

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3782995号

(P3782995)

(45) 発行日 平成18年6月7日(2006.6.7)

(24) 登録日 平成18年3月17日(2006.3.17)

(51) Int. Cl.	F I
HO 3 M 13/15 (2006.01)	HO 3 M 13/15
HO 4 B 1/707 (2006.01)	HO 4 J 13/00 D
HO 4 L 1/00 (2006.01)	HO 4 L 1/00 B

請求項の数 31 (全 37 頁)

(21) 出願番号	特願2002-574234 (P2002-574234)	(73) 特許権者	503447036
(86) (22) 出願日	平成14年3月20日(2002.3.20)		サムスン エレクトロニクス カンパニー
(65) 公表番号	特表2004-527949 (P2004-527949A)		リミテッド
(43) 公表日	平成16年9月9日(2004.9.9)		大韓民国キョンギード, スウォンーシ, ヨ
(86) 国際出願番号	PCT/KR2002/000471		ントンーク, マエタンードン 4 1 6
(87) 国際公開番号	W02002/075930	(74) 代理人	100064908
(87) 国際公開日	平成14年9月26日(2002.9.26)		弁理士 志賀 正武
審査請求日	平成15年8月11日(2003.8.11)	(74) 代理人	100089037
(31) 優先権主張番号	2001/14418		弁理士 渡邊 隆
(32) 優先日	平成13年3月20日(2001.3.20)	(72) 発明者	ジェーヨル・キム
(33) 優先権主張国	韓国 (KR)		大韓民国・4 3 5 - 0 4 2 ・キョンギード
(31) 優先権主張番号	2001/15294		・クンポーシ・サンボン・2ードン(番地
(32) 優先日	平成13年3月23日(2001.3.23)		なし)・サンボン・9ーダンジ・ベクトウ
(33) 優先権主張国	韓国 (KR)		ー・アパートメント・# 9 6 0 - 1 4 0 1

最終頁に続く

(54) 【発明の名称】 符号分割多重接続移動通信システムでの符号化／復号化装置及び方法

(57) 【特許請求の範囲】

【請求項 1】

4ビットの入力情報ビットを受信し、前記4ビットの入力情報ビットそれぞれを長さ16の相異なるウォルシュ符号に符号化し、排他的論理和器により論理和がとられた16個の符号化シンボルから構成された符号化シンボル列を利用して24個の符号化シンボルから構成された符号化シンボル列に出力する移動通信システムでの符号化方法において、

前記16個の符号化シンボルから構成された符号化シンボル列を1回反復して32個の符号化シンボルから構成された符号化シンボル列を出力する過程と、

前記32個の符号化シンボルのうち、0、1、2、3、4、5、6、16番目のシンボルを穿孔して前記24個の符号化シンボルから構成された前記符号化シンボル列を出力する過程と、を含むことを特徴とする方法。

10

【請求項 2】

7ビットの入力情報ビットを受信し、前記7ビットの入力情報ビットそれぞれを長さ32の相異なるウォルシュ符号とマスク関数に符号化し、排他的論理和器により論理和がとられた32個の符号化シンボルから構成された符号化シンボル列を利用して24個の符号化シンボルから構成された符号化シンボル列に出力する移動通信システムでの符号化方法において、

前記32個の符号化シンボルのうち、0、4、8、12、16、20、24、28番目のシンボルを穿孔して前記24個の符号化シンボルから構成された前記符号化シンボル列を出力する過程を含むことを特徴とする方法。

20

【請求項 3】

前記マスク関数は、‘ 0 1 1 1 0 1 1 1 0 0 1 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 ’と‘ 0 0 1 0 0 1 1 0 0 1 0 1 0 1 0 0 0 0 1 0 1 0 1 0 0 0 0 1 0 0 0 0 ’であることを特徴とする請求項 2 に記載の方法。

【請求項 4】

4 ビットの入力情報ビットを受信し、前記 4 ビットの入力情報ビットそれぞれを長さ 16 の相異なるウォルシュ符号に符号化し、排他的論理和器により論理和がとられた 16 個の符号化シンボルから構成された符号化シンボル列を利用して 24 個の符号化シンボルから構成された符号化シンボル列に出力する移動通信システムでの符号化装置において、

前記排他的論理和器からの前記 16 個の符号化シンボルから構成された符号化シンボル列を 1 回反復して 32 個の符号化シンボルから構成された符号化シンボル列を出力する反復器と、

10

前記符号化シンボル列の 32 個の符号化シンボルのうち、0、1、2、3、4、5、6、16 番目のシンボルを穿孔して前記 24 個の符号化シンボルから構成された前記符号化シンボル列を出力する穿孔器と、を含むことを特徴とする装置。

【請求項 5】

前記穿孔器により穿孔される穿孔位置 0、1、2、3、4、5、6、16 を貯蔵するメモリをさらに備えることを特徴とする請求項 4 に記載の装置。

【請求項 6】

7 ビットの入力情報ビットを受信し、前記 7 ビットの入力情報ビットそれぞれを長さ 32 の相異なるウォルシュ符号とマスク関数に符号化し、排他的論理和器により論理和がとられた 32 個の符号化シンボルから構成された符号化シンボル列を利用して 24 個の符号化シンボルから構成された符号化シンボル列に出力する移動通信システムでの符号化装置において、

20

前記排他的論理和器からの前記 32 個の符号化シンボルのうち、0、4、8、12、16、20、24、28 番目のシンボルを穿孔して前記 24 個の符号化シンボルから構成された前記符号化シンボル列を出力する穿孔器を含むことを特徴とする装置。

【請求項 7】

前記穿孔器により穿孔される穿孔位置 0、4、8、12、16、20、24、28 を貯蔵するメモリをさらに備えることを特徴とする請求項 6 に記載の装置。

30

【請求項 8】

前記マスク関数は、‘ 0 1 1 1 0 1 1 1 0 0 1 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 ’と‘ 0 0 1 0 0 1 1 0 0 1 0 1 0 1 0 0 0 0 1 0 1 0 1 0 0 0 0 1 0 0 0 0 ’であることを特徴とする請求項 6 に記載の装置。

【請求項 9】

4 ビット、または 7 ビットの入力情報ビットを受信し、前記 4 ビット、または 7 ビットの入力情報ビットそれぞれを長さ 16、または 32 の相異なるウォルシュ符号、またはマスクに符号化して、24 個の符号化シンボルから構成された符号化シンボル列に出力する移動通信システムでの符号化装置において、

前記入力情報ビットに 4 ビットが入力されるか、7 ビットが入力されるかを検査して前記符号化のための制御を遂行する制御部と、

40

前記制御部の制御下に前記長さ 16、または前記長さ 32 を有する相異なる 5 個のウォルシュ符号を選択的に出力するウォルシュ符号発生器と、

前記制御部の制御下に前記長さ 16、または前記長さ 32 を有する相異なる 2 個のマスクを選択的に出力するマスク発生器と、

前記 4 ビット、または 7 ビットの入力情報ビットと前記ウォルシュ符号発生器及び前記マスク発生器からのウォルシュ符号及びマスクを一对一に乗算して、16 個、または 32 個の符号化シンボルから構成された符号化シンボル列を出力する乗算器と、

前記乗算器からの符号化シンボル列の排他的論理和をとって一つの符号化シンボル列を出力する排他的論理和器と、

50

前記制御部の制御下に前記排他的論理和器からの前記符号化シンボル列を所定回数反復して、32個の符号化シンボルから構成された符号化シンボル列を出力する反復器と、

前記4ビットの入力情報ビットに対応した8個の穿孔位置と前記7ビットの入力情報ビットに対応した8個の穿孔位置を貯蔵するメモリと、

前記反復器からの前記32個の符号化シンボルから構成された符号化シンボル列を受信し、前記制御部の制御下に前記32個の符号化シンボルのうち、前記メモリから読み出した8個の該当穿孔位置の符号化シンボルを穿孔して前記24個の符号化シンボルから構成された前記符号化シンボル列を出力する穿孔器と、を含むことを特徴とする装置。

【請求項10】

前記制御部は、前記入力情報ビットに4ビットが入力されると、前記ウォルシュ符号発生器と前記マスク発生器が長さ16を有する相異なる7個のウォルシュ符号とマスクを出力するように制御し、前記反復器の反復回数を1回に決定し、前記穿孔器には前記4ビットの入力情報ビットに対応した8個の穿孔位置を穿孔するように制御することを特徴とする請求項9に記載の装置。

10

【請求項11】

前記制御部は、前記入力情報ビットに7ビットが入力されると、前記ウォルシュ符号発生器と前記マスク発生器が長さ32を有する相異なる7個のウォルシュ符号とマスクを出力するように制御し、前記反復器の反復回数を0回に決定し、前記穿孔器には前記7ビットの入力情報ビットに対応した8個の穿孔位置を穿孔するように制御することを特徴とする請求項9に記載の装置。

20

【請求項12】

前記4ビットの入力情報ビットに対応して前記メモリに貯蔵された8個の穿孔位置は、0、1、2、3、4、5、6、16番目のシンボルであることを特徴とする請求項9に記載の装置。

【請求項13】

前記7ビットの入力情報ビットに対応して前記メモリに貯蔵された8個の穿孔位置は、0、4、8、12、16、20、24、28番目のシンボルであることを特徴とする請求項11に記載の装置。

【請求項14】

前記7ビットの入力情報ビットに対応して前記メモリに貯蔵された8個の穿孔位置は、0、1、2、3、4、5、6、7番目のシンボルであることを特徴とする請求項11に記載の装置。

30

【請求項15】

前記マスク関数は、‘0111 0111 0010 0100 0110 0000 0000 0000’と‘0010 0110 0101 0100 0101 0100 0100 0100’であることを特徴とする請求項13に記載の装置。

【請求項16】

前記マスク関数は、‘0000 0000 1110 1000 1101 1000 1100 0000’と‘0000 0000 1100 0000 0111 1110 0010 1000’であることを特徴とする請求項14に記載の装置。

40

【請求項17】

4ビット、または7ビットの入力情報ビットを受信し、前記4ビット、または7ビットの入力情報ビットそれぞれを長さ16、または32の相異なるウォルシュ符号、またはマスクに符号化して24個の符号化シンボルから構成された符号化シンボル列に出力する移动通信システムでの符号化方法において、

前記入力情報ビットに4ビットが入力されるか、7ビットが入力されるかを検査する過程と、

前記入力情報ビットに4ビットが入力される場合、前記4ビットの入力情報ビットそれぞれを長さ16の相異なるウォルシュ符号に符号化して排他的論理和器により論理和がとられた16個の符号化シンボルから構成された符号化シンボル列を反復及び穿孔により2

50

4 個の符号化シンボルから構成された符号化シンボル列に出力する第 1 符号化過程と、

前記入力情報ビットに 7 ビットが入力される場合、前記 7 ビットの入力情報ビットそれぞれを長さ 3 2 の相異なるウォルシュ符号とマスクに符号化して排他的論理和器により論理和がとられた 3 2 個の符号化シンボルから構成された符号化シンボル列を穿孔により 2 4 個の符号化シンボルから構成された符号化シンボル列に出力する第 2 符号化過程と、を含むことを特徴とする方法。

【請求項 1 8】

前記第 1 符号化過程は、

前記 1 6 個の符号化シンボルから構成された符号化シンボル列を 1 回反復して 3 2 個の符号化シンボルから構成された符号化シンボル列を出力する段階と、

10

前記 3 2 個の符号化シンボルのうち、0、1、2、3、4、5、6、1 6 番目のシンボルを穿孔して前記 2 4 個の符号化シンボルから構成された前記符号化シンボル列を出力する段階と、を含むことを特徴とする請求項 1 7 に記載の方法。

【請求項 1 9】

前記第 2 符号化過程は、

前記 3 2 個の符号化シンボルのうち、0、4、8、1 2、1 6、2 0、2 4、2 8 番目のシンボルを穿孔して前記 2 4 個の符号化シンボルから構成された前記符号化シンボル列を出力する段階を含むことを特徴とする請求項 1 7 に記載の方法。

【請求項 2 0】

前記第 2 符号化過程は、

20

前記 3 2 個の符号化シンボルのうち、0、1、2、3、4、5、6、7 番目のシンボルを穿孔して前記 2 4 個の符号化シンボルから構成された前記符号化シンボル列を出力する段階を含むことを特徴とする請求項 1 7 に記載の方法。

【請求項 2 1】

7 ビットの入力情報ビットを受信し、前記 7 ビットの入力情報ビットそれぞれを長さ 3 2 の相異なるウォルシュ符号とマスク関数に符号化して排他的論理和器により論理和がとられた 3 2 個の符号化シンボルから構成された符号化シンボル列を 2 4 個の符号化シンボルから構成された符号化シンボル列に出力する移動通信システムでの符号化方法において、

前記 3 2 個の符号化シンボルのうち、0、1、2、3、4、5、6、7 番目のシンボルを穿孔して前記 2 4 個の符号化シンボルから構成された前記符号化シンボル列を出力する過程を含むことを特徴とする方法。

30

【請求項 2 2】

前記マスク関数は、‘0 1 1 1 0 1 1 1 0 0 1 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0’ と ‘0 0 1 0 0 1 1 0 0 1 0 1 0 1 0 0 0 1 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0’ であることを特徴とする請求項 2 1 に記載の方法。

【請求項 2 3】

7 ビットの入力情報ビットを受信し、前記 7 ビットの入力情報ビットそれぞれを長さ 3 2 の相異なるウォルシュ符号とマスク関数に符号化して排他的論理和器により論理和がとられた 3 2 個の符号化シンボルから構成された符号化シンボル列を 2 4 個の符号化シンボルから構成された符号化シンボル列に出力する移動通信システムでの符号化装置において、

40

前記排他的論理和器からの前記 3 2 個の符号化シンボルのうち、0、1、2、3、4、5、6、7 番目のシンボルを穿孔して前記 2 4 個の符号化シンボルから構成された前記符号化シンボル列を出力する穿孔器を含むことを特徴とする装置。

【請求項 2 4】

前記穿孔器により穿孔される穿孔位置 0、1、2、3、4、5、6、7 を貯蔵するメモリをさらに備えることを特徴とする請求項 2 3 に記載の装置。

【請求項 2 5】

前記マスク関数は、‘0 1 1 1 0 1 1 1 0 0 1 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0’

50

0 0 0 0 0 0 'と' 0 0 1 0 0 1 1 0 0 1 0 1 0 1 0 0 0 1 0 1 0 1 0 0 0 1 0 0 0 0 0 'であることを特徴とする請求項 2 3 に記載の方法。

【請求項 2 6】

4 ビット、または 7 ビットの入力情報ビットを受信し、前記 4 ビット、または 7 ビットの入力情報ビットそれぞれを長さ 3 2 の相異なるウォルシュ符号、またはマスクに符号化して 2 4 個の符号化シンボルから構成された符号化シンボル列に出力する移動通信システムでの符号化装置において、

