

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3947510号
(P3947510)

(45) 発行日 平成19年7月25日(2007.7.25)

(24) 登録日 平成19年4月20日(2007.4.20)

(51) Int. Cl.		F I		
HO3M 7/14	(2006.01)	HO3M 7/14		B
HO4L 1/00	(2006.01)	HO4L 1/00		F
HO4L 25/49	(2006.01)	HO4L 25/49		A

請求項の数 8 (全 25 頁)

(21) 出願番号	特願2003-336173 (P2003-336173)	(73) 特許権者	390019839
(22) 出願日	平成15年9月26日(2003.9.26)		三星電子株式会社
(65) 公開番号	特開2004-120763 (P2004-120763A)		Samsung Electronics
(43) 公開日	平成16年4月15日(2004.4.15)		Co., Ltd.
審査請求日	平成15年9月26日(2003.9.26)		大韓民国京畿道水原市八達区梅灘洞416番地
(31) 優先権主張番号	2002-058809	(74) 代理人	100070150
(32) 優先日	平成14年9月27日(2002.9.27)		弁理士 伊東 忠彦
(33) 優先権主張国	韓国 (KR)	(74) 代理人	100091214
(31) 優先権主張番号	2002-063534		弁理士 大貫 進介
(32) 優先日	平成14年10月17日(2002.10.17)	(74) 代理人	100107766
(33) 優先権主張国	韓国 (KR)		弁理士 伊東 忠重
		(72) 発明者	沈 載 晟
			大韓民国漢城市広津区紫陽1洞610-35番地

最終頁に続く

(54) 【発明の名称】 データ変調方法、データ復調方法及びコード配置方法

(57) 【特許請求の範囲】

【請求項1】

mビットのソースデータを最小連長制限 $d = 1$ 、最大連長制限 $k = 7$ に制限しつつ n ビット ($n = m \cdot d \cdot k$ 、n 及び m は正の整数) のコードワードを配列するコード配置方法において、

コードワード a はコードワード b に連結され、先行するコードワードであり、コードワード b はコードワード b 1 とコードワード b 2 から選択され、コードワード a とコードワード b 1 とが連結されるコード列をコード列 X 1、コードワード a とコードワード b 2 とが連結されるコード列をコード列 X 2 とする時、コードワード内のビット " 1 " の数が奇数かまたは偶数かによって次のコードワードの遷移を予測する INV パラメータが反対値を有するように、コードワード b 1 及び b 2 を配置する段階と、

コードワード a とコードワード b 1 または b 2 とが連結される時、境界規則によりコードワード a、コードワード b 1 または b 2 が他のコードワードに変換されるとしても、INV パラメータの特性を維持するように、コード列 X 1 及び X 2 を配置する段階と、を含むコード配置方法。

【請求項2】

コードワード a とコードワード b 1、b 2 間のビット 0 の数が $d (= 1)$ より小さい場合、境界規則においてコードワード a またはコードワード b 1、b 2 を変形することにより、変形されたコードワード間のビット " 0 " の数が $d (= 1)$ より大きいかまたは同一に、 $k (= 7)$ より小さいかまたは同一に配置されたことを特徴とする請求項 1 に記載の

10

20

コード配置方法。

【請求項 3】

前記境界規則により前記コード列 X 1 内のコードワード a 及びコード列 X 2 内のコードワード a が他のコードワードに変換されるが、それぞれのコード列で変換されたコードワードは同じ I N V 値を有するようにしてコードワード b 1 及び b 2 の I N V により結局それぞれのコード列 X 1 及び X 2 は相異なる I N V 値を有するように配置されたことを特徴とする請求項 1 に記載のコード配置方法。

【請求項 4】

m ビットのソースデータを最小連長制限 $d = 1$ 、最大連長制限 $k = 7$ に制限しつつ n ビット ($n \leq m$ 、 d 、 k 、 n 及び m は正の整数) のコードワードを配列するコード配置方法

10

において、
コードワード b はコードワード c に連結され、先行するコードワードであり、コードワード b はコードワード b 1 とコードワード b 2 から選択され、コードワード b 1 とコードワード c とが連結されるコード列をコード列 Y 1、コードワード b 2 とコードワード c とが連結されるコード列をコード列 Y 2 とする時、コードワード内のビット “ 1 ” の数が奇数かまたは偶数かによって次のコードワードの遷移を予測する I N V パラメータが反対値を有するように、コードワード b 1 及び b 2 を配置する段階と、

コードワード b 1 または b 2 がコードワード c と連結される時、境界規則によりコードワード b 1、b 2 またはコードワード c が他のコードワードに変換されるとしても、I N V パラメータの特性を維持するように、コード列 Y 1 及び Y 2 を配置する段階と、を含む

20

【請求項 5】

コンピュータで判読可能な媒体であって、

$n \leq m$ であり、 d 、 k 、 n 及び m は正の整数である時、m ビットのソースデータを n ビットのコードワードに変調するデータ変調方法の動作を行うコンピュータで判読可能な命令語を含み、ここで、前記コードワードは最小連長制限 d が 1 に制限され、最大連長制限 k は 7 に制限され、

前記変調方法の動作は、

コードワード a はコードワード b に連結され、先行するコードワードであり、前記コードワード b はコードワード b 1 とコードワード b 2 から選択され、前記コードワード a とコードワード b 1 とが連結されるコード列をコード列 X 1、コードワード a とコードワード b 2 とが連結されるコード列をコード列 X 2 とする時、1 つのコードワード内のビット “ 1 ” の数が奇数かまたは偶数かによって次のコードワードの遷移を予測する正反対の I N V パラメータを有するように前記コードワード b 1 及び b 2 を配列する段階と、

30

前記コードワード a とコードワード b 1 または b 2 とが連結される時、境界規則により前記コードワード a、コードワード b 1 または b 2 が他のコードワードに変換されるとしても前記コード列 X 1 及び X 2 を正反対の I N V パラメータを有するように配列する段階と、を含むことを特徴とするコンピュータで判読可能な媒体。

【請求項 6】

コンピュータで判読可能な媒体であって、

40

$n \leq m$ であり、 d 、 k 、 n 及び m は正の整数である時、m ビットのソースデータを n ビットのコードワードに変調するデータ変調方法の動作を行うコンピュータで判読可能な命令語を含み、ここで、前記コードワードは最小連長制限 d が 1 に制限され、最大連長制限 k は 7 に制限され、

前記変調方法の動作は、

コードワード b はコードワード c に連結され、先行するコードワードであり、前記コードワード b はコードワード b 1 とコードワード b 2 から選択され、前記コードワード b とコードワード c とが連結されるコード列をコード列 Y 1、コードワード b 2 とコードワード c とが連結されるコード列をコード列 Y 2 とする時、1 つのコードワード内のビット “ 1 ” の数が奇数かまたは偶数かによって次のコードワードの遷移を予測する正反対の I N

50

Vパラメータを有するように前記コードワードb 1及びb 2を配列する段階と、

前記コードワードcとコードワードb 1またはb 2とが連結される時、境界規則により前記コードワードa、コードワードb 1またはb 2が他のコードワードに変換されるとしても前記コード列Y 1及びY 2を正反対のINVパラメータを有するように配列する段階と、を含むことを特徴とするコンピュータで判読可能な媒体。

【請求項7】

ソースデータを最小連長制限がd、最大連長制限がk（ただし、dとkは正の整数）であるコードワードに変調するステップからなるデータ変調方法であって、前記変調ステップは、コードワードaのEnd ZeroとコードワードbのLead Zeroの和が前記最小連長制限より小さいか、あるいは前記最大連長制限より大きい場合、前記コードワードaのEnd Zeroと前記コードワードbのLead Zeroの前記和が前記最小連長制限以上かつ、前記最大連長制限以下となるよう前記コードワードaを他のコードワードに変えることからなり、前記コードワードbは前記コードワードaに接続され、前記コードワードaは先行するコードワードであり、前記End Zeroは前記コードワードaのLSBからMSBまでの連続する0の個数であり、前記Lead Zeroは前記コードワードbのMSBからLSBまでの連続する0の個数であることを特徴とするデータ変調方法。

10

【請求項8】

請求項7記載のデータ変調方法であって、前記最小連長制限dは1、前記最大連長制限kは7であり、前記コードワードaのEnd Zeroが0、かつ前記コードワードbのLead Zeroが0である場合、前記エンコーダは前記コードワードaをLSBが0であるようなコードワードに変えることを特徴とするデータ変調方法。

20

【発明の詳細な説明】

【技術分野】

【0001】

本発明はデータ変復調分野に係り、特にエラー伝播確率を減少させ、かつ高いコード効率及びDC抑圧能力を有する変調コードを提供するデータ変調方法、データ復調方法及びコード配置方法に関する。

【背景技術】

30

【0002】

マルチコーディング方式はDC抑圧能力がない変調コードにDC抑圧能力を付与するための方法である。これは入力データ列にaビットの付加情報を挿入し、この付加情報によって 2^a 個の他のランダムデータ列を作ってその 2^a 個のランダムデータ列にDC抑圧能力がない変調を行っても、そのうち最もDC成分が少ない変調されたデータ列を選択することによりDC抑圧能力を有させる方法である。

【0003】

従来には特許文献1の“Device for encoding/decoding n-bit source words into corresponding m-bit channel words, and vice versa”で例示したd = 1、k = 7、m = 2、n = 3のコードのコード率Rは約2%のリダンダンシーが含まれれば $R = 49 / 75 = 0.6533$ であり、コード効率 $R / C(d, k)$ は $R / C(d, k) = 0.6533 / 0.6793 = 96.2\%$ である。特許文献1で使われた変調コードを説明の便宜上以下“ACode”と称する。

