

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号

特許第7366484号
(P7366484)

(45)発行日 令和5年10月23日(2023.10.23)

(24)登録日 令和5年10月13日(2023.10.13)

(51)国際特許分類

F I

G 0 6 N 10/70 (2022.01)

G 0 6 N 10/70

G 0 6 F 11/10 (2006.01)

G 0 6 F 11/10

請求項の数 16 (全45頁)

(21)出願番号	特願2022-514800(P2022-514800)	(73)特許権者	517392436
(86)(22)出願日	令和3年9月9日(2021.9.9)		▲騰▼▲訊▼科技(深▲セン▼)有限公
(65)公表番号	特表2023-516521(P2023-516521 A)		司
(43)公表日	令和5年4月20日(2023.4.20)		中華人民共和國518057▲広▼▲東
(86)国際出願番号	PCT/CN2021/117490		▼省深▲セン▼市南山区高新区科技中一
(87)国際公開番号	WO2022/166199	(74)代理人	路 ▲騰▼▲訊▼大厦35▲層▼
(87)国際公開日	令和4年8月11日(2022.8.11)		100110364
審査請求日	令和4年3月4日(2022.3.4)	(74)代理人	弁理士 実広 信哉
(31)優先権主張番号	202110168322.7		100150197
(32)優先日	令和3年2月7日(2021.2.7)	(72)発明者	弁理士 松尾 直樹
(33)優先権主張国・地域又は機関	中国(CN)		▲鄭▼一▲聰▼
		(72)発明者	中華人民共和國518057▲広▼▲東
			▼省深▲セン▼市南山区高新区科技中一
			路 ▲騰▼▲訊▼大厦35▲層▼
			席 光磊
			最終頁に続く

(54)【発明の名称】 量子誤り訂正復号システム、方法、フォールトトレラント量子誤り訂正システム及びチップ

(57)【特許請求の範囲】

【請求項1】

量子誤り訂正復号システムであって、前記量子誤り訂正復号システムは複数の誤り訂正チップを含み、前記誤り訂正チップは、

量子回路のエラー症状情報を取得することであって、前記エラー症状情報は量子誤り訂正コードのスタビライザージェネレーターの固有値により構成されるデータアレイである、ことと、

ニューラルネットワークデコーダを動作させて前記エラー症状情報に対し復号処理を行い、エラー結果情報を獲得することであって、前記エラー結果情報は前記量子回路におけるエラーが発生する量子ビット及び相応なエラータイプを決定することに用いられる、ことと、に用いられ、

前記ニューラルネットワークデコーダのコア演算は数値量子化処理を経て獲得される符号なし固定小数点数の積和演算であり、

前記量子誤り訂正復号システムは正規化処理を経て獲得され、前記正規化処理は、

前記複数の誤り訂正チップ上で動作する各前記ニューラルネットワークデコーダが同じネットワーク構造を有することと、

前記ニューラルネットワークデコーダの隠れ層が畳み込み層及び全結合層のみを含むことと、

前記ニューラルネットワークデコーダの活性化関数が線形整流ユニットReLU活性化関数を使用することと、

各前記誤り訂正チップ上で1つの前記ニューラルネットワークデコーダを動作させることと、
 前記ニューラルネットワークデコーダが異なる量子回路の誤り訂正復号に適用されることと、
 のうちの少なくとも一つを含む、量子誤り訂正復号システム。

【請求項2】

量子誤り訂正復号システムであって、前記量子誤り訂正復号システムは複数の誤り訂正チップを含み、前記誤り訂正チップは、
 量子回路のエラー症状情報を取得することであって、前記エラー症状情報は量子誤り訂正コードのスタビライザージェネレーターの固有値により構成されるデータアレイである、
 ことと、
 ニューラルネットワークデコーダを動作させて前記エラー症状情報に対し復号処理を行い、エラー結果情報を獲得することであって、前記エラー結果情報は前記量子回路におけるエラーが発生する量子ビット及び相応なエラータイプを決定することに用いられる、ことと、に用いられ、
 前記ニューラルネットワークデコーダのコア演算は数値量子化処理を経て獲得される符号なし固定小数点数の積和演算であり、

前記ニューラルネットワークデコーダは少なくとも1つの隠れ層を含み、前記隠れ層の出力パラメータの計算過程は、

量子化後重み付けベクトル及び量子化後入力ベクトルを取得することであって、前記量子化後重み付けベクトル中に数値量子化処理を経て獲得される符号なし固定小数点数形式の重み付けパラメータを含み、前記量子化後入力ベクトルは数値量子化処理を経て獲得される符号なし固定小数点数形式の入力パラメータを含む、ことと、

前記量子化後重み付けベクトルと前記量子化後入力ベクトルとの内積に基づいて、符号なし固定小数点数形式の出力パラメータを獲得することと、である、請求項1に記載の量子誤り訂正復号システム。

【請求項3】

前記量子化後重み付けベクトルと前記量子化後入力ベクトルとの内積に基づいて、符号なし固定小数点数形式の出力パラメータを獲得することは、

前記誤り訂正チップのオンチップ内部メモリ中から既定パラメータを取得することであって、前記既定パラメータとは取る値が前記入力パラメータの影響を受けないパラメータを指す、ことと、

前記既定パラメータ、及び前記量子化後重み付けベクトルと前記量子化後入力ベクトルとの内積に基づいて、符号なし固定小数点数形式の前記出力パラメータを獲得することと、を含む、請求項2に記載の量子誤り訂正復号システム。

【請求項4】

前記既定パラメータ、及び前記量子化後重み付けベクトルと前記量子化後入力ベクトルとの内積に基づいて、符号なし固定小数点数形式の前記出力パラメータを獲得することは、
 下記【数式1】に従って計算し、符号なし固定小数点数形式の前記出力パラメータ O_q を獲得することを含む、請求項3に記載の量子誤り訂正復号システム。

【数1】

〔数式1〕

$$O_q = \underbrace{\frac{scale_w scale_a}{scale_o}}_{S_1} \left(\underbrace{\sum_i q_w[i] q_a[i]}_A - \underbrace{w_0 \sum_i q_a[i]}_B \right) + \underbrace{\frac{scale_b}{scale_o} (q_b - b_0)}_{S_2}$$

ここで、 q_w は数値量子化処理を経て獲得される符号なし固定小数点数形式の重み付け

パラメータであり、 q_a は数値量子化処理を経て獲得される符号なし固定小数点数形式の入力パラメータであり、 q_b は数値量子化処理を経て獲得される符号なし固定小数点数形式のオフセットパラメータであり、 $scale_w$ は重み付けパラメータに対応するスケーリング係数であり、 $scale_a$ は入力パラメータに対応するスケーリング係数であり、 $scale_o$ は出力パラメータに対応するスケーリング係数であり、 $scale_b$ はオフセットパラメータに対応するスケーリング係数を表し、 w_0 は重み付けパラメータに対応する参照固定小数点数であり、 b_0 はオフセットパラメータに対応する参照固定小数点数であり、 i は正の整数であり、且つ、前記既定パラメータは S_1 、 S_2 及び w_0 を含む。

【請求項5】

前記隠れ層の後、さらに活性化関数層を含み、前記活性化関数層は、前記出力パラメータに対し、

前記出力パラメータの符号桁を取得する処理と、

前記符号桁が前記出力パラメータが正数であることを指示する場合に、前記出力パラメータの各ビット桁の数値を変わないように保留する処理と、

前記符号桁が前記出力パラメータが負数であることを指示する場合に、前記出力パラメータの各ビット桁の数値を零にセットする処理と、を行うことに用いられる、請求項2に記載の量子誤り訂正復号システム。

【請求項6】

前記誤り訂正チップはオンチップ内部メモリと、制御ユニットと、算術計算モジュールと、を含み、

前記オンチップ内部メモリは前記ニューラルネットワークデコーダのパラメータ情報を記憶することに用いられ、

前記制御ユニットは前記算術計算モジュールが前記オンチップ内部メモリ中から前記パラメータ情報を読み取るように制御することに用いられ、

前記算術計算モジュールは前記オンチップ内部メモリ中から前記パラメータ情報を読み取り、前記パラメータ情報に基づいて前記積和演算を実行し、前記積和演算の結果に基づいて、前記ニューラルネットワークデコーダの隠れ層の出力パラメータを計算することに用いられる、請求項1または2に記載の量子誤り訂正復号システム。

【請求項7】

前記算術計算モジュールは、パイプライン方式を採用して、

前記オンチップ内部メモリ中から前記パラメータ情報を読み取るステップと、

前記パラメータ情報に基づいて前記積和演算を実行するステップと、

前記積和演算の結果に基づいて、前記ニューラルネットワークデコーダの隠れ層の出力パラメータを計算するステップと、を実行することに用いられる、請求項6に記載の量子誤り訂正復号システム。

【請求項8】

前記量子誤り訂正コードのスケールは L であり、前記ニューラルネットワークデコーダの数量は $(L^2 - 1) / N + 2$ であり、

1つのニューラルネットワークデコーダは X エラーに対応するエラー症状情報に基づいて、前記 X エラーが発生するか否かを判定することに用いられ、

1つのニューラルネットワークデコーダは Z エラーに対応するエラー症状情報に基づいて、前記 Z エラーが発生するか否かを判断することに用いられ、

前記 X エラーに対応するエラー症状情報は $(L^2 - 1) / 2N$ 個の小グループに画定され、各小グループは N 個のシンδροームビットを含み、各小グループのシンδροームビットは1つのニューラルネットワークデコーダによって復号されて、前記 X エラーが発生する量子ビットの位置情報を獲得し、

前記 Z エラーに対応するエラー症状情報は $(L^2 - 1) / 2N$ 個の小グループに画定され、各小グループは N 個のシンδροームビットを含み、各小グループのシンδροームビットは1つのニューラルネットワークデコーダによって復号されて、前記 Z エラーが発生する量子ビットの位置情報を獲得する、請求項1～7のいずれか一項に記載の量子誤り訂正

10

20

30

40

50

復号システム。

【請求項 9】

フォールトトレラント量子誤り訂正システムであって、前記フォールトトレラント量子誤り訂正システムは、量子回路と、量子処理コントローラ Q C P と、任意波形発生器 A W G と、デジタル採集器 D A Q と、複数の誤り訂正チップを含む量子誤り訂正復号システムと、を含み、

前記 A W G は、エラー症状測定に用いられる測定波形を前記量子回路に送信することに用いられ、

前記 D A Q は前記量子回路を経て増幅された後の測定波形を採集し、採集して得た前記測定波形に対し復調を行って前記量子回路のエラー症状情報を獲得し、前記 Q C P に前記エラー症状情報を送信することに用いられ、前記エラー症状情報は量子誤り訂正コードのスタビライザージェネレーターの固有値により構成されるデータアレイであり、

前記 Q C P は前記エラー症状情報を前記量子誤り訂正復号システムにおける複数の誤り訂正チップに送信することに用いられ、

前記誤り訂正チップは、ニューラルネットワークデコーダを動作させて前記エラー症状情報に対し復号処理を行い、エラー結果情報を獲得することに用いられ、前記エラー結果情報は前記量子回路におけるエラーが発生する量子ビット及び相応なエラータイプを決定することに用いられ、前記ニューラルネットワークデコーダのコア演算は数値量子化処理を経て獲得される符号なし固定小数点数の積和演算である、フォールトトレラント量子誤り訂正システム。

【請求項 10】

前記誤り訂正チップはさらに、前記 Q C P に前記エラー結果情報を送信することに用いられ、

前記 Q C P はさらに、前記エラー結果情報に基づいて、前記 A W G に誤り訂正命令を送信することに用いられ、前記誤り訂正命令中にエラーが発生する量子ビットの位置情報及び相応なエラータイプを含み、

前記 A W G はさらに、前記誤り訂正命令に基づいて、前記量子回路に誤り訂正波形を送信することに用いられ、前記誤り訂正波形は前記量子回路におけるエラーが発生する量子ビットを訂正することに用いられる、請求項 9 に記載のフォールトトレラント量子誤り訂正システム。

【請求項 11】

前記 Q C P と前記 A W G との間の通信、

前記 Q C P と前記 D A Q との間の通信、及び

前記 Q C P と前記誤り訂正チップとの間の通信のうちの、少なくとも 1 グループの通信は、低電圧差動信号 L V D S を採用する、請求項 9 に記載のフォールトトレラント量子誤り訂正システム。

【請求項 12】

複数の誤り訂正チップにより実行される、量子誤り訂正復号方法であって、前記方法は、

量子回路のエラー症状情報を取得するステップであって、前記エラー症状情報は量子誤り訂正コードのスタビライザージェネレーターの固有値により構成されるデータアレイである、ステップと、

ニューラルネットワークデコーダを動作させて前記エラー症状情報に対し復号処理を行い、エラー結果情報を獲得するステップであって、前記エラー結果情報は前記量子回路におけるエラーが発生する量子ビット及び相応なエラータイプを決定することに用いられる、ステップと、

前記ニューラルネットワークデコーダに対して正規化処理を行うステップであって、前記正規化処理は、

前記複数の誤り訂正チップ上で動作する各前記ニューラルネットワークデコーダが同じネットワーク構造を有することと、

前記ニューラルネットワークデコーダの隠れ層が畳み込み層及び全結合層のみを含むこと

10

20

30

40

50

と、

前記ニューラルネットワークデコーダの活性化関数が線形整流ユニット R e L U 活性化関数を使用することと、

各前記誤り訂正チップ上で1つの前記ニューラルネットワークデコーダを動作させることと、

前記ニューラルネットワークデコーダが異なる量子回路の誤り訂正復号に適用されることと、

のうちの少なくとも一つを含む、ステップと

を含み、

前記ニューラルネットワークデコーダのコア演算は数値量子化処理を経て獲得される符号なし固定小数点数の積和演算である、量子誤り訂正復号方法。

【請求項 1 3】

量子誤り訂正復号装置であって、前記装置は、取得ユニットと、復号ユニットと、を含み、

前記取得ユニットは、量子回路のエラー症状情報を取得することに用いられ、前記エラー症状情報は量子誤り訂正コードのスタビライザージェネレーターの固有値により構成されるデータアレイであり、

前記復号ユニットは、ニューラルネットワークデコーダを動作させて前記エラー症状情報に対し復号処理を行い、エラー結果情報を獲得することに用いられ、前記エラー結果情報は前記量子回路におけるエラーが発生する量子ビット及び相応なエラータイプを決定することに用いられ、

前記復号ユニットは、前記ニューラルネットワークデコーダに対して正規化処理を行うことにさらに用いられ、前記正規化処理は、

前記量子誤り訂正復号装置の複数の誤り訂正チップ上で動作する各前記ニューラルネットワークデコーダが同じネットワーク構造を有することと、

前記ニューラルネットワークデコーダの隠れ層が畳み込み層及び全結合層のみを含むことと、

前記ニューラルネットワークデコーダの活性化関数が線形整流ユニット R e L U 活性化関数を使用することと、

各前記誤り訂正チップ上で1つの前記ニューラルネットワークデコーダを動作させることと、

前記ニューラルネットワークデコーダが異なる量子回路の誤り訂正復号に適用されることと、

のうちの少なくとも一つを含む、

前記ニューラルネットワークデコーダのコア演算は数値量子化処理を経て獲得される符号なし固定小数点数の積和演算である、量子誤り訂正復号装置。

【請求項 1 4】

誤り訂正チップであって、前記誤り訂正チップは請求項 1 2 に記載の量子誤り訂正復号方法を実施することに用いられる、誤り訂正チップ。

【請求項 1 5】

非一時的コンピュータ可読記憶媒体であって、前記非一時的コンピュータ可読記憶媒体中に少なくとも1つのコンピュータプログラムが記憶されており、前記少なくとも1つのコンピュータプログラムはプロセッサによりロードされ且つ実行され、コンピュータに請求項 1 2 に記載の量子誤り訂正復号方法を実施させる、非一時的コンピュータ可読記憶媒体。

【請求項 1 6】

コンピュータプログラムであって、前記コンピュータプログラムはコンピュータ命令を含み、前記コンピュータ命令はプロセッサによりロードされ且つ実行され、コンピュータに請求項 1 2 に記載の量子誤り訂正復号方法を実施させる、コンピュータプログラム。

【発明の詳細な説明】

10

20

30

40

50

【技術分野】

【0001】

本願の実施例は量子技術分野に関し、特に量子誤り訂正復号システム、方法、フォールトトレラント量子誤り訂正システム及びチップに関する。

【0002】

本願は、2021年2月7日に提出された出願番号が第202110168322.7号であり、発明の名称が「量子誤り訂正復号システム、方法、フォールトトレラント量子誤り訂正システム及びチップ」である中国特許出願の優先権を要求し、その全部の内容は引用によって本願中に組み込まれている。

【背景技術】

【0003】

量子ビットは非常にノイズの影響を受けやすいため、現状の技術から見て、物理量子ビット上で量子計算を直接実現することはまだ現実的ではない。量子誤り訂正コード及びフォールトトレラント量子計算技術の発展に伴って、原則的には、ノイズがある量子ビットにおいて任意の精度の量子計算を実現することが可能である。