前記入力情報ビットに 4 ビットが入力されるか、7 ビットが入力されるかを検査して前記符号化のための制御を遂行する制御部と、

前記長さ 3 2 を有する相異なる 5 個のウォルシュ符号を選択的に出力するウォルシュ符号発生器と、 10

前記長さ 3 2 を有する相異なる 2 個のマスクを選択的に出力するマスク発生器と、

前記 4 ビット、または 7 ビットの入力情報ビットと前記ウォルシュ符号発生器及び前記マスク発生器からのウォルシュ符号及びマスクを一对一に乗算して 3 2 個の符号化シンボルから構成された符号化シンボル列を出力する乗算器と、

前記乗算器からの符号化シンボル列の排他的論理和をとって一つの符号化シンボル列を出力する排他的論理和器と、

前記 4 ビットの入力情報ビットに対応した 8 個の穿孔位置と前記 7 ビットの入力情報ビットに対応した 8 個の穿孔位置を貯蔵するメモリと、

前記排他的論理和器からの前記 3 2 個の符号化シンボルから構成された符号化シンボル列を受信し、前記制御部の制御下に前記 3 2 個の符号化シンボルのうち、前記メモリから読み出した 8 個の該当穿孔位置の符号化シンボルを穿孔して前記 2 4 個の符号化シンボルから構成された前記符号化シンボル列を出力する穿孔器と、を含むことを特徴とする装置 20

【請求項 2 7】

前記入力情報ビットに 4 ビットが入力されると、前記穿孔器により前記 4 ビットの入力情報ビットに対応した 0、1、2、3、4、5、6、1 6 番目のシンボルを穿孔することを特徴とする請求項 2 6 に記載の装置。

【請求項 2 8】

前記入力情報ビットに 7 ビットが入力されると、前記穿孔器により前記 7 ビットの入力情報ビットに対応した 0、4、8、1 2、1 6、2 0、2 4、2 8 番目のシンボルを穿孔することを特徴とする請求項 2 6 に記載の装置。 30

【請求項 2 9】

前記入力情報ビットに 7 ビットが入力されると、前記穿孔器により前記 7 ビットの入力情報ビットに対応した 0、1、2、3、4、5、6、7 番目のシンボルを穿孔することを特徴とする請求項 2 6 に記載の装置。

【請求項 3 0】

4 ビット、または 7 ビットの入力情報ビットを受信し、前記 4 ビット、または 7 ビットの入力情報ビットそれぞれを長さ 3 2 の相異なるウォルシュ符号、またはマスクに符号化して 2 4 個の符号化シンボルから構成された符号化シンボル列に出力する移動通信システムでの符号化方法において、 40

前記入力情報ビットに 4 ビットが入力されるか、7 ビットが入力されるかを検査する過程と、

前記 4 ビット、または前記 7 ビットの入力情報ビットが受信される場合、前記 4 ビット、または前記 7 ビットの入力情報ビットそれぞれを長さ 3 2 の相異なるウォルシュ符号に符号化して排他的論理和器により論理和がとられた 3 2 個の符号化シンボルから構成された符号化シンボル列を出力する過程と、

前記 4 ビットの入力情報ビットによる前記符号化シンボル列が受信される場合、前記符号化シンボル列を構成する前記 3 2 個の符号化シンボルのうち、0、1、2、3、4、5、6、1 6 番目のシンボルを穿孔して前記 2 4 個の符号化シンボルから構成された前記符 50

号化シンボル列を出力する過程と、

前記7ビットの入力情報ビットによる前記符号化シンボル列が受信される場合、前記符号化シンボル列を構成する前記32個の符号化シンボルのうち、0、4、8、12、16、20、24、28番目のシンボルを穿孔して前記24個の符号化シンボルから構成された前記符号化シンボル列を出力する過程と、を含むことを特徴とする方法。

【請求項31】

4ビット、または7ビットの入力情報ビットを受信し、前記4ビット、または7ビットの入力情報ビットそれぞれを長さ32の相異なるウォルシュ符号、またはマスクに符号化して24個の符号化シンボルから構成された符号化シンボル列に出力する移動通信システムでの符号化方法において、

10

前記入力情報ビットに4ビットが入力されるか、7ビットが入力されるかを検査する過程と、

前記4ビット、または前記7ビットの入力情報ビットが受信される場合、前記4ビット、または前記7ビットの入力情報ビットそれぞれを長さ32の相異なるウォルシュ符号に符号化して排他的論理和器により論理和がとられた32個の符号化シンボルから構成された符号化シンボル列を出力する過程と、

前記4ビットの入力情報ビットによる前記符号化シンボル列が受信される場合、前記符号化シンボル列を構成する前記32個の符号化シンボルのうち、0、1、2、3、4、5、6、16番目のシンボルを穿孔して前記24個の符号化シンボルから構成された前記符号化シンボル列を出力する過程と、

20

前記7ビットの入力情報ビットによる前記符号化シンボル列が受信される場合、前記符号化シンボル列を構成する前記32個の符号化シンボルのうち、0、1、2、3、4、5、6、7番目のシンボルを穿孔して前記24個の符号化シンボルから構成された前記符号化シンボル列を出力する過程と、を含むことを特徴とする方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は符号分割多重接続移動通信システムでの符号化／復号化装置及び方法に関するもので、特に、同期移動通信システムで使用する逆方向データ速度インジケータチャネル(Reverse Rate Indicator Channel、R-RICH)を伝送する装置及び方法に関する。

30

【背景技術】

【0002】

通常的に逆方向パケットチャネル(Reverse Supplemental Channel、R-SCCH)では基本的に可変速度伝送方式を支援している。ここで、可変速度伝送方式とは、端末が伝送するデータ速度を任意に変更できることを意味する。一般的に、データ速度が変わると、フレーム構成に使用される誤り訂正符号の符号速度、シンボル反復回数、帯域拡散のためのウォルシュ符号などの長さ及び種類などが変更される。従って、端末は現在伝送している逆方向パケットチャネルのデータ速度を基地局に知らせないと、基地局受信器が前記逆方向パケットチャネルを正しく受信できない。このような用途のため定義されたチャネルを逆方向データ速度インジケータチャネル(Reverse Rate Indicator Channel、R-RICH)と称する。

40

【0003】

前記逆方向パケットチャネルを通じて端末が伝送することができるデータ速度の数は、端末が同時に使用することができる逆方向パケットチャネルの個数によって変わる。前記逆方向パケットチャネルの個数は、呼設定時に逆方向に伝送するデータの量などを考慮して基地局により決定され、端末に通報される。従って、逆方向データ速度インジケータチャネルに伝送される情報ビットの数は、逆方向パケットチャネルの個数によって変更される。即ち、逆方向パケットチャネルの個数が1個である場合は、4ビットを利用して逆方向データ速度を知らせ、逆方向パケットチャネルの個数が2個である場合は、7ビットを

50

利用して逆方向データ速度を知らせる。一方、端末が同時に使用することができる逆方向パケットチャネルの個数は、基地局から別の指示を受信するまでは変更することができないので、逆方向データ速度インジケータチャネルには常に4ビット情報が伝送されるか、常に7ビット情報が伝送される。即ち、前記4ビット情報と7ビット情報が同時に伝送される場合はない。従来技術では前記逆方向データ速度インジケータチャネルで使用する誤り訂正符号に(24、4)、または(24、7)符号を定義している。

【0004】

図1は前記逆方向データ速度インジケータチャネルの送信器構造を示す図である。前記図1を参照すると、4ビット、または7ビットのデータ速度インジケータは符号器100に入力される。前記符号器100は前記データ速度インジケータを符号化して24個の符号化シンボルを出力する。すると、前記出力された符号化シンボルはシンボル反復器110に入力され、前記シンボル反復器110は前記入力された24個の符号化シンボルを16回反復して出力する。前記反復された符号化シンボルは信号変換器120に入力され信号変換されたシンボルが出力される。前記信号変換器120で遂行される信号変換は入力された信号に対して0は1に変換し、1は-1に変換して出力する動作により遂行される。前記信号変換されたシンボルは拡散器130に入力され、拡散された後に出力される。

【0005】

前記図1から分かるように、データ速度インジケータは4ビット、または7ビットに表現され、伝送前に24個の符号化シンボルに符号化される。この時、前記符号化シンボルに符号化された前記データ速度インジケータの伝送中にエラーが発生する場合、これに対応する逆方向パケットチャネルの符号速度及びシンボル反復回数、帯域拡散のためのウォルシュ符号の長さ及び種類などを不正確に示すようになり、その結果、受信器で逆方向パケットチャネルの正しい解釈が不可能になる。従って、前記データ速度インジケータは優秀な性能を有する(24、4)、または(24、7)符号器により符号化されるべきである。また、前記対応するパケットチャネルを解釈するためには最大限迅速に復号化されるべきである。

【発明の開示】

【発明が解決しようとする課題】

【0006】

従って、上述したような問題点を解決するための本発明の目的は、最適の性能を有するデータ速度インジケータ符号化方法及び装置を提供することにある。

【0007】

本発明の他の目的は、最小の複雑度を有するデータ速度インジケータ符号化方法及び装置を提供することにある。

【0008】

本発明のさらに他の目的は、拡張された一次リードミュラー符号を穿孔する方法を使用することにより、復号化過程で逆アーダマルを使用できるようになって、ハードウェア複雑度(Hardware Complexity)を最小化することができる方法及び装置を提供することにある。

【0009】

本発明のさらに他の目的は、拡張された一次リードミュラー符号を穿孔する方法を使用することにより、復号化過程で逆アーダマルを使用できるようになって、最適の符号語を使用することができる装置及び方法を提供することにある。

【0010】

本発明のさらに他の目的は、穿孔前の直交符号の長さを最小化することにより、ハードウェア複雑度を最小化することができる装置及び方法を提供することにある。

【0011】

本発明のさらに他の目的は、拡張された直交符号を穿孔することにより、ハードウェア複雑度を最小化するだけでなく、誤り訂正性能においても最適の符号を生成することができる装置及び方法を提供することにある。

10

20

30

40

50

【0012】

本発明のさらに他の目的は、ハードウェア複雑度を最小化し、誤り訂正性能において最適の符号を生成するために、(24、4)符号化及び(24、7)符号化を兼用する装置及び方法を提供することにある。

【課題を解決するための手段】

【0013】

前記目的を達成するための本発明の方法は、4ビットの入力情報ビットを受信し、前記4ビットの入力情報ビットそれぞれを長さ16の相異なるウォルシュ符号に符号化し、排他的論理和器(XOR operator)により論理和がとられた(XORing)16個の符号化シンボルから構成された符号化シンボル列を利用して24個の符号化シンボルから構成された符号化シンボル列を出力する移動通信システムでの符号化方法を提供する。前記符号化方法は、前記16個の符号化シンボルから構成された符号化シンボル列を1回反復して32個の符号化シンボルから構成された符号化シンボル列を出力する過程と、前記32個の符号化シンボルのうち、0、1、2、3、4、5、6、16番目のシンボルを穿孔して前記24個の符号化シンボルから構成された前記符号化シンボル列を出力する過程と、を含むことを特徴とする。

10

【0014】

前記目的を達成するための本発明の方法は、7ビットの入力情報ビットを受信し、前記7ビットの入力情報ビットそれぞれを長さ32の相異なるウォルシュ符号とマスク関数に符号化し、排他的論理和器により論理和がとられた32個の符号化シンボルから構成された符号化シンボル列を利用して24個の符号化シンボルから構成された符号化シンボル列に出力する移動通信システムでの符号化方法を提供する。前記符号化方法は、前記32個の符号化シンボルのうち、0、4、8、12、16、20、24、28番目のシンボルを穿孔して前記24個の符号化シンボルから構成された前記符号化シンボル列を出力する過程を含むことを特徴とする。

20

【0015】

前記目的を達成するための本発明の装置は、4ビット、または7ビットの入力情報ビットを受信し、前記4ビット、または7ビットの入力情報ビットそれぞれを長さ16、または32の相異なるウォルシュ符号、またはマスクに符号化して、24個の符号化シンボルから構成された符号化シンボル列に出力する移動通信システムでの符号化装置を提供する。前記符号化装置は、前記入力情報ビットに4ビットが入力されるか、7ビットが入力されるかを検査して前記符号化のための制御を遂行する制御部と、前記制御部の制御下に前記長さ16、または前記長さ32を有する相異なる5個のウォルシュ符号を選択的に出力するウォルシュ符号発生器と、前記制御部の制御下に前記長さ16、または前記長さ32を有する相異なる2個のマスクを選択的に出力するマスク発生器と、前記4ビット、または7ビットの入力情報ビットと前記ウォルシュ符号発生器及び前記マスク発生器からのウォルシュ符号及びマスクを一对一に乗算して、16個、または32個の符号化シンボルから構成された符号化シンボル列を出力する乗算器と、前記乗算器からの符号化シンボル列の排他的論理和をとって一つの符号化シンボル列を出力する排他的論理和器と、前記制御部の制御下に前記排他的論理和器からの前記符号化シンボル列を所定回数反復して、32個の符号化シンボルから構成された符号化シンボル列を出力する反復器と、前記4ビットの入力情報ビットに対応した8個の穿孔位置と前記7ビットの入力情報ビットに対応した8個の穿孔位置を貯蔵するメモリと、前記反復器からの前記32個の符号化シンボルから構成された符号化シンボル列を受信し、前記制御部の制御下に前記32個の符号化シンボルのうち、前記メモリから読み出した8個の該当穿孔位置の符号化シンボルを穿孔して前記24個の符号化シンボルから構成された前記符号化シンボル列を出力する穿孔器と、を含むことを特徴とする。

30

40

【0016】

前記目的を達成するための本発明の装置は、4ビット、または7ビットの入力情報ビットを受信し、前記4ビット、または7ビットの入力情報ビットそれぞれを長さ32の相異

50

なるウォルシュ符号、またはマスクに符号化して24個の符号化シンボルから構成された符号化シンボル列に出力する移動通信システムでの符号化装置を提供する。前記符号化装置は、前記入力情報ビットに4ビットが入力されるか、7ビットが入力されるかを検査して前記符号化のための制御を遂行する制御部と、前記長さ32を有する相異なる5個のウォルシュ符号を選択的に出力するウォルシュ符号発生器と、前記長さ32を有する相異なる2個のマスクを選択的に出力するマスク発生器と、前記4ビット、または7ビットの入力情報ビットと前記ウォルシュ符号発生器及び前記マスク発生器からのウォルシュ符号及びマスクを一对一に乗算して32個の符号化シンボルから構成された符号化シンボル列を出力する乗算器と、前記乗算器からの符号化シンボル列の排他的論理和をとって一つの符号化シンボル列を出力する排他的論理和器と、前記4ビットの入力情報ビットに対応した8個の穿孔位置と前記7ビットの入力情報ビットに対応した8個の穿孔位置を貯蔵するメモリと、前記排他的論理和器からの前記32個の符号化シンボルから構成された符号化シンボル列を受信し、前記制御部の制御下に前記32個の符号化シンボルのうち、前記メモリから読み出した8個の該当穿孔位置の符号化シンボルを穿孔して前記24個の符号化シンボルから構成された前記符号化シンボル列を出力する穿孔器と、を含むことを特徴とする。

