

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4713039号  
(P4713039)

(45) 発行日 平成23年6月29日 (2011. 6. 29)

(24) 登録日 平成23年4月1日 (2011. 4. 1)

(51) Int. Cl.

F I

H O 3 M 13/29 (2006. 01)

H O 3 M 13/29

G O 6 F 11/10 (2006. 01)

G O 6 F 11/10 3 3 O N

請求項の数 8 (全 45 頁)

(21) 出願番号 特願2001-524245 (P2001-524245)  
 (86) (22) 出願日 平成12年9月4日 (2000. 9. 4)  
 (65) 公表番号 特表2003-509943 (P2003-509943A)  
 (43) 公表日 平成15年3月11日 (2003. 3. 11)  
 (86) 国際出願番号 PCT/EP2000/008631  
 (87) 国際公開番号 W02001/020787  
 (87) 国際公開日 平成13年3月22日 (2001. 3. 22)  
 審査請求日 平成19年8月14日 (2007. 8. 14)  
 (31) 優先権主張番号 99117945. 8  
 (32) 優先日 平成11年9月15日 (1999. 9. 15)  
 (33) 優先権主張国 欧州特許庁 (EP)

(73) 特許権者 598036300  
 テレフオンアクチーボラゲット エル エ  
 ム エリクソン (パブル)  
 スウェーデン国 ストックホルム エスー  
 1 6 4 8 3  
 (74) 代理人 100076428  
 弁理士 大塚 康德  
 (74) 代理人 100112508  
 弁理士 高柳 司郎  
 (74) 代理人 100115071  
 弁理士 大塚 康弘  
 (74) 代理人 100116894  
 弁理士 木村 秀二

最終頁に続く

(54) 【発明の名称】 並列ターボ符号化の実装

(57) 【特許請求の範囲】

【請求項 1】

$$x_1(t) = I(t-1) \oplus \alpha_1 \cdot x_1(t-1) \oplus \alpha_2 \cdot x_2(t-1) \oplus \dots \oplus \alpha_N \cdot x_N(t-1)$$

$$\alpha_i \in \{0, 1\}$$

$$x_2(t) = x_1(t-1)$$

$$\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array}$$

$$x_N(t) = x_{N-1}(t-1)$$

10

$$Q_j(t) = \beta_{j0} \cdot I(t-1) \oplus$$

$$x_1(t-1) \cdot [\beta_{j0} \cdot \alpha_1 \oplus \beta_{j1}] \oplus$$

$$\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array}$$

$$x_N(t-1) \cdot [\beta_{j0} \cdot \alpha_N \oplus \beta_{jN}]$$

$$\beta_{ij} \in \{0, 1\}$$

20

$$j \in [1, \dots, N]$$

に応じたシリアルターボ符号化ブロック表現から、並列ターボ符号化を行う方法であって、

a) 並列度 n において、

$$x_1(t-1) = I(t-2) \oplus \alpha_1 \cdot x_1(t-2) \oplus$$

$$\alpha_2 \cdot x_2(t-2) \oplus \dots \oplus$$

$$\alpha_N \cdot x_N(t-2)$$

$$(2.x_1.1)$$

30

$$\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array}$$

$$x_1(t-(n-1)) = I(t-n) \oplus \alpha_1 \cdot x_1(t-n) \oplus$$

$$\alpha_2 \cdot x_2(t-n) \oplus \dots \oplus$$

$$\alpha_N \cdot x_N(t-n)$$

$$(2.x_1.n-1)$$

40

に対応する第 1 の内部状態について時間インデックス置換を行う工程と、

b)

$$X_i(t-1) = X_{i-1}(t-2) \quad (2.xi.1)$$

$$\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array}$$

$$X_i(t(n-1)) = X_{i-1}(t-n) \quad (2.xi.n-1)$$

に応じた残りの内部状態 ( $i=2, \dots, N$ ) について時間インデックス置換を行う工程と、

50

c)

$$\begin{aligned}
 Q_j(t-i) &= \beta_{j0} \cdot I(t-(i+1)) \oplus \\
 &\quad x_1(t-(i+1)) \cdot [\beta_{j0} \cdot \alpha_1 \oplus \beta_{j1}] \oplus \\
 &\quad x_N(t-(i+1)) \cdot [\beta_{j0} \cdot \alpha_N \oplus \beta_{jN}] \\
 &\quad i \in [1, \dots, n-1] \\
 &\quad (2.Q.i)
 \end{aligned}$$

10

に応じた出力信号について、

$$Q_j(t) = Q_{j0}(p)$$

$$Q_j(t-1) = Q_{j1}(p)$$

.

.

.

$$Q_j(t-(n-1)) = Q_{jn-1}(p)$$

$$j \in [1, \dots, M]$$

の並列出力ベクトルを導出するために時間インデックス置換を行う工程と、

d) 内部状態 $x_k(t)$  ( $k=1, \dots, N$ )のそれぞれについて内部状態置換処理を行う工程であって、

20

d 1) 最大時間インデックス要素を前記内部状態 $x_k(t)$ について $t_{\max}=t-1$ に設定する工程と、

d 2) 前記最大時間インデックス $t_{\max}$ を有する内部状態について、前記内部状態 $x_k(t)$ 表現をスキャンする工程と、

d 3) 式(2)を利用する状態変数置換工程を通して、前記内部状態 $x_k(t)$ の表現において $t_{\max}$ から $t_{\max}-1$ へ後方時間インデックス遷移を実行する工程と、

d 4)  $t_{\max}$ を1だけ減ずると共に、 $t_{\max}$ が $t-n$ よりも大きい間において前記ステップd 2)からd 4)を繰り返す工程とを備える工程と、

e) 内部状態置換処理を、各並列出力ベクトル $Q_j(t)$  ( $j=i, \dots, M$ )の各要素 $Q_j(t-1)$  ( $i=0, \dots, n-2$ )について実行する工程であって、

30

e 1) 前記最大時間インデックス要素を、考慮された並列出力ベクトル $Q_j(t)$ におけるベクトル要素 $Q_j(t-1)$ について、 $t_{\max}=t-i-1$ に設定する工程と、

e 2) 前記最大時間インデックスを有する内部状態について、前記ベクトル要素 $Q_j(t-i)$ の表現をスキャンする工程と、

e 3) 式(2)を利用する状態変数置換工程を通して、前記ベクトル要素 $Q_j(t-i)$ の表現において $t_{\max}$ から $t_{\max}-1$ へ後方時間インデックス遷移を実行する工程と、

e 4)  $t_{\max}$ を1だけ減ずると共に、 $t_{\max}$ が $t-n$ よりも大きい間において前記ステップe 2)からe 4)を繰り返す工程とを備える工程とを

備えることを特徴とする方法。

40

#### 【請求項2】

並列度 $n$ の並列ターボ符号化ブロックであって、

a) 入力信号 $I(t)$ の $n$ サンプル( $I(t-1), \dots, I(t-n)$ )を、前記並列ターボ符号化ブロックに格納するための手段( $I_0, \dots, I_7$ )と、

b) 前記並列ターボ符号化ブロックの少なくとも1の出力信号 $Q_j(t)$ の $n$ サンプル( $Q_j(t), \dots, Q_j(t-(n-1))$ )を格納するための、少なくとも1の手段( $Q_0, \dots, Q_7$ )と、

c) 遅延ユニット( $X_1, \dots, X_N$ )の列を備え、並列処理に適應され、前記列の少なくとも2つの遅延ユニット $X_1$ 、 $\dots$ 、 $X_N$ が、前記入力信号 $I(t)$ の $n$ 個のサンプル $I(t-1), \dots, I(t-n)$ のサブセットを直接受け、前記並列ターボ符号化ブロックの少なくとも1つの遅延ユニット $X_1$ 、 $\dots$ 、 $X_N$ の出力信号が、前記並列ターボ符号化ブ

50

ロックの少なくとも2つの遅延ユニットへ供給されるように、前記入力信号  $I(t)$  の  $n$  個のサンプルの並列処理に適応されたターボ符号化手段とを備え、

$$x_1(t) = I(t-1) \oplus \alpha_1 \cdot x_1(t-1) \oplus \alpha_2 \cdot x_2(t-1) \oplus \dots \oplus \alpha_N \cdot x_N(t-1)$$

$$\alpha_i \in \{0, 1\}$$

$$x_2(t) = x_1(t-1)$$

.

.

$$x_N(t) = x_{N-1}(t-1)$$

10

$$Q_j(t) = \beta_{j0} \cdot I(t-1) \oplus$$

$$x_1(t-1) \cdot [\beta_{j0} \cdot \alpha_1 \oplus \beta_{j1}] \oplus$$

$$\cdot$$

$$\cdot$$

$$x_N(t-1) \cdot [\beta_{j0} \cdot \alpha_N \oplus \beta_{jN}]$$

$$\beta_{ji} \in \{0, 1\}$$

20

$$j \in [1, \dots, M]$$

に対応するシリアルターボ符号化ブロックから導出可能な構造を、前記ターボ符号化手段が請求項1に記載の並列化されたターボ符号化を行なう際に有することを特徴とする並列ターボ符号化ブロック。

#### 【請求項3】

前記並列度が2であり、 $N = 3$ 、 $M = 1$ 、 $\alpha = [0, 1, 1]$ 、 $\beta = [1, 1, 0, 1]$ である場合に、前記ターボ符号化手段の構造が

$$Q_0(p) = I_0(p-1) \oplus I_1(p-1) \oplus x_1(p-1) \oplus x_2(p-1) \oplus x_3(p-1)$$

30

$$Q_1(p) = I_1(p-1) \oplus x_1(p-1) \oplus x_2(p-1),$$

ここで

$$x_1(p) = I_0(p-1) \oplus x_1(p-1) \oplus x_2(p-1),$$

$$x_2(p) = I_1(p-1) \oplus x_2(p-1) \oplus x_3(p-1), \text{ and}$$

40

$$x_3(p) = x_1(p-1).$$

であることを特徴とする請求項2に記載の並列ターボ符号化ブロック。

#### 【請求項4】

前記並列度が4であり、 $N = 3$ 、 $M = 1$ 、 $\alpha = [0, 1, 1]$ 、 $\beta = [1, 1, 0, 1]$ である場合に、前記ターボ符号化手段の構造が

$$Q0(p) = I0(p-1) \oplus I1(p-1) \oplus I2(p-1) \oplus I3(p-1) \oplus x3(p-1)$$

$$Q1(p) = I1(p-1) \oplus I2(p-1) \oplus I3(p-1) \oplus x1(p-1) \oplus x3(p-1)$$

$$Q2(p) = I2(p-1) \oplus I3(p-1) \oplus x1(p-1) \oplus x2(p-1) \oplus x3(p-1)$$

$$Q3(p) = I3(p-1) \oplus x1(p-1) \oplus x2(p-1),$$

10

ここで

$$x1(p) = I0(p-1) \oplus I2(p-1) \oplus I3(p-1) \oplus$$

$$x1(p-1) \oplus x3(p-1),$$

$$x2(p) = I1(p-1) \oplus I3(p-1) \oplus$$

$$x1(p-1) \oplus x2(p-1) \oplus x3(p-1), \text{ and}$$

20

$$x3(p) = I2(p-1) \oplus x1(p-1) \oplus x2(p-1).$$

であることを特徴とする請求項 2 に記載の並列ターボ符号化ブロック。

【請求項 5】

前記並列度が 8 であり、 $N = 3$ 、 $M = 1$ 、 $\mathbf{a} = [0, 1, 1]$ 、 $\mathbf{b} = [1, 1, 0, 1]$ である場合に、前記ターボ符号化手段の構造が

$$Q0(p) = I0(p-1) \oplus I1(p-1) \oplus I2(p-1) \oplus I3(p-1) \oplus$$

$$I6(p-1) \oplus x1(p-1) \oplus x2(p-1)$$

30

$$Q1(p) = I1(p-1) \oplus I2(p-1) \oplus I3(p-1) \oplus I4(p-1) \oplus$$

$$I7(p-1) \oplus x2(p-1) \oplus x3(p-1)$$

$$Q2(p) = I2(p-1) \oplus I3(p-1) \oplus I4(p-1) \oplus I5(p-1) \oplus x1(p-1)$$

$$Q3(p) = I3(p-1) \oplus I4(p-1) \oplus I5(p-1) \oplus I6(p-1) \oplus x2(p-1)$$

40

$$Q4(p) = I4(p-1) \oplus I5(p-1) \oplus I6(p-1) \oplus I7(p-1) \oplus x3(p-1)$$

$$Q5(p) = I5(p-1) \oplus I6(p-1) \oplus I7(p-1) \oplus x1(p-1) \oplus x3(p-1)$$

$$Q6(p) = I6(p-1) \oplus I7(p-1) \oplus x_1(p-1) \oplus x_2(p-1) \oplus x_3(p-1)$$

$$Q7(p) = I7(p-1) \oplus x_1(p-1) \oplus x_2(p-1),$$

ここで

$$x_1(p) = I0(p-1) \oplus I2(p-1) \oplus I3(p-1) \oplus$$

10

$$I4(p-1) \oplus I7(p-1) \oplus x_2(p-1) \oplus x_3(p-1),$$

$$x_2(p) = I1(p-1) \oplus I3(p-1) \oplus I4(p-1) \oplus I5(p-1) \oplus$$

$$x_1(p-1), \text{ and}$$

$$x_3(p) = I2(p-1) \oplus I4(p-1) \oplus I5(p-1) \oplus I6(p-1) \oplus x_2(p-1).$$

