

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2007-184095  
(P2007-184095A)

(43) 公開日 平成19年7月19日(2007.7.19)

(51) Int. Cl. F I テーマコード (参考)  
**G 1 1 B 20/14 (2006.01)** G 1 1 B 20/14 3 4 1 A 5 D 0 4 4  
**H 0 3 M 7/14 (2006.01)** H 0 3 M 7/14 B

審査請求 有 請求項の数 15 O L (全 33 頁)

<p>(21) 出願番号 特願2007-44997 (P2007-44997)                  (22) 出願日 平成19年2月26日 (2007.2.26)                  (62) 分割の表示 特願平10-315643の分割                  原出願日 平成10年11月6日 (1998.11.6)</p>	<p>(71) 出願人 000002185                  ソニー株式会社                  東京都港区港南1丁目7番1号                  (74) 代理人 100082131                  弁理士 稲本 義雄                  (72) 発明者 中川 俊之                  東京都港区港南1丁目7番1号 ソニー株式会社内                  Fターム(参考) 5D044 BC01 BC02 CC04 GL01 GL02                  GL20 GL21</p>
---	--

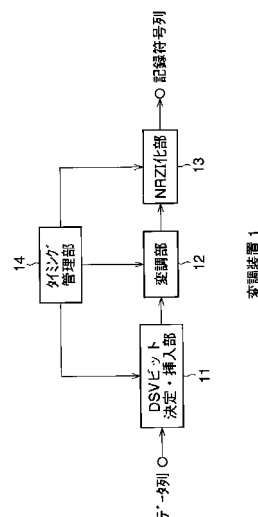
(54) 【発明の名称】 変調装置、変調方法、記録媒体、復調装置、復調方法

(57) 【要約】

【課題】 高線密度で記録再生ができるようにする。

【解決手段】 DSV制御ビット決定・挿入部 11 は、入力されたデータ列にDSV制御のためのDSV制御ビットを挿入し、変調部 12 に出力する。変調部 12 は、変換テーブルに従って、基本データ長が2ビットのデータを、基本符号長が3ビットの可変長符号に変換して、NRZI化部 13 に出力する。変調部 12 が有する変換テーブルは、最小ランの連続を所定の回数以下に制限する置き換えコード、ラン長制限を守るための置き換えコードを有し、前記置き換えコードは、置き換え制限が付け加えられており、さらに、データ列の要素内の「1」の個数を2で割ったときの余りと、符号語列の要素内の「1」の個数を2で割ったときの余りが、どちらも1あるいは0で一致するような変換規則を有する。

【選択図】 図1



## 【特許請求の範囲】

## 【請求項 1】

基本データ長が  $m$  ビットのデータを、変換テーブルに基づいて、基本符号長が  $n$  ビットの可変長符号 ( $d, k; m, n; r$ ) に変換して出力する変調手段を備える変調装置において、

前記変調手段の変換テーブルは、

データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、

最小ラン  $d$  の連続を所定の回数以下に制限する第1の置き換えコードと、

ラン長制限を守るための第2の置き換えコードと、

前記第1の置き換えコードと前記第2の置き換えコードとは異なる基礎コードとを有し

10

、  
前記第1の置き換えコード、または前記第2の置き換えコードには、第1の置き換え制限が加えられており、さらに、前記第1の置き換えコード、または前記第2の置き換えコードの発生回数を抑制するように、第2の置き換え制限が加えられている

ことを特徴とする変調装置。

## 【請求項 2】

前記第2の置き換え制限は、直前または直後の符号語列の少なくとも一方を参照して行う

20

ことを特徴とする請求項1に記載の変調装置。

## 【請求項 3】

前記第2の置き換え制限は、少なくとも直後のデータ列を参照して行う

ことを特徴とする請求項1に記載の変調装置。

## 【請求項 4】

前記第2の置き換え制限は、直後の符号語列または直後のデータ列を参照して行うとともに、さらに、前記直後の符号語列または前記直後のデータ列とは別の、直前の符号語列または直後の符号語列または直後のデータ列のうち少なくとも一つを参照して行う

ことを特徴とする請求項1に記載の変調装置。

## 【請求項 5】

前記入力されたデータの変調符号より得られた符号語列の同期信号を任意の位置に挿入する挿入手段を

30

さらに備えることを特徴とする請求項1に記載の変調装置。

## 【請求項 6】

前記挿入手段は、前記第1の置き換えコードを検出する第1の検出手段と、

前記第2の置き換えコードを検出する第2の検出手段と

をさらに備えることを特徴とする請求項5に記載の変調装置。

## 【請求項 7】

基本データ長が  $m$  ビットのデータを、基本符号長が  $n$  ビットの可変長符号 ( $d, k; m, n; r$ ) に変換する変調装置の変調方法において、

40

入力されたデータを、変換テーブルに従って、符号に変換する変換ステップを含み、

前記変換テーブルの変換コードは、

データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、

最小ラン  $d$  の連続を所定の回数以下に制限する第1の置き換えコードと、

ラン長制限を守るための第2の置き換えコードと、

前記第1の置き換えコードと前記第2の置き換えコードとは異なる基礎コードとを有し

、  
前記第1の置き換えコード、または前記第2の置き換えコードには、第1の置き換え制

50

限が加えられており、さらに、前記第 1 の置き換えコード、または前記第 2 の置き換えコードの発生回数を抑制するように、第 2 の置き換え制限が加えられている

ことを特徴とする変調方法。

【請求項 8】

基本データ長が  $m$  ビットのデータを、変換テーブルに基づいて、基本符号長が  $n$  ビットの可変長符号 ( $d, k; m, n; r$ ) に変換する変調方式により作成されたデータ列であって、

前記変調方式の変換テーブルは、

データ列の要素内の「1」の個数を 2 で割った時の余りと、変換される符号語列の要素内の「1」の個数を 2 で割った時の余りが、どちらも 1 あるいは 0 で一致するような変換規則と、

最小ラン  $d$  の連続を所定の回数以下に制限する第 1 の置き換えコードと、

ラン長制限を守るための第 2 の置き換えコードと、

前記第 1 の置き換えコードと前記第 2 の置き換えコードとは異なる基礎コードとを有し

、  
前記第 1 の置き換えコード、または前記第 2 の置き換えコードには、第 1 の置き換え制限が加えられており、さらに、前記第 1 の置き換えコード、または前記第 2 の置き換えコードの発生回数を抑制するように、第 2 の置き換え制限が加えられている

ことを特徴とするデータ構造を有するデータ列を記録した記録媒体。

【請求項 9】

前記第 2 の置き換え制限は、直前または直後の符号語列の少なくとも一方を参照して行う

ことを特徴とする請求項 8 に記載の記録媒体。

【請求項 10】

前記第 2 の置き換え制限は、少なくとも直後のデータ列を参照して行う

ことを特徴とする請求項 8 に記載の記録媒体。

【請求項 11】

前記第 2 の置き換え制限は、直後の符号語列または直後のデータ列を参照して行うとともに、さらに、前記直後の符号語列または前記直後のデータ列とは別の、直前の符号語列または直後の符号語列または直後のデータ列のうち少なくとも一つを参照して行う

ことを特徴とする請求項 8 に記載の記録媒体。

【請求項 12】

前記入力されたデータの変調符号より得られた符号語列の同期信号を任意の位置に挿入する挿入手段を

さらに備えることを特徴とする請求項 8 に記載の記録媒体。

【請求項 13】

前記挿入手段は、前記第 1 の置き換えコードを検出する第 1 の検出手段と、

前記第 2 の置き換えコードを検出する第 2 の検出手段と

をさらに備えることを特徴とする請求項 12 に記載の記録媒体。

【請求項 14】

基本符号長が  $n$  ビットの可変長符号 ( $d, k; m, n; r$ ) を、基本データ長が  $m$  ビットのデータに変換する復調装置において、

入力された符号を、変換テーブルに従って、データに変換する変換手段を備え、

前記変換テーブルの変換コードは、

データ列の要素内の「1」の個数を 2 で割った時の余りと、変換される符号語列の要素内の「1」の個数を 2 で割った時の余りが、どちらも 1 あるいは 0 で一致するような変換規則と、

最小ラン  $d$  の連続を所定の回数以下に制限する第 1 の置き換えコードと、

ラン長制限を守るための第 2 の置き換えコードと、

前記第 1 の置き換えコードと前記第 2 の置き換えコードとは異なる基礎コードとを有し

10

20

30

40

50

前記第1の置き換えコード、または前記第2の置き換えコードには、第1の置き換え制限が加えられており、さらに、前記第1の置き換えコード、または前記第2の置き換えコードの発生回数を抑制するように、第2の置き換え制限が加えられていることを特徴とする復調装置。

【請求項15】

基本符号長が $n$ ビットの可変長符号( $d, k; m, n; r$ )を、基本データ長が $m$ ビットのデータに変換する復調装置の復調方法において、

入力された符号を、変換テーブルに従って、データに変換する変換ステップを含み、

前記変換テーブルの変換コードは、

データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、

最小ラン $d$ の連続を所定の回数以下に制限する第1の置き換えコードと、

ラン長制限を守るための第2の置き換えコードと、

前記第1の置き換えコードと前記第2の置き換えコードとは異なる基礎コードとを有し

10

前記第1の置き換えコード、または前記第2の置き換えコードには、第1の置き換え制限が加えられており、さらに、前記第1の置き換えコード、または前記第2の置き換えコードの発生回数を抑制するように、第2の置き換え制限が加えられている

ことを特徴とする復調方法。

20

【発明の詳細な説明】

【技術分野】

【0001】

本発明は変調装置、変調方法、記録媒体、復調装置、復調方法に関し、特に、記録媒体にデータを高密度に記録する、あるいは高密度にデータが記録された記録媒体から再生する場合に用いて好適な変調装置、変調方法、記録媒体、復調装置、復調方法に関する。

【背景技術】

【0002】

データを所定の伝送路に伝送したり、または、例えば磁気ディスク、光ディスク、光磁気ディスク等の記録媒体に記録したりする際、伝送路や記録媒体に適するように、データの変調が行われる。このような変調方法の1つとして、ブロック符号が知られている。ブロック符号とは、データ列を $m \times i$ ビットからなる単位(以下データ語という)にブロック化し、このデータ語を適当な符号則に従って、 $n \times i$ ビットからなる符号語に変換するものである。そしてこの符号は、 $i = 1$ のときには固定長符号となり、また $i$ が複数個選べるとき、すなわち、1乃至 $i_{\max}$ (最大の $i$ )の範囲の所定の $i$ を選択して変換したときには可変長符号となる。このブロック符号化された符号は可変長符号( $d, k; m, n; r$ )と表される。

30

【0003】

ここで $i$ は拘束長と称され、 $i_{\max}$ は $r$ (最大拘束長)となる。また最小ラン $d$ は、符号系列内の連続する「1」の間に入る「0」の最小連続個数を示し、最大ラン $k$ は符号系列内の連続する「1」の間に入る「0」の最大連続個数を示している。

40

【0004】

ところで上述のようにして得られる可変長符号を、光ディスクや光磁気ディスク、例えばコンパクトディスク(CD)やミニディスク(MD)(商標)に記録する場合、上述のようにして得られた可変長符号に対して、「1」で反転、「0」で無反転とするNRZI(NonReturn to Zero Inverted)変調を行い、NRZI変調された可変長符号(以下、記録波形列と称する)を記録するようにしている。これをマークエッジ記録ともいう。これに対して、ISO規格の3.5inch・230MB容量の光磁気ディスク等では、記録変調された符号列が、NRZI変調されずに、そのまま記録される。これをマークポジション記録と言う。現在のように高記録密度

50

化された記録メディアでは、マークエッジ記録が用いられる場合が多い。

【0005】

記録波形列の最小反転間隔を  $T_{min}$  とし、最大反転間隔を  $T_{max}$  とするとき、線速方向に高密度記録を行うためには、最小反転間隔  $T_{min}$  は長い方が良く、換言すると、最小ラン  $d$  は大きい方が良い。またクロックの再生の面からは、最大反転間隔  $T_{max}$  は短い方が良く、換言すれば、最大ラン  $k$  は小さい方が良い。オーバーライト特性を考慮する場合、 $T_{max}/T_{min}$  が小さい方が望ましい。さらに、JitterやS/Nの点から、検出窓幅  $T_w=m/n$  が大きいことが重要になるなど、メディア（記録媒体）の条件と照らし合わせながら種々の変調方法が提案され、実用化されている。

【0006】

ここで具体的に、光ディスク、磁気ディスク、または光磁気ディスク等において、提案されたり、あるいは実際に使用されている変調方式について説明する。CDやMDで用いられるEFM (Eight to Fourteen Modulation) 符号  $((2,10;8,17;1))$  とも表記される)、DVD (Digital Video Disc) で用いられる8-16符号  $((2,10;1,2;1))$  とも表記される)、またはPD (Phase Change Optical Disk; 120mm、650MB容量) で用いられるRLL (Run Length Limited code)  $(2,7)((2,7;m,n;r))$  とも表記される)は、最小ラン  $d=2$  のRLL符号である。