10

【発明の効果】**【0017】**

上述したように本発明は拡張された一次リードミュラー符号を穿孔する装置及び方法を使用することにより、最適の性能と最小の複雑度を有するデータ速度インジケータ符号化方法及び装置を具現することができるだけでなく、最適の符号語を使用することができるという効果を有する。また、本発明は拡張された一次リードミュラー符号を穿孔する装置及び方法を使用することにより、復号化過程で逆アーダマルの使用ができるようになり、ハードウェア複雑度(Hardware Complexity)を最小化することができ、誤り訂正性能においても最適の符号を生成することができるという効果がある。最後に本発明は(24、4)符号器と(24、7)符号器を兼用に支援することによって効率的な符号化を遂行することができるという利点がある。

20

【発明を実施するための最良の形態】**【0018】**

以下、本発明の望ましい実施形態について添付図を参照しつつ詳細に説明する。下記の発明において、本発明の要旨のみを明瞭にする目的で、関連した公知機能又は構成に関する具体的な説明は省略する。

30

【0019】

通常的に、線形誤り訂正符号(Linear Error Correcting Code)の性能を示す尺度(measure)としては、誤り訂正符号の符号語(code word)のハミング距離(Hamming distance)分布がある。これはそれぞれの符号語で0でないシンボルの個数を意味する。例えば、所定符号語0111において、前記符号語のハミング距離は前記符号語に含まれた1の個数、即ち3が前記符号語のハミング距離である。この時、各種符号語のハミング距離値のうち、一番小さい値を最小距離($d_{\min}$: minimum distance)という。一方、前記誤り訂正符号の誤り訂正性能は前記符号語の最小距離が大きいほど優秀であることは自明であろう。これは参照文献“The Theory of Error-Correcting Codes”-F. J. Macwilliams, N. J. A. Sloane, North-Hollandに詳細に開示されている。また、最適符号(optimal code)になるための二進線形符号の入力と出力値による符号間の最小距離は、参照文献[1]“An Updated Table of Minimum-Distance Bounds for Binary Linear Codes”(A. E. Brouwer and Tom Verhoeff, IEEE Transactions on information Theory, VOL.39, NO.2, MARCH 1993)に開示されている。

40

【0020】

この時、通常的に前記データ速度インジケータを符号化する符号器には、(24、4)符号器、または(24、7)符号器が使用され、前記参照文献[1]を参照すると、入力が4ビットであり、出力が24である最適の(24、4)線形符号器は12の最小距離を有し、

50

入力が7ビットであり、出力が24である最適の(24、7)線形符号器は10の最小距離を有する。

【0021】

先ず、前記データ速度インジケータを符号化する(24、4)符号器について説明する。

【0022】

前記(24、4)符号器は、(15、4)シンプレックス符号を2回反復し、6シンボルを穿孔することにより、最適の(24、4)符号を得るようにする構成である。実際に、前記(24、4)符号を生成することができる多様な方法があるが、本発明の実施形態による一次リードミュラー符号を穿孔する方法を使用することにより、ハードウェア複雑度(Hardware Complexity)を最小化することができるだけでなく、最適の符号語を生成すること
10
ができる。また、前記穿孔前のシンプレックス符号の長さを最小化することにより、ハードウェア複雑度を最小化することができる。また、前記シンプレックス符号を穿孔することによりハードウェア複雑度を最小化することができるだけでなく、誤り訂正性能においても最適の符号を生成することができる。本発明の実施形態ではシンプレックス符号を利用して誤り訂正符号を生成すると仮定する。

【0023】

上述したように、(24、4)符号器から出力される符号語は、(15、4)シンプレックス符号発生器で出力された15個の符号化シンボルを2回反復して総長さ30のシンプレックス符号シンボルを出力し、前記シンプレックス符号シンボルのうち、6シンボルを穿孔したものである。ここで、前記反復された総長さ30のシンプレックス符号シンボルで
20
前記6シンボルを穿孔する穿孔位置を変更すると、符号語の最小距離($d_{\min}$: minimum distance)が相異なるようになる。そのため、前記(15、4)シンプレックス符号で優秀な誤り訂正性能を有する(24、4)符号器を生成するためには、一番大きな最小距離を得ることができる穿孔位置を求めるのが重要である。

【0024】

最適の(24、4)線形符号を生成するために必要な6個の穿孔位置は実験的に求めることができるが、その中、一番簡単な場合は{0、1、2、3、4、5}である。この場合、4個の情報ビットを伝送するために、本発明の符号化／復号化方法を使用する移動通信システムの送信器と受信器は、予め6個の穿孔位置を約束しているべきである。これは通信規格で決定するのが一般的であり、送信側が予め穿孔位置を知らせることも可能である
30
。

【0025】

上述したように、本発明による最適(24、4)符号の符号化について図2を参照して説明する。

【0026】

図2は本発明の実施形態による送信器に設けられた符号器の構造を示すブロック図である。前記図2を参照すると、4ビットの入力情報ビット(a_0 、 a_1 、 a_2 、 a_3)は、(15、4)シンプレックス符号器200に入力される。ここで、前記入力情報ビットはデータ速度インジケータである。前記シンプレックス符号器(リードミュラー符号器)200は、前記4ビットの入力情報ビット(a_0 、 a_1 、 a_2 、 a_3)を符号化して長さ15の符号化シンボル(符号シンボル列)に出力する。前記符号化シンボルはシンプレックス符号である。前記15個の符号化シンボルは反復器210に入力される。前記反復器210は前記受信された符号化シンボルを2回反復して30個の符号化シンボルを出力する。前記30個の符号化シンボルが穿孔器220に入力されると、前記穿孔器220は前記6個の最適の穿孔位置である0、1、2、3、4、5番目のシンボルを穿孔して24個の符号化シンボルを出力する。
40

【0027】

一般的な符号理論では入力情報と符号化シンボル間の対応関係を示すために生成行列(Generating Matrix)を定義する。前記のようにシンボル反復及び穿孔が含まれる場合、最終的な(24、4)符号器に対する生成行列は、下記の<数1>のようになる。
50

【0028】

【数1】

$$M = \begin{bmatrix} 101010101101010101010101 \\ 100110011011001100110011 \\ 100001111000111100001111 \\ 011111111000000011111111 \end{bmatrix}$$

【0029】

前記<数1>の生成行列は、前記4ビットの入力信号のうち、一番目の入力信号が1であると、一番目行の24個のシンボルを選択し、一番目の入力信号が0であると選択しない。前記4ビットの入力信号のうち、二番目の入力信号が1であると、二番目行の24個のシンボルを選択し、0であると選択しない。前記4ビットの入力信号のうち、三番目の入力信号が1であると、三番目行の24個のシンボルを選択し、0であると選択しない。前記4ビットの入力信号のうち、四番目の入力信号が1であると、四番目行の24個のシンボルを選択し、0であると選択しない。

10

【0030】

図7は前記生成行列による(24、4)符号器を示す図である。

【0031】

前記図7を参照すると、0、または1の値を有する入力情報ビットa0、a1、a2、a3が入力されると、a0は乗算器920に、a1は乗算器922に、a2は乗算器924に、a3は乗算器926にそれぞれ入力される。これと同時に前記信号発生器900はメモリに貯蔵されている前記生成行列を構成する4個のシンボル列を前記乗算器のうち、それぞれ対応する一つの乗算器に出力する。即ち、前記生成行列の一番目列に該当する長さ24であるシンボル列R1=1010 1010 1101 0101 0101 0101をメモリに貯蔵し、乗算器920に出力し、前記生成行列の二番目列に該当する長さ24であるシンボル列R2=1001 1001 1011 0011 0011 0011をメモリに貯蔵し、乗算器922に出力する。また、前記生成行列の三番目列に該当する長さ24であるシンボル列R3=1000 0111 1000 1111 0000 1111をメモリに貯蔵し、乗算器924に出力し、前記生成行列の四番目列に該当する長さ24であるシンボル列R4=0111 1111 1000 0000 1111 1111をメモリに貯蔵し、乗算器926に出力する。すると、前記乗算器920は前記シンボル列R1のそれぞれのシンボルと前記入力情報ビットa0を乗算して長さ24であるシンボル列を排他的論理和器940に出力し、前記乗算器922は前記シンボル列R2のそれぞれのシンボルと前記入力信号a1を乗算して長さ24であるシンボル列を前記排他的論理和器940に出力する。また、前記乗算器924は前記シンボル列R3のそれぞれのシンボルと前記入力信号a2を乗算して長さ24であるシンボル列を前記排他的論理和器940に出力し、前記乗算器926は前記シンボル列R4のそれぞれのシンボルと前記入力信号a3を乗算して長さ24であるシンボル列を前記排他的論理和器940に出力する。前記排他的論理和器940は前記4個の長さ24であるシンボル列をシンボル単位に排他的論理和をとって長さ24である符号化シンボル列を出力する。

20

30

40

【0032】

次に、前記データ速度インジケータを符号化する(24、7)符号器について説明する。

【0033】

前記(24、7)符号器は、(32、5)直交符号(一次リードミュラー符号(First order Reed Muller code)に2個のマスク(mask)関数を使用して符号語を拡張した拡張された直交符号で8シンボルを穿孔することにより、最適の(24、7)符号を得るようにする構成である。

【0034】

図3は、前記拡張された直交符号の構造を示している。前記図3を参照すると、M1と

50

M2を前記使用される2個のマスク関数とする場合、上位32個の符号語は前記32個の長さ32である直交符号語(W)を使用する。次の32個の符号語は、前記マスク関数M1と前記32個の直交符号語(W)をそれぞれ排他的論理和をとった32個の符号語(M1+W)を使用し、その次の32個の符号語は、前記マスク関数M2と前記32個の直交符号語(W)をそれぞれ排他的論理和をとった32個の符号語(M2+W)を使用する。最後の32個の符号語には前記マスク関数M1、M2と前記32個の直交符号語(W)を排他的論理和をとった32個の符号語(M1+M2+W)を使用する。従って、拡張された直交符号には総計 $2^7 = 128$ 個の符号語を使用することになる。この時、(24、7)符号を最適化するための前記2個のマスク関数は実験的に求めることができる。

【0035】

例えば、前記2個のマスク関数M1、M2は、次のようになる。

M1 = 0111 0111 0010 0100 0110 0000 00000000

M2 = 0010 0110 0101 0100 0101 0100 01000000

【0036】

一般的に、線形(24、7)符号を生成することができる多くの方法があるが、本発明の実施形態による拡張された一次リードミューラー符号を穿孔する方法を使用することにより、ハードウェア複雑度(Hardware Complexity)を最小化することができるだけでなく、最適の符号語を使用することができる。また、前記穿孔前の直交符号の長さを最小化することにより、ハードウェア複雑度を最小化することができる。また、前記拡張された直交符号を穿孔することによりハードウェア複雑度を最小化することができるだけでなく、誤り訂正性能においても最適符号を生成することができる。本発明の実施形態では拡張された直交符号を有して誤り訂正符号を生成すると仮定する。

【0037】

上述したように、(24、7)符号語は長さ(32、7)拡張された直交符号発生器で出力された32個の符号化シンボルのうち、8シンボルを穿孔したものである。ここで、前記32個の拡張された直交符号シンボルのうちで、前記8シンボルを穿孔する穿孔位置を変更すると、符号語の最小距離($d_{\min}$: minimum distance)が変わる。そのため、前記(32、7)拡張された直交符号で優秀な誤り訂正性能を有する(24、7)符号器を生成するためには、一番大きな最小距離を得ることができるようにする穿孔位置を求めることが重要である。

【0038】

前記のような(24、7)線形符号として最適の符号を生成するために必要な8個の穿孔位置は実験的に求めることができるが、一番簡単な場合は{0、4、8、12、16、20、24、28}である。この時、7個の情報ビットを伝送する本発明の符号化／復号化方法を使用する移動通信システムの送信器と受信器は8個の穿孔位置を予め約束しているべきである。これは通信規格で決定するのが一般的であり、送信側が予め穿孔位置を知らせることも可能である。

【0039】

図4は、本発明の実施形態による送信器に設けられた最適(24、7)符号器の構成を示している。

【0040】

前記図4を参照すると、7ビットの入力情報ビットa0、a1、a2、a3、a4、a5、a6は、(32、7)符号器400の拡張された直交符号器400に入力される。前記7ビットの入力情報ビットはデータ速度インジケータである。前記7ビットの入力情報ビットa0、a1、a2、a3、a4、a5、a6を受信した前記拡張された直交符号器400は、前記7ビットの入力情報ビットa0、a1、a2、a3、a4、a5、a6を符号化して長さ32の符号化シンボル(符号シンボル列)を出力する。前記拡張された直交符号器400から出力される前記32個の符号化シンボルは、穿孔器410に入力される。前記穿孔器410は前記32個の符号化シンボルで前記8個の最適の穿孔位置である0、4、8、12、16、20、24、28番目のシンボルを穿孔して24個の符号化シンボ

10

20

30

40

50

【 0 0 4 1 】

【 0 0 4 2 】

【数 2】

$$M = \begin{bmatrix} 101101101101101101101101 \\ 011011011011011011011011 \\ 000111000111000111000111 \\ 000000111111000000111111 \\ 000000000000111111111111 \\ 111111010100110000000000 \\ 010110101100101100100000 \end{bmatrix}$$

10

【 0 0 4 3 】

20

【 0 0 4 4 】

図 8 は前記生成行列による (24、7) 符号器を示す。

【 0 0 4 5 】

40

50

28に出力し、前記生成行列の六番目列に該当する長さ24であるシンボル列 $R_6 = 1111110101001100000000001111$ をメモリに貯蔵し、乗算器1030に出力する。最後に、前記生成行列の七番目列に該当する長さ24であるシンボル列 $R_7 = 010110101100101100100000$ をメモリに貯蔵し、乗算器1032に出力する。すると、前記乗算器1020は前記シンボル列 R_1 それぞれのシンボルと前記入力信号 a_0 を乗算して長さ24であるシンボル列を排他的論理和器1040に出力し、前記乗算器1022は前記シンボル列 R_2 それぞれのシンボルと前記入力信号 a_1 を乗算して長さ24であるシンボル列を前記排他的論理和器1040に出力する。前記乗算器1024は前記シンボル列 R_3 それぞれのシンボルと前記入力信号 a_2 を乗算して長さ24であるシンボル列を前記排他的論理和器1040に出力し、前記乗算器1026は前記シンボル列 R_4 それぞれのシンボルと前記入力信号 a_3 を乗算して長さ24であるシンボル列を前記排他的論理和器1040に出力する。前記乗算器1028は前記シンボル列 R_5 それぞれのシンボルと前記入力信号 a_4 を乗算して長さ24であるシンボル列を前記排他的論理和器1040に出力し、前記乗算器1030は前記シンボル列 R_6 それぞれのシンボルと前記入力信号 a_5 を乗算して長さ24であるシンボル列を前記排他的論理和器1040に出力する。前記乗算器1032は前記シンボル列 R_7 それぞれのシンボルと前記入力信号 a_6 を乗算して長さ24であるシンボル列を前記排他的論理和器1040に出力する。すると、前記排他的論理和器1040は前記7個の長さ24であるシンボル列をシンボル単位に排他的論理和をとって長さ24である符号化シンボル列を出力する。

