bitlänge eines Codeworts mit einer unbekanntem Bitlänge und ein decodiertes Codewort mit der Bitlänge her, die an den Zwischenspeicher **207** bzw. den Ausgabepuffer **209** angelegt werden sollen. Die Ausgabeseite des zweiten Multibit-Schieberegisters **205** stellt der VLC-Tabelle **206** ein Decodierfenster bereit, und wird durch die Bitlängen der decodierten Codewörter, die im ersten Zwischenspeicher **207** aufgelaufen sind, bis zu einem Takt gesteuert, der vor dem momentanen Takt liegt. Dem zweiten Multibit-Schieberegister **205** werden Codewörter mit variabler Länge aus den Zwischenspeichern **204a** und **204b** bereitgestellt. Jeder der Zwischenspeicher **204a** und **204b** hat eine Bitkapazität, die zumindest dem Codewort mit der längsten Länge entspricht. Der zweite Zwischenspeicher **204** liefert aufeinanderfolgende Bits mit mindestens zwei variabel langen Codewörtern. Eine Ausgabeseite des zweiten Multibit-Schieberegisters **205** ist an eine Eingabeseite der VLC-Tabelle **206** angeschlossen, um Bits des längsten zu decodierenden Codeworts der aufeinanderfolgenden Bits mit einer Länge anzulegen, die dem Zweifachen des längsten Codeworts entspricht, das im vierten Zwischenspeicher **204a** und im fünften Zwischenspeicher **204b** gespeichert ist, während Eingabedaten aus dem fünften Zwischenspeicher **204b** dem ersten Zwischenspeicher **202** und dem ersten Multibit-Schieberegister **203** bereitgestellt werden. Die VLC-Tabelle **206** legt die Länge des aus dem zweiten Multibit-Schieberegister **205** eingegangenen Codeworts an den ersten Zwischenspeicher **207** und das decodierte Codewort an den Ausgabepuffer **209** an. Beim nächsten Takt nach dem momentanen Takt wird das zweite Multibit-Schieberegister **205** durch die Längen der Codewörter angesteuert, die bis zu einem früheren Takt decodiert wurden und vom ersten Zwischenspeicher **207** her eingegangen sind. Dementsprechend wird ein Ausgabefenster des zweiten Multibit-Schieberegisters **205** um die Bitlängen der Codewörter verschoben, die bis zu einem früheren Takt decodiert wurden und vom ersten Zwischenspeicher **207** her eingegangen sind, um erneut ab dem ersten Bit eines zu decodierenden Codeworts zu beginnen. Da alle Bits im vierten Zwischenspeicher **202a** bei allen Takten transformiert werden, wird der fünfte Zwischenspeicher **204b** wieder mit einem entsprechenden Bitstrom aus dem ersten Multibit-Schieberegister **203** aufgefüllt, das eine Übergabestellenfunktion in einem jedem Takt entsprechenden Zyklus hat. Und zwar erscheint ein aufeinanderfolgender Bitstrom beginnend ab dem ersten Bit eines decodierten Codeworts auf der Eingabeseite des zweiten Multibit-Schieberegisters **205**. Und, obwohl das erste Multibit-Schieberegister **203** dem fünften Zwischenspeicher **204b** einen Bitstrom liefert, ist dieser Bitstrom nicht in der längs-

ten Verzögerungszeit enthalten, weil dieser Bitstrom direkt an den fünften Zwischenspeicher **204b** angelegt wird. Somit sollte der Bitstrom von bis zu einem früheren Takt decodierten Codewörtern, die auf der Eingabeseite des zweiten Multibit-Schieberegisters **205** erscheinen, fehlerfrei während eines Zyklus aktualisiert werden, der jedem Takt entspricht. Wie bereits erläutert, wird aus dem eine Übergabestelle bildenden ersten Multibit-Schieberegister **203** dem fünften Zwischenspeicher **204b** eine Ausgabe bereitgestellt. Einer Eingabeseite des eine Übergabestelle bildenden Multibit-Schieberegisters **203** wird ein Codewort von den beiden aufeinanderfolgenden Zwischenspeichern **202a** und **202b**, die Kapazität für die längste Codewortlänge haben, und dem Eingabepuffer **201** bereitgestellt. Der eine Übergabestelle bildende zweite und dritte Zwischenspeicher **202a** und **202b** haben eine Bitkapazität für die längste Codewortlänge, die vom Eingabepuffer **201** ausgegeben werden kann. Die Ausgabe aus dem ersten Multibit-Schieberegister **203** wird sowohl durch die Länge des decodierten Codeworts aus der Ausgabe aus der VLC-Tabelle **206** als auch der Ausgabe des Addierers **208a** gesteuert, der die Längen der bis zu einem Takt vor dem momentanen Takt decodierten Codewörter addiert, die vom sechsten Zwischenspeicher **208b** im Steuersignalgenerierungsteil **208** her eingegangen sind. Die Ausgabe des ersten Multibit-Schieberegisters **203** wird je nach den Bitlängen der decodierten Codewörter verschoben, die durch den Addierer **208a** addiert wurden, um Ausgaben des zweiten Multibit-Schieberegisters **205** und die aufeinanderfolgenden Bitströme auszugeben. Eine Bitfolgenreihe aus dem eine Übergabestelle bildenden ersten Multibit-Schieberegister **203** erscheint auf der Eingabeseite des zweiten Multibit-Schieberegisters **205**. Wenn die Bitlängen der decodierten Codewörter, die im Addierer **208a** addiert wurden, die längste Codewortlänge überschreiten, wird ein Übertrag generiert. Das bedeutet, dass alle in Speicherung im ersten eine Übergabestelle bildenden Zwischenspeicher **202a** befindlichen Bits über das erste Multibit-Schieberegister **203** zum fünften Zwischenspeicher **202a** übertragen wurden. Aufgrund dieses Übertrags werden alle im dritten Zwischenspeicher **202b** vorhandenen Bits zum zweiten Zwischenspeicher **202a** übertragen und die Codewörter aus dem Eingabepuffer **201** werden zu dem eine Übergabestelle bildenden zweiten Zwischenspeicher **202a** übertragen. Und gleichzeitig werden dem dritten Zwischenspeicher **202b** und dem ersten Multibit-Schieberegister **203** aus dem Eingabepuffer **201** neue 16 Bit große Daten bereitgestellt.

**[0007]** Wie bereits erläutert wurde, bildet bei der Implementierung eines qualitativ hochwertigen Bildsystems wie MPEG 1, MPEG 2 oder HDTV der Decoder für Codes mit variabler Länge in vielen Fällen einen Funktionsblock mit Schaltungen, die ihn umgeben und mit ihm verbunden sind. Und, obwohl die Ausga-

ben aus der VLC-Tabelle und decodierte Codebits direkt in Eingabepuffern in voneinander unterschiedlichen Funktionsblöcken gespeichert werden können, werden sie in den meisten Fällen an Schaltungen um die VLC-Tabelle als Eingabedaten angelegt oder als Steuersignale für andere Funktionsblöcke verwendet. Es ist deshalb notwendig, die Bits der Codewörter in kürzester Zeit zu decodieren und gleichzeitig das von der längsten Verzögerungszeit herrührende Problem zu verhindern. In diesem Fall tritt die längste Verzögerungszeit des in Frage kommenden Funktionsblocks in Teilen, welche die VLC-Tabelle mit sie umgebenden Schaltungen verbinden, und nicht in einer internen Schaltung des VLC-Decoders selbst auf, und zwar, wenn die diesen umgebenden Schaltungen kompliziert werden, was darüber hinaus noch das Problem aufwirft, dass die Geschwindigkeit eines schnellen Datenübertragungssystems eingeschränkt wird.

**[0008]** Dabei besteht mit Bezug auf den in [Fig. 2](#) gezeigten Decoder für Codes mit variabler Länge aus dem Stand der Technik das Problem, dass ein größer bemessener VLC-Decoder benötigt wird, weil das eine Übergabestelle bildende erste Multibit-Schieberegister und das decodierende zweite Multibit-Schieberegister gebraucht werden. Außerdem stellte sich das Problem, dass eine Verzögerungszeit in den Teilen auftritt, die den VLC-Decoder über die VLC-Tabelle **6** und das decodierende zweite Multibit-Schieberegister **205** mit den ihn umgebenden Schaltungen verbinden.

**[0009]** Ähnliche Probleme in Bezug auf Verzögerungszeit, die bei Decodern für Codes mit variabler Länge auftreten, sind in den Dokumenten US 5173695 und EP 0681404 offenbart.

**[0010]** Das Dokument US 5173695 offenbart in [Fig. 1](#) einen Decoder für Codes mit variabler Länge, der Folgendes umfasst:  
ein erstes Speicherteil, um ein eingegangenes Codewort im Ansprechen auf ein Lesesignal temporär zu speichern und auszugeben;  
ein Multibit-Schieberegister (Barrel Shifter), um ein daran befindliches Decodierausgabefenster im Ansprechen auf ein Schiebesignal beim Ausgeben einer vorbestimmten Anzahl von Bits des Codeworts aus dem Schaltteil zum Decodieren zu verschieben;  
eine Tabelle, um ein decodiertes Codewort und eine Bitlänge des decodierten Codeworts aus der vorbestimmten Anzahl von Bits, die vom Multibit-Schieberegister ausgegeben wurden, auszugeben;  
ein Steuersignalgenerierungsteil, um die Bitlänge eines aus der Tabelle erhaltenen Codeworts mit dem momentanen Takt beim Generieren des Schaltsignals, des Schiebesignals und des Lesesignals zu verwenden.

**[0011]** Das Dokument EP 0681404 offenbart in den

**Fig. 2A, 2B** einen Decoder für Codes mit variabler Länge, der dem im Dokument US 5245338 offenbarten ähnlich ist, wobei in **Fig. 2B** ferner noch Multiplexer **16** und **17** zwischen dem Ausgang der drei Zwischenspeicher **13, 14, 15** und dem ersten Multibit-Schieberegister **18** eingesetzt sind.

