

(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2016-184830

(P2016-184830A)

(43) 公開日 平成28年10月20日(2016.10.20)

(51) Int.Cl. F I テーマコード (参考)
 H03M 7/30 (2006.01) H03M 7/30 Z 5J064

審査請求 未請求 請求項の数 10 O L (全 40 頁)

(21) 出願番号 特願2015-63449 (P2015-63449)
 (22) 出願日 平成27年3月25日 (2015.3.25)

(出願人による申告)平成21年度~平成25年度、独立行政法人科学技術振興機構、戦略的創造研究推進事業(個人型研究(さきがけ))に関する委託研究、産業技術力強化法第19条の適用を受ける特許出願

(71) 出願人 504171134
 国立大学法人 筑波大学
 茨城県つくば市天王台一丁目1番1
 (71) 出願人 504174135
 国立大学法人九州工業大学
 福岡県北九州市戸畑区仙水町1番1号
 (74) 代理人 100100549
 弁理士 川口 嘉之
 (74) 代理人 100123319
 弁理士 関根 武彦
 (72) 発明者 山際 伸一
 茨城県つくば市天王台一丁目1番1 国立
 大学法人筑波大学内

最終頁に続く

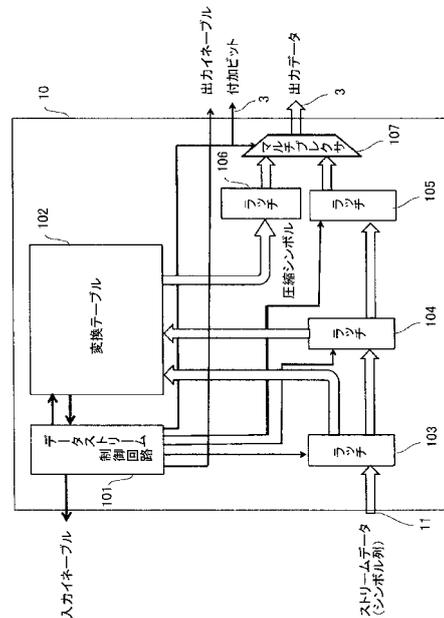
(54) 【発明の名称】 データ圧縮・解凍システム、データ圧縮方法及びデータ解凍方法、並びにデータ圧縮器及びデータ解凍器

(57) 【要約】

【課題】 転送データをリアルタイムに圧縮する場合においてデータの傾向に沿った変換規則を生成・適用する。

【解決手段】 データ圧縮方法は、固定長のデータであるシンボル単位でデータ列を圧縮する。そして、変換前に係る2以上のシンボルと変換後に係る1つのシンボルとの対応付けを示すエントリが登録される変換テーブルを検索し、データ列において連続する2以上のシンボルが、変換前に係る2以上のシンボルとして登録されていないと判断された場合、連続する2以上のシンボルを変換前に係る2以上のシンボルとするエントリを変換テーブルに登録すると共に、連続する2以上のシンボルを変換せずに出力し、データ列において連続する2以上のシンボルが、変換前に係る2以上のシンボルとして変換テーブルのエントリに登録されている場合、当該エントリにおいて対応付けられている変換後に係る1つのシンボルに、連続する2以上のシンボルを変換する。

【選択図】 図2



【特許請求の範囲】**【請求項 1】**

固定長のデータであるシンボル単位でデータ列を圧縮し、圧縮されたデータ列を受信側の装置へ送信する送信側の装置と、前記圧縮されたデータ列を受信し解凍する前記受信側の装置とを備えるデータ圧縮・解凍システムであって、

前記送信側の装置が、

変換前に係る 2 以上のシンボルと変換後に係る 1 つのシンボルとの対応付けを示すエントリが登録される変換テーブルを検索し、前記データ列において連続する 2 以上のシンボルが、変換前に係る 2 以上のシンボルとして登録されていないと判断された場合、前記連続する 2 以上のシンボルを変換前に係る 2 以上のシンボルとするエントリを前記変換テーブルに登録すると共に、前記連続する 2 以上のシンボルを変換せずに出し、

10

前記データ列において連続する 2 以上のシンボルが、変換前に係る 2 以上のシンボルとして前記変換テーブルのエントリに登録されている場合、当該エントリにおいて対応付けられている変換後に係る 1 つのシンボルに、前記連続する 2 以上のシンボルを変換して出し、

前記受信側の装置が、

解凍前に係るシンボルと解凍後に係る 2 以上のシンボルとの対応付けを示すエントリが登録される変換テーブルを検索し、前記データ列に含まれるシンボルが、解凍前に係るシンボルとして登録されていないと判断された場合、前記データ列に含まれるシンボル及び後続の所定数のシンボルを解凍後に係る 2 つ以上のシンボルとするエントリを前記変換テーブルに登録すると共に、前記データ列に含まれるシンボル及び後続する所定数のシンボルを変換せずに出し、

20

前記データ列に含まれるシンボルが、解凍前に係るシンボルとして前記変換テーブルのエントリに登録されている場合、当該エントリにおいて対応付けられている解凍後に係る 2 以上のシンボルに、当該前記データ列に含まれるシンボルを変換して出力する

データ圧縮・解凍システム。

【請求項 2】

コンピュータが、固定長のデータであるシンボル単位でデータ列を処理し、当該データ列を圧縮するデータ圧縮方法であって、

変換前に係る 2 以上のシンボルと変換後に係る 1 つのシンボルとの対応付けを示すエントリが登録される変換テーブルを検索し、前記データ列において連続する 2 以上のシンボルが、変換前に係る 2 以上のシンボルとして登録されていないと判断された場合、前記連続する 2 以上のシンボルを変換前に係る 2 以上のシンボルとするエントリを前記変換テーブルに登録すると共に、前記連続する 2 以上のシンボルを変換せずに出し、

30

前記データ列において連続する 2 以上のシンボルが、変換前に係る 2 以上のシンボルとして前記変換テーブルのエントリに登録されている場合、当該エントリにおいて対応付けられている変換後に係る 1 つのシンボルに、前記連続する 2 以上のシンボルを変換して出力する、

データ圧縮方法。

【請求項 3】

40

前記変換テーブルのエントリごとに、当該エントリが示す変換前に係る 2 以上のシンボルの、前記データ列における出現回数に応じた値を計数し、

計数された前記出現回数に応じた値が小さいエントリを優先して、前記変換テーブルに登録されたエントリを削除する、

請求項 2 に記載のデータ圧縮方法。

【請求項 4】

前記変換テーブルにエントリを登録する際、前記変換テーブルの空き領域のうち、エントリを始めに格納する開始位置と以降のエントリを格納する位置を決定するための規則とに基づいて決定された位置にエントリを登録し、

前記変換テーブルからエントリを削除する際、前記変換テーブルの領域のうち、登録さ

50

れているエントリを削除するか否か始めに判断する開始位置と、以降にエントリを削除するか否か判断する位置を決定するための規則とに基づいて候補の位置を決定し、決定された位置に登録されているエントリに対応付けられている、前記出現回数に応じた値が所定の閾値以下である場合に、当該エントリを削除する

請求項 3 に記載のデータ圧縮方法。

【請求項 5】

請求項 2 から 4 のいずれか一項に記載のデータ圧縮方法を繰り返し実施し、前段のデータ圧縮方法によって出力された前記圧縮データを入力として、後段のデータ圧縮方法を実施する

データ圧縮方法。

10

【請求項 6】

変換せずに出力された少なくとも一部の前記シンボル、又は変換して出力された前記シンボルを暗号化する

請求項 2 から 5 のいずれか一項に記載のデータ圧縮方法。

【請求項 7】

コンピュータが、通信相手の装置から受信するデータ列を、固定長のデータであるシンボル単位で処理し、当該データ列を解凍するデータ解凍方法であって、

解凍前に係るシンボルと解凍後に係る 2 以上のシンボルとの対応付けを示すエントリが登録される変換テーブルを検索し、前記データ列に含まれるシンボルが、解凍前に係るシンボルとして登録されていないと判断された場合、前記データ列に含まれるシンボル及び後続の所定数のシンボルを解凍後に係る 2 つ以上のシンボルとするエントリを前記変換テーブルに登録すると共に、前記データ列に含まれるシンボル及び後続する所定数のシンボルを変換せずに出力し、

20

前記データ列に含まれるシンボルが、解凍前に係るシンボルとして前記変換テーブルのエントリに登録されている場合、当該エントリにおいて対応付けられている解凍後に係る 2 以上のシンボルに、当該前記データ列に含まれるシンボルを変換して出力する

データ解凍方法。

【請求項 8】

前記通信相手の装置から設定の初期化の指示を受信した場合、前記変換テーブルに予め登録しておくエントリ、前記変換テーブルの空き領域のうちエントリを格納する開始位置、又は前記変換テーブルの領域のうち当該領域のエントリを削除するか否か判断する開始位置を初期化する

30

請求項 7 に記載のデータ解凍方法。

【請求項 9】

固定長のデータであるシンボル単位でデータ列を処理し、当該データ列を圧縮するデータ圧縮器であって、

変換前に係る 2 以上のシンボルと変換後に係る 1 つのシンボルとの対応付けを示すエントリが登録される変換テーブルと、

前記変換テーブルに基づいてデータ列に含まれるシンボルの変換を制御する制御回路と

40

、
を含み、

前記制御回路は、

前記データ列において連続する 2 以上のシンボルが、変換前に係る 2 以上のシンボルとして登録されていないと判断された場合、前記連続する 2 以上のシンボルを変換前に係る 2 以上のシンボルとするエントリを前記変換テーブルに登録すると共に、前記連続する 2 以上のシンボルを変換せずに出力し、

前記データ列において連続する 2 以上のシンボルが、変換前に係る 2 以上のシンボルとして前記変換テーブルのエントリに登録されている場合、当該エントリにおいて対応付けられている変換後に係る 1 つのシンボルに、前記連続する 2 以上のシンボルを変換して出力する、

50

データ圧縮器。

【請求項 10】

固定長のデータであるシンボル単位でデータ列を処理し、当該データ列を解凍するデータ解凍器であって、

解凍前に係るシンボルと解凍後に係る 2 以上のシンボルとの対応付けを示すエントリが登録される変換テーブルと、

前記変換テーブルに基づいてデータ列に含まれるシンボルの変換を制御する制御回路と

、

を含み、

前記制御回路は、

10

前記データ列に含まれるシンボルが、解凍前に係るシンボルとして登録されていないと判断された場合、前記データ列に含まれるシンボル及び後続の所定数のシンボルを解凍後に係る 2 つ以上のシンボルとするエントリを前記変換テーブルに登録すると共に、前記データ列に含まれるシンボル及び後続する所定数のシンボルを変換せずに出し、

前記データ列に含まれるシンボルが、解凍前に係るシンボルとして前記変換テーブルのエントリに登録されている場合、当該エントリにおいて対応付けられている解凍後に係る 2 以上のシンボルに、当該前記データ列に含まれるシンボルを変換して出力する

データ解凍器。

【発明の詳細な説明】

【技術分野】

20

【0001】

本発明は、データ圧縮・解凍システム、データ圧縮方法及びデータ解凍方法、並びにデータ圧縮器及びデータ解凍器に関する。

【背景技術】

【0002】

近年における、データストリームがネットワーク上を流れる環境下では、データストリームを形成するストリームデータに対するリアルタイム処理を行うため、ストリームデータを送受信する様々なエンティティ間におけるデータ伝送時間の短縮化が求められている。エンティティは、例えば、ネットワークに接続された様々な通信機器（端末装置、中継装置）である。また、データストリームは、通信機器内に搭載されたプロセッサ、LSI (Large Scale Integrated Circuit)、FPGA (Field Programmable Gate Array) のような様々なストリームデータに対する処理を行う電子回路チップ間を流れる。電子回路チップもエンティティの 1 つであり、エンティティ間の通信は、通信機器間の通信だけでなく、通信機器内部の電子回路チップ間の通信（いわゆる内部通信）を含む。

30

【0003】

近年では、ストリームデータ量が増大する傾向にある。或る量のストリームデータを送信側から受信側へ効率的に伝送する手法として、エンティティ間を結ぶ伝送路の周波数を上げる（伝送帯域を広げる）ことや、エンティティ間を複数の伝送路で結び、ストリームデータを並列に送信することが考えられる。しかしながら、これらの手法は、物理的、周波数的な限界がいずれ来ると考えられている。

40

【0004】

そこで、送信側装置でストリームデータの圧縮を行う技術が提案されている。すなわち、送信データ量の減少に伴うデータ伝送時間の短縮化を以て、データ伝送の効率化を図る。例えば、連続する 2 以上のシンボルが登録されている場合、2 以上のシンボルを 1 つのシンボルに変換する変換部と、変換部で 2 以上のシンボルが 1 つのシンボルに変換された場合は、当該 1 つのシンボルを出力し、そうでない場合は、2 以上のシンボルを出力する出力部とを含むデータ圧縮器が提案されている（例えば、特許文献 1）。このようにすれば、一定時間内の処理遅延で圧縮及び解凍処理を行うことができる。

【0005】

また、入力データをブロック単位で集計処理して符号化対象データ毎の発生度数を計数

50

し、発生度数が0の符号化対象データが存在する場合に1以上の整数を一様に加算し、あるブロックにおける発生度数に基づいて作成した適応ハフマン符号表に基づき、次のブロックに対して可変長符号化を実行するという技術も提案されている（例えば、特許文献2）。

【先行技術文献】

【特許文献】

【0006】

【特許文献1】特開2014-236449号公報

【特許文献2】特開平7-7436号公報

【発明の概要】

【発明が解決しようとする課題】

【0007】

例えば上述の特許文献1に記載された技術では、送信者と受信者との間で予め変換テーブル（ルックアップテーブル）を共有しておかなければならない。また、変換前後のデータシンボル（以下、単に「シンボル」とも呼ぶ）の対応関係として記憶されている情報のうち、変換前のシンボルが、データストリームに多く出現するほど圧縮効率が向上する一方、特にハードウェアによってデータ圧縮器及びデータ解凍器を実現する場合、変換前後のシンボルの対応関係を記憶しておくことができる容量は制限される。したがって、シンボルの出現傾向が異なるデータストリームに対し、共通に適用できる変換規則を用意しておくのは困難であった。

【0008】

本発明は、このような問題に鑑みてなされたものであり、転送データをリアルタイムに圧縮する場合において、データの傾向に沿った変換規則を生成及び適用することを目的とする。

【課題を解決するための手段】

【0009】

本発明の一側面に係るデータ圧縮・解凍システムは、固定長のデータであるシンボル単位でデータ列を処理し、装置間におけるデータ列の転送前後において当該データ列を圧縮及び解凍する。そして、送信側の装置が、変換前に係る2以上のシンボルと変換後に係る1つのシンボルとの対応付けを示すエントリが登録される変換テーブルを検索し、データ列において連続する2以上のシンボルが、変換前に係る2以上のシンボルとして登録されていないと判断された場合、連続する2以上のシンボルを変換前に係る2以上のシンボルとするエントリを変換テーブルに登録すると共に、連続する2以上のシンボルを変換せずに出力し、データ列において連続する2以上のシンボルが、変換前に係る2以上のシンボルとして変換テーブルのエントリに登録されている場合、当該エントリにおいて対応付けられている変換後に係る1つのシンボルに、連続する2以上のシンボルを変換して出力する。一方、受信側の装置が、解凍前に係るシンボルと解凍後に係る2以上のシンボルとの対応付けを示すエントリが登録される変換テーブルを検索し、データ列に含まれるシンボルが、解凍前に係るシンボルとして登録されていないと判断された場合、データ列に含まれるシンボル及び後続の所定数のシンボルを解凍後に係る2つ以上のシンボルとするエントリを変換テーブルに登録すると共に、データ列に含まれるシンボル及び後続する所定数のシンボルを変換せずに出力し、データ列に含まれるシンボルが、解凍前に係るシンボルとして変換テーブルのエントリに登録されている場合、当該エントリにおいて対応付けられている解凍後に係る2以上のシンボルに、当該データ列に含まれるシンボルを変換して出力する。