10

20

【0046】

前記(24、7)符号器のような符号器構造において、穿孔位置によって、性能だけではなく、符号器観点で複雑度を最小化させることができる。実際に、前記のような穿孔位置である{0、4、8、12、16、20、24、28}はそれぞれの穿孔位置が4の等間隔を有し、前記のような穿孔位置の規則性はハードウェア複雑度を減少させる利点を有する。しかし、穿孔位置を最前とすると、遅延のような方法を利用してハードウェア複雑度を減少させることができる。従って、性能を考慮して穿孔位置を最前とすると、前記2個のマスク関数は相異なる構成になる。

【0047】

前記図3を参照して上述したように相異なる穿孔位置を有する拡張された直交符号の構造を説明すると、 M_1 と M_2 が前記使用される2個のマスク関数である場合、上位32個の符号語は前記32個の長さ32である直交符号語(W)を使用する。次の32個の符号語は前記マスク関数 M_1 と前記32個の直交符号語(W)をそれぞれ排他的論理和をとった32個の符号語($M_1 + W$)を使用し、その次の32個の符号語は前記マスク関数 M_2 と前記32個の直交符号語(W)をそれぞれ排他的論理和をとった32個の符号語($M_2 + W$)を使用する。最後の32個の符号語には前記マスク関数 M_1 、 M_2 と前記32個の直交符号語(W)を排他的論理和をとった32個の符号語($M_1 + M_2 + W$)を使用する。従って、拡張された直交符号には総個数 $2^7 = 128$ 個の符号語を使用するようになる。この時、(24、7)符号を最適化するための前記2個のマスク関数は実験的に求めることができる。

30

【0048】

例えば、前記2個のマスク関数 M_1 、 M_2 は、次のようになる。

$M_1 = 00000000111010001101100011000000$

$M_2 = 00000000110000000111111000101000$

40

【0049】

一般的に、(24、7)線形符号を生成することができる多様な方法があるが、本発明の実施形態による拡張された一次リードミュラー符号を穿孔する方法を使用することにより、ハードウェア複雑度(Hardware Complexity)を最小化することができるだけでなく、最適の符号語を使用することができる。また、前記穿孔前の直交符号の長さを最小化させることにより、ハードウェア複雑度を最小化することができる。また、前記拡張された直交符号符号を穿孔することによりハードウェア複雑度を最小化することができるだけでは

50

なく、誤り訂正性能においても最適符号を生成することができる。本発明の実施形態では拡張された直交符号を有して誤り訂正符号を生成すると仮定する。

【0050】

上述したように、(24、7)符号語は長さ(32、7)拡張された直交符号発生器で出力された32個の符号化シンボルのうち、8シンボルを穿孔したものである。ここで、前記32個の拡張された直交符号シンボルのうちで、前記8シンボルを穿孔する穿孔位置を変更すると、符号語の最小距離($d_{\min}$: minimum distance)が相異なるようになる。そのため、前記(32、7)拡張された直交符号で優秀な誤り訂正性能を有する(24、7)符号器を生成するために、一番大きな最小距離を得ることができるようにする穿孔位置を求めるべきである。

10

【0051】

前記のような(24、7)線形符号として最適の符号を生成するために必要な8個の穿孔位置は実験的に求めることができる、その中、一番簡単な場合は{0、1、2、3、4、5、6、7}である。この場合、7個の情報ビットを伝送するために本発明の符号化／復号化方法を使用する移動通信システムの送信器と受信器は、予め8個の穿孔位置を約束しているべきである。これは通信規格で決定するのが一般的であり、送信側が予め穿孔位置を知らせることも可能である。

【0052】

図4は、本発明の他の実施形態による送信器に設けられた最適(24、7)符号器の構成を示している。

20

【0053】

前記図4を参照すると、7ビットの入力情報ビットa0、a1、a2、a3、a4、a5、a6は、(32、7)符号器400の拡張された直交符号器400に入力される。前記7ビットの入力情報ビットはデータ速度インジケータである。前記7ビットの入力情報ビットa0、a1、a2、a3、a4、a5、a6を受信した前記拡張された直交符号器400は、前記7ビットの入力情報ビットa0、a1、a2、a3、a4、a5、a6を符号化して長さ32の符号化シンボル(符号シンボル列)を出力する。前記拡張された直交符号器400から出力される前記32個の符号化シンボルは穿孔器410に入力される。前記穿孔器410は前記32個の符号化シンボルで前記8個の最適の穿孔位置である0、1、2、3、4、5、6、7番目のシンボルを穿孔して24個の符号化シンボルを出力する。

30

【0054】

一般的な符号理論では入力情報と符号化シンボル間の対応関係を示すために生成行列(Generating Matrix)を定義する。前記のようにシンボル穿孔が含まれる場合、最終的な(24、7)符号器に対する生成行列は、下記<数3>のようになる。

【0055】

【数3】

$$M = \begin{bmatrix} 0101010101010101010101 \\ 001100110011001100110011 \\ 000011110000111100001111 \\ 111111110000000011111111 \\ 000000001111111111111111 \\ 111010001101100011000000 \\ 110000000111111000101000 \end{bmatrix}$$

40

【0056】

前記<数3>の生成行列は、前記7ビットの入力信号のうち、一番目の入力信号が1で

50

あると、一番目行の24個のシンボルを選択し、0であると選択しない。前記7ビットの入力信号のうち、二番目の入力信号が1であると、二番目行の24個のシンボルを選択し、0であると選択しない。前記7ビットの入力信号のうち、三番目の入力信号が1であると、三番目行の24個のシンボルを選択し、0であると選択しない。前記7ビットの入力信号のうち、四番目の入力信号が1であると、四番目行の24個のシンボルを選択し、0であると選択しない。前記7ビットの入力信号のうち、五番目の入力信号が1であると、五番目行の24個のシンボルを選択し、0であると選択しない。前記7ビットの入力信号のうち、六番目の入力信号が1であると、六番目行の24個のシンボルを選択し、0であると選択しない。前記7ビットの入力信号のうち、七番目の入力信号が1であると、七番目行の24個のシンボルを選択し、0であると選択しない。この時、前記選択されたすべての列に対してシンボル単位に排他的論理和をとると、前記入力情報ビットに対応される符号化シンボルが出力される。

10

【0057】

図8は前記生成行列による(24、7)符号に対する符号器を示す。

【0058】

前記図8を参照すると、まず、0、または1の値を有する入力情報ビットa0、a1、a2、a3、a4、a5、a6が入力されると、a0は乗算器1020に、a1は乗算器1022に、a2は乗算器1024に、a3は乗算器1026に、a4乗算器1028に、a5は乗算器1030に、a6は乗算器1032にそれぞれ入力される。これと同時に、前記信号発生器1000はメモリに貯蔵されている前記生成行列を構成する7個のシンボル列を前記乗算器のうち、それぞれ対応する一つの乗算器に出力する。即ち、前記生成行列の一番目列に該当する長さ24であるシンボル列R1=0101 0101 0101 0101 0101 0101 0101をメモリに貯蔵し、乗算器1020に出力し、前記生成行列の二番目列に該当する長さ24であるシンボル列R2=0011 0011 0011 0011 0011 0011 0011をメモリに貯蔵し、乗算器1022に出力する。また、前記生成行列の三番目列に該当する長さ24であるシンボル列R3=0000 1111 0000 1111 0000 1111 0000 1111をメモリに貯蔵し、乗算器1024に出力し、前記生成行列の四番目列に該当する長さ24であるシンボル列R4=1111 1111 0000 0000 1111 1111 1111をメモリに貯蔵し、乗算器926に出力する。前記生成行列の五番目列に該当する長さ24であるシンボル列R5=0000 0000 1111 1111 1111 1111 1111をメモリに貯蔵し、乗算器1028に出力し、前記生成行列の六番目列に該当する長さ24であるシンボル列R6=1110 1000 1101 1000 1100 0000をメモリに貯蔵し、乗算器1030に出力する。最後に、前記生成行列の七番目列に該当する長さ24であるシンボル列R7=1100 0000 0111 1110 0010 1000をメモリに貯蔵し、乗算器1032に出力する。すると、前記乗算器1020は前記シンボル列R1それぞれのシンボルと前記入力信号a0を乗算して長さ24であるシンボル列を排他的論理和器1040に出力し、前記乗算器1022は前記シンボル列R2それぞれのシンボルと前記入力信号a1を乗算して長さ24であるシンボル列を前記排他的論理和器1040に出力する。前記乗算器1024は前記シンボル列R3それぞれのシンボルと前記入力信号a2を乗算して長さ24であるシンボル列を前記排他的論理和器1040に出力し、前記乗算器1026は前記シンボル列R4それぞれのシンボルと前記入力信号a3を乗算して長さ24であるシンボル列を前記排他的論理和器1040に出力する。前記乗算器1028は前記シンボル列R5それぞれのシンボルと前記入力信号a4を乗算して長さ24であるシンボル列を前記排他的論理和器1040に出力し、前記乗算器1030は前記シンボル列R6それぞれのシンボルと前記入力信号a5を乗算して長さ24であるシンボル列を前記排他的論理和器1040に出力する。前記乗算器1032は前記シンボル列R7それぞれのシンボルと前記入力信号a6を乗算して長さ24であるシンボル列を前記排他的論理和器1040に出力する。すると、前記排他的論理和器1040は前記7個の長さ24であるシンボル列をシンボル単位に排他的論理和をとって長さ24である符号化シンボル列を出力する。

20

30

40

50

【0059】

上述した(24、4)符号器と(24、7)符号器は、符号器構造観点で直交符号の構造から類推することができるとの共通点を有している。即ち、(24、4)符号器で使用された(15、4)シンプレックス符号は、(16、4)直交符号で0番目列を穿孔したものであり、(24、7)符号器で使用された(32、7)拡張された直交符号は、(32、5)符号器でマスク関数との追加的な2個の符号語の基底(basis)を使用して拡張した。従って、前記(24、4)符号器と(24、7)符号器は前記のような共通点を有しており、これを利用して後述される第1実施形態では前記相異なる大きさを有する(24、4)符号器と(24、7)符号器を一つの符号器に符号化することができる符号器を示す。

【0060】

10

第1実施形態(符号器)

図5は上述したように、相異なる長さを有する(24、4)符号器と(24、7)符号器をすべて遂行することができる符号器の構造を示す。即ち、前記図5の符号器は4ビット、または7ビットの入力情報ビットを長さ16、または32の相異なるウォルシュ符号、またはマスクに符号化して24個の符号化シンボルを有する符号化シンボル列に出力する。

【0061】

前記図5を参照すると、制御器510は前記入力情報ビットに4ビットが入力されるか、7ビットが入力されるかを検査して前記符号化のための制御を遂行する。即ち、前記制御器510は前記入力情報ビットに4ビットが入力されると、ウォルシュ符号発生器500とマスク発生器505が長さ16を有する相異なる5個のウォルシュ符号と2個のマスクを出力するように制御する。また、反復器550の反復回数を1回に決定し、穿孔器560には前記4ビットの入力情報ビットに対応した8個の穿孔位置を穿孔するように制御する。一方、前記制御部510は前記入力情報ビットに7ビットが入力されると、前記ウォルシュ符号発生器500と前記マスク発生器505が長さ32を有する相異なる5個のウォルシュ符号と2個のマスクを出力するように制御する。また、前記反復器550の反復回数を0回に決定し、前記穿孔器560には前記7ビットの入力情報ビットに対応した8個の穿孔位置を穿孔するように制御する。

20

【0062】

前記ウォルシュ符号発生器500は前記制御器510の制御下に長さ16または長さ32の相異なる複数のウォルシュ符号を選択的に出力する。例えば、4ビットの入力情報ビットが受信されると、前記制御器510の制御下に前記ウォルシュ符号発生器500は長さ16の相異なる5個のウォルシュ符号を出力する。しかし、7ビットの入力情報ビットが受信されると、前記制御器510の制御下に前記ウォルシュ符号発生器500は長さ32の相異なる5個のウォルシュ符号を出力する。一方、前記ウォルシュ符号発生器500から発生されるウォルシュ符号の個数は前記入力情報ビット(4ビットまたは7ビット)に対応して差別化することができる。即ち、4ビットの入力情報ビットが受信される場合には、長さ16を有する4個のウォルシュ符号を出力し、7ビットの入力情報ビットが受信される場合には、長さ32を有する5個のウォルシュ符号を出力する。

30

【0063】

前記マスク発生器505は前記制御器510の制御下に長さ16または長さ32のマスクを選択的に出力する。例えば、4ビットの入力情報ビットが受信されると、前記制御器510の制御下に前記マスク発生器505は長さ16の相異なる2個のマスクを出力する。しかし、7ビットの入力情報ビットが受信されると、前記制御器510の制御下に前記マスク発生器505は長さ32の相異なる2個のマスクを出力する。一方、前記マスク発生器505は前記制御器510による制御なし、2個の相異なるマスクを連続して出力するように具現されることができる。前記図5では前記制御器510により制御されないマスク発生器505の構成を示している。もし、前記制御器510の制御下に前記マスク発生器505が動作すると、前記制御器510からの制御信号が前記マスク発生器505に印加されるべきである。

40

【0064】

50

乗算器 520 乃至 532 は前記 4 ビットまたは前記 7 ビットの入力情報ビットと前記ウォルシュ符号発生器 500 と前記マスク発生器 505 からのウォルシュ符号及びマスクを一对一に乗算して、16 個または 32 個の符号化シンボルから構成された符号化シンボル列を出力する。もし、前記入力情報ビットとして 4 ビットが受信される場合、残りの入力情報ビット (a4、a5、a6) には“0”を提供することにより出力値に影響を与えないようにする。排他的論理和器 540 は前記乗算器 520 乃至 532 からの符号化シンボル列の排他的論理和をとって、16 個または 32 個の符号化シンボルから構成された一つの符号化シンボル列を出力する。前記反復器 550 は前記制御器 510 の制御下に前記排他的論理和器 540 からの前記符号化シンボル列を所定回数反復して、32 個の符号化シンボルから構成された符号化シンボル列を出力する。即ち、前記反復器 550 は 16 個の符号化シンボルから構成された符号化シンボル列が受信される場合、前記受信された符号化シンボル列を 1 回反復して 32 個の符号化シンボルから構成された符号化シンボル列を出力する。しかし、32 個の符号化シンボルから構成された符号化シンボル列が受信される場合には反復なしそのままに出力する。メモリ 570 は前記 4 ビットの入力情報ビットに対応した 8 個の穿孔位置と前記 7 ビットの入力情報ビットに対応した 8 個の穿孔位置を貯蔵する。前記 4 ビットの入力情報ビットに対応して前記メモリ 570 に貯蔵された 8 個の穿孔位置は、0、1、2、3、4、5、6、16 番目のシンボルであり、前記 7 ビットの入力情報ビットに対応して前記メモリ 570 に貯蔵された 8 個の穿孔位置は、0、4、8、12、16、20、24、28 番目のシンボル、または 0、1、2、3、4、5、6、7 番目のシンボルである。

