

(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES  
PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum  
Internationales Büro



(43) Internationales Veröffentlichungsdatum  
20. September 2001 (20.09.2001)

PCT

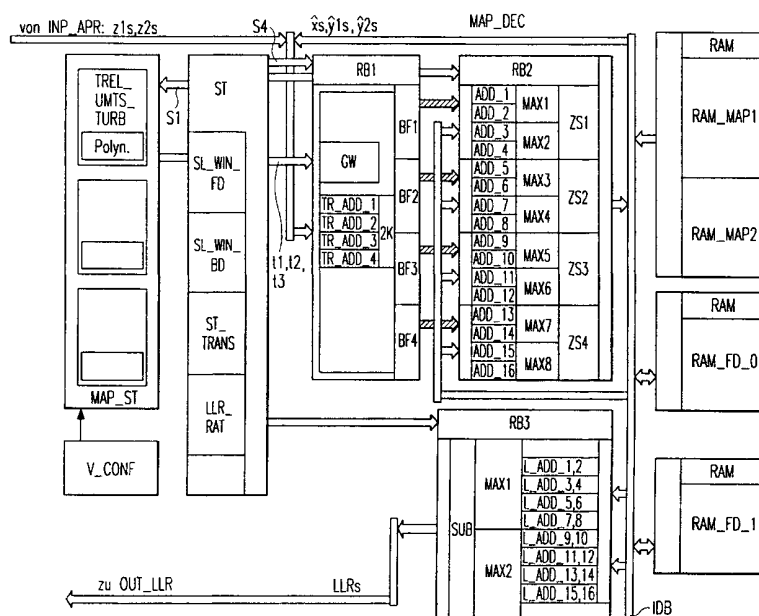
(10) Internationale Veröffentlichungsnummer  
**WO 01/69789 A2**

- (51) Internationale Patentklassifikation<sup>7</sup>: **H03M** (72) Erfinder; und  
(75) Erfinder/Anmelder (nur für US): **SCHMIDT, Peter** [DE/DE]; Bahnhofstrasse 32, 67167 Erpolzheim (DE).  
(21) Internationales Aktenzeichen: PCT/DE01/00983 **PLECHINGER, Jörg** [DE/DE]; Westermühlstrasse 16, 80469 München (DE). **SCHNEIDER, Michael** [DE/DE]; St.-Martin-Strasse 44A, 81541 München (DE). **DOETSCH, Markus** [DE/CH]; Haselozweg 26, Ch-3098 Schliern (CH). **KELLA, Tideya** [CM/DE]; Tumblingerstrasse 54/42, 80337 München (DE). **JUNG, Peter** [DE/DE]; Im Rabental 28, 67697 Otterberg (DE). **BECKER, Burkhard** [DE/DE]; Kolomanstrasse 8, 85737 Ismaning (DE).  
(22) Internationales Anmeldedatum: 12. März 2001 (12.03.2001)  
(25) Einreichungssprache: Deutsch  
(26) Veröffentlichungssprache: Deutsch  
(30) Angaben zur Priorität: 100 12 873.4 16. März 2000 (16.03.2000) DE  
(71) Anmelder (für alle Bestimmungsstaaten mit Ausnahme von US): **INFINEON TECHNOLOGIES AG** [DE/DE]; St.-Martin-Strasse 53, 81669 München (DE).  
(74) Anwalt: **LANGE, Thomas**; Dingolfinger Strasse 6, 81673 München (DE).  
(81) Bestimmungsstaaten (national): CN, JP, KR, US.

[Fortsetzung auf der nächsten Seite]

(54) Title: OPTIMIZED TURBO DECODER

(54) Bezeichnung: OPTIMIERTER TURBO-DECODIERER



(57) Abstract: The invention relates to a turbo decoder which is provided for decoding a data signal (D) transmitted over a disturbed channel and which comprises a symbol estimator (MAP\_DEC). Said estimator comprises a computing means which, by using the knowledge of the fault protection code used on the side of the transmitter, calculates transition metric values, forward and reverse recurrence metric values, and which calculates the output values (LLR) therefrom. The computing means comprise at least one hardware calculating module (RB1/2/3), which is constituted of at least one combinatory logic, for generating at least one type of said values.

[Fortsetzung auf der nächsten Seite]



WO 01/69789 A2



**(84) Bestimmungsstaaten (regional):** europäisches Patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

*Zur Erklärung der Zweibuchstaben-Codes, und der anderen Abkürzungen wird auf die Erklärungen ("Guidance Notes on Codes and Abbreviations") am Anfang jeder regulären Ausgabe der PCT-Gazette verwiesen.*

**Veröffentlicht:**

— *ohne internationalen Recherchenbericht und erneut zu veröffentlichen nach Erhalt des Berichts*

---

**(57) Zusammenfassung:** Ein Turbo-Decodierer zum Decodieren eines über einen gestörten Kanal übertragenen Datensignals (D) weist einen Symbolschätzer (MAP\_DEC) auf. Dieser umfaßt ein Rechenmittel, welches mit Kenntnis des senderseitig eingesetzten Fehlerschutzcodes Übergangs-Metrikwerte, Vorwärts- und Rückwärtsrekursions-Metrikwerte, und daraus die Ausgabewerte (LLR) berechnet. Das Rechenmittel umfaßt zumindest einen aus kombinatorischer Logik aufgebauten Hardware-Rechenbaustein (RB1/2/3) zur Erzeugung zumindest einer Sorte der genannten Werte.

## Optimierter Turbo-Decodierer

Die Erfindung betrifft einen Turbo-Decodierer zum Decodieren eines senderseitig mit einem Turbo-Code fehlerschutzcodierten, über einen gestörten Kanal übertragenen und in einem Empfänger detektierten Datensignals nach dem Oberbegriff des Anspruchs 1.

Ein Grundproblem der Nachrichtentechnik besteht darin, ein Datensignal möglichst fehlerfrei über einen gestörten Kanal zu übertragen. Um dies zu erreichen, wird das zu übertragende Datensignal einer Kanalcodierung unterzogen. Durch die Kanalcodierung wird das zu übertragende Signal an die Eigenschaften des Übertragungskanals angepaßt. Es sind eine Vielzahl unterschiedlicher Kanalcodierverfahren im Einsatz. Das Prinzip der Kanalcodierung (die auch als Fehlerschutzcodierung bezeichnet wird) beruht darauf, daß durch gezieltes Einbringen von Redundanz in das zu übertragende Signal ein effektiver Fehlerschutz erreicht wird.

Besondere Schwierigkeiten im Zusammenhang mit der Signalübertragung ergeben sich bei Mobilfunksystemen. Dies ist einerseits darauf zurückzuführen, daß bei Mobilfunksystemen aufgrund von zeitvarianten Mehrfachreflektionen, Gleichkanalstörungen, Dopplerverschiebung und Rauschen ausgesprochen starke Signalverzerrungen am Empfänger auftreten. Andererseits stehen im Empfänger einer Mobilstation aus Platz- und Kostengründen nur begrenzte Energie- und Hardware-Ressourcen zur Verfügung. In der Mobilfunktechnik wird daher angestrebt, mit möglichst wenig Signalverarbeitungsaufwand (Schonung der Ressource Energie) möglichst Hardware-effizient (d.h. platz- und kostengünstig) eine Rekonstruktion des gesendeten Signals mit möglichst geringer Fehlerrate zu erreichen.

Es ist ersichtlich, daß die genannten Forderungen in der Regel nicht gleichzeitig optimierbar sind, sondern daß ein sinnvoller Ausgleich oder Kompromiß zwischen diesen Forderungen

gen durch eine gezielte Software/Hardware-Auslegung des Empfängers geschaffen werden muß.

Ein modernes Verfahren der Fehlerschutzcodierung ist die sogenannte "Turbo-Codierung". Bei der Turbo-Codierung werden  
5 sogenannte "Turbo-Codes" eingesetzt. Turbo-Codes sind binäre, parallel verkettete rekursive Faltungscodes. Turbo-Codes stellen insbesondere bei der Übertragung großer Datenblöcke eine ausgesprochen leistungsfähige Form der Fehlerschutzco-  
10 dierung dar.

Zum Decodieren von Turbo-Codes wird im Empfänger ein Turbo-Decodierer eingesetzt. Ein solcher ist in dem den nächstliegenden Stand der Technik repräsentierenden Buch "Analyse und  
15 Entwurf digitaler Mobilfunksysteme" von P. Jung, Stuttgart, B.G. Teubner, 1997, auf den Seiten 343 bis 368, insbesondere Bild E.2, beschrieben. Dieser Turbo-Decodierer besteht aus zwei rekursiv verschalteten MAP-Symbolschätzern, zwischen welchen ein Turbo-Code-Verschachteler angeordnet ist.

20 Ein Hindernis bei der Implementierung von Turbo-Codes besteht darin, daß Turbo-Decodierer aufgrund der iterativen Signalverarbeitung und der zwei MAP-Symbolschätzer einen beträchtlichen Signalverarbeitungs- und Hardware-Aufwand erfordern.

25 Bisher werden Turbo-Codes daher nur bei weniger ressourcenbegrenzten Systemen, wie beispielsweise Satellitenkommunikationssystemen, eingesetzt. Im Bereich des Mobilfunks sind Turbo-Codes aufgrund der genannten Beschränkungen noch nicht im  
30 Einsatz.

Der Erfindung liegt die Aufgabe zugrunde, einen Turbo-Decodierer zu schaffen, der bei vertretbarem Hardware-Aufwand eine signalverarbeitungsarme und damit Energieresourcen-scho-  
35 nende Signaldecodierung ermöglicht. Insbesondere soll der erfindungsgemäße Turbo-Decodierer zum Einsatz in Mobilfunkempfängern geeignet sein.

Zur Lösung der Aufgabe sind die Merkmale des Anspruchs 1 vorgesehen.

5 Demnach besteht das Konzept der Erfindung darin, durch den (zumindest einen) in dem Rechenmittel vorgesehenen Hardware-Rechenbaustein bestehend aus kombinatorischer Logik eine gezielte Beschleunigung der Signalverarbeitung bei der Berechnung der Übergangs-Metrikwerte und/oder der Vorwärts- und  
10 Rückwärtsrekursions-Metrikwerte und/oder der Ausgabewerte herbeizuführen. Das heißt, daß zumindest eine der genannten Turbo-Decodier-Berechnungsprozeduren vollständig in Hardware ausgeführt ist, so daß eine zeit- und energieaufwendige Programm-Abarbeitung an dieser Stelle entfällt. Dadurch kann der  
15 gesamte Turbo-Decodier-Ablauf wesentlich beschleunigt werden.

Eine vorteilhafte Ausführungsform des erfindungsgemäßen Turbo-Decodierers kennzeichnet sich dadurch, daß das Rechenmittel einen ersten Hardware-Rechenbaustein zur Erzeugung von  
20 Übergangs-Metrikwerten enthält, welcher vier parallel angeordnete Addierer und wenigstens eine den Addierern nachgeschaltete Zweier-Komplement-Stufe umfaßt. Auf diese Weise können in kürzester Zeit gleichzeitig sechzehn Übergangs-Metriken (die acht unterschiedliche Werte annehmen können)  
25 berechnet werden.

Eine weitere vorteilhafte Ausgestaltung des erfindungsgemäßen Turbo-Decodierers kennzeichnet sich dadurch, daß das Rechenmittel einen zweiten Hardware-Rechenbaustein zur Erzeugung  
30 von Vorwärts- und Rückwärtsrekursions-Metrikwerten enthält, welcher eine Additionsstufe bestehend aus einer Anzahl Q von parallel angeordneten Addierern, und eine Maximierungsstufe bestehend aus parallel angeordneten Maximum-Einheiten umfaßt, wobei jede Maximum-Einheit an ihren Eingängen die Ausgänge  
35 von zwei Addierern entgegennimmt und den größeren der Addiererausgänge an ihrem Ausgang ausgibt. Auf diese Weise können

in kürzester Zeit gleichzeitig acht Vorwärts- oder Rückwärtsrekursions-Metrikwerte berechnet werden.

Vorzugsweise umfaßt der Symbolschätzer ein Steuermittel zur Steuerung des zweiten Hardware-Rechenbausteins, welches bewirkt, daß zur Berechnung der Vorwärts- und Rückwärtsrekursions-Metrikwerte Vorwärts- und Rückwärtsrekursionsintervalle mit gleitenden Intervallgrenzen innerhalb eines Datenblocks eingesetzt werden.

10

Eine schnelle Berechnung von Ausgabewerten wird durch das Vorsehen eines dritten Hardware-Rechenbausteins erreicht, welcher eine Additionsstufe bestehend aus einer Anzahl  $P$  von parallel angeordneten Addierern und eine Maximierungsstufe bestehend aus zwei parallel angeordneten Maximum-Einheiten umfaßt, wobei jede Maximum-Einheit an ihren Eingängen die Ausgänge von  $P/2$  Addierern entgegennimmt und den größten der Addiererausgänge an ihrem Ausgang ausgibt.

Selbstverständlich kann das Rechenmittel auch alle drei genannten Hardware-Rechenbausteine umfassen. In diesem Fall sind in dem Rechenmittel drei festverschaltete Datenpfade realisiert, welche die wichtigsten Berechnungsschritte in direkter Signalverarbeitung (ohne Programmabarbeitung) ausführen.

Vorzugsweise umfaßt der erfindungsgemäße Turbo-Decodierer ferner einen digitalen Signalprozessor (DSP) zur Durchführung von Turbo-Ver- und Entschachtelungsprozeduren und zur Berechnung von für den momentanen Kanalzustand repräsentativer statistischer Information gemäß einem vorgegebenen Ablaufprogramm. Dabei steht der DSP über eine bidirektionale Schnittstelle mit dem Symbolschätzer in Datenaustauschverbindung. Mit anderen Worten wird der Turbo-Decodierer in einen auf Software-Abarbeitung basierenden Rechenabschnitt (DSP mit Ablaufprogramm) und einen im wesentlichen auf Hardware-Ebene arbeitenden Rechenabschnitt (Symbolschätzer) aufgeteilt. Die-

se Aufteilung, bei der der Datentransfer zwischen den beiden Rechenabschnitten durch die bidirektionale Schnittstelle realisiert wird, hat sich unter dem Gesichtspunkt eines Kompromisses zwischen aufwandsgünstiger Signalverarbeitung und vertretbarem Hardware-Aufwand als besonders vorteilhaft herausgestellt.

Eine weitere vorteilhafte Variante des erfindungsgemäßen Turbo-Decodierers kennzeichnet sich dadurch, daß der Symbolschätzer während einer Iterationsschleife der Turbo-Decodier-Prozedur sowohl zur Berechnung von erster als auch zur Berechnung von zweiter Zuverlässigkeitsinformation herangezogen wird. Diese gemultiplexte Ansteuerung des Symbolschätzers bewirkt, daß der (bei Turbo-Code-Decodierung notwendigerweise erhebliche) Hardware-Aufwand noch in akzeptablen Grenzen gehalten werden kann.

Weitere vorteilhafte Ausgestaltungen der Erfindung sind in den Unteransprüchen angegeben.

