

(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2014-64065
(P2014-64065A)

(43) 公開日 平成26年4月10日(2014.4.10)

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

F I
H03M 13/19

テーマコード(参考)
5J065

審査請求 未請求 請求項の数 5 O L (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)

(71) 出願人 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
Fターム(参考) 5J065 AD05

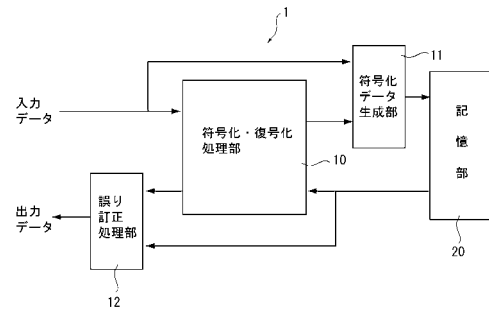
(54) 【発明の名称】 符号化・復号化装置及び演算回路設定方法

(57) 【要約】

【課題】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。

【解決手段】符号化・復号化処理部10は、所定の規則性を有する検査行列に基づいて入力データに対応するハミング符号の符号語を生成する符号化処理を行うとともに検査行列に基づいてハミング符号の受信語のエラー位置を検出する復号化処理を行い、ハミング符号が符号長 $2^m - 1$ 及び情報ビット数 $2^m - m - 1$ の2元ハミング符号である場合に、XOR演算回数を $(2^{m+1} - 2m - 2)$ 回処理して復号化処理する演算回路を備えている。

【選択図】 図1



【特許請求の範囲】

【請求項 1】

所定の規則性を有する検査行列に基づいて入力データに対応するハミング符号の符号語を生成する符号化手段と、前記検査行列に基づいてハミング符号の受信語のエラー位置を検出する復号化手段とを備える符号化・復号化装置であって、前記復号化手段は、ハミング符号が符号長 $2^m - 1$ 及び情報ビット数 $2^m - m - 1$ の 2 元ハミング符号である場合に、XOR 演算回数を $(2^{m+1} - 2m - 2)$ 回処理して復号化処理する演算回路を備えている符号化・復号化装置。

【請求項 2】

前記符号化手段は、ハミング符号が符号長 $2^m - 1$ 及び情報ビット数 $2^m - m - 1$ の 2 元ハミング符号である場合に、XOR 演算回数を $(2^{m+1} - 3m - 2)$ 回処理して符号化処理する演算回路を備えている請求項 1 に記載の符号化・復号化装置。

10

【請求項 3】

前記復号化手段又は前記符号化手段は、1 回の XOR 演算に起因する遅延時間を t とした場合に、前記演算回路の遅延時間が $(m - 1)t$ に設定されている請求項 1 又は 2 に記載の符号化・復号化装置。

【請求項 4】

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 , $j = 1, 2, \dots, 2^m - 1$ 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δροーム s_i , $i = 1, 2, \dots, m$ 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。

20

< 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 $b_{m-1} b_{m-2} \dots b_0$, $b_i \in \{0, 1\}$ について以下の値を求め、

【数 1 4】

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

30

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、 $b_{m-1} b_{m-2} \dots b_0$ 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、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。

40

【請求項 5】

$m = k - 1$ (k は、3 以上の自然数) におけるシンδροームを生成する演算回路に基づいて $m = k$ におけるシンδροームを生成する演算回路を設定する請求項 4 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。

50

【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】

10

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）を組み合わせて構成される。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。

20

【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

30

$$h_j^T = (h_{1,j}, h_{2,j}, \dots, h_{i,j}, \dots, h_{m,j}) \quad F_2^m$$

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）。

40

【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】

50

【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演算を組み合わ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。

