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

(10) **DE 698 37 077 T2** 2007.06.21

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 0 928 071 B1**
(21) Deutsches Aktenzeichen: **698 37 077.5**
(96) Europäisches Aktenzeichen: **98 403 283.9**
(96) Europäischer Anmeldetag: **23.12.1998**
(97) Erstveröffentlichung durch das EPA: **07.07.1999**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **14.02.2007**
(47) Veröffentlichungstag im Patentblatt: **21.06.2007**

(51) Int Cl.[8]: ***H03M 13/27*** (2006.01)
H03M 13/29 (2006.01)

(30) Unionspriorität:

9716669	**30.12.1997**	**FR**
9814084	**09.11.1998**	**FR**

(84) Benannte Vertragsstaaten:
DE, FR, GB

(73) Patentinhaber:
Canon K.K., Tokio/Tokyo, JP

(74) Vertreter:
TBK-Patent, 80336 München

(72) Erfinder:
Le Dantec, Claude, 35140 Saint Hilaire des Landes, FR; Piret, Philippe, 35510 Cesson-Sevigne, FR

(54) Bezeichnung: **Verschachteler für Turbo-Kodierer**

Beschreibung

[0001] Die vorliegende Erfindung betrifft eine Codierungsvorrichtung, ein Codierungsverfahren, eine Decodierungsvorrichtung und ein Decodierungsverfahren, und Systeme, die diese implementieren.

[0002] Sie betrifft gleichermaßen die Codierung von eine physikalische Größe darstellenden Daten, die Codierung von Daten in Form von Codes, die eine physikalische Größe modulieren können, die Decodierung von datenmodulierten Signalen, und die Decodierung von Daten, die eine physikalische Größe darstellen. Diese Daten können z.B. Bilder, Töne, Computerdaten, elektrische Größen oder gespeicherte Daten darstellen.

[0003] Die Erfindung findet eine Anwendung auf dem Gebiet der Faltungscodes. Wenn Letztere zur Implementierung einer iterativen Decodierung verwendet werden, so werden diese Codes drastisch verbessert, wenn ihre Codierer eine Permutationsvorrichtung enthalten. In diesem Fall werden sie üblicherweise als "Turbocodierer" bezeichnet, und der entsprechende iterative Decodierer wird als "Turbodecodierer" bezeichnet.

[0004] Auf diesen Fachgebieten dienen als Referenzdokumente zum einen die Artikel von C. BERROU, A. GLAVIEUX und P. THITIMAJSHIMA mit dem Titel "Near Shannon limit error-correcting coding and decoding: turbocodes", veröffentlicht mit dem Tagungsband "ICC'93", 1993, Seiten 1064-1070, und andererseits die Artikel von C. BERROU und A. GLAVIEUX mit dem Titel "Near Optimum error-correcting coding and decoding: turbo-codes", veröffentlicht in den IEEE Transactions on Communication, Band COM-44, Seiten 1261-1271, im Oktober 1996.

[0005] Die Gestaltung von Permutationsvorrichtungen wird jedoch bei weitem nicht perfekt beherrscht. Im Allgemeinen verwendet diese Vorrichtung quadratische oder rechteckige Matrizen, in denen eine Zeile nach der anderen geschrieben wird, und eine Spalte nach der anderen gelesen wird. Diese Matrizen sind im Allgemeinen sehr groß, zum Beispiel von einer Größe von 256 × 256.

[0006] Gemäß einem anderen Verfahren haben DOLINAR und DIVSALAR in einem Artikel mit dem Titel "Weight distributions for turbo-codes using random and nonrandom permutations", veröffentlicht durch Jet Propulsion Laboratory, mit "TDA Progress Report", Nummer 42-122, vom 15. August 1995, die Permutationen betrachtet, die durch Nummerieren der k Informationspositionen zwischen 0 und k – 1 die an einer Position i platzierte binäre Information an eine Position e i + f verschieben, für "günstig gewählte" Werte von e und f.

[0007] In diesem Dokument geben Sie nur ein Beispiel, in dem k eine Potenz von 2 ist. Zudem diskutieren Sie nicht den möglichen gegenseitigen Einfluss zwischen der Wahl der Permutationsvorrichtung und der Wahl der elementaren Faltungscodierer (2, 1), die zum Generieren der durch den Turbocodierer (3, 1) erzeugten codierten Sequenzen verwendet werden sollen.

[0008] Die Bewertung des entsprechenden Turbocodes besteht aus dem Simulieren seiner Verwendung auf einem Übertragungskanal mit verschiedenen Signal/Rausch-Verhältnis-Werten, und Messen des minimalen Wertes dieses Verhältnisses, für den ein vorbestimmter Wert der Fehlerwahrscheinlichkeit der binären Werte erreicht wird.

[0009] Die Verwendung von Simulationen als Bewertungswerkzeug kann jedoch zu einigen Problemen führen.

[0010] Zum Beispiel sei wie vorangehend erwähnt die Permutationsvorrichtung mit k = 65536 = 256 × 256 gewählt, und eine vorbestimmte Fehlerwahrscheinlichkeit von 10^{-5} sei gewählt, um die Performanz eines Turbocodes unter Verwendung dieser Vorrichtung zu simulieren. Folglich ist die mittlere Anzahl von Fehlern der binären Werte pro Block von 256 × 256 nahe 1, aber es ist dabei ungewiss, ob jedes binäre Informationselement die gleiche Fehlerwahrscheinlichkeit hat. Diese Fehlerwahrscheinlichkeit könnte für binäre Werte mit einer "unglücklichen Position" in der Permutationsvorrichtung recht hoch sein, und für "glücklichere" Positionen könnte diese Wahrscheinlichkeit viel kleiner sein.

[0011] Um in dieser Situation Abhilfe zu schaffen, ist ein mögliches Verfahren, einen stimmigen und gemeinsamen Entwurf der Permutationsvorrichtung und der beiden Faltungscodierer zu erzeugen, um eine vernünftige Gleichmäßigkeit der Fehlerrate der binären Werte nach dem Decodieren, als eine Funktion der Position der binären Information in der Permutationsvorrichtung, zu garantieren.

[0012] Ein weiteres Problem betrifft den Mangel an algebraischen Werkzeugen zur Spezifizierung der Permu-

tationsvorrichtungen. Es wäre hilfreich, Einrichtungen zum Spezifizieren einer Auswahl von Permutationsvorrichtungen mit einer für die Menge aller Permutationsvorrichtungen repräsentativen Performanz zur Verfügung zu haben.

[0013] Die Erfindung betrifft prinzipiell die Übertragung von durch Sequenzen binärer Symbole dargestellter Information:

$\underline{u} = (u_0, u_1, ..., u_{k-1})$,

bezeichnet als "Informationssequenzen", die codiert werden in einem Tripel binärer Sequenzen,

$\underline{v} = (\underline{a}, \underline{b}, \underline{c})$,

wobei jede dieser Sequenzen $\underline{a}$, $\underline{b}$ und $\underline{c}$ an sich die Sequenz $\underline{u}$ repräsentiert.

[0014] Im Rest dieser Beschreibung wird zur Darstellung einer Sequenz $\underline{u}$ die Form $\underline{u} = (\underline{u}_0, \underline{u}_1, ..., \underline{u}_{k-1})$ indifferent zusammen mit der zugehörigen Polynomform verwendet:

$u(x) = u_0x^0 + u_1x^1 + ... + u_{k-1}x^{k-1}$.

[0015] Äquivalente Notationen werden für die Sequenzen $\underline{a}$, $\underline{b}$ und $\underline{c}$ verwendet. Unter Verwendung dieser zweiten Darstellung ist zur Bestimmung des Tripels $\underline{v} = (\underline{a}, \underline{b}, \underline{c})$ bekannt:

– $a(x) = u(x)$ zu wählen,

– $b(x) = u(x) \cdot h_1(x)/g(x)$ zu wählen,
wobei $g(x)$ das Polynom ist, z.B. $g(x) = 1 + x + x^3$, das, entsprechend der Darstellung in einer Sequenz, der Sequenz (1, 1, 0, 1) entspricht; und $h_1(x)$ ist ein Polynom, z.B. $h_1(x) = 1 + x + x^2 + x^3$, das der Sequenz (1, 1, 1, 1) entspricht, und

– durch Bezeichnen einer durch Permutation der binären Daten der Sequenz $\underline{a}$ gebildeten Sequenz als $\underline{a}^*$ den Ausdruck $c(x) = a^*(x) \cdot h_2(x)/g(x)$ zu wählen,
wobei $h_2(x)$ ein Polynom ist, z.B. $h_2(x) = (1 + x^2 + x^3)$, das der Sequenz (1, 0, 1, 1) entspricht.

[0016] Eine beliebige Wahl der Polynome $g(x)$, $h_1(x)$, $h_2(x)$ und der Permutation, die den Verschachtler spezifiziert, der die permutierte Sequenz $\underline{a}^*$ mit der Sequenz $\underline{a}$ assoziiert, spezifiziert einen Codierer, der als "Turbocodierer" bezeichnet wird. Alle Sequenzen, die ein spezifizierter Turbocodierer erzeugen kann, werden als "Turbocode" bezeichnet.

[0017] Im Rest der Beschreibung wird der Begriff "erster Codierer" für den elementaren rekursiven Faltungscodierer verwendet, der die Sequenz $\underline{b}$ erzeugt, und der Begriff "zweiter Codierer" wird für denjenigen verwendet, der die Sequenz $\underline{c}$ erzeugt.

[0018] Die verwendeten Polynomdivisionen sind von dem Typ, der aus einer Division gemäß aufsteigenden Potenzen besteht, wie es für einen Fachmann wohlbekannt ist. Sie verwenden eine Modulo-2-Arithmetik. Die Sequenzen $\underline{a}$, $\underline{b}$, und $\underline{c}$ sind binäre Sequenzen, und im allgemeinen Fall haben die $\underline{b}$ und $\underline{c}$ definierenden Divisionen einen Rest.

[0019] Diese Art von Codierungsverfahren hat den Vorteil, dass sie sich für eine iterative Decodierung eignet, die leistungsfähig, relativ einfach und kostengünstig ist.

[0020] Bei ihrer Implementierung stellen sich mehrere Fragen.

I. Wie sollten die Polynome $g(x)$, $h_1(x)$ und $h_2(x)$ gewählt werden?

II. Wie sollte die Permutation der Terme der Sequenz $\underline{a}$ gewählt werden, die die Sequenz $\underline{a}^*$ erzeugt?

Unter den vorgeschlagenen Wahlmöglichkeiten sind nachfolgend drei beispielhafte Verschachtler, d.h. Operatoren, die zur Bildung der Sequenz $\underline{a}^*$ die Terme der Sequenz $\underline{a}^*$ permutieren, angegeben:

A) In dem ersten Beispiel wird nach dem Anordnen der Terme von $\underline{a}$ in einer rechteckigen Tabelle, sukzessiv Zeile für Zeile und, für jede Zeile, von links nach rechts, die Sequenz $\underline{a}^*$ durch Nehmen der Terme in dieser Tabelle sukzessiv Spalte für Spalte und, für jede Spalte, von oben nach unten, gebildet. Zum Beispiel transformiert in dem Fall von Sequenzen mit sechs Termen unter Verwendung einer Tabelle von zwei Zeilen mit drei Spalten der Verschachtler die Sequenz $\underline{a} = (a_0, a_1, a_2, a_3, a_4, a_5)$ in die Sequenz $\underline{a}^* = (a_0, a_3, a_1, a_4, a_2, a_5)$.

B) In einem zweiten Beispiel wird der i-te Term (i = 0, 1, 2, ...) a^*_i der Sequenz $\underline{a}^*$ als der Term a_j der Sequenz $\underline{a}$ gewählt, wobei $j = s \cdot i + t$ modulo der Anzahl von Termen in der Sequenz $\underline{a}$ berechnet wird, und s und t

Ganzzahlen sind. Falls zum Beispiel die Anzahl von Termen der Sequenz a sechs ist, und s = 5 und t = 3 ist, transformiert der Verschachtler die Sequenz a = (a_0, a_1, a_2, a_3, a_4, a_5) in die Sequenz a* = (a_3, a_2, a_1, a_0, a_5, a_4).

C) In dem dritten Beispiel ist die gewählte Permutation zufällig.

III. Wie kann vermieden werden, dass die b(x) definierende Division keinen Rest hat?

IV. Wie kann vermieden werden, dass die c(x) definierende Division einen Rest hat?

[0021] Eine Beantwortung dieser letzten beiden Fragen stellt die Lösung eines in der Literatur über Turbocodes häufig erwähnten Problems dar, und zwar das Problem des "Zurückkehrens zum Null-Zustand" des b und c definierenden elementaren Faltungscodierers. Da die Turbocodierer zwei elementare rekursive Codierer aufweisen, wobei der zweite eine Permutation a* der Sequenz a verwendet, ist es wünschenswert zu garantieren, dass die die Informationssequenz u(x) darstellenden Polynome a(x) und a*(x) gleichzeitig durch g(x) teilbar sind. Das Sicherstellen dieser Teilbarkeitsbedingung für a(x) ist leicht, da es hinreichend ist, a(x) aus u(x) zu konstruieren, und dabei u(x) mit Füllsymbolen in einer Anzahl gleich dem Grad von g(x) zu ergänzen, deren einzige Funktion es ist, die Abwesenheit eines Rests in der b(x) aus a(x) erzeugenden Division zu garantieren.

[0022] Das Wählen einer a*(x) aus a(x) erzeugenden Permutation, die sowohl die Teilbarkeit von a*(x) durch g(x) als auch gute Fehlerkorrekturleistungen für den somit spezifizierten Turbocode garantiert, ist andererseits schwieriger.

[0023] Dieses Problem kann zu Disparitäten zwischen den Fehlerwahrscheinlichkeiten nach dem Decodieren der verschiedenen Bits, die u(x) darstellen, führen.

[0024] In einem in Band 31 Nr. 1 des Journals "Electronic Letters" vom 5. Januar 1995 erschienenen Artikel offenbaren BARBULESCU und PIETROBON, dass ein Verschachtler beschrieben werden kann durch sukzessives und zyklisches Anordnen der Terme der Sequenz a in einer Anzahl von Sequenzen, die gleich dem um eins inkrementierten Grad des Polynoms g(x) ist, und das in einem solchen Fall Permutationen innerhalb einer jeden der somit gebildeten Sequenzen zu einer Gleichheit zwischen dem Rest der die Sequenz b definierenden Division und dem der die Sequenz c definierenden Division führen.

[0025] Im Gegensatz zu Aussagen in diesem Artikel ist dies jedoch nur zutreffend, falls das Polynom g(x) von der Form $\Sigma_{i=0 \text{ bis } m}\ x^i$ ist.

[0026] In einem mit den Berichten des durch das Institute of Technology of Lund (Schweden) (Department of Applied Electronics) organisierten Seminars "Turbo Coding" im August 1996 veröffentlichten Artikel mit dem Titel "Turbo-block-codes" haben C. BERROU, S. EVANO und G. BATTAIL offenbart, dass durch zyklisches Anordnen der Terme der Sequenz u in einer Anzahl von Spalten, die gleich ist einem Vielfachen des Grades N0 des Polynoms vom Typ $x^n - 1$ des kleinsten strikt positiven Grades, das durch g(x) teilbar ist, Permutationen innerhalb einer jeden somit gebildeten Spalte bedeuten, dass die Summe des Rests der die Sequenz b definierenden Division und des Rests der die Sequenz c definierenden Division Null ist, sodass die Verkettung der Sequenzen durch g teilbar ist.

[0027] Wie auch das vorhergehende Dokument beschränkt deshalb dieses Dokument die Wahl der Verschachtler auf bestimmte Formen, die unabhängig auf Untermengen der Terme der Sequenz a durch Anwenden interner Permutationen auf diese arbeiten. Es ist jedoch nicht garantiert, dass a(x) und a*(x) für sich durch g(x) teilbar sind. Das Einzige, das garantiert ist, ist die Teilbarkeit durch g(x) des Polynoms, das die Verkettung (a, a*) darstellt, die aus einer Ende-an-Ende-Aneinanderreihung der zwei Sequenzen a und a* besteht.

[0028] Es ergibt sich ein möglicher Verlust an Wirksamkeit des Decodierers, da dieser nicht über den Zustand informiert wird, den der Decodierer in dem Moment hatte, der das Ende der Berechnung von b und den Beginn der Berechnung von c markiert.

[0029] Der Artikel von W.J. BLACKERT et al. mit dem Titel "Turbo code termination and interleaver conditions", in Electronic Letters, Band 31, Nr. 24, 23. November 1995, Seiten 2082-2084, behandelt das Problem des Findens einer Permutation, die die Teilbarkeitsbedingung im Bereich der Turbocodierung einhält.

[0030] Keiner der zitierten Artikel schlägt eine wirksame Wahl des Verschachtlers vor.

[0031] Die vorliegende Erfindung schafft Abhilfe für diese Nachteile, indem sie einerseits Familien von Verschachtlern vorschlägt, die eine Rückkehr nach Null bzw. Return-to-Zero am Ende der Sequenz c garantieren,

wenn die Sequenz b nach Null zurückkehrt, und andererseits eine Auswahl von Verschachtlern vorschlägt, die breiter ist als die durch die vorangehend zitierten Artikel vorgeschlagene.

[0032] Zu diesem Zweck betrifft die vorliegende Erfindung gemäß einem ersten Aspekt ein Codierungsverfahren gemäß Anspruch 2, das dadurch gekennzeichnet ist, dass:

1) es berücksichtigt:

– eine vorbestimmte Ganzzahl M1, die gleich oder größer 2 ist,

– eine Zahl K, die größer oder gleich 1 ist, von Sequenzen a_i (i = 1, ..., K) von binären Daten, die eine physikalische Größe darstellen, wobei jede Sequenz a_i aufweist:

• eine Polynomdarstellung $a_i(x)$, die ein Vielfaches von einem vorbestimmten Polynom $g_i(x)$ ist, und

• eine Zahl von binären Daten, die gleich dem Produkt einer beliebigen Ganzzahl M und der Ganzzahl N0 ist, wobei die kleinste Ganzzahl derart ist, dass das Polynom $x^{N0} + 1$ durch jedes der Polynome $g_i(x)$ teilbar ist,

2) das Verfahren eine erste Operation zum Erzeugen einer Zahl K*M1 von so genannten "permutierten" Sequenzen a_{ij}^* (i = 1, ..., K; j = 1, ..., M1) umfasst, wobei jede Sequenz a_{ij}^*:

– durch eine Permutation der entsprechenden Sequenz a_i erhalten wird, wobei die Permutation in einer Darstellung, bei der die binären Daten von jeder Sequenz a_i zeilenweise in einer Tabelle von N0 Spalten und M Zeilen geschrieben werden, die Resultante von einer beliebigen Zahl von so genannten elementaren Permutationen ist, die jeweils

• entweder die Eigenschaft aufweisen zum Transformieren des zyklischen Codes der Länge N0 und mit einem Generatorpolynom $g_i(x)$ in einen äquivalenten zyklischen Code mit einem Generatorpolynom $g_{ij}(x)$, das gleich $g_i(x)$ sein kann, und durch Permutation der N0 Spalten der a_i darstellenden Tabelle arbeiten,

• oder eine beliebige Permutation der Symbole einer Spalte der Tabelle sind, und

– folglich eine Polynomdarstellung $a_{ij}^*(x)$ aufweist, die gleich einem Polynomprodukt $c_{ij}(x)g_{ij}(x)$ ist,

– wobei zumindest eine permutierte Sequenz a_{ij}^* verschieden ist von der entsprechenden Sequenz a_i,

3) das Verfahren eine zweite Operation zum Erzeugen von M1 redundanten Sequenzen umfasst, deren Polynomdarstellung für j = 1, ... M1 gleich $\Sigma f_{ij}(x)c_{ij}(x)$ ist, wobei jedes Polynom $f_{ij}(x)$ ein Polynom mit einem Grad ist, der höchstens gleich dem Grad des Polynoms $g_{ij}(x)$ mit den gleichen Indizes i und j ist.

