(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5999634号
(P5999634)
(45)発行日　平成28年9月28日(2016.9.28)　(24)登録日　平成28年9月9日(2016.9.9)

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

請求項の数 2　(全 16 頁)

(21)出願番号　特願2012-206395(P2012-206395)
(22)出願日　平成24年9月19日(2012.9.19)
(65)公開番号　特開2014-64065(P2014-64065A)
(43)公開日　平成26年4月10日(2014.4.10)
審査請求日　平成27年7月16日(2015.7.16)

(73)特許権者　504145320
国立大学法人福井大学
福井県福井市文京３丁目９番１号
(74)代理人　100111855
弁理士　川崎　好昭
(72)発明者　岩田　賢一
福井県福井市文京３丁目９番１号　国立大学法人福井大学内
(72)発明者　福間　慎治
福井県福井市文京３丁目９番１号　国立大学法人福井大学内
(72)発明者　大島　怜也
福井県福井市文京３丁目９番１号　国立大学法人福井大学内

最終頁に続く

(54)【発明の名称】演算回路設定方法

(57)【特許請求の範囲】
【請求項１】
符号長2^m-1及び情報ビット数2^m-m-1の２元ハミング符号の受信語r_j，ｊ＝１，２，…，2^m-1に対してＸＯＲ演算を複数回行ってエラー位置を検出するシンドロームｓ$_i$，ｉ＝１，２，…，ｍを生成する復号化処理の演算回路設定方法であって、以下の規則１から規則４に基づいて表記された順序でＸＯＲ演算を行うように演算回路を構成する演算回路設定方法。
＜規則１＞
左端の列に上から順にs_1からs_mを並べる。
＜規則２＞
ｊの二進表記である$b_{m-1}b_{m-2}\cdots b_0$，$b_i \in \{0, 1\}$について以下の値を求め、
【数１４】

$$i=\sum_{k=0}^{m-1} b_k$$

この値が同じものを小さい値１から大きい値ｍの順に左から右に列として配置し、各列ではｊの小さい値から大きい値の順に上から下に配置する。
＜規則３＞
規則２で作成した配置図において、$b_{m-1}b_{m-2}\cdots b_0$を値０か値１の１ビットを保持する節点ｊとし、各節点ｊの保持する値をｖ（ｊ）として、演算開始時刻での値ｖ（ｊ）をｒ

$_j$とする。左端の列に配置された節点ｊ＝2^{i-1}をs_iに対応させ、s_iを求めるr_jのＸＯＲ演算の演算式に基づいてr_jに対応する節点ｊ同士を線で結ぶ。
＜規則４＞
ｊ＜ｊ’を満たす節点ｊと節点ｊ’との間を結ぶ線がある場合には、演算開始時刻から１回のＸＯＲ演算に起因する遅延時間に基づいて設定されたタイミングでｖ（ｊ）及びｖ（ｊ’）のＸＯＲ演算を行い、節点ｊの値をＸＯＲ演算結果とする。
【請求項２】
ｍ＝ｋ－１（ｋは、３以上の自然数）におけるシンドロームを生成する演算回路に基づいてｍ＝ｋにおけるシンドロームを生成する演算回路を設定する請求項１に記載された演算回路設定方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、記憶処理装置、通信処理装置等のデータ処理装置に用いられる符号化・復号化装置及び符号化・復号化処理を行う演算回路の設定方法に関する。
【背景技術】
【０００２】
記憶処理装置や通信処理装置等のデータ処理装置は、一般的にデータの誤り検出及び訂正を行う誤り訂正機能を備えている。誤り訂正機能としては、１ビット訂正可能なハミング符号（短縮化ハミング符号語）を用いた符号化・復号化方式が提案されている。こうした符号化・復号化方式では、情報ビット列（情報ベクトル）及び生成行列に基づいてハミング符号を生成する符号化処理を実行するとともに、生成行列に対応する検査行列（パリティ検査行列）及びハミング符号に基づいてエラー位置を検出する復号化処理を実行する。
【０００３】
情報ビット列は、記憶処理装置では記録媒体に記憶される入力データが該当し、通信処理装置では送信データが該当する。そして、ハミング符号は、情報ビット列及びパリティビット列（パリティデータ）を組み合せて構成される。復号化処理では、記憶処理装置では記録媒体から読み出されるハミング符号からなる出力データのエラー位置を検出し、通信処理装置では受信データのエラー位置を検出して、そのエラー位置を示すデータ（以下「シンドローム（syndrome）」と称する）を生成する。エラー訂正処理では、復号化処理により検出されたエラー位置のエラービットを訂正してデータ出力処理を行う。
【０００４】
簡単な誤り訂正符号の１つである２元ハミング符号を用いた符号化・復号化装置における排他的論理和（ＸＯＲ）の演算処理について考える。２元体をF_2と表記し、その元を０と１で表す。２元体での加法については、以下の通り表記する。
【数１】

$$0 \oplus 0 = 1 \oplus 1 = 0, 1 \oplus 0 = 0 \oplus 1 = 1$$

なお、この表記において、２項間の記号は排他的論理和（ＸＯＲ）である。
２以上の任意の整数ｍに対して、$F_2=\{0, 1\}$上のすべての非零のｍ次元ベクトルを列として並べたｍ行2^m-1列の行列を検査行列として定義される符号は符号長2^m-1、情報ビット数2^m-m-1の２元ハミング符号であり、（2^m-1，2^m-m-1）ハミング符号と表記する。（2^m-1，2^m-m-1）ハミング符号を定める検査行列をＨと表記する。特に、検査行列のｊ列目h_j^Tを
$h_j^T=(h_{1,j}, h_{2,j}, \cdots, h_{i,j}, \cdots, h_{m,j}) \in F_2^m$
とし、

【数２】

$$\sum_{i=1}^{m} h_{i,j} \cdot 2^{m-i} = j$$

を満たすようにｈ$_{j}$を定めた検査行列をＨ$_{m}$と表記する。ただし、ｈ^{T}はｈの転置を表す。（２^{m}－１，２^{m}－ｍ－１）ハミング符号は最小距離が３であり、ハミング限界を等号で満たす完全符号である（非特許文献１参照）。

【０００５】

そして、（２^{m}－１，２^{m}－ｍ－１）ハミング符号における検査行列Ｈでの列の並べ方は任意であり、巡回符号である（２^{m}－１，２^{m}－ｍ－１）ハミング符号はシフトレジスタを用いたｍ段符号器や２^{m}－ｍ－１段符号器及び巡回ハミング符号の復号器を用いた符号化・復号化装置によって実現することができる（非特許文献１参照）。

【０００６】

これに対して、符号器及び復号器の高速化を図る方法として、符号語や復号結果の各ビットを並列に出力する並列符号器及び並列復号器が提案されている（特許文献１参照）。

【発明の概要】

【発明が解決しようとする課題】

【０００７】

こうした並列符号器及び並列復号器については、演算処理速度が高速化するものの演算回路が複雑化するため実用化に至っていないのが現状である。演算処理では２入力１出力のＸＯＲ演算を組み合せて処理するのが一般的であるが、こうしたＸＯＲ演算の回数は、演算処理速度の向上、演算回路の構成等に直接影響を与えるため、安定した並列処理を行う上で必要となる最適な演算回数の設定が求められている。

