(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4157041号
(P4157041)
(45)発行日　平成20年9月24日(2008.9.24)　(24)登録日　平成20年7月18日(2008.7.18)

(51)Int.Cl.　F I
H03M 13/19 (2006.01)　H03M 13/19

請求項の数 53　(全 49 頁)

(21)出願番号	特願2003-557101 (P2003-557101)
(86)(22)出願日	平成14年12月23日 (2002.12.23)
(65)公表番号	特表2005-514828 (P2005-514828A)
(43)公表日	平成17年5月19日 (2005.5.19)
(86)国際出願番号	PCT/US2002/041615
(87)国際公開番号	WO2003/056703
(87)国際公開日	平成15年7月10日 (2003.7.10)
審査請求日	平成17年10月3日 (2005.10.3)
(31)優先権主張番号	10/032,156
(32)優先日	平成13年12月21日 (2001.12.21)
(33)優先権主張国	米国(US)

(73)特許権者　501114844
デジタル　ファウンテン，　インコーポレイテッド
アメリカ合衆国　カリフォルニア　９４５３８，　フリーモント，スイート　３００　シビック　センター　ドライブ　３９１４１
(74)代理人　100076428
弁理士　大塚　康徳
(74)代理人　100112508
弁理士　高柳　司郎
(74)代理人　100115071
弁理士　大塚　康弘
(74)代理人　100116894
弁理士　木村　秀二

最終頁に続く

(54)【発明の名称】通信システムのための多段符号発生器及び復号器

(57)【特許請求の範囲】
【請求項１】
通信チャネルを介してソースから宛先へ送信されるデータを符号化する方法であって、該方法は、
入力シンボルの順序付けられた集合に送信されるべきデータをアレンジするステップと、
該入力シンボルから複数の冗長シンボルを作るステップと、
該入力シンボル及び該冗長シンボルを含むシンボルの組み合わせ集合から複数の出力シンボルを作るステップであって、入力シンボルの所与の集合に対する有効な出力シンボルの数は、該入力シンボルの数と独立であり、有効な出力シンボルであると決定される出力シンボルのうちのＮ個から該入力シンボルの順序付けられた集合を所望の程度の精度で再生できるように、少なくとも１つの出力シンボルは、該シンボルの組み合わせ集合中の２つ以上で且つ全部よりは少ない数のシンボルから作られる、ステップと
を包含する、方法。
【請求項２】
前記通信チャネルを介して前記複数の出力シンボルを送信するステップを更に含む、請求項１の方法。
【請求項３】
前記複数の出力シンボルを記憶媒体に蓄積するステップを更に含む、請求項１の方法。
【請求項４】

Ｎは前記入力シンボルの順序付けられた集合中の入力シンボルの数より大きい、請求項１の方法。

【請求項５】

Ｎは前記入力シンボルの順序付けられた集合中の入力シンボルの数より少ないか又はそれに等しい、請求項１の方法。

【請求項６】

前記入力シンボルの順序付けられた集合中の入力シンボルの数Ｋに基づいて作るべき冗長シンボルの数Ｒを決定するステップを更に含む、請求項１の方法。

【請求項７】

Ｋは入力シンボルの数の推定値である、請求項６の方法。

【請求項８】

前記複数の冗長シンボルはＬＤＰＣ符号に従って作られる、請求項１の方法。

【請求項９】

通信チャネルを介してソースから宛先へ送信されるデータを符号化する方法であって、該方法は、

入力シンボルの順序付けられた集合に送信されるべきデータをアレンジするステップと、

該入力シンボルから複数の冗長シンボルを作るステップと、

該入力シンボル及び該冗長シンボルを含むシンボルの組み合わせ集合から複数の出力シンボルを作るステップであって、可能な出力シンボルの数は該シンボルの組み合わせ集合中のシンボルの数と独立であり、任意のＮ個の出力シンボルから該入力シンボルの順序付けられた集合を所望の程度の精度で再生できるように、少なくとも１つの出力シンボルは、該シンボルの組み合わせ集合中の２つ以上で且つ全部よりは少ない数のシンボルから作られる、ステップと

を包含し、

複数の冗長シンボルは複数の第１冗長シンボルと複数の第２冗長シンボルとを含み、該複数の冗長シンボルを作るステップは、

該複数の第１冗長シンボルを該入力シンボルから作るステップと、

該複数の第２冗長シンボルを該第１冗長シンボル及び該入力シンボルから作るステップとを含む、方法。

【請求項１０】

前記複数の第１冗長シンボルはハミング符号に従って作られ、前記複数の第２冗長シンボルはＬＤＰＣ符号に従って作られる、請求項９の方法。

【請求項１１】

前記入力シンボルの順序付けられた集合中の入力シンボルの数Ｋに基づいて第１冗長シンボルの数Ｄ＋１を決定するステップと、

第２冗長シンボルの数Ｅを、作られる冗長シンボルの数Ｒ及びＤ＋１に基づいて決定するステップと

を更に含む、請求項１０の方法。

【請求項１２】

ＲをＫに基づいて決定するステップを更に含む、請求項１１の方法。

【請求項１３】

Ｋは入力シンボルの数の推定値である、請求項１１の方法。

【請求項１４】

Ｄは、２Ｄ－Ｄ－１≧Ｋとなる最小の整数であり、Ｅ＝Ｒ－Ｄ－１である、請求項１１の方法。

【請求項１５】

前記所望の精度は前記入力シンボルの完全な回復である、請求項１の方法。

【請求項１６】

前記所望の精度は高い確率での入力シンボルの完全な回復である、請求項１の方法。

【請求項１７】
前記所望の精度はＧ個の入力シンボルの回復であり、このＧは、前記入力シンボルの順序付けられた集合中の入力シンボルの数より少ない、請求項１の方法。
【請求項１８】
通信チャネルを介してソースから宛先へ送信されるデータを符号化する方法であって、該方法は、
入力シンボルの順序付けられた集合に送信されるべきデータをアレンジするステップと、
該入力シンボルから複数の冗長シンボルを作るステップと、
該入力シンボル及び該冗長シンボルを含むシンボルの組み合わせ集合から複数の出力シンボルを作るステップであって、可能な出力シンボルの数は該シンボルの組み合わせ集合中のシンボルの数と独立であり、任意のＮ個の出力シンボルから該入力シンボルの順序付けられた集合を所望の程度の精度で再生できるように、少なくとも１つの出力シンボルは、該シンボルの組み合わせ集合中の２つ以上で且つ全部よりは少ない数のシンボルから作られる、ステップと
を包含し、
任意の数の出力シンボルからせいぜいＧ個の入力シンボルを再生することができ、このＧは入力シンボルの順序付けられた集合中の入力シンボルの数より少ない、方法。
【請求項１９】
通信チャネルを介してソースから宛先へ送信されるデータを符号化する方法であって、該方法は、
入力シンボルの順序付けられた集合に送信されるべきデータをアレンジするステップと、
該入力シンボルから複数の冗長シンボルを作るステップであって、各冗長シンボルに対して、
ａ）ｔ個の別々の入力シンボルを分布に従って決定するステップと、
ｂ）各冗長シンボルをｔ個の別々の入力シンボルのＸＯＲとして計算するステップとを含む、ステップと、
該入力シンボル及び該冗長シンボルを含むシンボルの組み合わせ集合から複数の出力シンボルを作るステップであって、可能な出力シンボルの数は該シンボルの組み合わせ集合中のシンボルの数と独立であり、任意のＮ個の出力シンボルから該入力シンボルの順序付けられた集合を所望の程度の精度で再生できるように、少なくとも１つの出力シンボルは、該シンボルの組み合わせ集合中の２つ以上で且つ全部よりは少ない数のシンボルから作られる、ステップと
を包含する、方法。
【請求項２０】
ｔは全ての冗長シンボルについて同じである、請求項１９の方法。
【請求項２１】
ｔはＫ／２より大きな最小の奇数であり、Ｋは前記入力シンボルの順序付けられた集合中の入力シンボルの数である、請求項２０の方法。
【請求項２２】
前記分布は一様な分布である、請求項１９の方法。
【請求項２３】
通信チャネルを介してソースから宛先へ送信されるデータを符号化する方法であって、該方法は、
入力シンボルの順序付けられた集合に送信されるべきデータをアレンジするステップと、
該入力シンボルから複数の冗長シンボルを作るステップと、
該入力シンボル及び該冗長シンボルを含むシンボルの組み合わせ集合から複数の出力シンボルを作るステップであって、可能な出力シンボルの数は該シンボルの組み合わせ集合

中のシンボルの数と独立であり、任意のＮ個の出力シンボルから該入力シンボルの順序付けられた集合を所望の程度の精度で再生できるように、少なくとも１つの出力シンボルは、該シンボルの組み合わせ集合中の２つ以上で且つ全部よりは少ない数のシンボルから作られる、ステップと、

該複数の出力シンボルを該通信チャネルを介して送信するステップと

を包含し、該複数の出力シンボルを作るステップは、複数の出力シンボルを送信するステップと実質的に同時に実行される、方法。

【請求項２４】

前記複数の冗長シンボルを作るステップは、前記複数の出力シンボルを送信するステップと実質的に同時に実行される、請求項２３の方法。

【請求項２５】

前記複数の冗長シンボルを作るステップは、前記複数の出力シンボルを送信するステップより前に実行される、請求項２３の方法。

【請求項２６】

前記複数の出力シンボルを作るステップは第１装置を用いて実行され、前記複数の冗長シンボルを作るステップは該第１装置とは別の第２装置を用いて実行される、請求項１の方法。

【請求項２７】

通信チャネルを介してソースから宛先へ送信されるデータを符号化するシステムであって、該システムは、

複数の入力シンボルを受信するように結合されたスタティック符号器であって、該複数の入力シンボルは、送信されるべきデータから作られ、該スタティック符号器は、該入力シンボルに基づいて複数の冗長シンボルを作る冗長シンボル発生器を含む、スタティック符号器と、

該複数の入力シンボル及び該複数の冗長シンボルを受信するように結合されたダイナミック符号器であって、該複数の入力シンボルと該複数の冗長シンボルとを含むシンボルの組み合わせ集合から複数の出力シンボルを作る出力シンボル発生器を含み、入力シンボルの所与の集合に対する可能な有効な出力シンボルの数は、該入力シンボルの数と独立であり、有効な出力シンボルであると決定される出力シンボルのうちのＮ個から入力シンボルの順序付けられた集合を所望の精度で再生できるように、少なくとも１つの出力シンボルは、シンボルの該組み合わせ集合中の２以上で且つ全部よりは少ない数のシンボルから作られる、ダイナミック符号器と

を含む、システム。

【請求項２８】

Ｎは前記入力シンボルの順序付けられた集合中の入力シンボルの数より大きい、請求項２７のシステム。

【請求項２９】

Ｎは前記入力シンボルの順序付けられた集合中の入力シンボルの数より小さいか又はこれに等しい、請求項２７のシステム。

【請求項３０】

前記ダイナミック符号器及び通信チャネルに結合され、該通信チャネルを介して、前記出力シンボルを受信して該出力シンボルを送信する送信モジュールを更に含む、請求項２７のシステム。

【請求項３１】

通信チャネルを介してソースから宛先へ送信されるデータを符号化するシステムであって、該システムは、

複数の入力シンボルを受信するように結合されたスタティック符号器であって、該複数の入力シンボルは、送信されるべきデータから作られ、該スタティック符号器は、該入力シンボルに基づいて複数の冗長シンボルを作る冗長シンボル発生器を含む、スタティック符号器と、

該複数の入力シンボル及び該複数の冗長シンボルを受信するように結合されたダイナミック符号器であって、該複数の入力シンボルと該複数の冗長シンボルとを含むシンボルの組み合わせ集合から複数の出力シンボルを作る出力シンボル発生器を含み、可能な出力シンボルの数は該組み合わせ集合中のシンボルの数と独立であり、任意のＮ個の出力シンボルから入力シンボルの順序付けられた集合を所望の精度で再生できるように、少なくとも１つの出力シンボルは、該組み合わせ集合中の２以上で且つ全部よりは少ない数のシンボルから作られる、ダイナミック符号器と、

該ダイナミック符号器に結合されて、作られるべき各出力シンボルについてキーを作るキー発生器であって、該ダイナミック符号器は各キーを受け取るように結合される、キー発生器と

を含み、

該ダイナミック符号器は各出力シンボルを対応するキーに基づいて作る、システム。

【請求項３２】

前記スタティック符号器に結合されて、作られるべき冗長シンボルのうちの少なくとも幾つかの各々についてキーを作るキー発生器を更に含み、該スタティック符号器は各キーを受け取るように結合され、該スタティック符号器は該少なくとも幾つかの冗長シンボルの各々を対応するキーに基づいて作る、請求項２７のシステム。

【請求項３３】

前記スタティック符号器はＬＤＰＣ符号器を含む、請求項２７のシステム。

【請求項３４】

通信チャネルを介してソースから宛先へ送信されるデータを符号化するシステムであって、該システムは、

複数の入力シンボルを受信するように結合されたスタティック符号器であって、該複数の入力シンボルは、送信されるべきデータから作られ、該スタティック符号器は、該入力シンボルに基づいて複数の冗長シンボルを作る冗長シンボル発生器を含む、スタティック符号器と、

該複数の入力シンボル及び該複数の冗長シンボルを受信するように結合されたダイナミック符号器であって、該複数の入力シンボルと該複数の冗長シンボルとを含むシンボルの組み合わせ集合から複数の出力シンボルを作る出力シンボル発生器を含み、可能な出力シンボルの数は該組み合わせ集合中のシンボルの数と独立であり、任意のＮ個の出力シンボルから入力シンボルの順序付けられた集合を所望の精度で再生できるように、少なくとも１つの出力シンボルは、該組み合わせ集合中の２以上で且つ全部よりは少ない数のシンボルから作られる、ダイナミック符号器と

を含み、

該スタティック符号器は、第１冗長シンボル発生器を有する第１スタティック符号器と、第２冗長シンボル発生器を有する第２スタティック符号器とを更に含み、

該複数の冗長シンボルは第１複数の冗長シンボルと、第２複数の冗長シンボルとを含み、

該第１冗長シンボル発生器は該第１複数の冗長シンボルを該入力シンボルに基づいて作り、

該第２冗長シンボル発生器は該第２複数の冗長シンボルを該入力シンボル及び該第１複数の冗長シンボルに基づいて作る、システム。

【請求項３５】

前記第１スタティック符号器はハミング符号器を含み、前記第２スタティック符号器はＬＤＰＣ符号器を含む、請求項３４のシステム。

【請求項３６】

通信チャネルを介してソースから送信されるデータを受信する方法であって、該方法は、

出力シンボルを受信するステップであって、各出力シンボルは入力シンボル及び冗長シンボルの組み合わせ集合中の少なくとも１つのシンボルから作られ、少なくとも１つの出

力シンボルは、該組み合わせ集合中の２つ以上で且つ全部よりは少ないシンボルから作られ、可能な出力シンボルの数は該組み合わせ集合中のシンボルの数と独立であり、該入力シンボルは入力シンボルの順序付けられた集合からの入力シンボルであり、該冗長シンボルは該入力シンボルから作られる、ステップと、

該出力シンボルのうちの任意のＮ個を受信したときに、該組み合わせ集合中のシンボルの少なくとも部分集合を該Ｎ個の出力シンボルから再生するステップであって、該部分集合は複数の再生された入力シンボル及び複数の再生された冗長シンボルを含む、ステップと、

シンボルの少なくとも部分集合をＮ個の出力シンボルから再生するステップが入力シンボルを所望の精度で再生しなければ、再生されなかった入力シンボルの少なくとも幾つかを、複数の再生された冗長シンボル及び複数の再生された入力シンボルから再生するステップと

を包含する、方法。

【請求項３７】

該冗長シンボルは第１複数の冗長シンボルと第２複数の冗長シンボルとを含み、再生されなかった入力シンボルの少なくとも幾つかを再生するステップは、

該第１複数の冗長シンボルのうちの再生された冗長シンボルと該複数の再生された入力シンボルとから、再生されなかった入力シンボルのうちの少なくとも１つと該第２複数の冗長シンボルのうちの再生されなかった冗長シンボルとを再生するステップと、

もし該第１複数の冗長シンボルのうちの再生された冗長シンボルと該複数の再生された入力シンボルとから再生するステップが入力シンボルを所望の精度で再生しなければ、該第２複数の冗長シンボルのうちの冗長シンボルと該複数の復号された入力シンボルとから少なくとも１つの再生されなかった入力シンボルを再生するステップとを含む、請求項３６の方法。

【請求項３８】

再生されなかった入力シンボル及び該第２複数の冗長シンボルのうちの再生されなかった冗長シンボルのうちの或るシンボルはＬＤＰＣ復号器を用いて再生され、

該或る入力シンボルは該第２複数の冗長シンボルのうちの冗長シンボルからハミング復号器を用いて再生される、請求項３７の方法。

【請求項３９】

再生されなかった入力シンボルのうちの少なくとも或るシンボルを再生するステップは、再生されなかった入力シンボルの全てを再生することを含む、請求項３６の方法。

【請求項４０】

該組み合わせ集合中のシンボルの少なくとも部分集合を再生するステップと再生されなかった入力シンボルのうちの少なくとも或るシンボルを再生するステップとは、

受信された各出力シンボルについて該出力シンボルと関連する該組み合わせ集合中のシンボルを示す第１マトリックスを形成し、

各冗長シンボルについて該冗長シンボルと関連する入力シンボルを示す情報で該第１マトリックスを拡大し、

入力シンボルのうちの少なくとも或るシンボルを該拡大された第１マトリックスにより示される方程式系に対する解として再生することを含む、請求項３６の方法。

【請求項４１】

Ｎは入力シンボルの数より大きいか又はこれに等しい、請求項３６の方法。

【請求項４２】

Ｎは入力シンボルの数より小さい、請求項３６の方法。

【請求項４３】

再生されなかった入力シンボルのうちの少なくとも或る入力シンボルを再生することは、全ての入力シンボルを再生することを含む、請求項３６の方法。

【請求項４４】

再生されなかった入力シンボルのうちの少なくとも或る入力シンボルを再生することは

、入力シンボルのうちの全部よりは少数の入力シンボルを再生することを含む、請求項３６の方法。

【請求項４５】

ソースから通信チャネルを介して送信されたデータを受信するシステムであって、該システムは、

通信チャネルを介して送信された出力シンボルを受信するために該通信チャネルに結合されている受信モジュールであって、各出力シンボルは入力シンボル及び冗長シンボルの組み合わせ集合中の少なくとも１つのシンボルから作られ、少なくとも１つの出力シンボルは該組み合わせ集合中の２つ以上で且つ全部よりは少ないシンボルから作られ、可能な出力シンボルの数は該組み合わせ集合中のシンボルの数と独立であり、該入力シンボルは入力シンボルの順序付けられた集合からの入力シンボルであり、該冗長シンボルは該入力シンボルから作られる、受信モジュールと、

Ｎ個の出力シンボルを受信すると、該Ｎ個の出力シンボルから該組み合わせ集合中のシンボルの部分集合を復号するダイナミック復号器であって、該部分集合は、複数の復号された入力シンボルと複数の復号された冗長シンボルとを含む、ダイナミック復号器と、

復号されなかった入力シンボルがもしあるならばその入力シンボルのうちの少なくとも或る入力シンボルを複数の復号された冗長シンボルから復号するスタティック復号器も含む、システム。

【請求項４６】

該スタティック復号器はＬＤＰＣ復号器を含む、請求項４５のシステム。

【請求項４７】

該冗長シンボルは第１複数の冗長シンボルと第２複数の冗長シンボルとを含んでおり、該スタティック符号器は、

該第１複数の冗長シンボルのうちの復号された冗長シンボルと複数の復号された入力シンボルとから、復号されなかった入力シンボルと該第２複数の冗長シンボルのうちの復号されなかった冗長シンボルとのうちの少なくとも１つを復号する第１スタティック復号器と、

該第２複数の冗長シンボルのうちの冗長シンボルと該複数の復号された入力シンボルとから少なくとも１つの復号されなかった入力シンボルを復号する第２スタティック復号器とを含む、請求項４５のシステム。

