(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6045720号
(P6045720)
(45)発行日　平成28年12月14日(2016.12.14)　(24)登録日　平成28年11月25日(2016.11.25)

(51)Int.Cl.　F I
H03M 7/30 (2006.01)　H03M 7/30 Z

請求項の数 14　(全 20 頁)

(21)出願番号	特願2015-553054 (P2015-553054)
(86)(22)出願日	平成26年1月14日 (2014.1.14)
(65)公表番号	特表2016-509416 (P2016-509416A)
(43)公表日	平成28年3月24日 (2016.3.24)
(86)国際出願番号	PCT/EP2014/050517
(87)国際公開番号	WO2014/111351
(87)国際公開日	平成26年7月24日 (2014.7.24)
審査請求日	平成27年9月15日 (2015.9.15)
(31)優先権主張番号	13305055.9
(32)優先日	平成25年1月18日 (2013.1.18)
(33)優先権主張国	欧州特許庁 (EP)

(73)特許権者　391030332
アルカテル－ルーセント
フランス国、９２１００・ブローニュ－ビヤンクール、ルート・ドゥ・ラ・レーヌ・１４８／１５２
(74)代理人　100094112
弁理士　岡部　讓
(74)代理人　100106183
弁理士　吉澤　弘司
(74)代理人　100114915
弁理士　三村　治彦
(74)代理人　100120363
弁理士　久保田　智樹
(74)代理人　100125139
弁理士　岡部　洋

最終頁に続く

(54)【発明の名称】データ圧縮

(57)【特許請求の範囲】
【請求項１】
キャラクタの入力シーケンス（Ｉ）の、出力シーケンス（Ｏ）へのデータ圧縮のための方法であって、前記入力シーケンス（Ｉ）の中の前記キャラクタは、ｌ＋ｍ＝ｋであり、それぞれが2^m個の連続するコードからなる2^l個の範囲の形で配列される2^k個のコードの組のｋビットのコードによって表され、前記方法は、
－　符号化されるべきキャラクタ（Ｉ（ｉ））のｍ個の最下位ビットが、少なくとも１つの所定のｍビットのタグ（ＳＴ）とは、異なっており、また符号化されるべき前記キャラクタ（Ｉ（ｉ））の前記範囲（Ｒ（ｉ））が、前記入力シーケンス（Ｉ）の中の以前のキャラクタの前記範囲（Ｒ（ｉ－１））と同じである場合に、符号化されるべき前記キャラクタ（Ｉ（ｉ））のｍ個の最下位ビットに対応するｍビットの値（Ｖ（ｉ））を有する前記出力シーケンスへと、符号化されるべき前記キャラクタ（Ｉ（ｉ））を符号化するステップ（Ｓ９）
を含み、
－　そうでない場合に、前記少なくとも１つの所定のｍビットのタグ（ＳＴ）と、符号化されるべき前記キャラクタ（Ｉ（ｉ））の前記範囲（Ｒ（ｉ））を示すｌビットと、符号化されるべき前記キャラクタ（Ｉ（ｉ））のｍ個の最下位ビットに対応するｍビットの値（Ｖ（ｉ））とを有する前記出力シーケンス（Ｏ）へと、符号化されるべき前記キャラクタ（Ｉ（ｉ））を符号化するステップ（Ｓ３）
を含む方法。

【請求項２】
ｋ＝８であり、ｌ＝３であり、またｍ＝５である、請求項１に記載の方法。
【請求項３】
各範囲Ｒは、第１のフラグ（Ｓ０）と、第２のフラグ（Ｓ１）とを含んでおり、前記第１のフラグ（Ｓ０）のｍ個の最下位ビットは、第１の所定のタグに等しく、前記第２のフラグ（Ｓ１）のｍ個の最下位ビットは、第２の所定のタグに等しい、請求項１または２のいずれか１項に記載の方法。
【請求項４】
符号化されるべき前記キャラクタ（Ｉ（ｉ））の関数の形で前記第１の所定のタグと、前記第２の所定のタグとのうちの一方を選択するステップ（Ｓ１）を含む、請求項３に記載の方法。
【請求項５】
前記第１の所定のタグは、ｍビットの「０」から成り、また前記第２の所定のタグは、ｍビットの「１」から成る、請求項３または４のいずれか１項に記載の方法。
【請求項６】
複数の入力シーケンス（Ｉ）を含むテキスト（Ｔ）を圧縮するための方法であって、
各入力シーケンスに対して、
－　請求項１乃至５のいずれか１項に記載のデータ圧縮のための前記方法に従って、前記入力シーケンス（Ｉ）の関数の形で、出力シーケンス（Ｏ）を決定するステップ（Ｔ１）と、
－　前記入力シーケンス（Ｉ）と、前記出力シーケンス（Ｏ）とのうちの最も短いものを選択するステップ（Ｔ２、Ｔ３、Ｔ４）と、
－　前記選択されたシーケンス（Ｓ）を出力するステップ（Ｔ５）と
を含む方法。
【請求項７】
前記決定するステップ（Ｔ１）は、以前の入力シーケンス（Ｉ）に対応する前記選択されたシーケンス（Ｓ）を出力するステップ（Ｔ５）と同時に起こる、請求項６に記載の方法。
【請求項８】
複数の入力シーケンス（Ｉ）を含むテキスト（Ｔ）を圧縮するための方法であって、
以前の入力シーケンスを処理した後に非圧縮モードが選択されなかった各入力シーケンスに対して、
－　請求項１乃至５のいずれか１項に記載のデータ圧縮のための前記方法に従って、前記入力シーケンス（Ｉ）の関数の形で出力シーケンス（Ｏ）を決定するステップ（Ｕ２）と、
－　前記出力シーケンス（Ｏ）を出力するステップ（Ｕ２）と、
－　前記出力シーケンスが、前記入力シーケンスよりも短くない場合に、次の入力シーケンス（Ｉ）については非圧縮モード（Ｍ）を選択するステップ（Ｕ４）と
を含む方法。
【請求項９】
前記出力シーケンス（Ｏ）を決定するステップ（Ｕ２）と、前記出力シーケンス（Ｏ）を出力するステップ（Ｕ２）とは、同時に実行される、請求項８に記載の方法。
【請求項１０】
キャラクタの入力シーケンス（Ｉ）の、出力シーケンス（Ｏ）へのデータ圧縮のためのデバイス（２）であって、前記入力シーケンス（Ｉ）の中の前記キャラクタは、ｌ＋ｍ＝ｋであり、それぞれが2^{m}個の連続するコードからなる2^{l}個の範囲の形で配列される2^{k}個のコードの組のｋビットのコードによって表され、データ圧縮のための前記デバイス（２）は、
－　符号化されるべきキャラクタ（Ｉ（ｉ））のｍ個の最下位ビットが、少なくとも１つの所定のｍビットのタグ（ＳＴ）とは、異なっており、また符号化されるべき前記キャラ