【0004】

もしも単に量子情報を対象として記憶を行うとすれば、すべてのエラー症状を検出して収集し、且つ最後にすべての症状情報に応じて誤り訂正を行うことができる。このような誤り訂正方式は後処理と呼ばれる。しかしながら、フォールトトレラント量子計算を行うときには、量子回路自体はエラータイプをリアルタイムに変更するようになり、単に症状情報に依存するだけでは異なる時間と空間に発生するエラーを正確に追跡して位置特定することができない。量子計算をスムーズに行うことができるようにするために、エラー症状を獲得した後に、直ちに復号を行い、且つ量子アルゴリズムの各々の計算ステップの動作の前（又は次の誤り訂正の開始前）にエラー訂正を完了する必要がある。このような誤り訂正方式はリアルタイム誤り訂正と呼ばれ、リアルタイム誤り訂正は大規模な汎用量子計算を実現するのに不可欠な技術である。

【0005】

リアルタイム誤り訂正は量子誤り訂正復号システムの動作時間マージンに対して、とても高い剛性要求を提示するが、現状のいくつかの量子誤り訂正復号システムの設計は、リアルタイム誤り訂正の要求を満たすことができない。

【発明の概要】

【発明が解決しようとする課題】

【0006】

本願の実施例は量子誤り訂正復号システム、方法、フォールトトレラント量子誤り訂正システム及びチップを提供し、量子誤り訂正復号の所要時間を短縮させることができ、それによりリアルタイム誤り訂正の要求を満たす。前記技術的手段は以下の通りである。

【課題を解決するための手段】

【0007】

本願の実施例の1つの態様によれば、量子誤り訂正復号システムを提供し、前記量子誤り訂正復号システムは複数の誤り訂正チップを含み、前記誤り訂正チップは、

量子回路のエラー症状情報を取得することであって、前記エラー症状情報は量子誤り訂正コードのスタビライザージェネレーターの固有値により構成されるデータアレイである、ことと、

ニューラルネットワークデコーダを動作させて前記エラー症状情報に対し復号処理を行い、エラー結果情報を獲得することであって、前記エラー結果情報は前記量子回路におけるエラーが発生する量子ビット及び相応なエラータイプを決定することに用いられる、ことと、に用いられ、

前記ニューラルネットワークデコーダのコア演算は数値量子化処理を経て獲得される符号なし固定小数点数の積和演算である。

【0008】

本願の実施例の1つの態様によれば、フォールトトレラント量子誤り訂正システムを提供し、前記フォールトトレラント量子誤り訂正システムは、量子回路と、量子処理コントローラQCPと、任意波形発生器AWGと、デジタル採集器DAQと、複数の誤り訂正チップを含む量子誤り訂正復号システムとを含み、

前記AWGは、エラー症状測定に用いられる測定波形を前記量子回路に送信することに用いられ、

前記DAQは前記量子回路を経て増幅された後の測定波形を採集し、採集して得た前記測定波形に対し復調を行って前記量子回路のエラー症状情報を獲得し、前記QCPに前記エラー症状情報を送信することに用いられ、前記エラー症状情報は量子誤り訂正コードのスタビライザージェネレーターの固有値により構成されるデータアレイであり、

前記QCPは前記エラー症状情報を前記量子誤り訂正復号システムにおける複数の誤り訂正チップに送信することに用いられ、

前記誤り訂正チップは、ニューラルネットワークデコーダを動作させて前記エラー症状情報に対し復号処理を行い、エラー結果情報を獲得することに用いられ、前記エラー結果情報は前記量子回路におけるエラーが発生する量子ビット及び相応なエラータイプを決定することに用いられ、前記ニューラルネットワークデコーダのコア演算は数値量子化処理を経て獲得される符号なし固定小数点数の積和演算である。

【0009】

本願の実施例の1つの態様によれば、量子誤り訂正復号方法を提供し、前記方法は、

量子回路のエラー症状情報を取得するステップであって、前記エラー症状情報は量子誤り訂正コードのスタビライザージェネレーターの固有値により構成されるデータアレイである、ステップと、

ニューラルネットワークデコーダを動作させて前記エラー症状情報に対し復号処理を行い、エラー結果情報を獲得するステップであって、前記エラー結果情報は前記量子回路におけるエラーが発生する量子ビット及び相応なエラータイプを決定することに用いられる、ステップと、を含み、

前記ニューラルネットワークデコーダのコア演算は数値量子化処理を経て獲得される符号なし固定小数点数の積和演算である。

【0010】

本願の実施例の1つの態様によれば、量子誤り訂正復号装置を提供し、前記装置は、取得ユニットと、復号ユニットと、を含み、

前記取得ユニットは量子回路のエラー症状情報を取得することに用いられ、前記エラー症状情報は量子誤り訂正コードのスタビライザージェネレーターの固有値により構成されるデータアレイであり、

前記復号ユニットはニューラルネットワークデコーダを動作させて前記エラー症状情報に対し復号処理を行い、エラー結果情報を獲得することに用いられ、前記エラー結果情報は前記量子回路におけるエラーが発生する量子ビット及び相応なエラータイプを決定することに用いられ、

前記ニューラルネットワークデコーダのコア演算は数値量子化処理を経て獲得される符号なし固定小数点数の積和演算である。

【0011】

本願の実施例の1つの態様によれば、誤り訂正チップを提供し、前記誤り訂正チップは前記量子誤り訂正復号方法を実現することに用いられる。

【0012】

本願の実施例の1つの態様によれば、非一時的コンピュータ可読記憶媒体を提供し、前記非一時的コンピュータ可読記憶媒体中に少なくとも1つのコンピュータプログラムが記憶されており、前記少なくとも1つのコンピュータプログラムはプロセッサによりロードされ且つ実行され、コンピュータに前記量子誤り訂正復号方法を実現させる。

【0013】

本願の実施例の1つの態様によれば、コンピュータプログラム製品を提供し、前記コン

10

20

30

40

50

コンピュータプログラム製品はコンピュータプログラム又はコンピュータ命令を含み、前記コンピュータプログラム又は前記コンピュータ命令はプロセッサによりロードされ且つ実行され、コンピュータに前記量子誤り訂正復号方法を実現させる。

【発明の効果】

【0014】

本願の実施例が提供する技術的手段は少なくとも以下の有益な効果を含む。

【0015】

ニューラルネットワークデコーダを採用して量子誤り訂正復号を行うシステムにとって、該ニューラルネットワークデコーダのコア演算は数値量子化処理を経て獲得される符号なし固定小数点数の積和演算であり、それによりニューラルネットワークデコーダの所要のデータ量及び計算量を最大限に減少させて、リアルタイム誤り訂正の要求をより良く満たす。

【図面の簡単な説明】

【0016】

【図1】本願に例示的に示される回転表面コードの模式図である。

【図2】本願に例示的に示される表面コードエラーが発生する模式図である。

【図3】本願に例示的に示される症状測定回路の模式図である。

【図4】本願に例示的に示される三次元シンδροームビットアレイの模式図である。

【図5】本願に例示的に示される表面コードにおいてフォールトトレラント量子計算手段を実現する模式図である。

【図6】本願に例示的に示されるリアルタイムフォールトトレラント誤り訂正システムの構造図である。

【図7】本願に例示的に示される手段のアプリケーションシーンの模式図である。

【図8】図7に示す手段のアプリケーションシーンに関する誤り訂正復号過程の模式図である。

【図9】本願に例示的に示されるリアルタイムフィードバックフォールトトレラント誤り訂正のタイミング図である。

【図10】本願に例示的に示されるマルチニューラルネットワークフォールトトレラント復号フレームワークの模式図である。

【図11】本願に例示的に示されるエラー症状分割の模式図である。

【図12】本願に例示的に示されるRGが触発する浅いニューラルネットワークの模式図である。

【図13】本願に例示的に示される量子誤り訂正復号システムの模式図である。

【図14】本願に例示的に示される異なる症状ブロック組み合わせでの復号性能の模式図である。

【図15】本願に例示的に示されるSNNとMWPMとの性能比較の模式図である。

【図16】本願に例示的に示される2種の量子化方式の模式図である。

【図17】本願に例示的に示される8桁の固定小数点数と32桁の浮動小数点数との精度損失の対比図である。

【図18】本願に例示的に示される畳み込み計算の模式図である。

【図19】本願に例示的に示される誤り訂正チップの体系構造の模式図である。

【図20】本願に例示的に示されるフォールトトレラント量子誤り訂正システムの模式図である。

【図21】本願に例示的に示される量子誤り訂正復号過程の模式図である。

【図22】本願に例示的に示される量子誤り訂正復号装置の模式図である。

【発明を実施するための形態】

【0017】

本願の目的、技術的手段及び利点をより明確にするために、以下、図面と併せて本願の実施形態を更に詳細に記述する。

【0018】

本願の実施例に対して紹介及び説明を行う前に、まず、本願中に関するいくつかの名詞に対し解釈説明を行う。

【0019】

1：量子計算（Quantum Computation、QC）：量子状態の重ね合わせ及びもつれ性質を利用して特定の計算タスクを迅速に完了する方式である。

【0020】

2：量子誤り訂正コード（Quantum Error Correction、QEC）：量子状態を多体量子システムのヒルベルト空間における1つの部分空間中にマッピングして符号化する方式である。量子ノイズは符号化量子状態をその他の部分空間に移す。量子状態が所在する空間を連続的に観測する（症状抽出）ことによって、量子ノイズを評価して訂正すると同時に、符号化量子状態を干渉しないことを可能にし、それにより符号化量子状態を量子ノイズの干渉を受けないように保護する。具体的に言えば、1つの $[[n, k, d]]$ 量子誤り訂正コードは n （ n は正の整数である）個の物理量子ビットにおいて k （ k は正の整数である）個の論理量子ビットを符号化すると、任意のシングル量子ビット上に発生した任意の「数1」個のエラーを訂正するために用いられることを表す。

【0021】

【数1】

$$\lfloor (d-1)/2 \rfloor \quad (d \text{ は正の整数である})$$

【0022】

3：データ量子状態：量子計算時に量子情報を格納するために用いられるデータ量子ビットの量子状態である。

【0023】

4：スタビライザージェネレーター（stabilizer generator）：パリティチェック演算子とも呼ばれる。量子ノイズ（エラー）の発生は、あるいくつかのスタビライザージェネレーターの固有値を変更し得るため、それによりこれらの情報に応じて量子誤り訂正を行うことができる。

【0024】

5：エラー症状（error syndrome）：エラーがないときに、スタビライザージェネレーターの固有値は0であり、量子ノイズが発生するときに、あるいくつかの誤り訂正コードのスタビライザージェネレーター（パリティチェック演算子）の固有値は1に変わり得る。これらの0、1シンドロームビットから構成されるビットストリングはエラー症状と呼ばれる。

【0025】

6：症状測定回路：チェック回路とも呼ばれ、エラー症状を取得する量子回路である。一般的な場合に、該回路自体はノイズに汚染されるようになる。

【0026】

7：トポロジー量子誤り訂正コード（topological quantum code）：量子誤り訂正コードにおける一種の特殊なカテゴリである。この種類の誤り訂正コードの量子ビットは二次元以上の格子アレイ上に分布している。格子は1つの高次元多様体の離散構造（結晶格子多様体）を構成する。この時、誤り訂正コードのスタビライザージェネレーターは幾何学的に近く且つ有限な量子ビット上に定義される。従って幾何学的には近傍領域（比較的小さい範囲内に局限される）であり、且つ測定が容易である。この種類の誤り訂正コードの論理演算子が作用する量子ビットは格子点アレイの多様体において一種類のトポロジー的に非自明な幾何学的オブジェクトを構成する。

【0027】

8：表面コード（surface code）：表面コードは二次元結晶格子多様体において定義される一種類のトポロジー量子誤り訂正コードである。そのスタビライザージェネレーターは通常、4つの量子ビットによりサポートされ（境界箇所では2つの量子ビ

ットによりサポートされ)、論理演算子はアレイを跨ぐ帯状の非自明なチェーンである。表面コードの具体的な二次元構造(5×5、合計で49個のデータ量子ビット及び48個の補助量子ビットであり、任意の2つの量子ビット上に発生するエラーを訂正できる)は図1に示される。黒色ドット11は量子計算をするために用いられるデータ量子ビットを表し、十字12は補助量子ビットを表す。補助量子ビットは初期では $|0\rangle$ 又は $|+\rangle$ 状態で準備される。斜線で充填される正方形(又は半円形)、及び白色で充填される正方形(又は半円形)は2種の異なるタイプのスタビライザージェネレーターを表し、それぞれZエラー及びXエラーを検出することに用いられる。本願においては、図1に示す回転表面コード(rotated surface code)を使用することとなり、約半分の物理量子ビットを節約し、近頃の実験においてより便利に認証することができるからである。このような表面コードのパリティチェック演算子は「体内」では4つのデータ量子ビットに接続され、境界箇所では2つのデータ量子ビットに接続される。

10

【0028】

9：表面コードスケールL：表面コードアレイの周囲長の4分の1である。図1における表面コードアレイL=7は、任意の3つの量子ビット上に発生するエラーを訂正できることを表す。

【0029】

10：ホモロジークラス(homology class)：トポロジーでは、ホモロジー理論におけるホモロジークラスは境界が0である幾何学的サブオブジェクトの有限線形組み合わせで表される。もしこのような線形組み合わせの幾何学的オブジェクトは次元が1よりも大きい幾何学的オブジェクトの境界としてみなすことができるなら、それは「0」(ここでの0とは、トポロジーの意味においての平凡クラス一点に連続的に収縮できる幾何学的オブジェクトを指す)とホモロジーである。下記においては、「エラークラス」と混用することがある。

20

【0030】

11：Xエラー及びZエラー：物理量子ビットの量子状態において生じたランダムに生じたパウリX及びパウリZ進化エラーである。量子誤り訂正理論に応じて、もし誤り訂正コードがXエラー及びZエラーを訂正できるならば、シングル量子ビット上に発生する任意のエラーを訂正できる。

【0031】

30

12：フォールトトレラント量子誤り訂正(Fault Tolerant Quantum Error Correction、FTQEC)：真の量子計算におけるすべての操作過程は、量子ゲート及び量子測定を含んでいずれもノイズを帯びている。つまり、たとえ量子誤り訂正をするために用いられる回路自体であってもノイズを含有する。フォールトトレラント量子誤り訂正とは、誤り訂正回路を精巧に設計することによって、ノイズを帯びている誤り訂正回路を使用して誤り訂正を行うことができ、且つ依然としてエラーを訂正して時間に伴うエラーの拡散を阻止するという目的を達成できることを指す。

【0032】

13：フォールトトレラント量子計算(Fault Tolerant Quantum Computation、FTQC)：すなわち、量子誤り訂正保護を有する量子計算である。量子計算の過程においては、量子誤り訂正回路自体及び量子ビット測定を含む、任意の物理的操作はいずれもノイズを帯びている。もし古典的計算及び操作(たとえば、命令入力、及び誤り訂正コード復号等)がノイズを含有せず、且つ迅速に実行と仮定すれば、フォールトトレラント量子計算は、量子誤り訂正手段を合理的に設計し、且つ符号化した論理量子状態に対して特定方式の量子ゲート操作を行うことによって、ノイズ付き量子ビットを使用して量子計算を行う過程において、エラーを効果的に制御し訂正できることを確実にする技術的手段である。

40

【0033】

14：リアルタイムフィードバック誤り訂正(Real time feedback QEC)：いくつかの症状測定を行ってシンδροームビットを獲得した後に、低遅延高速

50

復号を行ってエラーの具体的な発生位置を獲得し、且つ次回の症状測定を行う前にエラーが発生するデータ量子ビットに対し物理的にエラー訂正操作を直接行う。

【0034】

15：物理量子ビット：実空間における物理デバイスを使用して実現される量子ビットであり、データ量子ビット及び補助量子ビットを含む。

【0035】

16：論理量子ビット：誤り訂正コードで定義されるヒルベルト部分空間における数学自由度である。論理量子ビットの量子状態の記述は通常、多体もつれ状態であり、一般的には、複数の物理量子ビットとヒルベルト空間とを組み合わせた二次元部分空間である。フォールトトレラント量子計算は誤り訂正コードによる保護を受ける論理量子ビットにおいて動作させる必要がある。

10

【0036】

17：量子ゲート／回路：物理量子ビット上に作用する量子ゲート／回路である。

【0037】

18：データエラー：データ量子ビット上に発生するエラーである。

【0038】

19：測定エラー：測定過程が完璧ではないことによるエラーである。

【0039】

20：閾値定理 (threshold theorem)：フォールトトレラント量子計算の要求に合致する量子計算手段に対し、すべての操作のエラー率がある1つの閾値よりも小さいときに、より良い誤り訂正コード、より多くの量子ビット、より多くの量子操作を使用することによって、計算の正確率を任意に1に近づけることができると同時に、これらの追加のリソースオーバーヘッドは量子計算の指数／多項式加速に対しては無視できる、という定理である。