[0033] In einer Darstellung, in der die binären Daten der Sequenz a in einer Tabelle mit N0 Spalten und M Zeilen eingeordnet sind, wurden vorangehend Abfolgen von Permutationen eingeführt, die von all den Permutationen genommenen sind, die einerseits die Automorphismen des binären zyklischen Codes der Länge N0 und mit einem Generatorpolynom g(x), die zumindest zwei der N0 Spalten der Tabelle miteinander permutieren, umfassen, und andererseits die ausschließlich auf Daten in der gleichen Spalte arbeitenden und zumindest zwei der Daten miteinander permutierenden Permutationen umfassen.

[0034] Die Erfinder haben entdeckt, dass all diese Abfolgen von Permutationen, und nur diese, für ein beliebiges Polynom a(x), dessen Division durch g(x) einen Null-Rest ergibt, garantieren, dass das permutierte Polynom a*(x) die gleiche Eigenschaft aufweist.

[0035] Für eine Untersuchung der Bedingungen, die die Wahl von g_{ij} bestimmen, sei der Leser auf Seite 234 des Buches von F.J. MACWILLIAMS und N.J.A. SLOANE mit dem Titel "The theory of error-correcting codes" verwiesen, veröffentlicht durch North-Holland im Jahre 1977, dessen siebte Auflage im Jahre 1992 gedruckt wurde.

[0036] Die gesamte in der vorliegenden Erfindung beschriebene Auswahl umfasst die in den beiden vorangehend erwähnten Artikeln offenbarten Verschachtler. Somit kann die im Sinne einer Fehlerrate als Funktion des Signal/Rausch-Verhältnisses ausgedrückte Performanz ohne Erhöhung der Komplexität des Turbocodierers oder des Turbodecodierers verbessert werden.

[0037] Gemäß einem zweiten Aspekt betrifft die vorliegende Erfindung ein Decodierungsverfahren gemäß Anspruch 13.

[0038] Aufgrund dieser Maßnahmen kann einerseits die Mehrheit der Spalten in der Tabelle durch Permutation bewegt werden, und andererseits kann in dieser beschränkten Auswahl die minimale Distanz des Turbocodes leichter analysiert und deshalb optimiert werden.

[0039] Dieser zweite Aspekt der Erfindung hat die gleichen Vorteile wie der erste Aspekt.

[0040] Die Erfinder haben beobachtet, dass ein Implementieren des Verfahrens der vorliegenden Erfindung

in jedem ihrer wie vorangehend offenbarten Aspekte den Vorteil hat, dass jede Fehlerschätzung durch den entsprechenden Decodierer konvergiert. Der Fall, in dem die Fehlerschätzung nicht konvergiert, ist deshalb durch die Implementierung der vorliegenden Erfindung ausgeschlossen.

[0041] Gemäß besonderen Merkmalen sind während der ersten Erzeugungsoperation alle Werte der Exponenten e_{ij} mit dem gleichen Wert des Index j identisch.

[0042] Aufgrund dieser Maßnahmen macht es das Codierungsverfahren, das die vorliegende Erfindung betrifft, möglich, alle Verschachtelungen bezüglich j fest in der gleichen Weise auszuführen. Es ist deshalb leicht zu implementieren.

[0043] Gemäß besonderen Merkmalen sind während der ersten Erzeugungsoperation alle Werte der Exponenten e_{ij} gleich einer Potenz von 2.

[0044] Aufgrund dieser Maßnahme sind alle Polynome g_{ij} identisch.

[0045] Gemäß besonderen Merkmalen umfasst das wie vorangehend in Kürze offenbarte Codierungsverfahren, das die vorliegende Erfindung betrifft, eine Operation zum Übertragen von, einerseits, Sequenzen $\underline{a}_i$, und andererseits einer Untermenge der Daten der anderen Sequenzen.

[0046] Aufgrund dieser Maßnahmen ist die Effizienz des Verfahrens gesteigert.

[0047] Weitere eine Codierungsvorrichtung und Decodierungsvorrichtung betreffende Aspekte der vorliegenden Erfindung sind in den Ansprüchen 8 bzw. 22 festgelegt.

[0048] Die Erfindung betrifft zudem:
– eine Informations-Speichervorrichtung, die durch einen Computer- oder Mikroprozessor gelesen werden kann und Instruktionen eines Computerprogramms speichert, dadurch gekennzeichnet, dass sie eine Implementierung des wie vorangehend in Kürze offenbarten Verfahrens der Erfindung ermöglicht, und
– einer teilweise oder vollständig entfernbaren Informations-Speichervorrichtung, die durch einen Computer oder einen Mikroprozessor gelesen werden kann und Informationen eines Computerprogramms speichert, dadurch gekennzeichnet, dass sie eine Implementierung des wie vorangehend in Kürze offenbarten Verfahrens der Erfindung ermöglicht.

[0049] Die Erfindung betrifft zudem:
– eine Vorrichtung zur Verarbeitung Sprache repräsentierender Signale, die eine wie vorangehend in Kürze offenbarte Vorrichtung umfasst,
– eine Datenübertragungsvorrichtung mit einem zur Implementierung eines Paketübertragungsprotokolls angepassten Senders, die eine wie vorangehend in Kürze offenbarte Vorrichtung umfasst,
– eine Datenübertragungsvorrichtung mit einem zur Implementierung des ATM-(Asynchronous Transfer Mode) Paketübertragungsprotokolls angepassten Senders, die eine wie vorangehend in Kürze offenbarte Vorrichtung aufweist,
– eine Datenübertragungsvorrichtung mit einem zur Implementierung des Paketübertragungsprotokolls angepassten Senders, auf einem Netzwerk des ETHERNET-(eingetragenes Warenzeichen) Typs,
– eine Netzwerkstation mit einer wie vorangehend in Kürze offenbarten Vorrichtung,
– einer Datenübertragungsvorrichtung mit einem auf einem drahtlosen Kanal sendenden Sender, die eine wie vorangehend in Kürze offenbarte Vorrichtung aufweist, und
– eine Vorrichtung zum Verarbeiten von Sequenzen von Signalen, die höchstens eintausend binäre Daten darstellen, die eine wie vorangehend in Kürze offenbarte Vorrichtung aufweist.

[0050] Da diese Codierungs- und Decodierungsvorrichtungen, diese Codierungs- und Decodierungsverfahren und diese Signalverarbeitungs-, Datenübertragungs- und Sequenzverarbeitungsvorrichtungen und dieses Netzwerk die gleichen besonderen Merkmale und die gleichen Vorteile wie das wie vorangehend in Kürze offenbarte Codierungsverfahren aufweisen, werden diese besonderen Merkmale und diese Vorteile hier nicht wiederholt.

[0051] Die Erfindung wird durch Lesen der mit Bezug auf die beigefügte Zeichnung angefertigten und nun folgenden Beschreibung klarer verständlich, wobei in der Zeichnung:

[0052] Fig. 1 schematisch eine elektronische Vorrichtung darstellt, die einen Codierer gemäß einem ersten

Ausführungsbeispiel der vorliegenden Erfindung umfasst,

[0053] Fig. 2 schematisch ein Betriebsablaufdiagramm eines wie in Fig. 1 veranschaulichten Codierers gemäß dem ersten Ausführungsbeispiel der vorliegenden Erfindung darstellt,

[0054] Fig. 3 schematisch ein Ablaufdiagramm darstellt, das die Schritte zur Bestimmung eines in der in Fig. 1 veranschaulichten Vorrichtung verwendeten Verschachtlers beschreibt,

[0055] Fig. 4 schematisch einen Codierer gemäß dem zweiten Ausführungsbeispiel der vorliegenden Erfindung darstellt,

[0056] Fig. 5 schematisch die allgemeine Form von auf mehreren Sequenzen von zu codierenden Symbolen arbeitenden Verschachtlern darstellt,

[0057] Fig. 6 eine Performanzkurve besonderer Beispiele für Methoden und Verfahren gemäß dem zweiten Ausführungsbeispiel der vorliegenden Erfindung darstellt,

[0058] Fig. 7 schematisch eine elektronische Vorrichtung darstellt, die einen Codierer gemäß einem zweiten Ausführungsbeispiel der vorliegenden Erfindung umfasst,

[0059] Fig. 8 schematisch ein Betriebsablaufdiagramm eines wie in Fig. 7 veranschaulichten Codierers gemäß dem zweiten Ausführungsbeispiel der vorliegenden Erfindung darstellt, und

[0060] Fig. 9 schematisch eine Decodierungsvorrichtung darstellt, die die vorliegende Erfindung betrifft.

[0061] In der folgenden Beschreibung wird die erste Steuersequenz immer von nicht-verschachtelten Informationssequenzen erhalten, obwohl sich der Rahmen der Erfindung auch auf den allgemeinen Fall erstreckt.

[0062] Die Beschreibung der Ausführungsbeispiele der Erfindung ist nachfolgend in zwei Teile unterteilt, die den Fall betreffen, in dem eine einzige Sequenz von Symbolen codiert wird, bzw. den Fall, in dem zwei Sequenzen von Symbolen gleichzeitig codiert werden.

I – ERSTES AUSFÜHRUNGSBEISPIEL

[0063] In der folgenden Beschreibung beschreibt der Begriff "Daten" sowohl Information darstellende Symbole, als auch zusätzliche oder redundante Symbole.

[0064] Vor dem Beginn der Beschreibung eines besonderen Ausführungsbeispiels werden nachfolgend die mathematischen Grundlagen seiner Implementierung gegeben.

[0065] In der Erfindung ist angegeben, dass es wünschenswert ist, dass die durch die Divisionen $b(x) = a(x)\cdot h_1(x)/g(x)$ bzw. $c(x) = a^*(x)\cdot h_2(x)/g(x)$ definierten Sequenzen b und c keinen Rest aufweisen, wobei g(x), $h_1(x)$ und $h_2(x)$ vorbestimmt sind.

[0066] Zu diesem Zweck wird die kleinste Zahl N0 gesucht, sodass g(x) das Polynom $x^{N0} - 1$ teilt, wobei $g(x) = 1 + \Sigma_{i=1 \text{ bis } m-1}\, g_i\cdot x^1 + x^m$ ein Polynom des vorbestimmten Grades m ist. Es ist bekannt, dass diese Zahl existiert. Beispielsweise $g(x) = 1 + x + x^3$, N0 = 7.

[0067] Dann wird durch Wählen einer beliebigen Zahl M eine Länge einer Sequenz a gleich M.N0 gewählt, was der Bestimmung der Länge (das heißt der Anzahl von binären Daten) der in der Sequenz a enthaltenen Sequenz u als gleich M.N0 minus dem Grad von g(x) entspricht.

[0068] Somit wird zur Bildung der Sequenz a der durch zu übertragende k binäre Daten u_i gebildeten Sequenz u eine dem Grad des Polynoms g(x) gleiche Anzahl zusätzlicher binärer Daten nebenangestellt, wobei die hinzugefügten Daten die Abwesenheit eines Rests bei der Division von a(x) durch g(x) garantieren.

[0069] Es sei daran erinnert, dass die hier modulo 2 durchgeführte Division auf den Koeffizienten der aufsteigenden Potenzen von a(x) durchgeführt wird.

[0070] Falls, zum Beispiel, die Sequenz u die Sequenz (1, 0, 0, 1, 0, 0) ist, und die Sequenz g die Sequenz

(1, 1, 0, 1) ist, schreibt sich die Division als:

```
1 0 0 1 0 0                    | 1 1 0 1
1 1 0 1                        | ---------
  1 1 0 1                      | 1 1 1 1 0 1
    1 1 0 1                    |
      1 1 0 1                  |
        1 1 0 1                |
          0 0 0 0              |
            1 1 0 1            |
-------------------------------
          0 0 0 0 0 0 0 0 1
```

was sich auch schreiben lässt als: (1, 0, 0, 1, 0, 0, 0, 0, 0) = (1, 1, 0, 1) × (1, 1, 1, 1, 0, 1) + (0, 0, 0, 0, 0, 0, 0, 0, 1), dass heißt auch (1, 0, 0, 1, 0, 0, 0, 0, 1) = (1, 1, 0, 1) × (1, 1, 1, 1, 0, 1), unter termweisem Addieren der Rest-Sequenz (0, 0, 0, 0, 0, 0, 0, 0, 1) zur zunächst durch m "0"en ergänzten Sequenz u.

[0071] Somit ist durch Ersetzen der Sequenz u = (1, 0, 0, 1, 0, 0) durch die Sequenz a = (1, 0, 0, 1, 0, 0, 0, 0, 1), die durch diese Addition gebildet wird, und deren erste binären Daten und die ersten binären Daten alle die binären Daten der Sequenz u sind, die Teilbarkeit des Polynoms a(x) durch das mit der Sequenz g = (1, 1, 0, 1) assoziierte Polynom g(x), und somit auch die Teilbarkeit von $a(x)\cdot h_1(x)$ durch g(x), garantiert, unabhängig von der mit dem Polynom $h_1(x)$ assoziierten Sequenz h_1, was die Definition der Sequenz b durch $b(x) = a(x)\cdot h_1(x)/g(x)$ liefert.

[0072] Um die Sequenz a*(x) zu bestimmen, die nach Permutation die gleichen binären Daten aufweist wie die Sequenz a, jedoch in einer unterschiedlichen Reihenfolge, wird ein Verschachtler gewählt, für den eine Darstellung wie folgt gegeben werden kann: Auf die in einer Tabelle mit N0 Spalten und M Zeilen eingeordneten binären Daten der Sequenz a wird zumindest eine Permutation in einer Menge von Permutationen durchgeführt, die einerseits die Automorphismen des binären zyklischen Codes der Länge N0 und mit dem Generatorpolynom g(x), die zumindest zwei der N0 Spalten der Tabelle miteinander permutieren, umfassen, und andererseits die ausschließlich auf Daten in der gleichen Spalte arbeitenden und zumindest zwei der Datenelemente miteinander permutierenden Permutationen umfassen, und nur eine oder mehrere Permutationen dieser Menge.

[0073] Dies ist der Fall, weil die Erfinder entdeckt haben, dass nur die dadurch darstellbaren Permutationen garantieren, dass für ein beliebiges Polynom a(x), dessen Division durch g(x) keinen Rest aufweist, das permutierte Polynom a(x) eine Division durch g(x) ohne Rest aufweist.

[0074] In einer derartigen Abfolge kann, wenn zum Beispiel N0 = 7 und $g(x) = 1 + x + x^3$, sukzessiv gefunden werden:

- eine Permutation der binären Daten der ersten Spalte,
- eine Permutation der binären Daten der dritten Spalte,
- die Ersetzung der zweiten Spalte durch die vierte, die Ersetzung der vierten Spalte durch die zweite, die Ersetzung der fünften Spalte durch die sechste, und die Ersetzung der sechsten Spalte durch die fünfte.

[0075] Mit Bezug auf den Automorphismus, wobei die Bezeichnung Cg verwendet wird für den binären zyklischen Code der Länge N0 und mit dem Generatorpolynom g(x), das heißt alle Vielfache von g(x), modulo $x^{N0} - 1$, werden die Permutationen der Koordinaten dieses Codes betrachtet, die ein beliebiges Wort dieses Codes in ein anderes Wort dieses Codes transformieren. Die Menge von Permutationen von Koordinaten, die diese Eigenschaft aufweisen, hat eine Gruppenstruktur und wird als die Cg-Automorphismusgruppe bezeichnet.

[0076] Für weitere Details sei auf die Arbeit von F.J. MACWILLIAMS und N.J.A. SLOANE verwiesen, "The theory of error-correcting codes", veröffentlicht durch North-Holland im Jahre 1977.

[0077] Unter all diesen Permutationen haben die Erfinder die folgenden Permutationen ausgewählt, die den Vorteil haben, dass sie nur eine kleine Familie darstellen, deren Elemente alle getestet werden können, um die effektivste Permutation auszuwählen.

[0078] Wie vorangehend erwähnt wird M ungerade gewählt, und g(x) derart, dass die entsprechende Zahl N0 ebenfalls ungerade ist. Durch Schreiben der sukzessiven Potenzen von 2 modulo M.N0 erhält man den so genannten Zyklus von 2, modulo M.N0. In diesem Zyklus wird durch Wählen eines beliebigen Terms e die folgende Permutation durchgeführt:
das Polynom a(x) ergibt nach Permutation das Polynom a*(x), wobei Letzteres definiert ist durch $a^*(x) = a(x^e)$, das heißt falls $\underline{a} = (a_0, a_i, a_2, ..., a_{M.N0-1})$ ist das erste binäre Datenelement von a* a_0, das zweite a_f, das dritte a_{2f}, ..., wobei f die Inverse von e ist, modulo M.N0, und die Vielfachen von f selbst modulo M.N0 berechnet werden.

[0079] Zum Beispiel ergibt sich durch Wiederholen von $g(x) = 1 + x + x^3$, und deshalb N0 = 7 und Wählen von M = 5, M.N0 = 35. Der Zyklus von 2 wird dann geschrieben als:
[1, 2, 4, 8, 16, 32, 29, 23, 11, 22, 9, 18].

[0080] Beispielsweise ist, für $e = 2^8 = 11$, f gleich 16, und die Sequenz $\underline{a}^*$ beginnt mit den binären Daten: a_0, a_{16}, a_{32}, a_{13}, a_{29}, a_{10} usw. ...

[0081] Diese Permutationen, die vorangehend beschrieben wurden und durch $a^*(x) = a(x^e)$ dargestellt werden können, wobei e in dem Zyklus von 2 modulo M.N0 ist, und wobei $a(x^e)$ modulo $x^{M.N0} – 1$ genommen wird, bilden eine kleine Familie, deren Elemente alle getestet werden können, um die effektivste auszuwählen.

[0082] Die Auswahllogik ist wie folgt: Zunächst wird ein durch ein quadratisches Polynom nicht teilbares g(x) vom Grad m gewählt. Diese Wahl legt den Wert von N0 als die kleinste Ganzzahl fest, sodass g(x) $x^{N0} – 1$ teilt. Zudem impliziert die Tatsache, dass das Polynom g(x) nicht durch ein quadratisches Polynom teilbar ist, dass N0 eine ungerade Zahl ist. Dann werden $h_1(x)$ und $h_2(x)$ von einem beliebigen, vorzugsweise jedoch einem höchstens dem Grad m von g(x) entsprechenden Grad gewählt, da das Maximum der Grade dieser drei Polynome g(x), $h_1(x)$ und $h_2(x)$ ein entscheidender Faktor für die Komplexität des Decodierers ist. Dann wird eine ungerade Ganzzahl M gewählt, und der Zyklus von zwei modulo M.N0 wird berechnet. Dann wird ein Element e in diesem Zyklus von 2 gewählt, um $a^*(x) = a(x^e)$ modulo $X^{M.N0-1} – 1$ anhand von a(x) zu spezifizieren, und verschiedene Testoperationen werden auf dem mit dem somit definierten Verschachtler assoziierten Turbocode durchgeführt.

[0083] Es sei zum Beispiel $g(x) = 1 + x + x^3$, was N0 = 7 bedeutet. Zudem sei $h_1(x) = 1 + x + x^2 + x^3$, $h_2(x) = 1 + x^2 + x^3$ und M = 21 gewählt, wobei dies zu M.N0 = 147 führt, und es möglich macht, den Zyklus von 2 modulo 147 als {1, 2, 4, 8, 16, 32, 64, 128, 109, 71, 142, 137, 127, 107, 67, 134, 121, 95, 43, 86, 25, 50, 100, 53, 106, 65, 130, 113, 79, 11, 22, 44, 88, 29, 58, 116, 85, 23, 46, 92, 37, 74} zu berechnen.