【0010】

本発明の他の側面に係るデータ圧縮方法は、コンピュータが、固定長のデータであるシンボル単位でデータ列を処理し、当該データ列を圧縮する。そして、変換前に係る2以上のシンボルと変換後に係る1つのシンボルとの対応付けを示すエントリが登録される変換テーブルを検索し、データ列において連続する2以上のシンボルが、変換前に係る2以上

10

20

30

40

50

のシンボルとして登録されていないと判断された場合、連続する2以上のシンボルを変換前に係る2以上のシンボルとするエントリを変換テーブルに登録すると共に、連続する2以上のシンボルを変換せずに出力し、データ列において連続する2以上のシンボルが、変換前に係る2以上のシンボルとして変換テーブルのエントリに登録されている場合、当該エントリにおいて対応付けられている変換後に係る1つのシンボルに、連続する2以上のシンボルを変換して出力する。

【0011】

また、本発明の他の側面に係るデータ解凍方法は、コンピュータが、通信相手の装置から受信するデータ列を、固定長のデータであるシンボル単位で処理し、当該データ列を解凍する。そして、解凍前に係るシンボルと解凍後に係る2以上のシンボルとの対応付けを示すエントリが登録される変換テーブルを検索し、データ列に含まれるシンボルが、解凍前に係るシンボルとして登録されていないと判断された場合、データ列に含まれるシンボル及び後続の所定数のシンボルを解凍後に係る2つ以上のシンボルとするエントリを変換テーブルに登録すると共に、データ列に含まれるシンボル及び後続する所定数のシンボルを変換せずに出力し、データ列に含まれるシンボルが、解凍前に係るシンボルとして変換テーブルのエントリに登録されている場合、当該エントリにおいて対応付けられている解凍後に係る2以上のシンボルに、当該データ列に含まれるシンボルを変換して出力する。

10

【発明の効果】

【0012】

本発明によれば、ストリームデータの傾向に沿った変換規則を生成及び適用することができる。

20

【図面の簡単な説明】

【0013】

【図1】データ圧縮/解凍システムの構成を示す図である。

【図2】データ圧縮器の一例を示す機能ブロック図である。

【図3】データ圧縮器の変換テーブルに登録されるエントリの一例を示す図である。

【図4】実施形態1に係る圧縮処理の一例を示す処理フロー図である。

【図5A】変換テーブルへのエントリの登録及び更新を説明するための図である。

【図5B】変換テーブルへのエントリの登録及び更新を説明するための図である。

【図5C】変換テーブルへのエントリの登録及び更新を説明するための図である。

30

【図5D】変換テーブルへのエントリの登録及び更新を説明するための図である。

【図5E】変換テーブルへのエントリの登録及び更新を説明するための図である。

【図5F】変換テーブルへのエントリの登録及び更新を説明するための図である。

【図5G】変換テーブルへのエントリの登録及び更新を説明するための図である。

【図5H】変換テーブルへのエントリの登録及び更新を説明するための図である。

【図5I】変換テーブルへのエントリの登録及び更新を説明するための図である。

【図6】データ解凍器の一例を示す機能ブロック図である。

【図7】データ解凍器の変換テーブルに登録されるエントリの一例を示す図である。

【図8】実施形態1に係る解凍処理の一例を示す処理フロー図である。

【図9A】変換テーブルへのエントリの登録及び更新を説明するための図である。

40

【図9B】変換テーブルへのエントリの登録及び更新を説明するための図である。

【図9C】変換テーブルへのエントリの登録及び更新を説明するための図である。

【図9D】変換テーブルへのエントリの登録及び更新を説明するための図である。

【図9E】変換テーブルへのエントリの登録及び更新を説明するための図である。

【図9F】変換テーブルへのエントリの登録及び更新を説明するための図である。

【図9G】変換テーブルへのエントリの登録及び更新を説明するための図である。

【図9H】変換テーブルへのエントリの登録及び更新を説明するための図である。

【図9I】変換テーブルへのエントリの登録及び更新を説明するための図である。

【図10】実施形態1に係る圧縮処理の一例を示す処理フロー図である。

【図11A】実施形態2に係る変換テーブルの更新を説明するための図である。

50

- 【図 1 1 B】実施形態 2 に係る変換テーブルの更新を説明するための図である。
- 【図 1 1 C】実施形態 2 に係る変換テーブルの更新を説明するための図である。
- 【図 1 1 D】実施形態 2 に係る変換テーブルの更新を説明するための図である。
- 【図 1 2】実施形態 1 に係る解凍処理の一例を示す処理フロー図である。
- 【図 1 3 A】実施形態 2 に係る変換テーブルの更新を説明するための図である。
- 【図 1 3 B】実施形態 2 に係る変換テーブルの更新を説明するための図である。
- 【図 1 3 C】実施形態 2 に係る変換テーブルの更新を説明するための図である。
- 【図 1 3 D】実施形態 2 に係る変換テーブルの更新を説明するための図である。
- 【図 1 4】実施形態 3 に係る圧縮処理の一例を示す処理フロー図である。
- 【図 1 5 A】実施形態 3 に係る変換テーブルの更新を説明するための図である。 10
- 【図 1 5 B】実施形態 3 に係る変換テーブルの更新を説明するための図である。
- 【図 1 5 C】実施形態 3 に係る変換テーブルの更新を説明するための図である。
- 【図 1 5 D】実施形態 3 に係る変換テーブルの更新を説明するための図である。
- 【図 1 6】実施形態 3 に係る解凍処理の一例を示す処理フロー図である。
- 【図 1 7 A】実施形態 3 に係る変換テーブルの更新を説明するための図である。
- 【図 1 7 B】実施形態 3 に係る変換テーブルの更新を説明するための図である。
- 【図 1 7 C】実施形態 3 に係る変換テーブルの更新を説明するための図である。
- 【図 1 7 D】実施形態 3 に係る変換テーブルの更新を説明するための図である。
- 【図 1 8】実施形態 4 に係るデータ圧縮器の一例を示す機能ブロック図である。
- 【図 1 9】実施形態 4 に係るデータ解凍器の一例を示す機能ブロック図である。 20
- 【図 2 0 A】実施形態 5 に係る変換テーブルへのエントリの登録及び更新を説明するための図である。
- 【図 2 0 B】実施形態 5 に係る変換テーブルへのエントリの登録及び更新を説明するための図である。
- 【図 2 0 C】実施形態 5 に係る変換テーブルへのエントリの登録及び更新を説明するための図である。
- 【図 2 0 D】実施形態 5 に係る変換テーブルへのエントリの登録及び更新を説明するための図である。
- 【図 2 0 E】実施形態 5 に係る変換テーブルへのエントリの登録及び更新を説明するための図である。 30
- 【図 2 0 F】実施形態 5 に係る変換テーブルへのエントリの登録及び更新を説明するための図である。
- 【図 2 0 G】実施形態 5 に係る変換テーブルへのエントリの登録及び更新を説明するための図である。
- 【図 2 0 H】実施形態 5 に係る変換テーブルへのエントリの登録及び更新を説明するための図である。
- 【図 2 0 I】実施形態 5 に係る変換テーブルへのエントリの登録及び更新を説明するための図である。
- 【図 2 1】実施形態 6 に係る、データ圧縮器を多段接続したデータ圧縮装置の一例を示すブロック図である。 40
- 【図 2 2】実施形態 6 に係る、データ解凍器を多段接続したデータ解凍装置の一例を示すブロック図である。
- 【図 2 3】実施形態 7 に係るデータフォーマットの一例を示す図である。
- 【図 2 4】実施形態 7 に係る通信の一例を示すシーケンス図である。
- 【図 2 5】実施形態 7 に係る通信の一例を示すシーケンス図である。
- 【図 2 6】スクランブル回路を含むデータ圧縮/解凍システムの一例を示す図である。
- 【図 2 7 A】実施形態 9 に係る変換テーブルへのエントリの登録及び更新を説明するための図である。
- 【図 2 7 B】実施形態 9 に係る変換テーブルへのエントリの登録及び更新を説明するための図である。 50

【図 2 7 C】実施形態 9 に係る変換テーブルへのエントリの登録及び更新を説明するための図である。

【図 2 7 D】実施形態 9 に係る変換テーブルへのエントリの登録及び更新を説明するための図である。

【図 2 7 E】実施形態 9 に係る変換テーブルへのエントリの登録及び更新を説明するための図である。

【図 2 7 F】実施形態 9 に係る変換テーブルへのエントリの登録及び更新を説明するための図である。

【図 2 8 A】実施形態 9 に係る変換テーブルへのエントリの登録及び更新を説明するための図である。

10

【図 2 8 B】実施形態 9 に係る変換テーブルへのエントリの登録及び更新を説明するための図である。

【図 2 8 C】実施形態 9 に係る変換テーブルへのエントリの登録及び更新を説明するための図である。

【図 2 8 D】実施形態 9 に係る変換テーブルへのエントリの登録及び更新を説明するための図である。

【図 2 8 E】実施形態 9 に係る変換テーブルへのエントリの登録及び更新を説明するための図である。

【図 2 8 F】実施形態 9 に係る変換テーブルへのエントリの登録及び更新を説明するための図である。

20

【発明を実施するための形態】

【0014】

以下、本発明の実施の形態について、図面に基づいて説明する。なお、実施の形態は本発明の一例であり、本発明の構成は以下の例には限られない。

【0015】

〔実施形態 1〕

図 1 は、データ圧縮 / 解凍システムを模式的に示す図である。図 1 において、データ圧縮 / 解凍システムは、送信側装置 1 と、送信側装置 1 が備え、信号線 1 1 で接続されたデータ圧縮器 1 0 と、受信側装置 2 と、受信側装置 2 が備え、信号線 2 1 で接続されたデータ解凍器 2 0 とを含む。また、データ圧縮器 1 0 とデータ解凍器 2 0 とは、伝送路 3 を介して接続されている。

30

【0016】

データ圧縮器 1 0 は、送信側装置 1 から受信側装置 2 へ送信するデータ（例えば、ストリームデータ）に対する圧縮処理を行い、圧縮データを出力する。また、圧縮データは、伝送路 3 を通ってデータ解凍器 2 0 に到達する。データ解凍器 2 0 は、圧縮データを元のデータに戻す解凍処理を行い、元のデータが受信側装置 2 に供給される。圧縮処理を行うことにより伝送路 3 へ送出されるデータ量が減少するため、あるサイズのデータを送信側装置 1 から受信側装置 2 に伝送するために要する時間は、圧縮処理を行わない場合に比べて短縮される。

【0017】

40

データ圧縮 / 解凍システムは、通信機器間の通信に適用されても良く、通信機器内の構成要素間の通信（いわゆる「内部通信」と呼ばれるような、電子回路チップ間通信）に適用されても良い。通信機器間の通信に適用する場合、例えば、データ圧縮器 1 0 は、送信側の通信機器に搭載され、データ解凍器 2 0 は、受信側の通信機器に搭載される。また、内部通信に適用する場合、例えば、データ圧縮器 1 0 及びデータ解凍器 2 0 は、通信機器や各種の情報処理装置（コンピュータ）内に構成要素の一つとして搭載される。

【0018】

なお、通信機器が双方向通信を行う場合、送信側及び受信側の通信機器のそれぞれにデータ圧縮器 1 0 及びデータ解凍器 2 0 が搭載され、上り通信と下り通信とのそれぞれにおいてデータの圧縮 / 解凍が行われる構成としてもよい。

50

【 0 0 1 9 】

< データ圧縮器 >

図 2 は、図 1 に示したデータ圧縮器 1 0 の一例を示すブロック図である。データ圧縮器 1 0 は、送信側装置 1 から入力されるストリームデータに対して可逆圧縮処理（「符号化処理」とも呼ぶ）を行い、少なくとも一部が圧縮されたストリームデータを出力する。なお、ストリームデータの形式は、テキストであってもバイナリであってもよい。また、データ圧縮器 1 0 は、ストリームデータを、固定長の処理単位であるシンボルの列として扱う。1 シンボルのサイズは任意であり、例えば 1 バイト（8 ビット）とすることができる。

【 0 0 2 0 】

データ圧縮処理は、ストリームデータに含まれる所定のシンボル列に対する変換ルールを規定する変換テーブル（「ルックアップテーブル」とも呼ぶ）に従い、連続するデータシンボル列（例えば 2 つのシンボルであり、「シンボルペア」とも呼ぶ）を、当該シンボル列よりサイズの小さい圧縮シンボル（例えば 1 つのシンボル）に変換することにより行われる。また、データ圧縮器 1 0 は、いわゆる動的符号化（「適応型符号化」とも呼ぶ）を行う。すなわち、データ圧縮器 1 0 は、データ圧縮処理を行うと共に、ストリームデータに含まれるシンボルの出現傾向に基づいて、変換ルールを生成及び変更する。

【 0 0 2 1 】

具体的には、データ圧縮器 1 0 は、データストリーム制御回路 1 0 1 と、変換テーブル 1 0 2 と、ラッチ 1 0 3 ~ ラッチ 1 0 6 と、マルチプレクサ 1 0 7 とを含み、これらが信号線で接続されている。データストリーム制御回路 1 0 1 は、他の構成要素にイネーブル信号を出力し、ストリームデータに含まれるシンボルをデータラッチ（単に「ラッチ」とも呼ぶ）に保持させたり、変換テーブル 1 0 2 の検索や更新を行わせたりする。なお、データストリーム制御回路 1 0 1 は、例えば L S I（Large Scale Integration）、A S I C（Application Specific Integrated Circuit）、P L D（Programmable Logic Device、例えば F P G A（Field-Programmable Gate Array））によって構成するようにしてもよい。また、データストリーム制御回路 1 0 1 は、入力イネーブル信号及び出力イネーブル信号をそれぞれ送受信し、他の装置と同期してデータを伝送する。データ圧縮器 1 0 の構成要素は、図示していないクロック信号に基づいて、同期して動作する。

【 0 0 2 2 】

変換テーブル 1 0 2 は、圧縮前後のシンボルの対応付けを規定するエントリを保持すると共に、エントリの読み書きを行う回路を含む記憶装置である。すなわち、変換テーブル 1 0 2 は、シンボルペアが被圧縮シンボル列として登録されているエントリが変換テーブルに存在する場合（すなわち、シンボルペアが変換テーブルにヒットした場合）、変換後の圧縮シンボルを出力する。このとき、変換テーブル 1 0 2 は、ヒットしたか否かを示す信号をデータストリーム制御回路 1 0 1 へ出力し、データストリーム制御回路 1 0 1 は、マルチプレクサ 1 0 7 からの出力を制御したり、出力されるシンボルが変換後のシンボルであるか否かを示すフラグ（「付加ビット」とも呼ぶ）を出力したりする。また、変換テーブル 1 0 2 は、ストリームデータに含まれるシンボルの出現頻度に応じて、エントリを追加及び削除する。

【 0 0 2 3 】

変換テーブル 1 0 2 には、例えば連想メモリ（C A M（Content Addressable Memory））を用いることができる。C A M は、入力されたデータワード（データ語）に対応するアドレスを出力する高速検索用のコンピュータメモリである。C A M が適用される場合には、シンボルペアがデータワードとして C A M に入力され、C A M はエントリがヒットしたときに、データワードに対応するアドレスとして圧縮シンボルを出力するとともに、真（True）信号を出力する。True 信号は、ヒットしたか否かを示す信号としてデータストリーム制御部に出力される。一方、エントリがヒットしなかった場合には、C A M からのアドレス（圧縮シンボル）の出力は行われず、偽（False）信号が出力される。

