

シミュレーション試行を前提とした量子コンピュータ・アーキテクチャ

古屋 良二郎[†] 大音 真由美^{††}
城 和 貴^{†††} 中 條 拓 伯[†]

近年、量子コンピューティングに関する研究が活発になっている。研究の潮流としては、効率的量子アルゴリズムに関する研究と実験物理学の領域において量子デバイスの実現を目指す研究に大別される。我々は、将来の量子コンピュータシステムの実現を想定とし、汎用的な量子コンピュータの抽象モデルを探索している。本論文では、シミュレーション試行を前提とした量子コンピュータ・アーキテクチャを提案し、量子命令セットおよび内部アーキテクチャを定義した。この抽象モデルは、種々の量子命令実行のシミュレーションを行う際の基礎設計部分となるものである。

Quantum Computer Architecture with the Intention of Trying Simulation

RYOUJIROU FURUYA[†], MAYUMI OTO^{††}, KAZUKI JOE^{†††}
and HIRONORI NAKAJO[†]

Quantum Computing is currently a hot research topic. In this research field, there are two main streams; one is the research on new efficient quantum algorithms, and the other is on implementation of quantum devices in the domain of experimental physics. On the supposition that a quantum computer system could be realized in the near future, we are investigating an abstract model of a general-purpose quantum computer. In this paper, we propose a quantum computer architecture and define a quantum instruction set and an internal architecture of the processor with a view to try simulation. This abstract model is a fundamental design part in conducting simulation of executing quantum instructions.

1. はじめに

近年、これまでのコンピュータに取って代わる新しい計算モデルとして量子力学的原理に基づく量子コンピューティングに関する研究が活発になっている。量子コンピューティングに関する研究が始まったのは、Feynman が量子力学的原理に基づくコンピュータを示唆したことから、1985 年に Deutsch が、量子コンピュータの数学的モデルである量子 Turing 機械を定式化したことによる。近年になって研究が活発になったのは、現在のコンピュータが電子デバイスの高速化・高密度集積化に伴って古典物理学の限界が指摘され、量子力学の適用が議論されるようになった背景をもとにしている。また、1994 年に Shor が大きな整数を量子

Turing 機械上で多項式時間内に素因数分解する量子アルゴリズムを考案¹⁾²⁾したことから Grover が高速に動作するデータベース検索アルゴリズムを提案³⁾したことによってブレークスルーがもたらされたことによる。現在の量子コンピューティング関連研究の潮流としては、効率的な量子アルゴリズムを模索する量子計算理論研究と、量子デバイスの実現を目的とする実験物理研究に大別される。

我々は、これらの手法の中間を補い各研究を関連付けることを目指し、汎用的な量子コンピュータ・アーキテクチャの抽象モデルを提案する。汎用的なシステムとして実現するためには、入力・出力・記憶・プロセッサ等の構成要素を抽象化する必要がある。現在のコンピュータも、これらの構成要素がハード・ソフト両面において一連の階層から構成される抽象化原理に基づいて設計されている。命令セットもまた、ハード・ソフト間の抽象的なインタフェースであり、コンピュータ・システムの複雑性に対応する上で重要な構成要素となる。従って、これらの相互作用をシミュレーションすることにより、最終的にプログラム化できる量子コンピュータ・モデルを明らかにすることができると考えた。

量子コンピュータのシミュレーション・モデルに関する研究の現状において、量子コンピュータ・シミュレータがいくつか実現されているが⁴⁾⁵⁾、それらは量子回路モデルレベルのシミュレータであり、量子コンピュータがどのような動作をして計算を行うかを理解

[†] 東京農工大学 工学部 情報コミュニケーション工学科

Department of Computer, Information and Communication Sciences, Faculty of Technology, Tokyo University of Agriculture and Technology, {furuya, nakajo}@nj.cs.tuat.ac.jp

