

(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2005-295216
(P2005-295216A)

(43) 公開日 平成17年10月20日(2005. 10. 20)

(51) Int.Cl.⁷
H04L 29/00

F I
H04L 13/00

Z

テーマコード (参考)
5 K O 3 4

審査請求 未請求 請求項の数 4 O L (全 21 頁)

(21) 出願番号	特願2004-107597 (P2004-107597)	(71) 出願人	304020498 サクサ株式会社 東京都目黒区下目黒二丁目2番3号
(22) 出願日	平成16年3月31日 (2004. 3. 31)	(74) 代理人	100110319 弁理士 根本 恵司
		(74) 代理人	100109977 弁理士 畑川 清泰
		(74) 代理人	100106806 弁理士 三谷 浩
		(74) 代理人	100099472 弁理士 杉山 猛
		(74) 代理人	100104503 弁理士 益田 博文

最終頁に続く

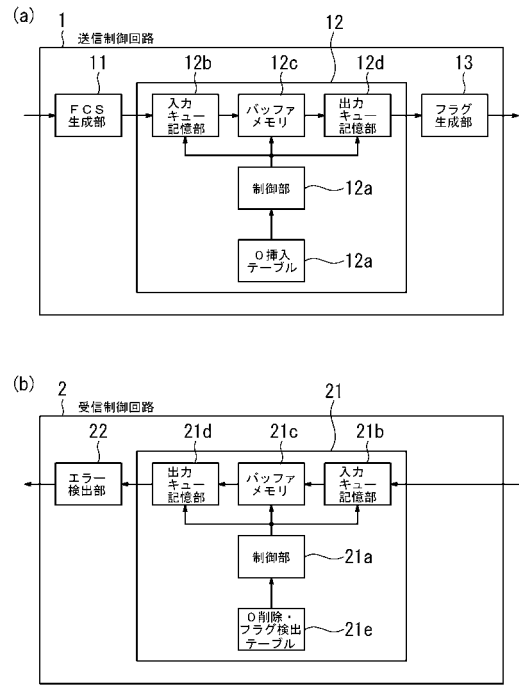
(54) 【発明の名称】 データ処理装置及びプログラム

(57) 【要約】

【課題】 0挿入及び0削除に必要な処理回数が入力キューのビット列の並びに影響されないようにして、処理時間を短縮する。

【解決手段】 0挿入部12では、入力キュー記憶部12bに記憶されたデータを8ビット単位でバッファメモリ12cに読み込み、0挿入テーブル12eを検索して、0挿入処理を行い、0挿入処理を受けたデータをバッファメモリ12cから出力キュー記憶部12dへ送出する。入力キューから選択される所定数の入力ビット列は、その並びに関係なく処理され、出力キューに付加されるので、0挿入処理の処理回数は入力キュー記憶部12bから読み込まれる入力キューのビット列の並びとは無関係となる。0削除及びフラグ検出部21の処理についても同様である。

【選択図】 図1



【特許請求の範囲】

【請求項 1】

入力キューに 0 挿入を行い出力キューを生成するデータ処理装置であって、前記入力キューから所定ビット数毎にビット列を選択して入力ビット列とする手段と、前回の入力ビット列と前回の出力ビット列との差異に対応する状態データ、前記所定ビット数のビット列の並び、出力ビット列、及び次回の状態データの組合せが予め記憶されたテーブルを検索して、前記入力ビット列に対応する出力ビット列及び次回の状態データを読み出す手段と、前記出力ビット列を前記出力キューに付加する手段とを備えたことを特徴とするデータ処理装置。

【請求項 2】

10

入力キューから 0 削除を行い出力キューを生成するデータ処理装置であって、前記入力キューから所定ビット数毎にビット列を選択して入力ビット列とする手段と、前回の入力ビット列と前回の出力ビット列との差異に対応する状態データ、前記所定ビット数のビット列の並び、出力ビット列、及び次回の状態データの組合せが予め記憶されたテーブルを検索して、前記入力ビット列に対応する出力ビット列及び次回の状態データを読み出す手段と、前記出力ビット列を前記出力キューに付加する手段とを備えたことを特徴とするデータ処理装置。

【請求項 3】

入力キューに 0 挿入を行い出力キューを生成するデータ処理装置のコンピュータを、前記入力キューから所定ビット数毎にビット列を選択して入力ビット列とする手段と、前回の入力ビット列と前回の出力ビット列との差異に対応する状態データ、前記所定ビット数のビット列の並び、出力ビット列、及び次回の状態データの組合せが予め記憶されたテーブルを検索して、前記入力ビット列に対応する出力ビット列及び次回の状態データを読み出す手段と、前記出力ビット列を前記出力キューに付加する手段として機能させることを特徴とするプログラム。

20

【請求項 4】

入力キューから 0 削除を行い出力キューを生成するデータ処理装置のコンピュータを、前記入力キューから所定ビット数毎にビット列を選択して入力ビット列とする手段と、前回の入力ビット列と前回の出力ビット列との差異に対応する状態データ、前記所定ビット数のビット列の並び、出力ビット列、及び次回の状態データの組合せが予め記憶されたテーブルを検索して、前記入力ビット列に対応する出力ビット列及び次回の状態データを読み出す手段と、前記出力ビット列を前記出力キューに付加する手段として機能させることを特徴とするプログラム。

30

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、HDL C (High-level Data Link Control Procedure) を用いたデータ処理する装置に関し、特に 0 挿入及び 0 削除の高速化を可能にしたデータ処理装置に関する。

【背景技術】

40

【0002】

HDL C は、IBM 社の SDLC (Synchronous Data Link Control) をベースに ISO (国際標準化機構) が定めたデータリンク層の標準規約である。HDL C のフレーム構成は、図 25 に示すように、フレームの開始及び終了を示す 8 ビットのフラグと、順次フラグに続く 8 ビットのアドレス部、8 ビット又は 16 ビットの制御部、任意ビット長の情報部、及び 16 ビットの FCS (フレーム・チェック・シーケンス) からなる。フラグのビットパターンは 01111110 と定められているため、フレームの開始と終了を示すフラグに挟まれた部分では、ビット 1 が 5 個連続する場合、送信側ではその次にビット 0 を挿入し (これを 0 挿入という)、受信側ではそのビット 0 を削除する (これを 0 削除という) ことで、フラグのビットパターンの独自性を保証している。

50

【0003】

従来、このような0挿入及び0削除を行う方法として、特許文献1に開示された「同期HDL Cのためのビット・スタッフィング方法及び装置」がある。以下、図26乃至51を参照しながら、特許文献1に開示されたビット・スタッフィング方法について説明する。

【0004】

まず、図26のフローチャート及び図27乃至36の状態変化を示す図を用いて0挿入方法を説明する。これらの図において、入力キューはこれから0挿入処理を受けるデータ、出力キューは0挿入処理を受けたデータである。

【0005】

まず、6オクテット（48ビット）入力キューの先頭から1オクテット（8ビット）のデータを選択する（図27（a）、図26ステップS31）。図27（a）において、枠で囲まれた01011011が選択された8ビットを示す。次に、0挿入用テーブルを検索して、（ア）出力キューに付加するビット数（以下、出力ビット数という）、（イ）入力キューから削除するビット数（以下、削除ビット数という）、及び（ウ）出力キューに付加するビット列（以下、出力ビット列という）を読み出す（図27（b）、ステップS32）。0挿入用テーブルには、入力される8ビットの全ての組み合わせ（ $2^8 = 256$ 通り）に対する上記（ア）、（イ）及び（ウ）のデータが予め格納されている。

【0006】

ここでは、出力ビット数＝6、削除ビット数＝6、出力ビット列＝010110である。選択された8ビットが全て入力キューから削除され、かつ出力キューに付加されない理由は、選択された8ビットの末尾の2ビット11は、その2ビットの次に1が3個続いた場合、その次に0挿入処理を行うことになるため、次に選択される先頭の3ビットと併せて処理する必要があるためである。従って、ここでは、出力キューに010110を付加し（ステップS33）、入力キューから010110を削除する（ステップS34）。図27（c）は、010110が削除された入力キュー、及び010110が付加された出力キューを示す。ここで、入力キューの010110に付与された二重線は削除されたビット列を表し、網がけは今回削除されたビット列を表す。また、出力キューに付与された下線は今回付加されたビット列を表す。

【0007】

以上のステップS31乃至34を入力キューのビット数が8より小さくなるまで繰り返す（ステップS35）。ここでは、図28（a）に示すように、11101111を選択する。この先頭の11は前回選択されたものの出力キューに付加されなかった2ビットである。テーブルを検索した結果は、図28（b）に示すように、出力ビット数＝4、削除ビット数＝4、出力ビット列＝1110である。従って、図28（c）に示すように、1110が入力キューから削除され、かつ出力キューに付加される。

【0008】

以後、図29乃至図35までステップS31乃至34の処理が繰り返し実行され、図35（c）に示す状態になると、ステップS35で”はい”と判定されるので、1フレームの処理が終わる。図30、31、32、34及び35では0挿入を行っている。なお、図35の入力キューのxxは次のフレームのデータである。