であることを特徴とする請求項 2 に記載の並列ターボ符号化ブロック。

20

【請求項 6】

コンピュータの内部メモリに直接にロード可能なコンピュータプログラムであって、  
前記コンピュータプログラムがコンピュータ上で実行される場合に、請求項 1 に記載の方法を実行するためのソフトウェアコード部を備えることを特徴とするコンピュータプログラム。

【請求項 7】

前記ソフトウェアコード部が、V H D L 形式であることを特徴とする請求項 6 に記載のコンピュータプログラム。

【請求項 8】

前記ソフトウェアコード部が、

30

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_arith.ALL;

ENTITY en_turbo_coder_rtl IS
  PORT(
    -- General:
    reset_p          : IN STD_LOGIC;
    clk32m            : IN STD_LOGIC; -- Clock
                                (rising edge triggered)
    input_8           : IN std_logic_vector(7 DOWNTO 0);
    input_4           : IN std_logic_vector(3 DOWNTO 0);
    input_2           : IN std_logic_vector(1 DOWNTO 0);
    input_1           : IN std_logic;

    -- turboCoding 2 bit parallel
    output_8          : OUT std_logic_vector(7 DOWNTO 0);
    output_4          : OUT std_logic_vector(3 DOWNTO 0);
    output_2          : OUT std_logic_vector(1 DOWNTO 0);
    output_1          : OUT std_logic;
  );
END en_turbo_coder_rtl;

ARCHITECTURE rtl OF en_turbo_coder_rtl IS
```

```

SIGNAL s1_x1          : std_logic;
SIGNAL s1_x2          : std_logic;
SIGNAL s1_x3          : std_logic;
SIGNAL s1_i           : std_logic;
SIGNAL s1_o           : std_logic;

SIGNAL s2_x1          : std_logic;
SIGNAL s2_x2          : std_logic;
SIGNAL s2_x3          : std_logic;
SIGNAL s2_i           : std_logic_vector(1 DOWNTO 0);
SIGNAL s2_o           : std_logic_vector(1 DOWNTO 0);

SIGNAL s4_x1          : std_logic;
SIGNAL s4_x2          : std_logic;
SIGNAL s4_x3          : std_logic;
SIGNAL s4_i           : std_logic_vector(3 DOWNTO 0);
SIGNAL s4_o           : std_logic_vector(3 DOWNTO 0);

SIGNAL s8_x1          : std_logic;
SIGNAL s8_x2          : std_logic;
SIGNAL s8_x3          : std_logic;
SIGNAL s8_i           : std_logic_vector(7 DOWNTO 0);
SIGNAL s8_o           : std_logic_vector(7 DOWNTO 0);

BEGIN

    tc_1: PROCESS (clk32m, reset_p) -- seriell building of
                                     turbo coder block TCB

    BEGIN

        IF reset_p = '1' THEN

```

10

20

30

40



```
s1_x1 <= '0';  
s1_x2 <= '0';  
s1_x3 <= '0';  
s1_i  <= '0';  
s1_o  <= '0';
```

```
ELSIF clk32m'EVENT AND clk32m = '1' THEN
```

10

```
s1_i  <= input_1;  
s1_x1 <= s1_i XOR s1_x2 XOR s1_x3;  
s1_x2 <= s1_x1;  
s1_x3 <= s1_x2;  
s1_o  <= s1_i XOR s1_x2 XOR s1_x1;
```

```
END IF;
```

20

```
END PROCESS tc_1;
```

```
output_1 <= s1_o;
```

```
tc_2: PROCESS (clk32m, reset_p) -- 2bit parallel building  
                                     of turbo coder block
```

30

```
BEGIN
```

```
IF reset_p = '1' THEN
```

```
s2_x1 <= '0';  
s2_x2 <= '0';  
s2_x3 <= '0';  
s2_i  <= (OTHERS => '0');  
s2_o  <= (OTHERS => '0');
```

40

```
ELSIF clk32m'EVENT AND clk32m = '1' THEN
```

```
    s2_i  <= input_2;
    s2_x1 <= s2_i(0) XOR s2_x1 XOR s2_x2;
    s2_x2 <= s2_i(1) XOR s2_x2 XOR s2_x3;
    s2_x3 <= s2_x1;
    s2_o(0) <= s2_i(0) XOR
                s2_i(1) XOR s2_x1 XOR s2_x2 XOR s2_x3;
    s2_o(1) <= s2_i(1) XOR s2_x1 XOR s2_x2;
```

10

```
END IF;
```

```
END PROCESS tc_2;
```

```
output_2 <= s2_o;
```

20

```
tc_4: PROCESS (clk32m, reset_p) -- 4bit parallel building
                                of turbo coder block
```

```
BEGIN
```

```
IF reset_p = '1' THEN
```

30

```
    s4_x1  <= '0';
    s4_x2  <= '0';
    s4_x3  <= '0';
    s4_i    <= (OTHERS => '0');
    s4_o    <= (OTHERS => '0');
```

```
ELSIF clk32m'EVENT AND clk32m = '1' THEN
```

40

```
    s4_i    <= input_4;
    s4_x1    <= s4_i(0) XOR s4_i(2) XOR s4_i(3)
```

```

                                XOR s4_x1 XOR s4_x3;
s4_x2    <= s4_i(1) XOR s4_i(3) XOR s4_x1
                                XOR s4_x2 XOR s4_x3;
s4_x3    <= s4_i(2) XOR s4_x1 XOR s4_x2;
s4_o(0)  <= s4_i(0) XOR s4_i(1) XOR s4_i(2)
                                XOR s4_i(3) XOR s4_x3;
s4_o(1)  <= s4_i(1) XOR s4_i(2) XOR s4_i(3)
                                XOR s4_x1 XOR s4_x3;
s4_o(2)  <= s4_i(2) XOR s4_i(3) XOR s4_x1
                                XOR s4_x2 XOR s4_x3;
s4_o(3)  <= s4_i(3) XOR s4_x1 XOR s4_x2;

END IF;

END PROCESS tc_4;

output_4 <= s4_o;

tc_8: PROCESS (clk32m, reset_p) -- 8bit parallel building
                                of turbo coder block

BEGIN

    IF reset_p = '1' THEN

        s8_x1    <= '0';
        s8_x2    <= '0';
        s8_x3    <= '0';
        s8_i      <= (OTHERS => '0');
        s8_o      <= (OTHERS => '0');

        ELSIF clk32m'EVENT AND clk32m = '1' THEN

```

```

s8_i    <= input_8;
s8_x1   <= s8_i(0) XOR s8_i(2) XOR s8_i(3)
          XOR s8_i(4) XOR s8_i(7)
          XOR s8_x2 XOR s8_x3;
s8_x2   <= s8_i(1) XOR s8_i(3) XOR s8_i(4)
          XOR s8_i(5) XOR s8_x1;
s8_x3   <= s8_i(2) XOR s8_i(4) XOR s8_i(5)
          XOR s8_i(6) XOR s8_x2;
s8_o(0) <= s8_i(0) XOR s8_i(1) XOR s8_i(2)
          XOR s8_i(3) XOR s8_i(6)
          XOR s8_x1 XOR s8_x2;
s8_o(1) <= s8_i(1) XOR s8_i(2) XOR s8_i(3)
          XOR s8_i(4) XOR s8_i(7)
          XOR s8_x2 XOR s8_x3;
s8_o(2) <= s8_i(2) XOR s8_i(3) XOR s8_i(4)
          XOR s8_i(5) XOR s8_x1;
s8_o(3) <= s8_i(3) XOR s8_i(4) XOR s8_i(5)
          XOR s8_i(6) XOR s8_x2;
s8_o(4) <= s8_i(4) XOR s8_i(5) XOR s8_i(6)
          XOR s8_i(7) XOR s8_x3;
s8_o(5) <= s8_i(5) XOR s8_i(6) XOR s8_i(7)
          XOR s8_x1 XOR s8_x3;
s8_o(6) <= s8_i(6) XOR s8_i(7) XOR s8_x1
          XOR s8_x2 XOR s8_x3;
s8_o(7) <= s8_i(7) XOR s8_x1 XOR s8_x2;

END IF;

END PROCESS tc_8;

output_8 <= s8_o;