クタ（Ｉ（ｉ））の前記範囲（Ｒ（ｉ））が、前記入力シーケンス（Ｉ）の中の以前のキャラクタの前記範囲（Ｒ（ｉ－１））と同じである場合にアクティブにされる、符号化されるべき前記キャラクタ（Ｉ（ｉ））のｍ個の最下位ビットに対応するｍビットの値（Ｖ（ｉ））を有する前記出力シーケンス（Ｏ）へと、符号化されるべき前記キャラクタ（Ｉ（ｉ））を符号化するための手段（Ｓ９）と、
－　そうでない場合にアクティブにされる、前記少なくとも１つの所定のｍビットのタグ（ＳＴ）と、符号化されるべき前記キャラクタ（Ｉ（ｉ））の前記範囲（Ｒ（ｉ））を示すｌビットと、符号化されるべき前記キャラクタ（Ｉ（ｉ））のｍ個の最下位ビットに対応するｍビットの値（Ｖ（ｉ））とを有する前記出力シーケンス（Ｏ）へと、符号化されるべき前記キャラクタ（Ｉ（ｉ））を符号化するための手段（Ｓ３）と
を備えているデバイス。
【請求項１１】
命令が、コンピュータによって実行されるときに、請求項１乃至９のいずれか１項に記載の方法を実行するための前記命令を含むコンピュータ・プログラム。
【請求項１２】
ビットのシーケンス（Ｏ）の、キャラクタの解凍されたシーケンス（Ｉ’）へのデータ解凍のための方法であって、前記解凍されたシーケンス（Ｉ’）の中の前記キャラクタは、ｌ＋ｍ＝ｋであり、それぞれが2^{m}個の連続するコードからなる2^{l}個の範囲の形で配列される2^{k}個のコードの組のｋビットのコードによって表され、前記方法は、
－　ビットの前記シーケンス（Ｏ）のｍ個の連続するビットが、少なくとも１つの所定のタグ（ＳＴ）に等しい場合に、
－　ビットの前記シーケンス（Ｏ）のｌ個の次のビットの関数の形で、復号されるべきキャラクタ（Ｉ’（ｉ））の前記範囲（Ｒ（ｉ））を決定するステップ（Ｖ１２、Ｖ１３）と、
－　ビットの前記シーケンス（Ｏ）の前記ｍ個の次のビットの関数の形で、復号されるべきキャラクタ（Ｉ’（ｉ））の値（Ｖ（ｉ））を決定するステップ（Ｖ１４、Ｖ１１）と、
－　前記範囲（Ｒ（ｉ））と、前記値（Ｖ（ｉ））との関数の形で、復号されるべき前記キャラクタ（Ｉ’（ｉ））を決定するステップ（Ｖ７）と
を含み、
－　そうでない場合に、
－　ビットの前記シーケンス（Ｏ）の前記ｍ個の連続するビットの関数の形で、復号されるべきキャラクタ（Ｉ’（ｉ））の前記値（Ｖ（ｉ））を決定するステップ（Ｖ１１）と、
－　以前のキャラクタの前記範囲と、前記値（Ｖ（ｉ））との関数の形で、復号されるべきキャラクタ（Ｉ’（ｉ））を決定するステップ（Ｖ７）と
を含む方法。
【請求項１３】
ビットのシーケンス（Ｏ）の、キャラクタの解凍されたシーケンス（Ｉ’）へのデータ解凍のためのデバイス（３）であって、前記解凍されたシーケンス（Ｉ’）の中の前記キャラクタは、ｌ＋ｍ＝ｋであり、それぞれが2^{m}個の連続するコードからなる2^{l}個の範囲の形で配列される2^{k}個のコードの組のｋビットのコードによって表され、データ解凍のための前記デバイス（３）は、
－　ビットの前記シーケンス（Ｏ）のｌ個の次のビットの関数の形で、復号されるべきキャラクタ（Ｉ’（ｉ））の前記範囲（Ｒ（ｉ））を決定し（Ｖ１２、Ｖ１３）、
－　ビットの前記シーケンス（Ｏ）のｍ個の次のビットの関数の形で、復号されるべきキャラクタ（Ｉ’（ｉ））の値（Ｖ（ｉ））を決定し（Ｖ１４、Ｖ１１）、また
－　前記範囲（Ｒ（ｉ））と、前記値（Ｖ（ｉ））との関数の形で、復号されるべき前記キャラクタ（Ｉ’（ｉ））を決定する（Ｖ７）
－　ための、ビットの前記シーケンス（Ｏ）のｍ個の連続するビットが、少なくとも１つ

の所定のタグ（ＳＴ）に等しい場合にアクティブにされる、第１の手段と、
－　ビットの前記シーケンス（Ｏ）の前記ｍ個の連続するビットの関数の形で、復号されるべきキャラクタ（Ｉ’（ｉ））の前記値（Ｖ（ｉ））を決定し（Ｖ１１）、また
－　以前のキャラクタの前記範囲と、前記値（Ｖ（ｉ））との関数の形で、復号されるべきキャラクタ（Ｉ’（ｉ））を決定する（Ｖ７）
－　ための、そうでない場合にアクティブにされる、第２の手段と
を備えているデバイス。
【請求項１４】
命令が、コンピュータによって実行されるときに、請求項１２に記載の方法を実行するための前記命令を含むコンピュータ・プログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、データ圧縮の分野に関する。
【背景技術】
【０００２】
データ圧縮は、様々な状況のために役に立つ。例えば、通信は、多くの場合に、異なるデバイスの間で比較的大量のデータを転送することを伴う。データ転送は、それを１つのデバイスから別のデバイスへと送信する前に、データを圧縮することによって合理化される可能性がある。データ圧縮は、ある与えられた帯域幅の上で通信され得る情報の量を増大させることができる。様々なデータ圧縮技法が、知られている。
【０００３】
いくつかの知られているデータ圧縮技法は、必ずしもより小さなファイル、またはより少量の情報をもたらすとは限らないこともある。例えば、小さなファイルの場合には、いくつかのデータ圧縮技法は、どのようなサイズの縮小も生じない。そのような場合には、データ圧縮技法は、ファイル・サイズの観点からどのような節約も提供することなく、追加のプロセッサ要件を加える。
【発明の概要】
【発明が解決しようとする課題】
【０００４】
したがって、改善されたデータ圧縮の方法およびデバイスを提案することが、本発明の実施形態の一目的であり、これらの方法およびデバイスは、先行技術の固有の短所を示すものではない。本発明の実施形態についての別の目的は、対応するデータ解凍の方法およびデバイスを提案することである。
【課題を解決するための手段】
【０００５】
それに応じて、本発明の実施形態は、キャラクタの入力シーケンスの、出力シーケンスへのデータ圧縮のための一方法に関し、そこでは、入力シーケンスの中のキャラクタは、ｌ＋ｍ＝ｋである、2^{m}個の連続するコードの2^{l}個の範囲の形で配列される2^{k}個のコードの組のｋビットのコードによって表され、本方法は、
－　符号化されるべきキャラクタのｍ個の最下位ビットが、少なくとも１つの所定のｍビットのタグとは、異なっており、また符号化されるべきキャラクタの範囲が、入力シーケンスの中の以前のキャラクタの範囲と同じである場合に、符号化されるべきキャラクタのｍ個の最下位ビットに対応するｍビットの値を有する出力シーケンスへと、符号化されるべき前記キャラクタを符号化するステップ
を含み、
－　そうでない場合に、前記少なくとも１つの所定のｍビットのタグと、符号化されるべきキャラクタの範囲を示すｌビットと、符号化されるべきキャラクタのｍ個の最下位ビットに対応するｍビットの値とを有する出力シーケンスへと、符号化されるべき前記キャラクタを符号化するステップ

