

状態空間の分割を用いた量子探索ハードウェアシミュレータ

高木 文博[†] 中西 正樹[†] 山下 茂[†] 渡邊 勝正[†]

[†] 奈良先端科学技術大学院大学 情報科学研究科
〒630-0101 奈良県生駒市高山町 8916 番地の 5
E-mail: tayahi-ta@is.aist-nara.ac.jp

あらまし 量子計算のシミュレーションを行うにあたって、現在の計算機アーキテクチャを利用する場合、指数規模の計算資源が必要とされる。そこで、計算機システムの状態空間を分割することでより規模の大きいシミュレーションを実現する手法を提案する。

キーワード 量子計算シミュレータ, 量子探索アルゴリズム

Quantum Search Hardware Simulator based on Decomposition of the State Space

Ayahiro TAKAKI[†], Masaki NAKANISHI[†], Shigeru YAMASHITA[†], and
Katsumasa WATANABE[†]

[†] Graduate School of Information Science, Nara Institute of Science and Technology, 8916-5 Takayama,
Ikoma-shi, Nara, 630-0101, Japan.
E-mail: tayahi-ta@is.aist-nara.ac.jp

Abstract Exponential resources are required to simulate a quantum computer by a classical one. We suggest the way to realize larger scale hardware quantum computer simulator based on decomposition of the state space.

Key words quantum computer simulator, quantum search algorithm

1. はじめに

現在の計算機アーキテクチャの核を成す半導体技術の進歩は目覚しく、計算機構成デバイスの微小化により動作速度の向上や省電力化が推し進められている。しかしながら、これらは古典力学的現象に基づいたアーキテクチャであり、デバイスを構成する原子の数が減るにつれ量子力学的な現象が支配的になり有効な演算が困難になると考えられている。

そのため、量子力学的現象が支配的なシステムにおいて計算機として動作する量子計算機が近年注目されている。現行の計算機では、計算機の状態は 0 または 1 のビット列として表されるが、量子計算機の状態は qubit と呼ばれる 0 と 1 の値を重ね合わせて持つことができるビットの列で表される。この qubit の列は、量子状態と呼ばれ、同時に複数の状態を保持できるため、計算量が飛躍的に減少することが期待されている。現在提案されている量子状態を利用した重要なアルゴリズムとして、Shor の量子因数分解アルゴリズム [1] と Grover の量子探索アルゴリズム [2] が挙げられる。

しかしながら、量子状態には外界からの擾乱を受けやすいと

いう特徴があり、多数の qubit による演算を可能にするデバイスの開発は、困難を極めている。

これらの事情が相俟って、多数の qubit での量子計算機の動作や、量子状態が外界から受ける擾乱の影響などの解明が求められており、現在量子計算シミュレータの研究が盛んに行われるようになってきている。

量子状態の状態遷移における並列性から、量子計算を現行の計算機でシミュレーションする場合、指数規模の演算時間もしくは計算資源を要する。主にソフトウェアでの効率的なシミュレーションの研究が広く行われているが [3]、量子状態遷移における 1 ステップを幾つもの演算命令によって実現しなければならないことや、並列プロセッサを用いた場合でもその並列度が高々数並列であることなどから、計算時間が増大する傾向にある。

また、ハードウェアによる実装も試みられており、量子的な並列性をハードウェアの並列性に置き換えることで高速に演算を可能にする研究も行われている [4]。しかしながら、指数規模の計算資源を要することから、FPGA や LSI などでの実装は、qubit 数が多くなるにつれ困難を極めるものと考えられる。

その主な理由は、シミュレータの汎用性や量子状態遷移において完全な並列性をもって演算を行うことを目指しているからだと考えられる。そこで、本稿では、高速化・省面積化をはかるための手法として、状態空間を分割し、各部分空間に対する演算を繰り返すことにより、全体のシミュレーションを行う方法を示す。また、対象とするアルゴリズムを量子探索アルゴリズムに絞ることにより、演算器構成の単純化をはかる。

2. 量子計算

本章では、量子計算を理解するうえで必要な概念である、量子状態と量子状態遷移について形式的な定義および説明を行う。ここでは、3qubit のシステムを例にとって説明を行う。