【０００８】

そこで、本発明は、並列処理を行う上で必要となる最小限のＸＯＲ演算回数により演算処理を行うことができる符号化・復号化装置を提供することを目的とする。

【課題を解決するための手段】

【００１０】

本発明に係る演算回路設定方法は、符号長２^{m}－１及び情報ビット数２^{m}－ｍ－１の２元ハミング符号の受信語ｒ$_{j}$，ｊ＝１，２，…，２^{m}－１に対してＸＯＲ演算を複数回行ってエラー位置を検出するシンドロームｓ$_{i}$，ｉ＝１，２，…，ｍを生成する復号化処理の演算回路設定方法であって、以下の規則１から規則４に基づいて表記された順序でＸＯＲ演算を行うように演算回路を構成する。

＜規則１＞

左端の列に上から順にｓ$_{1}$からｓ$_{m}$を並べる。

＜規則２＞

ｊの二進表記であるｂ$_{m-1}$ｂ$_{m-2}$…ｂ$_{0}$，ｂ$_{i}$ ∈{０，１}について以下の値を求め、

【数１４】

$$i = \sum_{k=0}^{m-1} b_k$$

この値が同じものを小さい値１から大きい値ｍの順に左から右に列として配置し、各列ではｊの小さい値から大きい値の順に上から下に配置する。

＜規則３＞

規則２で作成した配置図において、ｂ$_{m-1}$ｂ$_{m-2}$…ｂ$_{0}$を値０か値１の１ビットを保持する節点ｊとし、各節点ｊの保持する値をｖ（ｊ）として、演算開始時刻での値ｖ（ｊ）をｒ$_{j}$とする。左端の列に配置された節点ｊ＝２$^{i-1}$をｓ$_{i}$に対応させ、ｓ$_{i}$を求めるｒ$_{j}$のＸＯＲ演算の演算式に基づいてｒ$_{j}$に対応する節点ｊ同士を線で結ぶ。

＜規則４＞

ｊ＜ｊ’を満たす節点ｊと節点ｊ’との間を結ぶ線がある場合には、演算開始時刻から１回のＸＯＲ演算に起因する遅延時間に基づいて設定されたタイミングでｖ（ｊ）及びｖ（ｊ’）のＸＯＲ演算を行い、節点ｊの値をＸＯＲ演算結果とする。

【００１１】

さらに、上記の演算回路設定方法において、ｍ＝ｋ－１（ｋは、３以上の自然数）におけるシンドロームを生成する演算回路に基づいてｍ＝ｋにおけるシンドロームを生成する演算回路を設定する。

【発明の効果】

【００１２】

本発明によれば、並列処理を行う上で必要となる最小限のＸＯＲ演算回数により演算処理を行うことができるので、演算処理速度の向上、演算回路の簡略化を図ることができる。

【図面の簡単な説明】

【００１３】

【図１】本実施形態に係る符号化・復号化装置に関するブロック構成図である。

【図２】並列符号化処理に用いる演算回路の例に関する概略構成図である。

【図３】並列復号化処理に用いる演算回路の例に関する概略構成図である。

【図４】$R_m^{(s)}$をパスカルの三角形における最初の７段を用いた場合に関する説明図である。

【図５】算出された$R_m^{(s)}$の値を示すリストである。

【図６】並列復号化処理における演算回路に関する表記例を示す説明図である。

【図７】ｍ＝３の場合の簡略表記図を作成する手順の例を示す説明図である。

【図８】ｍ＝２，３，４，５の場合の簡略表記図をまとめて示した説明図である。

【図９】ｍ＝３におけるs_iの算法及び算法に用いる値ｊの一覧を示す説明図である。

【図１０】ｍ＝５におけるs_iの算法及び算法に用いる値ｊの一覧を示す説明図である。

【発明を実施するための形態】

【００１４】

以下、本発明に係る実施形態について詳しく説明する。図１は、本実施形態に係る符号化・復号化装置に関するブロック構成図である。符号化・復号化装置１は、符号化・復号化処理部１０、符号化データ生成部１１及び誤り訂正処理部１２を備えており、各部の機能を実現する回路構成を有するＩＣにより構成することができる。

【００１５】

この例では、符号化・復号化装置１は、記憶部２０に接続されており、記憶部２０に書き込むデータの符号化を行うとともに記憶部２０から読み出されるデータの復号化を行う。なお、符号化・復号化装置１を通信装置に適用する場合には、外部の通信ネットワークと送受信する送受信部に接続する。

【００１６】

符号化・復号化処理部１０は、装置の外部から転送される入力データに対して符号化処理を行うとともに記憶部２０から読み出されるデータを復号化処理する。符号化処理では、入力データの情報ビット列に基づいてパリティビット列からなるパリティデータを生成する。生成されたパリティデータは、入力データとともに符号化データ生成部１１に入力されて、入力データの情報ビット列及びパリティデータからなる符号化データが生成されて記憶部２０に送信されて記憶される。復号化処理では、記憶部２０から読み出された符号化データのエラー位置の検出を行い、検出されたエラー位置を示すシンドロームを生成する。生成されたシンドロームは、読み出された符号化データとともに誤り訂正処理部１２に入力されて、エラー位置に対応するデータを訂正したデータを出力データとして装置の外部に転送する。

【００１７】

通信装置に適用される場合には、入力データとして送信データを符号化・復号化処理部１０に入力し、符号化データ生成部１１から出力される符号化データを送受信部に転送し

て通信ネットワークを介して送信する。また、受信データは、送受信部で受信した後符号化・復号化処理部１０に入力してエラー位置の検出を行った後誤り訂正処理部１２により訂正処理を行って出力される。

【００１８】

次に、符号化・復号化処理部１０における演算処理について説明する。（2^m-1，2^m-m-1）ハミング符号の符号語ｃ、符号語ｃに対応させた情報ビット列ｂ、及び、符号語ｃを記憶した後読み出された受信語ｒをそれぞれ以下の通り表記する。