【請求項４８】

該第１スタティック復号器はＬＤＰＣ復号器を含み、該第２スタティック復号器はハミング復号器を含む、請求項４７のシステム。

【請求項４９】

該ダイナミック復号器は、

各々の受信された出力シンボルについて、該出力シンボルと関連する該組み合わせ集合中のシンボルを示す第１マトリックスを形成し、

各冗長シンボルについて該冗長シンボルと関連する入力シンボルを示す情報で該第１マトリックスを拡大し、

該拡大された第１マトリックスにより示される方程式系に対する解として該入力シンボルのうちの少なくとも或る入力シンボルを再生するステップを実行するように構成されたプロセッサを含む、請求項４５のシステム。

【請求項５０】

搬送波で具体化されるコンピュータ・データ信号であって、該コンピュータ・データ信号は、

複数の出力シンボルであって、該複数の出力シンボルは入力シンボルの順序付けられた集合と冗長シンボルとを含むシンボルの組み合わせ集合から作られたシンボルを表わし、該冗長シンボルは該入力シンボルから作られ、入力シンボルの所与の集合に対する可能な有効な出力シンボルの数は、該入力シンボルの数と独立であり、該データ信号の受信装置が有効な出力シンボルであると決定されるべき出力シンボルのうちのＮ個から入力シンボ

ルの順序付けられた集合を所望の精度で再生できるように、少なくとも１つの出力シンボルは、該シンボルの組み合わせ集合中の２つ以上で且つ全部よりは少ない数のシンボルから作られる、複数のシンボルを含む、コンピュータ・データ信号。

【請求項５１】

該シンボルの少なくとも部分集合を再生するステップは、該出力シンボルのうちの所定数Ｎ個の任意の出力シンボルが受信されたときに実行される、請求項３６の方法。

【請求項５２】

該シンボルの少なくとも部分集合を再生するステップは、該出力シンボルのいずれかＮ個が受信されたときに実行され、ここでＮは入力シンボルが所望の精度で再生され得るような数である、請求項３６の方法。

【請求項５３】

該シンボルの少なくとも部分集合を再生するステップは、該出力シンボルの受信と実質的に同時に実行される、請求項３６の方法。

【発明の詳細な説明】

【背景技術】

【０００１】

（発明の背景）

本発明は、通信システムにおけるデータの符号化及び復号に関し、特に、伝達されるデータ中のエラー及びギャップを補償するためにデータを符号化及び復号する通信システムに関する。

【０００２】

送信者と受信者との間でのファイルの伝送は多くの文献の主題となっている。好ましくは、受信者は、送信者によりチャネル経由で送信されたデータの正確なコピーを或るレベルの確実性で受け取ることを望む。チャネルが完全な忠実度を持っていない場合（殆ど全ての物理的に実現可能なシステムにあてはまることである）、伝送中に失われたり改ざんされたりしたデータをどのように扱うかということが１つの関心事である。原形が損なわれたデータが誤って受信されたデータであると常に受信者が分かるとは限らないので、失われたデータ（削除）の方が原形が損なわれたデータ（エラー）より処理しやすいということが良くある。削除及び／又はエラーを訂正するために多くのエラー訂正符号が開発されている。通常、使用される符号は、データが伝送されるチャネルの非忠実度及び伝送されるデータの性質に関する情報に基づいて選択される。例えば、チャネルが長い非忠実度期間を有すると分かっている場合、その様なアプリケーションにはバーストエラー符号がおそらく最も良く適しているであろう。短いエラーが希に生じると予想される場合には、単純なパリティー符号がおそらく最善であろう。

【０００３】

符号を選択する上でのもう一つの考慮事項は、送信に用いられるプロトコルである。“インターネット”（大文字“Ｉ”を伴う）として知られているネットワークの地球規模インターネットワークの場合には、データトランスポートのためにパケット・プロトコルが使用される。このプロトコルは、インターネットプロトコル又は略されて“ＩＰ”と称される。ＩＰネットワークを介してデータのファイル又はその他のブロックが送信されるとき、これは等サイズの入力シンボルに分割され、入力シンボルは連続するパケット内に置かれる。入力シンボルが実際にビットストリームに分解されていてもいなくても入力シンボルの“サイズ”はビット数で測ることができ、入力シンボルが2^M個のシンボルのアルファベットから選択されるときには入力シンボルはＭビットのサイズを有する。この様なパケット・ベースの通信システムでは、パケット指向の符号化方式が適切であろう。ネットワークで削除があったとしても意図された受信者が元のファイルの正確なコピーを復元することを可能にする送信は信頼できると称される。インターネットでは、時折発生する輻輳に起因してルータ内のバッファーリングメカニズムがその容量に達し、そのために、該メカニズムが入って来るパケットを落とさざるを得なくなるのでパケットが失われることが良くある。トランスポート中の削除からの保護は多くの研究の主題である。

【０００４】
トランスポート制御プロトコル（“ＴＣＰ”）は、受け取り通知メカニズムを有する広く使用されているポイントツーポイント・パケット制御方式である。ＴＣＰは１対１通信では良好なものであり、この場合送信者及び受信者は共に送信が何時行われて受信されるか合意し、またどの送信装置及び受信装置が使われるか合意する。しかし、ＴＣＰは、１対多数又は多数対多数の通信や、送信者及び受信者がデータをいつ何処で送信或いは受信するかを自主的に決める場合には適さないことが良くある。
【０００５】
ＴＣＰを使って、送信者は順序付けられたパケットを送信し、受信者は各パケットの受信を受け取り通知する。もしパケットが紛失すると、受け取り通知は送信者に送られなくて、送信者はそのパケットを送り直す。パケットが紛失する原因は幾つかある。ＴＣＰ／ＩＰのようなプロトコルでは、受け取り通知パラダイムは完全な故障を伴わずにパケットが紛失することを許すが、その理由は、受け取り通知が無いことに応答して又は受信者からの明示的要求に応答して紛失したパケットを単に送り直すことができることにある。いずれにしても、受け取り通知プロトコルは、紛失したパケットの数が大きいときに激しく使用される受信者から送信者へのバックチャネルを必要とする。
【０００６】
受け取り通知ベースのプロトコルは一般に多くのアプリケーションに適していて実際に現行のインターネット上で広く使われているけれども、該プロトコルは効率が良くなくて、米国特許第６，３０７，４８７号に記載されているような一定のアプリケーションについては時には完全に不可能である。この米国特許第６，３０７，４８７号は、マイケルＧ．ルビーに発行され、“通信システムのための情報付加符号発生器及び復号器（Ｉｎｆｏｒｍａｔｉｏｎ Ａｄｄｉｔｉｖｅ Ｃｏｄｅ Ｇｅｎｅｒａｔｏｒ ａｎｄ Ｄｅｃｏｄｅｒ ｆｏｒ Ｃｏｍｍｕｎｉｃａｔｉｏｎ Ｓｙｓｔｅｍｓ）”と題されている（以降、“ルビーＩ”）。更に、受け取り通知に基づくプロトコルは、１送信者が多数のユーザにファイルを同時に送る放送とは釣り合いが取れない。例えば、送信者が衛星チャネルを介してファイルを多数の受信者に放送しているとする。各受信者はいろいろなパターンのパケット紛失を経験するであろう。ファイルを確実に配達するために受け取り通知データ（肯定又は否定）に依拠するプロトコルは、各受信者から送信者へのバックチャネルを必要とし、これを設けることが法外に高価となる可能性がある。更に、これは、複雑で強力な送信者が受信者から送られた受け取り通知データの全てを適切に処理できることを必要とする。異なる受信者が異なるパケットの集合を紛失したならば、ほんの数人の受信者が受け取りそこなっただけのパケットが再放送されて他の受信者が無用の重複パケットを受信する結果となることがもう一つの欠点である。
【０００７】
受け取り通知ベースのプロトコルに代わる実際に時折使用されるプロトコルはカルーセルに基づくプロトコルである。カルーセル・プロトコルは、入力ファイルを等長の入力シンボルに分割し、各入力シンボルをパケット内に置き、連続的にサイクルして全てのパケットを送信する。カルーセルに基づくプロトコルの主な欠点は、受信者がたとえ１つのパケットであっても受け取りそこなった場合にその受信者が受け取りそこなったパケットを受け取るチャンスが来るまでにもう１回サイクル全体を待たなければならなくなることである。このことは、見方を変えれば、カルーセルに基づくプロトコルが大量の無用な重複データ受信の原因となり得ることを意味する。例えば、受信者がカルーセルの始めからパケットを受け取り、しばらくの間受信をやめ、その後に再び該カルーセルの始めから受信し始めれば、無用の重複パケットが多数受信されることになる。
【０００８】
上記の問題を解決するために提案されている１つの解決策は、受け取り通知ベースのプロトコルの使用を避け、代わりに、信頼性を高めるために、リードソロモン符号又はトルネード符号、或いは情報付加符号であるチェーンリアクション符号などの前進型誤信号訂正（ＦＥＣ）符号を使用することである。これらの符号では、内容から出力シンボルが作

られて、内容を構成する入力シンボルを単に送る代わりにその出力シンボルが送られる。リードソロモン符号或いはトルネード符号等の削除訂正符号は、固定された長さの内容について固定された数の出力シンボルを作る。例えば、Ｋ個の入力シンボルについてＮ個の出力シンボルを作ることができる。このＮ個の出力シンボルは、Ｋ個のオリジナル入力シンボルとＮ－Ｋ個の冗長シンボルから成ることができる。もし記憶装置が許すならば、サーバーは、各内容についての出力シンボルの集合を１回だけ計算し、カルーセル・プロトコルを用いて出力シンボルを送信する。

【０００９】

或るＦＥＣ符号に伴う問題は、該符号が過大な計算能力又はメモリを作動させることを必要とすることである。もう１つの問題は、符号化プロセスの前に出力シンボルの数を決定しなければならないことである。このことは、もしパケットの紛失率を過大に見積もれば非能率という結果に至る可能性を有し、またもしパケットの紛失率を過小に見積もれば故障という結果に至る可能性を有する。

【００１０】

普通のＦＥＣ符号では、作ることのできる、考えられる出力シンボルの数は、内容を分割して得られる入力シンボルの数と同じ桁の数である。通常、これら出力シンボルの殆ど又は全部が送信ステップの前の前処理ステップで作られる。これら出力シンボルは、長さが元の内容と同じか又は元の内容より僅かに長い出力シンボルの任意の部分集合から入力シンボルの全てを再生することができるという特性を持っている。

【００１１】

ルビーＩ（以降、“チェーンリアクション符号”と称する）に記載されている実施態様は、上記の問題点を扱う異なる形の前進型誤信号訂正を提供する。チェーンリアクション符号では、作ることのできる、考えられる出力シンボルのプールは通常は入力シンボルの数より桁違いに多く、その可能性のプールからランダム出力シンボルを非常に急速に作ることができる。チェーンリアクション符号では、送信ステップと同時に出力シンボルを必要に応じて急いで作ることができる。チェーンリアクション符号は、元の内容より僅かに長いランダムに作られた出力シンボルの集合の任意の部分集合から内容の全ての入力シンボルを再生することができるという特性を有する。

【００１２】

チェーンリアクション符号の１つの具体的形では、各出力シンボルは入力シンボルのうちの幾つかの排他的論理和（ＸＯＲ、

【００１３】

【数１】

$$\oplus$$

により表示する）として得られる。Ｋが入力シンボルの総数を表わすとすると、各入力シンボルは、平均で、ｃ＊ｌｎ（Ｋ）個の入力シンボルのＸＯＲであり、ここでｌｎ（Ｋ）はＫの自然対数であり、ｃは適切な定数である。例えば、Ｋが６０，０００であるとき、各出力シンボルは平均で２８．６８個の入力シンボルのＸＯＲであり、Ｋが１０，０００であるとき、各出力シンボルは平均で２２．８６個の入力シンボルのＸＯＲである。ＸＯＲの数が大きいと、各動作がメモリからデータを取り出し、ＸＯＲ演算を実行し、記憶場所を更新することを必要とするので、出力シンボルの計算時間が長くなる。

【００１４】

チェーンリアクション符号器により作られた出力シンボルの１つの特性は、充分な出力シンボルが受信されたならば受信者が直ちに元のファイルを回復できることである。具体的には、元のＫ個の入力シンボルを高い確率で回復するためには、受信装置はほぼＫ＋Ａ個の出力シンボルを必要とする。比Ａ／Ｋは“相対的受信オーバーヘッド”と称される。相対的受信オーバーヘッドは、入力シンボルの数Ｋと、復号器の信頼性とに依存する。例えば、１つの具体的実施態様において、Ｋが６０，０００である場合、５％の相対的受信オーバーヘッドは復号器が少なくとも$1-10^{-8}$の確率で入力ファイルを首尾良く復号することを保証し、Ｋが１０，０００に等しい場合、１５％の相対的受信オーバーヘッド

は復号器の同じ成功確率を保証する。一実施態様では、チェーンリアクション符号の相対的受信オーバーヘッドを（１３＊ｓｑｒｔ（Ｋ）＋２００）／（Ｋ）として計算することができ、ここでｓｑｒｔ（Ｋ）は入力シンボルの数Ｋの平方根である。この実施態様では、チェーンリアクション符号の相対的受信オーバーヘッドは、Ｋの小さい値について大きくなりがちである。

【００１５】

出力シンボルがＸＯＲ機能を用いて符号化される実施態様では、チェーンリアクション復号器の主な計算動作は記憶場所のＸＯＲを実行することである。このようなＸＯＲの数は、チェーンリアクション符号器の場合と同様に釣り合いが取れている。

【００１６】

チェーンリアクション符号は、パケットベースのネットワークを介する通信のために極めて有益である。しかし、該符号はかなり計算集中的である可能性がある。例えば、チェーンリアクション符号の或る具体的実施態様では、入力シンボルの個数Ｋが６０，０００であるとき、各出力シンボルの計算は、平均で２８．６８個のランダムに選択された入力シンボルを取り出して、それらに対してＸＯＲ演算を実行することを必要とする。サーバーが同時にサービスすることのできるファイルの数はどの出力シンボルにも必要とされる演算の数に逆比例するので、全ての出力シンボルに必要とされる演算の数を減らすのが有益であろう。後者を例えば３分の１に減らせば、１つのサーバーから同時にサービスを受けられるファイルの数は３倍に増える。

【００１７】

チェーンリアクション符号のもう１つの特性は、与えられたターゲット成功確率に対して割合に大きくなる可能性のある受信オーバーヘッドを必要とすることである。例えば、前述したように、チェーンリアクション符号の或る具体的実施態様では、もしＫが１０，０００ならば、１５％の相対的受信オーバーヘッドは少なくとも$1-10^{-8}$の復号成功確率を保証する。受信オーバーヘッドは、Ｋが小さいほど大きくなりがちである。例えば、チェーンリアクション符号の或る具体的実施態様では、Ｋが１０００ならば、６１％の相対的受信オーバーヘッドは同じ確率での復号の成功を保証する。更に、衛星ネットワークを介する内容の高速送信などの一定のアプリケーションに必要とされるように目標エラー確率を10^{-12}程度の数まで減少させれば、更に大きな受信オーバーヘッドが必要となる。

【発明の開示】

【課題を解決するための手段】

【００１８】

（発明の簡単な要約）

本発明の一実施態様により、通信チャネルを介してソースから宛先へ送信されるデータを符号化する方法が提供される。この方法は、入力シンボルの順序付けられた集合に働きかけるものであって、入力シンボルから複数の冗長シンボルを作るステップを含む。この方法は、入力シンボル及び冗長シンボルを含むシンボルの組み合わせ集合から複数の出力シンボルを作るステップも含んでおり、可能な出力シンボルの数は該シンボルの組み合わせ集合中のシンボルの数より遥かに大きく、そして任意の所定数の出力シンボルから入力シンボルの順序付けられた集合を所望の程度の精度で再生できるように、少なくとも１つの出力シンボルは、該シンボルの組み合わせ集合中の２つ以上で且つ全部よりは少ない数のシンボルから作られる。

【００１９】

本発明の他の実施態様によると、通信チャネルを介してソースから宛先へ送信されるデータを符号化するシステムが提供される。このシステムは、複数の入力シンボルを受信するように結合されたスタティック符号器を含み、その複数の入力シンボルは、送信されるべきデータから作られる。このスタティック符号器は、入力シンボルに基づいて複数の冗長シンボルを作る冗長シンボル発生器を含む。このシステムは、更に、複数の入力シンボル及び複数の冗長シンボルを受信するように結合されたダイナミック符号器を含む。該ダ

イナミック符号器は、複数の入力シンボルと複数の冗長シンボルとを含むシンボルの組み合わせ集合から複数の出力シンボルを作る出力シンボル発生器を含み、可能な出力シンボルの数は該組み合わせ集合中のシンボルの数より遥かに大きく、そして任意の所定数の出力シンボルから入力シンボルの順序付けられた集合を所望の程度の精度で再生できるように、少なくとも１つの出力シンボルは、該組み合わせ集合中の以上のシンボルから、及び該シンボルの組み合わせ集合中のシンボルの全部よりは少ない数のシンボルから作られる。

【００２０】

本発明のもう一つの実施態様によると、通信チャネルを介してソースから送信されるデータを受信する方法が提供される。この方法は、出力シンボルを受信するステップを含み、各出力シンボルは入力シンボル及び冗長シンボルの組み合わせ集合中の少なくとも１つのシンボルから作られ、少なくとも１つの出力シンボルは、該組み合わせ集合中の２つ以上で且つ全部よりは少ないシンボルから作られ、可能な出力シンボルの数は該組み合わせ集合中のシンボルの数より遥かに大きく、該入力シンボルは入力シンボルの順序付けられた集合からの入力シンボルであり、該冗長シンボルは該入力シンボルから作られる。この方法は、任意の所定数Ｎの出力シンボルを受信した後に該組み合わせ集合中のシンボルの少なくとも部分集合をＮ個の出力シンボルから再生するステップも含み、その部分集合は複数の再生された入力シンボル及び複数の再生された冗長シンボルを含む。この方法は、更に、シンボルの少なくとも部分集合をＮ個の出力シンボルから再生するステップが入力シンボルを所望の精度で再生しなければ、再生されなかった入力シンボルの少なくとも幾つかを、複数の再生された冗長シンボル及び／又は複数の再生された入力シンボルのうちの幾つかから再生するステップを含む。

【００２１】

本発明の更にもう一つの実施態様により、ソースから通信チャネルを介して送信されたデータを受信するシステムが提供される。該システムは、該通信チャネルを介して送信された出力シンボルを受信するために通信チャネルに結合されている受信モジュールを含み、各出力シンボルは入力シンボル及び冗長シンボルの組み合わせ集合中の少なくとも１つのシンボルから作られ、少なくとも１つの出力シンボルは該組み合わせ集合中の２つ以上で且つ全部よりは少ないシンボルから作られ、可能な出力シンボルの数は該組み合わせ集合中のシンボルの数より遥かに大きく、該入力シンボルは入力シンボルの順序付けられた集合からの入力シンボルであり、該冗長シンボルは該入力シンボルから作られる。該システムは、更に、所定数Ｎの出力シンボルを受信すると該Ｎ個の出力シンボルから該組み合わせ集合中のシンボルの部分集合を復号するダイナミック復号器も含み、該部分集合は複数の復号された入力シンボルと複数の復号された冗長シンボルとを含む。該システムは、更に、復号されなかった入力シンボルがもしあるならばその入力シンボルのうちの少なくとも幾つかを複数の復号された冗長シンボルから復号するスタティック復号器も含む。

【００２２】

本発明の更にもう一つの実施態様により、搬送波で具体化されるコンピュータ・データ信号が提供される。該コンピュータ・データ信号は複数の出力シンボルを含み、該複数の出力シンボルは入力シンボルの順序付けられた集合と冗長シンボルとを含むシンボルの組み合わせ集合から作られたシンボルを表わし、該冗長シンボルは該入力シンボルから作られ、可能な出力シンボルの数は該シンボルの組み合わせ集合中のシンボルの数より遥かに大きく、そして該データ信号の受信装置が任意の所定数の出力シンボルから入力シンボルの順序付けられた集合を所望の精度で再生できるように、少なくとも１つの出力シンボルは、該シンボルの組み合わせ集合中の２つ以上で且つ該シンボルの組み合わせ集合中のシンボルの全部よりは少ない数のシンボルから作られる。

【００２３】

本発明により多数の利益が達成される。例えば、或る特定の実施態様では、チャネルを介して伝送するためにデータを符号化する計算に係わる費用が低減される。もう一つの特定の実施態様では、その様なデータを復号する計算に係わる費用が低減される。実施態様

