学位請求論文

大規模集積回路を用いた
量子回路プロセッサに関する研究

平成13年12月14日

指導教官　　鳳　紘一郎　教授

東京大学大学院　工学系研究科
電子情報工学専攻

大内 真一

目 次

第 1章

序論

1.1 計算における未解決問題と量子コンピュータ

集積回路 (LSI, Large-Scale Integrated circuit) の発展と共に, 計算機は飛躍的な発展を遂げてきた. これはデバイスの微細化により扱える計算問題の規模が向上したことによるところが大きい. しかし, 集積回路の発展によっても原理的に扱いの難しい計算問題も多数存在する. 計算の複雑度を示す尺度として, 計算量クラスが用いられる. 代表的なクラスとして, P(Polynomial time), NP(Nondeterministic Polynomial time), および NP の部分集合として NP 完全 (NP complete) というクラスが存在する. これらは, 問題の入力サイズすなわち入力のビット数を用いて以下のように定義されている.

(1) P: 入力サイズの多項式で表される時間 (以下多項式時間と称する) で解を得られる計算アルゴリズムが存在する問題の集合全体. ここで計算は決定性チューリングマシン (Turing Machine, TM) により行うものとする.

(2) NP: 問題の解の候補が与えられた場合, 解の候補が問題の解であるか決定性 TM を用いて多項式時間で確認可能な問題の集合全体.

(3) NP 完全: NP 問題のうち, 完全性を満たす部分集合. NP 完全に属する問題は互いに多項式時間あるいは対数時間で他の問題に変換可能である.

このうち, 現在効率的に計算機で解くことが可能であると分かっている問題はクラス P に属する問題である. これに対し, クラス NP に属する問題を現在の計算機を使って多項式時間で解くことの可能なアルゴリズムは現時点では知られていない. そればかりか, 「クラス NP と P の包含関係がどうなっているか」つまり, 現在の計算機を使って NP 問題を多項式時間で解くことが原理的に可能であるかどうかすら分かっていない. これは P=NP 問題と呼ばれており, 計算科学における未解決問題とされている.

よってNP問題については, 入力サイズnに対し$O(2^n)$の解候補を1つ1つ確認していくことにより解を探すのような, 指数時間のアルゴリズムによって問題を解くしか, 現在のところ手段がないことになる. この場合, 問題の規模が大きい問題については計算時間が爆発し, 計算機での扱いが非常に困難になってしまう. しかし, 実用面で重要な問題にはNP完全に属するものが多く含まれているため, このP=NP問題の解決は重要な意味を持っている. 現在のところ, 近似アルゴリズムと分類される遺伝的アルゴリズムなどによって, この問題に対処する手法が一般的となっている. この取り扱いの理論的正当性は, 逆にP=NP問題が未解決であることに立脚している. また, NP問題の1つとされている因数分解を用いた公開鍵暗号が現在広く用いられているが, その安全性も同様にP=NP問題が未解決であることにより保証されていると言って良い.

さて, このP=NP問題に光を与えるものと近年期待されている量子コンピュータは, 可逆計算に関する検討 [1] をきっかけに, 1980年頃のR. P. Feynmanなどによりアイディア [2, 3] が出された. 以来, 従来方式のコンピュータを上回る性能を有するのではないかと考えられてきたが, これを実証する結果は長い間得られず, 量子コンピュータ研究は一時下火になっていた. しかし, 1994年になり, P. W. Shorにより, 因数分解が量子コンピュータを用いることにより多項式時間で解くことが可能であることが示された [10]. このブレークスルーにより, 現在量子コンピュータに対する関心が大変高くなっている. 量子コンピュータに対する期待は多岐に渡る. 様々なNP問題, あるいはNP完全問題の多項式アルゴリズムの実現をはじめ, 量子コンピュータ研究の発展の上にP=NP問題に関する知見が得られるのではないかと期待されている. また, 量子コンピュータが因数分解を多項式時間で解くことが可能であることから, 暗号の安全性に影響を与える可能性も指摘されており, 安全保証の立場からも研究が行われている.

1.2 量子コンピュータに関する研究の動向と集積回路技術の発展

量子コンピュータは, 現在理論的側面, 実験的側面の両方から研究が進められている. 理論的側面においては, 量子コンピュータ上で実行可能な新しいアルゴリズムの獲得が目標であり, 他方実験的側面においてはその実現可能性を示すことである. 量子コンピュータは従来の計算機とは異なり, 量子重ね合わせを用いたコンピュータモデルである. これが多項式時間でのNP問題の取り扱いを可能にするものと考えられる. しかし, この量子重ね合わせと言う物理現象は外乱に非常に影響を受けやすく, 実用的な量子コンピュータを作製することには大変な困難が伴う. 実験研究の進捗としては, 冷却イオントラップを用いるもの [16], 光子の偏光を利用するもの [17], 核磁気共鳴を用いるもの [18], 固体物性を用いるもの [19, 21, 22, 23, 24] など様々な提案と一部実験が報告されている. これまでもっとも大きな規模の実験を行なった例は5ビット相当の計算を核磁気共鳴により行なったもの [28] であるが, この核磁気共鳴を使った方法では10ビット相当が限界であるという指摘もある [29]. このように, 実験研究では, 未だ適当な物理的実装方法を模索している段階である.

このような状況を鑑み, この実験面での発展に先立ち量子コンピュータで何ができるかを知ることは非常に重要となる. この場合, 既存の計算システム上で量子コンピュータのシミュレーションを行なうことが量子アルゴリズム研究を進める上で重要な位置を占めてくると考えられる. しかし, 量子コンピュータは既存の計算機には存在しない量子重ね合わせを用いており, この重ね合わせ状態の振舞いを既存の計算機により模倣し NP 問題を解くことは, 計算時間の指数爆発を招くこととなる. よって, 量子コンピュータの大規模なシミュレーションを実現することは大変難しい.

しかし冒頭でも触れたように, 集積回路技術の発展により 1 チップ上で扱うことのできるデバイス数は年々指数関数的に増大している [30]. よって, 計算機の並列性は理論的には指数関数的に増大させることが可能であり, 並列計算に適した形の問題であれば非常に高速な処理可能が可能となると考えられる. この点で量子コンピュータの高速シミュレーションもある程度の規模で実現できる可能性がある.

1.3 本研究の目的

前述のような大規模で高速な量子コンピュータを作製する場合, 量子コンピュータのシミュレーションに適しており, 同時に巨大システムを構成するのに適した簡単なアーキテクチャを採用することが鍵となる. 本研究の第一目標はまず, このアーキテクチャを創出し, 集積回路を用いた量子コンピュータの大規模高速シミュレータを実現することにより, 量子コンピュータ研究に一助をなすことである.

さらに, このシミュレータのアーキテクチャおよび計算手法を拡張することにより, 集積回路技術を用いて NP 問題の時間計算量爆発の緩和に光を当てることが第二の目標となる.

このアーキテクチャは集積回路のスケーリングの先に存在するナノデバイスを用いた計算システムに拡張できるものへと発展させることが望まれる. これは本研究の範囲を越えるものであるが, 本研究はこれを念頭において進められる.

この目標に基づき, 本論文では第 2 章で量子コンピュータの理論について概観した後, 第 3 章で量子コンピュータにおける重ね合わせ状態の発展の様子を集積回路上の周期信号の重ね合わせ状態の変化によってエミュレートする離散時間システムの作製に述べる.

続いて第 4 章では, デバイスは大量に必要とするものの, この並列動作によって量子コンピュータと同等の計算時間である大きさの問題を解くことの可能な量子回路プロセッサについて述べる. さらに, 第 5 章では, この量子回路プロセッサを発展させて, 並列計算の新たな形を模索する.

第 2章

量子コンピュータの理論

2.1 はじめに

本研究の背景となる量子コンピュータは, NP 問題の多項式時間での取り扱いが期待されている.

この量子コンピュータの起源は, 1980 年頃の R. P. Feynman などのアイディアに遡ることができる [2, 3]. 可逆計算 [1] と関係の深いこののアイディアは, さらに P. Benioff[4] や D. Deutch ら [5] により数学的にモデル化され, 現在の量子コンピュータ研究の礎が築かれた. この量子コンピュータは, 量子効果を用いることにより, 従来の計算機に比べ高速な演算ができると予想されていたが, 1994 年に P. W. Shor によって NP 問題の 1 つとされる因数分解が量子コンピュータを用いて多項式時間で解かれることが示され [10], その期待は確かなものとなった. 今後の発展によっては, 量子コンピュータは NP 問題の解法, あるいは数学上の未解決問題とされる P=NP 問題の解決に寄与する可能性がある.

本章では, 本研究の主題である量子コンピュータについて基本的な事項を説明する.

2.2 量子コンピュータモデルと演算

2.2.1 量子ビット

通常のコンピュータにおいて, 情報は $\{0,1\}$ の 2 値により表現される. これらは, 例えば回路中のノードにおける電圧等, 古典的な物理量を担体として伝達される. これに対し, 量子コンピュータにおいてはビットを表す担体に古典的な物理量ではなく量子力学的な物理状態を用いる. すなわち, 例えばスピン 1/2 など 2 状態系の基底 $\{|\uparrow\rangle, |\downarrow\rangle\}$ を $\{0,1\}$ として用いる. これは量子ビット (quantum bit, qubit) と称され, 通常のビットとは異なる以下の特徴を有する.

量子ビットにおける論理の重ね合わせ

量子力学的な状態は重ね合わせが可能であるから, 量子ビットを情報担体として用いる量子コンピュータ上では論理の重ね合わせが可能である. これは, 数学的には論理を表す基底 $\{|0\rangle, |1\rangle\}$ の線形結合

$$|\psi\rangle = \omega_0|0\rangle + \omega_1|1\rangle \tag{2.1}$$

の形で表現される. ここで, 重ね合わせの重み係数 $\omega_0, \omega_1 \in C$ はそれぞれ 0 の状態と 1 の状態の確率振幅と称さる. ただし, C は複素数の体を表すものと定義する. 量子力学によれば, この系を観測した場合, 状態 $|0\rangle$ は確率 $|\omega_0|^2 = \omega_0\omega_0^*$ で, 状態 $|1\rangle$ は確率 $|\omega_1|^2 = \omega_1\omega_1^*$ でそれぞれ観測される. ただし, ω^* は共役複素数を表す. この確率は

$$\omega_0\omega_0^* + \omega_1\omega_1^* = 1 \tag{2.2}$$

を満たす. 上式を満たせば ω_0, ω_1 は任意に定めることが可能である.

量子レジスタと量子ビットのエンタングルメント

量子コンピュータでは, 複数の量子ビットからなる量子力学系を構成し, これをレジスタのように扱う.

複数の量子ビットから構成される系は, テンソル積で表現される. 例えば, 2 つの量子ビットからなる系を考える. 2 つの量子ビットのうち, 上位を m, 下位を l と名付け, それぞれの状態が

$$\left.\begin{aligned} |\psi^{(\mathrm{m})}\rangle &= \omega_0^{(\mathrm{m})}|0\rangle + \omega_1^{(\mathrm{m})}|1\rangle \\ |\psi^{(\mathrm{l})}\rangle &= \omega_0^{(\mathrm{l})}|0\rangle + \omega_1^{(\mathrm{l})}|1\rangle \end{aligned}\right\} \tag{2.3}$$

であったとする. ただし, $\omega_0^{(\mathrm{m})}, \omega_1^{(\mathrm{m})}$ および $\omega_0^{(\mathrm{l})}, \omega_1^{(\mathrm{l})}$ はそれぞれの量子ビットにおける状態 0, 1 の確率振幅を表す. このとき, 量子ビット m, l からなる系全体は, テンソル積

$$\begin{aligned} |\Psi\rangle &= |\psi^{(\mathrm{m})}\rangle \otimes |\psi^{(\mathrm{l})}\rangle \\ &= \omega_0^{(\mathrm{m})}\omega_0^{(\mathrm{l})}|0\rangle\otimes|0\rangle + \omega_0^{(\mathrm{m})}\omega_1^{(\mathrm{l})}|0\rangle\otimes|1\rangle + \omega_1^{(\mathrm{m})}\omega_0^{(\mathrm{l})}|1\rangle\otimes|0\rangle + \omega_1^{(\mathrm{m})}\omega_1^{(\mathrm{l})}|1\rangle\otimes|1\rangle \end{aligned} \tag{2.4}$$

で表現することが可能である. ただし, このようにテンソル積で表現可能な合成系は, 量子ビット同士は互いに相関の無い振舞いをする. たとえば, この合成系で量子ビット m および l 両方で 0 を観測する確率は $|\omega_0^{(\mathrm{m})}|^2|\omega_0^{(\mathrm{l})}|^2$ となるが, これは量子ビット m において状態 0 を観測する確率 $|\omega_0^{(\mathrm{m})}|^2$ 量子ビット l において状態 0 を観測する確率 $|\omega_0^{(\mathrm{l})}|^2$ の積となっており, 2 つの量子ビットでの観測は独立な事象であることを示している. この場合, ある量子ビットの振舞いは他の量子ビットとは無相関である.

これに対し, 2つの量子系が量子的な相関を持っている場合があり得る. 例えば, 光学結晶を励起光で励起した場合に発生される, 偏光に相関が存在する光子の EPR 対などがそれである. 例えば, 横偏光と縦偏光, あるいは右まわり偏光と左まわり偏光の相補的な2つの状態を $\{|0\rangle, |1\rangle\}$ と定義すると,

$$|\Psi\rangle = \omega_{00}|0\rangle \otimes |0\rangle + \omega_{11}|1\rangle \otimes |1\rangle \tag{2.5}$$

や

$$|\Psi\rangle = \omega_{01}|0\rangle \otimes |1\rangle + \omega_{10}|1\rangle \otimes |0\rangle \tag{2.6}$$

のように表される. これらが意味するところは, 量子ビット m で 0 が観測された場合は, 必ず量子ビット l では 0 が観測される, あるいは量子ビット m で 1 が観測された場合は必ず l では 1 が観測されるかのいずれかの場合しか存在しない, と言うような相関が合成系に存在することを示している. この場合, もはや (2.4) のようなテンソル積では系が表現できない. 2つの量子系が相関を持っていることをエンタングルメントと呼ぶ. このエンタングルメントの発生は量子コンピュータにおける演算と関わりが深く, 演算結果を表現する上で不可欠である. 量子コンピュータ上での演算とエンタングルメントの関係は, 詳しく後述する.

以下では $|0\rangle \otimes |0\rangle$ を $|00\rangle$ のように略記し, 2量子ビットの量子レジスタに格納される2ビットの語のように扱う. 一般に, n 量子ビットの量子レジスタの基底は

$$|x_{n-1}\rangle \otimes |x_{n-2}\rangle \ldots \otimes |x_1\rangle \otimes |x_0\rangle = |(x_{n-1}x_{n-2}\ldots x_1x_0)_2\rangle, \{x_i|0,1\} \tag{2.7}$$

のように表され, 2^n 次元のヒルベルト空間の単位ベクトルとなる. つまり, n 量子ビットの量子レジスタは 2^n 次元ヒルベルト空間を構成する. レジスタ上の重ね合わせ状態は

$$|\Psi\rangle = \sum_{x=0}^{2^n-1} \omega_x |x\rangle \tag{2.8}$$

と表記される. このとき,

$$\sum_{x=0}^{2^n-1} \omega_x \omega_x^* = 1 \tag{2.9}$$

が成り立つ.

2.2.2 量子ゲート操作と量子回路

量子回路

量子ビットを用いて演算を行う量子コンピュータのモデルとしては量子チューリングマシン [5] と量子回路の2種類が提案されているが, 両者は等価であることが示されている [6]. 量

子ビットの状態を変化させる操作は, どちらにおいても量子ビットに対してユニタリ変換を施すことにより行われる. ここでは, 量子回路について述べる.

ある量子ビットに対するNOT演算を行うことを考える. いま, 状態$|0\rangle, |1\rangle$をそれぞれ縦ベクトル

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.10}$$

と記述することにすると, NOT演算は行列で

$$U_{\mathrm{NOT}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{2.11}$$

と表され, 任意の状態にある量子ビット

$$|\Psi\rangle = \begin{bmatrix} \omega_0 \\ \omega_1 \end{bmatrix} \tag{2.12}$$

に対し,

$$|\Psi'\rangle = \begin{bmatrix} \omega_0' \\ \omega_1' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \omega_0 \\ \omega_1 \end{bmatrix} = \begin{bmatrix} \omega_1 \\ \omega_0 \end{bmatrix} \tag{2.13}$$

のように働く. このU_{NOT}はユニタリ対称性を持っているため, 実際に量子力学的な物理操作により実行可能である. 一般に, ユニタリ行列の形で記述可能な論理操作は物理系で実現可能である. 逆に, 量子コンピュータ上で量子ビットに行おうとする論理操作はユニタリ性を持っていなければならない. このユニタリな論理操作を量子ビットにいかなる順序で行うかを論理回路に似せたチャートで示すのが量子回路であり, その要素であるユニタリ変換による論理演算を量子ゲート操作と称する. 量子回路の例を図2.1に示す. 通常, 量子レジスタの初期状態を示すのが図の左端であり水平方向の線は各々の量子ビットを示している. 量子コンピュータの分野では通常, MSBを上側に, LSBを下側に記述する例が多い. 量子ゲート操作は, 量子回路の左側から右側に向かって順に実行されていく. すべてゲート操作が終了した図中右端の状態が量子回路の出力状態である. この出力状態の中から論理値の1つが, 観測操作により確率的にユーザ側に返される. このとき, 量子ゲート操作の手順, すなわちアルゴリズム問題の解として得るべき状態の振幅が強められるように組まれる.

量子回路における論理演算

論理回路における完全系は, ANDゲートとNOTゲートの組み合わせ, あるいはORゲートとNOTゲートの組み合わせ, NANDゲート等によって実現できる. この完全系が量子回路においても実現されなければならないが, 量子回路の場合には完全系を構成する論理ゲー

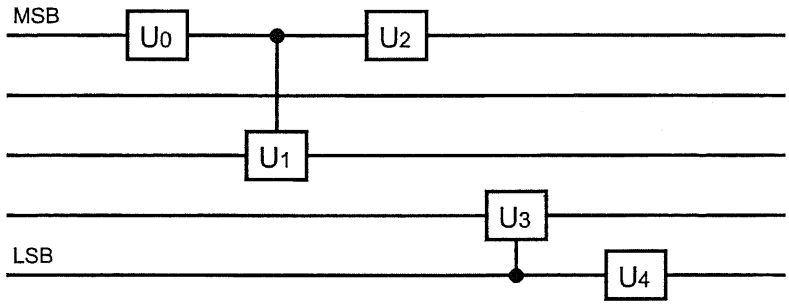


図 2.1: 量子回路の例. 上位量子ビットを上側に, 下位量子ビットを下側に記述し, これに対するゲート操作は左側から順に行われ, 回路右端がすべてのゲート操作を終了した出力状態を表す.

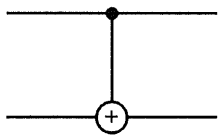


図 2.2: 制御 NOT ゲートのシンボル. 上位ビットがコントロールビット, 下位ビットがターゲットビットを示す.

トにユニタリ対称性が無ければならない. これを満たす要素ゲートの 1 つとして多用されるのが, 制御 NOT ゲート (Controlled NOT, CNOT) と呼ばれる量子ゲートである. CNOT ゲートのシンボルと真理値表を図 2.2 および表 2.1 に示す. これらに定義する通り, 制御 NOT ゲートは 2 量子ビットに作用するゲートであり, コントロールビットに 1 が入力された場合にのみターゲットビットの論理を反転する. これは, 通常の論理回路における排他的論理和 (Exclusive OR, XOR) と等価である. CNOT は数式では

$$U_{\mathrm{CNOT}} = |0\rangle\langle 0| \otimes E + |1\rangle\langle 1| \otimes U_{\mathrm{NOT}} \tag{2.14}$$

と定義される. ただし, E は 2×2 単位行列, つまり 1 量子ビットに対する恒等変換を示す. これを行列により表記すると,

$$U_{\mathrm{CNOT}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} E & O \\ O & U_{\mathrm{NOT}} \end{bmatrix} \tag{2.15}$$

となる. ただし O は 2×2 の零行列を示す. CNOT には当然重ねあわせ状態が入力可能である. 重ね合わせ状態に含まれる基底の 1 つ 1 つはそれぞれが真理値表に従い変換される.

表 2.1: 制御 NOT ゲートの真理値表.

入力		出力	
コントロール	ターゲット	コントロール	ターゲット
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	0

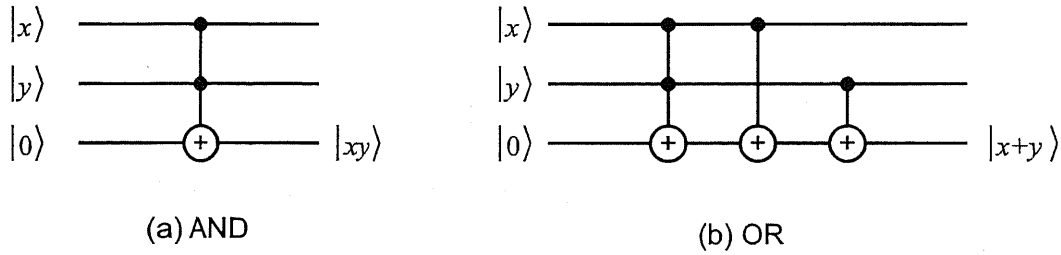


図 2.3: 量子回路における (a)AND ゲートおよび (b)OR ゲートの一般的な構成法.

ここで, 例として

$$|\psi\rangle = (\omega_0^{(\mathrm{m})}|0\rangle + \omega_1^{(\mathrm{m})}|1\rangle) \otimes \omega_1^{(\mathrm{l})}|1\rangle \tag{2.16}$$

に CNOT を作用させた場合を考えると,

$$\begin{aligned} |\psi'\rangle &= U_{\mathrm{CNOT}}|\psi\rangle \\ &= \omega_0^{(\mathrm{m})}\omega_1^{(\mathrm{l})}|01\rangle + \omega_1^{(\mathrm{m})}\omega_1^{(\mathrm{l})}|10\rangle \end{aligned} \tag{2.17}$$

と変換される. ここで, $|\psi'\rangle$ は, $|\psi\rangle$ がテンソル積で書けていたのにもかかわらず, 量子ビットのテンソル積では書けない状態になっており, U_{CNOT} によりエンタングルメントが生じたことを示している. このように, 演算の結果エンタングルメントが生ずることはしばしば起こる. このエンタングルメントは, 逆変換 U_{CNOT}^{-1} を $|\psi'\rangle$ に施すことにより解消される.

さて, NOT と CNOT により論理演算の完全系が実現されることは, CNOT により AND, OR が実現されることにより簡単に分かる. CNOT を用いて AND, OR の論理を計算する量子回路を図 2.3 に示す. ここで, AND ゲートに使われる制御ビットが 2 つあるゲートを特にトフォリゲートと呼ぶ. これらと前述の NOT ゲートを合わせれば完全系が構成される. ここで, 量子ゲートは通常の論理ゲートとは異なり, ユニタリ変換で実現しなければならない

ことから入力ビット数と出力ビット数を変えることが許されない. 同時にこれは可逆性を保証する. あるゲートの逆ゲートは, その逆変換で実現される.

さらに, 量子回路では振幅の大きさ, 位相をアルゴリズムで有効に用いるために, 完全系を構成する論理演算の他に任意のユニタリ変換が必要になる. A. Barenco らによれば, 1 量子ビットに対するゲート操作の要素

$$R_y(\theta) = \begin{bmatrix} \cos\theta & sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{2.18}$$

$$R_z(\alpha) = \begin{bmatrix} e^{i\alpha/2} & 0 \\ 0 & e^{-i\alpha/2} \end{bmatrix} \tag{2.19}$$

$$\Phi(\delta) = \begin{bmatrix} e^{i\delta} & 0 \\ 0 & e^{i\delta} \end{bmatrix} \tag{2.20}$$

と CNOT を組み合わせることにより, 任意の量子ビット数の量子回路において任意のユニタリ変換を実現できる [7]. このように, 任意のユニタリ変換を実現できる量子ゲートの集合のことを, 量子コンピュータの分野では万能集合 (universal set) 場合によっては万能量子ゲート (universal quantum gate) と呼んでいる.

同様に, D. Deutch らによって

$$A(\phi,\alpha,\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{i\alpha}\cos\theta & -ie^{i(\alpha-\phi)}\sin\theta \\ 0 & 0 & -ie^{i(\alpha+\phi)}\sin\theta & e^{i\alpha}\cos\theta \end{bmatrix} \tag{2.21}$$

のような形の 2 量子ビット量子ゲートが万能量子ゲートであることが示されている [8]. これは A. Barenco の万能量子ゲートと等価変換できることが分かる. 一般に, 万能量子ゲートを構成する要素の決め方はいくつも存在する. この万能量子ゲートの決め方はアルゴリズム記述のしやすさあるいは量子コンピュータを実現しようとする物理系の性質等により決定されると言える.

量子並列性

量子コンピュータにおける高速性はさまざまな形で説明されているが, ここでは量子並列性と称される性質について述べる.

いま, 例として n 量子ビットの量子回路において, ある i 番目の量子ビットに対してユニタリ変換

$$U = \begin{bmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{bmatrix} \tag{2.22}$$

を施す場合を考える. この量子ゲートを表す全体の行列 S は, テンソル積

$$S = \underbrace{E \otimes \cdots \otimes E \otimes U \otimes \overbrace{E \otimes \cdots \otimes E}^{i-1\text{ 項}}}_{n\text{ 項}} \tag{2.23}$$

で表される. これを行列に展開すると, S は $2^n \times 2^n$ 正方行列であり

$$S = \underbrace{\begin{bmatrix} T & & & \\ & T & & \\ & & \ddots & \\ & & & T \end{bmatrix}}_{2^{(n-i)}}, \tag{2.24}$$

ただし

$$T = \underbrace{\begin{bmatrix} u_{00} & & & & u_{10} & & & \\ & \ddots & & & & \ddots & & \\ & & u_{00} & & & & u_{10} & \\ u_{01} & & & & u_{11} & & & \\ & \ddots & & & & \ddots & & \\ & & u_{01} & & & & u_{11} & \end{bmatrix}}_{2^{(i-1)}} \underbrace{\qquad}_{2^{(i-1)}} \tag{2.25}$$

という 2^i 番目の非対角成分のみが非零のバンド対角行列となっている.

さて, 量子回路 S への入力状態を $|\Psi\rangle$, 出力状態を $|\Psi'\rangle$ と定義する. 変換 S は, 量子ビットが互いに相関無く独立な振舞いをしている状態

$$|\Psi\rangle = |\psi_{n-1}\rangle \otimes |\psi_{n-2}\rangle \otimes \cdots \otimes |\psi_1\rangle \otimes |\psi_0\rangle \tag{2.26}$$

に S を作用させると, その出力は

$$|\Psi'\rangle = |\psi_{n-1}\rangle \otimes |\psi_{n-2}\rangle \otimes \cdots \otimes U|\psi_{i-1}\rangle \otimes \cdots \otimes |\psi_1\rangle \otimes |\psi_0\rangle \tag{2.27}$$

のようにまとめることができる. これを古典的な演算装置で模倣することを考えると, n 個の無相関な働きをする装置を用いて量子コンピュータと同じ 1 ステップで可能なことが分かる. すなわち, n 個の装置のうち i 番目の装置の状態を, U を模倣するような変換により変化させればよい. しかし, 入力 $|\Psi\rangle$ にエンタングルメントがあると状況は異なる. 例えば, n 量子ビット全ての量子ビットが互いに従属である状態

$$|\Psi\rangle = \sum_{x=0}^{2^n-1} \omega_x |x\rangle \tag{2.28}$$

が入力された場合, 出力状態は

$$
\begin{aligned}
|\Psi'\rangle &= S|\Psi\rangle \\
&= U\begin{bmatrix} \omega_{00\ldots0\ldots00} \\ \omega_{00\ldots1\ldots00} \end{bmatrix} + U\begin{bmatrix} \omega_{00\ldots0\ldots01} \\ \omega_{00\ldots1\ldots01} \end{bmatrix} + \cdots + U\begin{bmatrix} \omega_{11\ldots0\ldots11} \\ \omega_{11\ldots1\ldots11} \end{bmatrix}
\end{aligned} \tag{2.29}
$$

のように表される. すなわち S は, あたかも量子ゲート操作のターゲットである i 番目の量子ビットのみが $|0\rangle, |1\rangle$ と異なる状態を対として選び出し, これに U を作用させる操作を, 2^{n-1} 個存在する対全てに対し遂行する. 量子コンピュータでは, エンタングルメントと重ね合わせの効果により, この操作を n 個の物理的実体上で 1 ステップで行うことが可能である. これは量子並列性と呼ばれる. これに対し, 等価な計算を古典的な演算装置で行う場合, ステップ数を $O(n)$ で演算を行おうとすると演算装置が $O(2^n)$ 必要となり, 逆に演算装置を $O(n)$ で行おうとするとステップ数が $O(2^n)$ 掛かることとなる.

2.3 量子アルゴリズム

量子コンピュータは, これまで効率的な解法が知られていなかった NP 問題に対する応用が期待される. 量子コンピュータ上で実行される量子アルゴリズムは, 前述の量子並列性が有効に働くように設計されなければならない. 現在考案されているもののうち, 良く知られているのが因数分解のアルゴリズム [10, 11] とデータ検索のアルゴリズム [12, 13, 14] である.

2.3.1 ショアの因数分解アルゴリズム

"整数を因数分解する" という問題は, 素因数分解とは異なり, "整数 N が与えられたときに, N の非自明な (1 と N 以外の) 因数が存在するならば, それを 1 組発見せよ" という問題として定義される. この問題の計算量を考えてみると, 例えば $\sqrt{N}$ 以下の各整数で割ってみるという素朴なものが考えられる. しかし, 試行 1 回を 1 ステップとして考えても, 最悪で $\sqrt{N} = 2^{\frac{1}{2}\log N}$ 回の試行が必要となる. これはすなわち入力の桁数に対して計算ステップ数が指数関数的に爆発していくことになるため, 現在のコンピュータで大きな整数の因数分解問題を解くことは事実上不可能である. 現在知られている最も効率的なアルゴリズムでも, $2^{(\log N)^{1/3}(\log\log N)^{2/3}}$ 程度のステップ数が必要である [9]. 現在このような因数分解の難しさが公開鍵暗号の原理に用いられている. 逆に, 因数からもとの整数 N を求めるはただのかけ算で, 必要ステップ数は入力桁数の多項式で抑えられる.