40

【0004】

特許文献2の“Method of allocating RLL code having enhanced DC suppression capability, modulation method, demodulation method, and demodulation apparatus therefor”で例示したd = 1、k = 8、m = 8、n = 12のコードのコード率Rは約2%のリダンダンシーが含ま

50

れれば $R = 32 / 49 = 0.6531$ であり、コード効率 $R / C(d, k)$ は $R / C(d, k) = 0.6531 / 0.6853 = 95.3\%$ である。特許文献 2 で使われた変調コードを説明の便宜上、以下 “B - Code” と称する。C はコードの d 及び k による容量を意味する。

【0005】

また、非特許文献 1 の 13 章に説明した Guided Scrambling 方法を使用してデータ 25 バイト毎にリダンダンシー 4 ビットを挿入して $d = 1$ 、 $k = 7$ 変調を行う場合、コード率 R は $R = 200 / 306 = 0.6536$ であり、コード効率 $R / C(d, k)$ は $R / C(d, k) = 0.6536 / 0.6793 = 96.2\%$ である。前記文献で使われた変調コードを説明の便宜上、以下 “C - Code” と称する。

10

【0006】

上記 3 つの従来技術の変調方法によるコード効率は $95.3\% \sim 96.2\%$ と類似しており、これらコードの DC 抑圧能力を表すパワースペクトル密度 (以下、PSD) 曲線は図 1 に示された通りである。

【0007】

しかし、前述した文献で例示したマルチモードコーディング方式も、十分な DC 抑圧能力を有するためには、データ列をランダムデータにするための付加情報の頻度数をそれだけ高めなければならない。また、高いコード効率を有する変調技法を開発しても DC 抑圧能力が十分でない場合もある。その例として、前述した特許文献 2 に開示された B - Code もリダンダンシーがなければ DC 抑圧は可能であるが、別途の付加ビットなしでは満足すべき DC 抑圧性能を有することができない。以下、リダンダンシーがなければ DC 抑圧は可能であるが、別途の付加ビットがなくかつ抑圧性能は落ちるコードを不十分な DC 抑圧変調コードと称する。

20

【0008】

図 2 は、入力データ列をランダムデータに変換する従来の多重化方法を説明するための図面であって、a ビットの付加情報を利用して入力データ列を 2^a 個の他のランダムデータ列にする方法の一例は特許文献 3 に開示されているが、従来には入力データに対して連続的にデータスクランブルを行っていた。

【0009】

図 2 において、所定ビット数のデータ $x_{i,0} \sim x_{i,u-1}$ (符号変調単位と称する) で構成された入力データ列

30

【0010】

【数 1】

$$\underline{y}_i$$

40

は多重化情報 s_t 及び $x_{i,0} \sim x_{i,u-1}$ に各々対応した排他的論理和素子による排他的論理和演算を通じてランダムデータ列

【0011】

【数 2】

$$f(\underline{y}_i / s_t)$$

50

に変換される。

【 0 0 1 2 】

すなわち、先頭の符号変調単位 $x_{i,0}$ 及び初期データ（多重化情報） s_t は、排他的論理和素子を利用して排他的論理和演算することにより、初期データを除外した先頭の符号変調単位の変換データ $y_{i,0}^t$ を生成する。次いで、前述した変換が終わった符号変調単位の変換データ $y_{i,0}^t$ 及び次の符号変調単位 $x_{i,1}$ は排他的論理和演算により次の変換データ $y_{i,1}^t$ を同じく生成する。以下、同様に、前記入力データ列

【 0 0 1 3 】

【 数 3 】

y_i

10

の最終の符号変調単位（ここでは、 $x_{i,u-1}$ ）まで、直前の符号変調単位の変換データと変換する符号変調単位との排他的論理和演算処理が反復される。

20

【 0 0 1 4 】

図 3 は、図 2 に示された多重化方法で入力データ列をランダムデータに変換した後、連長制限（Run Length Limited；以下、RLL）変調を行った結果を保存媒体に記録した後、再生する時に RLL ストリーム

【 0 0 1 5 】

【 数 4 】

$f'(y_i/s_i)$

30

を逆変換を通じて復元したストリーム

【 0 0 1 6 】

【 数 5 】

y_i

40

を示した図である。

【 0 0 1 7 】

データ逆変換時には逆変換前 RLL ストリーム

【 0 0 1 8 】

【数6】

$$f'(y_i/s_i)$$

の先頭の復調符号単位（多重化情報 s_t ）を除外した復調符号単位から順番に、逆変換対象である復調符号単位と前記復調符号単位直前の復調符号単位（初期データまたは逆変換前の復調符号単位）との排他的論理和演算を通じて復元されたストリーム 10

【0019】

【数7】

$$y_i$$

20

が生成される。

【0020】

すなわち、先頭の復調符号単位 $y_{i,0}^t$ と初期データ（多重化情報） s_t との排他的論理和演算により逆変換後に逆変換されたデータ $x_{i,0}$ が生成される。次いで、前記 $y_{i,0}^t$ （逆変換前の復調符号単位）と次の復調符号単位 $y_{i,1}^t$ との排他的論理和演算により逆変換後に次に逆変換されたデータ $x_{i,1}$ が同様に生成される。以下、同様にして、前記 RLL ストリーム 30

【0021】

【数8】

$$f'(y_i/s_i)$$

40

の最終の復調符号単位まで逆変換対象の復調符号単位と前記復調符号単位直前の復調符号単位との排他的論理和演算処理が反復される。

【0022】

このように、データ逆変換時には、逆変換直前の復調符号単位1つを復調符号単位の逆変換に利用しているために、エラーが発生すればその影響は前記復調符号単位及び次の復調符号単位まで及ぶ。例えば、逆変換前の復調符号単位 $y_{i,u-3}^{t*}$ でエラーが発生した場合には逆変換後のデータ $x_{i,u-3}^*$ 及び逆変換された次のデータ $x_{i,u-2}^*$ に影響を及ぼす。

【0023】

したがって、従来には RLL ストリーム 50

【 0 0 2 4 】

【 数 9 】

$$f'(y_i/s_i)$$

でエラーが発生すれば、それに該当するスクランブル前データ $x^*_{i-1, u-2, 3}$ だけでなく、その次のデータである $x^*_{i-1, u-2}$ までにエラーが伝播される問題点があった。このようなエラー伝播特性はスクランブルを利用したマルチモードコーディング方式の一般的な特徴であるといえる。

【特許文献1】米国特許第6,225,921号公報

【特許文献2】米国特許第6,281,815号公報

【特許文献3】韓国特許出願第1999-703183号(“デジタル変調回路、デジタル変調方法、デジタル復調回路及びデジタル復調方法”、三洋電機株式会社)

【特許文献4】韓国特許出願第2001-21360号

【非特許文献1】Kees A. Schouhamer Immink, “Code s for Mass Data Storage Systems,” Shann on Foundation Publishers, 1999

【発明の開示】

【発明が解決しようとする課題】

【 0 0 2 5 】

したがって、本発明の目的は、DC抑圧性能は3つの従来技術のように維持しつつ不十分なDC抑圧変調コード及びマルチモードコーディング方式を結合して高効率でかつDC抑圧能力に優れた変調コードを提供するデータ変調方法を提供するところにある。

【 0 0 2 6 】

本発明の他の目的は、エラー伝播確率を減らすデータ変調方法及びデータ復調方法を提供するところにある。

【 0 0 2 7 】

本発明のまた他の目的は、DC抑圧性能は維持しつつ入力データを不連続的にスクランブルして擬似ランダムデータ列に生成する多重化方法を採用したデータ変調方法及びデータ復調方法を提供するところにある。

【 0 0 2 8 】

本発明のさらに他の目的は、不十分なDC抑圧変調コードを配置するにおいて、連長制限を有するコードワードを生成し、コード列配置時に境界規則によってコードワードが取り替えられる場合にも最初のコード列特性をそのまま維持するようにコードワードを配置するコード配置方法を提供するところにある。

【課題を解決するための手段】

【 0 0 2 9 】

本発明によれば、前記目的はmビットのソースデータを最小連長制限d及び最大連長制限kに制限しつつnビット(n>m)のコードワードに変換するデータ変調方法において、入力データ列を一定長さに分割し、前記分割された入力データ列を多重化情報を利用して所定の多重化方式により多重化させ、多重化したデータ列を提供する段階と、前記多重化したデータ列に対して別途のビットが付加されたDC制御変換表を使用しない微弱DCフリーRLL変調を行う段階と、多重化してRLL変調されたコード列のうち最も小さなDC成分を有するコード列を提供する段階と、を含むデータ変調方法により達成される。

【 0 0 3 0 】

前記目的は、mビットのソースデータを最小連長制限d及び最大連長制限kに制限しつ

つ n ビット($n - m$)のコードワードに変換するデータ変調方法において、一定長さに分割された入力データ列を所定ビットの多重化情報を利用して複数種類のデータ列に多重化させるが、複数種類のそれぞれの多重化したデータ列に対して所定の多重化方法を利用して複数種類の擬似ランダムデータ列を生成する段階と、前記複数種類の擬似ランダムデータ列に対してRLL変調を行うことによって最適のDC抑圧が行われた変調されたコード列を生成する段階と、を含むデータ変調方法により達成される。

【0031】

また、前記目的は、 m ビットのソースデータを最小連長制限 d 及び最大連長制限 k に制限しつつ n ビット($n - m$)のコードワードに変換するデータ変調方法において、一定長さに分割された入力データ列を所定ビットの多重化情報を利用して複数種類のデータ列に多重化させるが、複数種類のそれぞれの多重化したデータ列に対して所定の多重化方法を利用して複数種類の擬似ランダムデータ列を生成する段階と、前記多重化したデータ列に対して別途のビットが付加されたDC制御変換表を使用しない微弱DCフリーRLL変調を行い、多重化させてRLL変調されたコード列のうち最もDC成分が小さなコード列を提供する段階と、を含むデータ変調方法により達成される。

