量子コンピュータ実現に向けた量子命令セットについて

大音 真由美[†] 中條 拓伯[††]
高田 司郎[†††] 城 和貴[†]

量子コンピュータ開発に向けた研究は始まったばかりで，量子素子・デバイスと，量子チューリング機械上のアルゴリズムの 2 つの興味深い課題が研究されている．これらの研究は非常に重要な基礎研究である．しかし，我々は，量子コンピュータを実現するためには，量子コンピュータ・アーキテクチャ，および，そのシステム・ソフトウェア側からの研究が不可欠であると確信している．そこで，本論文では，量子コンピュータ・アーキテクチャ構築の準備として，量子コンピュータ実現に向けた量子命令セットを提案する．具体的には，まず，現在までに提案されている 5 つの量子アルゴリズムに共通な量子基本操作を抽出する．次に，それら量子基本操作を実現するために，量子ユニットモデルとそのモデル上で実行する量子命令セットを提案する．さらに，この命令セットを上記 5 つの量子アルゴリズムのコーディングに適用することで，提案する量子命令セットの有効性を示す．最後に，今後の課題である量子コンピュータ・アーキテクチャの枠組みの一例を示す．

A Quantum Instruction Set for Real Quantum Computer

Mayumi Oto,[†] Hironori Nakajo,[††] Shiro Takata[†††]
and Kazuki Joe[†]

The research on quantum computers is just getting started and consists of two important issues: 1) quantum elements/devices and 2) algorithms for Quantum Turing Machine. These are quite important and basic research topics on quantum computers. However, we believe that an approach from the computer architecture and the system software side is indispensable to develop a quantum computer. In this paper, we propose a quantum instruction set for a real quantum computer as the first step toward quantum computer architecture research. First, we abstract common primitives of quantum computations through five well-known quantum algorithms. Secondly, in order to execute these quantum computations, we present a quantum unit model and a quantum instruction set for the model. Thirdly, we show the effectiveness of the proposed instruction set by applying it to these five quantum algorithms. Finally, we present a framework for a quantum computer architecture.

1. はじめに

1994 年，Shor の因数分解アルゴリズム[1] の提案を機に，インターネット上のセキュリティ問題など，社会に大きなインパクトを与える可能性を秘めた量子コンピュータに関する研究が注目されている．現在までの量子コンピュータの研究は 2 種類に大別される．量子コンピュータを構成する量子素子・デバイスの研究[2)~6)]，および，量子コンピュータの計算理論である量子チューリング機械の実現アルゴリズムの研究[7)~9)]である．前者は物理系，後者は，計算理論の分野で推進されている重要な基礎研究である．

しかし，これらの研究は，量子コンピュータを現在のノイマン型コンピュータにとって代わる将来のコンピュータ・システムとしてとらえたものではない．現在のコンピュータは，各種ノイマン型素子・デバイス技術とチューリング機械のみで創造されたのではなく，ノイマンがプログラム内蔵方式のシステムを提案したことが，汎用的な利用を可能としていることは，万人が認めるところである．したがって，量子コンピュータを現在のノイマン型コンピュータにとって代わる汎用のコンピュータ・システムとして実現するためには，コンピュータ・アーキテクチャ，および，システム・ソフトウェアの研究アプローチが不可欠かつ急務の研究課題である．そこで，我々は量子素子・デバイスが実現されることを前提に，量子コンピュータ・アーキテク

† 奈良女子大学
Nara Women's University
†† 東京農工大学
Tokyo University of Agriculture & Technology
††† ATR メディア情報科学研究所
ATR Media Information Science Laboratories

チャの構築を目標としている．まず，文献 10) において，汎用的な量子コンピュータの一例を示した．次に，文献 11) において，5 種類のアルゴリズムの具体的プログラミング例を示した．さらに，文献 12) において，それらのプログラムを実行するアーキテクチャの一例を示した．本論文では，上記で報告した内容を総括し，量子ユニットモデルの定義を行う．これによって，量子ユニットモデルと，そのモデル上で実行される量子命令セットとの関係がより明確になる．

本論文では，アーキテクチャ構築の準備として，量子命令セットを提案する．量子命令セットの定義には，現在提案されている 5 つの量子アルゴリズムに共通の量子基本操作をすべて抽出し，量子ユニットモデルとそのモデル上で実行する量子命令セットを定義するという方法を採る．そこで，まず，現在までに提案されている 5 つの量子アルゴリズムから，量子基本操作を抽出する．次に，それら量子基本操作を実現するために，量子ユニットモデルとそのモデル上で実行する量子命令セットを定義する．さらに，この命令セットを上記 5 つの量子アルゴリズムのコーディング例に適用することで，提案する量子命令セットの有効性を示す．最後に，今後の課題である量子コンピュータ・アーキテクチャの枠組みの一例を示す．

以下，2 章では量子基本操作の抽出，3 章では量子命令セットの提案，4 章では量子アルゴリズムのコーディング事例，5 章では量子コンピュータ・アーキテクチャ事例，最後の章ではまとめと今後の課題について，それぞれ述べる．

2. 量子基本操作の抽出

本章では，現在までに提案されている 5 つの量子アルゴリズムから，量子基本操作を抽出する．そのため，まず，これらアルゴリズムの代表格である Shor の因数分解のアルゴリズムから量子基本操作を抽出する．次に，その他の量子アルゴリズムからも共通な量子基本操作を抽出する．