20

【0040】

21：ニューラルネットワーク：人工ニューラルネットワークは大量の簡単な基本素子－ニューロンを相互に締結してなる自己適応非線形動的システムである。各ニューロンの構造及び機能は比較的簡単であるが、大量のニューロンの組み合わせで生じたシステム行為は非常に複雑であり、原則的には、任意の関数を表現できる。

【0041】

22：畳み込みニューラルネットワーク (Convolutional Neural Network、CNN)：畳み込みニューラルネットワークは畳み込み計算を含み且つ深さ構造を有する一種のフィードフォワードニューラルネットワークである。畳み込み層 (convolutional layer) は畳み込みニューラルネットワークのコア基盤であり、すなわち、離散二次元又は三次元フィルター (畳み込みカーネルとも呼び、それぞれ二次元又は三次元行列である) が二次元又は三次元データドットマトリックスと畳み込み操作を行う。

30

【0042】

23：全結合層 (fully connected layer)：全接続層、全結合ネットワーク又は全接続ネットワークとも呼ばれ、全結合層の各々の出力節点はいずれも前の層のすべての入力節点に連結され、その前に抽出した特徴を統合するために用いられる。

40

【0043】

24：線形整流層 (Rectified Linear Units layer、ReLU layer)：線形整流 (Rectified Linear Units、ReLU) $f(x) = \max(0, x)$ をニューラルネットワークの励起関数として使用する。

【0044】

25：誤差逆伝播アルゴリズム (BP (Back Propagation))：人工ニューラルネットワークにおける一種の教師あり学習アルゴリズムである。BPニューラルネットワークアルゴリズムは理論上では任意の関数に近づけることができ、基本構造は非線形変化ユニットからなり、とても高い非線形マッピング能力を有する。

50

【0045】

26: **FPGA: Field Programmable Gate Array**、フィールドプログラマブルゲートアレイチップである。

【0046】

27: **パイプライン (pipeline) 技術**: 演算タスクを行うときに、異なる部分に属する演算を重ねて操作を行う一種の準並行処理実現技術を指す。

【0047】

28: **ニューラルネットワークモデル量子化 (quantization)**: 量子化は、すなわち、各重み付けを表すための所要のビット数を減少させることによってオリジナルネットワークを圧縮する。ニューラルネットワークモデルは一般的には、いずれもとても大きな記憶空間を占有するようになる。もしハードウェアチップにおいてニューラルネットワークアルゴリズムを実現しようとするならば、これは大量の貴重なオンチップ記憶空間、チップ内レジスタ及び配線を占め、且つ計算速度に深刻な影響を与えることとなる。これらのパラメータは浮動小数点数タイプであるため、普通の可逆圧縮アルゴリズムはこのような場合を効果的に処理できない。もしモデルの正確率に影響を与えない場合にモデルの内部でその他の簡単な数値タイプ（たとえば、固定小数点演算）を採用して計算を行うことができるならば、消費されるハードウェア計算リソース（ハードウェア演算ユニット及び記憶ユニットを含む）は大幅に減少するようになる。リアルタイムフィードバックシステムの復号チップにとっては、量子化アルゴリズムは単位時間内の計算量を極めて大幅に減少させ、それにより遅延を減少させるようになる。

【0048】

29: **積和演算 (Multiply Accumulate, MAC 又は MA)**: デジタル信号プロセッサ又はいくつかのマイクロプロセッサにおける特殊な演算である。この演算操作を実現するハードウェア回路ユニットは、「乗累算器」と呼ばれる。このような演算の操作は、乗算の積の結果と累算器の値とを足し算してから累算器に保存することである。

【0049】

30: **LVDS (Low-Voltage Differential Signaling、低電圧差動信号)**: 一種の低電圧差動信号の電気標準であり、差動信号を使用してデータをシリアル伝送する一種の方法である。物理的な実現形態としては、通常、ツイストペア線を使用し、相対的に比較的低い電圧下で高速伝送を行うことができる。LVDSは一種の物理層の規格に過ぎず、プロトコル層の通信方法等に関するものではなく、ゆえに、とても低いディレイを有する。

【0050】

31: **AWG (Arbitrary Waveform Generator、任意波形発生器)**: デジタル信号を用いて配置される波形情報を受信し、デジタルーアナログ変換を経た後に、対応するアナログ波形を生じさせる。量子測定制御システムにおいては量子ビットを制御し及び読み取る波形を生じさせることに用いられる。

【0051】

32: **DAQ (Digital Acquisition、デジタル採集器)**: アナログ信号入力を受信し、アナログーデジタル変換を経た後に、対応するデジタル信号を獲得し、且つ次の各種の処理に用いられる。量子測定制御システムにおいては、量子チップの読み取り結果を受信することに用いられる。

【0052】

33: **QCP (Quantum Control Processor、量子制御プロセッサ)**: 量子命令を実行するモジュールであり、実験過程においてはDAQから結果を受信し、且つAWGを制御して量子操作に対応する波形を送信させる。量子測定制御システムにおいては、量子制御プロセッサは量子プログラムが生成する命令をリアルタイムに処理することによって、それによりプログラムフロー制御及びフィードバック制御を行う。

【0053】

34：量子測定制御システム（Quantum Control & Measurement System）：量子命令セットを動作させることによって量子プログラムの実行を実現し、且つ量子チップに入力信号を提供して制御を行い、及び量子チップに対し測定を行い、且つ結果を採集するシステムである。超伝導量子計算においては、1セットの量子測定制御システムは通常、以上で言及されたAWG、DAQ及びQCPを含む。

【0054】

以下、本願の技術的手段の出発点に対して紹介及び説明を行う。リアルタイムフィードバックフォールトトレラント量子誤り訂正（又は「リアルタイムフィードバックフォールトトレラント誤り訂正」と呼ばれる）である。

【0055】

量子ビットは非常にノイズの影響を受けやすい。従って、物理量子ビットにおいて量子計算を直接実現することは、現状の技術から見て、まだ現実的ではない。量子誤り訂正コード及びフォールトトレラント量子計算技術の発展は、原則的には、ノイズがある量子ビットにおいて、任意の精度の量子計算を実現する可能性を提供する。一般的に言えば、量子誤り訂正コードのスタビライザージェネレーターに対し測定（症状測定、又はパリティチェックとも呼ばれる）を行うことは、長距離量子ゲートを導入することを要し、且つ同時に追加の量子ビットによって複雑な量子補助状態を準備してフォールトトレラント誤り訂正を完了することを要求する。現状の実験手法の制限のために、現状では、高精度の長距離量子ゲートを実現する能力が未だなく、複雑な量子補助状態を準備する能力もない。一方、トポロジ量子誤り訂正コードを使用してフォールトトレラント量子誤り訂正及びフォールトトレラント量子計算をする手段は、長距離量子ゲート（スタビライザージェネレーターは空間において局所的に近傍するため）を使用し、複雑な量子補助状態（通常、シングルの量子ビットの標準状態のみを準備する必要がある）を準備する必要がなく、且つフォールトトレラント閾値は比較的高い。従って、現在の実験技術を利用して汎用フォールトトレラント量子計算を実現する可能性が比較的高い手段であると考えられる。本願は、表面コードという特殊なトポロジ量子誤り訂正コードを例として説明を行うが、コアの思想は任意のトポロジ量子誤り訂正コードに適用される。

【0056】

1つの誤り訂正コードとして、エラーが発生した後に、パリティチェックを行うことによってエラー症状を獲得することができる。その後、これらの症状に応じて、さらに誤り訂正コードに対する具体的な復号アルゴリズムによってエラーの発生位置及びタイプ（Xエラーであるか、Zエラーであるか、それとも両方があるか、すなわちYエラーであるか）を判断する必要がある。表面コード（又は一般的なトポロジ誤り訂正コード）にとっては、エラー及びエラー症状は具体的な空間的位置を有し、症状を引き起こすエラーがあるときに、対応する位置の補助量子ビットの固有値は1であり（該位置に1つの点粒子が出現したとみなすことができる）、エラーがないときに、対応する位置の補助量子ビットの固有値は0である。この時、復号は以下の問題にまとめられる。1つの空間デジタルアレイ（2次元又は3次元、数値は0又は1である）を与え、特定のエラー発生モデル（error model）—量子ビット上に発生するエラー確率分布に応じて、エラーが発生する可能性が最も大きいのがどの量子ビットであるか、及び具体的なエラータイプを推定し、そしてこの推定結果に応じてエラー訂正を行う。

【0057】

図2に示すように、表面コードエラーが発生する模式図を示す。量子ビットは二次元アレイの辺上にあり、エラー症状を測定する補助量子ビットは二次元アレイのノード上にある（これらの症状は完璧な測定である）。図2における黒色辺21はエラーが発生する量子ビットから形成されるエラーチェーンを表し、斜線で充填される丸部分22はエラーにより誘発される症状値が1である点を表す。点状症状によってチェーン状エラーを決定できれば復号を完了できる。

【0058】

もし症状測定過程自体が完璧でノイズ無しであるならば、直接これらの症状情報に応じ

10

20

30

40

50

て復号アルゴリズムを動作させ、エラーが具体的にどのデータ量子ビット上に発生するかを判定することができる。しかし、実際の状況においては、量子回路（症状測定回路を含む）はいずれもノイズを含有するものである。

【0059】

図3に示すように、Z及びXの2種類のスタビライザージェネレーターに対し症状測定を行う回路を例示的に示す。ここで、N、E、W、及びSは補助量子ビットの北、東、西、及び南にあるデータ量子ビットを表す。該順序は非常に重要視され、さもなければ復号性能を深刻に低減させるようになる。

【0060】

ゆえに、1回の症状測定だけによって獲得される症状情報は直接復号に用いることができない。それは、却って関連ノイズを導入するようになり、多数のデータ量子ビットにエラーが発生することを引き起こし、論理量子ビットを迅速に破壊するためである。通常、スケールがLである表面コードを対象として、一区切りの時間内にO(L)回の測定をして十分な症状情報（三次元症状履歴図としてみなすことができる。図4に示すように、ここで垂直な第3次元は時間を表す）を作成する必要がある、このようにしてこそ、これらの情報に応じて、この区切りの時間内に、どれらの症状が問題有りであるか、どれらが確実であるかを判定し、且つこの区切りの時間O(L)内にどれらのデータ量子ビット及び補助量子ビットにエラーが発生したかを判定し、且つデータ量子ビット上のエラーに対し訂正を行うことができる。この複数回の、症状測定－物理誤り訂正の過程は量子記憶過程が終わるまで絶えず繰り返す必要がある、この全過程はリアルタイムフォールトトレラント量子記憶と呼ばれる。理論的な結果から証明されるように、すべての量子ビット上に発生する様々なノイズ（記憶、準備、量子ゲート、及び測定等）はいずれも相互に独立するものであり、且つ強度がある1つの閾値よりも小さい場合に、Lを増加させることによってノイズの影響を任意に小さく減少させることができる。これはフォールトトレラント閾値定理である。閾値定理は任意の規模の汎用量子計算を実現できることを確実にする理論基盤である。

【0061】

図4は、複数回の症状測定が獲得する三次元シンδροームビットアレイの模式図を例示的に示す。図4においては、合計4つのセグメント41を含み、各々のセグメント41は1回の測定で取得するエラー症状情報を表し、浅い色の丸はデータ量子ビットを表し、濃い色の十字は補助量子ビットを表し、線42はZエラーが誘発する症状を表し、線43はXエラーが誘発する症状を表し、線44は測定エラーを表し、X及びZ等はデータ量子ビット上に等価発生するエラーを表す。

【0062】

リアルタイムフォールトトレラント量子記憶の実現に基づいて、更にフォールトトレラント量子計算を実現することができる。図5に大まかに示すように、ここでは表面コードを利用する1つの比較的古典的な計算手段を紹介する。まず、量子計算を実現するためには、1つの論理量子ビットでは十分ではなく、より多くの論理量子ビットを符号化する必要がある。このような符号化方式は表面において1つの結合領域のスタビライザージェネレーターに対する測定をキャンセルして1つのホールを形成することによって、2つの保護される自由度を実現し、1つの論理量子ビットを形成できる。ここでは、「自由度」の意味は、事前定義されていない変数であると考えられてもよい。ここではスタビライザージェネレーターの固有値を固定する必要がない。従って、このジェネレーターの固有値は0であってもよく1であってもよく、つまり2つの値の自由度である。フォールトトレラント操作の便宜上、通常、大きさが等しい2つのホールを選択し、4つの論理自由度を形成し、且つそれから2つの自由度を選んで1つの論理量子ビットを実現し、図5における線で結ばれる2つのホールのように、1つの論理量子ビット51を形成する。注意される点として、ホールの周囲長及び2つのホールを接続する直線の長さはこの論理量子ビット51に対する表面コードの保護能力を決める。フォールトトレラント量子計算を行うときに、このホールはシングル量子ビットとスタビライザージェネレーターとを交互に測定す

10

20

30

40

50

ることによって平面 (surface) における移動及びエンタングルメントを実現する必要がある。且つ、この過程においては中断せずに症状測定を行い、データ量子ビット上に発生するエラーを直ちに訂正する (リアルタイムフィードバック誤り訂正)。

【0063】

問題を簡略化するために、本願においては、量子内部メモリに対しリアルタイムフィードバック誤り訂正を実現することのみを考慮する。これは汎用フォールトトレラント量子計算を実現する肝心な一歩であり、且つ両者間はリアルタイムフィードバック制御の角度から実質的な相違点がない。本願においては、注意力は量子チップを制御する電子学／マイクロ波システムに集中し、且つ量子チップ (又はその他の量子計算物理本体) 自体の製造プロセスが既に十分に成熟していると仮定する。そのため準備される量子ビットが耐えるノイズは空間独立の仮定を満たし、且つ様々なノイズの強度はいずれも表面コードの閾値 (約<1%のある1つの値) よりも小さい。

【0064】

現状では、表面コード又はその他のいかなるトポロジ誤り訂正コードに対しリアルタイムフォールトトレラント誤り訂正を行うシステムは未だ存在していない。大まかに言えば、量子計算を実現する物理プラットフォームの種類 (超伝導、イオン井戸、中性冷原子、及び線形光学システム等を含む) にかかわらず、1つのフォールトトレラント量子誤り訂正 (FTQEC) システムの実現フレームワークは図6に示され、リアルタイムフィードバック誤り訂正過程は図6に基づいて以下のいくつかのステップに分けられる。

【0065】

ステップ1: 症状測定モジュール (syndrome measurement module) は症状測定を行う命令を発生し命令実行ユニット (Instruction Execution Unit、IEU) を経て任意波形発生器 (AWG) に到達させ、図3に示す症状測定回路に必要な波形シーケンスを生じさせ、量子チップを利用して症状測定を完了する。

【0066】

ステップ2: 獲得した測定信号を、DAQを経過させた後に、データ点に対して0及び1の状態区別を行い、シンドロームビット (syndrome bits) を生じさせ、且つ一時記憶する。

【0067】

ステップ3: 上記ステップ1及び2をO(L)回繰り返した後に、すべての症状情報を集め、すべてのシンドロームビットを復号モジュール (decoding module) に同時に伝送して復号を行う。

【0068】

ステップ4: 復号モジュールはエラーが発生する量子ビットの位置座標及びエラータイプを生じさせ、これらの情報をAWGに伝送する。

【0069】

ステップ5: AWGはエラーの発生位置及びタイプに応じてエラー訂正に必要な操作 (一般的にはX又はZゲートである) の波形 (光信号又はマイクロ波信号であり、底層量子ビットの物理的な実現形態により決まる) を生じさせ、相応な量子ビット上に作用して誤り訂正を完了する。

【0070】

汎用フォールトトレラント量子計算を行うことができる1つの誤り訂正システムは以下の2つの条件を満たす必要がある。

【0071】

条件1: 拡張可能性。フォールトトレラント量子計算に最終的に必要な物理量子ビットは数千万であるため、誤り訂正を行うことに用いられる電子学制御システムに消費されるリソースは論理量子ビットの点数の増大に伴って急激に増大しないよう確実にする必要がある。

【0072】

10

20

30

40

50

条件2：低ディレイ性。誤り訂正過程では、O（L）回の症状測定の収集が終了してから誤り訂正が完了するまでの間に症状測定を行うことができないため、量子ビットはノイズ下にさらされ、後続の誤り訂正効果に影響を与えるようになる。従って、リアルタイムフィードバック誤り訂正は非常に低いディレイを要求する。量子ビットの数量の増加に伴って、原則的には、復号の遅延増大はゆっくりであることを要する。

【0073】

上記条件1について、現在の実験は未だ数千万の物理量子ビットからとても遠く離れている。従って、未だ現状のすべての制御学システムの考慮範囲内にはない。

【0074】

上記条件2について、リアルタイム誤り訂正は上記ステップ3～5の過程において症状測定を行わないことを要する。つまり、この区切りの時間に、量子状態はノイズ環境中にさらされ、且つエラーに関連する情報を抽出できず、この時間が長いほど、誤り訂正効果への影響も大きい。従って、ステップ3～5の実行時間はできるだけ短い必要がある。具体的に分析すれば、ステップ3～5の実行時間は以下の1～5を含む。