10

【0032】

本発明の他の分野によれば、前記目的は、入力されるデジタルデータの各 n ビットを m ビット($n - m$)の復調符号単位に復調して所定長さの逆変換前のデータ列を生成する段階と、前記逆変換前のデータ列に対して、多重化情報を利用して不連続的にデスクランブルを行って逆変換されたデータ列を提供する段階と、を含むデータ復調方法により達成される。

20

【0033】

本発明のまた他の分野によれば、前記目的は、 m ビットのソースデータを最小連長制限 $d = 1$ 、最大連長制限 $k = 7$ に制限しつつ n ビット($n - m$)のコードワードを配列するコード配置方法において、コードワード a はコードワード b に連結され、先行するコードワードであるが、コードワード b はコードワード b_1 とコードワード b_2 のうち選択でき、コードワード a とコードワード b_1 とが連結されるコード列をコード列 X_1 、コードワード a とコードワード b_2 とが連結されるコード列をコード列 X_2 とする時、コードワード b_1 及び b_2 はコードワード内のビット“1”の数が奇数かまたは偶数かによって次のコードワードの遷移を予測するINVパラメータが反対値を有するように配置する段階と、コードワード a とコードワード b_1 または b_2 とが連結される時、境界規則によりコードワード a 、コードワード b_1 または b_2 が他のコードワードに変換されるとしてもコード列 X_1 及び X_2 はそのままINVパラメータ値が反対に維持されるように配置する段階と、を含むコード配置方法により達成される。

30

【0034】

本発明の方法の動作はコンピュータで判読可能な媒体のコンピュータで実行可能な命令語により具現できる。

【0035】

本発明のさらに他の様相及び利点は以下の説明によりさらに明確になり、また本発明の遂行により習得できる。

40

【発明の効果】**【0036】**

本発明は不十分なDC抑圧変調コードにマルチモードコーディング方式を結合してDC抑圧能力を向上させた高効率の変調コードを提供することにより記録密度側面で高い効率性を提供する。

【0037】

本発明は不十分なDC抑圧RLL変調時にコードワード間の連長制限条件を満足できなくてコードワードを他のコードワードに取り替える場合にも、コード列のDC抑圧能力を維持するようにコードワードを配置することによってコード列のさらに優秀なDC抑圧能力を提供する。

50

【 0 0 3 8 】

また、本発明はマルチコーディング方式において、入力データは擬似ランダムデータに不連続的にスクランブルされてDC成分が除去される。したがって、DC抑圧性能は維持しつつも一般的なスクランブルを利用したマルチモードコーディング方式よりエラー伝播確率を減らすことができる。

【 0 0 3 9 】

本発明の方法の動作はコンピュータで判読可能な媒体のコンピュータで実行可能な命令語により具現できる。

【 発明を実施するための最良の形態 】

【 0 0 4 0 】

以下、添付した図面を参照して本発明の望ましい実施例を説明する。

【 0 0 4 1 】

図4は、本発明によるデータ変調装置の一実施例によるブロック図である。入力データ列は式(1)に示されたように、 $x = (x_0, x_1, \dots, x_{k-1})$ に表示でき、 $v \times u$ 分割器10で入力データ列を式(2)に示されたように、 $v \times u (= k)$ と分けるが、すなわち、入力データ列を u バイト長を有する v 個のデータ列に分ける。

【 0 0 4 2 】

【 数 1 0 】

$$x = (x_0, x_1, \dots, x_{k-1}) \quad (1)$$

【 0 0 4 3 】

【 数 1 1 】

$$B_x = \begin{bmatrix} x_{0,0} & x_{0,1} & \dots & x_{0,u-1} \\ x_{1,0} & x_{1,1} & \dots & x_{1,u-1} \\ \cdot & \cdot & \cdot & \cdot \\ x_{i,0} & x_{i,1} & \dots & x_{i,j} & \dots & x_{i,u-1} \\ x_{v-1,0} & x_{v-1,1} & \dots & x_{v-1,u-1} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \dots \\ y_i \\ \dots \\ y_{v-1} \end{bmatrix} \quad (2)$$

ここで、 $x_{i,j} = x_{i \times u + j}$ である。

【 0 0 4 4 】

擬似スクランブルを利用した多重化器20は、分割器10により分割された $v \times u$ のそれぞれのデータ列に a ビットの付加情報を付けて $L = 2^a$ 個のデータ列に多重化させた後、付加された多重化情報 s によってデータ列を擬似ランダムデータに変換する。ランダムデータへの変換が終われば式(3)及び式(4)のように、一つの u バイト長を有するデータ列

【 0 0 4 5 】

【数 1 2】

 \underline{y}_i

に対して 2^a 個に多重化した相異なる内容の u バイトのデータが作られる。

10

【0 0 4 6】

【数 1 3】

$$\underline{C}_y = (\underline{C}_0, \underline{C}_1, \dots, \underline{C}_i, \dots, \underline{C}_{v-1}) \quad (3)$$

20

【0 0 4 7】

【数 1 4】

$$C_i = \begin{bmatrix} s_0, y^{0}_{i,0}, x_{i,1}, \dots, x_{i,q-1}, x_{i,q+1}, \dots, y^{0}_{i,p^*q}, \dots, x_{i,u-1} \\ s_1, y^1_{i,0}, x_{i,1}, \dots, x_{i,q-1}, y^1_{i,q}, x_{i,q+1}, \dots, y^1_{i,p^*q}, \dots, x_{i,u-1} \\ \vdots \\ s_{L-1}, y^{L-1}_{i,0}, x_{i,1}, \dots, x_{i,q-1}, y^{L-1}_{i,q}, x_{i,q+1}, \dots, y^{L-1}_{i,p^*q}, \dots, x_{i,u-1} \end{bmatrix} = \begin{bmatrix} f(\underline{y}_i / s_0) \\ f(\underline{y}_i / s_1) \\ \vdots \\ f(\underline{y}_i / s_{L-1}) \end{bmatrix} \quad (4)$$

30

ここで、 $u - 1 - q$ の倍数、 $p = 0, 1, \dots, r$ 、 $r = (u - 1) / q$ の商である。
 q はスクランブル間隔指数とも称する。関数

【0 0 4 8】

【数 1 5】

 $f(\underline{y}_i / s_i)$

40

は多重化情報 s を利用して入力データ列

【0 0 4 9】

【数 16】

 y_i

をランダムデータにした結果を意味する。

10

【0050】

同期及び多重化ID挿入器30は付加された多重化情報のビット数による複数(ここでは $L = 2^a$)のチャンネルを有することができ、 2^a 個に多重化した擬似ランダムデータ列に、すなわち、多重化情報が付加された多重化した擬似ランダムデータ列に同期パターンを挿入し、多重化情報を多重化識別子(以下、ID)に変換する。

【0051】

エンコーダ40は付加された多重化情報による複数(ここでは $L = 2^a$)チャンネルを有することができ、不十分なDC抑圧コードに対するRLLV変調を行う。ここで、 m ビットのソースデータを最小連長制限 d 及び最大連長制限 k に制限しつつ n ビット($n > m$)のコードワードに変換する変調をRLLV変調という。

20

【0052】

本発明のエンコーダ40は、別途のビットが付加された別途のDC抑圧用コード変換表を有していないために、リダンダンシーがなければDC抑圧は可能であるものの抑圧性能は落ちるコードを使用するが、所定連長制限条件(一実施例として、最小連長制限(d) = 1、最大連長制限(k) = 7とし、その例にRLLV(1、7、8、12)コード)に合うコードワードを生成し、前記連長制限条件別にコードワードでグルーピングし、ソースワードに対するコード列がDC制御能力を有するようにコードワードに配置されている主コード変換表及び前記所定連長制限条件を満足し、前記主コード変換表で不要なコードワードをとってDC制御用補助変換表を利用してRLLV変調を行う。

【0053】

同期及び多重化ID挿入器30において多重化情報の多重化IDへの変換は最小連長制限 $d = 2$ 、最大連長制限 $k = 7$ として最小連長制限を長くすることにより最小マーク(またはビット)の大きさを大きくして信号の干渉雑音を減らすことができる。他の実施例として、エンコーダ40の最小連長制限 $d = 2$ 、最大連長制限 $k = 10$ になり、その例にRLLV(2、10、8、15)コードを挙げられる。

30

【0054】

比較及び選択器50はRLLV変調された 2^a 個の変調ストリームに対してDC成分が最も少ない変調ストリーム一つを選択する。

【0055】

図5は、図4に示された多重化器に適用される擬似スクランブルを利用した多重化方法の一例を示した図であって、入力データを不連続的にスクランブルする方式を本発明では擬似スクランブルを利用した多重化方法と称する。

40

【0056】

スクランブルを利用した多重化方法は図2に示されたように、もしある位置にエラーが発生すればそのエラーによってその次のデータにもエラーが発生する。したがって、コード列のDC成分に影響を及ぼさない程度までデータスクランブルを不連続的に行えば、次のデータまでエラーが伝播される確率を減らすことができるという長所が生じる。