## ZUSAMMENFASSUNG DER ERFINDUNG

**[0012]** Die Ausführungsformen der vorliegenden Erfindung richten sich auf einen Decoder für Codes mit variabler Länge, der im Wesentlichen eines oder mehrere der Probleme umgeht, die von Einschränkungen und Nachteilen aus dem verwandten Stand der Technik herrühren.

**[0013]** Nach einer bevorzugten Ausführungsform wird ein Decoder für Codes mit variabler Länge bereitgestellt, der die längste Verzögerungszeit verkürzen kann, die in Teilen auftritt, die einen VLC-Decoder und diesen umgebende Schaltungen verbinden, und bei dem eine Chip-Fläche einer integrierten Schaltung eines Funktionsblocks bei der Herstellung verkleinert werden kann.

**[0014]** Zusätzliche Merkmale und Vorteile der Erfindung werden in der folgenden Beschreibung dargelegt, ergeben sich teilweise aus der Beschreibung oder lassen sich durch die praktische Umsetzung der Erfindung in Erfahrung bringen. Die Ziele und weitere Vorteile der Erfindung werden durch den Aufbau in die Tat umgesetzt und erreicht, der insbesondere in der schriftlichen Beschreibung und deren Ansprüchen sowie in den beigefügten Zeichnungen herausgestellt ist.

**[0015]** Um diese und weitere Vorteile zu erzielen, und in Übereinstimmung mit einem ersten Aspekt der Erfindung, wird ein Decoder für Codes mit variabler Länge bereitgestellt, der die Merkmale umfasst, die in den beigefügten Ansprüchen den Gegenstand von Anspruch 1 ausmachen.

**[0016]** Es sollte klar sein, dass sowohl die vorstehende allgemeine als auch die folgende ausführliche Beschreibung beispielhaft und erläuternd sind und eine weitere Erklärung der Erfindung, so wie sie beansprucht ist, bieten soll.

## KURZE BESCHREIBUNG DER ZEICHNUNGEN

**[0017]** Die begleitenden Zeichnungen, die mit aufgenommen sind, um ein weiteres Verständnis der Erfindung zu bieten, und die in diese technische Beschreibung eingegliedert sind und einen Teil von ihr bilden, stellen Ausführungsformen der Erfindung dar und dienen zusammen mit der Beschreibung dazu, deren Grundgedanken zu erklären.

**[0018]** [Fig. 1](#) stellt ein Funktionsblockschema eines

Systems dar, das einen Decoder für Codes mit variabler Länge umfasst;

[0019] [Fig. 2](#) stellt ein Blockschema dar, das einen Decoder für Codes mit variabler Länge aus dem Stand der Technik zeigt;

[0020] [Fig. 3](#) stellt ein Blockschema dar, das einen Decoder für Codes mit variabler Länge nach einer ersten Ausführungsform der vorliegenden Erfindung zeigt;

[0021] [Fig. 4](#) stellt einen Aufbau eines Bitstroms dar, der an den in [Fig. 3](#) gezeigten Eingabepuffer angelegt wird;

[0022] [Fig. 5](#) ist eine Tabelle zur Erläuterung des Funktionsablaufs des in [Fig. 3](#) gezeigten Decoders für Codes mit variabler Länge bei jedem Takt;

[0023] [Fig. 6](#) stellt ein Blockschema dar, das einen Decoder für Codes mit variabler Länge nach einer zweiten Ausführungsform der vorliegenden Erfindung zeigt; und

[0024] [Fig. 7](#) stellt ein Blockschema dar, das einen Decoder für Codes mit variabler Länge nach einer dritten Ausführungsform der vorliegenden Erfindung zeigt.

#### AUSFÜHRLICHE BESCHREIBUNG DER BEVORZUGTEN AUSFÜHRUNGSFORMEN

[0025] Nun wird im Einzelnen Bezug auf die bevorzugten Ausführungsformen der vorliegenden Erfindung genommen, wovon Beispiele in den begleitenden Zeichnungen dargestellt sind.

[0026] [Fig. 3](#) stellt ein Blockschema dar, das einen Decoder für Codes mit variabler Länge nach einer ersten Ausführungsform der vorliegenden Erfindung zeigt.

[0027] Mit Bezug auf [Fig. 3](#) umfasst der Decoder für Codes mit variabler Länge nach einer ersten Ausführungsform der vorliegenden Erfindung einen Eingabepuffer **310**, um Codewörter, die seriell an ihn angelegt werden, parallel anhand von 16 Bits zu speichern und die Codewörter im Ansprechen auf ein Lesesignal auszugeben, ein Zwischenspeicherteil **320**, um die Codewörter temporär zu speichern und auszugeben, die im Ansprechen auf das Lesesignal eingegangen sind, ein Multiplexerteil **330**, um die Codewörter aus dem Zwischenspeicherteil **320** zu empfangen und sie selektiv im Ansprechen auf ein Multiplexersignal auszugeben, ein Multibit-Schieberegister **340**, um ein Ausgabefenster von diesem im Ansprechen auf ein Verschiebungssignal zu verschieben und eine vorbestimmte Anzahl an Bits der Codewörter aus dem Multiplexerteil **330** zum Decodieren aus-

zugeben, einen Zwischenspeicher **350**, um die Codewörter aus dem Multibit-Schieberegister **340** temporär zu speichern und auszugeben, eine PLA (programmierbare Logikanordnung) **360** als VCL-Tabelle, um ein decodiertes Codewort und eine Bitlänge des Codeworts aus dem Zwischenspeicher **350** auszugeben, ein Steuersignalgenerierungsteil **370**, um die Bitlängen der decodierten Codewörter bis zu einem Takt genau vor dem momentanen Takt und eine Bitlänge des Codeworts aus der PLA **360** zum momentanen Takt zu addieren und die addierten Bitlängen bei der Bereitstellung des Lesesignals, des Multiplexersignals und des Schiebesignals für den Eingabepuffer **310**, das Zwischenspeicherteil **320**, das Multiplexerteil **330** bzw. das Multibit-Schieberegister **340** zu verwenden.

[0028] Das Zwischenspeicherteil **320** umfasst einen ersten Zwischenspeicher **321**, um die Codewörter aus dem Eingabepuffer **310** im Ansprechen auf das Lesesignal temporär zu speichern und auszugeben, einen zweiten Zwischenspeicher **322**, um die Codewörter aus dem ersten Zwischenspeicher **321** im Ansprechen auf das Lesesignal temporär zu speichern und auszugeben, einen dritten Zwischenspeicher **323**, um die Codewörter aus dem zweiten Zwischenspeicher **322** im Ansprechen auf das Lesesignal temporär zu speichern und auszugeben. Das Multiplexerteil **330** umfasst einen ersten Multiplexer **331**, um eines der Codewörter aus dem Eingabepuffer **310** und dem ersten und zweiten Zwischenspeicher **321** und **322** im Ansprechen auf das Multiplexersignal im Zwischenspeicherteil **320** auszuwählen und das ausgewählte Codewort als 16 Rückwärtsbits des Multibit-Schieberegisters **340** auszugeben, und einen zweiten Multiplexer **332**, um eines der Codewörter aus dem ersten, zweiten und dritten Zwischenspeicher **321**, **322** und **323** im Ansprechen auf das Multiplexersignal im Zwischenspeicherteil **320** auszuwählen und das ausgewählte Codewort als 16 Vorwärtsbits des Multibit-Schieberegisters **340** auszugeben. In diesem Fall fungiert das Multiplexerteil **330** als Schaltteil. Deshalb kann das Multiplexerteil **330** auch durch eine andere Vorrichtung ersetzt werden, die als Schaltteil fungiert. Der Einfachheit halber wird der Zwischenspeicher **350** nachstehend als vierter Zwischenspeicher bezeichnet.

[0029] Das Steuersignalgenerierungsteil **370** umfasst einen dritten Multiplexer **371**, um ein Anfangseinstellungssignal zu empfangen, auszuwählen und auszugeben und eine Bitlänge eines aus der PLA **360** eingegangenen Codeworts auszuwählen, das zum momentanen Takt decodiert wurde, einen fünften Zwischenspeicher **372**, der eine Summe der Bitlängen der Codewörter speichert, die bis zu einem Takt genau vor dem momentanen Takt decodiert wurden, einen Addierer **373**, um die Bitlänge aus dem dritten Multiplexer **371** und ein Ausgabedatenelement aus dem fünften Zwischenspeicher **372** zu ad-

dieren, um das Schiebesignal und ein Übertragungssignal zum Verschieben des Ausgabefensters des Multibit-Schieberegisters **340** auszugeben, einen sechsten Zwischenspeicher **374**, um das aus dem Addierer **373** eingegangene Übertragungssignal temporär zu speichern, und das gespeicherte Übertragungssignal beim nächsten Takt als Lesesignal auszugeben, und ein Multiplexsteuerteil **375**, um das aus dem Addierer **373** eingegangene Übertragungssignal und das aus dem sechsten Zwischenspeicher **324** eingegangene Lesesignal bei der Ausgabe des Multiplexsignals zu verwenden, um das Multiplexerteil **330** einen seiner Anschlüsse auswählen zu lassen. Hier ist die PLA **360**, bei der es sich um eine VLC-Tabelle handelt, eine Art Speicher. Dementsprechend kann die PLA **360** ein decodiertes Codewort und eine Bitlänge eines empfangenen Codeworts ausgeben. Deshalb kann die PLA durch eine andere Vorrichtung ersetzt werden, die als Speicher fungieren kann.

**[0030]** Der Funktionsablauf des Decoders für Codes mit variabler Länge nach der ersten Ausführungsform der vorliegenden Erfindung wird nachstehend im Einzelnen mit Bezug auf die [Fig. 4](#) und [Fig. 5](#) beschrieben. [Fig. 4](#) stellt einen Aufbau eines Bitstroms dar, der an den in [Fig. 3](#) gezeigten Eingabepuffer angelegt wird, und [Fig. 5](#) ist eine Tabelle zur Erläuterung des Funktionsablaufs des in [Fig. 3](#) gezeigten Decoders für Codes mit variabler Länge bei jedem Takt.