【0009】

次に、図37及び38のフローチャート及び図39乃至51の状態変化を示す図を用いて0削除方法を説明する。これらの図において、入力キューはこれから0削除処理を受けるデータ、出力キューは0削除処理を受けたデータである。また、図37は開始フラグ検出処理であり、図38は開始フラグ検出後の処理である。

【0010】

まず、0削除の前にフラグ検出を行うために、入力キューの先頭から1オクテット（8ビット）のデータを選択する（図39（a）、図37ステップS41）。次に、フラグ検出テーブルを検索して、（ア）入力キューから削除するビット数（削除ビット数）、及び

10

20

30

40

50

(イ) データ種別 (フラグ/通常データ) を読み出す (図 39 (b)、ステップ S 42)。ここでは、データ種別はフラグではなく、削除ビット数 = 1 である。従って、ステップ S 43 で ”いいえ” と判断され、ステップ S 45 で入力キューから先頭の 1 ビットが削除され、ステップ S 41 に戻る。フラグが検出される迄は出力キューにデータは付加されないで、図 39 (c) に示すように、1 が入力キューから削除され、出力キューに付加されるデータはない。

【0011】

再び入力キューから 8 ビットを選択する。今回は 0 1 1 1 1 1 1 0 が選択されている (図 40 (a))。フラグ検出テーブルを検索することにより、入力された 8 ビットがフラグであり、削除ビット数 = 8 であることが分かる。従って、ステップ S 43 で ”はい” と判断され、ステップ S 44 で入力キューからフラグの 8 ビットが削除され、開始フラグ検出処理を終える。

10

【0012】

開始フラグが検出されたので、次に図 38 に示されている 0 削除処理を実行する。入力キューの先頭から 1 オクテット (8 ビット) のデータを選択する (図 41 (a)、図 38 のステップ S 51)。ここでは、0 1 0 1 1 0 1 1 が選択されている。次に、0 削除テーブルを検索して、(ア) 出力キューに付加するビット数 (出力ビット数)、(イ) 入力キューから削除するビット数 (削除ビット数)、及び (ウ) データ種別 (通常データ/フラグ/アボート) を読み出す (図 41 (b)、ステップ S 52)。ここでは、図 41 (b) に示されているように、出力ビット数 = 5、削除ビット数 = 5、データ種別 = 通常データである。選択された 8 ビットの後部の 3 ビット 0 1 1 が出力キューに付加されない理由は、この 0 1 1 がフラグの先頭 3 ビットと同じであるため、その 3 ビットの次に 1 1 1 1 0 が続いた場合、終了フラグとなるからである。従って、ステップ S 53 ”通常データ” と判定され、ステップ S 54 で 0 1 0 1 1 が出力キューに付加され、ステップ S 55 で 0 1 0 1 1 が入力キューから削除された結果、図 41 (c) に示す状態となる。

20

【0013】

以後、図 42 乃至 50 迄ステップ S 51 乃至 55 の処理が繰り返し実行され、図 51 (a) に示すようにステップ S 53 でフラグが検出され、1 フレームの処理が終わる。図 44、45、46、48 及び 49 では 0 削除が行われている。1 フレームの処理が終わると、ステップ S 51 に戻って次のフレームの処理を行う。また、ステップ S 53 でアボートが検出された場合は、図 37 のステップ S 41 に戻ってフラグ検出処理を行う。

30

【特許文献 1】特表 2002-520952 号公報

【発明の開示】

【発明が解決しようとする課題】

【0014】

しかしながら、上記従来の 0 挿入方法及び 0 削除方法では、入力キューから選択したビット列のうち確定したビット数分だけを削除するため、削除処理が複雑になる。また、ビット列の並びにより処理回数が増減し、0 挿入方法では特に 1 が連続すると処理回数が極めて増加し、0 削除方法ではフラグの一部と同じビット並びのデータがあるとき、及び 0 削除が必要なビット列が連続するときに処理回数が多くなり、処理時間が長くなる。

40

【0015】

本発明はこのような問題点を解決するためになされたもので、0 挿入処理又は 0 削除処理の処理回数が入力キューのビット列の並びに影響を受けないようにして、処理時間を短縮することを目的とする。

【課題を解決するための手段】

【0016】

請求項 1 に係る発明は、入力キューに 0 挿入を行い出力キューを生成するデータ処理装置であって、前記入力キューから所定ビット数毎にビット列を選択して入力ビット列とする手段と、前回の入力ビット列と前回の出力ビット列との差異に対応する状態データ、前記所定ビット数のビット列の並び、出力ビット列、及び次回の状態データの組合せが予め

50

記憶されたテーブルを検索して、前記入力ビット列に対応する出力ビット列及び次回の状態データを読み出す手段と、前記出力ビット列を前記出力キューに付加する手段とを備えたことを特徴とするデータ処理装置である。

請求項 2 に係る発明は、入力キューから 0 削除を行い出力キューを生成するデータ処理装置であって、前記入力キューから所定ビット数毎にビット列を選択して入力ビット列とする手段と、前回の入力ビット列と前回の出力ビット列との差異に対応する状態データ、前記所定ビット数のビット列の並び、出力ビット列、及び次回の状態データの組合せが予め記憶されたテーブルを検索して、前記入力ビット列に対応する出力ビット列及び次回の状態データを読み出す手段と、前記出力ビット列を前記出力キューに付加する手段とを備えたことを特徴とするデータ処理装置である。

10

請求項 3 に係る発明は、入力キューに 0 挿入を行い出力キューを生成するデータ処理装置のコンピュータを、前記入力キューから所定ビット数毎にビット列を選択して入力ビット列とする手段と、前回の入力ビット列と前回の出力ビット列との差異に対応する状態データ、前記所定ビット数のビット列の並び、出力ビット列、及び次回の状態データの組合せが予め記憶されたテーブルを検索して、前記入力ビット列に対応する出力ビット列及び次回の状態データを読み出す手段と、前記出力ビット列を前記出力キューに付加する手段として機能させることを特徴とするプログラムである。

請求項 4 に係る発明は、入力キューから 0 削除を行い出力キューを生成するデータ処理装置のコンピュータを、前記入力キューから所定ビット数毎にビット列を選択して入力ビット列とする手段と、前回の入力ビット列と前回の出力ビット列との差異に対応する状態データ、前記所定ビット数のビット列の並び、出力ビット列、及び次回の状態データの組合せが予め記憶されたテーブルを検索して、前記入力ビット列に対応する出力ビット列及び次回の状態データを読み出す手段と、前記出力ビット列を前記出力キューに付加する手段として機能させることを特徴とするプログラムである。

20

【発明の効果】

【0017】

本発明によれば、入力キューから選択された所定数のビット列は、ビット列の並びに関係なく処理され、出力キューに付加されるので、0 挿入処理又は 0 削除処理の処理回数がビット列の並びに影響を受けなくなり、処理時間が短縮される。

【発明を実施するための最良の形態】

30

【0018】

以下、図面を参照しながら本発明の実施形態について説明する。

図 1 は本発明の実施形態に係る送信回路及び受信制御回路の構成を示すブロック図である。送信制御回路 1 は、FCS 生成部 11 と、その出力側に接続された 0 挿入部 12 と、その出力側に接続されたフラグ生成部 13 とを備えている。受信制御回路 2 は、0 削除及びフラグ検出部 21 と、その出力側に接続されたエラー検出部 22 とからなる。

【0019】

0 挿入部 12 は、入力キュー記憶部 12b と、その出力側に接続されたバッファメモリ 12c と、その出力側に接続された出力キュー記憶部 12d と、各部を制御する制御部 12a と、制御部 12a に接続された 0 挿入テーブル 12e とを備えている。また、0 削除及びフラグ検出部 21 は、入力キュー記憶部 21b と、その出力側に接続されたバッファメモリ 21c と、その出力側に接続された出力キュー記憶部 21d と、各部を制御する制御部 21a と、制御部 21a に接続された 0 削除及びフラグ検出テーブル 21e とを備えている。

40

【0020】

送信制御回路 1 の FCS 生成部 11 には、図示されていないフレーム生成部で生成された HDLC フレームデータが入力される。FCS 生成部 11 は、入力された HDLC フレームデータに FCS (誤り訂正符号) を付加し、0 挿入部 12 へ出力する。0 挿入部 12 では、制御部 12a の制御により、入力キュー記憶部 12b にデータを記憶すると共に、記憶されたデータを所定ビット単位でバッファメモリ 12c に読み込み、0 挿入テーブル