[0084] Durch sukzessives Testen der durch g(x) teilbaren Polynome a(x) vom Gewicht 2, 3, 4 und 5 wird gefolgert, dass die Wahl e = 25 "aussichtsreich" ist, da die a(x) Polynome vom Gewicht 2 dann einer codierten Sequenz $\underline{v} = (\underline{a}, \underline{b}, \underline{c})$ vom Gewicht ≥ 26 entsprechen, die a(x) Polynome vom Gewicht 3 dann einer codierten Sequenz $\underline{v} = (\underline{a}, \underline{b}, \underline{c})$ vom Gewicht ≥ 24 entsprechen, die a(x) Polynome vom Gewicht 4 dann einer codierten Sequenz $\underline{v} = (\underline{a}, \underline{b}, \underline{c})$ vom Gewicht ≥ 25 entsprechen, die a(x) Polynome vom Gewicht 5 dann einer codierten Sequenz $\underline{v} = (\underline{a}, \underline{b}, \underline{c})$ vom Gewicht 30 entsprechen usw.

[0085] Dies scheint ein Anzeichen für eine minimale Distanz von 24 zu sein, was den besten Wert darstellt, der gemäß dem vorangehend offenbarten Verfahren für N0 ≥ 7 und M = 21 mit dem wie vorangehend angegebenen g(x), $h_1(x)$ und $h_2(x)$ erhalten werden kann.

[0086] Ein weitere mögliche Wahl ist $g(x) = 1 + x + x^4$, was zu N0 = 15 bedeutet. Des weiteren sei $h_1(x) = 1 + x + x^2 + x^4$, $h_2(x) = 1 + x^3 + x^4$ und M = 27 gewählt. Dies führt zu M.N0 = 405 und macht es möglich, den Zyklus von 2 modulo 405 als {1, 2, 4, 8, 16, ..., 304, 203} zu berechnen. Er weist 108 Zahlen auf.

[0087] Durch sukzessive Elimination lässt sich folgern, dass die Wahlen e = 151, e = 362 und e = 233 besonders aussichtsreich sind.

[0088] Die Erfinder haben insbesondere für e = 151 die durch g(x) teilbaren Polynome a(x) mit einem Gewicht sukzessiv gleich 2, 3, 4, 5, 6 und 7 getestet, und derart, dass, falls dieses Gewicht größer oder gleich 5 ist, das als W(a(x)) bezeichnete Gewicht von a(x) gleich dem Gewicht von a(x) modulo $x^{15} + 1$ ist.

[0089] Wenn das Gewicht von a(x) der Ausdruck W(a(x)) ist, ist das entsprechende minimale Gewicht von $W(\underline{v})$ in der Tabelle angegeben:

W(a(x))	W(v) ≥
2	54
3	42
4	44
5	48
6	54
7	54

[0090] Auf die gleiche Weise und unter den gleichen Bedingungen, für e = 362, wenn das Gewicht von a(x) der Ausdruck W(a(x)), ist das entsprechende minimale Gewicht von W(v) in der Tabelle angegeben:

W(a(x))	W(v) ≥
2	54
3	42
4	42
5	50
6	56
7	54

[0091] Gemäß der Erfindung können insbesondere die bei 1, modulo N0, nicht-kongruenten Zahlen e verwendet werden.

[0092] Die Beschreibung eines besonderen Ausführungsbeispiels der vorliegenden Erfindung wird nun mit Bezug auf die Fig. 1 bis Fig. 3 fortgesetzt.

[0093] Fig. 1 veranschaulicht schematisch die Anordnung einer Netzwerkstation oder Computercodierungsstation in Form eines Blockschaltbildes. Diese Station weist eine Tastatur **111**, einen Bildschirm **109**, eine externe Informationsquelle **110** und einen Funksender **106** auf, die gemeinsam mit einem Eingabe-/Ausgabeanschluss **103** einer Verarbeitungskarte **101** verbunden sind.

[0094] Die Verarbeitungskarte **101** weist, miteinander durch einen Adress- und Datenbus **102** verbunden, auf:
- eine zentrale Verarbeitungseinheit **100**,
- einen Speicher mit wahlfreiem Zugriff **104**,
- einen Nur-Lese-Speicher ROM **105**,
- den Eingabe-/Ausgabeanschluss **103**.

[0095] Jede der in Fig. 1 veranschaulichten Komponenten ist für einen Fachmann auf dem Gebiet der Mikrocomputer und Übertragungssysteme, und allgemeiner auf dem Gebiet der Informationsverarbeitungssysteme, wohl bekannt. Diese gebräuchlichen Elemente sind deshalb hier nicht beschrieben. Es sei jedoch angemerkt, dass:
- die Informationsquelle **110** beispielsweise eine Schnittstellenperipherie, ein Sensor, ein Demodulator, ein externer Speicher oder ein anderes Informationsverarbeitungssystem (nicht gezeigt) ist, und vorzugsweise dazu angepasst ist, Sequenzen von Signalen, die Sprache, Dienstnachrichten oder Multimedia-Daten darstellen, in Form von Sequenzen von binären Daten zu liefern,
- der Funksender **106** dazu angepasst ist, ein Paketübertragungsprotokoll auf einem drahtlosem Kanal zu implementieren, und diese Pakete auf einem derartigen Kanal zu übertragen.

[0096] Es sei zudem angemerkt, dass das in der Beschreibung verwendete Wort "Register" in jedem der Speicher **104** und **105** sowohl einen Speicherbereich mit geringer Kapazität (einige binäre Daten), als auch einen Speicherbereich mit großer Kapazität (der das Speichern eines gesamten Programms ermöglicht) bezeichnet.

[0097] Der Speicher mit wahlfreiem Zugriff **104** speichert Daten, Variablen und Verarbeitungs-Zwischenergebnisse in Speicherregistern, die in der Beschreibung die gleichen Namen tragen wie die Daten, deren Werte sie speichern. Der Speicher mit wahlfreiem Zugriff **104** hat insbesondere:
- ein Register "primary_data", in dem in der Reihenfolge ihrer Ankunft auf dem Bus **102** die von der Informationsquelle **110** kommenden binären Daten in der Form einer Sequenz u gespeichert werden, und die später ergänzt werden, um eine Sequenz a zu bilden,
- ein Register "N°_data", das eine der Anzahl binärer Daten n-m in dem Register "binary_data" entsprechende Ganzzahl speichert,
- ein Register "intermediate_remainder", in dem sukzessiv die Zwischen-Reste der Division gespeichert

werden, wobei das Register zur Bildung der Sequenz a anhand der Sequenz u verwendet wird,
– ein Register "final_remainder", in dem die komplementären binären Daten in Form einer Sequenz gespeichert werden,
– ein Register "permuted_data", in dem wie mit Bezug auf Fig. 2 beschrieben die permutierten binären Daten in Form einer Sequenz a* in der Reihenfolge ihrer Ankunft auf dem Bus **102** gespeichert werden, und
– ein Register "a, b, c", in dem die binären Daten der gegenwärtig verarbeiteten Sequenz in der Reihenfolge ihrer Bestimmung durch die zentrale Einheit **100** gespeichert werden.

[0098] Der Nur-Lese-Speicher **105** ist dazu angepasst, in Registern, die der Einfachheit halber die gleichen Namen tragen wie die Daten, die sie speichern, zu speichern:
– das Betriebsprogramm der zentralen Verarbeitungseinheit **100**, in einem Register "program",
– die Sequenz g, in einem Register "g",
– den Grad m von g(x), in einem Register "degree",
– die Sequenz h_1, in einem Register "h_1",
– die Sequenz h_2, in einem Register "h_2",
– den Wert von N0, in einem Register "N0",
– den Wert von M, in einem Register "M", und
– die den Verschachtler definierende Tabelle, in einem Register "interlacer".

[0099] Die zentrale Verarbeitungseinheit **100** ist dazu angepasst, das in Fig. 2 beschriebene Ablaufdiagramm zu implementieren.

[0100] In Fig. 2, die den Betrieb eines wie in Fig. 1 veranschaulichten Codierers darstellt, ist ersichtlich, dass nach einer Initialisierungsoperation **300**, während der die Register des Speichers mit wahlfreiem Zugriff **104** initialisiert werden (N°_data = "0"), während einer Operation **301** die zentrale Einheit **100**, nachdem sie auf das Empfangen gewartet hat, ein zu übertragendes binäres Datenelement empfängt, es in dem Speicher mit wahlfreiem Zugriff **104** in dem Register "primary_data" positioniert, und den Zähler "N°_data" inkrementiert.

[0101] Als Nächstes bestimmt die zentrale Einheit **100** während eines Tests **302**, ob die in dem Register "N°_data" gespeicherte Ganzzahl gleich dem Produkt M.N0, von dem der Grad von g(x) subtrahiert wird, ist oder nicht, wobei M, N0 und der Grad m von g(x) in dem Nur-Lese-Speicher **105** gespeicherte Werte sind.

[0102] Wenn das Ergebnis des Tests **302** negativ ist, wird die Operation **301** wiederholt. Wenn das Ergebnis des Tests **302** positiv ist, wird während einer Operation **303** die Division des mit der in dem Register "primary_data" gespeicherten Sequenz von binären Daten assoziierte Polynoms u(x) durch das Polynom g(x) bis zum letzten Term (des höchsten Grades) von u(x) durchgeführt, wobei zu diesem Zweck das Register "intermediate_remainder" verwendet wird, und der Rest dieser Division wird im Speicher in dem Register "final_remainder" gespeichert.

[0103] Als Nächstes werden während einer Operation **304** die in dem Register "final_remainder" gespeicherten binären Daten der Sequenz u am Ende nebenangestellt, um die Sequenz a zu bilden. Die binären Daten der Sequenz a werden im Speicher in dem Register "a, b, c" gespeichert.

[0104] Als Nächstes wird während einer Operation **305** die während der Operation **303** durchgeführte Division mit den während der Operation **304** hinzugefügten zusätzlichen Daten fortgesetzt, und die Sequenz b wird in dem Register "a, b, c" vervollständigt.

[0105] Dann werden während einer Operation **306** die binären Daten der Sequenz a in der durch die in dem Nur-Lese-Speicher **105** gespeicherten Tabelle "interlacer" beschriebenen Reihenfolge sukzessiv von dem Register "a, b, c" gelesen. Die sukzessiv von diesem Lesen resultierenden Daten werden im Speicher in dem Register "permuted_data" des Nur-Lese-Speichers **104** gespeichert.

[0106] Als Nächstes wird während einer Operation **307** die Division des mit der in dem Register "permuted_data" gespeicherten Sequenz von permutierten binären Daten assoziierten Polynoms a*(x) durch das Polynom g(x) durchgeführt, wobei für diesen Zweck das Register "intermediate_remainder" verwendet wird. Das Ergebnis dieser Division wird im Speicher in dem Register "a, b, c" gespeichert, und entspricht den binären Daten der Sequenz c.

[0107] Während einer Operation **308** werden die Sequenzen b und c durch Berechnen des Produkts der mit den in den Registern "a, b, c" des Speichers mit freiem Wahlzugriff **104** gespeicherten Sequenzen b und c as-

soziierten Polynome bestimmt, und entsprechend die Polynome $h_1(x)$ und $h_2(x)$.

[0108] Es lässt sich beobachten, dass aufgrund der Erfindung Speicherelemente durch Durchführen der Division durch g(x) vor der Multiplikation mit $h_1(x)$ bzw. $h_2(x)$ eingespart werden.

[0109] Während einer Operation **309** werden die Sequenzen a, b und c gesendet, wobei für diesen Zweck der Sender **106** verwendet wird. Als Nächstes werden die Register des Speichers **104** wieder initialisiert, insbesondere wird der Zähler "N°_data" auf "0" zurückgesetzt und die Operation **301** wird wiederholt.

[0110] Als eine Variante lässt sich hier beobachten, dass während der Operation **309** die gesamte Sequenz a, aber nur eine Teilmenge, zum Beispiel von zwei Datenelementen eines, einer jeden der Sequenzen b und c gesendet wird. Diese Variante ist für einen Fachmann als Punktierung bekannt.

[0111] Mit Bezug auf das Decodieren lässt sich beobachten, dass durch Kennen der Polynome g(x), $h_1(x)$, $h_2(x)$ und des anhand der Sequenz a die permutierte Sequenz a* liefernden Verschachtlers ein Fachmann ohne jedwede technische Schwierigkeit weiß, wie der zum Decodieren und zur Korrektur eines beliebigen das Tripel von Sequenzen (a, b, c) betreffenden Fehlers angepasste Decodierer unter Verwendung des vorangehend betrachteten Verschachtlers und, wahlweise, des entsprechenden Entschachtlers zu erzeugen ist.

[0112] Zu diesem Zweck sei verwiesen auf:
– den Artikel von L.R. BAHL, J. COCKE, F. JELINEK und J. RAVIV mit dem Titel "Optimal decoding of linear codes for minimizing symbol error rate", veröffentlicht in dem Journal IEEE Transactions on Information Theory, im März 1974,
– den Artikel von J. HAGENAUER, E. OFFER and L. PAPKE mit dem Titel "Iterative decoding of binary block and convolutional codes", veröffentlicht im Journal IEEE Transactions on Information Theory, im März 1996,
– den Artikel von J. HAGENAUER und P. HOEHER mit dem Titel "A Viterbi algorithm with soft decision outputs and its applications", veröffentlicht mit den Berichten der Konferenz IEEE GLOBECOM, Seiten 1680-1686, im November 1989,
– den Artikel von J. HAGENAUER, P. ROBERTSON und L. PAPKE mit dem Titel "Iterative (turbo)decoding of systematic convolutional codes with the MAP and SOVA algorithms", veröffentlicht durch das Journal Informationstechnische Gesellschaft (ITG) Fachbericht, Seiten 21-29, Oktober 1994, und
– den Artikel von C. BERROU, S. EVANO und G. BATTAIL mit dem Titel "Turbo-block-codes", veröffentlicht mit den Berichten des durch das Technology Institute of Lund (Schweden) (Department of Applied Electronics) organisierten Seminars "Turbo Coding" im August 1996.

[0113] Fig. 3 stellt die Schritte eines Algorithmus dar, der den Wert von e bestimmt, der den zu verwendenden Verschachtler festlegt. Diese Schritte können durch einen Computer eines bekannten Typs (nicht dargestellt) durchgeführt werden, indem sich die Register "N0", "M", "interlacer", "d", "d_{max}", "e" und "j" in dem Speicher mit wahlfreiem Zugriff befinden.

[0114] Während einer Operation **501** wird die kleinste strikt positive Ganzzahl N0 gesucht, sodass g(x) das Polynom $x^{N0} - 1$ teilt, wobei $g(x) = 1 + \Sigma_{i=1 \text{ bis } m-1}\, g_i \cdot x^i + x^m$ das vorbestimmte Polynom vom Grad m ist, das der Sequenz g entspricht. Es ist bekannt, dass diese Zahl existiert. Zum Beispiel ist, für $g(x) = 1 + x + x^3$, N0 = 7. Zu diesem Zweck werden die Divisionen der Polynome $x^i - 1$ durch g(x) sukzessiv durchgeführt, beginnend mit einem Wert von i, der gleich dem Grad m von g(x) ist, und dann schrittweise i inkrementierend mit einem Inkrementierungsschritt von 1, bis der Rest der Division Null ist, modulo 2. Wenn der Rest Null ist, wird der Wert von i in das Register N0 abgelegt. Es sei daran erinnert, dass die hier durchgeführte Division modulo 2 auf den Koeffizienten der aufsteigenden Potenzen von $x^i - 1$ durchgeführt wird.

[0115] Dann wird unter Wählen einer derartigen ungeraden Zahl M, sodass das Produkt M.N0 größer oder gleich der Anzahl von in dem gleichen Rahmen zu übertragenden binären Daten u_i ist, addiert zum Grad m von g(x), während einer Operation **502** eine Länge der Sequenz a gleich M.N0 gewählt, was der Bestimmung der Länge (das heißt der Anzahl von binären Daten) der in der Sequenz a umfassten Sequenz u als M.N0 minus dem Grad m von g(x) entspricht.

[0116] Dann bestimmt die zentrale Einheit **100** während den Operationen **503** bis **509**, ob der mit e assoziierte Verschachtler berücksichtigt werden muss, was bedeutet, dass es keine Sequenz a mit niedrigem Gewicht gibt, für die die Sequenz v = (a, b, c) ebenfalls ein niedriges Gewicht hat.

[0117] In dem beschriebenen und dargestellten Ausführungsbeispiel besteht die Bestimmung von a* aus ei-

ner Ersetzung von $\underline{a} = (a_0, a_1, ...)$ durch $\underline{a}^* = (a_0, a_f, a_{2f}, ...)$, wobei die Vielfachen von f modulo M.N0 berechnet werden. Wenn f gleich einer von 1 unterschiedlichen Potenz von 2 ist, modulo M.N0, ist diese Permutation in der Tat von dem angegebenen Typ. Sie kann in der Tat dargestellt werden durch eine Permutation, die binäre Daten nur innerhalb jeder Spalte der Tabelle permutiert, gefolgt von, wenn f gleich einer von 1 verschiedenen Potenz von 2 ist, modulo N0, einer Permutation von zumindest zwei der Spalten miteinander, wobei diese letzte Permutation ein Automorphismus des binären zyklischen Codes der Länge N0 und mit einem Generatorpolynom g(x) ist. Wenn f gleich 1 ist, modulo N0, ist diese sich auf die Spalten beziehende Permutation die "triviale" Permutation oder "Identitäts-" Permutation, das heißt sie behält die Position der Spalten in der Tabelle bei.

[0118] In diesem besonderen Ausführungsbeispiel der vorliegenden Erfindung bestimmt die zentrale Einheit **100** zu diesem Zweck während der Operation **503** die sukzessiven Potenzen von 2 modulo M.N0, um den so genannten Zyklus von 2 modulo M.N0 zu bestimmen, wobei dieser Zyklus vollständig ist, sobald eine der Potenzen von 2 gleich 1 ist, modulo M.N0. Die Anzahl von Termen j dieses Zyklus wird in dem Register "j" gespeichert.

[0119] Dann werden während der Operation **504** die in den Registern "l" und "d_{max}" gespeicherten Zwischenwerte l und d_{max} mit dem Wert "1" beziehungsweise dem Wert "0" initialisiert.

[0120] Als Nächstes wird während einer Operation **505** der Wert von "l" um 1 inkrementiert, und der l-te Wert des Zyklus von 2 modulo M.N0 wird genommen.

[0121] Dann wird während einer Operation **506**, falls dieser Wert nicht gleich 1 modulo M.N0 ist, das Gewicht der Sequenz $\underline{v} = (\underline{a}, \underline{b}, \underline{c})$ für die Sequenzen $\underline{a}$ mit niedrigem Gewicht bestimmt, wobei die Permutation definiert ist durch $a^*(x) = a(x^e)$ (somit, falls $\underline{a} = (a_0, a_1, a_2, ..., a_{M.N0-1})$, ist das erste binäre Datum von $a^* = a_0$, das zweite a_f, das dritte a_{2f}, ..., wie vorangehend offenbart, wobei der Index modulo M.N0 berechnet wird).

[0122] Da die Distanz zwischen zwei Sequenzen das Gewicht (das heißt die Anzahl von von null verschiedenen binären Daten) der durch die Differenz der homologen binären Daten dieser Sequenzen gebildeten Sequenz ist, ist der Prozess zu diesem Zweck beschränkt auf die Analyse der Distanz der Sequenzen mit der Null-Sequenz, wobei die Polynome a(x) nach aufsteigendem Gewicht aufgezählt werden, die Summe der Gewichte der Sequenzen in ein und demselben Tripel $(\underline{a}, \underline{b}, \underline{c})$ gemessen wird, und eine Suche nach dem minimalen Gewicht für ein gegebenes e durchgeführt wird, für die Sequenzen $\underline{a}$ mit niedrigem Gewicht, und, nachdem all diese minimalen Gewichte für e in dem Zyklus von 2 bestimmt wurden, für den Wert von e, der dem höchsten Gewicht entspricht.

[0123] Die Distanz wird dann in dem Register "d" des Speichers mit wahlfreiem Zugriff **104** gespeichert. Als Nächstes wird, falls während eines Tests **507** der in dem Register "d" gespeicherte Wert größer als der in dem Register "d_{max}" gespeicherte Wert ist, während der Operation **508** der Wert des Registers "d_{max}" modifiziert, um den Wert d anzunehmen, und der Wert des l-ten Elements des betrachteten Zyklus wird im Speicher in dem Register "e" gespeichert.