に依存して、これらの利益のうちの１つ以上が達成可能である。これらの、及びその他の利益は、本明細書全体を通じて、そして以下ではいっそう具体的に、詳しく提供される。

【００２４】

本明細書の下記の部分及び添付図面から、本書で開示される発明の性質及び利点が更に理解されよう。

【発明を実施するための最良の形態】

【００２５】

（特定の実施態様の詳細な説明）

本発明は、米国特許第６，３０７，４８７号（ルビーＩ）と、米国特許第６，３２０，５２０号とを参照するが、この後者の米国特許第６，３２０，５２０号は、マイケルG.ルビーに発行され、“通信システムのための情報付加群符号発生器及び復号器（Ｉｎｆｏｒｍａｔｉｏｎ Ａｄｄｉｔｉｖｅ Ｇｒｏｕｐ Ｃｏｄｅ Ｇｅｎｅｒａｔｏｒ ａｎｄ Ｄｅｃｏｄｅｒ ｆｏｒ Ｃｏｍｍｕｎｉｃａｔｉｏｎ Ｓｙｓｔｅｍｓ）”と題されており（以降、“ルビーＩＩ”）、あらゆる目的のためにその開示全体を参照により本書に取り入れる。ルビーＩ及びルビーＩＩは、本発明の一定の実施態様に使用できるシステム及び方法について教示事項を提供している。しかし、これらのシステム及び方法は本発明にとって必要なものではなくて、他の多くの変化形、改造形及び代替物を使用することもできることが理解されよう。

【００２６】

本書に記載される特定の実施態様では、本明細書で使用されるいろいろな用語の意味及び範囲が説明され、その後に“多段符号化”と表示される符号化方式が説明される。

【００２７】

本書で説明される多段符号化は、データを複数の段で符号化する。通常、常にというわけではないけれど、第１段は所定量の冗長性をデータに付加する。第２段は、チェーンリアクション符号などを使用して、元のデータと符号化の第１段で計算された冗長シンボルとから出力シンボルを作る。本発明の１つの特定の実施態様では、受信されたデータは始めにチェーンリアクション復号プロセスを用いて復号される。もしこのプロセスが元のデータを完全に復号するのに成功しなかったならば、第２復号ステップを適用することができる。

【００２８】

本書に記載されている幾つかの実施態様の１つの利点は、後述するように、出力シンボルを作るために必要な算術演算が、チェーンリアクション符号化だけの場合より少なくなることである。第１段符号化と第２段符号化とを含む幾つかの特定の実施態様のもう一つの利点は、第１段符号化と第２段符号化とを別々の時に且つ／又は別々の装置により実行でき、従って計算負荷を分割できることである。このことは、例えば、符号化の一部分を送信と実質的に同時に実行することが希望される場合に、利点となり得る。特に、第１段符号化は送信より充分前に実行可能であるのに対して第２段符号化は送信と実質的に同時に実行可能である。しかし、或る実施態様では、第１段符号化及び第２段符号化の両方を送信と実質的に同時に、且つ／又は１つの装置で、実行可能であることが理解されよう。この開示を検討した後、当業者にとっては他の多くの変化形、改造形及び代替物が明らかとなろう。

【００２９】

多段符号化の実施態様では、第１段符号化中に入力ファイルから冗長シンボルが作られる。これらの実施態様では、第２段符号化時に、入力ファイルと冗長シンボルとの組み合わせから出力シンボルが作られる。これらの実施態様のうちの幾つかでは、出力シンボルを必要に応じて作ることができる。第２段がチェーンリアクション符号化を含む実施態様では、他の出力シンボルがどのように作られるかということを顧慮せずに各出力シンボルを作ることができる。これらの出力シンボルは、作られた後、パケット中に置かれて自分達の宛先へ送信され得るが、各パケットは１つ以上の出力シンボルを含む。その代わりとして、或いはそれと共に、非パケット化送信手法を使うこともできる。

【００３０】
本書で使われるとき、“ファイル”という用語は、１つ以上のソースに蓄積されていて１つのユニットとして１つ以上の宛先へ配送されることになる任意のデータを指す。ファイルサーバー又はコンピュータ記憶装置からの文書、画像、及びファイルは全て配達可能な“ファイル”の例である。ファイルは、既知のサイズ（ハードディスクに蓄積されている１メガバイトの画像など）のファイルであっても良く、或いはサイズ不明のファイルであっても良い（ストリーミング・ソースの出力から取り出されたファイルなど）。いずれにしても、ファイルは入力シンボルのシーケンスであり、その各入力シンボルは該ファイル内の位置及び値を持っている。

【００３１】
送信は、１つ以上の送信者からファイルを配達するためにチャネルを通して１つ以上の受信者へデータを送るプロセスである。送信者は、時には符号器と称されることもある。もし１つの送信者が完全なチャネルによって任意の数の受信者に接続されたならば、全てのデータが正しく受信されるであろうから、受信されたデータは入力ファイルの正確なコピーであり得る。ここで、チャネルが不完全であることを仮定するが、これは殆どの実世界チャネルの実状である。多くのチャネル欠陥のうちの、興味ある２つの欠陥は、データ削除とデータ不完全性（これはデータ削除の特別の場合であるとして取り扱われて良い）とである。データ削除は、チャネルがデータを紛失し或いはデータを落としたときに発生する。データの一部が既に通り過ぎてしまうまで受信者がデータの受信を開始しないとき、送信が終わる前に受信者が受信をやめたとき、受信者が送信されたデータの一部分だけを受信することを選んだとき、且つ／又は受信者がデータ受信を断続的に止めたり再開したりしたときに、データ不完全性が発生する。データ不完全性の１例として、移動する衛星送信者は、入力ファイルを表わすデータを送信しようとして、受信者が有効範囲内に入る前に送信を開始することがあろう。受信者が有効範囲内に入ると、衛星が有効範囲外に出るまでデータを受信することができ、この時点で受信者は衛星用アンテナを向け直して（その間、これはデータを受信していない）、有効範囲内に入った他の衛星から送信されつつある同じ入力ファイルに関するデータを受信し始めることができる。この明細書を読めば分かるはずであるが、データ不完全性はデータ削除の特別の場合であり、その理由は、受信者があたかも全時間にわたって有効範囲内にいたけれども受信者がデータ受信を開始した時点までの全てのデータをチャネルが紛失したかのように（受信者はそれと同じ問題を有する）受信者が該データ不完全性を取り扱うことができることにある。また、通信システムの設計に関して公知のように、検出可能なエラーを有する全てのデータブロック又はシンボルを単に落とすことによって、検出可能なエラーを削除と同等であると見なすことができる。

【００３２】
或る通信システムでは、受信者は複数の送信者により、或いは複数の接続を使用する１つの送信者により作られたデータを受信する。例えば、ダウンロードを高速化するために、受信者は、同じファイルに関するデータを送信する２つ以上の送信者と同時に接続することができよう。他の例として、マルチキャスト送信では、総伝送速度を受信者を送信者に接続するチャネルの帯域幅と釣り合わせるために、複数のマルチキャスト・データストリームが送信されて受信者がこれらのストリームのうちの１つ以上に接続できるようにする。全てのその様な場合に、１つの関心事は送信される全てのデータが受信者にとってはそれぞれ独立に使用されることを保証することである、即ち、送信速度が異なるストリームについて大幅に異なっているとき、及び任意のパターンのロスがある時でも複数ソース・データが該ストリーム間で冗長ではないことを保証することである。

【００３３】
一般に、通信チャネルは、データ伝送のために送信者と受信者を接続するものである。通信チャネルはリアルタイム・チャネルであって良くて、その場合には該チャネルはデータを入手するとデータを送信者から受信者に移動させ、或いは通信チャネルは、送信者から受信者へ移動するデータの一部又は全てを蓄積する記憶型チャネルであっても良い。後

者の例はディスク記憶装置又はその他の記憶装置である。この例では、データを作るプログラム又は装置は、データを記憶装置へ送る送信者であると見なすことができる。受信者は、その記憶装置からデータを読むプログラム又は装置である。記憶装置へデータを得るために送信者が使用するメカニズムと、その記憶装置自体と、該記憶装置からデータを得るために受信者が使用するメカニズムとが集合体としてチャネルを形成する。これらのメカニズム或いは記憶装置がデータを紛失する可能性があるならば、それは該通信チャネルにおけるデータ削除として取り扱われることになろう。

【００３４】

シンボルが削除される可能性のある通信チャネルによって送信者と受信者とが分離されているときには、入力ファイルの正確なコピーを送信せずに、削除箇所を回復するのに役立つ入力ファイルから作られたデータを送信することが好ましい。符号器は、このタスクを処理する回路、装置、モジュール或いは符号セグメントである。符号器の動作の１つの見方は、符号器が入力シンボルから出力シンボルを作ることであり、ここで入力シンボル値のシーケンスが入力ファイルを表わす。各入力シンボルは、入力ファイルの中での位置と値とを有する。復号器は、受信者により受信された出力シンボルから入力シンボルを復元する回路、装置、モジュール又は符号セグメントである。多段符号化では、符号器及び復号器は、更に、それぞれ異なるタスクを実行するサブモジュールに分割される。

【００３５】

多段符号化システムの実施態様では、符号器及び復号器は、更に、それぞれ異なるタスクを実行するサブモジュールに分割できる。例えば、或る実施態様では、符号器は、本書においてスタティック符号器と称されるものとダイナミック符号器とを含む。本書で使用されるとき、“スタティック符号器”は入力シンボルの集合から或る数の冗長シンボルを作る符号器であって、冗長シンボルの数は符号化の前に決定される。スタティック符号化符号の例は、リードソロモン符号、トルネード符号、ハミング符号、低密度パリティー検査符号（ＬＤＰＣ）符号などを含む。“スタティック復号器”という用語は、本書では、スタティック符号器により符号化されたデータを復号できる復号器を指すために使用される。

【００３６】

本書で使用されるとき、“ダイナミック符号器”は入力シンボルの集合から出力シンボルを作る符号器であり、可能な出力シンボルの数は入力シンボルの数より何桁も大きく、作られる出力シンボルの数は一定でなくても良い。ダイナミック符号器の一例は、ルビーＩ及びルビーＩＩに記載されている符号器などのチェーンリアクション符号器である。“ダイナミック復号器”という用語は、本書では、ダイナミック符号器により符号化されたデータを復号できる復号器を指すために使用される。

【００３７】

多段符号化の実施態様は、特定のタイプの入力シンボルに限定されなくても良い。通常、入力シンボルの値は或る正の整数Ｍについて2^{M}個のシンボルのアルファベットから選択される。その様な場合、入力シンボルは、入力ファイルからのデータのＭビットのシーケンスにより表示可能である。Ｍの値は、アプリケーションの使用法、通信チャネル、及び／又は出力シンボルのサイズに基づいて決定されることが良くある。更に、出力シンボルのサイズは、アプリケーション、チャネル、及び／又は入力シンボルのサイズに基づいて決定されることが良くある。或る場合には、出力シンボル値及び入力シンボル値が同じサイズだったならば（即ち、同数のビットにより表示可能であるか或いは同じアルファベットから選択されるならば）符号化プロセスを簡単化することができよう。その場合には、出力シンボル値サイズが制限されるときには入力シンボル値サイズが制限される。例えば、制限されたサイズのパケットに出力シンボルを入れることが望まれるであろう。もし出力シンボルと関連するキーを受信装置で復元するために該キーに関するデータを送信するならば、出力シンボルは、出力シンボル値と該キーに関する該データとを１パケットに収容するのに充分な小ささであるのが好ましいであろう。

【００３８】

例を挙げると、もし入力ファイルが複数メガバイトのファイルであれば、入力ファイルを、数千バイト、数百バイト、又は僅か数バイトを各々符号化する数千、数万或いは数十万の入力シンボルに分解することができよう。他の例として、パケット・ベースのインターネット・チャネルについては、１０２４バイトのサイズのペイロードを有するパケットが適切であろう（１バイトは８ビットである）。この例では、各パケットが１つの出力シンボルと８バイトの補助情報とを含むとすれば、８１２８ビット（（１０２４－８）＊８）の出力シンボルサイズが適切であろう。従って、入力シンボルのサイズをＭ＝（１０２４－８）＊８即ち８１２８ビットとして選択することができる。他の例として、或る衛星システムはＭＰＥＧパケット規格を使用し、この場合、各パケットのペイロードは１８８バイトから成る。この例では、各パケットが１つの出力シンボルと４バイトの補助情報とを含むとすると、１４７２ビット（（１８８－４）＊８）の出力シンボル・サイズが適切であろう。従って、入力シンボル・サイズをＭ＝（１８８－４）＊８即ち１４７２ビットとして選択することができる。多段符号化を使用する汎用通信システムでは、入力シンボル・サイズ（即ち、入力シンボルにより符号化されるビットの数Ｍ）等のアプリケーション特有のパラメータは、アプリケーションにより設定される変数であろう。

【００３９】

出力シンボルはそれぞれ値を有する。以下で考察する１つの好ましい実施態様では、各出力シンボルに、その“キー”と称される識別子も関連する。受信者が１つの出力シンボルを他の出力シンボルから識別できるように、受信者がそれぞれの出力シンボルのキーを容易に判定できることが好ましい。好ましくは、出力シンボルのキーは他の全ての出力シンボルのキーとは全く別のものである。従来技術においていろいろな形のキーイングが論じられている。例えば、ルビーＩは、本発明の実施態様に使用できる種々の形のキーイングを開示している。

【００４０】

多段符号化は、データ削除が予想される場合、或いは送信が始まるとき及び終わるときに受信者が正確に受信を始め及び終えない場合に特に有益である。この後者の状態は本書では“データ不完全性”と称される。削除イベントに関しては、多段符号化は、ルビーＩに記載されているチェーンリアクション符号化の利点の多くを共有する。具体的には、多段出力シンボルは情報付加的であって、入力ファイルを所望の精度で復元するために任意の適切な数のパケットを使用できる。多段符号化で作られた出力シンボルは情報付加的であるので、これらの条件は通信プロセスに悪影響を与えない。例えば、データ削除を生じさせるノイズのバーストに起因して百個のパケットが失われたならば、そのバーストの後に削除されたパケットの紛失に取って代わる余分の百個のパケットを拾うことができる。送信装置が送信を始めたときに受信装置がその送信装置に同調しなかったために数千個のパケットが紛失したならば、受信装置は、他の任意の送信期間から、或いは他の送信装置からでも、これら数千個のパケットを接受することができる。多段符号化では、受信装置は特定のパケットの集合を接受する様には制約されないので、受信装置は、１つの送信装置からパケットを幾つか接受し、他の送信装置へ転換し、パケットを幾つか紛失し、所与の送信の始まり或いは終了を逸し、それでもなお入力ファイルを復元することができる。受信装置・送信装置間の調整無しに送信に加わったり去ったりできる能力は、通信プロセスを簡単化するのに役立つ。

【００４１】

或る実施態様では、多段符号化を用いてファイルを送信することは、入力ファイルから入力シンボルを作り、形成し又は抽出することと、冗長シンボルを計算すること、各出力シンボルが該出力シンボルのキーに基づいて他の全ての出力シンボルと無関係に作られる場合に入力シンボル及び冗長シンボルを１つ以上の出力シンボルに符号化すること、該出力シンボルをチャネルを通して１つ以上の受信者へ送信することを含むことができる。更に、或る実施態様では、多段符号化を用いて入力ファイルのコピーを受信すること（及び復元すること）は、１つ以上のデータストリームから出力シンボルの集合又は部分集合を受信すること、及びその受信した出力シンボルの値及びキーから入力シンボルを復号する

ことを含むことができる。

【００４２】

図を参照して本発明のいろいろな局面を説明する。

【００４３】

（システムの概観）

図１は、多段符号化を使用する通信システム１００のブロック図である。通信システム１００では、入力ファイル１０１、又は入力ストリーム１０５が入力シンボル発生器１１０に提供される。入力シンボル発生器１１０は、入力ファイル又はストリームから１つ以上の入力シンボル（ＩＳ（０），ＩＳ（１），ＩＳ（２），・・・）のシーケンスを作り、その各入力シンボルは値及び位置（図１では括弧入り整数として表示されている）を有する。前述したように、入力シンボルについての可能な値、即ちそのアルファベット、は通常は2^M個のシンボルのアルファベットであるので、各入力シンボルは入力ファイルのＭビットについて符号化をする。Ｍの値は一般に通信システム１００を用いて決定されるが、汎用システムは、Ｍを使用毎に変更できるように入力シンボル発生器１１０のためのシンボルサイズ入力を包含するであろう。入力シンボル発生器１１０の出力は符号器１１５に提供される。

【００４４】

スタティックキー発生器１３０は、スタティックキーS_0，S_1，・・・のストリームを作る。作られるスタティックキーの数は、一般に制限され、符号器１１５の具体的実施態様に依存する。スタティックキーの発生については後に詳しく説明する。ダイナミックキー発生器１２０は、符号器１１５により作られる各出力シンボルのためにダイナミックキーを作る。各ダイナミックキーは、同じ入力ファイルのためのダイナミックキーの大部分がユニークであるように作られる。例えば、ルビーＩは、使用できるキー発生器の実施態様を開示している。ダイナミックキー発生器１２０及びスタティックキー発生器１３０の出力は符号器１１５に提供される。

【００４５】

ダイナミックキー発生器１２０により提供される各キーＩから、符号器１１５は、値Ｂ（Ｉ）を有する出力シンボルを、入力シンボル発生器により提供される入力シンボルから作る。以下で、符号器１１５の動作をもっと詳しく説明する。各出力シンボルの値は、そのキーに基づいて、入力シンボルのうちの１つ以上の入力シンボルの何らかの関数に基づいて、及び場合によっては入力シンボルから計算された１つ以上の冗長シンボルに基づいて、作られる。特定の出力シンボルを生じさせる入力シンボル及び冗長シンボルの集まりは、本書ではその出力シンボルの“関連シンボル”又は単にその“アソシエート”と称される。該関数（“価値関数”）及びアソシエートの選択については、以下で詳しく説明するプロセスに従って行われる。通常は、何時でもそうだというわけではないけれども、Ｍは入力シンボル及び出力シンボルについて同じである、即ち、両者共に同数のビットについて符号化をする。

【００４６】

或る実施態様では、入力シンボルの数Ｋは、アソシエートを選択するために符号器１１５により使用される。例えば入力がストリーミング・ファイルである場合など、Ｋが前以て知られていない場合には、Ｋは単なる推定値であって良い。値Ｋは、入力シンボルと、符号器１１５により作られた任意の中間シンボルとに記憶装置を割り当てるためにも符号器１１５により使用されるであろう。

【００４７】

符号器１１５は、出力シンボルを送信モジュール１４０に提供する。送信モジュール１４０には、ダイナミックキー発生器１２０から、その様な各出力シンボルのキーも提供される。送信モジュール１４０は出力シンボルを送信し、また使用されるキーイング方法に依存して送信モジュール１４０は送信される出力シンボルのキーに関するデータもチャネル１４５を介して受信モジュール１５０に送るであろう。チャネル１４５は削除チャネルであると仮定されるが、このことは通信システム１００が適切に動作するための必要条件

ではない。送信モジュール１４０が出力シンボルとそのキーに関する所用データとをチャネル１４５に送信し、受信モジュール１５０がシンボルと場合によってはそのキーに関するデータとをチャネル１４５から受信するようになっている限り、モジュール１４０，１４５及び１５０は任意の適切なハードウェア・コンポーネント、ソフトウェア・コンポーネント、物理的媒体、又はこれらの任意の組み合わせであって良い。Ｋの値は、アソシエートを決定するために使用される場合、チャネル１４５を介して送られても良く、或いは、Ｋの値は前もって符号器１１５及び復号器１５５の協定により設定されても良い。

【００４８】

前述したように、チャネル１４５は、例えばインターネットを通る経路又はテレビジョン送信装置からテレビジョン受信装置への放送リンク又は１つのポイントから他のポイントへの電話接続などのリアルタイム・チャネルであって良く、或いはチャネル１４５は例えばＣＤ－ＲＯＭ、ディスクドライブ、ウェブサイト等の記憶チャネルであって良い。チャネル１４５は、例えば一人の人が入力ファイルをパーソナルコンピュータから電話回線を介してインターネットサービスプロバイダ（ＩＳＰ）へ送信し、その入力ファイルがウェブサーバーに蓄積され、後にインターネットを介して受信者へ送られるときに形成されるチャネルなどの、リアルタイム・チャネルと記憶チャネルとの組み合わせであっても良いであろう。