$c=(c_1, c_2, \cdots, c_n) \in F_2^n,\ n=2^m-1$

$b=(b_1, b_2, \cdots, b_k) \in F_2^k,\ k=2^m-m-1$

$r=(r_1, r_2, \cdots, r_n) \in F_2^n$

そして、受信語ｒに対して復号化処理して類推される符号語Ｃを復号語とし、以下の通り表記する。

$C=(C_1, C_2, \cdots, C_n) \in F_2^n$

【００１９】

ハミング符号におけるシンドローム復号法では、受信語ｒに対する検査行列H_mによるシンドロームｓは、以下の通り表記し、

$s=(s_m, s_{m-1}, \cdots, s_1) \in F_2^m$

シンドロームｓを以下の通り定める。

$s=rH_m^T$

また、シンドロームｓに対して$(s)_2$を以下の通り定める。

【数３】

$$(s)_2=\sum_{i=1}^{m} s_i 2^{i-1}$$

そして、復号語Ｃを以下の通り定める。

【数４】

$$C_i=r_i \oplus I(i=(s)_2), i=1,2,\cdots,n$$

ここで、Ｉ(i=$(s)_2$)は、ｉが$(s)_2$に等しい場合には１となり、ｉが$(s)_2$に等しくない場合には０となる。そのため$(s)_2$が０ならば受信語ｒをそのまま復号語Ｃとし、$(s)_2$が１以上ｎ以下であるときは１箇所のエラー位置を推定して訂正する復号化処理が行われる。こうしたハミング符号におけるシンドローム復号法は、通信路が定常無記憶である２元対称通信路である場合には、最尤復号となる。

【００２０】

以上説明した符号化処理及び復号化処理について、以下の検査行列H_3

【数５】

$$B_n=\left(\sum_{i=0}^{k} 2^i, k=0,1,\cdots,n\right)$$

で定義される（７，４）ハミング符号による並列符号化処理及びシンドローム復号法による並列復号化処理を例にとり、ＸＯＲ演算回数を説明する。この例では、情報ビット列ｂ及びそれに対応する符号語ｃは以下の通りになる。

$b=(b_1, b_2, b_3, b_4) \in F_2^4$

$c=(c_1, c_2, c_3, c_4, c_5, c_6, c_7) \in F_2^7$

そして、c_3、c_5、c_6及びc_7のパリティビットを以下の通り定める。

$c_3=b_1$

$c_5=b_2$

$c_6=b_3$

$c_7 = b_4$
この場合、残りのパリティビットは以下の通り定められる。
【数６】

$$c_1 = c_3 \oplus c_5 \oplus c_7$$

$$c_2 = c_3 \oplus c_6 \oplus c_7$$

$$c_4 = c_5 \oplus c_6 \oplus c_7$$

こうして、４ビットの情報ビット列ｂの入力に対してパリティビットc_1，c_2及びc_4を定める符号器の一部は、それぞれ右辺の項のＸＯＲ演算を処理する回路により実現することができる。
【００２１】
図２は、並列符号化処理に用いる演算回路の例に関する概略構成図である。図２では、並列復号化処理を行う演算回路におけるＸＯＲ演算の順序を木構造で表記している。パリティビットc_iの演算処理の最後のＸＯＲ演算を根節点とし、c_iの演算処理に用いたＸＯＲ演算を内節点とし、c_iの演算処理に用いたb_iの集合を葉節点の集合とし、これらの節点を樹枝状に連接して枝とするツリー図で表記しており、入力側である右側から出力側である左側に枝をたどることでＸＯＲ演算の順序が設定されるようになっている。
【００２２】
図２（ａ）に示す例では、（７，４）ハミング符号における３ビットのパリティビットについて、それぞれ独立に２回のＸＯＲ演算を行っており、合計６回のＸＯＲ演算で各パリティビットを求めている。この場合、パリティビットを求める上記の演算式においてＸＯＲ演算は可換であることから、ＸＯＲ演算の順序を変えることができる。上記の演算式では、c_3及びc_7、c_5及びc_7並びにc_6及びc_7のそれぞれのＸＯＲ演算が共通している。そのため、例えば、共通するc_6及びc_7のＸＯＲ演算を先に処理して、その演算結果を演算式に代入すれば、c_2及びc_4を３回のＸＯＲ演算で求めることができるため、３ビットのパリティビットを合計５回のＸＯＲ演算で求めることができる。図２（ｂ）は、５回のＸＯＲ演算を行う場合の演算回路の概略構成図である。
【００２３】
上記の演算式のようにパリティビットを求める２項演算において２項の対に共通のＸＯＲ演算が存在する場合は、ＸＯＲ演算の回数を減らすことができる。以下の説明では、ＸＯＲの２項演算において２項の対に存在する共通の対を「冗長な対」と称し、冗長な対に基づいてＸＯＲ演算を削減することを「冗長な対の削減」と称する。なお、上記の演算式では、c_1、c_2及びc_4の３ビットのパリティビットを４回のＸＯＲ演算で求めることはできない。なぜなら、（c_3，c_7）、（c_5，c_7）及び（c_6，c_7）の３つの冗長な対のいずれかを削減した場合、２項の対には同じ対がなくすべて相異なっているため、冗長な対が存在しなくなってこれ以上ＸＯＲ演算の回数を削減することはできなくなる。
【００２４】
次に、（７，４）ハミング符号の並列復号化処理における、ＸＯＲ演算に起因する遅延時間について説明する。例えば、上記の検査行列H_3で定まる（７，４）ハミング符号の受信語ｒ及びそれに対応するシンドロームｓ以下の通りになる。
$r = (r_1, r_2, r_3, r_4, r_5, r_6, r_7) \in F_2^7$
$s = (s_3, s_2, s_1) \in F_2^3$
そして、s_3、s_2及びs_1は、以下の通り定まる。

【数７】

$$s_3 = r_4 \oplus r_5 \oplus r_6 \oplus r_7$$
$$s_2 = r_2 \oplus r_3 \oplus r_6 \oplus r_7$$
$$s_1 = r_1 \oplus r_3 \oplus r_5 \oplus r_7$$

【００２５】

図３は、並列復号化処理に用いる演算回路の例に関する概略構成図である。図３についても、図２と同様にツリー図により演算回路を表記している。上記のシンドロームビットの演算式においても冗長な対（r_6，r_7）、（r_5，r_7）及び（r_3，r_7）が存在する。そのため、冗長な対を削減することで、ＸＯＲ演算の回数を削減することができる。図３（ａ）は、冗長な対（r_6，r_7）を削減した演算回路の例を示している。なお，図３（ａ）では、r_i＝０、ｉ＝１，２，４とすれば，上記の情報ビット列ｂに対応するパリティビットc_i、ｉ＝１，２，４を求める演算回路として使用することもできる。

【００２６】

また、図３（ｂ）は、ＸＯＲ演算に起因する遅延時間の観点から構成した演算回路の例を示している。２入力１出力となる１回のＸＯＲ演算に起因する遅延時間をｔとすると、図３（ａ）に示す例ではシンドロームビットs_iを求める演算処理に演算開始から時間３ｔを要するが、図３（ｂ）では演算開始から時間２ｔとなり、遅延時間を短縮することができる。節点及び枝で構成されるs_iに関する木構造において、同時に演算処理されるＸＯＲ演算の段数を「木の深さ」とすれば、図３（ａ）に示す場合は木の深さが３となり、図３（ｂ）に示す例では木の深さが２となる。s_iに対応させたグラフの木の深さをd_iとすると、s_iのＸＯＲ演算に起因する遅延時間は$d_i t$となる。そして、並列復号化処理に用いる回路構成全体の遅延時間をｄｔとすれば、ｄは以下の通り定めることができる。