P. W. Shor によって考案された因数分解アルゴリズムは, 整数論から知られた合成数の性質を応用する. いま, 因数分解しようとする整数 N の方程式

$$
x^2 \equiv 1 (\mathrm{mod} N) \tag{2.30}
$$

を考える．この方程式の解は，“自乗して N で割った場合の剰余が1になるような x” であるが，自明の解 $x \equiv \pm 1 (\mathrm{mod} N)$ の他に非自明な解があるとすれば N は合成数であり，その非自明な解が求められれば効率的に N の因数を求められることが整数論から知られている．N と互いに素，すなわち最大公約数 $\gcd(N, y) = 1$ となるようなある整数に対して

$$y^r \equiv 1 (\mathrm{mod} N) \tag{2.31}$$

を満たす r は周期的に現れることが知られているので，その最小である位数 r を求めたときそれが偶数であれば，$x = y^{r/2}$ を (2.30) に代入して

$$(y^{r/2} - 1)(y^{r/2} + 1) \equiv 0 (\mathrm{mod} N) \tag{2.32}$$

を根拠に，$\gcd(y^{r/2} - 1, N), \gcd(y^{r/2} + 1, N)$ を因数の候補として探し出すことができる．最大公約数はユークリッドの互除法により多項式時間でできることが分かっているので，量子コンピュータですべきことは，ある y についての r の位数を多項式ステップ数で求めるという問題に置き換えられる．量子コンピュータを用いて行われるのは，この位数を求める操作である．

量子コンピュータでは，$N^2 < q < 2N^2$ を満たす q 以下の全ての整数 a について $y^a (\mathrm{mod}\, N)$ を計算し，これらの計算結果全体からの周期 r を求める．ここで，q の条件 $N^2 < q < 2N^2$ は，アルゴリズムの精度を保つために必要となるものである．この $y^a (\mathrm{mod} N)$ を最悪のケースで $O(N^2)$ 回繰り返して計算する代わりに量子コンピュータでは重ね合わせ処理を行なうため，多項式ステップ数で計算を終了する．ショアの方法による位数 r を求める手順は，以下のようになる．

(1) $l[\equiv \log q]$ 量子ビットのレジスタ1と $m[\equiv \log N]$ 量子ビットのレジスタ2を準備し，$|0\rangle$ に初期化する．

(2) q 以下の全ての整数の重ね合わせをレジスタ1に生成する．この操作は，レジスタ1の各量子ビットにユニタリ変換であるウォルッシュ・アダマール変換

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.33}$$

を作用させることにより遂行される．すなわち，

$$W : |0\rangle \mapsto \bigotimes_{i=0}^{m-1} \left(\frac{1}{\sqrt{2}} (|0\rangle_i + |1\rangle_i) \right) = \frac{1}{\sqrt{2^n}} \sum_{a=0}^{q-1} |a\rangle \tag{2.34}$$

のようにテンソル積によって重ね合わせが実現される．

(3) N と互いに素な x をランダムに選び,

$$S : |a\rangle|0\rangle \mapsto |a\rangle|x^a (\mathrm{mod} N)\rangle \tag{2.35}$$

のように, レジスタ2に x^a を N で割った余りを書き込むような量子ゲート操作を, 重ね合わせ状態に対し行なう. この変換 S を実行した後は, 2つのレジスタはエンタングルした状態になっている. すなわち, a と計算結果 $x^a (\mathrm{mod} N)$ はエンタングルされた状態になる. ここで, S を実行するためには補助量子ビットが N から $2N$ 個必要となる.

(4) 続いて, 量子フーリエ変換

$$\mathrm{DFT}_q : |a\rangle \mapsto \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} \exp(2\pi i ac/q) \tag{2.36}$$

をレジスタ1に対し行なうと, レジスタの状態は

$$\frac{1}{q} \sum_{a=0}^{q-1} \sum_{c=0}^{q-1} \exp(2\pi i ac/q)|c\rangle|x^a (\mathrm{mod} N)\rangle \tag{2.37}$$

のように変化する.

(5) ここで, レジスタの状態を観測する. この結果, ある状態 $|c\rangle|x^k (\mathrm{mod}\, N)\rangle$ を観測したとする. このとき, 観測確率は

$$\left| \frac{1}{q} \sum_{a : x^a \equiv x^k (\mathrm{mod}\ N)} \exp(2\pi i ac/q) \right|^2 \tag{2.38}$$

と計算される. ただし a についての和は, $x^a \equiv x^k (\mathrm{mod} N)$ にを満たす全ての a の項を含むことを意味する. この a が周期 r を持った数であることが数論からわかっているから, 整数 j, l を用いて $a = jr + l$ と表すことができる. ゆえに観測確率は

$$\left| \frac{1}{q} \sum_j \exp(2\pi i jrc/q) \exp(2\pi i lc/q) \right|^2 = \begin{cases} \text{定数} & c \text{が } q/r \text{ の整数倍の時} \\ 0 & \text{それ以外の時} \end{cases} \tag{2.39}$$

というようになり, 結局 $|c\rangle = |\lambda q/r\rangle, \lambda \in \mathbf{Z}$ となる状態が観測され, それ以外は観測されない.

(6) 最後に観測された c を基に q/c を既約分数の形で求めることにより, r の候補を得られるので, $\gcd(x^{r/2} - 1, N), \gcd(x^{r/2} + 1, N)$ から因数の候補が得られる. ここで, c に含まれる定数係数 λ が r と互いに素な場合があるが, 多項式回程度一連の操作を繰り返すことにより解を得られることが示されている.

上記の説明では, q/r が割り切れる場合について述べたが, 割り切れない場合にも最初に述べた条件 $N^2 < q < 2N^2$ を満たせば, 高い精度で求める状態を得られることが示されている [11].

このように, ショアの因数分解アルゴリズムは, 重ね合わせによる一括処理, 位相に対する処理と, これらによって可能となる観測による情報の抽出を効果的に利用する.

2.3.2 グローバーのデータベース検索アルゴリズム

L. K. Grover により提案されたアルゴリズム [12] は, N 個のファイルを含むソートされていないデータベースにおいて, 目的のデータを検索をするのような問題に適用される. この問題を解くためにシラミ潰しにデータを検索していくと, 平均して $N/2$ 回の検索が必要なことが予想される. グローバーのアルゴリズムは, 利用者が与えた条件にあったファイルに対応付けられた固有状態の確率振幅を充分大きく増幅して観測することにより, 適当なファイルを見つけ出すというもので, この増幅に要する演算ステップ数が $\sqrt{N}$ に抑えられる. 多項式時間のアルゴリズムではないが, 多く問題に対するの応用が試みられている [14].

ここでは, データベース検索を, N 個の解候補 a から

$$f(a) = \begin{cases} 1 & a = a_0 \\ \\ 0 & \text{その他の場合} \end{cases} \tag{2.40}$$

という関数を満たす a_0 を探す問題に置き換えて考える. いま, 解候補 a を入力すると $f(a)$ を 1 ステップで出力する, オラクルと呼ばれる装置を考える. これは, 計算量を計量するために用いられる仮想的なサブルーチンと言える. シラミ潰しの場合, 解候補 a をランダムに選び, 順次オラクルに代入して a_0 を見つけようとすると, オラクルへの質問は平均で $N/2$ 回必要となる. ここで, 状態を表すタグをバイナリで表すと, そのビット長は n は $n = \log N$ となるから, 結局オラクルへの質問回数は 2^{n-1} となり, 計算時間は指数時間かかることとなる. このように, 計算時間はオラクルへの質問回数として見積もることが可能である.

さて, グローバーの量子アルゴリズムではオラクルの量子版 "量子オラクル" を利用して問題を解くことになる. この量子オラクル S_f には質問を重ね合わせの形で入力でき, その出力は重ね合わせの形で返される. これを

$$S_f : |a\rangle \mapsto (-1)^{f(a)}|a\rangle \tag{2.41}$$

のように位相反転の形で得られたと仮定する. この U_f を全ての解候補の重ね合わせをに対して作用させ, 位相の反転したものが解であるが, これを重ね合わせから 1 回の観測操作を通じて知ることはできない.

Groverによれば, 位相の反転した状態の振幅を強めて高い確率で観測を行なうために, 行列要素が

$$d_{ij} = -\delta_{ij} + \frac{2}{N} = \begin{cases} -1 + \dfrac{2}{N} & i = j \\ \dfrac{2}{N} & i \neq j \end{cases} \tag{2.42}$$

となるような $N \times N$ ユニタリ行列 D を用いる. ただし, $i, j = 0, 1, 2, \ldots N-1$ であり, δ_{ij} はクロネッカーの δ を表す. この D は, 拡散変換と呼ばれ, 変換されるベクトルの全要素の確率振幅の平均値を中心に振幅を反転する働きを持つ. すなわち, D を 単位行列 E と全要素が $1/N$ からなる行列 P により

$$D = -E + 2P \tag{2.43}$$

と書き直すと, あるベクトル $|v\rangle$ は

$$D|v\rangle = (-E + 2P)|v\rangle = -|v\rangle + 2P|v\rangle \tag{2.44}$$

と変換される. このとき, $P|v\rangle$ の項は, その全ベクトル要素が $|v\rangle$ の全ベクトル要素の平均値となることより, 同平均値を a とおくと変換後のベクトル $D|v\rangle$ の i 番目の要素は

$$v_i' = -v_i + 2a, \tag{2.45}$$

すなわち, 各要素 v_i は平均値 a を中心に反転が行なわれることとなる. この D を用いて, 適当な解を抽出するアルゴリズムは次のように実行される.

(1) 式 (2.34) により, N 状態全ての重ね合わせを生成する.

(2) S_f を作用させ, 解の状態の振幅を反転させる.

(3) D を作用させ, 全状態の平均値を中心に位相反転を行なう.

(4) (2), (3) を $\lfloor(\pi/4)\sqrt{N}\rfloor$ 回続けた後, 観測を行なうことにより, 解の状態を観測できる. このとき, 失敗する確率は $1/N$ より小さい.

結果として, オラクルへの質問回数が $O(\sqrt{N})$ の計算量となる.

このアルゴリズム手続き (2),(3) を繰り返すときの, 振幅の変化について考える. いま, i 回同繰り返し操作を実行した時の解状態の振幅を k_i, それ以外の振幅を l_i とおく. $|0\rangle$ へ状態を初期化し, 手続き (1) を実行した時, それぞれの振幅は

$$\left.\begin{aligned} k_0 &= \frac{1}{\sqrt{N}} \\ l_0 &= \frac{1}{\sqrt{N}} \end{aligned}\right\} \tag{2.46}$$

と, 等しい. この状態から $i+1$ 回繰り返し操作を実行した時の振幅は, i 回目の値を使って,

$$\left.\begin{aligned} k_{i+1} &= \frac{N-2}{N}k_i + \frac{2(N-1)}{N}l_i \\ l_{i+1} &= \frac{N-2}{N}l_i - \frac{2}{N}k_i \end{aligned}\right\} \tag{2.47}$$

と書ける. いま, $\sin^2\theta = 1/N$ なる θ を導入し, 式 (2.47) を書き換えると,

$$\left.\begin{aligned} k_{i+1} &= \sin((2i+1)\theta) \\ l_{i+1} &= \frac{1}{\sqrt{N-1}}\cos((2i+1)\theta) \end{aligned}\right\} \tag{2.48}$$

と表される. (2.48) より, k_i が最大すなわち 1 となるのは, $(2i_0+1)\theta = \pi/2$, すなわち, 繰り返し回数が $i_0 = (\pi - 2\theta)/4\theta$ となる場合であることがわかる. N が非常に大きい, すなわち $\theta \simeq \sin\theta = 1/\sqrt{N}$ という近似を用いれば, 結果として $(\pi/4)\sqrt{N}$ にもっとも近い整数が, i_0 を与えることがわかる. なお, 以上では解状態が 1 つの場合について記述したが, 一般に解状態が t 個の場合にも同様のアルゴリズムが実行でき, 繰り返し回数 i_0 は $(\pi/4)\sqrt{N/t}$ となる.

D は, ウォルッシュ・アダマール変換 W と状態 $|0\rangle$ のみの位相反転を行なう行列

$$S_0 = \begin{bmatrix} -1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} \tag{2.49}$$

を用いて

$$D = -WS_0W \tag{2.50}$$

と書かれる. これらを実際に量子回路で実現することを考えると, D のうち, W は前述の通り 2×2 行列 H を各量子ビットに作用させることにより実現できる. また, S_0 および S_f は状態

$$|b\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{2.51}$$

に初期化された 1 量子ビットの補助ビット $|b\rangle$ を用いて,

$$\mathrm{U}_f : |a\rangle|b\rangle \mapsto |a\rangle|b \oplus f(a)\rangle, \tag{2.52}$$

という量子回路により実現される. ただし, $\oplus$ は排他的論理和と等価な1ビットの加算を示す. すなわち, CNOT を利用し, レジスタ $|a\rangle$ に含まれる状態が条件 $f(a)$ を満たす場合に補助ビット $|b\rangle$ の論理を反転する量子回路を作れば

$$\mathrm{U}_f : |a\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \mapsto (-1)^{f(a)}|a\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{2.53}$$

のように S_f の機能を実現できる. このとき, 振幅に変化は及ぼされない.

さて, 実際にグローバーのアルゴリズムをどのような問題に利用できるか考えてみると, オラクルの部分をどう実現するかが問題になってくるように思われる. 本来, オラクルは数学的な仮想のサブルーチンであり, グローバーのアルゴリズムは, このオラクルを使った量子計算機モデルを使って $O(\sqrt{N})$ で NP 完全問題を解くことができることを示したものに過ぎない. 換言すれば, オラクルが表す関数の性質を知るためのアルゴリズムと言える. このアルゴリズムを実際に使う場合, 量子オラクルが多項式段数の量子ゲートで実現され多項式時間でその設計が実現可能であれば, 計算量を $O(\sqrt{N})$ のままでアルゴリズムを実行可能である. 例えば, 充足可能性 (Satisfiability, SAT) 問題にグローバーのアルゴリズムを適用可能であることが知られている. SAT とは, 論理式 $f(x_0, x_1, x_2, \ldots, x_{n-1}) = 1$ を満たすことが可能か否か判定する問題である. ここで, 論理式 f は和積標準形で書かれる. この論理式 f を CNOT の組み合わせにより実現し, グローバーのアルゴリズムにおける U_f に組み込めば良い.

なお, 本節の最初に述べたように本アルゴリズムの原著論文において, Grover はデータベース検索を応用例として挙げているが, 実際にはデータベース検索には利用できないと指摘する文献がある [15]. ショアのアルゴリズムと同様な, エンタングルメントを利用した

$$|\Psi\rangle = \sum_x |x\rangle|g(x)\rangle \tag{2.54}$$

という状態を作り, x をデータベースのタグ, $g(x)$ をデータと考えれば, $g_0 = g(x_0)$ を与える x_0 を探すという問題がソートされていないデータベースを探すことに相当する. この場合, ユーザによって入力された g_0 と g を比較するオラクルによって $g(x)$ にグローバーのアルゴリズムを行ない, 結果を観測することにより, 状態 $|x_0\rangle|g_0\rangle$ を観測できるように思われるが, アルゴリズムの実装に用いる $D = -WS_0W$ が動作しない.

2.4 量子コンピュータの実現に向けた実験研究の動向

量子コンピュータの実現を目指して, 実際に量子効果の顕著に現れる物理系を使って量子計算を行なう試みが多数行なわれている. その中でも, 有機化合物分子の核スピンを核磁気共鳴 (nuclear macnetic resonance, NMR) によって操作する NMR 量子コンピュータ [18] では現在までに 5 量子ビットの計算が実現されている [27].

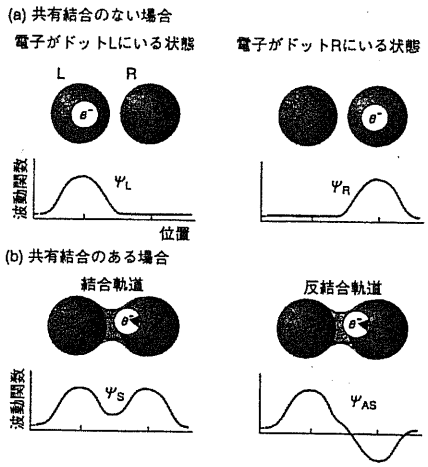


図 2.4: 結合量子ドットの波動関数の模式図 [20].

しかし, 実際的な量子計算の実現に必要な技術レベルについて考えてみると, その実現にはこれらはあまり向いていないと思われる. 5 量子ビットの実験に成功した NMR 量子計算機でも, 10 量子ビットを境に実現が困難になってくるという予想がある [29] のに対し, 量子アルゴリズムの実現には大量の量子ビットが必要になるからである. 例えば, ショアのアルゴリズムについて考えると, n ビットの整数の因数分解に必要な量子ビット数は $4n$ から $5n$ 程度であるが, 10 進 300 桁の因数分解を想定すると, 少なくとも約 3600 量子ビットが必要となる. このような大量の量子ビットを利用しようとすると量子ビットの集積が不可欠になるため, 前述の方法よりも固体物理を用いた系での実現を考える方がより現実的であるという見方が実験家の間では強い.

以下, いくつかの提案と実験について紹介する.

2.4.1 結合量子ドットにおける人工 2 準位系を用いた量子ビット

図 2.4 のように, 量子ドットを 2 つ近接した構造を作ったとき, ドット間の結合は, あたかもイオン結合や共有結合をしている分子ような特性を示す. 両ドット間のトンネル結合が単電子トンネルのように弱い場合, 電子は片方のドットに極在する. このとき, どちらかのドットに電子が入るとクーロン力により他方のドットには電子が入りにくくなる. しかし, 結合ドット上のゲートの電圧を変化させることによりトンネル結合強度を強くすると, 共鳴トンネルが起こるようなバイアス条件で電子が片方に極在することがなくなり, 両ドット間を往来するようなる. ここで, 左側に電子が存在する状態を $|\mathrm{L}\rangle$, 右側に電子が存在する場合を $|R\rangle$

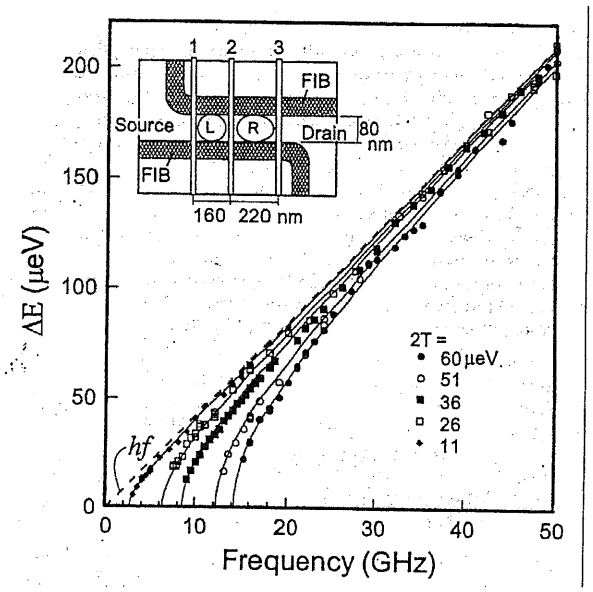


図 2.5: 報告された結合量子ドットデバイスの模式図と, 強くドットが結合した場合に生ずる、"結合量子ドット分子" のスペクトル解析 [19].

と定義すると, トンネル結合が強い場合には 2 つの状態

$$|\Psi_{\mathrm{A}}\rangle = \frac{1}{\sqrt{2}}\left(|\mathrm{L}\rangle - |\mathrm{R}\rangle\right) \tag{2.55}$$

$$|\Psi_{\mathrm{B}}\rangle = \frac{1}{\sqrt{2}}\left(|\mathrm{L}\rangle + |\mathrm{R}\rangle\right) \tag{2.56}$$

をとる可能性がある. これはちょうど, 共有結合する 2 原子分子のエネルギーの高い反結合状態 $|\Psi_{\mathrm{A}}\rangle$ とエネルギーの低い結合状態 $|\Psi_{\mathrm{B}}\rangle$ に対比される. これに対し, 前者のトンネル結合が弱い状態はイオン結合状態に対比される.

このような人工的な 2 準位系が結合ドットにより形成可能であり, 外界から与える電界やマイクロ波によって, この 2 状態間での遷移を起こさせることが可能であると実験的に示され, 量子ビットとしてこの 2 準位系を使える可能性が指摘された [19]. 図 2.5 は, $|\Psi_{\mathrm{A}}\rangle, |\Psi_{\mathrm{B}}\rangle$ の 2 状態間の遷移に共鳴する周波数を計ったものである. しかし実際には, 前述の疑似的な結合状態, 反結合状態は定常状態であり, 量子ビットとしてこれを用いるためには非定常状態での重ね合わせ制御を行なう技術が必要である.

2.4.2 単一クーパー対箱における人工2準位系を用いた量子ビット

結合量子ドットと類似の人工2準位系を単一クーパー対箱により実現する方法がある. この系は, 図2.6に示すように, "浴" 電極と "島" 電極が微小ジョセフソン接合に結合されたものである. この接合は, 超電導体としての性質と共にキャパシタとしての性質も持っている.

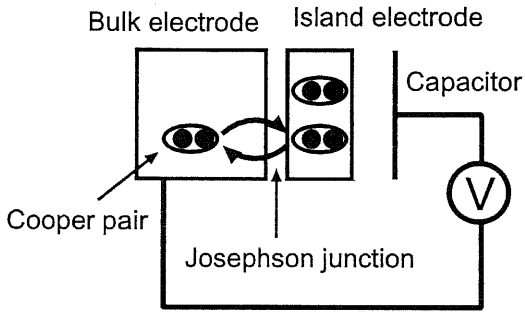


図2.6: 単一クーパー対箱の模式図. "島" 電極はジョセフソン接合によって "浴" 電極に接合された超電導体である.

このため, 島電極は単一電子トンネルと同じような性質を同時に持っている. 電子1つが島電極に帯電した場合のエネルギーを E_{C} とおく. この E_{C} に対してジョセフソン結合エネルギー E_{J} が,

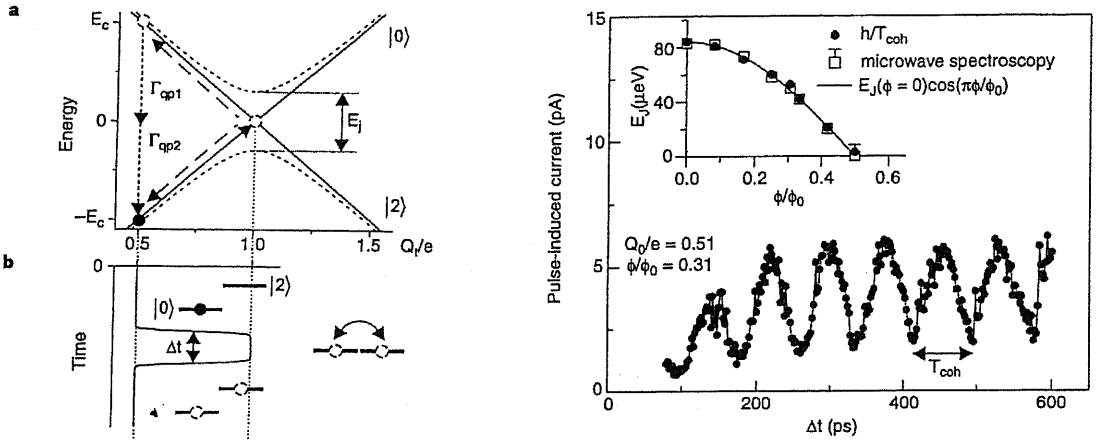
$$\Delta > E_{\mathrm{C}} > E_{\mathrm{J}} \gg k_{\mathrm{B}}T \tag{2.57}$$

の条件を満たす時, 単電子トンネルよりも単クーパー対トンネルが顕著に観測されるようになる. ここで, Δ は超電導ギャップ, k_{B} はボルツマン定数, T は温度を表す. よって, 島電極内部の余剰電子数 n によって系を $|n\rangle$ と記述することにすると, $|0\rangle$, $|2\rangle$ のように, 系は離散的な状態をとることになる. 系のエネルギーを図示すると図2.7(a)を得る. 図中実線で示すのは, $|0\rangle$ 状態と $|2\rangle$ 状態の静電エネルギーであり, これはゲートのバイアス条件, すなわち接合部に誘起される電荷に依存する. このバイアスによって系が静電エネルギーの交差する点に近付けられると, 系は前述の結合量子ドットと同様に,

$$|\Psi_{\mathrm{A}}\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |2\rangle\right) \tag{2.58}$$

$$|\Psi_{\mathrm{B}}\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |2\rangle\right) \tag{2.59}$$

と表される2つの結合状態を形成する. これらは, 図2.7中破線で示すような個有エネルギーをもつ.



(a,b) (c)

図 2.7: パルスによって量子状態を制御する手法を示す模式図. (a) エネルギーダイヤグラム中実線は静電エネルギー, 点線は結合状態によって生ずる固有エネルギー. (b) のようなパルスを系に印加することにより, 状態を重ね合わせに移動させ, 戻した時に生ずるトンネル電流をモニタすることによりにより, (c) のようなコヒーレントな振動の証拠が得られる [22].

いま, 始状態として $|0\rangle$ にある系に対し, (b) のようなパルスを電極から与え, 系を前述のクロスオーバー点に遷移させると, $|0\rangle$ は非定常状態となり, $|0\rangle$ と $|2\rangle$ の重ね合わせ状態に系は遷移する. これは, 単一のクーパー対がコヒーレントな振動を起こすことに対応する. このパルスがもとの電圧に戻されると, 確率 1/2 で $|0\rangle$ もしくは $|2\rangle$ の定常状態に戻る. NEC の中村らによって行なわれた実験は, この一連の動作の最後に系が $|2\rangle$ に戻されたときにながれるトンネル電流を特別な電極を作製して観測することにより, 系が $|0\rangle$, $|2\rangle$ の間をコヒーレントに振動していることを確かめたものであり, 結合量子ドットで課題になっていた非定常な重ね合わせを実現したことになる [22]. このパルスの長さを変化させることにより, NOT 操作やウォルッシュ·アダマール変換に一致する操作を実現でき, 1 量子ビットの量子コンピュータを実現できたことにもなった.

この実験系のコヒーレンスの続く時間, すなわちデコヒーレンス時間は $1\mu s$ と見積もられている. コヒーレント振動の周期が 100ps であり, これをゲート操作に必要な時間と考えると, 1 量子ビットの系でゲート操作が最大で 10^4 回行なえる可能性がある. しかし, 2 量子ビットでの相関を発生させる量子ゲート操作の実験は未だに行なわれていない. また, 2 量子ビット以上に量子ビットを増やすとデコヒーレンス時間徐々に短くなると考えられる. 量子状態を制御できる系では, 逆に系との結合が強いため, デコヒーレンスが起こりやすくなり, 更に量子ビット数が多くなれば, 外部と結合する部分が多くなるからである.

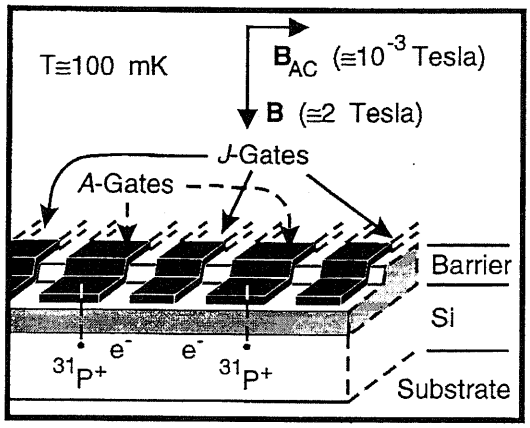


図 2.8: E. B. Kaneによって提案された単一核スピン量子コンピュータの模式図 [21]. A, J 両ゲートによって, 1 量子ビット, 2 量子ビットの演算が可能である.

デコヒーレンスにつよく, 量子ビットの相関を持たせる方法が必要となってくる.

2.4.3 半導体中の核スピンを用いた量子ビット

デコヒーレンスにつよい量子ビットが実現が期待されているもののうちの 1 つが, 核スピンを用いた量子コンピュータである. 核スピンは, 核磁気共鳴をつかって操作, 観測が行なわれるが, 核スピンは外界と良く分離されているため, デコヒーレンス時間が非常に長くなるものと考えられる.

固体中の核スピンを用いる場合, 量子ビットとして用いることのできる原子は核スピン 1/2 を持つ原子である. この量子ビットとして用いる原子を核スピンを持たない原子からなる結晶中に固定して用いる. E. B. Kaneによって提案された方法は, スピンを持たないシリコン同位体である ^{28}Si もしくは ^{30}Si 原子からなるウェハ中にスピン 1/2 を持つ燐原子 ^{31}P を埋め込み, この燐原子 1 個を量子ビット 1 個として扱うものである [21]. そのデバイス構造は, 図 2.8 に示すように SiO_2/Si 界面から 20nm 程度下方に 10–20nm 間隔で ^{31}P を埋め込み, SiO_2 薄膜を介して P 原子の直上に A ゲートと称する電極を設け, この A ゲートの間に J ゲートと称する電極を設けたものである. 核スピンは, 固有の周波数を持っており, これに共鳴する交流磁界によってそのスピンの向きを変えることができる. さらに, ドナーとして働いているこの P 原子の周囲の電子と核スピンは相互作用をしており, これが共鳴周波数に影響を与えている. したがって, 電子の波動関数の分布を A ゲートによって変化させることにより, 共鳴周波数は人為的に変化させることが可能である. この効果を用いて, 特定の P 原子の核スピンを選択的に変化させることができ, 1 量子ビットのゲート操作が実現される.

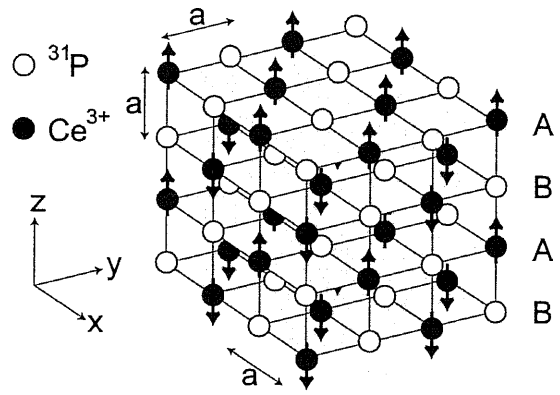


図 2.9: 結晶格子量子ビットに用いられる CeP 結晶構造. P 原子上のスピン 1/2 が計算に用いられる. 図中矢印は $4f$ 電子スピンの量子状態を表現している [24].

一方, J ゲートに電圧を印加するとその下部で電子の波動関数が濃くなり, この電子を介して隣合う 2 つの核スピンの状態を結合することができる. ここで, 結合状態に共鳴する交流磁界を用いてスピンを操作することにより, CNOT に対応した物理操作が可能となる. 提案によれば, 低温で系を動作させることによりデコヒーレンス時間は 10^6s にもなり, 少なくとも 10^5 回からおそらく 10^{10} 回程度のゲート操作が可能になると予想される.