【0007】

また、ISO規格の3.5inchMO (Magneto Optical; 640MB容量) で用いられるRLL  $(1,7)((1,7;2,3;r))$  とも表記される) は最小ラン  $d=1$  のRLL符号であり、この他にも、記録密度の高い光ディスクや光磁気ディスク等の記録再生ディスク装置では、最小マークの大きさや、変換効率のバランスの取れた、最小ラン  $d=1$  のRLL符号が用いられている。

【0008】

可変長RLL  $(1-7)$  符号の変換テーブルは、例えば以下のようなテーブルである。

<表1> RLL  $(1,7;2,3;2)$

	データ	符号
$i = 1$	1 1	0 0 x
	1 0	0 1 0
	0 1	1 0 x
$i = 2$	0 0 1 1	0 0 0 0 0 x
	0 0 1 0	0 0 0 0 1 0
	0 0 0 1	1 0 0 0 0 x
	0 0 0 0	1 0 0 0 1 0

【0009】

ここで変換テーブル内の記号  $x$  は、次に続くチャンネルビットが "0" であるときに "1" とされ、次に続くチャンネルビットが "1" であるときに "0" とされる(以下、同様)。最大拘束長  $r$  は2である。

【0010】

可変長RLL  $(1,7)$  のパラメータは  $(1,7;2,3,2)$  であり、記録波形列のビット間隔を  $T$  とすると、 $(d+1)T$  で表される最小反転間隔  $T_{min}$  は  $2(=1+1)T$  となる。データ列のビット間隔を  $T_{data}$  とすると、 $(m/n) \times 2$  で表される最小反転間隔  $T_{min}$  は、 $1.33(=(2/3) \times 2)T_{data}$  となる。また、 $(k+1)T$  で表される最大反転間隔  $T_{max}$  は、 $8(=7+1)T(=(m/n) \times 8T_{data}=(2/3) \times 8T_{data}=5.33T_{data})$  である。さらに検出窓幅  $T_w$  は  $(m/n) \times T_{data}$  で表され、その値は  $0.67(=2/3)T_{data}$  となる。

【0011】

ところで、表1のRLL  $(1,7)$  による変調を行ったチャンネルビット列においては、発生頻度としては  $T_{min}$  である  $2T$  が一番多く、以下  $3T$ 、 $4T$  と続く。 $2T$  や  $3T$  のようなエッジ情報が早い周期で多く発生するのは、クロック再生には有利となる場合が多い。

【0012】

ところが、さらに線記録密度を高くしていくと、今度は逆に、短いマークが問題となる。すなわち、最小ランである  $2T$  が連続して発生し続けると、記録波形に歪みが生じやす

10

20

30

40

50

くなる。なぜならば、2Tの波形出力は他の波形出力よりも小さく、例えばデフォーカスやタンジェンシャル・チルト等による影響を受け易いからである。また高記録密度において、最小マーク(2T)の連続した記録はノイズ等の外乱の影響も受け易くなる。従って、このようなパターン列は、データ再生時に誤りを起こり易くなる。この場合におけるデータ再生誤りのパターンとしては、連続する最小マークの先頭と最後が、シフトして誤るケースが多く、エラー伝搬長が長くなってしまふことになる。

【0013】

一方、記録媒体へのデータの記録、あるいはデータの伝送の際には、その記録媒体あるいは伝送路に適した符号化変調が行われるが、これらの変調符号に低域成分が含まれていると、例えば、ディスク装置のサーボ制御におけるトラッキングエラーなど、各種のエラー信号に変動が生じ易くなったり、あるいはジッタが発生し易くなったりする。従って、変調符号は、低域成分がなるべく抑制されている方が望ましい。

10

【0014】

低域成分を抑制する方法として、DSV(Digital Sum Value)制御がある。DSVとは、チャンネルビット列をNRZI化(レベル符号化)して記録符号列とし、そのビット列(データのシンボル)の"1"を「+1」、「0」を「-1」として、符号を加算していったときの総和を意味する。DSVは記録符号列の低域成分の目安となり、DSVの正負のゆれの絶対値を小さくすること、すなわち、DSV制御を行うことは、記録符号列の直流成分を除き、低域成分を抑制することになる。

【0015】

表1に示した、可変長RLL(1,7)テーブルによる変調符号は、DSV制御が行われていない。このような場合のDSV制御は、変調後の符号化列(チャンネルビット列)において、所定の間隔でDSV計算を行い、所定のDSV制御ビットを符号化列(チャンネルビット列)内に挿入することで実現される。

20

【0016】

しかしながら、DSV制御ビットは冗長ビットであるから、符号変換の効率から考えれば、DSV制御ビットはなるべく少ない方が良く、また、挿入されるDSV制御ビットによって、最小ランクdおよび最大ランクkは、変化しない方が良い。これは、(d,k)が変化すると、記録再生特性に影響を及ぼしてしまうからである。

【0017】

そこで本出願人は、例えば、先に出願した特願平10-150280号において、(d,k)=(1,7)で、さらに高記録密度に対応した変調方式として、表2の1,7PP符号を提案している。

30

<表2> 1,7PP(1,7;2,3;4)

データ	符号
11	*0*
10	001
01	010
0011	010 100
0010	010 000
0001	000 100
000011	000 100 100
000010	000 100 000
000001	010 100 100
000000	010 100 000
"110111	001 000 000(next 010)
00001000	000 100 100 100
00000000	010 100 100 100
if xx1 then	*0* = 000
xx0 then	*0* = 101

40

50

-----  
Sync & Termination

#01 000 000 001 (12 channel bits)

or

#01 001 000 000 001 000 000 001 (24 channel bits)

# = 0 not terminate case

# = 1 terminate case  
-----

## Termination table

00 000

0000 010 100

"110111 001 000 000(next010):

When next channel bits are '010',

convert '11 01 11' to '001 000 000' after

using main table and termination table.  
-----

10

## 【 0 0 1 8 】

表 2 の変換テーブルは、変換コードとして、それがないと変換処理ができない基礎コード（データ列（1 1）から（0 0 0 0 0 0）までのコード）、それがなくても変換処理は可能であるが、それがあると、より効果的な変換処理が可能となる置き換えコード（データ列（1 1 0 1 1 1）、（0 0 0 0 1 0 0 0）、（0 0 0 0 0 0 0 0）のコード）、および、符号を任意の位置で終端させるための終端コード（データ列（0 0）、（0 0 0 0）のコード）を含んでいる。

20

## 【 0 0 1 9 】

また、表 2 は、最小ラン d = 1、最大ラン k = 7 で、基礎コードの要素に不確定符号（\* を含む符号）を含んでいる。不確定符号は、直前および直後の符号語列の如何によらず、最小ラン d と最大ラン k を守るように、“0”が“1”に決定される。すなわち、表 2 において、変換する 2 ビットのデータ列が（1 1）であったとき、その直前の符号語列によって、“0 0 0”または“1 0 1”が選択され、そのいずれかに変換される。例えば、直前の符号語列の 1 チャネルビットが“1”である場合、最小ラン d を守るために、2 ビットのデータ（1 1）は、符号語“0 0 0”に変換され、直前の符号語列の 1 チャネルビットが“0”である場合、最大ラン k が守られるように、符号語“1 0 1”に変換される。

30

## 【 0 0 2 0 】

表 2 の変換テーブルの基礎コードは可変長構造を有している。すなわち、拘束長 i = 1 における基礎コードは、必要数の 4 つ（ $2^m = 2^2 = 4$ ）よりも少ない 3 つ（\* 0 \* , 0 0 1 , 0 1 0）で構成されている。その結果、データ列を変換する際に、拘束長 i = 1 だけでは変換出来ないデータ列が存在することになる。結局、表 2 において、全てのデータ列を変換するには（変換テーブルとして成り立つためには）、拘束長 i = 3 までの基礎コードを参照する必要がある。

40

## 【 0 0 2 1 】

また、表 2 の変換テーブルは、最小ラン d の連続を制限する置き換えコードを持っているため、データ列が（1 1 0 1 1 1）である場合、さらに後ろに続く符号語列が参照され、それが“0 1 0”であるとき、このデータ列は符号語“0 0 1 0 0 0 0 0 0”に置き換えられる。また、このデータ列は、後ろに続く符号語列が“0 1 0”以外である場合、2 ビット単位（（1 1）、（0 1）、（1 1））で符号語に変換されるので、符号語“\* 0 \* 0 1 0 \* 0 \*”に変換される。これによって、データを変換した符号語列は、最小ランの連続が制限され、最大でも 6 回までの最小ラン繰り返しとなる。

## 【 0 0 2 2 】

さらに表 2 の変換テーブルは、最大拘束長 r = 4 である。拘束長 i = 4 のコードは、最

50

大ランク = 7 を実現するための、置き換えコード (最大ラン補償コード) で構成されている。すなわち、データ (00001000) は、符号語 "000100100100" に変換され、データ (00000000) は、符号語 "010100100100" に変換されるようになされている。そして、この場合においても、最小ランド = 1 は守られている。

#### 【0023】

ところで、表 2 の変換コードは、データ列の要素内の "1" の個数を 2 で割った時の余りと、変換される符号語列の要素内の "1" の個数を 2 で割った時の余りが、どちらも 1 あるいは 0 で同一 (対応するいずれの要素も、"1" の個数が奇数または偶数) となるような変換規則を持っている。例えば、変換コードのうちのデータ列の要素 (000001) は、"010100100" の符号語列の要素に対応しているが、それぞれの要素の "1" の個数は、データ列では 1 個、対応する符号語列では 3 個であり、どちらも 2 で割ったときの余りが 1 (奇数) で一致している。同様に、変換コードのうちのデータ列の要素 (000000) は、"010100000" の符号語列の要素に対応しているが、それぞれ "1" の個数は、データ列では 0 個、対応する符号語列では 2 個であり、どちらも 2 で割ったときの余りが 0 (偶数) で一致している。

10

#### 【0024】

次に、DSV制御を行う方法について述べる。表 1 の RLL(1,7) 符号のような、変換テーブルに DSV制御が行われていない場合における、従来の DSV制御の方法の一例としては、データ列を変調した後、変調後のチャンネルビット列に、所定の間隔で、DSV制御ビットを少なくとも (d + 1) ビットだけ付加することで行われた。

20

#### 【0025】

表 2 のような変換テーブルにおいては、データ列と、変換される符号語列の関係を生かし、効率良く DSV制御を行うことができる。即ち、変換テーブルが、データ列の要素内の "1" の個数と、変換される符号語列の要素内の "1" の個数を、2 で割った時の余りが、どちらも 1 あるいは 0 で同一となるような変換規則を持っている時、上記のようにチャンネルビット列内に、「反転」を表す "1"、あるいは「非反転」を表す "0" の DSV制御ビットを挿入することは、データビット列内に、「反転」するならば (1) の、「非反転」ならば (0) の、それぞれ DSV制御ビットを挿入することと等価となる。

#### 【0026】

例えば、表 2 において、データ変換する 3 ビットが (001) と続いたときに、その後ろにおいて DSV制御ビットを挟むものとする、データは、(001 - x) (x は 1 ビットで、「0」又は「1」) となる。ここで x に「0」を与えれば、表 2 の変換テーブルで、

30

データ列	符号語列
0010	010 000

の変換が行われ、また、「1」を与えれば、

データ列	符号語列
0011	010 100

の変換が行われる。符号語列を NRZI 化して、レベル符号化したとき、これらは

40

データ列	符号語列	レベル符号列
0010	010 000	011111
0011	010 100	011000

となり、レベル符号列の最後の 3 ビットが相互に反転している。このことは、DSV制御ビット x の (1) と (0) を選択することによって、データ列内においても、DSV制御が行えることを意味する。

#### 【0027】

DSV制御による冗長度を考えると、データ列内の 1 ビットで DSV制御を行うということは、チャンネルビット列で表現すれば、表 2 の変換率 (m/n=2/3) より、1.5 チャンネルビットで DSV制御を行っていることに相当する。一方、表 1 のような RLL(1,7) テーブルにおいて DSV制

50



御を行うためには、チャンネルビット列においてDSV制御を行う必要があるが、この時、最小ランを守るためには、少なくとも2チャンネルビットが必要であり、表2のDSV制御と比較すると、冗長度がより大きくなってしまふ。換言すれば、表2のテーブル構造を持つ時、データ列内でDSV制御を行うことで、効率よくDSV制御を行うことができる。

【0028】

ところで一般的に、最大拘束長 $r$ が大きいほど、ビットシフト時の復調エラー（エッジビットの位置が1ビット分だけ、正規の位置よりも前方または後方にシフトすることによるエラー）の伝搬特性が悪くなる。