2.1 量子状態

量子計算機の状態を表す量子状態は、基底状態 $(|000\rangle, |001\rangle, \dots, |111\rangle)$ の線形結合で表され (式 (1))、その係数 $(w_{000}, w_{001}, \dots, w_{111})$ は確率振幅と呼ばれる複素数である。ここで、基底状態とは 2^3 次元ベクトル空間における正規直交基底である。(式 (2))

$$|\Psi\rangle = w_{000}|000\rangle + w_{001}|001\rangle + \dots + w_{111}|111\rangle \quad (1)$$

$$|000\rangle = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, |001\rangle = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \dots, |111\rangle = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \quad (2)$$

一般にこの基底状態の線形結合を”状態の重ね合わせ”と呼び、その基底状態の係数は、式 (3) の条件を満たしていなければならない。

$$|w_{000}|^2 + |w_{001}|^2 + \dots + |w_{111}|^2 = 1 \quad (3)$$

また、この量子状態を観測した場合、状態 $|x\rangle$ は、 $|w_x|^2$ の確率で観測される。(ここで、 x は 000, 001, $\dots$, 111 のいずれかである)

2.2 量子状態遷移

量子状態遷移は、代数学においてユニタリ作用素と呼ばれる式 (4) の条件を満たす行列 U で記述できる。

$$U^\dagger U = U U^\dagger = I \quad I: \text{単位行列} \quad (4)$$

量子状態遷移とはこのユニタリ作用素 U を量子状態 $|\Psi\rangle$ に作用させることであり、遷移後の量子状態を $|\Psi'\rangle$ とすると、これらの間には式 (5) が成り立つ。

$$|\Psi'\rangle = U|\Psi\rangle \quad (5)$$

ここで、式 (1) と式 (2) から $|\Psi\rangle, |\Psi'\rangle$ を式 (6) のように定め、 2^3 次元ベクトル空間のユニタリ作用素である行列 U を式 (7) のように定めると、式 (5) は行列表記で式 (8) のように記述することができる。

$$|\Psi\rangle = \begin{pmatrix} w_{000} \\ w_{001} \\ \vdots \\ w_{110} \\ w_{111} \end{pmatrix}, |\Psi'\rangle = \begin{pmatrix} w'_{000} \\ w'_{001} \\ \vdots \\ w'_{110} \\ w'_{111} \end{pmatrix} \quad (6)$$

$$U = \begin{pmatrix} U_{00} & U_{01} & \dots & U_{06} & U_{07} \\ U_{10} & U_{11} & \dots & U_{16} & U_{17} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ U_{60} & U_{61} & \dots & U_{66} & U_{67} \\ U_{70} & U_{71} & \dots & U_{76} & U_{77} \end{pmatrix} (U_{xy}: \text{複素数}) \quad (7)$$

$$\begin{pmatrix} w'_{000} \\ w'_{001} \\ \vdots \\ w'_{110} \\ w'_{111} \end{pmatrix} = \begin{pmatrix} U_{00} & U_{01} & \dots & U_{06} & U_{07} \\ U_{10} & U_{11} & \dots & U_{16} & U_{17} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ U_{60} & U_{61} & \dots & U_{66} & U_{67} \\ U_{70} & U_{71} & \dots & U_{76} & U_{77} \end{pmatrix} \begin{pmatrix} w_{000} \\ w_{001} \\ \vdots \\ w_{110} \\ w_{111} \end{pmatrix} \quad (8)$$

これは 2^3 個の基底状態に対する演算が同時に行われることを意味しており、これが量子並列性と呼ばれる量子計算の大きな特徴である。よって、 n qubit 系においては 2^n 個の基底状態に対して並列に演算が行われる。これは、量子計算機のシミュレーションを古典計算機上で行う場合、実現する qubit 数に対し計算資源が指数規模必要とされることを意味しており、実時間計算もしくは計算資源の充足を困難にする大きな要因となっている。