20

Die Erfindung wird nachfolgend anhand eines Ausführungsbeispiels unter Bezugnahme auf die Zeichnung erläutert; in dieser zeigt:

25 Fig. 1 eine schematische Darstellung der Luftschnittstelle eines Mobilfunksystems mit Sender und Empfänger;

Fig. 2 ein Blockschaltbild eines Turbo-Codierers zur Erzeugung eines Turbo-Codes;

30

Fig. 3 ein Blockschaltbild eines in Fig. 2 dargestellten RSC-Faltungscodierers;

35 Fig. 4 ein Funktions-Blockschaltbild zur Erläuterung der Arbeitsweise eines erfindungsgemäßen Turbo-Decodierers;

Fig. 5 ein Blockschaltbild des in Fig. 4 dargestellten Turbo-Decodierers;

5 Fig. 6 ein vereinfachtes Blockschaltbild des in Fig. 5 dargestellten MAP-Decodierers;

10 Fig. 7 eine schematische Darstellung zur Erläuterung einer segmentweisen Vorwärts- und Rückwärtsrekursionen bei der Berechnung von Zuverlässigkeitsinformation;

Fig. 8 ein Blockschaltbild der in Fig. 6 dargestellten Steuereinheit zur Erzeugung von standardspezifischer Code-Information;

15 Fig. 9a ein Schaubild, in welchem die möglichen Übergänge zwischen den Anfangs- und Endzuständen eines Turbo-Codierers bei Eingabe eines Bits des Wertes 0 dargestellt sind;

20 Fig. 9b ein Schaubild, in welchem die möglichen Übergänge zwischen den Anfangs- und Endzuständen eines Turbo-Codierers bei Eingabe eines Bits des Wertes 1 dargestellt sind;

25 Fig. 9c ein Schaubild, in welchem vier Gruppen von Übergängen, geordnet nach den Endzuständen, dargestellt sind;

Fig. 9d ein Blockschaltbild einer Rechenstufe zur Berechnung von theoretischer Information eines Turbo-Codes; und

30 Fig. 10 ein Blockschaltbild der Hardware-Architektur des in Fig. 6 dargestellten MAP-Decodierers.

35 Fig. 1 zeigt einen Sender S und einen Empfänger E eines Mobilfunksystems. Der Sender S ist beispielsweise in einer Basisstation und der Empfänger E in einer Mobilstation des Mobilfunksystems enthalten.

Der Sender S weist einen Turbo-Codierer TCOD, einen Modulator MOD sowie eine Sendeantenne SA auf.

5 Der Turbo-Codierer TCOD nimmt ein digitales Eingangssignal in Form von Datensymbolen (beispielsweise Bits)  $u_1, u_2, \dots$  entgegen. Im folgenden wird aufgrund der blockweisen Codierung eine endliche Folge  $U = (u_1, u_2, \dots, u_N)$  bestehend aus N Eingabesignal-Datensymbolen (z.B. Bits)  $u_n, n = 1, 2, \dots, N$  betrachtet, wobei die Anzahl N als Blockgröße bezeichnet wird.

Das Eingangssignal trägt eine zu übertragende Nutzinformation, beispielsweise eine Sprachnachricht. Es kann z.B. über eine Mikrofon-Verstärker-Analog/Digital-Umsetzer Schaltungskette  
15 (nicht dargestellt) erzeugt werden.

Der Turbo-Codierer TCOD fügt dem digitalen Eingangssignal zur Fehlerschutzcodierung Redundanz hinzu. Am Ausgang des Turbo-Codierers TCOD liegt ein fehlerschutzcodiertes Datensignal in  
20 Form einer Folge D bestehend aus K Datensymbolen (Bits),  $D = (d_1, d_2, \dots, d_K)$  vor.

Das Verhältnis  $N/K$  (Anzahl von Eingabebits/Anzahl von Ausgangsbits) wird als Coderate  $R_c$  eines Codierers bezeichnet.  
25

Ein Modulator MOD moduliert das fehlerschutzcodierte Datensignal auf ein Trägersignal auf. Das mit dem fehlerschutzcodierten Datensignal modulierte Trägersignal wird in nicht dargestellter Weise von einem Sendefilter spektral geformt  
30 und von einem Sendeverstärker verstärkt, bevor es als Funksignal FS über die Sendeantenne SA abgestrahlt wird.

Der Empfänger E weist eine Empfangsantenne EA, einen Demodulator DMOD und einen Turbo-Decodierer TDEC auf.  
35

Die Empfangsantenne EA empfängt das durch Umgebungseinflüsse und Interferenz mit Funksignalen anderer Teilnehmer gestörte Funksignal FS und führt es dem Demodulator DMOD zu.

- 5 Der Demodulator DMOD entzerrt das empfangene Funksignal FS unter Berücksichtigung der im Funkkanal erlittenen Signalstörungen. Ein am Ausgang des Demodulators DMOD bereitgestelltes entzerartes Datensignal liegt in Form einer Datensymbolfolge  $\hat{D} = (\hat{d}_1, \hat{d}_2, \dots, \hat{d}_K)$  vor, deren Elemente  $\hat{d}_1, \hat{d}_2, \dots, \hat{d}_K$
- 10 wertekontinuierliche Schätzwerte der Datensymbole  $d_1, d_2, \dots, d_K$  der fehlerschutzcodierten Datensymbolfolge D sind.

- Das entzerarte Datensignal wird dem Turbo-Decodierer TDEC zugeführt, an dessen Ausgang eine decodierte Ausgabesignalfolge  $\hat{U} = (\hat{u}_1, \hat{u}_2, \dots, \hat{u}_N)$  bereitgestellt wird. Die Elemente  $\hat{u}_1, \hat{u}_2, \dots, \hat{u}_N$  der decodierten Ausgabesignalfolge  $\hat{U}$  sind Hypothesen der Datensymbole  $u_1, u_2, \dots, u_N$  des senderseitigen Eingangssignals in Form von diskreten Werten aus dem Symbolvorrat (beispielsweise 0,1) des Eingangssignals.

- 20 Die Datensymbol-Fehlerrate wird durch die relative Häufigkeit von Fehlschätzungen  $u_n \neq \hat{u}_n, n = 1, 2, \dots$  definiert. Sie darf bei Mobilfunkanwendungen einen bestimmten maximal zulässigen Wert nicht überschreiten.

- 25 Vor der Beschreibung eines Ausführungsbeispiels des erfindungsgemäßen Turbo-Decodierers wird zum besseren Verständnis der Erfindung anhand Fig. 2 zunächst beispielhaft die Erzeugung eines Turbo-Codes erläutert.

- 30 Ein Turbo-Codierer TCOD weist zwei identische binäre rekursive systematische Faltungscodierer RSC1 und RSC2 auf, die in der Codiertechnik als RSC-(Recursive Systematic Convolutional-)Codierer bekannt sind. Dem zweiten RSC-Faltungscodierer RSC2 ist eingangsseitig ein Turbo-Code-Verschachteler IL vorgeschaltet, der eine blockweise Verschachtelung des Eingabedatensignals vornimmt. Die Ausgänge der beiden Faltungscodie-

rer RSC1 und RSC2 sind jeweils über Punktierer PKT1 bzw. PKT2 mit einer Multiplexereinrichtung MUX verbunden. Der Multiplexereinrichtung MUX wird ferner eine Signalfolge X zugeführt, die identisch mit der digitalen Eingabesignalfolge U ist.

5

Fig. 3 zeigt am Beispiel von RSC1 den Aufbau eines rekursiven Faltungscodierers. Der Faltungscodierer RSC1 weist eingangsseitig einen ersten Addierer ADD1 und ein dem ersten Addierer ADD1 nachgeschaltetes Schieberegister mit z.B. drei Speicherzellen T auf. An seinem Ausgang stellt der Faltungscodierer RSC1 eine Redundanz-Datenfolge  $Y1 = (y1_1, y1_2, \dots, y1_N)$  bereit, die von einem zweiten Addierer ADD2 gebildet wird.

Es wird deutlich, daß ein zu einem bestimmten Zeitpunkt n am Ausgang vorliegendes Redundanz-Datensymbol  $y1_n$  ( $n = 1, 2, \dots, N$ ) von dem aktuellen Eingabe-Datensymbol  $u_n$  der Eingabesignalfolge U sowie von dem Zustand des Schieberegisters abhängt.

Der Aufbau des zweiten Faltungscodierers RSC2 ist identisch mit dem Aufbau des ersten Faltungscodierers RSC1; an seinem Ausgang stellt RSC2 eine Redundanz-Datenfolge  $Y2 = (y2_1, y2_2, \dots, y2_N)$  bereit.

Die Signalfolge X kann als zweiter Ausgang des ersten Faltungscodierers RSC1 angesehen werden, das heißt bei dieser Betrachtungsweise umfaßt der erste Faltungscodierer RSC1 einen zweiten Ausgang, an dem die Datenfolge X ausgegeben wird, deren Elemente  $x_1, x_2, \dots, x_N$  identisch mit den Elementen  $u_1, u_2, \dots, u_N$  der Eingabesignalfolge U sind. Analoges gilt für den zweiten Faltungscodierer RSC2 und einen zweiten Ausgang  $X_I$  (nicht dargestellt) desselben, der identisch mit der verschachtelten Eingabesignalfolge U ist. Codierer mit dieser Eigenschaft werden allgemein als systematische Codierer bezeichnet.

Es werden dann pro Eingabe-Datensymbol  $u_n$  von jedem Faltungscodierer RSC1 bzw. RSC2 genau zwei Ausgabe-Datensymbole  $x_n$  und  $y_{1n}$  bzw.  $x_{I_n}$  (verschachtelte systematische Information) und  $y_{2n}$  ausgegeben. Jeder Faltungscodierer RSC1, RSC2 weist  
5 somit eine Coderate  $R_c = 0,5$  auf.

Die Multiplexereinrichtung MUX dient zur Einstellung der Coderate des Turbo-Codierers TCOD. Um auch für TCOD eine Coderate von z.B.  $R_c = 0,5$  zu erreichen, werden die beiden  
10 Redundanz-Teilfolgen Y1 und Y2 beispielsweise alternierend punktiert und multiplexiert. Die sich dabei ergebende Redundanz-Datenfolge  $Y = (y_{11}, y_{22}, y_{13}, y_{24}, \dots, y_{1N}, y_{2N})$  wird nachfolgend alternierend mit der systematischen Datenfolge X multiplexiert. Das sich bei dieser (speziellen) Form der Tur-  
15 bocodierung ergebende fehlerschutzcodierte Datensignal weist demzufolge (N sei als eine gerade Zahl angenommen) beispielsweise die Form  $D = (x_1, y_{11}, x_2, y_{22}, x_3, y_{13}, x_4, y_{24}, \dots, x_N, y_{2N})$  auf.

20 Der Faltungscodierer RSC1 kann als endlicher, getakteter Automat aufgefaßt und durch ein sog. Trellis-Diagramm mit M möglichen Zuständen beschrieben werden. Das Trellis-Diagramm des Faltungscodierers RSC1 mit einem Schieberegister aus 3 Zellen weist  $M = 2^3 = 8$  Knoten auf, die den möglichen Zustän-  
25 den des Schieberegisters entsprechen. Ein (beliebiger) erster Zustand  $m'$ , der durch Eingabe eines Eingangsbits ( $u_n = 0$  oder 1) in einen zweiten Zustand  $m$  übergeht, ist mit diesem im Trellis-Diagramm durch eine Verbindungslinie verbunden. Jede Redundanz-Teilfolge Y1 entspricht einem bestimmten Weg ent-  
30 lang Verbindungslinien durch das Trellis-Diagramm des RSC1-Codierers.

Trellis-Diagramme zur Veranschaulichung der Zustände von Codierern sind bekannt und werden hier nicht näher erläutert.

Fig. 4 zeigt ein Blockschaltbild zur Erläuterung der Arbeitsweise eines Ausführungsbeispiels eines erfindungsgemäßen Turbo-Decodierers TDEC.

- 5 Der Turbo-Decodierer TDEC umfaßt einen Demultiplexer DMUX, eine Statistik-Stufe STA, einen ersten und zweiten Speicher mit wahlfreiem Zugriff MAP\_RAM1, MAP\_RAM2, einen ersten und zweiten MAP-Symbolschätzer MAP1 und MAP2, eine erste und zweite Datenverarbeitungseinheit DVE1, DVE2 sowie eine Entscheidungslgik (Schwellenwertentscheider) TL.

- Die erste Datenverarbeitungseinheit DVE1 umfaßt einen Turbo-Verschachteler IL, eine Statistik-Stufe STA und einen Datenspeicher APR\_RAM. Die zweite Datenverarbeitungseinheit DVE2  
15 umfaßt einen Entschachteler DIL, ebenfalls eine Statistik-Stufe STA und ebenfalls einen Datenspeicher APR\_RAM.

- Die von dem Demodulator DMOD dem Turbo-Decodierer TDEC zugeführte entzerrte Datenfolge  $\hat{D} = (\hat{x}_1, \hat{y}_{11}, \hat{x}_2, \hat{y}_{22}, \hat{x}_3, \hat{y}_{13},$   
20  $\hat{x}_4, \hat{y}_{24}, \dots, \hat{x}_N, \hat{y}_{2N})$  wird von dem eingangsseitig vorgesehenen Demultiplexer DMUX in die entzerrte systematische Datenfolge  $\hat{X}$  (detektierte Version der Eingabesignalfolge  $U (= X)$ ) und die beiden entzerrten Redundanz-Teilfolgen  $\hat{Y}_1$  und  $\hat{Y}_2$  (detektierte Versionen der Redundanz-Teilfolgen  $Y_1$  und  $Y_2$ )  
25 aufgespalten. Mit  $\hat{x}_n, \hat{y}_{1n}, \hat{y}_{2n}$  ( $n = 1, 2, \dots, N$ ) werden die entzerrten (geschätzten) Versionen der senderseitig auftretenden Datensymbole  $x_n, y_{1n}, y_{2n}$  bezeichnet. (In der Zeichnung sind die Indizes  $n$  der jeweiligen Folgeelemente aus Gründen der Übersichtlichkeit weggelassen.)

- 30 Die Datensymbole  $\hat{x}_n$  (systematische Information) und  $\hat{y}_{1n}, \hat{y}_{2n}$  (Redundanz-Information) werden der Statistik-Stufe STA zugeführt. Basierend auf der Kenntnis des momentanen Zustands des Übertragungskanal werden dort die statistischen Datenwerte  
35  $\hat{x}_{sn}, \hat{y}_{1sn}, \hat{y}_{2sn}$  gemäß der folgenden Gleichungen berechnet.

$$\hat{x}_{sn} = \hat{x}_n \cdot \mu_{\hat{x}_n} / \sigma_{\hat{x}_n}^2$$

12

$$\begin{aligned}\hat{y}1s_n &= \hat{y}1_n \cdot \mu_{\hat{y}1n} / \sigma_{\hat{y}1n}^2 \\ \hat{y}2s_n &= \hat{y}2_n \cdot \mu_{\hat{y}2n} / \sigma_{\hat{y}2n}^2\end{aligned}\quad (1)$$

Dabei bezeichnen  $\mu_{\hat{x}n}$ ,  $\mu_{\hat{y}1n}$ ,  $\mu_{\hat{y}2n}$  die Mittelwerte und  $\sigma_{\hat{x}n}^2$ ,  
 5  $\sigma_{\hat{y}1n}^2$ ,  $\sigma_{\hat{y}2n}^2$  die Varianzen des weißen Rauschens bezüglich der zugehörigen Signalanteile.