10

【課題を解決するための手段】

【0009】

本発明に係る符号化・復号化装置は、所定の規則性を有する検査行列に基づいて入力データに対応するハミング符号の符号語を生成する符号化手段と、前記検査行列に基づいてハミング符号の受信語のエラー位置を検出する復号化手段とを備える符号化・復号化装置であって、前記復号化手段は、ハミング符号が符号長 $2^m - 1$ 及び情報ビット数 $2^m - m - 1$ の2元ハミング符号である場合に、XOR演算回数を $(2^{m+1} - 2m - 2)$ 回処理して復号化処理する演算回路を備えている。さらに、前記符号化手段は、ハミング符号が符号長 $2^m - 1$ 及び情報ビット数 $2^m - m - 1$ の2元ハミング符号である場合に、XOR演算回数を $(2^{m+1} - 3m - 2)$ 回処理して符号化処理する演算回路を備えている。さらに、前記復号化手段又は前記符号化手段は、1回のXOR演算に起因する遅延時間を t とした場合に、前記演算回路の遅延時間が $(m - 1)t$ に設定されている。

20

【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 、 $j = 1, 2, \dots, 2^m - 1$ 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δροーム s_i 、 $i = 1, 2, \dots, m$ 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。

30

<規則2>

j の二進表記である $b_{m-1}b_{m-2}\dots b_0$ 、 $b_i \in \{0, 1\}$ 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、 $b_{m-1}b_{m-2}\dots b_0$ 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。

40

<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δροームを生成する演算回路に基づいて $m = k$ におけるシンδροームを生成する

50

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。

10

【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。

20

【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。

30

【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δροームを生成する。生成されたシンδρο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。

40

【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

50

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。

$$\begin{aligned} c &= (c_1, c_2, \dots, c_n) \quad F_2^n, n = 2^m - 1 \\ b &= (b_1, b_2, \dots, b_k) \quad F_2^k, k = 2^m - m - 1 \\ r &= (r_1, r_2, \dots, r_n) \quad F_2^n \end{aligned}$$

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。

$$C = (C_1, C_2, \dots, C_n) \quad F_2^n$$

【 0 0 1 9 】

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、

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

シンドローム s を以下の通り定める。

$$s = r H_m^T$$

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 】

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

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。

【 0 0 2 0 】

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 】

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

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。

$$\begin{aligned} b &= (b_1, b_2, b_3, b_4) \quad F_2^4 \\ c &= (c_1, c_2, c_3, c_4, c_5, c_6, c_7) \quad F_2^7 \end{aligned}$$

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。

10

20

30

40

【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。

10

【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_j の集合を葉節点の集合とし、これらの節点を樹枝状に接続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。

20

【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。

30

【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δροーム s 以下の通りになる。

40

$$r = (r_1, r_2, r_3, r_4, r_5, r_6, r_7) \quad F_2^7$$

$$s = (s_3, s_2, s_1) \quad F_2^3$$

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。

10

【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 $d t$ とすれば、 d は以下の通り定めることができる。

20

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

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】

30

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。

40

【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、 $m = 2, 3, \dots$ に対して以下の式を満たす。

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

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

50

【 0 0 3 0 】

そして、 R_m については、以下の式を満たすようになる。

$$R_m = (m - 4)(2^{m-1} + 1) + 6 \cdot \dots \cdot (3)$$

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j₁及びj₂を1 ≤ j₁ < j₂ ≤ 2^m - 1とし、あるk個の行番号を表すi₁, i₂, ..., i_kを1 ≤ i₁ < i₂ < ... < i_k ≤ mとする。検査行列Hのj₁列目とj₂列目において、i₁, i₂, ..., 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。

【 0 0 3 1 】

10

(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り、「1」の要素がk₁個であるj₁列目と「1」の要素がk₂個であるj₂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、k₁ = k₂である場合にはmin{k₁, k₂} - 1回までXOR演算の回数を削除可能であり、k₁ ≠ k₂である場合にはk₁ - 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 】