3. 量子探索アルゴリズム

3.1 量子アルゴリズム

一般に量子アルゴリズムとは、ユニタリ作用素 U_k のシーケンスであり、そのユニタリ作用素を U_E とすると、式 (9) のように記述される。

$$U_E = U_n U_{n-1} U_{n-2} \dots U_k \dots U_2 U_1 U_0 \quad (9)$$

このユニタリ作用素 U_E に対して、一番右端の U_0 から量子状態に作用していく。

3.2 量子探索

本稿でシミュレータに実装する量子探索とは、あるオラクル関数 $f(x)$ について $f(x) = 1$ となるような解 x を探す量子アルゴリズムである。そのユニタリ作用素 U_{QS} は、式 (10) のように書ける。

$$U_{QS} = (DR_{af})^k \quad (10)$$

ここで R_{af} , D は、それぞれ、選択的回転 (selective rotation), 平均に関する反転変換 (reverse about average) と呼ばれるユニタリ作用素である。量子探索では、その qubit のサイズを n とすると、 $k = \sqrt{N}(N = 2^n)$ のとき式 (11) が成り立ち、その x は $f(x) = 1$ を満たすことが知られている。

$$|x\rangle = U_{QS}|\Psi_0\rangle \quad (11)$$

3.2.1 選択的回転

選択的回転 R_{af} は、オラクル関数 $f(x) = 1$ を満たす状態 $|x\rangle$ の確率振幅のみを反転するユニタリ作用素である。 $f(101) = 1$ とすると、この操作により確率振幅は図1の様に变化する。

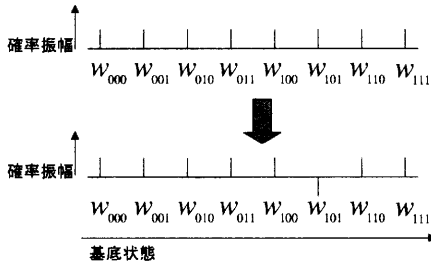


図1 3qubit 系の選択的回転

3.2.2 平均に関する反転変換

平均に関する反転変換 D は、各基底状態が持つ確率振幅の平均値でそれぞれの確率振幅を折り返すユニタリ作用素である。この操作で確率振幅は図2の様に变化する。

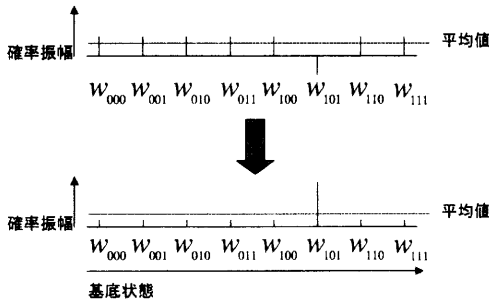


図2 3qubit 系の平均に関する反転変換

この平均に関する反転変換 D は、ウォルシュ-アダマール変換 W と選択的回転の特殊形 R_{0f} から構成され、式(12)のように書ける。

$$D = WR_{0f}W \quad (12)$$

3.2.3 量子探索回路

このユニタリ作用素 DR_{af} を量子回路で記述すると、図3のようになる。

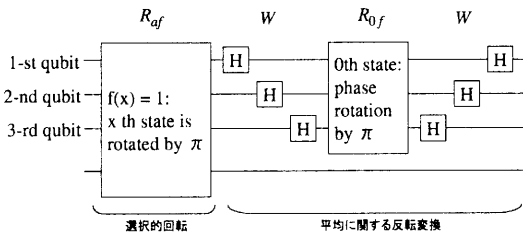


図3 3qubit に対する量子探索回路

3qubit の量子状態は、この量子探索回路に左から入ってい

く。すなわち、 2^3 個の状態が左から同時に入り、同様の作用を受けた後、同時に右へ出て行く。

本稿で提案するシミュレータのアーキテクチャについて図3で示した量子回路をもとに述べる。

4. アーキテクチャ

4.1 外部仕様

シミュレータへの入力は、オラクル関数 $f(x)$ の真値表とした。 n qubit 系のシミュレーションでは、長さ 2^n のビット列となる。

出力は、現在の量子状態を表す各確率振幅の値をセレクトによって選択したものとした。

4.2 量子ゲートの実装

図3の量子探索回路中の各量子ゲートは、シミュレータ上において 2^n 個の演算器の演算によってシミュレーションされる。実装する量子ゲートを例にその実装方法を示す。