【0075】

1：症状量子ビットの測定を完了した後に、信号を誤り訂正システムに伝送する時間、
2：DAQと0及び1との区別処理をし、且つシンドロームビットを復号モジュールに伝送する時間、

3：復号アルゴリズムが動作する時間、

4：復号結果をAWGに伝送する時間、及び

5：AWGは制御波形を生じさせ且つチップを伝送する時間。

【0076】

図7に参照されるように、本願の1つの実施例が提供する手段のアプリケーションシーンの模式図を示す。図7に示すように、該アプリケーションシーンは超伝導量子計算プラットフォームであってもよく、該アプリケーションシーンは、量子回路71、希釈冷凍機72、制御機器73及びコンピュータ74を含む。

【0077】

量子回路71は物理量子ビットに作用する一種の回路であり、量子回路71は量子チップ、例えば、絶対零度付近にある超伝導量子チップとして実現できる。希釈冷凍機72は超伝導量子チップに絶対零度の環境を提供することに用いられる。

【0078】

制御機器73は量子回路71に対し制御を行うことに用いられ、コンピュータ74は制御機器73に対し制御を行うことに用いられる。たとえば、編集済みの量子プログラムはコンピュータ74におけるソフトウェアを経て命令にコンパイルされ制御機器73（例えば、電子／マイクロ波制御システム）に送信され、制御機器73は上記命令を電子／マイクロ波制御信号に変換して希釈冷凍機72に入力し、10mKよりも小さい温度にある超伝導量子ビットを制御する。読み取りの過程はそれとは反対であり、読み取った波形は量子回路71に伝えられる。

【0079】

図8に示すように、本願の実施例はリアルタイムフィードバックフォールトトレラント誤り訂正を実現できる復号アルゴリズムを提供し、該復号アルゴリズムを制御機器73と組合せられる（例えば、該復号アルゴリズムを電子学／マイクロ波制御システム中に集積する）。制御機器73の総制御システム73a（例えば、中央ボードFPGA（Field Programmable Gate Array、フィールドプログラマブルゲートアレイ））が量子回路71からエラー症状情報を読み取った後に、総制御システム73aは、誤り訂正命令を制御機器73のリアルタイムフィードバックフォールトトレラント誤り訂正モジュール73bに発する。該誤り訂正命令中には上記量子回路71のエラー症状情報を含む。リアルタイムフィードバックフォールトトレラント誤り訂正モジュール73bはFPGA又はASIC（Application Specific Integrated Circuit、専用集積回路）チップであってもよい。リアルタイムフィード

10

20

30

40

50

バックフォールトトレラント誤り訂正モジュール 73b は上記復号アルゴリズムを動作させ、エラー症状情報に対し復号を行い、且つ復号して獲得したエラー結果情報を誤り訂正制御信号にリアルタイムに変換し、量子回路 71 に送信してエラー訂正を行い、それにより量子ビットに対し保護を行う。

【0080】

上記では、リアルタイムフィードバックフォールトトレラント誤り訂正の背景を紹介し、且つ図 7 及び図 8 と併せて制御システム全体を特徴付けした。フォールトトレラント量子計算を実現するには、まず量子内部メモリに対するリアルタイムフィードバック誤り訂正を実現する必要がある、本願の技術的手段の焦点でもある。ここでは、本願のシステム全体の実現の細部を展開して詳細に紹介する。

【0081】

図 7 及び図 8 は、量子内部メモリのリアルタイムフィードバックフォールトトレラント誤り訂正のシステムフレームワーク図を与えている。もしタイミングから出発するならば、誤り訂正タイミング全体は図 9 に示されており、スケール L を与えた後に、具体的な症状測定周期 T を選定する必要がある。図 9 においては、SM は症状測定を表し、星状符号は発生するある 1 つのエラー（記憶又は測定エラー）を表す。図 9 においては、 $T = 3$ 及び $T = 4$ を例として選ぶ。

【0082】

復号アルゴリズムの性能を研究するために、シミュレーションを行う必要がある。シミュレーションを行って復号過程をシミュレートするときに、毎回復号 (FTQEC) を完了した後に、いずれも 1 回の完璧な症状測定の理想的な復号 (ideal decoding、ID) を行うことでこの時間点に論理エラーが既に発生して誤り訂正の失敗を及ぼすかを判定することができる。論理エラーが発生したと判定するときに、合計経過時間を記録する。複数回のシミュレーションによって、この時間に対し平均値を求める。すなわち、論理ビットの寿命である。寿命に対し逆数を取ると、すなわち、復号エラー率である。MWPM (Minimum Weight Perfect Matching、最小重み完全マッチング) デコーダを使用する実際の観察に応じて、復号周期 T が大きいほど、誤り訂正効果が良好である。しかし、T が L の 2 倍を超えた後には、明らかな性能の向上はなくなり、論理エラー率はある 1 つの限度値に近い。それと同時に、復号の複雑度は迅速に増加するようになる（シンδροームビットの数が過度に蓄積する）。従って、 $T \leq 2L$ を選ぶ。

【0083】

FTQEC モジュールにおいて、浅いニューラルネットワーク復号 (Shallow Neural Network Decoder、SNND) アルゴリズムを使用してトロジー量子誤り訂正コードに対する復号を実現する。ここでは、まず浅いニューラルネットワーク復号アルゴリズムに対して紹介及び説明を行う。

【0084】

シンδροームビット自体がノイズに汚染されるため、複数のニューラルネットワークデコーダを使用して論理エラークラス及び具体的なシンδροームビット値をそれぞれ推定しなければならない。ここでは、量子誤り訂正コード（例えば、表面コード）のスケールが L であることを例とすると、更に入力したエラー症状情報を X クラス及び Z クラスの 2 つの大グループに大別でき、各大グループに含有されるシンδροームビットの数は総シンδροームビットの数の半分、すなわち、 $(L^2 - 1) / 2$ 個のビットであり、それぞれ X 及び Z エラーに対し誤り訂正を行うことに用いられる。X 大グループを例とすると、更に推定される必要があるシンδροームビットを $(L^2 - 1) / 2N$ 個の小グループに構成してもよく、各小グループは N (N は正の整数である) 個のシンδροームビットを含有し、各小グループのシンδροームビットは 1 つのニューラルネットワークを使用して推定を行い、すなわち、1 つの 2^N 分類問題に対応する。このように、X (又は Z) 大グループは X (又は Z) 論理エラーに対し分類推定を行う 1 つのニューラルネットワークが必要であり、残りの $(L^2 - 1) / 2N$ 個のニューラルネットワークは $(L^2 - 1) / 2N$ 個のシン

10

20

30

40

50

ドロームビット値に対し推定を行い、合計で $(L^2 - 1) / N + 2$ 個のニューラルネットワークが必要である。

【0085】

図10に示すように、マルチニューラルネットワークのフォールトトレラント復号フレームワークを例示的に示す。量子誤り訂正コードのスケールは L であり、ニューラルネットワークデコーダの数量は $(L^2 - 1) / N + 2$ である。ここで、1つのニューラルネットワークデコーダは X エラーに対応するエラー症状情報に基づいて、 X エラーが発生するか否かを判定することに用いられる。1つのニューラルネットワークデコーダは Z エラーに対応するエラー症状情報に基づいて、 Z エラーが発生するか否かを判定することに用いられる。 X エラーに対応するエラー症状情報は $(L^2 - 1) / 2N$ 個の小グループに画定され、各小グループは N 個のシンドロームビットを含み、各小グループのシンドロームビットは1つのニューラルネットワークデコーダによって復号されて、 X エラーが発生する量子ビットの位置情報を獲得する。 Z エラーに対応するエラー症状情報は $(L^2 - 1) / 2N$ 個の小グループに画定され、各小グループは N 個のシンドロームビットを含み、各小グループのシンドロームビットは1つのニューラルネットワークデコーダによって復号されて、 Z エラーが発生する量子ビットの位置情報を獲得する。即ち、2つのニューラルネットワークデコーダはエラータイプを推定し、残りのニューラルネットワークデコーダはエラーが発生する量子ビットの位置情報を推定し、この2つの部分の情報を統合すれば、具体的に物理量子ビット上に発生する全部のエラーを推断できる。

【0086】

通常、ニューラルネットワークを直接選んで訓練を行うが、 L が比較的大きい場合に、効果は非常に理想的ではない。拡張可能な手段を構築するために、RG (Renormalization Group、繰り込み群) デコーダの思想に応じてニューラルネットワークを構造する。コア思想は、3D CNNを使用して前の層で抽出された情報に対し特徴抽出を行うことを1段の復号過程としてみなすことができ、各層の出力がいずれも1グループの三次元配列であり、1つのソフトデコーダの出力確率分布であってもよいことにある。このように、各層の症状抽出時に、各三次元配列に対しブロック分割を行い、且つ三次元畳み込みカーネルを直接用いてこれらのブロック分割後の三次元サブ行列と畳み込むことができる。図11はRGが触発する症状分割の模式図である。より正確には、すなわち、畳み込みのストライドは、

$stride = kernel\ size = (l_x, l_y, l_z)$ であり、

ここで、 $stride$ は畳み込みのストライドを表し、 $kernel\ size$ は畳み込みカーネルのサイズを表し、 l_x 、 l_y 、及び l_z は畳み込みカーネルの x 、 y 、及び z という3つの方向上におけるスケールである。すべての畳み込みネットワークの後に、全結合層を使用して分類をし、ネットワーク全体は図12に示す。ここでの全結合層の規模の大きさは基本的に L から独立して相対的に固定する。実際の応用においては、畳み込みの移動ストライド ($stride$) は畳み込みカーネルのサイズよりもやや小さくてもよく、即ち、畳み込みカーネルと畳み込む隣接するサブ行列の間に重なる部分がある。同時に、 $stride$ 及び $kernel = (1, 1, 1)$ を適時に使用して、より多くの情報を抽出すると同時に、チャンネル ($channel$) の数量を圧縮することができる。訓練結果が表すように、これらの措置はいずれも訓練収束進度を加速するようになり、比較的良好な復号性能を獲得し、同時に、ネットワークの深さを増加させることがない。

【0087】

最後に、すべての訓練サンプルはいずれもある既知の復号アルゴリズムを使用して (例えば、MWP Mを使用し、原則的には、より良いデコーダを使用してもよい) 症状-エラーセットを生じさせ、その後、エラーを論理エラー及び対応する症状セットに分け、異なるネットワークの訓練に用いるように要求する。

【0088】

一旦モデルの訓練が終了すると、それをリアルタイム誤り訂正システムの復号モジュールに配備する必要がある。

【0089】

図13に参照されるように、本願の1つの実施例が提供する量子誤り訂正復号システムの模式図を示す。該量子誤り訂正復号システムは複数の誤り訂正チップを含み、該誤り訂正チップは以下のようなステップ（131～132）を実行することに用いられる。

【0090】

ステップ131：量子回路のエラー症状情報を取得し、該エラー症状情報は量子誤り訂正コードのスタビライザージェネレーターの固有値により構成されるデータアレイである。

【0091】

量子誤り訂正コードを採用して量子回路に対しエラー症状測定を行い、相応なエラー症状情報を獲得することができ、該エラー症状情報は量子誤り訂正コードのスタビライザージェネレーターの固有値により構成されるデータアレイである。選択可能に、エラー症状情報は0及び1からなる二次元又は三次元のデータアレイである。たとえば、エラーがないときに、スタビライザージェネレーターの固有値は0であり、エラーが発生するときに、スタビライザージェネレーターの固有値は1である。

【0092】

本願の実施例においては、量子誤り訂正コードは任意の形式のトポロジック量子誤り訂正コードであってもよい。量子誤り訂正コードが表面コードであることを例とすると、表面コードにとっては、エラー及びエラー症状は具体的な空間的位置を有し、症状を引き起こすエラーがあるときに、対応する位置の補助量子ビットの固有値は1であり（該位置に1つの点粒子が出現したとみなすことができる）、エラーがないときに、対応する位置の補助量子ビットの固有値は0である。ゆえに、表面コードにとっては、もし誤り訂正過程自体のエラーを考慮しない（即ち、測定過程は、もし完璧であれば、完璧な症状と呼ばれる）と、エラー症状情報は0及び1からなる1つの二次元アレイ画像であると考えられてもよい。

【0093】

また、症状測定過程が通常ノイズを含有することを考慮し、ゆえに、一区切りの時間内に複数回の測定をして、0及び1からなる1つの三次元データアレイをエラー症状情報として獲得することができる。具体的には、上記の図4の部分の紹介及び説明を参照することができるため、ここでは再度詳細に説明しない。

【0094】

ステップ132：ニューラルネットワークデコーダを動作させてエラー症状情報に対し復号処理を行い、エラー結果情報を獲得する。該エラー結果情報は量子回路におけるエラーが発生する量子ビット及び相応なエラータイプを決定することに用いられ、ここで、ニューラルネットワークデコーダのコア演算は数値量子化処理を経て獲得される符号なし固定小数点数の積和演算である。

【0095】

ニューラルネットワークデコーダは、ニューラルネットワークに基づいて構築されエラー症状情報に対し復号を行うことに用いられる機械学習モデルである。該ニューラルネットワークデコーダの入力データはすなわちエラー症状情報であり、出力データはすなわち該エラー症状情報に対応するエラー結果情報である。ニューラルネットワークデコーダが出力するエラー結果情報に基づいて、量子回路におけるエラーが発生する物理量子ビット及び相応なエラータイプを決定することができる。たとえば、量子回路におけるエラーが発生する物理量子ビットの位置、及び該位置箇所にエラーが発生する物理量子ビットのエラータイプ、例えば、Xエラーであるか、ZエラーであるかそれともYエラー（YエラーはXエラーもありZエラーもあることを表す）であるかを決定する。

【0096】

選択可能に、ニューラルネットワークデコーダは入力層、隠れ層及び出力層を含む。ここで、隠れ層の数量は少なくとも1つであり、通常の場合に、複数である。エラー症状情報はニューラルネットワークデコーダの入力層から入力され、若干の隠れ層の特徴抽出を経て、最終的にエラー結果情報を獲得し、出力層から出力する。

10

20

30

40

50

【0097】

本願においては、ニューラルネットワークデコーダに対しいくつかの設計規範を提示することによって、ニューラルネットワークデコーダのコア演算が数値量子化処理を経て獲得される符号なし固定小数点数の積和演算であるようにし、それによりニューラルネットワークデコーダの所要のデータ量及び計算量を最大限に減少させて、リアルタイム誤り訂正の要求をより良く満たす。

【0098】

本願の実施例においては、システム全体の設計及び配備を便利にすると同時に、なるべく復号に要する消費時間を短縮させるために、量子誤り訂正復号システム（例えば、ニューラルネットワークデコーダ、及び誤り訂正チップ等）に対しいくつかの正規化処理の要求を提示する。つまり、量子誤り訂正復号システムは正規化処理を通じて獲得される。正規化処理は以下の少なくとも一項を含むが、これらに限定されない。

【0099】

1：複数の誤り訂正チップ上で動作する各ニューラルネットワークデコーダは同じネットワーク構造を有する。

【0100】

本願においては、リアルタイムフォールトトレラント誤り訂正においては、すべてのニューラルネットワークデコーダが一致する遅延で推定結果を出力し、その後、これらの推定結果に応じてエラーの発生位置及びタイプを統合して判断することが望ましい。1回の復号は複数のニューラルネットワークデコーダを同時に動作させる必要があるため、本願は同じネットワーク構造及びハードウェア構造（FPGA／ASIC）を使用してニューラルネットワークの推定を実現する傾向がある。これは、FPGA及びASICにとっては、設計開発過程を極めて大幅に簡略化できる。殊にLが比較的大きく、関するニューラルネットワークデコーダの数量が比較的多いときに、もし各々のニューラルネットワークデコーダのハードウェア上での配備をいずれも再設計して最適化しようとするれば、これは膨大な負担をもたらす。これはニューラルネットワークデコーダの設計時に一致性を確実にすることを要求する。このように、モデルをハードウェアにマッピングして実行するときに、フロントエンドとバックエンドの設計及び製作はいずれも大幅に簡略化されるようになる。同時に、Lが比較的大きいときに、比較的多いニューラルネットワークデコーダを使用する必要があるときに、規模効果が生じる。つまり、同じ構造設計はFPGA／ASICのバッチテープアウト生産を便利にし、コストを低減させるようになる。従って、本願はすべてのニューラルネットワークデコーダがいずれも同じネットワーク構造を有することを要求する（例えば、各層の重み付けの数量、オフセットの数量等を含む）。後続のシミュレーション結果からも表されるように、このような選択によりもはや最適な復号性能に達することができる。