【0029】

表1と表2を比較すると、表1のRLL(1,7)符号では最大拘束長 $r$ は2であるのに対して、表2の1,7PP符号では最大拘束長 $r$ は4と大きく、ビットシフトに対する復調エラー伝搬の最悪伝搬長は、表1では2バイトであるが、表2では3バイトである。1,7PP符号は、高記録密度に対応した $(d,k)=(1,7)$ 符号であり、コンパクトな構成であるが、それでもエラー伝搬特性については従来のRLL(1,7)符号より不利であった。

10

【特許文献1】特開平11-346154号公報

【発明の開示】

【発明が解決しようとする課題】

【0030】

以上のように、RLL符号を高線密度にディスクに記録再生する場合、最小ラン $d$ の連続したパターンがあると、長いエラーが発生し易かった。また、 $(1,7;2,3)$ 符号においてDSV制御を行うには冗長ビットを挟む必要があったが、なるべくこの冗長ビットを少なくする必要性があった。このような状況を元に、上述したように最小ラン $d=1$ であるRLL符号 $(d,k;m,n)=(1,7;2,3)$ において、最小ランの連続する回数を制限し、さらに最小ラン及び最大ランを守りながら、効率の良い制御ビットで、DSV制御を行うことができる1,7PP符号が開発されが、その1,7PP符号は簡単な構造の変換テーブルを持つにもかかわらず、従来RLL(1,7)符号よりも長いエラー伝搬特性を持っていた。

20

【0031】

本発明はこのような状況に鑑みてなされたものであり、長いエラー伝搬を発生しやすい変換コードの発生回数を制限するテーブルを用い、さらに長いエラー伝搬を発生しやすいようなパターンの発生する回数を制限するテーブルを用いることにより、長いエラー伝搬を発生させないようにするものである。

30

【課題を解決するための手段】

【0032】

本発明の一側面の変調装置は、基本データ長が $m$ ビットのデータを、変換テーブルに基づいて、基本符号長が $n$ ビットの可変長符号 $(d,k;m,n;r)$ に変換して出力する変調手段を備える変調装置において、前記変調手段の変換テーブルは、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ラン $d$ の連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードと、前記第1の置き換えコードと前記第2の置き換えコードとは異なる基礎コードとを有し、前記第1の置き換えコード、または前記第2の置き換えコードには、第1の置き換え制限が加えられており、さらに、前記第1の置き換えコード、または前記第2の置き換えコードの発生回数を抑制するように、第2の置き換え制限が加えられていることを特徴とする。

40

【0033】

前記第2の置き換え制限は、直前または直後の符号語列の少なくとも一方を参照して行うようにすることができる。

【0034】

前記第2の置き換え制限は、少なくとも直後のデータ列を参照して行うようにすることができる。

50

## 【0035】

前記第2の置き換え制限は、直後の符号語列または直後のデータ列を参照して行うとともに、さらに、前記直後の符号語列または前記直後のデータ列とは別の、直前の符号語列または直後の符号語列または直後のデータ列のうち少なくとも一つを参照して行うようにすることができる。

## 【0036】

前記入力されたデータの変調符号より得られた符号語列の同期信号を任意の位置に挿入する挿入手段をさらに備えるようにすることができる。

## 【0037】

前記挿入手段は、前記第1の置き換えコードを検出する第1の検出手段と、前記第2の置き換えコードを検出する第2の検出手段とをさらに備えるようにすることができる。

## 【0038】

本発明の一側面の変調方法は、基本データ長が $m$ ビットのデータを、基本符号長が $n$ ビットの可変長符号( $d, k; m, n; r$ )に変換する変調装置の変調方法であって、入力されたデータを、変換テーブルに従って、符号に変換する変換ステップを含み、前記変換テーブルの変換コードは、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ラン $d$ の連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードと、前記第1の置き換えコードと前記第2の置き換えコードとは異なる基礎コードとを有し、前記第1の置き換えコード、または前記第2の置き換えコードには、第1の置き換え制限が加えられており、さらに、前記第1の置き換えコード、または前記第2の置き換えコードの発生回数を抑制するように、第2の置き換え制限が加えられていることを特徴とする。

## 【0039】

本発明の一側面の変調装置および方法においては、基本データ長が $m$ ビットのデータが、変換テーブルに基づいて、基本符号長が $n$ ビットの可変長符号( $d, k; m, n; r$ )に変換され、その変調処理が行われるときの変換テーブルは、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ラン $d$ の連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードと、第1の置き換えコードと第2の置き換えコードとは異なる基礎コードとを有し、第1の置き換えコード、または第2の置き換えコードには、第1の置き換え制限が加えられており、さらに、第1の置き換えコード、または第2の置き換えコードの発生回数を抑制するように、第2の置き換え制限が加えられている。

## 【0040】

本発明の一側面の記録媒体は、基本データ長が $m$ ビットのデータを、変換テーブルに基づいて、基本符号長が $n$ ビットの可変長符号( $d, k; m, n; r$ )に変換する変調方式により作成されたデータ列であって、前記変調方式の変換テーブルは、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ラン $d$ の連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードと、前記第1の置き換えコードと前記第2の置き換えコードとは異なる基礎コードとを有し、前記第1の置き換えコード、または前記第2の置き換えコードには、第1の置き換え制限が加えられており、さらに、前記第1の置き換えコード、または前記第2の置き換えコードの発生回数を抑制するように、第2の置き換え制限が加えられていることを特徴とするデータ構造を有するデータ列を記録している。

## 【0041】

前記第2の置き換え制限は、直前または直後の符号語列の少なくとも一方を参照して行うようにすることができる。

## 【0042】

前記第2の置き換え制限は、少なくとも直後のデータ列を参照して行うようにすることができる。

【0043】

前記第2の置き換え制限は、直後の符号語列または直後のデータ列を参照して行うとともに、さらに、前記直後の符号語列または前記直後のデータ列とは別の、直前の符号語列または直後の符号語列または直後のデータ列のうち少なくとも一つを参照して行うようにすることができる。

【0044】

前記入力されたデータの変調符号より得られた符号語列の同期信号を任意の位置に挿入する挿入手段をさらに備えるようにすることができる。

【0045】

前記挿入手段は、前記第1の置き換えコードを検出する第1の検出手段と、前記第2の置き換えコードを検出する第2の検出手段とをさらに備えるようにすることができる。

【0046】

本発明の一側面の記録媒体においては、基本データ長が $m$ ビットのデータを、変換テーブルに基づいて、基本符号長が $n$ ビットの可変長符号( $d, k; m, n; r$ )に変換する変調方式により作成されたデータ列が記録されており、その変調方式の変換テーブルは、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ラン $d$ の連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードと、第1の置き換えコードと第2の置き換えコードとは異なる基礎コードとを有し、第1の置き換えコード、または第2の置き換えコードには、第1の置き換え制限が加えられており、さらに、第1の置き換えコード、または第2の置き換えコードの発生回数を抑制するように、第2の置き換え制限が加えられている。

【0047】

本発明の一側面の復調装置は、基本符号長が $n$ ビットの可変長符号( $d, k; m, n; r$ )を、基本データ長が $m$ ビットのデータに変換する復調装置であって、入力された符号を、変換テーブルに従って、データに変換する変換手段を備え、前記変換テーブルの変換コードは、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ラン $d$ の連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードと、前記第1の置き換えコードと前記第2の置き換えコードとは異なる基礎コードとを有し、前記第1の置き換えコード、または前記第2の置き換えコードには、第1の置き換え制限が加えられており、さらに、前記第1の置き換えコード、または前記第2の置き換えコードの発生回数を抑制するように、第2の置き換え制限が加えられていることを特徴とする。

【0048】

本発明の一側面の復調方法は、基本符号長が $n$ ビットの可変長符号( $d, k; m, n; r$ )を、基本データ長が $m$ ビットのデータに変換する復調装置の復調方法であって、入力された符号を、変換テーブルに従って、データに変換する変換ステップを含み、前記変換テーブルの変換コードは、データ列の要素内の「1」の個数を2で割った時の余りと、変換される符号語列の要素内の「1」の個数を2で割った時の余りが、どちらも1あるいは0で一致するような変換規則と、最小ラン $d$ の連続を所定の回数以下に制限する第1の置き換えコードと、ラン長制限を守るための第2の置き換えコードと、前記第1の置き換えコードと前記第2の置き換えコードとは異なる基礎コードとを有し、前記第1の置き換えコード、または前記第2の置き換えコードには、第1の置き換え制限が加えられており、さらに、前記第1の置き換えコード、または前記第2の置き換えコードの発生回数を抑制するように、第2の置き換え制限が加えられていることを特徴とする。

【0049】

本発明の一側面の復調装置および方法は、基本符号長が $n$ ビットの可変長符号( $d, k$

10

20

30

40

50

; m , n ; r ) が、基本データ長が m ビットのデータに変換され、入力された符号が、変換テーブルに従って、データに変換されるときの変換コードは、データ列の要素内の「 1 」の個数を 2 で割った時の余りと、変換される符号語列の要素内の「 1 」の個数を 2 で割った時の余りが、どちらも 1 あるいは 0 で一致するような変換規則と、最小ラン d の連続を所定の回数以下に制限する第 1 の置き換えコードと、ラン長制限を守るための第 2 の置き換えコードと、前記第 1 の置き換えコードと前記第 2 の置き換えコードとは異なる基礎コードとを有し、前記第 1 の置き換えコード、または前記第 2 の置き換えコードには、第 1 の置き換え制限が加えられており、さらに、前記第 1 の置き換えコード、または前記第 2 の置き換えコードの発生回数を抑制するように、第 2 の置き換え制限が加えられている。

10

【発明の効果】

【0050】

本発明によれば、高線密度でエラーの少ない符号語列を記録再生することが可能となる。

【発明を実施するための最良の形態】

【0051】

本発明の実施の形態について説明するが、以下においては、説明の便宜上、変換される前のデータの「 0 」と「 1 」の並び（変換前のデータ列）を、( 0 0 0 0 1 1 ) のように、( ) で区切って表し、変換された後の「 0 」と「 1 」の並び（符号語列）を、" 0 0 0 1 0 0 1 0 0 " のように、" " で区切って表す。

20

【0052】

以下の表は、本発明のデータを符号に変換する変換テーブルの例を表している。

< 表 3 > 1,7PP-32 (plus pre0)

データ列	符号列
11	*0*
10	001
01	010
0011	010 100
0010	010 000
0001	000 100
000011	000 100 100
000010	000 100 000
000001	010 100 100
000000	010 100 000

30

```
#110111 (pre0)001 000 000(next010)
#00001000 000 100 100 100
#00000000 010 100 100 100
```

```
if xx1 then *0* = 000
  xx0 then *0* = 101
```

40

```
*1. # "0"+110111 + 01/001/00000 001 000 000
    # "0" + 110111 + "010" 001 000 000
    # "101-010-101" + "010" 001 000 000
```

【0053】

表 3 の変換テーブルは、1,7PP符号であり、さらに、表 2 と比較すると最小ランの連続を制限する変換コードの発生回数を抑制するように、直前及び直後の符号語列を参照することで、置き換え制限を行うものである。

【0054】

50

表 3 は、最小ラン d = 1、最大ラン k = 7 で、基礎コードの要素に不確定符号 (\* を含む符号) を持つ。不確定符号は、直前および直後の符号語列の如何によらず、最小ラン d と最大ラン k を守るように、" 0 " が " 1 " に決定される。すなわち、表 3 において、変換する 2 ビットのデータ列が ( 1 1 ) であったとき、その直前の符号語列によって、" 0 0 0 " または " 1 0 1 " が選択され、そのどちらかに変換される。例えば、直前の符号語列の 1 チャネルビットが " 1 " である場合、最小ラン d を守るために、2 ビットのデータ ( 1 1 ) は、符号語 " 0 0 0 " に変換され、直前の符号語列の 1 チャネルビットが " 0 " である場合、最大ラン k が守られるように、符号語 " 1 0 1 " に変換される。

#### 【 0 0 5 5 】

表 3 の変換テーブルは、可変長構造を有しており、その基礎コードは  $i = 1$  乃至 3 まで持つ。また、表 3 の変換テーブルは、最小ラン d の連続を制限する置き換えコードを持っているため、データ列が ( 1 1 0 1 1 1 ) である場合、さらに直前と直後の符号語が参照される。直前の符号語が、" 0 " であり、かつ、後ろに続く符号語列が " 0 1 0 " であった時、このデータ列は、符号語 " 0 0 1 0 0 0 0 0 0 " に置き換えられる。またこのデータ列は、直前と直後の符号語列が上記で示した場合でないとき、2 ビット単位 ( ( 1 1 ) , ( 0 1 ) , ( 1 1 ) ) で符号語に変換され、符号語 " \* 0 \* 0 1 0 \* 0 \* " に変換される。これにより、データを変換した符号語列は、最小ランの連続が制限され、最大でも 6 回までの最小ランの繰り返しとなる。