10

20

【0065】

前記穿孔器 560 は前記反復器 550 から前記 32 個の符号化シンボルから構成された符号化シンボル列を受信し、前記制御器 510 の制御下に前記 32 個の符号化シンボルのうち前記メモリ 570 から読み出した 8 個の該当穿孔位置の符号化シンボルを穿孔して、24 個の符号化シンボルから構成された前記符号化シンボル列を出力する。即ち、前記入力情報ビットとして 4 ビットが入力された場合、前記穿孔器 560 は前記制御器 510 の制御下に前記反復器 550 から出力される 32 個の符号化シンボルのうち、0、1、2、3、4、5、6、16 番目のシンボルを穿孔して、24 個の符号化シンボルから構成された符号化シンボル列を出力する。しかし、前記入力情報ビットとして 7 ビットが入力された場合、前記穿孔器 560 は前記制御器 510 の制御下に前記反復器 550 から出力される 32 個の符号化シンボルのうち、0、4、8、12、16、20、24、28 番目のシンボル、または 0、1、2、3、4、5、6、7 番目のシンボルを穿孔して、24 個の符号化シンボルから構成された符号化シンボル列を出力する。

30

【0066】

上述した図 5 を参照して前記符号器を説明するために、(24、4)符号器に動作する場合と(24、7)符号器に動作する場合を説明する。この時、符号器に入力される入力情報ビットはデータ速度インジケータと仮定する。

【0067】

一番目に、前記(24、7)符号器に動作する場合の符号器の動作を説明すると、7 ビットのデータ速度インジケータ a0、a1、a2、a3、a4、a5、a6 が前記符号器に入力される。一方、前記 7 ビットのデータ速度インジケータが前記符号器に入力されるとのビット情報が制御器 510 に入力される。前記制御器 510 はウォルシュ符号発生器 500 に長さ 32 を有するウォルシュ符号を生成するようにする制御信号を出力する。すると、前記ウォルシュ符号発生器 500 は長さ 32 であるウォルシュ符号 W1、W2、W4、W8、W16 を出力する。前記ウォルシュ符号発生器 500 から発生された前記 W1、W2、W4、W8、W16 はそれぞれに対応する乗算器に提供される。例えば、前記 W1 は乗算器 520 に提供され、前記 W2 は乗算器 522 に提供される。前記 W4 は乗算器 524 に提供され、前記 W8 は乗算器 526 に提供される。最後に前記 W16 は乗算器 528 に提供される。前記ウォルシュ符号発生器 500 から発生される長さ 32 のウォルシュ符号は下記<表 1>のようになる。

40

50

【0068】

【表1】

ウォルシュ番号	ウォルシュ符号
W1	0101 0101 0101 0101 0101 0101 0101 0101
W2	0011 0011 0011 0011 0011 0011 0011 0011
W4	0000 1111 0000 1111 0000 1111 0000 1111
W8	0000 0000 1111 1111 0000 0000 1111 1111
W16	0000 0000 0000 0000 1111 1111 1111 1111

10

【0069】

これと同時に、マスク発生器505はマスクM1=0111 0111 0010 0100 0110 0000 0000 0000を発生させ乗算器530に出力し、マスクM2=0010 0110 0101 0100 0101 0100 0100 0000を発生させ乗算器532に出力する。

【0070】

一方、前記7個のデータ速度インジケータは前記乗算器のうち対応するそれぞれの前記乗算器に提供される。前記7個のデータ速度インジケータと前記乗算器の対応関係は下記<表2>のようになる。

20

【0071】

【表2】

入力情報ビット	乗算器
a0	乗算器 520
a1	乗算器 522
a2	乗算器 524
a3	乗算器 526
a4	乗算器 528
a5	乗算器 530
a6	乗算器 532

30

【0072】

従って、前記乗算器それぞれに提供される入力情報ビットとウォルシュ符号及びマスクは下記<表3>のように示すことができる。

【0073】

【表3】

乗算器	ウォルシュ符号/マスク	入力情報ビット
乗算器 520	W1 = 0101 0101 0101 0101 0101 0101 0101 0101	a0
乗算器 522	W2 = 0011 0011 0011 0011 0011 0011 0011 0011	a1
乗算器 524	W4 = 0000 1111 0000 1111 0000 1111 0000 1111	a2
乗算器 526	W8 = 0000 0000 1111 1111 0000 0000 1111 1111	a3
乗算器 528	W16 = 0000 0000 0000 0000 1111 1111 1111 1111	a4
乗算器 530	M1 = 0111 0111 0010 0100 0110 0000 0000 0000	a5
乗算器 532	M2 = 0010 0110 0101 0100 0101 0100 0100 0000	a6

40

50

【0074】

前記乗算器は前記<表3>で示しているように自分に提供されるデータ速度インジケータをウォルシュ符号／マスクを構成するそれぞれのシンボルと乗算して加算器540に出力する。

【0075】

即ち、前記乗算器520はa0をW1それぞれのシンボルごとに乗算して加算器540に出力し、前記乗算器522はa1をW2それぞれのシンボルごとに乗算して前記加算器540に出力し、前記乗算器524はa2をW4それぞれのシンボルごとに乗算して前記加算器540に出力する。一方、前記乗算器526はa3をW8それぞれのシンボルごとに乗算して前記加算器540に出力し、前記乗算器528はa4をW16それぞれのシンボルごとに乗算して前記加算器540に出力する。また、前記乗算器530はa5をM1それぞれのシンボルごとに乗算して前記加算器540に出力し、前記乗算器532はa6をM2それぞれのシンボルごとに乗算して前記加算器540に出力する。前記加算器540は排他的論理和器を使用する。

10

【0076】

前記排他的論理和器540は前記乗算器520、522、524、526、528、530、532から出力された長さ32であるすべてのシンボル列をシンボル単位に排他的論理和をとって反復器550に出力する。

【0077】

前記排他的論理和器540から出力される長さ32の符号化シンボル列(Ws)は下記<式4>のように表現されることができる。

20

【0078】

<式4>

$$W_s = (W_1 \times a_0) + (W_2 \times a_1) + (W_4 \times a_2) + (W_8 \times a_3) + (W_{16} \times a_4) + (M_1 \times a_5) + (M_2 \times a_6)$$

【0079】

この時、前記制御器510は前記反復器550が前記排他的論理和器540の出力を反復する回数を制御する制御信号を出力する。(24、7)符号器では前記排他的論理和器540から出力されるシンボル列が長さ32を有することによって、前記反復器550は前記長さ32を有するシンボル列に対して反復を遂行してはいけない。このため前記制御器510は前記反復器550に入力信号をそのままに出力するように指示する制御信号を出力する。これにตอบสนองして前記反復器550は前記排他的論理和器540から出力された長さ32であるシンボル列をそのままに穿孔器560に出力する。

30

【0080】

この時点で前記制御器510は前記穿孔器560に(24、7)符号に対する穿孔位置を穿孔するように指示する制御信号を出力する。この時、制御信号として前記制御器510は前記データ速度インジケータの長さ情報(7ビット)のみを前記穿孔器560に提供することができる。すると、前記穿孔器560は穿孔位置が貯蔵されているメモリ570から前記データ速度インジケータが7ビットである場合に対応する穿孔位置を受信して、前記反復器550から提供される長さ32の符号化シンボルのうち、前記穿孔位置に該当する8個の該当符号化シンボルを穿孔する。即ち、前記穿孔器560は前記反復器550から提供される長さ32の符号化シンボルのうち、前記メモリ570からの穿孔位置に該当する8シンボルを穿孔する。例えば、前記メモリ570からの穿孔位置が0、4、8、12、16、20、24、28である場合、前記穿孔器560は前記長さ32の符号化シンボルのうち、0、4、8、12、16、20、24、28番目の符号化シンボルを穿孔して24個の符号化シンボルを出力する。

40

【0081】

前記(24、7)符号器の動作は、穿孔位置に0、4、8、12、16、20、24、28を有する場合に対する構成を示した。上述したようにハードウェア複雑度を低減するために、穿孔位置に0、1、2、3、4、5、6、7を有する(24、7)符号器への動作を

50

説明すると、7ビットのデータ速度インジケータa0、a1、a2、a3、a4、a5、a6が前記符号器に入力されると、前記7ビットのデータ速度インジケータが前記符号器に入力されるとのビット情報が制御器510に入力される。すると、前記制御器510はウォルシュ符号発生器500に長さ32を有するウォルシュ符号を生成するようにする制御信号を出力する。前記ウォルシュ符号発生器500は長さ32であるウォルシュ符号W1、W2、W4、W8、W16を出力する。前記ウォルシュ符号発生器500から発生された前記W1、W2、W4、W8、W16はそれぞれに対応する乗算器に提供される。例えば、前記W1は乗算器520に提供され、前記W2は乗算器522に提供される。前記W4は乗算器524に提供され、前記W8は乗算器526に提供される。最後に前記W16は乗算器528に提供される。前記ウォルシュ符号発生器500から発生される長さ32のウォルシュ符号は下記<表4>のようになる。

【0082】

【表4】

ウォルシュ番号	ウォルシュ符号
W1	0101 0101 0101 0101 0101 0101 0101 0101
W2	0011 0011 0011 0011 0011 0011 0011 0011
W4	0000 1111 0000 1111 0000 1111 0000 1111
W8	0000 0000 1111 1111 0000 0000 1111 1111
W16	0000 0000 0000 0000 1111 1111 1111 1111

【0083】

これと同時に、マスク発生器505はマスクM1=0000 0000 1110 1000 1101 1000 1100 0000を発生させ乗算器530に出力し、マスクM2=0000 0000 1100 0000 0111 1110 0010 1000を発生させ乗算器532に出力する。

【0084】

一方、前記データ速度インジケータの7個の入力情報ビットは前記乗算器のうち対応するそれぞれの乗算器に提供される。前記7個の入力情報ビットと前記乗算器間の対応関係は下記<表5>のようになる。

【0085】

【表5】

入力情報ビット	乗算器
a0	乗算器 520
a1	乗算器 522
a2	乗算器 524
a3	乗算器 526
a4	乗算器 528
a5	乗算器 530
a6	乗算器 532

【0086】

従って、前記乗算器それぞれに提供される入力情報ビットとウォルシュ符号及びマスクは下記<表6>のようになる。

【0087】

10

20

30

40

50

【表 6】

乗算器	ウォルシュ符号/マスク	入力情報ビット
乗算器 520	W1 = 0101 0101 0101 0101 0101 0101 0101 0101	a0
乗算器 522	W2 = 0011 0011 0011 0011 0011 0011 0011 0011	a1
乗算器 524	W4 = 0000 1111 0000 1111 0000 1111 0000 1111	a2
乗算器 526	W8 = 0000 0000 1111 1111 0000 0000 1111 1111	a3
乗算器 528	W16 = 0000 0000 0000 0000 1111 1111 1111 1111	a4
乗算器 530	M1 = 0000 0000 1110 1000 1101 1000 1100 0000	a5
乗算器 532	M2 = 0000 0000 1100 0000 0111 1110 0010 1000	a6

10

【0088】

前記乗算器は前記<表 6>で示しているように自分に提供されるデータ速度インジケータをウォルシュ符号/マスクを構成するそれぞれのシンボルと乗算して加算器 540 に出力する。

【0089】

即ち、前記乗算器 520 は a0 を W1 それぞれのシンボルごとに乗算して加算器 540 に出力し、前記乗算器 522 は a1 を W2 それぞれのシンボルごとに乗算して前記加算器 540 に出力し、前記乗算器 524 は a2 を W4 それぞれのシンボルごとに乗算して前記加算器 540 に出力する。一方、前記乗算器 526 は a3 を W8 それぞれのシンボルごとに乗算して前記加算器 540 に出力し、前記乗算器 528 は a4 を W16 それぞれのシンボルごとに乗算して前記加算器 540 に出力する。また、前記乗算器 530 は a5 を M1 それぞれのシンボルごとに乗算して前記加算器 540 に出力し、前記乗算器 532 は a6 を M2 それぞれのシンボルごとに乗算して前記加算器 540 に出力する。前記加算器 540 は排他的論理和器を使用する。

20

【0090】

すると、前記排他的論理和器 540 は前記乗算器 520、522、524、526、528、530、532 から出力された長さ 32 であるすべてのシンボル列をシンボル単位に排他的論理和をとって反復器 550 に出力する。

30

【0091】

前記排他的論理和器 540 から出力される長さ 32 の符号化シンボル列 (Ws) は下記<式 5>のように表現されることができる。

【0092】

<式 5>

$$Ws = (W1 \times a0) + (W2 \times a1) + (W4 \times a2) + (W8 \times a3) + (W16 \times a4) + (M1 \times a5) + (M2 \times a6)$$

【0093】

この時、前記制御器 510 は前記反復器 550 が前記排他的論理和器 540 の出力を反復する回数を制御する制御信号を出力する。(24、7)符号器では前記排他的論理和器 540 から出力されるシンボル列が長さ 32 を有することにより、前記反復器 550 の反復回数は 0 回になるものである。従って、前記制御器 510 は前記反復器 550 に入力信号をそのままに出力するように指示する制御信号を出力する。これに応答して前記反復器 550 は前記排他的論理和器 540 から出力された長さ 32 であるシンボルをそのままに穿孔器 560 に出力する。

40

【0094】

この時点で前記制御器 510 は前記穿孔器 560 に(24、7)符号に対する穿孔位置を穿孔するように指示する制御信号を出力する。すると、前記穿孔器 560 は穿孔位置が貯蔵されているメモリ 570 から穿孔位置を受信して前記反復器 550 から提供される長さ 32 の符号化シンボルのうち、8 個の該当符号化シンボルを穿孔する。即ち、前記穿孔器

50

560は前記反復器550から提供される長さ32の符号化シンボルのうち、前記メモリ570からの穿孔位置に該当する8シンボルを穿孔する。例えば、前記メモリ570からの穿孔位置が0、1、2、3、4、5、6、7である場合、前記穿孔器560は前記長さ32の符号化シンボルのうち、0、1、2、3、4、5、6、7番目の符号化シンボルを穿孔して、24個の符号化シンボルを出力する。