Die statistischen Datensymbolwerte  $\hat{x}s_n$  und  $\hat{y}1s_n$  werden in dem Datenspeicher MAP\_RAM1 abgelegt und die statistischen Datenwerte  $\hat{y}2s_n$  werden in dem Datenspeicher MAP\_RAM2 abgelegt.  
 10 Die Speichergrößen von MAP\_RAM1 und MAP\_RAM2 sind jeweils so bemessen, daß die statistischen Datenwerte eines gesamten Blocks (z.B. N = 5120) Platz finden.

15 Auf der Grundlage der für einen Datenblock repräsentativen systematischen Information  $\hat{x}s_n$  und Redundanz-Information  $\hat{y}1s_n$  und  $\hat{y}2s_n$  erfolgt eine iterative Turbo-Decodierung. Die Iterationsschleife wird durch die Einheiten MAP1, DVE1, MAP2, DVE2 und die Rückführung von Rückkoppel-Information  $z2s_n$  von  
 20 DVE2 zu MAP1 gebildet.

Genauer gesagt berechnet der erste MAP-Symbolschätzer MAP1 auf der Basis der Eingangs-Datenwerte  $\hat{x}s_n$  und  $\hat{y}1s_n$  und einer Anfangs-Rückkoppel-Information  $z2s_n(it\_no = 0)$  in einem ersten Iterationsschritt ( $it\_no = 1$ ) eine erste Folge von wertekontinuierlicher logarithmischer erster Zuverlässigkeitsinformation  $z1_n(it\_no = 1)$ . Diese erste Iterationsfolge logarithmischer Zuverlässigkeitsinformation wird in dem Turbo-Verschachteler IL verschachtelt und in der Statistik-Stufe  
 25 STA statistisch aufbereitet. Die statistische Aufbereitung erfolgt nach der Gleichung

$$z1s_n = z1_n \cdot \mu_{z1n} / \sigma_{z1n}^2 \quad (2)$$

Dabei bezeichnet  $\mu_{z1n}$  den Mittelwert und  $\sigma_{z1n}^2$  die Varianz des weißen Rauschens der ersten Zuverlässigkeitsinformation  $z1_n$ .

Zur Durchführung der genannten Prozeduren (Verschachtelung  
5 und statistische Aufbereitung) werden die Folgewerte  
 $z1s_n(it\_no = 1)$  in dem Datenspeicher mit wahlfreiem Zugriff  
APR\_RAM der ersten Datenverarbeitungseinheit DVE1 zwischenge-  
speichert.

10 Die in der ersten Datenverarbeitungseinheit DVE1 bei der er-  
sten Iteration erzeugten Datenwerte  $z1s_n(it\_no = 1)$  der ver-  
schachtelten Folge von erster Zuverlässigkeitsinformation  
(erste Iteration) werden dem zweiten MAP-Symbolschätzer MAP2  
zugeleitet. Dieser berechnet aus diesen Datenwerten und der  
15 Folge  $\hat{y}2s_n$  (für alle Iterationsdurchgänge identisch) eine  
Folge von zweiter logarithmischer Zuverlässigkeitsinformation  
 $z2_n(it\_no = 1)$ . Die einzelnen Werte der Folgen von erster und  
zweiter Zuverlässigkeitsinformation sind logarithmische a-  
Posteriori-Wahrscheinlichkeitsverhältnisse, sog. LLRs (Log-  
20 Likelihood Ratios).

Die Folge der zweiten Zuverlässigkeitsinformation  $z2_n(it\_no =$   
1) wird dem Turbo-Entschachteler DIL und der Statistik-Stufe  
STA der zweiten Datenverarbeitungseinheit DVE2 zugeleitet und  
25 dort entschachtelt und statistisch aufbereitet. Die statisti-  
sche Aufbereitung erfolgt nach der Gleichung

$$z2s_n = z2_n \cdot \mu_{z2n} / \sigma_{z2n}^2 \quad (3)$$

30 Dabei bezeichnet  $\mu_{z2n}$  den Mittelwert und  $\sigma_{z2n}^2$  die Varianz des  
weißen Rauschens der zweiten Zuverlässigkeitsinformation  $z2_n$ .

Der Datenspeicher mit wahlfreiem Zugriff APR\_RAM dient wie-  
derum der Zwischenspeicherung der in der beschriebenen Weise  
35 erzeugten Datenwerte  $z2s_n(it\_no = 1)$ . Gemäß Fig. 4 stehen sie

als Rückkoppel-Information dem ersten Symbolschätzer MAP1 während des nächsten Iterationsdurchlaufs ( $it\_no = 2$ ) zur Verfügung.

- 5 Nach einer vorgegebenen Anzahl von beispielsweise 5 Iterationsdurchläufen werden die am Ausgang des Turbo-Entschachtelers DIL der zweiten Datenverarbeitungseinheit DVE2 erzeugten Datenwerte  $z2_n(it\_no = 5)$  (entschachtelt) einem Schwellenwertentscheider TL zugeleitet. Der Schwellenwertentscheider TL bestimmt für jedes Element der angegebenen Datenfolge  
10 mit einem Wert  $\leq 0$  ein rekonstruiertes Datensymbol  $\hat{u}_n = 0$  und für jedes Element der genannten Datenfolge mit einem Wert  $> 0$  ein rekonstruiertes Datensymbol  $\hat{u}_n = 1$ .
- 15 Anhand der nachfolgenden Figuren wird deutlich, daß die in Fig. 4 mehrfach dargestellten Statistik-Stufen STA, die ersten und zweiten Symbolschätzer MAP1, MAP2 und die Datenspeicher mit wahlfreiem Zugriff APR\_RAM jeweils in Form ein und derselben Hardware-Struktur realisiert sein können. Die im  
20 unteren Bereich der Fig. 4 eingezeichneten Pfeile kennzeichnen in diesem Fall die mittels einer bidirektionalen Schnittstelle realisierten Datenübergabeorte, an welchen ein Datentransfer zwischen einem Symbolschätzer (enthält MAP1/2 und die zugeordneten Speicher MAP\_RAM1/2, schraffiert gezeichnet)  
25 und einen digitalen Signalprozessor (enthält DVE1/2, DMUX, STA, TL, ohne Schraffur) bewerkstelligt wird.

Es sei erwähnt, daß insbesondere die Statistikstufe STA auch in Hardware ausgeführt und in dem Symbolschätzer integriert  
30 sein kann.

Fig. 5 zeigt ein Blockschaltbild des bereits anhand Fig. 4 beschriebenen Turbo-Decodierers TDEC.

- 35 Der Turbo-Decodierer TDEC umfaßt einen MAP-Decodierer MAP\_DEC, welcher den beiden Symbolschätzern MAP1/2 in Fig. 4 entspricht und mit den beiden MAP-Datenspeichern MAP\_RAM1 und

- MAP\_RAM2 in Verbindung steht. Der MAP-Decodierer MAP\_DEC ist über einen bidirektionalen Datenbus DB und eine FMI-(flexible memory interface-)Schnittstelle FMI mit zwei programmierbaren DMA-(direkter Speicherzugriff: direct memory access-)Kanälen DMA\_KAN1 und DMA\_KAN2 verbunden. Die "Lage" dieser FMI-Schnittstelle FMI mit zugeordneten DMA-Kanälen DMA\_KAN1 und DMA\_KAN2 im Turbo-Decodierablauf ist wie bereits erwähnt in der Fig. 4 durch die Pfeile angezeigt.
- Die programmierbaren DMA-Kanäle DMA\_KAN1/2 umfassen jeweils interne DMA-Register DMA\_REG1, DMA\_REG2. Die internen DMA-Register DMA\_REG1/2 stehen über jeweilige Programmierleitungen IOD mit einem digitalen Signalprozessor DSP in bidirektionaler Datenaustauschverbindung.
- Der DSP hat über Datenleitungen und eine interne DSP-Schnittstelle IFD Zugriff auf den Datenspeicher APR\_RAM. Ferner stehen der DSP und die DSP-Schnittstelle IFD über eine bidirektionale Busstruktur bestehend aus einem Datenbus DD und einem Adressbus DA mit den DMA-Kanälen DMA\_KAN1/2 in Datenaustauschverbindung.
- Der DSP führt durch Abarbeitung einer herstellerseitig programmierten Software (Firmware) unter Zwischenspeicherung von Daten im Datenspeicher APR\_RAM die statistischen Berechnungen sowie die Ver- und Entschachtelungsprozeduren (siehe Fig. 4) und gegebenenfalls auch die von den Einheiten DMUX und TL ausgeführten Datenverarbeitungsschritte durch.
- Die Datenübermittlung über die DMA-Kanäle DMA\_KAN1/2 sowie die FMI-Schnittstelle FMI zu dem MAP-Decodierer MAP\_DEC wird über zwei Datenanforderungs-Steuersignale V\_TRQ\_M (Anforderung des Datentransfers über den ersten DMA-Kanal DMA\_KAN1) und V\_TRQ\_C (Anforderung des Datentransfers über den zweiten DMA-Kanal DMA\_KAN2) initiiert.

Über den ersten DMA-Kanal DMA\_KAN1 werden die Eingabedaten für die erste und die zweite Symbolschätzung (MAP1- und MAP2-Modus) übertragen. Für die erste Symbolschätzung sind dies einerseits die statistischen Datenwerte der systematischen Information  $\hat{x}s_n$  und die statistischen Datenwerte der ersten Redundanz-Information  $\hat{y}ls_n$ , welche lediglich einmal am Anfang eines Block-Decodierablaufes übertragen werden müssen, und andererseits sind dies die statistisch aufbereiteten Datenwerte  $z2s_n(it\_no)$  der Rückkoppel-Information, die in jedem Iterationsdurchlauf der Turbo-Decodierung übertragen werden.

Ein bei einer Wortbreite von 16 Bit beispielsweise verwendetes Datenformat der Eingabewerte für den ersten Symbolschätzer MAP1 ist in den beiden Tabellen 1 und 2 dargestellt.

15

Tabelle 1 (Datenübergabe an MAP1, einmal pro Block)

| Bits 0-3     | Bits 4-7 | Bits 8-11     | Bits 12-15 |
|--------------|----------|---------------|------------|
| $\hat{x}s_1$ |          | $\hat{y}ls_1$ |            |
| $\hat{x}s_2$ |          | $\hat{y}ls_2$ |            |
| $\hat{x}s_3$ |          | $\hat{y}ls_3$ |            |
| ...          |          | ...           |            |
| ...          |          | ...           |            |
| $\hat{x}s_N$ |          | $\hat{y}ls_N$ |            |

20

Tabelle 2 (Datenübergabe an MAP1, bei jedem Iterationsdurchlauf)

| Bits 0-3            | Bits 4-7 | Bits 8-11 | Bits 12-15 |
|---------------------|----------|-----------|------------|
| $z2s_1(it\_no=1-5)$ |          |           |            |
| $z2s_2(it\_no=1-5)$ |          |           |            |
| $z2s_3(it\_no=1-5)$ |          |           |            |
| ...                 |          |           |            |
| ...                 |          |           |            |
| $z2s_N(it\_no=1-5)$ |          |           |            |

Für die zweite Symbolschätzung (MAP2-Modus) werden dem MAP-Decodierer MAP\_DEC ebenfalls über den ersten DMA-Kanal DMA\_KAN1, die bidirektionale FMI-Schnittstelle FMI und den Datenbus DB die statistischen Datenwerte der zweiten Redundanz-Information  $\hat{y}_{2s_n}$  und - bei jedem Iterationsdurchlauf - die statistisch aufbereiteten Datenwerte der ersten Zuverlässigkeitsinformation  $z_{1s_n}(it\_no)$  zugeleitet. Ein mögliches Datenformat der Datenübertragung ist in den Tabellen 3 und 4 dargestellt:

10

Tabelle 3 (Datenübergabe für die zweite Symbolschätzung, einmal pro Block)

| Bits 0-3         | Bits 4-7 | Bits 8-11 | Bits 12-15 |
|------------------|----------|-----------|------------|
| $\hat{y}_{2s_1}$ |          |           |            |
| $\hat{y}_{2s_2}$ |          |           |            |
| $\hat{y}_{2s_3}$ |          |           |            |
| ...              |          |           |            |
| ...              |          |           |            |
| $\hat{y}_{2s_N}$ |          |           |            |

15 Tabelle 4 (Datenübergabe an MAP2, bei jedem Iterationsdurchlauf)

| Bits 0-3               | Bits 4-7 | Bits 8-11 | Bits 12-15 |
|------------------------|----------|-----------|------------|
| $z_{1s_1}(it\_no=1-5)$ |          |           |            |
| $z_{1s_2}(it\_no=1-5)$ |          |           |            |
| $z_{1s_3}(it\_no=1-5)$ |          |           |            |
| ...                    |          |           |            |
| ...                    |          |           |            |
| $z_{1s_N}(it\_no=1-5)$ |          |           |            |

Der zweite DMA-Kanal DMA\_KAN2 dient der Übertragung der Ergebniswerte (Ausgabedaten)  $z_{1n}$  und  $z_{2n}$  der Symbolschätzer

20 MAP1 bzw. MAP2 zum DSP.

Fig. 6 zeigt ein Blockschaltbild des MAP-Decodierers MAP\_DEC.

Die über den Datenbus DB übertragenen Daten werden einer internen Schnittstelle IF übergeben.