[0124] Nach der Operation **508**, oder wenn das Ergebnis des Tests **507** negativ ist, wird, solange der Wert von l kleiner als j ist, Test **509**, die Operation **505** wiederholt.

[0125] Die Tabelle "interlacer" wird dann wie folgt gebildet:
$a^*(x) = a(x^e)$, und somit, falls $\underline{a} = (a_0, a_1, a_2, ..., a_{M.N0-1})$, ist das erste binäre Datenelement von $a^* = a_0$, das zweite a_f, das dritte a_{2f}, ..., wie vorangehend offenbart, wobei der Index modulo M.N0 berechnet wird.

[0126] Gemäß einer nicht gezeigten Variante wird das Tripel $(\underline{a}, \underline{b}, \underline{c})$ wie folgt konstruiert:
– a(x) und $b(x) = a(x) \cdot h_1(x)/g(x)$ sind wie vorangehend definiert,
– mit einem gegebenen Polynom $g_2(x)$, das derart gewählt ist, sodass die kleinste Ganzzahl N2, für die $g_2(x)$ das Polynom $x^{N2} - 1$ teilt, gleich der kleinsten Ganzzahl N0 ist, für die g(x) das Polynom $x^{N0} - 1$ teilt, wird eine Permutation P gewählt, die ein beliebiges Wort des binären zyklischen Codes der Länge N0 und mit einem Generatorpolynom g(x) in ein Wort des binären zyklischen Codes der Länge N2 und mit einem Generatorpolynom $g_2(x)$ transformiert. Es lässt sich beobachten, dass eine derartige Permutation nur für die Polynome $g_2(x)$ existiert, die zu Cg äquivalente zyklische Codes generieren. Diese Verifikation ist für einen Fachmann wohlbekannt, und als Referenz sei auf Seite 234 des vorangehend erwähnten Buches von F.J. MACWILLIAMS und N.J.A. SLOANE hingewiesen,
– gemäß der Erfindung wird dann die Permutation, die aus der mit dem durch g(x) teilbaren Polynom a(x) assoziierten Sequenz $\underline{a}$, die mit dem durch $g_2(x)$ teilbaren Polynom a**(x) assoziierte Sequenz $\underline{a}$** erzeugt,

durch eine beliebige Permutation erzeugt, die aus a(x) eine wie vorangehend erklärt durch g(x) teilbare erste Sequenz a(x) erzeugt, gefolgt von einer gerade eben eingeführten Permutation P, die auf den Spalten der zunächst alle a und dann a* enthaltenen Tabelle mit M Zeilen und N0 Spalten arbeitet, um diese Spalten miteinander zu permutieren und a** zu erzeugen.

[0127] Der Rahmen der Erfindung ist nicht auf die beschriebenen und dargestellten Ausführungsbeispiele begrenzt, sondern erstreckt sich im Gegenteil auf jedwede Modifikation und Verbesserung innerhalb der Fähigkeiten eines Fachmanns.

[0128] Insbesondere wird ein Durchfluss bei Durchsätzen von einem Viertel, oder weniger, durch Hinzufügen von einem oder mehreren zusätzlichen Verschachtlern, für jeden Verschachtler durch Anwenden der vorangehend dargelegten Prinzipien bewerkstelligt. In all diesen Fällen kann eine Punktierung verwendet werden, um den Durchsatz des Codes zu erhöhen. Es sei daran erinnert, dass eine Punktierung darin besteht, dass nur einige der Prüfsymbole übertragen werden.

[0129] Zudem werden Vorrichtungen, die die Aufgabe der vorliegenden Erfindung sind, zum Ausführen der arithmetischen Berechnungen, Polynommultiplikation, Polynomdivision, der Verschachtelungsfunktion und der Funktionen zum elementaren Decodieren zweckmäßigerweise durch Implementieren dedizierter Schaltkreise produziert, die keinen Prozessor aufweisen (ein derartiger Prozessor kann trotzdem zur Steuerung des Betriebs dieser Vorrichtungen verwendet werden). Die Verwendung derartiger dedizierter Schaltkreise ermöglicht das Erreichen höherer Informationsflussraten.

[0130] Es sei angemerkt, dass in dem ersten Ausführungsbeispiel der Durchsatz ohne Punktierung nahe einem Drittel ist, da für eine zu codierende Sequenz mit n-m Symbolen zwei Sequenzen mit n Prüfsymbolen vorhanden sind.

II – ZWEITES AUSFÜHRUNGSBEISPIEL

[0131] Im zweiten Ausführungsbeispiel der vorliegenden Erfindung werden Durchsätze größer 1/3 betrachtet, die ohne Punktierung erreicht werden: für zwei zu codierende Sequenzen mit n-m Symbolen werden zum Beispiel zwei Sequenzen mit n Prüfsymbolen zugeführt, das bedeutet ein Durchsatz nahe ein halb.

[0132] Vor dem Beginn der Beschreibung des zweiten Ausführungsbeispiels werden nachfolgend die mathematischen Grundlagen seiner Implementierung angegeben.

[0133] Um das erste Ausführungsbeispiel einzuführen, wird eine mit Turbocodierern mit einem Durchsatz nahe einem Drittel zu verwendende Klasse algebraischer Verschachtler aufgezeigt. Der wesentliche Vorteil dieser Verschachtler ist die Einhaltung der Teilbarkeit des Informationspolynoms durch ein gegebenes Polynom g(x), das den Codierer teilweise charakterisiert. Ein Ergebnis ist die bessere Unabhängigkeit der Fehlerwahrscheinlichkeit pro codiertem Informationsbit von der Position des Bits in der Informationssequenz. Als weiteren Vorteil macht die algebraische Beschreibung dieser Verschachtler ihre Aufzählung und individuelle Evaluierung möglich. Zudem wird erwartet, dass die Performanz dieser kleinen Menge von Verschachtlern maßgeblich für die Performanz aller Verschachtler ist.

[0134] Unter vielen Umständen, wie beispielsweise im Zusammenhang mit drahtloser Übertragung, ist jedoch ein besserer Durchsatz notwendig.

[0135] Das zweite Ausführungsbeispiel betrifft insbesondere Durchsätze von mehr als oder nahe ein halb, ohne die Verwendung von Punktierungsverfahren.

IIa – Turbocodes mit hohem Durchsatz mit Verschachtlern vom "x-zu-x^e"-Typ

[0136] Der folgende systematische Faltungscodierer K × (K + 2) wird betrachtet:

$$G = \begin{matrix} 1 & 0 & \dots & 0 & h_1/g & f_1/g \\ 0 & 1 & \dots & 0 & h_2/g & f_2/g \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & h_K/g & f_K/g \end{matrix} \qquad (1)$$

in dem h_i, f_i und g Polynome mit binären Koeffizienten der den Verzögerungsoperator darstellenden Unbekann-

ten x sind.

[0137] Die übertragene Information wird in einer üblichen Weise durch ein K-Tupel $\underline{a} = (a_1, ..., a_K)$ von Polynomen $a_i = \Sigma_{j=0 \text{ bis } n-1}\ a_{ij} \cdot x^j$ dargestellt, mit binären Koeffizienten a_{ij}, und die Information wird in $\underline{v} = \underline{a}G$ codiert, was ein (K + 2)-Tupel von Sequenzen der Unbekannten x ist:

$$\underline{v} = [a_1, ..., a_K, (\Sigma_{i=1 \text{ bis } K}\ a_i h_i)/g, (\Sigma_{i=1 \text{ bis } K}\ a_i f_i)/g]$$

[0138] Es sei darauf hingewiesen, dass hier "Sequenzen" anstatt "Polynome" geschrieben wird, da die Summen nicht notwendigerweise Vielfache von g sind, und, falls sie es nicht sind, eine Division durch g eine Sequenz unendlich und letztlich periodisch macht.

[0139] Um die Performanz dieses Typs von Codierer zu verbessern, kann die letzte Komponente $(\Sigma_{i=1 \text{ bis } K}\ a_i \cdot f_i)/g$ von $\underline{v}$ durch $(\Sigma_{i=1 \text{ bis } K}\ a_i^* h_i)/g$ ersetzt werden, wobei a_i^* die aus der Sequenz a_i durch Permutation ihrer Koeffizienten erhaltene Sequenz darstellt. Die Transformation bei jeder Sequenz a_i in die Sequenz a_i^* wird als "Verschachtelung" bzw. "Interlacing" bezeichnet (vgl. zum Beispiel den Artikel von C. BERROU und A. GLAVIEUX mit dem Titel "Near Optimum error-correcting coding and decoding: turbocodes", veröffentlicht durch IEEE Transactions on Communication, Band COM-44, Seiten 1261 bis 1271, im Oktober 1996).

[0140] Fig. 4 gibt eine Veranschaulichung eines Codierers, der diese Operation durchführt. In dieser Figur lässt sich beobachten, dass der Codierer für K Sequenzen von Symbolen am Eingang am Ausgang sendet:

– diese K Sequenzen identischer Symbole,
– eine Sequenz von durch Bilden der Summe der Produkte der mit den Informationssequenzen a_i assoziierten Polynome und der vorbestimmten Polynome h_i und Dividieren dieser Summe durch ein vorbestimmtes Polynom g gebildeten Prüfsymbolen (Codierer **401**),
– eine Sequenz von Prüfsymbolen, die gebildet werden durch zunächst Verschachteln einer jeden Informationssequenz a_i durch einen Verschachtler I_i (Verschachtler **402** bis **405**), um eine Sequenz a_i^* bereitzustellen, dann Bilden der Summe der Produkte der mit den Sequenzen a_i^* assoziierten Polynome und vorbestimmter Polynome f_i, und Dividieren dieser Summe durch ein vorbestimmtes Polynom g (Codierer **406**).

[0141] Hier sei angemerkt, dass die in Fig. 1 veranschaulichte Menge von Verschachtlern I_i **402** bis **406** bereits eine Beschränkung des Verschachtlers **201** von Fig. 5 darstellt, in der jede Sequenz a_i^* Symbole der Sequenz a_j enthalten kann, wobei j ungleich i ist.

[0142] Gemäß den allgemeinen Merkmalen der vorliegenden Erfindung weist jeder Verschachtler I_i in einer Darstellung, in der die binären Daten einer jeden Sequenz a_i in einer Tabelle mit N0 Spalten und M Zeilen eingeordnet sind, auf:

– zumindest eine Permutation in einer Menge von Permutationen, die einerseits die Automorphismen des binären zyklischen Codes der Länge N0 und mit einem Generatorpolynom g(x), die zumindest zwei der N0 Spalten der Tabelle miteinander permutieren, umfassen, und andererseits die ausschließlich auf den Daten in der gleichen Spalte arbeitenden und zumindest zwei der Daten miteinander permutierenden Permutationen umfassen, und
– keine Permutation außerhalb der Menge.

[0143] Gemäß den besonderen Merkmalen der vorliegenden Erfindung werden auf Codierer mit einem Durchsatz von KI(K + 2), wobei K eine beliebige positive Ganzzahl darstellt, die Verschachtler des Typs "x-zu-x^e" angewendet, der mit Bezug auf das den Wert K = 1 betreffende erste Ausführungsbeispiel beschrieben wurde.

[0144] Nachfolgend wird nun eine Beschreibung des Falls K ≥ 2 (zweites Ausführungsbeispiel) gegeben.

[0145] Es sei ein Polynom $g(x) = 1 + g_1 x + g_2 x^2 + ... + g_{m-1}\ x^{m-1} + x^m$.

[0146] Es sei N0 die kleinste Ganzzahl, sodass g(x) den Ausdruck $x^{N0} + 1$ teilt.

[0147] Und schließlich sei n gleich einem ungeraden Vielfachen von N0: n = M.N0.

[0148] Das Polynom g(x) ist dann ein Teiler des Polynoms $x^n + 1$. Zum Beispiel kann, mit $g(x) = 1 + x + x^4$, n aus den Werten 15, 45, 75, ..., 255, ..., 405, ... gewählt werden. Andere diese Teilbarkeitseigenschaften betreffende Einzelheiten sind in der Arbeit von W.W. PETERSON und E.J. WELDON mit dem Titel "Error-correcting

codes", veröffentlicht durch MIT Press, Cambridge, Massachussets, 1972, zu finden.

[0149] Die Information wird durch eine Sequenz $\underline{u} = (\underline{u}_1, ..., u_K)$ dargestellt, in der jede der K Komponenten $\underline{u}_i$ durch ein Polynom $\underline{u}_i(x)$ vom formalen Grad n-m-1 mit binären Koeffizienten dargestellt wird. Jedem Polynom $\underline{u}_i(x)$ wird ein Abschluss $\Sigma_{j=n-m \text{ bis } n-1}(u_i)_j\, x^j$ hinzugefügt, ein Abschluss mit m Bits, der das Polynom $a_i = u_i + \Sigma_{j=n-m \text{ bis } n-1}(u_i)_j\, x^j$ durch g(x) teilbar macht.

[0150] Das sich ergebende K-Tripel $\underline{a} = (\underline{a}_1, ..., \underline{a}_K)$ wird dann codiert, um zwei Prüfsequenzen zu erzeugen.

[0151] Die erste ist durch $\Sigma_{i=1 \text{ bis } K}\, a_i h_i/g$ gegeben und ist ein Polynom, weil das Polynom g(x) die K Polynome a_i teilt. Die zweite ist durch $\Sigma_{i=1 \text{ bis } K}\, a_i{}^* f_i/g$ gegeben, wobei die Verschachtelung von $a_i(x)$ in $a_i{}^*(x)$ gegeben ist durch:

$$a_i{}^*(x) = a_i(x^e) \text{ modulo } x^n + 1 \qquad (2)$$

wobei e gleich einer Potenz von 2 ($e = 2^i$), modulo n genommen, ist. Dieser Typ einer "x-zu-x^e"-Permutation wurde im K = 1 betreffenden ersten Teil dargelegt und garantiert die Teilbarkeit von $a_i{}^*$ durch g, wenn a_i durch g teilbar ist.

[0152] Nun werden einige Beispiele mit K = 2 gegeben.

[0153] Es wird ein 2 × 4 Turbocodierer der in (1) beschriebenen und durch $g(x) = 1 + x + x^3$, $h_1(x) = f_2(x) = 1 + x^2 + x^3$, $h_2(x) = f_1(x) = 1 + x + x^2 + x^3$ und n = 147 festgelegten Form betrachtet.

[0154] Mit diesem Wert für n, 147, gibt es 42 Zahlen e, die jeweils der Rest, modulo 197, einer Potenz von 2 sind. Für jeden dieser Werte für e wurde der entsprechende Turbocode auf einem additiven weißen Gauß'schen Störkanal (für den Fachmann unter dem für "Additive White Gaussian Noise" stehenden englischen Akronym AWGN bekannt) für verschiedene Werte des Verhältnisses zwischen der Energie pro Bit, E_b, und der Rauschleistung pro Hertz (auch als spektrale Rauschleistungsdichte bezeichnet), N, simuliert.

[0155] Die entsprechenden Werte der Kurven der Fehlerraten pro Bit ("BER") als eine Funktion des Signal/Rausch-Verhältnisses sind in **Fig. 6** dargestellt, für n = 147, K = 2 und für drei verschiedene Werte für e: $e = 67 = 2^{14}$, $e = 32 = 2^5$ und $e = 71 = 2^9$. Die ersten zwei Werte stellen "gute" Werte für e unter den 42 möglichen Werten dar, und der letzte Wert für e stellt einige "weniger gute" Werte dar: 1, 2, 4, 109, 142 und 50.

[0156] Mit den gleichen g(x), $h_i(x)$, $f_i(x)$, für i = 1 oder 2, aber mit einem höheren Wert für n, können die gleichen Simulationen durchgeführt werden. Die Fälle n = 413 und n = 917 wurde ausgewählt, weil sie eine große Anzahl von Werten verschiedener Potenzen von 2, modulo n, bieten. Für n = 413 gibt es tatsächlich 174 derartiger verschiedener Werte, und für n = 917 gibt es 390 derartiger verschiedener Werte.

[0157] Die Erfinder haben beobachtet, dass mit der Implementierung der vorliegenden Erfindung keine Stufe in der Fehlerkurve für einen Wert der Fehlerwahrscheinlichkeit pro Bit nach Decodierung von ungefähr 10^{-5} aufzutreten scheint, wenn der Wert für e vernünftig gewählt wird. Es sei zudem angemerkt, dass die Verschachtelung der $a_i(x)$ Polynome zum Erzeugen von $a_i{}^*(x) = a_i(x^e)$ modulo $x^n + 1$ für jeden Wert von i durchgeführt werden kann, mit einem unterschiedlichen Wert für e, aber immer mit der Form einer l-ten Potenz von 2: $e = 2^1$. Verzugsweise werden die gleichen Verschachtler für alle Werte von i verwendet, was die Identität der k-Tupel von Symbolen einhält.

IIb – Beschreibung der Turbocodierer mit K ≥ 2

[0158] Beispielhaft wird der folgende 3 × 5 Turbocodierer betrachtet:

$$G = \begin{matrix} 1 & 0 & 0 & h_1/g & f_1/g \\ 0 & 1 & 0 & h_2/g & f_2/g \\ 0 & 0 & 1 & h_3/g & f_3/g \end{matrix}$$

wobei g ein irreduzibles Polynom mit binären Koeffizienten ist, das nicht durch ein quadratisches Polynom teilbar ist. Diese Matrix G codiert ein Informations-Tripel $\underline{u} = (\underline{u}_1, \underline{u}_2, \underline{u}_3)$ als:

$v = [a_1, a_2, a_3, (\Sigma a_i h_i)/g, (\Sigma a_i{}^* f_i)/g]$

wobei man a_i aus u_i und a_i^* aus a_i erhält, wie mit Bezug auf das erste Ausführungsbeispiel offenbart wurde.

[0159] Ein Element, das die minimalen Distanzen eines derartigen Codes beeinflusst, ist die minimale Anzahl von von Null verschiedenen Komponenten des 5-Tupels v.

[0160] Es ist möglich zu garantieren, dass ein beliebiges 5-Tupel $v = (v_1, ..., v_5)$ des Turbocodes zumindest drei von Null verschiedene Sequenzen v_i enthält.

[0161] Um dies zu demonstrieren, wird a_i^* explizit als $a_i(x^e)$ (reduziert modulo $x^n + 1$) geschrieben, und die Inverse von e modulo n durch "d" dargestellt:
ed = 1 (n), eine Gleichung in der "(n)" die Bedeutung von "modulo n" hat.

[0162] Es ist sodann zu beobachten, dass die Gleichung $\Sigma a_i(x^e)f_i(x) = 0$ modulo $(x^n + 1)$ äquivalent ist zur Gleichung $\Sigma a_i(x)f_i(x^d) = 0$ modulo $(x^n + 1)$.

[0163] Dies folgt aus der Tatsache, dass e und d relativ prim mit n sind, sodass man für jedes b(x) von einem Grad kleiner oder gleich n – 1 die modulo $(x^n + 1)$ genommenen Ausdrücke $b(x^e)$ und $b(x^d)$ einfach durch Permutation der Koeffizienten von b(x) erhält, und sodass dies, wenn das Polynom $a(x^e)$ modulo $(x^n + 1)$ als $[a(x)]^{\Pi(e)}$ bezeichnet wird, ergibt:

$[a(x)f(x)]^{\Pi(e)} = [a(x)]^{\Pi(e)} [f(x)]^{\Pi(e)}$ modulo $(x^n + 1)$.