END rtl;

```

のように定義されることを特徴とする請求項7に記載のコンピュータプログラム。

【発明の詳細な説明】

【0001】

〔発明の属する技術分野〕

本発明は、ターボ符号器の実装に関連する。より詳細には、本発明はターボ符号器の実装の並列化に関連する。

【0002】

10

20

30

40

50

## 〔従来の技術〕

電気通信システムの送信機のベースバンド部は大抵の場合、送信ビットの汎用エンコーダを他のパーツと共に含むものである。エンコーダは、冗長な情報を到来したビットサイズ  $k$  のデータストリームに追加するものである。これにより、ビット数は符号化レート  $r$  に基づき係数  $1/r$  倍で増加する。現在、符号化レート  $1/2$  及び  $1/3$  が公に利用されているが、他のレートも同様に利用することができる。従って、符号化されていない  $K$  ビットの各ブロックについて、上記エンコーダは  $K/r$  符号化ビットのブロックを出力する。

## 【0003】

電気通信システムにおけるデータストリームの受信側では、いくつかのビットがエアインタフェースでの送信において破壊されている場合でも、オリジナルデータストリームが受信機の符号化されたデータブロックから再計算される。

10

## 【0004】

最近では、通信データの送信におけるエラー制御のためにターボ符号器が導入されている。一般には、ターボ符号化は、送信に先立ち同一情報シーケンスの異なるインタリーブバージョンに2以上のコンポーネント符号を適用することを含む。公知技術としてのターボ符号化には、たとえば、Berrou等の "Turbo-codes (ターボ符号)" IEEE International Communication Conference, pp.1064-1070, May 1993 や Sklar の "A Primer on Turbo Code Concepts (ターボ符号概念入門)", IEEE Communications Magazine, pp.94-102, December 1997, があり、ここで与えられるであろうさらなる詳細や、上記文献はここにおいて引用により記載に代えるものである。

20

## 【0005】

図8はターボ符号器の一例を示す。図8に示すように、ターボ符号器は、同一構成の2つのターボ符号器ブロックTCB1とTCB2から構成される。2つのターボ符号器ブロックの相違は、一方が入力ブロックのビットを順序不変で受信し、他方がインタリーブされた順序で入力ビットを受信することである。各入力ビットについて、出力0, 1, 2のそれぞれにおいて3出力ビットが生成される。

## 【0006】

図8に示すように、ターボ符号器ブロックTCB1は第1のXORゲート100を入力段に備え、第2のXORゲート102を出力段に備える。その間において、3つの遅延ユニット104から108が存在し、各入力ビットを遅延させる。同様に、ターボ符号器ブロックTCB2は、第3のXORゲート110を入力段に、第4のXORゲート112を出力段に備える。その間において、3つの遅延ユニット114から118が存在し、各入力ビットを遅延させる。

30

## 【0007】

さらに図8に示されるように、ターボ符号器への入力信号は、ターボ符号器ブロック1へ直接供給され、その一方で、入力信号はインタリーバ120を介して、第2のターボ符号器ブロックTCB2へ供給される。出力0については、入力信号がそのまま何の修正もされずに転送される。

## 【0008】

図8に示すターボ符号器のためのパラメータは、各ターボ符号器ブロックにおける遅延ユニットの番号であり、さらに、異なるXORゲート100, 102, 110及び112への入力信号の供給を示すものである。

40

## 【0009】

図8に示すターボ符号器を端的に実現するには、入力及び出力レジスタ(図8には不図示)と同様に、ターボ符号化がビットワイズ手法(bitwise manner)で実行されるように、遅延ユニットを利用することになる。ここでは、遅延ユニットの出力がそれ以前のクロックサイクルにおける入力を表すように、すべての遅延ユニットが同一システムクロックで動作する。

## 【0010】

遅延ユニットをシリアルに利用してターボ符号器を端的に実現することにより、多くのレ

50

ジスタやXORゲートを必要としない一方で、その主な不利益は、ターボ符号化がシリアルに実行されることにある。このことは、システムクロックの1サイクルで1ビットしか符号化されないことを意味する。結果として、高ビットレートを必要とする場合には、システムクロック周波数は非常に高いものになってしまう。

【0011】

もし、たとえば1200チャンネルが符号化されるとして、それぞれが1ミリ秒に100ビットを有する音声チャンネルに関連するならば、必要なシステムクロック周波数は120MHzである。

【0012】

このように、たとえばASICやFPGA技術を利用して上述のターボ符号器を実現することは非常に困難である。

10

【0013】

専用ターボ符号器を各チャンネルに備えることは一つの解決策である一方で、異なるチャンネルは依然として並列に符号化されなければならないので、入力及び出力のビットストリームの複雑なハンドリングが要求される。このためには、正しい入力を正しい時間に正しいターボ符号器に供給するための非常に複雑な制御ロジックが導入されることになる。さらに、異なるターボ符号器の出力の扱いもまた、同様に複雑になるであろう。

【0014】

[発明の概要]

上記の観点から、本発明の目的はターボ符号器ブロックの処理速度を向上させることにあ

20

る。

【0015】

本発明によれば、この目的はターボ符号器の並列化(parallel realization)によって達成される。

【0016】

これまで、ターボ符号器ブロックの構造は、ターボ符号器ブロックの並列化のための基礎をなす汎用形式の表現で記載されていた。

【0017】

特に、ターボ符号器ブロックへの入力サンプルは、並列度を表すnを用いて、n次元の並列入力ベクトルの要素として変換される。

30

【0018】

ターボ符号器ブロックの汎用形式の表現は、この並列入力ベクトルの、少なくとも1の並列出力ベクトルへのマッピングを導くために利用される。

【0019】

より詳細には、内部状態変数置換処理(internal state variable substitution process)が、汎用形式表現における各内部状態変数に適用される。ここで、内部状態変数の表現は、それ以前に決定された時間インデックスが置換された内部状態変数の表現を利用した後方時間インデックス遷移の実行を通して置換された最大時間インデックス要素についてスキャンされる。これらの置換ステップは、各内部状態変数の表現が入力ベクトル要素及び並列化の度合いにより遅延するターボ符号器の内部状態変数の値にのみ依存するまで繰

40

り返される。

【0020】

さらに、置換処理は各並列出力ベクトルの各要素について同様に実行される。再度、各出力ベクトル内の各ベクトル要素の表現は、最大時間インデックスを有する内部状態変数についてスキャンされ、ベクトル要素の表現における後方時間インデックス遷移の決定が、各ベクトル要素の表現が入力ベクトル要素及び並列化の度合いにより遅延するターボ符号器の内部状態変数の値にのみ依存するまで再帰的行われる。

【0021】

それゆえに、本発明は、処理速度を向上させるために、多くのシリアルターボ符号器の代わりに唯一の並列化ターボ符号器ブロックのみを必要とするような、ターボ符号器ブロッ

50

クの並列化を提案する。これは、複数のターボ符号器ブロックに必要な複雑な入力及び出力制御を不要とする決定的な利益をもたらすものである。

【 0 0 2 2 】

たとえば、4ビット並列ターボ符号器ブロックでは、上記の例におけるシステムクロック周波数は30MHzの範囲になるので、FPGAやASIC技術を利用して簡単に実現することができる。ゆえに、並列化されたターボ符号器ブロック及びそこから導き出されるターボ符号器は、FPGA又はASIC技術ベースの実装を利用するための所定のスペックに見合う並列度に応じて、シリアルターボ符号器を越える高速度化を達成できる。このように、並列化されたターボ符号器ブロックは、異なるチャンネルを同時に複雑にハンドリングする必要もなく、ベースバンド部における低処理遅延を実現する複雑な電気通信システムの基礎を形成するものである。

10

【 0 0 2 3 】

さらに、並列化されたターボ符号器ブロック及び上記ターボ符号器は、シリアルターボ符号器と比較して、論理ゲート及びレジスタの追加余剰が最小限ですむ。

【 0 0 2 4 】

また、ターボ符号器の前及びその後続く処理ブロックもまた並列処理手法をサポートする場合は、シリアルターボ符号器及び（パラレルツースリアル又はその逆の）付加的コンバータを備えるエンコーダブロックに比べて、エンコーダブロック全体として完全な符号化ブロックが必要とするロジック及びレジスタはより少なくなる。

20

【 0 0 2 5 】

本発明の好適な実施形態によれば、コンピュータ上で走らせる場合に、本発明の並列化方法に対応するステップを実行するソフトウェアコード部を備えるコンピュータの内部メモリに直接ロード可能なコンピュータプログラム製品をも提供される。より好ましくは、ソフトウェアコード部は、VHDLタイプのものである。

【 0 0 2 6 】

それゆえに、開発サイクルが少なく、デザインをたとえばASICやFPGAなどの異なるハードウェア技術にマッピングする場合の高い柔軟性を有する並列ターボ符号器設計の高速な開発及び修正の基礎を実現するものである。

【 0 0 2 7 】

[ 好適な実施形態の説明 ]

30

以下において、本発明の好適な実施形態が図1乃至図7を参照して記載される。

【 0 0 2 8 】

特に、図1及び図2に示すようなターボ符号器ブロックの形式的な記載は、本発明に対応した並列化方法の記載の基礎として与えられるであろう。

【 0 0 2 9 】

図3を参照して、データ入力のタイムシーケンシャルストリームが、並列化されたターボ符号器ブロックにおいて扱われるために、どのように並列入力ベクトルにマッピングされるかが記載されてもよい。また、この並列化されたターボ符号器ブロックの並列出力ベクトルがデータ出力のタイムシーケンシャルストリームにマッピングされるかが記載されてもよい。

40

【 0 0 3 0 】

これに続いて、本発明に対応した並列化方法を利用するターボ符号器ブロックの並列化を実現するための異なる例は、図4から図7を参照して記載される。

【 0 0 3 1 】

[ 1 . シリアル型のターボ符号器ブロック、汎用記載 ]

図1は、シリアル型のターボ符号器ブロックのブロック線図である。

【 0 0 3 2 】

図1に示すように、ターボ符号器ブロックは、N個の遅延ユニット、 $X_1$ ,  $X_2$ ,  $\dots$ ,  $X_N$ を、たとえばフリップフロップ等で備える。遅延ユニット $X_1$ ,  $X_2$ ,  $\dots$ ,  $X_N$ の出力信号は、それぞれ $X_1(t)$ ,  $X_2(t)$ ,  $\dots$ ,  $X_N(t)$ とし、ここでtは、正数の時間

50

を表す指数である。第 1 の遅延ユニット  $X_1$  の入力には、入力 XOR ゲート 1 2 が提供され、最後の遅延ユニット  $X_N$  の出力には、出力 XOR ゲート 1 4 が提供される。

【 0 0 3 3 】

図 1 に示すように、入力信号  $I(t)$  は入力遅延ユニット 1 6 を介して供給され、出力 XOR ゲート 1 4 の出力信号は、出力遅延ユニット 1 8 を介して転送される。

【 0 0 3 4 】

また、図 1 を参照すると、出力信号  $X_1(t)$ ,  $X_2(t)$ ,  $\dots$ ,  $X_N(t)$  は、接続  $_1$ ,  $_2$ ,  $\dots$ ,  $_N$  を介して入力 XOR ゲート 1 2 に提供される。また、遅延ユニットに対する入力信号は、接続  $_0$ ,  $_1$ ,  $\dots$ ,  $_{N-1}$  を介して出力 XOR ゲート 1 4 へ提供される。さらに、遅延ユニット  $X_N$  の出力は、接続  $_N$  を介して出力 XOR ゲート 1 4 へ供給される。 10

【 0 0 3 5 】

形式的に 0 又は 1 の値を、接続  $_1$ ,  $_2$ ,  $\dots$ ,  $_N$  のそれぞれ、さらに接続  $_0$ ,  $_1$ ,  $\dots$ ,  $_{N-1}$  のそれぞれに割り当てると、N 個の遅延ユニットを有するどんなシリアルターボ符号器ブロックも、以下の式に従って記載することができる。

【 0 0 3 6 】

$$x_1(t) = I(t-1) \oplus \alpha_1 \cdot x_1(t-1) \oplus \alpha_2 \cdot x_2(t-1) \oplus \dots \oplus \alpha_N \cdot x_N(t-1)$$

$$\alpha_i \in \{0, 1\}$$

20

(1.x<sub>1</sub>)

$$x_2(t) = x_1(t-1)$$

(1.x<sub>2</sub>)

.

.

.

.

30

$$x_N(t) = x_{N-1}(t-1)$$

(1.x<sub>N</sub>)

$$Q(t) = \beta_0 \cdot [I(t-1) \oplus \alpha_1 \cdot x_1(t-1) \oplus \alpha_2 \cdot x_2(t-1) \oplus \dots \oplus \alpha_N \cdot x_N(t-1)] \oplus \beta_1 \cdot x_1(t-1) \oplus \beta_2 \cdot x_2(t-1) \oplus \dots \oplus \beta_N \cdot x_N(t-1) =$$

40

$$= \beta_0 \cdot I(t-1) \oplus \beta_0 \cdot \alpha_1 \cdot x_1(t-1) \oplus \beta_1 \cdot x_1(t-1) \oplus$$



$$\begin{aligned}
 & \cdot \\
 & \cdot \\
 & \cdot \\
 & \beta_0 \cdot \alpha_N \cdot x_N(t-1) \oplus \beta_N \cdot x_N(t-1) = \\
 & = \beta_0 \cdot I(t-1) \oplus \\
 & \quad x_1(t-1) \cdot [\beta_0 \cdot \alpha_1 \oplus \beta_1] \oplus \\
 & \quad \cdot \\
 & \quad \cdot \\
 & \quad \cdot \\
 & \quad x_N(t-1) \cdot [\beta_0 \cdot \alpha_N \oplus \beta_N] \\
 & \beta_i \in \{0, 1\} \\
 & (1.Q)
 \end{aligned}$$

10

【 0 0 3 7 】

20

図 2 に示すように、単一出力を有するターボ符号器ブロックについて上記のように与えられる一般的な表現は、シリアル型で複数の出力を有するターボ符号器ブロックについて一般化されてもよい。

【 0 0 3 8 】

図 1 のように、図 2 もまた、遅延ユニット  $X_1, X_2, \dots, X_N$  について、その出力が、それぞれ  $X_1(t), X_2(t), \dots, X_N(t)$  である。遅延ユニット  $X_1, X_2, \dots, X_N$  のそれぞれの出力信号  $X_1(t), X_2(t), \dots, X_N(t)$  は、図 1 においてすでに記載されているように、接続  $1, 2, \dots, N$  を介して入力 XOR ゲート 12 に提供される。また、入力データストリームは、入力遅延ユニット 16 を介して入力 XOR ゲート 12 に提供される。

30

【 0 0 3 9 】

図 2 に記載のシリアルターボ符号器ブロックは、複数の出力  $Q_1(t), \dots, Q_M(t)$  が与えられる点で、上記のターボ符号器ブロックとは異なる。図 2 に示されるように、各出力について、関連する出力 XOR ゲート 14 - 1、 $\dots$ 、14 - M が存在する。そのような出力 XOR ゲート 14 - 1、 $\dots$ 、14 - M のそれぞれにおいて、関連する出力遅延ユニット 18 - 1、 $\dots$ 、18 - M が接続されている。