**[0031]** Codewörter mit variablen Längen, die an den Eingabepuffer **310** von außerhalb angelegt werden, werden von diesem 16 Bit-weise nach einer feststehenden Zeichentaktfrequenz ausgegeben. Die längste Bitlänge der Codewörter mit variabler Länge beträgt 16 Bits. Wie hinlänglich bekannt ist, hat der erste, zweite bzw. dritte Zwischenspeicher **321**, **322** und **323** im Zwischenspeicherteil **320**, bei dem es sich um ein Bit-Register handelt, eine Kapazität für das längste Codewort. Eine Eingabe von 32 Bits in das Multibit-Schieberegister **340** wird aus dem ersten und zweiten Multiplexer **331** und **332** im Multiplexerteil **330** bereitgestellt, und ein 16-Bit-Schiebedecodierfenster wird als Ausgabefenster des Multibit-Schieberegisters **340** bereitgestellt, dessen Verschiebung durch das Schiebesignal aus dem Steuersignalgenerierungsteil **370** gesteuert wird. Der erste Multiplexer **331** und der zweite Multiplexer **332** stellen dem Multibit-Schieberegister **340** 16 Vorwärtsbits bzw. 16 Rückwärtsbits zur Verfügung.

**[0032]** Das erneuerte 16 Bit-Codewort aus dem Multibit-Schieberegister **340** wird über den vierten Zwischenspeicher **350** an die PLA **360** angelegt, die eine Art Speicher ist. Wie in [Fig. 2](#) gezeigt ist, umfasst die PLA **360** eine Codeworttabelle, eine decodierte Codeworttabelle und eine Codewortlängentabelle. Die Codeworttabelle in der PLA **360** erfasst Bitmuster von darin gespeicherten Codewörtern für ein

Codewort, das einem Bitmuster entspricht, das identisch mit der Reihenfolge von Bits eines empfangenen Codeworts ist. Die decodierte Codeworttabelle in der PLA **360** gibt Codewörter mit einer feststehenden Länge aus, die den Codewörtern mit variablen Längen entsprechen, die in der Codeworttabelle erfasst werden, also decodierte Codewörter. Und die Codewortlängentabelle gibt Bitlängen der Codewörter mit variablen Längen aus, die in der Codeworttabelle erfasst sind. Bei der Generierung des nächsten Takts wird die Codewortlänge aus der Codewortlängentabelle in der PLA **360** zu den Bitlängen der Codewörter, die bis zu einem Takt genau vor dem Takt decodiert wurden, im Addierer **373** addiert, und dieses akkumulierte Signal wird als Schiebesignal verwendet, um das Ausgabefenster des Multibit-Schieberegisters **340** zu verschieben. Dieser Addierer generiert auch ein Übertragungssignal „1“, wenn die Summe der akkumulierten Codewörter länger ist als die längste Bitlänge des Codeworts, und „0“, wenn sie kürzer ist, und stellt es dem sechsten Zwischenspeicher **374** bereit. Wenn das Übertragungssignal aus dem Addierer **373** und das Lesesignal aus dem sechsten Zwischenspeicher **374** auf 0 (Tiefpegel) bzw. 0 (Tiefpegel) sind, stellt das Multiplexsteuerteil **375** dem ersten Multiplexer **331** und dem zweiten Multiplexer **332** ein Multiplexsignal bereit, so dass der erste Multiplexer **331** und der zweite Multiplexer **332** das Ausgabesignal aus dem zweiten Zwischenspeicher **322** bzw. das Ausgabesignal aus dem dritten Zwischenspeicher **323** selektiv ausgeben, wenn das Übertragungssignal und das Lesesignal auf 1 (Hochpegel) bzw. 0 (Tiefpegel) sind, oder auf 0 bzw. 1, das Multiplexsteuerteil **375** dem ersten Multiplexer **331** und dem zweiten Multiplexer **332** ein Multiplexsignal bereitstellen, so dass der erste Multiplexer **331** und der zweite Multiplexer **332** das Ausgabesignal aus dem ersten Zwischenspeicher **321** bzw. das Ausgabesignal aus dem zweiten Zwischenspeicher **322** selektiv ausgeben, und wenn das Übertragungssignal und das Lesesignal auf 1 (Hochpegel) bzw. 1 (Hochpegel) sind, das Multiplexsteuerteil **375** dem ersten Multiplexer **331** und dem zweiten Multiplexer **332** ein Multiplexsignal bereitstellen, so dass der erste Multiplexer **331** und der zweite Multiplexer **332** selektiv das Ausgabesignal aus dem Eingabepuffer **321** bzw. das Ausgabesignal aus dem ersten Zwischenspeicher **321** ausgeben.

**[0033]** Beispielsweise angenommen, ein Codewort mit dem Datenstrom von [Fig. 4](#) wird an den in [Fig. 3](#) gezeigten Eingabepuffer angelegt. Bei der Generierung eines ersten Taktsignals werden beliebige 16 Bits zwangsweise an den dritten Multiplexer **371** im Steuersignalgenerierungsteil **370** als Anfangseinstellungssignal angelegt, um alle Übertragungssignale und Lesesignale auf „Hochpegel“ gehen zu lassen. Zur gleichen Zeit gibt der Eingabepuffer **310** 16 Bits  $a_1-a_8b_1-b_6c_1-c_2$  aus, die das erste Bit des empfangenen Codeworts enthalten. „1“ bedeutet hier Hochpegel und „0“ Tiefpegel. Da alle Übertrag- und Lesesig-

nale im Ansprechen auf ein Multiplexsignal aus dem Multiplexsteuerteil **375** auf 1, 1 sind, wählt in dieser Zeit der zweite Multiplexer **332** ein Codewort aus dem ersten Zwischenspeicher **321** aus, und der erste Multiplexer **331** wählt ein Codewort aus dem Eingabepuffer **310** aus. Das heißt, der erste Multiplexer **331** wählt ein Ausgabesignal  $a_1-a_8b_1-b_6c_1-c_2$  aus und gibt es aus, und das Multibit-Schieberegister **340**, welches 32 Bits erhält, verschiebt sein Ausgabefenster, ein Schiebedecodierfenster, um 16 Bits. In diesem Fall wird kein Signal vom Multibit-Schieberegister **340** ausgegeben.

**[0034]** Bei der Generierung eines zweiten Taktsignals werden beliebige 16 Bits zwangsweise an den dritten Multiplexer **371** im Steuersignalgenerierungsteil **370** als Anfangseinstellungssignal an das Übertragungssignal „1“ angelegt und lassen das Lesesignal durch das im Addierer **373** beim ersten Takt generierte Übertragungssignal auf „1“ gehen, was dazu führt, dass der Eingabepuffer **310** ein Codewort  $c_3-c_5d_1-d_{13}$  ausgibt. Aufgrund dessen werden die Daten  $a_1-a_8b_1-b_6c_1-c_2$ , die beim ersten Takt vom Eingabepuffer **310** ausgegeben werden, durch das Lesesignal im zweiten Zwischenspeicher **322** gespeichert, da alle Übertrag- und Lesesignale auf 1 und 1 sind, der zweite Multiplexer **332** gibt selektiv ein Signal  $a_1-a_8b_1-b_6c_1-c_2$  im Ansprechen auf das Multiplexsignal vom Multiplexsteuerteil **375** aus dem ersten Zwischenspeicher **321** aus, und der erste Multiplexer **331** gibt selektiv ein Signal  $c_3-c_5d_1-d_{13}$  aus. Und das Multibit-Schieberegister **340**, das die Signale aus dem ersten Zwischenspeicher **321** und dem Eingabepuffer **310** erhalten hat, verschiebt sein Schiebedecodierfenster aus dem momentanen Zustand um 16 Bits und überträgt das Codewort  $a_1-a_8b_1-b_6c_1-c_2$  vom ersten Zwischenspeicher **321** zum vierten Zwischenspeicher **350**.

**[0035]** Bei der Generierung eines dritten Taktsignals wird das Lesesignal durch das Übertragungssignal beim vorherigen zweiten Takt zu „1“, das Übertragungssignal wird zu „0“, weil das zu decodierende Signal  $a_1-a_8$  tatsächlich nur aus 8 Bits besteht, und der Eingabepuffer **310** gibt den nächsten Signalbitstrom  $d_{14}d_{15}e_1-e_{12}f_1f_2$  aus. Da das Lesesignal beim vorherigen zweiten Taktsignal auf „1“ ist, speichert der Zwischenspeicher **321** im Zwischenspeicherteil **320** ein Datenelement  $c_3-c_5d_1-d_{13}$  aus dem Eingabepuffer **310** beim vorherigen zweiten Takt, der zweite Zwischenspeicher **322** speichert ein Datenelement  $a_1-a_8b_1-b_6c_1-c_2$  aus dem ersten Zwischenspeicher **321** beim vorherigen zweiten Takt, und der vierte Zwischenspeicher **350** speichert ein Datenelement  $a_1-a_8b_1-b_6c_1-c_2$  aus dem Multibit-Schieberegister **340** beim vorherigen zweiten Takt. Da in diesem Moment das Übertrag- und Lesesignal auf 1 bzw. 0 sind, wählen der zweite Multiplexer **332** und der erste Multiplexer **331** das Datenelement  $a_1-a_8b_1-b_6c_1-c_2$  aus dem zweiten Zwischenspeicher **322** bzw. das Datenele-

ment  $c_3-c_5d_1-d_{13}$  aus dem ersten Zwischenspeicher **321** aus und legen es an die Eingabeseite des Multibit-Schieberegisters **340** an. Beim Eingang eines Signals (oder Datenelements) aus dem ersten und zweiten Multiplexer **331** und **332** verschiebt das Multibit-Schieberegister **340** sein Schiebedecodierfenster um die Länge  $a_1-a_8$  (8 Bits) des beim vorherigen zweiten Takt decodierten Codeworts und gibt im Ansprechen auf das Schiebesignal aus dem Steuersignalgenerierungsteil **370** ein Datenelement  $b_1-b_6c_1-c_5d_1-d_5$  aus. Während der Decoder für Codes mit variabler Länge in Betrieb ist, werden die zuvor erwähnten Schritte ständig wiederholt.