【 0 0 2 4 】

10

20

30

40

50

なお、変換テーブル102は、CAMとRAM(Random Access Memory)との組合せであってもよい。この場合、CAMが出力するRAMのアドレスに圧縮シンボルが記憶され、当該アドレスの圧縮シンボルがラッチ106に出力される。RAMはDRAMでもSRAMでも良いが、CAMとの連動に鑑み、高速動作が可能なものを選択するのが好ましい。

【0025】

ラッチ103～ラッチ106は、例えばDフリップフロップで構成される回路である。ラッチ103～ラッチ105は、ストリームデータを入力された順に固定長のシンボル単位で保持し、データパイプラインを形成する。すなわち、ラッチ103～ラッチ105は、ストリームデータからシンボルペアを順に抽出するための構成である。そして、ラッチ104に保持されるシンボルとラッチ103に保持されるシンボルとの組み合わせをシンボルペアとして、シンボル変換テーブル102のエントリにヒットするか否か判断される。ラッチ106は、シンボルペアが変換テーブル102にヒットした場合に出力される圧縮シンボルを保持する。

10

【0026】

マルチプレクサ107は、ラッチ105及びラッチ106と接続され、元のシンボルペア又は圧縮シンボルを出力するセレクタである。すなわち、マルチプレクサ107は、シンボルペアがルックアップテーブルのエントリにヒットしたか否かに応じて、元のシンボルペア又は圧縮シンボルを出力する。具体的には、データストリーム制御回路101から変換テーブルにヒットした旨の信号(True)が入力された場合、圧縮シンボルを出力する。一方、変換テーブルにヒットした旨の信号が入力されないとき(False)は、ラッチ105から入力される元のシンボルを順に2つ出力する。

20

【0027】

<変換テーブル>

図3は、データ圧縮器10が備える変換テーブル102の一例を示す。変換テーブル102は、変換の対象となる2つのシンボル「被圧縮シンボル列」(すなわち、変換前のシンボル列)と、圧縮処理によって変換される1つのシンボル「圧縮シンボル」(すなわち、変換後のシンボル)と、当該圧縮シンボルへのアクセス回数を表す「参照頻度」とを対応づけて記憶するメモリである。なお、入力シンボル列に含まれる2つのシンボルを、便宜上「シンボル0」及び「シンボル1」と呼ぶ。また、変換テーブル102に登録されるエントリ(「レコード」とも呼ぶ)の数は、変換テーブル102を構成するハードウェアの容量に応じた有限個である。本実施形態において、エントリは動的に追加及び削除されるが、処理の開始時において予め何らかのエントリに登録しておくようにしてもよい。本実施形態では、図3に示す1つの英字が1つのデータシンボルを表すものとする。

30

【0028】

<圧縮処理>

図4は、本実施形態に係る圧縮処理の工程を模式的に示す処理フロー図である。まず、データ圧縮器10のデータストリーム制御回路101は、変換テーブル102のエントリを初期化する(図4:S1)。本ステップでは、例えば変換テーブル102のエントリをすべて削除してもよいし、所定のエントリを予め登録しておくようにしてもよい。また、本ステップの処理は、例えば送信側装置1における1つのファイルの転送を開始する度に行うようにしてもよい。

40

【0029】

また、データ圧縮器10のラッチ103及びラッチ104は、ストリームデータからシンボルペアを抽出する(S2)。そして、データ圧縮器10の変換テーブル102は、シンボルペアが非圧縮シンボル列として登録されているか、エントリを検索する(S3)。

【0030】

そして、変換テーブル102に、シンボルペアを被圧縮シンボル列として保持するエントリが登録されている場合(S4:YES)、変換テーブル102は当該エントリに登録されている圧縮シンボルをラッチ106に出力し、データストリーム制御回路101は、

50

マルチプレクサ 107 に、ラッチ 106 が保持する圧縮シンボルを出力させる (S5)。

【0031】

一方、変換テーブル 102 に、シンボルペアを被圧縮シンボル列として保持するエントリが登録されていない場合 (S4:NO)、データストリーム制御回路 101 は、マルチプレクサ 107 に、ラッチ 105 へ伝送されてくる 2 つのシンボル (すなわち、S2 で抽出されたシンボルペア) を順に出力させる (S6)。また、変換テーブル 102 は、当該シンボルペアを被圧縮シンボル列とするエントリを登録する (S7)。

【0032】

そして、S5 又は S7 の後、ストリームデータに後続のシンボルペアが存在する場合 (S8:YES)、S2 に戻って処理を繰り返す。一方、ストリームデータに後続のシンボルペアが存在しない場合 (S8:NO)、圧縮処理を終了する。なお、ストリームデータの最後がシンボルペアを構成しない 1 つのシンボルである場合は、当該シンボルをそのまま出力するものとする。なお、上述の処理フローは一例であり、一部の処理を入れ替えたり並列に行ったりしてもよい。例えば、S6 及び S7 の順序を逆にしてもよい。

10

【0033】

<変換テーブルの生成及び更新>

次に、図 5A ~ 図 5I を用いて、変換テーブル 102 へのエントリを追加する具体的な例を説明する。図 5A ~ 図 5I は、変換テーブルの生成及び更新を説明するための図である。この例では、「A A B B C C D D C C B B C C A A」というストリームデータ (「入力データ」と呼ぶ) を、送信側装置 1 から受信側装置 2 へ送るものとする。また、データ圧縮器 10 は、入力データを 2 シンボルずつ処理する。すなわち、「A A」、「B B」、「C C」・・・という単位で処理をする。

20

【0034】

まず、図 5A に示すように、処理開始前の初期的な変換テーブル 102 には、エントリが登録されていないものとする。また、変換テーブル 102 の最大エントリ数 (容量) は 4 とする。そして、図 5B に示すように、入力データのうち先頭のシンボルペア「A A」が入力されると、変換テーブル 102 には被圧縮シンボル列として「A A」が登録されたエントリが存在しない (すなわち、変換テーブルのエントリにヒットしない) ため、データストリーム制御回路 101 は、シンボルペア「A A」を変換せず出力させる。併せて、変換テーブル 102 には、シンボルペア「A A」を被圧縮シンボル列とするエントリを追加する。なお、当該エントリには、圧縮シンボルとして、他のエントリの圧縮シンボルと重複しない値が割り当てられる。圧縮シンボルは、例えば、エントリに付される添数 (インデックス) であってもよい。図 5B の例では、圧縮シンボルとして「0」が登録されている。また、当該エントリの参照頻度として、当該シンボルペアの出現回数を示す 1 が登録される。

30

【0035】

次に、図 5C に示すように、入力データのうち次のシンボルペア「B B」が入力される。このときも、「B B」が変換テーブル 102 のエントリにヒットしないため、「B B」がそのまま出力されると共に、変換テーブル 102 には「B B」を被圧縮シンボル列とするエントリが追加される。また、追加されるエントリには、圧縮シンボルとして「1」が登録され、参照頻度に 1 が登録される。

40

【0036】

同様に、図 5D に示すように、次のシンボルペア「C C」が入力されると、「C C」がそのまま出力されると共に、変換テーブル 102 には新たなエントリが追加される。当該エントリには、被圧縮シンボル列として「C C」が登録され、圧縮シンボルとして「2」が登録され、参照頻度には 1 が登録されている。また、図 5E に示すように、次のシンボルペア「D D」が入力されると、「D D」がそのまま出力されると共に、変換テーブル 102 に新たなエントリが登録される。当該エントリには、被圧縮シンボル列として「D D」が登録され、圧縮シンボルとして「3」が登録され、当該エントリの参照頻度に 1 が登録されている。

50

【 0 0 3 7 】

次に、図 5 F に示すように、次のシンボルペア「 C C 」が入力されると、「 C C 」は変換テーブル 1 0 2 のエントリにヒットし、圧縮シンボル「 2 」が得られる。この場合、データストリーム制御回路 1 0 1 はマルチプレクサ 1 0 7 に圧縮シンボル「 2 」を出力させる。このとき、入力データに含まれる固定長のシンボル 2 つ（シンボルペア）が 1 つのシンボル（圧縮シンボル）に変換され、データが圧縮される。また、当該エントリの参照頻度には 1 が加算される。また、データストリーム制御回路 1 0 1 は、マルチプレクサ 1 0 7 からの当該出力が圧縮後のシンボルであることを示す付加ビットを伝送路 3 へ出力する。

【 0 0 3 8 】

同様に、図 5 G に示すように、次のシンボルペア「 B B 」が入力されると、「 B B 」が変換テーブル 1 0 2 のエントリにヒットし、圧縮シンボル「 1 」が得られる。よって、シンボルペア「 B B 」は圧縮シンボル「 1 」に変換されて出力されると共に、付加ビットが出力される。また、当該エントリの参照頻度には 1 が加算される。また、図 5 H に示すように、次のシンボルペア「 C C 」が入力された場合も、「 C C 」が変換テーブル 1 0 2 のエントリにヒットし、圧縮シンボル「 2 」が得られる。よって、シンボルペア「 C C 」は圧縮シンボル「 2 」に変換されて出力されると共に、当該エントリの参照頻度に 1 が加算される。そして、図 5 I に示すように、次のシンボルペア「 A A 」が入力された場合も、「 A A 」が変換テーブル 1 0 2 のエントリにヒットし、圧縮シンボル「 0 」が得られる。よって、シンボルペア「 A A 」は圧縮シンボル「 0 」に変換されて出力されると共に、当該エントリの参照頻度に 1 が加算される。

【 0 0 3 9 】

以上のようなデータ圧縮器 1 0 によれば、シンボルペアを 1 つの圧縮シンボルに置き換えた分だけ、伝送路 3 を流れるデータ量を削減することができる。また、処理単位を固定長のシンボルとすること等により、処理に要する時間を一定以下に抑えることができる。したがって、データ圧縮器 1 0 は例えば図 2 に示したようなハードウェアで実現できる。また、入力データに含まれるシンボルの出現傾向に基づいて変換テーブルにエントリを追加するため、事前に変換テーブルを用意する必要はない。このように、ストリームデータをリアルタイムに圧縮する場合において、ストリームデータの傾向に沿った変換規則を生成及び適用できるようになる。

【 0 0 4 0 】

< データ解凍器 >

図 6 は、図 1 に示したデータ解凍器 2 0 の一例を示すブロック図である。データ解凍器 2 0 は、データストリーム制御回路 2 0 1 と、変換テーブル 2 0 2 と、ラッチ 2 0 3 ~ ラッチ 2 0 5 とを含み、これらが信号線で接続されている。データストリーム制御回路 2 0 1 は、他の構成要素にイネーブル信号を出力し、ストリームデータに含まれるシンボルを順次ラッチに保持させたり、変換テーブル 2 0 2 の検索や更新を行わせたりする制御を行う。そして、データストリーム制御回路 2 0 1 は、伝送路 3 を介して入力された 1 つのシンボルが圧縮後のシンボルであるか否かを示す付加ビットに基づき、変換テーブル 2 0 2 から復号後のシンボル列を出力させたり、伝送路 3 を介して入力されたシンボルをそのまま出力させたりする。なお、データストリーム制御回路 2 0 1 は、例えば L S I、A S I C、P L D（例えば F P G A）によって構成するようにしてもよい。また、また、データストリーム制御回路 2 0 1 は、他の装置との間で入力イネーブル信号及び出力イネーブル信号をそれぞれ送受信し、他の装置と同期してデータを伝送する。データ圧縮器 2 0 の構成要素も、図示していないクロック信号に基づいて、同期して動作する。また、データ解凍器 2 0 は、データ圧縮器 1 0 側と同様の規則に基づいて、データ圧縮器 1 0 側と対応するエントリを変換テーブル 2 0 2 に追加及び更新する。

【 0 0 4 1 】

変換テーブル 2 0 2 は、復号前後のシンボルの対応付けを規定するエントリを保持すると共に、エントリの読み書きを行う回路を含む記憶装置である。すなわち、変換テーブル

10

20

30

40

50

202は、圧縮シンボルに対応付けて登録されている復号後のシンボル列を出力する。また、変換テーブル202も、ストリームデータに含まれるシンボルの出現頻度に応じて、エントリを追加及び削除する。

【0042】

なお、変換テーブル202は、例えばRAMによって構成することができる。例えば、圧縮シンボルをRAMのアドレスとし、RAMのデータ部に復号後のシンボル列（「復号シンボル列」とも呼ぶ）を記憶させる。また、参照頻度は、RAM又はレジスタによって構成することができる。

【0043】

ラッチ203～ラッチ206は、例えばDフリップフロップで構成される回路である。伝送路3を介してラッチ203に伝送されたシンボルが圧縮シンボルでない場合は、当該シンボルをラッチ204、ラッチ205へと順次伝送する。一方、ラッチ203に伝送されたシンボルが圧縮シンボルである場合は、当該シンボルはラッチ204以降へ伝送せず、シンボル変換テーブル202から復号シンボル列をラッチ204及びラッチ205へ出力し、復号シンボル列が順次受信側装置2へ出力される。

【0044】

また、伝送路3を介してラッチ203に伝送されたシンボルが圧縮シンボルでない（すなわち、付加ビットが0である）場合、データストリーム制御回路201は、シンボル変換テーブル202に、連続して入力される2つのシンボル（シンボルペア）を復号シンボル列とするエントリを追加する。このとき、圧縮シンボルは、データ圧縮器10側と同様の規則に基づいて割り当てられる。本実施形態では、データ圧縮器10がシンボルペアを初回は変換せずに出し、2回目の出現以降で変換して出力するため、データ解凍器20においても初回の入力時にデータ圧縮器10と対応するエントリを変換テーブル202に登録することができるようになっている。

【0045】

<変換テーブル>

図7は、データ解凍器20が備える変換テーブル202の一例を示す。変換テーブル202は、復号後の2つのシンボル「復号シンボル列」（すなわち、解凍処理における変換後のシンボル列）と、復号の対象となる1つのシンボル「圧縮シンボル」（すなわち、解凍処理における変換前のシンボル）と、当該圧縮シンボルへのアクセス回数を表す「参照頻度」とを対応づけて記憶するメモリである。なお、復号シンボル列に含まれる2つのシンボルを、便宜上「シンボル0」及び「シンボル1」と呼ぶ。また、変換テーブル202に登録されるエントリ（「レコード」とも呼ぶ）の数は、データ圧縮器10の変換テーブル102と同様とする。データ解凍器20の変換テーブル202も、処理の開始時において予めデータ圧縮器10の変換テーブル102と同一のエントリを登録しておくようにしてもよい。

【0046】

<解凍処理>

図8は、本実施形態に係る解凍処理の工程を模式的に示す処理フロー図である。まず、データ解凍器20のデータストリーム制御回路201は、変換テーブル202のエントリを初期化する（図8：S11）。本ステップでは、例えば変換テーブル202のエントリをすべて削除してもよいし、所定のエントリを予め登録しておくようにしてもよいが、データ圧縮器10の変換テーブル102のエントリと対応させておく。本ステップの処理も、例えば送信側装置1における1つのファイルの転送を開始する度に行うようにしてもよい。このようなタイミングは、例えば所定のプロトコルにおけるヘッダ情報で通知する。

【0047】

また、データ解凍器20のラッチ203は、ストリームデータからシンボル及び付加ビットを抽出する（S12）。そして、抽出されたシンボルが圧縮シンボルである場合（S13：YES、すなわち付加ビットが1である場合）、データストリーム制御回路201は、変換テーブル202から復号シンボル列を出力させる（S14）。