2.1 Shor の因数分解アルゴリズム

Shor の因数分解アルゴリズムは，因数分解したい整数 N と，N と互いに素である任意の整数 x に対して

$$x^r \equiv 1 \ mod \ N$$

を満たす最小の整数となる位数 r を求める算法である．上式は

$$(x^{r/2}-1)(x^{r/2}+1) \equiv 0 \ mod \ N$$

と変形でき，以下のような最大公約数 gcd が解となる．

$$\gcd(x^{r/2}-1, N), \gcd(x^{r/2}+1, N)$$

この量子アルゴリズムは，以下のように提案されている[13]．

(1) 入力 N に対して $2N^2 \le q \le 3N^2$ を満たす q と，$1 < x < N$ を満たす x を選ぶ．

(2) 量子レジスタ R1，R2 を生成する．

(3) R1 に 0 から $q-1$ までの重ね合わせの状態を作る．

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a, 0\rangle$$

ただし，$|R1, R2\rangle$ は，それぞれ量子レジスタ R1，R2 の結合状態を表す．

(4) R2 に $x^a \ mod \ N$ を計算した結果を保存する．

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a, x^a \ mod \ N\rangle$$

(5) R2 を観測する．観測して得た値を a' とする．

$$\frac{1}{\sqrt{\|A\|}} \sum_{a' \in A} |a', k\rangle$$
$$A = \{a' : x^{a'} \ mod \ N = k\}$$
$\|A\|$: 集合 A の要素数

(6) R1 に離散フーリエ変換を行う．

$$\frac{1}{\sqrt{\|A\|}} \sum_{a' \in A} \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} exp(2\pi i a' c/q)|c, k\rangle$$

(7) R1 を観測する．

(8) 得られた値は q/r の定数倍の形をしているので，連分数展開法によって $1/r$ の定数倍を得る．

位数を計算するのに十分な数のサンプルを得るために，以上のステップを繰返し位数 r を決定する．

2.2 量子基本操作の抽出

上記の Shor の因数分解アルゴリズムでは，ステップ (1) と (8) はノイマン型レジスタで閉じたノイマン型の計算である．その他のステップは，量子レジスタを用いた量子基本操作であり，**表 1** のように，量子レジスタの生成，ユニタリー変換，量子算術演算，量子観測などの量子基本操作が抽出できる．

以下，同様に，付録に記述したデータベース検索[14]，最小値検索[15]，中間値の評価[16]，および，複数解の検索[17] などのアルゴリズムでは，Shor の因数分解アルゴリズムで抽出した量子基本操作以外には，データベース検索におけるステップ (2)(a) の条件付きユニタリー変換のみが共通な量子基本操作として抽出で

表 1 量子アルゴリズムから抽出した量子基本操作
Table 1 Quantum calculations abstracted from quantum algorithms.

量子アルゴリズム	ステップ	量子基本操作
Shor の因数分解	(2)	量子レジスタの生成
	(3) (6)	ユニタリー変換
	(4)	量子算術演算
	(5) (7)	量子観測
データベース検索	(2) (a)	条件付きユニタリー変換

きる．

以上，表 1 にまとめたように，代表的な 5 つの量子アルゴリズムからは，量子レジスタの生成，ユニタリー変換，量子算術演算，量子観測，および，条件付きユニタリー変換などの量子基本操作が抽出できた．

3. 量子命令セットの提案

本章では，2 章で抽出した量子基本操作を実現する量子命令セットを提案する☆．まず，量子ユニットモデルを定義し，次に，そのモデル上での量子命令セットの仕様を記述する．

3.1 量子ユニットモデル

量子ユニットモデルは，量子レジスタファイル，量子初期化レジスタファイル，QR セレクタ，および量子 ALU（Arithmetic Logic Unit）で構成する．アルゴリズムの実行前は，すべての量子レジスタは初期化されているものとする．本論文では，量子レジスタの初期化を，量子レジスタ中のすべての qubit を $|0\rangle$ の状態にすることと定義する．

3.1.1 量子レジスタファイル

量子基本操作を行う際の状態数は，解決すべき問題によって異なる．一般に，状態数 2^n は，量子レジスタ中の量子ビット（以下，qubit と呼ぶ）n 個で表現できる．そこで，問題に依存しない量子レジスタのモデルとするために，各量子レジスタは無限個の qubit を持つ．また，各量子レジスタが持つ qubit の個数の値を量子レジスタの先頭に持ち，その量子命令の実行対象となる qubit の個数を表現する．これにより，たとえば，各量子命令のオペランドで qubit の個数指定が省略されたときは，その量子レジスタに最後に設定された qubit の個数を使用する．

任意の 2 行 2 列のユニタリー行列 U は，下記のユニタリー行列[18] を用いて表現される．

$$\exists\theta,\alpha,\beta,\delta\in R,\ U=\Phi(\delta)R_z(\alpha)R_y(\theta)R_z(\beta)$$

$$R_y(\theta)=\begin{pmatrix}\cos\theta/2 & \sin\theta/2\\ -\sin\theta/2 & \cos\theta/2\end{pmatrix}$$

$$R_z(\alpha)=\begin{pmatrix}e^{i\alpha/2} & 0\\ 0 & e^{-i\alpha/2}\end{pmatrix}$$

$$\Phi(\delta)=e^{i\delta}I$$

0 と 1 を表す状態を，それぞれ，$|0\rangle$ と $|1\rangle$ と表現する．θ に関しては，$\cos\theta/2$，$\sin\theta/2$ が，それぞれ，$|0\rangle$ と $|1\rangle$ の振幅比を表す．α と β に関しては，$\alpha+\beta$ が，位相の実数と虚数との比を表すので，$e^{i(\alpha+\beta)/2}$，$e^{-i(\alpha+\beta)/2}$ が，それぞれ，$|0\rangle$ と $|1\rangle$ の位相を表す．δ に関しては，$e^{i\delta}$ が，$|0\rangle$ および $|1\rangle$ 両方の振幅係数を表す U の行列式を 1 にする規格化のための変数である．同様に各 qubit の状態は，$Q(\delta,\theta,\alpha+\beta)$ と表現する．したがって，qubit の状態を $|0\rangle$ と $|1\rangle$ の重ね合わせ状態に変換すると，下記のように表現できる．

$$Q(\delta,\theta,\alpha+\beta)=e^{i\{\delta+(\alpha+\beta)/2\}}\cos\frac{\theta}{2}\,|0\rangle + e^{i\{\delta-(\alpha+\beta)/2\}}\sin\frac{\theta}{2}\,|1\rangle$$