【 0 0 4 0 】

図 2 にさらに示されているように、遅延ユニット  $X_1, X_2, \dots, X_N$  への各入力は、第 1 の出力 XOR ゲート 14 - 1 へ、接続  $1,0, 1,1, \dots, 1,N-1$  を介して供給され、さらに、第 2 の出力 XOR ゲート 14 - 2 へも、接続  $2,0, 2,1, \dots, 2,N-1$  を介して供給され、以下同様に続く。さらに、最終の遅延ユニット  $X_N$  の出力は、接続  $1,N$  を介して、第 1 の出力 XOR ゲート 14 - 1 へ供給され、接続  $1,N$  を介して、第 1 の出力 XOR ゲート 14 - 1 へ供給され、接続  $2,N$  を介して、第 2 の出力 XOR ゲート 14 - 2 へ供給され、以下同様に続く。最後に、各出力 XOR ゲートの出力信号が、ターボ符号器ブロック出力信号  $Q_1(t), \dots, Q_M(t)$  を導くために遅延される。

40

【 0 0 4 1 】

図 2 のターボ符号器ブロックの形式的な表現は、複数の出力  $Q_1(t), \dots, Q_M(t)$  について関連する出力表現が以下のように一般化される点で、上記のターボ符号器ブロックとは異なる。

【 0 0 4 2 】

50

$$\begin{aligned}
 Q_j(t) = & \beta_{j0} \cdot I(t-1) \oplus \\
 & x_1(t-1) \cdot [\beta_{j0} \cdot \alpha_1 \oplus \beta_{j1}] \oplus \\
 & \cdot \\
 & \cdot \\
 & \cdot \\
 & x_N(t-1) \cdot [\beta_{j0} \cdot \alpha_N \oplus \beta_{jN}]
 \end{aligned}$$

$$\beta_{ij} \in \{0, 1\}$$

$$j \in [1, \dots, M]$$

10

(1.Q)

【 0 0 4 3 】

上記の 1 以上の出力を有するシリアル型のターボ符号器ブロックの形式表現は、本発明に対応する並列化方法の表現の基礎となるであろう。

【 0 0 4 4 】

しかしながら、以下に本発明に対応する並列化方法を記載する前に、まず、シリアルデータ入力ストリームの並列入力ベクトルへのマッピングと、並列出力ベクトルをシリアルデータ出力ストリームへマッピングし直すことについて、図 3 を参照して説明する。

20

【 0 0 4 5 】

[ 2 . n 次の並列型ターボ符号器ブロック ]

図 3 の左側上部において、シリアルデータ入力ストリームに対応するタイムシーケンスが示されている。同様に、図 3 の右側上部には、シリアル出力ストリームに対応する時間シーケンスが示されている。

【 0 0 4 6 】

図 3 に示すように、並列入力ベクトル  $I_0, I_1, I_{n-1}$  は、データ入力値  $I(t-1), I(t-2), \dots, I(t-n)$  の系列から以下のように導き出されてもよい。

【 0 0 4 7 】

$$I(t-1) = I_0(p-1)$$

$$I(t-2) = I_1(p-1)$$

.

.

.

.

$$I(t-n) = I_{n-1}(p-1)$$

シリアルデータ入力ストリームから導き出された図 3 に示す並列入力ベクトルが、以下に記載する並列ターボ符号器ブロックにおいて処理されるとすると、あるものは並列出力ベクトル  $Q_{j0}, \dots, Q_{jn-1}$  に到達する。

40

【 0 0 4 8 】

図 3 に示すように、この並列出力ベクトルは、シリアルデータ出力ストリーム  $Q_j(t), \dots, Q_j(t-(n-1))$  へマッピングし直されてもよい。

【 0 0 4 9 】

$$Q_j(t) = Q_{j0}(p)$$

$$Q_j(t-1) = Q_{j1}(p)$$

.

.

.

$$Q_j(t-(n-1)) = Q_{jn-1}(p)$$

50

j [1, ..., M]

【 0 0 5 0 】

[ 2 . 1 並列化の一般的なアウトライン ]

本発明によれば、異なる時間スケールがシリアルドメイン及びパラレルドメインにおいて利用される。言い換えれば、シリアルタイムドメインにおける  $n$  個のシリアルデータ入力のセットが、並列タイムドメイン  $p$  における 1 クロックサイクルで並列に処理される並列入力ベクトルに対応する。

【 0 0 5 1 】

故に、結果としての並列システムクロックは、同量のデータを処理する場合に想定されるシリアルシステムクロックよりも  $n$  倍遅くなる。それゆえに、シリアル時間表現 ( $t - 1$ ) は、並列時間表現 ( $p - 1$ ) と置き換えられてもよく、 $n$  個のシリアル時間入力のシーケンスは、1 並列クロックサイクル前の ( $p - 1$ )、単一並列入力ベクトルとして書き換えられる。同様に、出力の系列も現在の並列サイクル ( $p$ ) についての並列出力ベクトルとして書き換えられる。

10

【 0 0 5 2 】

図 1 及び図 2 に応じたターボ符号器ブロックの一般的表現及び、さらに、シリアルデータ入力ストリームの並列入力ベクトルへのマッピング、及び並列出力ベクトルのシリアル出力データストリームへの逆マッピングを利用するので、本発明のアプローチは、 $n$  個のシリアルデータ入力アイテムを有する並列入力ベクトルの、並列化されたターボ符号器ブロックへの供給、及び、それぞれが  $n$  個の出力データアイテムを備える  $M$  個の出力ベクトルの計算に依拠する。

20

【 0 0 5 3 】

出力ベクトルの  $n$  個の要素は、シリアルターボ符号器ブロックの  $n$  システムクロック前における (並列化されたターボ符号器ブロックにおける 1 サイクルに対応する) 内部遅延ユニット  $X_1, \dots, X_N$  の値と、さらには、並列入力ベクトルにおいてこの  $n$  サイクルにおいて加算されたすべての入力データアイテムにのみ依存する。

【 0 0 5 4 】

以下に記載するように、本発明のアプローチは、並列出力ベクトルの出力要素と、並列ターボ符号器ブロックの内部遅延ユニットの次の並列サイクルについての、並列ターボ符号器ブロックの内部遅延ユニットの出力値と以前の並列サイクルの並列入力ベクトルに依存した出力値を表す等式の決定に依存している。

30

【 0 0 5 5 】

シリアル表現 (たとえば、式 (1.Q)) を参照すると、時間  $t$  におけるシリアル型のターボ符号器ブロックの各出力値は、遅延ユニット  $X_1, \dots, X_N$  の時間  $t - 1$  における出力値及び時間  $t - 1$  における入力データアイテムから計算される。

【 0 0 5 6 】

内部遅延ユニットのすべての出力値を、シリアル実装における 1 システムクロックサイクル前における値で置換し、この置換ステップを  $n$  シリアルシステムクロックサイクル前の出力値のみが残るまで繰り返すことにより、 $n$  シリアルサイクル又は同等な 1 並列サイクルについての出力結果の並列計算式が導出される。

40

【 0 0 5 7 】

言い換えれば、これらの等式は、 $n$  シリアルクロックサイクルが 1 パラレルクロックサイクルを表すので、並列化されたターボ符号器ブロックの基礎を形成する。

【 0 0 5 8 】

[ 2 . 2 第 1 の内部状態についての後方時間インデックス置換 ]

上述の再帰的置換処理の準備において、第 1 の遅延ユニット  $X_1$  の出力を以下のように表現することができる。

【 0 0 5 9 】

$$\begin{aligned}
 x_1(t) &= I(t-1) \oplus \alpha_1 \cdot x_1(t-1) \oplus \\
 &\quad \alpha_2 \cdot x_2(t-1) \oplus \dots \oplus \\
 &\quad \alpha_N \cdot x_N(t-1)
 \end{aligned}$$

【 0 0 6 0 】

これは、シリアルクロックサイクル  $t - 1$  ,  $\dots$  ,  $t - (n - 1)$  の内部状態を表現するために、以下において第 1 の内部状態と呼ばれる。

【 0 0 6 1 】

$$\begin{aligned}
 x_1(t-1) &= I(t-2) \oplus \alpha_1 \cdot x_1(t-2) \oplus & 10 \\
 &\quad \alpha_2 \cdot x_2(t-2) \oplus \dots \oplus \\
 &\quad \alpha_N \cdot x_N(t-2) \\
 &\hspace{15em} (2.x_1.1)
 \end{aligned}$$

$\cdot$   
 $\cdot$   
 $\cdot$

20

$$\begin{aligned}
 x_1(t-(n-1)) &= I(t-n) \oplus \alpha_1 \cdot x_1(t-n) \oplus \\
 &\quad \alpha_2 \cdot x_2(t-n) \oplus \dots \oplus \\
 &\quad \alpha_N \cdot x_N(t-n) \\
 &\hspace{15em} (2.x_1.n-1)
 \end{aligned}$$

【 0 0 6 2 】

[ 2 . 3 更なる内部状態 ( $i=2, \dots, N$ ) についての後方時間インデックス置換 ]

付随する遅延ユニット  $X_2$  ,  $\dots$  ,  $X_N$  の出力 (式 (1.x2) ,  $\dots$  , (1.xN) 参照) を以下のように表現することができる。

30

$$X_i(t) = X_{i-1}(t-1)$$

シリアルクロックサイクル  $t - 1$  ,  $\dots$  ,  $t - (n - 1)$  の内部状態を表現するために、これは更なる内部状態と呼ばれる。

$$X_i(t-1) = X_{i-1}(t-2) \hspace{10em} (2.xi.1)$$

$\cdot$   
 $\cdot$   
 $\cdot$

$$X_i(t(n-1)) = X_{i-1}(t-n) \hspace{10em} (2.xi.n-1)$$

【 0 0 6 3 】

[ 2 . 4 出力ベクトル要素についての後方時間インデックス置換 ]

40

さらに、ターボ符号器ブロックの少なくとも 1 の出力 (式 (1.Q) 参照) についての表現は、

$$\begin{aligned}
 Q_j(t) &= \beta_{j0} \cdot I(t-1) \oplus \\
 &\quad x_1(t-1) \cdot [\beta_{j0} \cdot \alpha_1 \oplus \beta_{j1}] \oplus \\
 &\quad \cdot \\
 &\quad \cdot \\
 &\quad \cdot \\
 &\quad x_N(t-1) \cdot [\beta_{j0} \cdot \alpha_N \oplus \beta_{jN}]
 \end{aligned}
 \tag{2.Q.0}$$

10

シリアルクロックサイクル  $t-1$  ,  $\dots$  ,  $t-(n-1)$  についての、以下の出力を表現するために利用されてもよい。

$$\begin{aligned}
 Q_j(t-1) &= \beta_{j0} \cdot I(t-2) \oplus \\
 &\quad x_1(t-2) \cdot [\beta_{j0} \cdot \alpha_1 \oplus \beta_{j1}] \oplus \\
 &\quad \cdot \\
 &\quad \cdot \\
 &\quad \cdot \\
 &\quad x_N(t-2) \cdot [\beta_{j0} \cdot \alpha_N \oplus \beta_{jN}]
 \end{aligned}
 \tag{2.Q.1}$$

20

$$\begin{aligned}
 &\cdot \\
 &\cdot \\
 &\cdot \\
 Q_j(t-(n-1)) &= \\
 &\quad \beta_{j0} \cdot I(t-n) \oplus \\
 &\quad x_1(t-n) \cdot [\beta_{j0} \cdot \alpha_1 \oplus \beta_{j1}] \oplus \\
 &\quad \cdot \\
 &\quad \cdot \\
 &\quad \cdot \\
 &\quad x_N(t-n) \cdot [\beta_{j0} \cdot \alpha_N \oplus \beta_{jN}]
 \end{aligned}
 \tag{2.Q.n-1}$$

30

【 0 0 6 4 】

[ 2 . 5 再帰的後方時間インデックス遷移ステップによる並列化 ]

[ 2 . 5 . 1 目的 ]

40

すでに上記において述べた本発明に対応する並列化処理の目的は、各並列出力ベクトル（図2に示す出力1、 $\dots$ 、Mごとに一つ）の要素を、 $n$ シリアルデータ入力アイテム及び現在の並列サイクルよりも1並列サイクル（ $n$ シリアルサイクルに相当）前の内部状態の関数として表現することである。

【 0 0 6 5 】

$Q_j(t-i) = f[I(t-1), \dots, I(t-n); x_1(t-n), \dots, x_N(t-n)]$   
 for  $i \in [0, \dots, n-1]$   
 $j \in [1, \dots, M]$   
 $\longleftrightarrow$   
 $Q_{ji}(p) = f[I_0(p-1), \dots, I_{n-1}(p-1); x_1(p-1), \dots, x_N(p-1)]$   
 (2. Qi)

【 0 0 6 6 】

[ 2 . 5 . 2 再帰的時間インデックス遷移ステップ ]

まず、この目的を実現するためにすべての内部状態が並列化される。すべての内部状態  $X_1(t)$ ,  $X_2(t)$ ,  $\dots$ ,  $X_N(t)$  について、この内部状態の表現において使用される最大時間インデックスが、まず動作のシリアル表現に対応する  $t - 1$  にセットされる。

10

【 0 0 6 7 】

全ての内部状態の表現が考慮され、上記の式 ( 2 ) のうちの一つを利用して、この現在の最大時間インデックスに先立つ時間インデックスに依存する同一の内部状態を表現する形式により、それぞれが現在の最大時間インデックスに依存する内部状態変数を置換するように再帰的にスキャンされる。

【 0 0 6 8 】

それゆえに、考慮された内部状態の表現は、現在の最大時間インデックスから、これに先行する時間インデックスへの遷移を行なうことで修正される。この遷移を以下では後方時間インデックス遷移と呼ぶ。後方時間インデックス遷移が、現在の最大時間インデックスに依存する全ての内部状態変数実行されると、現在の最大時間インデックスは 1 減少する。

20

【 0 0 6 9 】

この処理の再帰的反復は、最大時間インデックスが、 $t-n$  になるまで継続される。この理由は、この点において、サイクル  $t(p)$  における考慮される内部状態は、時間  $t - n(p - 1)$  における内部状態及びシリアル入力データアイテム  $I(t - 1)$ 、 $\dots$ 、 $I(t - n)$  にのみ依存して記載されるからである。

【 0 0 7 0 】

つぎに、再度、上述の後方時間インデックス遷移を内部状態について再帰的に実行することにより、各出力ベクトルの全ての要素が並列化される。

30

【 0 0 7 1 】

さらに、再帰的時間インデックス遷移ステップの途中段階における、現在の最大時間インデックスの修正では、内部状態のそれぞれの表現及び出力ベクトルの要素を、 $(a \text{ XOR } a \text{ XOR } b = b)$  の関係を利用することにより単純化することができる。言い換えれば、重複するターム  $(a \text{ XOR } a)$  は、それぞれの表現においてキャンセルされる。

【 0 0 7 2 】

図 2 に示すようなもっとも一般的な形式におけるシリアルターボ符号器ブロックの再帰的並列化は、以下のようにまとめることができる。

【 0 0 7 3 】

procedure recursive\_parallelization(n)

40