【0095】

二番目に、前記(24、4)符号器として動作する場合を説明すると、4ビットのデータ速度インジケータa0、a1、a2、a3が前記符号器に入力され、残りのデータ速度インジケータa4、a5、a6は初期化され入力される。前記初期化により前記a4、a5、a6には0が入力される。一方、前記制御器510には4ビットのデータ速度インジケータが前記符号器に入力されるとのビット情報が入力される。すると、前記制御器510はウォルシュ符号発生器500に長さ16を有するウォルシュ符号を生成するように指示する制御信号を出力する。すると前記ウォルシュ符号発生器500は長さ16であるウォルシュ符号W1、W2、W4、W8を出力する。前記ウォルシュ符号発生器500から発生された前記W1、W2、W4、W8はそれぞれに対応する乗算器に提供される。例えば、前記W1は乗算器520に提供され、前記W2は乗算器522に提供される。前記W4は乗算器524に提供され、前記W8は乗算器526に提供される。前記ウォルシュ符号発生器500から発生される長さ16のウォルシュ符号は下記<表7>のようになる。

【0096】

【表7】

ウォルシュ番号	ウォルシュ符号
W1	0101 0101 0101 0101
W2	0011 0011 0011 0011
W4	0000 1111 0000 1111
W8	0000 0000 1111 1111

【0097】

一方、前記データ速度インジケータの4個の入力情報ビットは前記乗算器のうち対応するそれぞれの乗算器に提供される。前記4個の入力情報ビットと前記乗算器の対応関係は下記<表8>で示しているようになる。

【0098】

【表8】

入力情報ビット	乗算器
a0	乗算器 520
a1	乗算器 522
a2	乗算器 524
a3	乗算器 526

【0099】

従って、前記乗算器それぞれに提供される入力情報ビットとウォルシュ符号は下記<表9>のようになる。

【0100】

10

20

30

40

【表 9】

乗算器	ウォルシュ符号	入力情報ビット
乗算器 520	W1 = 0101 0101 0101 0101	a0
乗算器 522	W2 = 0011 0011 0011 0011	a1
乗算器 524	W4 = 0000 1111 0000 1111	a2
乗算器 526	W8 = 0000 0000 1111 1111	a3

【0101】

10

前記乗算器は前記<表 9>で示しているように自分に提供されるデータ速度インジケータを、ウォルシュ符号を構成するそれぞれのシンボルと乗算して加算器 540 に出力する。

【0102】

即ち、前記乗算器 520 は a0 を W1 それぞれのシンボルごとに乗算して加算器 540 に出力し、前記乗算器 522 は a1 を W2 それぞれのシンボルごとに乗算して前記加算器 540 に出力する。一方、前記乗算器 524 は a2 を W4 それぞれのシンボルごとに乗算して前記加算器 540 に出力し、前記乗算器 526 は a3 を W8 それぞれのシンボルごとに乗算して前記加算器 540 に出力する。前記加算器 540 は排他的論理和器を使用する。

20

【0103】

一方、乗算器 528、530、532 には前記初期化により 0 の値を有する a4、a5、a6 がそれぞれ印加されることにより、前記ウォルシュ符号発生器 500 からの W16 とマスク発生器 505 からの M1、M2 に関わらず前記乗算器 528、530、532 の出力は前記加算器 540 の出力に影響を与えない。

【0104】

即ち、前記ウォルシュ符号生成器 500 から乗算器 528 に入力される W16 シンボル列の値に関わらず前記乗算器 528 からは 0 であるシンボル列が出力される。また、前記マスク生成器 505 から乗算器 530、532 に入力されるシンボル列の値に関わらず前記乗算器 530、532 からは 0 であるシンボル列が出力される。従って、前記乗算器 528、530、532 の出力は前記排他的論理和器 540 に入力されても影響を与えない。前記のように a4、a5、a6 を 0 に初期化することは、前記乗算器 528、530、532 の出力を遮断するスイッチ動作と類似である。

30

【0105】

すると、前記排他的論理和器 540 は前記乗算器 520、522、524、526、528、530、532 から出力された長さ 16 であるすべてのシンボル列をシンボル単位に排他的論理和をとって反復器 550 に出力する。

【0106】

前記排他的論理和器 540 から出力される長さ 16 の符号化シンボル列 (Ws) は下記<式 6>のように表現することができる。

40

【0107】

<式 6>

$$W_s = (W_1 \times a_0) + (W_2 \times a_1) + (W_4 \times a_2) + (W_8 \times a_3)$$

【0108】

この時、前記制御器 510 は前記反復器 550 が前記排他的論理和器 540 の出力を反復する回数を制御する制御信号を出力する。(24、4)符号器では前記排他的論理和器 540 から出力されるシンボル列が長さ 16 を有することにより、前記反復器 550 の反復回数は 1 回になる。従って、前記制御器 510 は前記反復器 550 に入力信号を 1 回反復するように指示する制御信号を出力する。これにตอบสนองして前記反復器 550 は前記排他的論理和器 540 から出力された長さ 16 であるシンボルを 1 回反復して長さ 32 のシンボ

50

ル列を穿孔器 5 6 0 に出力する。

【0 1 0 9】

この時点で前記制御器 5 1 0 は前記穿孔器 5 6 0 に (2 4、4) 符号に対する穿孔位置を穿孔するように指示する制御信号を出力する。すると、前記穿孔器 5 6 0 は穿孔位置が貯蔵されているメモリ 5 7 0 から穿孔位置を受信して前記反復器 5 5 0 から提供される長さ 3 2 の符号化シンボルのうち、8 個の該当符号化シンボルを穿孔する。即ち、前記穿孔器 5 6 0 は前記反復器 5 5 0 から提供される長さ 3 2 の符号化シンボルのうち、前記メモリ 5 7 0 からの穿孔位置に該当する 8 シンボルを穿孔する。例えば、前記メモリ 5 7 0 からの穿孔位置が 0、1、2、3、4、5、6、1 6 である場合、前記穿孔器 5 6 0 は前記長さ 3 2 の符号化シンボルのうち、0、1、2、3、4、5、6、1 6 番目の符号化シンボルを穿孔して、2 4 個の符号化シンボルを出力する。

10

【0 1 1 0】

第 2 実施形態 (符号器)

図 9 は本発明の実施形態による相異なる長さを有する (2 4、4) 符号器と (2 4、7) 符号器をすべて遂行することができる符号器の構造を示す。即ち、前記図 9 では 4 ビット、または 7 ビットの入力情報ビットそれぞれを長さ 3 2 の相異なるウォルシュ符号、またはマスクに符号化して 2 4 個の符号化シンボルから構成された符号化シンボル列に出力する移動通信システムでの符号器の構成を示している。この時、前記第 1 実施形態での符号器と異なり、前記第 2 実施形態による符号器にはシンボル反復器が含まれない。

【0 1 1 1】

20

前記図 9 を参照すると、制御器 1 1 1 0 は前記入力情報ビットに 4 ビットが入力されるか、7 ビットが入力されるかを検査して前記符号化のための制御を遂行する。即ち、制御器 1 1 1 0 は穿孔器 1 1 6 0 に前記入力情報ビットに対応した 8 個の穿孔位置を穿孔するようにする制御信号を提供する。前記ウォルシュ符号発生器 1 1 0 0 は長さ 3 2 のウォルシュ符号を出力する。例えば、前記入力情報ビットが受信されると、前記ウォルシュ符号発生器 1 1 0 0 は長さ 3 2 の相異なる 5 個のウォルシュ符号を出力する。マスク発生器 1 1 0 5 は長さ 3 2 のマスクを出力する。例えば、前記入力情報ビットが受信されると、前記マスク発生器 1 1 0 5 は長さ 3 2 の相異なる 2 個のマスクを出力する。

【0 1 1 2】

乗算器 1 1 2 0 乃至 1 1 3 2 は前記 4 ビットまたは 7 ビットの入力情報ビットと前記ウォルシュ符号発生器 1 1 0 0 と前記マスク発生器 1 1 0 5 からのウォルシュ符号及びマスクを一对一に乗算して、3 2 個の符号化シンボルから構成された符号化シンボル列を出力する。排他的論理和器 1 1 4 0 は前記乗算器 1 1 2 0 乃至 1 1 3 2 からの符号化シンボル列の排他的論理和をとって、3 2 個の符号化シンボルで構成される一つの符号化シンボル列を出力する。メモリ 1 1 7 0 は前記 4 ビットの入力情報ビットに対応した 8 個の穿孔位置と前記 7 ビットの入力情報ビットに対応した 8 個の穿孔位置を貯蔵する。前記 4 ビットの入力情報ビットに対応して前記メモリ 1 1 7 0 に貯蔵された 8 個の穿孔位置は、0、1、2、3、4、5、6、1 6 番目のシンボルであり、前記 7 ビットの入力情報ビットに対応して前記メモリ 1 1 7 0 に貯蔵された 8 個の穿孔位置は、0、4、8、1 2、1 6、2 0、2 4、2 8 番目のシンボル、または 0、1、2、3、4、5、6、7 番目のシンボルである。

30

40

【0 1 1 3】

前記穿孔器 1 1 6 0 は前記排他的論理和器 1 1 4 0 から前記 3 2 個の符号化シンボルから構成された符号化シンボル列を受信し、前記制御器 1 1 1 0 の制御下に前記 3 2 個の符号化シンボルのうち、前記メモリ 1 1 7 0 から読み出した 8 個の該当穿孔位置の符号化シンボルを穿孔して、前記 2 4 個の符号化シンボルから構成された前記符号化シンボル列を出力する。即ち、前記入力情報ビットとして 4 ビットが入力される場合、前記穿孔器 1 1 6 0 は前記制御器 1 1 1 0 の制御下に前記排他的論理和器 1 1 4 0 から出力される 3 2 個の符号化シンボルのうち、0、1、2、3、4、5、6、1 6 番目のシンボルを穿孔して、2 4 個の符号化シンボルから構成された符号化シンボル列を出力する。しかし、前記入

50

力情報ビットとして7ビットが入力される場合、前記穿孔器1160は前記制御器1110の制御下に前記排他的論理和器1140から出力される32個の符号化シンボルのうち、0、4、8、12、16、20、24、28番目のシンボル、または0、1、2、3、4、5、6、7番目のシンボルを穿孔して、24個の符号化シンボルから構成された符号化シンボル列を出力する。

【0114】

前記図9を参照して前記符号器を説明するために、(24、4)符号器に動作する場合と(24、7)符号器に動作する場合を分けて説明する。この時、符号器に入力される入力情報ビットはデータ速度インジケータと仮定する。

【0115】

一番目に、前記(24、7)符号器への動作を説明すると、7ビットのデータ速度インジケータa0、a1、a2、a3、a4、a5、a6が前記符号器に入力されると、前記7ビットのデータ速度インジケータが前記符号器に入力されるとのビット情報が制御器1110に入力される。すると、前記ウォルシュ符号発生器1100は長さ32であるウォルシュ符号W1、W2、W4、W8、W16を出力する。この時、前記ウォルシュ符号発生器1100は前記制御器1110の制御下に動作することもできるが、別の制御なし動作することもできる。前記図9では別の制御なし動作する構成を示している。前記ウォルシュ符号発生器1100から発生された前記W1、W2、W4、W8、W16はそれぞれに対応する乗算器に提供される。例えば、前記W1は乗算器1120に提供され、前記W2は乗算器1122に提供される。前記W4は乗算器1124に提供され、前記W8は乗算器1126に提供される。最後に前記W16は乗算器1128に提供される。前記ウォルシュ符号発生器1100から発生される長さ32のウォルシュ符号は下記<表10>のようになる。

【0116】

【表10】

ウォルシュ番号	ウォルシュ符号
W1	0101 0101 0101 0101 0101 0101 0101 0101
W2	0011 0011 0011 0011 0011 0011 0011 0011
W4	0000 1111 0000 1111 0000 1111 0000 1111
W8	0000 0000 1111 1111 0000 0000 1111 1111
W16	0000 0000 0000 0000 1111 1111 1111 1111

【0117】

これと同時に、マスク発生器1105はマスクM1=0111 0111 0010 0100 0110 0000 0000 0000を発生させ乗算器1130に出力し、マスクM2=0010 0110 0101 0100 0101 0100 0100 0000を発生させ乗算器1132に出力する。

【0118】

一方、前記7個のデータ速度インジケータは前記乗算器のうち対応するそれぞれの乗算器に提供される。前記7個のデータ速度インジケータと前記乗算器の対応関係は下記<表11>のようになる。

【0119】

【表 1 1】

入力情報ビット	乗算器
a0	乗算器 1120
a1	乗算器 1122
a2	乗算器 1124
a3	乗算器 1126
a4	乗算器 1128
a5	乗算器 1130
a6	乗算器 1132

10

【0 1 2 0】

従って、前記乗算器それぞれに提供されるデータ速度インジケータとウォルシュ符号及びマスクは下記＜表 1 2＞のように示すことができる。

【0 1 2 1】

【表 1 2】

乗算器	ウォルシュ符号/マスク	入力情報ビット
乗算器 1120	W1 = 0101 0101 0101 0101 0101 0101 0101 0101	a0
乗算器 1122	W2 = 0011 0011 0011 0011 0011 0011 0011 0011	a1
乗算器 1124	W4 = 0000 1111 0000 1111 0000 1111 0000 1111	a2
乗算器 1126	W8 = 0000 0000 1111 1111 0000 0000 1111 1111	a3
乗算器 1128	W16 = 0000 0000 0000 0000 1111 1111 1111 1111	a4
乗算器 1130	M1 = 0111 0111 0010 0100 0110 0000 0000 0000	a5
乗算器 1132	M2 = 0010 0110 0101 0100 0101 0100 0100 0000	a6

20

【0 1 2 2】

前記乗算器は前記＜表 1 2＞で示しているように自分に提供されるデータ速度インジケータをウォルシュ符号／マスクを構成するそれぞれのシンボルと乗算して加算器 1 1 4 0 に出力する。

【0 1 2 3】

即ち、前記乗算器 1 1 2 0 は a 0 を W 1 それぞれのシンボルごとに乗算して加算器 1 1 4 0 に出力し、前記乗算器 1 1 2 2 は a 1 を W 2 それぞれのシンボルごとに乗算して前記加算器 1 1 4 0 に出力し、前記乗算器 1 1 2 4 は a 2 を W 4 それぞれのシンボルごとに乗算して前記加算器 1 1 4 0 に出力する。一方、前記乗算器 1 1 2 6 は a 3 を W 8 それぞれのシンボルごとに乗算して前記加算器 1 1 4 0 に出力し、前記乗算器 1 1 2 8 は a 4 を W 1 6 それぞれのシンボルごとに乗算して前記加算器 1 1 4 0 に出力する。また、前記乗算器 1 1 3 0 は a 5 を M 1 それぞれのシンボルごとに乗算して前記加算器 1 1 4 0 に出力し、前記乗算器 1 1 3 2 は a 6 を M 2 それぞれのシンボルごとに乗算して前記加算器 1 1 4 0 に出力する。前記加算器 1 1 4 0 は排他的論理和器を使用する。

40

【0 1 2 4】

すると、前記排他的論理和器 1 1 4 0 は前記乗算器 1 1 2 0、1 1 2 2、1 1 2 4、1 1 2 6、1 1 2 8、1 1 3 0、1 1 3 2 から出力された長さ 3 2 であるすべてのシンボル列をシンボル単位に排他的論理和をとって穿孔器 1 1 6 0 に出力する。

