

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4909498号
(P4909498)

(45) 発行日 平成24年4月4日(2012.4.4)

(24) 登録日 平成24年1月20日(2012.1.20)

(51) Int.Cl.

F 1

HO3M 13/27 (2006.01)
HO3M 13/29 (2006.01)HO3M 13/27
HO3M 13/29

請求項の数 5 (全 14 頁)

(21) 出願番号 特願2004-53923 (P2004-53923)
 (22) 出願日 平成16年2月27日 (2004.2.27)
 (65) 公開番号 特開2005-244770 (P2005-244770A)
 (43) 公開日 平成17年9月8日 (2005.9.8)
 審査請求日 平成19年1月15日 (2007.1.15)
 審判番号 不服2010-21993 (P2010-21993/J1)
 審判請求日 平成22年9月30日 (2010.9.30)

(73) 特許権者 000004237
 日本電気株式会社
 東京都港区芝五丁目7番1号
 (74) 代理人 100123788
 弁理士 宮崎 昭夫
 (74) 代理人 100106138
 弁理士 石橋 政幸
 (74) 代理人 100127454
 弁理士 緒方 雅昭
 (72) 発明者 石田 一博
 東京都港区芝五丁目7番1号 日本電気株式会社内

最終頁に続く

(54) 【発明の名称】インターリーブパラメータ演算方法／プログラム／プログラム記録媒体／装置、携帯電話機

(57) 【特許請求の範囲】

【請求項 1】

IMT2000 の標準規格である 3GPP TS 25.212 で規定されるターボ符号のインターリーバ装置に対して、前記インターリーバ装置が送信データを符号化するのに必要な行内入れ替えパターン $U[i][j]$ と、行間入れ替えパターン $T[i]$ を出力するインターリーブパラメータ演算装置が、前記送信データのデータ長から求めた、前記行間入れ替えパターン $T[i]$ と、素数 p と、ベースシーケンス $s[j]$ と、素数列 $q[i]$ をもとに前記行内入れ替えパターン $U[i][j]$ を計算する過程で、中間値として求めるべき $y[i] = q[i] \times m \bmod (p - 1)$ を計算するインターリーブパラメータ演算方法において、

前記インターリーブパラメータ演算装置が、 i と変数 d_{iv} を 0 に初期化する第 1 のステップと、10

前記インターリーブパラメータ演算装置が、 $q[i] \leq d_{iv} + p - 1$ の場合、 d_{iv} の値に $p - 1$ を加算して保存し、 $q[i] < d_{iv} + p - 1$ の場合、 d_{iv} の値をそのまま保存する第 2 のステップと、

前記インターリーブパラメータ演算装置が、前記第 2 のステップで保存された d_{iv} の値をもとに、 $y[i] = q[i] - d_{iv}$ から、 $y[i]$ を求める第 3 のステップを有し、

前記インターリーブパラメータ演算装置が、 i がインターリーバの行数 R になるまで i をインクリメントして、前記第 2 のステップと前記第 3 のステップを繰り返し、すべての i に対して、 $y[i]$ を求めることを特徴とするインターリーブパラメータ演算方法。

【請求項 2】

請求項 1 に記載の前記インターリープパラメータ演算方法を、前記インターリープパラメータ演算装置となるべきコンピュータに実行させるプログラム。

【請求項 3】

請求項 1 に記載の前記インターリープパラメータ演算方法を、前記インターリープパラメータ演算装置となるべきコンピュータに実行させるプログラムを記録した、コンピュータ読み取りが可能な記録媒体。

【請求項 4】

IMT2000 の標準規格である 3GPP TS 25.212 で規定されるターボ符号のインターリーバ装置に対して、前記インターリーバ装置が送信データを符号化するのに必要な行内入れ替えパターン $U[i][j]$ と、行間入れ替えパターン $T[i]$ を出力するインターリープパラメータ演算装置であって、前記送信データのデータ長から求めた、前記行間入れ替えパターン $T[i]$ と、素数 p と、ベースシーケンス $s[j]$ と、素数列 $q[i]$ をもとに前記行内入れ替えパターン $U[i][j]$ を計算する過程で、中間値として求めるべき $y[i] = q[i] \times m o d(p - 1)$ を計算するインターリープパラメータ演算装置において

、
i と変数 $d_i v$ を 0 に初期化する第 1 の手段と、

$q[i] \leq d_i v + p - 1$ の場合、 $d_i v$ の値に $p - 1$ を加算して保存し、 $q[i] < d_i v + p - 1$ の場合、 $d_i v$ の値をそのまま保存する第 2 の手段と、