【0101】

2：ニューラルネットワークデコーダの隠れ層は畳み込み層及び全結合層のみを含む。

【0102】

本願においては、ニューラルネットワークデコーダの隠れ層は畳み込み層及び全結合層のみを使用することを許可し、それはこの2種類のアルゴリズムのコア演算がいずれも行列演算であり、並行化が容易であるからである。並行化を最大限にするために、さらにニューラルネットワークデコーダのネットワークの層数及び演算の複雑性を更に減少させる必要がある。つまり、畳み込み層及び全結合層のみを使用することを許可し、且つmax pooling（最大プーリング）、dropout（非活性化）又はbatch norm（バッチ正規化）等の技術の使用を諱めてモデルを更に簡略化する。これらの簡略化はデータ依存性を同時に減少させ、且つ回路実現の困難性を大幅に減少させることができる。実践により示されるように、これらの補助層を減少することで、訓練の困難性を高めること、又はオーバーフィッティング現象が生じることがない。

【0103】

3：ニューラルネットワークデコーダの活性化関数はReLU活性化関数を使用する。

【0104】

ハードウェア配備の実現を便利にするために、本願は、ニューラルネットワークデコーダのすべての活性化層（`activate layer`）が、いずれもReLU活性化関数を使用することを要求する。その理由は、デジタル回路を使用して固定小数点演算を行うときに（具体的な量子化操作は後続では言及されるようになる）、出力に対するReLU活性化関数の判定が固定小数点数の符号桁のみに決められ、且つ符号桁が1である負数を全部0にリセットすればよいことにある。これはハードウェア実現の困難性を極めて大きく簡略化することとなる。同時に、ReLU活性化関数を使用することで既に十分に良い復号性能を生じさせることができ、最適に近いことを達成する。

【0105】

ニューラルネットワークデコーダに対する上記簡略化処理を経た後に、ニューラルネットワークデコーダ全体のコア演算は一種類のみであり、配列間の積和演算（MA）であり、ハードウェア実現の困難性は更に低下する。

【0106】

4：各誤り訂正チップ上で1つのニューラルネットワークデコーダを動作させる。

【0107】

選択可能に、復号性能に影響を与えない場合には、さらになるべくネットワークの数量を減少させることを選択する必要がある。遅延（伝送遅延及び計算遅延を含む）をなるべく低くして減少させるために、ハードウェアリソースが十分である場合には、本願においては、各ニューラルネットワークデコーダがいずれも単一又は複数のFPGA／ASICにおいて完了できることを望ましい。即ち、単一のチップで複数のニューラルネットワークデコーダを演算する場合を考慮しない。このように、ネットワークの数量が多いほど、必要なチップの数量が多いことを表す。ネットワークの数量をなるべく減少させることは、より少ないチップ数を意味し、復号システム全体の立ち上げの困難性を緩和するようになる。下記においては、大まかな最適ネットワーク数量を選ぶ経験的な方法が提示されている。

【0108】

5：ニューラルネットワークデコーダは異なる量子回路の誤り訂正復号に適用される。

【0109】

選択可能に、本願においては、さらに、ニューラルネットワークデコーダが十分な柔軟性を有することが望ましい。つまり、本願はマルチニューラルネットワーク復号モデルが特定のノイズモデルを満たすことができるモデルであるだけでなく、同時に異なるノイズモデルを満たす必要もある。一歩下がって、訓練済みのモデルは異なるノイズモデルにとっても良く適格でない場合であっても、モデルの構造パラメータを修正しない場合にモデルを再訓練し、且つ比較的良い復号性能を獲得することができる。このように、再設計／最適化／テプアウトをしない場合に、オンチップ記憶の重み付け、及びオフセット等のデータのみを更新して復号アルゴリズムの更新を完了することで、異なる量子回路上によりよく適用される。

【0110】

以下、 $L=5$ を例としてモデルの変換による復号性能に対する影響を検討する。まず、 $L=5$ の具体的なモデルパラメータを与える。モデルはpytorchコードで以下のよう表される。

【0111】

```
super(CNN_3D_L_X, self).__init__()
self.conv1=nn.Conv3d(1, 500, stride=(4, 1, 3), kernel_size=(6, 3, 3), padding=0)
self.relu1=nn.ReLU()
self.fc1=nn.Linear(500*2*2*1, 400)
self.relu2=nn.ReLU()
self.fc2=nn.Linear(400, 80)
```

```
self.relu3=nn.ReLU()
self.fc3=nn.Linear(80,output_size)
【0112】
```

該モデルはReLUを除き、4層のみがあり、1層の3次元畳み込み層、及び3層の全結合層を含む。本願は2つのモデル校正エラークラスを使用するため、4つのモデルはシンドロームビット数値校正を行う。症状を推定するネットワークに対し、各ネットワークは6つのシンドロームビットを校正する必要がある、 $2^6 = 64$ 個の状態を有する。従って、ネットワークの最終的な出力層は一律に64と規定される。ネットワークパラメータは合計で約80万であり、約100万回の浮動小数点乗算演算が必要である。

【0113】

続いて、以下のいくつかの場合をそれぞれ考慮する。N=2, 3, 4, 6つのシンドロームビットを1グループとし、1つのニューラルネットワークを用いてその数値を推定する。すなわち、4分類、8分類、16分類及び64分類（すなわち、上記pytorchコードにおけるoutput_size変数に変化が発生し、残りは変わらない）の問題に対応する。ここでは、対応する総モデル数量はそれぞれ14、10、8、6である。L=5、及びT=10を使用し、且つp=0.006程度でノイズサンプリングを行い、MWPMを使用して訓練セット生成を行い、その後、すべてのモデルに対し同じ訓練を行う。異なるN値に対し、復号性能（論理量子ビットのエラー率）はほとんど区別できないことを見出すことができる。即ち、比較的大きなN（L=5に対し、 $2^N < 80$ である）を選択でき、ネットワークの数量を極めて大きく減少させると同時に、ほとんど復号性能に対し何らの影響も及ぼさない（図14参照）。このような現象はL=7及び9の時にも存在する。これは一種の普遍的な現象であると考えられる。すなわち、モデルの区別能力は十分に強く、規模が十分に大きい多分類問題に対応できる。しかし、output_sizeが最後から2層目のノード数（上記pytorchコードの例においては、80である）よりも遥かに大きいと、推定性能が低下するようになる。従って出力層のノード数は最後から2層目の出力ノード数よりも小さく制限される。

【0114】

L=7及び9に対し、L=5と類似し、上記規範を満たすモデルを選択し、p=0.006~0.007を使用してノイズサンプリングを行い、且つ古典的デコーダを使用して訓練セット生成を完了する。今のところは、依然として、最も深い研究を経て適用範囲が最も広いMWPMを選択してシミュレーションで生じたエラー症状を対象として復号を行い、獲得した復号データ及び症状データセットが訓練セットを生成している。今後は、例えば、強化学習を利用するより高い性能の復号アルゴリズムを使用して訓練セットを生成することを考慮するようにして、より良い復号性能を達成することを図る。具体的なモデルパラメータは、この前に与えられたL=5を除き、本願中には示されない。ここでは、L=5, 7, 9を使用して標準回路ノイズモデル（circuit level depolarization noise）によりサンプリングされたデータセットで訓練するデコーダを考慮すると、対応する復号周期はそれぞれT=10, 14, 12であり、それぞれN=6, 6, 8を使用して対応するモデルの総数はそれぞれ6, 10, 12である。使用されるモデルは設計上、いずれもモデル規範の要求に合致する。具体的な復号性能は図15に示す。示されているように、物理的なエラー率が比較的低い場合に、SNN（図におけるFTCNN）は、それに対し訓練を行うMWPMデコーダの性能とほとんど重なり、従って理論限界を達成する。エラー率が比較的高い場合に、SNNの性能は対応するMWPMよりも僅かに悪いが、これは許容可能である。これは、フォールトトレラント誤り訂正の正常作動区間が、物理的なエラー率が比較的低いことを要求するからである。例示的に、将来、強化学習を使用して訓練して獲得される新たなデコーダは、品質のより高い訓練セットを生じさせてSNNを訓練することを所望する。

【0115】

以下、ニューラルネットワークデコーダの数値量子化手段に対して紹介及び説明を行う。

【0116】

ニューラルネットワークモデルのパラメータは一般的に、いずれもとてと大きな記憶空間を占有するようになる。L = 5 を例とすると、現状使用されているパラメータの数量は約 80 万程度である。もしハードウェアチップ上にニューラルネットワークアルゴリズムを配備するとすれば、これらのデータは事前に F P G A / A S I C のオンチップ内部メモリに配備されなければならない。これはリアルタイム誤り訂正が主内部メモリ (D D R S D R A M (D o u b l e D a t a R a t e S y n c h r o n o u s D y n a m i c R a n d o m A c c e s s M e m o r y、ダブルデータレートシンクロナスダイナミックランダムアクセスメモリ) 又は G D D R S D R A M (G r a p h i c s D o u b l e D a t a R a t e S D R A M、グラフィックスダブルデータレートシンクロナスダイナミックランダムアクセスメモリ)) からのデータの読み取りによりもたらされる遅延に耐えることができないからである。これらのパラメータは浮動小数点型であるため、各数は 32 桁の空間を占有する必要がある、可逆圧縮アルゴリズムはそれらの記憶空間を圧縮することがとても困難である。従って、それらは大量の貴重なオンチップ内部メモリを占有する。同時に、浮動小数点数の乗算計算は比較的大きな計算リソースを消費し、且つ消費時間が比較的に長くなる。もしモデルの正確率に影響を与えない場合にモデルの内部がその他の簡単な数値タイプ (たとえば固定小数点演算) を採用して計算を行うことができるならば、必要なオンチップ記憶は減少し (8 桁の固定小数点を例とすると、4 倍の記憶及びより多くの計算量を減少する)、同時に、計算速度が大幅に向上し、消費される計算リソースは大幅に減少し得る。リアルタイムフィードバックフォルトトレラント誤り訂正システムのアルゴリズム演算チップに対して、量子化操作は、すなわち予め設定された各パラメータ及び中間計算結果の表示が要するビット桁数を減少させることによってオリジナルネットワークを圧縮し、同時に記憶効率及び計算効率を向上させる方法である。

【0117】

ニューラルネットワークの予測能力は比較的高いロバスト性を有し、一般的に、データ精度に対する要求が比較的低い。量子化によってデータ精度を低減させることはモデル推定能力への影響は有限である。一般的に言えば、図 16 に示すように、量子化手段は非飽和量子化及び飽和量子化に分けられる。非飽和量子化は、一定の範囲内の浮動小数点数を固定小数点数が表現できる区間 (図 16 は 8 桁符号の固定小数点数を例とすると、区間範囲は -128 ~ 127 の整数である) に完全にマッピングすることである。データ分布が非常に不均一な場合に、非飽和マッピングは少数のデータ点を比較的大きな範囲内にマッピングすると同時に、大量のデータ点がある 1 つの非常に小さい区間にマッピングすることにより区別することができないため、比較的大きな精度損失をもたらしやすい。飽和マッピングにとって、あるいくつかの絶対値が比較的大きく、統計的確率が比較的低い点は境界点 (図 16 は例であり、-128 又は 127 である) として統一してマッピングされる。このように、比較的多くの区間内では、固定小数点数は高い確率のデータを表すために用いられる。たとえば T e n s o r F l o w / p y t o r c h / T e n s o r R T 等の産業用 / オープンソースフレームワーク下で、通常、飽和量子化手段を使用してより良い予測精度を獲得する。これらのフレームワークは通常、重み付け、出力、及びオフセット等に対し異なる量子化方式を使用し、C P U / G P U 等の比較的に柔軟な命令セット体系への応用により適しており、その追加の複雑度はディレイ及び計算リソースの増加をもたらすようになる。産業用 / オープンソースフレームワークの入力出力データのタイプは制御できいため、細部は具体的なニーズに応じてニーズを満たすように微調整を行うことができない。

【0118】

本願の技術的手段においては、非飽和量子化を直接選択しているが、これは非飽和量子化の実現が十分に簡単であり、精度損失をほぼ無視して、性能ニーズを達成することができるからである。次に、該非飽和量子化手段の細部を詳細に紹介することとなる。だが、説明する必要がある点として、いくつかのその他の実施例においては、飽和量子化手段は同様に本願に適用され、且つ低エラー率の場合での量子化損失を更に減少させることができる可能性がとても高い。飽和量子化手段か非飽和量子化手段かにかかわらず、いずれ

も本願の保護範囲内にある。

【0119】

まず、本願は量子化手段に対し具体的な制限、すなわち、符号なし固定小数点数（`unsigned fixed point number`）のみをハードウェア乗算器に入力することを許可する点を提示する。このように選択する理由は、乗算はモデル全体において実行量が最も多い演算であり、可能な限りその演算を簡略化してハードウェアのリソースを十分に利用すべきであるという点である。ゆえに、一旦ある特定の桁数の符号なし固定小数点数タイプを使用してモデルにおける浮動小数点数（入力、重み付け、オフセット、及び層と層との間の中間入力出力データ等を含む）を表すことを選択すると、すべてのハードウェア乗算器の入力データはいずれも符号なし固定小数点整数でなければならない。たとえば乗算器の出力データ及び後続の累算にオーバーフローが生じたり、固定小数点数の表示範囲を超えたり、又は負数になったりしても、後続の乗算演算が一致している符号なし固定小数点数を継続的に使用するようにするために、本願は量子化アルゴリズムが、符号なし固定小数点数の区間範囲まで収縮、増幅、及び切り捨てを行うことを要求する。

10

【0120】

上記量子化手段の設計要求を満たすことに基づき、ニューラルネットワークデコーダの隠れ層の出力パラメータの計算過程は、以下の通りである。

【0121】

1：量子化後重み付けベクトル、及び、量子化後入力ベクトルを取得する。ここで、量子化後重み付けベクトルは、数値量子化処理を経て獲得される符号なし固定小数点数形式の重み付けパラメータを含み、量子化後入力ベクトルは数値量子化処理を経て獲得される符号なし固定小数点数形式の入力パラメータを含む。

20

2：量子化後重み付けベクトルと量子化後入力ベクトルとの内積に基づいて、符号なし固定小数点数形式の出力パラメータを獲得する。選択可能に、該出力パラメータは、固定の数値範囲に属する1つの符号なし固定小数点数である。

【0122】

次に、本願は具体的な量子化プロトコルを与える。

【0123】

浮動小数点数 x_f 及びその符号なし固定小数点数を x_q で表し、固定小数点数の表示範囲が $[q_{min}, q_{max}]$ であることを考慮する。同時に、 x_f の統計的な数値範囲が $[x_{min}, x_{max}]$ であることを考慮し、スケーリング係数 $scale_x$ を定義する。

30

【0124】

【数2】

$$scale_x = \frac{x_{max} - x_{min}}{q_{max} - q_{min}}$$

【0125】

もし $[x_{min}, x_{max}]$ が0点を含むなら、0に対応する固定小数点数 x_0 は、以下【数3】であるようにする。

40

【0126】

【数3】

$$x_0 = q_{min} - \frac{x_{min}}{scale_x}$$

【0127】

もし $x_{min} > 0$ であれば、 $x_0 = q_{min}$ であると設定し、もし $x_{max} < 0$ であれば

50

、 $x_0 = q_{max}$ であると設定する。該 x_0 は参照固定小数点数と呼ばれてもよい。

【0128】

どのような場合であっても、いずれも以下の汎用非飽和マッピング（量子化）規則を有する。

【0129】

【数4】

$$x_q = x_0 + \frac{x_f}{scale_x}$$

$$x_f = (x_q - x_0) scale_x$$

10

【0130】

ニューラルネットワークを使用して推定を行う問題に戻る。本願の上記で提示されたモデル規範に応じて、畳み込み層、全結合層及びReLU層のみを使用することを許可する。従って、畳み込みか行列とベクトルとの乗算かにかかわらず、そのコアの算術演算（arithmetic operation）はいずれも2つのベクトル w と a との内積である。

【0131】

【数5】

$$\sum_i w[i]a[i]$$

20

【0132】

ここで、 i は正の整数であり、ベクトル w / ベクトル a における要素の数量を表す。 $w[i]$ は浮動小数点数形式の重み付けパラメータである。 $a[i]$ は浮動小数点数形式の入力パラメータである。

【0133】

従って、次に、如何に非飽和量子化規則を使用して、固定小数点演算を使用してこのベクトル内積演算を近似するかを重点的に検討する。まず、 w 及び a とその固定小数点数 q_w 及び q_a との関係は以下の通りである。

30

【0134】

$$w[i] = (q_w[i] - w_0) scale_w$$

$$a[i] = (q_a[i] - a_0) scale_a$$

ここで、 q_w は数値量子化処理を経て獲得される符号なし固定小数点数形式の重み付けパラメータである。 q_a は数値量子化処理を経て獲得される符号なし固定小数点数形式の入力パラメータである。 $scale_w$ は重み付けパラメータに対応するスケール係数である。 $scale_a$ は入力パラメータに対応するスケール係数である。 w_0 は重み付けパラメータに対応する参照固定小数点数である。 a_0 は入力パラメータに対応する参照固定小数点数である。