を含む。
【０００６】
一実施形態においては、ｋ＝８であり、ｌ＝３であり、またｍ＝５である。
【０００７】
各範囲Ｒは、第１のフラグと、第２のフラグとを含むことができ、第１のフラグのｍ個の最下位ビットは、第１の所定のタグに等しく、第２のフラグのｍ個の最下位ビットは、第２の所定のタグに等しい。
【０００８】
本方法は、符号化されるべきキャラクタの関数の形で第１の所定のタグと、第２の所定のタグとのうちの一方を選択するステップを含むことができる。
【０００９】
例えば、第１の所定のタグは、ｍビットの「０」から成り、また第２の所定のタグは、ｍビットの「１」から成る。
【００１０】
本発明の実施形態は、複数の入力シーケンスを含むテキストを圧縮するための方法に関し、
－　本発明の一実施形態のデータ圧縮のための方法に従って、入力シーケンスのうちの１つの関数の形で、出力シーケンスを決定するステップと、
－　入力シーケンスと、出力シーケンスとのうちの最も短いものを選択するステップと、
－　選択されたシーケンスを出力するステップと
を含む。
【００１１】
本方法は、
－　以前の入力シーケンスに対応する選択されたシーケンスを出力するステップと同時に、本発明の一実施形態のデータ圧縮のための方法に従って、入力シーケンスのうちの１つの関数の形で出力シーケンスを決定するステップ
を含むことができる。
【００１２】
本発明の実施形態はまた、複数の入力シーケンスを含むテキストを圧縮するための一方法に関し、
－　本発明の一実施形態のデータ圧縮のための方法に従って、入力シーケンスのうちの１つの関数の形で出力シーケンスを決定するステップと、
－　出力シーケンスを出力するステップと、
－　出力シーケンスが、入力シーケンスよりも短くない場合に、次の入力シーケンスについては非圧縮モードを選択するステップと
を含む。
【００１３】
出力シーケンスを決定するステップと、出力シーケンスを出力するステップとは、同時に実行されることもある。
【００１４】
本発明の実施形態はまた、キャラクタの入力シーケンスの、出力シーケンスへのデータ圧縮のためのデバイスに関し、そこでは、入力シーケンスの中のキャラクタは、ｌ＋ｍ＝ｋである、2^{m}個の連続するコードの2^{l}個の範囲の形で配列される2^{k}個のコードの組のｋビットのコードによって表され、データ圧縮のためのデバイスは、
－　符号化されるべきキャラクタのｍ個の最下位ビットが、少なくとも１つの所定のｍビットのタグとは、異なっており、また符号化されるべきキャラクタの範囲が、入力シーケンスの中の以前のキャラクタの範囲と同じである場合にアクティブにされる、符号化されるべきキャラクタのｍ個の最下位ビットに対応するｍビットの値を有する出力シーケンスへと、符号化されるべき前記キャラクタを符号化するための手段と、
－　そうでない場合にアクティブにされる、前記少なくとも１つの所定のｍビットのタグ

と、符号化されるべきキャラクタの範囲を示すｌビットと、符号化されるべきキャラクタのｍ個の最下位ビットに対応するｍビットの値とを有する出力シーケンスへと、符号化されるべき前記キャラクタを符号化するための手段と
を備えている。

【００１５】

本発明の実施形態はまた、ビットのシーケンスの、キャラクタの解凍されたシーケンスへのデータ解凍のための一方法に関し、そこでは、解凍されたシーケンスの中のキャラクタは、ｌ＋ｍ＝ｋである、2^m個の連続するコードの2^l個の範囲の形で配列される2^k個のコードの組のｋビットのコードによって表され、本方法は、
－　ビットのシーケンスのｍ個の連続するビットが、少なくとも１つの所定のタグに等しい場合に、
－　ビットのシーケンスのｌ個の次のビットの関数の形で、復号されるべきキャラクタの範囲を決定するステップと、
－　ビットのシーケンスのｍ個の次のビットの関数の形で、復号されるべきキャラクタの値を決定するステップと、
－　前記範囲と、前記値との関数の形で、復号されるべきキャラクタを決定するステップと
を含み、
－　そうでない場合に、
－　ビットのシーケンスの前記ｍ個の連続するビットの関数の形で、復号されるべきキャラクタの値を決定するステップと、
－　以前のキャラクタの範囲と、前記値との関数の形で、復号されるべきキャラクタを決定するステップと
を含む。

【００１６】

本発明の実施形態はまた、ビットのシーケンスの、キャラクタの解凍されたシーケンスへのデータ解凍のためのデバイスに関し、そこでは、解凍されたシーケンスの中のキャラクタは、ｌ＋ｍ＝ｋである、2^m個の連続するコードの2^l個の範囲の形で配列される2^k個のコードの組のｋビットのコードによって表され、データ解凍のためのデバイスは、
－　ビットのシーケンスのｌ個の次のビットの関数の形で、復号されるべきキャラクタの範囲を決定し、
－　ビットのシーケンスのｍ個の次のビットの関数の形で、復号されるべきキャラクタの値を決定し、また
－　前記範囲と、前記値との関数の形で、復号されるべきキャラクタを決定する
－　ための、ビットのシーケンスのｍ個の連続するビットが、少なくとも１つの所定のタグに等しい場合にアクティブにされる、第１の手段と、
－　ビットのシーケンスの前記ｍ個の連続するビットの関数の形で、復号されるべきキャラクタの値を決定し、また
－　以前のキャラクタの範囲と、前記値との関数の形で、復号されるべきキャラクタを決定する
－　ための、そうでない場合にアクティブにされる、第２の手段と
を備えている。

【００１７】

本発明の実施形態はまた、命令が、コンピュータによって実行されるときに、前述の方法のうちの１つを実行するための前記命令を含むコンピュータ・プログラムを提供している。

【００１８】

添付の図面と併せて解釈される実施形態についての以下の説明を参照することにより、本発明についての上記および他の目的と、特徴とは、より明らかになり、また本発明、それ自体は、最もよく理解されるであろう。