```
int i, j, k;  
{
```

```
    /* parallelization of internal states */
```

```
    for( k=1; k<=N; k++ )
```

```
    {
```

```
        /* consider internal state  $x_k(t)$  */
```

10

```
        tmax = t-1;
```

```
        while( tmax > t-n )
```

```
        {
```

```
            scan representation of internal state  $x_k(t)$   
            for internal states with maximum time index  
            tmax;
```

20

```
            for( all internal states with maximum time  
                index )
```

```

    {
        execute backward time index transition
        from  $t_{\max}$  to  $t_{\max}-1$  through state variable
        substitution using eq. (2) above;
    }
    cancel double terms in representation of
    internal state  $x_k(t)$ ;
     $t_{\max} = t_{\max}-1$ ;
}
}

/* parallelization of output variables */

for( j=1; j<=M; j++ )
{
    for( i=0; i<=n-2; i++ )
    {
        /* consider output vector element  $Q_j(t-i)$  */
         $t_{\max} = t-i-1$ ;
        while(  $t_{\max} > t-n$  )
        {
            scan representation of  $Q_j(t-i)$  for
            internal states with maximum time index;

            for( all internal states with maximum time
            index )
            {
                execute backward time index
                transition from  $t_{\max}$  to  $t_{\max}-1$ 
                through state variable substitution
                using eq. (2) above;
            }
        }
    }
}

```