前記第 2 の手段が保存した $d_i v$ の値をもとに、 $y[i] = q[i] - d_i v$ から、 $y[i]$ を求める第 3 の手段と、

i がインターリーバの行数 R になるまで i をインクリメントして、前記第 2 の手段と前記第 3 の手段に処理を繰り返させて、すべての i に対して、 $y[i]$ を求める第 4 の手段を有することを特徴とするインターリープパラメータ演算装置。

【請求項 5】

請求項 4 に記載の前記インターリープパラメータ演算装置を有する携帯電話機。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、第 3 世代の移動通信システム (IMT2000) の無線アクセス方式 (RAN) の 1 つとして標準化されている広域 DS - CDMA (W-CDMA) において規格化されている、素数インターリーバと呼ばれるターボコーディング用の内部インターリーバに関する。

【背景技術】

【0002】

広域 DS - CDMA (W-CDMA) は、第 3 世代の移動通信システム (IMT2000) の無線アクセス方式 (RAN) の 1 つとして標準化され、その中で、素数インターリーバと呼ばれるターボコーディング用の内部インターリーバが規格化されている。その詳細な説明が、「3rd Generation Partnership Project ; Technical Specification Group Radio Access Network ; Multiplexing And Channel Coding (FDD) (Release 1999) 3GPP TS 25.212 V3.10.0 (2002-06) 4.2.3.2.3 章 Turbo Code Internal Interleaver 16 頁～19 頁」(非特許文献 1) に開示されている。以下の説明は、非特許文献 1 の記載にもとづく。

【0003】

IMT2000 (W-CDMA) の標準規格 3GPP TS 25.212 で規定されるターボ符号のインターリーバにおいて、送信するデータを符号化するのに必要不可欠な行内の入れ替えパターン $U[i][j]$ は、素数 p 、ベースシーケンス $s[j]$ 、後述のように定義される $r[i]$ を用いて、次のように定義されている。

【0004】

10

20

30

40

50

(数1)

$$U[i][j] = s[(j \times r[i]) \times \text{mod}(p - 1)] = s[x[i][j]]$$

ただし、 $x[i][j] = (j \times r[i]) \times \text{mod}(p - 1)$ 。

【0005】

ここで、 $r[i]$ は、行間の入れ替えパターン $T[i]$ と、後述のように定義される素数列 $q[i]$ を用いて、次のように定義されている。

【0006】

(数2)

$$r[T[i]] = q[i].$$

【0007】

素数列 $q[i]$ は、インターリーバの行数 R 、素数 p を用いて、次のように定義されている。

【0008】

(数3)

$$q[0] = 1$$

$q[i] (i = 1, 2, \dots, R - 1)$ は、 $\text{g.c.d}(q[i], p - 1) = 1$ かつ $q[i] > 6$ かつ $q[i] > q[i - 1]$ を満たす最小の素数。

ただし、 $\text{g.c.d}(x, y)$ は、 x と y の最大公約数。

【0009】

表1を参照すると、インターリーバの行数 $R = 20$ のときの、素数 p の各値に対する素数列 $q[i]$ の値が例示されている。

【0010】

10

20

【表 1 A】

p	23	29	43	47	53	59	67	79	83	103	107	113	127	131
q[0]	1	1	1	1	1	1	1	1	1	1	1	1	1	1
q[1]	7	11	11	7	7	7	7	7	7	7	7	11	11	7
q[2]	13	13	13	11	11	11	13	11	11	11	11	13	13	11
q[3]	17	17	17	13	17	13	17	17	13	13	13	17	17	17
q[4]	19	19	19	17	19	17	19	19	17	19	17	19	19	19
q[5]	23	23	23	19	23	19	23	23	19	23	19	23	23	23
q[6]	29	29	29	29	29	23	29	29	23	29	23	29	29	29
q[7]	31	31	31	31	31	31	31	31	29	31	29	31	31	31
q[8]	37	37	37	37	37	37	37	37	31	37	31	37	37	37
q[9]	41	41	41	41	41	41	41	41	37	41	37	41	41	41
q[10]	43	43	43	43	43	43	43	43	43	43	41	43	43	43
q[11]	47	47	47	47	47	47	47	47	47	47	43	47	47	47
q[12]	53	53	53	53	53	53	53	53	53	53	47	53	53	53
q[13]	59	59	59	59	59	59	59	59	59	59	59	59	59	59
q[14]	61	61	61	61	61	61	61	61	61	61	61	61	61	61
q[15]	67	67	67	67	67	67	67	67	67	67	67	67	67	67
q[16]	71	71	71	71	71	71	71	71	71	71	71	71	71	71
q[17]	73	73	73	73	73	73	73	73	73	73	73	73	73	73
q[18]	79	79	79	79	79	79	79	79	79	79	79	79	79	79
q[19]	83	83	83	83	83	83	83	83	83	83	83	83	83	83