[0164] Aus der Code-Theorie resultierend ist als "MDS" (was für "Maximum Distance Separable" steht) bekannt, dass ein beliebiges von Null verschiedenes v in dem durch G generierten Code stets zumindest drei von Null verschiedene Komponenten enthält, falls die folgenden zwei Bedingungen erfüllt sind:

– keines der Polynome h_i und f_j ist Null modulo $x^{N0} + 1$, wobei N0 die kleinste Ganzzahl ist, sodass g(x) den Ausdruck $x^{N0} + 1$ teilt, wobei N0 ungerade ist, da das Polynom g(x) durch kein polynomisches Quadrat teilbar ist, und

– keine Matrix 2 × 2 der Form

$h_i(x)$ $f_i(x^d)$

$h_j(x)$ $f_j(x^d)$

mit i ungleich j, eine Determinante aufweist die Null ist, modulo $x^{N0} + 1$.

[0165] Eine ähnliche Eigenschaft gilt für einen beliebigen Codierer eines Turbocodes der Dimension K und Länge N1, mit einer Anzahl von Redundanzen M1 = N1 – K > 2. Im letzteren Fall gleicht der Codierer:

$$G = \begin{matrix} 1 & 0 & \dots & 0 & h_{1,1}/g & h_{1,2}/g & \dots & h_{1,N1-K}/g \\ 0 & 1 & \dots & 0 & h_{2,1}/g & h_{2,2}/g & \dots & h_{1,N1-K}/g \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & h_{K,1}/g & h_{K,2}/g & \dots & h_{K,N1-K}/g \end{matrix}$$

und eine beliebige Informationssequenz $\underline{u} = (\underline{u}_1, ..., \underline{u}_K)$ wird zunächst codiert als $\underline{a} = (\underline{a}_1, ..., \underline{a}K)$, und dann als:

$v = [a_1, ..., a_K, (\Sigma a_i h_{i,1})/g, (\Sigma a_i(x^{e2})h_{i,2})/g, ..., (\Sigma a_i(x^{eN0-K})h_{i,K})/g]$

[0166] Es sei d_j die Inverse von e_j modulo n: $d_j e_j = 1(n)$. Jedes von Null verschiedene v in dem durch G generierten Code enthält immer zumindest N1 – K + 1 von Null verschiedene Komponenten, falls für jede Ganzzahl r, für die $1 \le r \le N1 – K$ gilt, jede Unter-Matrix r × r von G, extrahiert aus seinen zumindest N1 – K Spalten und mit $h_{i,j}(x)$ ersetzt durch $h_{i,j}(x^{dj})$, eine von Null verschiedene Determinante aufweist, modulo $x^{N0} + 1$.

[0167] Hierbei sei angemerkt, dass die Decodierung der Redundanz-Turbocodes N1 – K > 2 bereits betrachtet wurde. Zum Beispiel kann der Artikel von D. DIVSALAR und F. POLLARA mit dem Titel "Multiple Turbocodes for Deep Space Communications", TDA Progress Report 42-121, vom 15. Mai 1995, herangezogen werden.

[0168] Nun wird ein Verfahren zur Auswahl guter Kandidaten zum Erhalten eines guten Turbocodes der Dimensionen K und Länge N1 mit Verschachtlern des Typs "x-zu-x^e" beschrieben.

[0169] Zunächst wird der Fall betrachtet, in dem N1 – K = 2 ist.

[0170] Ein irreduzibles Polynom g(x) auf GF(2), vom Grad m ≥ 2, sodass N0 = 2^m – 1 ≥ N1, wird gewählt, und GF(2^m) wird genommen als die Menge von Polynomresten modulo g(x)·n wird ebenfalls als ein ungerades Vielfaches von N0 gewählt. Dann wird eine Matrix Γ mit 2 Zeilen und N1 Spalten konstruiert:

$$\Gamma = \begin{matrix} \alpha_1^{\,r} & \alpha_2^{\,r} & \cdots & \alpha_{N1}^{\,r} \\ \alpha_1^{\,s} & \alpha_2^{\,s} & \cdots & \alpha_{N1}^{\,s} \end{matrix}$$

wobei α_i, mit i = 1, ..., N1 verschiedene von Null verschiedene Elemente von GF(2^m) sind, und r und s sich modulo N0 unterscheidende Ganzzahlen sind. Die Wahl s = r + 1 ist zum Beispiel immer eine gute Wahl. Dies impliziert, dass alles unter der Matrix Γ nicht-singulär ist. Es sei Γ(1, 2) die Unter-Matrix der ersten zwei Spalten von Γ, $[\Gamma(1,2)]^{-1}$ Γ lässt sich wie folgt schreiben:

$$[\Gamma(1,2)]^{-1}\ \Gamma = \begin{matrix} 1 & 0 & \\ & & \Lambda \\ 0 & 1 & \end{matrix}$$

wobei Λ eine 2 × K Matrix auf GF(2^m) ist. Für einen gewählten Wert für e Potenz von 2 modulo n wird jedes Element in der zweiten Zeile von Λ ersetzt durch seine e-te Potenz, und die transponierte Matrix dieser Matrix wird als Δ bezeichnet. Die Elemente von Δ werden dann als Polynome vom Grad m – 1 der Unbekannten x interpretiert, und durch g(x) oder durch ein beliebiges anderes Teilerpolynom von x^{N0} + 1 geteilt, und die resultierende Matrix wird als P(x) bezeichnet.

[0171] Schließlich wird die Matrix G der Dimension K × (K + 2) definiert durch:

$$G = [I_K\ P(x)]$$

wobei I_K die Einheitsmatrix der Dimension K ist.

[0172] Für jeden Wert von e kann dieser Codierer G als ein Turbocodierer verwendet werden. Durch Simulation ist es dann möglich, den besten Wert für e zu wählen.

[0173] Der Fall N1 – K > 2 kann auf eine ähnliche Weise behandelt werden. In diesem Fall, mit M1 = N1 – K, wird eine Matrix Γ vom Typ M1 × N1 konstruiert:

$$\Gamma = \begin{matrix} \alpha_1^{\,r} & \alpha_2^{\,r} & \cdots & \alpha_{N1}^{\,r} \\ \alpha_1^{\,r+1} & \alpha_2^{\,r+1} & \cdots & \alpha_{N1}^{\,r+1} \\ \cdots & \cdots & \cdots & \cdots \\ \alpha_1^{\,r+M1-1} & \alpha_2^{\,r+M1-1} & \cdots & \alpha_{N1}^{\,r+M1-1} \end{matrix}$$

[0174] Mit Γ(1, M1) als die Unter-Matrix der ersten M1 Spalten von Γ lässt sich $[\Gamma(1, M1)]^{-1}$ Γ als $[I_{M1}\Lambda]$ schreiben, wobei I_{M1} die Einheitsmatrix der Dimension M1 und Λ eine Unter-Matrix auf GF(2^m) der Dimension M1 × K ist. Für gewählte Werte $e_t = 2^{it}$, für t = 2, ..., M1, werden die Elemente der t-ten Zeile von Λ durch ihre e_t-te Potenz ersetzt.

[0175] Die transponierte Matrix dieser Matrix wird dann als Δ bezeichnet, die Elemente von Δ werden als Polynome vom Grad m – 1 der Unbekannten x interpretiert, sie werden durch g(x) dividiert, und die resultierende Matrix der Dimensionen K × (N – K) wird als P(x) bezeichnet.

[0176] Die für verschiedene Werte des (M1 – 1)-Tupels (e_2, ..., e_{M1}) erhaltenen Codierer G = $[I_K\ P(x)]$ werden dann durch Simulationen analysiert, und die besten können ausgewählt werden.

IIc – Beschreibung einer Implementierung des zweiten Ausführungsbeispiels der vorliegenden Erfindung

[0177] Die Beschreibung des zweiten Ausführungsbeispiels der vorliegenden Erfindung wird nun mit Bezug auf die Fig. 7 und Fig. 8 für K = M1 = 2 fortgesetzt.

[0178] Fig. 7 veranschaulicht schematisch die Anordnung einer Netzwerkstation oder Computercodierungsstation in Form eines Blockschaltbildes. Diese Station weist eine Tastatur **711**, einen Bildschirm **709**, eine ex-

terne Informationsquelle **710** und einen Funksender **706** auf, die gemeinsam mit einem Eingabe-/Ausgabeanschluss **703** einer Verarbeitungskarte **701** verbunden sind.

[0179] Die Verarbeitungskarte **701** weist, miteinander durch einen Adress- und Datenbus **702** verbunden, auf:
- eine zentrale Verarbeitungseinheit **700**,
- einen Speicher mit wahlfreiem Zugriff **704**,
- einen Nur-Lese-Speicher ROM **705**,
- den Eingabe-/Ausgabeanschluss **703**.

[0180] Jedes der in Fig. 7 veranschaulichten Elemente ist für einen Fachmann auf dem Gebiet der Mikrocomputer und Übertragungssysteme, und allgemeiner auf dem Gebiet der Informationsverarbeitungssysteme, wohl bekannt. Diese gebräuchlichen Elemente sind deshalb hier nicht beschrieben. Es sei jedoch angemerkt, dass:
- die Informationsquelle **710** beispielsweise eine Schnittstellenperipherie, ein Sensor, ein Demodulator, ein externer Speicher oder ein anderes Informationsverarbeitungssystem (nicht gezeigt) ist, und vorzugsweise dazu angepasst ist, Sequenzen von Signalen, die Sprache, Dienstnachrichten oder Multimedia-Daten darstellen, in Form von Sequenzen von binären Daten zu liefern,
- der Funksender **706** dazu angepasst ist, ein Paketübertragungsprotokoll auf einem drahtlosem Kanal zu implementieren, und diese Pakete auf einem derartigen Kanal zu übertragen.

[0181] Es sei zudem angemerkt, dass das in der Beschreibung verwendete Wort "Register" in jedem der Speicher **704** und **705** sowohl einen Speicherbereich mit geringer Kapazität (einige binäre Daten), als auch einen Speicherbereich mit großer Kapazität (der das Speichern eines gesamten Programms ermöglicht) bezeichnet.

[0182] Der Speicher mit wahlfreiem Zugriff **704** speichert Daten, Variablen und Verarbeitungs-Zwischenergebnisse in Speicherregistern, die in der Beschreibung die gleichen Namen tragen wie die Daten, deren Werte sie speichern. Der Speicher mit wahlfreiem Zugriff **704** enthält insbesondere:
- ein Register "primary_data", in dem in der Reihenfolge ihrer Ankunft auf dem Bus **702** die von der Informationsquelle **710** kommenden binären Daten in der Form zweier Sequenzen $\underline{u}_1$ und $\underline{u}_2$ gespeichert werden, und die später ergänzt werden, um zwei Sequenzen $\underline{a}_1$ und $\underline{a}_2$ zu bilden,
- ein Register "N°_data", das eine der Anzahl binärer Daten n-m in dem Register "binary_data" entsprechende Ganzzahl speichert,
- ein Register "permuted_data", in dem in der Reihenfolge ihrer Übertragung auf dem Bus **702** die permutierten binären Daten, wie mit Bezug auf Fig. 8 beschrieben, in der Form zweier Sequenzen $\underline{a}_1^*$ und $\underline{a}_2^*$ gespeichert werden,
- ein Register "intermediate_remainder", in dem sukzessiv die Zwischen-Reste der Division gespeichert werden, wobei das Register zur Bildung einer jeden Sequenz $\underline{a}_i$ anhand der entsprechenden Sequenz $\underline{u}_i$ verwendet wird,
- ein Register "final_remainder", in dem die komplementären binären Daten in Form zweier Sequenzen gespeichert werden, und
- Register "$\underline{a}_1$, $\underline{b}_1$, $\underline{c}_1$" und "$\underline{a}_2$, $\underline{b}_2$, $\underline{c}_2$", in denen die binären Daten der Sequenzen in der Reihenfolge ihrer Bestimmung durch die zentrale Einheit **700** gespeichert werden.

[0183] Der Nur-Lese-Speicher **705** ist dazu angepasst, in Registern, die der Einfachheit halber die gleichen Namen tragen wie die Daten, die sie speichern, zu speichern:
- das Betriebsprogramm der zentralen Verarbeitungseinheit **700**, in einem Register "program",
- die Sequenz $\underline{g}$, in einem Register "$\underline{g}$",
- den Grad m von g(x), in einem Register "degree",
- die Sequenz $\underline{h} = \underline{h}_1 = \underline{f}_2$, in dem Register "$\underline{h}$",
- die Sequenz $\underline{f} = \underline{f}_1 = \underline{h}_2$, in dem Register "$\underline{f}$",
- den Wert von N0, in einem Register "N0",
- den Wert von N1, in einem Register "N1",
- den Wert von M1, in einem Register "M1",
- den Wert von M, in einem Register "M",
- die den auf der Sequenz a_1 arbeitenden Verschachtler definierende Tabelle, in einem Register "interlacer1", und
- die den auf der Sequenz a_2 arbeitenden Verschachtler definierende Tabelle, in einem Register "interlacer2".

[0184] Die hier betrachteten Verschachtler sind vom vorangehend beschriebenen Typ "x-zu-x^{e1}" bzw.

"x-zu-x^{e2}".

[0185] Die zentrale Verarbeitungseinheit **700** ist dazu angepasst, das in Fig. 8 beschriebene Ablaufdiagramm zu implementieren.

[0186] In Fig. 8, die den Betrieb eines wie in Fig. 7 veranschaulichten Codierers darstellt, ist ersichtlich, dass nach einer Initialisierungsoperation **800**, während der die Register des Speichers mit wahlfreiem Zugriff **704** initialisiert werden (N°_data = "0"), während einer Operation **801** die zentrale Einheit **700**, nachdem sie auf das Empfangen gewartet hat, ein zu übertragendes binäres Datenelement empfängt, es in dem Speicher mit wahlfreiem Zugriff **704** in dem Register "primary_data" positioniert, und den Zähler "N°_data" inkrementiert.

[0187] Hier sei angemerkt, dass die zentrale Einheit **700** zur Bildung der zwei Sequenzen $\underline{u}_1$ und $\underline{u}_2$ zunächst die Sequenz $\underline{u}_1$ und dann die Sequenz $\underline{u}_2$ bildet, wobei die primären Daten von der Informationsquelle **710** kommen.

[0188] Als Nächstes bestimmt die zentrale Einheit **700** während eines Tests **802**, ob die in dem Register "N°_data" gespeicherte Ganzzahl gleich dem Produkt M.N0, von dem der Grad m von g(x) subtrahiert wird, ist oder nicht, wobei M, N0 und der Grad m von g(x) in dem Speicher mit wahlfreiem Zugriff **705** gespeicherte Werte sind.

[0189] Wenn das Ergebnis des Tests **802** negativ ist, wird die Operation **801** wiederholt. Wenn das Ergebnis des Tests **802** positiv ist, werden während einer Operation **803** die Divisionen der mit den in dem Register "primary_data" gespeicherten Sequenzen von binären Daten $\underline{u}_1$ bzw. $\underline{u}_2$ assoziierten Polynome $u_1(x)$ und $u_2(x)$ durch das Polynom g(x) bis zu den letzten Termen (der höchsten Grade) von $u_1(x)$ und $u_2(x)$ durchgeführt, wobei zu diesem Zweck das Register "intermediate_remainder" verwendet wird. Die Reste dieser Divisionen werden im Speicher in dem Register "final_remainder" gespeichert. Die Ergebnisse dieser Divisionen stellen die ersten Elemente der Sequenzen $\underline{b}_1 = \underline{a}_1/g$ und $\underline{b}_2 = \underline{a}_2/g$ bereit.

[0190] Als Nächstes werden während einer Operation **804** die in dem Register "final_remainder" gespeicherten binären Daten den Sequenzen $\underline{u}_1$ bzw. $\underline{u}_2$ am Ende nebenangestellt, um die Sequenzen $\underline{a}_1$ und $\underline{a}_2$ zu bilden. Die binären Daten der Sequenzen $\underline{a}_1$ und $\underline{a}_2$ werden im Speicher in den Registern "$\underline{a}_1$, $\underline{b}_1$, $\underline{c}_1$" und "$\underline{a}_2$, $\underline{b}_2$, $\underline{c}_2$" gespeichert.

[0191] Als Nächstes werden während einer Operation **805** die während der Operation **803** durchgeführten Divisionen mit den während der Operation **804** hinzugefügten zusätzlichen Daten fortgesetzt, und die Sequenzen $\underline{b}_1$ und $\underline{b}_2$ werden in den Registern "$\underline{a}_1$, $\underline{b}_1$, $\underline{c}_1$" und "$\underline{a}_2$, $\underline{b}_2$, $\underline{c}_2$" vervollständigt.

[0192] Dann werden während einer Operation **806**:
– die binären Daten der Sequenz $\underline{a}_1$ in der durch die in dem Speicher mit wahlfreiem Zugriff **705** gespeicherten Tabelle "interlacer1" beschriebenen Reihenfolge jeweils sukzessiv in das Register "$\underline{a}_1$, $\underline{b}_1$, $\underline{c}_1$" gelesen, und
– die binären Daten der Sequenz $\underline{a}_2$ in der durch die in dem Speicher mit wahlfreiem Zugriff **705** gespeicherten Tabelle "interlacer2" beschriebenen Reihenfolge jeweils sukzessiv in das Register "$\underline{a}_2$, $\underline{b}_2$, $\underline{c}_2$" gelesen.

[0193] Die sukzessiv von diesen Lesevorgängen resultierenden Daten werden jeweils im Speicher in dem Register "permuted_data" des Speichers mit wahlfreiem Zugriff **704** gespeichert.

[0194] Als Nächstes werden während einer Operation **807** die Divisionen der jeweils mit den in dem Register "permuted_data" gespeicherten Sequenzen von permutierten binären Daten assoziierten Polynome $a_1^*(x)$ und $a_2^*(x)$ durch das Polynom g(x) durchgeführt, wobei für diesen Zweck das Register "intermediate_remainder" verwendet wird. Die Ergebnisse dieser Divisionen werden im Speicher in den Registern "$\underline{a}_1$, $\underline{b}_1$, $\underline{c}_1$" und "$\underline{a}_2$, $\underline{b}_2$, $\underline{c}_2$" gespeichert, und entsprechen den binären Daten der Sequenzen $\underline{c}_1$ und $\underline{c}_2$.

[0195] Während einer Operation **808** werden zwei so genannte "Prüf-" Sequenzen bzw. "redundante" Sequenzen bestimmt:
– die Sequenz $\underline{b}_s$ wird durch Berechnen des Produkts der mit den in den Registern "$\underline{a}_1$, $\underline{b}_1$, $\underline{c}_1$" und "$\underline{a}_2$, $\underline{b}_2$, $\underline{c}_2$" in dem Speicher mit freiem Wahlzugriff **704** gespeicherten Sequenzen $\underline{b}_1$ und $\underline{b}_2$ assoziierten Polynome bestimmt, und entsprechend die Polynome h(x) und f(x),
– die Sequenz $\underline{c}_s$ wird durch Berechnen des Produkts der mit den in den Registern "$\underline{a}_1$, $\underline{b}_1$, $\underline{c}_1$" und "$\underline{a}_2$, $\underline{b}_2$, $\underline{c}_2$" in dem Speicher mit freiem Wahlzugriff **704** gespeicherten Sequenzen $\underline{c}_1$ und $\underline{c}_2$ assoziierten Polynome

bestimmt, und entsprechend die Polynome f(x) und h(x).

[0196] Es sei darauf hingewiesen, dass aufgrund der Erfindung Speicherelemente durch Durchführen der Division durch g(x) vor den Multiplikationen mit h(x) bzw. f(x) eingespart werden.

[0197] Während einer Operation **809** werden die Sequenzen $\underline{a}_1$, $\underline{a}_2$, $\underline{b}_s$ und $\underline{c}_s$ übertragen, wobei für diesen Zweck der Sender **706** verwendet wird. Als Nächstes werden die Register des Speichers **704** wieder initialisiert, insbesondere wird der Zähler "N°_data" auf "0" zurückgesetzt und die Operation **801** wird wiederholt.

[0198] Als eine Variante sei hier angemerkt, dass während der Operation **809** die Sequenzen $\underline{a}_1$ und $\underline{a}_2$ als Ganzes gesendet werden, aber nur eine Teilmenge, zum Beispiel von zwei Datenelementen eines, einer jeden der Sequenzen $\underline{b}_s$ und $\underline{c}_s$ gesendet wird. Diese Variante ist für einen Fachmann als Punktierung bekannt.