【図面の簡単な説明】
【００１９】
【図１】本発明の実施形態において使用される符号化テーブルを示す図である。
【図２】本発明の一実施形態による、データを圧縮するためのデバイスの機能的表現を示す図である。
【図３】本発明の一実施形態による、データを解凍するためのデバイスの機能的表現を示す図である。
【図４】図２および／または３のデバイスの構造的表現を示す図である。
【図５】本発明の実施形態による、データを圧縮するための一方法のフローチャートである。
【図６】非同期モードにおいてデータを圧縮するための一方法のフローチャートであり、これは、図５の方法を使用している。
【図７】同期モードにおいてデータを圧縮するための一方法のフローチャートであり、これは、図５の方法を使用している。
【図８】本発明の実施形態による、データを解凍するための一方法のフローチャートである。
【発明を実施するための形態】
【００２０】
図面において示される様々な要素の機能は、専用のハードウェア、ならびに適切なソフトウェアに関連してソフトウェアを実行することができるハードウェアの使用を通して提供されることもあることに注意すべきである。プロセッサによって提供されるときに、それらの機能は、単一の専用プロセッサにより、単一の共用プロセッサにより、またはそれらのプロセッサのうちのいくつかが共用され得る複数の個別のプロセッサにより、提供されることもある。さらに、用語「プロセッサ」の明示的な使用は、ソフトウェアを実行することができるハードウェアだけを排他的に意味するように解釈されるべきではなく、また限定することなく、デジタル信号プロセッサ（ＤＳＰ：ｄｉｇｉｔａｌ　ｓｉｇｎａｌ　ｐｒｏｃｅｓｓｏｒ）ハードウェアと、ネットワーク・プロセッサと、特定用途向け集積回路（ＡＳＩＣ：ａｐｐｌｉｃａｔｉｏｎ　ｓｐｅｃｉｆｉｃ　ｉｎｔｅｇｒａｔｅｄ　ｃｉｒｃｕｉｔ）と、フィールド・プログラマブル・ゲート・アレイ（ＦＰＧＡ：ｆｉｅｌｄ　ｐｒｏｇｒａｍｍａｂｌｅ　ｇａｔｅ　ａｒｒａｙ）と、ソフトウェアを記憶するためのリード・オンリー・メモリ（ＲＯＭ：ｒｅａｄ　ｏｎｌｙ　ｍｅｍｏｒｙ）と、ランダム・アクセス・メモリ（ＲＡＭ：ｒａｎｄｏｍ　ａｃｃｅｓｓ　ｍｅｍｏｒｙ）と、不揮発性ストレージとを暗黙のうちに含むことができる。他のハードウェアが、従来のものおよび／またはカスタムもまた、含められることもある。それらの機能は、プログラム・ロジックのオペレーションを通して、専用のロジックを通して、プログラム制御と専用のロジックとの相互作用を通して、または手動でさえも実行されることもあり、特定の技法は、文脈からもっと具体的に理解されるように、実装者によって選択可能である。
【００２１】
本明細書における任意のブロック図は、本発明の原理を実施する実例となる回路の概念図を表すことが、当業者によってさらに理解されるべきである。同様に、任意のフローチャートは、コンピュータ読取り可能媒体の形で実質的に表され、またそのようにして、そのようなコンピュータまたはプロセッサが明示的に示されているか否かにかかわらず、コンピュータまたはプロセッサによって実行され得る様々なプロセスを表すことが、理解されるであろう。
【００２２】
図１は、キャラクタを符号化するために使用される符号化テーブル１を示すものであり、そこでは、各キャラクタは、対応する８ビットのコードによって符号化される。図１は、コードの１６進数の値を示すものである。例えば、キャラクタ「ａ」は、１６進数においてはｘ６１として符号化され、または２進数においては０１１０００００１として符号化される。以下では、本発明者等は、対応するコードのビットについて言及する代わりに、

手短にキャラクタのビットについて言及する可能性がある。
【００２３】
符号化テーブル１においては、キャラクタは、３２個の連続するコードの８個の範囲Ｒの形で配列される。例えば、範囲Ｒのうちの１つは、キャラクタ
【００２４】
【数１】
「Ä」
（ｘ４０）からキャラクタ
【００２５】
【数２】
「î」

（ｘ５Ｆ）へと進む。さらに各範囲Ｒは、図１に示されるように、２つのフラグに、すなわち偶数フラグＳ０と、奇数フラグＳ１とに関連づけられる。偶数フラグＳ０は、範囲Ｒの最初のキャラクタであり、また奇数フラグＳ１は、範囲Ｒの最後のキャラクタである。人は、偶数フラグＳ０が、同じ５つの最下位ビット、すなわち、００００００を共有し、また奇数フラグＳ１が、同じ５つの最下位ビット、すなわち、１１１１１を共有することを認識するであろう。さらに、人は、範囲Ｒの中のすべてのキャラクタが、同じ３つの最上位ビットを共有することを認識するであろう。
【００２６】
図１の上に示される符号化テーブル１は、よく知られている拡張されたＡＳＣＩＩテーブルの修正されたバージョンである。より正確には、拡張されたＡＳＣＩＩテーブルに関して、
－　４つのキャラクタは、すなわち、｛と、｜と、｝と、～とは、
【００２７】
【数３】
é

と、
【００２８】
【数４】
è

と、
【００２９】
【数５】
à

と、ＳＰとにそれぞれ変更され、
－　２つのキャラクタは、すなわち、＾は、@に相互に変更され、
－　４つのキャラクタは、すなわち、？と、＿と、～と、＾とは、
【００３０】
【数６】
Ï

と、
【００３１】
【数７】
Î

と、
【００３２】
【数８】

と、
【００３３】
【数９】

とにそれぞれ変更される。
【００３４】
ＡＳＣＩＩテーブルのこの修正されたバージョンは、多くの場合に、通常のテキスト・メッセージにおいて互いに近くに現れるキャラクタが、同じ範囲Ｒの中に配置されるという効果を有しており、これは、後で現れるように、改善された圧縮効率をもたらす。しかしながら、本発明の実施形態は、別の符号化テーブルを、例えば、標準の拡張されたＡＳＣＩＩテーブルを使用することができる。さらに、説明は、３２個の連続するコードの８個の範囲の形で配列される８ビットのコードを使用する符号化テーブル１を参照して継続されるであろう。しかしながら、本発明の実施形態は、他の任意の符号化テーブルを使用することができ、そこでは、2^k個のキャラクタは、ｌ＋ｍ＝ｋである、2^m個の連続するコードの2^l個の範囲の形で配列される2^k個のコードの組のうちからのｋビットのコードを用いて符号化される。
【００３５】
図２は、データ圧縮のためのデバイス２の機能的表現である。データ圧縮のためのデバイス２は、キャラクタの入力シーケンスＩを出力シーケンスＯへと圧縮する。同様に、図３は、データ解凍のためのデバイス３の機能的表現である。データ解凍のためのデバイス３は、データ圧縮のためのデバイス２によって提供される出力シーケンスＯを最初の入力シーケンスＩに等しい解凍されたシーケンスＩ’へと解凍する。
【００３６】
図４は、データ圧縮のためのデバイス２および／またはデータ解凍のためのデバイス３とすることができるデバイスの構造的表現である。図４のデバイスは、プロセッサ４と、不揮発性メモリ５と、揮発性メモリ６と、通信インターフェース７とを備えている。プロセッサ４は、揮発性メモリ６をワーキング・メモリとして使用して、不揮発性メモリ５に記憶されるコンピュータ・プログラムを実行することを可能にする。通信インターフェース７は、通信リンクまたはネットワークを通して少なくとも１つの他のデバイスと通信することを可能にする。
【００３７】
不揮発性メモリ５は、コンピュータ・プログラムＰ１、および／またはコンピュータ・プログラムＰ２を含むことができ、コンピュータ・プログラムＰ１の実行は、データを圧縮するための方法を実施し、コンピュータ・プログラムＰ２の実行は、データを解凍するための方法を実施する。
【００３８】
図４のデバイスは、例えば、パーソナル・コンピュータ、モバイル電話などのハンドヘルド端末、通信ネットワークにおけるサーバ．．．とすることができる。
【００３９】
図５は、データ圧縮のためのデバイス２によって実行される、データを圧縮するための一方法のフローチャートである。図５の方法は、ｉが、０からＬ－１に及ぶキャラクタＩ（ｉ）の入力シーケンスＩを出力シーケンスＯへと圧縮することを目指している。
【００４０】
本方法は、カウンタｉが、０（ステップＳ０）で初期化されて、開始される。以下では

、符号化されるべきキャラクタは、Ｉ（ｉ）と示される。

【００４１】

次いで、５ビットの同期化タグＳＴが、キャラクタＩ（ｉ）の関数の形で、決定される（ステップＳ１）。同期化タグＳＴは、データを圧縮するときに使用される同期モードを表しており、すなわち、ＳＴ＝１１１１１は、奇数同期モードに対応しており、またＳＴ＝００００００は、偶数同期モードに対応している。