$d = max\{d_i, i = 1, 2, \cdots, m\}$

すなわち、並列復号化処理に用いる各シンドロームビットの演算回路に対応する木構造の遅延時間のうち最も長い遅延時間が回路構成全体の遅延時間となる。

【００２７】

次に、並列符号化処理及び並列復号化処理におけるＸＯＲ演算回数の下界について説明する。（2^m-1，2^m-m-1）ハミング符号の並列符号化処理においてｍビットのパリティビットを求めるために必要とするＸＯＲ演算回数の最小値を$N_{m,e}$と表記し、（2^m-1，2^m-m-1）ハミング符号の並列復号化処理においてｍビットのシンドロームを求めるために必要とするＸＯＲ演算回数の最小値を$N_{m,d}$と表記する。

【００２８】

（2^m-1，2^m-m-1）ハミング符号の検査行列Ｈの各行は、それぞれ2^{m-1}個の「１」の要素があり、符号化処理においてｍビットのパリティビットの演算処理は、各パリティビットを独立して$2^{m-1}-2$回のＸＯＲ演算により行う場合、ＸＯＲ演算回数の合計は、ｍ（$2^{m-1}-2$）回となる。同様に、（2^m-1，2^m-m-1）ハミング符号の並列復号化処理においてｍビットのシンドロームの演算処理は、各シンドロームビットを独立して$2^{m-1}-1$回のＸＯＲ演算により行う場合、ＸＯＲ演算回数の合計は、ｍ（$2^{m-1}-1$）回となる。

【００２９】

ｍビットのパリティビット(又はシンドローム）の演算処理を各ビット毎に独立してＸＯＲ演算により行った場合に、冗長な対の削減可能なＸＯＲ演算回数をR_mと表記し、R_mのある上界を$R_m^{(s)}$と表記する。この場合、最小値$N_{m,e}$及び$N_{m,d}$は、ｍ＝２，３，…に対して以下の式を満たす。

$m(2^{m-1}-2) - R_m^{(s)} \leqq N_{m,e} < m(2^{m-1}-2)$ ・・・・（１）

$m(2^{m-1}-1) - R_m^{(s)} \leqq N_{m,d} < m(2^{m-1}-1)$ ・・・・（２）

【００３０】
そして、R_mについては、以下の式を満たすようになる。
$R_m \leqq （m-4）（2^{m-1}+1）+6$・・・・（３）
式（３）については、次のように証明することができる。（2^m-1，2^m-m-1）ハミング符号の検査行列Ｈのある２列の列番号を表すj_1及びj_2を$1 \leqq j_1 < j_2 \leqq 2^m-1$とし、あるｋ個の行番号を表す$i_1$，$i_2$，…，$i_k$を$1 \leqq i_1 < i_2 < \cdots < i_k \leqq m$とする。検査行列Ｈの$j_1$列目と$j_2$列目において、$i_1$，$i_2$，…，$i_k$行目の要素がすべて「１」であれば、冗長の対に対応するｋ回のＸＯＲ演算をまとめて１回のＸＯＲ演算で行うことで、ｋ－１回のＸＯＲ演算が削除可能となる。

【００３１】
（2^m-1，2^m-m-1）ハミング符号の検査行列Ｈは、すべての非零のｍ次元ベクトルを列として並べたｍ行2^m-1列の行列である。検査行列Ｈの各列はすべて異なっており、「１」の要素がk_1個であるj_1列目と「１」の要素がk_2個であるj_2列目に対応するＸＯＲ演算をまとめて１回のＸＯＲ演算により行うことで、$k_1 \neq k_2$である場合には$\min\{k_1, k_2\}-1$回までＸＯＲ演算の回数を削除可能であり、$k_1 = k_2$である場合にはk_1-2回までＸＯＲ演算の回数を削除可能である。したがって、以下の関係式が導かれる。

【数８】

$$R_m \leqq \sum_{k=2}^{m}(k-2)\binom{m}{k} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots(4)$$

$$= m2^{m-1} - \sum_{k=0}^{2} k\binom{m}{k} - 2\left\{2^m - \sum_{k=0}^{2}\binom{m}{k}\right\}$$

$$= m2^{m-1} - 2^{m+1} + m + 2 \quad \cdots\cdots\cdots\cdots\cdots(5)$$

$$= (m-4)(2^{m-1}+1) + 6$$

以上のとおり式（３）が証明される。そして、式（３）を用いて式（１）及び式（２）は、以下の通り表記でき、ＸＯＲ演算回数の下界が評価できる。
$2^{m+1}-3m-2 \leqq N_{m,e}$・・・・（６）
$2^{m+1}-2m-2 \leqq N_{m,d}$・・・・（７）
また、$R_m^{(s)}$については、以後式（３）により以下の通り表記する。
$R_m^{(s)} = （m-4）（2^{m-1}+1）+6$・・・・（８）
図４は、$R_m^{(s)}$をパスカルの三角形における最初の７段を用いた場合に関する説明図である。この場合、$R_2^{(s)}=0$となり、$m \geqq 3$では、以下の漸化式を満たす。
$R_m^{(s)} = 2R_{m-1}^{(s)} + 2^{m-1} - m$・・・・（９）
さらに、$R_m^{(s)}$を係数とする以下の母関数

【数９】

$$g(z) = \sum_m R_m^{(s)} z^m$$

は、式（５）により以下の式で与えられる（R. L. Graham, D. E. Knuth, and O. Patashnik, Concrete Mathematics, Addison-Wesley, 1989）。

【数１０】

$$g(z) = \frac{z}{(1-2z)^2} - \frac{2}{1-2z} + \frac{z}{(1-z)^2} + \frac{2}{1-z} = \frac{z^3}{(1-z)^2(1-2z)^2}$$

$m=2, 3, \cdots, 12$の場合の$R_m^{(s)}$の値を図５に示す。

【００３２】

$R_{n+3}^{(s)}$，$n=0, 1, 2, \cdots$は、次の２つのパターンで生成される数列に等しいことが知られている（N. J. A. Sloane, On-line Encyclopedia of Integer Sequences）。一つ目のパターンは、１から$n+2$までの整数の集合$A_n=\{1, 2, \cdots, n+2\}$における空集合を除くすべての部分集合の径の和に等しい。ここで、集合Ａの径は、集合Ａに属する最大値から集合Ａに属する最小値を引いた値として定義される。例えば、$n=0$ならば$A_0=\{1, 2\}$であり、空集合を除くすべての部分集合$\{1\}, \{2\}, \{1, 2\}$の径はそれぞれ０，０，１となってその合計は１となる。同様に$n=1$ならば$A_1=\{1, 2, 3\}$であり，空集合を除くすべての部分集合の径の和は６となる。