【0 1 2 5】

前記排他的論理和器 1 1 4 0 から出力される長さ 3 2 の符号化シンボル列 (W s) は下記

50

<式 7>のように表現されることができる。

【0126】

<式 7>

$$Ws = (W1 \times a0) + (W2 \times a1) + (W4 \times a2) + (W8 \times a3) + (W16 \times a4) + (M1 \times a5) + (M2 \times a6)$$

【0127】

この時、前記制御器1110は穿孔器1160に(24、7)符号に対する穿孔位置を穿孔するように指示する制御信号を出力する。すると、前記穿孔器1160は穿孔位置が貯蔵されているメモリ1170から穿孔位置を受信して前記反復器1150から提供される長さ32の符号化シンボルのうち、8個の該当符号化シンボルを穿孔する。即ち、前記穿孔器1160は前記排他的論理和器1140から提供される長さ32の符号化シンボルのうち、前記メモリ1170からの穿孔位置に該当する8シンボルを穿孔する。例えば、前記メモリ1170からの穿孔位置が0、4、8、12、16、20、24、28である場合、前記穿孔器1160は前記長さ32の符号化シンボルのうち、0、4、12、16、20、24、28番目の符号化シンボルを穿孔して、24個の符号化シンボルを出力する。

【0128】

前記(24、7)符号器の動作は、穿孔位置に0、4、8、12、16、20、24、28を有する場合に対する構成を示した。上述したようにハードウェア複雑度を低減するための穿孔位置に0、1、2、3、4、5、6、7を有する(24、7)符号器への動作を説明すると、7ビットのデータ速度インジケータa0、a1、a2、a3、a4、a5、a6が前記符号器に入力されると、前記7ビットのデータ速度インジケータが前記符号器に入力されるとのビット情報が制御器1110に入力される。すると前記ウォルシュ符号発生器1100は長さ32であるウォルシュ符号W1、W2、W4、W8、W16を出力する。前記ウォルシュ符号発生器1100から発生された前記W1、W2、W4、W8、W16はそれぞれに対応する乗算器に提供される。例えば、前記W1は乗算器1120に提供され、前記W2は乗算器1122に提供される。前記W4は乗算器1124に提供され、前記W8は乗算器1126に提供される。最後に前記W16は乗算器1128に提供される。前記ウォルシュ符号発生器1100から発生される長さ32のウォルシュ符号は下記<表13>のようになる。

【0129】

【表13】

ウォルシュ番号	ウォルシュ符号
W1	0101 0101 0101 0101 0101 0101 0101 0101
W2	0011 0011 0011 0011 0011 0011 0011 0011
W4	0000 1111 0000 1111 0000 1111 0000 1111
W8	0000 0000 1111 1111 0000 0000 1111 1111
W16	0000 0000 0000 0000 1111 1111 1111 1111

【0130】

これと同時に、マスク発生器1105はマスクM1=0000 0000 1110 1000 1101 1000 1100 0000を発生させ乗算器1130に出力し、マスクM2=0000 0000 1100 0000 0111 1110 0010 1000を発生させ乗算器1132に出力する。

【0131】

一方、前記7個のデータ速度インジケータは前記乗算器のうち対応するそれぞれの前記乗算器に提供される。前記7個のデータ速度インジケータと前記乗算器の対応関係は下記

<表 1 4>のようになる。

【0 1 3 2】

【表 1 4】

入力情報ビット	乗算器
a0	乗算器 1120
a1	乗算器 1122
a2	乗算器 1124
a3	乗算器 1126
a4	乗算器 1128
a5	乗算器 1130
a6	乗算器 1132

10

【0 1 3 3】

従って、前記乗算器それぞれに提供されるデータ速度インジケータとウォルシュ符号及びマスクは下記<表 1 5>のように示すことができる。

【0 1 3 4】

【表 1 5】

乗算器	ウォルシュ符号/マスク	入力情報ビット
乗算器 1120	W1 = 0101 0101 0101 0101 0101 0101 0101 0101	a0
乗算器 1122	W2 = 0011 0011 0011 0011 0011 0011 0011 0011	a1
乗算器 1124	W4 = 0000 1111 0000 1111 0000 1111 0000 1111	a2
乗算器 1126	W8 = 0000 0000 1111 1111 0000 0000 1111 1111	a3
乗算器 1128	W16 = 0000 0000 0000 0000 1111 1111 1111 1111	a4
乗算器 1130	M1 = 0000 0000 1110 1000 1101 1000 1100 0000	a5
乗算器 1132	M2 = 0000 0000 1100 0000 0111 1110 0010 1000	a6

20

30

【0 1 3 5】

前記乗算器は前記<表 1 5>で示しているように自分に提供されるデータ速度インジケータをウォルシュ符号／マスクを構成するそれぞれのシンボルと乗算して加算器 1 1 4 0 に出力する。

【0 1 3 6】

即ち、前記乗算器 1 1 2 0 は a 0 を W 1 それぞれのシンボルごとに乗算して加算器 1 1 4 0 に出力し、前記乗算器 1 1 2 2 は a 1 を W 2 それぞれのシンボルごとに乗算して前記加算器 1 1 4 0 に出力し、前記乗算器 1 1 2 4 は a 2 を W 4 それぞれのシンボルごとに乗算して前記加算器 1 1 4 0 に出力する。一方、前記乗算器 1 1 2 6 は a 3 を W 8 それぞれのシンボルごとに乗算して前記加算器 1 1 4 0 に出力し、前記乗算器 1 1 2 8 は a 4 を W 1 6 それぞれのシンボルごとに乗算して前記加算器 1 1 4 0 に出力する。また、前記乗算器 1 1 3 0 は a 5 を M 1 それぞれのシンボルごとに乗算して前記加算器 1 1 4 0 に出力し、前記乗算器 1 1 3 2 は a 6 を M 2 それぞれのシンボルごとに乗算して前記加算器 1 1 4 0 に出力する。前記加算器 1 1 4 0 は排他的論理和器を使用する。

40

【0 1 3 7】

すると、前記排他的論理和器 1 1 4 0 は前記乗算器 1 1 2 0、1 1 2 2、1 1 2 4、1 1 2 6、1 1 2 8、1 1 3 0、1 1 3 2 から出力された長さ 3 2 であるすべてのシンボル列をシンボル単位に排他的論理和をとって穿孔器 1 1 6 0 に出力する。

50

【0138】

前記排他的論理和器1140から出力される長さ32の符号化シンボル列(Ws)は下記<式8>のように表現されることができる。

【0139】

<式8>

$$Ws = (W1 \times a0) + (W2 \times a1) + (W4 \times a2) + (W8 \times a3) + (W16 \times a4) + (M1 \times a5) + (M2 \times a6)$$

【0140】

この時、前記制御器1110は穿孔器1160に(24、7)符号に対する穿孔位置を穿孔するように指示する制御信号を出力する。すると、前記穿孔器1160は穿孔位置が貯蔵されているメモリ1170から穿孔位置を受信して前記反復器1150から提供される長さ32の符号化シンボルのうち、8個の該当符号化シンボルを穿孔する。即ち、前記穿孔器1160は前記排他的論理和器1140から提供される長さ32の符号化シンボルのうち、前記メモリ1170からの穿孔位置に該当する8シンボルを穿孔する。例えば、前記メモリ1170からの穿孔位置が0、1、2、3、4、5、6、7である場合、前記穿孔器1160は前記長さ32の符号化シンボルのうち、0、1、2、3、4、5、6、7番目の符号化シンボルを穿孔して、24個の符号化シンボルを出力する。

10

【0141】

二番目に、前記(24、4)符号器に動作する場合を説明すると、4ビットのデータ速度インジケータa0、a1、a2、a3が前記符号器に入力され、残りのデータ速度インジケータa4、a5、a6は初期化され入力される。前記初期化により前記a4、a5、a6には0が入力される。すると、前記ウォルシュ符号発生器1100は長さ32であるウォルシュ符号W1、W2、W4、W8を出力する。前記ウォルシュ符号発生器1100から発生された前記W1、W2、W4、W8はそれぞれに対応する乗算器に提供される。例えば、前記W1は乗算器1120に提供され、前記W2は乗算器1122に提供される。前記W4は乗算器1124に提供され、前記W8は乗算器1126に提供される。前記ウォルシュ符号発生器1100から発生される長さ32のウォルシュ符号は下記<表16>のようになる。

20

【0142】

【表16】

30

ウォルシュ番号	ウォルシュ符号
W1	0101 0101 0101 0101 0101 0101 0101 0101
W2	0011 0011 0011 0011 0011 0011 0011 0011
W4	0000 1111 0000 1111 0000 1111 0000 1111
W8	0000 0000 1111 1111 0000 0000 1111 1111

【0143】

40

一方、前記4個のデータ速度インジケータは前記乗算器のうち、対応するそれぞれの乗算器に提供される。前記4個のデータ速度インジケータと前記乗算器の対応関係は下記<表17>のようになる。

【0144】

【表 17】

入力情報ビット	乗算器
a0	乗算器 1120
a1	乗算器 1122
a2	乗算器 1124
a3	乗算器 1126

【0145】

10

従って、前記乗算器それぞれに提供されるデータ速度インジケータとウォルシュ符号は下記<表 18>のように示すことができる。

【0146】

【表 18】

乗算器	ウォルシュ符号/マスク	入力情報ビット
乗算器 1120	W1 = 0101 0101 0101 0101 0101 0101 0101 0101	a0
乗算器 1122	W2 = 0011 0011 0011 0011 0011 0011 0011 0011	a1
乗算器 1124	W4 = 0000 1111 0000 1111 0000 1111 0000 1111	a2
乗算器 1126	W8 = 0000 0000 1111 1111 0000 0000 1111 1111	a3

20

【0147】

前記乗算器は前記<表 18>で示しているように自分に提供されるデータ速度インジケータを、ウォルシュ符号を構成するそれぞれのシンボルと乗算して加算器 1140 に出力する。

【0148】

即ち、前記乗算器 1120 は a0 を W1 それぞれのシンボルごとに乗算して加算器 1140 に出力し、前記乗算器 1122 は a1 を W2 それぞれのシンボルごとに乗算して前記加算器 1140 に出力する。一方、前記乗算器 1124 は a2 を W4 それぞれのシンボルごとに乗算して前記加算器 1140 に出力し、前記乗算器 1126 は a3 を W8 それぞれのシンボルごとに乗算して前記加算器 1140 に出力する。前記加算器 1140 は排他的論理和器を使用する。

30

【0149】

一方、乗算器 1128、1130、1132 には前記初期化により 0 の値を有する a4、a5、a6 がそれぞれ印加されることにより、前記ウォルシュ符号発生器 1100 からの W16 とマスク発生器 1105 からの M1、M2 に関わらず前記乗算器 1128、1130、1132 の出力は前記加算器 1140 の出力に影響を与えない。

【0150】

即ち、前記ウォルシュ符号生成器 1100 から乗算器 1128 に入力される W16 シンボル列の値に関わらず前記乗算器 1128 からは 0 であるシンボル列が出力される。また、前記マスク生成器 1105 から乗算器 1130、1132 に入力されるシンボル列の値に関わらず前記乗算器 1130、1132 からは 0 であるシンボル列が出力される。従って、前記乗算器 1128、1130、1132 の出力は前記排他的論理和器 1140 に入力されてもどのような影響を与えない。前記のように a4、a5、a6 を 0 に初期化することは、前記乗算器 1128、1130、1132 の出力を遮断するスイッチ動作と類似である。

40

【0151】

すると、前記排他的論理和器 1140 は前記乗算器 1120、1122、1124、1126、1128、1130、1132 から出力された長さ 32 であるすべてのシンボル

50

列をシンボル単位に排他的論理和をとって穿孔器 1 1 6 0 に出力する。

【0 1 5 2】

前記排他的論理和器 1 1 4 0 から出力される長さ 3 2 の符号化シンボル列 (W s) は下記<式 9>のように表現されることができる。

【0 1 5 3】

<式 9>

$$W s = (W 1 \times a 0) + (W 2 \times a 1) + (W 4 \times a 2) + (W 8 \times a 3)$$

【0 1 5 4】

この時点で前記制御器 1 1 1 0 は穿孔器 1 1 6 0 に (2 4、4) 符号に対する穿孔位置を穿孔するように指示する制御信号を出力する。すると、前記穿孔器 1 1 6 0 は穿孔位置が
10
貯蔵されているメモリ 1 1 7 0 から 4 ビット入力に対応した穿孔位置を受信して前記排他的論理和器 1 1 4 0 から提供される長さ 3 2 の符号化シンボルのうち、8 個の該当符号化シンボルを穿孔する。即ち、前記穿孔器 1 1 6 0 は前記排他的論理和器 1 1 4 0 から提供される長さ 3 2 の符号化シンボルのうち、前記メモリ 1 1 7 0 からの穿孔位置に該当する 8 シンボルを穿孔する。例えば、前記メモリ 1 1 7 0 からの穿孔位置が 0、1、2、3、4、5、6、1 6 である場合、前記穿孔器 1 1 6 0 は前記長さ 3 2 の符号化シンボルのうち、0、1、2、3、4、5、6、1 6 番目の符号化シンボルを穿孔して、2 4 個の符号化シンボルを出力する。

【0 1 5 5】

実施形態(復号器)

20

図 6 は前記図 5 の符号器に対応する復号器の構造を示す。

【0 1 5 6】

以下、本発明の実施形態による復号化動作を (2 4、4) 復号器と (2 4、7) 復号器に分けて説明する。

【0 1 5 7】

一番目に、(2 4、7) 符号器に対応した (2 4、7) 復号器の動作を説明すると、前記 (2 4、7) 復号器は前記符号器により符号化されそれぞれが +1、または -1 の値を有する 2 4 個の符号化シンボルから構成された符号化シンボル列を受信し、前記受信した符号化シンボル列は 0 挿入器 6 5 0 に入力される。

【0 1 5 8】

30

一方、制御器 6 3 0 は予め約束された符号長さの情報を受信して、(2 4、7) 符号器に対する穿孔位置に +1 と -1 の中間値である 0 を挿入するように指示する制御信号を出力する。前記 0 挿入器 6 5 0 は前記制御器 6 3 0 の制御下にメモリ 6 6 0 から 7 ビットの入力情報ビットに対応した穿孔位置に対する情報を受信する。例えば、前記 7 ビットの入力情報ビットに対応した穿孔位置は “0、4、8、1 2、1 6、2 0、2 4、2 8 番目” または “0、1、2、3、4、5、6、7 番目” になる。