10

20

30

40

【0 0 1 1】

【表 1 B】

p	137	139	149	157	173	191	197	199	211	223	229	233	239	etc
q[0]	1	1	1	1	1	1	1	1	1	1	1	1	1	1
q[1]	7	7	7	7	7	7	11	7	11	7	7	7	11	7
q[2]	11	11	11	11	11	11	13	13	13	11	11	11	13	11
q[3]	13	13	13	17	13	13	17	17	17	13	13	13	19	13
q[4]	19	17	17	19	17	17	19	19	19	17	17	17	23	17
q[5]	23	19	19	23	19	23	23	23	23	19	23	19	29	19
q[6]	29	29	23	29	23	29	29	29	29	23	29	23	31	23
q[7]	31	31	29	31	29	31	31	31	31	29	31	31	37	29
q[8]	37	37	31	37	31	37	37	37	37	31	37	37	41	31
q[9]	41	41	41	41	37	41	41	41	41	41	41	41	43	37
q[10]	43	43	43	43	41	43	43	43	43	43	43	43	47	41
q[11]	47	47	47	47	47	47	47	47	47	47	47	47	53	43
q[12]	53	53	53	53	53	53	53	53	53	53	53	53	59	47
q[13]	59	59	59	59	59	59	59	59	59	59	59	59	61	53
q[14]	61	61	61	61	61	61	61	61	61	61	61	61	67	59
q[15]	67	67	67	67	67	67	67	67	67	67	67	67	71	61
q[16]	71	71	71	71	71	71	71	71	71	71	71	71	73	67
q[17]	73	73	73	73	73	73	73	73	73	73	73	73	79	71
q[18]	79	79	79	79	79	79	79	79	79	79	79	79	83	73
q[19]	83	83	83	83	83	83	83	83	83	83	83	83	89	79

以上のように、行内の入れ替えパターン $U[i][j]$ は、素数 p 、ベースシーケンス $s[j]$ 、行間の入れ替えパターン $T[i]$ 、素数列 $q[i]$ を用いて計算されることになる。

【0012】

IMT2000 における携帯電話機などでは、まず、送信を行うデータのデータ長が求められる。そして、求められたデータ長（または、送信を行うデータを格納する行列の大きさ）にしたがって、素数 p 、ベースシーケンス $s[j]$ 、行間の入れ替えパターン $T[i]$ 、素数列 $q[i]$ が求められ、これらをもとに、行内の入れ替えパターン $U[i][j]$ が計算される。そして、計算された行内の入れ替えパターン $U[i][j]$ と、行間の入れ替えパ

10

20

30

40

50

ーン $T[i]$ をもとに、送信データが符号化され、符号化された送信データは携帯電話網に送出される。この符号化されたデータを受信した別の携帯電話機は、同じく計算された行内の入れ替えパターン $U[i][j]$ と、行間の入れ替えパターン $T[i]$ をもとに、符号化されている受信データを復号化する。

【0013】

素数 p 、ベースシーケンス $s[j]$ 、素数列 $q[i]$ は、送信を行うデータのデータ長（または、送信を行うデータを格納する行列の大きさ）から、一義的に求められるようになっている。素数列 $q[i]$ に関しては、（数3）および[表1]で示した。また、ベースシーケンス $s[j]$ の効率的な計算方法が、本出願人により、すでに提出されている（特願2003-298493）。また、行間の入れ替えパターン $T[i]$ には、4つの固定化されたパターンが存在し、いずれか1つが選択される。
10

【0014】

以上のように、行内の入れ替えパターン $U[i][j]$ は、素数 p 、ベースシーケンス $s[j]$ 、行間の入れ替えパターン $T[i]$ 、素数列 $q[i]$ を用いて計算されるが、計算の簡略化のため、行内の入れ替えパターン $U[i][j]$ を求めるための中間値として、次のように定義される $y[i]$ を求めたうえで、行内の入れ替えパターン $U[i][j]$ を計算することが通常である。

【0015】

(数4)

$$y[i] = w[i][1] = q[i] \times \text{mod}(p - 1) .$$

20

【0016】

ここで、 $w[i][j]$ は、次のように定義される。

【0017】

(数5)

$$w[i][j] = \times [T[i]][j] = (j \times r[T[i]]) \times \text{mod}(p - 1) = (j \times q[i]) \times \text{mod}(p - 1) .$$