我々が考えているモデルの範囲では，α と β を区別する必要はないため，$\gamma=\alpha+\beta$ と置き換え，以降は，$Q(\delta,\theta,\gamma)$ を用いる．以下，qubit のユニタリー変換例を示す．状態 $|0\rangle=Q(0,0,0)$ に初期化された qubit に，Walsh-Hadamard 変換のユニタリー行列

$$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$$

を適用すると，重ね合わせ状態は

$$Q(\delta,\theta,\gamma):Q(0,0,0)\mapsto Q'(\pi/2,3\pi/2,\pi)$$

のように更新される．ただし，$|0\rangle$ の振幅は $e^{i\pi/2}\times e^{i\pi/2}\times\cos 3\pi/4 = 1/\sqrt{2}$，$|1\rangle$ の振幅は $e^{i\pi/2}\times e^{-i\pi/2}\times\sin 3\pi/4 = 1/\sqrt{2}$ と計算され，この qubit の重ね合わせ状態は $1/\sqrt{2}(|0\rangle+|1\rangle)$ となる．

3.1.2 量子初期化レジスタファイル

本論文で定義する量子初期化レジスタは，アーキテクチャレベルでの存在を前提とするものであって，実装についての詳細はまだ検討されていない．ここでは，量子初期化レジスタの機能の一例を示すにとどめる．

文献 17) において，文献 14) の量子アルゴリズムを k 回実行するという手順があり，文献 14) を 1 回実行するごとに，量子レジスタの初期化が不可欠である．そこで，このような初期化専用の量子初期化レジスタを導入する．量子初期化レジスタは，量子レジスタと同様に無限個の qubit を持ち，スタック構造を形成する．ユニタリー変換など，量子レジスタを更新する量子基本操作には可逆性が要求される．したがって，量

☆ 過去にノイマン型の命令セットは増減を繰り返した．したがって，必要十分な量子命令セットを予想することは困難である．本論文では，現時点での十分性のみを示す．

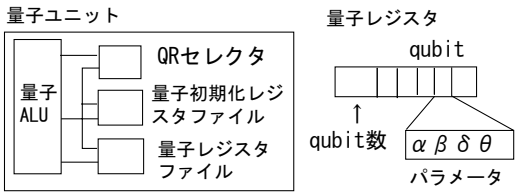

図 1 量子ユニットモデル
Fig. 1 A quantum unit model.

子レジスタの初期化はメモリそのものを 0 にするのではなく，量子レジスタと量子初期化レジスタとのレジスタ番号を交換することで実行する．

3.1.3 QR セレクタ

QR セレクタはノイマン型コンピュータのセレクタと同様に，量子レジスタのレジスタ番号の管理を行う．ノイマン型コンピュータでのレジスタ番号は固定されているが，本論文での QR セレクタはテーブル方式で可変とする．初期化命令が実行されたとき，QR セレクタはまずオペランドで指定された量子レジスタと同じサイズのメモリ領域を量子初期化レジスタに確保し，次にそれらのレジスタ番号を交換する．ここで量子レジスタならびに量子初期化レジスタの内容はまったく観測されないことに注意されたい．これによって，可逆のための情報を失うことなく量子レジスタの初期化を行うことができる．

3.1.4 量子 ALU

量子 ALU は本論文で提案する量子命令セットを実行する．具体的には，量子 ALU は，量子基本操作の対象となる量子レジスタの全 qubit に，オペランドで指定したユニタリー行列を適用する．

以上で定義した量子ユニットモデルを**図 1** に示す．

3.2 量子命令セットの定義

本節では 2 章で抽出した量子基本操作のそれぞれについて，3.1 節で定義した量子ユニットモデル上で実現する量子命令セットの仕様を定義する．

3.2.1 QSetLength

QSetLength は，この命令の実行後に実行される量子命令のオペランドで指定された量子レジスタ内で使用する qubit の個数（以後，量子レジスタの長さと呼ぶ）を指定する命令である．つまり，量子レジスタの先頭にある qubit の個数を設定する命令である．

3.2.2 QExchange

QExchange は，量子レジスタを初期化するための命令である．オペランドで指定された量子レジスタに対して，あらかじめその量子レジスタに QSetLength で設定された qubit の個数分だけ，量子初期化レジスタファイルにメモリ領域を確保する．次に，量子レジスタと量子初期化レジスタとのレジスタ番号を交換する．ただし，同じ量子レジスタに対して 2 回以上初期化命令が実行されるときは，2 回目以降は量子初期化レジスタ内でメモリ領域の確保とレジスタ番号の交換を行う．

3.2.3 QRP

QRP（Quantum Rotate Phase）は，量子レジスタで使用される全 qubit を位相回転するための命令である．qubit のパラメータを，$Q(\delta_0, \theta_0, \gamma_0)$ とし，それに適用するユニタリー行列のパラメータを，$U(\delta, \theta, \gamma)$ とする．このとき，ユニタリー行列の適用後の qubit の状態は，$Q'(\delta_0+\delta, \theta_0+\theta, \gamma_0+\gamma)$ のように表現される．