50

1 2 e を検索して、0 挿入処理を行い、0 挿入処理を受けたデータをバッファメモリ 1 2 c から出力キュー記憶部 1 2 d へ送出する。さらに、出力キュー記憶部 1 2 d に記憶されたデータをフラグ生成部 1 3 へ送るように制御する。詳細について後述するように、0 挿入テーブル 1 2 e には、前回入力キュー記憶部 1 2 b から読み込まれたビット列と前回出力キュー記憶部 1 2 d に送出されたビット列との差異に対応する状態データ、所定ビット数のビット列の並び、出力キュー記憶部 1 2 d へ送出されるビット列とそのビット数、及び入力キュー記憶部 1 2 b から読み込まれるビット列と出力キュー記憶部 1 2 d へ送出されるビット列との差異に対応する次の状態データの全ての組合せが予め記憶されている。また、入力キュー記憶部 1 2 b から読み込まれる所定数のビット列は、その並びに関係なく、入力キュー記憶部 1 2 b から削除されるので、0 挿入処理の処理回数は入力キュー記憶部 1 2 b から読み込まれるビット列の並びとは無関係となる。フラグ生成部 1 2 d は、入力されたデータにフラグ 0 1 1 1 1 1 0 を付加し、図示されていないデータ伝送路を介して受信制御回路 2 へ送信する。

10

【0 0 2 1】

受信制御回路 2 の 0 削除及びフラグ検出部 2 1 には、送信制御回路 1 から送信された H D L C フレームデータが図示されていないデータ伝送路を介して入力される。0 削除及びフラグ検出部 2 1 では、制御部 2 1 a の制御により、入力キュー記憶部 2 1 b にデータを記憶すると共に、記憶されたデータを所定ビット単位でバッファメモリ 2 1 c に読み込み、0 削除及びフラグ検出テーブル 2 1 e を検索して、0 削除及びフラグ検出処理を行い、0 削除及びフラグ検出処理を受けたデータをバッファメモリ 2 1 c から出力キュー記憶部 2 1 d へ送出する。さらに、出力キュー記憶部 2 1 d に記憶されたデータをエラー検出部 2 2 へ送るように制御する。詳細について後述するように、0 削除及びフラグ検出テーブル 2 1 e には、前回入力キュー記憶部 2 1 b から読み込まれたビット列と前回出力キュー記憶部 2 1 d に送出されたビット列との差異に対応する状態データ、所定ビット数のビット列の並び、出力キュー記憶部 2 1 d へ送出されるビット列とそのビット数、入力キュー記憶部 2 1 b から読み込まれるビット列と出力キュー記憶部 2 1 d へ送出されるビット列との差異に対応する次の状態データ、及びデータの種別（通常データ／フラグ／アボート）の全ての組合せが予め記憶されている。また、入力キュー記憶部 2 1 b から読み込まれる所定数のビット列は、その並びに関係なく、入力キュー記憶部 2 1 b から削除されるので、0 削除処理の処理回数は入力キュー記憶部 2 1 b から読み込まれるビット列の並びとは無関係となる。エラー検出部 2 2 は、入力されたデータの F C S を用いて誤り訂正処理を行い、図示されていないフレーム受信部へ送出する。

20

30

【0 0 2 2】

以下、図 2 乃至 1 1 を参照しながら、送信制御回路 1 の 0 挿入部 1 2 の動作を説明する。

まず、0 挿入テーブル 1 2 e の内容を説明する。本実施形態では、入力キュー記憶部 1 2 b から 8 ビットずつデータがバッファメモリ 1 2 c に読み込まれ、0 挿入が必要な場合、つまり 1 が 5 個続いた場合には 0 挿入処理を受け、出力キュー記憶部 1 2 d へ送出される。また、入力キュー記憶部 1 2 b から例えば図 2 8 (a) に示した 1 1 1 0 1 1 1 1 が読み込まれた場合、出力キュー記憶部 1 2 d へ送出されるのは図 2 8 (c) と同様、1 1 1 0 となり、1 1 1 1 は送出されない。本実施形態では、この 1 1 1 1 からなる残りビットを状態番号として保存し、次回に入力キュー記憶部 1 2 b から読み込まれる 8 ビットと同時に扱う。

40

【0 0 2 3】

残りビットは、1、1 1、1 1 1 及び 1 1 1 1 の 4 種類であるから、図 2 に示すように、状態番号 0 乃至 4 により残りビットの種類を定めている。そして、0 挿入テーブル 1 2 e には、入力キューから読み込まれる 8 ビットのビット列の並び、出力キュー記憶部 1 2 d へ出力されるビット列とそのビット数、及び状態番号の全ての組合せが予め記憶されている。

【0 0 2 4】

50

次に、0 挿入テーブル 1 2 e の作成手順を説明する。

[1] 状態番号が 0 のテーブルの場合

下記 (1) ~ (4) の手順で作成する。

(1) 対象とする 8 ビットについて、順にビットを調べ、0 挿入が必要か否かを判定する。

【0 0 2 5】

<判定の法則>

- ・ 0 挿入が必要か否かは 1 が 5 個連続しているか、途中で 0 が出現するかを確認する。
- ・ 1 の連続数が 5 個未満の場合、そのまま取り出しビットの並びとすることができる。
- ・ 1 の連続数が 5 個に達した場合、続けて 0 を挿入する事によってそれまで判定したビット列は、取り出しビットの並びとし、1 の連続数も 0 から数え直す。

【0 0 2 6】

(2) (1) の処理を行い、8 ビット全てを調べる。8 ビットの終了時点では以下の 3 つの状態が考えられる。

(ア) 8 ビット目が 0 で終了した状態 (1 の連続数が 0)

(イ) 8 ビット目で 1 の連続数が 5 になり、0 挿入をした直後の状態 (1 の連続数が 0)

(ウ) 8 ビット目では 1 の連続数が 1 以上 5 未満であり、次の取り出しビットに影響を与える状態

【0 0 2 7】

(3) (ア)、(イ) の状態の場合、それまで作成した、ビット列、作成済みのビット数がテーブルとなる。また、全てのビットが判定終了したので、残りのビットは 0 である。これは、残りのビット数が 0 であるので、次状態が 0 ということでもある。

【0 0 2 8】

(4) (ウ) の状態の場合は、判定済みのビット列とビット列のビット数をテーブルとする。また、終了時点での 1 の連続数を次の状態番号とする。

残りビット無しの場合 → 次状態：0

残りビット 1 の場合 → 次状態番号：1

残りビット 1 1 の場合 → 次状態番号：2

残りビット 1 1 1 の場合 → 次状態番号：3

残りビット 1 1 1 1 の場合 → 次状態番号：4

【0 0 2 9】

[2] 状態番号が 0 以外のテーブルの場合

下記 (1) ~ (5) の手順で作成する。

(1) 状態番号により、対象とする 8 ビットの前に以下のビットを加える。

状態番号：1 → 1 + (対象とする 8 ビット)

状態番号：2 → 1 1 + (対象とする 8 ビット)

状態番号：3 → 1 1 1 + (対象とする 8 ビット)

状態番号：4 → 1 1 1 1 + (対象とする 8 ビット)

【0 0 3 0】

(2) (1) で作成したビット列について、順にビットを調べ、0 挿入が必要か否かを判定する。

<判定の法則>

- ・ 0 挿入が必要か否かは 1 が 5 個連続しているか、途中で 0 が出現するかを確認する。
- ・ 1 の連続数が 5 個未満の場合、そのまま取り出しビットの並びとすることができる。
- ・ 1 の連続数が 5 個に達した場合、続けて 0 を挿入する事によってそれまで判定したビット列は、取り出しビットの並びとし、1 の連続数も 0 から数え直す。

【0 0 3 1】

(3) (2) の処理を行い、(1) で作成したビット全てを調べる。全ビットの終了時点では以下の 3 つの状態が考えられる。

(ア) 最後のビットが 0 で終了した状態 (1 の連続数が 0)

(イ) 最後のビットで1の連続数が5になり、0挿入をした直後の状態(1の連続数が0)

(ウ) 最後のビットでは1の連続数が1以上5未満であり、次の取り出しビットに影響を与える状態

【0032】

(4) (ア)、(イ)の状態の場合、それまで作成した、ビット列、作成済みのビット数がテーブルとなる。また、全てのビットが判定終了したので、残りのビットは0である。これは、残りのビット数が0であるので、次状態が0ということでもある。

【0033】

(5) (ウ)の状態の場合は、判定済みのビット列とビット列のビット数をテーブルとする。 10

残りビット無しの場合 → 次状態番号：0

残りビット1の場合 → 次状態番号：1

残りビット11の場合 → 次状態番号：2

残りビット111の場合 → 次状態番号：3

残りビット1111の場合 → 次状態番号：4

【0034】

[3] 0挿入テーブルの作成例(状態番号：2、入力ビット列：10111101の変換テーブル作成の場合)

(1) 状態2であるから、前回のビット列は11で終了した状態であり、これに10111101を合わせて判定ビット列は1110111101と考えることができる。 20

(2) (1)で作成したビット列について、0挿入の必要を考慮しながら、取り出しビットを作成すると、111011110+1と分割できる。

(3) よって、取り出しビットは、111011110、ビット数「9」を得る。残りの「1」については次状態に影響を与えるため、今回はデータを確定できず、次の状態を1とする。