【００４９】

チャネル１４５は削除チャネルであると仮定されているので、通信システム１００は、受信モジュール１５０から出る出力シンボルと送信モジュール１４０に入る出力シンボルとの間に１対１対応を仮定しない。実際は、チャネル１４５がパケットネットワークを含む場合には、通信システム１００は任意の２つ以上のパケットの相対的順序がチャネル１４５を通過するときに保存されるということを仮定できなくても良い。従って、出力シンボルのキーは、上記のキーイング方式のうちの１つ以上を用いて決定されるのであって、必ずしも出力シンボルが受信モジュール１５０を出る順序により決定されるのではない。

【００５０】

受信モジュール１５０は出力シンボルを復号器１５５に提供し、受信モジュール１５０がこれら出力シンボルのキーに関して受信するデータはダイナミックキー発生器１６０に提供される。ダイナミックキー発生器１６０は、受信された出力シンボルについてダイナミックキーを再生し、これらのダイナミックキーを復号器１５５に提供する。スタティックキー発生器１６３は、スタティックキーS_0，S_1，・・・を再生し、これを復号器１５５に提供する。スタティックキー発生器は、符号化及び復号の両プロセスのときに使用される乱数発生器１３５へのアクセスを有する。これは、乱数がその様な装置で作られるのであれば同じ物理的装置へのアクセスの形であって良く、或いは乱数を発生して同一動作を達成する同じアルゴリズムへのアクセスの形であっても良い。復号器１５５は、ダイナミックキー再生器１６０及びスタティックキー発生器１６３により提供されるキーを対応する出力シンボルと共に使用して入力シンボル（再びＩＳ（０），ＩＳ（１），ＩＳ（２），．．．）を復元する。復号器１５５は、回復された入力シンボルを入力ファイル・リアセンブラー１６５に提供するが、これは入力ファイル１０１又は入力ストリーム１０５のコピー１７０を作る。

【００５１】

（符号器）

図２は、図１に示されている符号器１１５の１つの具体的実施態様のブロック図である。符号器１１５はスタティック符号器２１０と、ダイナミック符号器２２０と、冗長性計算器２３０とを含む。スタティック符号器２１０は、次の入力即ち：ａ）入力シンボル発生器１１０により提供されて入力シンボル・バッファー２０５に蓄積された元の入力シンボルＩＳ（０），ＩＳ（１），．．．，ＩＳ（Ｋ－１）と；ｂ）元の入力シンボルの数Ｋと；ｃ）スタティックキー発生器１３０により提供されるスタティックキーS_0，S_1，．．．と；ｄ）冗長シンボルの数Ｒとを受信する。スタティック符号器２１０は、これらの入力を受け取ると、後述するようにＲ個の冗長シンボルＲＥ（０），ＲＥ（１），．．

．，ＲＥ（Ｒ－１）を計算する。通常は、常にそうというわけではないけれども、冗長シンボルは入力シンボルと同じサイズを有する。１つの特定の実施態様では、スタティック符号器２１０により作られた冗長シンボルは入力シンボル・バッファー２０５に蓄積される。入力シンボル・バッファー２０５は単に論理的であって良い、即ち、ファイルは１つの場所に物理的に蓄積されて良く、シンボル・バッファー２０５内での入力シンボルの位置は元のファイル内でのこれらのシンボルの位置の単なる名称変更であって良い。

【００５２】

ダイナミック符号器は、入力シンボル及び冗長シンボルを受け取り、以下で更に詳しく説明するように出力シンボルを作る。冗長シンボルが入力シンボル・バッファー２０５に蓄積される一実施態様では、ダイナミック符号器２２０は入力シンボル及び冗長シンボルを入力シンボル・バッファー２０５から受け取る。

【００５３】

冗長性計算器２３０は、入力シンボルの数Ｋから冗長シンボルの数Ｒを計算する。この計算について以下で更に詳しく説明する。

【００５４】

出力シンボルを作る速度が重要な資源である場合、入力ファイルは、スタティック符号器２１０を用いて符号化可能であり、出力シンボルの送信が始まる前に中間装置に蓄積される。この装置は、例えば、ダイナミック符号器２２０とは異なる物理的場所に取り付けられた記憶装置であって良く、これをダイナミック符号器２２０等の同じ物理的装置に包含させることができる。ファイルがダイナミック符号器２２０で符号化されるより充分前にスタティック符号器２１０で符号化される場合、ダイナミック符号器２２０を体現する計算装置が資源をスタティック符号化に専用する必要はない。従って、それは、例えば入力ファイルについて出力シンボルを作る速度を高め、他のファイルについて出力シンボルを作り、他のタスクを実行するなどの目的のためにダイナミック符号化により多くの資源を向けることができる。ダイナミック符号化の前にスタティック符号化を実行できるか或いは実行するべきであるか否かは具体的実施態様に依存する。

【００５５】

（スタティック符号器の概観）

図３及び４を参照してスタティック符号器２１０の一般的動作について説明する。図３は、静的符号化方法の一実施態様を示す略流れ図である。ステップ３０５で、何個の冗長シンボルが作られたか記録しておく変数ｊはゼロにセットされる。次に、ステップ３１０で、第１の冗長性シンボルＲＥ（０）は、入力シンボルＩＳ（０），・・・，ＩＳ（Ｋ－１）のうちの少なくともいくつかの関数F_0として計算される。その後、ステップ３１５で、変数ｊはインクリメントされる。次に、ステップ３２０で、全ての冗長シンボルが作られたか（即ち、ｊがＲ－１より大きいか）否かが試験される。もしイエスならば、流れは終了する。そうでなければ、流れはステップ３２５へ進む。ステップ３２５で、ＲＥ（ｊ）は入力シンボルＩＳ（０），．．．，ＩＳ（Ｋ－１）及び前に作られた冗長シンボルＲＥ（０），．．．，ＲＥ（ｊ－１）の関数Ｆｊとして計算され、このＦｊは入力シンボル及び冗長シンボルの一つ一つ全てに依存する関数でなくても良い。ステップ３１５，３２０及び３２５は、Ｒ個の冗長シンボルが計算され終わるまで反復される。

【００５６】

再び図１及び２を参照する。或る実施態様では、スタティック符号器２１０はスタティックキー発生器１３０から１つ以上のスタティックキーS_0，S_1，．．．を受け取る。これらの実施態様では、スタティック符号器２１０は、スタティックキーを使って関数F_0，F_1，．．．，F_{j-1}の一部又は全部を決定する。例えば、スタティックキーS_0を使って関数F_0を決定することができ、スタティックキーS_1を使って関数F_1を決定することができる、等々である。或いは、スタティックキーS_0，S_1，．．．のうちの１つ以上を使って関数F_0を決定することができ、スタティックキーS_0，S_1，．．．のうちの１つ以上を使って関数F_1を決定することができる、等々である。他の実施態様では、スタティックキーは不要であり、従ってスタティックキー発生器１３０は不要であ

る。

【００５７】

図２及び３を参照する。或る実施態様では、スタティック符号器２１０により作られた冗長シンボルを入力シンボル・バッファー２０５に蓄積することができる。図４は、スタティック符号器２１０の一実施態様の動作の略図である。具体的には、スタティック符号器２１０は、入力シンボル・バッファー２０５から受信された入力シンボルＩＳ（０），．．．，ＩＳ（Ｋ－１），ＲＥ（０），．．．，ＲＥ（ｊ－１）の関数Ｆｊとして冗長シンボルＲＥ（ｊ）を作り、それを逆に入力シンボル・バッファー２０５に蓄積する。関数F_0，F_1，．．．，F_{R-1}の正確な形は具体的アプリケーションに依存する。通常、常にそうであるというわけではないが、関数F_0，F_1，．．．，F_{R-1}は対応する独立変数の一部又は全部の排他的論理和を含む。前述したように、これらの関数は図１のスタティックキー発生器１３０により作られたスタティックキーを実際に使用しても良いしなくても良い。例えば、後述する一具体的実施態様では、始めの幾つかの関数はハミング符号を使用し、スタティックキーS_0，S_1，．．．を全く使用しないが、他の関数は低密度パリティー検査符号を実行し、スタティックキーを明示的に使用する。

【００５８】

（ダイナミック符号器の概観）

再び図２を参照する。ダイナミック符号器２２０は、入力シンボルＩＳ（０），．．．，ＩＳ（Ｋ－１）と冗長シンボルＲＥ（０），．．．，ＲＥ（Ｒ－１）と、自分が作る各出力シンボルのためのキーＩとを受け取る。元の入力シンボルと冗長シンボルとを含むコレクションは、以降は“ダイナミック入力シンボル”のコレクションと称される。図５は、ダイナミック符号器の一実施態様の略ブロック図である。この符号器は、ルビーＩに開示されている符号器の実施態様に類似している。ルビーＩは、この様な符号器の動作に関して詳しく開示している。

【００５９】

ダイナミック符号器５００は、重みセレクタ５１０と、アソシエータ５１５と、価値関数セレクタ５２０と計算装置５２５とを含む。図５に示されているように、Ｋ＋Ｒ個のダイナミック入力シンボルがダイナミックシンボル・バッファー５０５に蓄積される。一実施態様では、ダイナミックシンボル・バッファー５０５は図２の入力シンボル・バッファー２０５である。他の実施態様では、ダイナミックシンボル・バッファー５０５は入力シンボル・バッファー２０５とは別のものである。（図１に示されているダイナミックキー発生器１２０により提供される）ダイナミックキーＩは、重みセレクタ５１０、アソシエータ５１５及び価値関数セレクタ５２０への入力である。ダイナミック入力シンボルの数Ｋ＋Ｒもこれら３つのコンポーネント５１０，５１５及び５２０に提供される。計算装置５２５は、重みセレクタ５１０、アソシエータ５１５及び価値関数セレクタ５２０から出力を受け取り、且つダイナミックシンボル・バッファー５０５からシンボルを受け取るように結合されている。計算装置５２５は、出力シンボル値を作る。図５に示されている要素と同等の他の構成を使用しても良く、これは本発明の符号器の単なる一例に過ぎないことが理解されるべきである。例えば、ルビーＩ及びルビーＩＩは本発明の他の実施態様に使える他の符号器を開示している。

【００６０】

動作時には、Ｋ＋Ｒ個のダイナミック入力シンボルが入力シンボル・バッファー２０５から受け取られてダイナミック入力シンボル・バッファー５０５に蓄積される。前述したように、各ダイナミック入力シンボルは位置（例えば、入力シンボルの位置は入力ファイル内での自分の元の位置であって良い）と値とを有する。蓄積されたダイナミック入力シンボルの位置が判定可能である限り、ダイナミック入力シンボルはそれぞれの順でダイナミック入力シンボル・バッファー５０５に蓄積されなくても良い。

【００６１】

キーＩとダイナミック入力シンボルの数Ｋ＋Ｒとを使って、重みセレクタ５１０はキーＩを有する出力シンボルの“アソシエート”となるダイナミック入力シンボルの数Ｗ（Ｉ

）を決定する。キーＩと重みW（Ｉ）とダイナミック入力シンボルの数Ｋ＋Ｒとを使って、アソシエータ５１５は出力シンボルと関連するダイナミック入力シンボルの位置のリストＡＬ（Ｉ）を決定する。もしアソシエータ５１５が前もってW（Ｉ）を知らずにＡＬ（Ｉ）を作ることができるならば、W（Ｉ）を別に或いは明示的に計算しなくても良いことが理解されるべきである。ＡＬ（Ｉ）が計算されると、W（Ｉ）はＡＬ（Ｉ）に載っているアソシエートの数であるので、W（Ｉ）を容易に決定することができる。

【００６２】

アソシエータ５１５は、入力としてキーＩ、数Ｎ、及び数ｔを受け取って０とＮ－１との間の整数のリストＸ（０），．．．，Ｘ（ｔ－１）を作るマッピングである。好ましくは、これらの整数はそれぞれ他とは異なっていて自分達の範囲内に一様に分布している。例えば、図５のダイナミック符号器５００の場合、ＮはＫ＋Ｒに等しく、ｔはW（Ｉ）に等しく、ＡＬ（Ｉ）はリストＸ（０），．．．，Ｘ（ｔ－１）である。

【００６３】

アソシエータ５１５により与えられるマッピングは種々の形を取ることができる。それは、その出力をランダムにするために真にランダムなビット又は擬似ランダムなビットのソースへのアクセスを持つことができる。しかし、それは、同じキーＩ、同じＮ、及び同じｔについて符号器及び復号器の両方により同じ出力を作るように選択されるべきである。この要件を満たすために、キーＩでシーディングされた符号器及び復号器の両方により擬似ランダムなシーケンスを作ることができる。擬似ランダムなシーケンスの代わりに、出力を計算するために真にランダムなシーケンスを使うことができるであろうけれども、それが有益であるためには、出力を作るために使われるランダム・シーケンスを復号器に伝える必要があるであろう。

【００６４】

再び図５を参照する。Ｉ、W（Ｉ）及びＡＬ（Ｉ）が分かると、出力シンボルの値Ｂ（Ｉ）が価値関数Ｆ（Ｉ）に基づいて計算装置５２５により計算される。適切な価値関数の１つの特性は、該関数が、ＡＬ（Ｉ）により示されるアソシエートについての値を、出力シンボル値Ｂ（Ｉ）と、ＡＬ（Ｉ）により示される他のW（Ｉ）－１個のアソシエートについての値とから決定できるようにすることである。このステップで使われる１つの好ましい価値関数はＸＯＲ価値関数であるが、その理由は、該関数がこの特性を満たし、容易に計算され、容易に反転されることにある。しかし、代わりに、他の適切な価値関数を使用することもできよう。例えば、ルビーＩＩは、使用可能な他の適切な価値関数を開示している。

【００６５】

価値関数セレクタ５２０は、もし使用されれば、キーＩ及びＫ＋Ｒから価値関数Ｆ（Ｉ）を決定する。１つのバリエーションでは、価値関数Ｆ（Ｉ）は全てのＩについて同じ価値関数Ｆである。このバリエーションでは、価値関数セレクタ５２０は不要であり、計算装置５２５は価値関数Ｆを有するように構成されて良い。例えば、価値関数は全てのＩについてＸＯＲであって良く、即ち、出力シンボル値はそのアソシエートの全ての値のＸＯＲ（排他的論理和）である。

【００６６】

各キーＩについて、重みセレクタ５１０はＩ及びＫ＋Ｒから重みW（Ｉ）を決定する。１つのバリエーションでは、重みセレクタ５１０は、キーＩを使って始めにランダムな参照数を作り、次にこの数を使って、重みセレクタ５１０に蓄積されているか又は重みセレクタがアクセスできる分布テーブル内のW（Ｉ）の値を参照することによってW（Ｉ）を選択する。どのようにしてこの様な分布テーブルを形成してアクセスできるかを以下で説明する。重みセレクタ５１０がW（Ｉ）を決定すると、この値はアソシエータ５１５及び計算装置５２５に提供される。

【００６７】

リストＡＬ（Ｉ）、重みW（Ｉ）及び価値関数セレクタ５２０により提供される価値関数Ｆ（Ｉ）又は予め選択された価値関数Ｆを使って、計算装置５２５は、ダイナミック入

力シンボル・バッファー５０５内のＡＬ（Ｉ）により参照符付けされたW（Ｉ）個のダイナミック入力シンボルにアクセスして現在の出力シンボルについて値Ｂ（Ｉ）を計算する。ＡＬ（Ｉ）を計算するための手続きの例を以下で説明するが、代わりに他の適切な手続きを使っても良い。好ましくは、該手続きは、与えられた出力シンボルのためにアソシエートとして選択されるほぼ均等な機会を各入力シンボルに与え、もし復号器が利用できるＡＬ（Ｉ）を既に持っているのでなければ復号器が複製できるような方法で選択を行う。

【００６８】

ダイナミック符号器５００はＢ（Ｉ）を出力する。実際上、ダイナミック符号器５００は、図６に示されている動作を実行する、即ち、選択された入力シンボルの何らかの価値関数として出力シンボル値Ｂ（Ｉ）を作る。図示されている例では、価値関数はＸＯＲであり、出力シンボルの重みW（Ｉ）は３であり、関連するダイナミック入力シンボル（アソシエート）は位置０，２及びＫ＋Ｒ－２にあってそれぞれの値ＩＳ（０），ＩＳ（２）及びＲＥ（Ｒ－２）を有する。従って、出力シンボルは、Ｉのその値について：

【００６９】

【数２】

$$B(I) = IS(0) \oplus IS(2) \oplus RE(R\text{-}2)$$

として計算される。価値関数ＸＯＲが使われる場合には、冗長シンボルが元のシンボルＩＳ（０），．．．，ＩＳ（Ｋ－１）と同数のビットを有し、そしてこれらが出力シンボルと同数のビットを有することが分かる。

【００７０】

作られた出力シンボルは、前述したように送信されて受信される。本書では、出力シンボルうちの幾つかが無くなったり或いは順番が狂ったりしているかもしれず、或いは１つ以上の符号器により作られたということが仮定されている。しかし、受信された出力シンボルは、自分のキーの表示及び自分の値Ｂ（Ｉ）が正確であるという何らかの保証をもって受信されているということが仮定される。図１に示されているように、これらの受信された出力シンボルは、ダイナミックキー再生器１６０によりその表示から復元された対応するキー、値Ｋ及びＲ、及びスタティックキー発生器１６３により再生されたスタティックキーS_0，S_1，．．．と共に、復号器１５５への入力である。

【００７１】

（スタティック符号器）

スタティック符号器の主な機能は、たとえ削除があっても元のデータを復元できるように冗長情報を元のデータに付け加えることである。その様な冗長情報は、ダイナミック復号器が回復できない入力シンボルを復号器が回復するのを援助することができる。代表的アプリケーションでは、スタティック符号器は、たとえ削除があっても所望の精度で回復を保証するのに必要な冗長シンボルの数に関して、且つ／又は符号化プロセス及び／又は復号プロセスの計算費用に関して、効率的であるべきである。例えば、複数のアプリケーションにおいてダイナミック復号器の性能により必要とされる与えられた目標削除率ｐについて、目的は、たとえせいぜいデータの小部分ｐが紛失しても元のデータの速やかな回復を保証しながら冗長シンボルの数Ｒをなるべく少なくすることである。これらの要件を満たす符号の１つの種類は、当業者に周知されているＬＤＰＣ符号のそれである。これらの符号は、多くの場合に元のデータを回復できるけれども、元の入力シンボルのうちの２つ又は３つ以外は何でも回復できるという場合も希にある。或る実施態様では、ＬＤＰＣ符号化の前に、入力データは、始めに、削除が２つか３つあっても元のデータを回復できる符号を用いて符号化される。この最初の符号化は第１の複数の冗長シンボルを作る。この最初の符号化の後に、元の複数のシンボルと第１の複数の冗長シンボルとはＬＤＰＣ符号器を用いて符号化される。当業者に周知されていて以下で手短に説明される拡張ハミング符号は、３つ以下の削除があっても元のデータを回復することができ、そして少数の冗長シンボルを付加することによって該回復を実行するので、第１レベルの符号化の目的に適するものである。他の種類の符号化を使用できることが理解されるであろう。例えば、

或るアプリケーションでは、２つか３つの入力シンボル以外の全ての入力シンボルが回復されるということは容認できることであろう。その様なアプリケーションに適するのはＬＤＰＣ符号だけであろう。更に、特定のアプリケーションのためには、リードソロモン符号、トルネード符号などの他の種類の符号化が適しているであろう。従って、本発明の他の実施態様に従って多くの種類の符号化をそれぞれ単独で又は組み合わせて使用できることが理解されるであろう。

【００７２】

図７は、本発明に従うスタティック符号器の一具体的実施態様の略ブロック図である。スタティック符号器６００は、パラメータ計算装置６０５、ハミング符号器６１０、及び低密度パリティー検査（ｌｏｗ－ｄｅｎｓｉｔｙ－ｐａｒｉｔｙ－ｃｈｅｃｋ（ＬＤＰＣ））符号器６２０を含む。パラメータ計算装置６０５は、入力シンボルの数Ｋと、作られる冗長シンボルの数Ｒとを受け取り、パラメータＤ及びＥを作る。Ｄはハミング符号器６１０により作られる冗長シンボルの数の表示であり、ＥはＬＤＰＣ符号器６２０により作られる冗長シンボルの数の表示である。パラメータＤはハミング符号器６１０に提供され、パラメータＥはＬＤＰＣ符号器６２０に提供される。