【００４２】

ステップＳ１において、符号化されるべきキャラクタＩ（ｉ）が、偶数フラグＳ０のうちの１つである場合、偶数同期モードが、選択される（ＳＴ＝００００００）。それとは反対に、符号化されるべきキャラクタＩ（ｉ）が、奇数フラグＳ１のうちの１つである場合、奇数同期モードが、選択される（ＳＴ＝１１１１１）。初めてステップ１が、実行されるときに、符号化されるべきキャラクタＩ（ｉ）は、フラグＳ０と、Ｓ１とのうちの一方でないこともある。その場合には、デフォルト同期モードが、選択される。デフォルト同期モードは、偶数同期モード、または奇数同期モードとすることができる。

【００４３】

次いで、符号化されるべきキャラクタＩ（ｉ）の範囲Ｒ（ｉ）と値Ｖ（ｉ）とが、決定される（ステップＳ２）。範囲Ｒ（ｉ）は、キャラクタＩ（ｉ）の３つの最上位ビットに対応しており、また値Ｖ（ｉ）は、キャラクタＩ（ｉ）の５つの最下位ビットに対応している。

【００４４】

符号化されるべきキャラクタＩ（ｉ）は、同期化タグＳＴと、範囲Ｒ（ｉ）と、値Ｖ（ｉ）との連続に対応する１３ビットによって出力シーケンスＯの中で符号化される（ステップＳ３）。これは、図５の上にＯ＝Ｏ＋ＳＴ＋Ｒ＋Ｖとして示される。

【００４５】

その後に、データ圧縮のためのデバイス２は、入力シーケンスＩが、完全に圧縮されているかどうかを決定する（ステップＳ４）。入力シーケンスＩが、完全に圧縮されている場合、そのときには本方法は、終了する。そうでない場合には、カウンタｉは、１だけ増分され（ステップＳ５）、また本方法は、符号化されるべき次のキャラクタＩ（ｉ）を符号化するステップへと進む。

【００４６】

データを圧縮するためのデバイス２は、符号化されるべきキャラクタＩ（ｉ）が、フラグＳ０またはＳ１のうちの一方であるかどうかを決定する。肯定的な場合には、符号化されるべきキャラクタＩ（ｉ）は、以上で説明されるように、ステップＳ１からＳ３に従って符号化される。それとは反対に、符号化されるべきキャラクタＩ（ｉ）が、フラグＳ０またはＳ１のうちの一方でない場合、そのときには符号化されるべきキャラクタＩ（ｉ）の範囲Ｒ（ｉ）と、値Ｖ（ｉ）とは、ステップＳ２と同様な方法で決定される（ステップＳ７）。

【００４７】

次に、符号化されるべきキャラクタの範囲Ｒ（ｉ）は、以前のキャラクタＩ（ｉ－１）の範囲Ｒ（ｉ－１）と比較され、この以前のキャラクタＩ（ｉ－１）は、ステップＳ２またはＳ７のいずれかの以前の実行において、それ以前に決定されている。

【００４８】

符号化されるべきキャラクタの範囲Ｒ（ｉ）が、以前のキャラクタの範囲Ｒ（ｉ－１）と同じである場合、そのときには符号化されるべきキャラクタＩ（ｉ）は、値Ｖ（ｉ）に対応する５ビットだけ、出力シーケンスＯにおいて符号化される（ステップＳ９）。これは、図５の上にＯ＝Ｏ＋Ｖとして示される。次いで、データ圧縮のためのデバイス２は、以前に説明されるステップＳ４へと進む。

【００４９】

それとは反対に、符号化されるべきキャラクタの範囲Ｒ（ｉ）が、以前のキャラクタの範囲Ｒ（ｉ－１）とは異なる場合、そのときにはデータ圧縮のためのデバイス２は、ステ

ップＳ３へと進み、そこでは、符号化されるべきキャラクタＩ（ｉ）は、同期化タグＳＴと、範囲Ｒ（ｉ）と、値Ｖ（ｉ）との連続に対応する１３ビットだけ、出力シーケンスにおいて符号化される（ステップＳ３）。

【００５０】

図５の圧縮方法によれば、入力シーケンスＩの８ビットのキャラクタＩ（ｉ）が、

－　フラグＳ０と、Ｓ１とのうちの一方であるキャラクタについての、また以前のキャラクタとは異なる範囲の中のキャラクタについての、第１のキャラクタＩ（０）についての１３ビット、または

－　他の場合における５ビット

のうちの一方だけ、出力シーケンスＯにおいて符号化される。

【００５１】

図５の方法の圧縮利得Ｇは、１３ビットを用いて符号化されるキャラクタと、５ビットを用いて符号化されるキャラクタとの間の比に依存する。ある場合には、利得Ｇは、負であることもある。しかしながら、入力シーケンス１が、図１の上に示される第３の範囲と第４の範囲とに属している、ほとんど、連続するアルファベット・キャラクタから成る通常のテキスト・メッセージであることを仮定すると、利得Ｇは、一般的に正であろう。さらに、負の利得の効果は、図６および７を参照して以下で説明される非同期モードの方法、または同期モードの方法で制限される可能性がある。

【００５２】

さらに、キャラクタＩ（ｉ）を符号化することは、全体の入力シーケンスＩについての知識を必要としてはいない。対照的に、以前のキャラクタＩ（ｉ－１）の範囲Ｒ（ｉ－１）と、現在の同期化タグＳＴとが、知られていることを仮定すると、キャラクタＩ（ｉ）が、個別に符号化されることもある。言い換えれば、図５の方法は、シリアルの（キャラクタごとの）圧縮技法を提供している。キャラクタＩ（ｉ）が、連続して（例えば、送信システムにおいて）受信されるときに、これは、完全な入力シーケンスＩが、受信される前に符号化Ｉ（ｉ）を可能にする。出力シーケンスＯの（データ圧縮のためのデバイス２についての考えられたアプリケーションに応じた）バッファリングすること、記憶すること、および／または送信は、完全な入力シーケンスＩが、受信され、かつ／または圧縮される前に開始される可能性もある。

【００５３】

図６は、非同期モードに従って、連続するシーケンスＩから成るテキストＴを圧縮するための方法のフローチャートである。図６の方法は、データ圧縮のためのデバイス２によって実行される。

【００５４】

シーケンスＩは、ステップＴ１からＴ５に従って、逐次的に処理される。

【００５５】

シーケンスＩは、図５を参照して上記で説明されるシリアルな圧縮技法に従って、出力シーケンスＯへと圧縮される（ステップＴ１）。ステップＴ１は、シーケンスＩをバッファリングしながら／受信しながら、実行されることもある。

【００５６】

次いで、ビットの単位の、出力シーケンスＯの長さＬ（Ｏ）は、ビットの単位の、入力シーケンスＩの長さＬ（Ｉ）と比較される（ステップＴ２）。長さＬ（Ｏ）と、Ｌ（Ｉ）との両方は、入力シーケンスＩにおいて受信されるビットと、出力シーケンスＯに加えられるビットとをカウントすることにより、ステップＴ１において決定されることもある。