【００３３】

二つ目のパターンは、以下の通り設定された、２のべき乗の部分和の列B_nに対する畳み込みと等しい。

【数１１】

$$B_n=\left(\sum_{i=0}^{k}2^i, k=0,1,\cdots,n\right)$$

たとえば、$B_0=(1)$、$B_1=(1, 3)$、$B_2=(1, 3, 7)$に対して、畳み込みは、それぞれ以下の通りとなる。

B_0；１

B_1；$1\times3+3\times1=6$

B_2；$1\times7+3\times3+7\times1=23$

【００３４】

次に、上述のように求められたＸＯＲ演算回数の下界を下限とする回路構成について説明する。$(2^m-1, 2^m-m-1)$ハミング符号を用いた並列符号化処理及び並列復号化処理におけるＸＯＲ演算回数の下限は、式（６）及び式（７）により以下の通り設定される。

$N_{m,e}=2^{m+1}-3m-2=2(2^m-m-1)-m$・・・・（１０）

$N_{m,d}=2^{m+1}-2m-2=2(2^m-m-1)$・・・・（１１）

ここでは、$(2^m-1, 2^m-m-1)$ハミング符号の検査行列として上記の検査行列H_mを用い、２入力１出力となる１回のＸＯＲ演算に起因する遅延時間をｔとして、ＸＯＲ演算回数を下限に設定するとともに遅延時間$(m-1)$ｔとなる回路構成について述べる。まず、並列復号化処理における遅延時間が$(m-1)$ｔとなる回路構成について、$m=2$から順に再帰的に$m=2, 3, \cdots$を考える。シンドロームｓにおけるs_i，$i=1, 2, \cdots, m$の演算開始時刻を０とする。$m=2$の場合、s_1及びs_2は以下の通り設定される。

【数１２】

$$s_1=r_1\oplus r_3$$

$$s_2=r_2\oplus r_3$$

【００３５】

図６は、並列復号化処理における演算回路に関する表記例を示す説明図である。図６（ａ）では、s_1及びs_2を求める演算回路を図３と同様の表記方法で記載している。以後の説明を容易にするために、こうした演算回路の簡略表記の方法を次のように定める。簡略表記では、まず、以下の式を満たすｊを設定する。

【数１３】

$$j=\sum_{k=0}^{m-1}b_k2^k$$

そして、設定されたｊに関する二進数表記$(b_{m-1}b_{m-2}\cdots b_0)_2$を用いて$r_j$の代わりに単に$b_{m-1}b_{m-2}\cdots b_0$と表記する。そして、以下の規則１から規則４に従って表記する。

＜規則１＞
左端の列に上から順にs_1からs_mを並べる。
＜規則２＞
$b_{m-1}b_{m-2}\cdots b_0$，$b_i \in\{0, 1\}$について以下の値を求め、
【数１４】

$$i=\sum_{k=0}^{m-1} b_k$$

この値が小さい値１から大きい値ｍの順に右から左に列として分類し、各列では$j=(b_{m-1}b_{m-2}\cdots b_0)_2$の小さい値から大きい値の順に上から下に配置する。
＜規則３＞
規則２で作成した配置図において、$b_{m-1}b_{m-2}\cdots b_0$を値０か値１の１ビットを保持する節点とし、節点$(b_{m-1}b_{m-2}\cdots b_0)_2$又は節点ｊ，$j=(b_{m-1}b_{m-2}\cdots b_0)_2$と表記する。節点ｊの保持する値をｖ（ｊ）とし、時刻０での値ｖ（ｊ）を復号化処理ではr_jとする。なお，符号化処理では時刻０での値ｖ（ｊ）は、$j=2^{i-1}$，ｉ＝１，２，…，ｍを除きｖ（ｊ）＝c_jとし、$j=2^{i-1}$，ｉ＝２，…，ｍではｖ（ｊ）＝c_{j+1}とし、ｖ（１）＝c_3とする。
＜規則４＞
ｊ＜ｊ’を満たす節点ｊと節点ｊ’との間に、$(i_1)_\delta$（又は$(i_1, \cdots, i_k)_\delta$）とラベル付けされた線がある場合には、時刻（δ－１）ｔ～δｔの間にｖ（ｊ）及びｖ（ｊ’）のＸＯＲ演算を行い、時刻δｔでの節点ｊの値を以下のＸＯＲ演算結果とする。
【数１５】

$$v(j)\oplus v(j')$$

この場合、ラベル$(i_1)_\delta$（又は$(i_1, \cdots, i_k)_\delta$）は時刻（δ－１）ｔ～δｔの間に行われたＸＯＲ演算結果がs_{i1}（又は複数のs_{i1}，…，s_{ik}）を求めるために用いられることを意味する。以後、ラベル$(i_1)_\delta$（又は$(i_1, \cdots, i_k)_\delta$）の表記において、$i_k$をラベルにおけるｋ番目のインデックス又は単にインデックスと称し、δをラベルにおける下付の時刻δと称する。
【００３６】
ｍ＝ｋ－１におけるs_1，s_2，…，s_{k-1}を求める演算回路を表す簡略表記を用いてｍ＝ｋにおけるs_1，s_2，…，s_kを求める回路を表す簡略表記を定める手順を次に示す。なお、簡略表記において、時刻（ｍ－１）ｔに節点ｊ，$j=2^{i-1}$，ｉ＝１，２，…，ｍが保持する値がs_iの値となる。
＜手順１＞
次の２つの図を上記規則１及び２に従って１つの図にまとめる。
（１図）ｍ＝ｋ－１での簡略表記で各節点$(b_{k-2}b_{k-3}\cdots b_0)_2$の先頭ビットに０を追加した節点$(0\,b_{k-2}b_{k-3}\cdots b_0)_2$に変更した図
（２図）ｍ＝ｋ－１での簡略表記で各節点$(b_{k-2}b_{k-3}\cdots b_0)_2$の先頭ビットに１を追加した節点$(1\,b_{k-2}b_{k-3}\cdots b_0)_2$に変更し、$s_1$，$s_2$，…，$s_{k-1}$を除いた図
＜手順２＞
手順１で作成された図において、節点2^{i-1}と節点$2^{i-1}+2^{k-1}$とを結ぶ線を追加し、その線にラベルとして$(i)_{k-1}$を付与する処理を、ｉ＝１，…，ｋ－１としてｋ－１回行う。
＜手順３＞
手順２で作成された図において、s_kと節点2^{k-1}を追加する。
＜手順４＞
手順３で作成された図において、次の（手順４－１）をα＝１，…，ｋ－１としてｋ－１回行う。
（手順４－１）

節点2^{k-1}と節点$j+2^{\alpha-1}$とを結ぶ線を追加する。追加した線のラベルを$(k)_{\alpha}$に更新する。αが２以上であれば、（手順４－２）によりラベルにインデックスｋの追加を再帰的に繰り返す。

（手順４－２）

更新したラベルにおける下付の時刻をβとする。更新したラベル$(k)_{\beta}$，β＝α又はｋを含むすべてのラベル$(\cdot, k)_{\beta}$を付与された線で結ばれた右側の節点について、その節点から出ている線のラベルにおける下付の時刻γがγ＜βを満たすすべてのラベル$(\cdot)_{\gamma}$に対して、インデックスｋを追加して$(\cdot, k)_{\gamma}$とする処理を再帰的に繰り返す。