【００７３】

ハミング符号器６１０は、入力シンボル・バッファー６２５からの入力シンボルＩＳ（０），．．．，ＩＳ（Ｋ－１）と、入力シンボルの数Ｋと、パラメータＤとを受け取るように結合される。これらに応答してハミング符号器６１０はハミング符号に従ってＤ＋１個の冗長シンボルＨＡ（０），ＨＡ（１），．．．，ＨＡ（Ｄ）を作る。一実施態様では、入力シンボル・バッファー６２５は図２の入力シンボル・バッファー２０５である。ハミング符号化プロセスは、Ｄ＋１個の冗長シンボルを元のＫ個の入力シンボルに付け加えるが、このＤは$2^{D}-D-1 \geqq K$となるような最小の数である。当業者に知られているように、入力シンボルの全部はゼロではないような入力シンボルの可能な任意のセッティングについて複数の入力シンボルと対応する冗長シンボルとのうちの少なくとも４つのシンボルがゼロでない様に冗長シンボルは選択される。この特性は、少なくとも３つの削除の訂正を保証する。ハミング符号器６１０は、エラー訂正及び削除箇所訂正符号の分野で当業者に知られている任意の数の方法で実現可能である。

【００７４】

ＬＤＰＣ符号器６２０は、入力シンボルＩＳ（０），．．．，ＩＳ（Ｋ－１）と、入力シンボル及びハミング符号化された冗長シンボルの数Ｋ＋Ｄ＋１と、パラメータＥと、スタティックキーS_0，S_1，．．．とを受け取るように結合されている。これに応答して、ＬＤＰＣ符号器６２０は、ＬＤＰＣ符号に従ってＥ個の冗長シンボルを作る。ＬＤＰＣ符号器により計算された冗長シンボルの数ＥはＲ－Ｄ－１に等しく、このＲは冗長シンボルの数である。当業者に知られているように、ＬＤＰＣ符号を用いて情報を符号化する方法はいろいろある。ＬＤＰＣ符号は、メッセージ・ノードの集合と、検査ノードの集合と、メッセージ・ノードを検査ノードに接続するエッジとを含むグラフ構造によって記述される。妥当なＬＤＰＣコードワードの集合は、各検査ノードについて隣り合うメッセージ・ノードのＸＯＲがゼロであるようなメッセージ・ノードのセッティングのセットである。或るアプリケーションでは、メッセージ・ノードがすべて同じ次数を有すること、即ち同数の検査ノードに接続されることが好ましいが、その理由は、このことが符号器の具体化を簡単にすると共に復号器のエラー確率の計算を容易にすることにある。本発明の一具体的実施態様では、各メッセージ・ノードが接続される検査ノードの数は４である。この数は符号器の実行時間／計算負荷と復号器の故障の確率との容認可能な妥協点を提供する数であることが見出されている。更に、所与のメッセージ・ノードに隣接する検査ノードは検査ノードの集合の中からランダムに選択されるのが好ましいということも見出されている。ＬＤＰＣ符号器６２０は、エラー訂正及び削除箇所訂正符号の当業者に知られている任意の数の方法で具体化可能である。

【００７５】

図８は、図７に示されているスタティック符号器を使用する本発明の一実施態様の動作

を示している。具体的には、ハミング符号器６１０は、入力シンボル・バッファー２０５（図２）から入力シンボルを受け取り、Ｄ＋１個のハミング符号化された冗長シンボルを作り、これらは入力シンボル・バッファー２０５に蓄積される。ＬＤＰＣ符号器６２０は入力シンボル・バッファー２０５から入力シンボルとＤ＋１個のハミング符号化されている冗長シンボルとを受け取り、Ｅ個のＬＤＰＣ符号化されている冗長シンボルを作り、これらは入力シンボル・バッファー２０５に蓄積される。

【００７６】

前記のように、或る実施態様では、ＬＤＰＣ符号器６２０は、図１のスタティックキー発生器１３０により作られたスタティックキーS_0，S_1，．．．を受け取る。一実施態様では、スタティックキー発生器１３０は、シードを受け取るとランダムな参照数のシーケンス（スタティックキーS_0，S_1，．．．）を作る乱数発生器である。シードはいろいろな形を取ることができる。例えば、シードは真正乱数発生器の値であって良い。他の例として、シードはＣＰＵクロックから決定論的に得られるストリングであって良い。そのシードが何であっても、復号器によってスタティックキーの同じシーケンスを作れるようにシードは復号器へ伝えられるべきである。多くのアプリケーションで、大きすぎないシードを持つのが有利であろう。多くのアプリケーションで、シードは３２ビットの整数、又は６４ビットの整数であって良い。

【００７７】

図６に示されているスタティック符号器６００の一具体的実施態様では、パラメータＤは、2^D-D-1がＫより大きいか又は等しくなる最大の整数Ｄとして計算される。更に、パラメータＥはＲ－Ｄ－１として計算される。図９は、前記のようにパラメータＤ及びＥを計算する、図７のパラメータ計算装置６０５などのパラメータ計算装置の一実施態様を示す略流れ図である。始めに、ステップ７０５で、パラメータＤは１に初期化される。次に、ステップ７１０で、2^D-D-1がＫより小さいか否か判定される。否であれば、流れはステップ７３０へ進む。もしイエスならば、流れはステップ７２０へ進み、ここでパラメータＤはインクリメントされる。その後、流れはステップ７１０に戻る。Ｄが決定されると、ステップ７３０でパラメータＥがＲ－Ｄ－１として計算される。

【００７８】

再び図１を参照する。或る特定のアプリケーションでは、チャネル１４５を介して送られるファイル又はストリームは小さい。例えば、入力ファイルは、短いオーディオメッセージ又は数十キロバイトから成るウェブページの内容であって良い。前記のスタティック符号器の特定の実施態様は、この様なシナリオでは最適ではないであろう。例えば、前記した実施態様の幾つかは、メモリ及びプロセッサの速度が効率よく使われず、従ってデータ復元が低速であるという結果に至る可能性がある。又、前記実施態様の幾つかは、システムのユーザにより設定された信頼度パラメータ内でデータを復元するために大きな受信オーバーヘッドを必要とする可能性がある。更に、前記実施態様の幾つかは、信頼性が所望のレベルより低いデータ復元に至る可能性がある。

【００７９】

入力シンボルの数が減らされると復号器の故障確率が大きくなることが見出されている。このことの原因は、主として、元の内容のサイズが割合に小さければ符号化プロセスが元の内容に関する情報を充分に作らないことであるということも見出されている。従って、元のシンボルに関する情報をより多く伝える冗長シンボルを作る他の符号器実施態様が使われても良い。図１０は、本発明の一実施態様に従うその様な符号器の略流れ図であり、これについて説明する。

【００８０】

始めに、ステップ８０５で、変数ｉがゼロに初期化される。変数ｉは、既に作られた冗長シンボルの数を追跡する。ステップ８１０で、Ｋ／２より大きいか又は等しい最小の奇数として数ｔが計算される。ステップ８１５で、Ｋ、ｔ及びスタティックキーＳｉに基づいて値P_1，P_2，．．．，P_tが計算される。値P_1，P_2，．．．，P_tは、冗長シ

ンボルを作るために使われる入力シンボルの位置を表わす。一具体的実施態様では、P_1，P_2，．．．，P_tを作るために図５のアソシエータ５１５などのアソシエータが使われる。具体的には、値ｔをＷ（Ｉ）入力として提供することができ、値ＫをＫ＋Ｒ入力として提供することができ、スタティックキーＳｉをキーＩ入力として提供することができる。ｔの多様な値が同様の符号化効果をもたらすことに注意するべきであり、従ってこの特定の選択肢は単なる例に過ぎない。

【００８１】

ステップ８２０で、ＲＥ（ｉ）の値が値ＩＳ（P_1），ＩＳ（P_2），．．．，ＩＳ（P_t）のＸＯＲとして計算される。ステップ８２５で、変数ｉが次の冗長シンボルの計算を準備するために１だけインクリメントされ、ステップ８３０で、全ての冗長シンボルが計算されたか否か判定される。否であれば、流れはステップ８１５に戻る。

【００８２】

（復号器概観）

図１１は、本発明に従う復号器の一実施態様を示す略ブロック図である。復号器９００は、例えば、図１の復号器１５５を具体化するために使用可能である。

【００８３】

復号器９００はダイナミック復号器９０５とスタティック復号器９１０とを含む。ダイナミック復号器９０５は、図１の受信モジュール１５０から出力シンボルＢ（I_a），Ｂ（I_b），．．．を受け取り、ダイナミックキー再生器１６０からダイナミックキーI_a，I_b，I_c，．．．を受け取る。これらのデータを受け取ると、ダイナミック復号器９０５は、入力シンボルＩＳ（０），．．．，ＩＳ（Ｋ－１）と冗長シンボルＲＥ（０），．．．，ＲＥ（Ｒ－１）とを復元しようと試みる。本発明の幾つかの実施態様の１つの利点は、ダイナミック復号器９０５が入力シンボル全部の復号を終えなくても良いことである。ダイナミック復号器９０５が回復しなかった入力シンボルを復号するためにスタティック復号器９１０を使用することができる。

【００８４】

ダイナミック復号器９０５により回復された入力シンボル及び冗長シンボルは復元バッファー９１５に蓄積される。ダイナミック復号が完了すると、スタティック復号器９１０は、ダイナミック復号器９０５により回復されなかった入力シンボルがもしあるならば、その入力シンボルを回復しようと試みる。具体的には、スタティック復号器９１０は復元バッファー９１５から入力シンボル及び冗長シンボルを受け取る。更に、スタティック復号器９１０は、もしスタティックキーS_0，S_1，．．．が使用されるなら、スタティックキー発生器１３０（図１）からスタティックキーS_0，S_1，．．．を受け取る。再び図１を参照すると、１つの特定の実施態様では、スタティックキー発生器１３０を駆動する乱数発生器１３５へ通信チャネル１４５を通して共通シードを伝えることによってスタティックキーを再生することができる。回復された入力シンボルは入力ファイル・リアセンブラー１６５に提供される。

【００８５】

図１２は、本発明に従う復号方法の一実施態様を示す略流れ図である。ステップ１００５で、Ｑ個の出力シンボルが復号器により受け取られる。Ｑの値は、入力シンボルの数と、使用される具体的ダイナミック符号器とに依存して良い。Ｑの値は、復号器が入力シンボルを回復できる所望の精度にも依存して良い。例えば、復号器が全ての入力シンボルを高い確率で回復できることが希望される場合、Ｑは入力シンボルの数より大きい様に選択されるべきである。具体的には、或る実施態様では、入力シンボルの数が大きいとき、Ｑを元の入力シンボルの数より３％未満だけ大きくて良い。他のアプリケーションでは、入力シンボルの数が小さいとき、Ｑは入力シンボルの数より少なくとも１０％大きくて良い。具体的には、Ｑを入力シンボルの数Ｋプラス或る数Ａとして選択することが出来、このＡは、復号器が全ての入力シンボルを高い確率で確実に再生できるように選択される。数Ａの決定について以降で詳しく説明する。もし復号器が全ての入力シンボルは（時折或いは常に）復号できないということが容認可能であるならば、ＱはＫ＋Ａ未満であって良く

、Ｋに等しくても良く、或いはＫより小さくても良い。明らかに、符号化システム全体の１つの目的は、所望の精度に関して復号プロセスが成功する良好な蓋然論的保証を保ちながら数Ｑをなるべく減らすことであるということが良くある。

【００８６】

ステップ１０１０で、ダイナミック復号器９０５はＱ個の受信された出力シンボルから入力シンボル及び冗長シンボルを再生する。ステップ１００５及び１０１０を実質的に同時に実行できることが理解されるべきである。例えば、ダイナミック復号器９０５は、該復号器がＱ個の出力シンボルを受け取る前に入力シンボル及び冗長シンボルを再生し始めることができる。

【００８７】

ダイナミック復号器９０５がＱ個の出力シンボルを処理した後、入力シンボルが所望の精度で回復されているか否か判定される。所望の精度は、例えば、入力シンボルの全部であっても良く、或いは入力シンボルの全部よりは少ない数、パーセンテージ等であっても良い。もしイエスならば、流れは終了する。もしノーならば、流れはステップ１０２０へ進む。ステップ１０２０で、スタティック復号器９１０は、ダイナミック復号器９０５が回復できなかった入力シンボルを回復しようと試みる。スタティック符号器９１０がダイナミック符号器９０５により回復された入力シンボル及び冗長シンボルを処理し終えると、流れは終了する。

【００８８】

図１３は、本発明に従う復号方法の他の実施態様を示す略流れ図である。この実施態様は、図１２に関して説明したものと類似しており、ステップ１００５，１０１０，１０１５、及び１０２０を共通に含んでいる。しかし、ステップ１０２０の後に、流れはステップ１０３０に進み、ここで入力シンボルが所望の精度で回復されたか否か判定される。イエスならば、流れは終了する。ノーならば、流れはステップ１０３５へ進む。ステップ１０３５で、１つ以上の追加の出力シンボルが受け取られる。その後、流れはステップ１０１０へ逆に進み、ダイナミック復号器９０５及び／又はスタティック復号器９１０は、残っている回復されていない入力シンボルを回復しようと試みることができる。

【００８９】

図１４は、本発明に従う復号方法のもう一つの実施態様を示す略流れ図である。ステップ１０５５で出力シンボルが復号器により受信され、ステップ１０６０でダイナミック復号器９０５は受信された出力シンボルから入力シンボルと冗長シンボルとを再生する。その後、ステップ１０６５で、ダイナミック復号を終了するべきか否か判定される。この判定の基礎を、処理された出力シンボルの数、回復された入力シンボルの数、追加の入力シンボルが回復されつつある現在の速度、出力シンボルを処理して費やされた時間などのうちの１つ以上に置くことができる。

【００９０】

ステップ１０５５，１０６０及び１０６５を実質的に同時に実行できることが理解されなければならない。例えば、復号器が出力シンボルを受け取り続けているときにダイナミック復号器９０５は入力シンボル及び冗長シンボルを再生し始めることができる。更に、出力シンボルが受信されつつある間に且つ／又は出力シンボルがダイナミック復号器９０５により処理されつつある間にダイナミック復号プロセスを止めるか否かの評価を周期的に実行することができる。

【００９１】

ステップ１０６５で、ダイナミック復号を止めるべきではないと判定されたならば、流れはステップ１０５５に戻る。しかし、ステップ１０６５でダイナミック復号を終了すると決定されたならば、流れはステップ１０７０に進む。ステップ１０７０では、入力シンボルが所望の精度で回復されたか否か判定される。もしイエスならば、流れは終了する。もしノーならば、流れはステップ１０７５に進む。ステップ１０７５で、スタティック復号器９１０は、ダイナミック復号器９０５が回復できなかった入力シンボルを回復しようと試みる。ダイナミック符号器９０５により回復された入力シンボル及び冗長シンボルを

スタティック符号器９１０が処理し終えると、流れは終了する。

【００９２】

（ダイナミック復号器）

図１５は、本発明に従うダイナミック復号器の一実施態様を示す。ダイナミック復号器１１００は、図５に示されているダイナミック符号器５００のものと同様のコンポーネントを含んでいる。復号器１１００は、ルビーＩ及びルビーＩＩに開示されているチェーンリアクション復号器の実施態様と同様である。ダイナミック復号器１１００は、重みセレクタ５１０と、アソシエータ５１５と、価値関数セレクタ５２０と、出力シンボル・バッファー１１０５と、リデューサー１１１５と、復元装置１１２０と復元バッファー１１２５とを含む。符号器と同じく、価値関数セレクタ５２０と、価値関数の記述を蓄積するために割り当てられる出力シンボル・バッファー１１０５内のスペースとは任意のものであって、もし価値関数が全ての出力シンボルについて同じであるならば、使用されなくても良いであろう。復元バッファー１１２５のエントリーが幾つか示されており、幾つかの入力シンボルは復元され、その他はまだ不明で、疑問符により示されている。例えば、図１５において、位置０，２，５，６、及びＫ－１の入力シンボルと位置０及び２の冗長シンボルとは回復されており、位置１，３及び４の入力シンボルと位置１の冗長シンボルとはやがて回復されなければならない。

【００９３】

動作時には、キーＩ及び値Ｂ（Ｉ）を有する受信された各出力シンボルについて、復号器１１００は次のような動作を行う。キーＩは、価値関数セレクタ５２０と、重みセレクタ５１０とアソシエータ５１５とに提供される。Ｋ＋Ｒ及びダイナミックキーＩを使って重みセレクタ５１０は重みＷ（Ｉ）を決定する。Ｋ＋Ｒ、ダイナミックキーＩ、及びＷ（Ｉ）を使ってアソシエータ５１５は、出力シンボルと関連する入力シンボル及び冗長シンボルのＷ（Ｉ）個の位置のリストＡＬ（Ｉ）を作る。任意に、Ｋ＋Ｒ及びＩを使って、価値関数セレクタ５２０は価値関数Ｆ（Ｉ）を選択する。Ｉ、Ｂ（Ｉ）、Ｗ（Ｉ）及びＡＬ（Ｉ）、並びに任意にＦ（Ｉ）が出力シンボル・バッファー１１０５の一つのローに蓄積される。価値関数セレクタ５２０、重みセレクタ５１０及びアソシエータ５１５は、ダイナミック符号器２２０（図２）について記述したものと同じ動作を復号器１１０５に対して実行する。具体的には、図１５の価値関数セレクタ５２０、重みセレクタ５１０及びアソシエータ５１５により作られる価値関数Ｆ（Ｉ）、重みＷ（Ｉ）及びリストＡＬ（Ｉ）は、同じダイナミックキーＩについては図５に示されている対応する部分のものと同じである。Ｋ及びＲが入力ファイル毎に異なるならば、これらを例えばメッセージヘッダに含めるなどの任意の在来方法で符号器から復号器へ伝えることができる。

【００９４】

復元装置１１２０は、出力シンボル・バッファー１１０５を走査して、該バッファーに蓄積されている重み１即ちＷ（Ｉ）＝１を有し、ＡＬ（Ｉ）が１つのアソシエートだけの位置を記録している出力シンボルを探す。これらのシンボルは、本書では“復号可能な集合”のメンバーと称される。上記の特性を有する価値関数については、重み１の出力シンボルは復号可能な集合の中にあり、その理由は、ダイナミック入力シンボルの値をその出力シンボルから決定できることにある。もちろん、重み１を有するという条件以外の条件の下でダイナミック入力シンボルの復号を可能にする価値関数が使用されるならば、その条件は出力シンボルが復号可能な集合の中にあるか否か判定するために使われる。説明を明瞭にするために、ここに記載されている例は、復号可能な集合がこれらの重み１を有する出力シンボルであるということを仮定しており、これらの例の、他の価値関数復号可能条件への拡張は、この記述から明白であるはずである。

【００９５】

復号可能な集合の中にある出力シンボルを復元装置１１２０が発見すると、その出力シンボルの値Ｂ（Ｉ）と、任意に価値関数Ｆ（Ｉ）とがＡＬ（Ｉ）に載っているダイナミック入力シンボルを復元するために使われ、復元されたダイナミック入力シンボルは、復元バッファー１１２５の中の、その入力シンボル或いは冗長シンボルのために適切な位置に

置かれる。指示された入力シンボル或いは冗長シンボルが既に復元されていれば、復元装置１１２０は、新たに復元されたダイナミック入力シンボルを落とし、現存する復元された入力シンボル或いは冗長シンボルを上書きし、或いはその２つを比較してもし違っていればエラーを発することができる。価値関数が全てのアソシエートのＸＯＲである場合、入力シンボル或いは冗長シンボルの値は単純にその出力シンボルの値である。このように、復元装置１１２０は、入力シンボル及び冗長シンボルを復元するが、復号可能な集合の中の出力シンボルのみから復元を行う。復号可能な集合からの出力シンボルが入力シンボル或いは冗長シンボルを復元するために使われた後、出力シンボル・バッファー１１０５のスペースを節約するためにそれを削除することができる。“使い終わった”出力シンボルを削除すれば、復元装置１１２０が頻繁にその出力シンボルに再訪することが無いことが保証されることになる。

【００９６】

始めに、復元装置１１２０は、復元可能な集合のメンバーである出力シンボルが少なくとも１つ受信されるまで待つ。その１つの出力シンボルが使用されてしまうと、復号可能な集合は再び空となるが、他の何らかの出力シンボルがその１つの復元された入力シンボル或いは冗長シンボルと他の１つの入力シンボル或いは冗長シンボルとのみの関数である可能性がある。従って、復号可能な集合のメンバーから１つの入力シンボル或いは冗長シンボルを復元すると他の出力シンボルが復号可能な集合に付け加えられることになる可能性がある。出力シンボルを減少させてそれを復号可能な集合に付け加えるプロセスはリデューサー１１１５により実行される。