【0057】

図5において、 $x_{i,0} \sim x_{i,u-1}$ が各々所定ビット数よりなるデータで構成された u バイト長を有する入力データ列

50

【 0 0 5 8 】

【 数 1 7 】

 y_i

は、多重化情報 s_t と連続的に配置されずに毎 q 番目に配置された排他的論理和素子による排他的論理和演算を通じて擬似ランダムデータ関数 10

【 0 0 5 9 】

【 数 1 8 】

 $f(\underline{y_i}/s_i)$

に変換されている。 20

【 0 0 6 0 】

ここで、多重化情報 s_t は u バイト長を有する入力データ列

【 0 0 6 1 】

【 数 1 9 】

 y_i

を多重化させる多重化情報という時、多重化情報 s_t のビット数 a が入力データのビット数 m より小さいかまたは同一である場合いずれも適用される。ただし、 $a < m$ の場合は入力データの一部ビット (LSB (Least Significant Bit) から a ビットまたは MSB (Most Significant Bit) から a ビットまたは m ビットのデータ内の任意の a ビット) を利用して擬似ランダムデータにしても、 m ビット全部を利用して擬似ランダムデータにする場合とほとんど同じ性能を示す。

【 0 0 6 2 】

すなわち、先頭の変調するデータ (符号変調単位という) $x_{i,0}$ と初期データ (多重化情報) s_t との排他的論理和演算により、先頭のコード変調単位 $x_{i,0}$ (変調される最初のデータ) はデータ $y_{i,0}^t$ に変調される。符号変調単位 $x_{i,1}$ から $x_{i,q-1}$ までは排他的論理和演算処理が行われず、したがって、コード変調単位 $x_{i,1}$ から $x_{i,q-1}$ は変わらないまま出力される。前述した変換が終わった符号変調単位のデータ $y_{i,0}^t$ と q 番目の符号変調単位 $x_{i,q}$ との排他的論理和演算により次の変換データ $y_{i,q}^t$ が同様に生成される。以下、同様にして、 q 番目の符号変調単位で前記入力データ列 40

【 0 0 6 3 】

【数 2 0】

$$\underline{y}_i$$

の最終の符号変調単位まで排他的論理和演算処理が反復される。

10

【0064】

図6は、図5に示された擬似スクランブルを利用した多重化方法で変換されたデータを再生する場合のエラー伝播特徴を示した図面である。データ逆変換時、逆変換前のデータ列

【0065】

【数 2 1】

$$f'(\underline{y}_i/s_i)$$

20

はデコーダ（図示せず）により変調時に使われたRLL変調方式によってデータ列を復調するデコーダにより復調されたRLLストリームである。また、図6に示された構成は、逆変換前のデータ列について、多重化情報を利用して不連続的にデスクランブルして逆変換されたデータ列を提供する逆多重化器と称することができる。

【0066】

図6において、先頭の初期データ（多重化情報 s_t ）を除外した復調符号単位 $y_{i,0}^t$ から毎 q 番目に、逆変換対象である復調符号単位と q 番目以前の復調符号単位との排他的論理和演算を通じて復元されたデータが生成される。

30

【0067】

すなわち、先頭の復調符号単位 $y_{i,0}^t$ と初期データ（多重化情報） s_t との排他的論理和演算により逆変換後に逆変換されたデータ $x_{i,0}$ が生成される。復調符号単位 $x_{i,1}$ から $x_{i,q-1}$ までは排他的論理和演算処理が行われないので、復調符号単位 $x_{i,1}$ から $x_{i,q-1}$ までは変わらないまま出力される。その後、前述した $y_{i,0}^t$ （逆変換前の復調符号単位）と次の復調符号単位である q 番目の復調符号単位 $y_{i,q}^t$ との排他的論理和演算により逆変換後に次に逆変換されたデータ $x_{i,q}$ が生成される。以下、同様にして、逆変換前データ列

40

【0068】

【数 2 2】

$$f'(\underline{y}_i/s_i)$$

の最終の復調符号単位まで q 番目の復調符号単位で排他的論理和演算処理が反復される。

50

【 0 0 6 9 】

したがって、擬似ランダムデータよりなるデータを R L L 変調した後、光ディスクのような保存媒体に保存されたデータストリームを再生する時に R L L ストリーム

【 0 0 7 0 】

【 数 2 3 】

$$f'(y_i/s_i)$$

10

において、変調時に入力データが排他的論理和演算により他のデータに変換されていない $x^*_{i, q+1}$ のみエラーが発生した場合、他のデータにエラー伝播をしない。エラー伝播は変調時に入力データが排他的論理和演算により他のデータに変換されたデータに該当する再生された R L L ストリームにエラーが発生する場合にのみ発生する。

【 0 0 7 1 】

これは、図 5 に示された擬似スクランブルを利用した多重化方法でデータを擬似ランダムデータに変換すれば、図 2 に示された多重化方法で変換した時より排他的論理和演算の周期を q とした時にエラー伝播確率が $1/q$ に減るということを意味する。ここで、 q の値は R L L 変調した時に D C 抑圧性能を受容する水準で選択すればよい。なぜなら、 q の値が大きくなればエラー伝播の確率は落ちるが、D C 抑圧性能も低下する特徴があり、逆に q 値が小さくなれば D C 抑圧性能には有利であるが、エラー伝播確率が増加する特徴があるからである。

20

【 0 0 7 2 】

図 7 は、排他的論理和演算の周期 q による D C 抑圧能力の変化を示す P S D 曲線である。多重化情報 s_t のビット数 $a = 2$ ビットとし、一つの多重化長さ $u = 50$ バイトとし、変調する入力データのビット数 $m = 8$ とする時の排他的論理和演算の周期 q による変調コード列の D C 抑圧性能を示す。排他的論理和演算は、2 ビットの多重化情報 s_t を 8 ビットの入力データの L S B から 2 ビットだけ行った。P S D 曲線は下から排他的論理和演算周期 q をそれぞれ 1 バイト、5 バイト、10 バイト、15 バイト、20 バイトとして排他的論理和演算結果による D C 抑圧能力を示す。図 7 で分かるように、毎 5 バイトに排他的論理和演算を行っても D C 抑圧性能には大きい差がない。一方、エラー伝播率は $1/5$ に減らすことができる。

30

【 0 0 7 3 】

次に、本発明の一実施例による微弱 D C フリー R L L 変換コードについて説明する。(d 、 k 、 m 、 n) と表現される R L L コードにおいて、コードの性能を表現する要因のうち大きく記録密度の側面及び D C 成分を抑圧する能力により、そのコードの優秀な程度を評価する。ここで、 m はデータビット数 (ソースデータのビット数、情報ワードビット数ともいう)、 n は変調後のコードワードビット数 (チャンネルビット数ともいう)、 d はコードワード内で 1 と 1 間に存在できる連続する 0 の最小数 (最小連長制限という)、 k はコードワード内で 1 と 1 間に存在できる連続する 0 の最大数 (最大連長制限という) である。コードワード内のビット間隔は T (記録または再生時に使われるクロック信号周期に該当) として表す。

40

【 0 0 7 4 】

変調方法のうち記録密度を向上させうる方法は、 d 及び m は与えられた条件としたままコードワードのビット数 n を減らすことである。しかし、R L L コードはコードワード内で最小連長制限 d 及び最大連長制限 k を満足しなければならない。この (d 、 k) 条件を満足しつつデータビット数が m とする時、R L L (d 、 k) を満足するコードワードの数

50

は2m個以上であればよい。しかし、実際このようなコードを使用するためにはコードワードとコードワードとが連結される部分でも連長制限条件、すなわち、RL L (d, k) 条件を満足しなければならず、光ディスク記録/再生装置のようにコードのDC成分がシステム性能に影響を及ぼす場合には使用しようとするコードがDC抑圧能力を有していなければならない。

【0075】

本発明ではソースコードに対して変換されるコードワードのコード表は大きく2つ、すなわち、1)主変換表、2)DC制御用補助変換表として生成される。

【0076】

それぞれの変換表内のコードワード生成方式は下記の通りであり、最小連長制限が1、最大連長制限が7である(1,7)コードを例として説明する。

【0077】

図8は、主コード変換表のいろいろなコードワードグループ及び該当コードグループのコードワード特性を表で示したものである。

【0078】

コードの最小連長制限をd、最大連長制限をk、ソースデータのビット数をm、変調後のコードワードのビット数をnとし、コードワードのLSBからMSB方向に連続する0の数をEZ (End Zero)、MSBからLSB方向に連続する0の数をLZ (Lead Zero)とする時、d = 1、k = 7、m = 8、n = 12、0 EZ 5のコードワードをLZの条件によって分類すれば次の通りである。

【0079】

- (1) 1 LZ 7を満足するコードワードの数：210個
- (2) 0 LZ 4を満足するコードワードの数：316個
- (3) 0 LZ 2を満足するコードワードの数：264個

m = 8のソースデータの変調のためにはコードワードの数が最小256個以上でなければならないが、前記(1)の場合、コードワードの数が256個未満であるので他のLZ条件を満足するコードワードのうち一部をとって必要なコードワードの数を満足できる。この場合、前記(2)でLZ = 0のコードワードのうち1010xxxxxxxの51個を引いて前記(1)のグループに追加すれば、前記(1)グループに属するコードワードの数は261個になり、前記(2)のグループに属するコードワードの数は316 - 51 = 265個、前記(3)のグループに属するコードワードの数は264個で(1)~(3)グループいずれも256個以上のコードワードを保有するようになって、各条件に該当するグループのコードワードの数が8ビットのソースデータに対して256個という変調コードワードの最小数を満足できる。このうち、それぞれ256個のコードワードだけを選んで3個の主コード変換表MCG1~MCG3を作る。図8の表において、MCG (Main Code Group) 1は前記(1)の条件に該当するコードワード及び、(2)を満足するグループからとった一部(51個)のコードワードを含むグループの名称であり、MCG2及びMCG3は各々順に前記(2)、(3)の条件に該当するコードワードを含むグループの名称であり、これら主コードグループMCG1~MCG3各々から256個のコードワードだけソースコードに対する変換コードとして使われる。