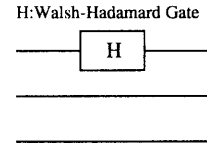


図4 3qubit 量子ゲート

図4に示した量子ゲート H は、下位1qubit に対して作用をし、上位2qubit の状態が同じであるような二つの状態、例えば $|000\rangle, |001\rangle$ に対して、 $H|000\rangle = 1/\sqrt{2}(|000\rangle + |001\rangle)$, $H|001\rangle = 1/\sqrt{2}(|000\rangle - |001\rangle)$ という状態遷移を同時に起こす。ここでゲートを通過する前の各状態の確率振幅を、 w_{000}, w_{001} とすると、通過後の各状態の確率振幅 w'_{000}, w'_{001} は、

$$w'_{000} = 1/\sqrt{2}(w_{000} + w_{001}) \quad (13)$$

$$w'_{001} = 1/\sqrt{2}(w_{000} - w_{001}) \quad (14)$$

となる。この演算を行うことで量子ゲートの動作を模倣することができる。上位2qubit の残りの組合せも合わせ3通りで並列に演算を行えば、図4の量子ゲートを1ステップで模倣可能である。すなわち、 n qubit 系を1ステップで模倣するには、 2^n 並列の演算を必要とする。

また、選択的回転変換の量子ゲートのシミュレーションは、位相反転の対象となる状態を判定し、その確率振幅の値に -1 をかけることで行う。

本稿では、より広い解空間に対応するためこの並列演算を適切に分割し数回にわたって実行することでシミュレーションを行う。

4.3 内部仕様

4.3.1 回路基本構成

回路の基本構成は図5のようになっている。

メモリモジュール (MemoryModule) には、32bit で表現された確率振幅が 2^{12} (4096) 個収められている。ウォルシュ-アダ

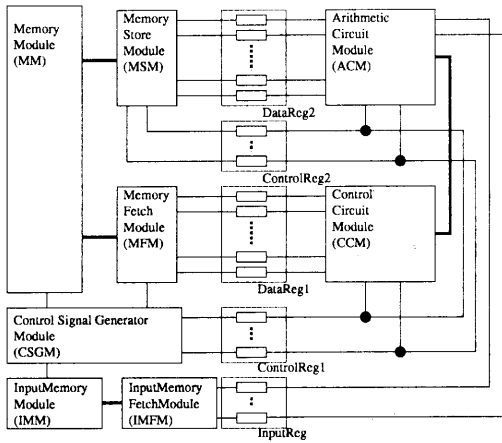


図5 シミュレータ回路ブロック図

マル変換 H では、2つの確率振幅をペアとして演算を行うので、2048組のペアを計算する必要がある、これらを32組のペアに分けて64回計算を繰り返すことでシミュレーションする。選択的回転変換 R_{af} , R_{of} では、各基底状態を個別に判定し確率振幅の符号の反転を行う。この場合も、確率振幅を64個ずつの集合に分けて64回の計算によってシミュレーションする。

実際の演算のサイクルの手順は、次に示すとおりに行う。

1. メモリモジュールから、64個の確率振幅をメモリフェッチモジュール (MemoryFetchModule) を通してデータレジスタ1(DataReg1)に値をセットする。この際、制御信号生成回路 (ControlSignalGeneratorModule) からの演算対象である量子ゲートの序数情報がフェッチモジュールの制御に用いられる。制御信号生成回路の出力はコントロールレジスタ1(ControlReg1)にセットされ、そのまま、コントロールレジスタ2(ControlReg2)へ次のクロックで代入する。

2. 制御回路 (ControlCircuitModule) 上でデータレジスタ1から送られてくる確率振幅についてペアを決定する。コントロールレジスタ1には、シミュレーション中の量子ゲートの序数情報、後述するメモリフェッチモードを識別する情報、書き込みアドレス情報などの制御情報が格納されている。ウォルシュアダマル変換時は、コントロールレジスタ1の情報を元に、確率振幅 w_x とペアになる確率振幅 w_y を $w_y^{(0)}$, $w_y^{(1)}$, ..., $w_y^{(5)}$ から決定し、演算回路に送る。選択的回転変換時には、演算回路で入力 w_y に関する不要な選択制御を省くため、コントロールレジスタ1からの制御信号は全て0にセットし、確率振幅 w_y に0を出力する。(図6-2.【制御回路】参照)