[0199] Mit Bezug auf das Decodieren sei angemerkt, dass durch Kennen der Polynome g(x), h(x), f(x) und der anhand der Sequenzen $\underline{a}_1$ und $\underline{a}_2$ die permutierten Sequenzen $\underline{a}_1$* bzw. $\underline{a}_2$* liefernden Verschachtler G_1 und G_2 ein Fachmann ohne jedwede technische Schwierigkeit weiß, wie der zum Decodieren und zur Korrektur eines beliebigen das Quadrupel von Sequenzen ($\underline{a}_1$, $\underline{a}_2$, $\underline{b}_s$, $\underline{c}_s$) betreffenden Fehlers angepasste Decodierer unter Verwendung der vorangehend betrachteten Verschachtler und, möglicherweise, der entsprechenden Entschachtler zu erzeugen ist.

[0200] In Fig. 9 ist zu erkennen, dass eine zum Decodieren der durch die in den Fig. 4 bis Fig. 8 veranschaulichte Decodierungsvorrichtung gesendeten Sequenzen angepasste Decodierungsvorrichtung im Wesentlichen aufweist:

– einen dem Codierer **401** entsprechenden Decodierer **901**, der die Schätzungen von übertragenen Sequenzen, $\underline{v}_1$ bis $\underline{v}_{K+1}$, als auch nachfolgend offenbarte K extrinsische Informationssequenzen $\underline{w'''}_1$ bis $\underline{w'''}_K$ empfängt, und K Schätzsequenzen a posteriori $\underline{w}_1$ bis $\underline{w}_K$ bereitstellt,

– K Verschachtler **902**, die zu den im Codierer verwendeten Verschachtlern **402** bis **405** identisch sind, und die jeweils die Sequenzen $\underline{w}_1$ bis $\underline{w}_K$ empfangen und dann jeweils als $\underline{w'}_1$ bis $\underline{w'}_K$ verschachteln,

– einen zweiten Decodierer **903**, der dem sowohl die Sequenzen $\underline{w'}_1$ bis $\underline{w'}_K$ als auch die Sequenzen $\underline{v}_{K+2}$ empfangenden Codierer **406** entspricht, und einerseits K a posteriori Schätz-Sequenzen $\underline{w''}_1$ bis $\underline{w''}_K$, und andererseits eine geschätzte Sequenz $\underline{a}'$ bereitstellt, und

– K Entschachtler **904**, die Umkehreinrichtungen der Verschachtler **402** bis **405**, die die Sequenzen $\underline{w''}_1$ bis $\underline{w''}_K$ empfangen und die Sequenzen $\underline{w'''}_1$ bis $\underline{w'''}_K$ bereitstellen.

[0201] Die geschätzte Sequenz $\underline{a}'$ wird nur einer vorbestimmten Anzahl von Iterationen folgend berücksichtigt (vgl. den vorangehend zitierten Artikel "Near Shannon limit error-correcting coding and decoding: turbocodes").

[0202] Gemäß der vorliegenden Erfindung haben die zum Decodieren verwendeten Verschachtler und Entschachtler jeweils die gleichen Merkmale wie die zum Codieren verwendeten Verschachtler, und sind vorzugsweise vom Typ "x-zu-x^e". Sowohl zum Codieren als auch zum Decodieren sind die Exponenten e_{ij} für einen identischen Wert von j vorzugsweise gleich. Sowohl zum Codieren als auch zum Decodieren sind die Werte der Exponenten e_{ij} auch vorzugsweise alle Potenzen von 2.

[0203] Zudem wird beim Initialisieren der Decodierer **901** und **903** die Tatsache berücksichtigt, dass die Codierer **401** und **406** jeweils einen initialen Zustand und einen finalen Null-Zustand haben.

[0204] Bezüglich dem Decodieren sei verwiesen auf:

– den Artikel von L.R. BAHL, J. COCKE, F. JELINEK und J. RAVIV mit dem Titel "Optimal decoding of linear codes for minimizing symbol error rate", veröffentlicht in dem Journal IEEE Transactions on Information Theory, im März 1974,

– den Artikel von J. HAGENAUER, E. OFFER and L. PAPKE mit dem Titel "Iterative decoding of binary block and convolutional codes", veröffentlicht im Journal IEEE Transactions on Information Theory, im März 1996,

– den Artikel von J. HAGENAUER und P. HOEHER mit dem Titel "A Viterbi algorithm with soft decision outputs and its applications", veröffentlicht mit den Berichten der Konferenz IEEE GLOBECOM, Seiten 1680-1686, im November 1989,

– den Artikel von J. HAGENAUER, P. ROBERTSON und L. PAPKE mit dem Titel "Iterative (turbo)decoding of systematic convolutional codes with the MAP and SOVA algorithms", veröffentlicht durch das Journal Informationstechnische Gesellschaft (ITG) Fachbericht, Seiten 21-29, Oktober 1994, und

– den Artikel von C. BERROU, S. EVANO und G. BATTAIL mit dem Titel "Turbo-block-codes", veröffentlicht mit den Berichten des durch das Technology Institute of Lund (Schweden) (Department of Applied Electro-

nics) organisierten Seminars "Turbo Coding" im August 1996.

Patentansprüche

1. Verschachtelungsverfahren (**300** bis **309**), das eine Permutation verwendet, welche ein Teil einer Menge von Permutationen ist, die aus einer Sequenz $\underline{a}$ von binären Daten, die eine physikalische Größe darstellt und durch ein Polynom a(x) dargestellt wird, das durch ein Teilerpolynom g(x) teilbar ist, eine permutierte Sequenz $\underline{a}$* (**306**) liefern, die durch ein Polynom a*(x) dargestellt wird, wobei das Polynom a*(x) durch das Polynom g(x) teilbar ist (**307**), um eine Sequenz $\underline{c}$ von binären Daten zu bilden, in der die binären Daten der Sequenz $\underline{a}$ in einer Tabelle von N0 Spalten und M Zeilen angeordnet sind, wobei N0 die kleinste Ganzzahl ist, so dass x^{N0} + 1 durch das Polynom g(x) teilbar ist, wobei die Menge von Permutationen die Permutationen umfasst, die ausschließlich auf Datenelemente von ein und derselben Spalte wirken und zumindest zwei der Datenelemente miteinander permutieren, **dadurch gekennzeichnet**, dass das Verschachtelungsverfahren zumindest eine Permutation verwendet, die aus den Automorphismen von einem binären zyklischen Code einer Länge N0 mit einem Generatorpolynom ausgewählt wird, das äquivalent ist zu g(x) und zumindest zwei der N0 Spalten der Tabelle miteinander permutiert.

2. Codierungsverfahren (**800** bis **809**), das:
1) berücksichtigt:
– eine vorbestimmte Ganzzahl M1, die gleich oder größer 2 ist,
– eine Zahl K, die größer oder gleich 1 ist, von Sequenzen $\underline{a}_i$(i = 1, ..., K) von binären Daten, die eine physikalische Größe darstellen, wobei jede Sequenz $\underline{a}_i$ aufweist:
• eine Polynomdarstellung $a_i(x)$, die ein Vielfaches von einem vorbestimmten Polynom $g_i(x)$ ist, und
• eine Zahl von binären Daten, die gleich dem Produkt einer beliebigen Ganzzahl M und der Ganzzahl N0 ist, wobei die kleinste Ganzzahl derart ist, dass das Polynom x^{N0} + 1 durch jedes der Polynome $g_i(x)$ teilbar ist;
dadurch gekennzeichnet, dass:
2) das Verfahren eine erste Operation (**806**) zum Erzeugen einer Zahl K*M1 von so genannten „permutierten" Sequenzen $\underline{a}_{ij}$*(i = 1, ..., K; j = 1, ..., M1) umfasst, wobei jede Sequenz $\underline{a}_{ij}$*:
– durch eine Permutation der entsprechenden Sequenz $\underline{a}_i$ erhalten wird, wobei die Permutation in einer Darstellung, bei der die binären Daten von jeder Sequenz $\underline{a}_i$ zeilenweise in einer Tabelle von N0 Spalten und M Zeilen geschrieben werden, die Resultante von einer beliebigen Zahl von so genannten elementaren Permutationen ist, die jeweils eine Eigenschaft aufweisen zum Transformieren eines zyklischen Codes einer Länge N0 und mit einem Generatorpolynom, das äquivalent ist zu $g_i(x)$, in einen äquivalenten zyklischen Code mit einem Generatorpolynom $g_{ij}(x)$, das gleich $g_i(x)$ sein kann und zumindest zwei der N0 Spalten miteinander permutiert,
– folglich eine Polynomdarstellung a_{ij}*(x) aufweist, die gleich einem Polynomprodukt $c_{ij}(x)g_{ij}(x)$ ist, wobei $c_{ij}(x)$ der Quotient von a_{ij}*(x) durch $g_{ij}(x)$ ist,
– wobei zumindest eine permutierte Sequenz $\underline{a}_{ij}$* verschieden ist von der entsprechenden Sequenz $\underline{a}_i$,
3) das Verfahren eine zweite Operation (**808**) zum Erzeugen von M1 redundanten Sequenzen umfasst, deren Polynomdarstellung für j = 1, ... M1 gleich $\Sigma f_{ij}(x)c_{ij}(x)$ ist, wobei jedes Polynom $f_{ij}(x)$ ein Polynom mit einem Grad ist, der höchstens gleich dem Grad des Polynoms $g_{ij}(x)$ mit den gleichen Indizes i und j ist.

3. Codierungsverfahren gemäß Anspruch 2, dadurch gekennzeichnet, dass das Polynom $g_{ij}(x)$ der Generator des kleinsten zyklischen Codes einer Länge N0 ist, der das Polynom $g_i(x^{eij})$ modulo (x^{N0+1}) enthält, und jede Sequenz a_{ij}* eine Polynomdarstellung aufweist, die gleich a_{ij}*(x) = $a_i(x^{eij})$ modulo $(x^n + 1)$ ist, wobei:
– n das Produkt der Zahl M und der Ganzzahl N0 ist,
– e_{ij} und n teilerfremde Zahlen sind.

4. Codierungsverfahren gemäß Anspruch 3, dadurch gekennzeichnet, dass während der ersten Erzeugungsoperation (**806**) alle Werte der Exponenten e_{ij} mit dem gleichen Wert des Index j identisch sind.

5. Codierungsverfahren gemäß Anspruch 3 oder 4, dadurch gekennzeichnet, dass während der ersten Erzeugungsoperation (**806**) alle Werte der Exponenten e_{ij} gleich einer Potenz von 2 sind.

6. Codierungsverfahren gemäß Anspruch 2, 3 oder 4, dadurch gekennzeichnet, dass es eine Operation zum Übertragen (**809**) einerseits von Sequenzen $\underline{a}_i$ und andererseits von einer Untermenge von Daten der anderen Sequenzen umfasst.

7. Verschachtler (**100**, **101**, **104**, **105**) mit einer Permutationseinrichtung (**100**, **104**, **105**), die eine Permutation verwendet (**306**), die ein Teil einer Menge von Permutationen ist, die aus einer Sequenz $\underline{a}$ von binären

Daten, die eine physikalische Größe darstellt und durch ein Polynom a(x) dargestellt wird, das durch ein Teilerpolynom g(x) teilbar ist, eine permutierte Sequenz $\underline{a}^*$ liefern, die durch ein Polynom a*(x) dargestellt wird, wobei das Polynom a*(x) durch das Polynom g(x) teilbar ist (**307**), um eine Sequenz $\underline{c}$ von binären Daten zu bilden (**308**), in der die binären Daten der Sequenz $\underline{a}$ in einer Tabelle von N0 Spalten und M Zeilen angeordnet sind, wobei N0 die kleinste Ganzzahl ist, so dass $x^{N0} + 1$ durch das Polynom g(x) teilbar ist, wobei die Menge von Permutationen die Permutationen umfasst, die ausschließlich auf Datenelemente vor ein und derselben Spalte wirken werden und zumindest zwei der Datenelemente miteinander permutieren, dadurch gekennzeichnet, dass die Permutationseinrichtung zumindest eine Permutation verwendet, die aus den Automorphismen von einem binären zyklischen Code einer Länge N0 mit einem Generatorpolynom ausgewählt wird, das äquivalent ist zu g(x) und zumindest zwei der N0 Spalten der Tabelle miteinander permutiert.

8. Codierungsvorrichtung (**700**, **701**, **704**, **705**), die eine Verarbeitungseinrichtung (**700**, **704**, **705**) umfasst, die angepasst ist zum:
1) Berücksichtigen von:
– einer vorbestimmten Ganzzahl M1, die gleich oder größer 2 ist,
– einer Zahl K, die größer oder gleich 1 ist, von Sequenzen $\underline{a}_i$(i = 1, ...; K) von binären Daten, die eine physikalische Größe darstellen, wobei jede Sequenz $\underline{a}_i$ aufweist:
• eine Polynomdarstellung $a_i(x)$, die ein Vielfaches von einem vorbestimmten Polynom $g_i(x)$ ist, und
• eine Zahl von binären Daten, die gleich dem Produkt einer beliebigen Ganzzahl M und der Ganzzahl N0 ist, wobei die kleinste Ganzzahl derart ist, dass das Polynom $x^{N0} + 1$ durch jedes der Polynome $g_i(x)$ teilbar ist;
dadurch gekennzeichnet, dass die Verarbeitungseinrichtung angepasst ist zum:
2) Erzeugen (**806**) einer Zahl von K*M1 von so genannten „permutierten" Sequenzen $\underline{a}_{ij}^*$(i = 1, ..., K; j = 1, ..., M1), wobei jede Sequenz $\underline{a}_{ij}^*$:
– durch eine Permutation der entsprechenden Sequenz $\underline{a}_i$ erhalten wird, wobei die Permutation in einer Darstellung, bei der die binären Daten von jeder Sequenz $\underline{a}_i$ zeilenweise in einer Tabelle von N0 Spalten und M Zeilen geschrieben werden, die Resultante von einer beliebigen Zahl von so genannten elementaren Permutationen ist, die jeweils eine Eigenschaft aufweisen zum Transformieren eines zyklischen Codes einer Länge N0 und mit einem Generatorpolynom, das äquivalent ist zu $g_i(x)$, in einen äquivalenten zyklischen Code mit einem Generatorpolynom $g_{ij}(x)$, das gleich $g_i(x)$ sein kann und zumindest zwei der N0 Spalten der Tabelle miteinander permutiert,
– folglich eine Polynomdarstellung $a_{ij}^*(x)$ aufweist, die gleich einem Polynomprodukt $c_{ij}(x)g_{ij}(x)$ ist, wobei $c_{ij}(x)$ der Quotient von $a_{ij}^*(x)$ durch $g_{ij}(x)$ ist,
– wobei zumindest eine permutierte Sequenz $\underline{a}_{ij}^*$ verschieden ist von der entsprechenden Sequenz $\underline{a}_i$,
3) Erzeugen (**808**) von M1 redundanten Sequenzen, deren Polynomdarstellung für j = 1, ..., M1 gleich $\Sigma f_{ij}(x)c_{ij}(x)$ ist, wobei jedes Polynom $f_{ij}(x)$ ein Polynom mit einem Grad ist, der höchstens dem Grad des Polynoms $g_{ij}(x)$ mit den gleichen Indizes i und j ist.

9. Codierungsvorrichtung gemäß Anspruch 8, dadurch gekennzeichnet, dass das Polynom $g_{ij}(x)$ ein Generator des kleinsten zyklischen Codes einer Länge N0 ist, der das Polynom $g_i(x^{eij})$ modulo (x^{N0+1}) enthält, und jede Sequenz a_{ij}^* eine Polynomdarstellung aufweist, die gleich $a_{ij}^*(x) = a_i(x^{eij})$ modulo $(x^n + 1)$ ist, wobei:
– n das Produkt der Zahl M und der Ganzzahl N0 ist,
– e_{ij} und n teilerfremde Zahlen sind.

10. Codierungsvorrichtung gemäß Anspruch 9, dadurch gekennzeichnet, dass die Verarbeitungseinrichtung (**700**, **704**, **705**) angepasst ist zum Verwenden der Exponenten e_{ij}, die identisch sind, wenn sie den gleichen Wert des Index j aufweisen.

11. Codierungsvorrichtung gemäß Anspruch 9 oder 10, dadurch gekennzeichnet, dass die Verarbeitungseinrichtung (**700**, **704**, **705**) angepasst ist zum Verwenden der Exponenten e_{ij}, die jeweils einen Wert aufweisen, der gleich einer Potenz von 2 ist.

12. Codierungsvorrichtung gemäß Anspruch 9, 10 oder 11, dadurch gekennzeichnet, dass sie eine Übertragungseinrichtung (**703**, **706**) aufweist, die zum Übertragen einerseits von den Sequenzen $\underline{a}_i$ und andererseits von einer Untermenge der Daten der anderen Sequenzen angepasst ist.

13. Decodierungsverfahren, das:
1) berücksichtigt:
– eine vorbestimmte Ganzzahl M1, die gleich oder größer 2 ist,
– eine Zahl K, die größer oder gleich 1 ist, von Sequenzen $\underline{a}_i$(i = 1, ..., K) von binären Daten, die eine physikalische Größe darstellen, wobei jede Sequenz $\underline{a}_i$ aufweist:

• eine Polynomdarstellung $a_i(x)$, die ein Vielfaches von einem vorbestimmten Polynom $g_i(x)$ ist, und
• eine Zahl von binären Daten, die gleich dem Produkt einer beliebigen Ganzzahl M und der Ganzzahl N0 ist, wobei die kleinste Ganzzahl derart ist, dass das Polynom $x^{N0} + 1$ durch jedes der Polynome $g_i(x)$ teilbar ist;
dadurch gekennzeichnet, dass:
2) das Verfahren eine Operation paralleler Turbodecodierung umfasst, die für jede $\underline{a}_i$ M1 Permutationsoperationen umfasst, die so genannte „permutierte" Sequenzen $\underline{a}_{ij}$* erzeugen, wobei zumindest eine Permutationsoperation nicht Identität ist, wobei jede Permutation in einer Darstellung, bei der die binären Daten von jeder Sequenz $\underline{a}_i$ zeilenweise in einer Tabelle mit N0 Spalten und M Zeilen geschrieben werden, die Resultante von einer beliebigen Zahl von so genannten elementaren Permutationen ist, die jeweils eine Eigenschaft aufweisen zum Transformieren eines zyklischen Codes einer Länge N0 und mit einem Generatorpolynom, das äquivalent ist zu $g_i(x)$, in einen äquivalenten zyklischen Code mit einem Generatorpolynom $g_{ij}(x)$, das gleich $g_i(x)$ sein kann und zumindest zwei der N0 Spalten der Tabelle miteinander permutiert, so dass a_{ij}*(x) durch $g_{ij}(x)$ teilbar ist, wobei die Operation paralleler Turbodecodierung K Sequenzen von Symbolen unter Verwendung der Generatorpolynome $g_{ij}(x)$ decodiert.

14. Decodierungsverfahren gemäß Anspruch 13, dadurch gekennzeichnet, dass das Polynom $g_{ij}(x)$ ein Generator des kleinsten zyklischen Codes einer Länge N0 ist, der das Polynom $g_i(x^{eij})$ modulo (x^{N0+1}) enthält, und jede Sequenz $\underline{a}_{ij}$* eine Polynomdarstellung aufweist, die gleich a_{ij}*(x) = $a_i(x^{eij})$ modulo $(x^n + 1)$ ist, wobei:
– n das Produkt der Zahl M und der Ganzzahl N0 ist,
– e_{ij} und n teilerfremde Zahlen sind.

15. Decodierungsverfahren gemäß Anspruch 14, dadurch gekennzeichnet, dass während der Permutationsoperation alle Werte der Exponenten e_{ij} mit dem gleichen Wert des Index j identisch sind.