【００９７】

リデューサー１１１５は、復元された入力シンボル或いは冗長シンボルの位置を記録してあるリストＡＬ（Ｉ）を有する出力シンボルを発見するために出力シンボル・バッファー１１０５及び復元バッファー１１２５を走査する。

リデューサー１１１５がキーＩを有するその様な“減少可能な”出力シンボルを発見すると、リデューサー１１１５は位置ｈに存する回復されたダイナミック入力シンボルの値ＩＳ（ｈ）を得て、Ｂ（Ｉ）、Ｗ（Ｉ）及びＡＬ（Ｉ）を次のように変更する：

Ｂ（Ｉ）は

【００９８】

【数３】

$$B(I) \oplus IS(h)$$

にリセットされ、

Ｗ（Ｉ）はＷ（Ｉ）－１にリセットされ、

ＡＬ（Ｉ）はｈを除いたＡＬ（Ｉ）にリセットされる。

【００９９】

上の方程式では価値関数が全てのアソシエートの値のＸＯＲであることが前提とされている。ＸＯＲはそれ自身の逆であることに注意しなければならない。もしそうではなくて、出力シンボルを計算するために元は他の価値関数が使用されたのであれば、その価値関数の逆がここでリデューサー１１１５により使用されることになる。明らかに、２つ以上のアソシエートについて値が分かっているならば、Ｂ（Ｉ）を不明の何らかのアソシエート値のみに依存させるために（そして、それに応じてＷ（Ｉ）及びＬ（Ｉ）を調整する）上記方程式の均等物を計算することができる。

【０１００】

リデューサー１１１５の動作は、出力シンボル・バッファー１１０５内の出力シンボルの重みを減少させる。出力シンボルの重みが１に減少されると（或いは他の価値関数については他の復号可能条件が生じる）、その出力シンボルは復号可能な集合のメンバーになり、これに対して復元装置１１２０が働きかけることができるようになる。実際には、充分な数の出力シンボルが受信されると、リデューサー１１１５及び復元装置１１２０はチェーンリアクション復号を作り、復元装置１１２０は復号可能な集合を復号してより多く

のダイナミック入力シンボルを回復し、リデューサー１１１５はこれらの新たに回復された入力シンボル或いは冗長シンボルを使ってより多くの出力シンボルを減少させ、それらは復号可能な集合に付け加えられ、復号可能な集合が空になるまでこの様な動作が行われる。

【０１０１】

図１５に示されている復号器は、記憶装置、計算サイクル或いは転送時間をあまり考慮せずに簡単な方法で入力シンボル及び冗長シンボルを部分的に復元する。復号器メモリ、復号時間或いは転送時間（これは、受信された出力シンボルの数を束縛する）が制限されている場合には、これらの制限されている資源をより良く使用するために復号器を最適化することができる。この様な最適化の例が、例えば、ルビーＩ及びルビーＩＩに開示されている。該最適化を多段符号のダイナミック復号にも使うことができる。更に、他のバリエーション及び同等の復号器を使えることが理解されよう。

【０１０２】

（スタティック復号器）

図１６は、スタティック復号器の一実施態様を示す略ブロック図である。この実施態様は、図７を参照して記載されたスタティック符号器などのスタティック符号器でデータが符号化されるときに、使用可能である。スタティック復号器１２００は、ＬＤＰＣ復号器１２０５とハミング復号器１２１０とを含む。ＬＤＰＣ復号器１２０５は、復元バッファー１２１５から入力シンボル及び冗長シンボルを受け取り、ダイナミック復号器の復号ステップの後に回復されていない復元バッファー１２１５のシンボルを復元しようと試みる。或る実施態様では、復元バッファー１２１５は復元バッファー１１２５（図１５）である。ＬＤＰＣ復号器１２０５は、スタティックキー発生器１３０により作られたスタティックキーS_0，S_1，．．．を受け取る。更に、ＬＤＰＣ復号器１２０５は、入力シンボルの数Ｋ、冗長ハミング・シンボルの数Ｄ、及び冗長ＬＤＰＣシンボルの数Ｅを受け取る。ＬＤＰＣ復号器１２０５は、なるべく多くの入力シンボル及び冗長シンボルを当業者に周知されている方法で回復し、これらの値を、復元バッファー１２１５内のこれらに対応する位置に書き込む。

【０１０３】

ハミング復号器１２１０は、復元バッファー１２１５から入力シンボル及び冗長シンボルを受け取るようにも結合されている。更に、ハミング復号器１２１０は、入力シンボルの数Ｋと、数Ｄとを受け取り、ここでＤ＋１は冗長ハミング・シンボルの数である。ハミング復号器１２１０は、ダイナミック復号器及びＬＤＰＣ復号器１２０５により回復されなかった入力シンボルを回復しようと試みる。ＬＤＰＣ復号器１２０５の目的は、入力シンボル及び冗長シンボルをなるべく多く回復することであるのに対して、ハミング復号器１２１０は入力シンボルＩＳ（０），ＩＳ（１），．．．，ＩＳ（Ｋ－１）を回復しようと試みるに過ぎない。

【０１０４】

ＬＤＰＣ復号器及びハミング復号器の多数のバリエーションが当業者に周知されており、それらを本発明のいろいろな実施態様に使用できる。１つの具体的実施態様では、ハミング復号器はガウス消去アルゴリズムを用いて具体化される。ガウス消去アルゴリズムの多数のバリエーションが当業者に周知されており、それらを本発明のいろいろな実施態様に使用することができる。

【０１０５】

一定のアプリケーションでは、上記のものとは異なるタイプの図１に示されている復号器１５５を使う方が良い。例えば、もし入力シンボルの数Ｋがあまり大きくなくて、例えば１０００未満であれば、出力シンボルの受信の蓋然論的プロセスに係わる平方偏差によって余儀なくされて、ダイナミック復号器及びスタティック復号器が所用の数の削除を訂正できるようにするために復号器１５５はＫよりかなり大きな数の出力シンボルを集めざるを得なくなるであろう。その様な場合、異なるタイプの復号器を使うことができる。ガウス消去法を用いてデータを復号するその様な復号器の実施態様について図１７，１８及

び１９を参照して説明する。

【０１０６】

始めに、再び図１を参照すると、復号器１５５は、受信モジュール１５０から出力シンボルＢ（I_a）、Ｂ（I_b），．．．を受け取り、ダイナミックキー再生器１６０からキーI_a，I_b，．．．を受け取り、スタティックキー発生器１３０からキーS_0，S_1，．．．を受け取る。更に、それは、入力シンボルの値Ｋ及び冗長シンボルの値Ｒを受け取る。この入力を受け取ると、それは入力シンボルＩＳ（０），．．．，ＩＳ（Ｋ－１）を復元しようと試み、これらは入力ファイル・リアセンブラー１６５に渡されて更に処理される。

【０１０７】

図１７を参照すると、復号器１３００はダイナミック・マトリックス発生器１３０５とスタティック・マトリックス発生器１３１０とを含む。ダイナミック・マトリックス発生器１３０５は、出力シンボルＢ（Ｉａ），Ｂ（Ｉｂ），．．．、ダイナミックキーＩａ，Ｉｂ，．．．、及びパラメータＫ及びＲを受け取る。更に、ダイナミック・マトリックス発生器１３０５は、何個の出力シンボルを集めるべきか記述するもう一つのパラメータＡを受け取る（即ち、集められる出力シンボルの数はＫ＋Ａである）。パラメータＡの決定は、通常はダイナミック符号化及びスタティック符号化に用いられる方法に依存し、以下で詳しく説明される。以下の記述では、集められたＫ＋Ａ個の出力シンボルはＢ（０），Ｂ（１），．．．，Ｂ（Ｋ＋Ａ－１）と称される。これらのパラメータが受信されると、

Ｃ＊Ｔｒａｎｓｐｏｓｅ（ＩＳ（０），．．．，ＩＳ（Ｋ－１），ＲＥ（０），．．．，ＲＥ（Ｒ－１））＝

Ｔｒａｎｓｐｏｓｅ（Ｂ（０），．．．，Ｂ（Ｋ＋Ａ－１））

の形の線型方程式の系がダイナミック・マトリックス発生器１３０５により確立され、ここでＣは（Ｋ＋Ａ）ｘ（Ｋ＋Ｒ）のフォーマットのマトリックスである。ダイナミック・マトリックス発生器１３０５によるマトリックスＣの作成について以下で更に詳しく説明する。

【０１０８】

スタティック・マトリックス発生器１３１０は、マトリックスＣをダイナミック・マトリックス発生器１３０５から受け取り、キーS_0，S_1，．．．を使ってＣを追加のＲ個のローだけ大きくして方程式系

Ｍ＊Ｔｒａｎｓｐｏｓｅ（ＩＳ（０），．．．，ＩＳ（Ｋ－１），ＲＥ（０），．．．，ＲＥ（Ｒ－１）＝

Ｔｒａｎｓｐｏｓｅ（Ｂ（０），．．．，Ｂ（Ｋ＋Ａ－１），０，．．．，０）

を得る。ここで右辺のベクトルの終わりの方のＲ個のエントリーはゼロであり、Ｍは（Ｋ＋Ａ＋Ｒ）ｘ（Ｋ＋Ｒ）のフォーマットである。最後に、線形方程式系ソルバー１３１５が、この方程式系Ｍを解いて入力シンボルＩＳ（０），．．．，ＩＳ（Ｋ－１）の一部又は全部を得るために使用される。一具体的実施態様では、方程式系ソルバー１３１５はガウス消去アルゴリズムを使って線形方程式系を解く。

【０１０９】

図５のダイナミック符号器５００と図２のスタティック符号器２１０とを参照してダイナミック・マトリックス発生器１３０５及びスタティック・マトリックス発生器１３１０について更に詳しく説明する。図１８は、ダイナミック・マトリックス発生器１３０５により使用される方法の一実施態様を示す略流れ図である。ステップ１４０５で、ダイナミック・マトリックス発生器１２０５は、フォーマット（Ｋ＋Ａ）ｘ（Ｋ＋Ｒ）のマトリックスＣを全部ゼロに初期化する。次に、ステップ１４１０で、キーＩａ，Ｉｂ，．．．が重みセレクタ５１０及びアソシエータ５１５と関連して使用されて、それぞれ重みＷ（０），．．．，Ｗ（Ｋ＋Ａ－１）及びリストＡＬ（０），．．．，ＡＬ（Ｋ＋Ａ－１）を作る。リストＡＬ（Ｋ）の各々は０，．．．，Ｋ＋Ｒ－１の範囲内のＷ（ｋ）個の整数を含

む。ステップ１４１５で、これらの整数はＡＬ（ｋ）＝（ａ（０），．．．，ａ（Ｗ（ｋ）－１））でＣ（ｋ，１）を計算するために使われ、エントリーＣ（ｋ，ａ（０），．．．，Ｃ（ｋ，ａ（Ｗ（ｋ）－１））は１にセットされる。上記のように、マトリックスＣは、未知数（ＩＳ（０），．．．，ＩＳ（Ｋ－１），ＲＥ（０），．．．，ＲＥ（Ｒ－１））について、受信されたシンボル（Ｂ（０），．．．，Ｂ（Ｋ＋Ａ－１））の項で方程式系をもたらす。その理由は次の通りである、即ち、ダイナミック符号器が重みＷ（ｋ）とアソシエート・リストＡＬ（ｋ）＝（ａ（０），．．．，ａ（Ｗ（ｋ）－１））とを選択すると、対応する出力シンボルＢ（ｋ）は

【０１１０】

【数４】

$$B(k) = L(a(0)) \oplus L(a(1)) \oplus \ldots \oplus L(a(W(k)-1)),$$

として得られ、ここでＬ（ｊ）は復元バッファー１９２５内の位置ｊの未知数の値を意味する。これらの方程式は、０とＫ＋Ａ－１との間のｋの全ての値について蓄積されると、所望の方程式系をもたらす。

【０１１１】

図１９は、スタティック・マトリックス発生器１３１０により使用される方法の一実施態様を示す略流れ図である。この実施態様について、図１０を参照して説明する。図１０のステップ８２０で、冗長シンボルＲＥ（ｉ）が

【０１１２】

【数５】

$$RE(i) = IS(P_1) \oplus \ldots \oplus IS(P_t), \text{ and } P_1, P_2, \ldots, P_t$$

として計算され、キーＳｉが受信されるとP_1，P_2，．．．，P_tがステップ８１５のように計算されることに注意しなければならない。このことは、

【０１１３】

【数６】

$$IS(P_1) \oplus \ldots \oplus IS(P_t) \oplus RE(i) = 0$$

であることを意味する。復元バッファーの位置に関して言えば、このことは

【０１１４】

【数７】

$$L(P_1) \oplus \ldots \oplus L(P_t) \oplus L(i+K) = 0$$

であることを意味する。Ｍのエントリー（ｉ，P_1），．．．，（ｉ，P_t），（ｉ，ｉ－Ａ）を１に等しくセットし、ここでｉがＫ＋ＡからＫ＋Ａ＋Ｒ－１まで変化すると、受信されたシンボルＢ（０），．．．，Ｂ（Ｋ＋Ａ－１）に関して未知数（ＩＳ（０），．．．，ＩＳ（Ｋ－１），ＲＥ（０），．．．，ＲＥ（Ｒ－１））についての線形方程式系を記述するマトリックスＭが得られる。

【０１１５】

ステップ１５０５で、（Ｋ＋Ｒ＋Ａ）ｘ（Ｋ＋Ｒ）のフォーマットのマトリックスＭは、Ｍの始めのＫ＋Ａ個のローをダイナミック・マトリックス発生器１３０５により計算されたマトリックスＣに等しくすることによって、初期化される。Ｍの残りのローはゼロに初期化される。次に、ステップ１５１０で、変数ｉはＫ＋Ａに初期化される。この変数は、Ｍの最後のＲ個のローを追跡する。ステップ１５１２で、冗長シンボルｉ－Ｋ－Ａのアソシエートの数ｔが計算される。このステップは、図８のステップ８１０と同様である。具体的には、図８に与えられているスタティック符号化プロセスのときにｔの他の選択肢の方が好ましければ、その選択肢は、ステップ１５１２で計算される変数ｔについても採用されるべきである。ステップ１５１５で、アソシエータ５１５は、スタティックキーＳｉ、入力シンボルの数Ｋ、及び整数ｔから０とＫ－１との間のインデックスP_1，．．．

，P_tを計算する。マトリックスMの対応する位置がステップ１５３０で１にセットされる。ステップ１５４０のインクリメントとステップ１５５０の試験とは、Mの最後のR個のローの全てが尋ねられ計算されることを保証する。

【０１１６】

或る場合には、図１７，１８及び１９に示されている実施態様は、復号を首尾良く行うために他の実施態様よりは相対的に少ない出力シンボルを集めることを考慮に入れているので、本書に記載された他の実施態様よりおそらく適切であろう。復号器の選択は、アプリケーションと、例えば集められる出力シンボルの数が決定的な資源であるか否かということなどに大幅に左右される。

【０１１７】

（アソシエータ実施態様）

再び図５を参照する。アソシエータ５１５の一実施態様がルビーＩに記載されている。その場合、数Nは素数であるべきである。動作に関して、この実施態様がＡＬ（Ｉ）を計算するために使われるとき、入力サイズＫ＋Ｒは素であるように調整される。本発明では、Ｋ＋Ｒが素となるように冗長シンボルの数は充分に大きく選択される。或るアプリケーションでは、入力Nが素数であるという条件はかなり制限的である。

【０１１８】

Nが素数でなくても良いアソシエータ５２０を具体化する方法の他の実施態様が図２０に示されている。始めに、ステップ１８０５で、変数ｋがゼロに初期化される。次に、ステップ１８１０で、ランダム整数Ｙが作られる。１つの具体的実施態様では、出力シンボルについてのキーＩを使って乱数発生器にシーディングする。次に、ステップ１８１５で、０とＮ－１との間の数を作るためにＹはNを法として取られる。ステップ１８２０で、候補の数Ｙは、前に作られた他の数Ｙ（Ｘ（０），Ｘ（１），．．．））と比較される。数Ｙが前に作られていれば、流れはステップ１８１０に戻る。そうでなければ、ステップ１８２５で、それは、リストＸ（０），Ｘ（１）．．．の中に含められる。次に、ステップ１８３０で、Ｗ（Ｉ）個の数が作られたか否か判定される。否であれば、流れはステップ１８１０に戻る。図２０に示されている流れの結果はＷ（Ｉ）個の数Ｘ（０），Ｘ（１），．．．，Ｘ（Ｗ（Ｉ）－１）のリストであり、このリスト中の各数Ｘは０とＮ－１との間のユニークな整数である。次に、ステップ１８３５で、リストＡＬ（Ｉ）は数Ｘ（０），Ｘ（１），．．．Ｘ（Ｗ（Ｉ）－１）としてセットされる。

【０１１９】

（重みセレクタの実施態様）

符号器／復号器の性能及び効率は、図２に示されているようにダイナミック符号器２２０により作られる出力シンボルの重みの分布に依存し、一部の分布は他より良好である。具体的には、入力シンボルの数Ｋと比較される集められた出力シンボルの数の超過分を記述するパラメータＡの選択は、主として重み分布の選択により左右される。重み選択の動作に関する局面を以下で説明し、その後に或る重要な重み分布について説明する。これらの考えを示すために図２１のブロック図と図２２の流れ図とを用いる。

【０１２０】

図５に示されている重みセレクタ５１０のタスクは次の通りである。即ち、キーＩと長さＫ＋Ｒとを受け取ると、重みセレクタは、重みと称される０からＫ＋Ｒ－１までの範囲の整数Ｗ（Ｉ）を出力する。理想的には一様なランダム分布で整数を一様に作るアソシエータ５１５とは異なって、重みセレクタ５１０の出力は、望ましくは一様でなくて、以下で説明するように一定の重みに有利に偏っている。

【０１２１】

図２１に示されているように、重みセレクタ５１０は２つのプロセスＷＴ＿ＩＮＩＴ１９０５及びＷＴ＿ＣＡＬＣ１９１０、及び２つのテーブルＷＴ＿ＲＢＩＴＳ１９１５及びＷＴ＿ＤＩＳＴＲＩＢ１９２０を含む。プロセスＷＴ＿ＩＮＩＴ１９０５は、第１のキーが取り入れられるときにテーブルＷＴ＿ＤＩＳＴＲＩＢ１９２０を初期化するために１回だけ実施される。ＷＴ＿ＤＩＳＴＲＩＢ１９２０のデザインはシステムの重要な局面であ

り、後に詳しく検討される。プロセスＷＴ＿ＣＡＬＣ１９１０は、各呼び出し毎にキーＩに基づいて重みＷ（Ｉ）を作るために実施される。図２２の流れ図に示されているように、ＷＴ＿ＣＡＬＣ１９１０は、キーＩと、テーブルＷＴ＿ＲＢＩＴＳ１９１５に蓄積されているランダム・ビットとを使って乱数Ｔを作る（２００５）。次に、Ｔの値はテーブルＷＴ＿ＤＩＳＴＲＩＢ１９２０のロー番号Ｎを選択するために使われる。

【０１２２】

図２１に示されているように、ＷＴ＿ＤＩＳＴＲＩＢ１９２０のレンジ（ＲＡＮＧＥ）コラムのエントリーは値ＭＡＸ＿ＶＡＬで終わる正の整数の増加シーケンスであり、ＷＴコラムは値ＭＡＸ＿ＷＴで終わる正の整数の増加シーケンスである。Ｔの可能な値の集合は０とＭＡＸ＿ＶＡＬ－１との間の整数である。望ましい特性は、Ｔが、可能な値の範囲内のどの値にも一様な確率で等しくなりそうだという特性である。Ｎの値は、ＲＡＮＧＥ（Ｎ－１）≦Ｔ＜ＲＡＮＧＥ（Ｎ）を満たすＮが見出されるまでＲＡＮＧＥコラムを調べることにより決定される（２０１０）。Ｎが見出されると、Ｗ（Ｉ）の値は、テーブルＷＴ＿ＤＩＳＴＲＩＢのＷＴコラムのＮ番目のエントリーであるＷＴ（Ｎ）にセットされ、これは戻される重みである（２０１５，２０２０）。図２１で図示されているテーブル例についてＴが３８，５００に等しければ、Ｎは４であることが見出され、従ってＷ（Ｉ）はＷＴ（４）＝８にセットされる。好ましい実施態様では、ＷＴ＿ＤＩＳＴ１９２０のローは、Ｎが大きくなるときにＲＡＮＧＥ（Ｎ）－ＲＡＮＧＥ（Ｎ－１）の値が小さくなってゆくように編成される。これは、第１ローから連続探索を行うときにＴの値に対応する重みを見出すためにＷＴ＿ＤＩＳＴ１９２０を調べる平均探索時間を最小限度にする。他の実施態様ではおそらく他のロー編成が好ましく、バイナリーサーチ等の他の探索方法を使用することができる。