```

cancel double terms in representation of
Qj(t-i);
tmax = tmax-1;
    }
}
}
} /* end procedure */

```

10

## 【0074】

以下において、上記の並列化アプローチの適用について説明するために、この並列化処理はより詳細に特定の例と関連して説明される。

## 【0075】

以下に示すように、各並列化ターボ符号器ブロックは、例えば、フリップフロップ $l_0, \dots, l_7$ の第1のグループのような、入力信号 $I(t)$ の $n$ 個のサンプル $l(t-1), \dots, l(t-n)$ を並列化ターボ符号器ブロックにさらに供給するための格納部と、例えば、フリップフロップ $Q_0, \dots, Q_7$ の第2のグループのような、並列化ターボ符号器ブロックの少なくとも1つの出力信号 $Q_j(t), j=1, \dots, M$ の $n$ 個のサンプル $Q_j(t), \dots, Q_j(t-(n-1))$ を格納する少なくとも1つの格納部を備える。

20

## 【0076】

さらに、並列化ターボ符号器ブロックは、 $n$ 個の遅延ユニット $X_1, \dots, X_N$ の列(bank)を備え、当該列の少なくとも2つの遅延ユニット $X_1, \dots, X_N$ が、入力信号 $I(t)$ の $n$ 個のサンプル $l(t-1), \dots, l(t-n)$ のサブセットを直接受け、並列化ターボ符号器ブロックの少なくとも1つの遅延ユニット $X_1, \dots, X_N$ の出力信号が、並列化ターボ符号器ブロックの少なくとも2つの遅延ユニットへ供給されるように、入力信号 $I(t)$ の $n$ 個のサンプルの並列処理に適応されている。

## 【0077】

図4は、図8に示すターボ符号化において利用される、シリアル型の特定のターボ符号器ブロックのブロック線図である。

30

## 【0078】

ここで、入力遅延ユニット16及び出力遅延ユニット18の提供は、シリアル実装する場合の前提条件ではないが、並列化された場合に、図3に示すような入力ベクトル及び出力ベクトルの要素を訂正するために必要である。

## 【0079】

図4に示すシリアルターボ符号器ブロックは、上記の式(1)に従い形式的表現を利用して $N=3, M=1, \mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3] = [0, 1, 1], \mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4] = [1, 1, 0, 1]$ とすることにより簡単化してもよい。以下においては、いかにしてこのシリアルターボ符号器ブロックが、 $n=2, 4$ 及び8ビットの並列ターボ符号器ブロックに並列化されるかを説明する。

40

## 【0080】

[3.1 例： $N=3, M=1, \mathbf{A} = [0, 1, 1], \mathbf{B} = [1, 1, 0, 1]$ の場合の2ビット並列ターボ符号化]

式(1)より、図4に示されるターボ符号器ブロックについての表現として、以下を得ることができる。

## 【0081】

$$x_1(t) = I(t-1) \oplus x_2(t-1) \oplus x_3(t-1)$$

$$x_2(t) = x_1(t-1)$$

$$x_3(t) = x_2(t-1)$$

$$Q(t) = I(t-1) \oplus x_1(t-1) \oplus x_2(t-1)$$

【 0 0 8 2 】

2 シリアル時間サイクルについての後方時間インデックス遷移が必要である。

【 0 0 8 3 】

$$x_1(t) = I(t-1) \oplus x_2(t-1) \oplus x_3(t-1)$$

10

$$\Leftrightarrow x_1(t) = I(t-1) \oplus x_1(t-2) \oplus x_2(t-2)$$

並列回路の場合：

$$\Rightarrow x_1(p) = I_0(p-1) \oplus x_1(p-1) \oplus x_2(p-1)$$

$$x_2(t) = x_1(t-1)$$

$$\Leftrightarrow x_2(t) = I(t-2) \oplus x_2(t-2) \oplus x_3(t-2)$$

並列回路の場合：

20

$$\Rightarrow x_2(p) = I_1(p-1) \oplus x_2(p-1) \oplus x_3(p-1)$$

$$x_3(t) = x_2(t-1)$$

$$\Leftrightarrow x_3(t) = x_1(t-2)$$

並列回路の場合：

$$\Rightarrow x_3(p) = x_1(p-1)$$

30

$$Q(t) = I(t-1) \oplus x_1(t-1) \oplus x_2(t-1)$$

$$\Leftrightarrow Q(t) = I(t-1) \oplus [I(t-2) \oplus x_2(t-2) \oplus x_3(t-2)] \oplus x_1(t-2)$$

並列回路の場合：

$$\Rightarrow Q_0(p) = I_0(p-1) \oplus I_1(p-1) \oplus x_1(p-1) \oplus x_2(p-1) \oplus x_3(p-1)$$

$$Q(t-1) = I(t-2) \oplus x_1(t-2) \oplus x_2(t-2)$$

並列回路の場合：

$$\Rightarrow Q_1(p) = I_1(p-1) \oplus x_1(p-1) \oplus x_2(p-1)$$

40

【 0 0 8 4 】

図 5 は、この並列化処理の結果を示す。すなわち、図 4 に示すターボ符号器ブロックを 2 ビット並列化した場合のブロック線図である。

【 0 0 8 5 】

[ 3 . 2 例：N = 3 , M = 1、 = [ 0 , 1 , 1 ]、 = [ 1 , 1 , 0 , 1 ] の場合の 4 ビット並列ターボ符号化 ]

式 ( 1 ) より、図 4 に示されるターボ符号器ブロックについての表現として、以下を得ることができる。

【 0 0 8 6 】

50

$$x_1(t) = I(t-1) \oplus x_2(t-1) \oplus x_3(t-1)$$

$$x_2(t) = x_1(t-1)$$

$$x_3(t) = x_2(t-1)$$

$$Q(t) = I(t-1) \oplus x_1(t-1) \oplus x_2(t-1)$$

【 0 0 8 7 】

4 シリアル時間サイクルについての後方時間インデックス遷移が必要である。

【 0 0 8 8 】

$$x_1(t) = I(t-1) \oplus x_2(t-1) \oplus x_3(t-1)$$

10

$$\Rightarrow x_1(t) = I(t-1) \oplus x_1(t-2) \oplus x_2(t-2)$$

$$\Leftrightarrow x_1(t) = I(t-1) \oplus I(t-3) \oplus x_2(t-3) \oplus x_3(t-3) \oplus x_1(t-3)$$

$$\Leftrightarrow x_1(t) = I(t-1) \oplus I(t-3) \oplus x_1(t-4) \oplus x_2(t-4) \oplus$$

$$I(t-4) \oplus x_2(t-4) \oplus x_3(t-4)$$

並列回路の場合：

$$\Rightarrow x_1(p) = I_0(p-1) \oplus I_2(p-1) \oplus I_3(p-1) \oplus x_1(p-1) \oplus x_3(p-1)$$

$$x_2(t) = x_1(t-1)$$

10

$$\Leftrightarrow x_2(t) = I(t-2) \oplus x_2(t-2) \oplus x_3(t-2)$$

$$\Leftrightarrow x_2(t) = I(t-2) \oplus x_1(t-3) \oplus x_2(t-3)$$

$$\Leftrightarrow x_2(t) = I(t-2) \oplus [I(t-4) \oplus x_2(t-4) \oplus x_3(t-4)] \oplus x_1(t-4)$$

並列回路の場合：

$$\Rightarrow x_2(p) = I_1(p-1) \oplus I_3(p-1) \oplus x_1(p-1) \oplus x_2(p-1) \oplus x_3(p-1)$$

$$x_3(t) = x_2(t-1)$$

20

$$\Leftrightarrow x_3(t) = x_1(t-2)$$

$$\Leftrightarrow x_3(t) = [I(t-3) \oplus x_2(t-3) \oplus x_3(t-3)]$$

$$\Leftrightarrow x_3(t) = I(t-3) \oplus x_1(t-4) \oplus x_2(t-4)$$

並列回路の場合：

$$\Rightarrow x_3(p) = I_2(p-1) \oplus x_1(p-1) \oplus x_2(p-1)$$

$$Q(t) = I(t-1) \oplus x_1(t-1) \oplus x_2(t-1)$$

30

$$\Leftrightarrow Q(t) = I(t-1) \oplus [I(t-2) \oplus x_2(t-2) \oplus x_3(t-2)] \oplus x_1(t-2)$$

$$\Leftrightarrow Q(t) = I(t-1) \oplus I(t-2) \oplus x_1(t-3) \oplus x_2(t-3) \oplus$$

$$[I(t-3) \oplus x_2(t-3) \oplus x_3(t-3)]$$

$$\Leftrightarrow Q(t) = I(t-1) \oplus I(t-2) \oplus I(t-3) \oplus$$

$$[I(t-4) \oplus x_2(t-4) \oplus x_3(t-4)] \oplus x_2(t-4)$$

並列回路の場合：

40

$$\Rightarrow Q_0(p) = I_0(p-1) \oplus I_1(p-1) \oplus I_2(p-1) \oplus I_3(p-1) \oplus x_3(p-1)$$

$$Q(t-1) = I(t-2) \oplus x_1(t-2) \oplus x_2(t-2)$$

$$\Leftrightarrow Q(t-1) = I(t-2) \oplus [I(t-3) \oplus x_2(t-3) \oplus x_3(t-3)] \oplus$$

$$x_1(t-3)$$

$$\Leftrightarrow Q(t-1) = I(t-2) \oplus I(t-3) \oplus x_1(t-4) \oplus x_2(t-4) \oplus$$

$$[I(t-4) \oplus x_2(t-4) \oplus x_3(t-4)]$$

並列回路の場合：

10

$$\Rightarrow Q1(p) = I1(p-1) \oplus I2(p-1) \oplus I3(p-1) \oplus x_1(p-1) \oplus x_3(p-1)$$

$$Q(t-2) = I(t-3) \oplus x_1(t-3) \oplus x_2(t-3)$$

$$\Leftrightarrow Q(t-2) = I(t-3) \oplus [I(t-4) \oplus x_2(t-4) \oplus x_3(t-4)] \oplus$$

$$x_1(t-4)$$

並列回路の場合：

$$\Rightarrow Q2(p) = I2(p-1) \oplus I3(p-1) \oplus x_1(p-1) \oplus x_2(p-1) \oplus x_3(p-1)$$

20

$$Q(t-3) = I(t-4) \oplus x_1(t-4) \oplus x_2(t-4)$$

並列回路の場合：

$$\Rightarrow Q3(p) = I3(p-1) \oplus x_1(p-1) \oplus x_2(p-1)$$

【 0 0 8 9 】

図 6 は、この並列化処理の結果を示す。すなわち、図 4 に示すターボ符号器ブロックを 4 ビット並列化した場合のブロック線図である。

30

【 0 0 9 0 】

[ 3 . 3 例：N = 3 , M = 1、 = [ 0 , 1 , 1 ]、 = [ 1 , 1 , 0 , 1 ] の場合の 8 ビット並列ターボ符号化 ]

式 ( 1 ) より、図 4 に示されるターボ符号器ブロックについての表現として、以下を得ることができる。

【 0 0 9 1 】

$$x_1(t) = I(t-1) \oplus x_2(t-1) \oplus x_3(t-1)$$

$$x_2(t) = x_1(t-1)$$

40

$$x_3(t) = x_2(t-1)$$

$$Q(t) = I(t-1) \oplus x_1(t-1) \oplus x_2(t-1)$$

【 0 0 9 2 】

8 シリアル時間サイクルについての後方時間インデックス遷移が必要である。

【 0 0 9 3 】

$$x_1(t) = I(t-1) \oplus x_2(t-1) \oplus x_3(t-1)$$

$$\Leftrightarrow x_1(t) = I(t-1) \oplus x_1(t-2) \oplus x_2(t-2)$$

$$\Leftrightarrow x_1(t) = I(t-1) \oplus [I(t-3) \oplus x_2(t-3) \oplus x_3(t-3)] \oplus x_1(t-3)$$

$$\Leftrightarrow x_1(t) = I(t-1) \oplus I(t-3) \oplus x_1(t-4) \oplus x_2(t-4) \oplus$$

$$[I(t-4) \oplus x_2(t-4) \oplus x_3(t-4)]$$

$$\Leftrightarrow x_1(t) = I(t-1) \oplus I(t-3) \oplus I(t-4) \oplus$$

10

$$[I(t-5) \oplus x_2(t-5) \oplus x_3(t-5)] \oplus x_2(t-5)$$

$$\Leftrightarrow x_1(t) = I(t-1) \oplus I(t-3) \oplus I(t-4) \oplus I(t-5) \oplus x_2(t-6)$$

$$\Leftrightarrow x_1(t) = I(t-1) \oplus I(t-3) \oplus I(t-4) \oplus I(t-5) \oplus x_1(t-7)$$

$$\Leftrightarrow x_1(t) = I(t-1) \oplus I(t-3) \oplus I(t-4) \oplus I(t-5) \oplus$$

$$[I(t-8) \oplus x_2(t-8) \oplus x_3(t-8)]$$

20

並列回路の場合：

$$\Rightarrow x_1(p) = I_0(p-1) \oplus I_2(p-1) \oplus I_3(p-1) \oplus I_4(p-1) \oplus$$

$$I_7(p-1) \oplus x_2(p-1) \oplus x_3(p-1)$$

$$x_2(t) = x_1(t-1)$$

$$\Leftrightarrow x_2(t) = [I(t-2) \oplus x_2(t-2) \oplus x_3(t-2)]$$

$$\Leftrightarrow x_2(t) = I(t-2) \oplus x_1(t-3) \oplus x_2(t-3)$$

$$\Leftrightarrow x_2(t) = I(t-2) \oplus [I(t-4) \oplus x_2(t-4) \oplus x_3(t-4)] \oplus$$

30

$$x_1(t-4)$$

$$\Leftrightarrow x_2(t) = I(t-2) \oplus I(t-4) \oplus x_1(t-5) \oplus x_2(t-5) \oplus$$

$$[I(t-5) \oplus x_2(t-5) \oplus x_3(t-5)]$$

$$\Leftrightarrow x_2(t) = I(t-2) \oplus I(t-4) \oplus I(t-5) \oplus$$

$$[I(t-6) \oplus x_2(t-6) \oplus x_3(t-6)] \oplus x_2(t-6)$$

$$\Leftrightarrow x_2(t) = I(t-2) \oplus I(t-4) \oplus I(t-5) \oplus I(t-6) \oplus x_2(t-7) \quad 10$$

$$\Leftrightarrow x_2(t) = I(t-2) \oplus I(t-4) \oplus I(t-5) \oplus I(t-6) \oplus x_1(t-8)$$

並列回路の場合：

$$\Rightarrow x_2(p) = I_1(p-1) \oplus I_3(p-1) \oplus I_4(p-1) \oplus I_5(p-1) \oplus x_1(p-1)$$

$$x_3(t) = x_2(t-1)$$

$$\Leftrightarrow x_3(t) = x_1(t-2)$$

$$\Leftrightarrow x_3(t) = [I(t-3) \oplus x_2(t-3) \oplus x_3(t-3)] \quad 20$$

$$\Leftrightarrow x_3(t) = I(t-3) \oplus x_1(t-4) \oplus x_2(t-4)$$

$$\Leftrightarrow x_3(t) = I(t-3) \oplus [I(t-5) \oplus x_2(t-5) \oplus x_3(t-5)] \oplus x_1(t-5)$$

$$\Leftrightarrow x_3(t) = I(t-3) \oplus I(t-5) \oplus x_1(t-6) \oplus x_2(t-6) \oplus$$

$$[I(t-6) \oplus x_2(t-6) \oplus x_3(t-6)]$$

$$\Leftrightarrow x_3(t) = I(t-3) \oplus I(t-5) \oplus I(t-6) \oplus$$

$$[I(t-7) \oplus x_2(t-7) \oplus x_3(t-7)] \oplus x_2(t-7) \quad 30$$

$$\Leftrightarrow x_3(t) = I(t-3) \oplus I(t-5) \oplus I(t-6) \oplus I(t-7) \oplus x_2(t-8)$$

並列回路の場合：

$$\Rightarrow x_3(p) = I_2(p-1) \oplus I_4(p-1) \oplus I_5(p-1) \oplus I_6(p-1) \oplus x_2(p-1)$$

$$Q(t) = I(t-1) \oplus x_1(t-1) \oplus x_2(t-1)$$

$$\Leftrightarrow Q(t) = I(t-1) \oplus [I(t-2) \oplus x_2(t-2) \oplus x_3(t-2)] \oplus x_1(t-2)$$

$$\Leftrightarrow Q(t) = I(t-1) \oplus I(t-2) \oplus x_1(t-3) \oplus x_2(t-3) \oplus \quad 40$$

$$[I(t-3) \oplus x_2(t-3) \oplus x_3(t-3)]$$

$$\Leftrightarrow Q(t) = I(t-1) \oplus I(t-2) \oplus I(t-3) \oplus$$

$$[I(t-4) \oplus x_2(t-4) \oplus x_3(t-4)] \oplus x_2(t-4)$$

$$\Leftrightarrow Q(t) = I(t-1) \oplus I(t-2) \oplus I(t-3) \oplus I(t-4) \oplus x_2(t-5)$$

$$\Leftrightarrow Q(t) = I(t-1) \oplus I(t-2) \oplus I(t-3) \oplus I(t-4) \oplus x_1(t-6)$$

$$\Leftrightarrow Q(t) = I(t-1) \oplus I(t-2) \oplus I(t-3) \oplus I(t-4) \oplus \quad 10$$

$$[I(t-7) \oplus x_2(t-7) \oplus x_3(t-7)]$$

$$\Leftrightarrow Q(t) = I(t-1) \oplus I(t-2) \oplus I(t-3) \oplus I(t-4) \oplus I(t-7) \oplus$$

$$x_1(t-8) \oplus x_2(t-8)$$

並列回路の場合：

$$\Rightarrow Q_0(p) = I_0(p-1) \oplus I_1(p-1) \oplus I_2(p-1) \oplus I_3(p-1) \oplus \quad 20$$

$$I_6(p-1) \oplus x_1(p-1) \oplus x_2(p-1)$$

$$Q(t-1) = I(t-2) \oplus x_1(t-2) \oplus x_2(t-2)$$

$$\Leftrightarrow Q(t-1) = I(t-2) \oplus [I(t-3) \oplus x_2(t-3) \oplus x_3(t-3)] \oplus$$

$$x_1(t-3)$$

$$\Leftrightarrow Q(t-1) = I(t-2) \oplus I(t-3) \oplus x_1(t-4) \oplus x_2(t-4) \oplus \quad 30$$

$$[I(t-4) \oplus x_2(t-4) \oplus x_3(t-4)]$$

$$\Leftrightarrow Q(t-1) = I(t-2) \oplus I(t-3) \oplus I(t-4) \oplus$$

$$[I(t-5) \oplus x_2(t-5) \oplus x_3(t-5)] \oplus x_2(t-5)$$

$$\Leftrightarrow Q(t-1) = I(t-2) \oplus I(t-3) \oplus I(t-4) \oplus I(t-5) \oplus x_2(t-6)$$

$$\Leftrightarrow Q(t-1) = I(t-2) \oplus I(t-3) \oplus I(t-4) \oplus I(t-5) \oplus x_1(t-7)$$

$$\Leftrightarrow Q(t-1) = I(t-2) \oplus I(t-3) \oplus I(t-4) \oplus I(t-5) \oplus \quad 40$$



$$[I(t-8) \oplus x_2(t-8) \oplus x_3(t-8)]$$

並列回路の場合：

$$\Rightarrow Q1(p) = I1(p-1) \oplus I2(p-1) \oplus I3(p-1) \oplus I4(p-1) \oplus$$

$$I7(p-1) \oplus x_2(p-1) \oplus x_3(p-1)$$

$$Q(t-2) = I(t-3) \oplus x_1(t-3) \oplus x_2(t-3)$$

$$\Leftrightarrow Q(t-2) = I(t-3) \oplus [I(t-4) \oplus x_2(t-4) \oplus x_3(t-4)] \oplus \quad 10$$

$$x_1(t-4)$$

$$\Leftrightarrow Q(t-2) = I(t-3) \oplus I(t-4) \oplus x_1(t-5) \oplus x_2(t-5) \oplus$$

$$[I(t-5) \oplus x_2(t-5) \oplus x_3(t-5)]$$

$$\Leftrightarrow Q(t-2) = I(t-3) \oplus I(t-4) \oplus I(t-5) \oplus$$

$$[I(t-6) \oplus x_2(t-6) \oplus x_3(t-6)] \oplus x_2(t-6) \quad 20$$

$$\Leftrightarrow Q(t-2) = I(t-3) \oplus I(t-4) \oplus I(t-5) \oplus I(t-6) \oplus x_2(t-7)$$

$$\Leftrightarrow Q(t-2) = I(t-3) \oplus I(t-4) \oplus I(t-5) \oplus I(t-6) \oplus x_1(t-8)$$

並列回路の場合：

$$\Rightarrow Q2(p) = I2(p-1) \oplus I3(p-1) \oplus I4(p-1) \oplus I5(p-1) \oplus x_1(p-1)$$

$$Q(t-3) = I(t-4) \oplus x_1(t-4) \oplus x_2(t-4)$$

$$\Leftrightarrow Q(t-3) = I(t-4) \oplus [I(t-5) \oplus x_2(t-5) \oplus x_3(t-5)] \oplus \quad 30$$

$$x_1(t-5)$$

$$\Leftrightarrow Q(t-3) = I(t-4) \oplus I(t-5) \oplus x_1(t-6) \oplus x_2(t-6) \oplus$$

$$[I(t-6) \oplus x_2(t-6) \oplus x_3(t-6)]$$

$$\Leftrightarrow Q(t-3) = I(t-4) \oplus I(t-5) \oplus I(t-6) \oplus$$

$$[I(t-7) \oplus x_2(t-7) \oplus x_3(t-7)] \oplus x_2(t-7) \quad 40$$

$$\Leftrightarrow Q(t-3) = I(t-4) \oplus I(t-5) \oplus I(t-6) \oplus I(t-7) \oplus x_2(t-8)$$

並列回路の場合：

$$\Rightarrow Q3(p) = I3(p-1) \oplus I4(p-1) \oplus I5(p-1) \oplus I6(p-1) \oplus x_2(p-1)$$

$$Q(t-4) = I(t-5) \oplus x_1(t-5) \oplus x_2(t-5)$$

$$\Leftrightarrow Q(t-4) = I(t-5) \oplus [I(t-6) \oplus x_2(t-6) \oplus x_3(t-6)] \oplus$$

$$x_1(t-6)$$

$$\Leftrightarrow Q(t-4) = I(t-5) \oplus I(t-6) \oplus x_1(t-7) \oplus x_2(t-7) \oplus$$

10

$$[I(t-7) \oplus x_2(t-7) \oplus x_3(t-7)]$$

$$\Leftrightarrow Q(t-4) = I(t-5) \oplus I(t-6) \oplus I(t-7) \oplus$$

$$[I(t-8) \oplus x_2(t-8) \oplus x_3(t-8)] \oplus x_2(t-8)$$

並列回路の場合：

$$\Rightarrow Q4(p) = I4(p-1) \oplus I5(p-1) \oplus I6(p-1) \oplus I7(p-1) \oplus x_3(p-1)$$

20

$$Q(t-5) = I(t-6) \oplus x_1(t-6) \oplus x_2(t-6)$$

$$\Leftrightarrow Q(t-5) = I(t-6) \oplus [I(t-7) \oplus x_2(t-7) \oplus x_3(t-7)] \oplus$$

$$x_1(t-7)$$

$$\Leftrightarrow Q(t-5) = I(t-6) \oplus I(t-7) \oplus x_1(t-8) \oplus x_2(t-8) \oplus$$

$$[I(t-8) \oplus x_2(t-8) \oplus x_3(t-8)]$$

30

並列回路の場合：

$$\Rightarrow Q5(p) = I5(p-1) \oplus I6(p-1) \oplus I7(p-1) \oplus x_1(p-1) \oplus x_3(p-1)$$

$$Q(t-6) = I(t-7) \oplus x_1(t-7) \oplus x_2(t-7)$$

$$\Leftrightarrow Q(t-6) = I(t-7) \oplus [I(t-8) \oplus x_2(t-8) \oplus x_3(t-8)] \oplus$$

$$x_1(t-8)$$

並列回路の場合：

$$\Rightarrow Q6(p) = I6(p-1) \oplus I7(p-1) \oplus x_1(p-1) \oplus x_2(p-1) \oplus x_3(p-1)$$

40

$$Q(t-7) = I(t-8) \oplus x_1(t-8) \oplus x_2(t-8)$$

並列回路の場合：

$$\Rightarrow Q7(p) = I7(p-1) \oplus x_1(p-1) \oplus x_2(p-1)$$

【 0 0 9 4 】

図 7 A 及び図 7 B は、この並列化処理の結果を示す。すなわち、図 4 に示すターボ符号器ブロックを 8 ビット並列化した場合のブロック線図である。

【 0 0 9 5 】

50

#### [ 4 . V H D L コード ]

上記のステップ・バイ・ステップの再帰的並列化は、異なる代替デザインを比較するために論理合成ツールとともに利用することもできる。いかなる特定のシステムにも制限することなく、ここでは、論理合成ツールを利用した論理回路の設計例として、超高速ハードウェア記述言語 V H D L を典型的な例として引用する。

##### 【 0 0 9 6 】

V H D L コードを利用して、ターボ符号器ブロックを並列化を実現することの利点は、A S I C や F P G A のような特定の技術に拘束されることなく、非常に短時間で設計内容を認識できることである。

##### 【 0 0 9 7 】

このように、実際の技術はターボ符号器ブロック並列化結果を通して規定されないが、符号化のために論理合成ツールにより扱われるパラメータとして設定され、また、異なる設計オプションを比較するために簡単に変更することができる。

##### 【 0 0 9 8 】

以下において、V H D L コードは、図 5、図 6 及び図 7 に示す 2、4 及び 8 ビット並列化ターボ符号器ブロックについてリストされる。V H D L コード言語の使用は、技術として公然知られたものであり、上記の並列化結果との関係は自明であるから、詳細についての説明はここでは行わない。

##### 【 0 0 9 9 】

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_arith.ALL;

ENTITY en_turbo_coder_rtl IS
  PORT(
    -- General:
    reset_p          : IN STD_LOGIC;
    clk32m           : IN STD_LOGIC; -- Clock
                                (rising edge triggered)
    input_8           : IN std_logic_vector(7 DOWNTO 0);
    input_4           : IN std_logic_vector(3 DOWNTO 0);
    input_2           : IN std_logic_vector(1 DOWNTO 0);
    input_1           : IN std_logic;

    -- turboCoding 2 bit parallel
    output_8          : OUT std_logic_vector(7 DOWNTO 0);
    output_4          : OUT std_logic_vector(3 DOWNTO 0);
    output_2          : OUT std_logic_vector(1 DOWNTO 0);
    output_1          : OUT std_logic;
  );
END en_turbo_coder_rtl;

ARCHITECTURE rtl OF en_turbo_coder_rtl IS
```

```

SIGNAL s1_x1          : std_logic;
SIGNAL s1_x2          : std_logic;
SIGNAL s1_x3          : std_logic;
SIGNAL s1_i           : std_logic;
SIGNAL s1_o           : std_logic;

SIGNAL s2_x1          : std_logic;
SIGNAL s2_x2          : std_logic;
SIGNAL s2_x3          : std_logic;
SIGNAL s2_i           : std_logic_vector(1 DOWNTO 0);
SIGNAL s2_o           : std_logic_vector(1 DOWNTO 0);

SIGNAL s4_x1          : std_logic;
SIGNAL s4_x2          : std_logic;
SIGNAL s4_x3          : std_logic;
SIGNAL s4_i           : std_logic_vector(3 DOWNTO 0);
SIGNAL s4_o           : std_logic_vector(3 DOWNTO 0);

SIGNAL s8_x1          : std_logic;
SIGNAL s8_x2          : std_logic;
SIGNAL s8_x3          : std_logic;
SIGNAL s8_i           : std_logic_vector(7 DOWNTO 0);
SIGNAL s8_o           : std_logic_vector(7 DOWNTO 0);

BEGIN

    tc_1: PROCESS (clk32m, reset_p) -- seriell building of
                                     turbo coder block TCB

    BEGIN

        IF reset_p = '1' THEN