3.2.4 QRPS

QRPS（Quantum Rotate Phase Selectively）は，条件付きの QRP 命令である．つまり，指定された条件 Cond に満たす状態を探し出し，その状態だけを更新するような qubit のみを位相回転する命令である．

たとえば 2 個の qubit を，それぞれ，A，B とする．この量子レジスタが平坦な重ね合わせ状態にある（すべての状態が等しい確率で観測される）とき，条件 $C(S_i)=1$ を満たす状態の位相のみを ϕ 回転する．今，この条件を満たす状態を S_2 とする．QRPS 命令で量子レジスタの状態が変換される様子は以下の式で表される．

$$\begin{pmatrix} 1 & & & \\ & 1 & & \\ & & e^{i\phi} & \\ & & & 1 \end{pmatrix} \frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ e^{i\phi} \\ 1 \end{pmatrix}$$

変換前の A と B の状態は $1/\sqrt{2}(|0\rangle+|1\rangle)$，すなわち，$Q(\pi/2, 3\pi/2, \pi)$ である．量子 ALU が条件をみたす状態を探し出すと，条件付きユニタリー変換によって，B だけを

$$Q((\pi+\phi)/2, 3\pi/2, \pi+\phi) \mapsto 1/\sqrt{2}(e^{i\phi}|0\rangle+|1\rangle)$$

に変換する．

量子レジスタ全体の状態は，以下の式のようになり，状態 S_2 の位相のみを ϕ 回転する．

$$\begin{aligned} &\frac{1}{\sqrt{2}}|0\rangle\left\{\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\right\} \\ &\quad+\frac{1}{\sqrt{2}}|1\rangle\left\{\frac{1}{\sqrt{2}}(e^{i\phi}|0\rangle+|1\rangle)\right\} \\ &\quad=\frac{1}{2}(|00\rangle+|01\rangle+e^{i\phi}|10\rangle+|11\rangle) \end{aligned}$$

また，文献 15) の条件付き状態の印つけは，QRPS 命令のオペランドで $\theta=\pi$ とし，Q-R_i，Q-R_j を別々

表 2 量子命令セット
Table 2 A quantum instruction set.

量子基本操作	量子命令	ニーモニック	オペランド	量子命令の仕様
量子レジスタの生成	レジスタ長の指定	QSetLength	Q-R$_i$,N-R$_j$	Q-R$_i$* の長さを N-R$_j$** にセットする
	レジスタの初期化	QExchange	I-Reg,Q-R$_i$	I-Reg*** を Q-R$_i$ にコピーして初期化する
ユニタリー変換	位相回転	QRP	Q-R$_i$, θ	Q-R$_i$ で使用する全 qubit の位相を θ 回転する
	条件付き位相回転	QRPS	Cond,Q-R$_i$,Q-R$_j$†, θ	条件 Cond を満たす Q-R$_i$ に対応する Q-R$_j$ の位相のみ θ 回転する
量子算術計算	加算演算	QAdd	Q-R$_i$,Q-R$_j$	Q-R$_i$ と Q-R$_j$ の加算結果を Q-R$_i$ に保存する
	乗算演算	QMultiply	Q-R$_i$,Q-R$_j$	Q-R$_i$ と Q-R$_j$ の乗算結果を Q-R$_i$ に保存する
	指数演算	QExp	Q-R$_i$,Q-R$_j$,Q-R$_k$	Q-R$_j$ の Q-R$_k$ 乗を Q-R$_i$ に保存する
	剰余演算	QMod	Q-R$_i$,Q-R$_j$,Q-R$_k$	Q-R$_i$ を Q-R$_j$ で剰余した商を Q-R$_k$，余りを Q-R$_i$ に保存する
量子観測	観測	QObserve	Q-R$_i$,N-R$_j$	Q-R$_i$ の観測結果を N-R$_j$ に保存する
—	（ノイマン型命令）	CPhase	Matrix,N-R$_i$	行列 Matrix の回転する位相を計算して N-R$_i$ に保存する

*量子レジスタ i，**ノイマン型レジスタ j，***量子初期化レジスタ，†Q-R$_j$ が Q-R$_i$ と同じ場合は省略可.

に指定することによっても実現される．

3.2.5 量子算術演算

QAdd，QMultiply，QExp，および，QMod は，量子レジスタに対する量子算術計算であり，それぞれ，加算演算，乗算演算，指数演算，および，剰余演算を行うための命令である☆．

3.2.6 QObserve

QObserve は，量子レジスタを量子観測して測定値を得るための命令である．観測された測定値は，オペランドで指定されたノイマン型レジスタに転送する．

たとえば，ある量子レジスタは，n 個の qubit を持ち，i 番目の qubit の重ね合わせ状態が $a_i|0\rangle + b_i|1\rangle$ の場合，量子レジスタの状態 Ψ は，以下の式で表される．

$$\Psi = (a_0 a_1 \ldots a_{n-1})|0\rangle + (b_0 a_1 \ldots a_{n-1})|1\rangle + \ldots + (b_0 b_1 \ldots b_{n-1})|n-1\rangle$$

したがって，$|0\rangle, |1\rangle, \ldots, |n-1\rangle$ が観測される確率は，それぞれ，$|a_0 a_1 \ldots a_{n-1}|^2, |b_0 a_1 \ldots a_{n-1}|^2, \ldots, |b_0 b_1 \ldots b_{n-1}|^2$ となる．ただし，$|x_0 x_1 \ldots x_{n-1}|^2$ は，$|2^0 x_0 + 2^1 x_1 + \ldots + 2^{n-1} x_{n-1}\rangle$ が観測される確率である．