#### 【 0 0 5 6 】

表 3 の最小ラン d の連続を制限する置き換えコードは、直前の符号語列と、直後の符号語列の両方を参照して決定される。このうち、直前の符号語列は、変換の際に参照されなくても最小ラン d の連続は 6 回までに制限されるが、表 3 のようにされることにより、置き換えコードの発生回数を減らすことができる。

#### 【 0 0 5 7 】

そして、表 3 に示した変換コードでは、最大拘束長  $r = 4$  である。拘束長  $i = 4$  のコードは、最大ラン k = 7 を実現するための置き換えコード (最大ラン補償コード) で構成されている。すなわち、データ ( 0 0 0 0 1 0 0 0 ) は符号語 " 0 0 0 1 0 0 1 0 0 1 0 0 " に変換され、データ ( 0 0 0 0 0 0 0 0 ) は符号語 " 0 1 0 1 0 0 1 0 0 1 0 0 " に変換される。なお、この場合にも最小ラン d = 1 は守られている。

#### 【 0 0 5 8 】

また、表 3 の変換コードは、データ列の要素内の " 1 " の個数を 2 で割った時の余りと、変換される符号語列の要素内の " 1 " の個数を 2 で割った時の余りが、どちらも 1 あるいは 0 で同一 (対応するいずれの要素も、" 1 " の個数が奇数または偶数) となるような変換規則を持っている。例えば、変換コードのうちのデータ列の要素 ( 0 0 0 0 0 1 ) は、" 0 1 0 1 0 0 1 0 0 " の符号語列の要素に対応しているが、それぞれの要素の " 1 " の個数は、データ列では 1 個、対応する符号語列では 3 個であり、どちらも 2 で割ったときの余りが 1 (奇数) で一致している。同様にして、変換コードのうちのデータ列の要素 ( 0 0 0 0 0 0 ) は、" 0 1 0 1 0 0 0 0 0 " の符号語列の要素に対応しているが、それぞれ " 1 " の個数は、データ列では 0 個、対応する符号語列では 2 個であり、どちらも 2 で割ったときの余りが 0 (偶数) で一致している。

#### 【 0 0 5 9 】

< 表 4 > 1,7PP-32 (plus  $i=4$ )

データ列	符号列
11	*0*
10	001
01	010
0011	010 100
0010	010 000
0001	000 100
000011	000 100 100

10

20

30

40

50

```

000010      000 100 000
000001      010 100 100
000000      010 100 000

```

```

-----
#110111      001 000 000(next010)
#00001000-x  000 100 100 100
#00000000-x  010 100 100 100
# x = 01, 001

```

```

-----
if xx1 then *0* = 000
   xx0 then *0* = 101

```

10

```

-----
*1. # 110111 + 01/001/00000  001 000 000
    # 110111 + "010"        001 000 000
    # "*0*-010-101" + "010"  001 000 000
*2. # 000010 + 0001/00001    000 100 100 100
    # "000-100-000" + "000"  000 100 100 100
    # 000000 + 0001/00001    010 100 100 100
    # "010-100-000" + "000"  010 100 100 100

```

#### 【 0 0 6 0 】

20

表 4 に示した変換テーブルは、1,7PP符号であり、さらに、表 2 と比較すると最大ランを補償する変換コードの発生回数を抑制するように、直後の符号語列（データ列）を参照することで、置き換え制限を行っている。

#### 【 0 0 6 1 】

表 4 は、最小ラン  $d = 1$ 、最大ラン  $k = 7$  で、基礎コードの要素に不確定符号（\*を含む符号）を持つ。また、表 4 の変換テーブルは可変長構造を有しているので、基礎コードは  $i = 1$  乃至 3 までを持つ。また、表 4 の変換テーブルは、最小ラン  $d$  の連続を制限する置き換えコードを持っているため、データ列が（1 1 0 1 1 1）である場合、後ろに続く符号語列を参照し、それが " 0 1 0 " であった時、このデータ列は符号語 " 0 0 1 0 0 0 0 0 0 " に置き換えられる。またこのデータ列は、直後の符号語列が上記でない場合、2 ビット単位（（1 1）,（0 1）,（1 1））で符号語に変換されるので、符号語 " \* 0 \* 0 1 0 \* 0 \* " に変換される。これによって、データを変換した符号語列は、最小ランの連続が制限され、最大でも 6 回までの最小ランの繰り返しとなる。

30

#### 【 0 0 6 2 】

そして、表 4 の変換テーブルにおいては、最大拘束長  $r = 4$  である。拘束長  $i = 4$  のコードは、最大ラン  $k = 7$  を実現するための、置き換えコード（最大ラン補償コード）で構成されている。すなわち、データ（0 0 0 0 1 0 0 0）であったとき、さらに、後ろに続くデータ列を参照し、それが（0 1）又は（0 0 1）であった時、このデータ列は、符号語 " 0 0 0 1 0 0 1 0 0 1 0 0 " に変換される。またデータ（0 0 0 0 0 0 0 0）は、さらに、後ろに続くデータ列が（0 1）又は（0 0 1）であった時、符号語 " 0 1 0 1 0 0 1 0 0 1 0 0 " に変換される。なお、この場合にも、最小ラン  $d = 1$  は守られている。

40

#### 【 0 0 6 3 】

このことを換言すると、拘束長  $i = 4$  のコードは、最大ラン  $k = 7$  を実現するための、置き換えコード（最大ラン補償コード）は、データの 6 ビットが（0 0 0 0 1 0）であり、さらに、後ろに続く符号語列が、" 0 0 0 " であった場合、このデータ列は、符号語 " 0 0 0 1 0 0 1 0 0 1 0 0 " に変換される。また、データの 6 ビットが（0 0 0 0 0 0）であり、さらに、後ろに続く符号語列が " 0 0 0 " であった場合、符号語 " 0 1 0 1 0 0 1 0 0 1 0 0 " に変換される。

#### 【 0 0 6 4 】

表 4 の、最大ラン  $k = 7$  を実現する置き換えコードは、データ列 8 ビットに加え、最大

50

で3ビット多く見て、合計データ列11ビットが参照されて決定される。変換の際に、データ列8ビットまで参照するだけでも、最大ランク = 7は実現するが、表4に示したようにすることにより、置き換えコードの発生回数を減らすことが可能となる。

【0065】

また、表4の変換コードは、表3と同様に、データ列の要素内の"1"の個数を2で割った時の余りと、変換される符号語列の要素内の"1"の個数を2で割った時の余りが、どちらも1あるいは0で同一となるような変換規則を持っている。

【0066】

<表5> 1,7PP-32 (plus \*0\*)

データ列	符号列	
11	*0*	10
10	001	
01	010	
0011	010 100	
0010	010 000	
0001	000 100	
000011	000 100 100	
000010	000 100 000	
000001	010 100 100	
000000	010 100 000	20

---

#110111	001 000 000(next010/*0*)
#00001000	000 100 100 100
#00000000	010 100 100 100

---

```

if xx1 then *0* = 000
  xx0 then *0* = 101

```

---

*1. # 110111 + 01/001/00000	001 000 000	
# 110111 + "010"	001 000 000	30
# "*0*-010-101" + "010"	001 000 000	

---

*2. # 110111 + 11/not(110111-01/00/11)	001 000 000	
# 110111 + "*0*"	001 000 000	
# "*0*-010-101" + "*0*"	001 000 000	

【0067】

表5の変換テーブルは、1,7PP符号でありさらに、表2と比較すると最小ランの最も繰り返される、6回連続の発生する回数を抑制するように、2種類の直後の符号語列を参照することで、置き換え制限が行われている。

【0068】

表5は、最小ランd = 1、最大ランク = 7で、基礎コードの要素に不確定符号(\*を含む符号)をもち、可変長構造を有しているため、基礎コードはi = 1乃至3までを持つ。また、表5の変換テーブルは、表3や表4と同様に、最小ランdの連続を制限する置き換えコードを持っているため、データ列が(110111)である場合、後ろに続く符号語列を参照し、それが"010"又は"\*0\*"であった時、このデータ列は符号語"001000000"に置き換えられる。また、このデータ列は、上の2つの条件以外である場合、2ビット単位((11),(01),(11))で符号語に変換されるので、符号語"\*0\*010\*0\*"に変換される。これによって、データを変換した符号語列は、最小ランの連続が制限され、最大でも6回までの最小ランの繰り返しとなる。

【0069】

10

20

30

40

50

表 5 の最小ラン d の連続を制限する置き換えコードは、2 種類の直後の符号語列を参照して決定される。参照される 2 つの直後の符号語列のうち、片方は、変換の際に参照しなくても最小ラン d の連続は 6 回までに制限されるが、表 5 のような変換テーブルとすることにより、最小ランの 6 回連続する発生回数を減らすことが可能となる。

【 0 0 7 0 】

そして、表 5 の変化コードは、表 3 や表 4 と同様に、最大拘束長  $r = 4$  である。拘束長  $i = 4$  のコードは、最大ラン  $k = 7$  を実現するための置き換えコード（最大ラン補償コード）で構成されている。また、表 5 の変換コードは、表 3 や表 4 と同様にデータ列の要素内の " 1 " の個数を 2 で割った時の余りと、変換される符号語列の要素内の " 1 " の個数を 2 で割った時の余りが、どちらも 1 あるいは 0 で同一となるような変換規則を持っている。

10

【 0 0 7 1 】

一般的に、最大拘束長  $r$  が大きいほど、ビットシフト（再生時にエッジビットの位置が 1 ビット分だけ、正規の位置よりも前方または後方にシフトする）時の復調エラーの伝搬特性が悪くなる。これにより、拘束長の大きいデータ変換の部分が、長い復調エラー伝搬を発生させている場合が考えられる。すなわち、拘束長の大きい変換の発生回数を少なくすることは、エラーの伝搬特性が向上することになる。従って、表 2 にある 1,7PP 符号は、表 3 や表 4 にあるような構造とすることにより、エラー伝搬特性の向上が可能となる。また表 5 のように、最小ランの最も多く連続する、6 回の発生回数を減らすことによっても、長いエラー伝搬を減らすことになり、もって、エラー伝搬特性の向上が可能となる。

20

【 0 0 7 2 】

表 3 乃至表 5 はそれぞれ独立であるから、これらを複合させたテーブルを構成させることが出来る。例えば、以下に示す表 6 は、表 3 と表 4 を組み合わせたものであり、拘束長の大きい変換である置き換えコードの発生回数を、さらに減らすことが出来るような変換コードを 2 つ持ったテーブルである。

【 0 0 7 3 】

< 表 6 > 1,7PP-32 (plus  $i=4$ ,  $pre0$ )

データ列	符号列
11	*0*
10	001
01	010
0011	010 100
0010	010 000
0001	000 100
000011	000 100 100
000010	000 100 000
000001	010 100 100
000000	010 100 000

30

```

-----
#110111 (pre0)001 000 000(next010)
#00001000-x 000 100 100 100
#00000000-x 010 100 100 100
# x = 01, 001

```

40

```

-----
if xx1 then *0* = 000
  xx0 then *0* = 101

```

```

-----
*1. # "0"+110111 + 01/001/00000 001 000 000
    # "0" + 110111 + "010" 001 000 000
    # "101-010-101" + "010" 001 000 000

```

50



```

-----
*2. # 000010 + 0001/00001      000 100 100 100
     # "000-100-000" + "000"    000 100 100 100
     # 000000 + 0001/00001      010 100 100 100
     # "010-100-000" + "000"    010 100 100 100

```

## 【 0 0 7 4 】

ところで、上記表 3 乃至表 6 の変換テーブルによって発生された符号語列（チャンネルビット列）中の、任意の位置に同期信号を挿入する場合、この変換テーブルは可変長構造を有しているため、任意の位置で符号を終端させるために終端用テーブルが規定され、必要に応じて用いられるようになされている。

10

## 【 0 0 7 5 】

以下の表 7 は、本発明の表 3 乃至表 6 における、同期信号及び終端テーブルの一例を示している。

< 表 7 > Sync & Termination for 表 3 乃至表 6

```

#01 000 000 001 (12 channtl bits)
# = 0 not terminate case
# = 1 terminate case

```

-----  
Termination table

```

00    000
0000  010 100

```

20

## 【 0 0 7 6 】

例えば、任意の位置で同期信号を挿入する際、まず直前と直後の符号語列との接続において、最小ランク及び最大ランクが守られるように接続ビットが設定され、接続ビットの間に同期信号用のユニークなパターンが設定される。同期信号パターンとして、最大ランク = 7 を破るパターンを与えたとすると、例えば、" # 0 1 0 0 0 0 0 0 0 0 1 " となる。この同期信号パターンの先頭の " # " は接続用ビットで、" 0 " か " 1 " のどちらかに設定される。" # " の次の第 2 チャンネルビット目は、最小ランクを守るために " 0 " に設定される。第 3 チャンネルビット目から、同期信号パターンとして、 $k = 8$  となる 9 T のユニークなパターンが与えられる。すなわち、" 1 " と " 1 " の間に、" 0 " が 8 個連続して並ぶ。