**[0036]** Nachstehend wird ein Decoder für Codes mit variabler Länge nach einer zweiten Ausführungsform der vorliegenden Erfindung erläutert. In der zweiten Ausführungsform wird anstelle des im Stande der Technik verwendeten zweiten Multibit-Schieberegisters auch ein Multiplexerteil verwendet, das über eine Schaltfunktion verfügt.

**[0037]** [Fig. 6](#) stellt ein Blockschema dar, das den Decoder für Codes mit variabler Länge nach der zweiten Ausführungsform der vorliegenden Erfindung zeigt.

**[0038]** Mit Bezug auf [Fig. 6](#) umfasst der Decoder für Codes mit variabler Länge nach der zweiten Ausführungsform der vorliegenden Erfindung einen Eingabepuffer **610**, um aus einem seriell an ihn angelegten Codewort parallele Codewörter mit 16 Bits zu machen und die parallelen Codewörter im Ansprechen auf ein Lesesignal auszugeben, ein Zwischenspeicherteil **620**, um die aus dem Eingabepuffer **610** erhaltenen Codewörter im Ansprechen auf das Lesesignal temporär zu speichern und auszugeben, ein Multiplexerteil **630**, um die Codewörter aus dem Zwischenspeicherteil **620** und die Codewörter aus dem Eingabepuffer **610** im Ansprechen auf ein Multiplexsignal auszuwählen und auszugeben, ein Multibit-Schieberegister **640**, um ein Schiebeausgabedecodierfenster im Ansprechen auf ein Schiebesignal um die Längen von Codewörtern zu verschieben, die bis zu einem früheren Taktsignal decodiert wurden, und eine vorbestimmte Anzahl von Bits (in diesem Falle 16 Bits) der Codewörter aus dem Multiplexerteil **630** auszugeben, eine PLA (programmierbare Logikanordnung) **650**, um ein decodiertes Codewort und eine Bitlänge des decodierten Codeworts, das dem Codewort entspricht, aus dem Multibit-Schieberegister **640** auszugeben, und ein Steuersignalgenerierungsteil **660**, um die Bitlängen der bis zu einem Takt genau vor dem momentanen Takt decodierten Codewörter und eine Bitlänge des beim momentanen Takt decodierten Codeworts, das von der PLA **650** her eingegangen ist, zu addieren, und die addierten Bitlängen bei der Bereitstellung des Lesesignals, des Multiplexsignals und des Schiebesignals zu verwenden.

**[0039]** Das Zwischenspeicherteil **620** umfasst einen ersten Zwischenspeicher **621**, um die Codewörter aus dem Eingabepuffer **610** im Ansprechen auf das Lesesignal temporär zu speichern und auszugeben, und ein zweites Zwischenspeicherteil **622**, um die Codewörter aus dem ersten Zwischenspeicher **621** im Ansprechen auf das Lesesignal temporär zu speichern und auszugeben. Das Multiplexerteil **630** umfasst einen ersten Multiplexer **631**, um eines der Codewörter aus dem Eingabepuffer **610** und dem ersten Zwischenspeicher **621** im Zwischenspeicherteil **620** im Ansprechen auf das Multiplexersignal auszuwählen und das ausgewählte Codewort als 16 Rückwärtsbits des Multibit-Schieberegisters **640** auszugeben, und einen zweiten Multiplexer **632**, um eines der Codewörter aus dem ersten und zweiten Zwischenspeicher **621** und **622** im Zwischenspeicherteil **620** im Ansprechen auf das Multiplexersignal auszuwählen und das ausgewählte Codewort als 16 Vorwärtsbits des Multibit-Schieberegisters **640** auszugeben.

**[0040]** Das Steuersignalgenerierungsteil **660** umfasst einen dritten Multiplexer **661**, um ein Anfangseinstellungssignal zu empfangen und auszugeben und eine Bitlänge eines zum momentanen Takt decodierten Codeworts aus der PLA **650** auszuwählen, einen dritten Zwischenspeicher **662**, der eine Summe der Bitlängen der bis zu einem Takt genau vor dem momentanen Takt decodierten Codewörter speichert, einen Addierer **663**, um die Bitlänge eines zum momentanen Zeitpunkt decodierten, vom dritten Multiplexer **661** her eingegangenen Codeworts und Bitlängen von bis zu einem Takt genau vor dem momentanen decodierten, aus dem dritten Zwischenspeicher **662** her eingegangenen Codewörtern zu addieren, um das Schiebesignal und ein Übertragungssignal auszugeben, einen vierten Zwischenspeicher **664**, um das vom Addierer **663** her eingegangene Übertragungssignal temporär zu speichern und das gespeicherte Übertragungssignal bei einem auf den momentanen Takt folgenden Takt als Lesesignal auszugeben, und ein Multiplexerteil **665**, um das aus dem vierten Zwischenspeicher **664** her eingegangene Übertragungssignal bei der Ausgabe des Multiplexersignals zur Auswahl eines der Eingangsanschlüsse des Multiplexerteils **630** zu verwenden. Bei dem Schiebesignal handelt es sich um ein Signal, um das Decodierausgabefenster des Multibit-Schieberegisters **640** um eine vorbestimmte Anzahl von Bits zu verschieben, und bei dem Übertragungssignal handelt es sich um ein Signal „1“ oder „0“, das generiert wird, wenn die Ausgabe des Addierers **663** größer oder kleiner als die längste Bitlänge ist.

**[0041]** Nun wird der Funktionsablauf des Decoders für Codes mit variabler Länge nach der zweiten Ausführungsform der vorliegenden Erfindung erläutert.

**[0042]** Beim seriellen Empfang von Codewörtern mit variablen Längen stellt der Eingabepuffer **610** die

Codewörter dem ersten Zwischenspeicher **621** parallel in Form von 16 Bits mit einer konstanten Zeichengeschwindigkeit im Ansprechen auf ein Lesesignal bereit. Die 32 Bits werden durch den ersten und zweiten Multiplexer **631** und **632** im Multiplexerteil **630** an das Multibit-Schieberegister **640** angelegt. Bei dem Ausgabefenster des Multibit-Schieberegisters **640** handelt es sich um ein Schiebedecodierfenster für 16 Bits. Je nachdem, ob das Lesesignal aus dem vierten Zwischenspeicher **664** „0“ oder „1“ lautet, wählt das Multiplexerteil **665** einen der beiden Eingangsanschlüsse des ersten und zweiten Multiplexers **631** und **632** aus. „0“ und „1“ stellen einen niedrigen bzw. einen hohen Wert dar. Entsprechend geben der erste Multiplexer **631** und der zweite Multiplexer **632** ein Codewort aus dem Eingabepuffer **610** bzw. ein Codewort aus dem ersten Zwischenspeicher **621** im Ansprechen auf ein Multiplexersignal aus dem Multiplexerteil **665** aus, wenn das Lesesignal „1“ lautet, und geben ein Codewort aus dem ersten Zwischenspeicher **621** bzw. ein Codewort aus dem zweiten Zwischenspeicher **622** aus, wenn das Lesesignal „0“ lautet. Die erneuerten 16 Bits aus dem Multibit-Schieberegister **640** werden mit der PLA **650** verbunden, bei der es sich um eine VLC-Tabelle handelt.

**[0043]** Die PLA **650**, die eine Art Speicher ist, kann durch irgendeine andere Vorrichtung ersetzt werden, die über eine Speicherfunktion verfügt. Das System und der Funktionsablauf der PLA **650** stimmen mit der in [Fig. 3](#) gezeigten PLA **360** überein. Das heißt, die PLA **650** gibt ein decodiertes Codewort aus, das den vom Multibit-Schieberegister **640** her eingegangenen 16 Bits entspricht, und gibt eine Länge des decodierten Codeworts an den dritten Multiplexer **661** im Steuersignalgenerierungsteil **660** aus.

**[0044]** Bei der Generierung des nächsten Taktsignals werden die Länge des zum momentanen Takt decodierten, aus der PLA **640** her eingegangenen Codeworts und die Bitwerte der bis zu einem Takt genau vor dem momentanen Takt (bei dem es sich um ein Signal vom dritten Zwischenspeicher **662** handelt) decodierten Codewörter im Addierer **663** addiert und als Schiebesignal verwendet, um das Schiebesignalausgabefenster, bei dem es sich um das Ausgabefenster des Multibit-Schieberegisters **640** handelt, bei einem auf den momentanen Takt folgenden Takt zu verschieben. Dabei gibt der Addierer **663** „0“ aus, wenn der Additionswert kleiner ist als ein Bitlängenwert eines voreingestellten längsten Codeworts, und gibt „1“ aus, wenn der Additionswert größer ist als ein Bitlängenwert des voreingestellten längsten Codeworts.

**[0045]** Die zweite Ausführungsform des Decoders für Codes mit variabler Länge ist ein Modell, das sich für einen Fall eignet, bei dem im Decoder für Codes mit variabler Länge keine hohe Geschwindigkeit benötigt wird. Wie in [Fig. 6](#) gezeigt ist, ist der Zwischenspeicher **350**, der in [Fig. 3](#) enthalten ist, hier ausge-

geschlossen, um Systemen angepasst zu werden, die keine hohe Geschwindigkeit im Decoder benötigen.

**[0046]** [Fig. 7](#) stellt ein Blockschema dar, das einen Decoder für Codes mit variabler Länge nach einer dritten Ausführungsform der vorliegenden Erfindung zeigt.