10

20

30

40

50

【 0 0 4 8 】

一方、抽出されたシンボルが圧縮シンボルでない場合（S 1 3 : N O , すなわち付加ビットが 0 である場合）、データストリーム制御回路 2 0 1 は、入力されたシンボルを変換せずに 2 つ出力させる（S 1 5 ）。また、変換テーブル 2 0 2 に、S 1 5 で出力させた 2 つのシンボルを復号シンボル列とするエントリが登録される（S 1 6 ）。なお、S 1 5 及び S 1 6 の処理を実行する順序は、並列であってもよいし、逆であってもよい。

【 0 0 4 9 】

そして、S 1 4 又は S 1 6 の後、ストリームデータに後続のシンボルが存在する場合（S 1 7 : Y E S ）、S 1 2 に戻って処理を繰り返す。一方、ストリームデータに後続のシンボルペアが存在しない場合（S 1 7 : N O ）、解凍処理を終了する。

10

【 0 0 5 0 】

< 変換テーブルの生成及び更新 >

次に、図 9 A ~ 図 9 I を用いて、変換テーブル 2 0 2 へのエントリを追加する具体的な例を説明する。図 9 A ~ 図 9 I は、データ解凍器 2 0 における変換テーブルの生成及び更新を説明するための図である。この例では、「A A B B C C D D 2 1 2 0」というストリームデータ（「入力データ」と呼ぶ）が、伝送路 3 を介してデータ解凍器 2 0 へ送られてきたものとする。

【 0 0 5 1 】

まず、図 9 A に示すように、処理開始前の初期的な変換テーブル 2 0 2 には、エントリが登録されていないものとする。また、変換テーブル 2 0 2 の最大エントリ数（容量）も 4 とする。また、入力データのうち先頭のシンボル「A」が入力されると共に、付加ビットとして 0 が入力されるものとする。すなわち、シンボル「A」は圧縮シンボルではないため、図 9 B に示すように、次のシンボルと併せたシンボルペアを読み出し、変換テーブル 2 0 2 にエントリを登録する。エントリは、シンボルペア「A A」を復号シンボル列とし、圧縮シンボルには「0」が登録される。また、当該エントリの参照頻度として、当該シンボルの出現回数を示す 1 が登録される。

20

【 0 0 5 2 】

次に、図 9 C に示すように、入力データのうち次のシンボル「B」も圧縮シンボルではないと判断され、シンボルペア「B B」が変換テーブル 2 0 2 に追加される。また、追加されるエントリには、圧縮シンボルとして「1」が登録され、参照頻度に 1 が登録される。

30

【 0 0 5 3 】

同様に、図 9 D に示すように、次のシンボル「C」も圧縮シンボルではないと判断され、シンボルペア「C C」が変換テーブル 2 0 2 に追加される。当該エントリには、復号シンボル列として「C C」が登録され、圧縮シンボルとして「2」が登録され、参照頻度には 1 が登録されている。また、図 9 E に示すように、次のシンボル「D」も圧縮シンボルではないため、シンボルペア「D D」が変換テーブル 2 0 2 に登録される。当該エントリには、復号シンボル列として「D D」が登録され、圧縮シンボルとして「3」が登録され、当該エントリの参照頻度に 1 が登録されている。

【 0 0 5 4 】

その後、次のシンボル「2」が入力されると、付加ビットとして 1 が入力される。このとき、図 9 F に示すように、変換テーブル 2 0 2 から圧縮シンボル「2」に対応付けられている復号シンボル列「C C」が得られる。よって、データストリーム制御回路 2 0 1 は、ラッチ 2 0 4 及びラッチ 2 0 5 に、復号シンボル「C C」を出力させる。このようにして、圧縮したデータを元のデータに戻すことができる。また、当該エントリの参照頻度には 1 が加算される。

40

【 0 0 5 5 】

同様に、図 9 G に示すように、次のシンボル「1」は、変換テーブル 2 0 2 のエントリに基づき、復号シンボル列「B B」に変換される。また、当該エントリの参照頻度には 1 が加算される。また、図 9 H に示すように、次のシンボル「2」は、変換テーブル 2 0 2

50

のエントリに基づき、復号シンボル列「CC」に変換される。また、当該エントリの参照頻度には1が加算される。そして、図9Iに示すように、次のシンボル「0」は、変換テーブル202のエントリに基づき、復号シンボル列「AA」に変換される。また、当該エントリの参照頻度には1が加算される。

【0056】

以上のようなデータ解凍器20によれば、データ圧縮器10が圧縮したストリームデータを復号することができる。また、処理単位を固定長のシンボルとすること等により、処理に要する時間を一定以下に抑えることができる。したがって、データ解凍器20は、例えば図6に示したようなハードウェアで実現できる。また、ストリームデータに含まれるシンボルの出現傾向に基づいてデータ圧縮器10側と同様のルールで変換テーブルにエントリを追加するため、事前に変換テーブルを用意する必要はない。このように、ストリームデータをリアルタイムに解凍する場合において、ストリームデータの傾向に沿った変換規則を生成及び適用できるようになる。

10

【0057】

〔実施形態2〕

次に、データ圧縮器10の変換テーブル102、及びデータ解凍器20の変換テーブル202からエントリを削除する例を説明する。なお、実施形態2においては、実施形態1と共通する構成については同一の符号を付して説明を省略し、主として相違点について説明する。本実施形態では、変換テーブル102又は変換テーブル202にエントリを追加する際、空きがなければエントリの削除を行う。削除の対象は、ストリームデータにおけるシンボルペアの出現頻度に基づき、出現頻度（すなわち、変換テーブル202の「参照頻度」）の低いシンボルペアを被圧縮シンボル列として保持するエントリとする。ここで、参照頻度の値が所定の閾値（「削除閾値」とも呼ぶ）以下のエントリを削除するものとする。本実施形態では、削除閾値が1と定められているものとする。また、変換テーブル102からエントリを削除する処理は、例えば図2に示したデータストリーム制御回路101が制御する。

20

【0058】

< 圧縮処理 >

図10は、本実施形態に係る圧縮処理の一例を示す処理フロー図である。図10の処理フローでは、変換テーブルにエントリを追加する処理（S7）の前に、変換テーブルに空きがなければエントリを削除する処理（S21）を行う。本ステップでは、削除するエントリを変換テーブル102からシーケンシャルに全検索して決定する。なお、S6の処理と、S21及びS7の処理とを実行する順序は、並列であってもよいし、逆であってもよい。

30

【0059】

図11A～図11Dは、変換テーブル102からのエントリの削除を説明するための図である。例えば、図11Aに示す状態において、シンボルペア「EE」が入力されたものとする。このような場合、シンボルペア「EE」は変換テーブル102のエントリにヒットしないため、図10のS6においてそのまま出力される。そして、変換テーブル102には新たなエントリを追加する空き容量がないため、図10のS21においてエントリの削除が行われる。本ステップでは、変換テーブル102に登録されているエントリの参照頻度の値をシーケンシャルに全件読み出し、参照頻度の値が削除閾値「1」以下のエントリを削除する。ここで、参照頻度の値が削除閾値以下のエントリが存在しない場合、変換テーブル102に登録されているすべてのエントリからシーケンシャルに参照頻度の値を1ずつ減算し、参照頻度の値が削除閾値以下になったエントリをすべて削除する。図11Aの例では、参照頻度が1以下のエントリが存在しないため、図11Bに示すように、すべてのエントリから参照頻度を1ずつ減算する。すると、被圧縮シンボル列が「AA」及び「BB」のエントリの参照頻度が削除閾値「1」になるため、図11Cに示すように、S21においてデータストリーム制御回路101は当該エントリを変換テーブル102から削除させる。その後、図11Dに示すように、変換テーブル102の空き領域をシーケ

40

50

ンシャルに検索し、入力されたシンボルペア「EE」を被圧縮シンボル列とするエントリが変換テーブル102に追加される(図10:S7)。

【0060】

このように、ストリームデータにおけるシンボルペアの出現傾向に応じて変換テーブル102の内容を変更すれば、利用可能なメモリの容量の範囲で圧縮処理を行うことができる。また、データ解凍器20側でも同様のルールに基づいて変換テーブル202のエントリを削除及び追加すれば、データ解凍器20は圧縮されたストリームデータを適切に解凍できる。

【0061】

<解凍処理>

図12は、本実施形態に係る解凍処理の一例を示す処理フロー図である。図12の処理フローでは、変換テーブルにエントリを追加する処理(S16)の前に、変換テーブルに空きがなければエントリを削除する処理(S31)を行う。本ステップでは、削除するエントリを変換テーブル202からシーケンシャルに全検索して決定する。

【0062】

図13A~図13Dは、変換テーブル202からのエントリの削除を説明するための図である。例えば、図13Aに示すエントリの状態において、シンボルペア「EE」が入力されたものとする。このような場合、シンボルペア「EE」は変換テーブル102のエントリにヒットしないため、図12のS15においてそのまま出力される。そして、変換テーブル102には新たなエントリを追加する空き容量がないため、図12のS31においてエントリの削除が行われる。本ステップでは、変換テーブル102に登録されているエントリの参照頻度の値をシーケンシャルに全件読み出し、参照頻度の値が削除閾値「1」以下のエントリを削除する。ここで、図13Aのように参照頻度の値が削除閾値以下のエントリが存在しない場合、変換テーブル102に登録されているすべてのエントリからシーケンシャルに参照頻度の値を1ずつ減算し、参照頻度の値が削除閾値以下になったエントリをすべて削除する。図13Aの例では、参照頻度が1以下のエントリが存在しないため、図13Bに示すように、エントリから順に参照頻度を1ずつ減算する。すると、被圧縮シンボル列が「AA」及び「BB」のエントリの参照頻度が削除閾値「1」になるため、図13Cに示すように、データストリーム制御回路201は当該エントリを変換テーブル202から削除させる。その後、図13Dに示すように、変換テーブル202の空き領域をシーケンシャルに検索し、入力されたシンボルペア「EE」を被圧縮シンボル列とするエントリが変換テーブル202に追加される(図12:S16)。

【0063】

このように、データ解凍器20は、変換テーブル202に空きがない場合、データ圧縮器10と同様の規則に基づいて変換テーブル202からエントリを削除する。すなわち、データ圧縮器10の変換テーブル102のエントリと、データ解凍器20の変換テーブル202のエントリとが同期したタイミングで削除及び追加される。したがって、シンボルの変換規則が変更されてもデータ解凍器20は圧縮後のストリームデータを適切に解凍できるようになっている。

【0064】

〔実施形態3〕

次に、データ圧縮器10の変換テーブル102、及びデータ解凍器20の変換テーブル202からエントリを削除する他の例を説明する。なお、実施形態3においては、実施形態2と共通する構成については同一の符号を付して説明を省略し、主として相違点について説明する。本実施形態でも、変換テーブル102又は変換テーブル202にエントリを追加する際、空きがなければエントリの削除を行う。すなわち、削除の対象は、ストリームデータにおけるシンボルペアの出現頻度に基づき、出現頻度の低いシンボルペアを被圧縮シンボル列として保持するエントリとする。ただし、本実施形態では、直近に入力された所定数のシンボルペアを保持するスタック構造のメモリ(単に「スタック」とも呼ぶ)を用いて、変換テーブルに新たに登録されたシンボルペアを優先的に削除する。

10

20

30

40

50

【 0 0 6 5 】

< 圧縮処理 >

図 1 4 は、本実施形態に係る圧縮処理の一例を示す処理フロー図である。図 1 4 の処理フローでも、実施形態 2 と同様に、変換テーブル 1 0 2 にエントリを追加する処理 (S 7) の前に、変換テーブルに空きがなければエントリを削除する処理 (S 2 1) を行う。本ステップでは、変換テーブル 1 0 2 のエントリを削除すると共にスタックのエントリも削除する。ただし、本ステップでは、スタックに記憶されている直近に登録されたエントリを削除対象とする。また、図 1 4 の処理フローでは、変換テーブルにエントリを追加する処理 (S 7) においてスタックにもエントリを追加する。そして、スタックの容量が溢れた場合も、変換テーブルからエントリを削除すると共にスタックを空にする (S 4 1) 。

10

【 0 0 6 6 】

図 1 5 A ~ 図 1 5 D は、変換テーブル 1 0 2 からのエントリの削除を説明するための図である。本実施形態では、ストリームデータ「 A A B B C C D D C C B B C C A A 」が入力され、変換テーブル 1 0 2 に 1 5 A に示すエントリが保持されている状態において、シンボルペア「 E E 」が入力されたものとする。なお、スタックには 2 つのシンボルペアが保持され、図 1 5 A の例では、シンボルペア「 E E 」の前に 2 つのシンボルペア「 C C A A 」が入力されたものとする。なお、スタックの容量は、変換テーブルのエントリ数以下の任意の容量であれば好ましいが、本実施例では変換テーブルのエントリ 2 つ分とする。この場合、図 1 4 の S 2 1 において変換テーブル 1 0 2 に空きがないためエントリを削除する。すなわち、図 1 5 B に示すように、スタックのエントリに記憶されていたシンボルペアについて、変換テーブル 1 0 2 のエントリから参照頻度を 1 減算すると共にスタックからエントリを削除する。そして、図 1 5 C に示すように、変換テーブル 1 0 2 において参照頻度を減算することにより削除閾値以下となったエントリを削除する。また、図 1 5 D に示すように、入力されたシンボルペア「 E E 」を被圧縮シンボル列とするエントリが変換テーブル 1 0 2 に追加される (図 1 4 : S 7) 。なお、本ステップでは、スタックにもシンボルペア「 E E 」に係るエントリが追加される。なお、 S 6 の処理と、 S 2 1 、 S 7 及び S 4 1 の処理とを実行する順序は、並列であってもよいし、逆であってもよい。

20

【 0 0 6 7 】

このように、スタックに記憶されている直近に登録されたエントリを削除対象とし、ストリームデータにおけるシンボルペアの出現傾向に応じて変換ルールを変更することができる。また、データ解凍器 2 0 側でも同様のルールに基づいて変換テーブル 2 0 2 のエントリを削除及び追加すれば、データ解凍器 2 0 は圧縮されたストリームデータを適切に解凍できる。

30

【 0 0 6 8 】

< 解凍処理 >

図 1 6 は、本実施形態に係る解凍処理の一例を示す処理フロー図である。図 1 6 の処理フローでも、実施形態 2 と同様に、変換テーブル 2 0 2 にエントリを追加する処理 (S 1 6) の前に、変換テーブルに空きがなければエントリを削除する処理 (S 3 1) を行う。本ステップでは、変換テーブル 1 0 2 のエントリを削除すると共にスタックのエントリも削除する。ただし、本ステップでは、スタックに記憶されている直近に登録されたエントリを削除対象とする。また、図 1 6 の処理フローでは、変換テーブルにエントリを追加する処理 (S 1 6) においてスタックにもエントリを追加する。そして、スタックの容量が溢れた場合も、変換テーブルからエントリを削除すると共にスタックを空にする (S 5 1) 。

40

【 0 0 6 9 】

図 1 7 A ~ 図 1 7 D は、変換テーブル 2 0 2 からのエントリの削除を説明するための図である。本実施形態では、ストリームデータ「 A A B B C C D D 2 1 0 」が入力され、変換テーブル 1 0 2 に図 1 7 A に示すエントリが保持されている状態において、シンボルペア「 E E 」が入力されたものとする。なお、シンボルペア「 E E 」の前に 2 つの圧縮シンボル「 1 」及び「 0 」が入力され、スタックには復号後のシンボルペアが 2 つ保持されて