- 5 Neben der internen Schnittstelle IF und den beiden MAP-Datenspeichern MAP\_RAM1 (z.B. N x 16 Bit) und MAP\_RAM2 (z.B. N x 8 Bit) umfaßt der MAP-Decodierer MAP\_DEC einen Eingabedatenspeicher INP\_APR, einen Ausgabedatenspeicher OUT\_LLRL, ein Konfigurations-Register V\_CONF, eine DMA-Anforderungseinheit
- 10 DMA\_REQ zur Erzeugung der Anforderungs-Steuersignale V\_TRQ\_C und V\_TRQ\_M und ein Modul zum Erzeugen von standardspezifischer Code-Information (Trellis-Information) MAP\_ST. Ferner umfaßt der MAP-Decodierer MAP\_DEC eine Steuereinheit ST\_TRANS zur Steuerung der Berechnung von Übergangs-Metrikwerten, eine
- 15 Steuereinheit SL\_WIN\_FD zur Steuerung der Berechnung von Vorwärtsrekursions-Metrikwerten, eine Steuereinheit SL\_WIN\_BD zur Steuerung der Berechnung von Rückwärtsrekursions-Metrikwerten und eine Steuereinheit LLR\_RAT zur Steuerung der Berechnung von Werten der ersten und zweiten Zuverlässigkeits-
- 20 information  $z1_n(it\_no)$  bzw.  $z2_n(it\_no)$ . Die vier Steuereinheiten stehen über einen internen Datenbus IDB mit einem drei Hardware-Rechenbausteine RB1, RB2, RB3 umfassenden Rechenmittel RM in Datenaustauschverbindung. Ferner kann über den internen Datenbus IDB auf zwei Datenspeicher für Vorwärtsrekursions-Metrikwerte, nämlich RAM\_FD\_0 und RAM\_FD\_1, beide z.B.
- 25 20 x 128 Bit groß, zugegriffen werden. RAM\_FD\_0 beinhaltet Vorwärtsrekursions-Metrikwerte, die aufgrund eines Übergangs mit einem Eingangsbit von 0 erzeugt werden. RAM\_FD\_1 beinhaltet Vorwärtsrekursions-Metrikwerte, die aufgrund eines Übergangs mit einem Eingangsbit von 1 erzeugt werden.
- 30

Die Funktionsweise des in Fig. 6 dargestellten MAP-Decodierers ist wie folgt:

- 35 Im Rahmen eines Initialisierungsvorgangs wird dem MAP-Decodierer MAP\_DEC über den Datenbus DB zunächst mitgeteilt, welcher Code der senderseitigen Codierung des zu übertragenden

Datensignals zugrundeliegt. Unter den verfügbaren Codes ist erfindungsgemäß zumindest ein Turbo-Code. Entsprechend dem zugrundliegenden Code werden in dem Modul MAP\_ST codespezifische Eingangsdatenwerte für den ersten Rechenbaustein RB1 erzeugt. Dies wird in Verbindung mit Fig. 8 noch näher erläutert.

Zur Decodierung eines gesamten Datenblocks werden zunächst die Datenwerte  $\hat{x}_n$ ,  $\hat{y}_{1n}$  und  $\hat{y}_{2n}$  der (im DSP bereits statistisch aufbereiteten) systematischen Information und der ersten und zweiten Redundanz-Information für einen Datenblock ( $n = 1, \dots, N$ ) in die Datenspeicher MAP\_RAM1/2 geladen.

Zu Beginn der Decodierung arbeitet der MAP-Decodierer MAP\_DEC im MAP1-Modus. Auf der Basis des Speicherinhalts von MAP\_RAM1 und der Anfangs-Rückkoppel-Information  $z_{2n}(it\_no = 0)$  werden unter der Steuerung der Steuereinheit ST\_TRANS in dem ersten Rechenbaustein RB1 Übergangs-Metrikwerte, unter der Steuerung der Steuereinheiten SL\_WIN\_FD und SL\_WIN\_BD in dem zweiten Rechenbaustein RB2 Vorwärts- und Rückwärtsrekursions-Metrikwerte und unter der Steuerung der Steuereinheit LLR\_RAT mittels des dritten Rechenbausteins RB3 die logarithmischen Werte der ersten Zuverlässigkeitsinformation bezüglich des ersten Iterationsdurchlaufs  $z_{1n}(it\_no = 1)$  erzeugt. Diese Werte werden über den Ausgabedatenspeicher OUT\_LL\_R, die interne Schnittstelle IF und den Datenbus DB dem DSP zugeleitet.

Wie bereits anhand von Fig. 4 beschrieben, werden die von dem DSP erzeugten (verschachtelten und statistisch aufbereiteten) Datenwerte  $z_{1n}(it\_no = 1)$  nach Anforderung mittels des Steuersignals V\_TRQ\_C zum MAP-Decodierer MAP\_DEC zurückgegeben und zunächst in dem Eingabedatenspeicher INP\_APR zwischengespeichert.

Der MAP-Decodierer MAP\_DEC arbeitet nun im MAP2-Modus, d.h., daß die vorstehend beschriebene Berechnungsprozedur jetzt auf der Grundlage der im zweiten MAP-Datenspeicher MAP\_RAM2 ge-

speicherten Datenwerte ( $\hat{y}2s_n$ ) und den in dem Eingabedaten-  
 speicher INP\_APR zwischengespeicherten Werte durchgeführt  
 wird. Die Ausgabe der Werte der zweiten Zuverlässigkeits-  
 information  $z2_n(it\_no = 1)$  erfolgt wiederum über den Ausgabe-  
 5 datenspeicher OUT\_LLRL.

Nachfolgend wird die Berechnung der LLRs mittels Vorwärts-  
 und Rückwärtsrekursion anhand der Berechnung der ersten Zu-  
 verlässigkeitsinformation  $z1_n$  näher erläutert.

10

Jeder Wert  $z1_n$  ist ein wertekontinuierliches, logarithmisches  
 Wahrscheinlichkeitsverhältnis für das uncodierte Datensymbol  
 $u_n$  der Eingabesignalfolge U,

$$15 \quad z1_n = \ln \left\{ \frac{P(u_n = 1 | \hat{X}, \hat{Y}1, Z2)}{P(u_n = 0 | \hat{X}, \hat{Y}1, Z2)} \right\} \quad (4)$$

Mit  $P(u_n = 1 | \hat{X}, \hat{Y}1, Z2)$  bzw.  $P(u_n = 0 | \hat{X}, \hat{Y}1, Z2)$  werden die be-  
 dingten Wahrscheinlichkeiten dafür bezeichnet, daß das Daten-  
 symbol  $u_n$  gleich 1 bzw. gleich 0 ist unter der Bedingung, daß  
 20 die Folgen  $\hat{X}, \hat{Y}1, Z2$  beobachtet werden. Diese bedingten Wahr-  
 scheinlichkeiten sind "Wahrscheinlichkeiten a-Posteriori", da  
 von einem eingetretenen Ereignis (den "gemessenen" Folgen  
 $\hat{X}, \hat{Y}1, Z2$ ) auf die Wahrscheinlichkeiten der diesem Ereignis  
 zugrundeliegenden, uncodierten Datensymbole  $u_1$  bis  $u_N$  rückge-  
 25 schlossen wird.

Der Zustand des senderseitigen Faltungscodierers RSC1 zum  
 Zeitpunkt n (d.h. beim Eingabe-Datensymbol  $u_n$ ) wird mit  $S_n$   
 bezeichnet.

30

Die bedingten a-Posteriori-Wahrscheinlichkeiten in der Glei-  
 chung (4) lassen sich als Summen von Einzel-a-Posteriori-  
 Wahrscheinlichkeiten über die  $M = 2^{L-1}$  möglichen Zustände  $S_n =$   
 $0, 1, \dots, M-1$  des Codierers RSC1 ausdrücken, wobei mit L die  
 35 Rückgriffstiefe des erzeugten Turbo-Codes bezeichnet wird:

$$z1_n = \ln \left\{ \frac{\sum_{m=0}^{M-1} P(u_n = 1, S_n = m | \hat{X}, \hat{Y}_1, Z_2)}{\sum_{m=0}^{M-1} P(u_n = 0, S_n = m | \hat{X}, \hat{Y}_1, Z_2)} \right\} \quad (5)$$

Die Einzelwahrscheinlichkeiten können in der folgenden Form  
5 geschrieben werden

$$P(u_n = i, S_n = m | \hat{X}, \hat{Y}_1, Z_2) = \alpha_n^i(m) \cdot \beta_n(m) \quad i = 0, 1$$

mit

$$\begin{aligned} 10 \quad \alpha_n^i(m) &= P(u_n = i, S_n = m | R_1^N) \\ \beta_n(m) &= \frac{P(R_{n+1}^N | S_n = m)}{P(R_{n+1}^N | R_1^N)} \end{aligned} \quad (6)$$

wobei zur Vereinfachung der Schreibweise die Folge

$$R_v^\mu = (R_v, \dots, R_\mu), \quad 1 \leq v < \mu \leq N \quad (7)$$

15

bestehend aus den Wertetripeln  $R_n = (\hat{x}_n, \hat{y}_{1n}, z_{2n})$  von systematischer Information, Redundanz-Information und Rückkoppel-Information definiert wird.

20 Der Ausdrücke  $\alpha_n^i(m)$  und  $\beta_n(m)$  werden als Vorwärtsrekursions- und Rückwärtsrekursions-Metriken bezeichnet, da sie unter der Voraussetzung, daß die Übergangs-Metrikwerte  $\gamma_n^i(R_n, m', m)$  bekannt sind, rekursiv ermittelt werden können. Die Übergangs-Metrikwerte  $\gamma_n^i(R_n, m', m)$  sind die Übergangswahrscheinlichkeiten von einem ersten Zustand  $S_{n-1} = m'$  in einen zweiten Zustand  $S_n = m$  des Codierers RSC1 im Trellis-Diagramm, d.h.

25

$$\gamma_n^i(R_n, m', m) = P(u_n = i, S_n = m, R_n | S_{n-1} = m') \quad (8)$$

Eine detaillierte Beschreibung der Rekursionen (bei Verwendung einer optimalen MAP-Symbolschätzung) ist in dem Kapitel  
30 E.3.3 "Rekursive MAP-Symbolschätzung" des genannten Buchs von

- P. Jung auf den Seiten 353 bis 361 angegeben. Die Rekursionen laufen herkömmlicherweise über den gesamten Block, d.h. die Vorwärtsrekursion beginnt beim Zeitpunkt  $n = 1$  (erstes Bit der Folgen  $\hat{X}, \hat{Y}_1, Z_2$ : MAP1, bzw. der Folgen  $\hat{Y}_2, Z_1$ : MAP2) und
- 5 endet im Zeitpunkt  $n = N$  (letztes Bit der Folgen  $\hat{X}, \hat{Y}_1, Z_2$ : MAP1, bzw. der Folgen  $\hat{Y}_2, Z_1$ : MAP2), und die Rückwärtsrekursion beginnt beim Zeitpunkt  $n = N$  und endet im Zeitpunkt  $n = 1$ .
- 10 Demgegenüber werden bei dem hier beschriebenen Ausführungsbeispiel des MAP-Decodierers MAP\_DEC zur Berechnung der LLRs  $z_{1n}$  und  $z_{2n}$  pro Datenblock vorzugsweise mehrere Rückwärts- und Vorwärtsrekursionen durchgeführt, die sich jeweils (nur) über Teilintervalle des Datenblocks erstrecken.
- 15 In der Steuereinheit SL\_WIN\_FD wird eine Intervalllänge (Fenstergröße) für die Vorwärtsrekursionen von beispielsweise DV = 20 festgelegt. Beginnend bei  $n = 0$  werden in einem ersten Vorwärtsrekursionslauf VR1 (siehe Fig. 7) in dem zweiten
- 20 Rechenbaustein RB2 zunächst z.B. die ersten 20 Vorwärtsrekursions-Metrikwerte Werte  $\alpha_0^i(m), \dots, \alpha_{19}^i(m)$  bestimmt und wie bereits erwähnt in den Vorwärtsrekursions-Speicherbereichen RAM\_FD\_0 (für  $i = 0$ ) bzw. RAM\_FD\_1 (für  $i = 1$ ) zwischengespeichert.
- 25 Für die zugehörige erste Rückwärtsrekursion RR1 sei in SL\_WIN\_BD eine Intervalllänge für Rückwärtsrekursionen (Fenstergröße) von beispielsweise DR = 80 festgelegt. (Vorzugsweise bestimmt sich die Länge des Rückwärtsrekursionsfensters gemäß der Beziehung  $DR = L \times DV$ .) Somit beginnt
- 30 die erste Rückwärtsrekursion unter der Steuerung von SL\_WIN\_BD bei  $n = 79$  und läuft bis  $n = 0$ . Es werden dabei in dem zweiten Rechenbaustein RB2 die entsprechenden Rückwärtsrekursions-Metrikwerte  $\beta_n(m)$  berechnet.
- 35 Zur Berechnung der LLRs im Datenblocksegment  $n = 0, 1, \dots, 19$  werden sämtliche 20 berechneten Werte für  $\alpha_n^i(m)$  und die letz-

ten zwanzig berechneten Werte für  $\beta_n(m)$  herangezogen. Die Berechnung erfolgt wie bereits erwähnt im dritten Rechenbaustein RB3 unter der Steuerung von LLR\_RAT. Eine Speicherung der Rückwärtsrekursions-Metrikwerte ist dabei nicht erforderlich, da sie zur Berechnung der LLRs sofort weiterverarbeitet werden können.

Nach der Berechnung der ersten 20 LLRs werden die beiden Rekursionsfenster jeweils um 20 Werte "nach rechts" verschoben.

Die zweite Vorwärtsrekursion VR2 beginnt demzufolge bei  $n = 20$  und läuft bis  $n = 39$ . Bei der Zwischenspeicherung der Werte  $\alpha_{20}^i(m), \dots, \alpha_{39}^i(m)$  können die bei der ersten Vorwärtsrekursion VR1 gewonnenen Werte in RAM\_FD\_0 bzw. RAM\_FD\_1 überschrieben werden. Die zugehörige zweite Rückwärtsrekursion RR2 beginnt bei  $n = 99$  und läuft bis  $n = 20$  zurück. Die letzten 20 berechneten Werte für  $\beta_n(m)$  können wiederum ohne Abspeicherung direkt an den Rechenbaustein RB3 übergeben werden. Zur Berechnung der LLRs im Blocksegment  $n = 20, 21, \dots, 39$  werden auch wieder sämtliche 20 berechneten Werte für  $\alpha_n^i(m)$  und die letzten zwanzig berechneten Werte für  $\beta_n(m)$  herangezogen.

Diese segmentweise Bestimmung der LLRs mit gleitenden Vorwärts- und Rückwärtsrekursionsfenstern wird, von LLR\_RAT gesteuert, in der beschriebenen Weise fortgeführt, bis sämtliche LLRs des Datenblockes berechnet sind. Günstig ist der geringe Speicherplatzbedarf für RAM\_FD\_0 und RAM\_FD\_1 sowie die Tatsache, daß die Rückwärtsrekursions-Metrikwerte nicht abgespeichert werden müssen.

Nachfolgend wird die näherungsweise Berechnung der Übergangsmetrikwerte, der Vorwärts- und Rückwärtsrekursions-Metrikwerte und der LLR's in dem Rechenmittel RM anhand der Fig. 8, 9a-d und 10 näher erläutert.

Das Modul MAP\_ST umfaßt drei Untermodule TREL\_UMTS\_CONV, TREL\_UMTS\_TURB, und TREL\_GSM\_CONV/TREL\_EDGE\_CONV. Alle Untermodule enthalten einen eigenen Zustandsgenerator FSM.