**[0047]** Mit Bezug auf [Fig. 7](#) umfasst der Decoder für Codes mit variabler Länge nach der dritten Ausführungsform der vorliegenden Erfindung einen Eingabepuffer **710**, um ein seriell an ihn angelegtes Codewort in parallele Codewörter mit 16 Bits umzuwandeln und die parallelen Codewörter im Ansprechen auf ein Lesesignal auszugeben, ein Zwischenspeicherteil **720**, um die vom Eingabepuffer **710** her eingegangenen Codewörter im Ansprechen auf das Lesesignal temporär zu speichern und auszugeben, ein Multiplexerteil **730**, um die Codewörter aus dem Zwischenspeicherteil **720** im Ansprechen auf ein Multiplexersignal auszuwählen und auszugeben, ein Multibit-Schieberegister **740**, um ein Schiebedecodierfenster (oder ein Ausgabefenster) im Ansprechen auf ein Schiebesignal zu verschieben und eine vorbestimmte Anzahl von Bits des Codeworts aus dem Multiplexerteil **730** auszugeben, einen Zwischenspeicher **750**, um das Codewort aus dem Multibit-Schieberegister **740** temporär zu speichern und auszugeben, eine PLA (programmierbare Logikeinheit) **760**, um ein decodiertes Codewort und eine Bitlänge des decodierten Codeworts auszugeben, die dem Codewort aus dem Zwischenspeicher **750** entspricht, ein Steuersignalgenerierungsteil **760**, um eine Summe von Bitlängen der genau bis zu einem Takt vor dem momentanen Takt decodierten Codewörter und eine Bitlänge eines zum momentanen Takt decodierten, von der PLA **760** her eingegangenen Codeworts zu addieren und die addierten Bitlängen bei der Bereitstellung des Lesesignals, des Multiplexersignals und des Übertragungssignals zu verwenden, und einen Ausgabepuffer **780**.

**[0048]** Das Zwischenspeicherteil **720** umfasst einen ersten Zwischenspeicher **721**, um die Codewörter aus dem Eingabepuffer **710** im Ansprechen auf das Lesesignal temporär zu speichern und auszugeben, und einen zweiten Zwischenspeicher **722**, um die Codewörter aus dem ersten Zwischenspeicher **721** im Ansprechen auf das Lesesignal temporär zu speichern und auszugeben. Das Multiplexerteil **730** umfasst einen ersten Multiplexer **731**, um eines der Codewörter aus dem Eingabepuffer **710** und dem ersten Zwischenspeicher **721** im Zwischenspeicherteil **720** im Ansprechen auf das Multiplexersignal auszuwählen und das ausgewählte Codewort als 16 Rückwärtsbits des Multibit-Schieberegisters **740** auszugeben, und einen zweiten Multiplexer **732**, um eines der Codewörter aus dem ersten und zweiten Zwischenspeicher **721** und **722** im Zwischenspeicherteil **720** im Ansprechen auf das Multiplexersignal auszugeben und

das ausgewählte Codewort als 16 Vorwärtsbits des Multibit-Schieberegisters **740** auszugeben. Der Einfachheit halber wird der Zwischenspeicher **750** als dritter Zwischenspeicher bezeichnet.

**[0049]** Das Steuersignalgenerierungsteil **770** umfasst einen dritten Multiplexer **771**, um ein Anfangseinstellungssignal zu empfangen und auszugeben und eine Bitlänge eines Codeworts zu empfangen und auszugeben, das zum momentanen Takt in der PLA **760** decodiert wurde, einen vierten Zwischenspeicher **772**, der eine Summe der Bitlängen der bis zu einem Takt genau vor dem momentanen Takt decodierten Codewörter speichert, einen Addierer **773**, um das vom dritten Multiplexer **771** her eingegangene Codewort und die vom vierten Zwischenspeicher **772** her eingegangenen Codewörter zu addieren, um das Schiebesignal zum Verschieben des Ausgabefensters (oder Schiebedecodierfensters) des Multibit-Schieberegisters **740** zu erzeugen und „0“ auszugeben, wenn der Additionswert größer ist als eine voreingestellte längste Bitlänge eines Codeworts, und „1“ auszugeben, wenn der Additionswert kleiner ist als die voreingestellte längste Bitlänge eines Codeworts, und ein Multiplexesteuerteil **775**, um das vom Addierer **773** her eingegangene Lesesignal bei der Ausgabe des Multiplexersignals zu verwenden, um einen der Eingänge des ersten und zweiten Multiplexers **731** und **732** auszuwählen.

**[0050]** Die dritte Ausführungsform des Decoders für Codes mit variabler Länge mit dem vorstehend erwähnten System wird dann verwendet, wenn Schaltungen, die den Decoder umgeben und mit Ausgangsseiten des Decoders verbunden sind, um das Lesesignal anzulegen, nicht kompliziert sind.

**[0051]** Der Funktionsablauf des Decoders für Codes mit variabler Länge nach der dritten Ausführungsform der vorliegenden Erfindung wird nun erläutert.

**[0052]** Der Eingabepuffer **710** gibt seriell eingegangene, variabel lange Codewörter anhand von 16 Bits mit einer konstanten Zeichengeschwindigkeit im Ansprechen auf ein Lesesignal parallel aus. Die 32 Bits aus dem ersten und zweiten Multiplexer **731** und **732** im Multiplexerteil **730** werden an eine Eingangsseite des Multibit-Schieberegisters **740** angelegt. Bei dem Ausgabefenster des Multibit-Schieberegisters **740** handelt es sich um eine Schiebedecodierfenster für 16 Bits. Wenn das Lesesignal „1“ lautet, geben der erste Multiplexer **731** und der zweite Multiplexer **732** das Codewort aus dem Eingabepuffer **710** bzw. das Codewort aus dem ersten Zwischenspeicher **721** im Ansprechen auf das Multiplexersignal aus, das vom Multiplexesteuerteil **774** bereitgestellt wird, und wenn das Lesesignal „0“ lautet, geben der erste Multiplexer **731** und der zweite Multiplexer **732** das Codewort aus dem ersten Zwischenspeicher **721** bzw. das Co-

dewort aus dem zweiten Zwischenspeicher **722** im Ansprechen auf das Multiplexsignal aus, das vom Multiplexsteuerteil **774** bereitgestellt wird. Die Ausgangsseite des Multibit-Schieberegisters **740** ist an den dritten Zwischenspeicher **750** angeschlossen, und die Ausgangsseite des dritten Zwischenspeichers **750** ist mit der PLA **760** verbunden, bei der es sich um eine VLC-Tabelle handelt. Da die PLA **760** eine Art Speicher ist, kann sie durch irgendeine andere Vorrichtung ersetzt werden, die über eine Speicherfunktion verfügt.

**[0053]** System und Funktionsablauf der PLA **760** stimmen mit der in [Fig. 3](#) gezeigten PLA **360** überein. Das heißt, die PLA **760** gibt ein decodiertes Codewort und eine Bitlänge des decodierten Codeworts aus, die einem empfangenen Codewort entsprechen. Bei der Generierung des nächsten Signals werden die Bitlänge des zum momentanen Takt decodierten, von der PLA **760** her eingegangenen Codeworts und Werte der bis zu einem Takt genau vor dem momentanen Takt decodierten Codewörter, die im vierten Zwischenspeicher **772** gespeichert sind, im Addierer **773** addiert und direkt verwendet, um das Schiebende- codierausgabefenster am Multibit-Schieberegister **740** zu verschieben. Und je nach dem Status des Lesesignals (oder Übertragungssignals) stellt das Multiplexsteuerteil **774** das Multiplexsignal bereit, um einen der beiden Eingangsanschlüsse des ersten Multiplexers **731** und des zweiten Multiplexers **732** auszuwählen.

**[0054]** Wie erläutert wurde, hat der die Erfindung verkörpernde Decoder für Codes mit variabler Länge gegenüber dem Decoder aus dem Stand der Technik die folgenden Vorteile:

Erstens kann die Verwendung der Multiplexer anstelle des Multibit-Schieberegisters die Fläche einer integrierten Schaltung auf derselben Verzögerungsbasis reduzieren.

**[0055]** Zweitens ermöglicht es die Verkürzung der Verzögerungszeit, die zwischen einem Decoder für Codes mit variabler Länge und den ihn umgebenden Schaltungen besteht, einen Funktionsblock mit höherer Betriebsgeschwindigkeit zu implementieren.

**[0056]** Dem Fachmann auf dem Gebiet wird klar sein, dass verschiedene Abwandlungen und Veränderungen an dem beschriebenen Decoder für Codes mit variabler Länge vorgenommen werden können, ohne dass dabei vom Rahmen der Erfindung abgewichen würde. Somit soll die vorliegende Erfindung die Abwandlungen und Veränderungen dieser Erfindung abdecken, vorausgesetzt, sie fallen in den Rahmen der beigefügten Ansprüche und ihrer Entsprechungen.

## Patentansprüche

1. Decoder für Codes mit variabler Länge, Folgendes umfassend:  
 ein erstes Speicherteil, um ein eingegangenes Codewort im Ansprechen auf ein Lesesignal temporär zu speichern und auszugeben;  
 ein Schaltteil, um die Codewörter aus dem ersten Speicherteil und das am ersten Speicherteil eingegangene Codewort im Ansprechen auf ein Schaltsignal selektiv auszugeben;  
 ein Multibit-Schieberegister (Barrel Shifter), um ein Decodierausgabefenster daran im Ansprechen auf ein Schiebesignal beim Ausgeben einer vorbestimmten Anzahl von Bits des Codeworts aus dem Schaltteil zum Decodieren zu verschieben;  
 eine Tabelle, um ein decodiertes Codewort und eine Bitlänge des decodierten Codeworts aus der vorbestimmten Anzahl von Bits, die vom Multibit-Schieberegister ausgegeben wurden, auszugeben; und  
 ein Steuersignalgenerierungsteil, um die Bitlänge eines aus der Tabelle erhaltenen Codeworts mit dem momentanen Takt beim Generieren des Schaltsignals, des Schiebesignals und des Lesesignals zu verwenden.