この系はしかし, 原子 1 つを扱うことから, スピンを持つシリコンの同位体 ^{29}Si や結晶欠陥を完全に排除するような高度な技術が要求される. このため現在のところ実験をできる状態にはなく, 個々の要素技術を開発する段階にある.

一方, 固体中の核スピンを同じように用いるが, 原子単体ではなく, 結晶格子中に存在する ^{31}P を利用して量子ビットを実現する方法が提案されている [24]. これは図 2.9 のような NaCl 構造をもつセリウム燐 (CeP) 単結晶を利用するものである. Ce はスピンを持たない原子であり, P も ^{31}P 以外の同位体を持たないため, 結晶中の格子点に整列した P 原子の核スピンを大量に利用して, 多数の量子ビットからなる量子コンピュータを実現できる. CeP の結晶に対し, $x-y$ 面上に均一な z 軸方向への静磁界 B_z を印加すると, 交流磁界によって全ての P 原子に共通のスピン操作が可能であるが, 静磁界を $x-y$ 面上で勾配をつけた $B_z(x,y)$ という形にすると共鳴周波数に空間依存性が出るため, これに共鳴する周波数を持つ交流磁界によって, 結晶中一部分の原子の核スピンを任意に操作できるようになる. また, $B_z(x,y)$ のように 2 次元的な勾配にしておくことにより, z 方向には共通の性質を持ったスピンが線状に並ぶため, スピンの状態を観測する時に 1 個の原子から得られる信号よりも強い信号が得られる. 結晶中には規則的に核がならんでいるため, 相隣る 2 つのスピンは互いに結合する. この結合を断ち切るために, 例えば常に片方のスピンを反転し続けるような方法で結合

を解消しておく．この系で量子ビットにゲート操作を行なう場合は，対象となる核に結合解除用の交流磁界を加えるのを中止し，ゲート操作用に対応した周波数の交流磁界を与える．2つの量子ビットに跨るCNOTゲート操作を行なう場合は，対象となる隣あった核に再び結合を起こさせ，然るべき周波数の交流磁界を与える．このようにして情報処理が可能になる．量子ビットの読み出しは，異なる原子層の核スピンとの相互作用，あるいはCe原子に存在する電子との相互作用を利用する．この系のデコヒーレンス時間は山口らによれば，100s程度になると予想されている[24].

また，ごく最近，同様な原理を用いた全シリコン量子ビットが提案された[25]．これは，核スピンのない^{28}Siのみからなる，$(\bar{1}\,\bar{1}\,2)$面に対し1°傾いた(111)基板に形成されるテラス構造上に，スピン1/2を有する^{29}Siを堆積させて原子鎖をつくり，これを1つの量子ビットとして利用するものである．

これら2つの提案の長所は，デコヒーレンス時間の非常に長い量子ビット数を大量に入手できる点であるが，同時に短所もある．それは，量子ビットが多数の原子から形成されるため，系は混合状態になってしまうことである．このような系からは強い信号が発生される代わりに，観測がアンサンブル平均の形でしか得られないことである．このため，十分な信号対雑音比を得るためには観測を繰り返さなければならなくなり，アルゴリズムによっては，正しい結果を得るまでに指数時間がかかってしまう場合もあることが指摘されている．この問題を解決するアルゴリズム的な方法が必要である．

2.5 量子コンピュータ研究におけるシミュレーションの意義

NP問題を解くことが可能ではないかと期待される量子コンピュータの実現に向けて，このように様々な方法が提案されているが，その実現へ向けての取り組みには技術課題が多く，大変時間のかかる研究になると予想される．

開発を困難にしているのは，第1にデコヒーレンスの問題である．単一クーパー対箱のように人工的に固体中に2状態系をつくる場合，これを外部から制御できるが，デコヒーレンスが大きくなると予想される．第2に，人工2状態系で量子ビットの相関を発生させる機構を実現することの難しさである．これらを克服し，年次に対し指数的に量子ビットの集積数を増大できなければ，ショアのアルゴリズムのように大量に量子ビットを用いるアルゴリズムを実現するのは難しい．一方，結晶格子中の核スピンを用いる場合は単一の原子核スピンを用いる場合よりも実現可能性は大きいと思われるが，アルゴリズムにおける観測ついて難があるため，混合状態を使った場合のアルゴリズムを新しく考えていく必要がある．

このような事情から，アルゴリズムの研究が大変重要性を帯びてくる．量子アルゴリズムの研究の意義は多岐に渡る．

(1) 量子コンピュータの計算能力をより正確に把握すること．特に，現在まで有効と判明しているのは，ショアのアルゴリズムのみであり，NP問題のうちの1つが解けたに過ぎ

ず, 量子コンピュータの使い道が他にあるのか正確に把握しなければならない. またNP完全問題については, グローバーにより示されたオラクルを用いる計算モデル以外の方法によって, 多項式時間アルゴリズムが組めるかどうか, 理論面からも研究しなければならない. 量子コンピュータの物理系での実現方法に関して今後膨大な知識の集積を行なう動機付けのためにも, アルゴリズム研究は大変重要である.

(2) 少数の量子ビット数で有効に動作する量子アルゴリズムの可能性について議論すること. デコヒーレンスなど技術的な限界点があるため, ショアのアルゴリズムでの議論に見られるように量子ビット数が入力の多項式で抑えられるだけでは不十分であり, 限られた量子ビット数で動作するアルゴリズムの実装方法を考える必要がある. アルゴリズム, 物理の両方から量子ビット数の問題に取り組まなければならない.

(3) NMR量子コンピューティングにおいて, 観測がアンサンブル平均を利用することによって起因するアルゴリズムの速度低下を抑える, 別のアルゴリズムを考えること. これは, 混合状態としてモデル化される系の理論的研究と同時に進められなければならない.

このようなアルゴリズム研究において, 特に重要性を増してくるのはシミュレーション技術である.

また, デコヒーレンスの影響を考察する上でも量子コンピュータのシミュレーションは大変重要である. 想定した系で使われる量子ビット単体のデコヒーレン時間は実験からおおよそ予想されるが, それが量子ビットを多数連結した系でどのような増加を示すかは計算を行なわなければ分からない. 想定した実現形式が実際的な量子コンピュータに応用できるかを判断する材料として, この情報は必須といえる.

シミュレーションに要求される技術課題としては, 大規模な量子コンピュータのシミュレーションを高速に処理することである. しかし, 量子コンピュータで行なわれる計算はその性質上, シミュレーション規模を大きくすれば指数関数的な計算量の増大を招くことになる. さらに, 実際の物理を踏まえた量子コンピュータのシミュレーションを行なう場合, 量子ビット数が増えた場合, 結合する雑音要素がその計算量の増大は更に深刻なものとなる.

量子コンピュータのシミュレーションは, 現在のところ汎用のワークステーションによって行なわれるのが一般的である. これまでに, 汎用ワークステーション Sun Enterprice 4500によって1時間で30bitの因数分解をショアのアルゴリズムにより実行できる汎用ソフトウェアシミュレータが発表されている [26]. しかし, 汎用プロセッサシステムによる計算は, 計算手法がハードウェアにより制限されるため, デバイスの並列動作による計算の高速性を十分に引き出すことができない. さらなる高性能なシミュレーションを行なうためには, ハードウェアの開発の段階で量子コンピュータのシミュレーションに最適化したアーキテクチャを導入する必要がある. このアーキテクチャが集積回路上のデバイスを効率的に利用できるものであれば, シミュレータの性能は大変高くなり, 集積回路上のデバイスの微細化, 高性能化

によって計算速度を飛躍的に上昇させていくことが可能となる.

2.6 まとめ

本章では, 本研究の動機となっている量子コンピュータの理論について概観し, 様々な物理的実現形式の提案を紹介した. 現在のところ, 実際に量子コンピュータの実現に関して多くの技術課題が存在していることを考えると, シミュレーションを利用した量子コンピュータの研究が重要性を帯びてくることが指摘される. 量子効果を用いない古典的な計算システム上で大規模な量子計算を高速に再現する手法について考える必要がある. この方法は, 現在のコンピュータの根幹となる集積回路での実現に適したものでなければならない.

第 3章

ディジタルフィルタ群を用いた量子回路エミュレータ

3.1 はじめに

量子コンピュータの研究において, シミュレーションが重要な位置を占めてくることについて, 前章で述べた. 本研究でも古典的な計算機上で量子計算を再現することを重要視し, これについて集積回路の立場から解を提案することに取り組む.

量子コンピュータの動作を再現する上で重要な事項としては, 重ね合わせをまとめて処理可能であること, そして n ビットの論理を表す計算装置全体が関連性を持って動作していることである. 集積回路のような古典的な計算デバイス上でこれら 2 つを同時に満たすようなものを作ることは, 不可能であるが, 量子重ね合わせに類似した古典的な重ね合わせを実現することは可能と考えられる.

本章では, LSI 上で波動関数がユニタリ変換によって変化していく過程をディジタルフィルタを用いて模倣する量子コンピュータのエミュレータについて述べる. 本エミュレータは, 状態の重ね合わせをディジタルの周期信号として表し, これを周波数変換器と FIR(Finite Impulse Response) フィルタを組み合わせた系によって変化させ, 量子演算に対応した計算を行うものである.

3.2 量子コンピュータエミュレータ

3.2.1 エミュレータ上での状態表現

LSIにおける情報処理は, あるノードの電圧値によって行なわれる. すなわち, LSIは古典的な計算装置のうちの 1 つといえる. この LSI 上で, 重ね合わせ状態に対する単一の計算プ

ロセスを再現するために, 古典的な重ね合わせ状態を利用する.

すでに述べたように, n 量子ビットの量子回路には 2^n の状態が存在する. これらは単一の n ビット論理値と対応し, 1つのまとまった量子重ね合わせ状態の中に任意の重みを持って保存される. これに対し本ハードウェアエミュレータにおいては, 単一の n ビット論理値と回路中の単一の周期信号を一対一対応させる. すなわち,

$$|a\rangle \longleftrightarrow e^{j\omega_a t}, \tag{3.1}$$

のような対応関係を定義する. ただし, $a = 0, 1, 2, \ldots, 2^n - 1$ であり, ω_a は角周波数を表す. このとき, 状態 $|a\rangle$ の確率振幅 c_a は, スペクトル成分 $e^{j\omega_a t}$ の複素振幅 $X(a)$ に対応付けられる. よって, 重ね合わせ状態 $|\psi\rangle$ は時間依存の周期信号 $x(t)$ によって表現される.

$$|\psi\rangle = \sum_{a=0}^{2^n-1} c_a|a\rangle \longleftrightarrow x(t) = \sum_{a=0}^{2^n-1} X(a)e^{j\omega_a t}. \tag{3.2}$$

ここで $\{c_a \mid a = 0, 1, 2, \ldots, 2^n - 1\}$ および $\{X_a \mid a = 0, 1, 2, \ldots, 2^n - 1\}$ は複素数で

$$\sum_{a=0}^{2^n-1} c_a c_a^* = \sum_{a=0}^{2^n-1} X(a)X^*(a) = 1 \tag{3.3}$$

のように正規化されているとする. 式 (3.3) は, 各状態の観測確率が信号のパワースペクトルの大きさに対応付けられることを示している. 以上の対応付けは, 量子系の 2^n 次元ヒルベルト空間を, 2^n の離散周波数を基本ベクトルとする空間へ単純に写像を行なっていることを意味する.

係数の組 $\{X(a) \mid a = 0, 1, 2, \ldots, 2^n - 1\}$ は, 2^n 個で完全な周期をなす離散時間信号系列の離散フーリエ変換 (DFT, Discrete Fourier Transform) により得られる. 式 (3.2) に対応して, この時系列を

$$\{x(k) \mid k = t/\Delta,\ k = 0, 1, 2, \ldots, 2^n - 1\} \tag{3.4}$$

と表現する. ただし Δ は2つの連続する信号間の時間間隔を表し, これを用いて角周波数 ω_a は

$$\omega_a = \frac{2\pi a}{2^n \Delta} \tag{3.5}$$

と書き下される. このような性質から, 本エミュレータは離散時間システムの形で実現される. ここで, $\{x(k)\}$ は $\{X(a)\}$ と同様に複素数である. 離散信号処理を行なう系ではしばしば実部のみが扱われるが, 本エミュレータでは時系列の実部, 虚部両方を利用する. これにより, 任意の $X(a)$ について $X(-a) = X^*(a)$ というような対称性を生ずることなく, 独立に表現することが可能になる. これは後述する回路構成に深く関わる.

以下においては, 任意の状態は

$$|X\rangle = \begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ \vdots \\ X(N-1) \end{bmatrix} \tag{3.6}$$

のようにベクトルのかたちで表記することとする. ただし, $N \equiv 2^n$ と定義する.

3.2.2 並列FIRフィルタによる万能量子ゲート

量子回路における演算処理を表現するには, 入力状態 $|X\rangle$ を出力状態 $|Y\rangle$ に変換する伝達特性を持った回路機能が必要となる. 本エミュレータでは, これをフィルタ回路の集合体によって実現した. 一般的な $N \times N$ ユニタリ行列 $\hat{U}$ による変換

$$|Y\rangle = \hat{U}|X\rangle \tag{3.7}$$

の機能は, 図3.1に示すような, N 個の並列な複素FIR(finite impulse responce)フィルターと "C" と書かれた $\{X(a)\}$ の要素を巡回シフトする操作, すなわち巡回的周波数変換回路により実現できる. この回路機能は次のように説明される.

いま, $\hat{U}$ を行列要素 u_{ab} に書き下すと任意の $|Y\rangle$ の要素が

$$Y(a) = \sum_{b=0}^{N-1} u_{ab} X(b) \tag{3.8}$$

のように表されることをフィルタ回路により表現したい. しかし, FIRフィルタの周波数特性を $\{H(a) \mid a = 0, 1, 2, \ldots N-1\}$ と表すと, FIRフィルタ単体ではその線形性のために

$$Y(a) = H(a)X(a) \tag{3.9}$$

のようにある周波数成分の振幅を変化させることしかできない. そこで, 図3.1のようにFIRフィルタを N 個用意し, そのうちの i 番目 $\{H_i(a)\}$ の入力には式(3.10)で表される周波数変換 "C" を i 回繰り返し施したものを与えることにする. この "C" 単体では

$$X(a) \mapsto X(a-1)_{\bmod N}, \tag{3.10}$$

のような演算を行なう. ただし上式は $\{x(k)\}$ の周期性による

$$X(-l)_{\bmod N} = X(N-l), \forall l \in \mathbf{Z} \tag{3.11}$$

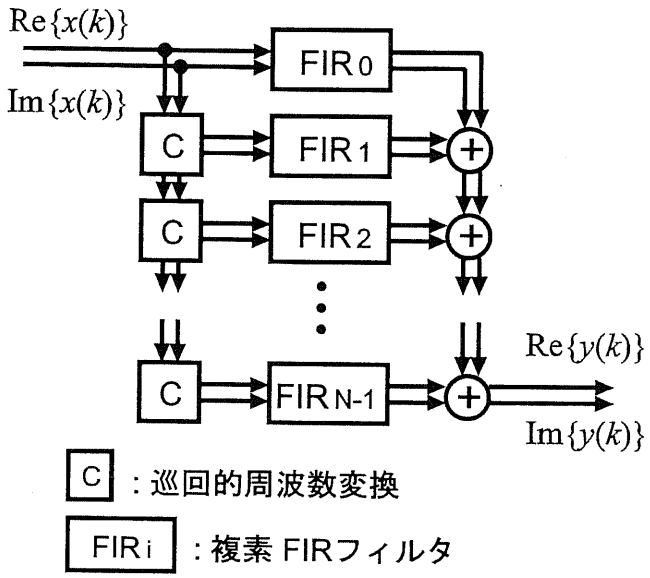


図 3.1: ユニタリ変換 $\hat{U}$ の機能を表現する回路構成. N 個の並列な FIR フィルタと $\{X(a)\}$ の巡回的周波数変換 "C" から成る.

という性質を有する. この N 個の FIR の全出力を加算することにより,

$$Y(a) = \sum_{i=0}^{N-1} H_i(a) X(a-i)_{\bmod N} \tag{3.12}$$

を得ることとなり, 式 (3.8) と対応をとることが可能となる. つまりこの回路においては, ユニタリ変換の要素 $\hat{U} = [u_{ab}]$ が FIR フィルタの周波数特性成分 $\{H_i(a) \mid i, a = 0, 1, 2, \ldots, N-1\}$ により,

$$u_{ab} \longleftrightarrow H_{(a-b)_{\bmod N}}(a) \tag{3.13}$$

のような対応関係を持って表現されている. これを行列の形で表現すれば,

$$\hat{U} = \begin{bmatrix} H_0(0) & H_{N-1}(0) & H_{N-2}(0) & \cdots & H_2(0) & H_1(0) \\ H_1(1) & H_0(1) & H_{N-1}(1) & \cdots & H_3(1) & H_2(1) \\ H_2(2) & H_1(2) & H_0(2) & \cdots & H_4(2) & H_3(2) \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ H_{N-1}(N-1) & H_{N-2}(N-1) & H_{N-3}(N-1) & \cdots & H_1(N-1) & H_0(N-1) \end{bmatrix}. \tag{3.14}$$

のようになる.

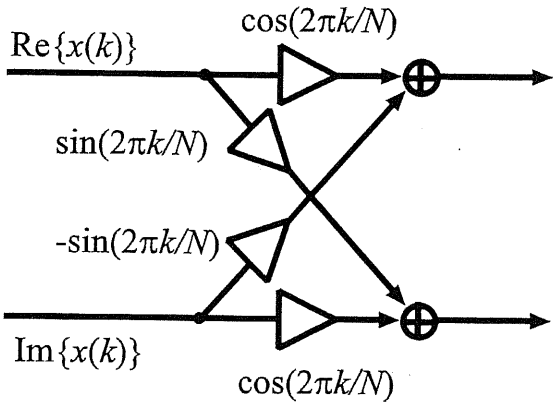


図 3.2: 巡回的周波数変換の回路構成. 一対の局部発振器からの信号がゲインとして入力信号の実部, 虚部に乗算される.

巡回的周波数変換は, 図 3.2 に示す通り, 一組の時間依存信号源 $\sin(2\pi k/N), \cos(2\pi k/N)$ により実現される. これは, 良く知られているように式 (3.10) の逆 DFT が

$$X(a-1)_{\bmod N} \overset{IDFT}{\longmapsto} x(k)e^{j2\pi k/N}. \tag{3.15}$$

によって与えられることから導かれる.

本エミュレータのアーキテクチャは, 離散時間系であればどのようなディジタルフィルタで構成しても, スイッチトキャパシタフィルタで構成しても実現可能である. ただし, 実現方法それぞれに付きまとう誤差が存在するので, 注意を要する.

3.3 ディジタルフィルタ群を用いたエミュレータシステム

エミュレータをディジタルフィルタによって実現した. この構成を図 3.3 に示す. 前述の並列 FIR フィルタは複素ディジタルフィルタとして実装した. FIR 係数は, プログラムとして与えられるゲート操作のシーケンスに基づいてパソコンで計算され, 周期信号を計算する前に専用回路にロードされる. 変換係数のロード終了後に, 初期状態を表すディジタルの周期信号を専用回路に送る. この初期状態は専用回路内で変換され, 終状態がパソコンに戻される. 戻されたディジタル周期波形はパソコン内で高速フーリエ変換され, これにより観測確率に相当するデータに戻される.

専用回路を PLD(Programmable Logic Device) に実装した. 回路の仕様を表 3.1 に示す. 実際の回路では同期設計を容易化するために, 図 3.1 に "C" として示す周波数変換器を縦列に 8 個繋げた部分を, 8 個の異なる局部発信機を用いて構成された周波数変換器で実現した. すな

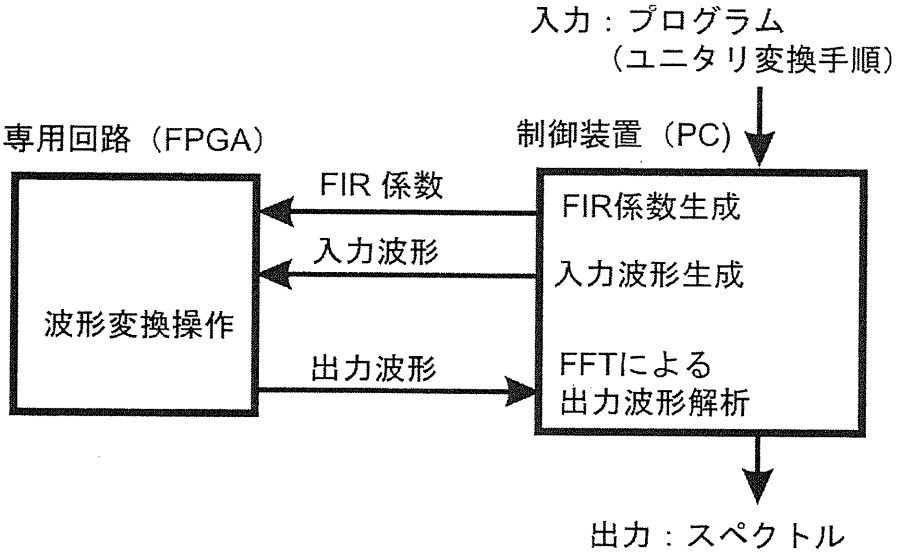


図 3.3: エミュレータシステムの構成．PC と FPGA 上に実現される専用回路からなる．

わち，k 番目の FIR に入力される信号は，処理に先立ち複素数 $\cos(2\pi k/8)+i\sin(2\pi k/8)), k = 0,1,2,\ldots,8$ を乗算されることにより，k 回 "C" を通ったのと等価になる．FIR 内部回路は，図 3.3 のように実現した．数値の扱いを符号付固定小数点 8 ビット長とし，最上位ビットを符号ビット，次のビットを 2^{-1} 位と定めた．これは，確率振幅が必ず $\omega\omega^* \leq 1$ に収まることを意識したものである．量子化は量子化幅 $\Delta = 2^{-7}$ の線形量子化，丸めは零捨一入により行った．計算の内部精度は 2 ビットを余分に設けた符号付 10 ビットとした．

表 3.1: エミュレータの主要仕様

PLD タイプ	Altera EPF10K250A-3
PLD 使用数	8 枚
量子ビット数	3
ゲート数	1.0×10^6
クロック速度	5MHz

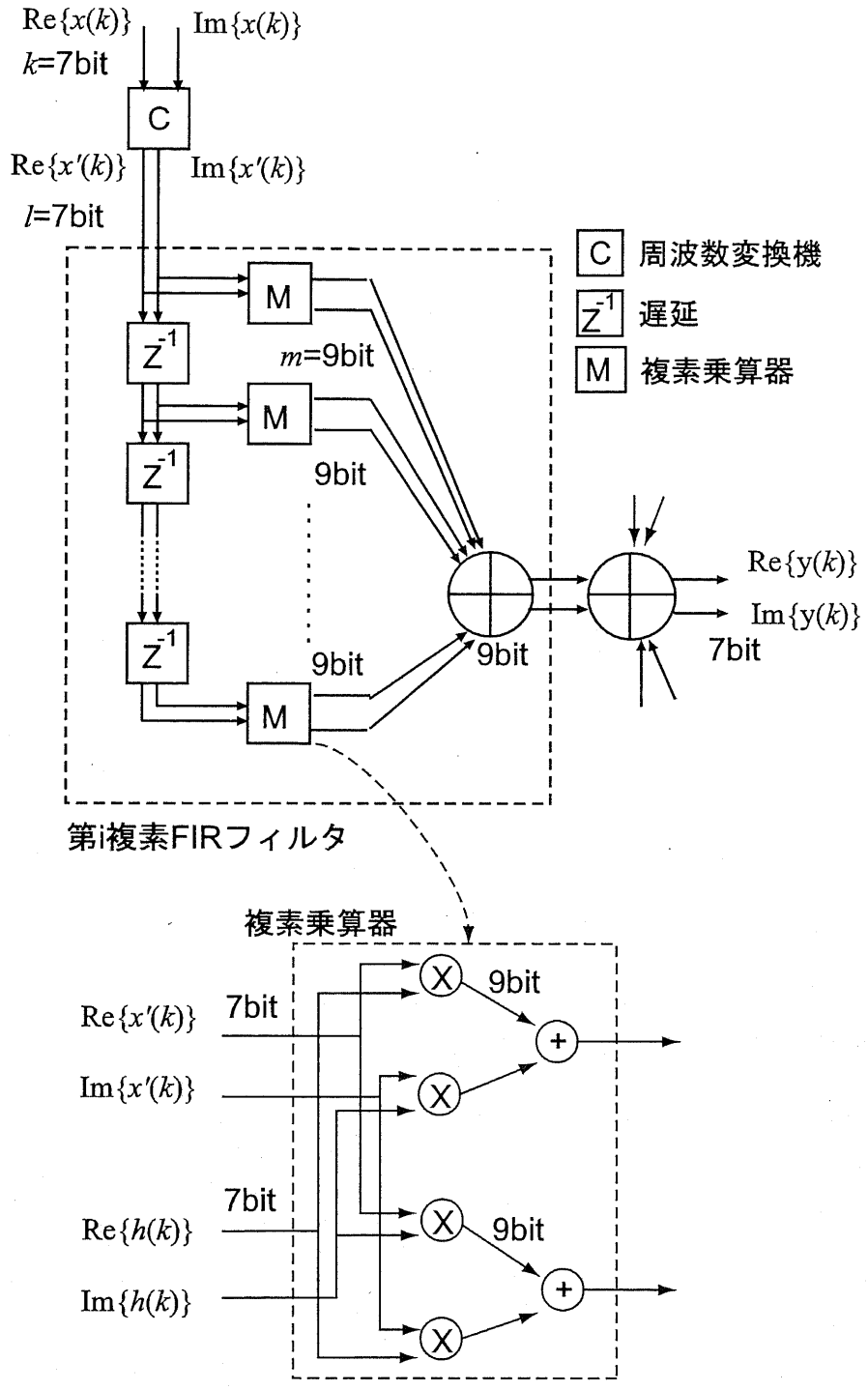


図 3.4: FIR フィルタの内部構成．外部入力は符号付 8 ビット，量子化幅 $\Delta = 2^{-7}$ であり，乗算器出力以下は符号付 10 ビット，量子化幅 $\Delta = 2^{-9}$ である．丸めはすべて零捨一入で行った．

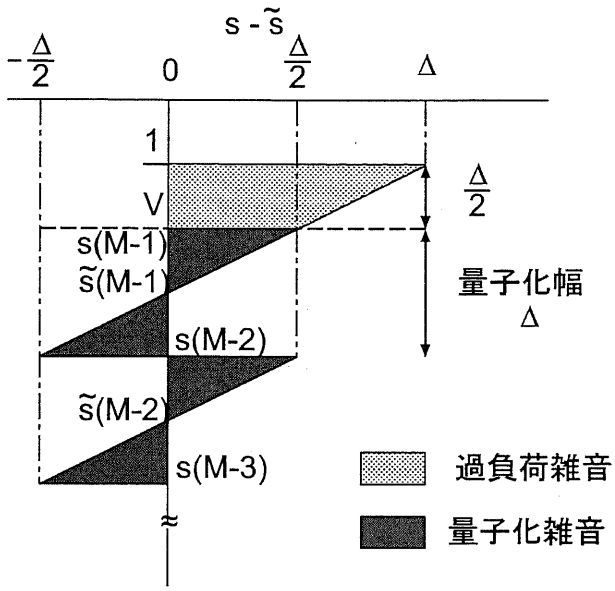


図 3.5: 誤差の分布の原理．量子化幅 Δ で $|s| < 1$ の範囲を量子化した場合．丸めは零捨一入を仮定．

3.4 エミュレータの評価

3.4.1 計算誤差の評価

ディジタルフィルタでエミュレータを構成した場合の精度について評価する．ディジタルフィルタ内では一般的に真値を有限ビット長のディジタル値で表現するため，計算には誤差が入り込む．この誤差は，ビット長，内部精度，丸め方法などに依存し，その結果生ずるのは量子化誤差 e_q と過負荷誤差 e_o の 2 種類に分類される．以下では数式上で現れる真値を s，ディジタル値を $\tilde{s}$ と表示する．エミュレータにおける数値の扱いを量子化幅 $\Delta = 2^{-7}$ の線形量子化，丸めを零捨一入によって行ったので，誤差 $e(= s - \tilde{s})$ は図 3.4.1 のように分布する．量子化方法および丸め方法より，量子化が規則的に行われる範囲は $|s| < 1 - \Delta/2$ であり，この範囲で起こる誤差が量子化誤差 e_q である．また，この領域外では値はすべて $\pm(1-\Delta)$ によって代表され，このときに生ずる誤差が過負荷誤差 e_o である．今，真値 s の出現頻度に偏りがないと仮定すると，量子化誤差の確率分布は一定となる．このとき，誤差の分散，すなわち雑音電力の平均値はどの代表値 $\tilde{s}$ の周りでも

$$E(e_q) = \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} e_q^2 de_q = \frac{\Delta^2}{12} \equiv \epsilon_q \tag{3.16}$$

と一定になる．ただし，本論分では $E(\cdot)$ を自乗平均値として定義する．同様に過負荷雑音の平均値も，本エミュレータにおいては

$$E(e_{\rm o}) = \frac{2}{\Delta}\int_{\Delta/2}^{\Delta} e_{\rm o}^2 {\rm d}e_{\rm o} = \frac{7\Delta^2}{12} \equiv \epsilon_{\rm o} \tag{3.17}$$

となる．

回路全体での誤差の累積と語のビット長の関係について考える．回路への入力値の真値の実部および虚部を $(x_{\rm r}, x_{\rm i})$，周波数変換におけるディジタル局部発信機から出力される信号の真値を $(c_{\rm r}, c_{\rm i})$, FIR フィルタ係数の真値を $(h_{\rm r}, h_{\rm i})$，期待される回路の出力値を $(y_{\rm r}, y_{\rm i})$ と表記し，それぞれの計算機内部での値を $(\tilde{x}_{\rm r}, \tilde{x}_{\rm i})$, $(\tilde{c}_{\rm r}, \tilde{c}_{\rm i})$, $(\tilde{h}_{\rm r}, \tilde{h}_{\rm i})$, $(\tilde{y}_{\rm r}, \tilde{y}_{\rm i})$ と表す．これらはそれぞれ誤差 $(\delta x_{\rm r}, \delta x_{\rm i})$, $(\delta c_{\rm r}, \delta c_{\rm i})$, $(\delta h_{\rm r}, \delta h_{\rm i})$, $(\delta y_{\rm r}, \delta y_{\rm i})$ を持っている．入力 $(\tilde{x}_{\rm r}, \tilde{x}_{\rm i})$ を周波数変換回路に入力し，複素乗算を行なう．すなわち，4つの部分積 $\tilde{x}_{\rm r}\tilde{c}_{\rm r}$, $\tilde{x}_{\rm i}\tilde{c}_{\rm i}$, $\tilde{x}_{\rm r}\tilde{c}_{\rm i}$, $\tilde{x}_{\rm i}\tilde{c}_{\rm r}$ を求める場合は，例えば

$$\begin{aligned} & E(\tilde{x}_{\rm r})E(\tilde{c}_{\rm r}) \\ &= E(x_{\rm r}+\delta x_{\rm r})E(c_{\rm r}+\delta c_{\rm r}) \\ &= \{E(x_{\rm r})+E(\tilde{x}_{\rm r})\}\{E(c_{\rm r})+E(\delta c_{\rm r})\} \\ &\sim E(x_{\rm r})E(c_{\rm r})+E(x_{\rm r})E(\delta c_{\rm r})+E(c_{\rm r})E(\delta x_{\rm r}) \end{aligned} \tag{3.18}$$