40

【0135】

ベクトル w と a との内積結果を O と記し、 O とその量子化 O_q とも対応関係を有する。

【0136】

$$O = (O - O_q) scale_o$$

ここで、 $scale_o$ は出力パラメータに対応するスケール係数である。

【0137】

同時に、 O の完全な表現式は以下の通りである。

【0138】

【数6】

50

$$O = \sum_i w[i] a[i] + b$$

$$= \sum_i scale_w scale_a (q_w[i] - w_0) (q_a[i] - a_0) + b$$

【0139】

ここで、bは浮動小数点数形式のオフセットパラメータである。

【0140】

便宜上、n（nは正の整数である）桁の固定小数点数を使用すると、 $q_{min} = 0$ 、 $q_{max} = 2^n$ である。aが取る値は症状自体（1層目のネットワーク）であるか、前の層の計算結果がReLUを経た後の出力結果であるため、この前のマッピングルールに応じて、 $a_0 = 0$ 及び $O_0 = 0$ となる。上の式を変換した後に、 O_q と q_w 、 q_a 及び q_b との関係を獲得できる。

【0141】

【数7】

$$O_q = \underbrace{\frac{scale_w scale_a}{scale_o}}_{S_1} \left(\underbrace{\sum_i q_w[i] q_a[i]}_A - w_0 \underbrace{\sum_i q_a[i]}_B \right) + \underbrace{\frac{scale_b}{scale_o} (q_b - b_0)}_{S_2}$$

【0142】

ここで、 q_w は数値量子化処理を経て獲得される符号なし固定小数点数形式の重み付けパラメータである。 q_a は数値量子化処理を経て獲得される符号なし固定小数点数形式の入力パラメータである。 q_b は数値量子化処理を経て獲得される符号なし固定小数点数形式のオフセットパラメータである。 $scale_w$ は重み付けパラメータに対応するスケールリング係数である。 $scale_a$ は入力パラメータに対応するスケールリング係数である。 $scale_o$ は出力パラメータに対応するスケールリング係数である。 $scale_b$ はオフセットパラメータに対応するスケールリング係数を表す。 w_0 は重み付けパラメータに対応する参照固定小数点数である。 b_0 はオフセットパラメータに対応する参照固定小数点数である。iは正の整数である。

【0143】

O_q はReLUを経た後に、次の層の入力 q_a として次の計算を行う。

【0144】

ここで、この表現式における各量に対し分析を行う。

【0145】

【数8】

$$1 : S_1 = \frac{scale_w scale_a}{scale_o}$$

【0146】

【数8】はw、a及びOのスケールリング係数によって決められる。 $scale_w$ はモデル重み付けパラメータのスケールリング係数である。一旦モデルが与えられた（訓練が完了した）後は、 w_{max} 及び w_{min} はいずれも決定された数字である。しかし、 $scale$

a 及び $scale_o$ は出力層の最大（小）値によって決められるため、異なる入力に対しては、これらの値はいずれも異なる。従って、1つのテストサンプルセットにおいてアルゴリズムを動作させ、各層の出力の最大（小）値を収集して統計的な平均値を計算し、最終的に $scale_a$ 及び $scale_o$ を得ることしかできない。ここでのテストサンプルのエラー率は必ず実際の量子ビットのエラー率と一致し、量子化後のモデルが実際のエラー率下で最適であることを保証する。注意される点として、 S_1 は一般的に浮動小数点数であり、且つ 1 よりも小さい可能性がある。すべての演算をいずれも固定小数点数にするために、それに対し 2 の、ある 1 つのべき乗を乗じて（具体的な理由は後続で詳述される） m とし、1 つの正の整数 [数 9] に近似させる必要がある。

【0147】

【数 9】

$$\tilde{S}_1$$

【0148】

【数 9】は $[q_{min}, q_{max}]$ に制限される必要がなく、【数 8】に対する演算（乗算）は 1 回のみである。従って、特殊な最適化済みの乗算器をカスタマイズできる。

【0149】

【数 10】

$$_2 : S_2 = \frac{scale_b}{scale_o} (q_b - b_0)$$

【0150】

【数 10】は S_1 の場合と類似する。テストサンプルセットでモデルを動作させることによって $scale_o$ を獲得する必要がある。 S_2 は同様に 1 つの浮動小数点数であり、しかし、一般的に 1 よりもはるかに大きく、従ってそれを直接切り捨て、小数部分を除去し、1 つの近似される整数 [数 11] を獲得する。

【0151】

【数 11】

$$\tilde{S}_2$$

【0152】

【数 12】

$$_3 : A = \sum_i q_w[i] q_a[i]$$

【0153】

【数 12】は、すなわち、固定桁固定小数点数の積和演算であり、モデル全体における演算量が最も大きいコア部分である。ここでは q_w 及び q_a がいずれも $[0, 2^n]$ にあることを要求する。 A の計算量はすべてのステップのうち最も大きいものである。従って、すべての乗算がいずれも一致する乗算器を使用することを要求する。注意される点として、一般的な場合に、 A は固定小数点符号なし数の表現範囲 $[0, 2^n]$ からオーバーフローするようになり、従って A に対し区間範囲制限をしない。

【0154】

【数 13】

10

20

30

40

50

$$4 : B = w_0 \sum_i q_a[i]$$

【0155】

【数13】は固定桁固定小数点数の総和演算であり、最後に w_0 と掛け算する。同様に、 B も $[0, 2^n]$ からオーバーフローするようになり、従って、 B に対しても区間範囲制限をしない。

【0156】

実際の演算においては、 A 及び B はいずれも正の整数であるが、それらの差は負数である可能性がある。このような場合を処理するために、 A 及び B を獲得した後に、それに対して統一して0を補充してある1つの固定長さ（具体的な場合次第で定める）に拡大し、その後、それを2つの符号付き整数としてみなして減算して $A - B$ を獲得し、獲得した結果を【式14】と掛け算し、1つの符号付き整数【式15】を獲得する。この結果は既に 2^m 倍増幅されているため、獲得した結果をビット操作レベルで右へ m （ m は正の整数である）桁シフトする必要がある、 $S_1(A - B)$ の近似を獲得する。その後、獲得した結果と【式16】とに対して符号付き数の加算を行い、 $S_1(A - B) + S_2$ の符号付き整数近似を獲得する。

【0157】

【数14】

$$\tilde{S}_1$$

【0158】

【数15】

$$\tilde{S}_1(A - B)$$

【0159】

【数16】

$$\tilde{S}_2$$

【0160】

また、ニューラルネットワークデコーダの各隠れ層の後に、さらに活性化関数層を含んでもよい。該活性化関数層は上記のモデル設計規範に従ってReLU活性化関数を使用することができ、隠れ層の出力パラメータに対し以下のような処理を行うことに用いられる。出力パラメータの符号桁を取得し、符号桁が出力パラメータが正数であることを指示する場合に、出力パラメータの各ビット桁の数値を変わらないように保留し、符号桁が出力パラメータが負数であることを指示する場合に、出力パラメータの各ビット桁の数値を零にセットする。

【0161】

1つの例においては、 $A = 4$ 、 $B = 8$ と仮定し、8桁二進数で表すならば、それぞれ $A = 00000100$ 、 $B = 00001000$ である。 $A - B < 0$ であるが、この2つは符号なし数であり、そして後続では処理される必要がある数は8桁の表示範囲よりも大きくなる。従って、ここでは、この2つの数をまずより多くの桁で表される符号付き桁数に拡大する。16桁に拡大することを例とすると、 $A = 00000000000000100$ 、 $B = 00000000000001000$ である。ここでは A 及び B は符号なし数ではなく、2の補数（two's complement）の符号付き表示としてみなされ、1桁目は符号数（0は正であり、1は負である）である。このように、 $A - B = -4$ は1

10

20

30

40

50

1 1 1 1 1 1 1 1 1 1 1 1 1 0 0として表され、これは1つの符号付き数である。その後、ReLU層は1桁目が0であるかそれとも1であるかのみに応じて出力を決めればよく、1桁目が0であると、結果が0よりも大きいことを表し、元の結果を保留し、1桁目が1であると、結果が0以下であることを表し、すべてのビットを全部0にリセットする。

【0162】

L = 5を例として量子化手段の効果を観察する。標準的な8桁の固定小数点量子化手段を選ぶ。ここでは、合計で6つのモデルあり、各々のモデルは合計で4層あり、1つの畳み込み層及び3つの全結合層を含む。畳み込み層に対し、 $m = 8$ を設定し、全結合層に対し、 $m = 16$ を設定する。 m の設定は一定の任意性を有し、具体的な数値は性能損失の場合次第で定める必要がある。また、異なる物理的なエラー率に対し、入力出力層の統計的性質はいずれも異なる。従って特定のエラー率を対象として S_1 及び S_2 を決定するように使用する必要がある。

【0163】

量子化の影響を定量的にテストするために、それぞれ $p = 0.001$, 0.0015 , 0.002 に対応する、3つの校正データセットを選定する。サンプルセットの大きさは20000であり、 S_1 及び S_2 の数値を決定するのに用いられる。同様に、3つのテストセットも選び、規模は40000程度である。テスト結果は図17に示すように、32桁の浮動小数点数に比べて、8桁の固定小数点数の予測精度損失は6つのモデルに対しいずれも非常に小さく、エラー率は平均的な状況で10%上昇し、許容可能範囲内にある。 p が比較的小さいときに、エラー率の精度損失は相対的に比較的大きく、これはこの時に獲得できる非零サンプル点が比較的小なく、 S_1 及び S_2 の分散が比較的大きいことに関係する可能性がある。従って、後続では p 値が比較的小さい時に飽和量子化を使用して改良を行うことを考慮できる。

【0164】

以上のように、本願の実施例が提供する技術的手段は、ニューラルネットワークデコーダを採用して量子誤り訂正復号を行うシステムにとって、該ニューラルネットワークデコーダに対し正規化処理を行うことによって、そのコア演算が数値量子化処理を経て獲得される符号なし固定小数点数の積和演算となり、それによりニューラルネットワークデコーダの所要のデータ量及び計算量を最大限に減少させ、リアルタイム誤り訂正の要求をより良く満たす。

【0165】

また、ニューラルネットワークデコーダは原則的には、任意のフォールトトレラント誤り訂正復号アルゴリズムをシミュレートでき、且つモデル構造の最適化後に、元のデコーダの性能に近づけ、且つすべての演算はいずれも行列演算であり、大規模な並行化に適する。

【0166】

以下、ニューラルネットワークデコーダのハードウェアチップ（例えば、FPGA／ASIC）における動作過程に対して紹介及び説明を行う。

【0167】

量子化手段を決定した後に、ニューラルネットワークデコーダをそれぞれFPGA／ASICに配備することができる。ハードウェアリソースを最大限に利用するために、各々のFPGA／ASICに対し、多くとも1つのニューラルネットワークデコーダのみを配備することを許可する。すべてのニューラルネットワークデコーダの構造はいずれも同じである。従って、1つのモデルの配備を完了した後に、その他のモデルの配備も同様な方式で完了できる。

【0168】

モデル規範に従って、各々のモデルは3次元畳み込み層及び全結合層のみを有し、その演算規則は図18に示されている。この2種の演算のコアはいずれもベクトル内積であり、入力データを大量に繰り返して応用する必要があるため、入力データをバッチコピーしてモデルのパラメータと積和演算をすることによって並行度を最大化することができる。

原則的には、もし使用可能な計算リソースが無限であり、各々の3次元畳み込み層及び全結合層の乗算がいずれも1回の乗算周期内に完了できれば、最速ではK（Kは正の整数である）個の乗算計算周期内でモデル推定を完了できる。ここでは、Kはモデルの層数である。

【0169】

しかし、実際の状況下では、計算リソース、殊に貴重な乗算計算リソースは有限である。FPGAを例とすると、オンチップのDSP（Digital Signal Processor、デジタル信号プロセッサ）は一般的に、数千個の8桁の固定小数点計算器を提供でき、つまり、多くとも数千回の8桁の乗算演算を同時に行う。配線の制限を考慮すると、同時に乗算演算を行うリソースはまた更に低減する。ニューラルネットワークデコーダは数百万回の8桁の乗算計算が必要である。総計算量が固定し、且つすべての計算のタイプも固定するため、計算時間を短縮させるためには、できるだけオンチップの計算リソースを同時に使用する必要がある。

【0170】

一方、計算アーキテクチャの汎用性も考慮する必要がある。すなわち、アーキテクチャは異なるアーキテクチャモデルを実現する必要がある。この点は、殊にASICを使用する時に最も重要であり、もし一種の構造のモデルのみしか実現できない場合、一旦システムを切り替える（たとえば、具体的なノイズニーズに応じてモデルの構造、ハイパーパラメータ等を改める）必要があれば、新たなシステム／ノイズモデルに応じて、フロントエンド及びバックエンドの更新及びテープアウトを再び行う必要があり、時間及び材料コストは非常に高価である。以下、本願は、モデルの推定演算を加速する1つの汎用完全多重化並行演算アーキテクチャを提示する。例示的には、使用可能なオンチップ内部メモリが十分に多いときに、該アーキテクチャは、原則的には、任意の上述した規範に従うモデルの動作を並行化して加速することができる。

【0171】

まず、この前に、全計算過程はいずれも複数のステップのベクトルの内積に分解され得ることに注意した。量子化内積演算と併せて、その前に下の式【数17】が既に与えられた。

【0172】

【数17】

$$O_q = \underbrace{\frac{scale_w scale_a}{scale_o}}_{S_1} \left(\underbrace{\sum_i q_w[i] q_a[i]}_A - w_0 \underbrace{\sum_i q_a[i]}_B \right) + \underbrace{\frac{scale_b}{scale_o} (q_b - b_0)}_{S_2}$$

【0173】

ここでは、 S_1 、 S_2 、 q_w 、及び w_0 は、モデルが与えられ（訓練が完了している）、且つテストセットを経て S_1 及び S_2 を校正した後に、いずれも決定された値であり、具体的な入力とは無関係である。これらの数値は事前にオンチップ内部メモリに記憶される必要があり、内積を行う必要があるときに、オンチップ内部メモリの相応な位置から迅速に読み取られることで便利になる（遅延は1つのクロック周期にあり、数ナノ秒である）。A及びBの2項の計算は互いに依存せず、同時に行うことができる。ここで、Aに含まれる乗算の数は最も多く、主要な計算リソースを占める。BはAに比べて、計算量を無視できる。1つの周期内に1層のすべての演算を完了できないため、同時にそのうちの部分の内積演算を完了することしかできず、ひいては内積演算の一部の乗算演算及び累積を同時に完了する。このような内積の計算ステップは3つのステップ、すなわち、データ内部

メモリからのパラメータ読み取り (Read、「R」と略称)、積和演算 (Multiplication Accumulation、「MA」と略称)、及び量子化 (Quantization、「Q」と略称) を含んでもよい。従って、この3つのステップをパイプライン化でき、図19に示すように各層の出力結果をオンチップレジスタに格納する。
【0174】

例示的な実施例においては、図19に示すように、誤り訂正チップの体系構造全体は、オンチップ内部メモリ (On-chip Memory)、制御ユニット (Control Unit) 及び算術計算モジュール (Arithmetic Module) を含む。

【0175】

オンチップ内部メモリはニューラルネットワークデコーダのパラメータ情報を記憶することに用いられ、上記において紹介された既定パラメータ (例えば、 S_1 、 S_2 、 q_w 、及び w_0 等を含む。図19におけるSCは q_w を除く既定パラメータ、例えば S_1 、及び S_2 等を表す) 及びモデル構造情報 (Model Specsであり、モデルの各種のハイパーパラメータを含む) を含む。これらのデータはいずれも静的なものである。従って、全量子計算過程の開始前に上位機PC (Personal Computer、パーソナルコンピュータ) からオンチップ内部メモリに読み込むことができる。

【0176】

制御ユニットは算術計算モジュールがオンチップ内部メモリ中から上記パラメータ情報を読み取るように制御することに用いられる。

【0177】

算術計算モジュールはオンチップ内部メモリ中から上記パラメータ情報 (即ちR) を読み取り、上記パラメータ情報に基づいて積和演算 (即ちMA) を実行し、積和演算の結果に基づいて、ニューラルネットワークデコーダの隠れ層の出力パラメータ (即ちQ) を計算することに用いられる。たとえば、内積演算を行って出力パラメータを獲得するときに、オンチップ内部メモリ中から既定パラメータを取得することによって、該既定パラメータ及び量子化後重み付けベクトルと量子化後入力ベクトルとの内積に基づいて、符号なし固定小数点数形式の出力パラメータを獲得する。ここで、既定パラメータとは例えば S_1 、 S_2 、 q_w 、及び w_0 等を含み、取る値が入力パラメータの影響を受けないパラメータを指す。