2. Decoder nach Anspruch 1, für Codes mit variabler Länge, darüber hinaus umfassend:  
 ein zweites Speicherteil, um das Codewort aus dem Multibit-Schieberegister, die das decodierte Codewort ausgebende Tabelle und die Bitlänge aus der vorbestimmten Anzahl von Bits, die im zweiten Speicherteil gespeichert sind, temporär zu speichern und auszugeben.

3. Decoder nach Anspruch 1 oder 2, für Codes mit variabler Länge, wobei das Schaltteil mindestens einen Multiplexer umfasst.

4. Decoder nach Anspruch 1 oder 2, für Codes mit variabler Länge, darüber hinaus umfassend:  
 einen Eingabepuffer, um ein serielles Codewort zu empfangen und das Codewort in Paralleleinheiten aus einer vorbestimmten Anzahl von Bits an das erste Speicherteil anzulegen; und  
 einen Ausgabepuffer, um ein aus der Tabelle erhaltenes decodiertes Codewort aus dem Decoder für Codes mit variabler Länge auszugeben.

5. Decoder nach Anspruch 1 oder 2, für Codes mit variabler Länge, wobei das erste Speicherteil mindestens einen Zwischenspeicher umfasst.

6. Decoder nach Anspruch 1 oder 2, für Codes mit variabler Länge, darüber hinaus einen Zwischenspeicher umfassend, der zwischen dem Multibit-Schieberegister und der Tabelle angeschlossen ist.

7. Decoder nach Anspruch 2, für Codes mit variabler Länge, wobei das erste Speicherteil umfasst: einen ersten Zwischenspeicher, um ein eingegangenes Codewort im Ansprechen auf das Lesesignal temporär zu speichern und auszugeben; einen zweiten Zwischenspeicher, um das aus dem ersten Zwischenspeicher eingegangene Codewort im Ansprechen auf das Lesesignal temporär zu speichern und auszugeben; und einen dritten Zwischenspeicher, um aus dem zweiten Zwischenspeicher eingegangene Daten im Ansprechen auf das Lesesignal temporär zu speichern und auszugeben; wobei das Schaltteil umfasst: einen ersten Multiplexer, um die Codewörter aus dem ersten und zweiten Zwischenspeicher und das am ersten Zwischenspeicher eingegangene Codewort zu empfangen, und eines der Codewörter im Ansprechen auf ein Multiplexersignal auszuwählen und auszugeben; und einen zweiten Multiplexer, um die Codewörter aus dem ersten, zweiten und dritten Zwischenspeicher zu empfangen und eines der Codewörter aus dem ersten, zweiten und dritten Zwischenspeicher im Ansprechen auf das Multiplexersignal auszuwählen und auszugeben; wobei das Multibit-Schieberegister das Codewort aus dem ersten Multiplexer als Vorwärtsbits und das Codewort aus dem zweiten Multiplexer als Rückwärtsbits empfängt; wobei das zweite Speicherteil einen vierten Zwischenspeicher umfasst, um das Codewort aus dem Multibit-Schieberegister temporär zu speichern und auszugeben; wobei die Tabelle ein decodiertes Codewort und eine Bitlänge des decodierten Codeworts entsprechend dem Codewort aus dem vierten Zwischenspeicher ausgibt; wobei der Decoder für Codes mit variabler Länge umfasst: einen dritten Multiplexer, um eine der Bitlängen aus der Tabelle und ein Anfangseinstellungssignal selektiv auszugeben; einen fünften Zwischenspeicher, um Bitlängen der Codewörter, die bis zu einem Takt genau vor dem momentanen Takt decodiert wurden, zu speichern; einen Addierer, um die Bitlänge aus dem dritten Multiplexer und die Bitlängen aus dem fünften Zwischenspeicher zu addieren und den Additionswert bei der Ausgabe des Schiebesignals und eines Übertragungssignals zu verwenden; einen sechsten Zwischenspeicher, um das Übertragungssignal temporär zu speichern und das gespeicherte Übertragungssignal als Lesesignal auszugeben; und ein Multiplexersignalgenerierungsteil, um das Übertragungssignal und das Lesesignal bei der Ausgabe des Multiplexersignals zu verwenden.

8. Decoder nach Anspruch 7, für Codes mit variabler Länge, wobei der erste Multiplexer und der

zweite Multiplexer das Codewort aus dem zweiten Zwischenspeicher bzw. das Codewort aus dem dritten Zwischenspeicher im Ansprechen auf das Multiplexersignal ausgeben, wenn sich das Übertragungssignal und das Lesesignal jeweils auf Tiefpegel befinden, das Codewort aus dem ersten Zwischenspeicher bzw. das Codewort aus dem zweiten Zwischenspeicher im Ansprechen auf das Multiplexersignal ausgeben, wenn sich das Übertragungssignal und das Lesesignal auf Hoch- bzw. Tiefpegel oder Tief- bzw. Hochpegel befinden, und das Codewort aus dem ersten Zwischenspeicher bzw. das am ersten Zwischenspeicher eingegangene Codewort im Ansprechen auf das Multiplexersignal ausgeben, wenn sich das Übertragungssignal und das Lesesignal jeweils auf Hochpegel befinden.

9. Decoder nach Anspruch 7, für Codes mit variabler Länge, wobei der dritte Multiplexer das Einstellsignal ausgibt, wenn nur das Einstellsignal in einem Anfangsstadium eingeht, und die Bitlänge des vorliegenden Codeworts ausgibt, wenn die Bitlänge aus der Tabelle erhalten wird.

10. Decoder nach Anspruch 7, für Codes mit variabler Länge, darüber hinaus umfassend: einen Eingabepuffer, um das seriell eingegangene Codewort an den ersten Zwischenspeicher im Ansprechen auf das Lesesignal parallel auszugeben; und einen Ausgabepuffer, um das decodierte Codewort aus der Tabelle aus dem Decoder für Codes mit variabler Länge auszugeben.

11. Decoder nach Anspruch 7, für Codes mit variabler Länge, wobei sich das Übertragungssignal auf Tiefpegel befindet, wenn der Additionswert kleiner ist als eine voreingestellte längste Bitlänge der Codewörter, und sich auf Hochpegel befindet, wenn der Additionswert größer ist als die voreingestellte längste Bitlänge der Codewörter.

12. Decoder nach Anspruch 1, für Codes mit variabler Länge, wobei das erste Speicherteil umfasst: einen ersten Zwischenspeicher, um ein eingegangenes Codewort im Ansprechen auf das Lesesignal temporär zu speichern und auszugeben; und einen zweiten Zwischenspeicher, um das aus dem ersten Zwischenspeicher eingegangene Codewort im Ansprechen auf das Lesesignal temporär zu speichern und auszugeben; wobei das Schaltteil umfasst: einen ersten Multiplexer, um das Codewort aus dem ersten Zwischenspeicher und das am ersten Zwischenspeicher eingegangene Codewort zu empfangen, und eines der Codewörter im Ansprechen auf ein Multiplexersignal auszuwählen und auszugeben; und einen zweiten Multiplexer, um die Codewörter aus

dem ersten und zweiten Zwischenspeicher zu empfangen und eines der Codewörter aus dem ersten und zweiten Zwischenspeicher im Ansprechen auf das Multiplexsignal auszuwählen und auszugeben; wobei das Multibit-Schieberegister das Codewort aus dem ersten Multiplexer als Vorwärtsbits und das Codewort aus dem zweiten Multiplexer als Rückwärtsbits empfängt; wobei der Decoder für Codes mit variabler Länge umfasst:

- einen dritten Multiplexer, um eine der Bitlängen aus der Tabelle und ein Anfangseinstellungssignal selektiv auszugeben;
- einen dritten Zwischenspeicher, um die Bitlängen der Codewörter zu speichern, die bis zu einem Takt vor dem momentanen Takt decodiert wurden;
- einen Addierer, um die Bitlänge aus dem dritten Multiplexer und die Bitlängen aus dem dritten Zwischenspeicher zu addieren, um den Additionswert als Schiebesignal und ein Übertragungssignal auszugeben;
- einen vierten Zwischenspeicher, um das Übertragungssignal temporär zu speichern und das gespeicherte Übertragungssignal als Lesesignal auszugeben; und ein Multiplexsignalgenerierungsteil, um das Lesesignal aus dem vierten Zwischenspeicher bei der Ausgabe des Multiplexsignals zu verwenden.

13. Decoder nach Anspruch 12, für Codes mit variabler Länge, wobei der erste Multiplexer und der zweite Multiplexer das Codewort aus dem ersten Zwischenspeicher bzw. das am ersten Zwischenspeicher eingegangene Codewort im Ansprechen auf das Multiplexsignal ausgeben, wenn sich das Lesesignal auf Hochpegel befindet, und ein Signal aus dem ersten Zwischenspeicher bzw. ein Signal aus dem zweiten Zwischenspeicher im Ansprechen auf das Multiplexsignal ausgeben, wenn sich das Lesesignal auf Tiefpegel befindet.

14. Decoder nach Anspruch 12, für Codes mit variabler Länge, wobei der dritte Multiplexer das Anfangseinstellungssignal ausgibt, wenn nur das Anfangseinstellungssignal in einem Anfangsstadium eingeht, und die Bitlänge ausgibt, wenn die Bitlänge aus der Tabelle erhalten wird.

15. Decoder nach Anspruch 12, für Codes mit variabler Länge, darüber hinaus umfassend:

- einen Eingabepuffer, um das seriell eingegangene Codewort an den ersten Zwischenspeicher im Ansprechen auf das Lesesignal durch eine vorbestimmte Anzahl von Bits parallel auszugeben; und
- einen Ausgabepuffer, um das decodierte Wort aus der Tabelle aus dem Decoder für Codes mit variabler Länge auszugeben.

16. Decoder nach Anspruch 12, für Codes mit variabler Länge, wobei sich das Übertragungssignal auf Tiefpegel befindet, wenn der Additionswert kleiner ist als eine voreingestellte längste Bitlänge, und sich auf

Hochpegel befindet, wenn der Additionswert größer ist als die voreingestellte längste Bitlänge.