```

10

20

30

40

```
s1_x1 <= '0';  
s1_x2 <= '0';  
s1_x3 <= '0';  
s1_i  <= '0';  
s1_o  <= '0';
```

```
ELSIF clk32m'EVENT AND clk32m = '1' THEN
```

10

```
s1_i  <= input_1;  
s1_x1 <= s1_i XOR s1_x2 XOR s1_x3;  
s1_x2 <= s1_x1;  
s1_x3 <= s1_x2;  
s1_o  <= s1_i XOR s1_x2 XOR s1_x1;
```

```
END IF;
```

20

```
END PROCESS tc_1;
```

```
output_1 <= s1_o;
```

```
tc_2: PROCESS (clk32m, reset_p) -- 2bit parallel building  
                                     of turbo coder block
```

30

```
BEGIN
```

```
IF reset_p = '1' THEN
```

```
s2_x1 <= '0';  
s2_x2 <= '0';  
s2_x3 <= '0';  
s2_i  <= (OTHERS => '0');  
s2_o  <= (OTHERS => '0');
```

40

```
ELSIF clk32m'EVENT AND clk32m = '1' THEN
```

```

    s2_i  <= input_2;
    s2_x1 <= s2_i(0) XOR s2_x1 XOR s2_x2;
    s2_x2 <= s2_i(1) XOR s2_x2 XOR s2_x3;
    s2_x3 <= s2_x1;
    s2_o(0) <= s2_i(0) XOR
                s2_i(1) XOR s2_x1 XOR s2_x2 XOR s2_x3;
    s2_o(1) <= s2_i(1) XOR s2_x1 XOR s2_x2;

```

10

```
END IF;
```

```
END PROCESS tc_2;
```

```
output_2 <= s2_o;
```

20

```

tc_4: PROCESS (clk32m, reset_p) -- 4bit parallel building
                                of turbo coder block

```

```
BEGIN
```

```
IF reset_p = '1' THEN
```

30

```

    s4_x1  <= '0';
    s4_x2  <= '0';
    s4_x3  <= '0';
    s4_i    <= (OTHERS => '0');
    s4_o    <= (OTHERS => '0');

```

```
ELSIF clk32m'EVENT AND clk32m = '1' THEN
```

40

```

    s4_i    <= input_4;
    s4_x1    <= s4_i(0) XOR s4_i(2) XOR s4_i(3)

```

```

                                XOR s4_x1 XOR s4_x3;
s4_x2    <= s4_i(1) XOR s4_i(3) XOR s4_x1
                                XOR s4_x2 XOR s4_x3;
s4_x3    <= s4_i(2) XOR s4_x1 XOR s4_x2;
s4_o(0)  <= s4_i(0) XOR s4_i(1) XOR s4_i(2)
                                XOR s4_i(3) XOR s4_x3;
s4_o(1)  <= s4_i(1) XOR s4_i(2) XOR s4_i(3)
                                XOR s4_x1 XOR s4_x3;
s4_o(2)  <= s4_i(2) XOR s4_i(3) XOR s4_x1
                                XOR s4_x2 XOR s4_x3;
s4_o(3)  <= s4_i(3) XOR s4_x1 XOR s4_x2;

                                10

END IF;

                                20
END PROCESS tc_4;

output_4 <= s4_o;

tc_8: PROCESS (clk32m, reset_p) -- 8bit parallel building
                                of turbo coder block
                                30
BEGIN

    IF reset_p = '1' THEN

        s8_x1    <= '0';
        s8_x2    <= '0';
        s8_x3    <= '0';
        s8_i     <= (OTHERS => '0');
        s8_o     <= (OTHERS => '0');
                                40

    ELSIF clk32m'EVENT AND clk32m = '1' THEN

```



```

s8_i    <= input_8;
s8_x1   <= s8_i(0) XOR s8_i(2) XOR s8_i(3)
          XOR s8_i(4) XOR s8_i(7)
          XOR s8_x2 XOR s8_x3;
s8_x2   <= s8_i(1) XOR s8_i(3) XOR s8_i(4)
          XOR s8_i(5) XOR s8_x1;
s8_x3   <= s8_i(2) XOR s8_i(4) XOR s8_i(5)
          XOR s8_i(6) XOR s8_x2;
s8_o(0) <= s8_i(0) XOR s8_i(1) XOR s8_i(2)
          XOR s8_i(3) XOR s8_i(6)
          XOR s8_x1 XOR s8_x2;
s8_o(1) <= s8_i(1) XOR s8_i(2) XOR s8_i(3)
          XOR s8_i(4) XOR s8_i(7)
          XOR s8_x2 XOR s8_x3;
s8_o(2) <= s8_i(2) XOR s8_i(3) XOR s8_i(4)
          XOR s8_i(5) XOR s8_x1;
s8_o(3) <= s8_i(3) XOR s8_i(4) XOR s8_i(5)
          XOR s8_i(6) XOR s8_x2;
s8_o(4) <= s8_i(4) XOR s8_i(5) XOR s8_i(6)
          XOR s8_i(7) XOR s8_x3;
s8_o(5) <= s8_i(5) XOR s8_i(6) XOR s8_i(7)
          XOR s8_x1 XOR s8_x3;
s8_o(6) <= s8_i(6) XOR s8_i(7) XOR s8_x1
          XOR s8_x2 XOR s8_x3;
s8_o(7) <= s8_i(7) XOR s8_x1 XOR s8_x2;

END IF;

END PROCESS tc_8;

output_8 <= s8_o;

END rtl;

```

【 0 1 0 0 】

[ 5 . ターボ符号化の実現 ]

本発明の背景となる技術についての上記記載は図 8 と関連して行われた一方で、図 8 はまた、以下に記載するように本発明にも関連する。

【 0 1 0 1 】

これまで記載した本発明のように、M 個の出力を有するターボ符号器ブロックの並列化について述べる。しかしながら、本発明によれば、完全なターボ符号化の構成は、図 8 に示

10

20

30

40

50

されるような並列化された2つのターボ符号器ブロックの例示を必要とする(ここでは、両方のブロックにおいて $M = 1$ が想定される。 )。

【0102】

あるターボ符号器ブロックは、図8における出力1を導くノーマルな入力データストリームに関連し、他方は、出力2を導くインタリーブされた入力データストリームに関連する。入力データストリームは、出力0へも直接に転送されるので、ターボ符号化の3出力が、パラレルクロックサイクルについて $3n$ ビットの幅を有する。ここで、 $n$ は並列度である。