3. 演算回路 (ArithmeticCircuitModule) 上で制御回路から送られてくる確率振幅のペアの演算を行う。ウォルシュアダマル変換時は、コントロールレジスタ1の量子ゲートの情報と確率振幅の序数を元に、 $x < y$ のときは $w_y + w_x$, $x > y$ のときは $w_y - w_x$ を行い、ウォルシュアダマル変換を2回行うたびに右1bit算術シフトを行う。選択的回転変換時は、コントロールレジスタ1の量子ゲートの情報と入力メモリモジュール (InputMemoryModule) からの入力レジスタ (InputReg) に格納されるオラクル関数の真理値および確率振幅の序数を元に、

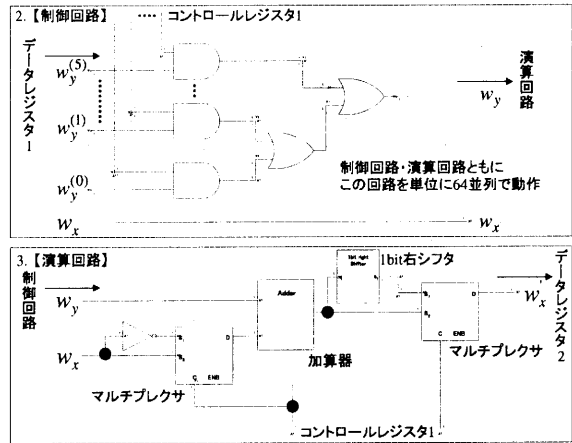


図6 制御回路及び演算回路構成

$w_y + w_x$ (符号反転なし) もしくは $w_y - w_x$ (符号反転あり) の演算を行う。これらの結果 w_x' をデータレジスタ2(DataReg2)にセットする。(図6-3.【演算回路】参照)

4. データレジスタ2にセットされた値をメモリストアモジュール (MemoryStoreModule) を通してメモリモジュールに書き出す。この際、書き込みのためのアドレスはコントロールレジスタ2に、書き込みイネーブル信号はコントロールレジスタ2の情報を元に生成される。

これらの4つの一連の手順(1.→2.→3.→4.)を1サイクルとすると、12qubitの量子計算をシミュレーションする場合、このサイクルを64サイクル繰り返すことで一つの量子ゲートをシミュレーションすることができる。

よって、選択的回転変換 R_{af} と平均に関する反転変換 D の合成作用素 DR_{af} は、 R_{af} 1回, D 1回, ウォルシュアダマル変換 H (2×12) 回で構成されるので、この64サイクルを26回繰り返すことでシミュレーションされる。

12qubitのシミュレーションでは、式(11)を満たす状態に移移するまでに、このユニタリ作用素 DR_{af} の演算を49回前後繰り返す。

4.3.2 レジスタ

レジスタには、前述したように確率振幅の値を格納したデータレジスタと各回路モジュールの制御情報を格納したコントロールレジスタの2種類がある。

データレジスタは、一般的には複素数を表現するために虚部と実部を割り当てる必要があるが、実装するアルゴリズムの特性上、虚部が演算対象とならないためレジスタに格納する情報は実部のみとしている。レジスタ一つあたりビット幅は32ビットとり、上位1桁目は符号ビットとし2桁目は1の位としている。演算は固定小数点で行われる。

コントロールレジスタは、制御信号生成回路 (ControlSignalGeneratorModule) からのものと入力データメモリ (InputMemoryModule) からのものがあり、それぞれ、36, 64ビットのビット幅をとる。

4.3.3 メモリ

図7に実装するメモリを示す。モジュール右下を基準に左右

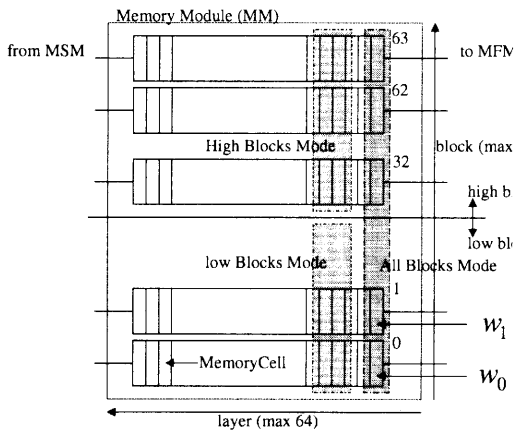


図7 メモリブロック図

方向をレイヤと呼び、上下方向をブロックと呼ぶ。レイヤ、ブロックそれぞれ左および上を上位と定める。

各メモリブロックは、0番から63番まで64個の同時読み書き可能な同期式メモリを使用し、データ幅は32bitである。これらの32bit単位のデータを収めた区画をメモリセル(MC)と呼ぶことにする。一つのメモリブロックには64個のメモリセルを構成しており、 i 番目のブロックの j 番目のレイヤのメモリセルを MC_{i-j} と表記することにする。