【0178】

算術計算モジュール中におけるR、MA、及びQの3つの部分はパイプライン方式を採用して実行し、それぞれ内部メモリ及び上段の出力からデータを読み取り、積和演算を行い、出力を量子化し、且つ出力結果をオンチップ一時レジスタ (register) に記憶する。算術計算モジュールはすべてのオンチップ乗算及び加算リソースを十分に利用する必要があり、これは前の層の出力データをファンアウト (fan out) してモジュール内の異なる乗算ユニット及び加算ユニットにコピーする必要がある。読み取りのパイプライン構造のため、オンチップ内部メモリからの読み取りによりもたらされるディレイは無視できる。内部メモリにおけるモデル構造パラメータ (層数、各層の重み付け、及びオフセットの数等) は制御ユニットに初期化されて、各層のモデルネットワークの計算周期を判断するのに用いられ、且つ1層のネットワーク計算が完了した後に、複数の内部メモリチャネル中から相応なネットワーク層のデータを選別して算術計算モジュールに伝送して出力レジスタを次の層の入力レジスタとして切り替えることをデータセレクター (Multiplexer、MUX) に通知する。モデルがいずれも浅い層のものであるため、制御回路自体の複雑性はモデル複雑度の増加に伴って迅速に増加することがなく、従って大型モデルに適用される。一方、モデルが比較的簡単である場合に、このような完全多重化アーキテクチャは追加の複雑度を増加させると同時に、それによるメリットが有限であるようになる可能性がある。ゆえに、具体的な場合を考慮して、使用される実用的なアーキテクチャは図19を基に適宜修正されるようになる。例示的には、多くとも2段のネットワークの算数演算を1本のパイプラインとして直列接続してもよく、このように層と層との間のデータウェイトによりもたらされる遅延 (次の層の入力は前の層の計算結果を

10

20

30

40

50

ウェイトする必要がある、この時、計算を行うことができず、オンチップ計算リソースの浪費をもたらす)を減少させることができる。しかし、図19に参照されるように、データの依存性のため、同一時間に、多くとも2層のネットワークのパイプラインを直列して1本のより長いパイプラインを形成することしかできない。具体的に如何に計算リソースをパイプラインの各部分に割り当てるかも異なるモデルのハイパーパラメータに応じて調整を行う必要がある。しかし、このリソースの具体的な割り当ても制御ユニットにより決められ、プログラマブルである。

【0179】

図20に示すように、本願の1つの例示的な実施例はさらにフォールトトレラント量子誤り訂正システムを提供し、該フォールトトレラント量子誤り訂正システムは、量子回路と、QCP 220と、AWG 230と、DAQ 240と、複数の誤り訂正チップ251を含む量子誤り訂正復号システム250と、を含む。

10

【0180】

AWG 230はエラー症状測定に用いられる測定波形を量子回路に送信することに用いられる。

【0181】

DAQ 240は量子回路を経て増幅された後の測定波形を採集し、採集して得た測定波形に対し復調を行って量子回路のエラー症状情報を獲得し、QCP 220にエラー症状情報を送信することに用いられ、該エラー症状情報は量子誤り訂正コードのスタビライザージェネレーターの固有値により構成されるデータアレイである。

20

【0182】

QCP 220はエラー症状情報を量子誤り訂正復号システム250における複数の誤り訂正チップ251に送信することに用いられる。

【0183】

誤り訂正チップ251は、ニューラルネットワークデコーダを動作させてエラー症状情報に対し復号処理を行い、エラー結果情報を獲得することに用いられ、該エラー結果情報は量子回路におけるエラーが発生する量子ビット及び相応なエラータイプを決定することに用いられる。ここで、ニューラルネットワークデコーダのコア演算は数値量子化処理を経て獲得される符号なし固定小数点数の積和演算である。

【0184】

例示的な実施例においては、誤り訂正チップ251はさらに、QCP 220にエラー結果情報を送信することに用いられる。

30

【0185】

QCP 220はさらに、エラー結果情報に基づいて、AWG 230に誤り訂正命令を送信することに用いられ、該誤り訂正命令中にエラーが発生する量子ビットの位置情報及び相応なエラータイプを含む。

【0186】

AWG 230はさらに、誤り訂正命令に基づいて、量子回路に誤り訂正波形を送信することに用いられ、該誤り訂正波形は量子回路におけるエラーが発生する量子ビットを訂正することに用いられる。

40

【0187】

選択可能に、以下の少なくとも1グループの通信はLVDS(低電圧差動信号)を採用する。

【0188】

1: QCPとAWGとの間の通信。具体的に言うと、QCPのハードウェアチップ(例えば、FPGA)とAWGのハードウェアチップ(例えば、FPGA)との間の通信である。

【0189】

2: QCPとDAQとの間の通信。具体的に言うと、QCPのハードウェアチップ(例えば、FPGA)とDAQのハードウェアチップ(例えば、FPGA)との間の通信であ

50

る。

【0190】

3：QCPと誤り訂正チップとの間の通信。具体的に言うと、QCPのハードウェアチップ（例えば、FPGA）と誤り訂正チップ（例えば、FPGA）との間の通信である。

【0191】

本願の実施例においては、LVDS標準を使用することによってシステムの信号伝送リンク全体のディレイを低減させる。誤り訂正システム全体は、ディレイに対し非常に高い要求を有するため、いかなる現状の高速通信プロトコルフレームワークもデータ伝送にはほとんど使用できない。これは、これらのプロトコルがとても高いデータレートに達することができる（高速SerDesを使用するプロトコルは20Gbps以上の速度に達することができる）が、データの伝送時に、送信及び受信側の両方においていずれも通信プロトコルの変換を経由する必要があるからである。もし伝送されるデータ量がとても大きければ、これらのプロトコル層のオーバーヘッドは無視できるようになるが、量子誤り訂正のような単一回のデータが数十bitオーダーである伝送については、数百ナノ秒のプロトコルオーバーヘッドは許容不能である。ゆえに、本願の実施例はLVDS標準を使用して信号伝送を行うことで、比較的高いレートに達することができ、且つ複数ペアの同時伝送を許可し、プロトコル層に追加のディレイが生じることがなく、それによりフィードバックリンク全体の信号伝送ディレイは要求を満たすことができる。

10

【0192】

以下、各ステップに使用されるLVDSチャンネル数及びレートに対し例示的な紹介及び説明を行う。

20

【0193】

1：DAQのADC（Analog to Digital Converter、アナログデジタル変換器）からDAQのFPGAまで：ADCは量子回路が返信した測定波形をデジタル信号に変換し、FPGAに伝えて処理を行って量子ビットの測定結果を獲得する。本システム中に使用されるADCの規格は、12bitの分解能、1Gbpsのデータレートであり、ゆえに、各塊のADCからFPGAまで12ペアのLVDSを使用し、500MHzで動作させ、毎回5つの点を伝送し、単一回の伝送に5nsが必要である。

【0194】

2：DAQのFPGAからQCPのFPGAまで：各塊のDAQのFPGAの単一回の伝送データの大きさは40bitであり、以下のいくつかの部分からなる。1塊のFPGAは4グループの測定結果が生じるようになり、各グループの測定結果は8つのqubit（量子ビット）を含み、各qubitの状態は1bitを用いて表すことができ（各qubitは測定後に、0及び1の2種の異なる状態のみがある）、同時に毎回の伝送に8bitの伝送ヘッダが必要であり、有効操作等を指示するために用いられ、すなわち、

30

40bit=8bit(header)+8(qubits)×4(group)×1(bit)である。

【0195】

各塊のDAQのFPGAからQCPのFPGAまで8ペアのLVDS線を使用し、伝送クロック周波数は200MHzであり、単一回の伝送の完了に15nsの時間が必要である。

40

【0196】

3：QCPのFPGAと誤り訂正チップとの双方向伝送：2回の伝送は基本的に以上と同様である。

【0197】

4：QCPのFPGAからAWGのFPGAまで：QCPがAWGに伝えるデータは波形選択コードであり、すなわち、記憶中からどのような波形（量子操作）を選択して送信し出すかをAWGに告知する。QCPのFPGAから各塊のAWGのFPGAまでの単一回の伝送データの大きさは48bitであり、以下のいくつかの部分からなる。各塊のAWGのFPGAは8つのアナログチャンネルを有し、各チャンネルは32種の波形、すなわち

50

大きさが5 b i tの波形選択コードを記憶し、同様に毎回の伝送に8 b i tの伝送ヘッダが必要である。すなわち、

$48 \text{ b i t} = 8 \text{ b i t} (\text{header}) + 8 (\text{channel}) \times 5 (\text{bits})$ である。

【0198】

Q C PのF P G Aから各塊のA W GのF P G Aまで8ペアのL V D S線を使用し、伝送クロック周波数は200 M H zであり、単一回の伝送の完了に15 n sの時間が必要である。

【0199】

5：A W GのF P G AからA W GのD A C (D i g i t a l t o A n a l o g C o n v e r t e r、アナログデジタル変換器)まで：A W Gは誤り訂正波形データをD A Cに送信してアナログ波形(すなわち誤り訂正波形)に変換し、且つ最終的に量子回路に作用させる。本システムが使用するD A Cの規格は16 b i tの分解能、2 G b p sの伝送率であり、ゆえに、F P G Aから各塊のD A Cまで32ペアのL V D Sを使用し、500 M H zで動作させ、1回で10個の点のデータを伝送し、10 n sの時間をかけることを要する。

【0200】

いくつかの説明を補充する。まず、理解できるように、このセットのシステムにおいてはすべての測定結果をQ C Pに伝え戻してから、Q C Pにより誤り訂正チップに伝えるようになる。明らかなように、ここでは測定結果を誤り訂正チップに直接伝えることは一定の時間を節約するようになるが、このようにすると、各塊のD A QのF P G Aに32ペアのL V D S接続線を別途増加させることを要するようになり、現状使用されているF P G Aのピンリソースの使用が不十分になることを招くようになる。この他、本システムにおいては、F P G A間のデータ伝送のL V D S周波数はD A / A DからF P G Aまでの間のL V D S周波数よりも低く、その理由として、ハードウェアの実現時に、異なるモジュール間の物理的距離のため、F P G A間のP C B (P r i n t e d C i r c u i t B o a r d、プリント回路基板)配線が比較的長くなり、もし周波数が高すぎればデータずれ等の場合が出現するようになる可能性がある。ゆえに、現状ではまず保守的な200 M H zを採用する。

【0201】

同時に以上で計算した時間はデジタル論理がL V D Sデータを伝送するための所要の時間であり、信号が実際にP C B線において伝送される時間は算入されていない。しかし、実際の測定計算によれば、最も遠いF P G A間のデータ伝送における、配線上で要する消費時間は2～3 n sのみであるため、受信側の論理的な消費時間に含められ、改めて単独で計算する必要はない。

【0202】

1セットの測定制御システムの所要のアナログチャンネル数(A W G及びD A Qモジュール)は量子回路に含まれるビット数量の増大に伴って線形的に上昇するが、アナログチャンネル数の増加もL V D S伝送線の数量の上昇を意味することを見出すことができる。現在の量子測定制御システムにおいては、主にF P G Aを使用して各モジュールのデジタル論理を実現するが、F P G Aは有限な数量のピンを提供することしかできず、そのため1塊のQ C Pに対応するF P G Aが制御可能なアナログチャンネル数が制限されていることになる。同時に、Q C Pにおいて動作する量子命令セットは同様に量子命令処理速度の問題に直面しており、ゆえに、制御される必要がある量子ビットの数が上昇するときに、Q C Pの制御能力はボトルネックに陥ってしまう。

【0203】

量子ビットの数量が特に多くないとき(約数十や百にのぼるビット)には、1セットの中央式アーキテクチャを使用して制御を行うことができる。より多くの量子ビットを制御する必要があるとき(約数百や千にのぼる)には、アーキテクチャ上からQ C Pが直面している拡張可能性の問題を解決する必要がある。ここでのアーキテクチャ設計は主に量子

10

20

30

40

50

命令セットに関連する内容に関する。

【0204】

以下、1つの具体的な例を与え、本願の技術的手段に対して紹介及び説明を行う。

【0205】

L = 5 を例に、ニューラルネットワークデコーダに基づくフォールトトレラント量子誤り訂正システムを実現し、且つ復号遅延を計算する。具体的なニーズは以下の通りである。

【0206】

現状、L = 5 のリアルタイムフィードバック誤り訂正アルゴリズムは計6つのニューラルネットワークデコーダを有し、各デコーダは4つの隠れ層（行列乗算ネットワークとも呼ばれてもよい）を含む。

【0207】

1層目の行列は500個の $6 \times 4 \times 3$ 行列と1つの $10 \times 4 \times 3$ 行列との畳み込みであり、 $10 \times 4 \times 3$ 行列は0、1状態の判断結果（入力シンδροームビット）であり、 $6 \times 4 \times 3$ 行列は重み付け（weight）であり、

2層目の行列は1つの 2000×400 行列と1つの2000ベクトルとの積であり、

3層目の行列は1つの 400×80 行列と1つの400ベクトルとの積であり、

4層目の行列は1つの 80×64 行列と1つの64（又は2）ベクトルとの積であり、

各層の計算結果はいずれも量子化切り捨てを行う必要があり、8bitの符号なし固定小数点データに処理される。

【0208】

実現手段は以下の通りである。

【0209】

Intel Stratix 10 シリーズ FPG A チップを採用してニューラルネットワークアルゴリズムを実現し、該 FPG A チップは5760個のDSP IP（乗算器IP）リソースを含み、多くとも11520個のINT8（8桁の整数）乗算をサポートする。

【0210】

各 FPG A チップは1つのニューラルネットワークデコーダを実現でき、システム全体は計6つのニューラルネットワークデコーダがあり、6枚の FPG A チップが必要である。L = 5 のため、ニューラルネットワークデコーダは4層のみを有し、最後の2層の計算量が非常に小さく、上記において紹介された手段を完全に使用することにより比較的複雑な制御回路を使用することになるため、オンチップリソースを消費すると同時に多くの並行処理を可能にする訳ではなく、現状では採算が合わない。ゆえに、本実現実例は上記において記載される完全多重化手段を完全に踏襲するのではなく、各層の行列演算がいずれも自己の専用回路を有する手段を採用する。

【0211】

手段のブロック図は図21に示すものを参照する。各層のネットワークの入力は3つの部分を含み、それぞれ、

1：前の層のネットワークの計算結果（1層目のネットワークの対応する入力量子ビットの状態判断結果である）、

2：重み付けパラメータ（上位機によって配置を行う）、及び

3：量子化切り捨てデータ（上位機によって配置を行う）であり、

ここで、重み付けパラメータ及び量子化切り捨てデータは計算をスタートする前に、上位機によってオンチップ記憶リソース中に配置され、状態判断結果を受信した後に、計算をスタートする。

【0212】

1層目のネットワーク及び2層目のネットワークの計算過程はパイプ処理を形成する。

【0213】

3層目のネットワーク及び4層目のネットワークの計算過程はパイプ処理を形成する。

【0214】

10

20

30

40

50

各FPGAチップが、多くとも11520個のINT8乗算を同時にサポートするが、各ニューラルネットワークデコーダが計1.2M回の乗算を有する。ゆえに、11520個のINT8乗算を4層のネットワークに割り当てることを考慮する必要がある。又は、各層のネットワークの各クロック周期内での計算回数を考慮する必要がある。割り当ての原則は以下の通りである。

【0215】

1：DSP IPリソースの使用率は最も大きい。

2：総計算ディレイは最も小さい。

3：FPGAチップ配線は通過する。

【0216】

選択可能な割り当て戦略は以下の「表1」に示し、表におけるデータは予測値である。

【0217】

【表1】

「表1」

		1層目のネットワーク	2層目のネットワーク	3層目のネットワーク	4層目のネットワーク	総計
手段1	各クロック周期に計算される乗算の数	896	6400	3200	512	11008
	乗算器IPの数量	0	3200	1600	256	5056
	クロック周期の数	146	0	34	0	213
	時間 (ns)	730	0	170	0	1065
手段2	各クロック周期に計算される乗算の数	1120	8000	1600	256	10976
	乗算器IPの数量	0	4000	800	128	4928
	クロック周期の数	121	0	44	0	198
	時間 (ns)	605	0	220	0	990

【0218】

上記3つの原則に応じて、最終的に手段2で指定される割り当て戦略を使用した。

【0219】

1層目のネットワークにおいては、10×6×3行列のデータが1bitであり、従って、乗算演算は普通の論理リソースを使用して実現でき、DSP IPを使用する必要がない。2層目のネットワークの計算過程は1層目のネットワークの計算過程とパイプ処理を形成し、従って、2層目のネットワークの計算時間は0であると考えられてもよい。3層目及び4層目のネットワークの計算も同様である。