【0080】

図9は、DC制御用補助変換表のいろいろなコードワードグループ及び該当コードグループのコードワード特性を表で示したものである。

【0081】

DC制御用補助変換表内のコードワードはd = 1、k = 7、m = 8、n = 12のコードワードのうち6 EZ 7を満足するコードワード及びMCGで残るコードワードとLZ = 5、6またはLZ = 3のコードワードとを合わせてDC制御用補助コードグループACGのコードワードとして使用する。このコードワードの生成条件を具体的に説明すれば次の通りであり、各項目は順に図4の表でDC制御用補助変換表の名称はACG1、ACG2、ACG3として示されている。

10

20

30

40

50

【 0 0 8 2 】

ACG1 : 6 EZ 7、LZ 0を満足するコードワード8個 + MCG1で残るコードワード5個 + 6 EZ 7、LZ = 0、また1010xxxxxxxのコードワード2個 = 15個

ACG2 : 6 EZ 7、0 LZ 6を満足するコードワード12個 + 0 EZ 5、5 LZ 6を満足するコードワード21個 + MCG2で残るコードワード9個 - 6 EZ 7、LZ = 0、また1010xxxxxxxのコードワード2個 = 40個

ACG3 : 6 EZ 7、0 LZ 3を満足するコードワード10個 + 0 EZ 5、LZ = 3を満足するコードワード33個 + MCG3で残るコードワード8個 = 51個

図10は、図8で説明した主変換表及び図9に示されたDC制御用補助変換表を用いて以前コードワードaのエンドゼロ数EZ__aによって決定される次のコードグループ(M = ncgdet)、すなわち、以前コードワードaの次のコードグループを示すパラメータncgを整理した表である。コードワードbは、以前コードワードaのエンドゼロ数EZ__aによってそのコードワードbが属しているコードグループが定められるが、EZ__a = 0であれば次のコードワードbが属しているコードグループは1 (= MCG1)であり、1 EZ__a 3であれば次のコードワードbが属しているコードグループは2 (= MCG2)であり、4 EZ__a 7であればコードワードbが属しているコードグループは3 (= MCG3)である。

10

【 0 0 8 3 】

一方、コードワードaとコードワードbとが連結される地点でも連長制限(d、k)条件を満足しなければならない。図11は、コードワードaとbとが連結される時に連長制限の条件のために考慮しなければならないことを示したものである。図11で、コードワードaのエンドゼロ数EZ__aとコードワードbのリードゼロ数LZ__bとを合算した値が最小連長制限d以上、最大連長制限k以下であってこそ連長制限の条件を満足できる。

20

【 0 0 8 4 】

図12は、図11を通じて説明した連長制限の条件を満足できない場合が発生する場合、コード変換前後のパラメータINVの変化を示している。パラメータINVは次のコードワードの遷移が分かるパラメータであって、コードワード内でビット“1”の数が偶数であればINV値は“0”であり、コードワード内でビット“1”の数が奇数であればINVの値は“1”である。付加的に、DSV(Digital Sum Value)はコードワードストリームでのデジタル合算値であって、DSVの絶対値が小さければ直流成分または低周波成分が少ないことを意味する。CSVはコードワード内でのデジタル合算値であって、CSVが小さければコードワードの直流成分または低周波成分が少ないことを意味するが、コード変換中の直流成分や低周波成分の評価にはCSVが利用される。コードワードストリームで現コードワードまで累積されたINVの値が“0”であれば次のコードワードのCSV値をそのコードワード以前までの累積されたDSV値にそのまま合算してDSV値を更新し、累積されたINV値が“1”であれば次のコードワードのCSV値の符号を反転させてそのコードワード以前までの累積されたDSV値に合算してDSV値を更新する。

30

【 0 0 8 5 】

図12を参照すれば、コードワードbは以前コードワードaのEZによってそのコードワードbが属しているコードグループが定められるが、主変換表及びDC制御用補助変換表でコードワード数が足りなくて他のコード変換表からコードワードをとったコードグループが指定される場合に(d、k)条件を満足できない場合が発生する。図12では、d EZ__a + LZ__b kに違反する場合を例示したが、この場合にはコードワードaのEZが変わるが、このように連長制限条件を満足できなくてコードワード変化が起きることを境界規則という。コードワードストリーム内のビット“1”の数が偶数か奇数かを示すパラメータINVは境界規則によりコード変化前の状態から変わる可能性がある。したがって、コードワードはDC制御のためのコード変換表内であらかじめ決定された順序で配列される。

40

50

【 0 0 8 6 】

図 1 3 は、D C 制御可能なコードワード b 1、b 2 によるコード列の分岐例を示した図である。本発明のコード変換において最も大きい特徴のうち一つは、D C 制御をするために選択可能な 2 つのコード変換表内のコードワードは I N V (コードワードストリーム内のビット“ 1 ”の数が偶数または奇数を表す) 特徴を逆に維持するようになっている。このようにすれば、2 つのコード変換表内のコードワードは I N V が逆であるので、両コードワードのうち一つは D C 制御が最適の方向に進むからである。しかし、前記境界規則で説明したように、I N V に変化が発生する場合があるが、この場合は D C 制御が可能な時点での選択可能な 2 つのコード変換表にいずれも同じ現象、すなわち、I N V が 2 つの選択可能なコード変換表で同様に变化すれば何の問題もない。このために本発明では次のよ

10

【 0 0 8 7 】

まず、図 1 3 の A、すなわち、コードワード a とコードワードと b が連結される地点でコードワード b として選択可能な b 1 及び b 2 がある場合、E Z__a が“ x x x x x x x x x x 1 0 1 ”の時、L Z__b 1 (コードワード b 1 のリードゼロ数) が“ 1 0 1 x x x x x x x x ”であれば、その時、L Z__b 2 (コードワード b 2 のリードゼロ数) が各々“ 1 0 1 x x x x x x x x ”となる。すなわち、リードゼロ数が“ 1 0 1 x x x x x x x x ”のコードワードは M C G 1 及び A C G 1 内の同じ位置に配置し、E Z が“ x x x x x x x x 1 0 1 ”のあらゆるコードは各々 M C G 1 及び A C G 1、M C G 2 及び A C G 2、M C G 3 及び A C G 3 内の同じ位置に配置して、コードワード a のエンドゼロ数が

20

【 0 0 8 8 】

次に、図 1 3 の B、すなわち、コードワード b とコードワード c とが連結される点でコードワード b 1 と b 2 とが各々コードワード c と連結される時、境界規則によりコードワード b 1、b 2 またはコードワード c が他のコードワードに変換されるとしてもコードワード b 1 とコードワード c とが連結されるコード列及びコードワード b 2 とコードワード c とが連結されるコード列はそのままパラメータ I N V 値が逆に維持される。

【 0 0 8 9 】

同期パターン及び多重化 I D について説明する。

30

【 0 0 9 0 】

ビット 1 と 1 間の 0 の最大数 $k = 7$ に制限された変調方式で同期パターンで $k = 7$ の制限を違反する“ 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 ”を使用する。

【 0 0 9 1 】

S y n c P a t t e r n : 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0

多重化 I D はデータ列の多重化のために 4 ビットの多重化情報を利用して図 1 4 に図示されたように 6 ビットに変換するが、この場合、一つのデータ列は $L = 2^4 = 16$ 種類のランダムデータ列に変換される。

【 0 0 9 2 】

図 1 5 A ないし図 1 5 E は、前述した事項を考慮して生成及び配置した主変換コード表である。

40

【 0 0 9 3 】

図 1 6 は前述した事項を考慮して生成及び配置した補助 D C 制御用コード変換表である。D C 制御用補助変換表内のコードワードを使用する時には、以前のコードワード(図 1 3 のコードワード a) と後続のコードワード(図 1 3 のコードワード c) との連長制限違反如何を検査した後、違反しない場合にのみ使用しなければならない。

【 0 0 9 4 】

図 1 7 は、本発明で提案した方式通り R L L (1、7) 変調を行った後、コード列に対する P S D 曲線を示した図である。

50

【0095】

図18は、本発明で提案したRLL(1,7)コードと従来のコードA-Code、B-Code、C-Codeとを比較して得られたコード率及びコード効率を整理した表である。本発明のRLL(1,7)コードは、従来のコード列と類似したDC抑圧能力を有しつつもコードの効率が極めて記録密度を約2%高めうる長所がある。

【0096】

図19は、RLL(2,10,8,15)コードを本発明に適用した場合、従来のDVDで使われているEFMP(Eight to Fourteen Modulation Plus)コードとDC抑圧能力を比較したPSD曲線を示している。ここで、EFMPコードは主変換コード表と共に別途のDC制御用コード変換表を使用してDC抑圧を行うコードである。RLL(2,10,8,15)コードの一例は同一出願人により出願された特許文献4に開示されており、本発明では付加ビットを使用したDC抑圧用コード変換表を使用せずに主変換表及びDC制御用補助変換表を使用する不十分なDC抑圧RLL変調を行っている。

10

【0097】

図20は、従来のEFMPコードと本発明が適用されたRLL(2,10,8,15)コードを比較してコード率及びコード効率を整理した表である。本発明で提案した方式のRLL(2,10,8,15)コードがEFMPと類似したDC抑圧能力を有しつつも高いコード効率を有して記録密度も約5.4%向上する。

【0098】

本発明のいくつかの実施例が図示及び説明されたが、このような実施例は、特許請求の範囲により限定される本発明の思想及び原理を離脱しない範囲内で当業者により変更できることはもちろんである。

20

【産業上の利用可能性】

【0099】

本発明はデジタルデータを保存する保存媒体に広範囲に利用できるが、HD-DVD(High Density Digital Versatile Disc)のような高密度保存媒体に効果的に適用できる。