【0018】

以下、具体例を用いて、従来技術による、行内の入れ替えパターン $U[i][j]$ の計算方法を説明する。

【0019】

30

まず、送信を行うデータのデータ長から、素数 p 、ベースシーケンス $s[j]$ 、行間の入れ替えパターン $T[i]$ 、素数列 $q[i]$ が求められる。例えば、次のような、 p 、 $s[j]$ 、 $T[i]$ 、 $q[i]$ が得られたとする。

【0020】

(設定パラメータ)

$$p = 7$$

$$s[j] = [1, 3, 2, 6, 4, 5] (j = 0 \sim 5)$$

$$T[i] = [4, 3, 2, 1, 0] (i = 0 \sim 4) (R = 5)$$

$$q[i] = [1, 7, 11, 13, 17] (i = 0 \sim 4) (R = 5) .$$

【0021】

40

まず、中間値 $y[i]$ が、（数4）、 p 、 $q[i]$ をもとに、次のように計算される。図4を参照すると、 $y[i]$ を求めるためのフローチャートが示されている。

【0022】

(計算プロセス1)

$$y[0] = q[0] \times \text{mod}(7 - 1) = 1 \times \text{mod} 6 = 1$$

$$y[1] = q[1] \times \text{mod}(7 - 1) = 7 \times \text{mod} 6 = 1$$

$$y[2] = q[2] \times \text{mod}(7 - 1) = 11 \times \text{mod} 6 = 5$$

$$y[3] = q[3] \times \text{mod}(7 - 1) = 13 \times \text{mod} 6 = 1$$

$$y[4] = q[4] \times \text{mod}(7 - 1) = 17 \times \text{mod} 6 = 5 .$$

【0023】

50

次に、 $x[i][1]$ が、(計算プロセス1)で求められた $y[i]$ 、 $T[i]$ 、(数4)、(数5)をもとに、次のように求められる。

【0024】

(計算プロセス2)

$$\begin{aligned}y[i] &= w[i][1] = x[T[i]][1] \\y[0] &= w[0][1] = x[T[0]][1] = x[4][1] = 1 \\y[1] &= w[1][1] = x[T[1]][1] = x[3][1] = 1 \\y[2] &= w[2][1] = x[T[2]][1] = x[2][1] = 5 \\y[3] &= w[3][1] = x[T[3]][1] = x[1][1] = 1 \\y[4] &= w[4][1] = x[T[4]][1] = x[0][1] = 5\end{aligned}$$

よって、 $x[i][1] = [5, 1, 5, 1, 1]$ ($i = 0 \sim 4$) ($R = 5$)。

【0025】

次に、 $x[i][j]$ が、(計算プロセス2)で求められた $x[i][1]$ 、 p をもとに、次のように(数1)を参考にして、剰余演算を行わずに求められる。

【0026】

(計算プロセス3)

$$\begin{aligned}x[0][0] &= 0 \\x[0][1] &= 5 \\x[0][2] &= x[0][1] + x[0][1] - (p - 1) = 5 + 5 - 6 = 4 \\x[0][3] &= x[0][2] + x[0][1] - (p - 1) = 4 + 5 - 6 = 3 \\x[0][4] &= x[0][3] + x[0][1] - (p - 1) = 3 + 5 - 6 = 2 \\x[0][5] &= x[0][4] + x[0][1] - (p - 1) = 2 + 5 - 6 = 1 \\(\times[0][1] + \times[0][1], \times[0][2] + \times[0][1], \times[0][3] + \times[0][1], \times[0][4] + \times[0][1]) &\text{が } p - 1 \text{ より大きいので、 } p - 1 \text{ だけ減算される。} \\x[1][0] &= 0 \\x[1][1] &= 1 \\x[1][2] &= x[1][1] + x[1][1] = 1 + 1 = 2 \\x[1][3] &= x[1][2] + x[1][1] = 2 + 1 = 3 \\x[1][4] &= x[1][3] + x[1][1] = 3 + 1 = 4 \\x[1][5] &= x[1][4] + x[1][1] = 4 + 1 = 5 \\(\times[0][1] + \times[0][1], \times[0][2] + \times[0][1], \times[0][3] + \times[0][1], \times[0][4] + \times[0][1]) &\text{が } p - 1 \text{ より小さいので、そのまま出力される。} \\x[2][0] &= 0 \\x[2][1] &= 5 \\x[2][2] &= x[2][1] + x[2][1] - (p - 1) = 5 + 5 - 6 = 4 \\x[2][3] &= x[2][2] + x[2][1] - (p - 1) = 4 + 5 - 6 = 3 \\x[2][4] &= x[2][3] + x[2][1] - (p - 1) = 3 + 5 - 6 = 2 \\x[2][5] &= x[2][4] + x[2][1] - (p - 1) = 2 + 5 - 6 = 1 \\(\times[0][1] + \times[0][1], \times[0][2] + \times[0][1], \times[0][3] + \times[0][1], \times[0][4] + \times[0][1]) &\text{が } p - 1 \text{ より大きいので、 } p - 1 \text{ だけ減算される。} \\x[3][0] &= 0 \\x[3][1] &= 1 \\x[3][2] &= x[3][1] + x[3][1] = 1 + 1 = 2 \\x[3][3] &= x[3][2] + x[3][1] = 2 + 1 = 3 \\x[3][4] &= x[3][3] + x[3][1] = 3 + 1 = 4 \\x[3][5] &= x[3][4] + x[3][1] = 4 + 1 = 5 \\(\times[0][1] + \times[0][1], \times[0][2] + \times[0][1], \times[0][3] + \times[0][1], \times[0][4] + \times[0][1]) &\text{が } p - 1 \text{ より小さいので、そのまま出力される。} \\x[4][0] &= 0 \\x[4][1] &= 5\end{aligned}$$