^{††} 奈良女子大学大学院 人間文化研究科

Graduate School of Human Culture, Nara Women's University, oto@ics.nara-wu.ac.jp

^{†††} 奈良女子大学 理学部 情報科学科

Department of Information & Computer Sciences, Faculty of Sciences, Nara Women's University
joe@ics.nara-wu.ac.jp

する手助けになるようなものである。

我々が提案する汎用的な量子コンピュータ・アーキテクチャの抽象モデルは、シミュレーション試行を前提とした基礎設計部分である。

第2章は量子命令セットの定義とその説明を述べ、第3章は汎用的な量子コンピュータ・アーキテクチャの構成ユニットを示す。第4章は個々の量子命令の流れを説明し、最後にまとめとしてシミュレータの提案と今後の課題について述べる。

2. 汎用的量子コンピュータ命令セット

表1に、本稿で提案する汎用的量子コンピュータ・アーキテクチャにおける量子ユニット (Q-Unit: Quantum Unit) で使用する量子命令セット (QIS: Quantum Instruction Set) を示す。これは、代表的な量子アルゴリズム¹⁾等を検証した結果により、実現するために必要とされる機能を全て抽出し、量子命令セットとして定義したものである。

表1において、オペランドに指定されているものは、それぞれ以下のものを指す。

- Q-Ri, Q-Rj, Q-Rk ⇒ 量子レジスタ
- I-Reg. ⇒ 量子初期化レジスタ
- N-Ri, N-Rj ⇒ ノイマン型レジスタ

命令	オペランド	内容
QExchange	I-Reg., Q-Ri	Q-Ri ← I-Reg.
QObserve	Q-Ri, N-Rj	N-Rj ← Q-Ri
QSetLength	Q-Ri, N-Rj	N-Rj ← length(Q-Ri)
QMultiply	Q-Ri, Q-Rj	Q-Ri ← Q-Ri mul Q-Rj
QMod	Q-Ri, Q-Rj, Q-Rk	商: Q-Rk ← Q-Ri mod Q-Rj, 余: Q-Ri ← Q-Ri mod Q-Rj
QRPS	Cond, Q-Ri, Q-Rj, θ	Condを満たすQ-Riの状態に対応するQ-Rjの位相を θ 回転
QRP	Q-Ri, θ	Q-Riで使用する全てのqubitの位相を θ 回転
QAdd	Q-Ri, Q-Rj	Q-Ri ← Q-Ri add Q-Rj
QExp	Q-Ri, Q-Rj, Q-Rk	Q-Ri ← Q-Ri ^{Q-Rk}
CPhase	Matrix, N-Ri	N-Ri ← Matrixの回転位相

2.1 データ移動系命令

2.1.1 QExchange 命令と量子初期化レジスタ (Initial Q-Register) の構造

QExchange 命令は量子初期化レジスタを使用する。QExchange 命令は量子初期化レジスタの値を量子レジスタにコピーすることで量子レジスタの初期化を実現するものであるが、量子初期化レジスタが $|0\rangle$ を無限に持つレジスタと仮定しているため、ハードウェアの構成という観点から限界があった。量子コンピュータ自体、無限次元の複素線形ベクトル空間とみなせるが、システムは何らかの物理的なもので実現されることから、レジスタを無限長と仮定するのは困難である。

量子初期化レジスタは、 $|0\rangle$ を無限にもつレジスタで

ありスタック構造をなしている。QExchange 命令が処理されると、スタックポインタがインクリメントされる。Q-Register と initial-Q-Register とが1対1に対応しているため可逆な命令となる (図1左部)。しかし、このままでは initial-Q-Register が無限長となってしまうので、その対処策としてスタックポインタを固定し、initial-Q-Register はリードアクセスのみ許すものとする (図1右部)。従って、擬似的に $|0\rangle$ を無限にもつレジスタと見せかけることによって上記の問題点を解決し、また命令の可逆性は失われない。

量子命令に可逆性を考慮しない場合は、量子初期化レジスタは必要でない。量子レジスタに $|0\rangle$ をストアすることによって QExchange 命令を実現することができる。