【0220】

最終的に、ネットワークモデル全体の計算時間は1010ナノ秒である。

【0221】

この実現形式は、既に実際のボードにおいて認証されて通過している。

【0222】

具体的な認証方法は、それぞれ該回路及びPCプログラムに同じデータ及びパラメータを入力し、両者の出力を比較することであり、実測したところ、両者の出力が一致するこ

とを見出した。

【0223】

要するに、フォールトトレラント誤り訂正は大規模な汎用量子計算への不可避的な道であり、基礎的な技術である。本願が提供する技術的手段は、浅いニューラルネットワークのフォールトトレラント復号アルゴリズムをFPGA高速フィードバック制御システムに実装し、リアルタイムフォールトトレラント復号を実現する（総遅延は $1.2\mu\text{s}$ （マイクロ秒）程度であり、ニューラルネットワーク演算子を最適化した最新のFPGAに置換すると、ディレイを更に 700ns 程度に減少することが期待できる）。既知のMWPM手段（シングルコア 3.7GHz Intel Xeon CPU）よりも約 10000 倍程度ディレイを減少し、最適化したMWPMよりも約 20 倍以上時間を短縮し、リアルタイム誤り訂正の要求を達成することができ、重要な技術的意味を有する。

10

【0224】

また、本願の1つの例示的な実施例はさらに、量子誤り訂正復号方法を提供し、該方法は誤り訂正チップにより実行され、該方法は以下のステップを含んでもよい。

【0225】

1：量子回路のエラー症状情報を取得し、該エラー症状情報は量子誤り訂正コードのスタビライザージェネレーターの固有値により構成されるデータアレイであり、

2：ニューラルネットワークデコーダを動作させてエラー症状情報に対し復号処理を行い、エラー結果情報を獲得し、該エラー結果情報は量子回路におけるエラーが発生する量子ビット及び相応なエラータイプを決定することに用いられ、

20

ここで、ニューラルネットワークデコーダのコア演算は数値量子化処理を経て獲得される符号なし固定小数点数の積和演算である。

【0226】

方法実施例における詳細に説明されていない細部に対しては、上記その他の実施例における紹介及び説明を参照することができるため、ここでは再度詳細に説明しない。

【0227】

また、本願の1つの例示的な実施例はさらに量子誤り訂正復号装置を提供し、図22に参照されるように、該装置は、取得ユニット2201と、復号ユニット2202と、を含み、

取得ユニット2201は、量子回路のエラー症状情報を取得することに用いられ、エラー症状情報は量子誤り訂正コードのスタビライザージェネレーターの固有値により構成されるデータアレイであり、

30

復号ユニット2202は、ニューラルネットワークデコーダを動作させてエラー症状情報に対し復号処理を行い、エラー結果情報を獲得することに用いられ、エラー結果情報は量子回路におけるエラーが発生する量子ビット及び相応なエラータイプを決定することに用いられ、

ここで、ニューラルネットワークデコーダのコア演算は数値量子化処理を経て獲得される符号なし固定小数点数の積和演算である。

【0228】

説明する必要がある点として、上記実施例が提供する装置はその機能を実現するにあたり、上記各機能ユニットの区分分けのみを例に説明したが、実際の応用においては、必要に応じて上記機能を異なる機能ユニットに割り当てて完成させることができる。すなわち、機器の内部構造を異なる機能ユニットに区分分けして、以上に記述された全部又は部分の機能を完了することができる。また、上記実施例が提供する装置は方法の実施例と同一の発想に属し、その具体的な実現過程は詳しくは方法の実施例を参照することができるため、ここでは再度詳細に説明しない。

40

【0229】

また、本願の1つの例示的な実施例はさらに誤り訂正チップを提供し、該誤り訂正チップは上記量子誤り訂正復号方法を実現することに用いられる。

【0230】

50

また、本願の1つの例示的な実施例はさらに非一時的コンピュータ可読記憶媒体を提供し、該非一時的コンピュータ可読記憶媒体中に少なくとも1つのコンピュータプログラムが記憶されており、該少なくとも1つのコンピュータプログラムがコンピュータ機器のプロセッサによりロードされ且つ実行され、コンピュータに上記量子誤り訂正復号方法を実現させる。

【0231】

一種の可能な実現方式では、上記非一時的コンピュータ可読記憶媒体は読み出し専用メモリ (Read-Only Memory、ROM)、ランダムアクセスメモリ (Random Access Memory、RAM)、読み出し専用光ディスク (Compact Disc Read-Only Memory、CD-ROM)、磁気テープ、フロッピーディスク及び光データ記憶機器等であってもよい。

10

【0232】

また、本願の1つの例示的な実施例はさらにコンピュータプログラム製品を提供し、該コンピュータプログラム製品はコンピュータプログラム又はコンピュータ命令を含み、該コンピュータプログラム又はコンピュータ命令はプロセッサによりロードされ且つ実行され、コンピュータに上記量子誤り訂正復号方法を実現させる。

【0233】

説明する必要がある点として、上記の実施例の紹介においては、主に回転表面コードを例としたが、本願の技術的手段は、高次元トーリックコード (high dimensional toric code)、双曲面コード (hyperbolic code) 及びカラーコード (color code) 等を含むがこれらに限定されない、境界を有するその他の表面コードへも展開し、その他のトポロジー量子誤り訂正コードへも展開することができる。しかし、その他のトポロジー量子誤り訂正コードを使用するときには、症状測定時 (パリティチェック時)、one-shot 又は flag qubit と類似する技術を使用する必要があるため、手段の複雑性を増加させ、且つ誤り訂正性能を低減させるようになる可能性がある。また、上記の実施例は主に超伝導量子計算プラットフォームを例としたが、アルゴリズムが汎用タイプであるため、本願はトポロジー量子誤り訂正コードを使用するいかなる量子計算物理プラットフォームにも適用できる。

20

【0234】

また、エンタングルメント空洞等の方法を使用して Clifford (クリフォード) を実現して量子計算を行うときに、本願の技術的手段は修正を加えて同様に適用できる。結晶格子手術手段 (lattice surgery) を使用して論理量子ゲートを実現するときに、本願の技術的手段は修正を加えて同じく適用できる。論理 Clifford ゲートの実現に対し誤り訂正を行うことに適用できるため、本願も魔法状態蒸留 (magic state distillation) を行うことに同様に適用できる。このように、本願の制御システムはトポロジー量子誤り訂正コードを使用する汎用フォールトトレラント量子計算に対しリアルタイムフィードバック誤り訂正を行うことができる。

30

【0235】

理解すべきであるように、本明細書中に言及される「複数」とは2つ又は2つ以上を指す。「及び／又は」は、関連対象の関連関係を記述し、3種の関係が存在してもよいことを表し、たとえば、A及び／又はBは、Aが単独で存在すること、A及びBが同時に存在すること、及びBが単独で存在することという3種の場合を表してもよい。文字「／」は一般的に、前後の関連対象が一種の「又は」の関係であることを表す。また、本明細書中に記述されたステップの番号は、ステップ間の一種の可能な実行先後順序を例示的に示すものに過ぎず、いくつかのほかに実施例においては、上記ステップは番号の順序に従って実行されなくてもよく、例えば、2つの異なる番号のステップは同時に実行され、又は2つの異なる番号のステップは図示とは反対する順序に従って実行され、本願の実施例ではこれを限定しない。

40

【0236】

上述は単に本願の例示的な実施例であり、且つ本願を制限するのに用いられず、本願の

50

精神及び原則内に作られるいかなる修正、均等物への置換や改良等もいずれも本願の保護範囲内に含まれるべきである。

【符号の説明】

【 0 2 3 7 】

1 1 黒色ドット

1 2 十字

2 1 黒色辺

2 2 丸部分

4 1 セグメント

5 1 論理量子ビット

10

7 1 量子回路

7 2 希釈冷凍機

7 3 制御機器

7 3 a 総制御システム

7 3 b 訂正モジュール

7 4 コンピュータ

2 5 0 訂正復号システム

2 5 1 訂正チップ

2 2 0 1 取得ユニット

20

2 2 0 2 復号ユニット

30

40

50

【図面】

【図 1】

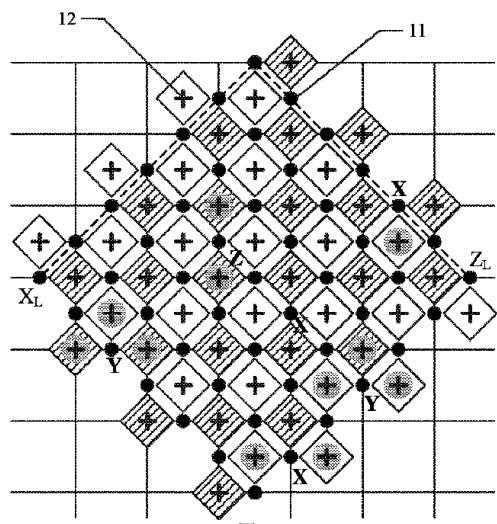


図 1

【図 2】

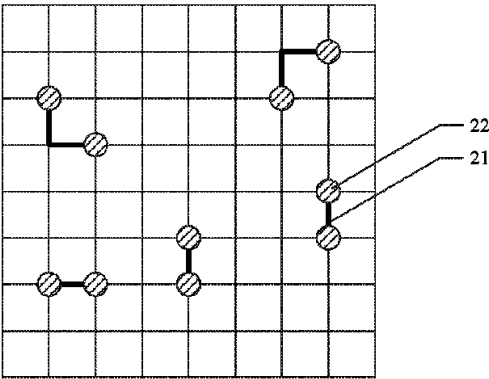


図 2

【図 3】

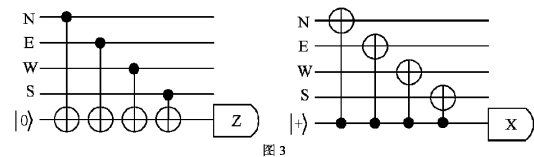
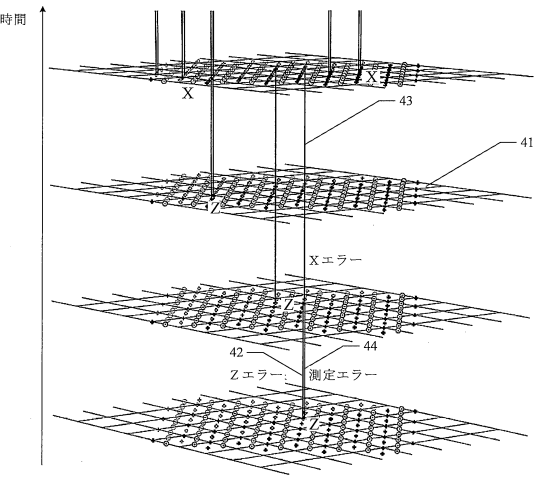


図 3

【図 4】



10

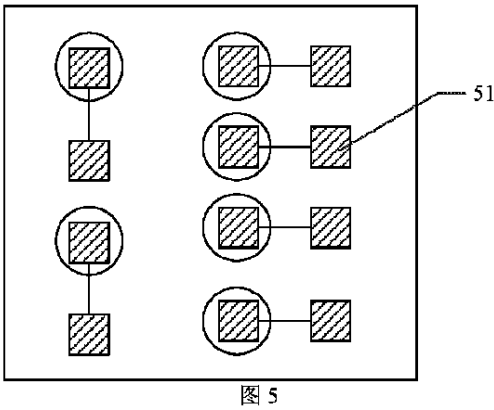
20

30

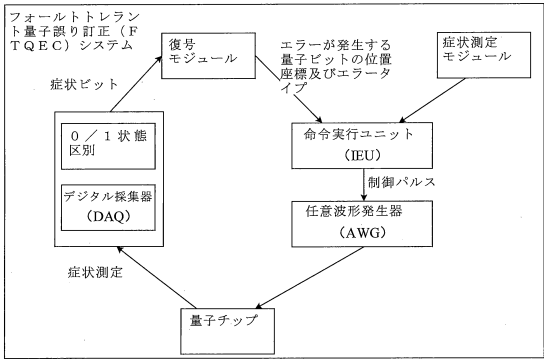
40

50

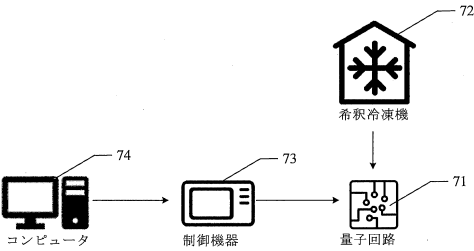
【図 5】



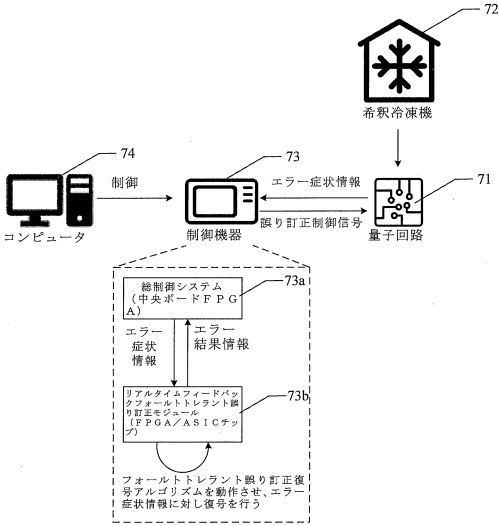
【図 6】



【図 7】



【図 8】



10

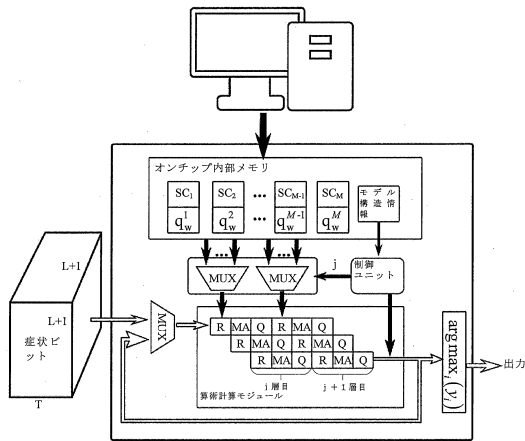
20

30

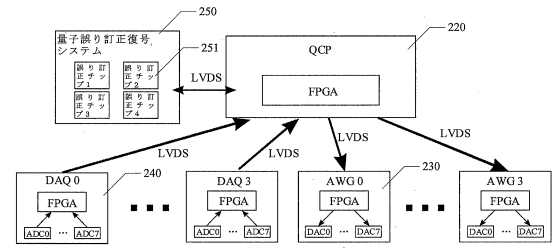
40

50

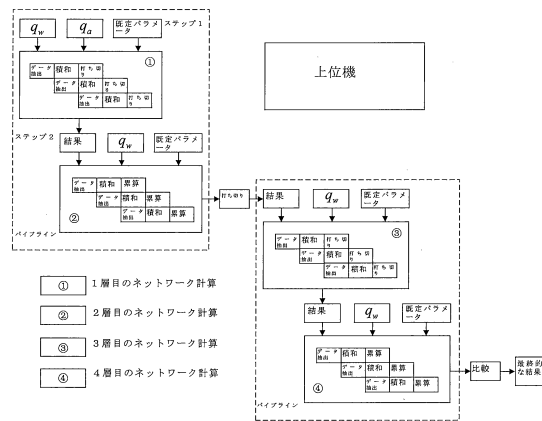
【図 19】



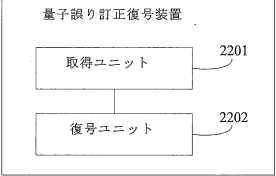
【図 20】



【図 21】



【図 22】



10

20

30

40

50

フロントページの続き

- 中華人民共和国 5 1 8 0 5 7 ▲広▼▲東▼省深▲セン▼市南山区高新区科技中一路 ▲騰▼▲訊▼
大厦 3 5 ▲層▼
- (72)発明者 ▲張▼ 孟禹
中華人民共和国 5 1 8 0 5 7 ▲広▼▲東▼省深▲セン▼市南山区高新区科技中一路 ▲騰▼▲訊▼
大厦 3 5 ▲層▼
- (72)発明者 ▲張▼ ▲華▼良
中華人民共和国 5 1 8 0 5 7 ▲広▼▲東▼省深▲セン▼市南山区高新区科技中一路 ▲騰▼▲訊▼
大厦 3 5 ▲層▼
- (72)発明者 ▲劉▼ 福明
中華人民共和国 5 1 8 0 5 7 ▲広▼▲東▼省深▲セン▼市南山区高新区科技中一路 ▲騰▼▲訊▼
大厦 3 5 ▲層▼
- (72)発明者 ▲張▼ ▲勝▼誉
中華人民共和国 5 1 8 0 5 7 ▲広▼▲東▼省深▲セン▼市南山区高新区科技中一路 ▲騰▼▲訊▼
大厦 3 5 ▲層▼
- 審査官 武田 広太郎
- (56)参考文献 中国特許出願公開第 1 1 1 5 1 0 1 5 8 (CN, A)
国際公開第 2 0 2 0 / 0 6 5 8 7 4 (WO, A1)
米国特許出願公開第 2 0 1 9 / 0 1 7 9 7 9 5 (US, A1)
米国特許出願公開第 2 0 1 9 / 0 0 4 4 5 4 2 (US, A1)
- (58)調査した分野 (Int.Cl., DB名)
G 0 6 N 1 0 / 7 0
G 0 6 F 1 1 / 1 0