10

20

30

40

50

$x[4][2] = x[4][1] + x[4][1] - (p - 1) = 5 + 5 - 6 = 4$
 $x[4][3] = x[4][2] + x[4][1] - (p - 1) = 4 + 5 - 6 = 3$
 $x[4][4] = x[4][3] + x[4][1] - (p - 1) = 3 + 5 - 6 = 2$
 $x[4][5] = x[4][4] + x[4][1] - (p - 1) = 2 + 5 - 6 = 1$
 $(x[0][1] + x[0][1], x[0][2] + x[0][1], x[0][3] + x[0][1], x[0][4] + x[0][1]$
 が $p - 1$ より大きいので、 $p - 1$ だけ減算される)。

【0027】

最後に、 $x[i][j]$ 、 $s[i]$ 、(数1)をもとに、 $U[i][j]$ が求められる。

【0028】

(計算プロセス4)

$U[0][0] = s[x[0][0]] = s[0] = 1$
 $U[0][1] = s[x[0][1]] = s[5] = 5$
 $U[0][2] = s[x[0][2]] = s[4] = 4$
 $U[0][3] = s[x[0][3]] = s[3] = 6$
 $U[0][4] = s[x[0][4]] = s[2] = 2$
 $U[0][5] = s[x[0][5]] = s[1] = 3$
 $U[1][0] = s[x[1][0]] = s[0] = 1$
 $U[1][1] = s[x[1][1]] = s[1] = 3$
 $U[1][2] = s[x[1][2]] = s[2] = 2$
 $U[1][3] = s[x[1][3]] = s[3] = 6$
 $U[1][4] = s[x[1][4]] = s[4] = 4$
 $U[1][5] = s[x[1][5]] = s[5] = 5$
 $U[2][0] = s[x[2][0]] = s[0] = 1$
 $U[2][1] = s[x[2][1]] = s[5] = 5$
 $U[2][2] = s[x[2][2]] = s[4] = 4$
 $U[2][3] = s[x[2][3]] = s[3] = 6$
 $U[2][4] = s[x[2][4]] = s[2] = 2$
 $U[2][5] = s[x[2][5]] = s[1] = 3$
 $U[3][0] = s[x[3][0]] = s[0] = 1$
 $U[3][1] = s[x[3][1]] = s[1] = 3$
 $U[3][2] = s[x[3][2]] = s[2] = 2$
 $U[3][3] = s[x[3][3]] = s[3] = 6$
 $U[3][4] = s[x[3][4]] = s[4] = 4$
 $U[3][5] = s[x[3][5]] = s[5] = 5$
 $U[4][0] = s[x[4][0]] = s[0] = 1$
 $U[4][1] = s[x[4][1]] = s[5] = 5$
 $U[4][2] = s[x[4][2]] = s[4] = 4$
 $U[4][3] = s[x[4][3]] = s[3] = 6$
 $U[4][4] = s[x[4][4]] = s[2] = 2$
 $U[4][5] = s[x[4][5]] = s[1] = 3$ 。

【非特許文献1】「3rd Generation Partnership Project ; Technical Specification Group Radio Access Network ; Multiplexing And Channel Coding (FDD) (Release 1999) 3GPP TS 25.212 V3.10.0 (2002-06) 4.2.3.2.3章 Turbo Code Internal Interleaver 16頁～19頁」

【発明の開示】

【発明が解決しようとする課題】

【0029】

携帯電話機などに使用されるプロセッサでは、その回路規模が抑えられているため、剩

10

20

30

40

50

余演算を1桁ずつ行うものが多く、プロセッサのデータ幅が16 bitの場合、(計算プロセス1)の $y[i]$ の各行の演算(図4のステップ43)には16クロックを要することになる。さらに、図4の条件分岐(ステップ45)に2クロック、その以外の処理(ステップ44)に1クロックを要するとすると、図4のステップ43からステップ45の1回の繰り返し当たり、19クロックを要する。その結果、(計算プロセス1)には、合計で($19 \times R$)クロック(R の最大値は20なので、最大380クロック)を要することになる。