- 5 Eine Steuereinheit ST dient zur zeitlichen Steuerung und Koordination der Zustandsgeneratoren FSM und der Rechenbausteine RB1, RB2, RB3. Hierzu steht die Steuereinheit ST über Steuerdatenleitungen S1, S2 und S3 mit den Zustandsgeneratoren FSM der Untermodule TREL\_UMTS\_CONV, TREL\_UMTS\_TURB,  
10 TREL\_GSM\_CONV/TREL\_EDGE\_CONV in Verbindung und steuert ferner über eine weitere Steuerdatenleitung S4 den ersten Rechenbaustein RB1. Weitere Steuerdatenleitungen führen zu den anderen Rechenbausteinen RB2, RB3.
- 15 Jedes Untermodul von MAP\_ST berechnet theoretische Information (wird auch als sogenannte Trellis-Information bezeichnet), welche für einen bestimmten Code charakteristisch ist. Die Berechnung der theoretischen Information erfolgt jeweils auf der Grundlage eines Code-spezifischen Polynoms. Den in  
20 den Untermodulen TREL\_GSM\_CONV bzw. TREL\_EDGE\_CONV berechneten Werten der theoretischen Information liegen die GSM-(Global System for Mobile Communication-) und EDGE-(enhanced data services for GSM evolution-)Standards zugrunde. Den von dem Untermodul TREL\_UMTS\_TURB berechneten theoretischen Werten  
25 t1, t2, t3 liegt der UMTS-(Universal Mobile Telecommunications Systems-)Standard für Turbo-Decodierung zugrunde.

Im folgenden wird nur der Fall der Anwahl des Untermoduls TREL\_UMTS\_TURB durch V\_CONF betrachtet.

30

- In UMTS ist ein Turbo-Code der Rückgriffstiefe  $L = 4$  vorgesehen. Folglich kann das Schieberegister des Faltungscodierers RSC1 im Turbo-Codierer (siehe Fig. 3) genau 8 Zustände einnehmen, nämlich 000, 001, 010, ... usw. In den Fig. 9a und 9b  
35 ist jeder dieser 8 möglichen Zustände durch ein mit den Index-Zahlen  $m = 0$  bis 7 durchnummeriertes Kästchen dargestellt.

Fig. 9a zeigt die Zustandsübergänge  $m'$  zu  $m$ , die auftreten, wenn der zunächst (bei Schritt  $k-1$ ) in einem (beliebigen) Ausgangszustand befindliche Turbo-Codierer durch Hinzufügen eines Eingabebits  $a_k$  des Wertes 0 (d.h.  $i = 0$ ) in einen darauffolgenden Zustand (bei Schritt  $k$ ) gebracht wird. Ein entsprechendes Schaubild, das die bei Hinzufügen eines Eingabebits  $a_k$  des Wertes 1 (d.h.  $i = 1$ ) auftretenden Zustandsübergänge darstellt, ist in Fig. 9b gezeigt.

10

Für jeden möglichen Zustandsübergang (von  $k-1$  nach  $k$ ) werden drei theoretische Werte  $t1(i)$ ,  $t2(i)$ ,  $t3(i)$  berechnet.

15

Die Werte  $t1(i)$  und  $t3(i)$  sind identisch mit dem hinzugefügten Eingabebit  $a_k$  (d.h.  $t1(i) = t3(i) = i$ ). Der Wert  $t2$  wird durch das Code-spezifische Polynom berechnet.

20

Eine mögliche Berechnungsschaltung für den Wert  $t2$  aus  $a_k$  ist in Fig. 9d dargestellt. Die Berechnungsschaltung umfaßt ein aus drei Speicherzellen  $T$  bestehendes Schieberegister und vier Addierer  $ADD$ , die in der dargestellten Weise in der Schaltung angeordnet sind. Das Schieberegister wird von dem Zustandsgenerator  $FSM$  mit einem Ausgangszustand (z.B.  $m' = 0$ , d.h. (000)) belegt und mit einem Eingabebit  $a_k$  versorgt. Bei Hinzufügen des Eingabebits  $a_k = i$  werden die Werte  $t2(i)$ ,  $i = 0, 1$ , bezogen auf den jeweiligen Ausgangszustand, z.B.  $m' = 0$ , generiert. Dieser Vorgang wird für alle Ausgangszustände des Schieberegisters durchgeführt.

25

30 Nachfolgend wird anhand Fig. 10 der Aufbau der Rechenbausteine  $RB1$ ,  $RB2$  und  $RB3$  erläutert. Die Rechenbausteine  $RB1$ ,  $RB2$ ,  $RB3$  sind schnelle Datenpfade, die komplett aus kombinatorischer Logik aufgebaut sind.

35

Der zur Berechnung der Übergangs-Metrikwerte  $\gamma_n^i(R_n, m', m)$  vorgesehene erste Rechenbaustein  $RB1$  nimmt die für  $a_k = i$ ,  $i = 0, 1$  erzeugten theoretischen Werte  $t1(i)$ ,  $t2(i)$ ,  $t3(i)$  entgegen.

gen und setzt diese in einem Schaltkreis GW in Gewichtungsfaktoren  $w1(i)$ ,  $w2(i)$ ,  $w3(i)$  um. Die Gewichtungsfaktoren  $w1(i)$ ,  $w2(i)$ ,  $w3(i)$  berechnen sich nach:

$$\begin{aligned} 5 \quad w1(i) &= 1-2 \cdot t1(i) \\ w2(i) &= 1-2 \cdot t2(i) \\ w3(i) &= 1-2 \cdot t3(i) \end{aligned} \quad (9)$$

Die Gewichtungsfaktoren  $w1(i)$ ,  $w2(i)$ ,  $w3(i)$  sind Vorzeichenwerte, d.h. +1, -1. Mit diesen Vorzeichenwerten werden nun durch  
10 Multiplikation mit den "Meßwerten" die Übergangs-Metrikwerte  $\gamma_n^i(R_n, m', m)$  berechnet.

Im MAP1-Modus stehen als Meßwerte  $\hat{x}s_n$ ,  $\hat{y}1s_n$  und die Rückkop-  
15 pel-Information  $z2s_n(it\_no)$  zur Verfügung. Die Berechnung der Übergangs-Metrik eines bestimmten der insgesamt 16 Übergänge (siehe Fig. 9a und 9b) erfolgt nach der Gleichung

$$\gamma_n^i(R_n, m', m) = w1(i) \cdot \hat{x}s_n + w2(i) \cdot \hat{y}1s_n + w3(i) \cdot z2s_n(it\_no)$$

20 (10)

Da  $w1(i)$ ,  $w2(i)$ ,  $w3(i)$  Vorzeichenwerte sind, kann  $\gamma_n^i(R_n, m', m)$   
maximal 8 verschiedene Werte annehmen. Da diese Werte symme-  
trisch zu 0 sind, müssen nur 4 Werte und ihre jeweiligen ne-  
25 gierten Werte berechnet werden.

Diese Berechnungen werden parallel, d.h. gleichzeitig mit 4  
Addierern TR\_ADD\_1, TR\_ADD\_2, ..., TR\_ADD\_4 (von denen jeder  
gemäß Gleichung (10) drei Eingänge aufweist) und der nachge-  
30 schalteten Zweier-Komplement-Stufe 2K in RB1 durchgeführt.  
Die Zweier-Komplement-Stufe 2K bildet zu einem berechneten  
Wert den entsprechenden negierten Wert.

Der Zweier-Komplement-Stufe 2K ist ein Demultiplexer (nicht  
35 dargestellt) nachgeschaltet. Dieser verteilt die 8 berechne-  
ten Übergangs-Metrikwerte  $\gamma_n^i(R_n, m', m)$  auf 4 Speicher BF1,  
BF2, ..., BF4 mit jeweils 4 Speicherplätzen.

Die Verteilung wird so vorgenommen, daß in jedem Speicher BF1-4 die Übergangs-Metrikwerte  $\gamma_n^i(R_n, m', m)$  eines "Butterfly" abgespeichert sind. Ein Butterfly bezeichnet eine Gruppe bestehend aus 4 bestimmten Übergangs-Metrikwerten  $\gamma_n^i(R_n, m', m)$ .  
 5 In Fig. 9c sind die 4 Butterflies dargestellt. Der erste Butterfly B1 umfaßt die 4 Übergangs-Metrikwerte, die auf die Endzustände 0 und 1 hinführen, der zweite Butterfly B2 umfaßt die 4 Übergangs-Metrikwerte, die auf die Endzustände 2 und 3 hinführen, ..., und der vierte Butterfly B4 umfaßt die 4 Übergangs-Metrikwerte, die auf die Endzustände 6 und 7 hinführen.  
 10

Im MAP2-Modus stehen lediglich zwei "Meßwerte"  $\hat{y}_{2s_n}$  und  $z_{1s_n}(it\_no)$  zur Verfügung. Die Berechnung des Übergangs-Metrikwertes eines bestimmten der insgesamt 16 Übergänge erfolgt nach der Gleichung:  
 15

$$\gamma_n^i(G_n, m', m) = w2(i) \cdot \hat{y}_{2s_n} + w3(i) \cdot z_{1s_n}(it\_no) \quad (11)$$

20 (hierbei wird mit  $G_n$  die Folge bestehend aus den Wertepaaren  $(z_{1n}, \hat{y}_{2n})$  bezeichnet).

Da nur zwei Vorzeichenwerte  $w2(i)$ ,  $w3(i)$  zu berücksichtigen sind, kann  $\gamma_n^i(G_n, m', m)$  maximal 4 verschiedene Werte annehmen.  
 25 Diese Werte können entweder von den 4 Addierern TR\_ADD\_1-4 direkt oder durch 2 Addierer und die nachgeschaltete Zweier-Komplement-Stufe 2K berechnet werden.

Der weitere Verarbeitungsweg ist wie beim MAP1-Modus.  
 30

Die Steuerung des Rechenbausteins RB1 zur Kombination der theoretischen Werte (bzw. der Gewichtungsfaktoren) mit den "Meßwerten" wechselseitig gemäß den Gleichungen (10) und (11) erfolgt durch die Steuereinheit ST\_TRANS.  
 35

Die 16 berechneten Übergangs-Metrikwerte werden durch vier parallele Datenbusleitungen (jeweils eine Datenbusleitung pro Speicher BF1-4) dem zweiten Rechenbaustein RB2 zugeleitet.

- 5 Der zweite Rechenbaustein RB2 führt die rekursive Berechnung der Vorwärts- und Rückwärtsrekursions-Metrikwerte durch. Er weist 16 parallel angeordnete Addierer ADD\_1, ADD\_2, ..., ADD\_16, 8 parallel angeordnete Maximum-Einheiten MAX1, MAX2, ..., MAX8 und 4 parallel angeordnete Zwischenspeicher ZS1, 10 ZS2, ZS3 und ZS4 auf.

- Jeder der Addierer ADD\_1-4 addiert einen Übergangs-Metrikwert zu einem Vorwärts- oder Rückwärtsrekursions-Metrikwert, welcher im vorhergehenden Rekursionsschritt bestimmt wurde - 15 d.h. im Fall der Vorwärtsrekursion zu einem Ausgangszustand und im Fall der Rückwärtsrekursion zu einem Endzustand.

- Für die Berechnung der Vorwärtsrekursions-Metrikwerte betrachten wir zunächst exemplarisch das Butterfly B1. Der erste Addierer ADD\_1 addiert zu dem im vorhergehenden Rekursionsschritt errechneten Vorwärtsrekursions-Metrikwert 20  $\alpha_{n-1}(m' = 0)$  die Übergangs-Metrik  $\gamma_n^0(R_n, m' = 0, m = 0)$ , wobei sich ein erster möglicher Wert  $\alpha_n^0(m = 0)$  für den neuen Vorwärtsrekursions-Metrikwert  $\alpha_n(m = 0)$  ergibt, und der zweite 25 Addierer ADD\_2 addiert zu dem im vorhergehenden Rekursionsschritt errechneten Vorwärtsrekursions-Metrikwert  $\alpha_{n-1}(m' = 4)$  die Übergangs-Metrik  $\gamma_n^1(R_n, m' = 4, m = 0)$ , wobei sich ein zweiter möglicher Wert  $\alpha_n^1(m = 0)$  für den neuen Vorwärtsrekursions-Metrikwert  $\alpha_n(m = 0)$  ergibt. Der hochgestellte Index 30 der möglichen Vorwärtsrekursions-Metrikwerte gibt das den jeweiligen Übergang bewirkende Eingabebit an.

- Diese beiden Möglichkeiten, zum Endzustand  $m = 0$  zu gelangen, sind gemäß dem obersten in Fig. 9c dargestellten Butterfly B1 35 alternativ.

Die weiteren Addierer arbeiten in analoger Weise, d.h. die Addierer ADD\_3 und ADD\_4 berechnen entsprechende Ausdrücke bezüglich der beiden Alternativen, um zu dem anderen Endzustand  $m = 1$  des erstes Butterfly B1 zu gelangen, usw..

5

Allgemein gilt, daß jedem solchen alternative Übergangs-Metrikwerte verarbeitenden Paar von Addierern ADD\_1/2, ADD\_3/4, ..., ADD\_15/16 jeweils eine Maximum-Einheit MAX1, MAX2, ..., MAX8 nachgeschaltet ist. Die Maximum-Einheiten  
10 MAX1, MAX2, ..., MAX8 wählen durch Verwerfen jeweils des kleineren der beiden möglichen Werte  $\alpha_n^0(m)$ ,  $\alpha_n^1(m)$  für  $\alpha_n(m)$  den wahrscheinlicheren der beiden alternativen Übergänge aus. Diese Vorwärtsrekursions-Metrikwerte  $\alpha_n(m)$  bilden den Ausgangspunkt für den nächsten Rekursionsschritt  $n$  nach  $n+1$ .

15

In RB2 werden daher gleichzeitig 8 Vorwärtsrekursions-Metrikwerte  $\alpha_n(m)$  für 4 Butterflies in paralleler Datenverarbeitung bestimmt. Die 8 berechneten Vorwärtsrekursions-Metrikwerte  $\alpha_n(m)$  werden temporär zum Zwecke der Rekursionsberechnung in  
20 den Zwischenspeichern ZS1, ZS2, ZS3, ZS4 abgelegt. Die Ergebnisse  $\alpha_n^0(m)$  und  $\alpha_n^1(m)$  der Addierer ADD\_1-16 werden über den Datenbus IDB den Vorwärtsrekursions-Speicherbereichen RAM\_FD\_0 bzw. RAM\_FD\_1 zugeleitet und dort abgespeichert. Sie werden später im Rechenbaustein RB3 verwendet.

25

Bei der in UMTS eingesetzten Datenrate von 384 kbit/s müssen alle 16 möglichen Vorwärtsrekursions-Metrikwerte  $\alpha_n^0(m)$ ,  $\alpha_n^1(m)$  innerhalb eines 52 MHz-Zyklus berechnet und in den Vorwärtsrekursions-Speicherbereichen RAM\_FD\_0 und RAM\_FD\_1 gespeichert  
30 werden.

Die Berechnung der Rückwärtsrekursions-Metrikwerte erfolgt analog, jedoch auf der Grundlage von Übergängen von einem Endzustand  $m$  zu einem Anfangszustand  $m'$ . Die bei der Rückwärtsrekursion für den betrachteten Zeitpunkt  $n$  berechneten Rückwärtsrekursions-Metrikwerte werden ebenfalls temporär

35

(d.h. nur zum Zwecke der Rekursion zur Verwendung in dem jeweils nächsten Rekursionsschritt) in dem jeweiligen Zwischenspeicher ZS1, ZS2, ZS3 bzw. ZS4 abgelegt. Die Zwischenspeicher beinhalten somit die Vorwärts- und Rückwärtsrekursions-  
 5 Metrikwerte eines Butterflies für einen bestimmten Zeitpunkt n.