という自乗平均値を持つことになる．ただし，δ の自乗の項は省略した．入力値 $(x_{\rm r}, x_{\rm i})$, $(c_{\rm r}, c_{\rm i})$, $(h_{\rm r}, h_{\rm i})$ を仮数部 k ビットで量子化した時，$\tilde{x}_{\rm r}\tilde{c}_{\rm r}$ などの部分積は $2k$ ビットとなるから，これを 2^{-l} の桁までで丸めて続く加算を行い，複素数の乗算を完了する．ここまでが周波数変換操作になる．このとき，加算において新しい誤差が介入する．この誤差の自乗平均値を $\epsilon_{\rm q1}$ とおくと，結局周波数変換器の出力 $(\tilde{x'}_{\rm r}, \tilde{x'}_{\rm i})$ の自乗平均値は，

$$\begin{aligned} E(\tilde{x'}_{\rm r}) &= E(\tilde{x}_{\rm r})E(\tilde{c}_{\rm r})+E(\tilde{x}_{\rm i})E(\tilde{c}_{\rm i})+2\epsilon_{\rm q1} \\ &\sim 2E(x)E(c)+2E(x)E(\delta c)+2E(c)E(\delta x)+2\epsilon_{\rm q1} \end{aligned} \tag{3.19}$$

などと与えられる．ここで，簡単化のために，すべての変数は実部，虚部とも同じ信号分布をしていると仮定し，これらを x, c, h および $\delta x, \delta c, \delta h$ という記号で代表することとする．

ここで，$E(x)$, $E(c)$, $E(h)$ および $E(\delta x)$, $E(\delta c)$, $E(\delta h)$ について考える．本研究では，時系列と周波数領域すなわち量子系の状態ベクトル X の関係を結ぶ離散フーリエ変換として

$$X(k) = \sum_{i=0}^{N-1} x(n)e^{-j2\pi ki/N} \tag{3.20}$$

$$x(k) = \frac{1}{N}\sum_{i=0}^{N-1} X(k)e^{j2\pi ki/N} \tag{3.21}$$

を用いた．今，確率の和すなわち $X(k)$ の自乗和が 1 であることを考えると，(3.21) 式から $E(x_\mathrm{r}) = E(x_\mathrm{i}) = E(x) = 1/2N^2$ となる．また，cについては交流信号 $e^{j\omega t}$ を掛ける操作なので，電力としては常に 1 であり $E(c_\mathrm{r}) = E(c_\mathrm{i}) = E(c) = 1/2$ となる．hについては変換行列がベクトルの長さを変えないユニタリ変換であることを考えると，$E(h_\mathrm{r}) = E(h_\mathrm{i}) = E(h) = 1/2N^2$ となる．一方，誤差について考えると，x_r および x_i の最大値は高々$1/N$，h および c の最大値は 1 となるから，過負荷誤差の現れる可能性があるのは h と c である．h, cが量子化の可能なレンジから出る頻度を p_h, p_c とおくと，結局誤差は，

$$E(\delta x) = \epsilon_{\mathrm{q}0} \tag{3.22}$$

$$E(\delta h) = (1-p_h)\epsilon_{\mathrm{q}0} + p_h\epsilon_\mathrm{o} \tag{3.23}$$

$$E(\delta c) = (1-p_c)\epsilon_{\mathrm{q}0} + p_c\epsilon_\mathrm{o} \tag{3.24}$$

となる．ただし，ϵ_q は，入力値 x, c, h を 2^{-k} の桁までで丸めを行ったことによる量子化誤差を示す．以上より，周波数変換器の出力における誤差 $\delta x'$ は

$$\begin{aligned} E(\delta x') &\sim 2E(x)E(\delta c) + 2E(c)E(\delta x) + 2\epsilon_{\mathrm{q}1} \\ &= \frac{1}{N^2}\{(1-p_c)\epsilon_{\mathrm{q}0} + p_c\epsilon_\mathrm{o}\} + \epsilon_{\mathrm{q}0} + 2\epsilon_{\mathrm{q}1} \end{aligned} \tag{3.25}$$

と近似できる．

周波数変換器出力 x' は FIR に入力され，内部の各乗算器によって h 倍され，乗算器の出力が $N = 2^n$ 項足し合わせられ FIR より出力される．さらに FIR は N 個あり，すべての FIR 出力の和をとって回路の出力が得られる．すなわち，回路出力における信号は FIR の乗算器出力を N^2 倍したものであり，

$$\begin{aligned} &2N^2\{E(\tilde{x'})E(\tilde{h}) + \epsilon_{\mathrm{q}2}\} \\ &\quad\sim 2N^2\{E(x')E(h) + E(x')E(\delta h) + E(h)E(\delta x') + \epsilon_{\mathrm{q}2}\} \end{aligned} \tag{3.26}$$

となる．ただし，$\epsilon_{\mathrm{q}2}$ は，乗算器出力結果を 2^{-m} の桁までで丸めることによるよって生ずる誤差である．結局誤差の累積は

$$E(\delta y) = E(\delta h) + E(\delta x') + 2N^2\epsilon_{\mathrm{q}2} \tag{3.27}$$

となる．この結果をフーリエ変換するため，パワースペクトルに現れる誤差は

$$\epsilon = 2N^2E(\delta y) \tag{3.28}$$

である．

結果として得られるスペクトルに関して考える．正解として得たい状態のパワースペクトルの真値を P_0 とおくとき，他の状態のパワースペクトルの合計は $1 - P_0$ となる．これに誤

差 ϵ が加わるとき，正解の状態は $P_0+\epsilon$, その他の合計は $1-P_0+(N-1)\epsilon$ となる．この2つの比が1以上，つまり誤差を含めた上での答の観測確率が $1/2$ 以上であるという条件を達成するには

$$P_0+\epsilon > 1-P_0+(N-1)\epsilon \tag{3.29}$$

を満たさなければならない．ここから誤差 ϵ は

$$\epsilon < \frac{2P_0-1}{N-2} \tag{3.30}$$

に収まることが必要となる．

ここから各部に必要な精度を求めると，

$$E(\delta y) < \frac{2P_0-1}{2N^2(N-2)} \tag{3.31}$$

より，

$$\begin{aligned}&\left\{(1-p_h)N^2+(1-p_c)+1\right\}(N-2)\epsilon_{\mathrm{q0}}\\&\quad+\left\{p_hN^2+p_c\right\}(N-2)\epsilon_0+2(N-2)\epsilon_{\mathrm{q1}}+2N^2(N-2)\epsilon_{\mathrm{q2}} < P_0-\frac{1}{2}\end{aligned} \tag{3.32}$$

という関係が得られる．近似として，N^3 が付く項のみを扱うと，

$$(1-p_h)N^3\epsilon_{\mathrm{q0}}+p_hN^3\epsilon_{\mathrm{O}}+2N^3\epsilon_{\mathrm{q2}}[\equiv\epsilon_{\mathrm{obs}}] < P_0-\frac{1}{2} \tag{3.33}$$

という関係が最終的に得られる．ここから，周波数変換器における乗算器出力の丸めの影響は小さく，もっとも大きい効果を持っているのは，FIR内部の乗算器出力の丸め方法，および h の丸め方法であることがわかる．ここで (3.33) 式の左辺は正解の状態を観測する際に入り込む誤差と言える．この左辺を ϵ_{obs} とおくと，$P_0-\epsilon_{\mathrm{obs}}$ が $1/2$ 以上の時に正しい状態を観測できることになる．ϵ_{obs} を図示すると，図 3.6 のようになる．式および図からわかるように，n が大きい時には k を大きくしなければ十分な精度が得られないことがわかる．n と k はおおよそ比例関係にある．また，内部精度 m には入力精度 k に対応して適当なビット幅が存在する．今回作製したエミュレータの規模 $n=3$ に対し，$k=7, m=9$ という設定はグラフから適当であると言える．

3.4.2　量子ビット数とハードウェア量の関係

ここまでの議論により，量子ビット数 n が増加するとビット精度がこれに比例することが分かるが，量子ビット数 n に対し，FIRと周波数変換器の必要数が 2^n であり，さらにFIRのタップ数が 2^n 必要となるから，これらをあわせると，ハードウェアの必要数は $O(n2^{2n})$

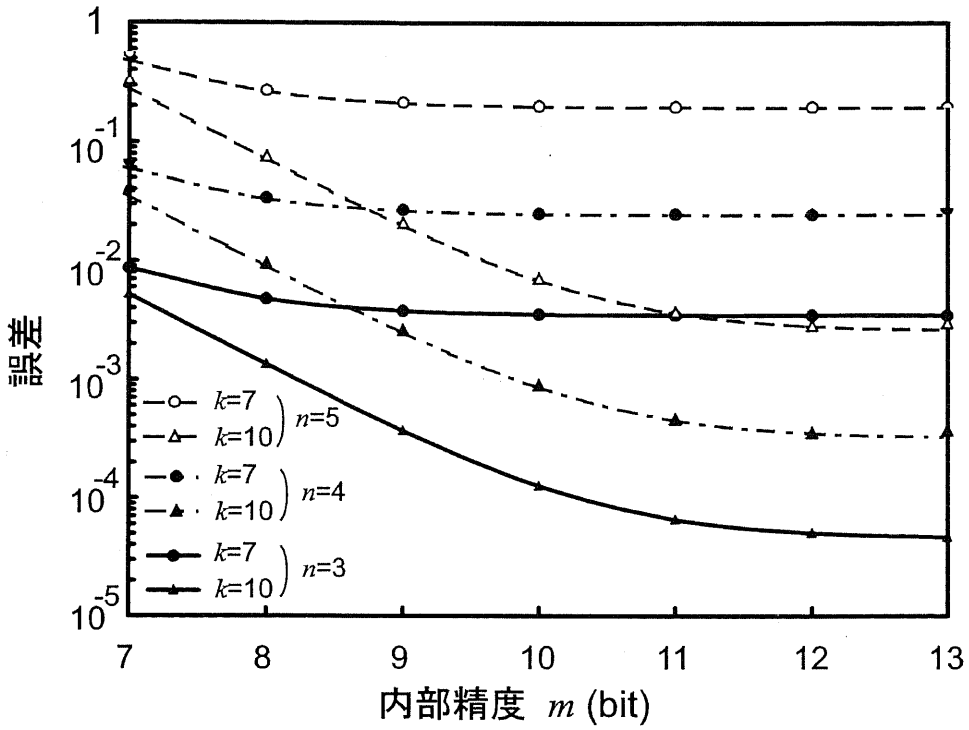


図 3.6: 誤差に対する外部入力値の量子化ビット幅 k と FIR 内部乗算器の出力ビット幅 m の寄与.

必要となる．加えて，演算ステップ数は FIR タップ数が 2^n であり，計算ステップ数もこれと同じ $O(2^n)$ 必要な構造となっている．

これは，2^n 次元ヒルベルト空間で並列に行なわれる演算を古典重ね合わせに置き換え，さらにこれを実時間処理したことにより発生した問題と言える．すなわち，実時間処理をハードウェアで行なう場合，周波数領域では単一であった情報が時間領域においては 2^n で 1 組のタイムスロット全体に展開されてしまうことにより，演算ステップ数も $O(2^n)$ となっている．

3.5 グローバーのアルゴリズムの実験

本エミュレータシステムを用いて実際にグローバーのアルゴリズムを実行した．実験は，3 量子ビットの場合，すなわち $N = 2^3 = 8$ の場合について行った．与える問題は $a =$

$\{0,1,2,\ldots,7\}$ の中から条件

$$f(a) = \begin{cases} 1 & a = a_0 \\ 0 & \text{その他の場合} \end{cases} \tag{3.34}$$

に合致する単一の項目 a_0 を探し出すという単純な問題に置き換えた．グローバーのアルゴリズムで使われる基本的な 3 つのユニタリ変換は 3 量子ビットの場合，ウォッルシュ・アダマール変換

$$W = H \otimes H \otimes H, \tag{3.35}$$

関数 f を表す条件付位相反転行列 S_f

$$S_f = \begin{bmatrix} (-1)^{f(0)} & & & \\ & (-1)^{f(1)} & & \\ & & \ddots & \\ & & & (-1)^{f(7)} \end{bmatrix}, \tag{3.36}$$

そして，位相変換

$$S_0 = \begin{bmatrix} -1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} \tag{3.37}$$

の 3 つである．ただし，

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{3.38}$$

である．

グローバーのアルゴリズムの初段では，まず初期状態 $|X_{\text{init}}\rangle = |0\rangle$ に全基底の重ね合わせを生成する．

$$|X_{\text{init}}\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{W} |X_1\rangle = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \tag{3.39}$$

続いて，$|X_1\rangle$ に変換行列 S_f を掛け，$|a_0\rangle$ の位相を反転する．もし，$a_0 = 3$ だったと仮定すると，以下のようになる．

$$|X_1\rangle \overset{S_f}{\longmapsto} |X_2\rangle = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \tag{3.40}$$

この $|X_2\rangle$ に変換行列 $-WS_0W$ を作用させると基底 $|a_0\rangle$ の確率振幅は増幅される．

$$|X_2\rangle \overset{-WS_0W}{\longmapsto} |X_3\rangle = \frac{1}{4\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 5 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \tag{3.41}$$

ここで用いた変換行列の組み合わせ $-WS_0WS_f$ を合計で $\sqrt{N}$ 回繰り返すことにより，a_0 に対応する基底 $|a_0\rangle$ の観測確率が1に近づく．ここで扱った3量子ビットの場合，繰り返し回数は2回ということになる．3量子ビットのグローバーのアルゴリズムをまとめると，以下のようなユニタリ変換 U にまとめられる．

$$U = -WS_0WS_f(-WS_0WS_f)W. \tag{3.42}$$

この変換の結果，始状態から終状態への遷移は

$$|X_{\text{init}}\rangle \overset{U}{\longmapsto} |X_{\text{fin}}\rangle = \frac{1}{8\sqrt{2}} \begin{bmatrix} -1 \\ -1 \\ -1 \\ 11 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}. \tag{3.43}$$

表 3.2: 各実験における数値処理方法および精度

実験	量子化幅 Δ	内部精度	丸め方法
(a)	2^{-7}	10ビット	零捨一入
(b)	2^{-7}	7ビット	零捨一入
(c)	2^{-7}	7ビット	切り捨て
(d)	2^{-6}	6ビット	切り捨て

のように起こり，正解を観測する確率は 0.95 となる．

この U を PC で FIR の係数に変換して専用回路にロードし，実験を行った．実験は，数値処理方法の異なる 4 種類の回路について行った．これを表 3.2 にまとめる．PC 側から専用回路に渡される $|X_{\rm init}\rangle = |0\rangle$ に対応した信号は振幅 0.125 の直流であり，これが変換を受けると図 3.7 のような交流信号となる．図 3.7 の実部には偶対称性が，虚部には奇対称性がそれぞれ現れている．この時系列波形を PC 内部で FFT したものが $|X_{\rm fin}\rangle$ に対応する．これを図 3.8 に示す．図には実部のみが表示されているが，虚部は現れないので省略した．これは，実時間波形で対象性が現れることと一致している．さらにこの時系列波形のパワースペクトルをとると，図 3.9 を得る．この振幅が理論上の観測確率に対応している．図 3.8,3.9 では，量子化幅 Δ，内部処理精度，および丸めの手法を変化させた場合の各結果をまとめて表示している．

正規化されたパワースペクトルを見ると，内部処理精度 7 ビットで丸めに切捨てを用いた (d) でも解基底のパワーは 1/2 以上となっている．3 量子ビットのグローバーのアルゴリズムを実行するには，このような扱いが限界と考えられる．

3.6 まとめ

量子コンピュータの数理を LSI 上で再現できる量子計算エミュレータを作製し，動作を確認した．本エミュレータは，量子系の基底状態をディジタルの周期信号として表現し，それらの重ねあわせた周期信号を量子系の重ね合わせ状態に対応させ，この周期信号を FIR フィルタの集成回路で変化させ量子コンピュータの演算に対応させるものである．実際に 3 量子ビットの量子アルゴリズムを実行し，これにまつわる誤差について議論した．

本機の性質として，正確なエミュレーションを行おうとする場合デバイス数は $O(n2^{2n})$ 必要であり，計算ステップ数も $O(2^n)$ となる．このため，大きな量子ビット数の量子コンピュータをエミュレートしようとすると計算ステップ数も増えていく．しかし，理論的にはデバイスを $O(2^n)$ 持っているため，さらに計算を効率化し，ステップ数を $\mathrm{Poly}(n)$ のオーダーに減らすことが可能なはずである．別の方式を検討する必要がある．

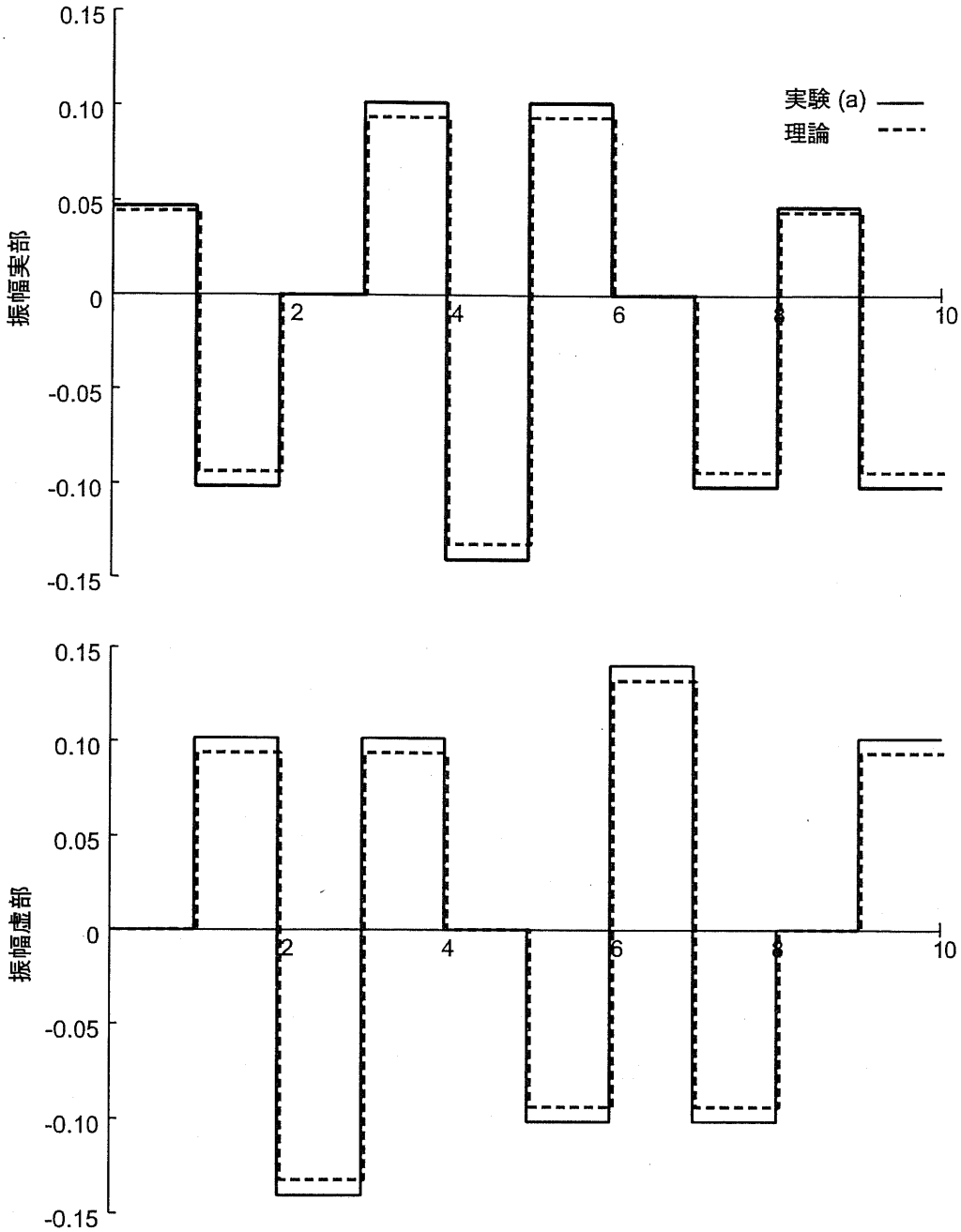


図 3.7: グローバーのアルゴリズムを実行した場合に得られる時系列信号．実部に偶対称性，虚部に奇対称性が生ずる．

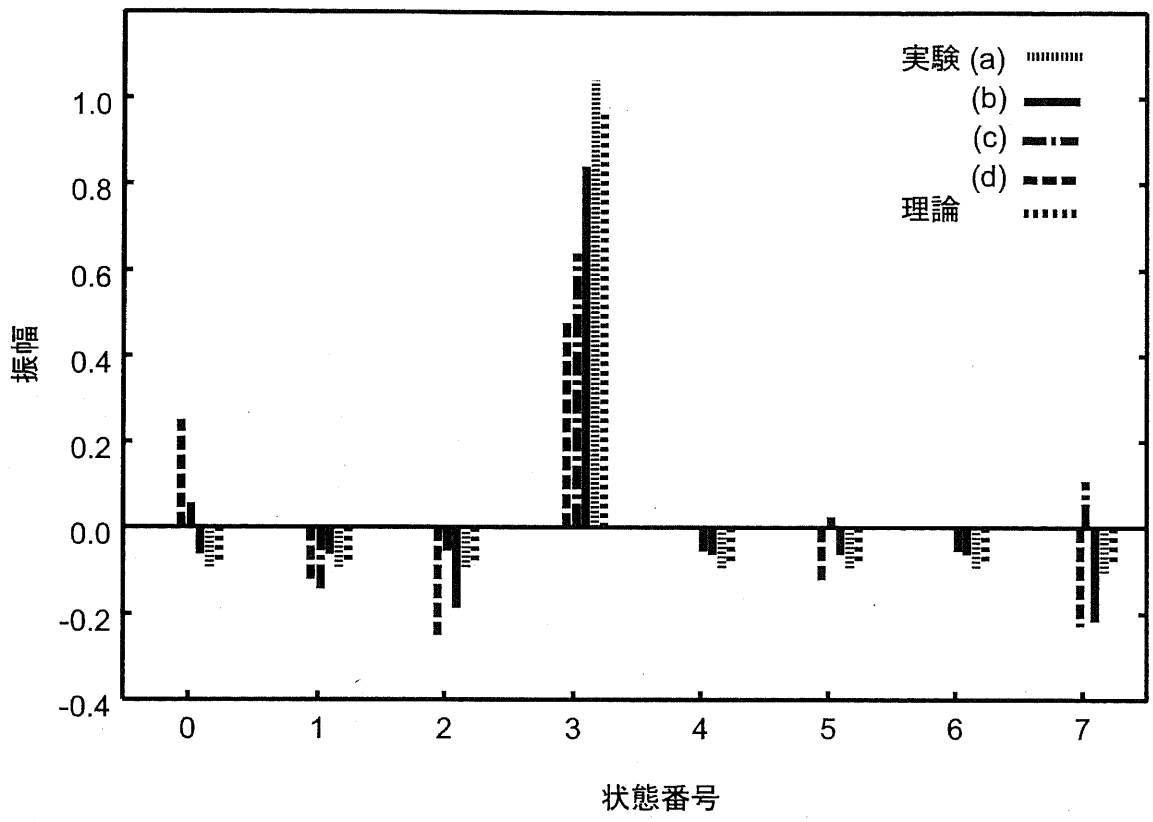


図 3.8: 出力信号の周波数スペクトル．状態ベクトル $|X\rangle$ に対応する

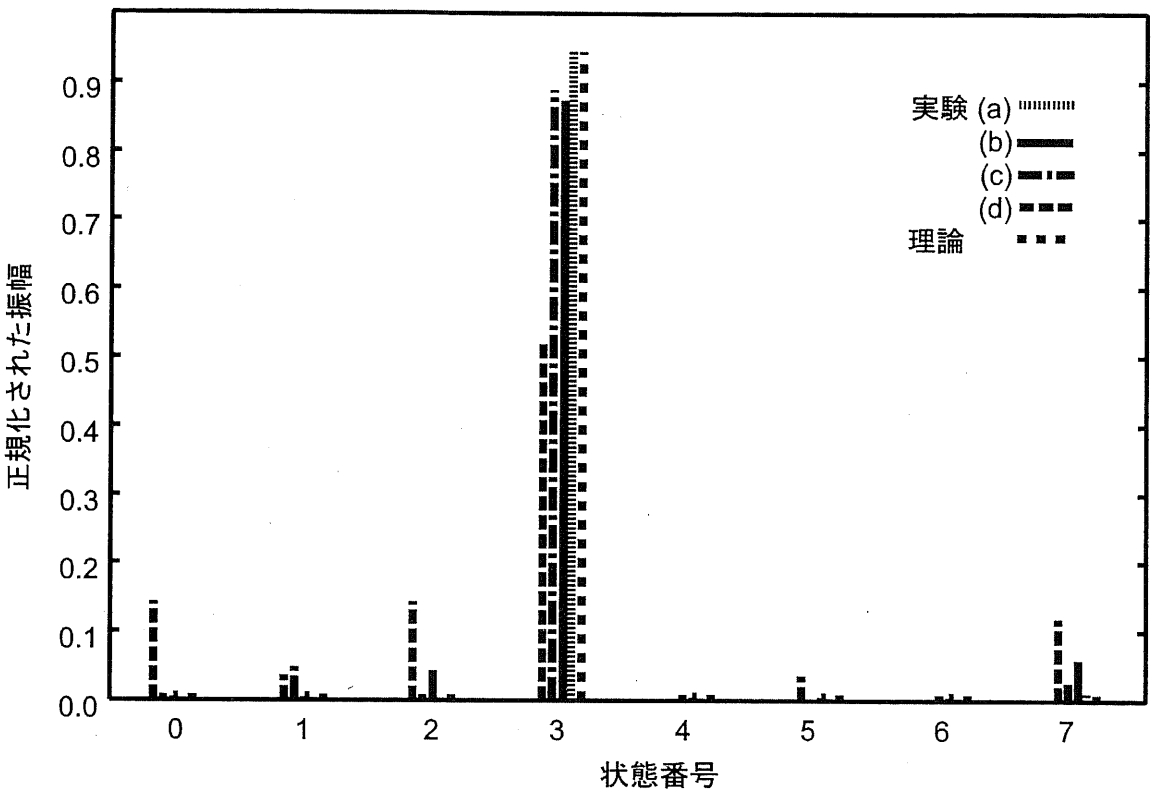


図 3.9: 正規化したパワースペクトル．観測確率に対応する．ここで，これらのスペクトルは誤差を含んでいる．

第 4章

再帰的ハードウェア構造を用いた量子回路プロセッサ

4.1 はじめに

前章では, 量子重ね合わせを古典重ね合わせに変換し, 実時間処理するエミュレータについて検討した. しかし, このエミュレータには, 量子ビットの数に対しデバイス数, 演算時間が共に指数爆発してしまうという欠点があった. これを解決し, より効率的なハードウェアシミュレーションを実現する必要がある

量子効果の働かない古典力学系の一つである集積回路で量子アルゴリズムを動作させようとする場合, 本質的に指数爆発を回避することは不可能であるが, 時間およびデバイス数のどちらか一方で指数爆発を容認すればもう一方を多項式に抑えることは理論的に可能である. 例えば, n 量子ビット規模の計算を 2^n 個のハードウェア要素に分担させることにより, すくなくとも計算時間爆発を緩和することは可能である. すなわちこれは, 計算量を時間領域から作業空間領域に転嫁することと言える. この手法は, 近年の集積回路の進展によりある程度まで可能なってきていると思われる. このときに重要となってくる事項は, 量子コンピュータのシミュレーション合わせたハードウェアの最適化を行ない, より大量のデバイスを効率的に動作させるハードウェアアーキテクチャを検討することと言える.

本章では, 量子回路プロセッサ (QCP, Quantum-Circuit Processor) と命名した, 高並列プロセッサについて述べる. QCP は, 量子重ね合わせを古典重ね合わせで模倣する第 3 章のハードウェアエミュレータとは異なり, 数値化された量子回路の動作を表現するユニタリ変換を高並列に処理するため, 計算時間を爆発させることはない.

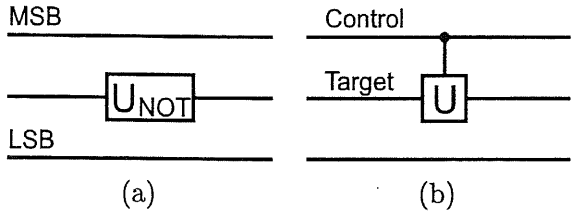


図 4.1: 量子ゲート操作の例.

4.2 量子回路プロセッサ

4.2.1 計算手法

量子回路プロセッサは, n 量子ビットの量子回路で行われる計算を, 2^n 個の演算エレメント (PE, Processing Element) を用い, 高並列にシミュレートする. すなわち, 量子系で重ね合わされていた計算は, 本システムにおけるハードウェアの並列動作に展開されることとなる.