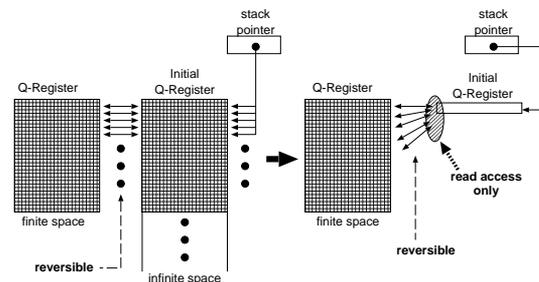


図1 量子初期化レジスタの構造

2.1.2 QObserve, QSetLength 命令

QObserve 命令は量子レジスタの値をノイマン型レジスタへ移すことによって量子計算結果を得るための命令である。QObserve 命令によって量子計算結果の観測が行われるものと仮定する。QSetLength 命令は、それ自身が直接データ移動を担う命令ではないが、QExchange 命令による量子レジスタの初期化に伴い、量子計算中に使用される量子レジスタのビット長をノイマン型レジスタに指定するための命令である。

2.2 算術論理演算系命令

表1の中で、算術論理演算を行う命令は QMultiply, QMod, QRPS, QRP, QAdd, QExp, CPhase である。

QMultiply 命令は、量子レジスタ間の乗算を行い、結果を量子レジスタに格納する命令である。QMod 命令は、量子レジスタ間で剰余演算を行い、結果を量子レジスタに格納する命令である。QRPS 命令は、条件 cond を満たす量子レジスタの状態に対応する量子レジスタの位相を θ 回転させる命令である。QRP 命令は、指定した量子レジスタ内で使用する全ての量子ビットの位相を θ の値だけ位相を回転する命令である。QAdd, QExp 命令は、それぞれ加算、指数演算を行うための命令である。

QRP 命令は、ノイマン型ユニットで使う CPhase 命令と組み合わせることによって回転命令を一般化している。CPhase 命令によって Matrix の回転する位相が計算され、ノイマン型レジスタに値が格納される。 $|0\rangle$ 状態にある1量子ビットを重ね合わせ状態に発展させる場合は、CPhase 命令によって回転角 $\theta = \frac{\pi}{4}$ をノイマン型レジスタに記憶させ、それを QRP 命令のオペランドとして指定することによって等しい重み付けの重ね合わせ状態を生成することができる。CPhase 命令に

よって任意の回転角が指定できるので、様々な重ね合わせ状態を生成することができる。

$$\hat{U}(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

$$\hat{U}\left(\frac{\pi}{4}\right)|0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$\underbrace{\hat{U}\left(\frac{\pi}{4}\right) \otimes \hat{U}\left(\frac{\pi}{4}\right) \otimes \dots \otimes \hat{U}\left(\frac{\pi}{4}\right)}_{\text{指定した量子レジスタのビット長}}$$

3. プロセッサの構造

本稿で提案する量子コンピュータ・アーキテクチャは、量子命令と通常の命令が混在することを許す汎用的なものである。量子命令は量子ユニットで実行され、通常の命令はノイマン型ユニットで実行される。

図2に、我々の提案する量子コンピュータ・アーキテクチャにおけるプロセッサの内部ブロック図を示す。プロセッサは、量子命令を処理する量子ユニット(Q-Unit:Quantum Unit)と通常の命令を処理するノイマン型ユニット(N-Unit:Neumann type Unit)に大きく分けられる。量子コンピュータの汎用性を考える場合、現有のアルゴリズムより高速に動作する量子アルゴリズムのみを量子回路で実現させ、その他の部分をノイマン型アーキテクチャで実現させることが望ましい。これは、効率的な量子アルゴリズムが現在のところまだ数少ないことと、量子コンピュータ全体を量子回路モデルで実現したとしてもノイマン型アーキテクチャに対するメリットが考えにくいことによる。Shorの素因数分解アルゴリズムを例にとると、量子アルゴリズムと現有のアルゴリズムとを融合させ、効率的に計算を行う量子コンピュータとみることもできる。また、これまでのプログラム内蔵方式とソフトウェアを柔軟に継承することが可能である。