$$R_m \leq \sum_{k=2}^m (k-2) \binom{m}{k} \dots \dots \dots (4)$$

20

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

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

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

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。

30

$$2^{m+1} - 3m - 2 = N_{m,e} \dots \dots (6)$$

$$2^{m+1} - 2m - 2 = N_{m,d} \dots \dots (7)$$

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。

$$R_m^{(s)} = (m - 4)(2^{m-1} + 1) + 6 \dots \dots (8)$$

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 = 3では、以下の漸化式を満たす。

$$R_m^{(s)} = 2R_{m-1}^{(s)} + 2^{m-1} - m \dots \dots (9)$$

さらに、 $R_m^{(s)}$ を係数とする以下の母関数

【 数 9 】

40

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

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 1 0 】

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

50

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

【 0 0 3 2 】

$R_{n+3}^{(s)}$, $n = 0, 1, 2, \dots$ 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 $A_n = \{1, 2, \dots, n + 2\}$ 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。

10

【 0 0 3 3 】

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 1 1】

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

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。

$$B_0; 1$$

$$B_1; 1 \times 3 + 3 \times 1 = 6$$

$$B_2; 1 \times 7 + 3 \times 3 + 7 \times 1 = 23$$

20

【 0 0 3 4 】

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。

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

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

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 を使い、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 $m = 2, 3, \dots$ を考える。シンドローム s における s_i , $i = 1, 2, \dots$, 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。

30

【数 1 2】

$$s_1 = r_1 \oplus r_3$$

$$s_2 = r_2 \oplus r_3$$

【 0 0 3 5 】

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。

40

【数 1 3】

$$j = \sum_{k=0}^{m-1} b_k 2^k$$

そして、設定された j に関する二進数表記 $(b_{m-1} b_{m-2} \dots b_0)_2$ を用いて r_j の代わりに単に $b_{m-1} b_{m-2} \dots b_0$ 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。

50

< 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 >

$b_{m-1} b_{m-2} \dots b_0$, $b_i \in \{0, 1\}$ について以下の値を求め、

【数 1 4】

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

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 $j = (b_{m-1} b_{m-2} \dots b_0)_2$ の小さい値から大きい値の順に上から下に配置する。

10

< 規則 3 >

規則 2 で作成した配置図において、 $b_{m-1} b_{m-2} \dots b_0$ を値 0 か値 1 の 1 ビットを保持する節点とし、節点 $(b_{m-1} b_{m-2} \dots b_0)_2$ 又は節点 j , $j = (b_{m-1} b_{m-2} \dots b_0)_2$ 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、 $j = 2^{i-1}$, $i = 1, 2, \dots, m$ を除き $v(j) = c_j$ とし、 $j = 2^{i-1}$, $i = 2, \dots, m$ 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) (又は (i_1, \dots, i_k)) とラベル付けされた線がある場合には、時刻 $(-1)t \sim t$ の間に $v(j)$ 及び $v(j')$ の XOR 演算を行い、時刻 t での節点 j の値を以下の XOR 演算結果とする。

20

【数 1 5】

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

この場合、ラベル (i_1) (又は (i_1, \dots, i_k)) は時刻 $(-1)t \sim t$ の間に行われた XOR 演算結果が s_{i_1} (又は複数の s_{i_1}, \dots, s_{i_k}) 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) (又は (i_1, \dots, i_k)) 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、 i_k をラベルにおける下付の時刻 i_k と称する。

【0 0 3 6】

30

$m = k - 1$ における s_1, s_2, \dots, 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, \dots, 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 , $j = 2^{i-1}$, $i = 1, 2, \dots, m$ 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 $(b_{k-2} b_{k-3} \dots b_0)_2$ の先頭ビットに 0 を追加した節点 $(0 b_{k-2} b_{k-3} \dots b_0)_2$ に変更した図

(2 図) $m = k - 1$ での簡略表記で各節点 $(b_{k-2} b_{k-3} \dots b_0)_2$ の先頭ビットに 1 を追加した節点 $(1 b_{k-2} b_{k-3} \dots b_0)_2$ に変更し、 s_1, s_2, \dots, s_{k-1} を除いた図

40

< 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、 $i = 1, \dots, k - 1$ 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 $i = 1, \dots, k - 1$ として $k - 1$ 回行う。

(手順 4 - 1)