30

## 【 0 0 7 7 】

そして表 3 乃至表 6 に適用できる終端用テーブルは、表 2 と同様に実現出来き、その終端用テーブルは表 7 のように、

```

00    000
0000  010 100

```

となる。終端用テーブルが必要になるのは、データ列と符号語列の対の数が 4 つ ( $2^m = 2^2 = 4$ ) よりも少ない拘束長  $i$  の基礎コードに対してである。

## 【 0 0 7 8 】

この終端用テーブルにより、データ (00) は符号 " 0 0 0 " に変換され、データ (0000) は符号 " 0 1 0 1 0 0 " に変換される。これにより、同期信号を挿入する際、その直前のデータを符号に変換することができなくなる（同期信号の直前までの符号を終端させることができなくなる）ことを防ぐことが可能となる。

40

## 【 0 0 7 9 】

同期信号パターンの接続用ビット " # " は、終端用のテーブルを用いる場合と、用いない場合とを区別するためのものである。すなわち、同期信号として与えられた、先頭の第 1 チャンネルビット目の " # " は、終端コードを用いたときは「 1 」とされ、そうでないときは「 0 」とされる。このようにすることにより、テーブルの違い（終端コードを用いたか否か）を、間違いなく識別することができる。

## 【 0 0 8 0 】

50

以上のように終端用テーブルを与えれば、同期信号パターンと合わせ、任意の位置で同期信号を挿入した際も、終端させることができる。

【0081】

図1は、上述した変換テーブルを用いて変調処理を行う変調装置の構成を示すブロック図である。ここでは、データ列が表6に従って、可変長符号(d, k; m, n; r) = (1, 7; 2, 3; 4)に変換される場合を例に挙げて説明する。

【0082】

変調装置1に入力されたデータ列は、DSV制御ビットである「1」あるいは「0」を決定し、任意の間隔で挿入するDSV制御ビット決定・挿入部11に入力される。DSVビット決定・挿入部11から出力されたデータは、変調部12に入力され、変調処理が施されて、NRZI化部13に出力される。NRZI化部13は、入力されたデータを、記録符号列に変換する。タイミング管理部14は、タイミング信号を生成し、各部に供給してタイミングを管理する。

10

【0083】

図2は、変調装置1の他の構成例を示すブロック図である。変調装置1に入力されたデータ列は、DSV制御ビットである「1」あるいは「0」を、所定の位置に挿入するコントロールビット挿入部21に入力される。コントロールビット挿入部21から出力されたデータは、変調部12に入力され、変調処理が施されて、NRZI化してレベル符号化する、NRZI化部13に出力される。NRZI化部13から出力されたデータは、入力された2通りのレベル符号列から、DSV制御された方を選択してこれを記録符号列とする、DSVビット決定部22に入力される。また、タイミング管理部14は、タイミング信号を生成し、各部に供給してタイミングを管理する。

20

【0084】

図3は、変調部12の構成例を示すブロック図である。シフトレジスタ31は、データを2ビットずつシフトさせながら、拘束長判定部32、置き換え制限を含む最小ラン連続制限コード検出部33、置き換え制限を含むラン長制限補償コード検出部34、および変換部35-1乃至35-4に出力するようになされている。このときシフトレジスタ31は、各部が、その処理を行うのに必要なビット数を各部に供給する。

【0085】

拘束長判定部32は、データの拘束長iを判定し、マルチプレクサ36に出力するようになされている。置き換え制限を含む最小ラン連続制限コード検出部33は、最小ランの連続を制限する専用のコードを検出したとき、その拘束長を表す検出信号を拘束長判定部32に出力する。また、置き換え制限を含むラン長制限補償コード検出部34は、表6で示した最大ランを補償する専用のコードを検出したとき、その拘束長を表す検出信号を拘束長判定部32に出力する。

30

【0086】

置き換え制限を含む最小ラン連続制限コード検出部33により専用のコードが検出されたとき、あるいは置き換え制限を含むラン長制限補償コード検出部34により専用のコードが検出されたとき、拘束長判定部32は、対応する拘束長をマルチプレクサ36に出力する。このとき、拘束長判定部32は、別の拘束長を判定している場合があるが、置き換え制限を含む最小ラン連続制限コード検出部33、または置き換え制限を含むラン長制限補償コード検出部34からの専用コードによる検出出力があれば、拘束長判定部32は、そちらの出力を優先させて拘束長を決定する。換言すれば、拘束長判定部32は、より大きい拘束長を選択する。

40

【0087】

変換部35-1乃至35-4は、内蔵されている変換テーブルを参照し、供給されたデータに対応する変換コードが登録されているか否かを判断し、登録されている場合は、そのデータに対応する符号語に変換した後、変換後の符号語をマルチプレクサ36に出力するようになされている。また、対応するデータが変換テーブルに変換コードとして登録されていない場合、変換部35-1乃至35-4は、入力されたデータを破棄するようにな

50

されている。

【0088】

マルチプレクサ36は、拘束長判定部32より供給される拘束長*i*に対応する変換部35-i (*i* = 1乃至4)が変換した符号を選択し、その符号を、シリアルデータとして、バッファ37を介して出力するようになされている。

【0089】

また各部の動作のタイミングは、タイミング管理部14から供給されるタイミング信号に同期して管理されている。

【0090】

次に、変調部12の動作について説明する。シフトレジスタ31より、拘束長判定部32、置き換え制限を含む最小ラン連続制限コード検出部33、置き換え制限を含むラン長制限補償コード検出部34、および各変換部35-1乃至35-4に、変調部12に入力されたデータが3ビット単位で、それぞれの部が判定等に必要なビット数だけ供給される。

【0091】

拘束長判定部32は、例えば表6に示す変換テーブルの基礎コード部分を内蔵しており、この変換テーブルを参照して、データの拘束長*i*を判定し、判定結果(拘束長*i*)をマルチプレクサ36に出力する。

【0092】

置き換え制限を含む最小ラン連続制限コード検出部33は、表6に示す変換テーブルのうちの、最小ランの連続を制限する置き換えコード(表6の場合、データ(110111)と、直前の符号語"0"及び後ろに続く符号語"010"を参照して、変換する部分)を内蔵しており、この変換テーブルを参照して、最小ランの連続を制限するコードを検出したとき、拘束長*i* = 3の検出信号を拘束長判定部32に出力する。

【0093】

また、置き換え制限を含むラン長制限補償コード検出部34は、表6に示す変換テーブル中の、最大ランを守る置き換えコード(表6の場合、データ(00001000-x)および(00000000-x)、x:(01)又は(001))を内蔵しており、この変換テーブルを参照して、最大ランを守る置き換えコードを検出したとき、拘束長*i* = 4の検出信号を拘束長判定部32に出力する。

【0094】

拘束長判定部32は、置き換え制限を含む最小ラン連続制限コード検出部33から拘束長*i* = 3の検出信号が入力された場合、その時、別の拘束長を判定していたとしても、その拘束長を選択せず、置き換え制限を含む最小ラン連続制限コード検出部33の検出に対応する拘束長*i*(表6の例の場合*i* = 3)を選択し、マルチプレクサ36に出力する。同様に拘束長判定部32は、置き換え制限を含むラン長制限補償コード検出部34から拘束長*i* = 4の検出信号が入力された場合、その時、別の拘束長を判定していたとしても、その拘束長を選択せず、置き換え制限を含むラン長制限補償コード検出部34の検出に対応する拘束長*i*(表6の例の場合*i* = 4)を選択し、マルチプレクサ36に出力する。

【0095】

このようにすることは、結局、置き換え制限を含む最小ラン連続制限コード検出部33、または置き換え制限を含むラン長制限補償コード検出部34における拘束長の判定結果と、拘束長判定部32における拘束長の判定結果が、それぞれ異なった場合には、大きい方の拘束長を最終的な拘束長として選択すればよいことを意味している。

【0096】

図4は、変調部12の他の構成例を示すブロック図である。シフトレジスタ31は、データを2ビットずつシフトさせながら、拘束長判定部32、最小ラン・最大ラン補償コード検出部41、最小ラン連続制限コード検出部42、直前・直後ビット参照部43、および、変換部35-1乃至35-4に出力するようになされている。このときシフトレジスタ31は、各部が、その処理を行うのに必要なビット数を供給する。

10

20

30

40

50

## 【 0 0 9 7 】

最小ラン連続制限コード検出部 4 2、最小ラン・最大ラン補償コード検出部 4 1 は、置き換え制限の付加されていない第 1 と第 2 の置き換えコード検出をする。そして、直前・直後ビット参照部 4 3 は、表 6 で示した置き換え制限を付加する規則を与えており、直前・直後ビット参照部 4 3 から出力され、拘束長判定部 3 2 へ入力された信号は、図 3 で示した拘束長判定部 3 2 への信号と、同様の信号である。他の部分は、図 3 と同様なので、その説明は省略する。

## 【 0 0 9 8 】

次に、図 5 を参照し、図 3 に示した拘束長判定部 3 2、置き換え制限を含む最小ラン連続制限コード検出部 3 3、及び置き換え制限を含むラン長制限補償コード検出部 3 4 の動作を、具体例を上げて説明する。

10

## 【 0 0 9 9 】

置き換え制限を含むラン長制限補償コード検出部 3 4 は、表 6 に示す変換テーブルの、( 0 0 0 0 1 0 0 0 - x ) および ( 0 0 0 0 0 0 0 0 - x ) の変換部分を持ち、入力された 8 ビットのデータが、これと一致した場合、さらに、次のデータを参照して、続くデータが ( 0 1 ) 又は ( 0 0 1 ) であったとき、拘束長  $i = 4$  の検出信号を拘束長判定部 3 2 に出力する。換言すると、入力された 6 ビットのデータが ( 0 0 0 0 1 0 ) と ( 0 0 0 0 0 0 ) に一致した場合、その直後に続く符号語列が " 0 0 0 " であったときに、拘束長  $i = 4$  の検出信号を拘束長判定部 3 2 に出力することになる。

## 【 0 1 0 0 】

置き換え制限を含む最小ラン連続制限コード検出部 3 3 は、表 6 に示す変換テーブルの、データ ( 1 1 0 1 1 1 ) と直前の符号 " 0 " と直後の符号 " 0 1 0 " の変換部分を持ち、入力された 6 ビットのデータが、( 1 1 0 1 1 1 ) であり、その直前の符号語が " 0 " であり、かつ、直後の 3 符号語が、" 0 1 0 " である場合、拘束長  $i = 3$  の検出信号を拘束長判定部 3 2 に出力する。そして、3 符号語 " 0 1 0 " の部分を、データ変換前のデータ列で表せば、( 0 1 )、( 0 0 1 )、又は ( 0 0 0 0 0 ) となる。従って、置き換え制限を含む最小ラン連続制限コード検出部 3 3 は、直前符号 " 0 " + ( 1 1 0 1 1 1 ) + ( 0 1 / 0 0 1 / 0 0 0 0 0 ) の変換部分を持ち、入力された 6 ビットのデータに加えて、直前の 1 符号の他に、直後の 5 ビットのデータまでを、さらに参照し、それらが、これらのいずれかと一致する場合、拘束長  $i = 3$  の検出信号を拘束長判定部 3 2 に出力する。

20

30

## 【 0 1 0 1 】

また拘束長判定部 3 2 は、表 6 に示すテーブルの基礎コード部分を内蔵しており、入力された 6 ビットのデータが、( 0 0 0 0 1 1 )、( 0 0 0 0 1 0 )、( 0 0 0 0 0 1 )、あるいは ( 0 0 0 0 0 0 ) のいずれかに一致する場合、拘束長  $i = 3$  と判定する。また、入力された 4 ビットのデータが ( 0 0 1 1 )、( 0 0 1 0 )、( 0 0 0 1 ) のいずれかに一致する場合、拘束長判定部 3 2 は、拘束長  $i = 2$  と判定する。さらに、入力された 2 ビットのデータが ( 1 1 )、( 1 0 )、( 0 1 ) のいずれかに一致する場合、拘束長判定部 3 2 は、拘束長  $i = 1$  と判定する。

## 【 0 1 0 2 】

ところで、入力されたデータが、例えば ( 0 0 0 0 1 0 ) であったとき、拘束長判定部 3 2 は、拘束長  $i = 3$  と判定する。しかしながら、上記の 6 ビットに加え、さらに続くデータが、( 0 0 0 1 ) 又は ( 0 0 0 0 1 ) であったとき、置き換え制限を含むラン長制限補償コード検出部 3 4 により、拘束長  $i = 4$  と判定される。このような場合、置き換え制限を含むラン長制限補償コード検出部 3 4 からの出力信号が優先され、拘束長  $i = 4$  と決定される。

40

## 【 0 1 0 3 】

このようにして、表 6 のテーブルに従って、最大拘束長である 8 ビットと、必要な場合は、直前あるいは直後の符号語列が参照されて、全ての ( 1 ) と ( 0 ) からなるデータ列より拘束長が決定される。