50

いる。なお、データ解凍器 20 のスタックの容量は、データ圧縮器 10 のスタックと同一であるものとし、変換テーブルのエントリ 2 つ分とする。この場合、図 16 の S 31 において変換テーブル 202 に空きがないためエントリを削除する。すなわち、図 17 B に示すように、スタックのエントリに記憶されていたシンボルペアについて、変換テーブル 102 のエントリから参照頻度を 1 減算すると共にスタックからエントリを削除する。そして、図 17 C に示すように、変換テーブル 202 において参照頻度を減算することにより削除閾値以下となったエントリを削除する。その後、図 17 D に示すように、入力されたシンボルペア「EE」を被圧縮シンボル列とするエントリが変換テーブル 202 に追加される(図 16 : S 16)。なお、本ステップでは、スタックにもシンボルペア「EE」に係るエントリが追加される。

10

【0070】

このように、スタックに記憶されている直近に登録されたエントリを削除対象とし、ストリームデータにおけるシンボルペアの出現傾向に応じて変換ルールを変更することができる。また、データ圧縮器 10 の変換テーブル 102 のエントリと、データ解凍器 20 の変換テーブル 202 のエントリとが同期したタイミングで削除及び追加される。したがって、シンボルの変換規則が変更されてもデータ解凍器 20 は圧縮後のストリームデータを適切に解凍できるようになっている。

【0071】

〔実施形態 4〕

本実施形態でも、データ圧縮器 10 の変換テーブル 102 及びデータ解凍器 20 の変換テーブル 202 からエントリを削除する。また、本実施形態においても、上述の実施形態と共通する構成については同一の符号を付して説明を省略し、主として相違点について説明する。本実施形態でも、エントリを追加する際、空きがなければ変換テーブルからエントリの削除を行う。具体的には、ストリームデータにおける出現頻度が低いシンボルペアを被圧縮シンボル列として保持するエントリを削除する。ただし、本実施形態では、シーケンシャルに全件のエントリを検索するのではなく、各エントリを並列に処理する。

20

【0072】

<データ圧縮器>

図 18 は、実施形態 4 に係るデータ圧縮器 10 の一例を示すブロック図である。図 18 のデータ圧縮器 10 は、例えば変換テーブルに登録可能なエントリ数のエントリ削除回路 108 を有する。エントリ削除回路 108 は、それぞれがデータストリーム制御回路 101 から指示を受け、変換テーブルの参照頻度の値を減算すると共に、削除閾値以下となったエントリを削除する処理を並列に実行する。複数のエントリ削除回路を設けることで、回路規模は大きくなる一方、エントリの削除に要する時間は短縮することができる。このようなエントリ削除回路は、LSI や FPGA 等によって構成することができる。

30

【0073】

<圧縮処理>

本実施形態においてエントリの削除を行うタイミングは、例えば上述した実施形態 2 と同様である。すなわち、データ圧縮器 10 は、図 10 に示した処理フローに従って動作する。ただし、図 11 B に示す参照頻度の減算処理は、並列に実行される。

40

【0074】

<データ解凍器>

図 19 は、実施形態 4 に係るデータ解凍器 20 の一例を示すブロック図である。図 19 のデータ解凍器 20 も、複数(例えば変換テーブルに登録可能なエントリ数)のエントリ削除回路 206 を有する。複数のエントリ削除回路 206 は、それぞれがデータストリーム制御回路 201 から指示を受け、変換テーブルの参照頻度の値を減算すると共に、削除閾値以下となったエントリを削除する処理を並列に実行する。データ解凍器 20 のエントリ削除回路 206 も、複数設けることで回路規模は大きくなる一方、エントリの削除に要する時間は短縮することができる。データ解凍器 20 のエントリ削除回路も、LSI や FPGA 等によって構成することができる。

50

【 0 0 7 5 】

< 解凍処理 >

本実施形態においてエントリの削除を行うタイミングも、例えば上述した実施形態 2 と同様である。すなわち、データ解凍器 20 は、図 12 に示した処理フローに従って動作する。ただし、図 13 B に示す参照頻度の減算処理は、並列に実行される。

【 0 0 7 6 】

〔実施形態 5〕

本実施形態でも、データ圧縮器 10 の変換テーブル 102 及びデータ解凍器 20 の変換テーブル 202 からエントリを削除する。また、本実施形態においても、上述の実施形態と共通する構成については同一の符号を付して説明を省略し、主として相違点について説明する。本実施形態では、変換テーブルに空きがない場合に初めてシーケンシャルに全件のエントリを検索するのではなく、例えば、データストリーム制御回路が、変換テーブルからのエントリの削除と変換テーブルへのエントリの追加とを並行して行うものとする。すなわち、図 4 の圧縮処理における S7 や、図 8 の解凍処理における S16 で変換テーブルからエントリの削除も行う。

【 0 0 7 7 】

また、データストリーム制御回路 101 及びデータストリーム制御回路 201 は、変換テーブルにおけるエントリの登録位置を示す情報（「登録インデックス」とも呼ぶ）と、変換テーブルにおけるエントリの削除位置を示す情報（「削除インデックス」とも呼ぶ）とを有するものとする。登録インデックス及び削除インデックスは、それぞれ所定の規則に基づいて変換テーブル上の位置を巡回する。例えば、登録インデックスは、毎回変換テーブルの先頭から順に空き領域を探索して決定するようにしてもよいし、前回エントリを追加した位置から次に表れる空き領域を探索するようにしてもよい。そして、見つかった空き領域にエントリが追加される。また、削除インデックスも、変換テーブルを参照する毎に 1 つずつエントリの挿入位置を巡回する。そして、変換テーブルにおいて削除インデックスの位置にエントリが挿入されていた場合、当該エントリの参照頻度が削除閾値以下であるときは、当該エントリを削除する。一方、参照頻度が削除閾値以下でないときは、当該エントリの参照頻度を 1 減算する。

【 0 0 7 8 】

< 変換テーブルの生成及び更新 >

図 20 A ~ 図 20 I は、変換テーブルへのエントリの追加及び削除を説明するための図である。ここでは、「A A B B C C D D C C B B C C A A D D E E」というストリームデータが入力されるものとする。また、登録インデックスは、変換テーブルの先頭から最初に表れる空き領域を探索する。そして、削除インデックスは、変換テーブルの先頭から順にエントリの登録位置を巡回するものとする。

【 0 0 7 9 】

図 20 A に示すように、初期状態において登録インデックス及び削除インデックスは、先頭のエントリ登録位置を示している。この場合、図 20 B に示すように、入力されたシンボルペア「A A」は変換テーブルの 1 番目に登録される。また、登録インデックスは変換テーブルを先頭から探索した場合の最初の空き領域である 2 番目の領域を示す。また、削除インデックスは、次のエントリ登録位置に移動し、変換テーブルの 2 番目の領域を示す。

【 0 0 8 0 】

そして、次のシンボルペア「B B」が入力されると、図 20 C に示すように、登録インデックスが示す 2 番目の領域に被圧縮シンボル列「B B」が登録され、そのまま出力される。また、登録インデックスは、変換テーブルを先頭から探索した場合の最初の空き領域である 3 番目の領域を示す。削除インデックスは、次のエントリ登録位置に移動し、変換テーブルの 3 番目の領域を示す。同様に、次のシンボルペア「C C」が入力されると、図 20 D に示すように、登録インデックスが示す 3 番目の領域に被圧縮シンボル列「C C」が登録され、そのまま出力される。その後、登録インデックスは、変換テーブルを先頭か

10

20

30

40

50

ら探索した場合の最初の空き領域である4番目の領域を示す。削除インデックスは、次のエントリ登録位置に移動し、変換テーブルの4番目の領域を示す。

【0081】

そして、次のシンボルペア「DD」が入力されると、図20Eに示すように、登録インデックスが示す4番目の領域に被圧縮シンボル列「DD」が登録され、そのまま出力される。また、登録インデックスは、変換テーブルに空き領域がないため移動しない。一方、削除インデックスは、次のエントリ登録位置に移動し、変換テーブルの1番目の領域を示す。このとき、1番目に登録されているエントリは参照頻度が1であり、削除閾値以下であると判断され、図20Fに示すように当該エントリは削除される。そして、登録インデックスは、変換テーブルを先頭から探索した場合の最初の空き領域である1番目の領域へ移動する。

10

【0082】

そして、次のシンボルペア「CC」が入力されると、図20Gに示すように、被圧縮シンボルペアが「CC」である3番目のエントリの参照頻度が1加算され、圧縮シンボル「2」が出力される。また、登録インデックスは、変換テーブルを先頭から探索した場合の最初の空き領域である1番目の領域をすでに示しているため、移動しない。一方、削除インデックスは、次のエントリ登録位置に移動し、変換テーブルの2番目の領域を示す。このとき、2番目に登録されているエントリは参照頻度が1であり、削除閾値以下であると判断され、当該エントリは削除される。

20

【0083】

また、次のシンボルペア「BB」が入力されると、図20Hに示すように、登録インデックスが示す1番目の領域に被圧縮シンボル列「BB」が登録され、そのまま出力される。一方、削除インデックスは、次のエントリ登録位置に移動し、変換テーブルの3番目の領域を示す。このとき、3番目に登録されているエントリは参照頻度が削除閾値より大きい2であったため、当該エントリは削除されないが、当該工程において参照頻度は1減算され、「1」になる。

【0084】

そして、次のシンボルペア「CC」が入力されると、図20Iに示すように、被圧縮シンボルペアが「CC」である3番目のエントリの参照頻度が1加算され、圧縮シンボル「2」が出力される。また、登録インデックスは、変換テーブルを先頭から探索した場合の最初の空き領域である2番目の領域をすでに示しているため、移動しない。一方、削除インデックスは、次のエントリ登録位置に移動し、変換テーブルの4番目の領域を示す。このとき、4番目に登録されていたエントリは参照頻度が1であり、削除閾値以下であると判断され、当該エントリは削除される。

30

【0085】

以上のように、本実施形態では登録と削除とが並行して行われる。なお、データ解凍器20においても対応する処理が行われる。このようにすれば、削除のための処理量を抑えることができるため、好ましい時間で処理が完了すると共に、回路規模の増大も抑えることができる。

【0086】

なお、登録インデックスは、変換テーブルの先頭から最初に表れる空き領域でなく、最後に削除されたエントリの場所を指すようにしてもよい。例えば、図20Fにおいて、1行目のエントリが削除された場合、次に他のエントリが削除されるまで登録インデックスは1行目の位置を指すものとする。このようにすれば、テーブルの空きエントリを探すためのハードウェアを削減することができるとともに、登録処理に必要なクロック数を削減することができる。

40

【0087】

また、削除インデックスは、変換テーブルの先頭から順にエントリの登録位置を巡回しつつ、変換テーブルに空きがある場合には参照頻度が1であってもエントリを削除しないようにしてもよい。例えば、図20Fにおいて削除したエントリにその後登録がなされる

50

まで、削除インデックスの巡回は継続するがエントリの削除は行わない。したがって、図 20G においては 2 行目のエントリが削除されない。このようにすれば、変換テーブルにできるだけ多くのエントリを保持しておくことができるため、圧縮処理及び解凍処理においてシンボルペアの変換テーブルへのヒット率が向上する。

【0088】**〔実施形態 6〕**

上述のデータ圧縮器 10、データ解凍器 20 を、それぞれ複数直列に接続して圧縮率を向上させたデータ圧縮装置、データ解凍装置を示す。実施形態 6 も、実施形態 1 ~ 5 と共通する構成を含むので、共通する構成については同一の符号を付して説明を省略し、主として相違点について説明する。

【0089】

図 21 は、データ圧縮器 10 を多段接続したデータ圧縮装置の一例を示す。図 21 に示すデータ圧縮装置は、直列に接続された 4 つのデータ圧縮器 10 から形成される。但し、データ圧縮装置を形成するデータ圧縮器の数は適宜選択可能である。ここで、データ圧縮装置に含まれる複数のデータ圧縮器 10 を、送信側装置 1 (図示せず) に近い方から順に (すなわち伝送路 3 から遠い順に) 1 段目、2 段目、3 段目、4 段目と呼ぶ。

【0090】

1 段目のデータ圧縮器 10 は、圧縮処理の結果として得られた圧縮シンボルを含むシンボル列と、当該シンボル列を形成する各シンボルに対応するフラグ (付加ビット) が、シンボル列に含まれるシンボル順と一致する順序で並べられた付加ビット列とを出力する。

【0091】

2 段目のデータ圧縮器 10 は、1 段目からのシンボル列及び付加ビットに対する 9 ビット単位の圧縮処理によって得られたシンボル列と、当該シンボル列に対応するフラグ群で形成された付加ビット列とを出力する。同様に、3 段目のデータ圧縮器 10 は、2 段目からのシンボル列及び付加ビットに対する 10 ビット単位の圧縮処理によって得られたシンボル列と、当該シンボル列に対応するフラグ群で形成されたビット列とを出力する。同様に、4 段目のデータ圧縮器 10 は、3 段目からのシンボル列及び付加ビットに対する 11 ビット単位の圧縮処理によって得られたシンボル列と、当該シンボル列に対応するフラグ群で形成されたビット列とを出力する。そして、4 段目のデータ圧縮器 10 の出力は、伝送路 3 を介してデータ解凍器 20 へ送信される。

【0092】

図 22 は、データ解凍器 20 を多段接続したデータ解凍装置の一例を示す。図 22 に示すデータ解凍装置は、直列に接続された 4 つのデータ解凍器 20 から形成される。但し、データ解凍装置を形成するデータ解凍器の数は、データ圧縮装置を形成するデータ圧縮器の数以上であればよい。ここで、データ解凍装置に含まれる複数のデータ解凍器 20 を、受信側装置 2 (図示せず) に近い方から順に (伝送路 3 から遠い順に) 1 段目、2 段目、3 段目、4 段目と呼ぶ。

【0093】

4 段目のデータ解凍器 20 には、データ圧縮装置による 4 段階の圧縮処理の結果として得られた圧縮シンボルを含むシンボル列と、各段階においてシンボルが圧縮されたか否かを示すフラグ (付加ビット) を、シンボル列に含まれるシンボルと一致する順序で並べた付加ビット列とが入力される。4 段目のデータ解凍器 20 は、データ圧縮装置の 4 段目のデータ圧縮器 10 が出力した付加ビットに基づいて、11 ビットのシンボル単位で解凍処理を行う。

【0094】

同様に、3 段目のデータ解凍器 20 には、4 段目のデータ解凍器 20 の解凍処理の結果として得られたシンボル列と、データ圧縮装置の 3 段目においてシンボルが圧縮されたか否かを示すフラグ (付加ビット) を、シンボル列に含まれるシンボルと一致する順序で並べたビット列とが入力される。すなわち、3 段目のデータ解凍器 20 は、データ圧縮装置の 3 段目のデータ圧縮器 10 が出力した付加ビットに基づいて、10 ビットのシンボル単

10

20

30

40

50

位で解凍処理を行う。

【0095】

また、2段目のデータ解凍器20には、3段目のデータ解凍器20の解凍処理の結果として得られたシンボル列と、データ圧縮装置の2段階目においてシンボルが圧縮されたか否かを示すフラグ(付加ビット)を、シンボル列に含まれるシンボルと一致する順序で並べたビット列とが入力される。すなわち、2段目のデータ解凍器20は、データ圧縮装置の2段目のデータ圧縮器10が出力した付加ビットに基づいて、9ビットのシンボル単位で解凍処理を行う。

【0096】

そして、1段目のデータ解凍器20には、2段目のデータ解凍器20の解凍処理の結果として得られたシンボル列と、データ圧縮装置の1段階目においてシンボルが圧縮されたか否かを示すフラグ(付加ビット)を、シンボル列に含まれるシンボルと一致する順序で並べたビット列とが入力される。すなわち、1段目のデータ解凍器20は、入力された付加ビットに基づいて、8ビットのシンボル単位で解凍処理を行う。

10

【0097】

このようにすれば、データ圧縮器10による圧縮前のストリームデータが復号され、受信側装置2に出力される。また、本実施形態に係る圧縮処理を重畳的に適用することで、圧縮率を向上させることができる。