【００５７】

選択されたシーケンスＳは、この比較に応じて選択される。より正確には、出力シーケンスＯが、入力シーケンスＩよりも短い場合、言い換えれば、ステップＴ１の圧縮方法が、正の利得Ｇを有する（Ｌ（Ｏ）＜Ｌ（Ｉ））場合、そのときには出力シーケンスＯが、選択される（ステップＴ３）。それとは反対に、出力シーケンスＯが、入力シーケンスＩよりも短くない場合、そのときには入力シーケンスＩが、選択される（ステップＴ４）。

言い換えれば、ステップＴ１において、出力シーケンスＯを決定した後に、より短いシーケンスが、ステップＴ２～Ｔ４において選択される。

【００５８】

最終的に、選択されたシーケンスＳは、出力され、例えば、それは、データ圧縮のためのデバイス２のアプリケーションに応じて、バッファリングされ、記憶され、かつ／または送信される（ステップＴ５）。

【００５９】

選択されたシーケンスの出力（ステップＴ５）は、テキストＴの次の入力シーケンスＩの圧縮（ステップＴ１）と並行して起こる可能性がある。

【００６０】

図６の非同期モードによれば、テキストＴを圧縮することが、対応する入力シーケンスＩと同じ長さか、または対応する入力シーケンスＩよりも短い長さかのいずれかを有するときに、選択されたシーケンスＳは、出力する。その結果として、たとえ入力シーケンスのいくつかが、負の利得Ｇを用いて、ステップＴ１において圧縮されるとしても、テキストＴを圧縮するためのグローバル利得は、正になるであろう。

【００６１】

図６の非同期モードにおいては、入力シーケンスは、シーケンスＯと、Ｉとのうちの一方が、選択され、また出力され得る（ステップＴ２～Ｔ５の）前に、完全に受信され、圧縮される必要がある（ステップＴ１）。それゆえに、非同期モードは、以上で論じられるように、シリアルな圧縮を完全に利用することができない。

【００６２】

図７は、同期モードに従って、連続するシーケンスＩから成るテキストＴを圧縮するための方法のフローチャートである。図７の方法は、データ圧縮のためのデバイス２によって実行される。

【００６３】

モードＭは、３つの以下の値、すなわち、

－　入力シーケンスＩが、圧縮されるべきではないことを意味している、ＷＳ、

－　入力シーケンスＩが、図５のステップＳ１の第１の実行についてのデフォルト・モードとして偶数同期モードを使用することにより圧縮されるべきであることを意味している、Ｓ０、または

－　入力シーケンスＩが、図５のステップＳ１の第１の実行についてのデフォルト・モードとして奇数同期モードを使用することにより圧縮されるべきであることを意味している、Ｓ１

のうちの１つを取ることができる。

【００６４】

モードＭは、ＷＳ、Ｓ０またはＳ１において、初期化されることもある。次いで、シーケンスＩは、ステップＵ１からＵ６に従って、逐次的に処理される。

【００６５】

最初に、データを圧縮するためのデバイス２は、モードＭが、ＷＳまたはＳ０／Ｓ１であるかどうかを決定した（ステップＵ１）。

【００６６】

モードＭが、Ｓ０とＳ１とのうちの一方である場合、そのときにはシーケンスＩは、図５を参照して上記で説明されるシリアルな圧縮技法に従って、出力シーケンスＯへと圧縮される（ステップＵ２）。ステップＵ２の圧縮のために、図５のステップＳ１において使用されるデフォルト・モード（偶数または奇数）は、モードＭによって指定される。

【００６７】

ステップＵ２は、シーケンスＩをバッファリングしながら／受信しながら、実行されることもある。さらに、出力シーケンスＯは、ステップＵ１中に、漸進的に出力されることもあり、例えば、それは、（データ圧縮のためのデバイス２のアプリケーションに応じて）バッファリングされ、記憶され、かつ／または送信されることもある。

【００６８】
次いで、ビットの単位の、出力シーケンスＯの長さＬ（Ｏ）は、ビットの単位の、入力シーケンスＩの長さＬ（Ｉ）と比較される（ステップＵ３）。長さＬ（Ｏ）と、Ｌ（Ｉ）との両方は、入力シーケンスＩにおいて受信されるビットと、出力シーケンスＯに加えられるビットとをカウントすることにより、ステップＵ１において決定されることもある。
【００６９】
次の入力シーケンスＩを処理するためのモードＭは、この比較に応じて選択される。より正確には、出力シーケンスＯが、入力シーケンスＩよりも短い場合、言い換えれば、ステップＵ１の圧縮方法が、正の利得Ｇを有する（Ｌ（Ｏ）＜Ｌ（Ｉ））場合、そのときにはモードＭ＝Ｓ０と、Ｍ＝Ｓ１とのうちの一方が、選択される（ステップＵ５）。それとは反対に、出力シーケンスＯが、入力シーケンスＩよりも短くない場合、そのときにはモードＭ＝ＷＳが、選択される（ステップＵ４）。
【００７０】
対照的に、ステップＵ１において、モードＭが、ＷＳである場合、そのときには入力シーケンスＩは、圧縮なしで出力され（ステップＵ６）、また次いで、次のシーケンスのためのモードＭは、以前に説明されるステップＵ５において、Ｓ０またはＳ１に設定される。
【００７１】
ステップＵ４またはＵ５の後に、次の入力シーケンスＩの処理は、このシーケンスＩのために選択されるモードＭを用いてステップＵ１において開始される。
【００７２】
図７の同期モード圧縮技法によれば、出力シーケンスＯは、入力シーケンスＩの受信と圧縮と並行して、漸進的に出力される（すなわち、バッファリングされ、記憶され、送信される．．．）こともある。
【００７３】
出力シーケンスＯは、対応する入力シーケンスＩよりも長いこともあり、言い換えれば、ステップＵ２のシリアルな圧縮の利得Ｇは、負であることもある。
【００７４】
入力シーケンスＩを圧縮するための利得Ｇが、負である場合、そのときには次の入力シーケンスＩを圧縮するための利得Ｇもまた、負になることもある可能性が高いことが、仮定されることもある。両方の入力シーケンスＩが、様々な特別なキャラクタが範囲Ｒの変更を伴うテキストＴの一部分に関連している場合に、これは、事例となる可能性がある。この観点から、図７の方法によれば、利得が、ステップＵ３において負のＧであるときに、モードＭ＝ＷＳが、ステップＵ４において選択され、また次の入力シーケンスＩは、圧縮されず（ステップＵ１およびＵ６）、それによって利得Ｇ＝０を提供している。これは、テキストＴを圧縮するための総利得に対する入力シーケンスＩを圧縮する、可能性のある負の利得Ｇの影響を制限することを可能にする。
【００７５】
図７の例においては、１つの入力シーケンスＩだけが、ステップＵ６の後に圧縮されず、次いで、モードＭは、ステップＵ５において、Ｓ０またはＳ１へと戻され、また次の入力シーケンスＩは、再び圧縮される。別の例（図示されず）においては、モードＭ＝Ｓ０、またはＭ＝Ｓ１へと戻る前に、複数の入力シーケンスが、圧縮されないこともある。
【００７６】
図８は、データ解凍のためのデバイス３によって実行される、データを解凍するための一方法のフローチャートである。図８の方法は、出力シーケンスＯを決定するために、データ圧縮のためのデバイス２によって使用される入力シーケンスＩに等しい解凍されたシーケンスＩ’へと、データ圧縮のためのデバイス２によって提供される出力シーケンスＯを解凍することを目指している。
【００７７】
以下の表記法が、図８の上で使用されており、すなわち、