(4) 以上から、状態番号：2、入力ビット列：10111101の変換テーブルとして、

次状態番号：1、出力ビット数：9、出力ビット列=111011110

が得られる。 30

【0035】

図3は0挿入テーブル12eの一部を抜粋したものである。この図に示されているように、状態番号と入力ビット列が決まれば、それに対応して次状態番号、出力ビット数及び出力ビット列が定まる。なお、入力キュー記憶部12bから最初に読み込まれるビット列の状態番号は0である。

【0036】

次に図4のフローチャート及び図5乃至11の状態変化を示す図を用いて0挿入方法を説明する。これらの図において、入力キューは入力キュー記憶部12bに記憶されているビット列、出力キューは出力キュー記憶部12dに記憶されているビット列である。なお、この入力キューのビット列の並びは従来例における入力キューのビット列の並びと同じ 40

【0037】

まず、6オクテット(48ビット)入力キューの先頭から1オクテット(8ビット)のデータを選択する(図5(a)、図4のステップS1)。図5(a)において、枠で囲まれた01011011が選択された8ビットを示す。次に、0挿入用テーブルを検索して、(ア)次回の状態番号(次状態番号)、(イ)出力キューに付加するビット数(出力ビット数)、(ウ)及び出力キューに付加するビット列(出力ビット列)を読み出す(図5(b)、ステップS2)。0挿入用テーブルには、入力される8ビットの全ての組み合わせと状態番号数の積($2^8 \times 5 = 1280$ 通り)に対する上記(ア)、(イ)及び(ウ)のデータが予め格納されている。 50

【0038】

ここでは、次状態番号＝2、出力ビット数＝6、出力ビット列＝010110である。なお、入力キューから選択された01011011の状態（初期状態）は0である。ここで、選択された8ビットの全てが出力キューに付加されない理由は、従来例（図27）を参照しなから説明したように、選択された8ビットの末尾の2ビット11は、その2ビットの次に1が3個続いた場合、その次に0挿入処理を行うことになるため、次に選択される先頭の3ビットと併せて処理する必要がある、未確定であるためである。従って、ここでは、出力キューに010110を付加する（ステップS3）。一方、入力キューからは01011011を削除すると共に次状態番号を保存する（ステップS4）。図5（c）において01011011が削除された入力キュー、010110が付加された出力キューを示す。ここで、入力キューの010110に付与された二重線は削除されたビット列を表し、網がけは今回削除されたビット列を表す。また、出力キューに付与された下線は今回付加されたビット列を表す。

【0039】

ここで、図5（c）と図27（c）とを比較すると、出力ビット列及び出力ビット数は本実施形態も従来例も同じであるが、削除ビット数は、従来例では出力ビット数と同じであるのに対し、本実施形態では選択されたビット数（＝8）と同じになる。そして、本実施形態では、従来例では入力キューから削除せずに残した11を削除し、次状態番号2として保存している。

【0040】

以上のステップS1乃至4を入力キューのビット列がなくなるまで繰り返す（ステップS5）。ここでは、図6（a）に示すように、10111101を選択する。この入力ビット列の状態番号は前回保存された2である。この入力ビット列及び状態番号を用いて0挿入テーブル12eを検索すると、図6（b）に示すように、次状態番号＝1、出力ビット数＝9、出力ビット列＝111011110である。ここで、出力ビット列の先頭の2ビット11は、入力キューから前回選択されて削除されたものの出力ビットにならず、次状態番号により保存されていたものである。次に、111011110を出力キューに付加し（ステップS3）、10111101を入力キューから削除すると共に次状態番号（＝1）を保存する（ステップS4）ことで、図6（c）に示す状態となる。

【0041】

以後、図7乃至10までステップS1乃至4の処理が繰り返し実行され、図11に示す状態になると、ステップS5で”はい”と判定されるので、次のステップS6で状態番号を判断し、その状態番号に対応する数の1を出力キューに付加して1フレームの処理が終わる。図7、8、9及び10では0挿入を行っている。

【0042】

このように、本実施形態の0挿入処理では、入力キューから選択した1オクテット（8ビット）を入力ビット列の並びに関係なく全て削除しているため、処理が簡単であり、かつ入力キューのオクテット数と同じ回数のループ処理により完了する。

【0043】

次に、図12乃至24を参照しながら、受送信制御回路2の0削除及びフラグ検出部21の動作を説明する。

【0044】

まず、0削除及びフラグ検出テーブル21eの内容を説明する。本実施形態では、入力キュー記憶部21bから8ビットずつデータがバッファメモリ21cに読み込まれ、0削除が必要な場合、つまり1が5個続いた場合には0削除処理を受け、出力キュー記憶部21dへ送出される。また、フラグである01111110が読み込まれた場合、フラグ検出処理を行う。ここで、フレームの最初のフラグである開始フラグが検出されたときは、それ以後読み込まれるデータに対して0削除処理を開始し、フレームの最後のフラグである終了フラグが検出されたときは、1フレームの0削除処理を終える。また、0挿入処理と同様、入力キューから選択したビット列の内、出力キューに付加されない残りビット列

を識別するために状態番号を保存する。

【0045】

状態番号は、図12に示されているように、状態0から12までの13通りになる。そして、0削除及びフラグ検出テーブル21eには、入力キュー21bから読み込まれる8ビットのビット列の並び、出力キュー記憶部21dへ出力されるビット列とそのビット数、データの種別（通常データ／フラグ／アボート）及び状態番号の全ての組合せが予め記憶されている。

【0046】

次に、0削除及びフラグ検出テーブル21eの作成手順を説明する。

[1] 状態0のテーブルの場合

下記(1)～(5)の手順により作成する。

【0047】

(1) 対象とする8ビットについて、順にビットを調べ、フラグの有無と0削除が必要か否かを判定する。

＜判定の法則＞

- ・0挿入が必要か否かは1が5個連続しているか、途中で0が出現するかを確認する。ただし、フラグの存在も考慮する為、1から始まる1の連続の場合と、0から始まる1の連続は区別する必要がある。

- ・1から始まる1の連続の場合で、1の連続数が5未満の場合、そのまま取り出しビットの並びとする。

- ・1から始まる1の連続の場合で、1の連続数が5個で次に続くビットが0の場合、その0を削除したものを取り出しビットの並びとする。

- ・1から始まる1の連続の場合で、1の連続数が6個になる場合、無効なデータであるので、種別を「無効（アボート）」とする。

- ・0から始まる1の連続の場合で、1の連続数が5未満の場合、そのまま取り出しビットの並びとする。

- ・0から始まる1の連続の場合で、1の連続数が5個で次に続くビットが0の場合、その0を削除したものを取り出しビットの並びとする。

- ・0から始まる1の連続の場合で、1の連続数が6個で次に続くビットが0の場合、そのビット並びはフラグをあらわすので、種別を「フラグ」とする。このとき、フラグに該当する01111110の並びは取り出しビットには含めない。

- ・0から始まる1の連続の場合で、1の連続数が7個になる場合、無効なデータであるので、種別を「無効」とする。

【0048】

(2) (1)の処理を行い、8ビット全てを調べる。8ビットの終了時点では以下の4つの状態が考えられる。

(ア) 8ビット目が0で終了した場合

(イ) 7ビット目まで1が5個連続し、8ビット目の0を削除して終了した場合

(ウ) 8ビット目では1の連続数が1以上6未満で有り、次の取り出しビットに処理が依存する場合

(エ) データ種別が「異常」な場合

【0049】

(3) (ア)、(イ)の状態の場合、データ種別は「通常」、作成済みのビット列とそのビット列のビット数がテーブルになる。また、全てのビットが判定終了したので、残りビットは0である。これは、残りのビット数が0であるので、次状態が0ということである。

【0050】

10

20

30

40

50

(4) (ウ)の状態の場合は、データ種別は「通常」、作成済みのビット列と、そのビット数がテーブルになる。また、終了時点での確定していないビットの状態により、次状態番号を決める。

- 残りビットが0の場合→次状態番号：1
- 残りビットが1の場合→次状態番号：2
- 残りビットが01の場合→次状態番号：3
- 残りビットが11の場合→次状態番号：4
- 残りビットが011の場合→次状態番号：5
- 残りビットが111の場合→次状態番号：6
- 残りビットが0111の場合→次状態番号：7
- 残りビットが1111の場合→次状態番号：8
- 残りビットが01111の場合→次状態番号：9
- 残りビットが11111の場合→次状態番号：10
- 残りビットが011111の場合→次状態番号：11
- 残りビットが0111111の場合→次状態番号：12