【0098】

また、本実施形態では、データ圧縮装置で行われる圧縮処理の段数以上の解凍処理を行うことができれば、それぞれの段数が異なっても圧縮データを解凍することができる。すなわち、圧縮側の段数の方が復号側の段数より少ない場合でも、シンボルのデータ幅に基づいて圧縮側の段数に対応する段数のデータ解凍器20が解凍処理を行うようにする。ストリームデータの内容によっては、ある程度の段数以上にデータ圧縮器10を増やしても圧縮率の改善が頭打ちになる場合もあるところ、図21に示す圧縮段数制御部12を追加して圧縮処理の段数を変更できるようにしてもよい。例えば、各段階のデータ圧縮器10においてシンボルを変換した割合又は圧縮率を算出し、圧縮段数制御部12は、シンボルを変換した割合又は圧縮率が所定の閾値を下回った場合に、それ以降の段階の圧縮処理を行わないように制御する。なお、このような処理は、各段階のデータ圧縮器10が自律的に判断するようにしてもよい。上述の通り、データ圧縮装置において行われる圧縮処理の段数を変更しても、図22に示したデータ解凍装置の構成により対応する段数の解凍処理を行うことができる。

20

30

【0099】

〔実施形態7〕

上記の実施形態に示した技術は、様々な伝送路に適用することができる。例えば、伝送路3は、光ケーブルやHDMI(High-Definition Multimedia Interface, 登録商標)、USB(Universal Serial Bus)等の有線であってもよいし、Bluetooth(登録商標)等の無線であってもよい。また、送信側装置1、データ圧縮器10、データ解凍器20、及び受信側装置2が同期して、変換テーブル102及び変換テーブル202のエントリや、実施形態5に示した登録インデックス、削除インデックスの開始位置を予め定められた状態に初期化するようにしてもよい。このような指示は、直接信号線を介して送信されてもよいし、例えばデータ圧縮器が圧縮データを予め定められたフォーマットに基づいてパケット化して送信する場合、例えばヘッダに初期化を指示するフラグを含めるようにしてもよい。なお、圧縮対象のデータはストリームデータに限られず、例えばUSB等を介して送受信されるデータ列であってもよい。

40

【0100】

図23は、圧縮データを独自のプロトコルでパケット化して送信する場合のフォーマットの一例を示す図である。図23に示すパケットの例は、伝送路3が16ビット幅、4チャンネル、最大データ長1023シンボルであるものとする。ヘッダには、多重化された伝送路のいずれを使用するか特定する「Channel#」(2ビット)、受信できたか否かを応答

50

する「Ack/Nack」(1ビット)、変換テーブル等の初期化を指示する「Flush」(1ビット)、シンボルの復号を実行せずシンボルをそのまま通過させるバイパスを指示する「Bypass」(1ビット)、「Data」部の長さを示す「Data Length」(10ビット)の領域を含む。また、「Data」部の各々には1シンボル、最大で1023シンボルを格納することができるものとする。このようにすれば、シンボルの境界が明確になる。さらにフッタには、伝送データの誤り検出及び訂正を行うための「Checksum」の領域を有する。このようなフォーマットを採用し、例えば最終段のデータ圧縮器10の後でパケット化する。また、ヘッダの「Flush」部に1が入っていれば、変換テーブルのエントリ等を初期化する。この場合において、「Flush」が1であり且つ「Datalength」が0のときは変換テーブルを予め定められた値に初期化し、「Flush」が1であり且つ「Datalength」が0でないときは「Data」部を利用して変換テーブルの初期値を送信するようにしてもよい。「Flush」が1であり且つ「Datalength」が0でないとき、例えば、「Data」部の各々にシンボルペアを格納して送信し、受信側は「Data」部に格納された順にシンボルペアを変換テーブルのエントリに登録する。また、パケットの送信前にマンチェスタ符号等によるプリアンブル信号を送信し、同期をとるようにしてもよい。

10

【0101】

図24は、本実施形態に係る通信の一例を示すシーケンス図である。長方形で囲われた情報は、圧縮側と解凍側とにおいてそれぞれの状態を示しており、各段階において所定の記憶領域に記憶される。このような記憶領域は、上述のチャネル数分設けられるものとする。圧縮側には、パケットの再送に用いられるパケットサイズのバッファ「Pbuf」と、当該プロトコルの制御に用いられる「Seq」レジスタとが設けられている。また、解凍側には、直近の送信内容(前回のAck又はNack)を保持しておく「Abuf」と、当該プロトコルの制御に用いられる「Seq」レジスタと、タイムアウト用カウンタが設けられている。また、初期状態では、AbufにAckがセットされ、圧縮側のSeqが0、解凍側のSeqが1とされる。なお、Seqは、送信時に0と1とを切り替え(トグル)し、再送時には切り替えをしない。

20

【0102】

具体的には、エラーなく解凍側からAck1が送信された後、圧縮側から送信がなければ復号側からタイムアウトを送信する。その後、圧縮側から新たに通信を開始する場合、変換テーブルのエントリ等を初期化するため、Flush1を送信する。一方、解凍側で問題なく受信できると、Ack2を送信する。その後、エラー無く通信が確立すれば、圧縮側から圧縮データの送信を開始し、Data1、Data2、Data3・・・と送信される。

30

【0103】

図25は、本実施形態に係る通信の他の例を示すシーケンス図である。図25に示すように、解凍側は、Ack/Nackを送信した後、タイマーを始動させ、タイムアウトの判断を行う。一方、圧縮側でエラーを検出した場合、特に応答せず、解凍側でのタイムアウトを待つ。解凍側でタイムアウトを待つことにより、圧縮側からの送信を継続させることができる。

40

【0104】

具体的には、圧縮側からFlush1が送信され、解凍側でチェックサムエラー等が発生したものとする。この場合、解凍側からはNack1が送信され、解凍側ではタイマーを始動する。一方、Nack1を受信した圧縮側はFlush1を再送する。そして、解凍側でタイムアウト前にFlush1を受信した場合、Ack2を送信し、タイマーを始動する。一方、圧縮側でAck2の受信時にチェックサムエラー等を検出した場合、圧縮側は何も応答しない。その後、解凍側でタイムアウトが発生すると、Ack2を再送し、タイマーを始動する。また、圧縮側から送信データがなければ、タイムアウトを繰り返し、Ack2を再送する。一方、圧縮側で圧縮データの送信を開始すると、Data1が送信される。また、解凍側で問題なく受信できれば、Ack3を送信する。

50

【0105】

以上のように、解凍側のタイムアウトを利用することで、エラーが発生した場合も適切に通信を継続することができる。

【 0 1 0 6 】

〔実施形態 8〕

上述の圧縮処理及び解凍処理は、転送されるデータ量を削減するだけでなく、転送されるデータの内容をいわゆる換字式暗号によって暗号化するという側面を有している。すなわち、初期の変換テーブルのエントリや、登録インデックスの初期値、削除インデックスの初期値がわからなければ圧縮データを復号できない。ここで、上述の実施形態に係る圧縮処理及び解凍処理は、変換テーブルに登録されていないシンボルペアの出現時にはデータ圧縮器 10 はシンボルペアを変換せずに出力するという特徴を有している。このように
10

することで、データ解凍器 20 が作成する変換テーブルにも対応するエントリを登録することができる一方、変換せずに伝送されるシンボルペアを第三者が盗み見ることで当該暗号が解読されるおそれがある。そこで、本実施形態では、変換せずに出力されるシンボルペアの少なくとも一部についてスクランブルをかけ、暗号強度を向上させる。

【 0 1 0 7 】

図 26 は、スクランブル回路を含むデータ圧縮 / 解凍システム（データ暗号化 / 復号化システム）の一例を示す図である。本実施形態に係るデータ圧縮 / 解凍システムは、データ圧縮器 10 の出力に対し、伝送路 3 へ送出する前にスクランブルをかけるスクランブル回路 13 を備えている。また、伝送路 3 を介して受信した圧縮データに対し、データ解凍器 20 が解凍処理を行う前にスクランブルを解除するスクランブル回路 23 を備えている
20

【 0 1 0 8 】

このようなデータ圧縮 / 解凍システムにおいて、スクランブル回路 13 及びスクランブル回路 23 が、所定の暗号化方式により伝送路 3 を流れるデータにスクランブルをかける。暗号化方式は特に限定されないが、例えば次のような手法を採用することができる。なお、本実施形態ではシンボルは 8 ビットであるものとする。

【 0 1 0 9 】

まず、データ圧縮器 10 及びデータ解凍器 10 は、あらかじめ変換テーブルの初期状態のデータセット（初期の変換テーブルのエントリ、登録インデックスの初期値、及び削除インデックスの初期値：共通鍵）を保持しているものとする。そして、データ圧縮器 10
30

及びデータ解凍器 10 は、変換テーブルの初期状態のデータセットから例えば MD5 のチェックサム A を求める。また、スクランブル回路 13 及びスクランブル回路 23 は、あらかじめ定められたパスワード（共通鍵）を保持しているものとする。そして、当該パスワードに基づいて例えば MD5 のチェックサム B を求める。そして、圧縮されたシンボルのインデックスを「i」として、下記の X (i) を算出する。

$$X (i) = A [i + 8 : i] \% B$$

なお、上記の式はシンボルが 8 ビットの場合を表し、インデックス i は 1 からカウントするものとする。MD5 によれば 32 文字の 16 進数からなるチェックサムが得られ、A [i + 8 : i] は当該チェックサムのビットを巡回させて用いている。そして、下記の式により X (i) と圧縮されたシンボル S (i) との排他的論理和をとれば、スクランブルを
40

かけた値 S ' (i) を得ることができ、S ' (i) が伝送路 3 へ送信される。

$$S ' (i) = X (i) \text{ xor } S (i)$$

【 0 1 1 0 】

本実施形態では、データ解凍器 20 及びスクランブル回路 23 も、上述の共通鍵を有しているため、データ圧縮器 10 及びスクランブル回路 13 と同様に X (i) を求めることができる。そして、下記の式により X (i) と S ' (i) との排他的論理和をとれば、スクランブルを解除して圧縮されたシンボル S (i) を求めることができる。

$$S (i) = X (i) \text{ xor } S ' (i)$$

【 0 1 1 1 】

実施形態 8 によれば、圧縮しつつ暗号化の強度も向上させることができる。

10

20

30

40

50

【 0 1 1 2 】

〔 実施形態 9 〕

ハードウェアによってデータ圧縮器 10 及びデータ解凍器 20 を形成する場合、変換テーブルへのエントリの登録がボトルネックとなりデータパイプライン全体の流れが遅延する（「ブロック」されるとも呼ぶ）おそれがある。そこで、本実施形態では、変換テーブルへのエントリの追加にかかる時間（処理サイクル）の間、後続のシンボルが変換テーブルのエントリにヒットしない場合であってもシンボルを変換テーブルに追加せずにそのまま出力する。また、解凍側においても変換テーブルへのエントリの追加を行う場合、対応する時間（処理サイクル）だけ後続のシンボルをそのまま出力する。このようにすれば、出力スループットを向上させることができる。また、このような圧縮処理及び解凍処理を、ノンブロック圧縮及びノンブロック復号と呼ぶものとする。ノンブロック圧縮及び復号は、上述した実施形態の各々に適用することができる。

10

【 0 1 1 3 】

例えば、1シンボルが8ビットであって、圧縮前の2シンボルを圧縮後において1シンボルに変換する場合において、変換テーブルへのエントリの追加にかかる時間が、無変換であれば4シンボルを出力できるだけの処理サイクルであるものとする。この場合において、シンボル列「A A B B C C D D C C B B A A B B A A C C」を処理する例を説明する。なお、図 2 7 A に示すように、変換テーブルの初期状態はエントリが登録されていないものとする。

【 0 1 1 4 】

ノンブロック圧縮を行う場合、図 2 7 B に示すように、最初のシンボルペア「A A」をそのまま出力すると共に、被圧縮シンボル列を「A A」とするエントリを変換テーブルに登録する。なお、図 2 7 B の例ではエントリが登録されているが、エントリの追加には4シンボルを処理するだけの時間がかかるため、図 2 7 C に示すように、後続のシンボル列「B B C C」は変換テーブルに登録せずそのまま出力する。

20

【 0 1 1 5 】

その後、図 2 7 D に示すように、シンボルペア「D D」が入力されると、変換テーブルにはヒットせず、且つ変換テーブルへの先のエントリの登録が完了しているため、シンボルペア「D D」をそのまま出力すると共に、被圧縮シンボル列を「D D」とするエントリを変換テーブルに登録する。そして、図 2 7 E に示すように、変換テーブルへのエントリの登録にかかる時間は、後続のシンボル列「C C B B」を変換テーブルに登録せずに出力する。

30

【 0 1 1 6 】

また、図 2 7 F に示すように、次のシンボルペア「A A」が入力された場合、シンボルペア「A A」は変換テーブルのエントリにヒットするため、エントリにおいて対応付けられている圧縮シンボル「0」を出力する。このように、変換テーブルへのエントリ登録にかかる時間に出力をブロックしない分、スループットを向上させることができる。

【 0 1 1 7 】

同様に、ノンブロック復号を行う場合、図 2 8 A に示すように変換テーブルの初期状態は、圧縮側と同様にエントリが登録されていない状態にする。そして、図 2 8 B に示すように、最初のシンボルペア「A A」が入力されると、変換テーブルにはヒットしないため、シンボル列「A A」を出力すると共に、変換テーブルの復号シンボル列を「A A」とするエントリを変換テーブルに追加する。また、図 2 8 C に示すように、後続のシンボル列「B B C C」は、変換テーブルに登録せずにそのまま出力する。

40

【 0 1 1 8 】

その後、図 2 8 D に示すように、シンボルペア「D D」が入力されると、変換テーブルにはヒットせず、且つ変換テーブルへの先のエントリの登録が完了しているため、シンボルペア「D D」をそのまま出力すると共に、復号シンボル列を「D D」とするエントリを変換テーブルに登録する。そして、図 2 8 E に示すように、変換テーブルへのエントリの登録にかかる時間は、後続のシンボル列「C C B B」を変換テーブルに登録せずに出力す

50

る。

【 0 1 1 9 】

また、図 2 8 F に示すように、次のシンボル「 0 」が入力された場合、シンボル「 0 」は圧縮シンボルとして変換テーブルのエントリに登録されているため、当該エントリにおいて対応付けられている復号シンボル「 A A 」に変換して出力する。このように、解凍側においても、変換テーブルへのエントリ登録にかかる時間に出力をブロックしない分、スループットを向上させることができるようになる。

【 0 1 2 0 】

〔その他の変形例〕

また、圧縮前のストリームデータに含まれるシンボルの種類が予めわかっている場合、変換テーブルにおいて圧縮後のシンボルとして、圧縮前のストリームデータには存在しないシンボルを登録しておくようにしてもよい。このようにすれば、付加ビットを参照しなくても圧縮されたシンボルであるか否か判断できるようになる。

10

【 0 1 2 1 】

例えば、ストリームデータが所定の文字コードが直列に並べられたテキストデータである場合には、圧縮前のシンボルとして 2 文字のペアが登録され、圧縮後のシンボルとしては当該文字コードにおいて未使用のビット列が割り当てられる。また、例えば、ストリームデータが遺伝子データ（「 A 」、「 G 」、「 T 」及び「 C 」の核酸コード配列）である場合には、ルックアップテーブルの圧縮前の 2 シンボルとして、「 A 」、「 G 」、「 T 」及び「 C 」から 2 文字を取り出した順列が登録され、圧縮後の 1 シンボルとして、「 A 」、