－　Ｉ’（ｉ）は、解凍されたシーケンスＩ’のｉ番目の８ビットのキャラクタのことを意味しており、ｉは、０からＬ－１に及んでおり、また

－　Ｏ［ｊ；ｊ＋ｘ］は、ｊ番目のビットから開始される、出力シーケンスＯのｘビットのことを意味している。

【００７８】

言い換えれば、カウンタｉは、８ビットのキャラクタのことを意味しており、またカウンタｊは、個別のビットのことを意味している。

【００７９】

最初に、カウンタｉと、カウンタｊとは０（ステップＶ０）において初期化される。

【００８０】

次いで、データを解凍するためのデバイス３は、出力シーケンスＯの最初の５ビットＯ［ｊ；ｊ＋４］が、同期化タグＳＴに対応するかどうかを決定する（ステップＶ１）。実際に、図５のデータを圧縮するための方法についての説明から明らかなように、出力シーケンスＯの最初の５ビットＯ［ｊ；ｊ＋４］は、通常、同期化タグ１１１１１と、０００００とのうちの一方であるべきである。Ｏ［ｊ；ｊ＋４］が、１１１１１または０００００とは異なる場合、エラーが、起こる（ステップＶ２）。

【００８１】

エラーが、起こらなかった場合、そのときには出力シーケンスＯの次の３ビットを使用して、キャラクタＩ’（ｉ）の範囲Ｒを決定する。これは、図８の上に、５だけｊを増分すること（ステップＶ３）、およびＯ［ｊ；ｊ＋２］の関数の形でＲを決定すること（ステップＶ４）として示されている。

【００８２】

その後に、出力シーケンスＯの次の５ビットを使用して、キャラクタＩ’（ｉ）の値Ｖを決定する。これは、図８の上に、３だけｊを増分すること（ステップＶ５）、およびＯ［ｊ；ｊ＋４］の関数の形でＶを決定すること（ステップＶ６）として示されている。

【００８３】

次いで、キャラクタＩ’（ｉ）は、範囲Ｒと、値Ｖとの関数、すなわち、Ｉ’（ｉ）＝Ｒ＋Ｖの形で決定される（ステップＶ７）。

【００８４】

次いで、カウンタｉと、カウンタｊとは、次のキャラクタを処理するために増分され、すなわち、ｉ＝ｉ＋１と、ｊ＝ｊ＋５となる（ステップＶ９）。

【００８５】

その後に、データを解凍するためのデバイス３は、出力シーケンスＯの次の５ビットＯ［ｊ；ｊ＋４］が、同期化タグＳＴに対応するかどうかを決定する（ステップＶ１０）。

【００８６】

Ｏ［ｊ；ｊ＋４］が、同期化タグ１１１１１と、０００００とのうちの一方でない場合、それは、キャラクタＩ’（ｉ）が、以前のキャラクタＩ’（ｉ－１）と同じ範囲Ｒの中にあり、またフラグＳ０と、Ｓ１とのうちの一方ではなく、またその５ビットの値Ｖだけによって符号化されていることを意味している。それゆえに、値Ｖは、Ｏ［ｊ；ｊ＋４］の関数の形で決定される（ステップＶ１１）。次いで、ステップＶ７において、キャラクタＩ’（ｉ）は、以前のキャラクタＩ’（ｉ）についてと同じ範囲Ｒを使用することによって決定される。

【００８７】

対照的に、Ｏ［ｊ；ｊ＋４］が、同期化タグ１１１１１と、０００００とのうちの一方である場合、それは、キャラクタＩ’（ｉ）が、以前のキャラクタＩ’（ｉ－１）とは異なる範囲Ｒの中にあり、かつ／またはフラグＳ０と、Ｓ１とのうちの一方であり、また５ビットの同期化タグと、３ビットの範囲Ｒと、５ビットの値Ｖとによって符号化されていることを意味している。したがって、出力シーケンスＯの次の３ビットを使用して、キャラクタＩ’（ｉ）の範囲Ｒを決定する。これは、図８の上に、５だけｊを増分すること（ステップＶ１２）、およびＯ［ｊ；ｊ＋２］の関数の形でＲを決定すること（ステップＶ

１３）として示されている。その後に、出力シーケンスＯの次の５ビットを使用して、キャラクタＩ’（ｉ）の値Ｖを決定する。これは、図８の上に、３だけｊを増分すること（ステップＶ１４）、およびＯ［ｊ；ｊ＋４］の関数の形でＶを決定すること（ステップＶ１１）として示されている。最後に、ステップＶ７において、キャラクタＩ’（ｉ）は、ステップＶ１３のアップデートされた範囲Ｒを使用することによって決定される。

【００８８】

もちろん、ステップＶ７からＶ１４のループは、出力シーケンスＯが、完全に処理されているときに、終了する。これは、例えば、ステップＶ８におけるテストによって決定される。

【００８９】

図８のデータを解凍するための方法は、図５のデータを圧縮する方法によって圧縮される出力シーケンスＯを解凍することを可能にする。同期化タグ１１１１１と、００００００との使用は、キャラクタＩ’（ｉ）が、出力シーケンスＯの５ビットの値Ｖだけに基づいて決定されるべきであるか、または５ビットの同期化タグＳＴと、３ビットの範囲Ｒと、５ビットの値Ｖとを含む１３ビットのグループに基づいて決定されるべきであるかを決定することを可能にする。

【００９０】

以下の表は、４６４ビットである、５８個のキャラクタの入力シーケンス＜＜あるメッセージが、電話番号８８８における貴方のボイスメールの上で待っている＞＞を圧縮する一例を与えるものである。キャラクタは、符号化テーブル１を用いて符号化される。それゆえに、とりわけ、空間キャラクタは、ｘ２０の代わりにｘ７Ｅとして符号化される。

【００９１】

【表１－１】

入力 シーケンスIの キャラクタ	入力 シーケンスIのコード （16進数）	出力シーケンスOのビット
A	41	1111 1010 0 0001
	7E	1111 1011 1 1110
m	6D	0 1101
e	65	0 0101
s	73	1 0011
s	73	1 0011
a	61	0 0001
g	67	0 0111
e	65	0 0101
	7E	1 1110
i	69	0 1001
s	73	1 0011
	7E	1 1110
w	77	1 0111
a	61	0 0001
i	69	0 1001
t	74	1 0100
i	69	0 1001
n	6E	0 1110
g	67	0 0111
	7E	1 1110
o	6F	0 1111
n	6E	0 1110
	7E	1 1110

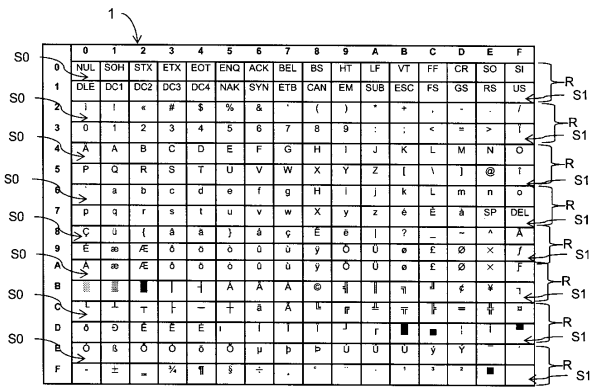
【表１－２】

y	79	1 1001
o	6F	0 1111
u	75	1 0101
r	72	1 0010
	7E	1 1110
v	76	1 0110
o	6F	0 1111
i	69	0 1001
c	63	0 0011
e	65	0 0101
m	6D	0 1101
a	61	0 0001
i	69	0 1001
l	6C	0 1100
	7E	1 1110
a	61	0 0001
t	74	1 0100
	7E	1 1110
p	70	1 0000
h	68	0 1000
o	6F	0 1111
n	6E	0 1110
e	65	0 0101
	7E	1 1110
n	6E	0 1110
u	75	1 0101
m	6D	0 1101
b	62	0 0010
e	65	0 0101
r	72	1 0010
	7E	1 1110
8	38	1111 1001 1 1000
8	38	1 1000
8	38	1 1000