【００３７】

図６（ｂ）は、以上説明した手順により簡略表記した例を示している。ｍ＝２の場合にs_1及びs_2を求める図６（ａ）と同様の演算回路を簡略表記している。図６（ｂ）に示すｍ＝２の場合の簡略表記図を用いて，ｍ＝３の場合の簡略表記図を作成する手順の例を図７に示す。図７では、線のラベルが１つ前の手順におけるラベルと同じ場合にはその表記を省略している。手順１では、０を追加した節点（００１）、（０１０）及び（０１１）に変更した図及び１を追加した節点（１０１）、（１１０）及び（１１１）に変更した図を１つの図にまとめて併記している。手順２では、節点（００１）と（１０１）とを結ぶ線を追加してラベル$(1)_2$を付与し、節点（０１０）と（１１０）とを結ぶ線を追加してラベル$(2)_2$を付与している。手順３では、s_3及び節点（１００）を追加し、手順４－１では、節点（１００）と（１０１）とを結ぶ線を追加してラベル$(3)_1$を付与し、節点（１００）と（１１０）とを結ぶ線を追加してラベル$(3)_2$を付与している。手順４－２では、節点（１１０）と（１１１）とを結ぶ線のラベルを$(2, 3)_1$に更新している。この場合、ＸＯＲ演算に起因する遅延時間は２ｔとなる。

【００３８】

図８は、ｍ＝２，３，４，５の場合の簡略表記図をまとめて示している。図８では、ｍ＝４，５の図では、インデックスの要素数が１個の場合にラベル表示を省略している。図８に示すｍ＝３の簡略表記図よりｍ＝３におけるs_iの算法を導き出すことができる。図９は、ｍ＝３におけるs_iの算法（図９（ａ））及び算法に用いる値ｊの一覧（図９（ｂ））を示す。値ｊは、以下の通り設定される。

【数１６】

$$j_k^{(i,w)},\ 1 \leqq i, w \leqq m,\ 1 \leqq k \leqq \binom{m}{w-1}$$

れる．ここで，wは$j_k^{(i,w)} = (b_{m-1}, b_{m-2} \cdots, , b_0)_2$において

$\sum_{n=0}^{m-1} b_n = w$であることを表す．

図１０は、ｍ＝５におけるs_iの算法（図１０（ａ））及び算法に用いる値ｊの一覧（図１０（ｂ））を示す。値ｊは、図９に示す例と同様に設定される。図９（ｂ）及び図１０（ｂ）では、同じｗ毎に区切りの横線を追加している。

【００３９】

図９に示す算法及びそれに用いる値ｊに基づいて図３（ｂ）に示す演算回路を得ることができる。すなわち、ｍ＝３の場合に手順１から手順４を適用することで、図３（ｂ）に示す演算回路が設定される。手順２及び手順４－１により，簡略表記において節点ｊの右からｋ，ｋ≦ｍ－１本の線が出ている場合には、各線の左側の節点ｊ’に対応する値ｊ’の小さい順（図８では、ｋ本の線の上からの順）に時刻０～ｔ，ｔ～２ｔ，…，（ｋ－１）ｔ～ｋｔ毎にその線に対応するＸＯＲ演算が行われる。一般に、２以上の自然数ｍの場合に手順１～手順４を適用することで、並列復号化処理におけるｓ＝（s_1，s_2，…，s_m）＝ｒH_m^Tを満たすs_i，ｉ＝１，２，…，ｍのある算法が定まり、（2^m-1，2^m-m-1）ハミング符号の並列復号化処理を行う演算回路を得ることができる。例えば、図８に示すｍ＝５の簡略表記に基づいて、図１０に示すｍ＝５におけるs_iの算法及び算法に用いる値が得られる。

【００４０】

なお、ｍにおける算法に用いる値ｊが設定されていれば、m_s≦ｍにおける値ｊを次の手順Ａにより求めることができる。

【数１７】

$$\text{(手順 A)}\quad m\text{ における }j_k^{(i,w)}\text{ の表から各 }w,\ 1\leqq w\leqq m_s\text{ 毎に}$$
$$j<2^{m_s}\text{ を満たす }j_k^{(i,w)},\ k=k_1,k_2,\cdots,k_{\binom{m_s-1}{w-1}}\text{ のみを残す.}$$
$$\text{更に，各 }w,\ 1\leqq w\leqq m_s\text{ 毎に }k_1<k_2<\cdots<k_{\binom{m_s-1}{w-1}}\text{ の順}$$
$$\text{番を保持し，}j_1^{(i,w)}\text{ から }j_{\binom{m_s-1}{w-1}}^{(i,w)}\text{ とする.}$$

手順Ａによりｍ＝５における値ｊのリスト（図１０（ｂ））からｍ＝２，３，４における値ｊのリストが得られる。そのため、ｍ＝ｋの簡略表記に対応する演算回路を用いて、ｍがｋより小さい演算回路として利用することが可能となり、演算回路の共通化や演算回路の変更を容易に行うことができ、演算回路のコストダウン及びフレキシビリティを高めることができる。

【００４１】

手順１～手順４のアルゴリズムにより得られる簡略表記においてラベルのインデックスの要素数がｋ，ｋ≧２個のものは冗長な対のＸＯＲ演算に対応しており、対応するｋ回のＸＯＲ演算をまとめて１回のＸＯＲ演算で行うことで、ｋ－１回のＸＯＲ演算を削除可能である。ｍにおけるs_i，ｉ＝１，２，…，ｍのある算法を与える手順１～手順４のアルゴリズムにより得られる削減可能なＸＯＲ演算の回数$R_m^{(a)}$は、図８に基づいて以下の通りとなる。

$R_2^{(a)}=0$
$R_3^{(a)}=1$
$R_4^{(a)}=6$
$R_5^{(a)}=23$

【００４２】

検査行列H_mにおいて各行における要素が１の個数は2^{m-1}個であり、s_iを独立して演算する場合に用いられるＸＯＲ演算の回数は$2^{m-1}-1$回である。一方、ｍにおけるs_i，ｉ＝１，２，…，ｍの算法を与える手順１～手順４では、手順４－１においてラベル$(m)_\alpha$，α＝１，２，…，ｍ－１が付与されて、インデックスｍが追加される回数はｍ－１回であり、手順４－２においてインデックスｍが追加される回数をS_mとすると、手順１～手順４における再帰構造から$S_2=0$であり、３≦ｎ≦ｍにおいて以下の漸化式を満たす。

$S_n=2S_{n-1}+n-2$

この漸化式より、ｍ≧２に対して、$S_m=2^{m-1}-m$であり、ｍにおける簡略表記のラベルのインデックスｍに基づく削除可能なＸＯＲ演算の回数はS_m回であることがわかる。また、ｍにおけるs_i，ｉ＝１，２，…，ｍの算法を与える手順１より、ｍにおける簡略表記から、ラベルのインデックスｍに基づくもの以外で削除可能なＸＯＲ演算の回数は$2R_{m-1}^{(a)}$回であることがわかる。そのため、ｍ≧３において