16. Decodierungsverfahren gemäß Anspruch 14 oder 15, dadurch gekennzeichnet, dass während der Permutationsoperation alle Werte der Exponenten e_{ij} gleich einer Potenz von 2 sind.

17. Decodierungsverfahren gemäß einem der Ansprüche 13 bis 16, dadurch gekennzeichnet, dass es eine Operation zum Empfangen einerseits von Sequenzen $\underline{a}_i$ und andererseits von einer Untermenge der Daten von anderen Sequenzen umfasst, die aus einer Codierung der Sequenzen $\underline{a}_i$ resultieren.

18. Verschachtelungsverfahren (**300** bis **309**), das aus einer Sequenz $\underline{a}$ von binären Daten, die eine physikalische Größe darstellt und mit einem Polynom a(x) in Zusammenhang steht, das durch ein Teilerpolynom g(x) teilbar ist, und dessen Koeffizienten aufsteigender Reihenfolge die binären Daten der Sequenz $\underline{a}$ sind, eine permutierte Sequenz $\underline{a}$** liefert, die mit einem Polynom a**(x) in Zusammenhang steht, dessen Koeffizienten aufsteigender Reihenfolge die binären Daten der Sequenz $\underline{a}$** sind, wobei das Polynom a**(x) dazu bestimmt ist, durch ein Teilerpolynom $g_2(x)$ geteilt zu werden (**307**), um eine Sequenz $\underline{c}$ von binären Daten zu bilden (**308**), und wobei $\underline{a}$ eine Zahl von binären Daten aufweist, die gleich dem Produkt einer beliebigen Ganzzahl M und der Ganzzahl N0 ist, wobei N0 die kleinste Ganzzahl ist, so dass $x^{N0} - 1$ durch das Polynom g(x) teilbar ist,
dadurch gekennzeichnet, dass das Verfahren in einer Darstellung, bei der die binären Daten der Sequenz $\underline{a}$ in einer Tabelle von N0 Spalten und M Zeilen angeordnet sind, verwendet:
– zumindest eine Permutation (**306**), die aus den Automorphismen eines binären zyklischen Codes einer Länge N0 mit einem Generatorpolynom ausgewählt wird, das äquivalent ist zu g(x) und zumindest zwei der N0 Spalten der Tabelle miteinander permutiert, wodurch eine Sequenz a*(x) erzeugt wird, die durch g(x) teilbar ist, und
– eine Permutation der Spalten der Tabelle, die das Polynom a*(x) in das Polynom a**(x) transformiert, das durch das Polynom $g_2(x)$ teilbar ist.

19. Verschachtelungsverfahren gemäß Anspruch 18, dadurch gekennzeichnet, dass die Permutationsoperation (**306**) wie folgt bewirkt wird:
$a^*(x) = a(x^e)$ modulo $x^{M.N0} - 1$,
wobei e ein Ganzzahlenwert gleich einer Potenz von 2 modulo M.N0 und M eine ungerade Zahl ist.

20. Verschachtelungsverfahren gemäß Anspruch 18 oder 19, dadurch gekennzeichnet, dass es eine Operation zum Bestimmen von zumindest einer zweiten Sequenz, die durch Implementierung eines Verfahrens gemäß Anspruch 18 oder 19 permutiert wird, aus der Sequenz $\underline{a}$ von binären Daten umfasst.

21. Verschachtelungsverfahren gemäß Anspruch 18, 19 oder 20, dadurch gekennzeichnet, dass es eine Operation (**309**) zum Übertragen einerseits von der Sequenz $\underline{a}$ und andererseits von einer Untermenge der Daten von den anderen Sequenzen umfasst.

22. Decodierungsvorrichtung, die eine Verarbeitungseinrichtung umfasst, die angepasst ist zum:
1) Berücksichtigen von:
– einer vorbestimmten Ganzzahl M1, die gleich oder größer 2 ist,
– einer Zahl K, die größer oder gleich 1 ist, von Sequenzen $\underline{a}_i$(i = 1, ...; K) von binären Daten, die eine physikalische Größe darstellen, wobei jede Sequenz $\underline{a}_i$ aufweist:
• eine Polynomdarstellung $a_i(x)$, die ein Vielfaches von einem vorbestimmten Polynom $g_i(x)$ ist, und
• eine Zahl von binären Daten, die gleich dem Produkt einer beliebigen Ganzzahl M und der Ganzzahl N0 ist, wobei die kleinste Ganzzahl derart ist, dass das Polynom $x^{N0} + 1$ durch jedes der Polynome $g_i(x)$ teilbar ist;
dadurch gekennzeichnet, dass die Verarbeitungseinrichtung angepasst ist zum:
2) Durchführen einer Operation paralleler Turbodecodierung, die für jede $\underline{a}_i$ M1 Permutationsoperationen umfasst, die so genannte permutierte Sequenzen $\underline{a}_{ij}$* erzeugen, wobei zumindest eine Permutationsoperation nicht Identität ist, wobei jede Permutation in einer Darstellung, bei der die binären Daten von jeder Sequenz a_i zeilenweise in einer Tabelle mit N0 Spalten und M Zeilen geschrieben werden, die Resultante von einer beliebigen Zahl von so genannten elementaren Permutationen ist, die jeweils eine Eigenschaft aufweisen zum Transformieren eines zyklischen Codes einer Länge N0 und mit einem Generatorpolynom, das äquivalent ist zu $g_i(x)$, in einen äquivalenten zyklischen Code mit einem Generatorpolynom $g_{ij}(x)$, das gleich $g_i(x)$ sein kann und zumindest zwei der N0 Spalten der Tabelle miteinander permutiert, so dass a_{ij}*(x) durch $g_{ij}(x)$teilbar ist, wobei die Operation paralleler Turbodecodierung von K Sequenzen von Symbolen die Generatorpolynome $g_{ij}(x)$ verwendet.

23. Decodierungsvorrichtung gemäß Anspruch 22, dadurch gekennzeichnet, dass das Polynom $g_{ij}(x)$ ein Generator des kleinsten zyklischen Codes einer Länge N0 ist, der das Polynom $g_i(x^{eij})$ modulo (x^{N0+1}) enthält, und jede Sequenz $\underline{a}_{ij}$* eine Polynomdarstellung aufweist, die gleich a_{ij}*(x) = $a_i(x^{eij})$ modulo $(x^n + 1)$ ist, wobei:
– n das Produkt der Zahl M und der Ganzzahl N0 ist,
– e_{ij} und n teilerfremde Zahlen sind.

24. Decodierungsvorrichtung gemäß Anspruch 23, dadurch gekennzeichnet, dass die Verarbeitungseinrichtung angepasst ist zum Verwenden von Werten der Exponenten e_{ij}, die identisch sind, wenn die Exponenten den gleichen Wert des Index j aufweisen.

25. Decodierungsvorrichtung gemäß Anspruch 23 oder 24, dadurch gekennzeichnet, dass die Verarbeitungseinrichtung angepasst ist zum Verwenden von Werten der Exponenten e_{ij}, die jeweils gleich einer Potenz von 2 sind.

26. Decodierungsvorrichtung gemäß einem der Ansprüche 22 bis 25, dadurch gekennzeichnet, dass sie eine Empfangeinrichtung umfasst, die angepasst ist zum Empfangen einerseits von Sequenzen $\underline{a}_i$ und andererseits von einer Untermenge der Daten von anderen Sequenzen, die aus einer Codierung der Sequenzen $\underline{a}_i$ resultieren.

27. Verschachtler (**101**), der angepasst ist zum Liefern, aus einer Sequenz $\underline{a}$ von binären Daten, die eine physikalische Größe darstellt und mit einem Polynom a(x) in Zusammenhang steht, das durch ein Teilerpolynom g(x) teilbar ist, und dessen Koeffizienten aufsteigender Reihenfolge die binären Daten der Sequenz $\underline{a}$ sind, einer permutierten Sequenz $\underline{a}$**, die mit einem Polynom a**(x) in Zusammenhang steht, dessen Koeffizienten aufsteigender Reihenfolge die binären Daten der Sequenz $\underline{a}$** sind, wobei das Polynom a**(x) dazu bestimmt ist, durch ein Teilerpolynom $g_2(x)$ geteilt zu werden, um eine Sequenz $\underline{c}$ von binären Daten zu bilden, und wobei $\underline{a}$ eine Zahl von binären Daten aufweist, die gleich dem Produkt einer beliebigen Ganzzahl M und der Ganzzahl N0 ist, wobei N0 die kleinste Ganzzahl ist, so dass $x^{N0} – 1$ durch das Polynom g(x) teilbar ist,
dadurch gekennzeichnet, dass der Verschachtler eine Permutationseinrichtung aufweist, die angepasst ist, in einer Darstellung, bei der die binären Daten der Sequenz $\underline{a}$ in einer Tabelle von N0 Spalten und M Zeilen angeordnet sind, zu implementieren:
– zumindest eine Permutation (**306**), die aus den Automorphismen eines binären zyklischen Codes einer Länge N0 mit einem Generatorpolynom ausgewählt wird, das äquivalent ist zu g(x) und zumindest zwei der N0 Spalten der Tabelle miteinander permutiert, wodurch eine Sequenz a*(x) erzeugt wird, die durch g(x) teilbar ist, und
– eine Permutation der Spalten der Tabelle, die das Polynom a*(x) in das Polynom a**(x) transformiert, das durch das Polynom $g_2(x)$ teilbar ist.

28. Verschachtler (**101**) gemäß Anspruch 27, dadurch gekennzeichnet, dass der Verschachtler (**101**) angepasst ist zum Durchführen der Permutation wie folgt:
a*(x) = $a(x^e)$ modulo $x^{M.N0} – 1$,

wobei e ein Ganzzahlenwert gleich einer Potenz von 2 modulo M.N0 und M eine ungerade Zahl ist.

29. Verschachtler (**101**) gemäß Anspruch 27 oder 28, dadurch gekennzeichnet, dass er eine Übertragungseinrichtung (**106**) umfasst, die angepasst ist zum Übertragen einerseits von der Sequenz a und andererseits von einer Untermenge der Daten der anderen Sequenzen.

30. Datenübertragungsvorrichtung mit einem Sender, der zum Implementieren eines Paketübertragungsprotokolls angepasst ist, dadurch gekennzeichnet, dass sie eine Codierungsvorrichtung (**701**) gemäß einem der Ansprüche 8 bis 12 oder eine Decodierungsvorrichtung gemäß einem der Ansprüche 22 bis 26 oder einen Verschachtler (**101**) gemäß einem der Ansprüche 7 oder 27 bis 29 umfasst.

Es folgen 9 Blatt Zeichnungen

Anhängende Zeichnungen

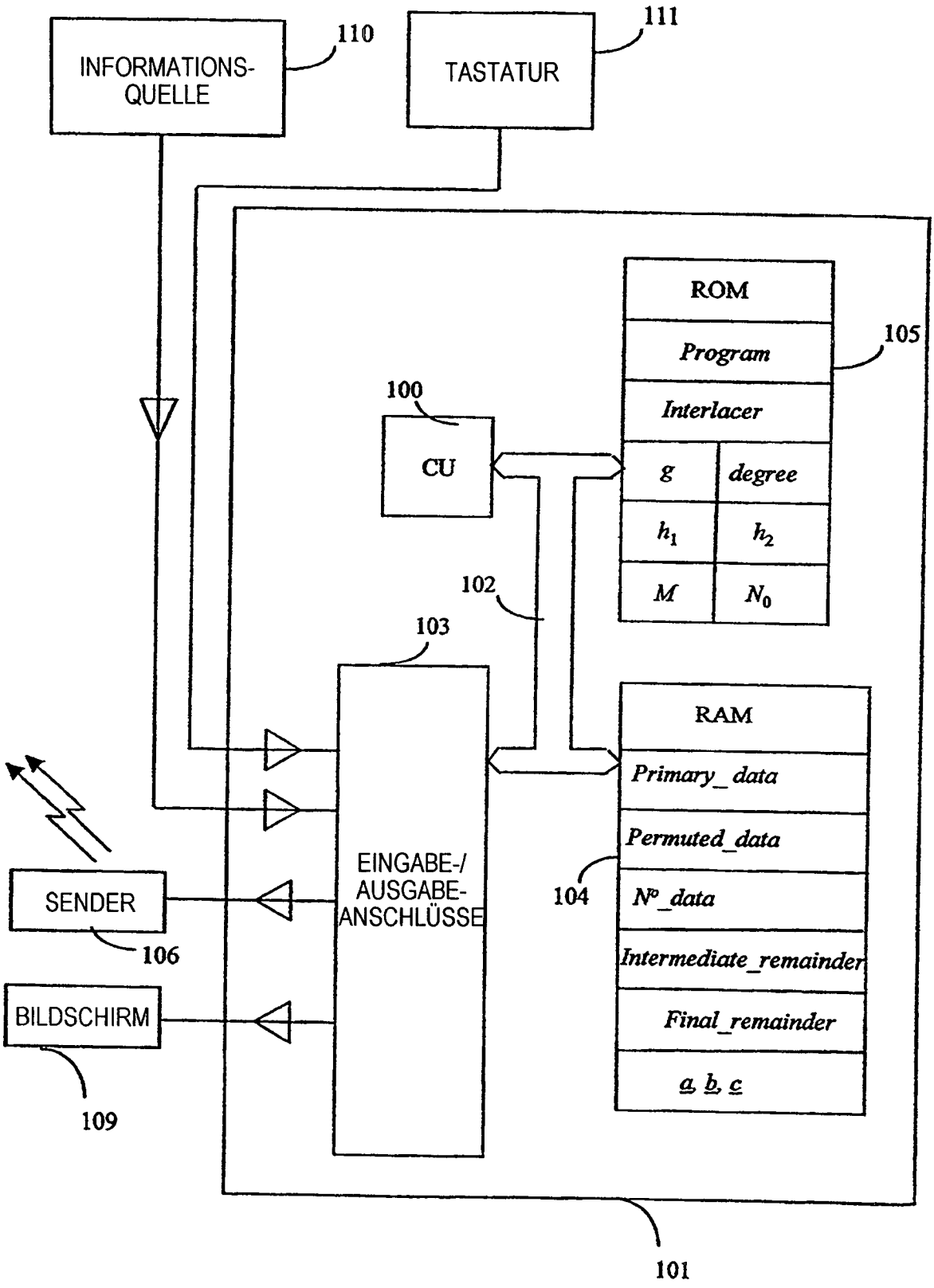

Fig.1

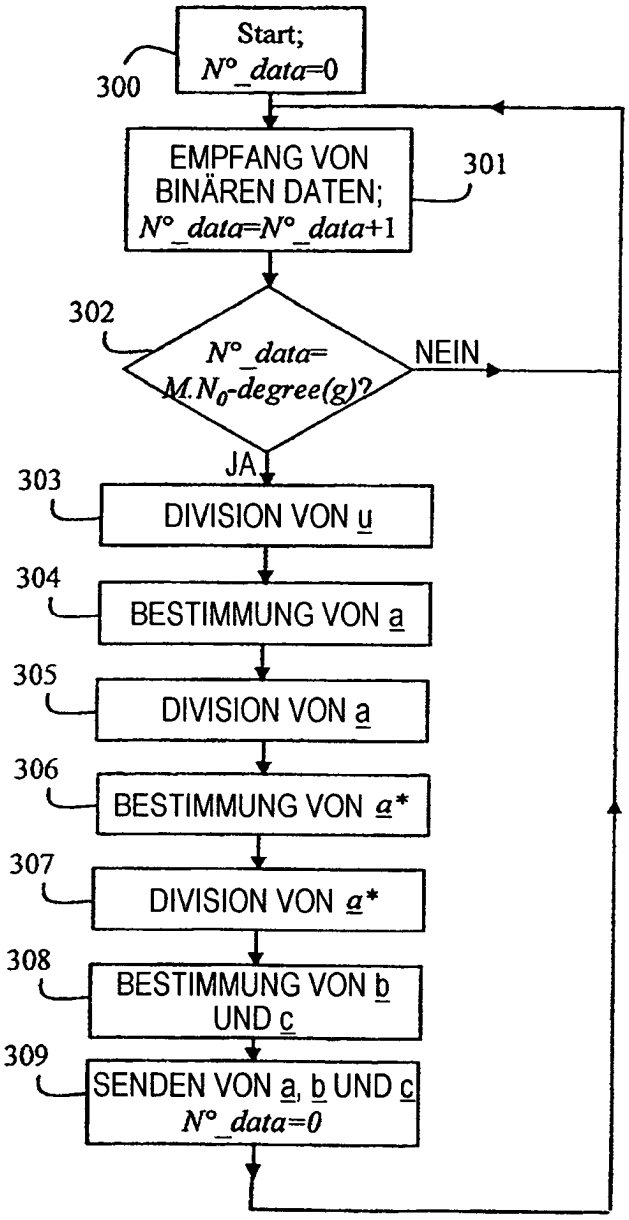
Start;
N°_data=0
300
EMPFANG VON BINÄREN DATEN;
N°_data=N°_data+1
301
302
N°_data= M.N0-degree(g)?
NEIN
JA
303
DIVISION VON u
304
BESTIMMUNG VON a
305
DIVISION VON a
306
BESTIMMUNG VON a*
307
DIVISION VON a*
308
BESTIMMUNG VON b UND c
309
SENDEN VON a, b UND c
N°_data=0

Fig.2

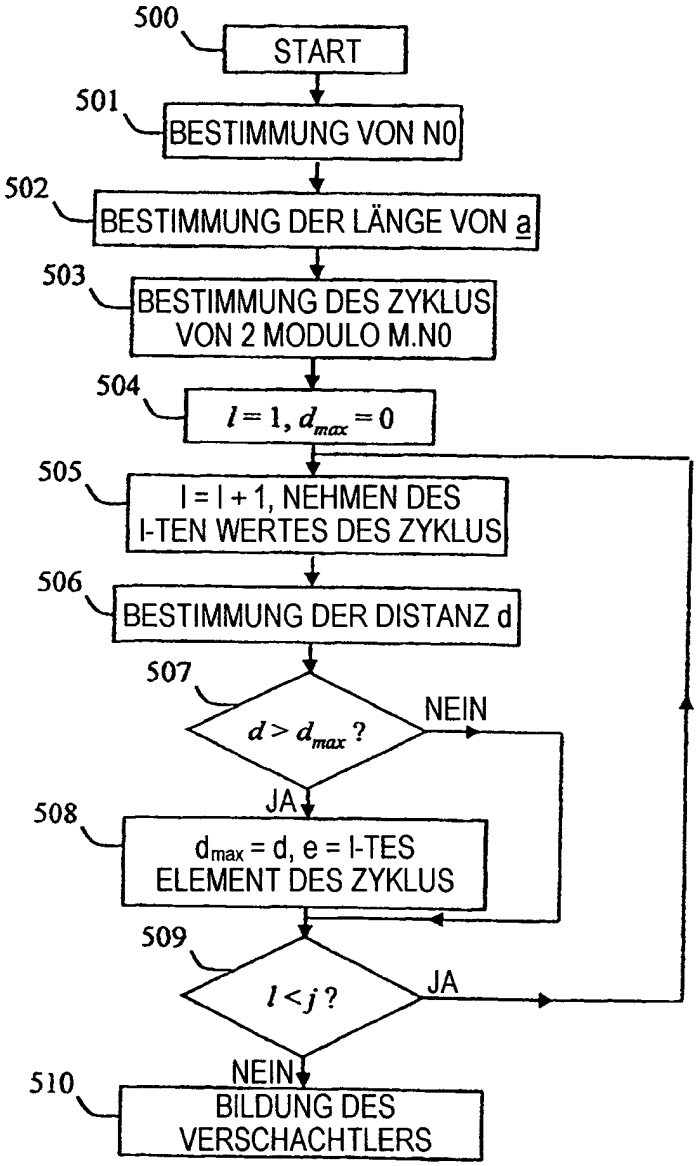
500
START
501
BESTIMMUNG VON N0
502
BESTIMMUNG DER LÄNGE VON a
503
BESTIMMUNG DES ZYKLUS VON 2 MODULO M.N0
504
$l = 1, d_{max} = 0$
505
I = I + 1, NEHMEN DES I-TEN WERTES DES ZYKLUS
506
BESTIMMUNG DER DISTANZ d
507
$d > d_{max}$?
NEIN
JA
508
d_{max} = d, e = I-TES ELEMENT DES ZYKLUS
509
$l < j$?
JA
NEIN
510
BILDUNG DES VERSCHACHTLERS