17. Decoder nach Anspruch 2, für Codes mit variabler Länge, wobei das erste Speicherteil umfasst:

- einen ersten Zwischenspeicher, um ein erhaltenes Codewort im Ansprechen auf das Lesesignal temporär zu speichern und auszugeben; und
- einen zweiten Zwischenspeicher, um das aus dem ersten Zwischenspeicher eingegangene Codewort im Ansprechen auf das Lesesignal temporär zu speichern und auszugeben;

wobei das Schaltteil umfasst:

- einen ersten Multiplexer, um das Codewort aus dem ersten Zwischenspeicher und das am ersten Zwischenspeicher eingegangene Codewort zu empfangen, und eines der Codewörter im Ansprechen auf ein Multiplexsignal auszuwählen und auszugeben; und
- einen zweiten Multiplexer, um die Codewörter aus dem ersten und zweiten Zwischenspeicher zu empfangen und eines der Codewörter aus dem ersten und zweiten Zwischenspeicher im Ansprechen auf das Multiplexsignal auszuwählen und auszugeben; wobei das Multibit-Schieberegister das Codewort aus dem ersten Multiplexer als Vorwärtsbits und das Codewort aus dem zweiten Multiplexer als Rückwärtsbits empfängt;

wobei das zweite Speicherteil einen dritten Zwischenspeicher umfasst, um das Codewort aus dem Multibit-Schieberegister temporär zu speichern und auszugeben;

wobei die Tabelle ein decodiertes Codewort und eine Bitlänge des decodierten Codeworts des Codeworts aus dem dritten Zwischenspeicher im Ansprechen auf das Codewort aus dem dritten Zwischenspeicher ausgibt;

wobei der Decoder für Codes mit variabler Länge umfasst:

- einen dritten Multiplexer, um eine der Bitlängen aus der Tabelle und ein Anfangseinstellungssignal auszugeben;
- einen vierten Zwischenspeicher, um eine Summe von Bitlängen der Codewörter, die bis zu einem Takt vor dem momentanen Takt decodiert wurden, zu speichern;
- einen Addierer, um das Anfangseinstellungssignal oder die Bitlänge aus dem dritten Multiplexer und die Bitlängen aus dem vierten Zwischenspeicher zu addieren, um das Schiebesignal und ein Übertragungssignal als Lesesignal auszugeben; und
- ein Multiplexsignalgenerierungsteil, um das Übertragungssignal bei der Ausgabe des Multiplexsignals zu verwenden.

18. Decoder nach Anspruch 17, für Codes mit variabler Länge, wobei der erste Multiplexer und der zweite Multiplexer das Codewort aus dem ersten Zwischenspeicher bzw. das am ersten Zwischenspeicher eingegangene Codewort im Ansprechen auf das

Multiplexsignal ausgeben, wenn sich das Lesesignal auf Hochpegel befindet, und das Codewort aus dem ersten Zwischenspeicher bzw. das Codewort aus dem zweiten Zwischenspeicher im Ansprechen auf das Multiplexsignal ausgeben, wenn sich das Lesesignal auf Tiefpegel befindet.

19. Decoder nach einem der Ansprüche 1, 7, 12 oder 17, für Codes mit variabler Länge, wobei es sich bei der Tabelle um eine programmierbare Logikanordnung handelt.

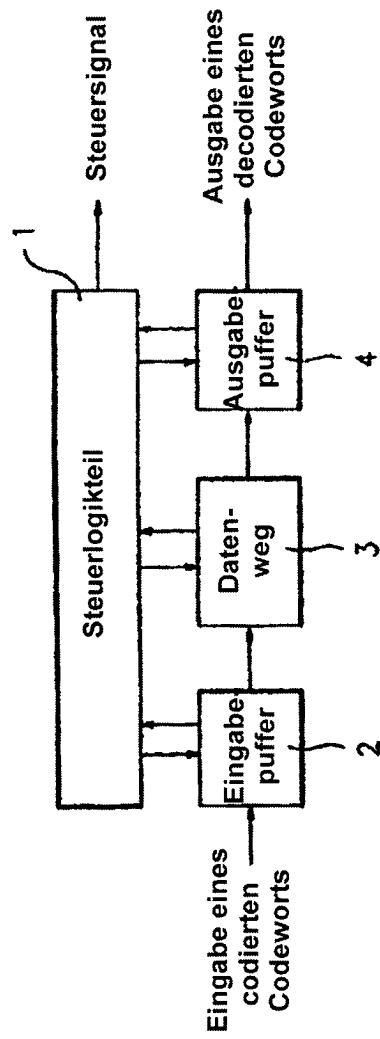
20. Decoder nach Anspruch 17, für Codes mit variabler Länge, wobei der dritte Multiplexer das Anfangseinstellungssignal ausgibt, wenn nur das Anfangseinstellungssignal in einem Anfangsstadium eingeht, und die Bitlänge anstatt des Anfangseinstellungssignals ausgibt, wenn die Bitlänge aus der Tabelle empfangen wird.

21. Decoder nach Anspruch 17, für Codes mit variabler Länge, darüber hinaus umfassend:  
einen Eingabepuffer, um das seriell eingegangene Codewort an den ersten Zwischenspeicher im Ansprechen auf das Lesesignal parallel auszugeben;  
und  
einen Ausgabepuffer, um das decodierte Codewort aus der Tabelle aus dem Decoder für Codes mit variabler Länge nach außen bereitzustellen.

22. Decoder nach Anspruch 17, für Codes mit variabler Länge, wobei sich das Übertragungssignal auf Tiefpegel befindet, wenn der Additionswert kleiner ist als eine voreingestellte längste Bitlänge eines Codeworts, und sich auf Hochpegel befindet, wenn der Additionswert größer ist als die voreingestellte längste Bitlänge des Codeworts.