両ユニット間には、通常のビットを量子ビットに変換するQtoNTU(Quantum to Neumann Transformation Unit)と量子ビットを通常のビットに変換するNtoQTU(Neumann to Quantum Transformation Unit)が存在する。これらは目的の処理を達成することができる何らかの装置であると仮定する。また各ブロック間を接続するデータパスは、量子ビットを伝達することができる量子ワイヤであると仮定する。

4. 個々の量子命令の流れ

4.1 命令フェッチ

図3に、命令フェッチから命令分岐までの流れを示す。命令レジスタ(Instruction Register)に格納された命令は、最上位ビットを制御ユニット(Control Unit)に送り、制御ユニットは送られたビットを判別し、制御信号NorQ(0or1)によってQ/NDS(Quantum/Neumann

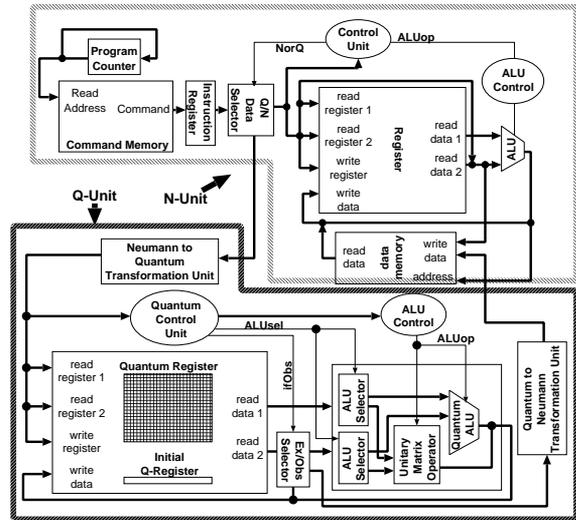


図2 汎用的量子コンピュータ・アーキテクチャの一例

Data Selector)を制御する。Q/NDSにより、通常の命令(ノイマン型命令)はノイマン型ユニット、量子命令は、変換ユニットNtoQTUを経て量子ユニットへ配分される。

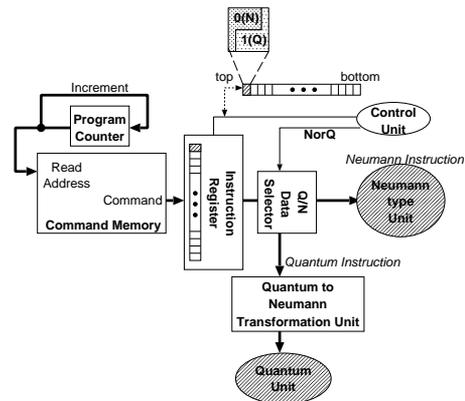


図3 命令フェッチしてPCを繰り上げる部分と命令を分岐させる部分

4.2 データ移動系命令

図4に、データ移動系命令の中でQExchange命令とQObserve命令の流れを示す。QExchange命令の場合、量子レジスタと量子初期化レジスタにそれぞれアクセスし、量子レジスタを初期化する。量子初期化レジスタへのアクセスは、その構造上読み出しのみ許すものとする。QObserve命令の場合、計算結果の観測を行うので量子初期化レジスタへのアクセスは行われない。量子制御ユニット(Quantum Control Unit)より制御信号ifObsがEx/Obs Selectorに作用し変換ユニットQtoNTUを経て通常のレジスタへ結果が格納される。