3.2.7 CPhase

CPhase は，与えられた 2 行 2 列のユニタリー行列の位相回転パラメータ（δ，α，β，および θ）を，ノイマン型で求めるためのアセンブラのマクロ命令である．

具体的には，ユニタリー行列の成分を $a_{00}, \ldots, a_{11}$ とすると，

$$\begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} = e^{i\delta} \cdot \begin{pmatrix} e^{i(\alpha+\beta)/2}\cos\theta/2 & e^{i(\alpha-\beta)/2}\sin\theta/2 \\ -e^{-i(\alpha-\beta)/2}\sin\theta/2 & e^{-i(\alpha+\beta)/2}\cos\theta/2 \end{pmatrix}$$

の連立方程式を解く．

以上，定義した量子命令セットをまとめると，**表 2** のようになる．

3.2.8 量子コピー命令

参考までに，本論文では用いなかった量子コピー命令（QCopy）について述べる．量子コピー命令は，qubit そのものをコピーするのではなく，オペランドで指定されたコピー元とコピー先の量子レジスタについて，アルゴリズムの開始時に量子 ALU が同じサイズの領域を確保し，それぞれにまったく同じ操作を行うことで実現できる．

4. 量子アルゴリズムのコーディング事例

この章では，我々が提案した量子命令セットの有効性を示すために，Shor の因数分解アルゴリズムを，量子命令セットを用いたコーディング例を示す．残りの 4 つについては，付録にて記述する．

ただし，ノイマン型命令のコードについては，本論文の範囲外として，機能概要を { } で囲った日本語で記述する．また，コメント行は ; で始まる．以下，Shor の因数分解アルゴリズムのコーディングを記述する．

今，整数 N の因数分解を求める問題とする．その整数の入力を N とする．

```
Read N
Load N, N-R_N
```

☆ 我々は，量子論理演算（量子 NOT など）を，量子演算（量子 ADD など）を構成する量子回路と見なしており，その観点から量子レジスタを単位とする量子命令を提案している．

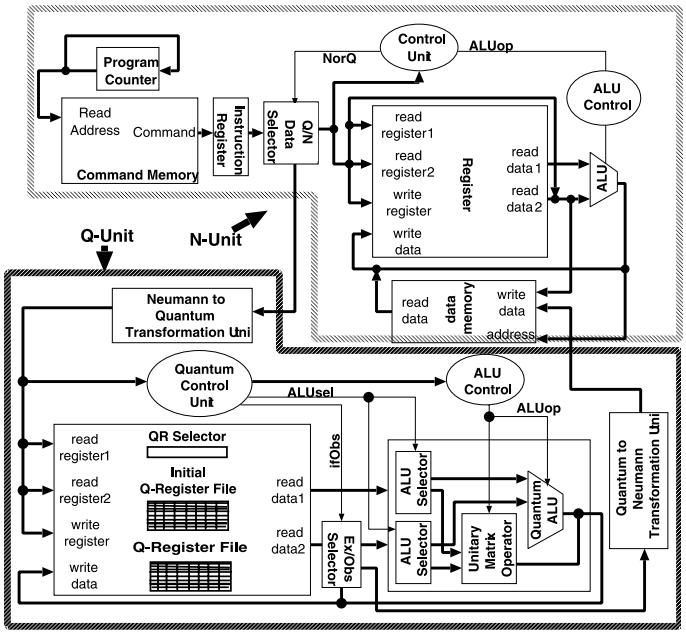

図 2 プロセッサの内部ブロック図
Fig. 2 Block diagram of processor.

{$2^n \geq N$ を満たす最小の n を計算する }
Store n,N-R$_n$
L1 {$2N^2 \leq q \leq 3N^2$ を満たす q を選ぶ }
Store q,N-R$_q$
{$1 < x \leq N-1$ を満たす x を選ぶ }
Store x,N-R$_x$
{gcd $(N, x) \neq 1$ ならば x を出力して終了 }
Read M
CPhase M, N-R$_M$
; 行列 M のパラメータを計算して
N-R$_M$ に保存
{ 以下 L2 まで $O(\log(q))$ 回繰り返す }
QSetLength N-R$_n$,Q-R$_1$
QSetLength N-R$_n$,Q-R$_2$
QSetLength N-R$_n$,Q-R$_3$
QExchange I-Reg,Q-R$_1$
QExchange I-Reg,Q-R$_2$
QExchange I-Reg,Q-R$_3$
QRP N-R$_M$,Q-R$_1$
QExp Q-R$_2$,Q-R$_1$,N-R$_x$
QMod Q-R$_2$,N-R$_N$,Q-R$_3$
QObserve Q-R$_2$,N-R$_k$
; Q-R$_2$ を観測
QRP N-R$_M$,Q-R$_1$
QObserve Q-R$_1$,N-R$_k$
; Q-R$_1$ を観測
L2 {N-R$_k$ を連分数展開法でサンプルを得る }
{gcd$(x^{r/2}-1, N)$ と gcd$(x^{r/2}+1, N)$ の両方が 1 でなければ，それらを解とする．
そうでなければ，L1 へ戻る }
Stop