【0030】

本発明の目的は、(計算プロセス1)の $y[i]$ を求めるのに要するクロック数を大幅に減らし、(計算プロセス1)の処理時間を削減することにある。

10

【課題を解決するための手段】

【0031】

上記目的を達成するために、本発明は、(計算プロセス1)の $y[i]$ を求める際に剩余演算を行わずに、代わりに加算と減算のみで、 $y[i]$ を計算する。具体的には、以下のように計算する。

【0032】

[表1]を参照すると、すべての p において、 $q[i+1] - q[i] < p - 1$ なので、

(数6)

$$z[i] = f\ l\ o\ o\ r (q[i] / (p - 1))$$

20

ただし、 $f\ l\ o\ o\ r (q[i] / (p - 1))$ は、 $q[i] / (p - 1)$ の商

と、 $z[i]$ を定義すると、

(数7)

$$z[i] \times (p - 1) \leq q[i] < (z[i] + 1) \times (p - 1)$$

$$z[i] \times (p - 1) \leq q[i + 1] < q[i] + p - 1 < (z[i] + 2) \times (p - 1)$$

となる。よって、 $z[i]$ は、

30

(数8)

$$z[0] = 0$$

$$q[i] < z[i - 1] + p - 1 \text{ の場合、 } z[i] = z[i - 1]$$

$$q[i] \geq z[i - 1] + p - 1 \text{ の場合、 } z[i] = z[i - 1] + p - 1$$

ただし、 $i = 1, \dots, R - 1$

で表される。これを、 $i = 1$ から順に計算することで、剩余演算を行わずに、加算と減算のみで、 $z[i]$ を求めることができる。そして、

(数9)

$$y[i] = q[i] - z[i]$$

40

により、剩余演算を行わずに、加算と減算のみで、 $y[i]$ を求めることができる。

【発明の効果】

【0033】

以上説明したように、本発明によれば、 $y[i]$ を求めるために、クロック数を多く要する剩余演算を行わなくてもよいので、プロセッサの動作時間を削減することができる。また、プロセッサの動作時間の削減により、プロセッサの消費電力を削減することができる。また、剩余演算器を用いて装置を構成していた場合と比べて、剩余演算器が不要となることで、その分、装置の回路規模を削減できる。また、装置の回路規模の削減により、装

50

置の消費電力が削減できる。

【発明を実施するための最良の形態】

【0034】

次に、本発明の実施の形態について、図面を参照して詳細に説明する。

【0035】

本発明の $y[i]$ の計算方法は、[課題を解決するための手段]で説明した通りであるが、ここでは、具体例を用いて説明する。

【0036】

ここでも、[背景技術]で用いた（設定パラメータ）を用いる。（設定パラメータ）、（数8）、（数9）をもとに、[背景技術]の（計算プロセス1）に対応する $y[i]$ の計算を行う。 10

【0037】

まず、（数8）、 $q[i]$ 、 p をもとに、 $z[i]$ が計算される。

【0038】

（ $z[i]$ の計算）

$$\begin{aligned} z[0] &= 0 \\ 7 = q[1] - z[0] + p - 1 &= 6 \text{ なので、 } z[1] = z[0] + p - 1 = 6 \\ 11 = q[2] - z[1] + p - 1 &= 12 \text{ なので、 } z[2] = z[1] = 6 \\ 13 = q[3] - z[2] + p - 1 &= 12 \text{ なので、 } z[3] = z[2] + p - 1 = 12 \\ 17 = q[4] - z[3] + p - 1 &= 18 \text{ なので、 } z[4] = z[3] = 12. \end{aligned} \quad 20$$

【0039】

次に、（ $z[i]$ の計算）で求められた $z[i]$ 、 $q[i]$ 、（数9）をもとに、 $y[i]$ が求められる。

【0040】

（ $y[i]$ の計算）

$$\begin{aligned} y[0] = q[0] - z[0] &= 1 - 0 = 1 \\ y[1] = q[1] - z[1] &= 7 - 6 = 1 \\ y[2] = q[2] - z[2] &= 11 - 6 = 5 \\ y[3] = q[3] - z[3] &= 13 - 12 = 1 \\ y[4] = q[4] - z[4] &= 17 - 12 = 5. \end{aligned} \quad 30$$