計算手法について, 3 量子ビットの量子回路の場合を例にとり以下説明する. 量子回路においてある量子ビットに量子ゲート操作が行なわれるとき, 式 (2.29) に示したようにゲート操作のターゲットビットのみが 0,1 と異なる状態の対に 2×2 ユニタリ行列 U が作用する. 図 4.1(a) のような 3 量子ビットの量子回路のうちの第 2 量子ビット $|x_1\rangle$ にゲート操作が行なわれる例では, 状態の対 $|(x_20x_0)_2\rangle, |(x_21x_0)_2\rangle$ が 2×2 ユニタリ行列によって変換を受ける. すなわち,

$$\begin{bmatrix} \omega'_{000} \\ \omega'_{010} \end{bmatrix} = U_{\mathrm{NOT}} \begin{bmatrix} \omega_{000} \\ \omega_{010} \end{bmatrix}, \tag{4.1}$$

$$\begin{bmatrix} \omega'_{001} \\ \omega'_{011} \end{bmatrix} = U_{\mathrm{NOT}} \begin{bmatrix} \omega_{001} \\ \omega_{011} \end{bmatrix}, \tag{4.2}$$

$$\begin{bmatrix} \omega'_{100} \\ \omega'_{110} \end{bmatrix} = U_{\mathrm{NOT}} \begin{bmatrix} \omega_{100} \\ \omega_{110} \end{bmatrix}, \tag{4.3}$$

$$\begin{bmatrix} \omega'_{101} \\ \omega'_{111} \end{bmatrix} = U_{\mathrm{NOT}} \begin{bmatrix} \omega_{101} \\ \omega_{111} \end{bmatrix}, \tag{4.4}$$

のような 4 つの変換が, 1 つのゲート操作で同時に行なわれることとなる. 更に, このゲート操作にコントロールビットが加わると, コントロールビットが状態 $|1\rangle$ にある対のみで変換が行なわれ, 他は変換が行なわれない. すなわち図 4.1(b) の場合には

$$\begin{bmatrix} \omega'_{000} \\ \omega'_{010} \end{bmatrix} = E \begin{bmatrix} \omega_{000} \\ \omega_{010} \end{bmatrix}, \tag{4.5}$$

$$\begin{bmatrix} \omega'_{001} \\ \omega'_{011} \end{bmatrix} = E \begin{bmatrix} \omega_{001} \\ \omega_{011} \end{bmatrix}, \tag{4.6}$$

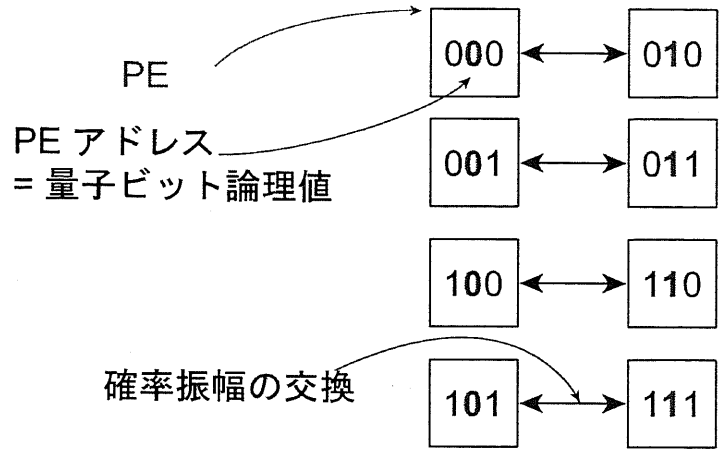


図 4.2: 量子計算プロセッサの計算手法の概念. 3 量子ビットの例.

$$\begin{bmatrix} \omega'_{100} \\ \omega'_{110} \end{bmatrix} = U \begin{bmatrix} \omega_{100} \\ \omega_{110} \end{bmatrix}, \tag{4.7}$$

$$\begin{bmatrix} \omega'_{101} \\ \omega'_{111} \end{bmatrix} = U \begin{bmatrix} \omega_{101} \\ \omega_{111} \end{bmatrix}, \tag{4.8}$$

のような変換が行なわれる. ただし, E は 2×2 単位行列を表す. このような量子並列演算をハードウェア上で 1 ステップで再現するために, 本プロセッサでは $2^3 = 8$ 個の PE の並列動作を利用する. システム中の 8 個の PE は, それぞれ 3 ビットのアドレス $000, 001, 010, \ldots, 111$ を持っており, これに対応する状態 $|(x_2x_1x_0)_2\rangle$ の確率振幅データ $\omega_{(x_2x_1x_0)_2}$ をローカルメモリに保持している. 各 PE は, 単一の量子ゲート操作を指定する命令に応じて確率振幅データを互いに授受し, 渡されたデータとローカルなデータを基にゲート操作後の振幅データを計算し, 結果をローカルメモリに書き戻す. 例えば式 (4.1)—(4.4), および (4.5)—(4.8) に示した第 2 量子ビットがターゲットとして指定された場合は, 図 4.2 のような PE 対 $\{000-010, 001-011, 100-110, 101-111\}$ すべてで処理が行なわれる. この並列動作により, 1 ステップで量子回路における演算が模倣され, PE の数に対応した規模の量子回路での演算が量子コンピュータと同等のステップ数で可能となる. この並列処理は, SIMD(Single-Instruction-stream-Multiple-Data-stream) 型の処理の 1 種と言える.

第 2 量子ビットにゲート操作が行なわれる図 4.1 の例と同様に, この 3 量子ビットの量子回路において第 1 量子ビットにゲート操作が行なわれる場合には $\{000-001, 010-011, 100-101, 110-111\}$, 第 3 量子ビットにゲート操作が行なわれる場合には $\{000-100, 001-101, 010-110, 011-111\}$ のような, 各 PE 対が形成されなければならない. 以上をまとめると, 3 量子ビットの量子回路の場合, 図 4.3 のようなハイパーキューブ型の通信網が必

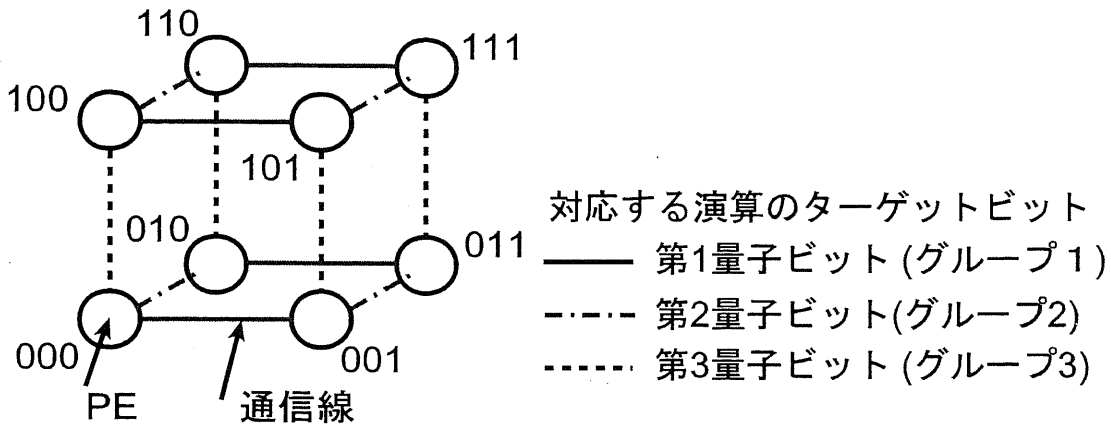


図 4.3: ハイパーキューブ型通信網の模式図.

要となることがわかる. この通信網を構成する PE 間リンクは, 量子ゲート操作ターゲットビットの指定に応じて, ターゲットビットに対応する PE アドレスビットのことなる対を有効にするもののみが演算中に有効となる.

一般に n 量子ビットの量子回路をシミュレートするハードウェアでは n 次元ハイパーキューブを形成する PE 間通信網が必要となる.

4.2.2 ハードウェアアーキテクチャ

実際に前述の n 次元ハイパーキューブ網を集積回路の 2 次元平面上で実現することは, 配線が複雑となり困難を伴う. そこで, 本システムではこれをスイッチマトリックスを用いたフラクタルツリー構造により実現した. 図 4.4 に 3 量子ビット部分の模式図を示す. 同図中に示す通り, ネットワーク全体は 2 分木構造を持っており, 木の分岐点にスイッチマトリックスが配置され, 木の葉側の端点に PE が配置される. 一般に, n 量子ビット規模の計算を行なう場合は $(n-1)$ 層のスイッチマトリックスが必要となる. 各スイッチマトリックスは, 葉側のデータを根側へ, 根側のデータを葉側への受け渡すか, もしくは葉側の 2 つの枝のデータを交換するという, 2 通りの振舞いをする. スイッチマトリックス全体の振舞いは, 命令により指定された量子ゲート操作ターゲットビットにより規定される. すなわち, 第 1 量子ビットをゲート操作のターゲットとして指定する場合, 第 1 層のスイッチが葉側の PE データの交換を行なうことにより, 目的とする PE 対の演算が実現される. 同様に第 2 量子ビットがターゲットビットの場合は第 2 層のスイッチが葉側の PE データ交換を行ない, 第 3 量子ビットがターゲットビットの場合は第 3 層のスイッチが葉側のデータ交換を行なう. ここで, ターゲットビットに対応しない層は, 葉側と根側の間のデータ受渡しを行なう. これによ

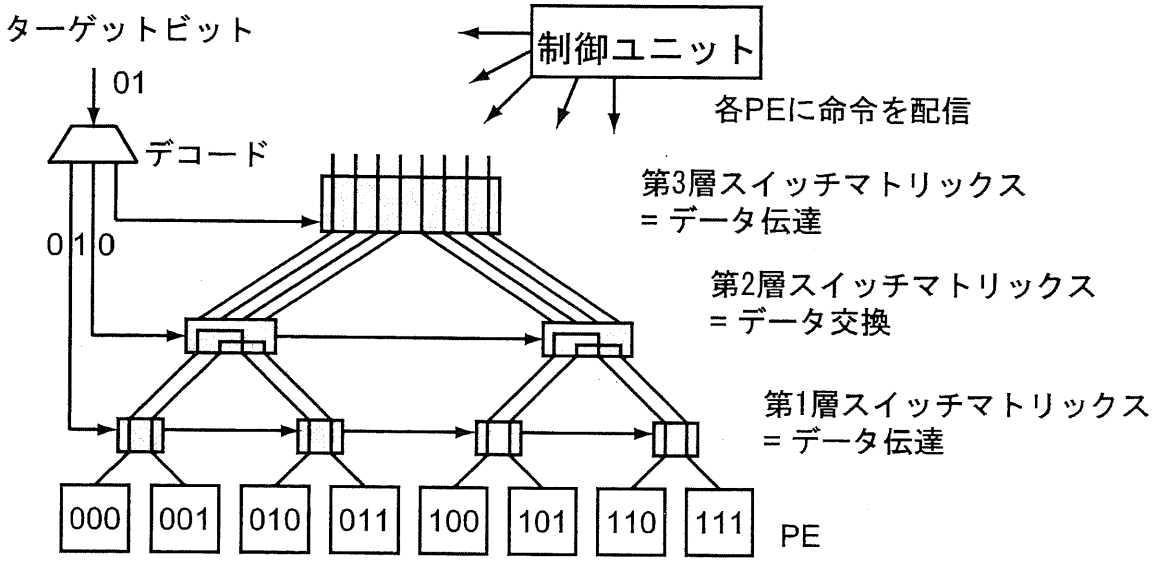


図 4.4: 2 次元平面上のフラクタルツリー構造で構成したネットワーク.

り, 前述のハイパーキューブ網が実現される.

このフラクタルツリー網を構成するスイッチマトリックスは, 2 次元平面上に配置する場合 H ツリー状の配置が可能である. 図 4.5 に 5 量子ビット規模のチップの例を示す. 再帰的な H 形のツリー構造の分岐点にスイッチマトリックスを配置し, 葉側の端点に PE を配置する. この平面上では, 各スイッチマトリックスに対し右半面に位置する PE のアドレスは 0, 左半面に位置する PE のアドレスは 1 という再帰構造を持つことになる. この再帰構造の利用により, 大規模なシステムを構築する場合の設計コストを低減することが可能となる. すなわち図 4.6 のように, $(n+1)$ 量子ビットのシステムは n 量子ビットのシステムと適当なバス幅のスイッチマトリックスを組み合わせることにより実現され, これを繰り返すことにより任意の大きさのシステムを効率良く生成することができる. このアーキテクチャは, 集積回路を用いた大規模並列システムに適しており, 論理合成や配置配線等の設計コストの削減にも繋がる.

PE の内部構成は, 図 4.5 に示す通り, 複素積和演算を主に行なう算術演算ユニットと, 振幅データの実部, 虚部, および基底の観測確率に対応するこれらの自乗和を保持するレジスタからなる. 制御ユニットは各部に対し, 命令および制御信号, 変換行列の係数をブロードキャストする. 制御ユニットからの制御信号により PE 間リンクが指定され, リンクによって PE に送られる対の振幅データと, ブロードキャストされた変換係数, PE 内部のローカルレジスタからのデータの 3 つから更新すべき振幅データを算術演算ユニットで計算し, ロー

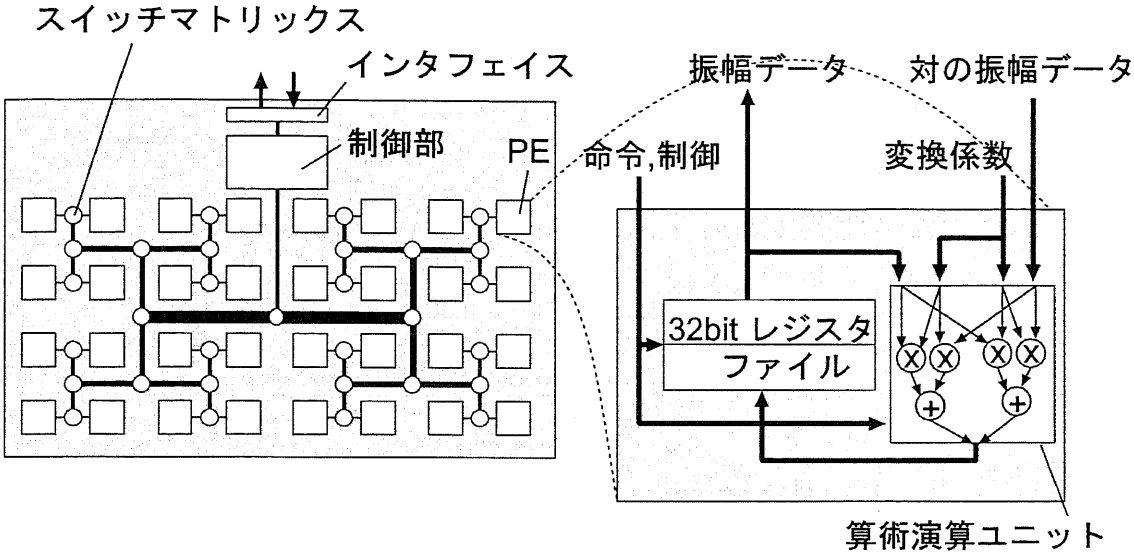


図 4.5: 量子回路プロセッサの構成. チップ上にフラクタルに PE を配置する.

カルレジスタに書き戻す. 以上の操作が演算サイクルとなる.

4.2.3 命令セットと PE 内部データパス

命令セットは, 第 2 章で述べた万能量子回路が構成可能な命令群により構成されなければならない. 万能量子回路を構成する要素量子ゲート群の決め方は何通りか考えられるが, 各命令に対応する PE 内部のデータパスによってハードウェアが効率的に利用されるような命令の機能分割が行なわれなければならない. 本プロセッサでは, ROT (ROTation), PHAS0 (PHASe shift on the state 0), PHAS1 (PHASe shift on the state 1), と名付ける 1 量子ビットに作用する量子ゲート操作命令, およびこれらに任意の数のコントロールビットを付加可能な CROT (Controlled ROTation), CPHAS0, CPHAS1 の計 6 つの量子ゲート操作命令を実装した.

ROT はターゲットビットに対し, ユニタリ行列

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{4.9}$$

で表されるようなゲート操作を行なう. 一方, PHAS0 および PHAS1 は, ターゲットビット

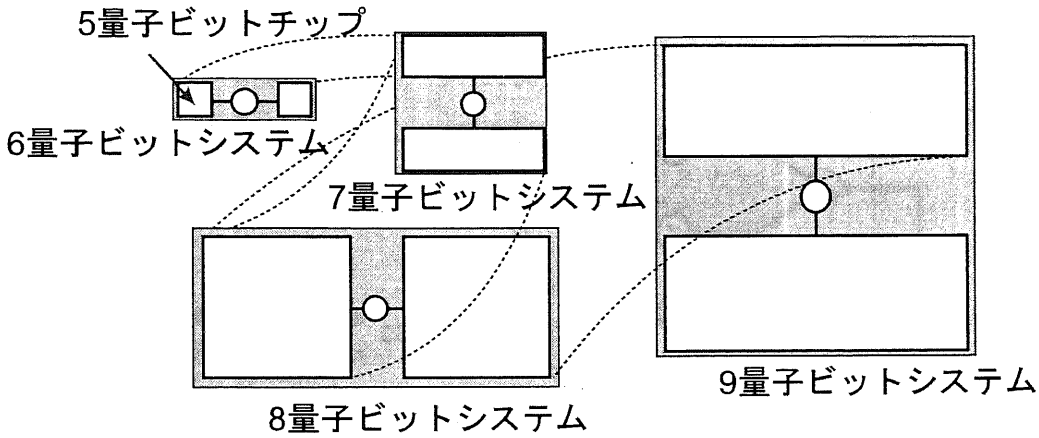


図 4.6: システムの量子ビット数拡張法.

に対し,

$$P_0(\phi_0) = \begin{bmatrix} e^{i\phi_0} & 0 \\ 0 & 1 \end{bmatrix}, \tag{4.10}$$

および

$$P_1(\phi_1) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi_1} \end{bmatrix} \tag{4.11}$$

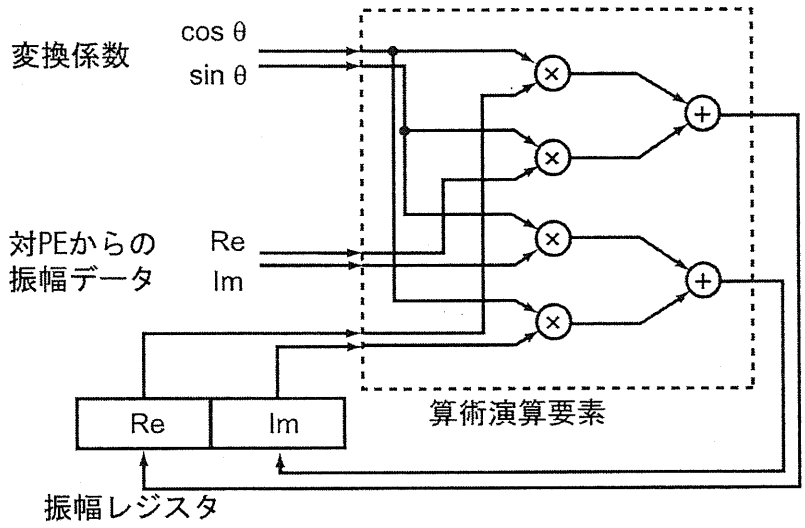
と表されるようなゲート操作を行なう. これら 3 命令によって, 1 量子ビットに対する任意のユニタリ変換が構成可能となる. また, CROT, CPHAS0, CPHAS1 は, 指定されたコントロールビットが全て 1 である状態に対し式 (4.9)—(4.11) で表される変換をそれぞれ行なう. この 6 種類のゲート操作命令によって第 2 章で示した万能量子ゲートが構成可能となる.

このように機能を 6 命令に分割することにより, 1 量子ビットに対し任意のユニタリ変換を実行可能な命令を実装する場合に比べ, 乗算器を 1PE あたり 8 個から 4 個に, 加算器を 4 個から 2 個に減らすことが可能となる. 例えば状態 0 側の PE では, ROT の場合

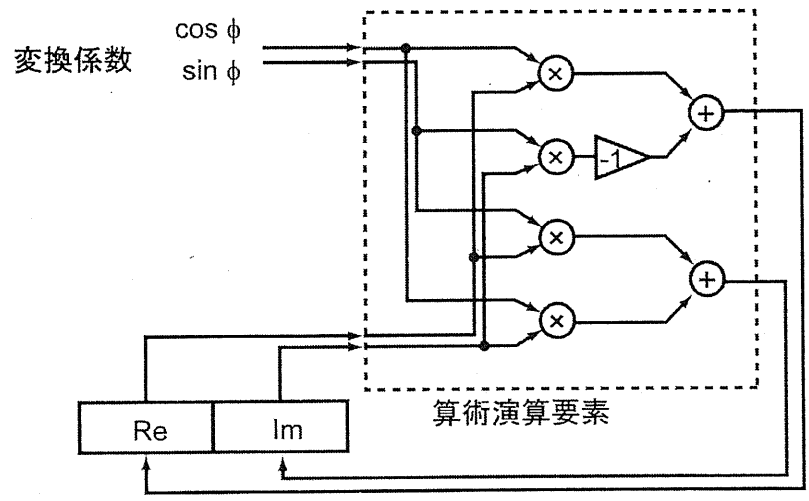
$$\left.\begin{aligned} \Re[\omega_0'] &= \Re[\omega_0]\cos\theta + \Re[\omega_1]\sin\theta \\ \Im[\omega_0'] &= \Im[\omega_0]\cos\theta + \Im[\omega_1]\sin\theta \end{aligned}\right\} \tag{4.12}$$

のような計算が実行され, PHAS0 の場合は

$$\left.\begin{aligned} \Re[\omega_0'] &= \Re[\omega_0]\cos\phi_0 - \Im[\omega_0]\sin\phi_0 \\ \Im[\omega_0'] &= \Re[\omega_0]\sin\phi_0 + \Im[\omega_0]\cos\phi_0 \end{aligned}\right\} \tag{4.13}$$



(a) ROT, CROTのデータパス



(b) PHAS, CPHASのデータパス

図 4.7: (a)ROT 系ゲート操作命令および (b)PHAS 系ゲート操作命令のデータパス.

のような計算が実行される．これら式(4.12), (4.13)に示す計算は，図4.7に示すデータパスで実現されることになるが，ROT系，PHAS系のどちらの命令でもPE内部で乗算，加算の数が一致しているため，ハードウェア資源を効率的に稼働させることが可能になる．ただし，PHAS0, PHAS1のいずれかが実行される時は半数のPEで，乗数1の乗算が行なわれ，ハードウェアが休止することと等価になる．このため，PHAS0とPHAS1を1命令にまとめる実装方法も考えられるが，この場合には係数$(\cos\phi_0, \sin\phi_0)$と$(\cos\phi_1, \sin\phi_1)$の2組をブロードキャストする必要が生じ，ROTに比べてブロードキャスト量が2倍になる．これに対し，PHASを状態0と1に分けたことによりステップ数は増えるが，その増加量は1量子ゲートに対し1ステップ程度であり，量子ビット数に対して指数関数的にアルゴリズム全体のステップ数を増やすものではない．PHASを2つに分けることは，配線領域のオーバーヘッドを抑える効果がある．

代表的な量子ゲート操作であるウォルッシュ·アダマール変換およびCNOT変換を専用命令で記述すると次のようになる．

ウォルッシュ·アダマール変換:

PHAS0(π)
ROT($\pi/6$)

CNOT:

ROT($\pi/2$)
PHAS1(π)

観測のシミュレートについても同様に，ステップ数のオーダーはPoly(n)とならければならない．以下，3量子ビットの量子回路を例として観測のシミュレート方法について述べる．量子ビット3つのうち第1量子ビットに観測が行なわれる場合，この第1量子ビットで状態0が観測される確率Prob(XX0)，もしくは1が観測される確率Prob(XX1)が必要である．これらは，

$$\mathrm{Prob(XX0)} = |\omega_{000}|^2 + |\omega_{010}|^2 + |\omega_{100}|^2 + |\omega_{110}|^2 \tag{4.14}$$

および

$$\mathrm{Prob(XX1)} = |\omega_{001}|^2 + |\omega_{011}|^2 + |\omega_{101}|^2 + |\omega_{111}|^2 \tag{4.15}$$

のように，PE内部の振幅レジスタ値の実部，虚部の自乗和を，PEアドレスの第1ビットが0のもの同士，1のもの同士で足し合わせたものである．これらを計算し，得られた値に基づき状態の組$\{000, 010, 100, 110\}$(以下グループ0と称する)と$\{001, 011, 101, 111\}$(同様にグループ1と称する)のうちのどちらかの振幅データを消去することにより，量子系における観測と等価な効果を得ることができる．この一連の操作は，本プロセッサにおいてPCAL

(Probability CALculation), PSUM (Probability SUMmation) および REDUCE と名付けた3命令によって実される. 観測操作の第1ステップで用いられる PCAL 命令は, それぞれの PE 内部で振幅の実部, 虚部の自乗和

$$\mathrm{Prob}(i) = |\omega_i|^2 = \Re[\omega_i]^2 + \Im[\omega_i]^2 \tag{4.16}$$

を計算し, これを単基底の観測確率として PE 内部の確率レジスタに保存する. この演算も, (4.16) より PHAS, ROT と同様に乗算4つと加算2つで実現可能である. 続いて PSUM 命令によって確率レジスタに保存されたデータを積算する. 単一の PSUM 命令は, ターゲットビットとして指定された PE アドレスのビットのみが 0, 1 と異なる PE の確率データを加算し, 双方の PE の確率レジスタに結果を書き戻す. この PE 対は ROT, PHAS のゲート操作命令と同様に, ハイパーキューブ網により生成される. この例のように第1量子ビットにおける観測をシミュレートしようとする場合, この PSUM を第3量子ビット, 第2量子ビットに続けて行なう. まず, 第3量子ビットに対して PSUM を行なうと, PSUM の実行前の確率レジスタ値 Prob_k は次式のような Prob_{k+1} に更新される.

$$\begin{aligned}
\mathrm{Prob}_{k+1}(000) &= \mathrm{Prob}_{k+1}(100) = \mathrm{Prob}_k(000) + \mathrm{Prob}_k(100),\\
\mathrm{Prob}_{k+1}(001) &= \mathrm{Prob}_{k+1}(101) = \mathrm{Prob}_k(001) + \mathrm{Prob}_k(101),\\
\mathrm{Prob}_{k+1}(010) &= \mathrm{Prob}_{k+1}(110) = \mathrm{Prob}_k(010) + \mathrm{Prob}_k(110),\\
\mathrm{Prob}_{k+1}(011) &= \mathrm{Prob}_{k+1}(111) = \mathrm{Prob}_k(011) + \mathrm{Prob}_k(111).
\end{aligned}$$

続いて第2量子ビットに対して PSUM を行なうと,

$$\begin{aligned}
\mathrm{Prob}_{k+2}(000) &= \mathrm{Prob}_{k+2}(010) = \mathrm{Prob}_{k+1}(000) + \mathrm{Prob}_{k+1}(010),\\
\mathrm{Prob}_{k+2}(001) &= \mathrm{Prob}_{k+2}(011) = \mathrm{Prob}_{k+1}(001) + \mathrm{Prob}_{k+1}(011),\\
\mathrm{Prob}_{k+2}(100) &= \mathrm{Prob}_{k+2}(110) = \mathrm{Prob}_{k+1}(100) + \mathrm{Prob}_{k+1}(110),\\
\mathrm{Prob}_{k+2}(101) &= \mathrm{Prob}_{k+2}(111) = \mathrm{Prob}_{k+1}(101) + \mathrm{Prob}_{k+1}(111)
\end{aligned}$$

のように確率レジスタ値更新が起こる. 結果として

$$\mathrm{Prob}_{k+2}(000) = \mathrm{Prob}_{k+2}(100) = \mathrm{Prob}_{k+2}(010) = \mathrm{Prob}_{k+2}(110) = \mathrm{Prob}(\mathrm{XX0}),$$

および

$$\mathrm{Prob}_{k+2}(001) = \mathrm{Prob}_{k+2}(101) = \mathrm{Prob}_{k+2}(011) = \mathrm{Prob}_{k+2}(111) = \mathrm{Prob}(\mathrm{XX1})$$

のように, グループ0に属する PE の各確率レジスタには Prob(XX0) が, グループ1に属する PE の確率レジスタには Prob(XX1) がそれぞれ書き込まれた状態となる. この結果を最

後に REDUCE 命令により比較する．RUDCUCE 命令により観測対象である第一量子ビットがターゲットビットと指定されると，$\{000-001, 010-011, 100-101, 110-111\}$ の PE 対が形成され，全対の中で確率値の比較演算が同時に行われる．比較の結果に応じてどちらかの PE 上の振幅データは消去されるが，この方法として 4 つが選択可能である．

(1) 絶対 0 モード：確率の大小にかかわらず状態 0 に確率が存在すればグループ 0 の振幅を残し他方を消去する．確率が存在しない場合は ZERO フラグを立てホストに演算が遂行されなかったことを通知し，計算を一時停止する．

(2) 絶対 1 モード：絶対 0 モードと同様の操作をグループ 1 に関して行う．

(3) 比較モード：確率値 Prob(0) と Prob(1) を比較し，確率の大きいグループを残し，他方を消去する．確率が等しい場合は EVEN フラグを立てホストに演算が遂行されなかったことを通知し，計算を一時停止する．

(4) 確率モード：Prob(0) と Prob(1) に従い，確率的に一方のグループを残し，他方を消去する．

以上をまとめると，3 量子ビットの量子回路において第 1 量子ビットの観測を行う場合，

```
PCAL
PSUM(2)
PSUM(1)
REDUCE(0, mode)
```

の 4 ステップで観測をシミュレート可能である．一般に，n 量子ビットの量子回路をシミュレートする場合は $(n+1)$ ステップで観測のシミュレートを遂行可能である．

なお，REDUCE 命令の確率モード以外のモードは本来の量子計算では許されない操作であるが，例えばこれらのモードを用いて重ね合わせ状態から最小値を求める多項式ステップアルゴリズムなどを実現可能である．すなわち，REDUCE を絶対 0 モードで行い，演算が遂行されなかった場合にのみ絶対 1 モードで REDUCE を実行しなおすと言う操作を全ての量子ビットに行うことにより，最大で $n(2n+2)$ ステップで最小値を恣意的に求めることができる．

以上の命令に加え停止命令と状態 $000\ldots0$ への量子回路初期化命令を含め，命令セット全体は 11 命令より構成される．各命令とその機能を表 4.1 にまとめる．

4.2.4　システムの制御

システムの制御は，図 4.5 に示す通り，チップあるいはシステムにつき 1 個の制御ユニットによって集中的に行われる．制御ユニットの状態遷移図を図 4.8 に示す．ここで，外部から入

表 4.1: 命令セットおよび命令機能の概要

ニモニック	機能
PHAS0(q_t, ϕ_d)	ターゲットビット q_t の状態0に対する位相シフト.
PHAS1(q_t, ϕ_d)	ターゲットビット q_t の状態1に対する位相シフト.
CPHAS0(q_c, q_t, ϕ_d)	q_c に示された量子ビットをコントロールビットとする量子ビット q_t の状態0の位相シフト. n 量子ビットの量子回路場合 q_c フィールドは n ビット幅で，1が立っている量子ビットが制御を行うことを示す.
CPHAS1(q_c, q_t, ϕ_d)	q_c に示された量子ビットをコントロールビットとする量子ビット q_t の状態1の位相シフト.
ROT(q_t, θ)	標的ビット q_t に対する回転.
CROT(q_c, q_t, θ)	q_c に示された量子ビットをコントロールビットとする回転.
PROB	基底の観測確率を確率振幅より求める.
PSUM(q_t)	PEアドレスの q_t ビット目のみが異なるPEの確率レジスタ値を加算し，計算結果をそれぞれのレジスタに書き戻す.
REDUCE(q_t, mode)	PEアドレスの q_t ビット目のみが異なるPEの確率レジスタ値を比較し，modeで指定された方法で状態を減らす. 絶対0モード (00): 比較結果にかかわらず，状態0に対応したレジスタ値を残す. 絶対1モード (11): 比較結果にかかわらず，状態1に対応したレジスタ値を残す. 比較モード (01): 比較結果にしたがって，確率の大きい側を残す. 確率モード (10): 確率に従いどちらを残すか決定する.
INIT	状態 $00\ldots0$ への初期化.
HALT	停止命令.