12qubitのシミュレータを構成するにあたって、4096個の確率振幅の値は最下位のメモリセル MC_{0-0} から順に $MC_{0-0} = w_0, MC_{1-0} = w_1, \dots, MC_{62-63} = w_{4095}, MC_{63-63} = w_{4096}$ と昇順に格納する。

メモリデータのフェッチ方法は、3種類のモードに分類されており、それぞれ、AllBlocksMode(ABM), LowBlocksMode(LBM), HighBlocksMode(HBM) の3種類の手法を使い分ける。これは、効率よくシミュレータを動作させるための一つの手法であり、状態数4096にわたる全状態空間を状態数64の部分空間に分割し効率よくシミュレーションを行うことを可能にしている。

ABMでは、同じレイヤのメモリセル64個が演算対象としてフェッチされる。これは、量子ゲートにおいて第6qubit以下に1qubitのウォルシュアダマール変換が作用する場合、もしくは、選択的回転変換を行う場合のモードである。例えば、3qubit目にウォルシュアダマール変換が作用する場合、確率振幅のペアは、 $[w_0, w_4], [w_1, w_5], [w_2, w_6], [w_3, w_7], \dots$ である。これはメモリセルでの表記にすると、 $[MC_{0-0}, MC_{4-0}], [MC_{1-0}, MC_{5-0}], [MC_{2-0}, MC_{6-0}], [MC_{3-0}, MC_{7-0}], \dots$ である。(図8参照) 6qubit目のウォルシュアダマール変換まで、ペアとなるメモリセルのレイヤ番号は同じになるので、同じレイヤのメモリセルのデータをフェッチして確率振幅のペアを決定することが可能である。このモードで64ステップの演算が完了すれば一つの量子ゲートの演算が完了したことになる。

LBMおよびHBMでは、2つのレイヤのメモリデータを、LowBlocksもしくはHighBlocksのどちらかのブロックからだけフェッチする。この動作モードは、量子ゲ

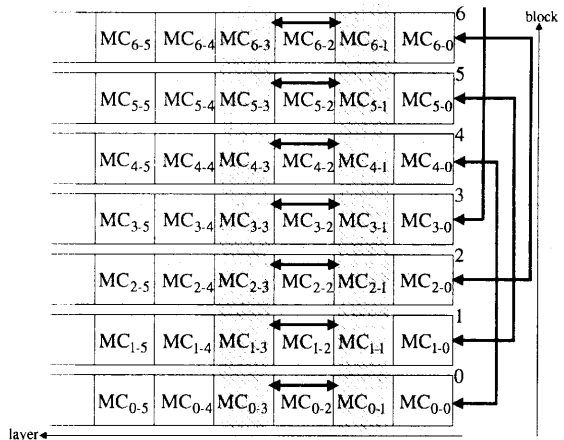


図8 確率振幅のペア決定例

ートの第7qubit以降に1qubitのウォルシュアダマール変換が作用する場合に用いられる。例えば、8qubit目にウォルシュアダマール変換が作用する場合、確率振幅のペアは、 $[w_0, w_{128}], \dots, [w_{64}, w_{192}], [w_{65}, w_{193}], [w_{66}, w_{194}], [w_{67}, w_{195}], \dots$ である。 $[w_{64}, w_{192}]$ からの64組のペアをメモリセルでの表記にすると、 $[MC_{0-1}, MC_{0-3}], [MC_{1-1}, MC_{1-3}], [MC_{2-1}, MC_{2-3}], [MC_{3-1}, MC_{3-3}], \dots, [MC_{63-1}, MC_{63-3}]$ である。(図8参照) すなわち、レイヤ1とレイヤ3のメモリセルを上位ブロックと下位ブロックに分けてフェッチすることで、確率振幅のペアを決定することが可能である。また、同じメモリブロックからは同時に2つ以上のデータを読み出すことはできないので、この動作モードではデータフェッチに2ステップを要する。よって、LBM, HBM, それぞれ64ステップの演算の完了を経て、計128ステップで一つの量子ゲートの演算が完了する。データレジスタ1へのメモリデータの格納は、常に下位レイヤのメモリデータを下位側に格納し、上位レイヤのメモリデータを上位側に格納する方法をとる。これにより、制御回路および演算回路の制御が簡単化される。