10

(5) (エ)の状態の場合、データ種別を「異常」とする。

【0051】

[2] 状態番号が0以外のテーブルの場合

下記(1)～(6)の手順により作成する。

(1) 状態により、対象とする8ビットの前に以下のビットを加える。

20

- 状態番号：1→0+（対象とする8ビット）
- 状態番号：2→1+（対象とする8ビット）
- 状態番号：3→01+（対象とする8ビット）
- 状態番号：4→11+（対象とする8ビット）
- 状態番号：5→011+（対象とする8ビット）
- 状態番号：6→111+（対象とする8ビット）
- 状態番号：7→0111+（対象とする8ビット）
- 状態番号：8→1111+（対象とする8ビット）
- 状態番号：9→01111+（対象とする8ビット）
- 状態番号：10→11111+（対象とする8ビット）
- 状態番号：11→011111+（対象とする8ビット）
- 状態番号：12→0111111+（対象とする8ビット）

30

【0052】

(2) (1)で作成したビット列について、順にビットを調べ、フラグの有無と0削除が必要か否かを判定する。

<判定の法則>

・0挿入が必要か否かは1が5個連続しているか、途中で0が出現するかを確認する。
ただし、フラグの存在も考慮する為、1から始まる1の連続と、0から始まる1の連続とは区別する必要がある。

・1から始まる1の連続の場合で、1の連続数が5未満の場合、そのまま取り出しビットの並びとする。

40

・1から始まる1の連続の場合で、1の連続数が5個で次に続くビットが0の場合、その0を削除したものを取り出しビットの並びとする。

・1から始まる1の連続の場合で、1の連続数が6個になる場合、無効なデータであるので、種別を「無効」とする。

・0から始まる1の連続の場合で、1の連続数が5未満の場合、そのまま取り出しビットの並びとする。

・0から始まる1の連続の場合で、1の連続数が5個で次に続くビットが0の場合、その0を削除したものを取り出しビットの並びとする。

50

・ 0 から始まる 1 の連続の場合で、1 の連続数が 6 個で次に続くビットが 0 の場合、そのビット並びはフラグをあらわすので、種別を「フラグ」とする。このとき、フラグに該当する 0 1 1 1 1 1 0 の並びは取り出しビットには含めない。

・ 0 から始まる 1 の連続の場合で、1 の連続数が 7 個になる場合、無効なデータであるので、種別を「無効」とする。

【0 0 5 3】

(3) (2) の処理を行い、(1) で作成したビット全てを調べる。全てのビットの終了時点では以下の 4 つの状態が考えられる。

(ア) 8 ビット目が 0 で終了した場合

10

(イ) 7 ビット目まで 1 が 5 個連続し、8 ビット目の 0 を削除して終了した場合

(ウ) 8 ビット目では 1 の連続数が 1 以上 6 未満で有り、次の取り出しビットに処理が依存する場合

(エ) データ種別が「異常」な場合

【0 0 5 4】

(4) (ア)、(イ) の状態の場合、データ種別は「通常」、作成済みのビット列とそのビット列のビット数がテーブルになる。また、全てのビットが判定終了したので、残りビットは 0 であるこれは、残りのビット数が 0 であるので、次状態が 0 ということである。

【0 0 5 5】

(5) (ウ) の状態の場合は、データ種別は「通常」、作成済みのビット列と、そのビット数がテーブルになる。また、終了時点での確定していないビットの状態により、次状態番号を決める。

20

残りビットが 0 の場合→次状態番号：1

残りビットが 1 の場合→次状態番号：2

残りビットが 0 1 の場合→次状態番号：3

残りビットが 1 1 の場合→次状態番号：4

残りビットが 0 1 1 の場合→次状態番号：5

残りビットが 1 1 1 の場合→次状態番号：6

残りビットが 0 1 1 1 の場合→次状態番号：7

残りビットが 1 1 1 1 の場合→次状態番号：8

30

残りビットが 0 1 1 1 1 の場合→次状態番号：9

残りビットが 1 1 1 1 1 の場合→次状態番号：1 0

残りビットが 0 1 1 1 1 1 の場合→次状態番号：1 1

残りビットが 0 1 1 1 1 1 1 の場合→次状態番号：1 2

(6) (エ) の状態の場合、データ種別を「異常」とする。

【0 0 5 6】

[3] 0 削除テーブルの作成例（状態番号：1 2、入力ビット列：0 0 1 0 1 1 0 1 の変換テーブルの作成の場合）

(1) 状態番号：1 2 であるから、前回のビット列は 0 1 1 1 1 1 1 で終了した状態であり、これに今回の 0 0 1 0 1 1 0 1 を合わせて判定ビット列は 0 1 1 1 1 1 1 0 0 1 0 1 1 0 1 と考えることができる。

40

(2) (1) で作成したビット列について、フラグの存在と 0 削除の必要を考慮しながら、取り出しビットを作成すると、0 1 1 1 1 1 1 0 + 0 1 0 1 1 + 0 1 に分割できる。よって、ひとつ目の 0 1 1 1 1 1 1 0 から種別はフラグ、取り出しビットは 0 1 0 1 1、ビット数 5 を得る。

残りの 0 1 については次状態に影響を与えるため、今回はデータを確定できず、次状態番号を 3 とする。

以上から、状態番号：1 2、入力ビット列：0 0 1 0 1 1 0 1 の変換テーブルとして、

次状態番号：3、出力ビット数：5、出力ビット列 = 1 1 1 0 1 1 1 1 0

50

が得られる。

【0057】

図13は0削除及びフラグ検出テーブル21eの一部を抜粋したものである。この図に示されているように、状態番号と入力ビット列が決まれば、それに対応してデータ種別が判定されると共に次状態番号、出力ビット数及び出力ビット列が定まる。なお、入力キュー記憶部12bから最初に読み込まれるビット列の状態番号は0である。

【0058】

次に、図14及び15のフローチャート並びに図16乃至24の状態変化を示す図を用いて0削除及びフラグ検出方法を説明する。これらの図において、入力キューはこれから0削除及びフラグ検出処理を受けるデータ、出力キューは0削除及びフラグ検出処理を受けたデータである。また、図14は開始フラグ検出処理であり、図15は開始フラグ検出後の処理である。なお、この入力キューのビット列の並びは従来例における入力キューのビット列の並びと同じである。

【0059】

まず、0削除の前にフラグ検出を行うために、入力キューの先頭から1オクテット（8ビット）のデータを選択する（図16（a）、図14のステップS11）。図16（a）において、枠で囲まれた10111111が選択された8ビットを示す。次に、0削除及びフラグ検出テーブル21eを検索して、（ア）次状態番号、（イ）データ種別（通常データ／フラグ／アボート）、（ウ）出力キューに付加するビット数（出力ビット数）、及び（エ）出力キューに付加するビット列（出力ビット列）を読み出す。（図16（b）、ステップS12）。ここでは、次状態番号＝12、データ種別＝通常、出力ビット数＝1、出力ビット列＝1である。従って、フラグではないから、ステップS13で”いいえ”と判断し、ステップS14で入力キューから8ビットを削除すると共に状態番号12を保存した後にステップS11に戻る。フラグが検出される迄は出力キューにデータは付加されないので、図16（c）に示すように、10111111が入力キューから削除され、出力キューに付加されるデータはない。

【0060】

再び入力キューから8ビットを選択する（ステップS11）。今回は00101101が選択されている（図17（a））。この入力ビット列の状態番号は前回保存された12である。この入力ビット列及び状態番号を用いて0削除及びフラグ検出テーブル21eを検索すると（ステップS12）、次状態番号＝3、データ種別＝フラグ、出力ビット数＝5、出力ビット列＝01011である（図17（b））。フラグが検出されているから、ステップS13で”はい”と判断し、ステップS15で出力キューに出力ビット列を付加した後、ステップS16で00101101を削除すると共に状態番号3を保存し、開始フラグ検出処理を終える。

【0061】

開始フラグが検出されたので、次に図15に示されている0削除処理を実行する。入力キューの先頭から1オクテット（8ビット）のデータを選択する（図18（a）、図15のステップS21）。ここでは、11011110が選択されている。次に、0削除及びフラグ検出テーブル21eを検索すると（ステップS22）、次状態番号＝1、データ種別＝通常、出力ビット数＝9、出力ビット列＝011101111である（図18（b））。従って、ステップS23通常データと判定し、ステップS24で011101111を出力キューに付加し、ステップS25で入力キューから11011110を削除すると共に状態番号1を保存する（図18（c））。

【0062】

以後、図19乃至図24迄ステップS21乃至25の処理が繰り返し実行され、図24に示すようにステップS23でフラグが検出され、1フレームの処理が終わる。図19、20、21、22、及び24では0削除が行われている。1フレームの処理が終わると、ステップS21に戻って次のフレームの処理を行う。また、ステップS23でアボートが検出された場合は、図14のステップS11に戻ってフラグ検出処理を行う。

【0063】

このように、本実施形態の0削除処理では、入力キューから選択した1オクテット（8ビット）を入力ビット列の並びに関係なく全て削除しているため、処理が簡単であり、かつ入力キューのオクテット数と同じ回数のループ処理により完了する。