【図面の簡単な説明】

【0100】

- 【図1】従来コードのPSD曲線を示した図である。
- 【図2】従来の多重化方法を説明するための図である。
- 【図3】図2に示された従来の多重化方法で変換されたデータを再生する場合のエラー伝播特性を示した参考図である。
- 【図4】本発明によるデータ変調装置の一実施例によるブロック図である。
- 【図5】図4に示された多重化器に適用される擬似スクランブルを利用した多重化方法を説明するための図である。
- 【図6】図5に示された擬似スクランブルを利用した多重化方法で変換されたデータを再生する場合のエラー伝播特性を示した参考図である。
- 【図7】本発明のデータ変調方法によるDC抑圧能力を示したPSD曲線である。
- 【図8】主変換コードグループのコードワード特性を整理した表である。
- 【図9】DC制御用補助変換コードグループのコードワード特性を整理した表である。
- 【図10】エンドゼロ数EZによって決定される次のコードグループncgを整理した表である。
- 【図11】コードワードa及びbが連結される時に連長制限の条件を説明するための図である。
- 【図12】図11により連長制限の条件を満足できない場合のコード変換前後のパラメータINVの変化を示した表である。
- 【図13】DC制御可能なコードワードb1、b2によるコード列の分岐例を示した図である。

30

40

50

【図14】図4に示された同期及び多重化ID挿入器における多重化情報の多重化IDへの変換例を示した変換表である。

【図15A】前述した事項を考慮して生成及び配置した主変換コード表である。

【図15B】前述した事項を考慮して生成及び配置した主変換コード表である。

【図15C】前述した事項を考慮して生成及び配置した主変換コード表である。

【図15D】前述した事項を考慮して生成及び配置した主変換コード表である。

【図15E】前述した事項を考慮して生成及び配置した主変換コード表である。

【図16】前述した事項を考慮して生成及び配置したDC制御用補助変換コード表である。

【図17】本発明のRLL(1,7)コードのPSD曲線を示した図である。

10

【図18】本発明のRLL(1,7)コードと既存のコードとの記録密度及び記録効率を比較した表である。

【図19】本発明のRLL(2,10)コードのPSD曲線を示した図である。

【図20】本発明のRLL(2,10)コードと既存のEFMPコードとの記録密度及び記録効率を比較した表である。

【符号の説明】

【0101】

10 vXu分割器

20 多重化器

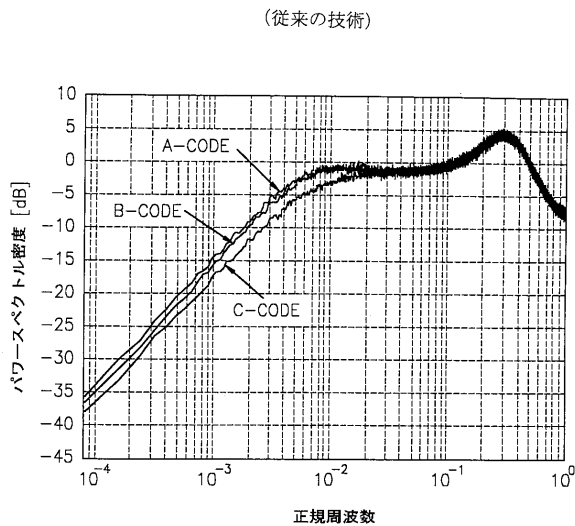
30 同期及び多重化ID挿入器

40 エンコーダ

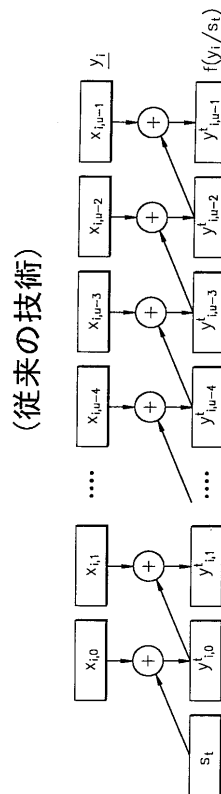
50 比較及び選択器

20

【図1】

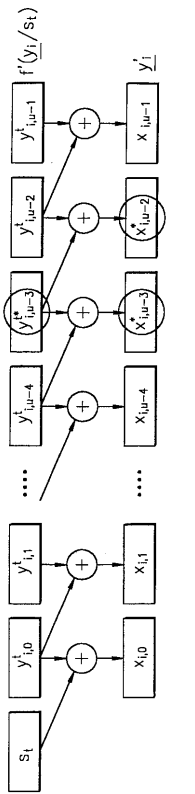


【図2】

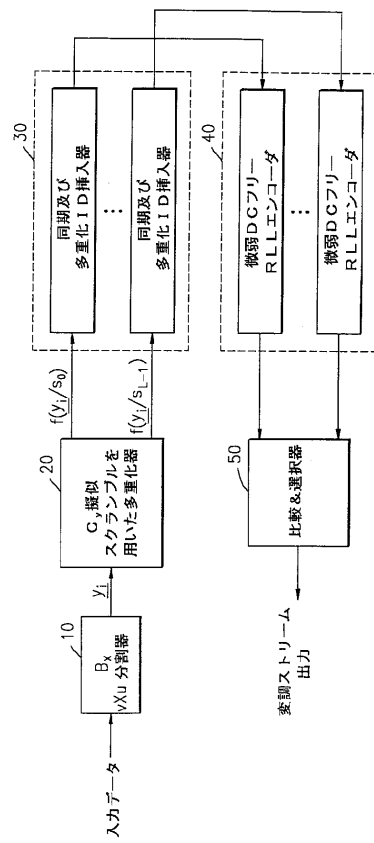


【 図 3 】

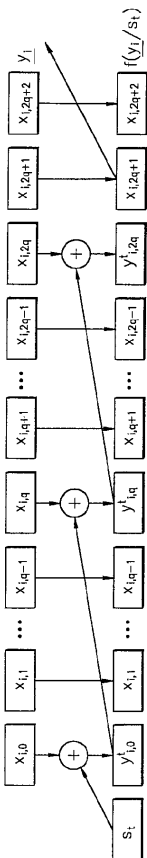
(従来の技術)



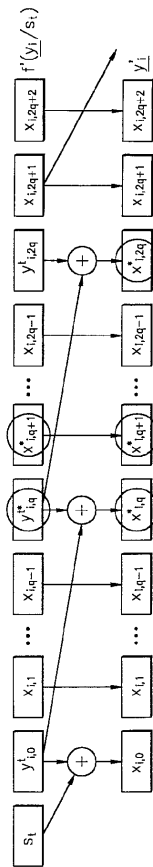
【 図 4 】



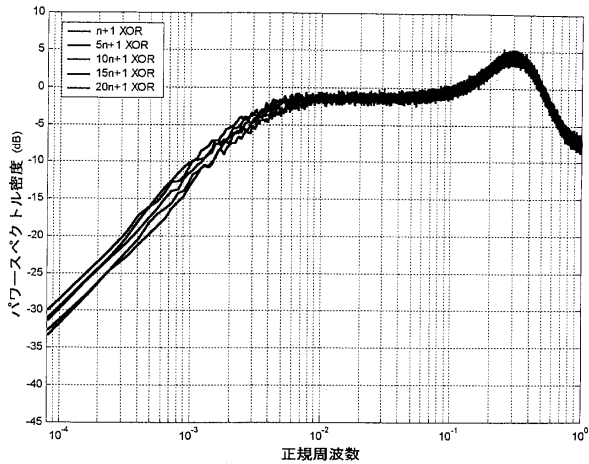
【 図 5 】



【 図 6 】



【 図 7 】



【 図 8 】

	LZ (EZ)	数	補足	数	消去	数	合計
MCG1	1 ≤ LZ ≤ 7 (0 ≤ EZ ≤ 5)	210	1010xxxxxxx (0 ≤ EZ ≤ 5)	51			261(5)
MCG2	0 ≤ LZ ≤ 4 (0 ≤ EZ ≤ 5)	316			1010xxxxxxx (0 ≤ EZ ≤ 5)	51	265(9)
MCG3	0 ≤ LZ ≤ 2 (0 ≤ EZ ≤ 5)	264					264(8)

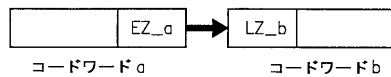
【 図 9 】

	LZ (EZ)	数	補足1	数	補足2	数	補足3/消去	数	合計
ACG1	1 ≤ LZ ≤ 7 (0 ≤ EZ ≤ 7)	8		MCG1からの ビット	5	1010xxxxxxx (0 ≤ EZ ≤ 7)	2	15	
ACG2	0 ≤ LZ ≤ 6 (0 ≤ EZ ≤ 7)	12	5 ≤ LZ ≤ 6 (0 ≤ EZ ≤ 5)	21	MCG2からの ビット	9	1010xxxxxxx (0 ≤ EZ ≤ 7)	40	
ACG3	0 ≤ LZ ≤ 3 (0 ≤ EZ ≤ 7)	10	LZ=3 (0 ≤ EZ ≤ 5)	33	MCG3からの ビット	8		51	

【 図 10 】

EZ_a	M=ncgdet (コードワードa)
0	1
1-3	2
4-7	3

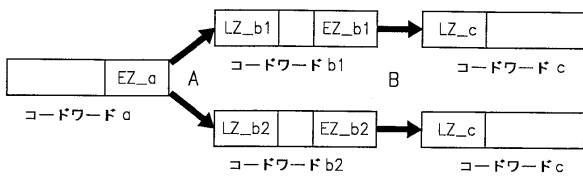
【 図 11 】



【 図 12 】

コードワードa	コードワードb	INVの変化
xxxxxxx0101 → xxxxxx0100	1010xxxxxxx	変化有
xxxxxxx01001 → xxxxxx01010		変化無
~		
xxx10000001 → xxx100001010		