$R_m^{(a)}=2R_{m-1}^{(a)}+S_m=2R_{m-1}^{(a)}+2^{m-1}-m$・・・・（１２）

である。$R_2^{(a)}=R_2^{(s)}=0$、式（９）及び式（１２）により、ｍ≧２において

$R_m^{(a)}=R_m^{(s)}=(m-4)(2^{m-1}+1)+6$

が成り立つ。

【００４３】

以上のことより、（2^m-1，2^m-m-1）ハミング符号の並列復号化処理においてs_i，ｉ＝１，２，…，ｍを求めるＸＯＲ演算回数が式（１１）を満たすとともに遅延時間（ｍ－１）ｔである演算回路を構成することができる。（2^m-1，2^m-m-1）ハミング符号の並列符号化処理においてｍビットのパリティビット

【数１８】

$$c_{2^{i-1}}, i = 1, 2, \cdots, m$$

を求める演算回路の簡易表記は、ｍにおけるｓ$_{i}$，ｉ＝１，２，…，ｍの算法を与える手順１～手順４で得られる簡易表記において、ｓ$_{i}$を［数１８］に示すパリティビットに変更し、節点１及び節点３を結ぶ線並びにｉ＝１，２，…，ｍでの節点2^{i-1}及び節点2^{i-1}＋１を結ぶ線とそれら線のラベル（ｉ’）$_{1}$，ｉ’＝１，２，…，ｍを削除することで得られる。こうして得られた簡易表記から［数１８］に示すｍビットのパリティビットを求める演算回路が定まり，そのＸＯＲ演算回数は式（１０）を満たす。

【００４４】

手順１～手順４で定まる（$2^{m}-1$，$2^{m}-m-1$）ハミング符号の並列復号化処理でシンドロームを求める演算回路は、$2 \leqq m_{s} \leqq m$を満たす（$2^{ms}-1$，$2^{ms}-m_{s}-1$）ハミング符号の並列符号化処理でのパリティビットを求める演算回路や並列復号化処理でのシンドロームを求める演算回路に利用可能である。さらに、これらの演算回路は、$1 \leqq k_{s} \leqq 2^{ms}-m_{s}-1$，$2 \leqq m_{s} \leqq m$を満たす（$k_{s}+m_{s}$，$k_{s}$）ハミング符号の並列符号化処理でのパリティビットを求める演算回路や並列復号化処理でのシンドロームを求める演算回路に利用可能である。検査行列がH_{m}でなく一般の検査行列Ｈ場合には、列の置換に対して演算回路への入力の配置をＨの列に合わせて並び直せば、上述した手順を適用できる。そのため、検査行列がハミング符号の検査行列に分解できるＢＣＨ符号のシンドロームの演算にも適用可能である。

【００４５】

手順１～手順４で得られるｍにおける簡易表記は、ｍ個の集合のベン図と関連付けることができ、集合{１，２，…，ｍ}の空集合を除いたべき集合での包含の関係によるハッセ図においてある規則に従って線を除いた図と関連付けることもできる。

【符号の説明】

【００４６】

１・・・符号化・復号化装置、１０・・・符号化・復号化処理部、１１・・・符号化データ生成部、１２・・・誤り訂正処理部、２０・・・記憶部

【先行技術文献】

【特許文献】

【００４７】

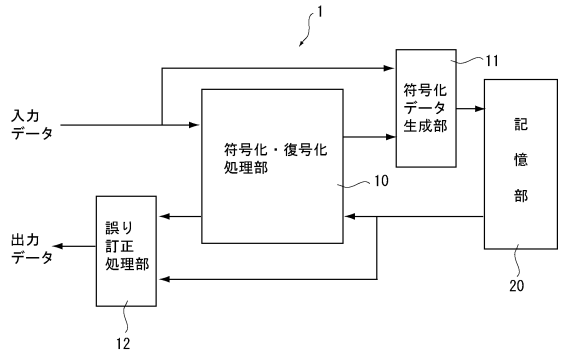
【特許文献１】米国特許第７，２９３，２２２号明細書

【非特許文献】

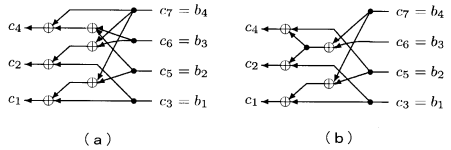
【００４８】

【非特許文献１】F. J. MacWilliams and N. J. A. Sloane, The Theory of Error-Correcting Codes, North-Holland, 1977

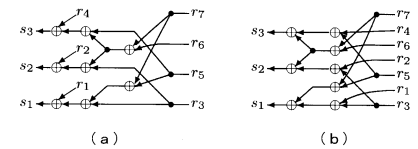
【図１】



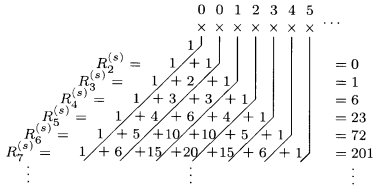
【図２】



【図３】



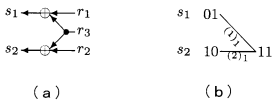
【図４】



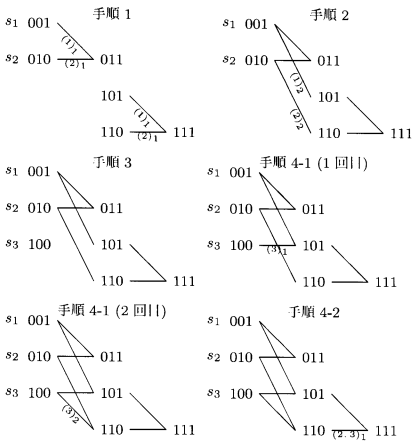
【図５】

m	2	3	4	5	6	7	8	9	10	11	12
$R_m^{(s)}$	0	1	6	23	72	201	522	1291	3084	7181	16398

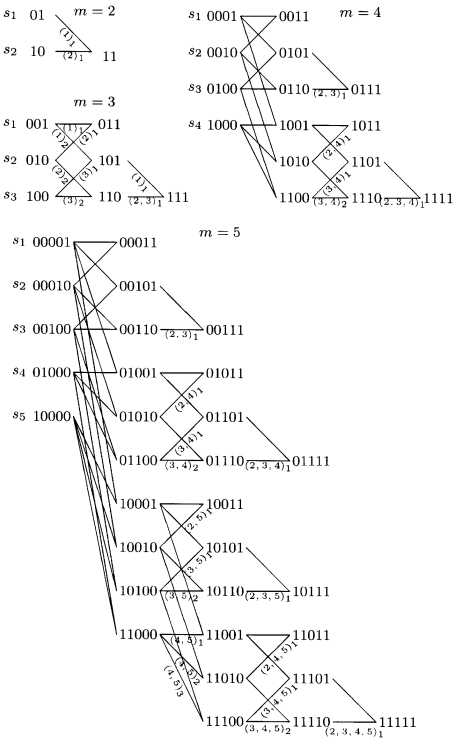
【図６】



【図７】



【図８】



【図９】

（ａ）

時刻 $=0\sim t$	時刻 $=t\sim 2t$
$x_{1,1}^{(i)}=r_{j_1^{(i,1)}}\oplus r_{j_1^{(i,2)}}$ $x_{1,2}^{(i)}=r_{j_2^{(i,2)}}\oplus r_{j_1^{(i,3)}}$	$s_i=x_{1,1}^{(i)}\oplus x_{1,2}^{(i)}$