Es folgen 6 Blatt Zeichnungen

FIG. 1  
Stand der Technik



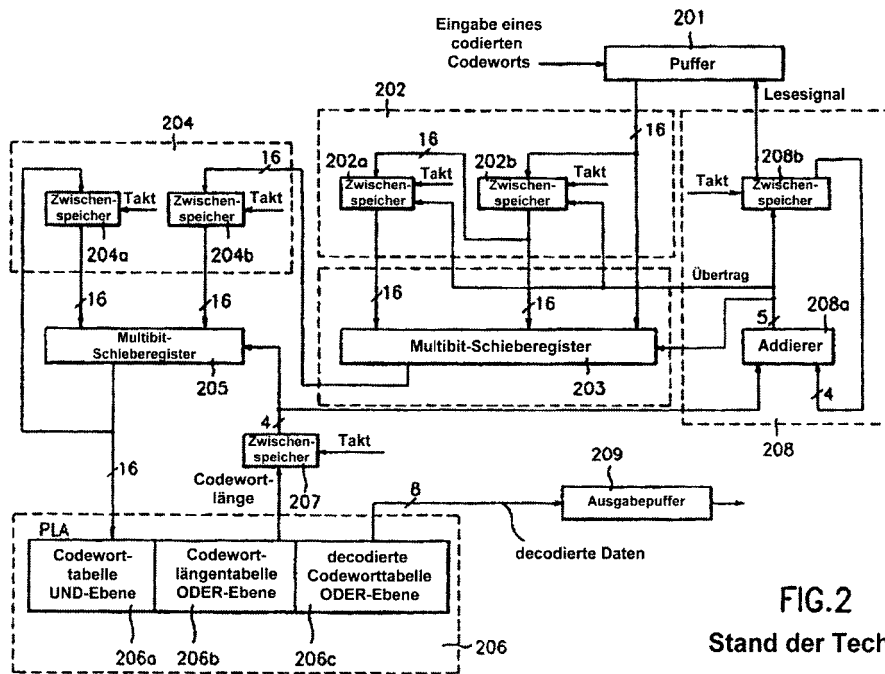


FIG. 2  
Stand der Technik

FIG.3

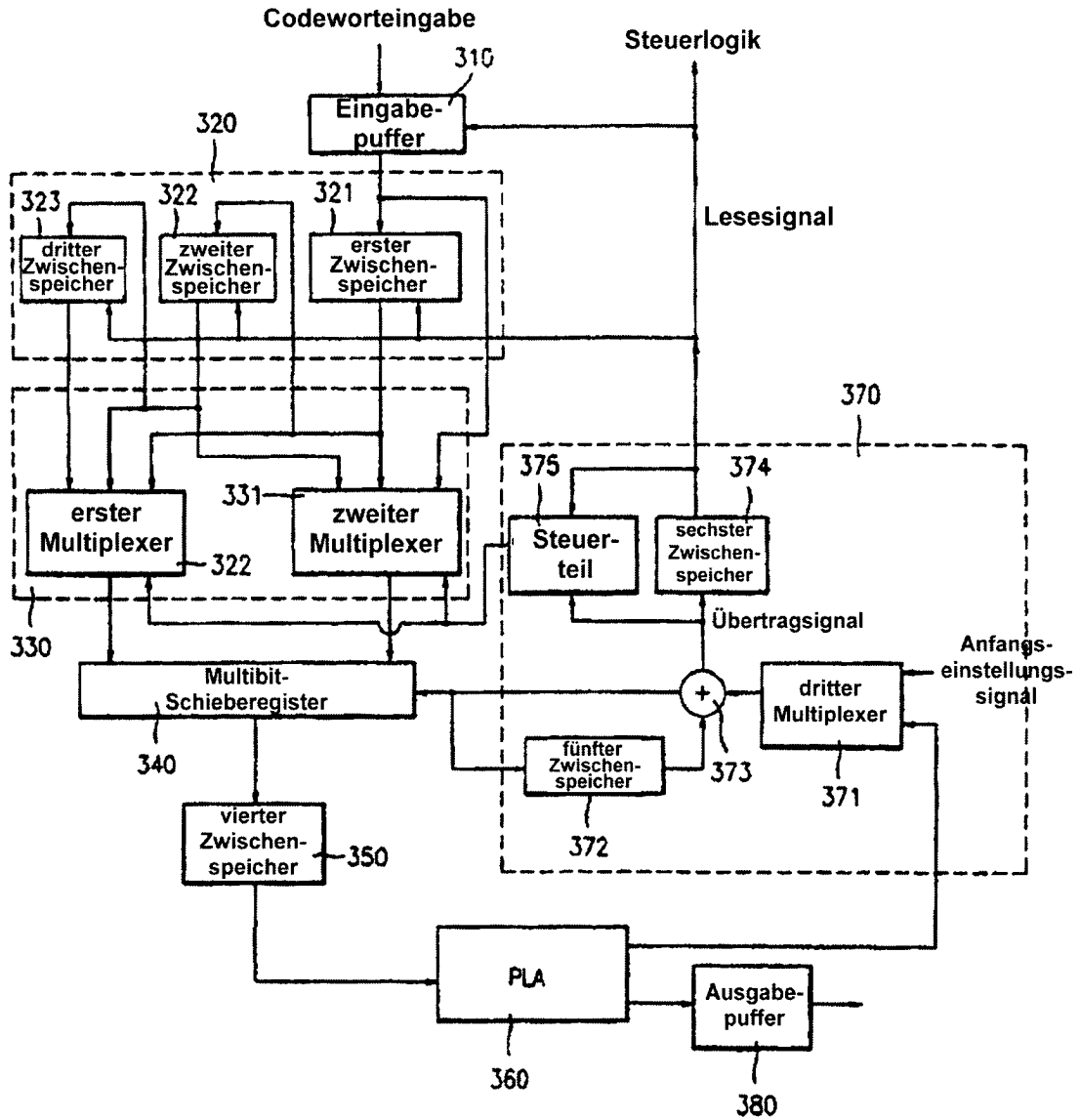


FIG.4

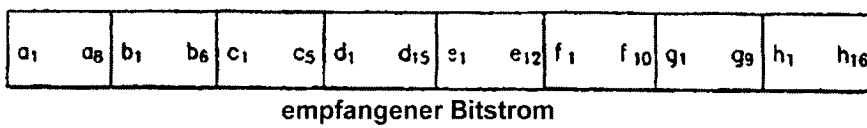


FIG.5

Takt	Zwischen- speicher D <sub>0</sub>	Zwischen- speicher D <sub>1</sub>	Zwischen- speicher D <sub>2</sub>	Zwischen- speicher D <sub>3</sub>	erster Multi- plexer	zweiter Multi- plexer	Ausgabe aus Multibit- Schiebe- register	Lesesignal	Addresser	tats. Ver- schiebe- wert	aufgezwin- gener An- fangswert	Puffer- ausgabe	Übertrag
0						a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>		1		0	16	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	1
1		a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>			a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	c <sub>7</sub> c <sub>6</sub> d <sub>7</sub> d <sub>6</sub>	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	1		0	16	c <sub>7</sub> c <sub>6</sub> d <sub>7</sub> d <sub>6</sub>	1
2		a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	c <sub>7</sub> c <sub>6</sub> d <sub>7</sub> d <sub>6</sub>	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	c <sub>7</sub> c <sub>6</sub> d <sub>7</sub> d <sub>6</sub>	b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub> d <sub>7</sub> d <sub>6</sub>	1		8		d <sub>7</sub> d <sub>6</sub> c <sub>7</sub> c <sub>6</sub> a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub>	0
3	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	c <sub>7</sub> c <sub>6</sub> d <sub>7</sub> d <sub>6</sub>	d <sub>7</sub> d <sub>6</sub> c <sub>7</sub> c <sub>6</sub> a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub>	b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub> d <sub>7</sub> d <sub>6</sub>	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	c <sub>7</sub> c <sub>6</sub> d <sub>7</sub> d <sub>6</sub>	c <sub>7</sub> c <sub>6</sub> d <sub>7</sub> d <sub>6</sub>	0	14	14		b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub> a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub>	0
4	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	c <sub>7</sub> c <sub>6</sub> d <sub>7</sub> d <sub>6</sub>	d <sub>7</sub> d <sub>6</sub> c <sub>7</sub> c <sub>6</sub> a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub>	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	c <sub>7</sub> c <sub>6</sub> d <sub>7</sub> d <sub>6</sub>	d <sub>7</sub> d <sub>6</sub> c <sub>7</sub> c <sub>6</sub> a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub>	d <sub>7</sub> d <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	0	19	3		a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	1
5	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	c <sub>7</sub> c <sub>6</sub> d <sub>7</sub> d <sub>6</sub>	d <sub>7</sub> d <sub>6</sub> c <sub>7</sub> c <sub>6</sub> a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub>	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	d <sub>7</sub> d <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub> d <sub>7</sub> d <sub>6</sub>	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	1	3	2		a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	1
6	c <sub>7</sub> c <sub>6</sub> d <sub>7</sub> d <sub>6</sub>	d <sub>7</sub> d <sub>6</sub> c <sub>7</sub> c <sub>6</sub> a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub>	b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub> d <sub>7</sub> d <sub>6</sub>	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	d <sub>7</sub> d <sub>6</sub> c <sub>7</sub> c <sub>6</sub> a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub>	b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub> d <sub>7</sub> d <sub>6</sub>	b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub> d <sub>7</sub> d <sub>6</sub>	1	2	14			0
7	d <sub>7</sub> d <sub>6</sub> c <sub>7</sub> c <sub>6</sub> a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub>	b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub> d <sub>7</sub> d <sub>6</sub>	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub> d <sub>7</sub> d <sub>6</sub>	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	0	14	8			1
8	d <sub>7</sub> d <sub>6</sub> c <sub>7</sub> c <sub>6</sub> a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub>	b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub> d <sub>7</sub> d <sub>6</sub>	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	1	8	1			1
9	b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub> d <sub>7</sub> d <sub>6</sub>	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>		a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>	a <sub>7</sub> a <sub>6</sub> b <sub>7</sub> b <sub>6</sub> c <sub>7</sub> c <sub>6</sub>			1	1	1			1

FIG.6

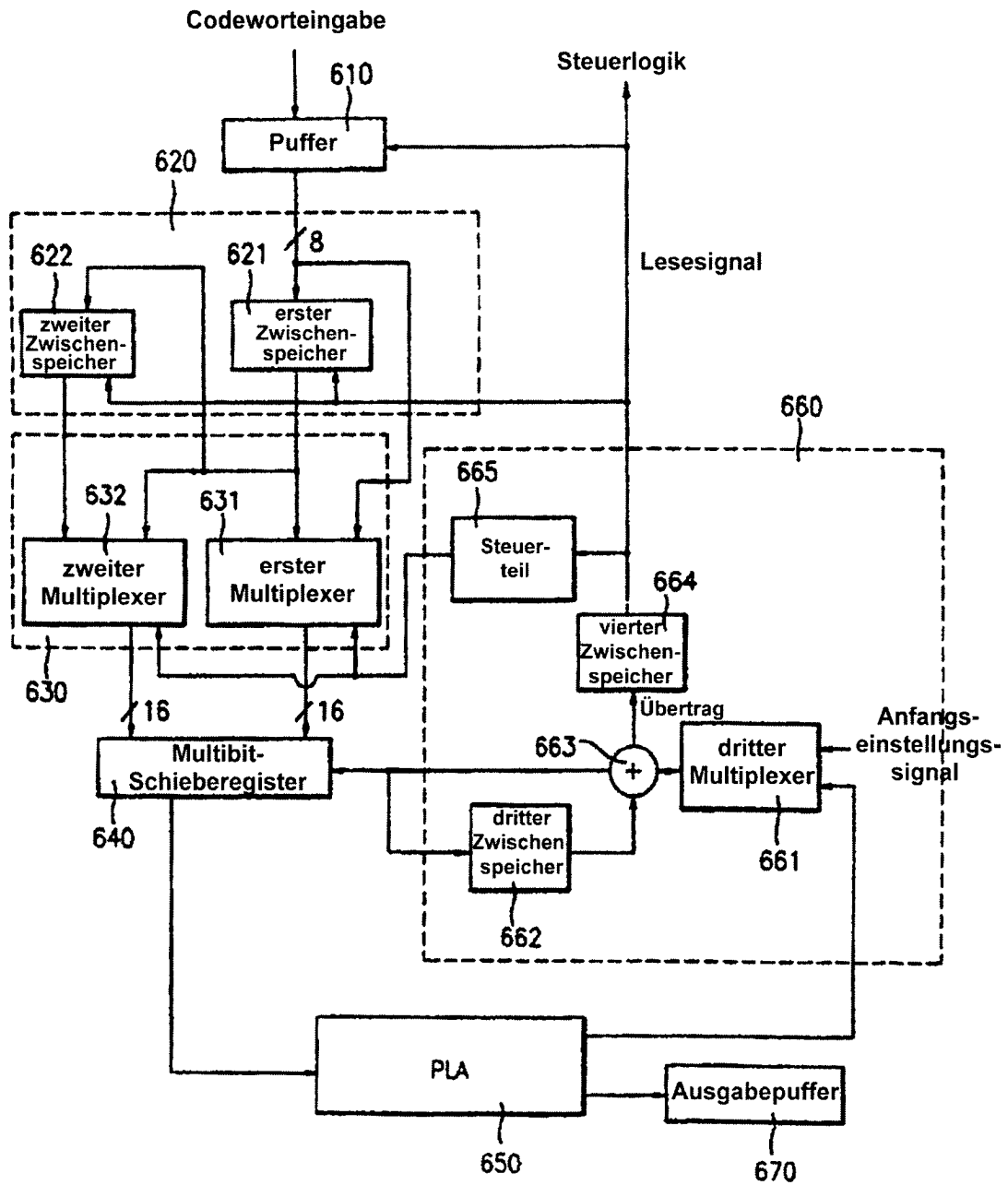


FIG.7