Eine Abspeicherung der berechneten Rückwärtsrekursions-Metrikwerte über den Rekursionslauf ist (anders wie bei den  
 10 Vorwärtsrekursions-Metrikwerten in RAM\_FD\_0 bzw. RAM\_FD\_1) nicht erforderlich, da die Rückwärtsrekursions-Metrikwerte im dritten Rechenbaustein RB3 sofort weiterverarbeitet werden können.

15 Der zur Berechnung der LLR-Ausgabewerte vorgesehene dritte Rechenbaustein RB3 umfaßt 16 Addierer L\_ADD\_1, L\_ADD\_2, ..., L\_ADD\_16, zwei Maximum-Einheiten MAX\_1 und MAX\_2 sowie einen Subtrahierer SUB.

20 Der dritte Rechenbaustein RB3 setzt die in der Gleichung (5) angegebene Beziehung gemäß einem Näherungsausdruck

$$\max_{m=0, \dots, 7} \{ \alpha_n^1(m) + \beta_n(m) \} - \max_{m=0, \dots, 7} \{ \alpha_n^0(m) + \beta_n(m) \} \text{ für die Berechnung}$$

der LLRs um. Der Näherungsausdruck ist in dem Artikel  
 "Comparision of Turbo-Code Decoders Applied to Short Frame  
 25 Transmission Systems", Peter Jung, IEEE Journal of Selected Areas in Communications, Vol. 14, No. 3, April 1996 vorgeschlagen, welcher durch Bezugnahme Gegenstand der vorliegenden Schrift wird. Die ersten 8 Addierer L\_ADD\_1-8 sind für die Addition der Vorwärtsrekursions-Metrikwerte zu  $i = 1$  mit  
 30 den zugehörigen Rückwärtsrekursions-Metrikwerten (d.h. für die Hypothese  $u_n = 1$ ) zuständig, d.h. jeder Addierer bildet für einen der 8 Zustände die Summe  $\alpha_n^1(m) + \beta_n(m)$ . Die zweiten 8 Addierer L\_ADD\_9-16 sind für die Addition der Vorwärtsrekursions-Metrikwerte zu  $i = 0$  mit den zugehörigen Rückwärtsrekursions-Metrikwerten (d.h. für die Hypothese  $u_n = 0$ ) zustän-  
 35

dig, d.h. jeder Addierer bildet für einen der 8 Zustände die Summe  $\alpha_n^0(m) + \beta_n(m)$ .

- Die Maximum-Einheit MAX\_1 nimmt die Ausgänge der ersten 8 Addierer L\_ADD\_1-8 entgegen und die Maximum-Einheit MAX\_2 nimmt die Ausgänge der zweiten 8 Addierer L\_ADD\_9-16 entgegen. Jede Maximum-Einheit MAX\_1 bzw. MAX\_2 findet durch Vergleich aller acht Addierer-Ausgänge den jeweils maximalen Addierer-Ausgang und leitet diesen dem Subtrahierer SUB weiter. Damit ist unter allen Übergängen verursacht durch ein Eingabebit  $i = 1$  durch MAX\_1 der Wert mit der maximalen Wahrscheinlichkeit und unter allen Übergängen verursacht durch ein Eingabebit  $i = 0$  durch MAX\_2 der Wert mit der maximalen Wahrscheinlichkeit bestimmt. Der Subtrahierer SUB berechnet unter Anwendung der Logarithmus-Eigenschaft  $\ln(a/b) = \ln(a) - \ln(b)$  als Differenz dieser Ergebniswerte die Zuverlässigkeitsinformation, d.h. den LLR-Wert  $z1_n(it\_no)$  (im MAP1-Modus) bzw. den LLR-Wert  $z2_n(it\_no)$  (im MAP2-Modus).
- Da sowohl in dem zweiten Rechenbaustein RB2 als auch in dem dritten Rechenbaustein RB3 jeweils 16 Addierer enthalten sind, können diese beiden Rechenbausteine auch in einem gemeinsamen Rechenbaustein mit insgesamt ebenfalls nur 16 Addierern zusammengefaßt werden. In diesem Fall werden die 16 Addierer gemultiplext betrieben, d.h. die Vorwärts- und Rückwärtsrekursions-Metrikwerte und die LLRs müssen alternierend berechnet werden. Der Vorteil besteht in einer Reduzierung des Hardware-Aufwands.

## Patentansprüche

1. Turbo-Decodierer zum Decodieren eines senderseitig mit einem Turbo-Code fehlerschutzcodierten, über einen gestörten  
5 Kanal übertragenen und in einem Empfänger (E) detektierten Datensignals ( $\hat{D}$ ), mit
- wenigstens einem Symbolschätzer (MAP\_DEC), welcher umfaßt:
    - einen Eingang (IF) zur Entgegennahme von auf das detek-
    - 10 tierte Datensignal ( $\hat{D}$ ) zurückgehenden Datensymbolen,
    - einen Ausgang zur Bereitstellung von Ausgabewerten (LLR),  
auf deren Basis geschätzte Datensymbole des senderseitig  
der Fehlerschutzcodierung zugrundeliegenden Datensignals  
(U) ermittelbar sind, und
    - ein Rechenmittel (RM), welches mit Kenntnis des sendersei-
    - 15 tig eingesetzten Fehlerschutzcodes
    - Übergangs-Metrikwerte,
    - Vorwärts- und Rückwärtsrekursions-Metrikwerte, und daraus
    - die Ausgabewerte (LLR)
    - berechnet,
    - 20 d a d u r c h g e k e n n z e i c h n e t,
    - daß das Rechenmittel (RM) einen aus kombinatorischer Logik  
aufgebauten Hardware-Rechenbaustein (RB1; RB2; RB3) zur Er-  
zeugung zumindest einer Sorte dieser Werte enthält.
- 25 2. Turbo-Decodierer nach Anspruch 1,  
d a d u r c h g e k e n n z e i c h n e t,
- daß das Rechenmittel (RM) einen ersten Hardware-Rechenbau-
  - stein (RB1) zur Erzeugung von Übergangs-Metrikwerten ent-
  - hält, welcher
  - 30 -- vier parallel angeordnete Addierer (TR\_ADD\_1-4) und wenig-
  - stens eine den Addierern nachgeschaltete Zweier-Komple-
  - ment-Stufe (2K) umfaßt.
3. Turbo-Decodierer nach Anspruch 1,  
35 d a d u r c h g e k e n n z e i c h n e t,

- daß das Rechenmittel (RM) einen ersten Hardware-Rechenbaustein (RB1) zur Erzeugung von Übergangs-Metrikwerten enthält, welcher
- acht parallel angeordnete Addierer umfaßt.

5

4. Turbo-Decodierer nach Anspruch 2 oder 3,  
dadurch gekennzeichnet,  
- daß der erste Hardware-Rechenbaustein (RB1) zur Erzeugung  
von Übergangs-Metrikwerten vier Speicher (BF) zur Abspei-  
10 cherung von jeweils vier Übergangs-Metrikwerten enthält.

5. Turbo-Decodierer nach einem der vorhergehenden Ansprüche,  
dadurch gekennzeichnet,  
- daß der Symbolschätzer (MAP\_DEC) ein Steuermittel  
15 (ST\_TRANS) zur Steuerung des ersten Hardware-Rechenbausteins (RB1) aufweist, welches bewirkt, daß der erste Rechenbaustein (RB1) die Übergangs-Metrikwerte wechselweise nach einer ersten und einer zweiten Rechenvorschrift bestimmt.

20

6. Turbo-Decodierer nach einem der vorhergehenden Ansprüche,  
dadurch gekennzeichnet,  
- daß das Rechenmittel (RM) einen zweiten Hardware-Rechenbaustein (RB2) zur Erzeugung von Vorwärts- und Rückwärtsrekursions-Metrikwerten enthält, welcher  
25 -- eine Additionsstufe bestehend aus einer Anzahl Q von parallel angeordneten Addierern (ADD\_1-16), und  
-- eine Maximierungsstufe bestehend aus parallel angeordneten Maximum-Einheiten (MAX1-8) umfaßt, wobei jede Maximum-  
30 Einheit (MAX1-8) an ihren Eingängen die Ausgänge von zwei Addierern (ADD-1-16) entgegennimmt und den größeren der Addiererausgänge an ihrem Ausgang ausgibt.

7. Turbo-Decodierer nach Anspruch 6,  
35 dadurch gekennzeichnet,  
- daß  $Q = 16$ .

8. Turbo-Decodierer nach einem der Ansprüche 6 oder 7,  
dadurch gekennzeichnet,  
- daß der Symbolschätzer (MAP\_DEC) ein Steuermittel  
(SL\_WIN\_FD, SL\_WIN\_BD) zur Steuerung des zweiten Hardware-  
5 Rechenbausteins (RB2) aufweist, welches bewirkt, daß zur  
Berechnung der Vorwärts- und Rückwärtsrekursions-Metrik-  
werte Vorwärts- und Rückwärtsrekursionsintervalle mit glei-  
tenden Intervallgrenzen innerhalb eines Datenblocks einge-  
setzt werden.
- 10 9. Turbo-Decodierer nach einem der vorhergehenden Ansprüche,  
dadurch gekennzeichnet,  
- daß das Rechenmittel (RM) einen dritten Hardware-Rechen-  
baustein (RB3) zur Erzeugung von Ausgabewerten (LLRs) um-  
15 faßt, welcher  
-- eine Additionsstufe bestehend aus einer Anzahl P von par-  
allel angeordneten Addierern (L\_ADD\_1-16), und  
-- eine Maximierungsstufe bestehend aus zwei parallel ange-  
ordneten Maximum-Einheiten (MAX\_1, MAX\_2) umfaßt, wobei  
20 jede Maximum-Einheit (MAX\_1, MAX\_2) an ihren Eingängen die  
Ausgänge von P/2 Addierern entgegennimmt und den größten  
der Addiererausgänge an ihrem Ausgang ausgibt.
10. Turbo-Decodierer nach Anspruch 9,  
25 dadurch gekennzeichnet,  
- daß  $P = 16$ .
11. Turbo-Decodierer nach Anspruch 9 oder 10,  
dadurch gekennzeichnet,  
30 - daß der dritte Hardware-Rechenbaustein (RB3) zur Erzeugung  
der Ausgabewerte (LLRs) einen Subtrahierer (SUB) zur Bil-  
dung der Differenz der Ausgänge der beiden Maximum-Einheit  
(MAX\_1, MAX\_2) aufweist.
- 35 12. Turbo-Decodierer nach einem der vorhergehenden Ansprüche,  
dadurch gekennzeichnet,

- daß der Symbolschätzer (MAP\_DEC) ein Steuermittel (LLR\_RAT) zur Steuerung des dritten Hardware-Rechenbausteins (RB3) aufweist, welches bewirkt, daß der dritte Rechenbaustein (RB3) Vorwärts- und Rückwärtsrekursions-Metrikwerte innerhalb von gleitenden Intervallgrenzen kombiniert.

13. Turbo-Decodierer nach einem der Ansprüche 6 bis 12, dadurch gekennzeichnet,
- daß der zweite Hardware-Rechenbaustein (RB2) zur Erzeugung von Vorwärts- und Rückwärtsrekursions-Metrikwerten und der dritte Hardware-Rechenbaustein (RB3) zur Erzeugung der Ausgabewerte (LLRs) gemeinsame, im Multiplexbetrieb arbeitende Addierer (ADD\_1-16 bzw. L\_ADD\_1-16) verwenden.
14. Turbo-Decodierer nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet,
- daß der Turbo-Decodierer (TDEC) ferner
    - einen digitalen Signalprozessor (DSP) zur
    - Durchführung von Turbo-Ver- und Entschachtelungsprozeduren und
    - zur Berechnung von für den momentanen Kanalzustand repräsentativer statistischer Information gemäß einem vorgegebenen Ablaufprogramm, und
    - eine bidirektionale Schnittstelle (FMI), über welche der digitalen Signalprozessor (DSP) mit dem Symbolschätzer (MAP\_DEC) in Datenaustauschverbindung steht, umfaßt.

15. Turbo-Decodierer nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet,
- daß der Symbolschätzer (MAP\_DEC) während eines Iterationsdurchlaufs sowohl zur Berechnung von erster als auch zur Berechnung von zweiter Zuverlässigkeitsinformation eingesetzt wird.