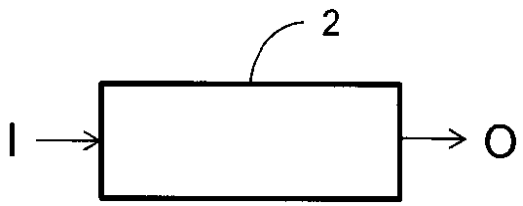
【００９２】
出力シーケンスＯは、入力シーケンスＩの４６４ビットと比べて、３１４ビットを含んでいる。利得Ｇは、（（４６４－３１４）／４６４）＝３２．３％である。
【００９３】
本方法の実施形態は、専用のハードウェアおよび／またはソフトウェア、あるいは両方の任意の組合せによって実行される可能性がある。
【００９４】
本発明の原理は、特定の実施形態に関連して上記で説明されているが、この説明は、例としてのみ行われ、また、添付の特許請求の範囲において規定される本発明の範囲に対す

る限定として行われてはいないことを明確に理解すべきである。

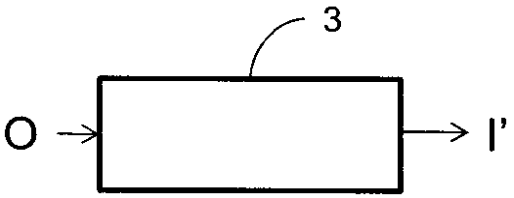
【図１】



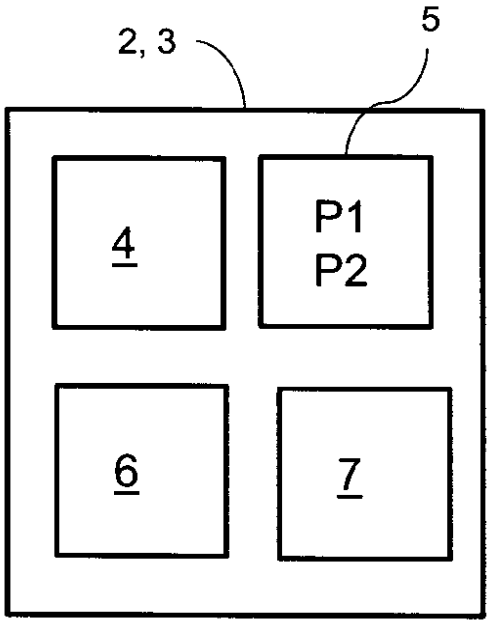
【図２】



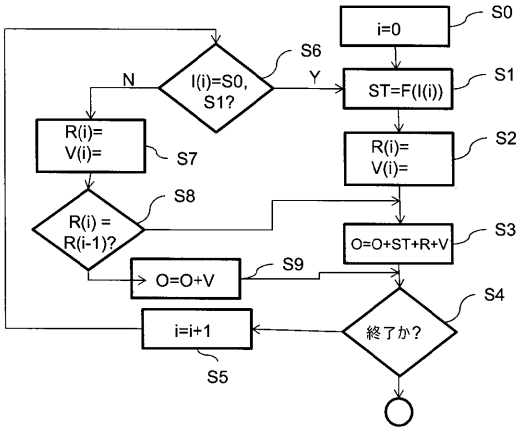
【図３】



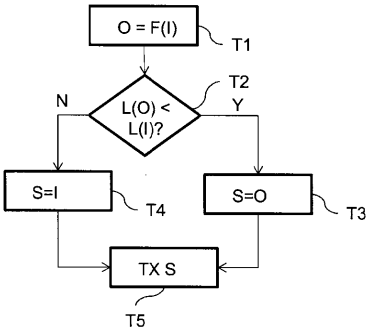
【図４】



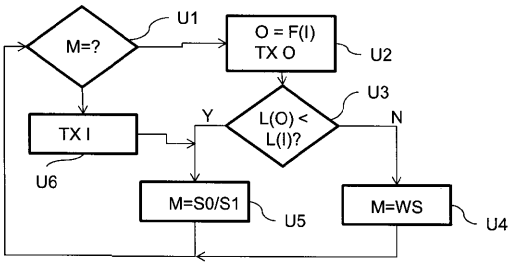
【図５】


i=0
S0
I(i)=S0, S1?
S6
N
Y
ST=F(I(i))
S1
R(i)= V(i)=
S7
R(i)= V(i)=
S2
R(i) = R(i-1)?
S8
O=O+ST+R+V
S3
O=O+V
S9
終了か?
S4
i=i+1
S5

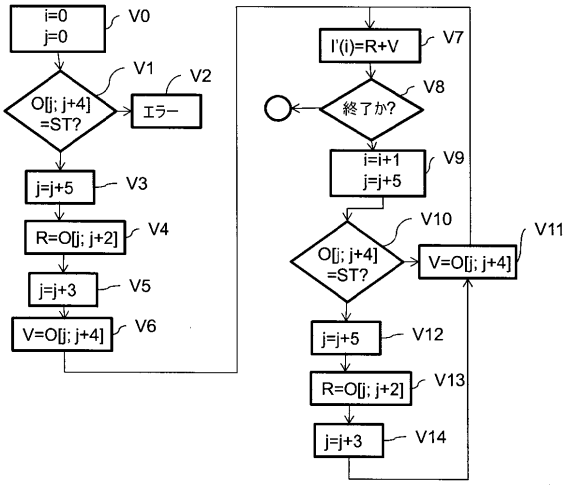
【図６】


O = F(I)
T1
L(O) < L(I)?
T2
N
Y
S=I
T4
S=O
T3
TX S
T5

【図７】


M=?
U1
O = F(I) TX O
U2
TX I
U6
Y
L(O) < L(I)?
U3
N
M=S0/S1
U5
M=WS
U4

【図８】


i=0 j=0
V0
O[j; j+4] =ST?
V1
V2
エラー
j=j+5
V3
R=O[j; j+2]
V4
j=j+3
V5
V=O[j; j+4]
V6
I'(i)=R+V
V7
終了か?
V8
i=i+1 j=j+5
V9
O[j; j+4] =ST?
V10
V=O[j; j+4]
V11
j=j+5
V12
R=O[j; j+2]
V13
j=j+3
V14

フロントページの続き

(72)発明者　デネス，アラン
フランス　エフ－７８１４０　ヴェリジー，リュ　フロンヴァル，３

審査官　北村　智彦

(56)参考文献　特開平０４－０４９４１４（ＪＰ，Ａ）
特開平０４－３６７０２７（ＪＰ，Ａ）
特開２０００－０３２１１７（ＪＰ，Ａ）
米国特許第０５４４４４４５（ＵＳ，Ａ）
特開昭５８－１１９２３０（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０３Ｍ　　３／００－１１／００