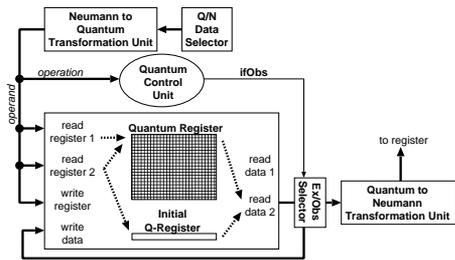


図 4 データ移動系命令のデータパス

4.3 算術論理演算系命令

図 5 に、算術論理演算系命令の中で QMultiply 命令および QMod 命令の流れを示す。算術演算を実行するユニットを制御する ALU Control は、特別な演算を処理する Unitary Matrix Operator と通常の算術演算命令 (QAdd 等) を処理する Quantum ALU のどちらを使用するかを制御線 ALUOp によって制御する。制御線 ALUUse は、Unitary Matrix Operator と Quantum ALU のどちらにデータを提供するかを判別する ALU Selector を制御するものである。

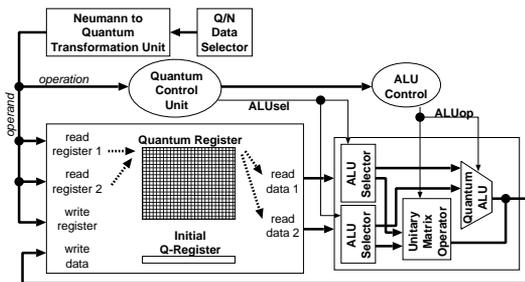


図 5 QMultiply 命令および QMod 命令を実行する部分

5. まとめと今後の課題

今回提案した汎用的量子コンピュータ・アーキテクチャは、量子命令セットおよびプロセッサの内部アーキテクチャの定義によりシミュレータを試行する上での基礎設計部分となる。

現在設計中のシミュレータは、これらの仕様をもとにプロセッサの各ブロックをモジュール化し、各命令は量子回路におけるプリミティブな制御 NOT および二重制御 NOT, ユニタリ作用素の標準形によって内部的にマクロ化する方法で作成する。開発言語は C++ を用いている。

シミュレーションによって、量子命令セットによる量子アルゴリズムのコーディングの有効性や抽象化されたインタフェースを検証することができ、さらなる改善を図ることが可能となる。

シミュレータには、現在、以下の問題を含み検討を行っている。

- QMultiply および QMod 命令を実現する量子回路, ユニタリ作用素の導出および計算手法問題の検討。
- 量子命令に対する不可逆性を適用してよいか。不可逆性の適用による量子初期化レジスタの必要性の検討。
- 量子初期化レジスタを有限長のスタック構造としたことによる何らかの誤差および悪影響の検討。
- 量子アルゴリズムを直接命令セットで記述したことでオペランド内に条件判定部分が含まれる問題および高級言語へ依存させる必要性の検討。

以上の問題点を克服しつつ、ノイマン型コンピュータ上での汎用的量子コンピュータ・アーキテクチャ・シミュレータの実装を進めている。

参考文献

- 1) Peter W. Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring", *Proceedings 35th Annual Symposium on Foundations of Computer Science*(1994), pp.124-134
- 2) Peter W. Shor, "Quantum Computing", *ICM (International Congress of Mathematicians) Proceeding Paper*(1998)
- 3) Lov K. Grover, "A fast quantum mechanical algorithm for database search", *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*(1996), pp.212-219
- 4) Kevin M. Obenland and Alvin M. Despain, "A Parallel Quantum Computer Simulator", *Submitted to High Performance Computing* (1998)
- 5) Kevin M. Obenland, "Feasibility of a Quantum Computer Architecture", *Dissertation Proposal* (1996)
- 6) Yao, A. "Quantum Circuit Complexity", *Proceedings of the 34th IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA* (1993), pp.352-360.
- 7) A.Barenco, C.H.Bennett, R.Cleve, D.P.DiVincenzo, N.Margolus, P.Shor, T.Sleator, J.A.Smolin, and H.Weinfurter, "Elementary gates for quantum computation", *Phys.Rev.A52* (1995), pp3457-3467
- 8) M.Oto, H.Nakajo, K.Joe, "A Possible Instruction Set for Quantum Computer Architectures", *The 2001 International Conference on Parallel and Distributed Processing Techniques and Applications PDPTA '2001 Volume III*, pp.1221-1227(2001.6)
- 9) 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 Vol.2000, No.16*, pp.77-80