1/9

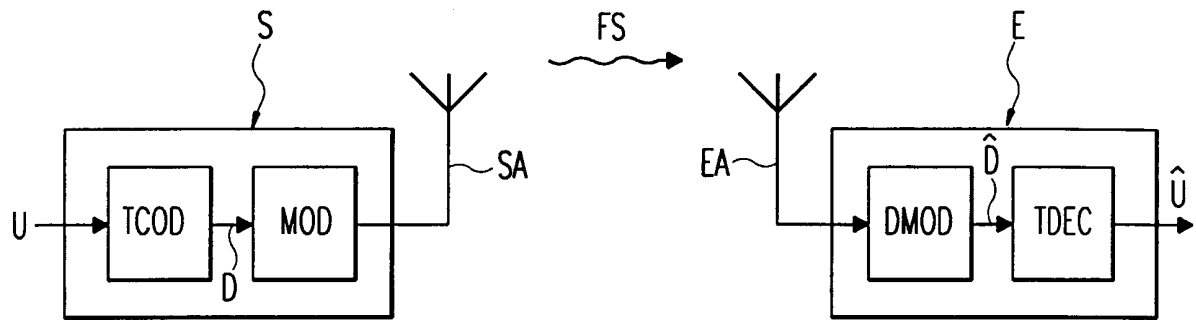


Fig. 1

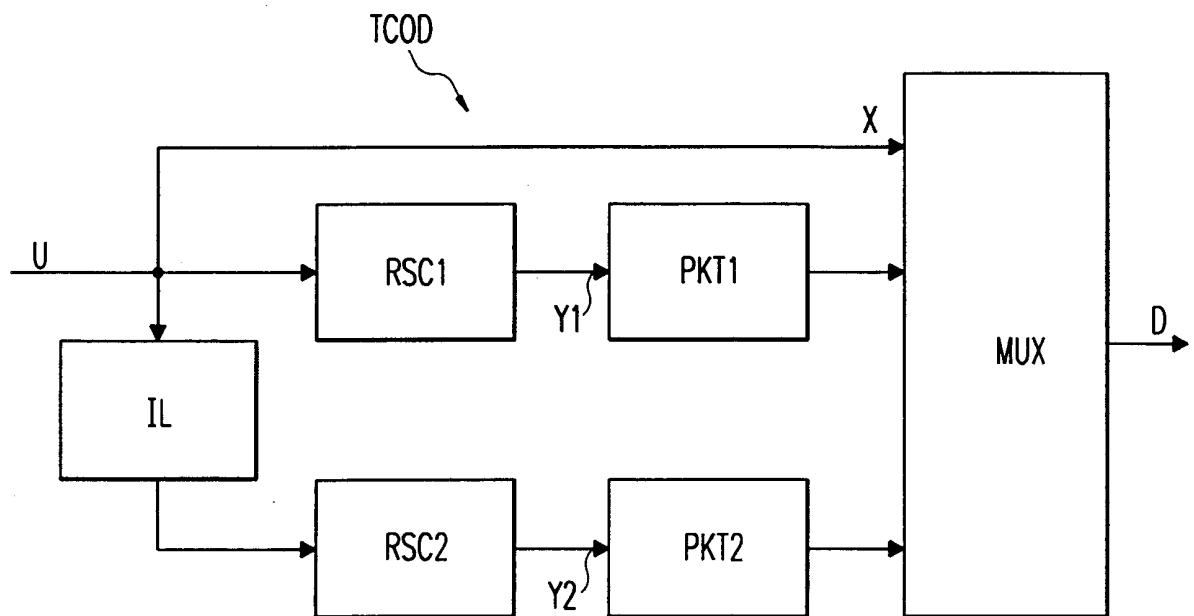


Fig. 2

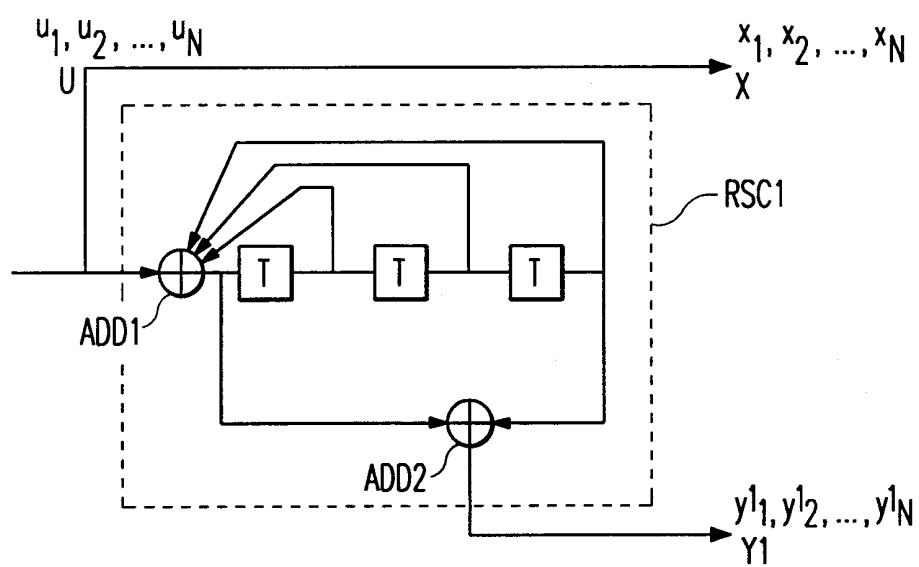


Fig. 3

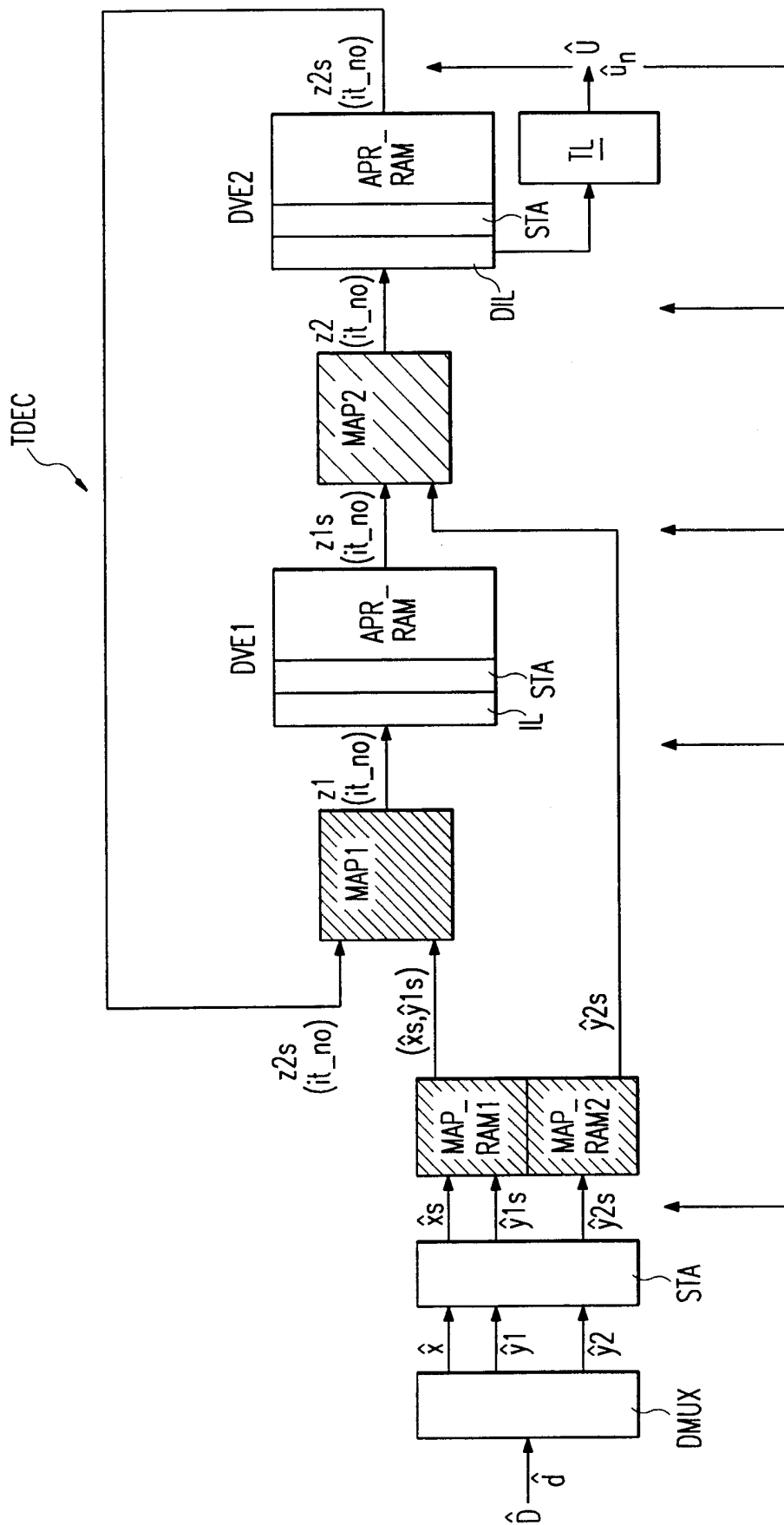


Fig. 4

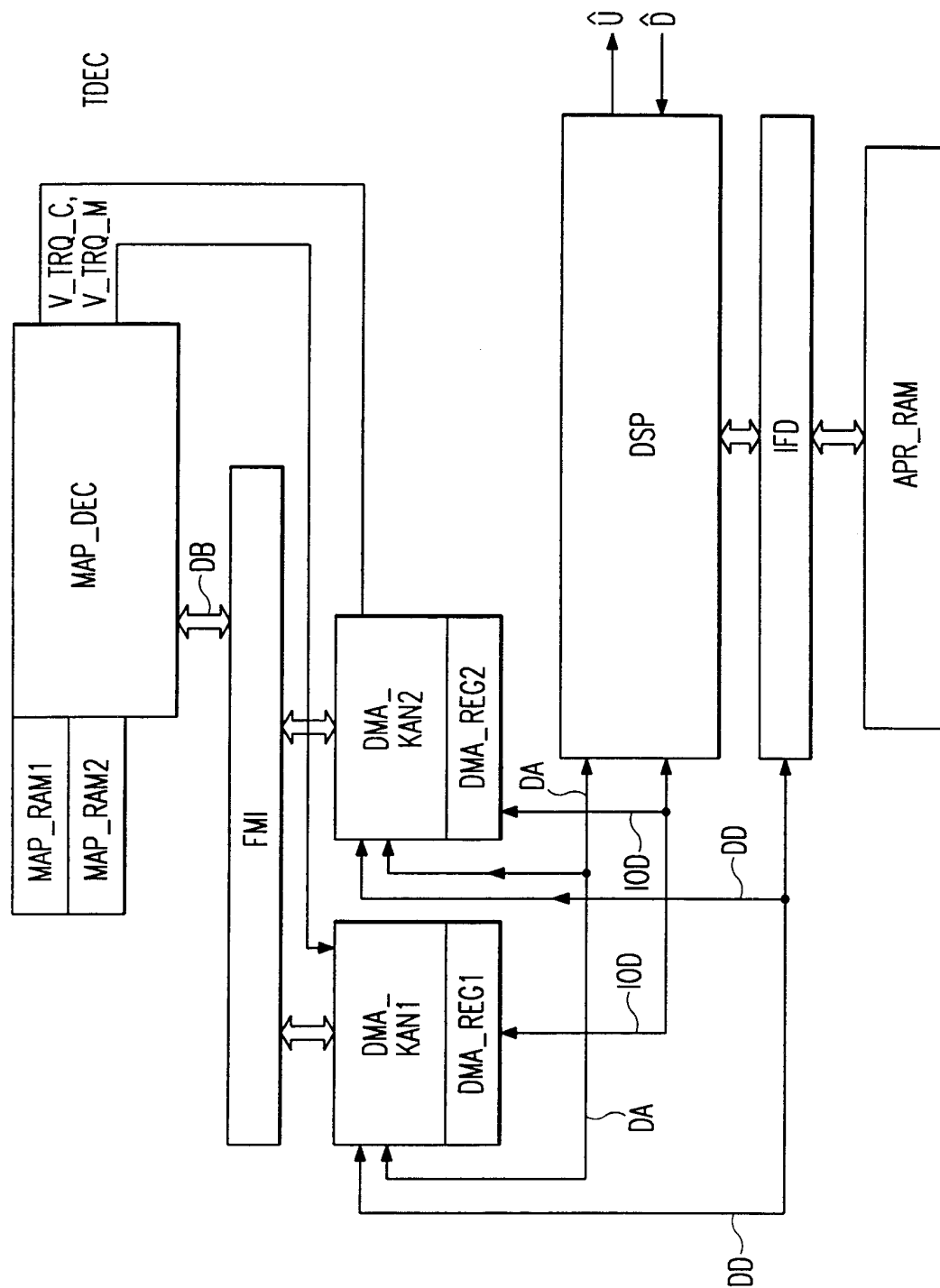


Fig. 5

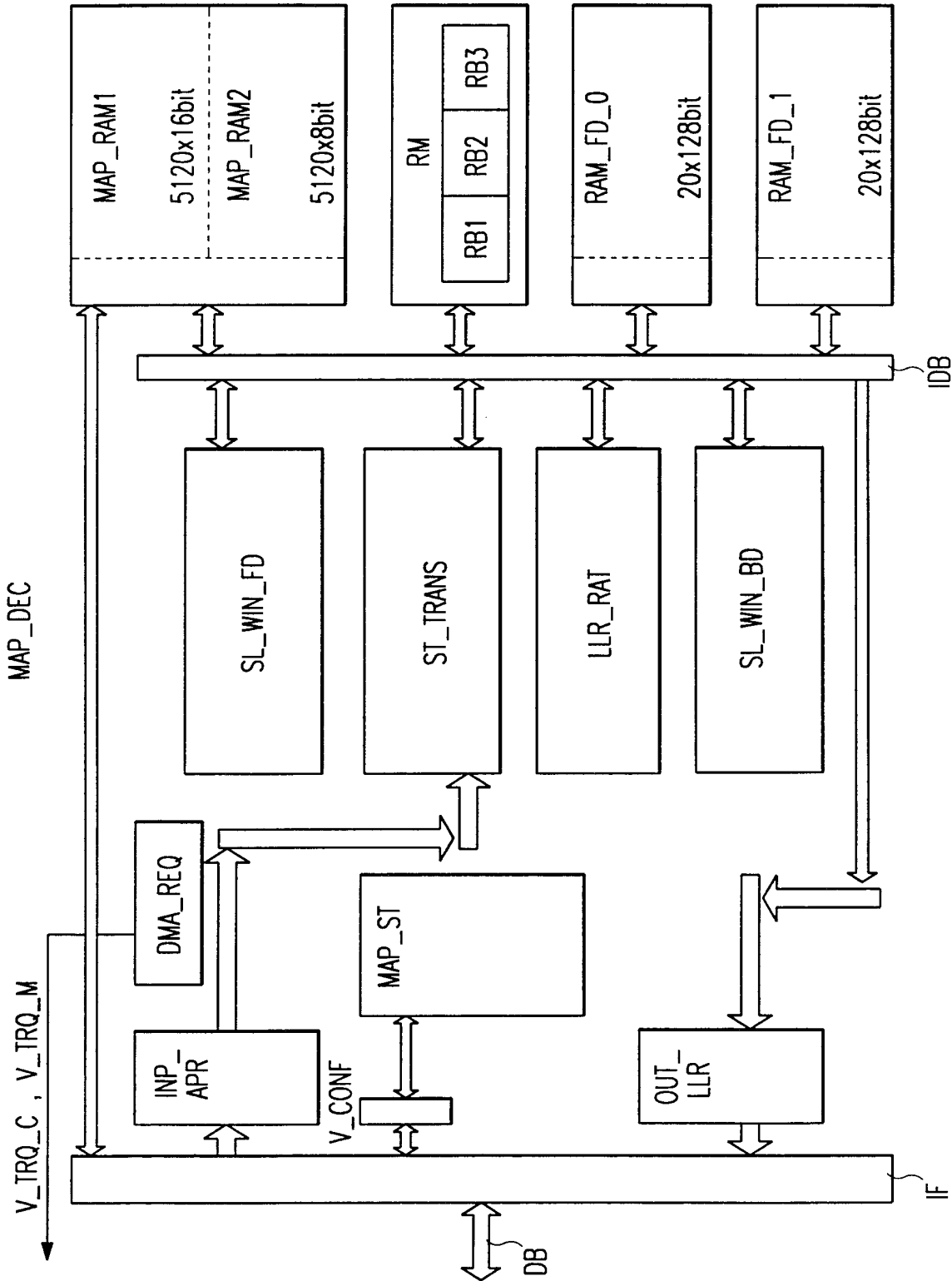


Fig. 6

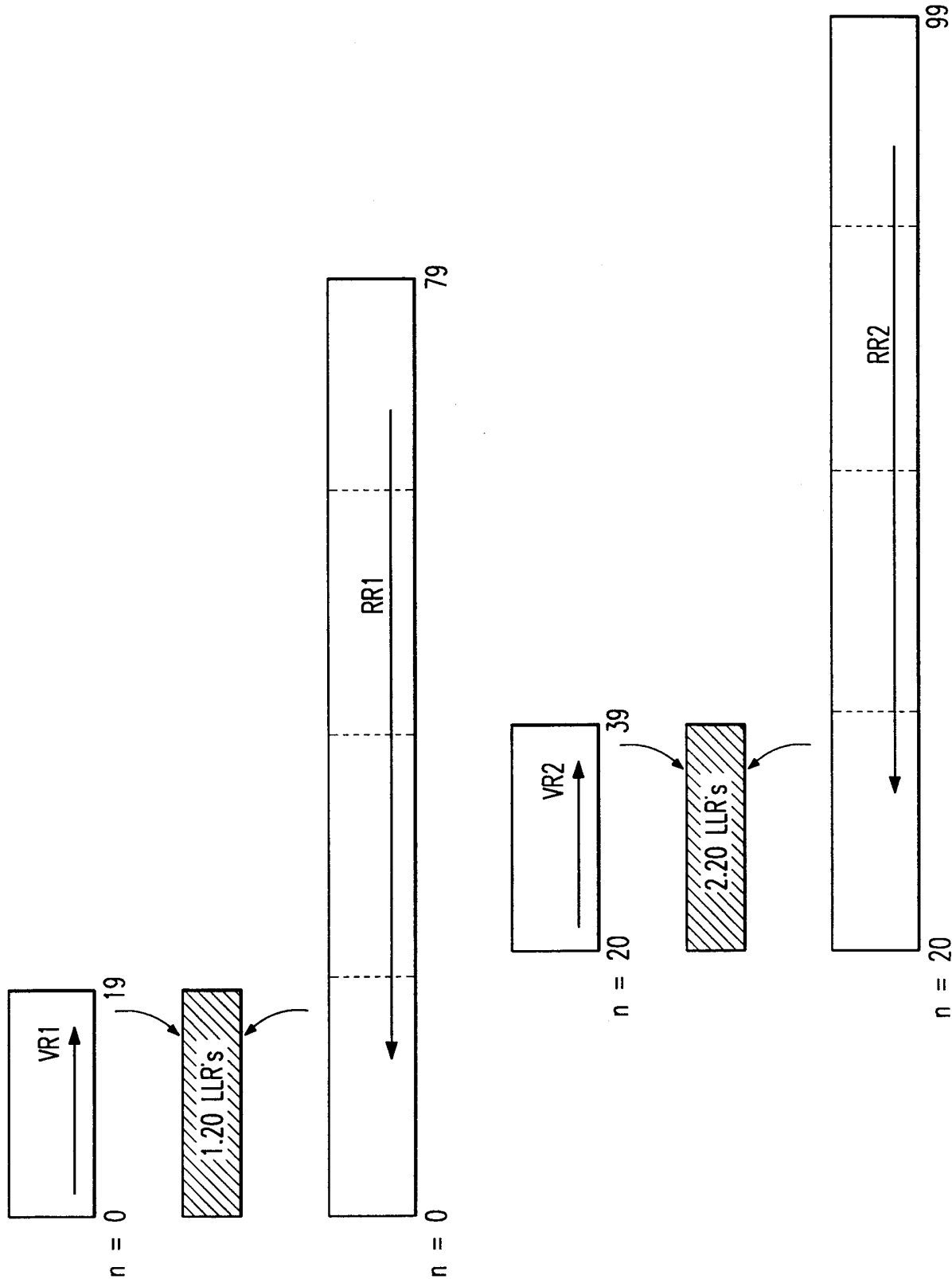