（ｂ）

$i=1$	$i=2$	$i=3$
$j_1^{(1,1)}=1$	$j_1^{(2,1)}=2$	$j_1^{(3,1)}=4$
$j_1^{(1,2)}=3$	$j_1^{(2,2)}=3$	$j_1^{(3,2)}=5$
$j_2^{(1,2)}=5$	$j_2^{(2,2)}=6$	$j_2^{(3,2)}=6$
$j_1^{(1,3)}=7$	$j_1^{(2,3)}=7$	$j_1^{(3,3)}=7$

【図１０】

（ａ）

時刻 $=0\sim t$	時刻 $=t\sim 2t$	時刻 $=2t\sim 3t$
$x_{1,1}^{(i)}=r_{j_1^{(i,1)}}\oplus r_{j_1^{(i,2)}}$	$x_{2,1}^{(i)}=x_{1,1}^{(i)}\oplus x_{1,2}^{(i)}$	$x_{3,1}^{(i)}=x_{2,1}^{(i)}\oplus x_{2,2}^{(i)}$
$x_{1,2}^{(i)}=r_{j_2^{(i,2)}}\oplus r_{j_1^{(i,3)}}$	$x_{2,2}^{(i)}=x_{1,3}^{(i)}\oplus x_{1,4}^{(i)}$	$x_{3,2}^{(i)}=x_{2,3}^{(i)}\oplus x_{2,4}^{(i)}$
$x_{1,3}^{(i)}=r_{j_3^{(i,2)}}\oplus r_{j_2^{(i,3)}}$	$x_{2,3}^{(i)}=x_{1,5}^{(i)}\oplus x_{1,6}^{(i)}$	
$x_{1,4}^{(i)}=r_{j_4^{(i,2)}}\oplus r_{j_3^{(i,3)}}$	$x_{2,4}^{(i)}=x_{1,7}^{(i)}\oplus x_{1,8}^{(i)}$	
$x_{1,5}^{(i)}=r_{j_4^{(i,3)}}\oplus r_{j_1^{(i,4)}}$		
$x_{1,6}^{(i)}=r_{j_5^{(i,3)}}\oplus r_{j_2^{(i,4)}}$		
$x_{1,7}^{(i)}=r_{j_6^{(i,3)}}\oplus r_{j_3^{(i,4)}}$		時刻 $=3t\sim 4t$
$x_{1,8}^{(i)}=r_{j_4^{(i,4)}}\oplus r_{j_1^{(i,5)}}$		$s_i=x_{3,1}^{(i)}\oplus x_{3,2}^{(i)}$

（ｂ）

$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
$j_1^{(1,1)}=1$	$j_1^{(2,1)}=2$	$j_1^{(3,1)}=4$	$j_1^{(4,1)}=8$	$j_1^{(5,1)}=16$
$j_1^{(1,2)}=3$	$j_1^{(2,2)}=3$	$j_1^{(3,2)}=5$	$j_1^{(4,2)}=9$	$j_1^{(5,2)}=17$
$j_2^{(1,2)}=5$	$j_2^{(2,2)}=6$	$j_2^{(3,2)}=6$	$j_2^{(4,2)}=10$	$j_2^{(5,2)}=18$
$j_3^{(1,2)}=9$	$j_3^{(2,2)}=10$	$j_3^{(3,2)}=12$	$j_3^{(4,2)}=12$	$j_3^{(5,2)}=20$
$j_4^{(1,2)}=17$	$j_4^{(2,2)}=18$	$j_4^{(3,2)}=20$	$j_4^{(4,2)}=24$	$j_4^{(5,2)}=24$
$j_1^{(1,3)}=7$	$j_1^{(2,3)}=7$	$j_1^{(3,3)}=7$	$j_1^{(4,3)}=11$	$j_1^{(5,3)}=19$
$j_2^{(1,3)}=11$	$j_2^{(2,3)}=11$	$j_2^{(3,3)}=13$	$j_2^{(4,3)}=13$	$j_2^{(5,3)}=21$
$j_3^{(1,3)}=19$	$j_3^{(2,3)}=19$	$j_3^{(3,3)}=21$	$j_3^{(4,3)}=25$	$j_3^{(5,3)}=25$
$j_4^{(1,3)}=13$	$j_4^{(2,3)}=14$	$j_4^{(3,3)}=14$	$j_4^{(4,3)}=14$	$j_4^{(5,3)}=22$
$j_5^{(1,3)}=21$	$j_5^{(2,3)}=22$	$j_5^{(3,3)}=22$	$j_5^{(4,3)}=26$	$j_5^{(5,3)}=26$
$j_6^{(1,3)}=25$	$j_6^{(2,3)}=26$	$j_6^{(3,3)}=28$	$j_6^{(4,3)}=28$	$j_6^{(5,3)}=28$
$j_1^{(1,4)}=15$	$j_1^{(2,4)}=15$	$j_1^{(3,4)}=15$	$j_1^{(4,4)}=15$	$j_1^{(5,4)}=23$
$j_2^{(1,4)}=23$	$j_2^{(2,4)}=23$	$j_2^{(3,4)}=23$	$j_2^{(4,4)}=27$	$j_2^{(5,4)}=27$
$j_3^{(1,4)}=27$	$j_3^{(2,4)}=27$	$j_3^{(3,4)}=29$	$j_3^{(4,4)}=29$	$j_3^{(5,4)}=29$
$j_4^{(1,4)}=29$	$j_4^{(2,4)}=30$	$j_4^{(3,4)}=30$	$j_4^{(4,4)}=30$	$j_4^{(5,4)}=30$
$j_1^{(1,5)}=31$	$j_1^{(2,5)}=31$	$j_1^{(3,5)}=31$	$j_1^{(4,5)}=31$	$j_1^{(5,5)}=31$

フロントページの続き

審査官　岡　裕之

(56)参考文献　米国特許出願公開第２００７／０２７７０８５（ＵＳ，Ａ１）
大島　怜也　他，ハミング符号の並列符号器と並列復号器におけるXOR演算回数の最適化，電子情報通信学会技術研究報告，２０１２年　9月２０日，Vol.112，No.215，pp.25-30，IT2012-35

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０３Ｍ　１３／１９
ＩＥＥＥ　Ｘｐｌｏｒｅ
ＣｉＮｉｉ