5. 量子コンピュータ・アーキテクチャ事例

先に定義した量子ユニットモデルで，量子命令セットを実行するための量子コンピュータ・アーキテクチャ事例を**図 2** [12] に示す☆．

プロセッサは，量子命令を処理する量子ユニット（Q-unit）と通常の命令を処理するノイマン型ユニット（N-unit）に大きく分けられる．N-unit にある制御ユニット（Control Unit）は，Quantum/Neumann データセレクタ（Q/NDS）を制御し，命令レジスタから送られてきた命令が量子型なのかノイマン型なのかを判別する．Q/NDS によって量子命令は量子ユニットへ，通常の命令はノイマン型ユニットへ送られる．

量子ユニットに送られた命令は，ビットを量子ビットに変換するユニット QtoNTU（Quantum to Neumann Transformation Unit）を通り，オペレーション

☆ 実装に関する詳細は，量子アルゴリズムからの要請と，実装上からの要請とを満たす必要がある．現在はともに研究段階にあるため，本論文では何らかの方法で実装が可能であるという立場をとる．

コードは量子制御ユニット（Quantum Control Unit）へ，オペランドは量子レジスタへ送られる．

ALU Control は制御線 ALUop によって，Unitary Matrix Operator と Quantum ALU のうち使用するユニットにフラグビットを立てる．ALU Selector は量子制御ユニットによって制御され，フラグビットを読んでユニットを判別しデータを提供する．量子制御ユニットからの制御信号 ifObs が Ex/Obs Selector に作用し，初期化命令の場合は QR セレクタの指示に従って量子レジスタおよび量子初期化レジスタにアクセスされる．観測命令の場合は，量子ビットをビットに変換するユニット NtoQTU（Neumann to Quantum Transformation Unit）を通り，ノイマン型レジスタに結果が格納される．

6. 結　　論

本論文では，5 つの代表的アルゴリズムから量子基本操作を抽出し，それら量子基本操作を実現するための，量子ユニットモデルと，そのモデル上で実行する量子命令セットを定義した．次に，これらアルゴリズムのコーディング例を示すことで量子命令セットの有効性を示した．そして，量子命令を実行する量子コンピュータ・アーキテクチャを例示し，量子初期化レジスタの仕様の一例をあげた．その実装の可能性については検討の余地がある．他のアーキテクチャとして，量子スタック・マシン方式との親和性を検討することを今後の課題とする．

また，量子コンピュータ・アーキテクチャのシミュレータを構築するためには，提案した量子ユニットモデル上での量子命令セットの計算量は重要な評価項目であり，今後の課題とする．特に，量子観測命令や以下で述べるユニタリー行列のテンソル積の計算量は重要である．Grover のデータベース検索アルゴリズムに記述されている拡散変換は N 行 N 列であり，この行列をノイマン型ユニットが CPhase 命令で実行するには，2 行 2 列のユニタリー行列のテンソル積に分解する必要がある．このときの計算量は，量子ユニットがその他の命令を実行する計算量よりも多いと予測される．さらに，この計算量からノイマン型ユニットの並列化が検討課題となることが予想され，今後の重要な課題である．

参 考 文 献

1) Shor, P.: Algorithms for Quantum Computation: Discrete Logarithms and Factoring, *Proc. 35th Annual Symposiun on Foundations of Computer Science*, pp.124–134 (1994).
2) Lloyd, S.: A Potentially Realizable Quantum Computer, *Science*, Vol.261, pp.1569–1571 (1993).
3) Cirac, J. and Zollar, P.: Quantum Computations with Cold Trapped Ions, *Physical Review Letters*, Vol.74, pp.4091–4094 (1995).
4) Turchette, Q., Hood, C., Lange, W., Mabuchi, H. and Kimble, H.: Measurement of Conditional Phase Shifts for Quantum Logic, *Physical Review Letters*, Vol.75, No.25, pp.4710–4713 (1995).
5) Cory, D., Fahmy, A. and Havel, T.: Nuclear Magnetic Resonance Spectoroscopy: An Experimentally Accessible Paradigm for Quantum Computing, *Proc. 4th Workshop on Physics and Computation*, pp.87–91 (1996).
6) Gershenfeld, N., Chuang, I. and Lloyd, S.: Bulk Quantum Computation, *Proc. 4th Workshop on Physics and Computation*, p.134 (1996).
7) Deutsch, D.: Quantum Theory, the Church-Turing Principle, and the Universal Quantum Computer, *Proc. Royal Society London*, Vol.A400, pp.97–117 (1985).
8) Lloyd, S.: Universal Quantum Simulators, *Science*, Vol.273, pp.1073–1078 (1996).
9) Zalka, C.: Efficient Simulation of Quantum Systems by Quantum Computers, *Proc. Royal Society London*, Vol.A454, pp.313–322 (1998).
10) 大音真由美，中條拓伯，城　和貴：汎用量子コンピュータ・アーキテクチャの構想，情報処理学会シンポジウムシリーズ新しい計算パラダイムシンポジウム 2000 論文集，Vol.2000, No.16, pp.77–80 (2000).
11) Oto, M., Nakajo, H. and Joe, K.: A Possible Instruction Set for Quantum Computer Architectures, *The 2001 International Conference on Parallel and Distributed Processing Techniques and Applications*, Vol.3, pp.1221–1227 (2001).
12) 古屋良二郎，大音真由美，中條拓伯，城　和貴：量子コンピュータの命令セットアーキテクチャの一提案とそのシミュレータの構想，第 43 回プログラミングシンポジウム報告集，pp.173–184 (2002).
13) C.P. ウィリアムズ，S.H. クリアウォータ（著），西野哲朗，荒井　隆，渡邉　昇（訳）：量子コンピューティング，Springer (2000).
14) Grover, L.: A fast quantum mechanical algorithm for database search, *Proc. 28th Annual ACM Symposium on the Theory of Computing*, pp.212–219 (1996).
15) Durr, C. and Høyer, P.: A quantum algorithm for finding the minimum, quant-ph/9607014, Los Alamos preprint archive (1996).