【図13】



【図14】

多重化情報	変換された多重化ID	次の状態
0	100001	1
1	100010	2
2	100100	2
3	100101	1
4	101000	2
5	101001	1
6	000100	2
7	000101	1
8	001000	2
9	001001	1
A	001010	2
B	010000	3
C	010001	1
D	010010	2
E	010100	2
F	010101	1

【図15A】

データ番号	MCG1			MCG2			MCG3		
	MSB	LSB	NCG	MSB	LSB	NCG	MSB	LSB	NCG
000	000001010000	3	000010010000	3	001000001001	1			
001	000010010000	3	000010001001	1	001000001010	2			
002	000000100000	3	000010001010	2	001000010010	2			
003	000101010000	3	000010010010	2	001000010101	1			
004	001001010000	3	000010010101	1	001000010100	2			
005	001010010000	3	000010010100	2	001000010101	1			
006	001010100000	3	000010000001	1	001000000010	2			
007	000001010000	3	000010000010	2	001001010000	3			
008	101001010000	3	000010100000	3	001001001001	1			
009	101001000000	3	000010010001	1	001000000100	2			
010	000100010000	3	000010100010	2	001000001000	2			
011	000000101001	1	000010100010	2	001000001001	1			
012	000000101010	2	000010100101	1	001010010000	3			
013	000001001010	2	000010101000	2	001001010001	1			
014	000001010101	1	000010101001	1	001000101010	2			
015	010001010000	3	000010000100	2	001001001010	2			
016	000001000001	1	000101010000	3	001001010101	1			
017	000001000010	2	000100000001	1	001001010010	2			
018	000001001000	2	000010001000	2	001000010001	1			
019	000001001001	1	000010001001	1	001000010010	2			
020	000001000001	1	000010000101	2	001001000010	3			
021	000001010010	2	000100001000	3	001010001001	1			
022	000001010100	2	000100010000	3	001001010100	2			
023	000010010101	1	000100010000	3	001000010100	2			
024	000010001010	2	000101010000	3	001010010101	1			
025	000010000001	1	000101000000	3	001010010010	2			
026	000000101000	2	010010010000	3	001000100001	1			
027	010010010000	3	001000100000	3	001000100100	2			
028	000010000001	1	010010100000	3	001000010000	3			
029	000001000010	2	000100000000	3	001001000001	1			
030	000001000100	2	001000010000	3	001000010100	2			
031	000001000101	1	000100010101	2	001000000100	2			
032	010010100000	3	000100001010	2	001000100101	1			
033	000010101001	1	000100101001	1	010000100000	3			
034	000001001000	2	000100000010	2	010010010000	3			
035	000010000010	2	010100100000	3	010010100000	3			
036	000010000101	1	000101001001	1	010000100000	3			
037	000010000100	2	000100000100	2	001000100000	3			
038	000001010001	1	000100001000	2	010000100000	3			
039	000010010010	2	000100000101	1	100000101000	3			
040	000010001001	1	000101010001	1	010000010000	3			
041	000010010100	2	000100010100	2	010000100000	3			
042	000010100010	2	000101001010	2	010101000000	3			
043	000001001001	1	000101010101	1	100001001000	3			
044	000010100100	2	000101010010	2	001010100001	2			
045	001000000001	1	000010100001	1	001010100000	2			
046	000010001000	2	000100010010	2	001010100000	2			
047	010100010000	3	001000100000	3	001010100101	1			
048	000100010001	1	000100001001	1	001010100100	2			
049	000010101010	2	000100010100	2	001010000001	1			
050	000100000010	2	00010001000100	2	001000100000	2			
051	000100000101	1	000100010001	1	001001000000	2			
052	010100100000	3	000100100100	2	001001000100	2			
053	000101001001	1	000100101000	2	001010000001	2			
054	000100000100	2	000101000010	1	001010100001	2			
055	000100001000	2	000101000010	2	001010101000	2			
056	000101010101	1	000100010101	1	100010100000	3			
057	000100101010	2	000101010100	2	010000101001	1			
058	000010010001	1	100001010000	3	010000000100	2			
059	000010101000	2	001001001001	1	010000001000	2			
060	000101010001	1	001000000010	2	010000001001	1			

【図15B】

データ番号	MCG1			MCG2			MCG3		
	MSB	LSB	NCG	MSB	LSB	NCG	MSB	LSB	NCG
061	000101001010	2	001000000100	2	100100010000	3			
062	000101010010	2	001000000101	1	010001001001	1			
063	001000000101	1	100010010000	3	010000101010	2			
064	000101010100	2	001001010001	2	010000100110	1			
065	000010100001	1	001000001000	2	010001010101	2			
066	0001000001010	2	0010000101010	2	010001010010	2			
067	000100100000	3	0010001010101	1	001010101001	1			
068	000100001001	1	001001001010	2	001010000100	2			
069	000100010010	2	000100010001	1	010001010001	2			
070	000100010100	2	000101000100	2	010001010100	2			
071	000100010101	1	001010001001	1	010000001010	2			
072	000100100010	2	001001010010	2	010000010101	1			
073	000100100100	2	001001010100	2	010001000100	2			
074	000100100101	1	001001010101	1	010000001001	1			
075	000100101000	2	001010001010	2	001010001000	1			
076	001000101001	1	000100100001	1	100000010000	3			
077	001000000010	2	000101001000	2	010000010001	1			
078	000101001000	2	010000010000	3	001010101010	2			
079	000100100101	1	000101000001	1	010000001010	1			
080	001000000101	2	001000001010	2	010100000101	1			
081	001000001000	2	001000010100	2	010000001000	1			
082	001001010101	1	001000010101	1	010000010001	2			
083	001000000100	1	001001000001	1	010001000010	2			
084	001001010001	1	001001001001	2	010010100101	2			
085	001000101010	2	001001010100	2	010010100100	2			
086	001001001010	2	001001010010	1	010000000001	1			
087	001010010101	1	001010000100	2	010000001010	2			
088	001001010010	2	001000001001	1	100000010000	3			
089	000100010001	1	001000010100	2	010001000001	1			
090	000101000010	2	010000010000	3	010000010100	2			
091	001010001001	1	001000010001	1	010000100010	2			
092	001001010100	2	001000100010	2	010000001001	1			
093	001010001010	2	001000100100	2	100101010000	3			
094	001010100101	1	001000100101	1	010010000001	1			
095	001010010010	2	010101010000	3	010000010000	2			
096	000100100001	1	001000100001	1	010000101000	2			
097	000100010000	2	001000010000	2	010000000001	1			
098	001000010000	3	001001000001	2	010001000010	2			
099	000101000001	1	001001000001	1	010010010001	1			
100	0010000001010	2	001001000010	2	010010100000	2			
101	001000010010	2	001010100001	1	010010100001	2			
102	000101000100	2	001010000100	2	010100001010	2			
103	001001000100	1	001001000100	2	010100001000	2			
104	001010010100	2	001010000100	1	010100010101	1			
105	001010100010	2	001010000001	2	010100010100	2			
106	010000000101	1	010000010001	1	010000010101	2			
107	001010100100	2	001010100100	2	010001000100	2			
108	001000010001	1	100010100000	3	010100000001	1			
109	001000010100	2	010000100001	2	010100000000	1			
110	001000100000	3	010000000100	2	010001001000	2			
111	001000100001	1	010000000100	2	010000000100	2			
112	001000100010	2	010000000001	1	010000000101	1			
113	001000100100	2	001000100000	3	010100000001	1			
114	001000100101	1	100010100000	3	010100000001	1			
115	010000010000	3	010000010100	2	010010000100	2			
116	001000100001	1	010000010010	2	010010000100	2			
117	001000010000	2	010000010101	1	010010000101	2			
118	001000100010	2	010000100010	2	010010100100	2			
119	001000100101	1	010000000001	1	010100000001	1			
120	001001000100	2	001010000100	2	010100000010	2			