## 【 0 1 0 4 】

50



i = 1	1 1	* 0 *
	1 0	0 0 1
	0 1	0 1 0

は、次のような配列となっても良い。

	データ	符号
i = 1	1 1	* 0 *
	1 0	0 1 0
	0 1	0 0 1

この場合でも、データ列の要素の「1」の個数と、符号語列の要素の「1」の個数は、それぞれ2で割った時の余りが、どちらも1あるいは0で一致するようにする。

10

#### 【0 1 1 3】

この他にも、表3乃至表6の、データ列の各要素の(1)と(0)を反転させても良い。すなわち、

データ列	符号列
11	*0*
10	001
01	010
0011	010 100
0010	010 000
0001	000 100

20

が、次のようになっても良く、この場合でも、データ列の要素の「1」の個数と、符号語列の要素の「1」の個数は、それぞれ2で割った時の余りが、どちらも1あるいは0で一致している。

データ列	符号列
00	*0*
01	001
10	010
1100	010 100
1101	010 000
1110	000 100

30

#### 【0 1 1 4】

図6は、上述した処理により変調されたデータを復調する復調装置51の構成を示すブロック図である。復調装置51は、可変長符号(d, k; m, n; r) = (1, 7; 2, 3; 4)を、表6の逆変換テーブルを用いてデータ列に復調する。

#### 【0 1 1 5】

復調装置51は、伝送路より伝送されてきた信号、または、記録媒体より再生された信号は、コンパレートし、逆NRZI化して記録符号列から符号語列に変換する、コンパレート・逆NRZI化部52に入力される。コンパレート・逆NRZI化部52から出力された信号は、復調テーブル(逆変換テーブル)に基づいて復調する復調部53に入力される。復調部53により復調されたデータ列は、任意の間隔で挿入されているデータ列内のDSV制御ビットを除去し、元のデータ列を復元するDSV制御ビット除去部54に入力される。DSV制御ビット除去部54から出力されたデータ列は、バッファ44に入力され、そのシリアルデータを一旦記憶し、所定の転送レートで読み出し、出力する。タイミング管理部56は、タイミング信号を生成し、各部に供給してタイミングを管理する。

40

#### 【0 1 1 6】

図7は、復調部53の構成例を示すブロック図である。拘束長判定部61は、コンパレート・逆NRZI化部52により、デジタル化された信号の入力を受け、拘束長iを判定する。また最小ラン連続制限コード検出部62は、デジタル化された信号から、最小ランの連続を制限するために与えられた専用のコードを検出し、それに対応する検出信号を拘束長判定部61に送る。さらに、ラン長制限補償コード検出部63は、入力された信号が

50

ら、最大ランを補償するために与えられた専用のコードを検出し、それに対応する検出信号を拘束長判定部 6 1 に送る。

【 0 1 1 7 】

逆変換部 6 4 - 1 乃至 6 4 - 4 は、 $n \times i$  ビットの可変長符号を、 $m \times i$  ビットのデータに逆変換するテーブルを有している。マルチプレクサ 6 5 は、逆変換部 6 4 - 1 乃至 6 4 - 4 からの出力のいずれかを、拘束長判定部 6 1 の判定結果に対応して選択し、シリアルデータとして出力する。

【 0 1 1 8 】

次に図 7 に示した復調部 5 3 の動作について説明する。伝送路より伝送されてきた信号、あるいは記録媒体より再生された信号は、コンパレート・逆 NRZI 化部 5 2 に入力され、コンパレートされる。さらに、逆 NRZI 符号 ( " 1 " がエッジを示す符号 ) のデジタル信号とされて、拘束長判定部 6 1 に入力され、表 6 に示す変換テーブル ( 逆変換テーブル ) の基礎コード部分に従って、拘束長の判定処理が行われる。拘束長判定部 6 1 の判定結果 ( 拘束長 ) はマルチプレクサ 6 5 に出力される。

10

【 0 1 1 9 】

復調部 5 3 に入力されたデジタル信号は、最小ラン連続制限コード検出部 6 2 にも入力される。最小ラン連続制限コード検出部 6 2 は、表 6 に示す、変換テーブルのうちの、最小ランの連続を制限する置き換えコード ( 表 6 の場合、符号語 " 0 0 1 0 0 0 0 0 0 " を変換する部分 ) を内蔵しており、この逆変換テーブルを参照して、最小ランの連続を制限するコード " 0 0 1 0 0 0 0 0 0 not 1 0 0 " を検出したとき、拘束長  $i = 3$  の検出信号を拘束長判定部 6 1 に出力する。ここで " not 1 0 0 " とは、" 1 0 0 " 以外の符号語を意味する。

20

【 0 1 2 0 】

さらに、入力されたデジタル信号は、ラン長制限補償コード検出部 6 3 にも入力される。ラン長制限補償コード検出部 6 3 は、表 6 に示す変換テーブルの中の、最大ランを守る置き換えコード ( 表 6 の場合、符号語列 " 0 0 0 1 0 0 1 0 0 1 0 0 " 及び " 0 1 0 1 0 0 1 0 0 1 0 0 " ) を内蔵しており、この逆変換テーブルを参照して、最大ランを守る置き換えコードを検出したとき、拘束長  $i = 4$  の検出信号を拘束長判定部 6 1 に出力する。

30

【 0 1 2 1 】

図 8 は、拘束長判定部 6 1、最小ラン連続制限コード検出部 6 2、およびラン長制限補償コード検出部 6 3 の判定処理について説明する図である。すなわち、ラン長制限補償コード検出部 6 3 は、表 6 に示すテーブルの、" 0 0 0 1 0 0 1 0 0 1 0 0 " あるいは " 0 1 0 1 0 0 1 0 0 1 0 0 " の逆変換部分を持ち、入力された 1 2 ビットの符号語列が、これと一致する場合、拘束長  $i = 4$  の検出信号を拘束長判定部 6 1 に出力する。

【 0 1 2 2 】

最小ラン連続制限コード検出部 6 2 は、表 6 に示すテーブルの " 0 0 1 0 0 0 0 0 0 " の逆変換部分を持ち、入力された 1 2 ビットの符号語列が、" 0 0 1 0 0 0 0 0 0 not 1 0 0 " と一致する場合、拘束長  $i = 3$  の検出信号を拘束長判定部 6 1 に出力する。

40

【 0 1 2 3 】

また拘束長判定部 6 1 は、表 6 に示す逆変換テーブルを内蔵しており、入力された 9 ビット又は 1 2 ビットの符号語列が、" 0 0 0 1 0 0 1 0 0 " , " 0 0 0 1 0 0 0 0 0 not 1 0 0 " , " 0 1 0 1 0 0 1 0 0 " , あるいは " 0 1 0 1 0 0 0 0 0 not 1 0 0 " のいずれかに一致する場合、拘束長  $i = 3$  と判定する。これに当てはまらない場合、入力された 6 ビット又は 9 ビットの符号語列が、" 0 1 0 1 0 0 " , " 0 1 0 0 0 0 not 1 0 0 " , あるいは " 0 0 0 1 0 0 " のいずれかに一致する場合、拘束長  $i = 2$  と判定する。さらに、これに当てはまらない場合、入力された 3 ビットの符号語列が、" 0 0 0 " , " 1 0 1 " , " 0 0 1 " , あるいは " 0 1 0 " のいずれかに一致するときに、拘束長判定部 6 1 は、拘束長  $i = 1$  と判定する。

50

【 0 1 2 4 】

なお、拘束長判定部 6 1、最小ラン連続制限コード検出部 6 2、及びラン長制限補償コード検出部 6 3 の拘束長判定の処理は、拘束長の小さい方から、 $i = 1$ 、 $i = 2$ 、 $i = 3$ 、 $i = 4$  の順番で行うようにしてもよい。

【 0 1 2 5 】

拘束長を、その小さい方から、 $i = 1$ 、 $i = 2$ 、 $i = 3$ 、 $i = 4$  の順番で判定する場合に、入力された符号語列が例えば、" 0 0 0 1 0 0 1 0 0 1 0 0 " であるとき、拘束長判定部 6 1 において、拘束長の小さいほうから順に、一致または不一致を判定していくと、拘束長  $i = 1$ 、拘束長  $i = 2$ 、拘束長  $i = 3$ 、または拘束長  $i = 4$  と、全ての拘束長  $i$  にあてはまることになる。このような場合、それぞれ判定された拘束長  $i$  から最大のものを選択し、決定するようにすればよい。

10

【 0 1 2 6 】

逆変換部 6 4 - 1 乃至 6 4 - 4 のうち、例えば、逆変換部 6 4 - 1 には、アドレス " 1 0 1 " および " 0 0 0 " にデータ ( 1 1 ) が、アドレス " 0 0 1 " にデータ ( 1 0 ) が、そしてアドレス " 0 1 0 " にデータ ( 0 1 ) が、それぞれ書き込まれている。以下、逆変換部 6 4 - 2 乃至 6 4 - 4 の各逆変換テーブルも、同様に、それぞれ対応するデータが書き込まれており、供給された  $3 \times i$  ビットの符号語列を、 $2 \times i$  ビットのデータ列に変換し、そのデータ語をマルチプレクサ 6 5 に出力する。

【 0 1 2 7 】

マルチプレクサ 6 5 は、逆変換部 6 4 - 1 乃至 6 4 - 4 より供給されたデータのいずれかを、拘束長判定部 6 1 の拘束長判定結果に対応して選択し、シリアルデータとして出力する。

20

【 0 1 2 8 】

表 6 の逆変換テーブルを示すと、次の表 8 のようになる。

< 表 8 > 逆変換テーブル ( 1 , 7 ; 2 , 3 ; 4 )

符号語列	復調データ列
i=4 : limits k to 7	
000 100 100 100	00001000
010 100 100 100	00000000
i=3 : Prohibit Repeated Minimum Transition Runlength	
001 000 000(not 100)	110111
-----	
i=3 000 100 100	000011
000 100 000(not 100)	000010
010 100 100	000001
010 100 000(not 100)	000000
i=2 010 100	0011
010 000(not 100)	0010
000 100	0001
i=1 101	11
000	11
001	10
010	01

30

40

【 0 1 2 9 】

次に、図 9 のフローチャートを参照して、DSV制御ビット除去部 5 4 の動作について説明する。DSV制御ビット除去部 5 4 は、内部にカウンタを有しており、ステップ S 1 において、復調部 5 3 よりデータ列のビットが入力されると、その数をカウントする。ステップ S 2 において、カウント値が DSV制御ビットを挿入する所定のデータ間隔に達したか否かが判定され、任意のデータ間隔ではないと判定された場合、ステップ S 3 に進み、復調部 5 3 より入力されたデータが、そのままバッファ 5 5 に出力される。これに対して、ス

50



ステップ S 2 において、所定のデータ間隔であると判定された場合、そのビットは DSV 制御ビットであるから、ステップ S 3 の処理はスキップされ、ステップ S 4 に進む。すなわち、このような場合には、そのビットはバッファ 5 5 に出力されず、廃棄される。

【 0 1 3 0 】

次に、ステップ S 4 において、次のデータを入力する処理が実行される。そして、ステップ S 5 において、全てのデータに対する処理が終了したか否かが判定され、まだ処理していないデータが存在する場合には、ステップ S 1 に戻り、それ以降の処理が繰り返し実行される。ステップ S 5 において、全てのデータが処理されたと判定された場合、このフローチャートの処理は終了される。

【 0 1 3 1 】

以上のようにして、DSV 制御ビット除去部 5 4 より出力されるデータは、DSV 制御ビットが除去されたデータである。このデータは、バッファ 5 5 を介して出力される。

【 0 1 3 2 】

ところで、データ変調時に同期信号 ( S y n c ) を挿入する場合の変調装置の構成例を図 1 0 と図 1 1 に、復調装置の構成例を図 1 1 に、それぞれ示す。これらの実施の形態においても、データ列が表 6 に従って、可変長符号 ( d , k ; m , n ; r ) = ( 1 , 7 ; 2 , 3 ; 4 ) に変調され、また復調されるものとする。

【 0 1 3 3 】

所定の間隔で同期信号を挿入する変調装置 7 1 においては、図 1 0 に示すように、DSV 制御ビット決定・挿入部 7 2 の出力は、変調部 7 3 と SYNC 決定部 7 4 に供給される。SYNC 決定部 7 4 には、変調部 7 3 からの出力も供給される。SYNC 決定部 7 4 は、これらの入力された信号から同期信号を決定し、その出力を SYNC 挿入部 7 5 に出力している。SYNC 挿入部 7 5 は、変調部 7 3 より入力される変調信号に、SYNC 決定部 7 4 より入力される同期信号を挿入し、NRZI 化部 7 6 に出力している。その他の構成は、図 1 に示した変調装置 1 と同様であるので、その説明は省略する。