【０１２３】

（重み分布を選択する）

所与の符号化プロセスについて、重み分布はａ）なるべく少ない出力シンボル、ｂ）なるべく少ない演算、及びｃ）なるべく高い信頼度で入力ファイルを充分に復元できるように、選択されるべきである。通常、出力シンボルについての重み分布即ち全Ｉ上でのＷ（Ｉ）の分布と、出力シンボル上でのアソシエートの分布即ち全てのＩにわたるＡＬ（Ｉ）の所属関係とを正しく選択することによって、これら全ての最適化基準を満たすことができる。重み分布とアソシエートの選択に関する分布とに関わらずに復号プロセスを使用できるけれども、好ましい実施態様は近最適性能を得るために特に選択される重み分布及びアソシエートの選択に関する分布とを使用するということが強調されるべきである。実際、選択された分布を少し変化させても性能は僅かに変化するだけであろうから、多くの分布は良く働くであろう。

【０１２４】

１つの好ましい実施態様において分布を決定する１つの方法を説明する。使用される実際の重み分布は、入力シンボルの数Ｋに依存する。分布を、範囲（K_{min}，K_{max}）、因子β、及び相対的オーバーヘッドαと共に、以下に示す。このことは次のような意味を有する、即ち、$K_{min} \leqq K \leqq K_{max}$のＫが与えられると、冗長シンボルの数Ｒがβ＊Ｋより大きいか又は等しい最小の整数として計算され；集められる出力シンボルの数は少なくとも（１＋α）＊Ｋであるべきである、即ち、上記パラメータＡはα＊Ｋより大きいか又は等しい最小の整数である。アソシエータ５２０の第１バージョンが使用される場合には、Ｒは、更にＫ＋Ｒが素であるという条件、即ちＲがβ＊Ｋより大きいか又は等しい最小の素数であるという条件、を満たすべきである。もしＫ＋Ｒが素数であることをアプリケーションが必要としなければ、β＊Ｋより大きいか又は等しい最小の整数としてＲを選択することができる。

【０１２５】

分布自体は、

【０１２６】

【表１】

重み１	P1
重み２	P2
重み３	P3
……	……

の形式のテーブルとして与えられ、ここでＰ１は重み１の対応する確率であり、Ｐ２は重み２の対応する確率であり、ここでＰ１，Ｐ２，．．．の和は１である。このことは、図２１のテーブルＷＴ＿ＤＩＳＴＲＩＢ１９２０が

【０１２７】

【表２】

重み１	MAX_VAL*P1
重み２	MAX_VAL*(P1+P2)
重み３	MAX_VAL*(P1+P2+P3)
……	……

の形式を有することを意味する。

【０１２８】

本書中のテーブルを計算するときに使われる一般的指針について説明する。設計の１つの目標は、ゼロでない数の換算重み１の出力シンボルをダイナミック復号プロセスのなるべく深くに持つことである。ダイナミック復号の終了までずっとこの数がゼロより大きいことが好ましいであろう。しかし、このことは出力シンボルの平均重みが少なくとも入力シンボルの数Ｋの対数に比例するならば可能であるに過ぎないということが数学的解析により明らかにされており、ルビーＩに記載されている重み分布のうちの幾つかはこの様に設計されている。本発明の或る実施態様は、この平均重みをＫに依存しない一定数まで顕著に低減させる。その結果として、ダイナミック復号プロセス全体にわたって換算重み１の出力シンボルの数がゼロより大きいと期待することはできない。

【０１２９】

重み分布のデザインの最初のステップは、ダイナミック入力シンボルの部分ｘがまだ回復されていないときにダイナミック復号プロセス中に復号可能な集合の中に現在存する出力シンボルからその値を得ることのできるダイナミック入力シンボルの期待数についての式を得ることである。この式は、重み１，２，．．．，ｋの出力シンボルの部分Ｐ１，Ｐ２，．．．，Ｐｋと、集められた出力シンボルの数と入力シンボル及び冗長シンボルの数との比との関数である。以降は、この比をγで表示する。γ＝（１＋α）（１＋β）であることが分かる。この量を数学的に解析すると、その様なダイナミック入力シンボルの期待される数を

$$K*(1+\beta)*(x-e^{-(1+\gamma)*\omega(1-x)}) \qquad (1)$$

として表わせることが分かり、ここでｘは、ダイナミック入力シンボルの、ダイナミック復号プロセス中にまだ回復されていない部分を表わし、ω（ｘ）は多項式

$$P1+2*P2*x+3*P3*x^{2}+\ldots+k*Pk*x^{k-1} \qquad (2)$$

である。

この数は１つの量の期待値に過ぎず、これは本質的に統計的であるので、変化する。この変動を分析すると、それがまだ回復されていない入力シンボルの期待される数の平方根に、即ちｘ＊Ｋ＊（１＋β）の平方根に比例することが明らかとなる。換算重み１の出力シンボルに隣接する入力シンボルの数が常に正であることを程よく保証するためには、Ｐ１，．．．，Ｐｋ及びγは

$$K*(1+\beta)*(x-e^{-(1+\gamma)*\omega(1-x)})>c*sqrt(x*K*(1+\beta)) \quad (3)$$

となるように選択されるべきであり、この不等式は、与えられた正の実数εと１との間のｘのあらゆる任意の値に対して成り立つべきであり、ｃは１より大きい正の実数である。ｃが大きいほど、復号プロセスが成功するという保証が良好となる。εが小さいほど、ダイナミック復号プロセスの終了時に回復されないまま残っている入力シンボルが少なくなる。ω（１）が小さいほど、出力シンボルの平均重みが小さくなる。これらの束縛条件が与えられたとき、与えられたε及び与えられたｃについて、全ての係数が負ではなく、εと１との間のｘの全ての値について上記不等式を満たし、且つω（１）がなるべく小さい多項式ω（ｘ）を計算することができる。この最適化は、例えば、上記不等式を適宜操作した後にシンプレックス・アルゴリズムを用いるなどして、いろいろな方法で実行可能である。

【０１３０】

以上の記述に関連して、実際のテーブルを提示する。上の記述において定数ｃは、復号器全体についてエラー確率が10^{-10}より小さくなることを保証するように選択された。Ｋ＞４９２５１については、エラー確率は10^{-12}未満である。これらのテーブルは、単に使用可能な重み分布の例として提供されているに過ぎない。他の重み分布を使用することもできることが理解されるべきである。

【０１３１】

（テーブル１）

Ｋの範囲：９９００－１４８００、β＝０．００８１、α＝０．１１８７

【０１３２】

【表３】

1	0.018235
2	0.477562
3	0.153565
4	0.102006
5	0.034651
7	0.048352
8	0.06084
18	0.058325
19	0.008401
70	0.008451
71	0.029613

（テーブル２）

Ｋの範囲：１４８０１－１９６８０、β＝０．０１２１、α＝０．０８４

【０１３３】

【表４】

1	0.019314
2	0.483582
3	0.160754
4	0.081631
5	0.067541
8	0.094528
18	0.041968
19	0.019462
66	0.007987
67	0.023233

（テーブル３）
Ｋの範囲：１９６８１－２９５１０、β＝０．０１５１、α＝０．０７６９

【０１３４】

【表５】

1	0.013531
2	0.488250
3	0.164810
4	0.070953
5	0.084243
8	0.050093
9	0.042547
19	0.055060
62	0.00501988
63	0.025491

（テーブル４）
Ｋの範囲：２９５１１－４９２５０、β＝０．０１６１、α＝０．０６７４

【０１３５】

【表６】

1	0.013876
2	0.489087
3	0.162276
4	0.081638
5	0.069880
8	0.081339
9	0.014424
18	0.017712
19	0.040774
66	0.014680
67	0.014314

（テーブル５）
Ｋの範囲：４９２５１－６４７８０、β＝０．０１５、α＝０．０５５８

【０１３６】

【表７】

1	0.009117
2	0.492843
3	0.165983
4	0.072707
5	0.082303
8	0.056347
9	0.036917
19	0.055616
65	0.022195
66	0.005972

（テーブル６）
Ｋの範囲：６４７８１－７９０８０、β＝０．０１１４、α＝０．０５

【０１３７】

【表８】

1	0.007969
2	0.493570
3	0.166220
4	0.072646
5	0.082558
8	0.056058
9	0.037229
19	0.055590
65	0.025023
66	0.003135

（テーブル７）
Ｋの範囲：７９０８１－９８６２３、β＝０．０１１３４、α＝０．０４７
【０１３８】
【表９】

1	0.007544
2	0.49361
3	0.166458
4	0.071243
5	0.084913
8	0.049633
9	0.043365
19	0.045231
20	0.010157
66	0.010479
67	0.017365

（テーブル８）
Ｋの範囲：９８６２４－１１８３４９、β＝０．０１３７７、α＝０．０４２４
【０１３９】

【表１０】

1	0.006495
2	0.495044
3	0.168010
4	0.067900
5	0.089209
8	0.041731
9	0.050162
19	0.038837
20	0.015537
66	0.016298
67	0.010777

（テーブル９）

Ｋの範囲：１１８３５０－∞、β＝０．０１５７９、α＝０．０３９３

【０１４０】

【表１１】

1	0.004807
2	0.496472
3	0.166912
4	0.073374
5	0.082206
8	0.057471
9	0.035951
18	0.001167
19	0.054305
65	0.018235
66	0.009100

例えば、Ｋ＝３３，０００ならば、冗長シンボルの数は、Ｋ＋Ｒが素数となるようにＫ＊０．０１６１＝５３１．３より大きな最小の整数Ｒであって良い。即ち、Ｒ＝５３３である。集められる出力シンボルの数は少なくとも（１＋０．０６７４）＊Ｋであるべきであり、これは３５２２５である。

【０１４１】

テーブル１についての平均重みはおよそ６．７５であり、テーブル２から９までについての平均重みはおよそ６である。前述したルビーＩの１実施態様ではＫが６０，０００であるときに平均重みは２８．６８であり、これと比べるとこれらの平均重みはかなり低い。

【０１４２】

（少数の入力シンボルについての符号化）
どんなサイズの入力ファイルについても低い相対的オーバーヘッドが好ましい。上のテーブルから分かるように、入力シンボルの数Ｋが小さくなると相対的オーバーヘッドαは大きくなる。例えば、入力ファイルが１０，０００バイトを有し、各シンボルが１バイトからなるようにＫが１０，０００であるように選択されると、オーバーヘッドはおよそ１１％であり、これは或るアプリケーションのためには望ましくない。オーバーヘッドを減少させる１つの方法は、入力シンボルのサイズを小さくするという犠牲を払って入力シンボルの数Ｋを増やすことであろう。例えば、約７％のオーバーヘッドが望まれる場合、Ｋを４０，０００であるように選択することができる。この場合、入力シンボルのサイズは２ビットとなる。しかし、非常に小さなサイズの入力シンボルを使うと実際上は計算に欠陥が生じることになる。

【０１４３】
この問題の解決策は、図１７に関して説明した実施態様などの、ガウス消去法に基づく復号器を使用することである。この復号器は、他の実施態様に関して前述したスタティック符号器のうちのいずれかと組み合わされたチェーンリアクション復号器のように計算に関して効率的ではないけれども、この復号器については故障確率が非常に小さいこととオーバーヘッドとを考慮すると、この解決策は一定のアプリケーションに対しては望ましいであろう。

【０１４４】
具体的には、入力シンボルの数Ｋが８００と９８９９との間にあるとき、ダイナミック符号器２２０のために下記の重み分布を使用することができる：
（テーブル１０）
Ｋの範囲：８００－１６００、β＝０．０８、α＝０．０５

【０１４５】
【表１２】

2	0.39
3	0.095
4	0.095
5	0.095
10	0.095
19	0.095
30	0.095
130	0.04

図１０に関して説明したスタティック符号器などのスタティック符号器の動作に従って０．０８＊Ｋ個の冗長シンボルが作られることになる。復号は、図１７に関して説明した復号器等の復号器を用いて実行される。

【０１４６】
例として、入力ファイルが１６，０００バイトのサイズを有すると仮定する。入力シンボルは各々１６バイトからなるように選択され、入力シンボルの数Ｋは１０００となる。図１０のスタティック符号器が使用されて８０個の冗長シンボルを作る。次に、ダイナミック符号器２２０が上記の重み分布と共に使用されて出力シンボルを作る。受信装置は、（１＋α）＊Ｋ＝１０５０個の出力シンボルを集め、これらを図１７の復号器に提供する。ダイナミック・マトリックス発生器１３０５は、１０５０ｘ１０８０のフォーマットのマトリックスＣを作る。スタティック・マトリックス発生器１３３０は１１３０ｘ１０８０のフォーマットのマトリックスＭを作り、これを線型方程式系ソルバー１３４０に渡し

、これは、元の１０００個の入力シンボル即ち入力ファイルを復号しようと試みる。

【０１４７】

（或る多段符号の或る特性）

上記の例の殆どにおいて、入力シンボル及び出力シンボルは或る数のビットのために符号化を行い、各出力シンボルは１つのパケット内に置かれる（パケットは、その全体が受信されるか或いは全体が失われるトランスポートの単位である）。或る実施態様では、通信システムは、各パケットが数個の出力シンボルを含むように改造される。出力シンボル値のサイズは、幾つかの要素に基づいて、最初にファイルを入力シンボルに分割するときに入力シンボル値のサイズにより決定されるサイズにセットされる。復号プロセスは、各パケットが受信されるときに出力シンボルが束になって到着することを除いて、実質的に変更されないままである。

【０１４８】

入力シンボル及び出力シンボルのサイズの設定は、普通は、ファイルのサイズと、出力シンボルが伝送されることになる通信システムとにより規定される。例えば、通信システムがデータのビットをまとめて決まったサイズのパケットにするか或いは他の方法でビットをまとめるならば、シンボルのサイズの設計はパケットのサイズ或いはグループ化サイズから始まる。そこから、設計者は１つのパケット又はグループで何個の出力シンボルが運ばれるか決定し、それが出力シンボルのサイズを決定する。平易にするために、設計者はおそらく入力シンボルのサイズを出力シンボルのサイズに等しく設定するであろうが、もし入力データが異なる入力シンボルのサイズをより好都合にするならば、それを使うことができる。

【０１４９】

上記の符号化プロセスは、元のファイルに基づく出力シンボルを含むパケットのストリームを作る。そのストリーム中の各出力シンボルは他の全ての出力シンボルに依存せずに作られ、作ることのできる出力シンボルの数には上限も下限も無い。各出力シンボルにキーが関連付けられる。そのキーと、入力ファイルの或る内容とが出力シンボルの値を決定する。連続的に作られる出力シンボルが連続的キーを持つ必要はなく、或るアプリケーションではキーのシーケンスをランダムに作ること、或いは該シーケンスを擬似ランダムに作ることが好ましいであろう。

【０１５０】

多段復号化は、もし元のファイルをＫ個の等サイズの入力シンボルに分割でき、各出力シンボル値が入力シンボル値と同じ長さであるならば、平均でＫ＋Ａ個の出力シンボルから非常に高い確率でファイルを回復することができるという特性を有し、ここでＡはＫと比べると小さい。例えば、上で導入した重み分布については、Ａの値がα＊Ｋを上回る確率は、もしＫが１９，６８１より大きければせいぜい10^{-12}であり、またＫのどんな値についてもせいぜい10^{-10}である。特定の出力シンボルはランダム又は擬似ランダムな順序で作られ、特定の出力シンボルの転送中の紛失はランダムであると仮定されるので、入力ファイルを回復するために必要な出力シンボルの実際の数には小さな分散が存在する。Ｋ＋Ａ個のパケットの特定のコレクションが入力ファイル全体を復号するには充分でない場合、受信装置が出力パケットの１つ以上のソースからもっと多くのパケットを集めることができるならば、入力ファイルは依然として回復可能である。

【０１５１】

出力シンボルの数はＩの解像度のみにより制限されるので、Ｋ＋Ａ個より充分に多くの出力シンボルを作ることができる。例えば、もしＩが３２ビットの数であれば、４０億個の異なる出力シンボルを作ることができ、ファイルはＫ＝５０，０００個の入力シンボルを含むことができる。或るアプリケーションでは、これら４０億個の出力シンボルのうちの少数の出力シンボルを作って送信することができ、また、可能な出力シンボルのうちの非常に小さな一部分で、また（入力シンボルのサイズが出力シンボルのサイズと同じであると仮定して）Ｋ個より僅かに多い出力シンボルで入力ファイルを回復できるという優れた確率で、入力ファイルを回復できるということは殆ど確実である。

【０１５２】
或るアプリケーションでは、入力シンボルの全部を復号することはできないこと、或いは割合に低い確率でではあるけれども入力シンボルの全部を復号することができるということは容認可能であろう。その様なアプリケーションでは、受信装置は、Ｋ＋Ａ個の出力シンボルを受け取った後に入力シンボルの全部を復号しようと試みるのを止めて良い。或いは、受信装置は、Ｋ＋Ａ個未満の出力シンボルを受け取った後に出力シンボルを受け取るのを止めて良い。或るアプリケーションでは、受信装置はＫ個又はそれ未満の出力シンボルを受け取るだけで良い。従って、本発明の或る実施態様では所望の精度が入力シンボル全部の完全な回復でなくても良いということが理解されよう。
【０１５３】
更に、不完全な回復が容認される或るアプリケーションでは、入力シンボルの全部は回復できない様に、或いは入力シンボルを完全に回復するためには入力シンボルの数より遥かに多数の出力シンボルを受信する必要があるように、データを符号化することができる。この様な符号化は、一般に、必要な計算費用が少ないので、符号化の計算費用を減らす１つの容認可能な方法であろう。
【０１５４】
上記の図中のいろいろな機能ブロックをハードウェア及び／又はソフトウェアの組み合わせにより具体化できること、及び、特定の実施態様において該ブロックのうちのあるものの機能の一部又は全部を組み合わせることができるということが理解されよう。同様に、本書に記載したいろいろな方法をハードウェア及び／又はソフトウェアの組み合わせによって実施できることも理解されよう。
【０１５５】
以上の記述は説明であって限定をするものではない。当業者にとっては、この開示を検討すれば、本発明のいろいろなバリエーションが明白になろう。従って、本発明の範囲は、以上の記述に関連して決定されるべきではなくて、添付されている請求項とその同等物の範囲全体に関連して決定されるべきである。
【図面の簡単な説明】
【０１５６】
【図１】図１は、本発明の一実施態様に従う通信システムのブロック図である。
【図２】図２は、本発明の一実施態様に従う符号器のブロック図である。
【図３】図３は、本発明の一実施態様に従って冗長シンボルを作る方法の略ブロック図である。
【図４】図４は、本発明の一実施態様に従うスタティック符号器の基本動作の略ブロック図である。
【図５】図５は、本発明の一実施態様に従うダイナミック符号器の略ブロック図である。
【図６】図６は、本発明の一実施態様に従うダイナミック符号器の基本動作の略ブロック図である。
【図７】図７は、本発明の一実施態様に従うスタティック符号器の略ブロック図である。
【図８】図８は、本発明の一実施態様に従うスタティック符号器の基本動作の略ブロック図である。
【図９】図９は、スタティック符号器の１つの特定の実施態様に従って符号化パラメータを計算する方法の略図である。
【図１０】図１０は、本発明のもう一つの実施態様に従うスタティック符号器の略流れ図である。
【図１１】図１１は、本発明の一実施態様に従う復号器の略ブロック図である。
【図１２】図１２は、本発明の一実施態様に従う復号器の動作の略流れ図である。
【図１３】図１３は、本発明のもう一つの実施態様に従う復号器の動作の略流れ図である。
【図１４】図１４は、本発明の更にもう一つの実施態様に従う復号器の動作の略流れ図である。