20

【 0 1 2 2 】

また、上記の実施形態では 2 つのシンボルを 1 つのシンボルに変換したが、圧縮前のシンボルの組合せは 2 つには限られない。複数のシンボルを、それ未満の数のシンボルに置き換えるという構成であれば、データ圧縮器として機能する。ただし、変換テーブルのエントリとストリームデータのシンボルペアとの一致率を向上させるという観点や、変換テーブルの容量を低減させるといった観点からは、2 つのシンボルを 1 つのシンボルに圧縮する態様が好ましい。

【 0 1 2 3 】

また、上記の実施形態では、1 番目のシンボル及び 2 番目のシンボルのシンボルペアが変換テーブルに登録されていない場合、次の処理対象は 3 番目のシンボル及び 4 番目のシンボルとし、これらのシンボルペアが変換テーブルに登録されているか判断していた。ここで、1 番目のシンボル及び 2 番目のシンボルのシンボルペアが変換テーブルに登録されていない場合、次の処理対象を 2 番目のシンボル及び 3 番目のシンボルのシンボルペアとしてもよい。このようにすれば、圧縮率の向上が期待できる。

30

【 0 1 2 4 】

以上説明した実施形態及び変形例の構成は、適宜組み合わせることができる。

【符号の説明】

【 0 1 2 5 】

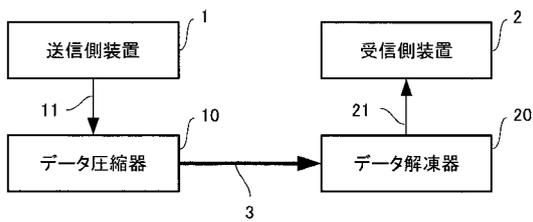
- | | | |
|---------------|--------------|--|
| 1 | 送信側装置 | |
| 1 0 | データ圧縮器 | |
| 1 0 1 | データストリーム制御回路 | |
| 1 0 2 | 変換テーブル | |
| 1 0 3 ~ 1 0 6 | ラッチ | |
| 1 0 7 | マルチプレクサ | |
| 2 | 受信側装置 | |
| 2 0 | データ解凍器 | |
| 2 0 1 | データストリーム制御回路 | |
| 2 0 2 | 変換テーブル | |
| 2 0 3 ~ 2 0 5 | ラッチ | |

40

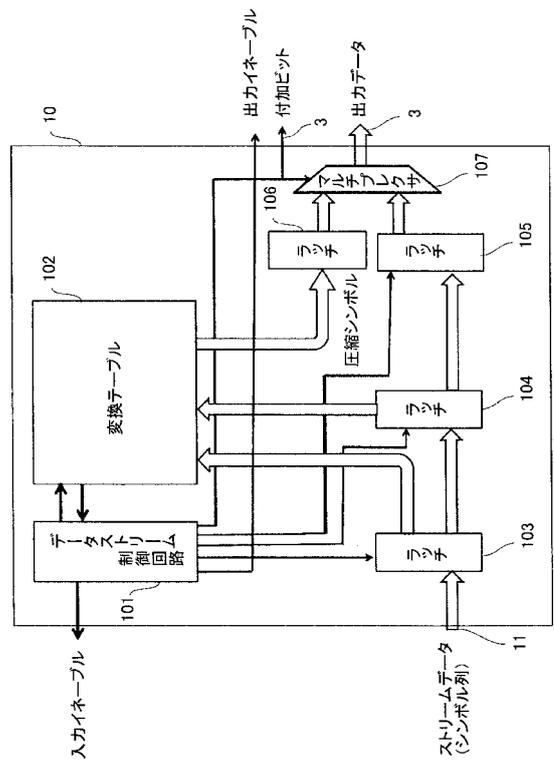
50

3 伝送路

【 図 1 】



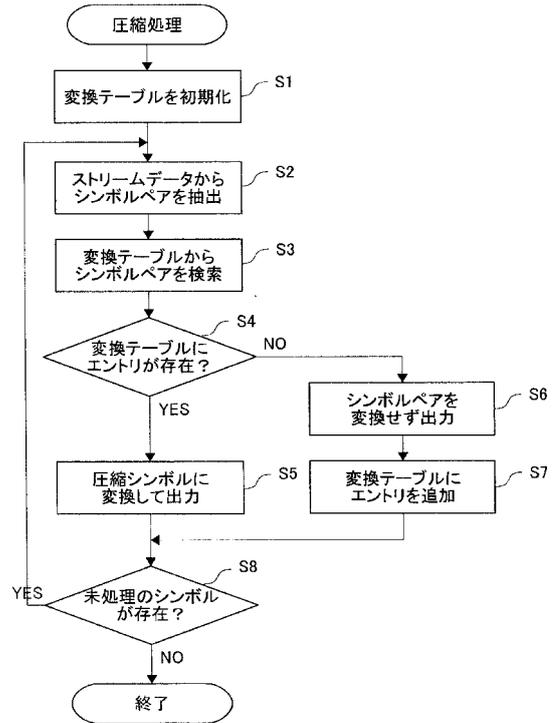
【 図 2 】



【 図 3 】

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
B	B	1	1
C	C	2	2
D	D	3	1
⋮	⋮	⋮	⋮

【 図 4 】



【 図 5 A 】

入力データ: AABBCDDCCBBCCAA

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		

出力データ:

【 図 5 C 】

入力データ: AABBCDDCCBBCCAA

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
B	B	1	1

出力データ: AAB B

【 図 5 B 】

入力データ: AABBCDDCCBBCCAA

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1

出力データ: AA

【 図 5 D 】

入力データ: AABBCDDCCBBCCAA

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
B	B	1	1
C	C	2	1

出力データ: AABBC C

【 図 5 E 】

入力データ:AABBCCDDCCBBCCAA

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
B	B	1	1
C	C	2	1
D	D	3	1

出力データ:AABBCCDD

【 図 5 F 】

入力データ:AABBCCDDCCBBCCAA

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
B	B	1	1
C	C	2	2
D	D	3	1

出力データ:AABBCCDD2

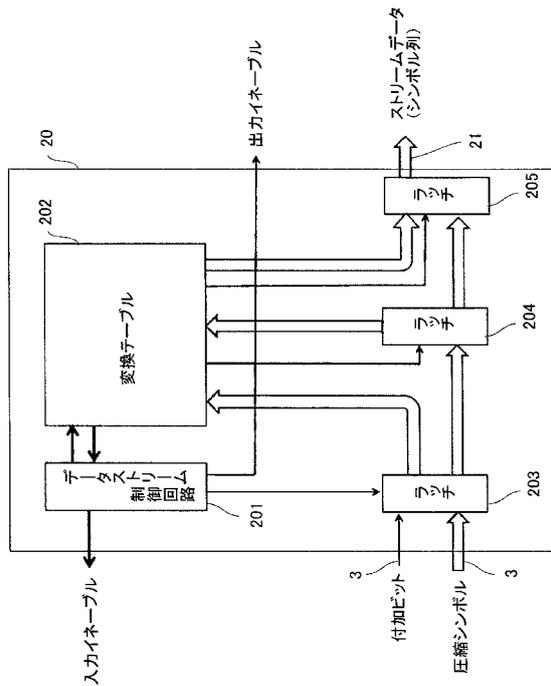
【 図 5 G 】

入力データ:AABBCCDDCCBBCCAA

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
B	B	1	2
C	C	2	2
D	D	3	1

出力データ:AABBCCDD21

【 図 6 】



【 図 5 H 】

入力データ:AABBCCDDCCBBCCAA

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
B	B	1	2
C	C	2	3
D	D	3	1

出力データ:AABBCCDD212

【 図 5 I 】

入力データ:AABBCCDDCCBBCCAA

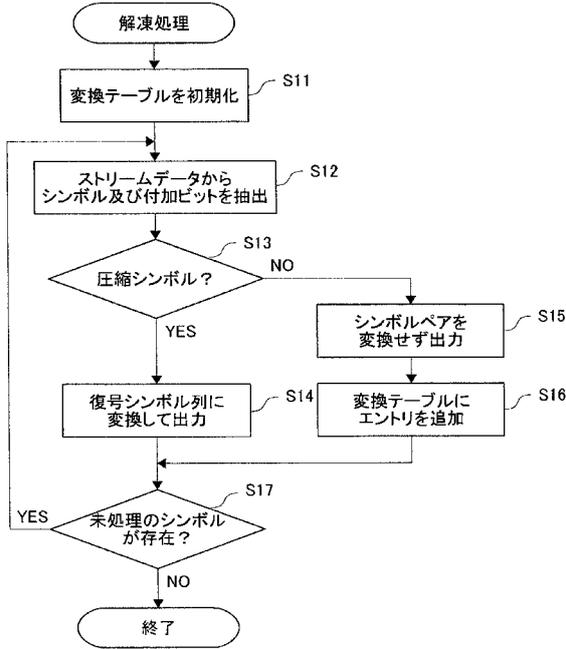
被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	2
B	B	1	2
C	C	2	3
D	D	3	1

出力データ:AABBCCDD2120

【 図 7 】

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
B	B	1	1
C	C	2	2
D	D	3	1
⋮	⋮	⋮	⋮

【 図 8 】



【 図 9 A 】

入力データ: AABCCDD2120

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		

出力データ:

【 図 9 B 】

入力データ: AABCCDD2120

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1

出力データ: AA

【 図 9 C 】

入力データ: AABCCDD2120

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
B	B	1	1

出力データ: AAB

【 図 9 E 】

入力データ: AABCCDD2120

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
B	B	1	1
C	C	2	1
D	D	3	1

出力データ: AABCCDD

【 図 9 D 】

入力データ: AABCCDD2120

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
B	B	1	1
C	C	2	1

出力データ: AABCC

【 図 9 F 】

入力データ: AABCCDD2120

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
B	B	1	1
C	C	2	2
D	D	3	1

出力データ: AABCCDDCC

【図 9 G】

入力データ: AABBCDD2120

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
B	B	1	2
C	C	2	2
D	D	3	1

出力データ: AABBCDDCCBB

【図 9 H】

入力データ: AABBCDD2120

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
B	B	1	2
C	C	2	3
D	D	3	1

出力データ: AABBCDDCCBBCC

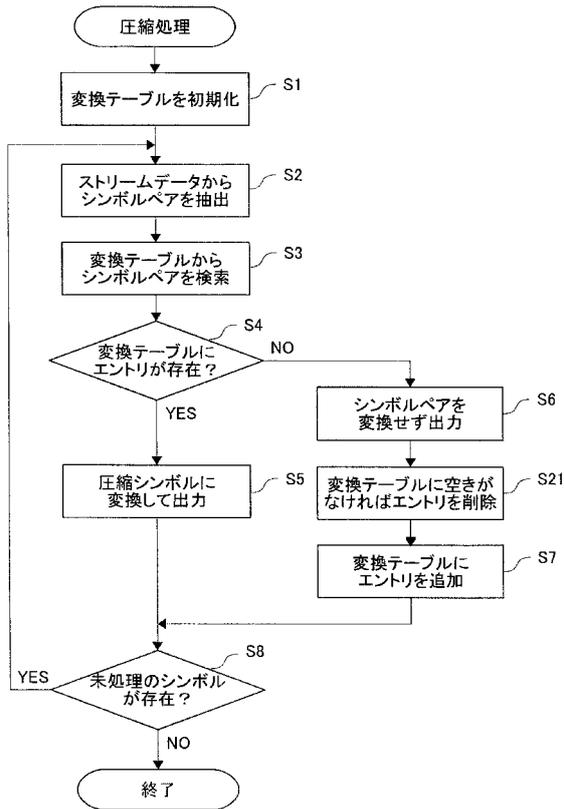
【図 9 I】

入力データ: AABBCDD2120

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	2
B	B	1	2
C	C	2	3
D	D	3	1

出力データ: AABBCDDCCBBCCAA

【図 1 0】



【図 1 1 A】

入力データ: AABBCDDCCBBCCAADDDDDDEE

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	2
B	B	1	2
C	C	2	3
D	D	3	4

出力データ: AABBCDD210333EE

【図 1 1 B】

入力データ: AABBCDDCCBBCCAADDDDDDEE

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
B	B	1	1
C	C	2	2
D	D	3	3

出力データ: AABBCDD210333EE

【図 1 1 C】

入力データ: AABBCDDCCBBCCAADDDDDDEE

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
C	C	2	2
D	D	3	3

出力データ: AABBCDD210333EE

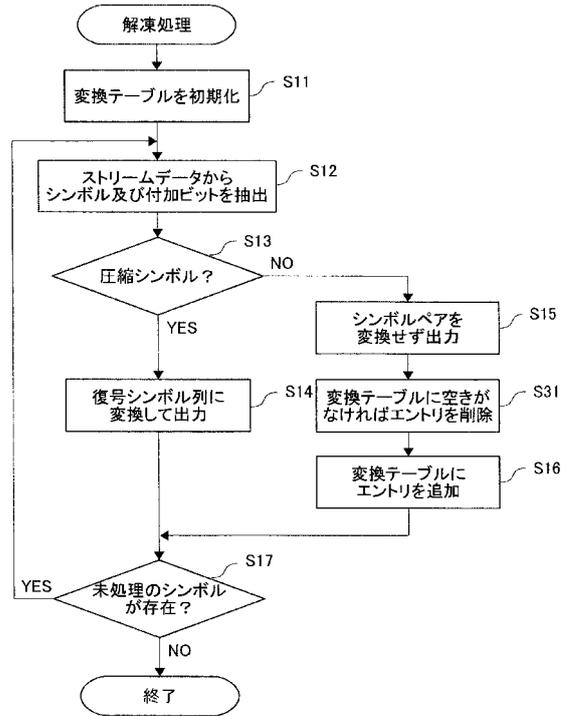
【図 1 1 D】

入力データ: AABBCDDCCBBCCAADDDDDDEE

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
E	E	0	1
C	C	2	2
D	D	3	3

出力データ: AABBCDD210333EE

【図 1 2】



【図 1 3 A】

入力データ: AABBCDD210333EE

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	2
B	B	1	2
C	C	2	3
D	D	3	4

出力データ: AABBCDDCCBBCCAADDDDDDEE

【図 1 3 D】

入力データ: AABBCDD210333EE

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
E	E	0	1
C	C	2	2
D	D	3	3

出力データ: AABBCDDCCBBCCAADDDDDDEE

【図 1 3 B】

入力データ: AABBCDD210333EE

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
B	B	1	1
C	C	2	2
D	D	3	3

出力データ: AABBCDDCCBBCCAADDDDDDEE

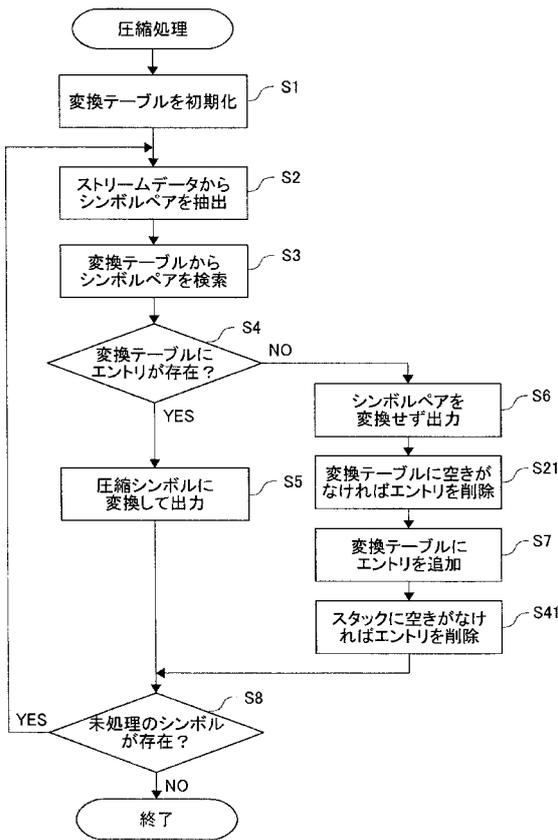
【図 1 3 C】

入力データ: AABBCDD210333EE

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
C	C	2	2
D	D	3	3

出力データ: AABBCDDCCBBCCAADDDDDDEE

【図14】



【図15A】

入力データ: AABBCDDCCBBCCAEE

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	2
B	B	1	2
C	C	2	3
D	D	3	1

スタック

シンボル0	シンボル1	圧縮シンボル	参照頻度
A	A	0	2
C	C	2	3

出力データ: AABBCDD210EE

【図15B】

入力データ: AABBCDDCCBBCCAEE

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
B	B	1	2
C	C	2	2
D	D	3	1

スタック

シンボル0	シンボル1	圧縮シンボル	参照頻度

出力データ: AABBCDD210EE

【図15C】

入力データ: AABBCDDCCBBCCAEE

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
B	B	1	2
C	C	2	2
D	D	3	1

スタック

シンボル0	シンボル1	圧縮シンボル	参照頻度

出力データ: AABBCDD210EE

【図15D】

入力データ: AABBCDDCCBBCCAEE

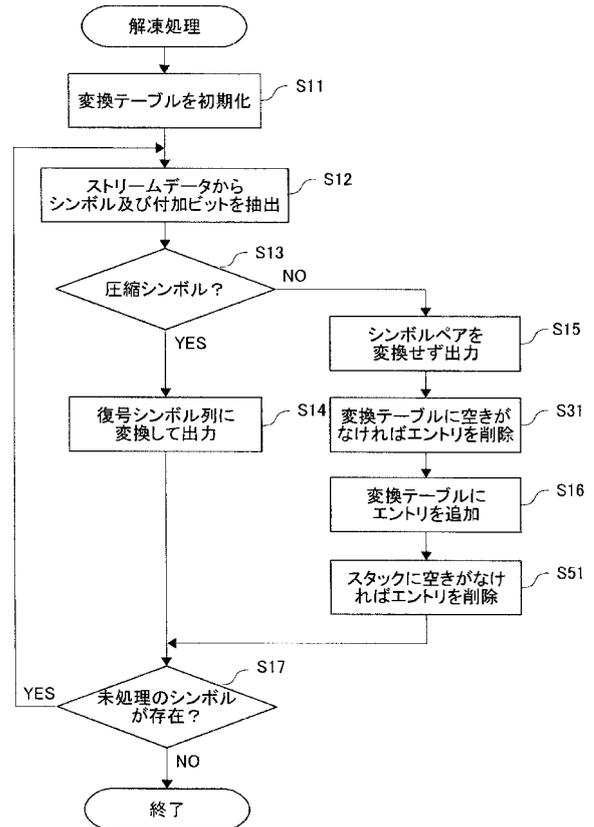
被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
E	E	0	1
B	B	1	2
C	C	2	2
D	D	3	1

スタック

シンボル0	シンボル1	圧縮シンボル	参照頻度

出力データ: AABBCDD210EE

【図16】



【図 17 A】

入力データ:AABBCCDD210EE

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	2
B	B	1	2
C	C	2	3
D	D	3	1

スタック			
シンボル0	シンボル1	圧縮シンボル	参照頻度
A	A	0	2
C	C	2	3

出力データ:AABBCCDDCCBBCCAEE

【図 17 C】

入力データ:AABBCCDD210EE

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
B	B	1	2
C	C	2	2
D	D	3	1

スタック			
シンボル0	シンボル1	圧縮シンボル	参照頻度

出力データ:AABBCCDDCCBBCCAEE

【図 17 B】

入力データ:AABBCCDD210EE

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
B	B	1	2
C	C	2	2
D	D	3	1

スタック			
シンボル0	シンボル1	圧縮シンボル	参照頻度

出力データ:AABBCCDDCCBBCCAEE

【図 17 D】

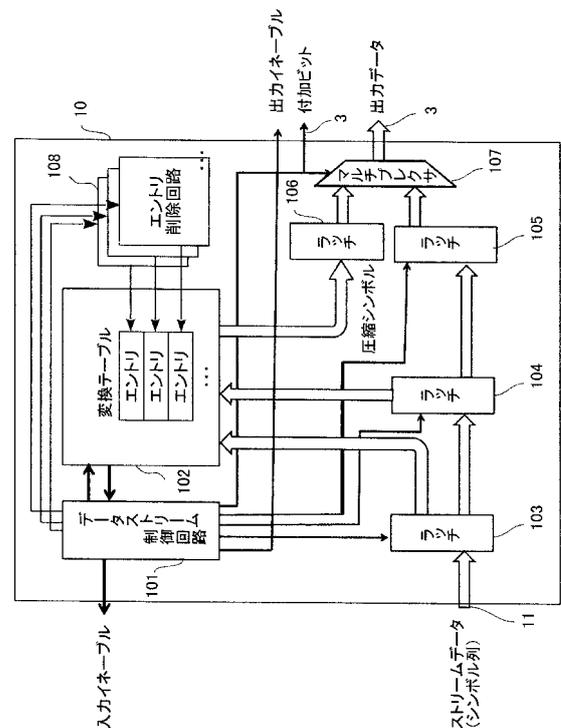
入力データ:AABBCCDD210EE

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
E	E	0	1
C	C	2	2
D	D	3	3

スタック			
シンボル0	シンボル1	圧縮シンボル	参照頻度

出力データ:AABBCCDDCCBBCCAEE

【図 18】



【 図 2 0 D 】

入力データ: AABBCDDCCBBCCAAADDEE				
被圧縮シンボル列		圧縮シンボル	参照頻度	
シンボル0	シンボル1			
A	A	0	1	
B	B	1	1	
C	C	2	1	
登録インデックス→				
出力データ: AABBC				
←削除インデックス				

【 図 2 0 E 】

入力データ: AABBCDDCCBBCCAAADDEE				
被圧縮シンボル列		圧縮シンボル	参照頻度	
シンボル0	シンボル1			
A	A	0	1	
B	B	1	1	
C	C	2	1	
D	D	3	1	
登録インデックス→				
出力データ: AABBCDD				
←削除インデックス				

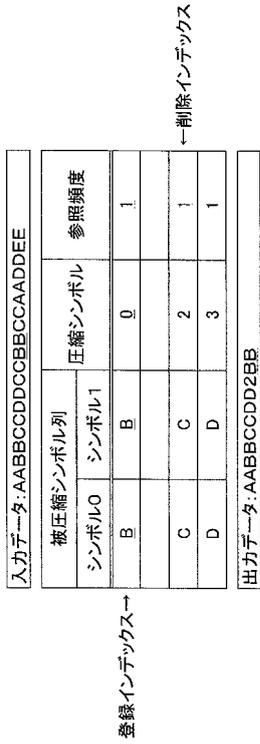
【 図 2 0 F 】

入力データ: AABBCDDCCBBCCAAADDEE				
被圧縮シンボル列		圧縮シンボル	参照頻度	
シンボル0	シンボル1			
B	B	1	1	
C	C	2	1	
D	D	3	1	
登録インデックス→				
出力データ: AABBCDD				
←削除インデックス				

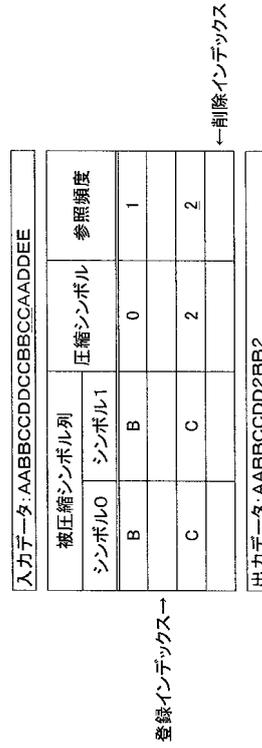
【 図 2 0 G 】

入力データ: AABBCDDCCBBCCAAADDEE				
被圧縮シンボル列		圧縮シンボル	参照頻度	
シンボル0	シンボル1			
C	C	2	2	
D	D	3	1	
登録インデックス→				
出力データ: AABBCDD2				
←削除インデックス				

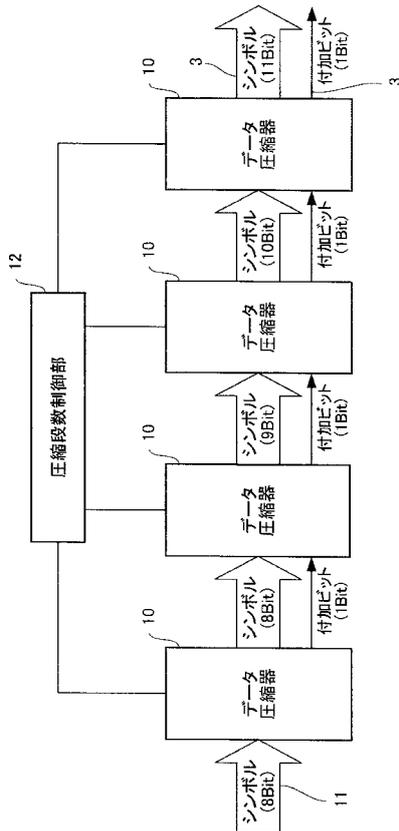
【図 20H】



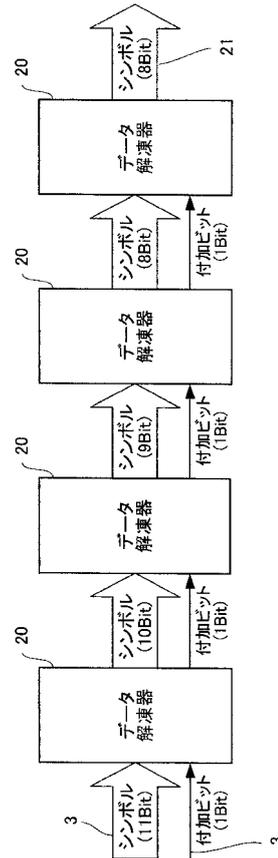
【図 20I】



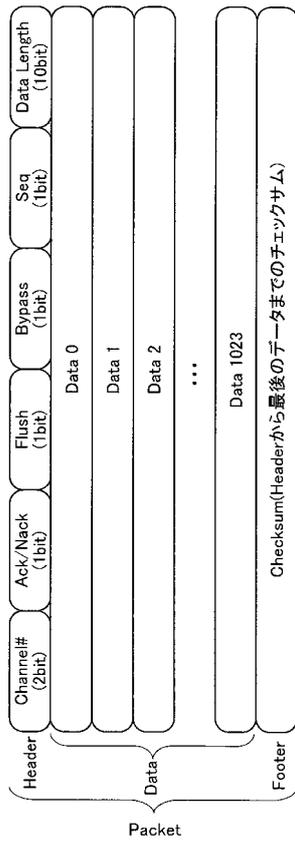
【図 21】



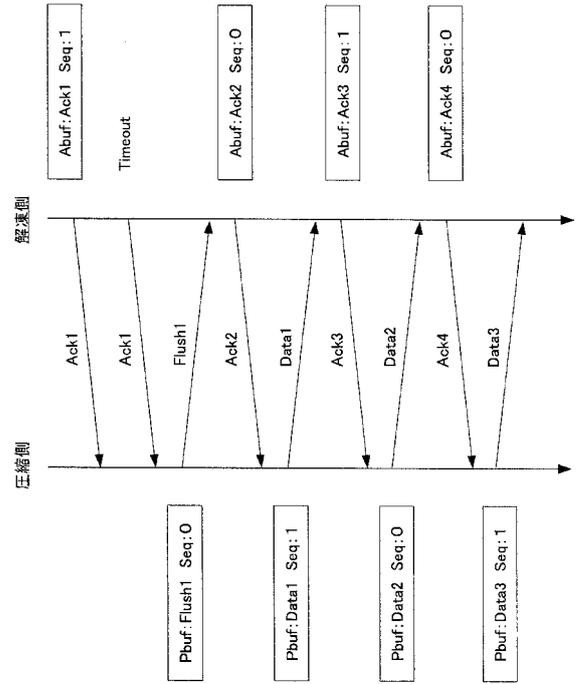
【図 22】



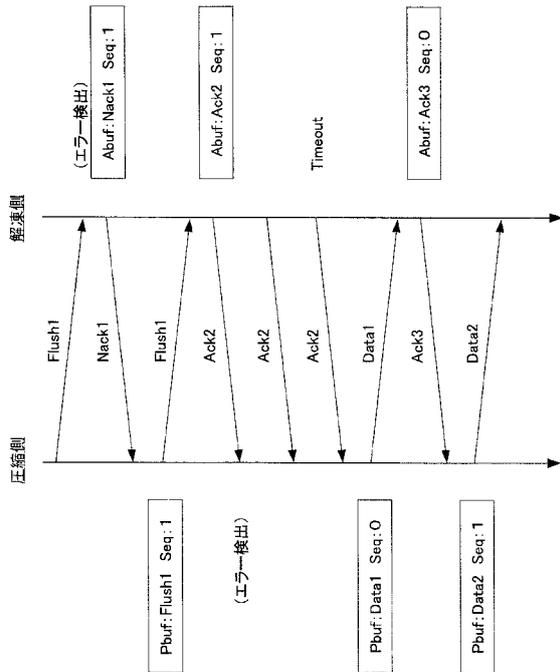
【 図 2 3 】



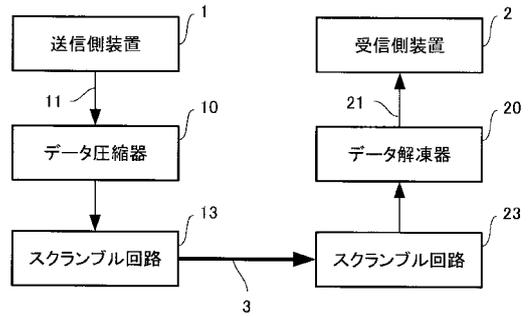
【 図 2 4 】



【 図 2 5 】



【 図 2 6 】



【図 2 7 A】

入力データ: AABBCDDCCBBAACC

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		

出力データ:

【図 2 7 B】

入力データ: AABBCDDCCBBAACC

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1

出力データ: AA

【図 2 7 C】

入力データ: AABBCDDCCBBAACC

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1

出力データ: AABBC

【図 2 8 A】

入力データ: AABBCDDCCBBO

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		

出力データ:

【図 2 8 B】

入力データ: AABBCDDCCBBO

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1

出力データ: AA

【図 2 8 C】

入力データ: AABBCDDCCBBO

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1

出力データ: AABBC

【図 2 7 D】

入力データ: AABBCDDCCBBAACC

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
D	D	1	1

出力データ: AABBCDD

【図 2 7 E】

入力データ: AABBCDDCCBBAACC

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
D	D	1	1

出力データ: AABBCDDCCBB

【図 2 7 F】

入力データ: AABBCDDCCBBAACC

被圧縮シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	2
D	D	1	1

出力データ: AABBCDDCCBBO

【図 2 8 D】

入力データ: AABBCDDCCBBO

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
D	D	1	1

出力データ: AABBCDD

【図 2 8 E】

入力データ: AABBCDDCCBBO

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	1
D	D	1	1

出力データ: AABBCDDCCBB

【図 2 8 F】

入力データ: AABBCDDCCBBO

復号シンボル列		圧縮シンボル	参照頻度
シンボル0	シンボル1		
A	A	0	2
D	D	1	1

出力データ: AABBCDDCCBBA

フロントページの続き

(72)発明者 坂本 比呂志

福岡県北九州市戸畑区仙水町 1 - 1 国立大学法人九州工業大学内

Fターム(参考) 5J064 AA02 AA04 BB13 BC01 BD02