【 0 1 3 4 】

SYNC 決定部 7 4 は、同期信号パターンを 1 2 符号語とすると、同期信号を、 " # 0 1 0 0 0 0 0 0 0 0 1 " と決定する。 " # " は、同期信号の挿入により区切られた、直前のデータ列 ( DSV 制御ビットは含んで良い ) に依存しており、区切られたデータ列を変換テーブルに従って変調した際に、終端テーブルを用いて終端させた場合、 " # " = " 1 " とされ、また終端テーブルを用いずに、表 2 のテーブルにより終端した場合、 " # " = " 0 " とされる。

【 0 1 3 5 】

変調部 7 3 は、終端テーブルを用いた場合、 " # " = " 1 " を、用いない場合、 " # " = " 0 " を、SYNC 決定部 7 4 に出力する。SYNC 決定部 7 4 は、変調部 7 3 から、 " # " の値の入力を受けると、これを同期信号の先頭ビットに挿入する。そして、その同期信号を SYNC 挿入部 7 5 に出力する。

【 0 1 3 6 】

SYNC 挿入部 7 5 は、SYNC 決定部 7 4 から出力された同期信号を、変調部 7 3 からの出力に挿入し、NRZI 化部 7 6 に出力する。その他の動作は、図 1 に示した変調装置 1 と同様である。

【 0 1 3 7 】

同期信号が挿入された後の最初のデータは、その先頭から ( 同期信号の直前のデータを考慮することなく ) 変換処理される。変調部 7 3 と SYNC 決定部 7 4 は、同期信号が挿入される所定の間隔をカウントするためのカウンタを備え、そのカウント値に対応して、同期信号の位置を決定する。

【 0 1 3 8 】

図 1 1 は、他の変調装置 7 1 の構成を示すブロック図である。変調装置 7 1 に入力された信号は、DSV 制御ビットである「 1 」あるいは「 0 」をデータ列の所定の位置に挿入するコントロールビット挿入部 8 1 に入力され、さらに 2 通りのデータ列を変調する変調部

10

20

30

40

50

73にされる。変調部73からの出力は、所定の間隔で同期信号を挿入する同期信号挿入部75にされ、さらに、NRZI化してレベル符号化する、NRZI化部76にされる。そして、NRZI化部76から出力された信号は、2通りのレベル符号列から、DSV制御された方を選択してこれを記録符号列とする、DSVビット・SYNC決定部82にされる。また、変調装置71は、タイミング信号を生成し、各部に供給してタイミングを管理するタイミング管理部77を備える。

【0139】

そして、10または11に示した変調装置71の変調部73は、例えば、表6によるデータ変換の詳細を示した、3または4で示した変調部12と同様の構成で実現することが可能である。

10

【0140】

12は、10または11の変調装置71により変調され、出力された符号を復調する復調装置の構成例を示すブロック図である。所定の伝送経路を介して、復調装置91にされた符号は、コンパレート・逆NRZI化部92にされ、符号語列にされる。コンパレート・逆NRZI化部92から出力された符号語列は、復調部93とSYNC識別部94にされる。SYNC識別部94は、された符号を用いて、同期信号を識別し、識別信号を復調部93とSYNC除去部95にする。SYNC除去部95は、復調部93からされた復調信号から、SYNC識別部94の出力に対応して同期信号を除去し、同期信号を除去した信号をDSV制御ビット除去部96にする。

【0141】

20

SYNC識別部94は、内蔵するカウンタで符号語をカウントし、そのカウント値から所定の間隔で挿入されている同期信号の位置を決定する。同期信号パターンの位置が判明したとき、SYNC識別部94は、次に変調時に定めた“#”の部分を読み取る。即ち同期信号ビット部分の先頭ビットを読み取り、それを復調部93にする。復調部93は、先頭ビットが“1”であれば、その直前の符号の復調には、表6の終端テーブルを用いる。また先頭ビットが“0”であれば、復調部93は、その直前の符号の復調には、表6の変換コードのテーブルを用いる。これ以外の同期信号ビットは、情報を持たないビットであるから不要となる。

【0142】

SYNC識別部94は、同期信号を構成するビットを識別する識別信号をSYNC除去部95にする。SYNC除去部95は、復調部93からされたデータから、SYNC識別部94からされた識別信号によって指定された同期信号ビットだけを除去し、DSV制御ビット除去部96にする。

30

【0143】

そして、12における復調装置91の復調部93は、例えば、表6によるデータ変換の逆変換詳細を示した、7に示した復調部53と同様の構成で実現することが可能である。

【0144】

ここで、実施の形態で説明した変換テーブルを用いた変調結果を検証した結果を示す。T<sub>min</sub>の連続を制限し、かつ、データ列内においてDSV制御ビットを挿入したデータ列を変調した表6は、さらに、置き換えコードの変換条件を付加し、復調エラー伝搬を低減させる変換コードとなっている。シミュレーションは、従来の表2に従った1,7PP符号と、本実施の形態を用いた表6に従った1,7PP符号の比較を行った。

40

【0145】

任意に作成したランダムデータ13,107,200bitを、56data-bitおきにDSV制御ビットを1bitを挿入することでDSV制御した後、表2または表6の変調コードテーブルを用いて、符号語列(チャンネルビット列)に変換した場合の結果は以下の通りである。

【0146】

各結果の数値は以下のようにして計算した。

Ren\_cnt[1 to 10]: 最小ランの繰り返し1回乃至10回の各発生数。

50

T\_size[2 to 10]: 2 T乃至10 Tの各ランの発生数。

Sum : Number of bits. ビット総数。

Total : Number of runlengths. 各ラン(2 T, 3 T, ...)の発生総数

Average Run : (Sum/Total)

run分布の数値 : (T\_size[i] \* (i)) / (Sum), i=2,3,4,,10

【0147】

以下に示す表9の2 T乃至10 Tの欄に示す数値が、このラン分布の数値を表す。

【0148】

Tminの連続する分布の数値 : (Ren\_cnt[i] \* (i)) / T\_size[2T], i=1,2,3,4...10

【0149】

表9のRMTR(1)乃至RMTR(7)の欄に示す値が、この最小ランの連続する分布の数値を表す

10

max-RMTR : 最小ランの繰り返す、最大回数。

peak DSV : 符号語列のDSV制御を行う過程において、DSV値を計算したときのDSV値のプラス側のピーク及びマイナス側のピークをいう。DSV制御ビットとして56データ列おきにDSV制御ビットを挿入した場合の冗長率は、56データ列に対してDSV制御ビット1bitであるから、冗長度は、1.75%(1/(1+56))である。

【0150】

<表9> \*\*\* 1,7PP comparison \*\*\*

	<表2> 従来1,7PP	<表6> 新1,7PP
Average Run	3.3665	3.3723
Sum	20011947	20011947
Total	5944349	5934288

20

2T	0.2256	0.2293
3T	0.2217	0.2111
4T	0.1948	0.1949
5T	0.1499	0.1515
6T	0.1109	0.1111
7T	0.0579	0.0672
8T	0.0392	0.0349
9T	-----	-----

30

RMTR(1)	0.3837	0.3803
RMTR(2)	0.3107	0.3032
RMTR(3)	0.1738	0.1799
RMTR(4)	0.0938	0.0968
RMTR(5)	0.0299	0.0319
RMTR(6)	0.0081	0.0079
RMTR(7)	-----	-----
max-RMTR	6	6

40

=====  
peak DSV # -36to36 # -36to37  
("#":56data-bit+1dc-bit, 1.75%)

【0151】

上述した結果より、従来を表2に従った1,7PP符号と、本実施の形態の表6に従った1,7PP符号は、それぞれ最小ランd=1と最大ランk=7、及び最小ランの連続は6回までに制限されていることが確認されるとともに、peak DSVの結果より、データ列内でDSV制御を行うことができる(peak DSVの値が所定の範囲内に納められている)ことが示された。

50

また、表 9 より、表 2 と表 6 のテーブルの違いによるラン分布及び、最小ラン連続回数分布に特性の差がないことがわかった。

【 0 1 5 2 】

次に、復調のシミュレーション結果について述べる。復調は、比較が可能なように表 2 と表 6 とともに、同様な復調方法により行った。正常に復調が行われるのを確認するとともに、符号語列において任意にビットシフトエラーを発生させ、この時の復調エラー伝搬特性を調べた。復調エラー伝搬とは、符号語列内の一箇所のビットシフトエラーに対し、復調時に何ビットだけエラーを伝搬したかであり、これを 8 ビット単位とすることでバイトエラーとした。

【 0 1 5 3 】

<表 1 0 >	*** Shift error response ***		
	<表 2 >	<表 6 >	
	従来 1,7PP	新 1,7PP	
worst case	3 Bytes	3 Bytes	
(dc bit)	exclude.	exclude	
Byte error(0)	0.177	0.183	
Byte error(1)	0.673	0.672	
Byte error(2)	0.150	0.144	
Byte error(3)	0.000	0.000	
Average -			
Byte error rate	0.973Byte	0.961Byte	

10

20

【 0 1 5 4 】

表 1 0 の結果より、表 2 と表 6 とともに 1,7PP の最悪エラー伝搬は 3 バイトであるが、実際の発生頻度はほとんどないことが確認された。また、表 6 による、置き換えコードの変換条件を付加し、復調エラー伝搬を低減させる変換コードを与えたことによって、長いエラー伝搬の発生率が 15.0% から 14.4% に減り、また平均バイトエラー伝搬長も減少させることができた。

【 0 1 5 5 】

本実施の形態における、他の効果として、表 5 の場合を考える。このテーブルを用いた場合は、平均エラー伝搬長が減少すると同時に、RMTR 特性を変化させることができる。

30

【 0 1 5 6 】

<表 1 1 >	*** 1,7PP comparison ***		
	<表 2 >	<表 6 >	<表 5 >
	従来 1,7PP	新 1,7PP	新 1,7PP
RMTR(1)	0.3837	0.3803	0.3886
RMTR(2)	0.3107	0.3032	0.3099
RMTR(3)	0.1738	0.1799	0.1839
RMTR(4)	0.0938	0.0968	0.0866
RMTR(5)	0.0299	0.0319	0.0261
RMTR(6)	0.0081	0.0079	0.0049
RMTR(7)	-----	-----	-----
max-RMTR	6	6	6

40

【 0 1 5 7 】

表 5 は、長い RMTR の発生回数を少なくすることが出来るので、表 9 と同様なシミュレーションを行うと、RMTR は最大 6 回までで同様でありながらさらに、RMTR(4), RMTR(5), RMTR(6) といった大きい回数の発生頻度が減少する。特に表 1 1 より、最大の RMTR(6) の出現確率が、0.008 前後から 0.005 程度になり、約 2/3 に減少させることが示された。

【 0 1 5 8 】

このように、大きな RMTR をさらに低下させることによって、エラーが発生した時の長いエラー伝搬を未然に防ぐことが出来ることになり、従ってエラー伝搬値も向上することが

50

期待できる。

【0159】

1,7PP符号は、最小ラン  $d = 1$ 、最大ラン  $k = 7$ 、変換率  $m / n = 2 / 3$  の変換テーブルにおいて、最小ラン長の繰り返し回数を制限する置き換えコードを設けるようにしたので、

(1) 高線密度での記録再生、及び、タンジェンシャル・チルトに対する許容度を向上させることができる。

(2) 信号レベルが小さい部分が減少し、AGCやPLL等の波形処理の精度が向上し、総合特性を高めることができる。

(3) 従来と比較して、ビタビ復号等の際のパスメモリ長を短く設計することができ、回路規模を小さくすることができる。 10

【0160】

また、変換テーブルの要素内の「1」の個数と、変換される符号語列の要素内の「1」の個数を、2で割った時の余りが、どちらも1あるいは0で一致するようにしたので、

(4) DSVの制御のための冗長ビットを少なくすることができる。

(5) 最小ラン  $d = 1$  かつ  $(m, n) = (2, 3)$  においては、1.5符号語でDSV制御を行うことができる。

(6) 冗長度が少ない上に、最小ランと最大ランを守ることができる。

【0161】

さらに本テーブルは特に、ラン長制限を守る置き換えコードを設けるようにしたので、 20

(7) テーブルをコンパクトにできる。

【0162】

そして、本実施の形態において、復調時のエラー伝搬長が長くなるという点を、エラー伝搬が大きくなる変換コードの発生回数を減らすテーブル構造とすることにより、

(8) ビットシフト時の復調エラー伝搬を、表2の従来の1,7PPよりも少なくできる。

【0163】

なお、本明細書中において、上記処理を実行するコンピュータプログラムをユーザに提供する提供媒体には、磁気ディスク、CD-ROMなどの情報記録媒体の他、インターネット、デジタル衛星などのネットワークによる伝送媒体も含まれる。 30