Fig. 7

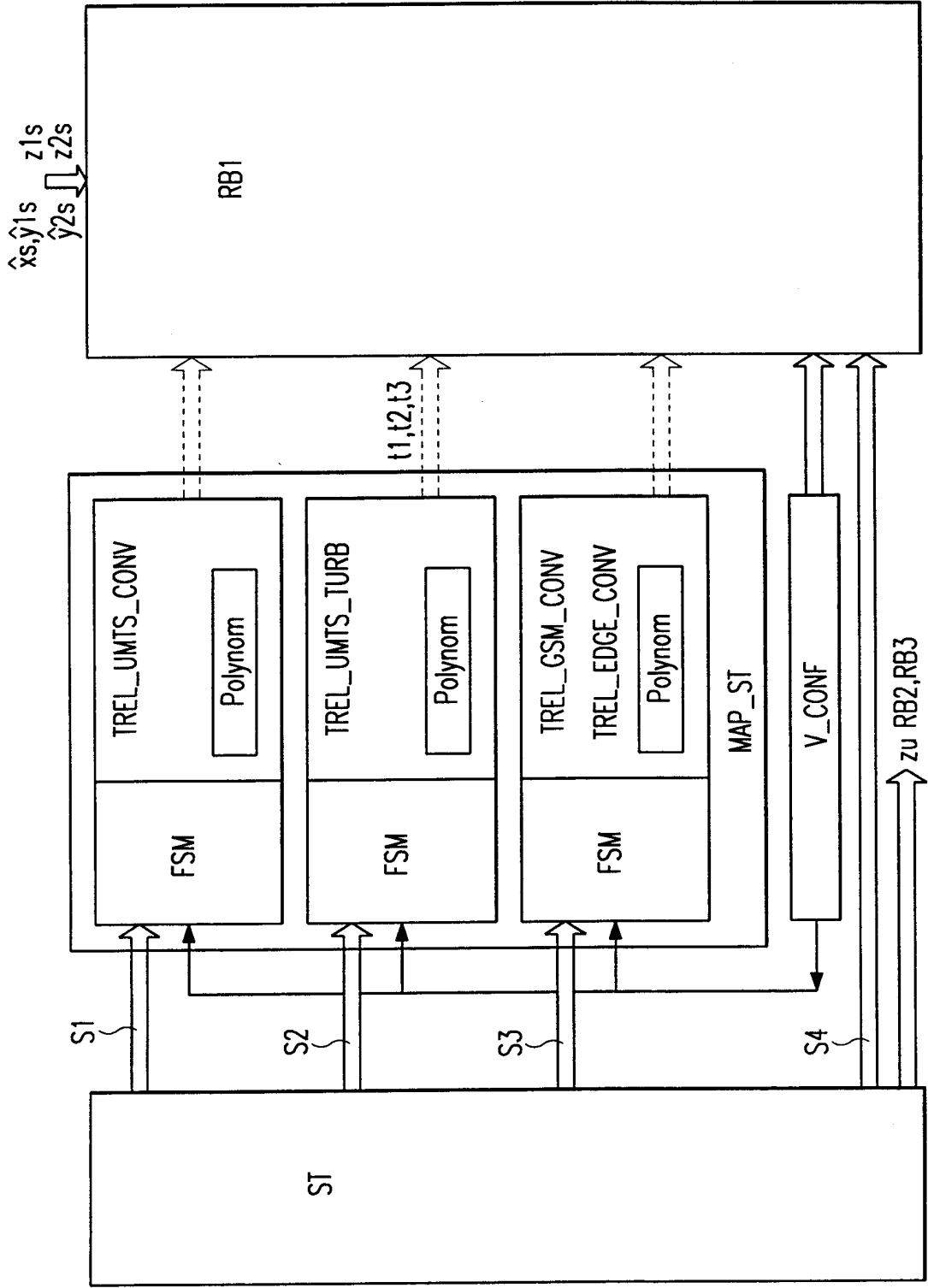


Fig. 8

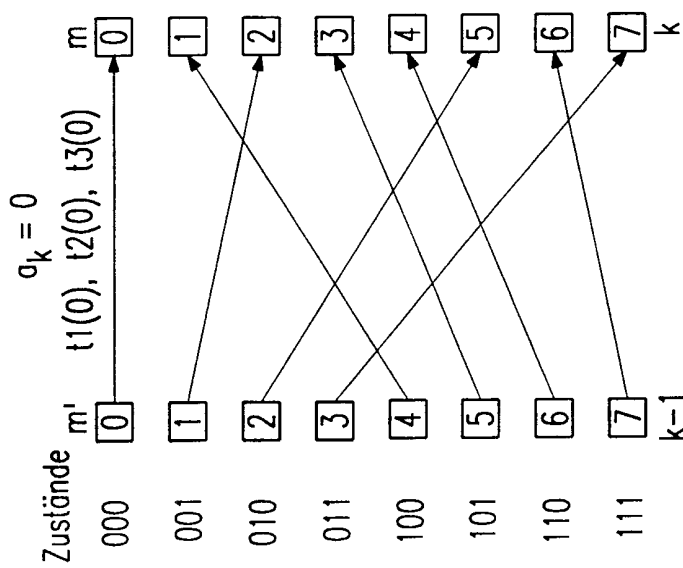


Fig. 9a

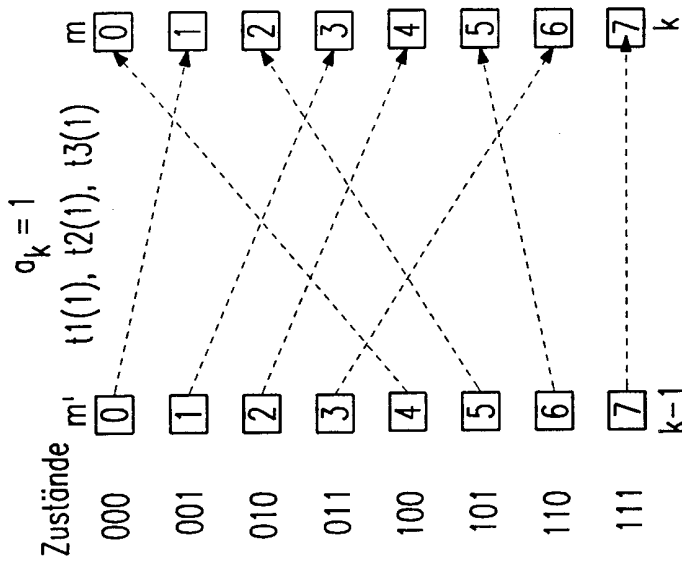


Fig. 9b

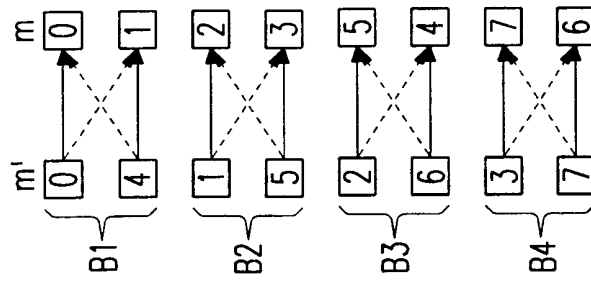


Fig. 9c

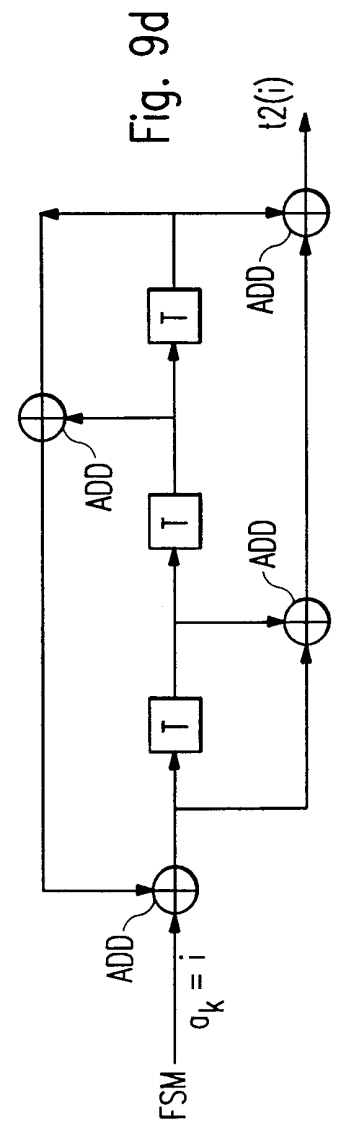


Fig. 9d

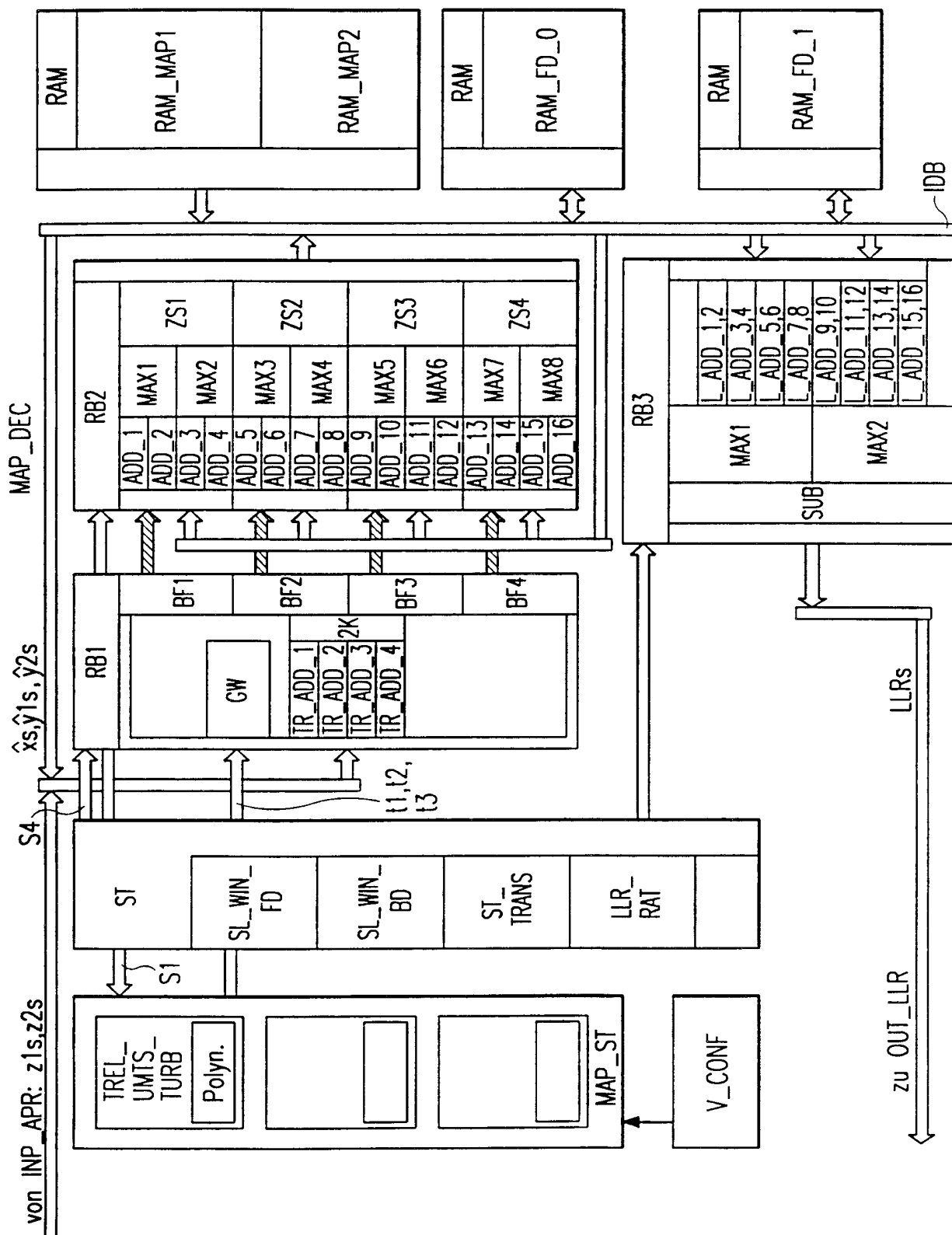


Fig. 10