【0 1 5 9】

従って、前記 0 挿入器 6 5 0 は前記受信した符号化シンボル列をなす 2 4 個の符号化シンボルのうち、前記メモリ 6 6 0 からの穿孔位置に 0 を挿入して 3 2 個のシンボルから構成された受信シンボル列をシンボル累積器 6 0 0 に出力する。すると、前記制御器 6 3 0
40
は前記 (2 4、7) 符号器による反復回数だけ反復されるシンボルを累積するように指示する制御信号を前記シンボル累積器 6 0 0 に出力する。この時、(2 4、7) 符号器はシンボル反復を遂行しないので前記シンボル累積器 6 0 0 は前記受信した 3 2 個の受信シンボルをそのままに出力する。すると、前記出力された 3 2 個の受信シンボルはウォルシュ符号との相関度計算器 6 2 0 に入力され、これと同時に乗算器 6 0 2、6 0 4、6 0 6 にそれぞれ入力される。すると、前記マスク生成器 6 1 0 は長さ 3 2 であるマスク関数 M 1 を生成して乗算器 6 0 2 に出力し、長さ 3 2 であるマスク関数 M 2 を生成して乗算器 6 0 4 に出力する、また、長さ 3 2 であるマスク関数 M 1 + M 2 を生成して乗算器 6 0 6 に出力する。この時、前記マスク関数は穿孔位置によって多様である。それぞれの穿孔位置が使用される場合、符号器で使用されたマスク関数が使用される。すると、前記乗算器 6 0 2 は
50

前記受信されたシンボルと前記マスク関数M 1をシンボル単位に乗算して出力し、前記乗算器6 0 4は前記受信されたシンボルと前記マスク関数M 2をシンボル単位に乗算して出力する。また、前記乗算器6 0 6は前記受信されたシンボルと前記マスク関数M 1 + M 2をシンボル単位に乗算して出力する。

【0 1 6 0】

この時、スイッチ6 5 2は前記制御器6 3 0の制御下に前記乗算器6 0 2で出力されたシンボル列を相関度計算器6 2 2に提供する。スイッチ6 5 4は前記制御器6 3 0の制御下に前記乗算器6 0 4で出力されたシンボル列を相関度計算器6 2 4に提供する。スイッチ6 5 6は前記制御器6 3 0の制御下に前記乗算器6 0 6で出力されたシンボル列を相関度計算器6 2 6に提供する。すると、前記相関度計算器6 2 0は前記長さ3 2である受信されたシンボル列を長さ3 2である3 2個のウォルシュ符号と相関度をすべて計算して、最高の相関度を有するウォルシュ符号番号、相関度値、そして、前段でマスク関数を使用されないことを示すマスク番号0を相関度比較器6 4 0に出力する。

10

【0 1 6 1】

前記相関度計算器6 2 2は前記長さ3 2である受信されたシンボル列とマスク関数M 1が乗算されたシンボル列を長さ3 2である3 2個のウォルシュ符号と相関度をすべて計算して、最高の相関度を有するウォルシュ符号番号、相関度値、そして、前段でマスク関数を使用されることを示すマスク番号1を前記相関度比較器6 4 0に出力する。前記相関度計算器6 2 4は前記長さ3 2である受信されたシンボル列とマスク関数M 2が乗算されたシンボル列を長さ3 2である3 2個のウォルシュ符号と相関度をすべて計算して、最高の相関度を有するウォルシュ符号番号、相関度値、そして、前段で使用されたマスク番号を示す2を前記相関度比較器6 4 0に出力する。前記相関度計算器6 2 6は前記長さ3 2である受信されたシンボル列とマスク関数M 1 + M 2が乗算されたシンボル列を長さ3 2である3 2個のウォルシュ符号と相関度をすべて計算して、最高の相関度を有するウォルシュ符号番号、相関度値、そして、前段で使用されたマスク番号を示す3を前記相関度比較器6 4 0に出力する。すると前記相関度比較器6 4 0は前記相関度計算器6 2 0、6 2 2、6 2 4、6 2 6から受信した値のうち、最大の相関度値を選択し、それに対するウォルシュ符号番号とマスク番号を接続して復号化ビットとして出力する。

20

【0 1 6 2】

二番目に、前記図5の(2 4、4)符号器に対応した復号器の動作を説明する。前記(2 4、4)符号器は、前記符号器により符号化され、それぞれが+ 1または- 1の値を有する2 4個の符号化シンボルから構成された符号化シンボル列を受信し、前記受信した符号化シンボル列は0挿入器6 5 0に入力される。

30

【0 1 6 3】

一方、制御器6 3 0は予め約束された符号長さの情報を受信して(2 4、4)符号器に対する穿孔位置に+ 1と- 1の中間値である0を挿入するように指示する制御信号を出力する。前記0挿入器6 5 0は前記制御器6 3 0の制御下にメモリ6 6 0から4ビットの入力情報ビットに対応した穿孔位置に対する情報を受信する。例えば、前記4ビットの入力情報ビットに対応した穿孔位置は“ 0、1、2、3、4、5、6、1 6 ”番目になることができる。従って、前記0挿入器6 5 0は前記受信した符号化シンボル列をなす2 4個の符号化シンボルのうち、前記メモリ6 6 0からの穿孔位置に0を挿入して3 2個のシンボルから構成された受信シンボル列をシンボル累積器6 0 0に出力する。すると、前記制御器6 3 0は前記(2 4、4)符号器による反復回数だけ反復されるシンボルを累積するように指示する制御信号を前記シンボル累積器6 0 0に出力する。

40

【0 1 6 4】

この時、(2 4、4)符号器が1回のシンボル反復を遂行したので、前記シンボル累積器6 6 0は前記受信した3 2個の受信シンボルのうち、反復位置にある2シンボルずつを加算して長さ1 6であるシンボル列を出力する。すると、前記出力された1 6個の受信シンボルは相関度計算6 2 0に入力され、これと同時に乗算器6 0 2、6 0 4、6 0 6にそれぞれ入力される。この時、前記乗算器6 0 2、6 0 4、6 0 6の出力端のスイッチ6 5 2

50

、654、656は前記制御器630の制御下にターンオフ(Turn off)されるので、前記乗算器602、604、606の出力は捨てられる。すると、前記制御器630は符号長さと同一のウォルシュ符号との相関度を計算するように指示する制御信号(即ち、長さ16であるウォルシュ符号と相関度計算するように指示する制御信号)を前記相関度計算器620に出力する。この時、前記相関度計算器620は前記長さ16である受信されたシンボル列を長さ16である16個のウォルシュ符号と相関度をすべて計算し、最高の相関度を有するウォルシュ符号番号、相関度値、そして、マスク番号を示す値を相関度比較器640に出力する。すると前記相関度比較器640は前記ウォルシュ符号との相関度計算器620から受信した値のうち、ウォルシュ番号とマスク番号を接続して復号化ビットに出力する。

10

【図面の簡単な説明】

【0165】

【図1】 通常的な符号分割多重接続移動通信システムでの逆方向データ速度インジケータチャネルの送信器構造を示した図である。

【図2】 本発明の実施形態による符号分割多重接続移動通信システムでの最適(24、4)符号化器の構成を示している図である。

【図3】 本発明の実施形態による符号語構成の一例を示している図である。

【図4】 本発明の実施形態による符号分割多重接続移動通信システムでの最適(24、7)符号化器の構成を示している図である。

【図5】 本発明の実施形態による符号分割多重接続移動通信システムで最適(24、4)符号化器と最適(24、7)符号化器を同時に支援する符号化器の構成を示している図である。

20

【図6】 本発明の実施形態による符号分割多重接続移動通信システムでの復号化器の構成を示している図である。

【図7】 本発明の実施形態による生成行列を利用した(24、4)符号器の構成を示している図である。

【図8】 本発明の実施形態による生成行列を利用した(24、7)符号器の構成を示している図である。

【図9】 本発明の実施形態による符号分割多重接続移動通信システムで最適(24、4)符号化器と最適(24、7)符号化器を同時に支援する符号化器の構成を示している図である。

30

【符号の説明】

【0166】

500 ウォルシュ符号発生器

505 マスク発生器

510 制御器

520～532 乗算器

540 排他的論理和器

550 反復器

560 穿孔器

570 メモリ

40

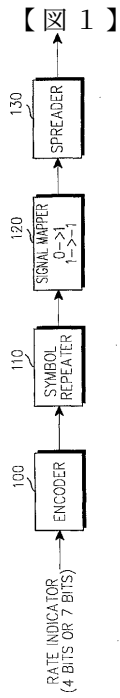


FIG. 1

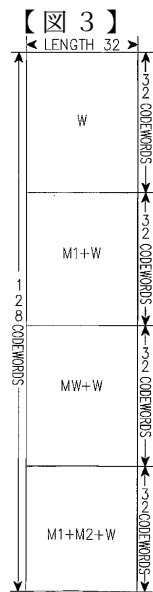


FIG. 3

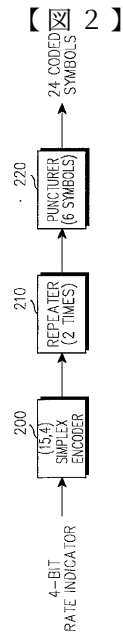


FIG. 2

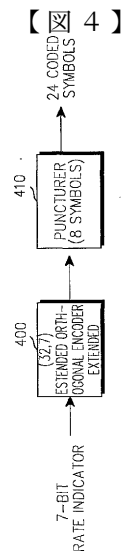


FIG. 4

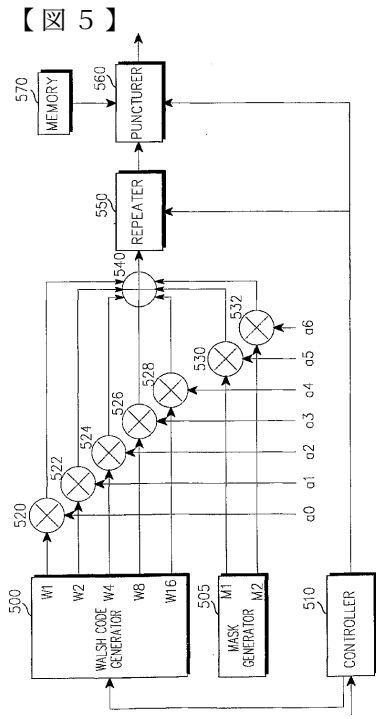


FIG. 5

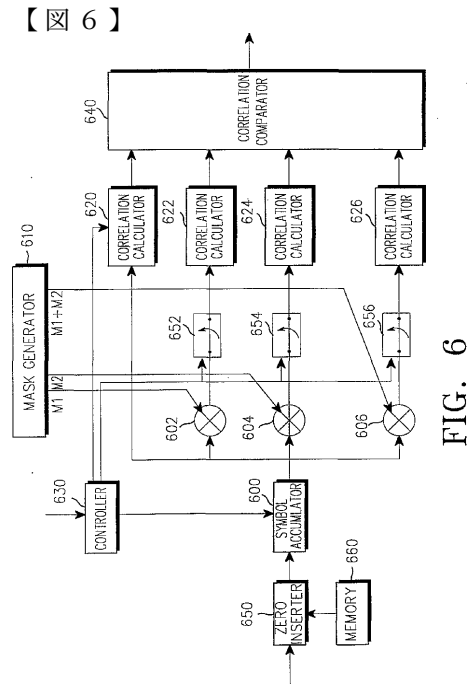


FIG. 6

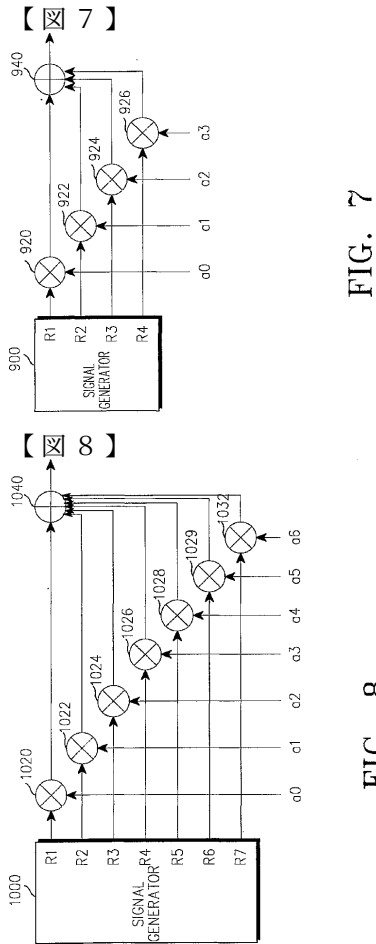


FIG. 7

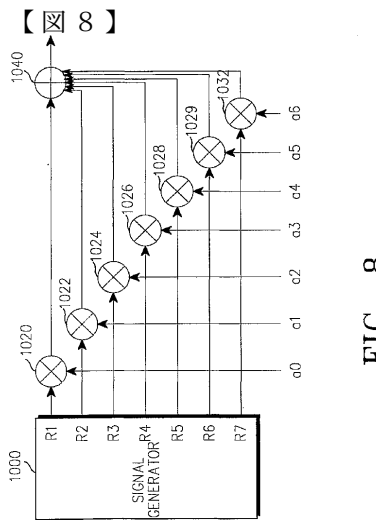


FIG. 8

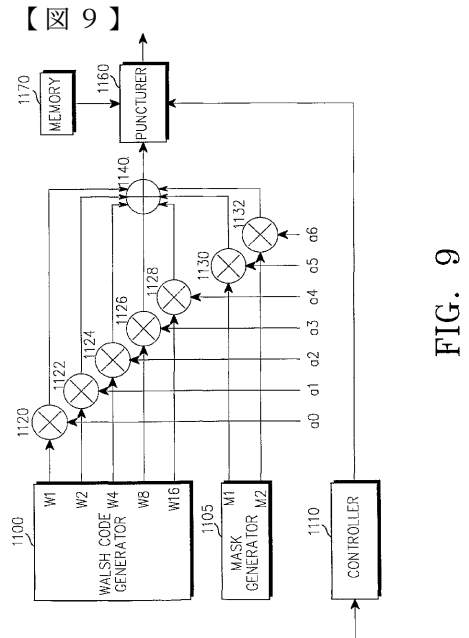


FIG. 9

フロントページの続き

(31)優先権主張番号 2001/15787

(32)優先日 平成13年3月26日(2001.3.26)

(33)優先権主張国 韓国(KR)

(72)発明者 ホーキュ・チョイ

大韓民国・137-030・ソウル・ソチョグ・チャンウォン・ドン・56-2

(72)発明者 チェースン・チャン

大韓民国・427-010・キョンギド・クァチョンシ・チュンガン・ドン・(番地なし)・
ジュゴン・アパートメント・#1102-203

(72)発明者 ヨンスン・キム

大韓民国・135-283・ソウル・カンナムグ・デチ・3・ドン・(番地なし)・サンヨン・
アパートメント・#6-607

審査官 藤井 浩

(56)参考文献 国際公開第01/003366 (WO, A1)

SAMSUNG ELECTRONICS CO.LTD, Dynamic split mode for TFCI, TSG-RAN WORKING GROUP 1 MEETING #17, 2000年10月21日, TSGR1#17(00)1269, URL, http://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_17/Docs/PDFs/R1-00-1269.pdf

(58)調査した分野(Int.Cl., DB名)

H03M 13/00 - 13/53

H04B 1/707

H04L 1/00