力される制御信号は，計算機のリセットを行う $\overline{\text{RST}}$, 計算実行停止を制御する RUN の 2 つであり, 逆にプロセッサ外部に返される制御信号は計算機が停止していることを示す停止フラグ HALT, および REDUCE 命令で計算が停止したことを示す EVEN フラグと ZERO フラグの計 3 つである. ただし実現形式によっては, ここでホストと呼んでいる機能も制御部に含めることが可能である.

制御フローは次のようになる.

(1) 電源投入後等にリセット信号 $\overline{\text{RST}}$ の入力により，強制的にアイドル状態 IDLE に入る.

(2) RUN 信号が入力されると命令フェッチステージ IFETCH に入り，外部から入力された命令を命令レジスタにフェッチする.

(3) 続くデコードステージ DEC ではフェッチされた命令を解釈する. その結果停止命令 HALT が検出された場合は，停止フラグ HALT を立て停止状態 STALL に入る. そうでなければ，解釈した結果を命令コードレジスタに入力し，PE およびスイッチマトリックスに制御信号等をブロードキャストし, 命令実行ステージ EXE に移行する.

(4) EXE ステージにおいて計算が正常に遂行された場合は, IF ステージに移行し, 次ステップの命令を実行する. 一方 REDUCE 命令で停止フラグが立てられた場合, STALL 状態に移行する.

(5) STALL 状態に入った場合, ホスト側で停止原因を示すフラグ EVEN または ZERO をを検出し RUN 信号を立ち下げ, IDLE 状態に入る. ここでホストが必要な操作を行った後再び RUN 信号を立ち上げることにより, 命令の実行を再スタートする.

4.3 本アーキテクチャの評価

本システムの性能, 特に本システムが現実に 1 チップでどの規模の計算に対応可能かを知るために, テクノロジ・ロードマップ [30] をもとに簡単に見積もった.

図 4.7 のデータパスとネットワークをあわせた実現方式として次の 4 通りを仮定する.

(1) PE は, 実部, 虚部それぞれについて, 乗算を 1 ステップで行う L ビット並列積和演算器と L ビットの振幅専用レジスタを備え, この他に L ビット確率専用レジスタ 1 つを持つ構成とし, ネットワークは実部, 虚部それぞれ L ビットを並列伝送可能な構成とする. EXE ステージに 1 クロックサイクルを要し, 1 演算サイクルは 3 クロックサイクルとなる.

(2) 乗算器を $2L$ ビット算術シフトレジスタと L ビット加算器で構成したもの. ネットワークは (1) と同様の並列伝送とする. 乗算を L ステップで行なうため, EXE ステージに $(L+1)$ クロックサイクル要し, 演算サイクルは $(L+3)$ クロックサイクルとなる.

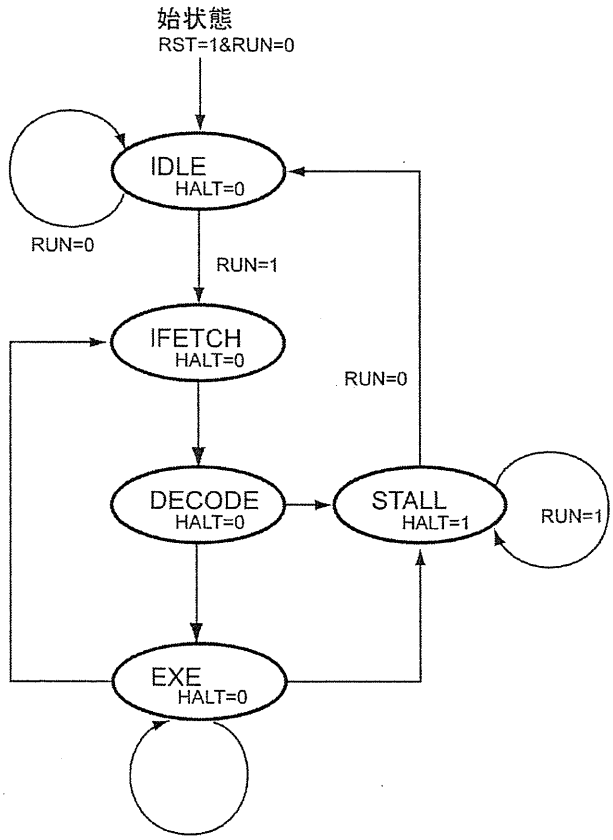


図 4.8: プロセッサの状態遷移図. EXE 状態には内部状態が存在する.

(3) 方式 (2) と同じ PE を用い, L ステップかけて振幅データ伝送する直列伝送を組み合わせた構成. 演算サイクルは $(2L+3)$ クロックサイクルとなる.

(4) PE 内部の計算を方式 (3) の半分の加算器 2 個で実現する方法. クロックサイクルが 2 倍必要となるため, 演算サイクルは $(3L+4)$ クロックサイクル必要となる.

この 4 モデルそれぞれについて, 1 チップに収まる量子ビット数と計算速度の関係を求めた. この結果を図 4.9 に示す. ここで, 振幅は n 量子ビットの量子回路において, 少なくとも全ての論理基底の重ねあわせ状態

$$|\phi\rangle = \frac{1}{\sqrt{2^n}} \sum^{2^n} |i\rangle$$

を表現できなければならない. よって, 振幅データのビット長 L は, 固定少数点で最低でも $L > n/2$ を満たさなければならない. ただし, L は $O(n)$ となる. ここでは, 量子ビット数 16

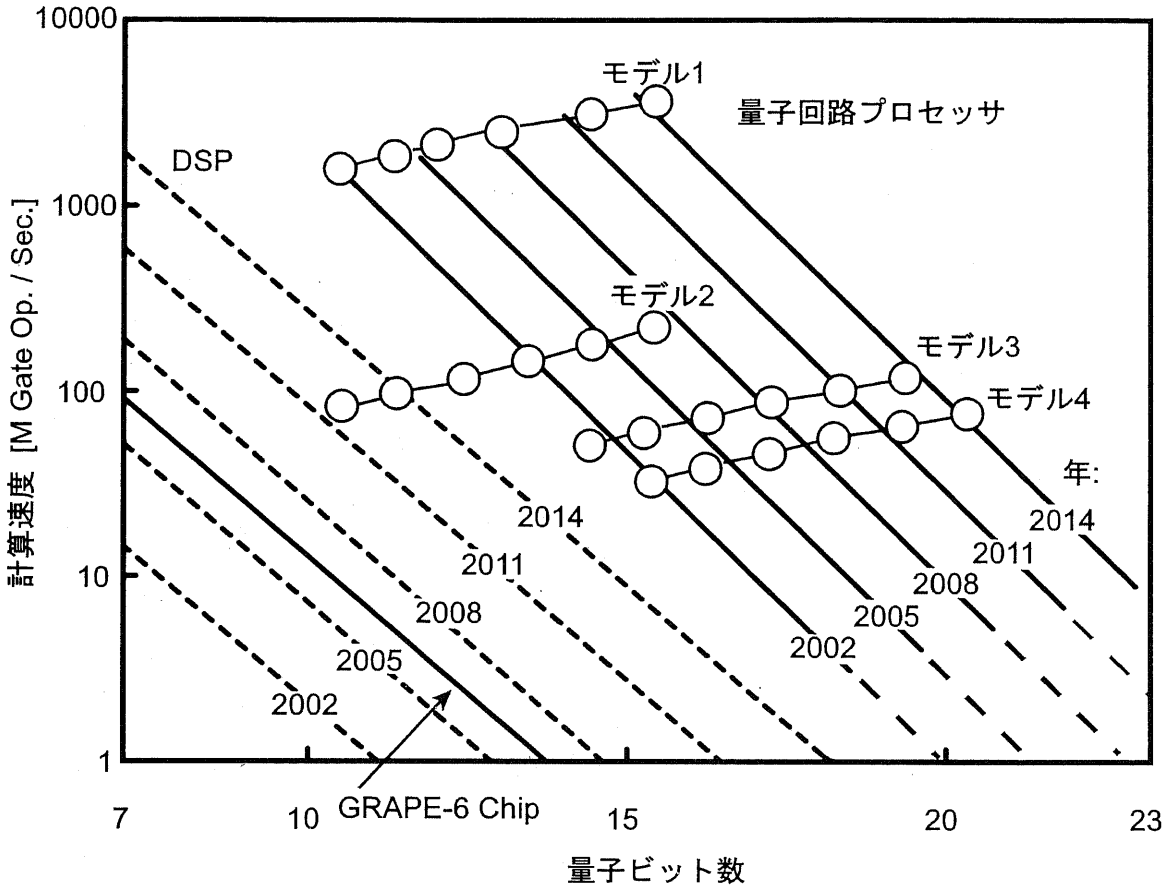


図 4.9: 量子回路プロセッサの性能見積もり．2014 年には 1 チップで 20 量子ビットが可能となる．

量子ビットまでを $L = 16, 32$ 量子ビットまでを $L = 32$ と仮定し，PE を構成するのに必要なトランジスタ数が占有する面積を PE1 つあたりのチップ面積をとして，文献 [30] のテクノロジロードマップを基に算出した．一方ネットワークの実装に要する面積は，グローバル配線で図 4.5 のようなネットワークを実装した場合の占有面積として，グローバル配線ルールを同じく文献 [30] に発表されている最小線幅 f の 4 倍程度と仮定し算出した．動作速度に関しては，クロック速度がネットワークに支配されると仮定し，グローバルクロック速度を PE のクロック速度として計算した．比較として，同じ計算を TI 社製 DSP TMS320C64x と同等のアーキテクチャで行った場合の速度を併記する．ただし，クロック速度は同一としテクノロジの進展による性能の向上を 3 年で 4 倍程度と仮定した．また，ニュートン力学系の多対問題に特化した並列計算システム GRAPE [31, 32] に搭載されているプロセッサチップの計算能力を併記する．ただし，これは 0.25μm プロセスを用いたものであり，1 チップあたりの性能が 30GFLOPS であることを参考に記入した．

図より，我々のシステムでは計算の並列化により，モデル 1 では 10 量子ビット程度で DSP に対し約 1000 倍の高速化が可能であることが分かる．モデル 1 から 2 への変更は，乗算器に要するデバイス数が減少することから量子ビット数が増加するが，その幅は 1 量子ビットから 2 量子ビットまでで計算速度の低下に対して得られる効果は少ない．これに対して，モデル 2 から 3 へと，ネットワークを並列伝送から直列伝送にした場合には，計算速度のペナルティに対して十分な量子ビット数の増加が見込める．すなわち，通信部分の効率化が性能に大変大きな影響を与えることを示唆している．また図より，本システムは 1 チップあたり 20 量子ビット程度で動作するように実現方法を調整するのが最も妥当であると考えれられる．将来的には 25 量子ビット以上を 1 チップで実現できるようになると予想される．

4.4 PLD を用いた実装

アルテラ社製のプログラマブルロジックデバイス (PLD, Programmable Logic Device) EP20K1500EFC33-3 を用いて，量子回路プロセッサの実装を行ない，実現可能性について検証を行なった．

4.4.1 5qubit 量子回路プロセッサ

前節で性能評価用モデルとして仮定した実装方式 (2) と同じ形式で量子回路プロセッサを作製した．すなわち，図 4.7 に示すデータパスの積和演算器を，算術シフトレジスタおよび加算器それぞれ 2 つによって実現した．確率振幅の計算精度は 16 ビット固定小数点とた．この積和演算器の回路構成を図 4.10 に示す．積和演算器は，まず図中実線で示すパスで 16 ビットの乗算を行ない，続いてこの積項を図中破線で示すパスで 32 ビット加算し，最後に零捨一入して 16 ビットに丸めを行なうことにより，積和項を出力する．図 4.7 の算術演算ユニット全体はこの積和演算器 2 つで構成されることとなる．

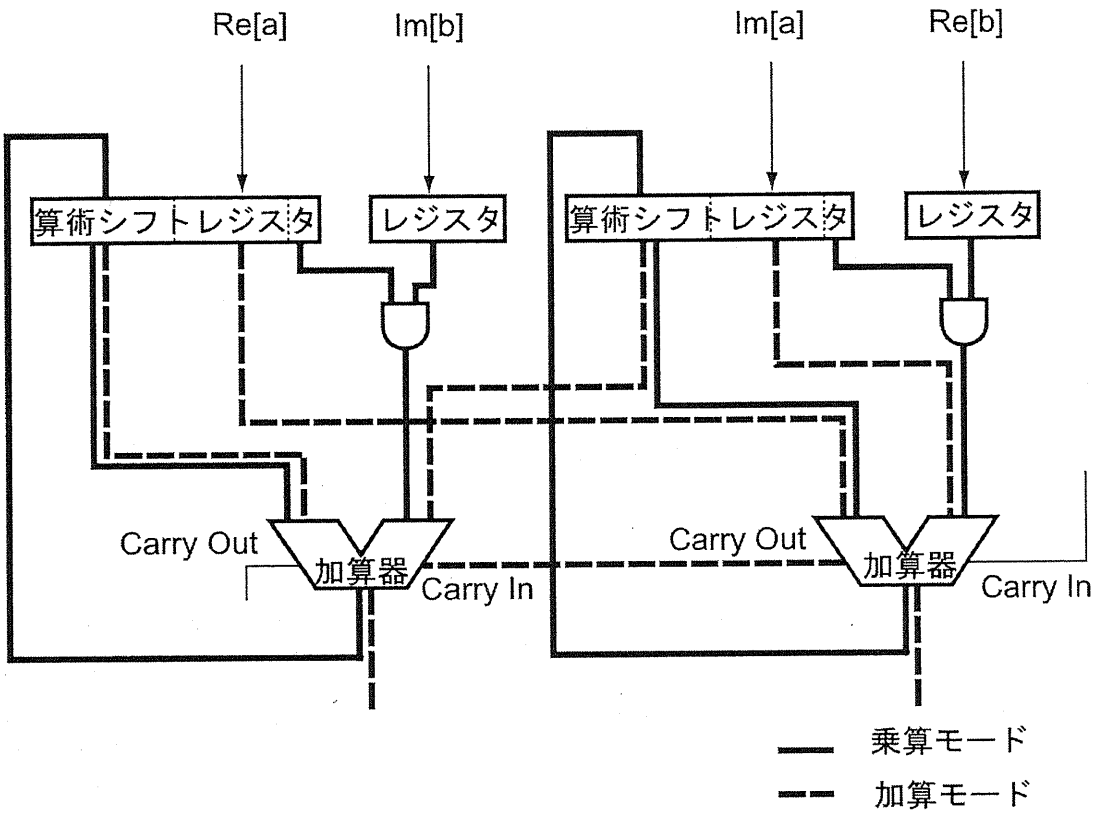


図 4.10: 複素乗算器の内部回路構成．同回路 2 つ分が全体の複素乗算機になる．

PE内部の演算を制御するために，図4.8のEXEステージに図4.11に示すような内部状態を定義した．量子ゲート操作命令実行時のPE内部の動作は以下の通りである．

(1) 確率振幅の実部もしくは虚部の計算に必要な確率振幅と変換係数のそれぞれ実部もしくは虚部のどちらかを適当なレジスタに代入する (MULI).

(2) 図4.10実線のパスにより算術シフトと加算を繰り返し，乗算を行なう (MUL, MULF).

(3) (2)を繰り返すことにより算術シフトレジスタに出力される乗算の結果を図4.10破線のパスにより32ビットの加減算を行ない，結果をシフトレジスタに書き戻す (ADD, ADDP).

(4) 32ビットから16ビットへ零捨一入を行ない，結果を振幅レジスタに書き戻す (ROFF).

また，PSUMおよびREDUCEの場合は16ビット加算器2つを用いた加減算により，加算と比較を行なう．ここで，量子回路の状態をモニタできるように，プロセッサからは各々の量子ビットの観測結果を示すコードが外部に常に返される．この観測結果を表すコードは2ビット長で，

- 00: 該当する量子ビットで状態0が観測された状態，
- 01: 状態0および1の等確率の重ねあわせにある状態，
- 11: 状態1が観測された状態，
- 10: 未観測状態

の4状態をモニタ可能である．このコードを専用のレジスタに格納する操作は，REDUCE, INIT両命令のSTOREステージで行なわれる．また，本プロセッサでは観測の度に行う規格化を省略しているが，確率振幅の実部，虚部それぞれに量子ビット数以上の十分なビット数を計算に用いているため，問題とはならない．

以上の実装の結果，最終的に表4.2のような動作性能が得られた．同チップの動作クロック周波数はネットワーク部分に支配されている．また，プロセッサが命令を実行するのに必要なクロックサイクル数は表4.3のようになった．

4.4.2 8qubit量子回路プロセッサ

前節の1チッププロセッサよりも更に大規模な計算に対応するために，4.3で行なった性能評価のモデル (4) に準ずる方式の実装を行ない，さらにPLDの構造を踏まえた回路の最適化を行った．これにより，8量子ビット規模のプロセッサを実現した．

この実装形式では図4.12に示すユニット2つで1基底分の計算が可能となる．PE間のリンクをシリアルデータ転送とし，PEを振幅専用レジスタとブース乗算器，シリアル加算器に

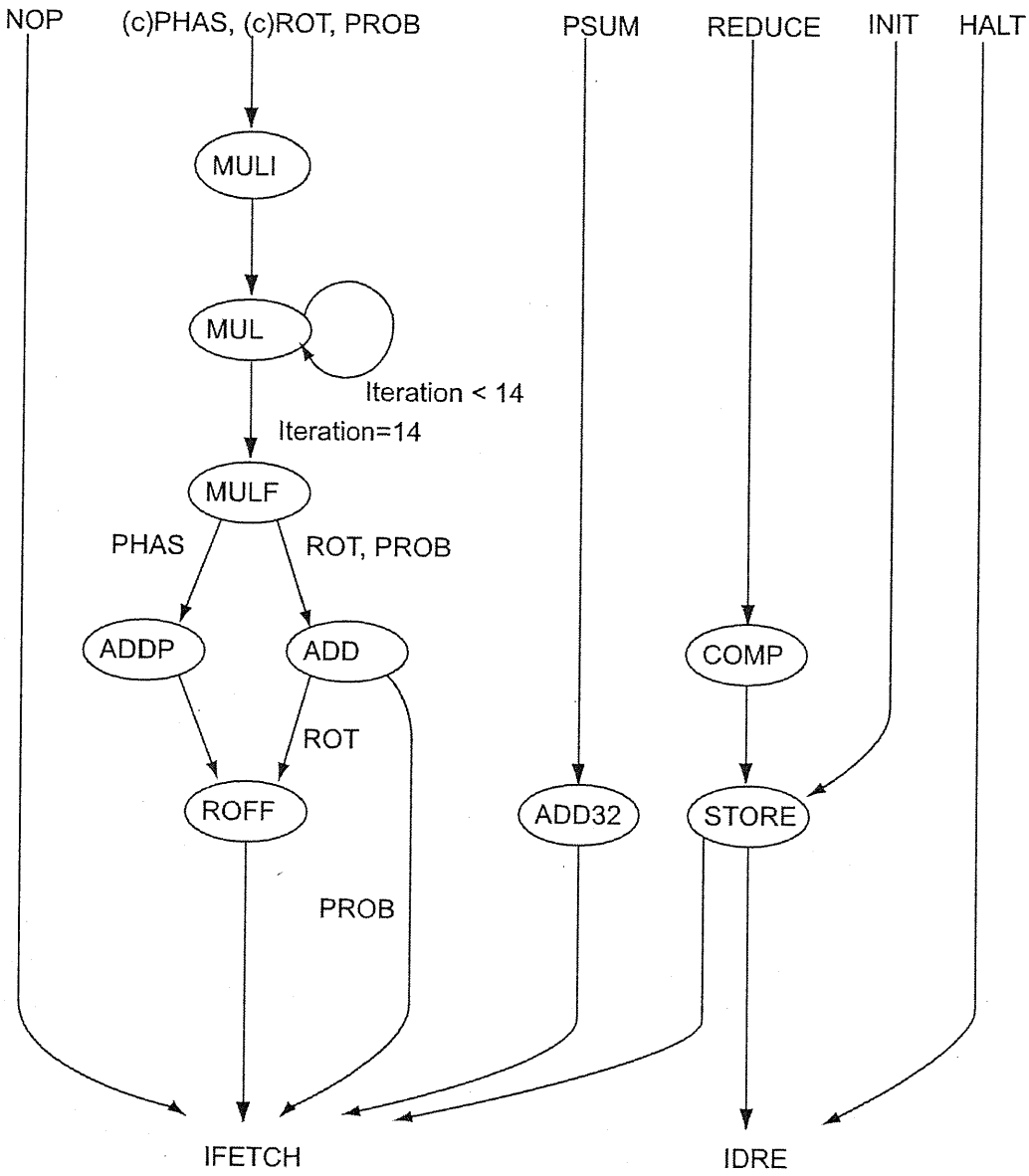


図 4.11: EXE ステージの内部状態.

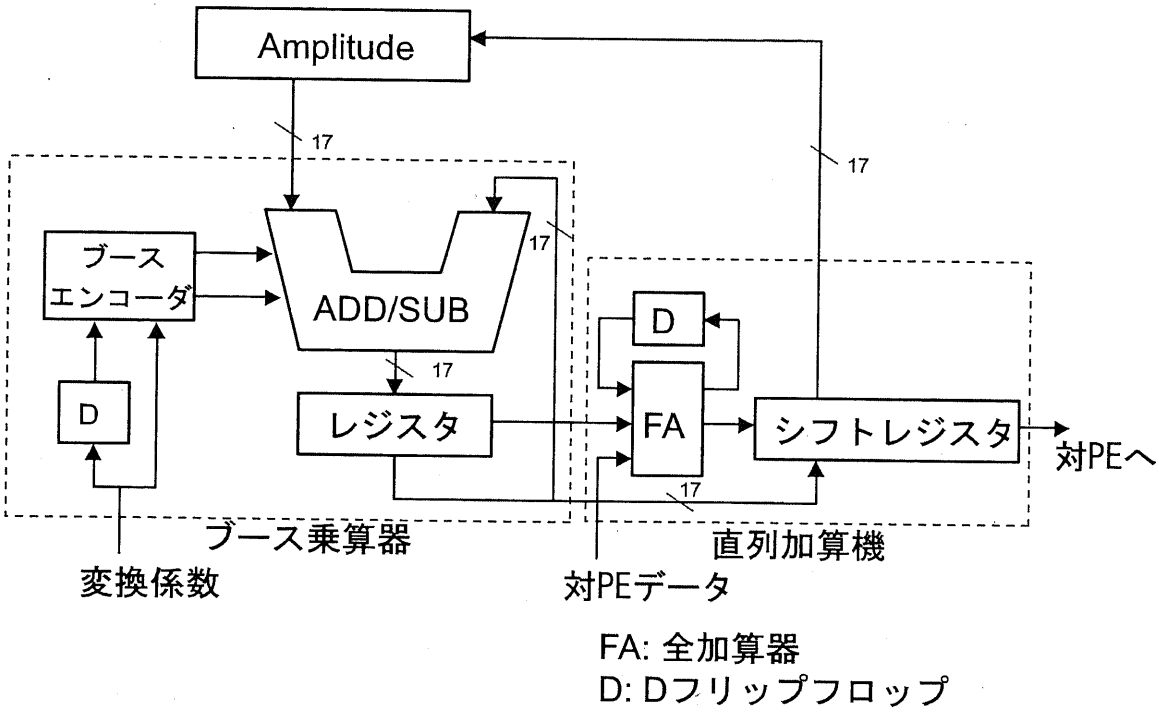


図 4.12: 8 量子ビットプロセッサ PE 内部の回路構成. 同図のユニットを実部, 虚部に 1 個づつ, つまり 1 基底に 2 つが必要となる. こののユニットは, Amplitude と表示する振幅専用レジスタ, ブース乗算器, シリアル加算器から構成される.

表 4.2: 5 量子ビットチップの主要仕様

PLD タイプ	Altera EP20K1500EFC33-3
PLD 使用数	1 枚
量子ビット数	5
ゲート数	1.1×10^6
クロック速度	15MHz

表 4.3: 5qubit 量子回路プロセッサの命令実行に必要なクロックサイクル数

命令	クロックサイクル数
PHAS0	20
PHAS1	20
CPHAS0	20
CPHAS1	20
ROT	20
CROT	20
PROB	19
PSUM	2
REDUCE	3
INIT	2
HALT	1

より構成した. 必要トランジスタ数を減らすことが可能となった. このユニットを使って, 量子ゲート操作を行なう動作は以下のようになる.

(1) 振幅レジスタ値と Coeff 端子から与えられる変換行列係数 (cos, sin) の片方との積をブース乗算器により求め,

(2) 乗算結果をシリアル加算器のシフトレジスタに転送 (MUL1).

(3) 続いて振幅レジスタ値と変換行列係数 (cos, sin) のうちの残りとの積を同様にブース乗算器により求める (MUL2).

(4) シリアル加算器内部のシフトレジスタから (1) で求めた値を ext 出力端子からリンクに送出すると同時に, 対の PE からリンクにより運ばれて ext 入力端子から入力される

表 4.4: 8 量子ビット規模の量子回路プロセッサの命令実行に必要なクロックサイクル数

命令	クロックサイクル数
PHAS0	57
PHAS1	57
CPHAS0	57
CPHAS1	57
ROT	57
CROT	57
PROB	58
PSUM	19
REDUCE	22
INIT	2
HALT	1

表 4.5: 8 量子ビットチップの主要仕様

PLD タイプ	Altera EP20K1500EFC33-3
PLD 使用数	1 枚
量子ビット数	8
ゲート数	1.3×10^6
クロック速度	30MHz

値を 1 ビットシリアル全加算器により加算し, 結果をシフトレジスタに順次格納する (ADD).

(5) シリアル加算器のシフトレジスタ値を振幅レジスタに転送する (STORE).

この他, レジスタのリセット等に数クロック要する. 命令実行に必要なクロックサイクル数を表 4.4 にまとめる.

この実装の結果, 表 4.5 に示すよう動作性能が得られた. この場合も同様に, ネットワーク部分の動作速度により全体の動作性能が決定される.

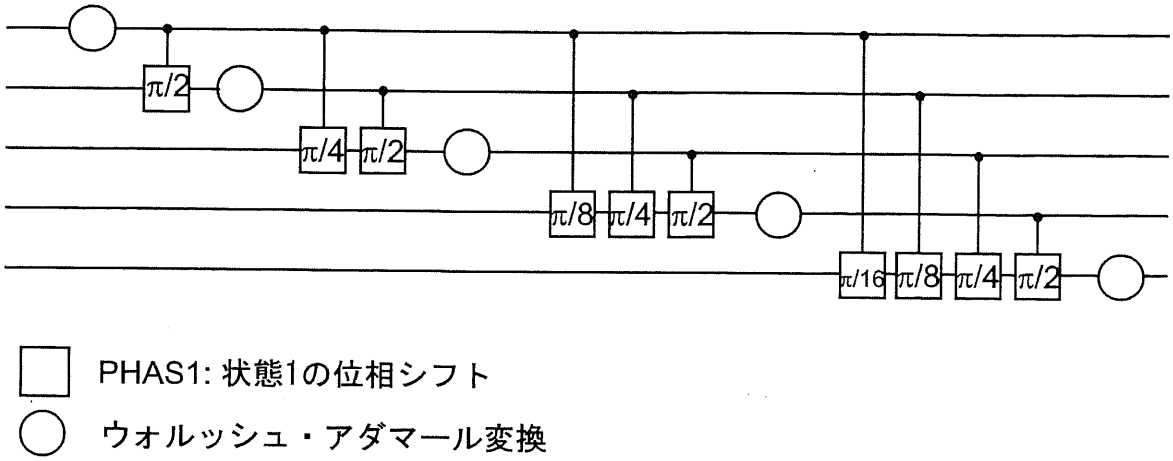


図 4.13: 5 量子ビット相当の量子フーリエ変換を示す量子回路.

4.4.3 量子フーリエ変換の実験

量子回路プロセッサ上で量子フーリエ変換の実験を行った. 5 量子ビット相当の量子フーリエ変換の回路を図 4.13 に示す. 量子フーリエ変換は, 高速フーリエ変換 (FFT, Fast Fourier Transform) と全く等価な振幅計算を行なう. 図 4.13 に示す部分は量子フーリエ変換本体であり, 量子回路プロセッサ上では 25 命令で実現されるこの他に FFT のシャッフリングに相当する量子ビットのスワップが必要となるが, 量子回路プロセッサ上では 12 ステップとなり, 合計 37 命令で通常のフーリエ変換と同じ機能を実現できる. プログラムリストを図 4.14 に示す. このルーチンに, 例えば状態 $(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \otimes |0\rangle$ を入力として与えると, $(|0\rangle + |1\rangle) \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle$ が返される.

量子フーリエ変換を量子回路プロセッサで行なった場合と TI 社製の固定小数点 DSP TMS320C6416 で処理した場合で比較を行なった. 通常の FFT は, シャッフリングを除くと複素乗算数が $(1/2)N \log N = n2^n/2$ 回, 加算が $N \log N = n2^n$ 回で実行される. TI 社製 DSP は, 複素乗算を 1 演算サイクルで実行可能であることから, 表 4.6 のような性能が推定される. ステップ数については DSP に比べ量子回路プロセッサでは 1/10 程度となっているが, 動作周波数が低いために実行時間で比較すると DSP に性能が劣る. 実装をカスタムプロセスで行ない, 動作周波数が 600MHz 程度と同等なレベルに上げられた場合, 8 量子ビット相当の量子フーリエ変換で 4.94ms となり, DSP と同程度の性能を得ることとなる.