16) Grover, L.: A fast quantum mechanical algorithm for estimating the median, quant-ph/960701, Los Alamos preprint archive (1996).
17) Boyer, M., Brassard, G., Høyer, P. and Tapp, A.: Tight bounds on quantum searching, *Proc. 4th Workshop on Physics and Computation*, pp.36–43 (1996).
18) 上坂吉則：量子コンピュータの基礎数理，コロナ社 (2000).

付　　録

付録では，1) Grover のデータベース検索アルゴリズム[14]，2) Durr らの最小値検索アルゴリズム[15]，3) Grover の中間値を評価するアルゴリズム[16]，および，4) Boyer らのアルゴリズム[17] について，それぞれ，そのアルゴリズムとコーディング事例を記述する．

A.1 データベース検索

N 個の要素から与えられた条件を満たすただ 1 つの解を，$O(\sqrt{N})$ のステップで 1/2 の確率で探し出す．ノイマン型アルゴリズムでは $O(N/2)$ のステップが必要である．

N 個の要素を S_1 から S_N までラベル付けした量子レジスタを重ね合わせ状態にして，与えられた条件を満たす状態 S_ν の振幅だけを増幅する．

(1) 量子レジスタ R1 に 0 から $N-1$ までの重ね合わせ状態を入力する．
(2) 以下，(a)，(b) を $O(\sqrt{N})$ 回繰り返す．
　(a) 任意の状態 S に対して，条件付き位相回転を行う．
　　$C(S)=1$ のとき，位相を π 回転する．
　　$C(S)=0$ のときは，何もしない．
　(b) 拡散変換 D を適用する．ただし，D は次のように定義される．
　　$D_{ij}=2/N$, if $i \neq j$
　　$D_{ii}=-1+2/N$
(3) R1 を観測する．

以下，このアルゴリズムのコーディング例を記述する．

```
　Read N
　Load N, N-R_N
　{2^n ≥ N を満たす最小の n を計算する }
　Store n, N-R_n
L1　QSetLength Q-R_1, N-R_n
　QExchange Q-R_1, I-Reg
　Read M
　CPhase M, N-R_M
　QRP Q-R_1, N-R_M
　Read D
　CPhase D, N-R_D
　{ 以下 L2 まで O(√N) 回繰り返す }
　QRPS "C(Q-R_1)=1", Q-R_1, π
L2　QRP Q-R_1, N-R_D
　QObserve Q-R_1, N-R_k
　; Q-R_1 を観測
　{ もし C(N-R_k) = 1 でなければ L1 へ戻る }
　Stop
```

A.2 最小値検索

ソートされていない $T[0]$ から $T[N-1]$ までの N 個の要素から，$T[y]$ が最小であるような y を探し出す．

まず，任意の整数を選び，それを閾値 y とする．$T[y]$ より小さい複数の $T[i]$ を選んだとき，i に相当する mark 用レジスタの表に 1 をセットする．

文献 17) によって mark 用レジスタから複数解を探す．R1 を観測して，得られた値 y' が条件を満たせば，y' を新たな閾値とする．これを閾値がリストの最小値になる確率が十分大きくなるまで繰り返す．

(1) 0 から $N-1$ までの任意の y を選び，R2 に保存する．
(2) すべての実行時間が $22.5\sqrt{N}+1.4lg^2N$☆を超えるまで以下繰り返す．
　(a) R1 に 0 から $N-1$ までの重ね合わせの状態を入力する．$T[j]<T[y]$ となるすべての j に対応する mark 用レジスタの表に 1 を入力する．
　(b) 文献 17) を適用．mark 用レジスタの表に入力されている 1 を探す．
　(c) R1 を観測する．得られた値 y' が，$T[y']<T[y]$ ならば，y を y' に更新して (2) へ戻る．

以下，このアルゴリズムのコーディング例を記述する．

```
　Read N
　Load N, N-R_N
　{2^n ≥ N を満たす最小の n を計算する }
　Store n, N-R_n
　{0 ≤ y ≤ N − 1 を満たす y を選ぶ }
　Store y, N-R_y
　Load M
　CPhase M, N-R_M
　{ すべての実行時間が 22.5√N + 1.4lg^2 N
　を超えるまで，以下 L3 までを繰り返す }
　QSetLength Q-R_1, N-R_n
```

☆ lg は 2 進対数

QExchange Q-R$_1$, I-Reg
QRP Q-R$_1$,N-R$_M$
{N-R$_m$ を生成する }
QRPS “T(Q-R$_1$) < T(N-R$_y$),Q-R$_1$, N-R$_m$”
文献 17) を適用する
QObserve Q-R$_1$, N-R$_k$
L3 { もし T(N-R$_k$ < T(N-R$_y$) ならば
N-R$_k$ を N-R$_y$ にセットする }
Stop