【0064】

なお、以上の説明では、最適値の8ビット単位で入力キューからビット列を読み込むものとしたが、8ビット以外のビット数単位で読み込むようにしてもよい。また、本発明は、HDL C手順だけでなく、ビット列の並びを特定の規則に従って変換する処理であり、かつ変換の前後にビット数が増減する処理に対しても同様に適用できる。

【図面の簡単な説明】

10

【0065】

【図1】図1は本発明の実施形態に係る送信回路及び受信制御回路の構成を示すブロック図である。

【図2】本発明の実施形態における0挿入テーブルの状態番号を示す図である。

【図3】本発明の実施形態における0挿入テーブルの抜粋を示す図である。

【図4】本発明の実施形態における0挿入処理のフローチャートである。

【図5】本発明の実施形態における0挿入処理の状態変化を示す図である。

【図6】本発明の実施形態における0挿入処理の状態変化を示す図である。

【図7】本発明の実施形態における0挿入処理の状態変化を示す図である。

【図8】本発明の実施形態における0挿入処理の状態変化を示す図である。

20

【図9】本発明の実施形態における0挿入処理の状態変化を示す図である。

【図10】本発明の実施形態における0挿入処理の状態変化を示す図である。

【図11】本発明の実施形態における0挿入処理の状態変化を示す図である。

【図12】本発明の実施形態における0削除及びフラグ検出テーブルの状態番号を示す図である。

【図13】本発明の実施形態における0削除及びフラグ検出テーブルの抜粋を示す図である。

【図14】本発明の実施形態における開始フラグ検出処理のフローチャートである。

【図15】本発明の実施形態における0削除処理のフローチャートである。

【図16】本発明の実施形態における開始フラグ検出処理の状態変化を示す図である。

30

【図17】本発明の実施形態における開始フラグ検出処理の状態変化を示す図である。

【図18】本発明の実施形態における0削除処理の状態変化を示す図である。

【図19】本発明の実施形態における0削除処理の状態変化を示す図である。

【図20】本発明の実施形態における0削除処理の状態変化を示す図である。

【図21】本発明の実施形態における0削除処理の状態変化を示す図である。

【図22】本発明の実施形態における0削除処理の状態変化を示す図である。

【図23】本発明の実施形態における0削除処理の状態変化を示す図である。

【図24】本発明の実施形態における0削除処理の状態変化を示す図である。

【図25】HDL Cフレームの構成を示す図である。

【図26】従来の0挿入処理のフローチャートである。

40

【図27】従来の0挿入処理の状態変化を示す図である。

【図28】従来の0挿入処理の状態変化を示す図である。

【図29】従来の0挿入処理の状態変化を示す図である。

【図30】従来の0挿入処理の状態変化を示す図である。

【図31】従来の0挿入処理の状態変化を示す図である。

【図32】従来の0挿入処理の状態変化を示す図である。

【図33】従来の0挿入処理の状態変化を示す図である。

【図34】従来の0挿入処理の状態変化を示す図である。

【図35】従来の0挿入処理の状態変化を示す図である。

【図36】従来の0挿入処理の状態変化を示す図である。

50

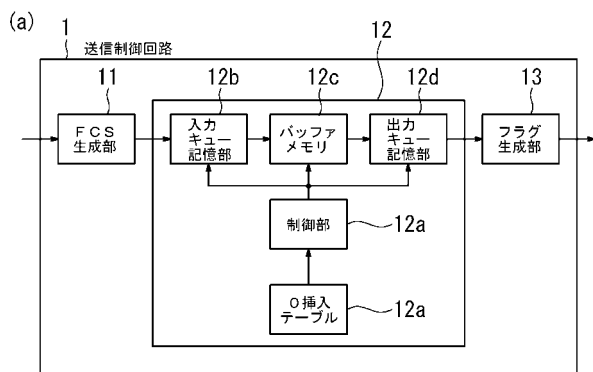
- 【図 3 7】 従来の開始フラグ検出処理のフローチャートである。
 【図 3 8】 従来の 0 削除処理のフローチャートである。
 【図 3 9】 従来の開始フラグ検出処理の状態変化を示す図である。
 【図 4 0】 従来の開始フラグ検出処理の状態変化を示す図である。
 【図 4 1】 従来の 0 削除処理の状態変化を示す図である。
 【図 4 2】 従来の 0 削除処理の状態変化を示す図である。
 【図 4 3】 従来の 0 削除処理の状態変化を示す図である。
 【図 4 4】 従来の 0 削除処理の状態変化を示す図である。
 【図 4 5】 従来の 0 削除処理の状態変化を示す図である。
 【図 4 6】 従来の 0 削除処理の状態変化を示す図である。
 【図 4 7】 従来の 0 削除処理の状態変化を示す図である。
 【図 4 8】 従来の 0 削除処理の状態変化を示す図である。
 【図 4 9】 従来の 0 削除処理の状態変化を示す図である。
 【図 5 0】 従来の 0 削除処理の状態変化を示す図である。
 【図 5 1】 従来の 0 削除処理の状態変化を示す図である。
 【符号の説明】
 【0 0 6 6】

1 2 . . . 0 挿入回路、1 2 a, 2 1 a . . . 制御部、1 2 b, 2 1 b . . . 入力キュー記憶部、1 2 d, 2 1 d . . . 出力キュー記憶部、1 2 e . . . 0 挿入テーブル、2 1 . . . 0 削除及びフラグ検出部、2 1 e . . . 0 削除及びフラグ検出テーブル。

10

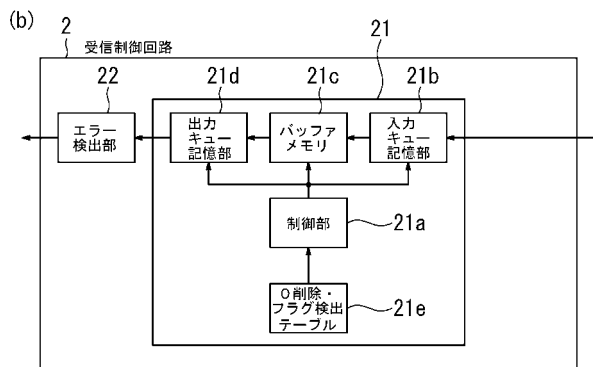
20

【図 1】



【図 2】

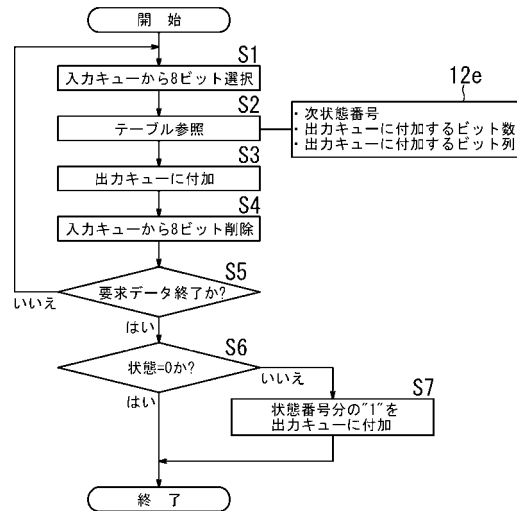
状態番号	残りビット
0	なし
1	1
2	11
3	111
4	1111



【図 3】

状態番号	入力ビット列	次状態番号	出力ビット数	出力ビット列
0	0x58 (01011000)	0	8	01011000
	0x59 (01011001)	1	7	0101100
	0x5A (01011010)	0	8	01011010
	0x5B (01011011)	2	6	010110
	0x5C (01011100)	0	8	01011100
	0x5D (01011101)	1	7	0101110
1	0x5E (01011110)	0	8	01011110
	0xF4 (11110100)	0	10	111100100
	0xF5 (11110101)	1	9	11110010
	0xF6 (11110110)	0	10	111100110
	0xF7 (11110111)	3	7	111100
	0xF8 (11110000)	0	10	111101000
2	0xF9 (11110001)	1	9	11110100
	0xFA (11110010)	0	10	111101010
	0xB9 (10111001)	1	9	111011100
	0xBA (10111010)	0	10	1110111010
	0xBB (10111011)	2	8	11101110
	0xBC (10111100)	0	10	1110111100
3	0xBD (10111101)	1	9	111011110
	0xBE (10111110)	0	11	11101111100
	0xBF (10111111)	1	10	1110111110
	0xE8 (11101000)	0	12	111110101000
	0xE9 (11101001)	1	11	11111010100
	0xEA (11101010)	0	12	111110101010
4	0xEB (11101011)	2	10	1111101010
	0xEC (11101100)	0	12	111110101100
	0xED (11101101)	1	11	11111010110
	0xEE (11101110)	0	12	111110101110
	0xEF (11101111)	4	8	1111010
	0xFA (11110100)	0	12	111110111010
5	0xFB (11110101)	2	10	1111101110
	0xFC (11110110)	0	12	111110111100
	0xFD (11110111)	1	11	11111011110
	0xFE (11111000)	0	13	1111101111100
	0xFF (11111001)	1	12	111110111110
	0xFF (11111010)	1	12	111110111110