50

節点 2^{k-1} と節点 $j + 2^{k-1}$ とを結ぶ線を追加する。追加した線のラベルを (k) に更新する。 k 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 t とする。更新したラベル (k) , $t = t$ 又は k を含むすべてのラベル (\cdot, k) 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 t が t を満たすすべてのラベル (\cdot) に対して、インデックス k を追加して (\cdot, k) 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。

10

20

【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】

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

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

$$\sum_{n=0}^{m-1} b_n = w \text{ であることを表す。}$$

30

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 $0 \sim t$ 、 $t \sim 2t$ 、 \dots 、 $(k - 1)t \sim kt$ 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 $s = (s_1, s_2, \dots, s_m) = r H_m^T$ を満たす s_i 、 $i = 1, 2, \dots, m$ 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。

40

【0040】

50

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】

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

10

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 $k, k \geq 2$ 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 $s_i, i = 1, 2, \dots, m$ のある算法を与える手順 1 ~ 手順 4 のアル

20

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。

$$\begin{aligned} R_2^{(a)} &= 0 \\ R_3^{(a)} &= 1 \\ R_4^{(a)} &= 6 \\ R_5^{(a)} &= 23 \end{aligned}$$

【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 $s_i, i = 1, 2, \dots, m$ 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) $i = 1, 2, \dots, m - 1$ 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、 $3 \leq n \leq m$ において以下の漸化式を満たす。

30

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

この漸化式より、 $m \geq 2$ 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 $s_i, i = 1, 2, \dots, m$ 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 $2 R_{m-1}^{(a)}$ 回であることがわかる。そのため、 $m \geq 3$ において

40

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

である。 $R_2^{(a)} = R_2^{(s)} = 0$ 、式 (9) 及び式 (12) により、 $m \geq 2$ 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 $s_i, i = 1, 2, \dots, m$ 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】

$$C_{2^{i-1}, i} = 1, 2, \dots, m$$

を求める演算回路の簡易表記は、 m における s_i , $i = 1, 2, \dots, m$ の算術を与える手順 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 $i = 1, 2, \dots, m$ 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') , $i' = 1, 2, \dots, m$ 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。

10

【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、 $2^{m_s} - m$ を満たす $(2^{m_s} - 1, 2^{m_s} - m_s - 1)$ 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、 $1 - k_s$

$2^{m_s} - m_s - 1, 2^{m_s} - m$ 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。

20

【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 $\{1, 2, \dots, m\}$ 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】