A.3 中間値の評価

N 個の要素の中から，ϵ $(0.1\epsilon_0 < |\epsilon| < \epsilon_0)$ の精度で中間値を求める．

閾値 μ より小さい値を持つ状態の振幅だけを増幅し観測する．それを繰り返して得たサンプルを元に ϵ を評価して中間値を求める．

(1) 以下 (a) から (e) までを $O(1/\theta^2)$ 回繰り返す．
- (a) R1 に 0 から $N-1$ までの重ね合わせ状態を入力する．
- (b) 任意の状態 S に対して，条件付き位相回転を行う．$V(S) > \mu$ のとき，位相を $\pi/2$ 回転する．
- (c) シフト変換 S を適用する．S は以下のように定義される．
  $S_{pq} = 1/N + i/N$, if $p \neq q$;
  $S_{pp} = 1/N - i/N(N-1)$
- (d) 以下 (i) から (iv) までを $O(1/\epsilon_0)$ 回繰り返す．
  - (i) 任意の状態 S に対して，条件付き位相回転を行う．$V(S) < \mu$ のとき，位相を π 回転する．
  - (ii) 拡散変換 D を適用する．
  - (iii) 任意の状態 S に対して，条件付き位相回転を行う．$V(S) > \mu$ のとき，位相を π 回転する．
  - (iv) 拡散変換 D を適用する．
- (e) R1 を観測する．

以下，このアルゴリズムのコーディング例を記述する．

Read N
Load N, N-R$_N$
{$2^n \geq N$ を満たす最小の n を計算する }
Store n, N-R$_n$
{ 以下 L4 までを $O(1/\theta^2)$ 回繰り返す }
QSetLength Q-R$_1$, N-R$_n$
QExchange Q-R$_1$, I-Reg
Load μ
Load ϵ_0
Load θ
Load M
CPhase M, N-R$_M$
QRP Q-R$_1$, N-R$_M$
QRPS “V(Q-R$_1$) > μ”, Q-R$_1$, $\pi/2$
Load S
CPhase S, N-R$_S$
Load D
CPhase D, N-R$_D$
QRP Q-R$_1$, N-R$_S$
; シフト変換
{ 以下 L5 までを $O(1/\epsilon_0)$ 回繰り返す }
QRPS “V(Q-R$_1$) < μ”, Q-R$_1$, π
QRP Q-R$_1$, N-R$_D$
; 拡散変換
QSPR “V(Q-R$_1$) > μ”, Q-R$_1$, π
QRP Q-R$_1$, N-R$_D$
Branch L5
Branch L4
QObserve Q-R$_1$, N-R$_k$
Stop

A.4 複数解の検索

解の数が不明の N 個の要素の中から，解を探し出す．条件 $T[i] = x$ を満たす状態の振幅だけを増幅し，観測を行い任意の解を 1 個取り出す．それを繰り返すことで，複数解を得る．

(1) $m = 1, \lambda = 6/5$ に初期設定する☆．
(2) k $(0 \leq k \leq m-1)$ をランダムに選ぶ．
(3) 文献 14) を k 回繰り返す．
(4) レジスタを観測して i を得る．
(5) $T[i] = x$ なら終了する．$T[i] \neq x$ なら，m を $min(\lambda m, \sqrt{N})$ にセットして (2) へ戻る．

以下，このアルゴリズムのコーディング例を記述する．

Store 1, N-R$_m$
Store 6/5, N-R$_\lambda$
L7 {k をランダムに選ぶ }
Store k, N-R$_k$
データベース検索アルゴリズム[14] を k 回繰り返す
{ もし観測した値 i が $T(i) = x$ を満たすなら i を表示して終了する．

☆ m は，k の上限を設定する変数で，step5 で更新される．λ は，$1 < \lambda \leq 4/3$ を満たす任意の数でよい．

そうでなければ m を $\min(\lambda m, \sqrt{N})$ に
セットして L7 へ戻る }
Stop

(平成 14 年 1 月 31 日受付)
(平成 14 年 4 月 7 日再受付)
(平成 14 年 5 月 7 日採録)

大音真由美（学生会員）

1998 年奈良女子大学理学部物理学科卒業．2000 年同大学大学院人間文化研究科博士前期課程修了．現在同大学院博士後期課程在学中．量子コンピュータ・アーキテクチャの研究に興味を持つ．

中條　拓伯（正会員）

1961 年生まれ．1985 年神戸大学工学部電気工学科卒業．1987 年同大学大学院工学研究科修了電子工学専攻．1989 年神戸大学工学部システム工学科（後に情報知能工学科）助手を経て，現在，東京農工大学工学部情報コミュニケーション工学科助教授．工学博士．1998 年より 1 年間 Illinois 大学 Urbana-Champaign 校 Center for Supercomputing Research and Development（CSRD）にて，Visiting Research Assistant Professor．プロセッサアーキテクチャ，分散共有メモリ，クラスタコンピューティングに関する研究に従事．電子情報通信学会，IEEE CS 各会員．

高田　司郎（正会員）

1979 年大阪大学基礎工学部情報工学科卒業．1993 年奈良先端科学技術大学院大学情報科学研究科前期課程入学．1999 年同科後期課程修了．1979 年（株）CSK 入社．1993 年（株）けいはんな入社．1999 年（株）ATR 知能映像通信研究所入所．2001 年（株）ATR メディア情報科学研究所客員研究員，現在に至る．工学博士．人工知能，コミュニケーション，合理的エージェント，形式的仕様記述等に興味を持つ．人工知能学会，日本ソフトウェア科学会，日本ロボット学会，IEEE 各会員．

城　　和貴（正会員）

大阪大学理学部数学科卒業．日本 DEC，ATR 視聴覚研究所（日本 DEC より出向),（株）クボタ・コンピュータ事業推進室で勤務．1993 年奈良先端科学技術大学院大学情報科学研究科博士前期課程入学，1996 年同研究科後期課程修了，同年同研究科助手．1997 年和歌山大学システム工学部情報通信システム学科講師，1998 年同学科助教授．1999 年奈良女子大学理学部情報科学科教授．工学博士．画像処理，文字認識，ニューラルネット，並列計算機アーキテクチャ，自動並列化コンパイラ，並列計算機の解析モデル，視覚化等の研究に従事．IEEE 会員．