この比較は, 並列化の効果が顕著に現れない場合があることを示唆している. その原因としては, CPHAS0,1 命令の実行中に並列化の効果が損なわれていることが挙げられる. すなわち, 制御ビットが 1 量子ビット入る場合 PE の半数で, 2 量子ビット入る場合には PE の

```
/*qunatum FT*/
PHAS0(4, pi)
ROT(4, pi/2)
ROT(4, pi/4)
CPHAS1(4, 3, pi/2)
PHAS0(3, pi)
ROT(3, pi/2)
ROT(3, pi/4)
CPHAS1(4, 2, pi/4)
CPHAS1(3, 2, pi/2)
PHAS0(2, pi)
ROT(2, pi/2)
ROT(2, pi/4)
CPHAS1(4, 1, pi/8)
CPHAS1(3, 1, pi/4)
CPHAS1(2, 1, pi/2)
PHAS0(1, pi)
ROT(1, pi/2)
ROT(1, pi/4)
CPHAS1(4, 0, pi/16)
CPHAS1(3, 0, pi/8)
CPHAS1(2, 0, pi/4)
CPHAS1(1, 0, pi/2)
PHAS0(0, pi)
ROT(0, pi/2)
ROT(0, pi/4)
/*SWAP*/
CROT(4, 0, pi/2)
CPHAS1(4, 0, pi)
CROT(0, 4, pi/2)
CPHAS1(0, 4, pi)
CROT(4, 0, pi/2)
CPHAS(4, 0, pi)
CROT(3, 1, pi/2)
CPHAS1(3, 1, pi)
CROT(1, 3, pi/2)
CPHAS1(1, 3, pi)
CROT(3, 1, pi/2)
CPHAS1(3, 1, pi)
```

図 4.14: 5 量子ビット相当の量子フーリエ変換用プログラムリスト．量子フーリエ変換本体に 25 ステップ，量子フーリエ変換後のスワップ操作に 12 ステップ要する．ここで，実装の簡単化のために，回転角を $\{\theta, \phi_0, \phi_1 | \pi, \pi/2, \pi/4, \pi/8, \ldots \pi/2^{n-1}\}$ と決め，各回転角の sin, cos の数表をハードウェア上に実装した．マシン語は回転角を数表のインデックスにより指定する．これにより，角度を直接指定する場合よりもステップ数が若干増加した．

表 4.6: 5, 8 量子ビット相当のフーリエ変換を行なった場合の所要ステップ数および所要時間の比較

チップ	5 量子ビット FFT		8 量子ビット FFT	
	ステップ数	時間 (ms)	ステップ数	時間 (ms)
TI DSP	240	0.40	3072	5.12
5 量子ビット QCP	27	36.00	—	—
8 量子ビット QCP	27	51.30	52	98.80

3/4 でユニタリ変換が行なわれないことになる. 最悪で, ターゲットビット以外のすべての量子ビットがコントロールビットとして働く場合には, 2PE でしか乗算が行なわれないことになる.

4.4.4 ショアのアルゴリズムの実験

8qubit 量子回路プロセッサを用いてショアのアルゴリズムの模擬を行なった. ここでは, 奇素数からなる最小の合成数 $N = 15[= 3 \times 5]$ の因数分解を例にとって実験を行なった. 前述の通り, 正確にショアのアルゴリズムを実行し, 多項式ステップ数で結果を得るためには, 因数分解すべき合成数のビット数 n に対し, 少なくとも $5n$ 量子ビットが必要となる. しかし, 本実験では扱える量子ビットが 8 量子ビットであるため, $2n$ 量子ビットに相当する大きさとなっている. 重ね合わせ $|a\rangle|x^a(\mathrm{mod}N)\rangle$ を作るために, 量子ビットの上位 4 量子ビットをレジスタ 1, 下位 4 量子ビットをレジスタ 2 と定義する. この 2 つの量子レジスタを用いて, 図 4.15 に示す量子回路で表現されるアルゴリズムを実行し, 因数分解を行なった. 手順は以下の通りになる.

(1) 両レジスタを $|0000\rangle$ に初期化する.

(2) 0 から 15 までの整数 a の重ね合わせをレジスタ 1 に生成する.

(3) レジスタ 1 の量子ビットをコントロールビットに持つユニタリ変換 π^i により, レジスタ 2 において $7^a(\mathrm{mod}15)$ を計算する. この結果, $|a, 7^a(\mathrm{mod}15)\rangle$ の重ね合わせが発生する.

(4) レジスタ 1 に量子フーリエ変換を行ない,

(5) 全量子ビットを観測することにより, レジスタ 1 から得られる結果から周期 r を求める.

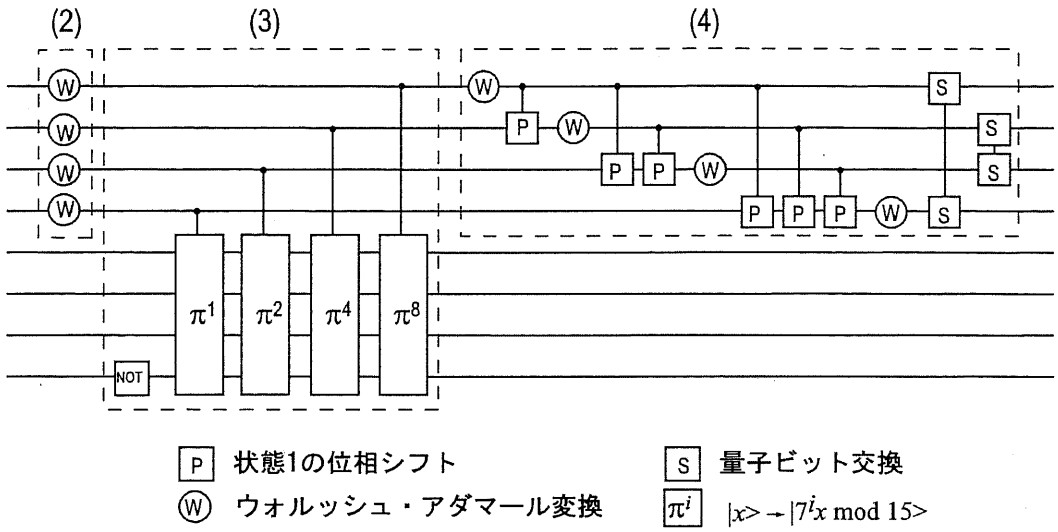


図 4.15: ショアのアルゴリズムを模擬する 8 量子ビットの量子回路.

このルーチンで用いたユニタリ変換 π^i はこの変換の入力 x に対し

$$x \longmapsto \begin{cases} 7^i x (\mathrm{mod} 15) & \text{制御条件が満たされた場合} \\ x & \text{その他の場合} \end{cases} \tag{4.17}$$

を計算する特殊なものである [27, 28]. このとき, 入力 $x = 1$ に対する変換 $\pi^8 \cdot \pi^4 \cdot \pi^2 \cdot \pi^1$ となっている図 4.15 の (3) 部分は

$$7^{(a_3 a_2 a_1 a_0)_2} \equiv 7^{2^3 a_3} \cdot 7^{2^2 a_2} \cdot 7^{2^1 a_1} \cdot 7^{2^0 a_0} (\mathrm{mod} 15) \tag{4.18}$$

を行なうことと等価になっている. この部分については, 通常は汎用の量子回路で設計することとなり, 実現方法によって $2n$ から $3n$ の量子ビットが必要となるため, 量子ビットが足りなくなる. そこで本研究では, 真理値表が表 4.7 で与えられる変換を量子ゲート命令により実現することにより, n 量子ビットで剰余部分を計算することとした. これは図 4.16(a) の量子回路で実現される [27, 28]. ただし, 表に示す以外の入力が与えられた場合には, 本来は剰余を返さなければならないが, 図の量子回路では入力値をそのまま返す. しかし, 式 (4.18) の機能を実現するのには十分である. これと同様に π^2 も構成することが可能であり, また π^4, π^8 はこの場合は恒等変換に等しい. ただしこの方法は, 設計の段階で剰余をすべて知らなければならないため, 設計コストに指数時間必要となり実際の計算では使用できない. また, 量子回路を小数段の量子ゲートで組むためには, 発見的な方法によらなければならない. 本実験においては, このゲート配列を山登り法により探索した. このゲートを図 4.16(b) に示す.

表 4.7: ユニタリ変換 π^i の真理値表.

π^1		π^2		π^4		π^8	
入力 x	出力 $f(x)$	入力 x	出力 $f(x)$	入力 x	出力 $f(x)$	入力 x	出力 $f(x)$
1	7	1	4	1	1	1	1
4	13	4	1	4	4	4	4
7	4	7	13	7	7	7	7
13	1	13	7	13	13	13	13

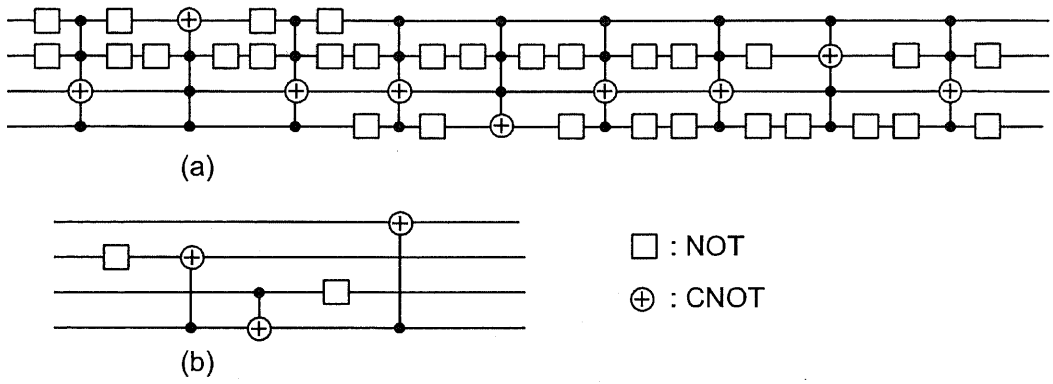


図 4.16: ユニタリ変換 π^1 を実現する量子回路. (a) は文献 [27, 28] に基づき組んだ量子回路である. これとは別に山登り法で (b) が得られる.

手順 (5) において, レジスタ 1 で観測されるのは, 求めるべき $7^a \pmod{15}$ の周期 r ではなく, フーリエ変換により強められる $2^4/r$ の整数倍である. いま, この例では周期が $r = 4$ であるため, 観測の結果 $\{0, 4, 8, 12\}$ のうちのいずれかが観測されることになる. ここから単一の r が正しく推定できるのは 4 と 12 であり, 推定した r をもとに, $\gcd(7^{r/2} - 1, 15)$ および $\gcd(7^{r/2} + 1, 15)$ から 15 の因数の候補を特定する. よって, 1 回のルーチンの試行により正しく解を導く確率は 1/2 ということになり, 実験は 3 および 5 を得るまで手続き (1)—(5) を繰り返す必要がある.

10000 回の実験における試行回数の統計結果を図 4.17 に示す. ここで, 観測を含めた手続き (1)—(5) は 131 命令で実行される. 実験の結果, 平均試行回数は 1.9177 回となった. これは, ハードウェア実装した疑似乱数発生装置が正しく動作していることを示している.

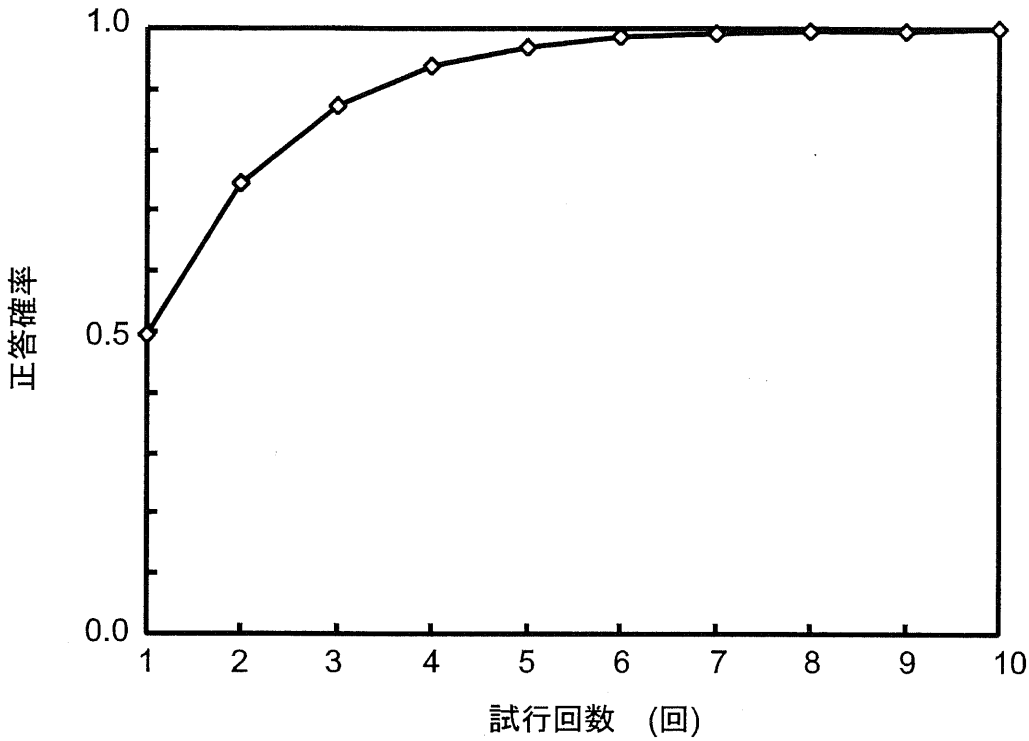


図 4.17: 試行回数と正答率. 横軸は試行回数, 縦軸はその試行回数により正答が得らる確率を示す. 実験は 10000 回行なった.

4.5　まとめ

量子回路によって記述された量子アルゴリズムを量子コンピュータと同じステップ数で処理する並列プロセッサを試作し, 実現可能性について検討した. プロセッサのハードウェアーキテクチャを量子ゲート操作のシミュレーションに合わせて最適化する上でフラクタルな構造を利用することから, 大規模なシミュレーションシステムの設計を効率化することが可能となった.

PLD を利用した実装を実際に行ない, 量子フーリエ変換とショアの因数分解のアルゴリズムを実際に実行した. 実験を行なった結果, ネットワーク部分の性能を向上させることが重要であることが明らかになった. これを解決する手段としては, 通信をパイプライン化することなどが考えられる.

本章で行なった試作機は, 5 量子ビットあるいは 8 量子ビット規模に留まった. より実際的なアルゴリズムを実行するためには, さらに大規模なシステムを構築する必要がある. そ

の場合, ローカルメモリをPE内部に実装し, 複数の量子ビットの振幅データを1PEに担当させて計算を行なうという解が考えられる. この時, 量子回路の規模とゲート操作実行速度の関係を考慮して, PE当たりのメモリ容量とPE数の関係を適正な値にする必要がある.

第 5章

確率振幅を1ビットで表現する量子回路プロセッサ

5.1 はじめに

前節で述べた量子回路プロセッサでは, 量子コンピュータのシミュレーションに特化したものであったが, これを汎用プロセッサに発展させる手法について考える.

そもそも, 量子回路プロセッサは汎用の計算が可能であるが, すべての計算は量子力学で許される変換によって行なわれなければならないため, アルゴリズムが却って複雑になり, 一般的な計算の高速化は期待できない. しかし, 量子回路プロセッサは古典系の一種である集積回路を用いた計算システムであり, 量子力学的には許されない演算操作も簡単な拡張で実装可能である. 例えば, ユニタリ対称性を持たない行列による変換操作や系を任意の状態へ初期化すること, あるいは量子回路プロセッサの REDUCE 命令のような任意の状態を重ね合わせから除去するような操作が, 通常の量子回路における演算操作と共に利用できる. このような古典的な操作を量子回路プロセッサに導入し量子アルゴリズムの非効率な部分を補うことによって, 新しい計算パラダイムを開くと期待される.

そこで本章では, 量子回路プロセッサの再帰構造を持ったハードウェア上で量子アルゴリズムを模した情報処理を行なう, 新しいタイプの並列演算手法について検討する. この検討においては, 将来的にナノデバイスでシステムを構築することを念頭に, 量子回路プロセッサでは実部, 虚部をそれぞれ 16 ビットで表現していた確率振幅を 1 ビットで表現するなど, 大幅な変更を行ない, より単純な PE をより大量に用いることにより, 設計コストを下げつつ並列性の高い計算を実現することついて堀下げる.

5.2 計算手法と命令アーキテクチャ

量子コンピュータのシミュレーションを行なう際, 振幅のビット幅は $O(n)$ だけ必要になることは前述の通りである. いま, シミュレーションに用いた計算手法を簡単化して量子アルゴリズムと同等の計算を行なうことを考えると, 1ビットで振幅情報を表現しこれを操作することにより演算を行なうことが, その究極的な姿として想定される. ここで, 実部, 虚部合わせて32ビットで表現していた振幅を1ビットに圧縮すると, 位相が表現できなくなる. 位相情報を持たない1ビット振幅データに対しては, 通常の量子ゲート操作は有効に働かないため, これに適した演算操作の集合を考えなければならない.

ここで, 再度量子回路を用いて量子アルゴリズムを実行することを考える. 図4.15に示すようなショアのアルゴリズムを例にとってもわかる通り, 量子アルゴリズムはいずれも大別して3つの手続きによって構成される.

(1) $|0\rangle$ への初期化を行なう.

(2) ウォルッシュ·アダマール変換を各量子ビットに作用させ, 解候補の重ね合わせ状態を生成する.

(3) CNOT で構成される論理演算を行なう.

(4) 量子フーリエ変換等を利用して, 解候補のうちから適当な状態の確率振幅を強める.

(5) 観測を行ない, 適当な状態を取り出す.

手続き (1), (2) を行なう場合, ウォルッシュ·アダマール変換の入力は常に $|0\rangle$ であり, これに対し変換を行なった後の状態は

$$|\Psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n} |x\rangle \tag{5.1}$$

と, 各状態の確率振幅は全て共通である. さらに, CNOT で構成される論理演算を実行している間は量子ビット数などの条件によらず, 2値の振幅 $\{0, 1/\sqrt{2^n}\}$ を与えられた条件によってPE間で交換しているに過ぎない. よって, これを $\{0,1\}$ の2値として, PEに振幅 $\{0,1\}$ を代入することにより, ウォルッシュ·アダマール変換がシミュレート可能であり, また, PEの1ビット振幅データをPE間で交換することによって手続き (3) もシミュレート可能となる.

手続き (4), (5) は, 振幅が1ビットの場合は直接シミュレートできないが, 前章の量子回路プロセッサのREDUCE命令を絶対観測モードにより行なう場合と同様に, PEアドレスがある条件に合致するPEの振幅データのみを恣意的に残すことによって, 後述するとおり, 量子フーリエ変換等の振幅を強める操作と観測の組合せを用いずとも目的を達成することが可能である.

上記のような1ビットの振幅を操作する命令セットを次のように定義した.

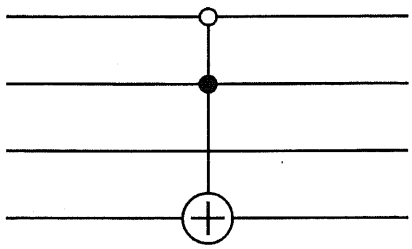


図 5.1: 拡張 CNOT のシンボル. 白丸で示されたビットが 0 でかつ黒丸で示されたビットが 1 の場合にターゲットビットにおける NOT 操作を行なう.

CNOT

量子回路における CNOT と同様な機能を有する. すなわち, 量子回路においてターゲットとして指定されたビットがコントロールビットとして指定されたビットの条件により反転されるように, PE アドレスのうちのターゲットビットに相当するビットが 0,1 と異なる PE の対のうち, コントロールビットにより指定された PE アドレスに合致する PE 対のみで振幅データの交換を行なう. ここで, 反転を行なう条件は, 複数のコントロールビットの各々に対し 0 もしくは 1 どちらの場合に反転を行なうか指定が可能であるように拡張を行なった. コントロールビットの指定は, {0, 1, x} からなるビットマスクにより行なう. ただし x は不定を示す. ビットマスクが満たされた場合にのみ, 演算が行なわれる.

例えば, コントロールビットを表すビットマスクを 01xx のように指定し, ターゲットビットを最下位ビットとする場合, 最下位ビットが異なる PE 対のうち, PE アドレスの上位 2 ビットが 01 である対でデータ交換を行なう. これを図 5.1 のようなシンボルで表現することにする. 図中白丸は, 該当するビットが 0 の時に N0T 操作をターゲットビットに対して行なうことを示している.

この操作により量子回路における CNOT がエミュレートでき, 従って完全な論理操作が実現できる.

GEN/ELIM

GEN (geneerate), ELIM (eliminate) は, 指定された状態の振幅を 1/0 にする. CNOT と同様に, 状態の指定はビットマスクにより行なわれる. 例えばビットマスク 01xx を指定すると, 上位 2 ビットが 01 である状態はすべて操作の対象となる.

この操作により, ウォルッシュ·アダマール変換をエミュレートできる. すなわち, 4 量子ビットの量子回路であればビットマスク xxxx による GEN を行ない, 全 PE の振幅データを 1 とすることで全ての状態の重ね合わせを生成できる.

表 5.1: ディジタル量子回路プロセッサの全命令とその機能

ニモニック	機能
CNOT(q_t, b)	ビットマスクbが満たされる場合, ターゲットビット q_t に対し, NOT 演算を行なう. ビットマスクは $(0, 1, x)$ からなる. ただし, xは不定.
GEN(b)	ビットマスクbを満たす状態の振幅を1にする.
ELIM(b)	ビットマスクbを満たす状態の振幅を0にする.
OBS(b)	ビットマスクbを満たす状態の振幅の論理和をとり, ホストに結果を返す. このときプロセッサは, ホストから再始動されるまでの間停止状態となる.

OBS

同様に, アドレスがビットマスクを満たすPEの振幅の論理和を計算し, 結果をホストに返す. 例えば01xxの場合は, アドレスの上位2ビットが01であるPEの振幅の論理和を計算する. 計算実行後はホストによりリスタートされるまで停止状態に入る.

返された結果によってホストがプロセッサをリスタートする方法を選択するようにプログラミングを行なえば, 様々なアルゴリズムを実行できる. 例えばこのOBSと前述のELIMを組み合わせることにより, 前章の絶対観測と類似の操作を行なうことが可能となる. まず, ビットマスク0によってOBSを行なうと, 0の状態に1つでも振幅データ1のPEがあれば, ホストに1が返される. これを確認した後, プロセッサリスタート時にビットマスク1によりELIMを行なえば, 絶対0モードでREDUCEを行なったことと等価になる.

以上4命令の機能を表5.1にまとめる. 振幅を0, 1の2値で表すことから, この命令アーキテクチャをディジタル量子回路プロセッサと命名した.

5.3 ディジタル量子回路プロセッサを用いた計算アルゴリズム

基本的な動作と, 量子コンピュータで行なわれるGroverのアルゴリズムを用いたSATの解法およびShorの因数分解を模倣するアルゴリズムについて述べる.

5.3.1 最小値および最大値検索

ディジタル量子回路プロセッサでは, 前章の量子回路プロセッサのREDUCE命令を使った方法と類似の手法により, 重ね合わせに属する状態のうちの最小値, 最大値を多項式ステップ数で検索することが可能である. 仮定する問題は $\sum_x |x\rangle|f(x)\rangle$ という重ね合わせ状態の $f(x)$ の最小値のみを導くと言うものである.

アルゴリズムは, OBSとELIMを組み合わせて2分木検索と類似の操作を行なうものである. 例えば最小値検索の場合, 次の基本操作を $|f(a)\rangle$ に相当するPEアドレスビットに対して行なうことにより達成される.

(1) 最上位アドレスビットが '0' であるPEのうち, 少なくとも1つに振幅が1であるPEが存在することをOBS(0xx...x)により確認する.

(2) OBSの結果が1であれば, 最上位アドレスビットが '1' であるPEが持つ振幅をELIM(1xx...x)により0にする. このとき, $f(x)$ の最上位ビットは '0' と確定する. 反対に, OBSの結果が0であれば何もしない. このときは, $f(x)$ の最上位ビットは '1' と確定する.

(3) 上記2操作をPEアドレスの最下位ビットまで順に行なう.

最大値検索の場合, OBSのビットマスクは1xx...xとなり, ELIMを実行するアドレスビットは0xx...xとなる. この基本操作を同じく最下位に向かって順に行なうことにより遂行される.

この操作を行なった後, x と $f(x)$ が1対1対応していれば, x にも単一の解が得られるため, 上記の手続きをPEアドレスの x に相当するビットに順に行なうことにより, 最小値を与える x を求めることが可能となる. この操作は, x のビット長の多項式程度のステップ数で実行可能である.

5.3.2 充足可能性問題 (SAT) の多項式時間アルゴリズム

充足可能性問題 (SAT) については, 第2章でグローバーのアルゴリズムに関連して触れた. 充足可能性問題は, 任意のブール式が充足可能か否かを判定するものである. ここでブール式は, 和積標準形で書かれる. 例えば,

$$f(x_0, x_1, x_2, \ldots, x_{n-1}) = (x_0 + \overline{x_4} + \overline{x_7})(\overline{x_1} + \overline{x_2} + x_8)\cdots \tag{5.2}$$

のように3つのリテラルの和の積をとったものが論理式として与えられた場合, この充足可能性判定は3充足可能性問題 (3-SAT) と呼ばれる. n 変数のSATの場合, 論理の組合せは 2^n 通りあるため, シラミ潰しを行なうと計算に $O(2^n)$ ステップかかる. SATはNP完全問題であることが知られている. 先に述べたように, この問題を量子コンピュータで解く場合は,

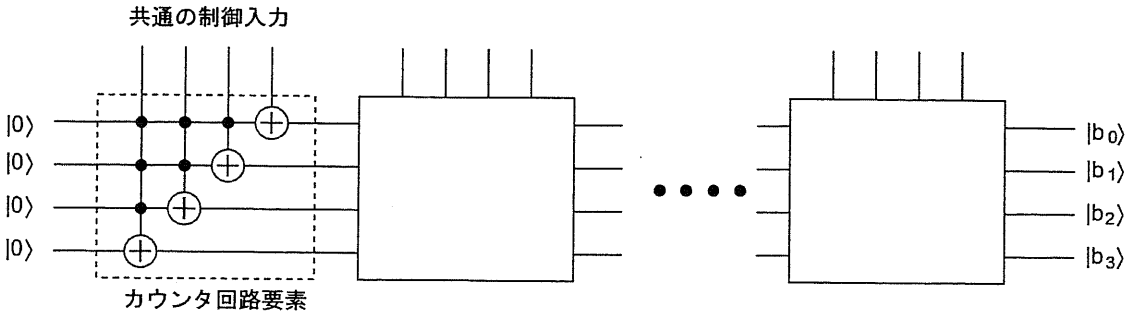


図 5.2: 量子カウンタ. $\log m$ 量子ビットのカウンタ回路要素を直列に m 段繋げると, 全要素のうちのいくつに制御信号が入力されたかををカウントし, 結果を出力する.

$O(2^{n/2})$ ステップかかる. しかし, ディジタル量子回路プロセッサでは多項式ステップ数で SAT を解くことができる.

式 (5.2) の括弧で括られた, 3 つのリテラルの和を節と呼ぶ. この節の全てが充足であったとき, f が充足である. いま, f の否定を考えると, これは逆に積和標準形で

$$\overline{f(x_0, x_1, x_2, \ldots, x_{n-1})} = \overline{x_0} x_4 x_7 + x_1 x_2 \overline{x_8} + \cdots \tag{5.3}$$

と書ける. よって, 積項の値の論理和が 0, すなわち値が 1 となる積項数が 0 ならば f は満たされる. そこで, カウンタと名付ける図 5.2 の量子回路を使って問題を解く. この量子カウンタは, カウンタ回路要素 m 段から構成される. 量子ビットを $\log m$ 個使うことにより, カウンタ回路要素のいくつに制御信号 1 が入力されたかを m までカウントし, 出力する. n 変数, 節数 m の SAT を解くには, 変数 $x_0, x_1, \ldots, x_{n-1}$ に対する値の割り当て方全てに対しこのカウンタを用いて積項の値が 1 となる数を数えあげ, カウンタの出力が $|0\rangle$ となる場合の有無を確認すれば良い. すなわち, 各カウンタ回路要素に対する制御入力に, 各積項の論理式を指定しておけば良い. m 個の節全てについてこの操作を行なえば, 数え上げが終了する. 例えば, 式 (5.3) を実現する量子回路を図 5.3 に示す. 同回路は変数 $x_0, x_1, \ldots, x_{n-1}$ を入力する量子ビット n 個とカウンタ用量子ビット $\log m$ 個からなる. 変数部に全ての値の割り当て方の重ね合わせを入力すれば, カウンタ出力にそれら全てに対応する充足された積項の数が重ね合わせて出力されるため, 前述の最小値検索をカウンタ部に行ない結果が $|0\rangle$ と一致することが確認されれば, f は充足可能と判定される. このとき変数側には, 式 f を充足する値の割り当て方の重ね合わせが存在するため, 適当に 2 分検索を行なうことにより少なくとも 1 つの割り当て方が得られる.

手続きを以下にまとめる.

(1) 変数代入用の量子ビット n ビットとカウンタ用量子ビット $\log m$ ビットを準備し, 変数

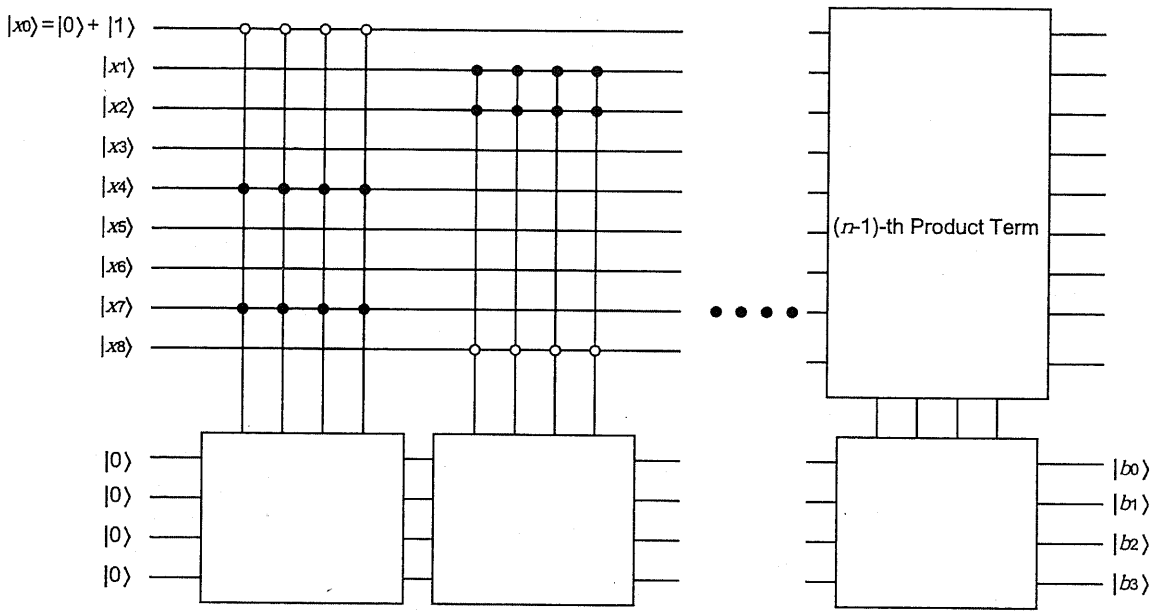


図 5.3: 式 (5.3) の充足可能性を判定するための量子回路. n 変数, m 節の問題の場合, 量子ビット数は $n + \log m$, 段数は $m \log m$ 段となる.