30

【特許文献】

【0047】

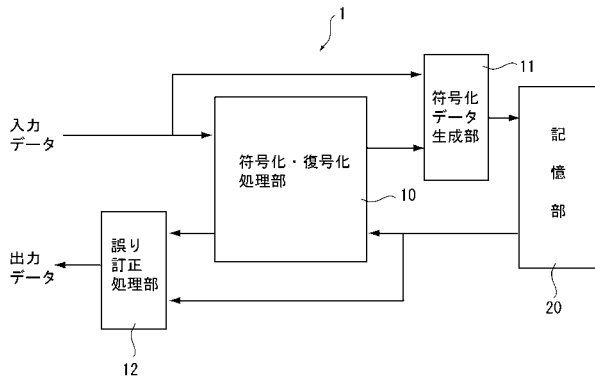
【特許文献 1】米国特許第 7, 293, 222 号明細書

【非特許文献】

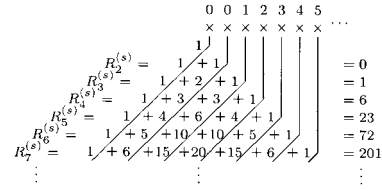
【0048】

【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 】



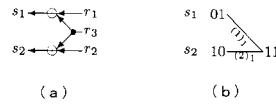
【 図 4 】



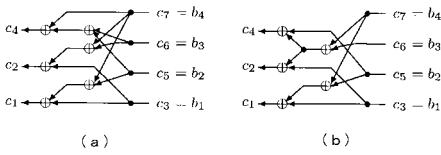
【 図 5 】

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

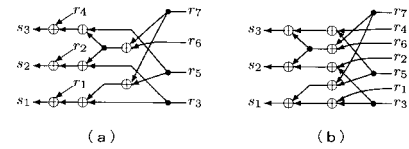
【 図 6 】



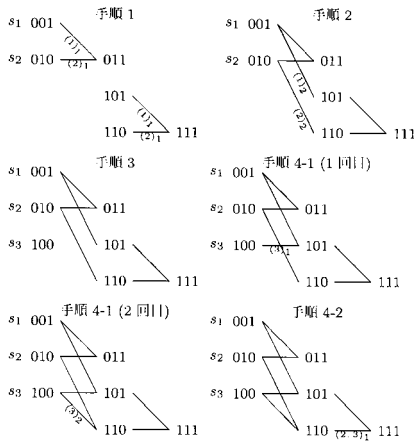
【 図 2 】



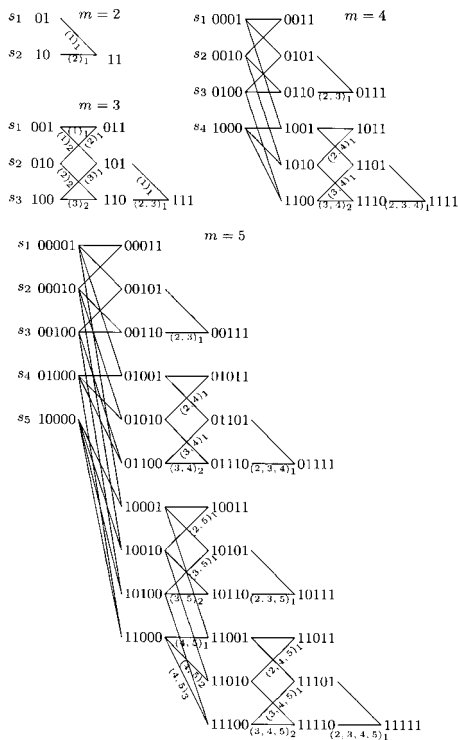
【 図 3 】



【 図 7 】



【 図 8 】



【 図 9 】

	時刻 = 0 ~ t	時刻 = t ~ 2t																
(a)	$x_{1,1}^{(i)} = r_{j_1^{(i,1)}} \oplus r_{j_1^{(i,2)}}$	$s_i = x_{1,1}^{(i)} \oplus x_{1,2}^{(i)}$																
	$x_{1,2}^{(i)} = r_{j_2^{(i,2)}} \oplus r_{j_1^{(i,3)}}$																	
(b)	<table border="1"> <tr> <td>i = 1</td> <td>i = 2</td> <td>i = 3</td> </tr> <tr> <td>$j_1^{(1,1)} = 1$</td> <td>$j_1^{(2,1)} = 2$</td> <td>$j_1^{(3,1)} = 4$</td> </tr> <tr> <td>$j_1^{(1,2)} = 3$</td> <td>$j_1^{(2,2)} = 3$</td> <td>$j_1^{(3,2)} = 5$</td> </tr> <tr> <td>$j_2^{(1,2)} = 5$</td> <td>$j_2^{(2,2)} = 6$</td> <td>$j_2^{(3,2)} = 6$</td> </tr> <tr> <td>$j_1^{(1,3)} = 7$</td> <td>$j_1^{(2,3)} = 7$</td> <td>$j_1^{(3,3)} = 7$</td> </tr> </table>	i = 1	i = 2	i = 3	$j_1^{(1,1)} = 1$	$j_1^{(2,1)} = 2$	$j_1^{(3,1)} = 4$	$j_1^{(1,2)} = 3$	$j_1^{(2,2)} = 3$	$j_1^{(3,2)} = 5$	$j_2^{(1,2)} = 5$	$j_2^{(2,2)} = 6$	$j_2^{(3,2)} = 6$	$j_1^{(1,3)} = 7$	$j_1^{(2,3)} = 7$	$j_1^{(3,3)} = 7$		
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 】

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

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