【図面の簡単な説明】

【図1】 1出力を備えるターボ符号器ブロックのブロック線図である。

10

【図2】 複数出力を備えるターボ符号器ブロックのブロック線図である。

【図3】 並列入力ベクトルへのデータ入力の時間系列ストリームのマッピングを表す図である。

【図4】 シリアル型の特定のターボ符号器ブロックのブロック線図である。

【図5】 図4に示すターボ符号器ブロックを2ビット並列化した場合のブロック線図である。

【図6】 図4に示すターボ符号器ブロックを4ビット並列化した場合のブロック線図である。

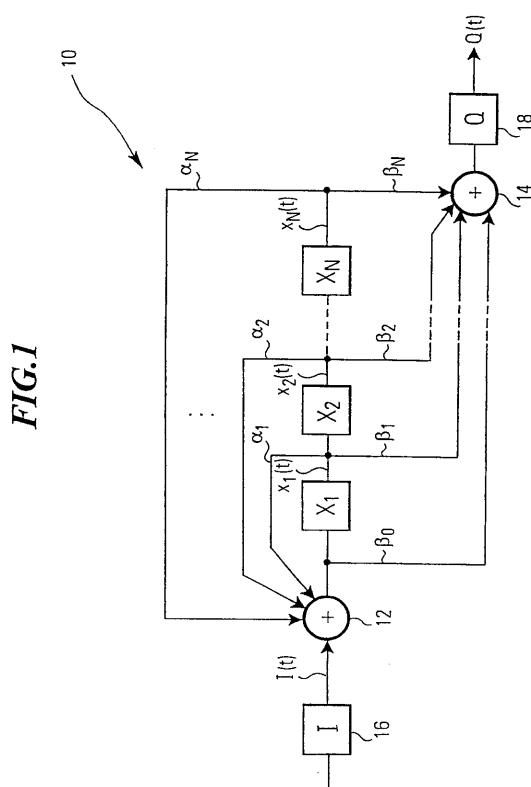
【図7A】 図4に示すターボ符号器ブロックを、入力部を8ビット並列化した場合のブロック線図である。

20

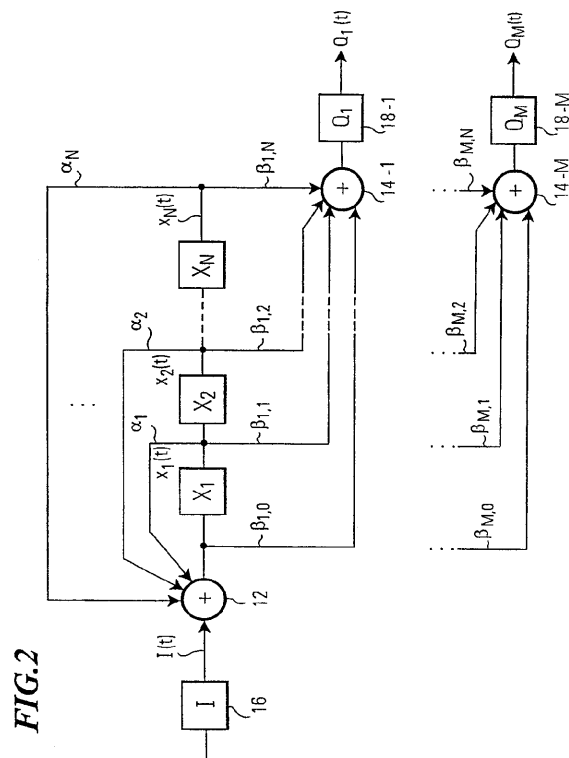
【図7B】 図4に示すターボ符号器ブロックの出力部を並列化した場合のブロック線図である。

【図8】 ターボ符号器ブロックTCB1及びTCB2と同様にインタリーブを備えるターボ符号器の構成図である。

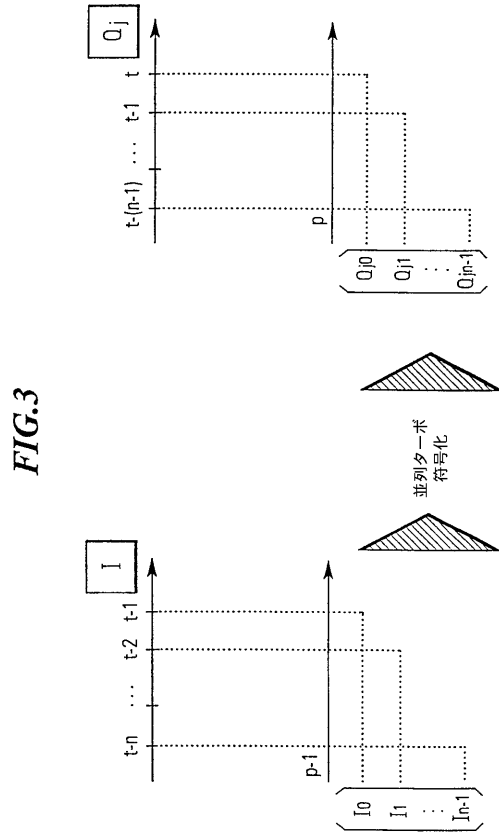
【図1】



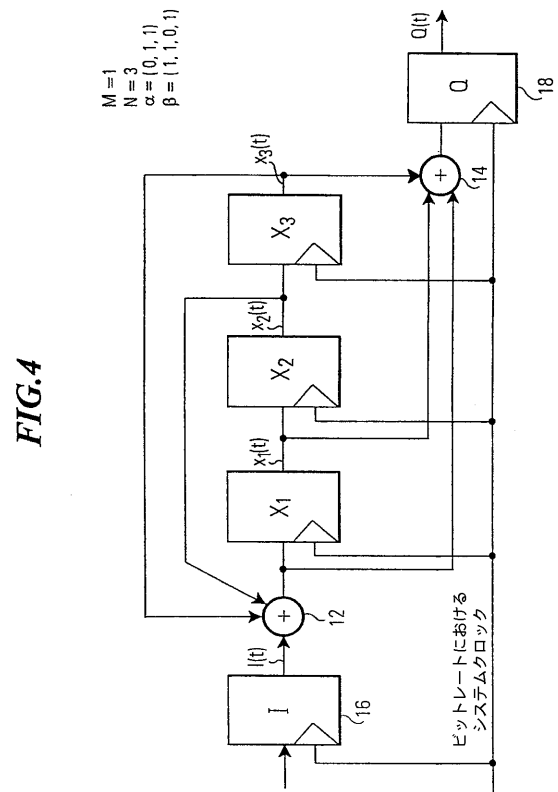
【図2】



【図 3】

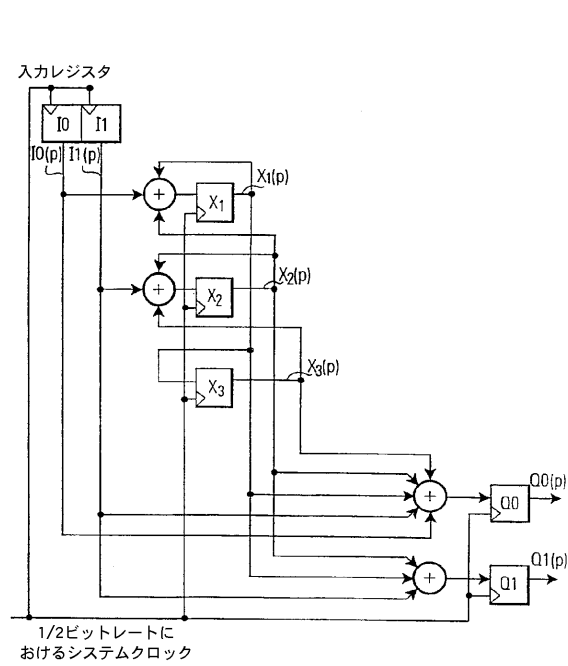


【図 4】



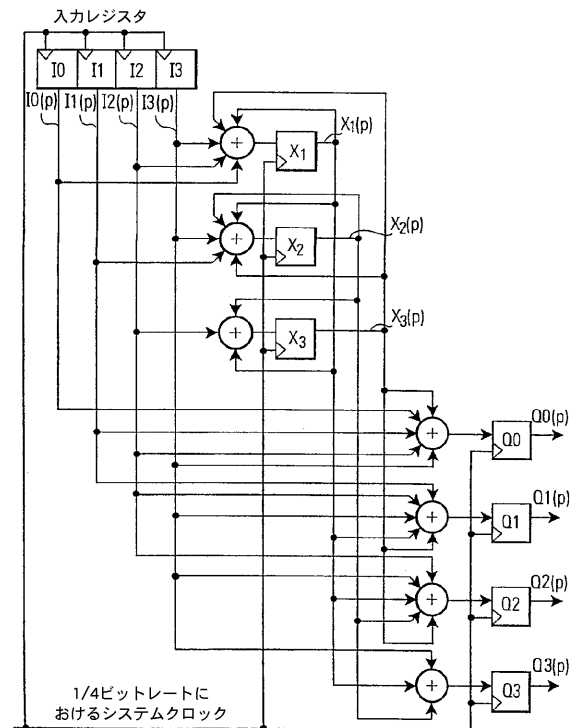
【図 5】

FIG.5



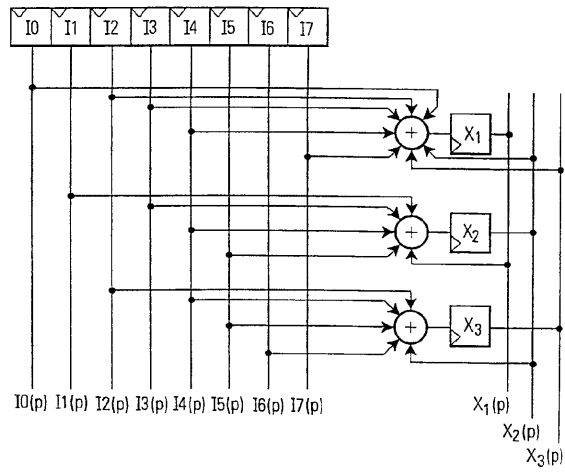
【図 6】

FIG.6



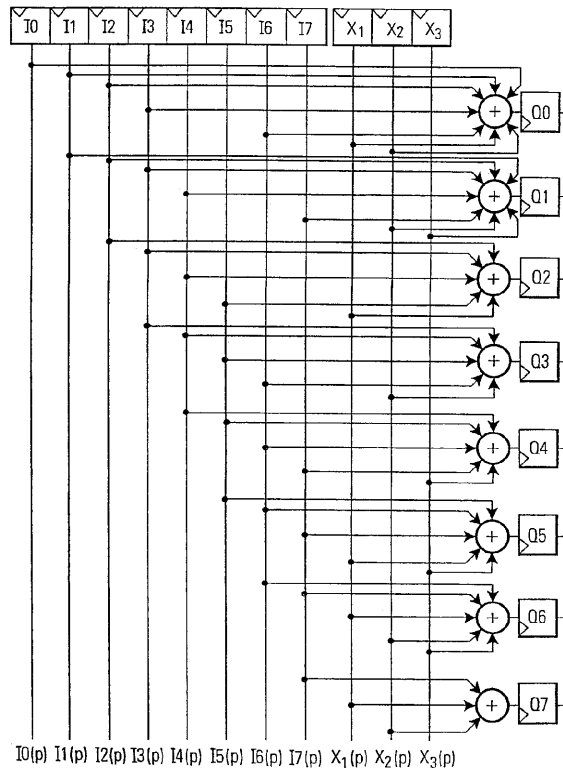
【図 7 A】

FIG. 7A



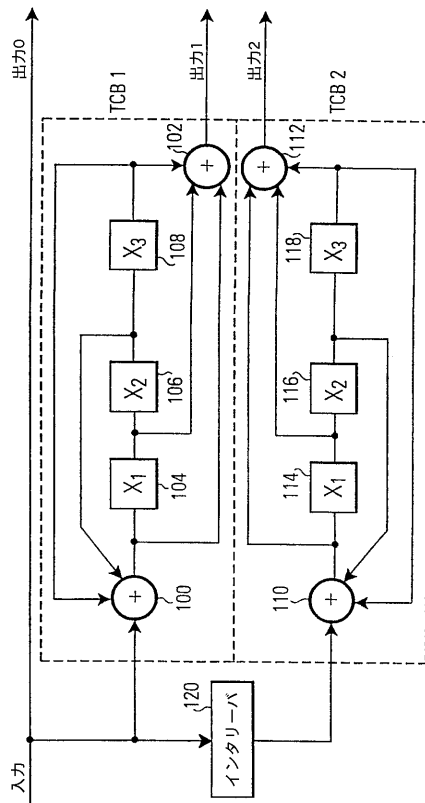
【図 7 B】

FIG. 7B



【図 8】

FIG. 8



---

フロントページの続き

- (72)発明者 ミュルスベルゲル, ゲルド  
ドイツ国 ブベンレウス デー - 9 1 0 8 8 , ヨハネスシュトラーセ 1 0  
(72)発明者 スピュルライン, ゲオルグ  
ドイツ国 ヒルスハイド デー - 9 6 1 1 4 , バンホフシュトラーセ 3 3

審査官 渡辺 未央子

- (56)参考文献 特開昭 6 1 - 2 1 4 6 2 3 ( J P , A )  
特開平 0 7 - 2 5 3 8 9 5 ( J P , A )  
特開 2 0 0 0 - 1 1 4 9 8 6 ( J P , A )

(58)調査した分野(Int.Cl. , D B 名)

H03M 13/29  
G06F 11/10