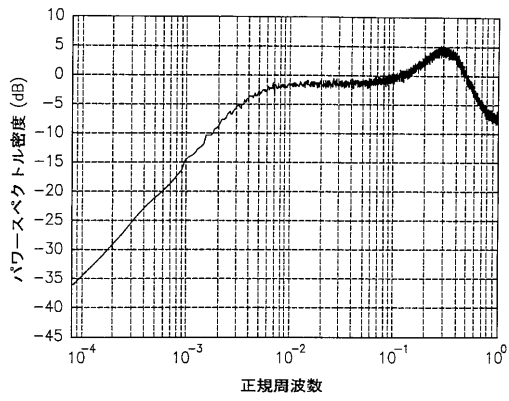
【 図 15 C 】

データ番号	MCG1				MCG2				MCG3			
	コードワード				コードワード				コードワード			
	MSB	LSB	NCG		MSB	LSB	NCG		MSB	LSB	NCG	
121	001010100001	1	2	1	010010001001	1	2	1	010101001001	1	2	1
122	001010101000	1	2	2	010001010100	2	2	2	010100000010	2	2	2
123	001001001000	1	2	2	010010001010	2	2	2	010100000100	2	2	2
124	001001000101	1	2	1	010010101011	1	2	1	010100000101	1	2	1
125	001010000010	1	2	1	010100100100	1	2	1	010100000100	1	2	1
126	010000101001	1	2	1	010000000001	1	2	1	010101000001	1	2	1
127	010000001000	1	2	1	001010001000	1	2	1	010100100100	1	2	1
128	001010000001	1	2	1	100000010000	3	2	1	100100100000	3	2	1
129	010001001000	1	2	1	001010101001	1	2	1	100000101001	1	2	1
130	010000001000	1	2	2	001010101010	2	2	2	100000101010	2	2	2
131	010000101010	1	2	2	010000001010	2	2	2	010101000010	2	2	2
132	010001010101	1	2	1	010000010101	1	2	1	010100100101	1	2	1
133	001010001000	1	2	1	010010100001	1	2	1	010100100001	1	2	1
134	010001010001	1	2	1	010010010100	1	2	1	010100101010	1	2	1
135	010001001010	1	2	2	010010000100	2	2	2	010101000100	2	2	2
136	010001010010	1	2	2	010010100101	2	2	2	010101000101	2	2	2
137	010001010101	1	2	1	010010100100	1	2	1	010000101000	1	2	1
138	010001010100	1	2	2	010000001001	2	2	2	010101001001	2	2	2
139	001010101001	1	2	1	010000100100	1	2	1	100000001000	3	2	1
140	001010000100	1	2	1	100000000000	3	2	1	010101000101	1	2	1
141	010010001001	1	2	1	010000010001	1	2	1	010101000100	1	2	1
142	010010001010	1	2	2	010000010100	2	2	2	100010001001	1	2	1
143	010010001001	1	2	2	010000100010	2	2	2	100000101010	2	2	2
144	010010100101	1	2	1	010000001001	1	2	1	100001001010	2	2	2
145	010010010100	1	2	2	010001010000	2	2	2	100001010101	1	2	1
146	010000001001	1	2	1	010001000001	1	2	1	100001000100	1	2	1
147	001010101010	1	2	2	010000100100	2	2	2	100000001001	1	2	1
148	010000100000	1	2	8	010000101000	8	2	8	010101010010	2	2	2
149	010000010001	1	2	1	010001000101	1	2	1	100001000001	1	2	1
150	010000001010	1	2	2	010001000010	2	2	2	100001010000	2	2	2
151	010000001001	1	2	2	010010100001	2	2	2	100010001010	2	2	2
152	001010000001	1	2	1	010100001010	2	2	1	100010010101	1	2	1
153	010010000001	1	2	1	010100001001	1	2	1	100010010010	1	2	1
154	010010100010	1	2	2	010100001010	2	2	2	100000010001	1	2	1
155	010010100100	1	2	2	010100001000	2	2	2	010101010100	2	2	2
156	010100010101	1	2	1	010100010101	1	2	1	100001010001	1	2	1
157	010010101000	1	2	2	010100100100	2	2	2	100000000001	1	2	1
158	010000100001	1	2	1	010001000001	1	2	1	100000001010	2	2	2
159	010000001000	1	2	1	010001000000	1	2	1	100000010010	2	2	2
160	010101010000	1	2	3	010001000100	3	2	3	010101010101	1	2	1
161	010001000001	1	2	1	010001000001	1	2	1	100010100001	1	2	1
162	010000100010	1	2	2	010001000010	2	2	2	100010100010	2	2	2
163	010000100100	1	2	2	010010000010	2	2	2	100010100100	2	2	2
164	010000001010	1	2	1	010010000010	1	2	1	100010100101	1	2	1
165	010100001010	1	2	2	010100001010	2	2	2	100010101000	2	2	2
166	010010000001	1	2	1	010010010001	1	2	1	100001000001	1	2	1
167	010000101000	1	2	2	010010000100	2	2	2	100000001000	2	2	2
168	010001000010	1	2	2	010010000100	2	2	2	100100000100	1	2	1
169	010000100101	1	2	1	010100000101	1	2	1	100010000001	1	2	1
170	010001000010	1	2	2	010101010010	2	2	2	100000100100	2	2	2
171	010001000001	1	2	1	010100100001	1	2	1	100000100100	2	2	2
172	010100001010	1	2	2	010100100010	2	2	2	100000010101	1	2	1
173	010100010001	1	2	1	010100000001	1	2	1	100100000101	1	2	1
174	010100010010	1	2	2	010100000010	2	2	2	100001010000	1	2	1
175	010100010100	1	2	2	010100000100	2	2	2	100000010000	2	2	2
176	010100100101	1	2	1	010101010101	1	2	1	100001000010	1	2	1
177	010100100010	1	2	2	010100001000	2	2	2	100000100101	1	2	1
178	010010101001	1	2	1	010100000101	1	2	1	100001000100	2	2	2
179	010000100000	1	2	2	010101000010	2	2	2	100100010000	1	2	1
180	010100100100	1	2	2	100100100000	3	2	3	100100010010	2	2	2

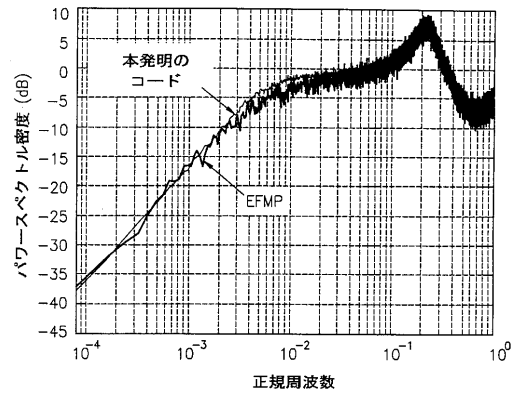
【 図 15 D 】

データ番号	MCG1				MCG2				MCG3			
	コードワード				コードワード				コードワード			
	MSB	LSB	NCG		MSB	LSB	NCG		MSB	LSB	NCG	
181	010100000001	1	2	1	100000101001	1	2	1	100100100001	1	2	1
182	010100000010	1	2	2	100000100010	2	2	2	100100100010	2	2	2
183	010100000100	1	2	2	100000100100	2	2	2	100100100010	2	2	2
184	010001000101	1	2	1	100000100101	1	2	1	100100100101	1	2	1
185	010100010000	1	2	2	100000101000	2	2	2	100100100100	2	2	2
186	010100010001	1	2	2	100000101001	2	2	2	100100000001	1	2	1
187	010000000000	1	2	2	100000000000	2	2	2	100000100000	2	2	2
188	010101010101	1	2	1	100000101001	1	2	1	100100101000	1	2	1
189	010100000101	1	2	2	100000100001	2	2	2	100100100001	1	2	1
190	010100000010	1	2	2	100000101010	2	2	2	100001000010	2	2	2
191	010100100001	1	2	2	100000100101	2	2	2	100001000010	2	2	2
192	010101000010	1	2	2	100000100010	2	2	2	100001000101	1	2	1
193	010101000101	1	2	2	100000100101	2	2	2	100100100000	1	2	1
194	010000000100	1	2	2	100000000100	2	2	2	100100100001	1	2	1
195	010100000000	1	2	2	100000100000	2	2	2	100001000010	2	2	2
196	010100000101	1	2	2	100000100101	2	2	2	100001010010	2	2	2
197	010100010101	1	2	2	100000101001	2	2	2	100001000010	2	2	2
198	010101010101	1	2	2	100000101001	2	2	2	100001010001	1	2	1
199	010101000100	1	2	2	100000100000	2	2	2	100101010001	1	2	1
200	010101000100	1	2	2	100000100000	2	2	2	100101000010	2	2	2
201	010101010010	1	2	2	100000100010	2	2	2	100101000010	2	2	2
202	010101000100	1	2	2	100000100000	2	2	2	100100000100	2	2	2
203	010101010001	1	2	2	100000100100	2	2	2	100100000100	2	2	2
204	010101000101	1	2	2	100000100101	2	2	2	100100000101	1	2	1
205	101000000100	2	2	2	100001000010	2	2	2	100100000100	2	2	2
206	101000000101	2	2	2	100000000101	2	2	2	100100000001	1	2	1
207	101000000101	2	2	2	100000000101	2	2	2	100100000001	2	2	2
208	101000000100	2	2	2	100000000100	2	2	2	100100000100	2	2	2
209	101000010001	2	2	2	100001000001	2	2	2	100100000100	2	2	2
210	101000000010	2	2	2	100000000010	2	2	2	100100000000	2	2	2
211	101010100000	3	2	3	100000001010	3	2	3	100100000001	1	2	1
212	101001000001	1	2	1	100000001001	1	2	1	100100000100	1	2	1
213	101000000100	2	2	2	100001000001	2	2	2	100100000100	2	2	2
214	101000000100	2	2	2	100001000000	2	2	2	100100000100	2	2	2
215	101000000101	1	2	1	100000000101	1	2	1	100100000101	2	2	2
216	101000000101	1	2	1	100001000101	1	2	1	100100000100	1	2	1
217	101000000100	2	2	2	100000000100	2	2	2	100100000100	2	2	2
218	101000010100	2	2	2	100000000010	2	2	2	100100000000	2	2	2
219	101000010010	2	2	2	100000000010	2	2	2	100100000000	2	2	2
220	101000101001	1	2	1	100000							

【 図 1 7 】



【 図 1 9 】



【 図 1 8 】

	d	k	R(m/n)	R/C(d,k)
A-CODE	1	7	0.6533	96.2%
B-CODE	1	8	0.6531	95.3%
C-CODE	1	7	0.6536	96.2%
本発明のコード	1	7	0.6650	97.9%

【 図 2 0 】

	d	k	R(m/n)	R/C(d,k)
EFMP	2	10	0.5	96.2%
本発明のコード	2	10	0.5284	97.9%

フロントページの続き

- (72)発明者 金 珍 漢
大韓民国京畿道水原市八達区梅灘3洞1237-2番地205号
- (72)発明者 丁 奎 海
大韓民国京畿道水原市八達区梅灘4洞214-26番地201号

審査官 高野 洋

- (56)参考文献 国際公開第98/017005(WO, A1)
特開2000-286710(JP, A)
特開2002-135121(JP, A)
特開2000-339871(JP, A)
特開2003-032120(JP, A)

- (58)調査した分野(Int.Cl., DB名)
- | | |
|------|-------|
| H03M | 7/14 |
| H04L | 1/00 |
| H04L | 25/49 |