【0041】

得られた結果は、[背景技術]の（計算プロセス1）で得られた結果と一致している。

【0042】

以降、 $U[i][j]$ を求めるために、[背景技術]の（計算プロセス2）から（計算プロセス4）と全く同じ計算を行う。（計算プロセス2）から（計算プロセス4）には、剩余演算は含まれていない。

【0043】

以下、この $y[i]$ の計算方法を採用したインターリーブパラメータ演算装置について、詳細に説明する。

【実施例】

【0044】

図1を参照すると、本発明の一実施例の構成が示されている。1は、インターリーブパラメータ演算装置1であり、プロセッサ11と、 $q[i]$ メモリ12と、 $y[i]$ メモリ14を含む。2は、インターリーバである。インターリーブパラメータ演算装置1と、インターリーバ2は、携帯電話機などに搭載される。

【0045】

インターリーブパラメータ演算装置1は、送信を行うデータを受信し、そのデータ長を求める。プロセッサ11が、求められたデータ長から、 p 、 $s[j]$ 、 $T[i]$ 、 $q[i]$ 、次いで中間値 $y[i]$ 、次いで $U[i][j]$ を計算する。プロセッサ11は、計算した $q[i]$ 、 $y[i]$ を、それぞれ $q[i]$ メモリ12、 $y[i]$ メモリ14に格納する。そして、インター 50

リープパラメータ演算装置1は、インターリーバ2に、U[i][j]、T[i]を出力する。

【0046】

インターリーバ2は、送信を行うデータを受信し、インターリープパラメータ演算装置1から受信したU[i][j]、T[i]をもとに、送信を行うデータを符号化し、これを出力する。

【0047】

図2を参照すると、プロセッサ11が行う計算のうち、上述した本発明のy[i]の計算をルーチン化したフローチャートが示されている。

【0048】

ステップ32で、i、divを0に初期化する。ここで、変数divは、z[i]に対応する。ステップ33で、(数8)に対応する判定を行い、q[i] < div + p - 1の場合、ステップ34で、divの値にp - 1を加算する。q[i] < div + p - 1の場合、そのままステップ35に進む。ステップ35で、y[i] = q[i] - divから、y[i]を求める。ステップ36で、iをインクリメントし、iがRになるまで、ステップ33からステップ36の処理を繰り返す。
10

【0049】

ここで、条件分岐(ステップ33、37)は2クロック、それ以外の処理(ステップ34、35、36)は1クロックに相当するので、ステップ33からステップ37の1回の繰り返し当たり、7クロックを要する。これは、[発明が解決しようとする課題]で検討したように、従来の方法が、1回の繰り返し当たり19クロックを要していたので、結局、12クロック分の処理量削減効果が得られたことになる。Rの最大値が20なので、最大240クロックの処理量削減が可能となる。
20

【0050】

図3を参照すると、インターリープパラメータ演算装置1の実装例が示されている。

【0051】

12は、q[i]の値を格納するメモリである。72は、-(p - 1)の値を有する定数ブロックである。73は、-divの値を有するレジスタである。74は、メモリアドレスを生成するカウンタである。14は、y[i]の値を格納するメモリである。76は、メモリ12のアドレスiの値を選択するセレクタである。77は、q[i] - div、または、q[i] - (div + p - 1)の値で出力を選択するセレクタである。78は、-(div + p - 1)、または、-divの値で出力を選択するセレクタである。79は、メモリ14の格納先を選択するセレクタである。
30

【0052】

次に、この実装例の動作を説明する。初期状態では、カウンタ74、レジスタ73の値i、divは、それぞれ「0」、「0」を保持している。セレクタ76により、メモリ12のアドレスiのデータq[i]が選択される。q[i]と、レジスタ73の値 - divと、定数ブロック72の値 -(p - 1)より、q[i] - (div + p - 1)が求められる。q[i] - (div + p - 1) < 0の場合、セレクタ77、78は、それぞれq[i] - div、-divを選択し、それ以外の場合は、それぞれq[i] - (div + p - 1)、-(div + p - 1)を選択する。セレクタ77の選択値はメモリ14のアドレスiに格納され、セレクタ78の選択値はレジスタ73に格納される。以上の処理が、カウンタ74の値iがR - 1になるまで繰り返される。以上により、メモリ14にy[i]が生成される。
40

【0053】

なお、インターリープパラメータ演算装置1は、専用のハードウェアにより実現する以外に、その機能を実現するためのプログラムを、コンピュータ読み取りが可能な記録媒体に記録して、この記録媒体に記録されたプログラムを、インターリープパラメータ演算装置1となるべきコンピュータに読み込ませて実行することにより、実現するものでもよい。コンピュータ読み取りが可能な記録媒体とは、フロッピーディスク、光磁気ディスク、CD-ROM等の記録媒体、コンピュータシステムに内蔵されるハードディスク装置等の記憶装置を指す。さらに、コンピュータ読み取りが可能な記録媒体とは、インターネット
50