【図 4】



【図 5】

(a) 1回目	状態: 0 (初期状態)
入力キュー	01011011101111011110111110111111101111111111111111
(b) テーブル参照結果	次状態: 2 出力ビット数: 6 出力ビット列: 010110
(c) 1回目終了	
入力キュー	01011011101111011110111110111111101111111111111111
出力キュー	010110

【図 6】

(a) 2回目	状態: 2
入力キュー	01011011101111011110111110111111101111111111111111
(b) テーブル参照結果	次状態: 1 出力ビット数: 9 出力ビット列: 111011110
(c) 2回目終了	
入力キュー	01011011101111011110111110111111101111111111111111
出力キュー	010110111011110

【図 9】

(a) 5回目	状態: 4
入力キュー	01011011101111011110111110111111101111111111111111
(b) テーブル参照結果	次状態: 3 出力ビット数: 10 出力ビット列: 1111011110
(c) 5回目終了	
入力キュー	01011011101111011110111110111111101111111111111111
出力キュー	0101101110111101111001111101011111011111011110

【図 7】

(a) 3回目	状態: 1
入力キュー	01011011101111011110111110111111101111111111111111
(b) テーブル参照結果	次状態: 3 出力ビット数: 7 出力ビット列: 1111100
(c) 3回目終了	
入力キュー	01011011101111011110111110111111101111111111111111
出力キュー	010110111011110111100

【図 10】

(a) 6回目	状態: 3
入力キュー	01011011101111011110111110111111101111111111111111
(b) テーブル参照結果	次状態: 1 出力ビット数: 12 出力ビット列: 11111011110
(c) 6回目終了	
入力キュー	01011011101111011110111110111111101111111111111111
出力キュー	0101101110111101111001111101011111011111011111011110

【図 8】

(a) 4回目	状態: 3
入力キュー	01011011101111011110111110111111101111111111111111
(b) テーブル参照結果	次状態: 4 出力ビット数: 8 出力ビット列: 11111010
(c) 4回目終了	
入力キュー	01011011101111011110111110111111101111111111111111
出力キュー	01011011101111011110011111010

【図 11】

最終処理	状態: 1
入力キュー	
出力キュー	01011011101111011110011111010111110111110111110111101

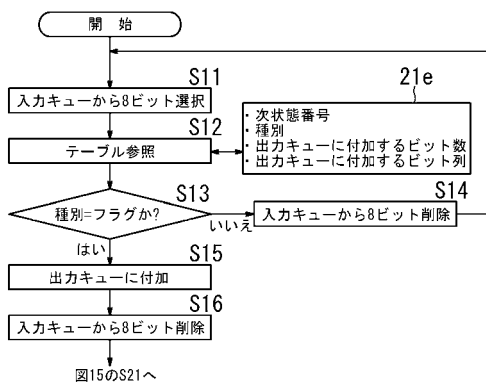
【図 1 2】

状態番号	残りビット
0	なし
1	0
2	1
3	01
4	11
5	011
6	111
7	0111
8	1111
9	01111
10	11111
11	011111
12	0111111

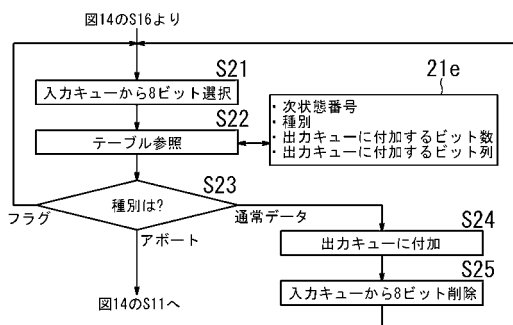
【図 1 3】

状態番号	入力ビット列	次状態番号	データ種別	出力ビット数	出力ビット列
0	0xBB(10111011)	5	通常	5	10111
	0xBC(10111100)	1	通常	7	1011110
	0xBD(10111101)	3	通常	6	101111
	0xBE(10111110)	1	通常	7	1011111
	0xBF(10111111)	12	通常	1	1
1	0xF6(11110110)	1	通常	1	0
	0xF7(11110111)	7	通常	4	0111
	0xF8(11110000)	1	通常	1	0
	0xF9(11110001)	3	通常	2	01
	0xFA(11110010)	1	通常	1	0
	0xFB(11110011)	4	通常	2	11
	0xFC(11110100)	1	フラグ	1	0
2	0xF2(11110010)	1	通常	1	0
	0xF3(11110011)	5	通常	3	011
	0xF4(11110100)	1	通常	1	0
	0xF5(11110101)	3	通常	2	01
	0xF6(11110110)	1	通常	1	0
	0xF7(11110111)	6	通常	3	111
	0xF8(11110000)	0	無効		-
3	0xDA(11011010)	1	通常	1	0
	0xDB(11011011)	5	通常	3	011
	0xDC(11011100)	1	通常	1	0
	0xDD(11011101)	3	通常	2	01
	0xDE(11011110)	1	通常	1	0
	0xDF(11011111)	11	通常	6	011111
3	0xF2(11110010)	1	通常	1	0
	0xF3(11110011)	5	通常	3	011
	0xF4(11110100)	1	通常	1	0
	0xF5(11110101)	3	通常	2	01
	0xF6(11110110)	1	通常	1	0
	0xF7(11110111)	6	通常	3	111
	0xF8(11110000)	1	通常	1	0
	0xF9(11110001)	3	フラグ	2	01
	0xFA(11110010)	1	フラグ	1	0
	0xFB(11110011)	5	フラグ	3	011
6	0x7A(01111010)	1	通常	1	0
	0x7B(01111011)	5	通常	3	011
	0x7C(01111100)	1	通常	1	0
	0x7D(01111101)	2	通常	1	1
	0x7E(01111110)	0	フラグ	0	-
	0x7F(01111111)	0	無効	-	-
12	0x2A(00101010)	1	フラグ	1	0
	0x2B(00101011)	5	フラグ	3	011
	0x2C(00101100)	1	フラグ	1	0
	0x2D(00101101)	3	フラグ	2	01
	0x2E(00101110)	1	フラグ	1	0
	0x2F(00101111)	9	フラグ	5	01111

【図 1 4】



【図 1 5】



【図 1 6】

(a) 1回目	状態: 0 (初期状態)
入力キュー	101111100101101110111101111001111010111101110111101111011110111110
(b) テーブル参照結果	次状態: 12 種別: 通常 出力ビット数: 1 出力ビット列: 1
(c) 1回目終了	
入力キュー	101111100101101110111101111001111010111110111011101111011110111110
出力キュー	NULL

【図 1 7】

(a) 2回目	状態: 12
入力キュー	1011111001011011101111001111010111101110111011110111101111011110
(b) テーブル参照結果	次状態: 3 種別: フラグ 出力ビット数: 5 出力ビット列: 01011
(c) 2回目終了	
入力キュー	1011111001011011101111011110011110101111011101110111101111011110
出力キュー	01011

【図 1 8】

(a) 3回目	状態: 3
入力キュー	1011111001011011101111001111010111101110111011110111101111011110
(b) テーブル参照結果	次状態: 1 種別: 通常 出力ビット数: 9 出力ビット列: 01101111
(c) 3回目終了	
入力キュー	1011111001011011101111001111010111101110111011110111101111011110
出力キュー	01011011101111

【図 1 9】

(a) 4回目	状態: 1
入力キュー	1011111001011011101111001111010111101110111011110111101111011110
(b) テーブル参照結果	次状態: 3 種別: 通常 出力ビット数: 6 出力ビット列: 011111
(c) 4回目終了	
入力キュー	1011111001011011101111001111010111101110111011110111101111011110
出力キュー	010110111011101111

【図 20】

(a) 5回目	状態: 3
入力キュー	10111110010101101110111100111101011101110111010111110
(b) テーブル参照結果	次状態: 3 種別: 通常 出力ビット数: 7 出力ビット列: 0111111
(c) 5回目終了	
入力キュー	10111110010101101110111100111101011101110111010111110
出力キュー	010110111011110111101111011111

【図 21】

(a) 6回目	状態: 3
入力キュー	1011111001010110111011110011110101110111010111110
(b) テーブル参照結果	次状態: 6 種別: 通常 出力ビット数: 6 出力ビット列: 0111111
(c) 6回目終了	
入力キュー	1011111001010110111011110011110101110111010111110
出力キュー	010110111011110111101111011111

【図 22】

(a) 7回目	状態: 6
入力キュー	1011111001010110111011110011110101110111010111110
(b) テーブル参照結果	次状態: 2 種別: 通常 出力ビット数: 9 出力ビット列: 111011111
(c) 7回目終了	
入力キュー	1011111001010110111011110011110101110111010111110
出力キュー	010110111011110111101111101111101111

【図 23】

(a) 8回目	状態: 2
入力キュー	1011111001010110111011110011110101110111010111110
(b) テーブル参照結果	次状態: 3 種別: 通常 出力ビット数: 6 出力ビット列: 1111111
(c) 8回目終了	
入力キュー	1011111001010110111011110011110101110111010111110
出力キュー	0101101110111101111011111011111011111111