【図面の簡単な説明】

【0164】

【図1】本発明を適用した変調装置の一実施の形態の構成を示すブロック図である。

【図2】変調装置の他の構成を示すブロック図である。

【図3】変調部の構成を示すブロック図である。

【図4】変調部の他の構成を示すブロック図である。

【図5】変調について説明する図である。

【図6】復調装置の構成を示すブロック図である。

【図7】復調部の構成を示すブロック図である。

【図8】復調について説明する図である。 40

【図9】DSV制御ビットの除去処理を説明するフローチャートである。

【図10】変調装置の他の構成を示すブロック図である。

【図11】変調装置のさらに他の構成を示すブロック図である。

【図12】復調装置の他の構成を示すブロック図である。

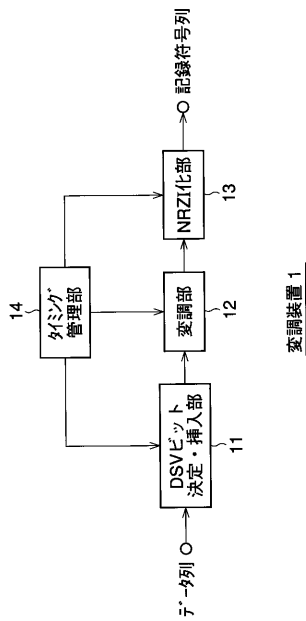
【符号の説明】

【0165】

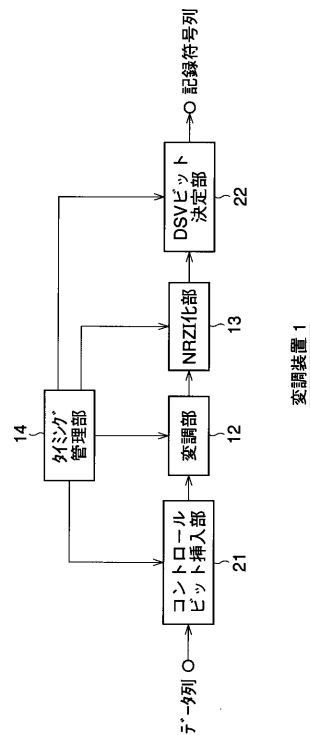
11 DSVビット決定・挿入部, 12 変調部, 13 NRZI化部, 14 タイミング管理部, 21 コントロールビット挿入部, 22 DSVビット決定部, 32 拘束長判定部, 33 置き換え制限を含む最小ラン連続制限コード検出部, 34 置き換え制限を含むラン長制限補償コード検出部, 41 最小ラン・最大ラン補償コード 50

検出部， 4 2 最小ラン連続制限コード検出部， 4 3 直前・直後ビット参照部，  
 5 2 コンパレート・逆NRZI化部， 5 3 復調部， 5 4 DSV制御ビット除去部，  
 6 1 拘束長判定部， 6 2 最小ラン連続制限コード検出部， 6 3 ラン長制限補償  
 コード検出部

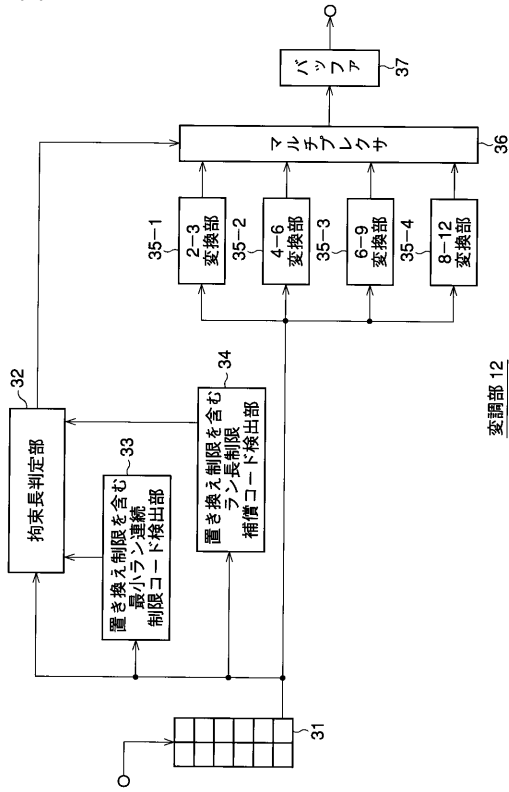
【 図 1 】



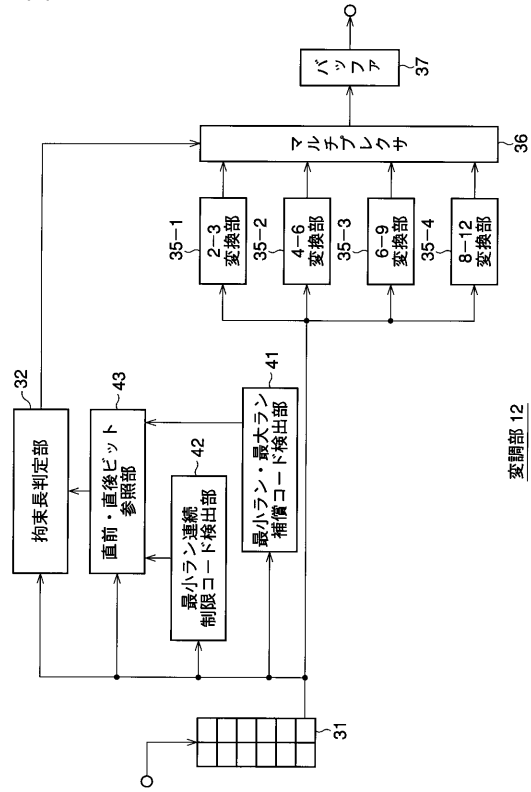
【 図 2 】



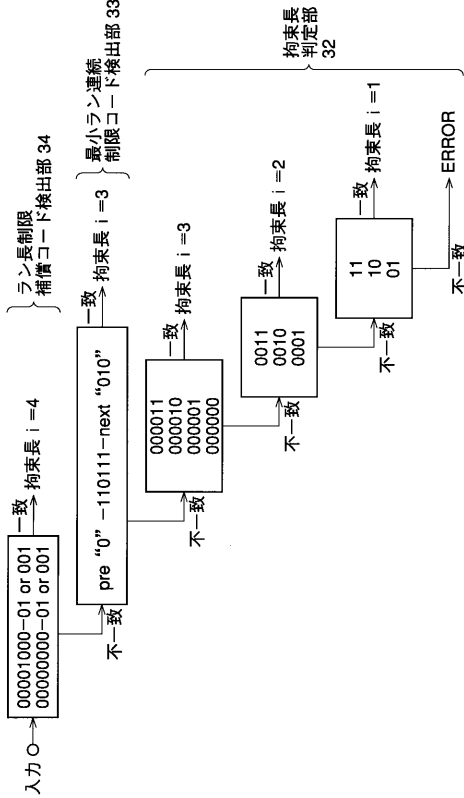
【 図 3 】



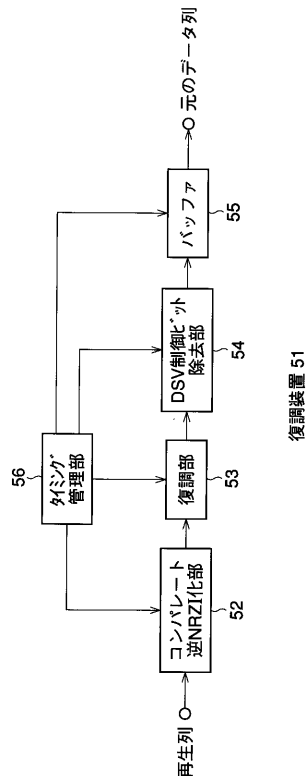
【 図 4 】



【 図 5 】



【 図 6 】

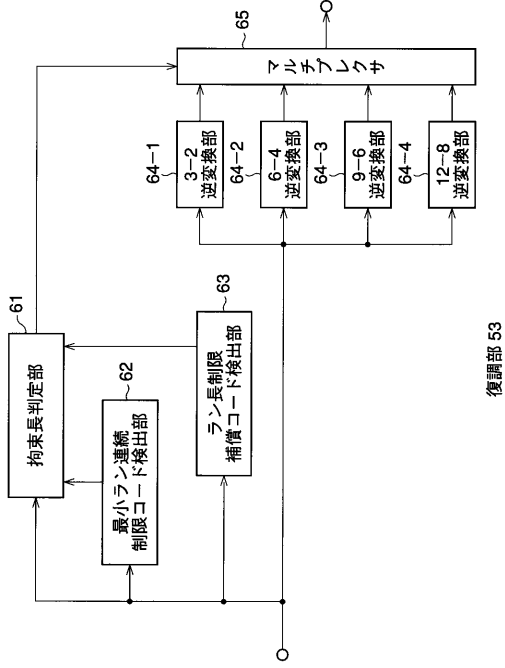


変調部 12

変調部 12

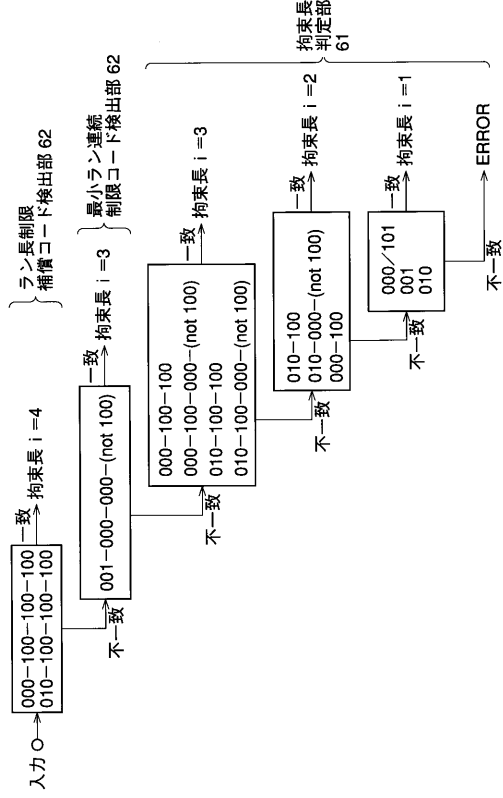
復調装置 51

【 図 7 】

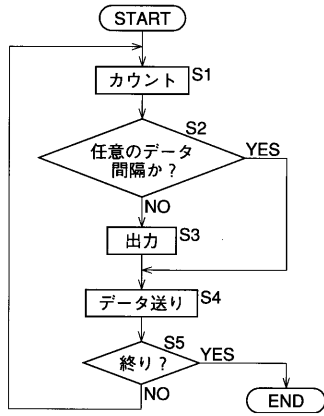


復調部 53

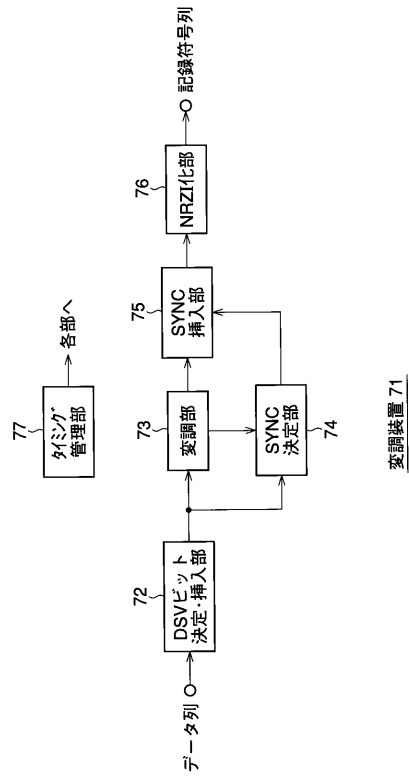
【 図 8 】



【 図 9 】



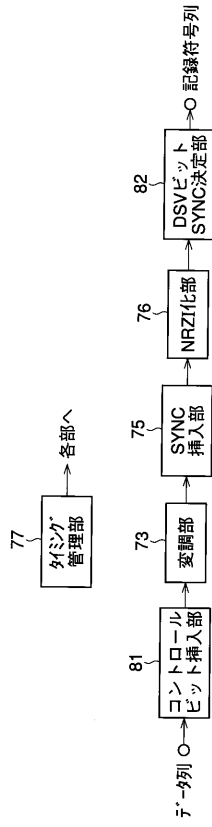
【 図 10 】



変調装置 71

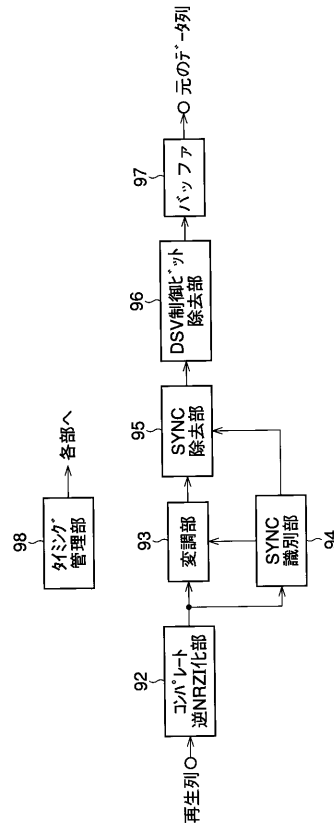


【 図 1 1 】



変調装置 71

【 図 1 2 】



変調装置 91