を介してプログラムを送信する場合のように、短時間の間に、動的にプログラムを保持するもの（伝送媒体もしくは伝送波）、コンピュータシステム内部の揮発性メモリのように、一定時間プログラムを保持しているものも含む。

【図面の簡単な説明】

【0054】

【図1】本発明の一実施例の構成を示した図である。

【図2】本発明による、 $y[i]$ を計算するためのフローチャートである。

【図3】インターリープパラメータ演算装置の実装例を示した図である。

【図4】従来技術による、 $y[i]$ を計算するためのフローチャートである。

【符号の説明】

【0055】

1 インターリープパラメータ演算装置

2 インターリーバ

11 プロセッサ

12 $q[i]$ メモリ

14 $y[i]$ メモリ

31 ~ 38 ステップ

72 定数ブロック

73 レジスタ

74 カウンタ

76 セレクタ

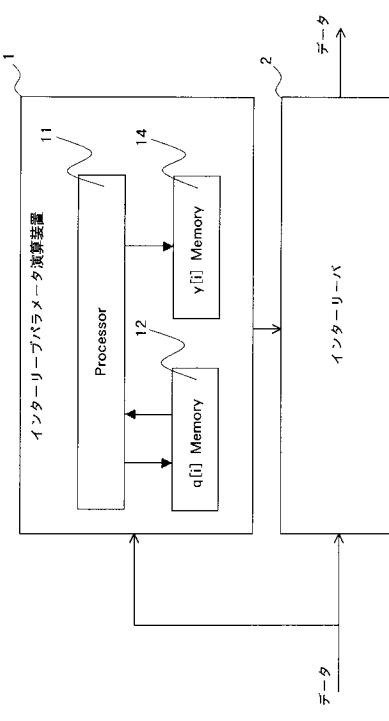
77 セレクタ

78 セレクタ

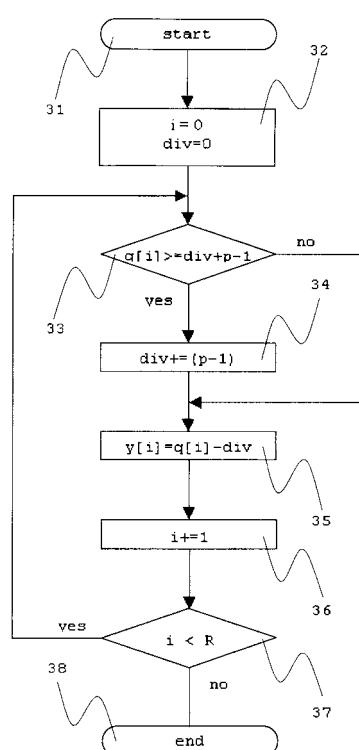
79 セレクタ

41 ~ 46 ステップ

【図1】



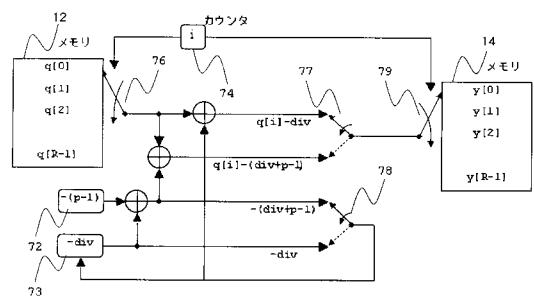
【図2】



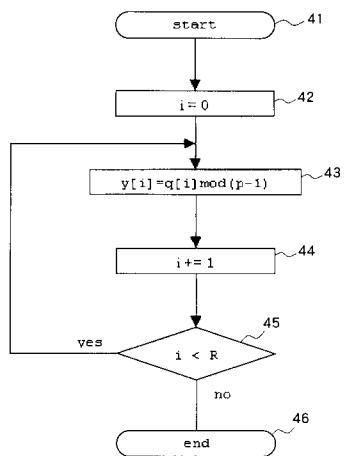
10

20

【図3】



【図4】



フロントページの続き

合議体

審判長 藤井 浩

審判官 神谷 健一

審判官 田中 庸介

(56)参考文献 特開2002-190743(JP,A)

特開2001-267934(JP,A)

特開2005-072843(JP,A)

須田博人 他, 素体を利用したターボ符号用インターリーバ, 電子情報通信学会論文誌, 200
2年11月, Vol. J85-A, No. 11, pp. 1168-1181

(58)調査した分野(Int.Cl., DB名)

H03M 13/27

H04L 1/00

G11B 20/18