Fig.3

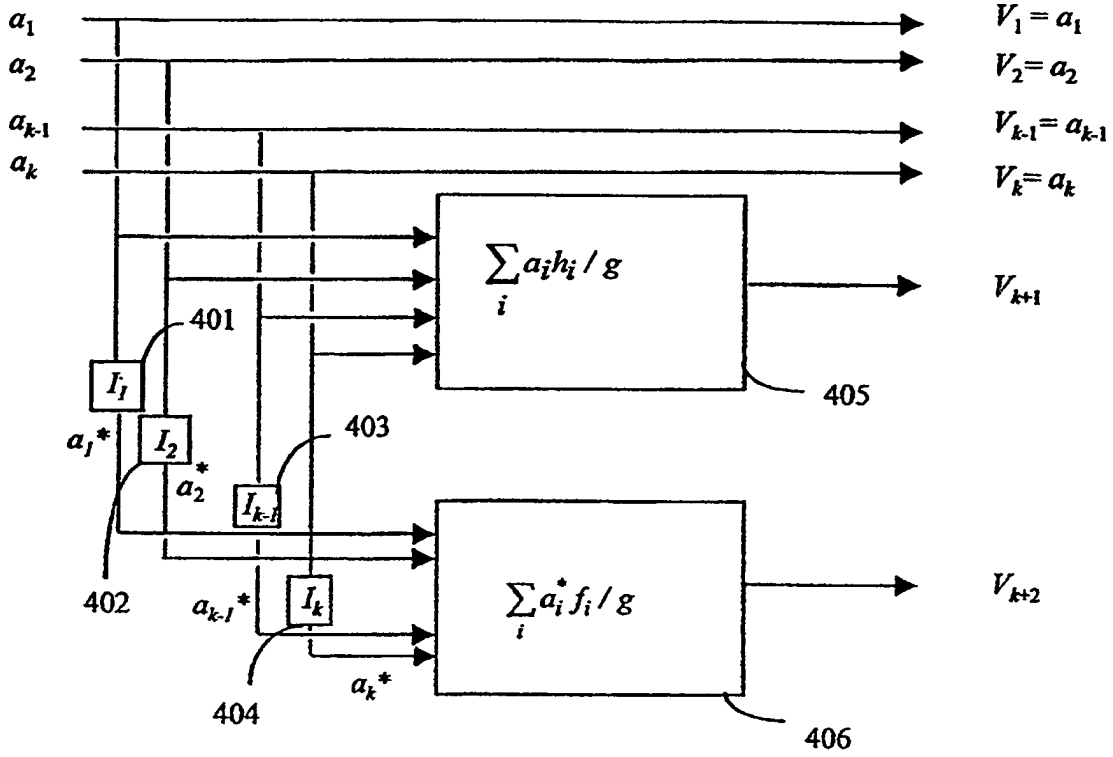
a_1
a_2
a_{k-1}
a_k
$V_1 = a_1$
$V_2 = a_2$
$V_{k-1} = a_{k-1}$
$V_k = a_k$
$\sum_i a_i h_i / g$
V_{k+1}
401
405
I_1
I_2
I_{k-1}
403
a_1^*
a_2^*
402
a_{k-1}^*
I_k
$\sum_i a_i^* f_i / g$
V_{k+2}
a_k^*
404
406

Fig.4

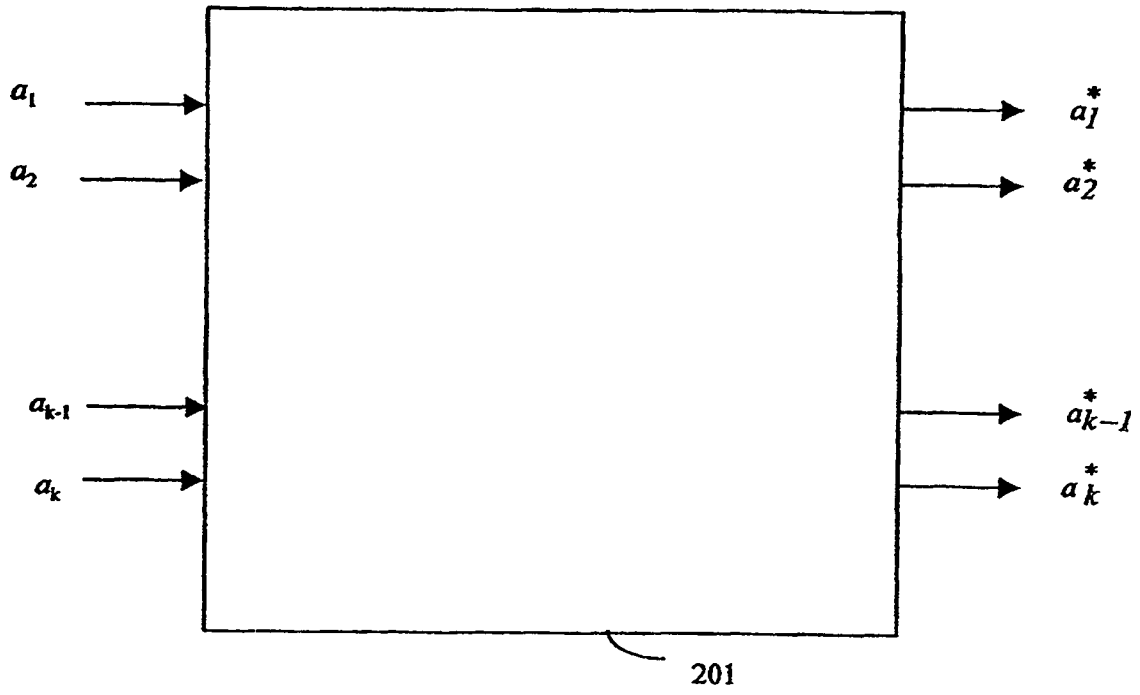

Fig.5

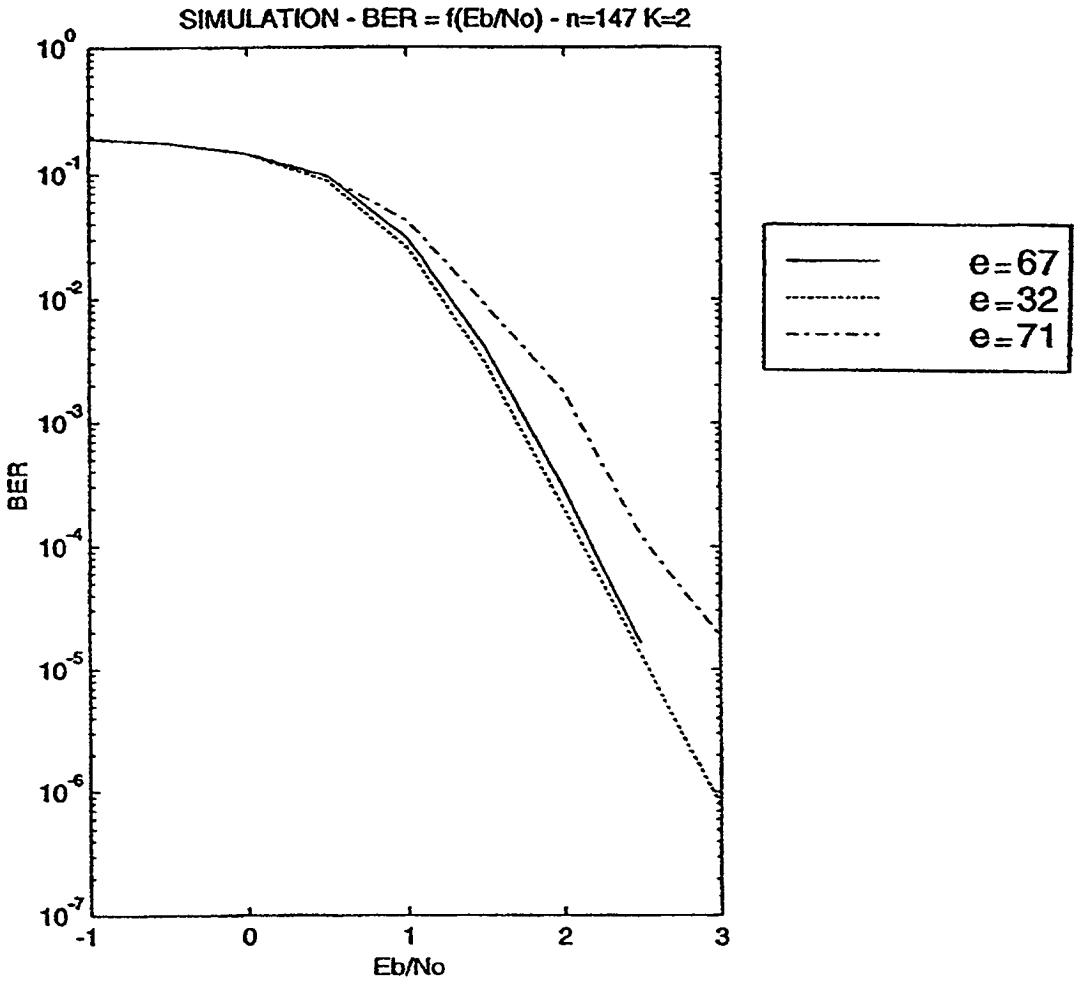

Fig.6

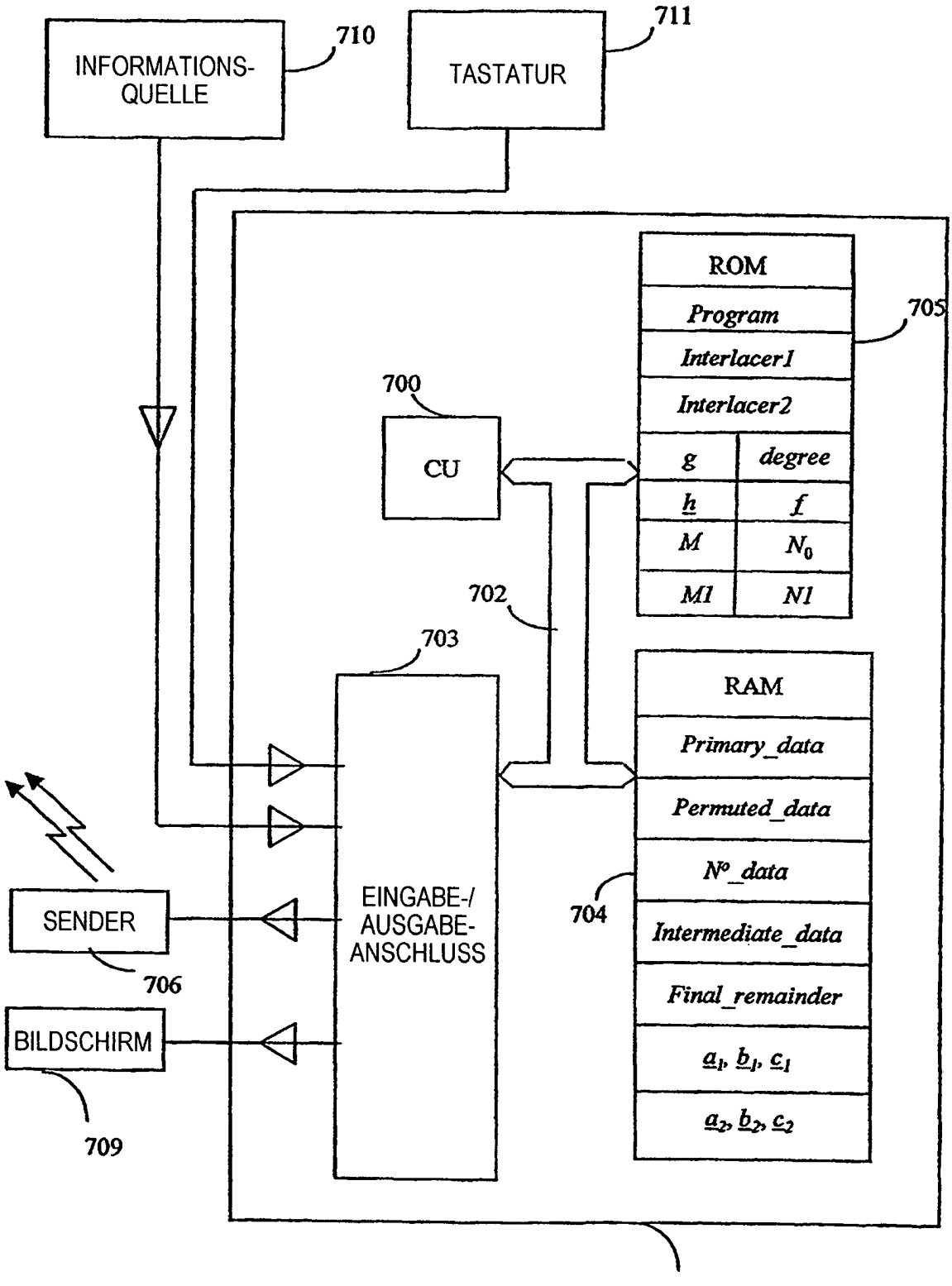
INFORMATIONS-QUELLE
710
TASTATUR
711
ROM
Program
Interlacer1
Interlacer2
g
degree
h
f
M
N0
M1
N1
705
700
CU
702
703
RAM
Primary_data
Permuted_data
N°_data
Intermediate_data
Final_remainder
a1, b1, c1
a2, b2, c2
704
EINGABE-/ AUSGABE-ANSCHLUSS
SENDER
706
BILDSCHIRM
709
701

Fig.7

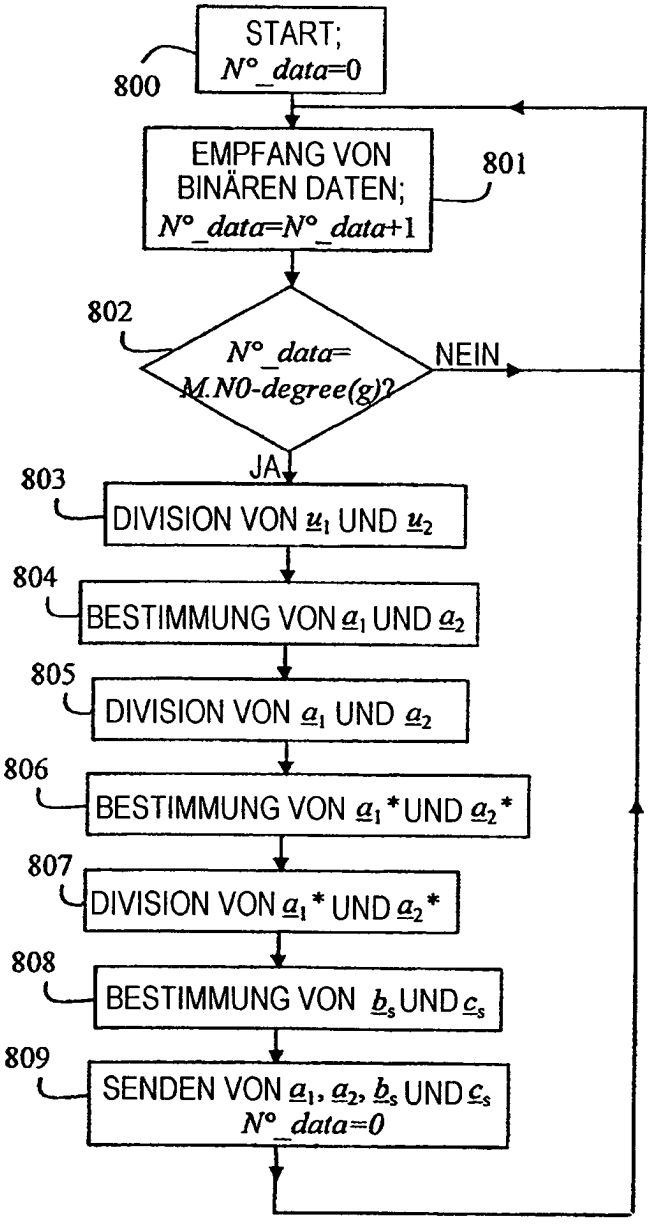
START;
N°_data=0
800
EMPFANG VON BINÄREN DATEN;
N°_data=N°_data+1
801
802
N°_data= M.N0-degree(g)?
NEIN
JA
803
DIVISION VON u_1 UND u_2
804
BESTIMMUNG VON a_1 UND a_2
805
DIVISION VON a_1 UND a_2
806
BESTIMMUNG VON a_1*UND a_2*
807
DIVISION VON a_1* UND a_2*
808
BESTIMMUNG VON b_s UND c_s
809
SENDEN VON a_1, a_2, b_s UND c_s
N°_data=0

Fig.8

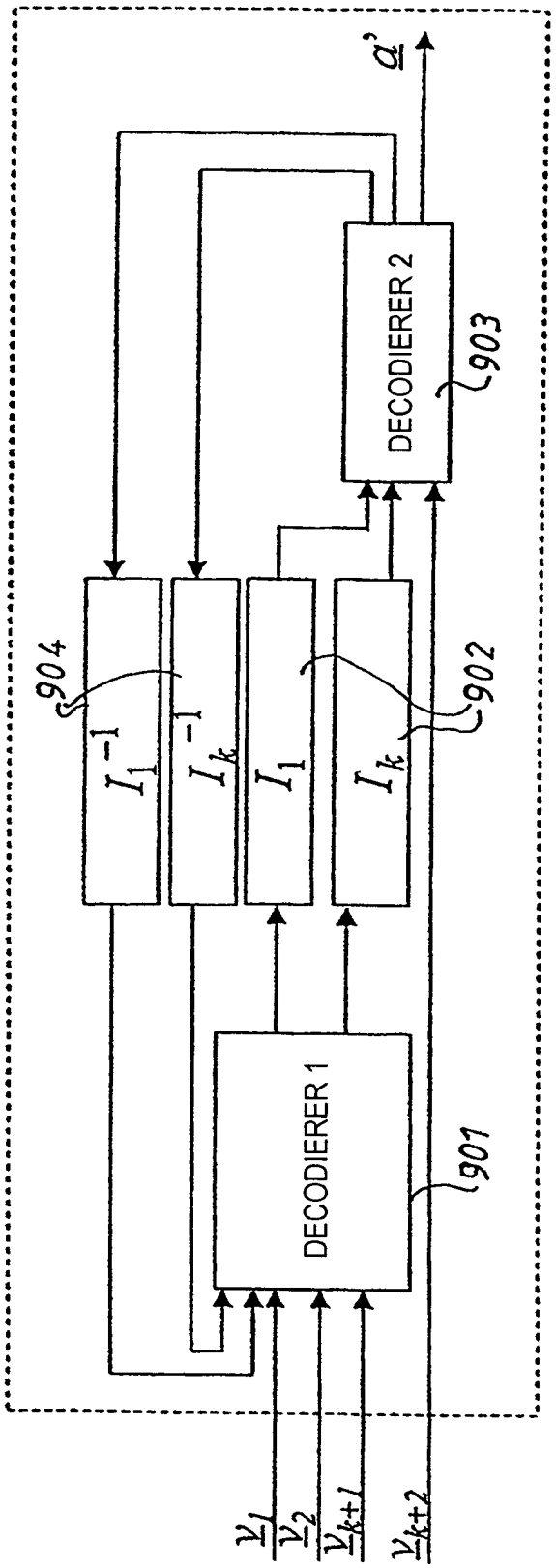
a'
DECODIERER 2
903
DECODIERER 1
901
902
904
I_1^{-1}
I_k^{-1}
I_1
I_k
y_1
y_2
y_{k+1}
y_{k+2}

Fig. 9