【図 27】

(a) 1回目	
入力キュー	0101101110111011110111110111110111110111111111111
(b) テーブル参照結果	出力ビット数: 6 削除ビット数: 6 出力ビット列: 010110
(c) 1回目終了	
入力キュー	0101101110111101111011111011111101111110111111111
出力キュー	010110

【図 28】

(a) 2回目	
入力キュー	0101101110111101111011111011111101111110111111111
(b) テーブル参照結果	出力ビット数: 4 削除ビット数: 4 出力ビット列: 1110
(c) 2回目終了	
入力キュー	0101101110111101111011111011111101111110111111111
出力キュー	0101101110

【図 29】

(a) 3回目	
入力キュー	0101101110111101111011111011111101111110111111111
(b) テーブル参照結果	出力ビット数: 5 削除ビット数: 5 出力ビット列: 11110
(c) 3回目終了	
入力キュー	0101101110111101111011111011111101111110111111111
出力キュー	010110111011110

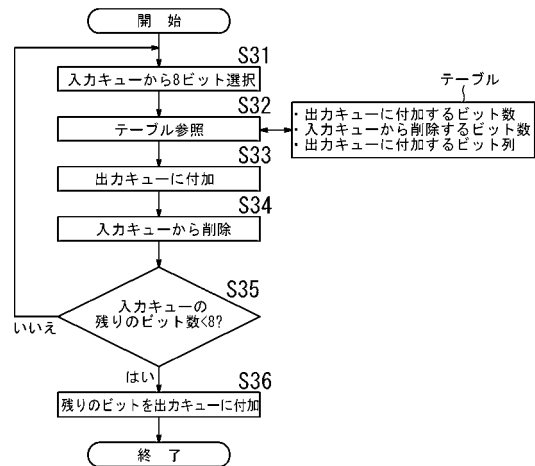
【図 24】

(a) 9回目	状態: 3
入力キュー	1011111001010110111011110011110101110111010111110
(b) テーブル参照結果	次状態: xx 種別: 75% 出力ビット数: 0 出力ビット列: 無し
(c) 9回目終了	
入力キュー	1011111001010110111011110011110101110111010111110
出力キュー	0101101110111101111011111011111011111101111111111

【図 25】

フラグ (8)	アドレス部 (8)	制御部 (8/16)	情報部 (任意長)	FCS (16)	フラグ (8)
01111110					01111110

【図 26】



【図 30】

(a) 4回目	
入力キュー	0101101110111101111011111011111101111110111111111
(b) テーブル参照結果	出力ビット数: 7 削除ビット数: 6 出力ビット列: 1111100
(c) 4回目終了	
入力キュー	0101101110111101111011111011111101111110111111111
出力キュー	010110111011110111101111100

【図 31】

(a) 5回目	
入力キュー	0101101110111101111011111011111101111110111111111
(b) テーブル参照結果	出力ビット数: 8 削除ビット数: 7 出力ビット列: 11111010
(c) 5回目終了	
入力キュー	0101101110111101111011111011111101111110111111111
出力キュー	01011011101111011110011111010

【図 32】

(a) 6回目	
入力キュー	0101101110111101111011111011111101111110111111111
(b) テーブル参照結果	出力ビット数: 6 削除ビット数: 5 出力ビット列: 111110
(c) 6回目終了	
入力キュー	0101101110111101111011111011111101111110111111111
出力キュー	0101101110111101111001111101011110

【 図 3 3 】

(a) 7回目
入力キュー

(b) テーブル参照結果 出力ビット数：4 削除ビット数：4 出力ビット列：1110

(c) 7回目終了
入力キュー
出力キュー

【図 3 4】

[illegible]

【图 3 5】

(a)	9回目 入力キュー	010110111011110111110111111011111101111111111111111111111111xx
(b)	テーブル参照結果	出力ビット数 : 6 削除ビット数 : 5 出力ビット列 : 111110
(c)	9回目終了 入力キュー 出力キュー	010110111011110111111011111101111110111111111111111111111111xx 0101011011101111011110011110101111011011101101110111101111111110

【図 3 6】

[illegible]

【图 3 9】

(a) 1回目
入力キュー 101111100101101101110111100111101011110110111101111011110111110

(b) テーブル参照結果 出力ビット数: 1 削除ビット数: 1 データ種類: 通常

(c) 1回目終了
入力キュー +0111111001011011011101111001111010111101101111011110111110
出力キュー NULL

【图 4 0】

(a) 2回目
入力キュー

(b) テーブル参照結果 出力ビット数：0 削除ビット数：8 データ種類 フラグ

(c) 2回目終了
入力キュー
出力キュー

【图 4 1】

[illegible]

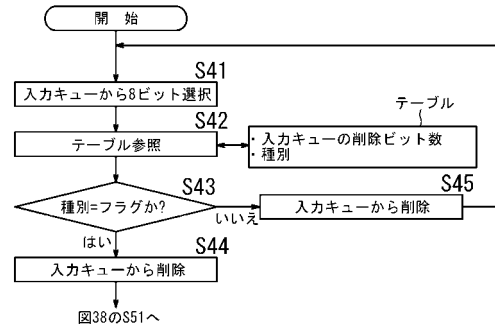
【图 4-2】

(a) 4回目
入力キュー

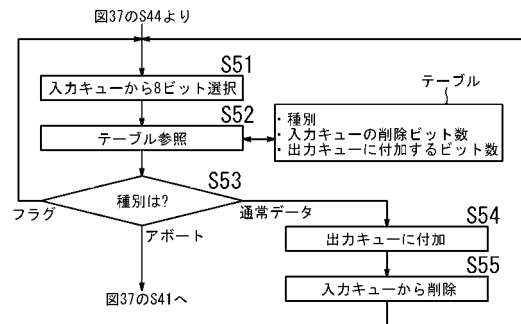
(b) テーブル参照結果 出力ビット数: 4 | 削除ビット数: 4 | データ種類: 通常

(c) 4回目終了
入力キュー
出力キュー

【図 3 7】



【 図 3 8 】



【 図 4 3 】

(a)	5回目 入力キュー	+011111100+0110110111011111010111101101111011110111110
(b)	テーブル参照結果	出力ビット数：5 削除ビット数：5 データ種類：通常
(c)	5回目終了 入力キュー 出力キュー	+011111100+01101101110111110011110101111011011110111110111110111110 01011011101111

【 図 4 4 】

(a) 6回目
入力キュー 1011111001011011101111111011111010111101110111011101111011110111110111110111110

(b) テーブル参照結果 出力ビット数：6 削除ビット数：7 データ種類：通常

(c) 6回目終了
入力キュー 1011111001011011101111101111101011110111011101111011110111101011111010111110
出力キュー 01011011011101111

【图 4 5】

(a) 7回目
入力キュー

(b) テーブル参照結果 出力ビット数: 6 削除ビット数: 7 データ種類: 通常

(c) 7回目終了
入力キュー
出力キュー

【图 4 6】

(a) 8回目
入力キュー 101111100101101110111100111101111011110111101111011110111101111011110111101111110

(b) テーブル参照結果 出力ビット数：7 削除ビット数：8 データ種類：通常

(c) 8回目終了
入力キュー 1011111001011011101111001111010111101111011110111101111011110111101111011110111110
出力キュー 0101101110111101111101111110111111011111

【图 4 7】

(a) 9回目
入力キュー 101111100101101110111100111101011110111011110111101011110

(b) テーブル参照結果 出力ビット数: 3 削除ビット数: 3 データ種類: 通常

(c) 9回目終了
入力キュー 10111110010110111011110011110101111011110111101111010111110
出力キュー 010110110111111101111101111101111111

【图 48】

(a) 10回目
入力キュー

(b) テーブル参照結果 出力ビット数：6 削除ビット数：7 データ種類：通常

(c) 10回目終了
入力キュー
出力キュー

【图 4 9】

[illegible]

【 ㊦ 5 0 】

(a)

12回目	
入力キュー	40111110010110111011110011110401111011101111011110101111010111110

(b) テーブル参照結果 出力ビット数：1 削除ビット数：1 データ種類：通常

(c)

12回目終了	
入力キュー	40111110010110111011110011110401111011101111011110101111040111110
出力キュー	0101101110111110111101111011111101111110111111111

【図 5 1】

(a) 13回目
入力キュー

(b) テーブル参照結果 出力ビット数: 0 削除ビット数: 8 データ種類: フラグ

(c) 13回目終了
入力キュー
出力キュー

フロントページの続き

(72)発明者 黒田 浩司

東京都目黒区下目黒二丁目1番7号テレコムクリエイトシステム株式会社内

(72)発明者 宮崎 真克

東京都目黒区下目黒二丁目2番3号株式会社田村電機製作所内

Fターム(参考) 5K034 AA02 FF02 FF13 HH05 HH12 KK24 MM29 PP07