【図１５】図１５は、本発明の一実施態様に従うダイナミック復号器の略ブロック図である。
【図１６】図１６は、本発明の一実施態様に従うスタティック復号器の略ブロック図である。
【図１７】図１７は、本発明のもう一つの実施態様に従うスタティック復号器の略ブロック図である。
【図１８】図１８は、本発明の実施態様に従う復号器の動作の略流れ図である。
【図１９】図１９は、本発明の実施態様に従う復号器のもう一つの動作の略流れ図である。
【図２０】図２０は、本発明の一実施態様に従うアソシエータの略流れ図である。
【図２１】図２１は、本発明の１つの特定の実施態様に従う重みセレクタの略ブロック図である。
【図２２】図２２は、本発明の実施態様に従う重みセレクタにより使用され得るプロセスの略流れ図である。

【図１】

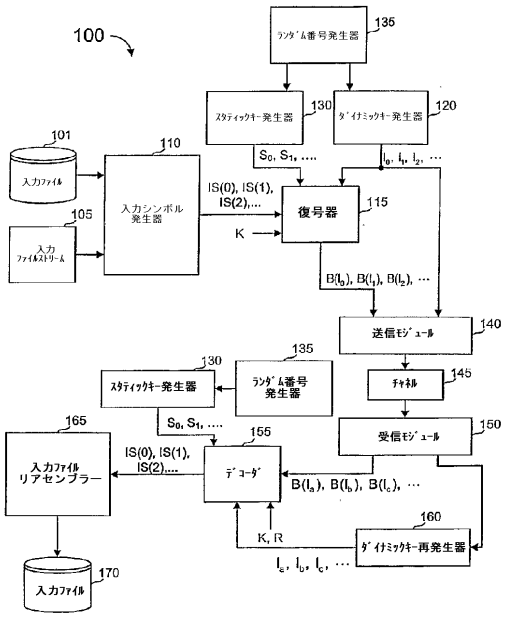

Figure 1

【図２】

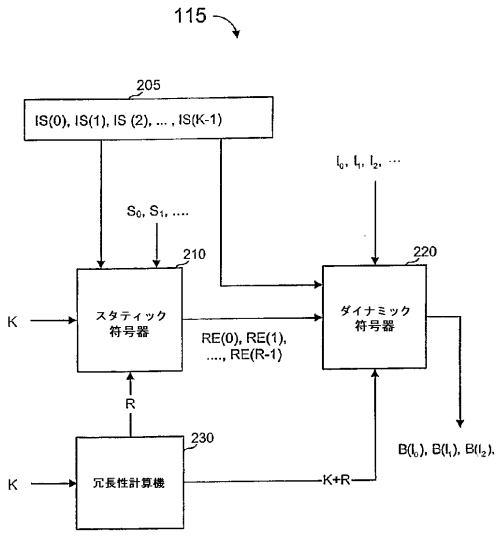

Figure 2

【図３】

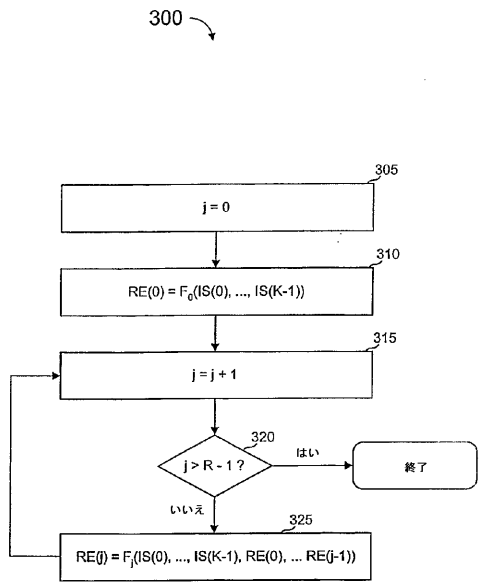

Figure 3

【図４】

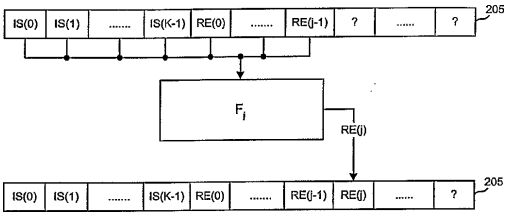

Figure 4

【図５】

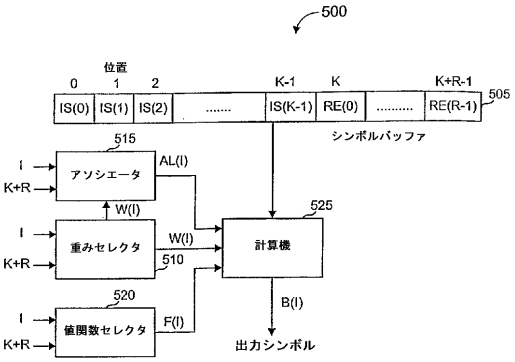

Figure 5

【図６】

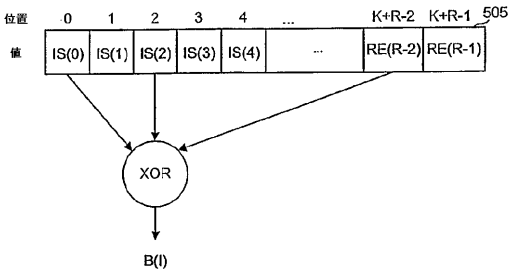

Figure 6

【図７】

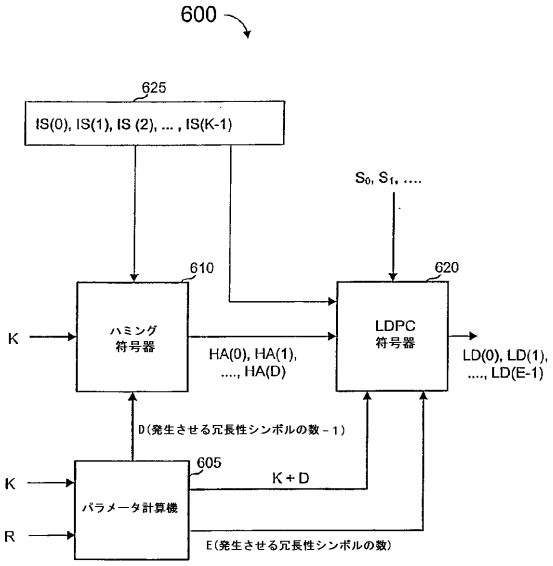

Figure 7

【図８】

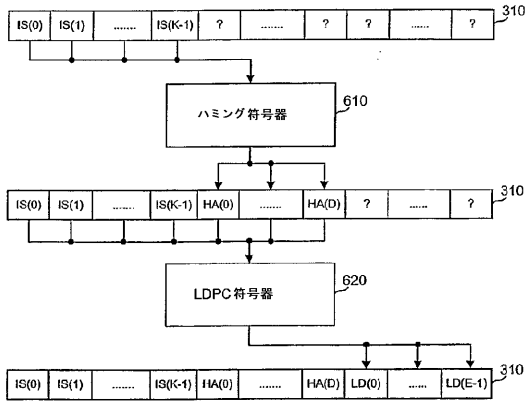

Figure 8

【図９】

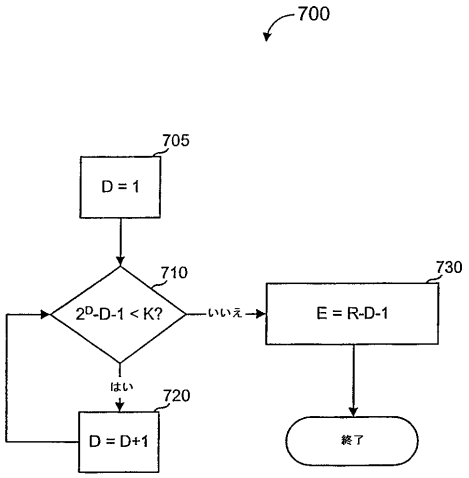

Figure 9

【図１０】

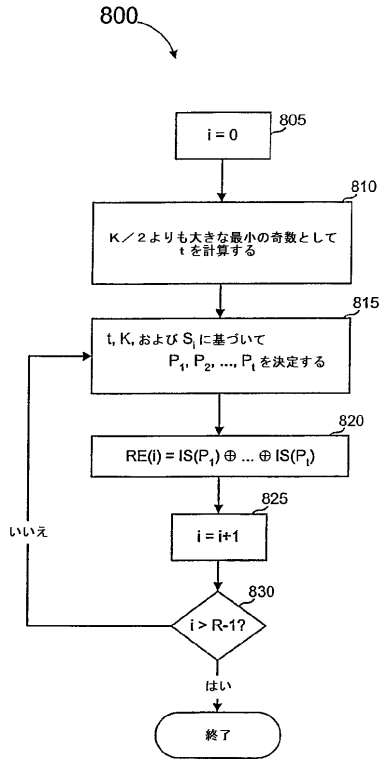

Figure 10

【図１１】

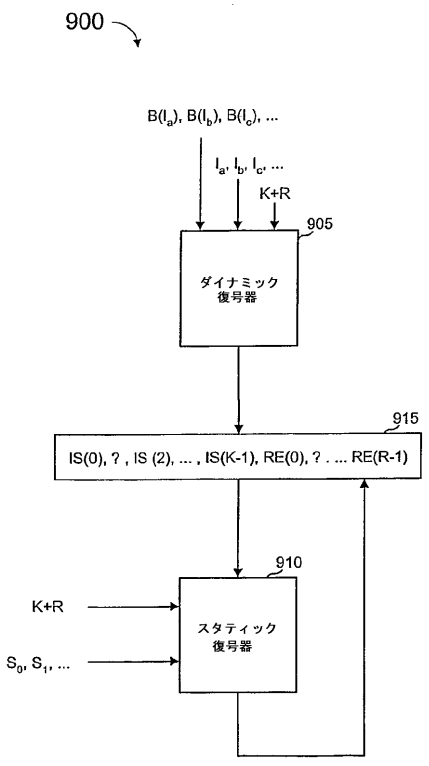

Figure 11

【図１２】

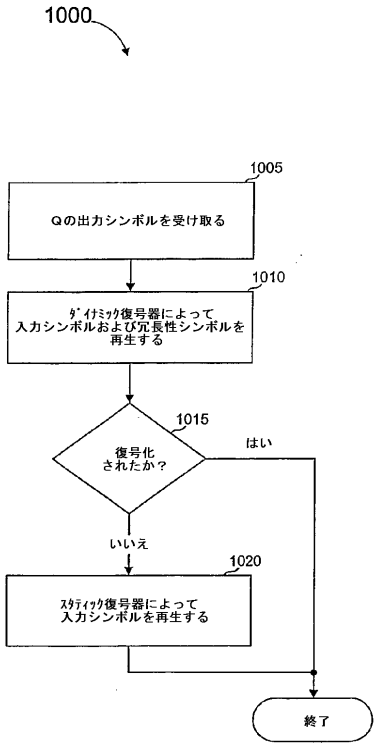

Figure 12

【図１３】

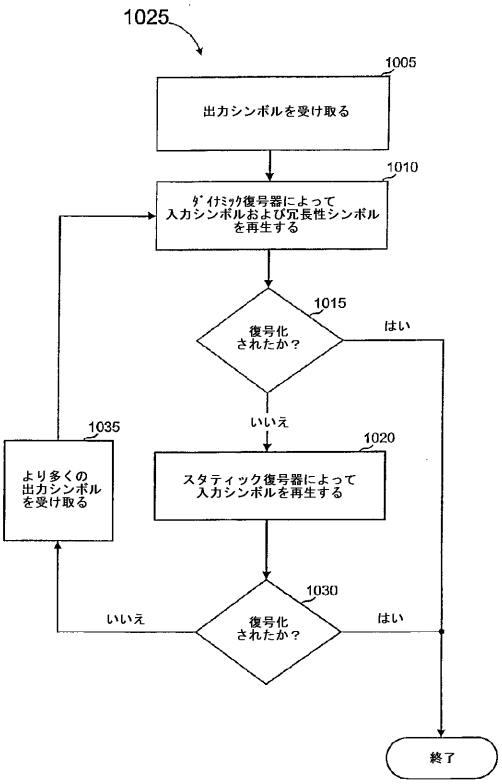

Figure 13

【図１４】

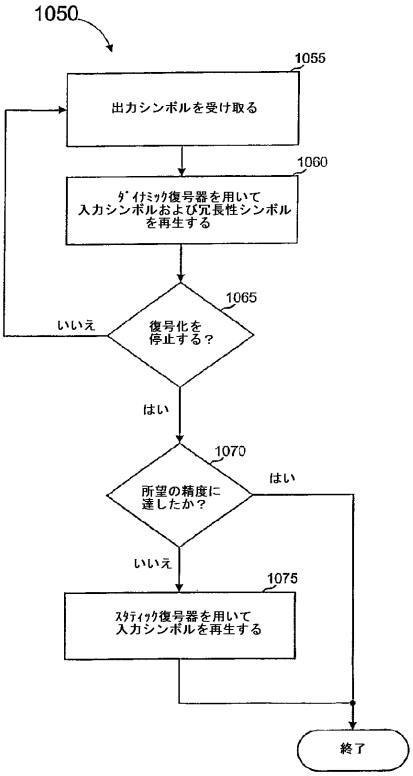

Figure 14

【図１５】

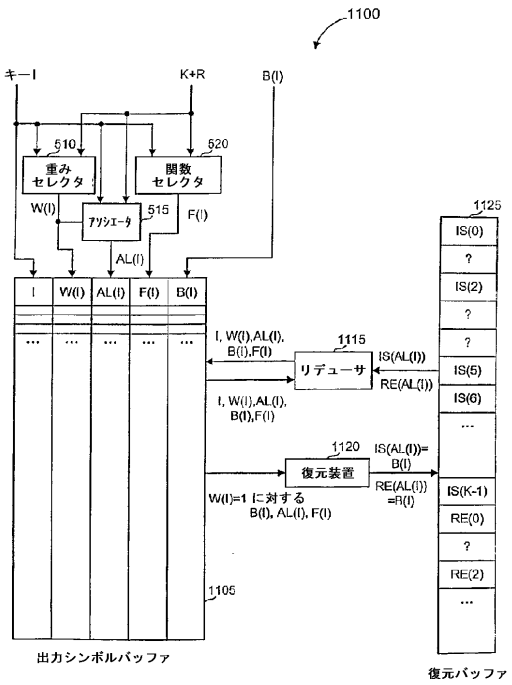

Figure 15

【図１６】

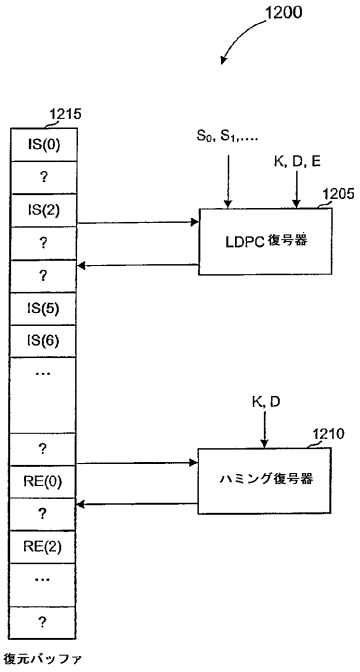

Figure 16

【図１７】

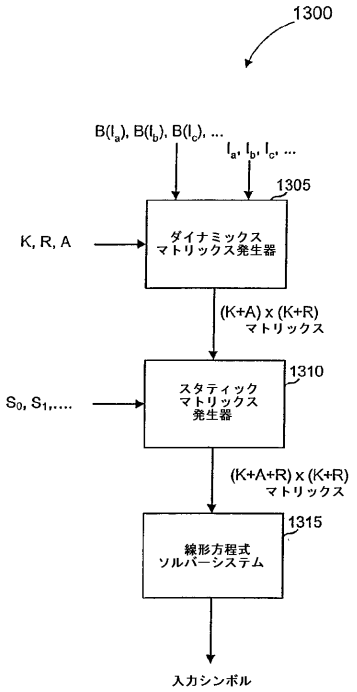

Figure 17

【図１８】

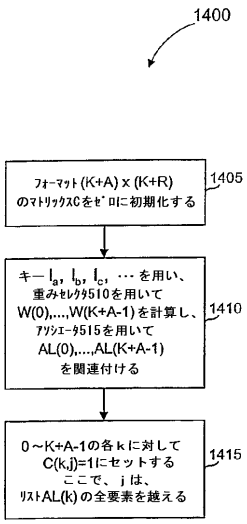

Figure 18

【図１９】

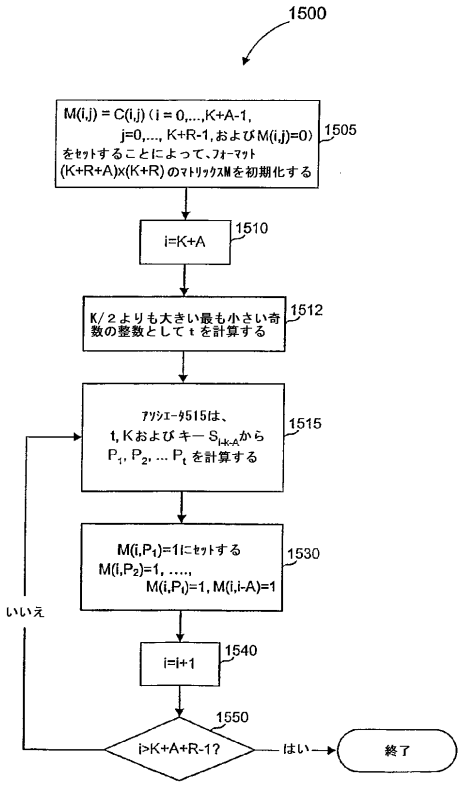

Figure 19

【図２０】

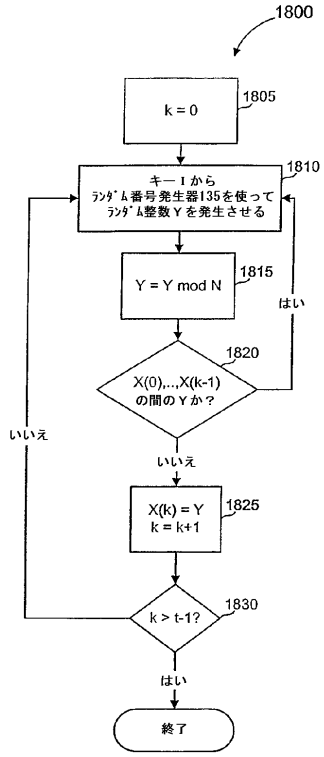

Figure 20

【図２１】

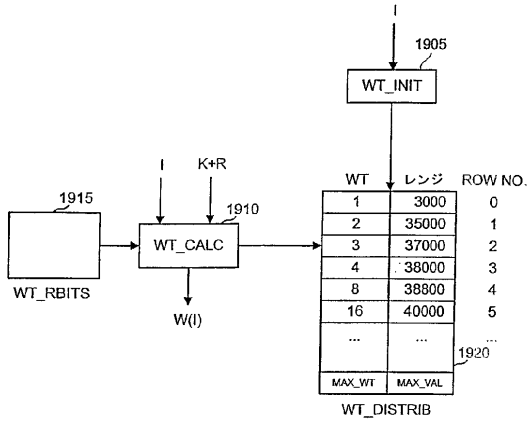

Figure 21

【図２２】

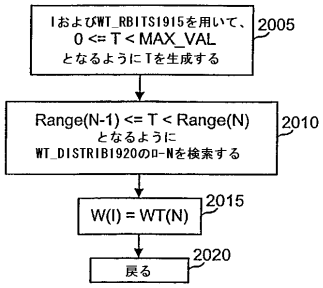

Figure 22

フロントページの続き

(74)代理人 100130409
弁理士 下山 治
(74)代理人 100134474
弁理士 坂田 恭弘
(74)代理人 100078282
弁理士 山本 秀策
(74)代理人 100062409
弁理士 安村 高明
(74)代理人 100113413
弁理士 森下 夏樹
(72)発明者 ショクロラヒ, アミン エム.
アメリカ合衆国 カリフォルニア ９４７０８, バークレイ, リーガル ロード ９０４
(72)発明者 ラッセン, ソレン
アメリカ合衆国 カリフォルニア ９４１１０, サン フランシスコ, バーレット ストリート ２３３
(72)発明者 ルビー, マイケル
アメリカ合衆国 カリフォルニア ９４７０８, バークレイ, ミラー アベニュー １１３３

審査官 矢頭 尚之

(56)参考文献 特開２００１－２２３６５５（ＪＰ，Ａ）

(58)調査した分野(Int.Cl., ＤＢ名)
H03M 13/19