4.3.4 メモリフェッチ回路

メモリデータのフェッチを制御する回路で、ABMでは、メモリからのデータをそのままレジスタに流し、LBMおよびHBMでは、0ステップ目で下位レジスタに、1ステップ目で上位レジスタにメモリデータを格納する。

4.3.5 メモリストア回路

メモリデータのストアを制御する回路で、ABMでは、レジスタからのデータをそのままメモリブロックに流し、LBMおよびHBMでは、それぞれ下位のメモリブロック、上位のメモリブロックにデータを書き込む。このとき、下位側のデータレジスタ2は下位レイヤのメモリセルに、上位側のデータレジスタ2は上位レイヤのメモリセルに書き込まれる。また、メモリブロックのライトイネーブル信号の生成も行っている。

4.3.6 制御信号生成回路

制御信号生成回路では、制御回路など各回路で利用される制御情報を生成する。その主な構成要素は、ユニタリ作用素 DR_{af} の演算回数を表すレジスタ、 DR_{af} 内部の演算経過情報

を表すレジスタ, メモリデータのフェッチ経過を表すレジスタ, 読み込みアドレスなどのレジスタやデータフェッチモードの識別ビットなどである。

4.3.7 演算回路

量子探索アルゴリズムは, 加算, 減算, 符号反転, 2 の割算のみでシミュレーション可能であるため, 演算回路における確率振幅のペアを演算する各演算ユニットは一つの加算器と一つの 1bit シフタのみで実現されている。また, ウォルシュ-アダマール変換のシミュレーションで現れる乗算 ($\times 1/\sqrt{2}$) は, その変換のシミュレーションを 2 回行った後, 各確率振幅を 2 で割ることで代用可能なので, 右 1 ビット算術シフト演算によって代用している。

5. ま と め

状態空間の分割を用いて 12qubit の量子探索アルゴリズムを効率よくシミュレーションするハードウェアアーキテクチャの提案を行った。現在, 提案アーキテクチャに従ったシステムを FPGA 上に構築中であり, 面積の規模, 及び動作周波数などのデータ収集が課題である。また, 既に C 言語で量子探索ソフトウェアシミュレータを作成済みであり, 速度比較及び考察も課題として挙げる。

提案アーキテクチャでは, メモリスケジューリングを工夫することで回路全体を極力絶え間なく利用し, 効率の良い演算を行うことが可能である。現在, 外部システム (PC もしくは FPGA の外部メモリなど) との連携により更なる解空間の拡充も検討しているが, 提案アーキテクチャはそのモジュールとして再利用可能である。外部とやり取りできるデータ幅には限りがあるので, ハードウェアの稼働効率を下げずに演算を続ける仕組みを考案することが当面の課題である。

また, 解空間の拡充のほかに, 量子計算機が外界からの擾乱すなわちエラーを受けた際のシミュレーションも非常に重要である。特に, 量子探索アルゴリズムは, 量子因数分解アルゴリズムに比べ, 解を得るまでのステップ数が多く, 実際の量子計算機において外界の環境から紛れ込んでくるエラーに対して大きな影響を受けることが予想されている。そのため, エラーのシミュレーションは重要な意味を持つと考えられるので, これについても今後設計の視野に入れていくことを考えている。

謝 辞

本研究は, 一部, 文部科学省科学研究費補助金 (課題番号: 15700014) ならびに大川情報通信基金研究助成による。

文 献

- [1] P. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," Proc. 35th Symp. on Foundations of Computer Science, pp. 124-134, 1994.
- [2] L. Grover, "A fast quantum mechanical algorithm for database search," Proc. 28th Symp. on the Theory of Computing, pp. 212-219, 1996.
- [3] J. Niwa, K. Matsumoto, H. Imai, "General-Purpose parallel simulator for quantum computing," Physical Review A 66, 062317 (2002).
- [4] S. O'uchi, M. Fujishima, K. Hoh, "AN 8-QUBIT