代入用の各ビットに $|0\rangle, |1\rangle$ の重ね合わせ状態を生成し, カウンタ用の $\log m$ ビットは $0\rangle$ に初期化する. これは, 変数用の n ビットに xx...x, カウンタ用の $\log m$ ビットには $00\ldots0$ が適用されるビットマスクにより GEN を実行することにより実現される.

(2) 式 (5.3) に示す積項の第1項目のリテラルをコントロールとするカウンタ回路要素を実行する.

(3) 同操作を m 項全てに対して行なう.

(4) 最後にカウンタ出力部に対して最小値検索を行なう. この結果が $|0\rangle$ であれば, 充足可能であると判定される.

(5) 充足する論理値の割り当ては, 変数を適当に2分検索することにより得られる.

本アルゴリズムに必要な量子ビット数は $n + \log m$, 計算ステップ数は $m \log m$ と観測に必要なステップ数の和となる.

5.3.3 因数分解アルゴリズム

Shor の因数分解アルゴリズムと類似の手法により, 因数分解をディジタル量子回路プロセッサによって行なうことが可能である. アルゴリズムの目的は, Shor の方法と同様に

$$x^a \equiv 1 (\mathrm{mod} N) \tag{5.4}$$

の解 a の位数を求めることである. ウォルッシュ・アダマール変換により全ての整数の重ね合わせを生成することがエミュレートできること, CNOT を用いた論理演算が可能であることはこれまで述べたとおりであり, これらを使えば状態 $\sum_a |a\rangle|x^a(\mathrm{mod} N)\rangle$ を生成することは量子回路と全く同様に可能である. 問題は, 量子フーリエ変換を用いずに, いかに a の位数を求めるかである.

この手法を考えるために, 式 (5.4) に立ち戻って考える. この方程式の解は, 求める位数を r とおくと, $a = 0, r, 2r, 3r, \ldots$ となる. ここから r を求めるために, Shor のアルゴリズムでは (5.4) の左辺を全ての a について求めておき, 剰余のうちの1つ M を観測によって定め, これを与える $a = l, r+l, 2r+l, 3r+l, \ldots$ の周期性をフーリエ変換によって導いた. これは, 剰余の1を恣意的に選ぶことができないために行なわれる操作であるが, ディジタル量子回路プロセッサにおいては剰余1を選ぶことが多項式ステップで可能である. すなわち, 重ね合わせ $|x^a(\mathrm{mod} N)\rangle$ に対し, ビットマスク (1xx...xx), (x1x...xx), ..., (xx...1x), (xx...x0) により ELIM を連続して行なうことによって $|x^a \equiv 1(\mathrm{mod} N)\rangle$ のみを残すことが可能である. この操作を行なうことにより, 重ね合わせのうち $|a\rangle$ の部分には $a = 0, r, 2r, 3r, \ldots$ しか残らないことになる. よって, $|a = 0\rangle$ を予め ELIM により除外しておけば, 前述の最小値検索を用いて r を確率的な方法によらずに導くことができる. よって, a の重ね合わせには,

$\log N = n$ ビット準備しておけば十分計算が実行可能であり，量子ビット数を減らすことも可能となる．

アルゴリズムをまとめると，次のような一連の操作となる．

(1) 全ての状態の振幅を0に初期化する．これは，全ビットをxとするビットマスクxx...xにより，ELIMを実行することにより実現される．

(2) ウォルッシュ·アダマール変換のエミュレーションにより $|a\rangle$ の重ね合わせを生成する．これは，$|a\rangle$ のビットにxxx...x，その他のビットに000...0が適用されるビットマスクによりGENを行なうことで実現される．

(3) CNOTを適当に用いて重ね合わせ状態 $\sum_a |a\rangle|x^a(\mathrm{mod}N)\rangle$ を生成する．

(4) 自明な解 $a = 0$ を重ね合わせから除外する．これは，$|a\rangle$ に000...00，$|x^a(\mathrm{mod}N)\rangle$ にxxx...xxが適用されるビットマスクによりELIMを行なうことにより実現される．

(5) $|x^a(\mathrm{mod}N)\rangle$ が1となるものと対応する a のみを重ね合わせ状態中に残す．これは，ELIMを n 回程度行なうことにより実現される．このときビットマスクは，$|a\rangle$ のビットには常にxxx...xが，$|x^a(\mathrm{mod}N)\rangle$ のビットには(1xx...x)，(x1x...xx)，...，(xx...1x)，そしてxx...x0が順に適用される．

(6) $|a\rangle$ に関する最小値検索を行なう．これにより位数 r を得ることができる．

以上により得られた r が奇数である場合は，x を変更し，再びアルゴリズムを最初から実行する．偶数ならば，得られた r から N の因数の候補 $\gcd(x^{r/2-1}, N), \gcd(x^{r/2+1}, N)$ を求め，ユークリッドの互助法により因数であることを確かめる．この手続きにより因数が得られなかった場合は，x を変更し，再びアルゴリズムを実行する．

以上の位数 r を求めるアルゴリズムは，多項式ステップで実行可能であり，かつ確率的な手続きを行なうことなく実行可能である．また，r が偶数でかつ r から因数の候補が得られる確率は1/2より大きいことが示されている．

ディジタル量子回路プロセッサで8ビットの因数分解を行なうプログラムを図5.4に示す．ただし，CNOTで $x^a(\mathrm{mod}N)$ を求める際に必要な補助ビットは省略してある．

5.4 PLDを用いた実装

ハードウェア·アーキテクチャの基本は量子回路プロセッサと同じである．ただし，ロジック部分に関しては加算器等がなくなるため，大変シンプルになる．

ディジタル量子回路プロセッサの実装では，前章までの実装形式とは異なり，新しく振幅データストレージ用にメモリを付加した．振幅更新の計算を行なう際，このメモリに対し随時データの読み出し/書き込みを行なう．図5.5にPEの内部構造を示す．メモリを適当な大

```
ELIM(0000000_000000000)          /* (1) */
GEN(xxxxxxxx_00000000)        /* (2) */
/*
 logic process with CNOT            (3)
*/
ELIM(00000000_xxxxxxxx)          /* (4) */
ELIM(xxxxxxxx_1xxxxxxx)          /* (5) */
ELIM(xxxxxxxx_x1xxxxxx)
ELIM(xxxxxxxx_xx1xxxxx)
ELIM(xxxxxxxx_xxx1xxxx)
ELIM(xxxxxxxx_xxxx1xxx)
ELIM(xxxxxxxx_xxxxx1xx)
ELIM(xxxxxxxx_xxxxxx1x)
ELIM(xxxxxxxx_xxxxxxx0)
OBS(0xxxxxxx_xxxxxxxx)           /* (6) */
/* restarted by host */
ELIM(1xxxxxxx_xxxxxxxx)
OBS(x0xxxxxx_xxxxxxxx)
/* restarted by host */
ELIM(x1xxxxxx_xxxxxxxx)
OBS(xx0xxxxx_xxxxxxxx)
/* restarted by host */
ELIM(xx1xxxxx_xxxxxxxx)
OBS(xxx0xxxx_xxxxxxxx)
/* restarted by host*/
ELIM(xxx1xxxx_xxxxxxxx)
OBS(xxxx0xxx_xxxxxxxx)
/* restarted by host*/
ELIM(xxxx1xxx_xxxxxxxx)
OBS(xxxxx0xx_xxxxxxxx)
/* restarted by host*/
ELIM(xxxxx1xx_xxxxxxxx)
OBS(xxxxxx0x_xxxxxxxx)
/* restarted by host*/
ELIM(xxxxxx1x_xxxxxxxx)
OBS(xxxxxxx0_xxxxxxxx)
```

図5.4: 振幅1ビットで因数分解を行なうプログラム.

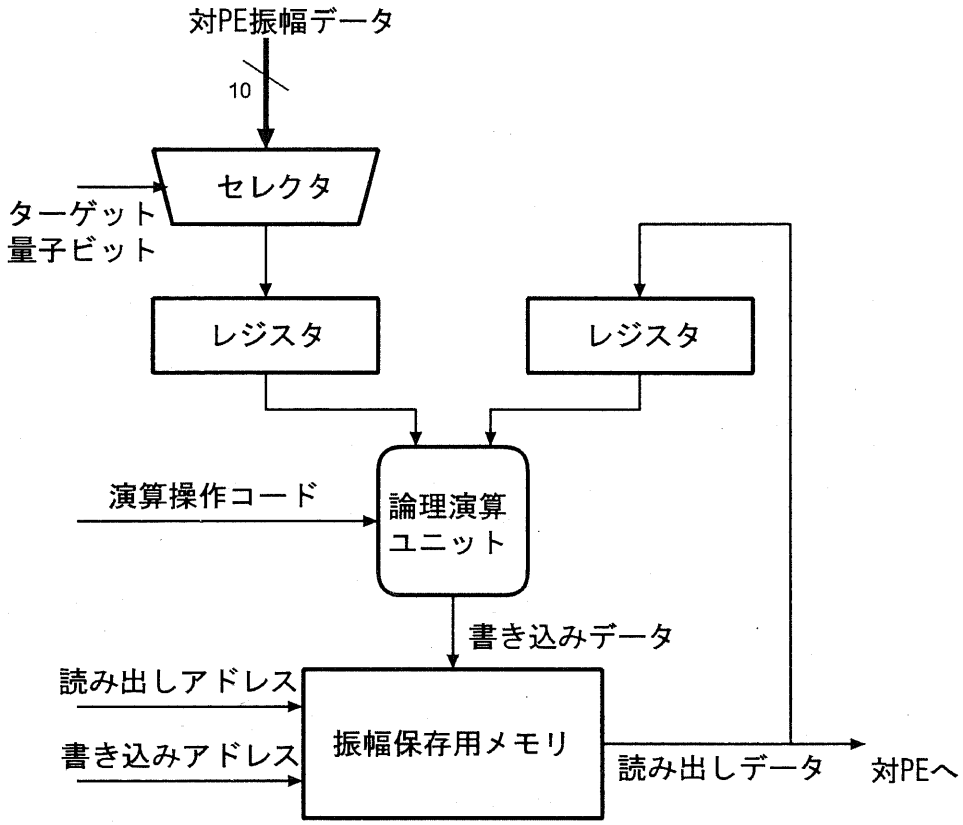


図 5.5: ディジタル量子回路プロセッサの PE 内部回路. PE はそれぞれ振幅データ保持用のメモリを有する.

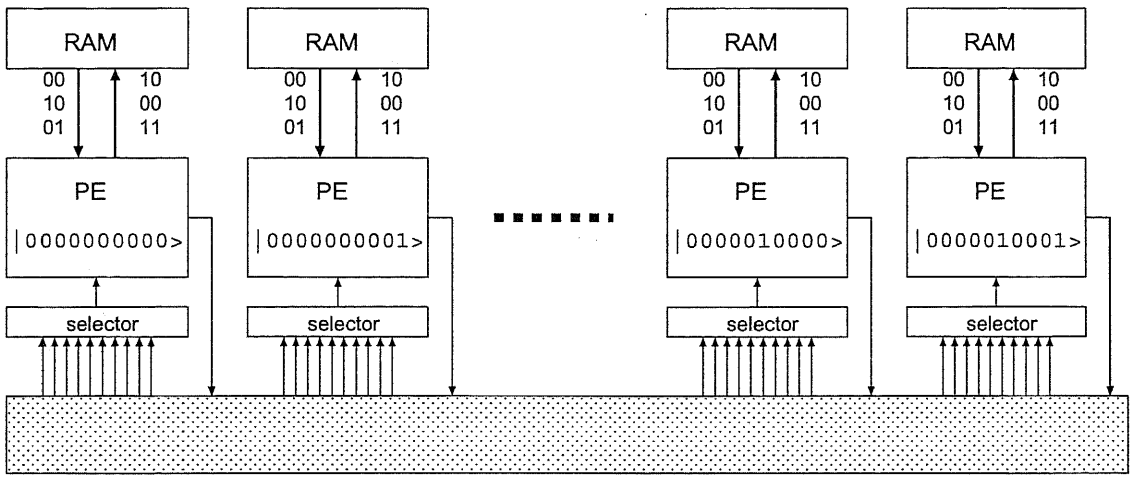
表 5.2: プロセッサの主要な仕様.

PLD タイプ	Altera EP20K1500EFC33-3
PLD 使用数	1 枚
等価的量子ビット数	16
PE 数	1024 (10 量子ビット相当)
メモリ容量	64 キロビット (64 ビット ×1024)
ゲート数	1.3×10^6
クロック速度	60MHz

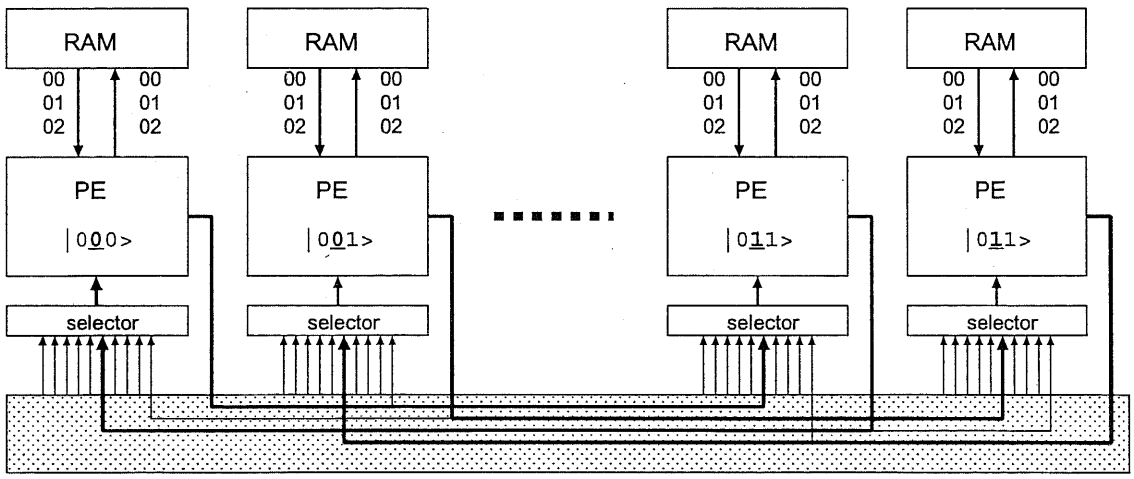
きさで各 PE に実装することにより, 計算速度と量子ビット数の間のトレードオフを任意にとることが可能である. 例えば, 各 PE に実装されたメモリ容量が 2^l ビットで, 同 PE が 2^m 個であれば, 全体で $(l+m)$ 量子ビット相当の計算が可能となる. ただし, 1PE が担当するデータ数 2^l だけ速度低下が起こる. この速度低下は, ユーザが許容できる程度に抑えられるよう留意されなければならない. $(6+10)$ 量子ビット規模のディジタル量子回路プロセッサを, 実際に量子回路プロセッサと同じ Altera EP20KE33 に実装した. ここで, l, m の規模は, 利用した PLD の回路構成を考えて最適化を行なった結果得られたものである.

プロセッサの動作は, PE 間通信が行なわれる場合と行なわれない場合に分けられる. PE 間の通信が行なわれない場合, PE 内部のデータを適当な順で読み出して計算を行なうことにより, PE 単位で $l = 6$ 量子ビット相当の計算が行なえる. 同じ動作を $m = 10$, すなわち 2^{10} 個の PE が同時に行なうため, 結果として 16 量子ビットの量子回路の下位 6 ビット分の計算が PE 間通信なしで行なわれる. 例えば, 下位から 5 ビット目に CNOT の演算が行なわれる場合は, 図 5.6(a) に示すように, 各 PE のローカルメモリに格納された $2^6 = 64$ 個のデータがアドレス 16 進表記で 00, 10, 01, 11,... の順でロードされ, パイプライン処理により 10, 00, 11, 01, ... のように交換されたデータが順に書き込まれる. 一方, PE 通信が行なわれるのは, 上位 10 ビットのうちの 1 つに演算が行なわれるような場合である. この時は, PE 内部メモリのデータ全てが一斉に交換され順次パイプラインで新しい振幅データが生成され, メモリに書き戻される. 例えば, 下位から $6+5$ ビット目に CNOT を行なうような場合, 図 5.6(b) のように PE アドレスの 5 ビット目が互いに異なる PE 対が通信を行ない, 64 個のデータが対の PE に順々に転送される. 転送されたデータは順に図 5.5 の "pertner in" に格納され, メモリから読み出され "self in" に格納されたデータと共に PE 内部で処理を行なわれた後, 順に PE のメモリに格納される.

このプロセッサの動作仕様を表 5.2 に示す. ここで, 等価的な量子ビット数は本実装の場合, PLD 内部の SRAM の大きさおよびロジック数によって決定したものである. また, 各命令の必要クロックサイクル数は, PE 内部に実装された SRAM のデータ転送量によって支配



(a)



(b)

図 5.6: プロセッサの動作. (a)PE 間通信が起こらない場合と (b) 起こる場合をそれぞれ示す.

表 5.3: 16 量子ビットプロセッサの命令実行に必要なクロックサイクル数

命令	クロックサイクル数
CNOT	64
GEN	64
ELIM	64
OBS	95

されるため, 表5.3のように決まる. 結果として, 16ビット相当の量子ゲート操作1つに相当する論理演算を1秒間に 9.4×10^5 回行なうことが可能である. すなわち, 状態1つの生成を演算1回と数えると 61GOPS(Giga Operations Per Second) 相当の演算能力を有することになる.

5.5 まとめ

本章では, 量子アルゴリズムを模した演算を量子回路プロセッサと同じアーキテクチャのハードウェアで行なうことを通じ, 新しいタイプの並列演算を提案する試みを行なった. その結果, 単純な PE を多数用い, 各 PE 上で 1bit のデータを取り扱うことにより, 量子コンピュータと同様な演算を行なうことが可能であることを示した. このプロセッサが通常のプロセッサの能力を凌駕するとは現時点では考えにくいが, 将来的に, ナノデバイスによる巨大な集積システムを構築する際に, その単純な構造がシステム設計に大変有利に働く場合があると期待される. 本提案方式は, このような観点から今後の発展を考えるべき技術と言える.

第 6章

結論

6.1 本研究の主たる成果

量子コンピュータ研究の進捗を念頭に, 本研究では集積回路の技術を駆使して高速かつ大規模なシミュレータを作製する手法を提案し, その実現可能性について議論した. 以下得られた結果をまとめる.

第 3 章では, 量子コンピュータの重ね合わせ状態を集積回路中の周期信号に置き換え, これをディジタルフィルタ・クラスタにより変換するエミュレータについて述べた. 実際に 3qubit 規模のシステムを PLD で実現し, グローバーのアルゴリズムの実験を行なった. このシステムを用いて, 量子コンピュータの内部状態の変化を模倣することが可能である.

第 4 章では, デバイス数 $O(2^n)$ で n 量子ビットの量子コンピュータを多項式時間でシミュレート可能な量子回路プロセッサについて述べた. これは, SIMD 型並列処理で行列を高速に処理するものである. 3 章で述べたエミュレータとは異なり, デバイスを $O(2^n)$ 用いて計算時間を多項式に抑えることが可能である. プロセッサのハードウェアアーキテクチャは, 拡張性に優れたフラクタル構造を持っており, 設計コストを抑え, システムの大規模化に寄与するものと考えられる.

第 5 章では, 量子回路プロセッサ上で行なわれる量子回路のシミュレーションと同等の演算を 1bit の振幅で行なうことを通じて, 新しいタイプの並列演算を行なう試みを行なった. 実際に Shor のアルゴリズムを簡単な論理回路の集合体で行なうことが可能であることを示した. この方式をディジタル量子回路プロセッサと命名した. これは, 現時点では通常のプロセッサの性能を上回るものになるとは言えないが, ナノスケールデバイスにより構成された巨大な集積計算システムを構築する際に有用な技術として捉えて行くべきと考えられる.

6.2 総括

本研究では, 大規模でかつ高速な量子コンピュータのシミュレータを集積回路によって実現することについて考えてきた. ここで得られた知見をもとに, 集積回路と量子コンピュータ, およびその境界領域の将来像を最後に議論する.

本研究は, P=NP 問題をその動機として出発した. この P=NP 問題は計算科学の分野で今後も議論が引き続いて行なわれることになるが, 現在のところ, この結果の予想としては P≠NP という見方が大勢を占めている. よって, NP 問題に対応が可能な手法として, 現在広く行なわれているように, 近似アルゴリズムを用いた方法をとるか, 我々の提案した手法のようにデバイスの並列動作により大規模かつ高速な計算を行なっていくかの 2 つが依然として有力と言える. 本研究で作製した並列プロセッサは, 大量のデバイスを多並列に動作させることにより量子コンピュータのシミュレーションを高速に行なうものであり, 後者の手法の一つと言える.

しかし, 本研究で考えたプロセッサは, 単純に汎用プロセッサをアレイ状に並べて並列処理を行なうのではなく, 非常に単純な構造の PE を大量に用い, これら全体で因数分解などの実用的な処理を行なうものに発展させられる. 今後, 集積回路の分野ではナノメートルデバイスの集積技術の開発に力が注がれていくこととなるが, 究極的の微細デバイスによって構成される巨大システムが実現されるようになった際に, このような非常に単純な要素を規則的に並べることによって高性能計算が実現できる計算機方式は, 設計コストや実現可能性の面から重要性を増してくるかも知れない. このような期待を念頭に, PE を更に簡単化し, PE 間の通信を効率化する方向に研究を進めていく必要があろう.

また, 量子計算の分野においても本論中で述べたように, 量子ビット数を減らす工夫が必要である. 集積回路の集積度が上がれば, 量子アルゴリズムの研究に対して本研究で提案したプロセッサの寄与が大きくなっていくことが期待される. また, ここで得られたアルゴリズム高性能化の成果は, 前述の新しい多並列処理方式にも適用可能であり, 2 つの研究は相補的に発展していくものと思われる.

量子コンピュータと集積回路の境界領域にはじめて取り組んだ本研究が, この 2 つの分野における今後の発展に寄与することを期待する.

本研究に関する発表

学会誌論文 (査読あり)

(1) S. O'uchi, M. Fujishima and K. Hoh: "Fractally-Structured CMOS Processor for Quantum-Circuit Emulation", Jpn. J. Appl. Phys. **41** (2002), to be published.

国際会議講演 (査読あり)

(1) S. O'uchi, M. Fujishima and K. Hoh: "Emulation of Quantum Computing by Parallel Finite Impulse Responses", in *Extended Abstract of the 1999 Int. Conf. on Solid State Devices and Materials*, Tokyo, Japan (1999) p. 96.

(2) S. O'uchi, M. Fujishima and K. Hoh: "A Programmable SIMD Processor for Universal Quantum-Circuit Simulator", in *Extended Abstract of the 2001 Int. Conf. on Solid State Devices and Materials*, Tokyo, Japan (2001) p. 402.

(3) S. O'uchi, M. Fujishima and K. Hoh: "An 8-Qubit Quantum-Circuit Processor", 2002 IEEE International Symposium on Circuits on Circuits and Systems, Scottsdale, Arizona, USA (2002) to be presented.

国際会議講演 (査読なし)

(1) M. Fujishima, S. O'uchi and K. Hoh: "Programmable Quantum Computing Emulator Implemented on FPGA Boards", in *Bulletin of the Stefan University Vol. 13*, FSRC-Frontire Science Research Conf. Science and Technology of Silicon Materials–2001, La Jolla, CA, USA (2001) p. 23.

国内大会講演および研究会講演

(1) 大内 真一, 今村 晃, 千葉 智子, 藤島 実, 鳳 紘一郎: 第 2 回量子情報技術研究会資料 (電子情報通信学会 量子情報技術時限研究専門委員会, 大阪, 1999) p. 27.

(2) S. O'uchi, M. Fujishima and K. Hoh: "Hardware Emulation of Quantum Computing: Toward the Realization of Quantum-Comparable Computation Ability on Silicon", *1st CREST Symp. on Function Evolution of Materials and Devides based on Electron/Photon Related Phenomenon*, Tokyo, Japan (2000) p. 81.

(3) 大内 真一, 藤島 実, 鳳 紘一郎: "量子回路プロセッサ", 応用物理学会学術講演会講演予稿集 (2001) p. 127.

(4) 大内 真一, 藤島 実, 鳳 紘一郎: "8-qubit CMOS 量子プロセッサ", 第 5 回量子情報技術研究会資料 (電子情報通信学会 量子情報技術時限研究専門委員会, 厚木, 2001).

(5) M. Fujishima, S. O'uchi and K. Hoh: "Fractal-Structured Quantum Processor", in *2nd CREST Symp. on Function Evolution of Materials and Devices based on Electron/Photon Related Phenomenon*, Tokyo, Japan (2001) p. 86.

(6) 千葉 智子, 大内 真一, 藤島 実, 鳳 紘一郎: "プログラマブル量子計算エミュレータ", 電子情報通信学会 回路とシステム研究会 (2001).

(7) 千葉 智子, 大内 真一, 藤島 実, 鳳 紘一郎: "プログラマブル量子計算エミュレータ", 電子情報通信学会第 62 回総合大会講演論文集情報・システム 1 (2001) p. 66.

(8) 鳳 紘一郎, 藤島 実, 大内 真一, 千葉智子: "シリコン集積回路による量子コンピューティングへの挑戦", 第 13 回電気電子情報学術振興財団ワークショップ, 発展する電子・光子機能制御研究 講演資料集 (2001) p. 1.

(9) 鈴木康文, 斉藤康祐, 大内真一, 藤島実, 鳳 紘一郎: "16-qubit ディジタル量子回路プロセッサ", 第 49 回応用物理学関係連合会講演会 (2002) 発表予定.

(10) 小野内雅文, 吉本晴洋, 大内真一, 藤島実, 鳳 紘一郎: "14-qubit 量子回路プロセッサ", 第 49 回応用物理学関係連合会講演会 (2002) 発表予定.

(11) 杉浦学, 大内真一, 藤島実, 鳳 紘一郎: "SPMD を用いる量子回路プロセッサ", 第 49 回応用物理学関係連合会講演会 (2002) 発表予定.

参考文献

[1] C. H. Bennett, "logical Reversiblity of Computation", IBM J. Research and Development **21**, 525 (1973).

[2] R. P. Feynman, "Simulating Physics with Computers", Int. J. Theoretical Physics **21**, 467 (1982).

[3] R. P. Feynman, "Quantum Mechanical Computers", Optics News **11**, 11 (1985).

[4] P. Benioff, "The Computer as a Physical System: A Microscopic Quantum Mechanical Hamiltonian Model of Computers as Tepresented by Turing Machines", J. Statistical Phys. **22**, 563 (1980).

[5] D. Deutch, "Quantum Theory, the Church-Turing Principle, and the Universal Quantum Computer", Proc. R. Soc. Lond. A **400**, 97 (1985).

[6] A. Yao, "Quantum Circuit Complexity", in *Proc. 34th IEEE Symp. on Foundation of Computer Science*, Los Alamitos, CA, USA, IEEE Computer Society Press, 352 (1993).

[7] A. Barenco,, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. W. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, "Elementary Gates for Qautum Computation", Phys. Rev. A **52**, 52 (1995).

[8] D. Deutch, A. Barenco, and A. Ekert, "Universality in Quantum Computation", Proc. R. Soc. Lond. A **449**, 669 (1995).

[9] A. Lenstra and H. Lenstra, "The Number Field Sieve", in *Proc. 22nd ACM Symp. Theory of Computing*, 564 (1990).

[10] P. W. Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring", in *Proc. 35th Ann. Symp. on Fundations of Computer Science*, IEEE Press, 124 (1994).

[11] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Qunatum Computer", SIAM J. Computing **26**, 1484 (1997), quant-ph/9508027.

[12] L. K. Grover, "A Fast Quantum Mechanical Algorithm for Database Search", in *Proc. STOC 1996*, 212 (1996), qunat-ph/9605043.

[13] L. K. Grover, "Quantum Mechanics Helps in Searching for a Needle in a Haystack", Phys. Rev. Lett. **79**, 325 (1997).

[14] M. BOyer, G. Brassard, P. Hoyer, and A. Tapp, "Tight Bounds on Quantum Searching", in *Proc. PhysComp 1996*, quant-ph/9605034.

[15] 細谷 暁夫、「量子コンピュータの基礎」、サイエンス社 (1999).

[16] J. Cirac and P. Zoller, "Quantum Computations with Cold Trapped Ions", Phys. Rev. Lett. **74**, 4091 (1995).

[17] I. L. Chuang and Y. Yamamoto, "A Simple Quantum Computer", Phys. Rev. A **52**, 3554 (1995).

[18] I. L. Chuang, L. M. K. Vandersypen, X. Zhou, D. W. Leung and S. Lloyd, "Experimantal Realization of a Quantum Algorithm", Nature **393**, 143 (1998).

[19] T. H. Oosterkamp, T. Fujisawa, W. G. van der Wiel, K. Ishibashi, T. V. Hijiman, S. Tarucha, and L. P. Kouwenhoven, "Micorwave Spectroscopy of a Quantum-Dot Molecule", Nature **395**, 873 (1998).

[20] 藤澤 利正、「量子ドット分子の物理と量子計算」、科学 **69**, 546 (2001).

[21] B. E. Kane, "A Silicon-Based Nuclear Spin Quantum Computer", Nature **393**, 133 (1998).

[22] Y. Nakamura, Yu. A. Pashkin, and J. S. Tsai, "Coherent Control of Macroscopic Quantum States in a Single-Cooper-Pair Box", Nature **398**, 786 (1999).

[23] D. V. Averin, "Solid-State Qubits under Control", Nature **398**, 748 (1999).

[24] F. Yamaguchi and Y. Yamamoto, "Crystal Lattice Quantum Computer", Appl. Phys. A **68**, 1 (1999).

[25] T. D. Ladd, J. R. Goldman, F. Yamaguchi, Y. Yamamoto, E. Abe and K. M. Itoh, "An All Silicon Quantum Computer", quant-ph/0109039.

[26] J. Niwa, "General-Purpose Fast Simulator for Quantum Computing", 第5回量子情報技術研究会資料 (電子情報通信学会 量子情報技術時限研究専門委員会, 厚木, 2001) 123.

[27] M. D. Price, S. S. Somaroo, C. H. Tseng, J. C. Gore, A. F. Fahmy, T. F. Havel and D. G. Cory, "Construction and Implementation of NMR Quantum Logic Gates for Two Spin Systems", J. Magnetic Resonance **140**, 371 (1999).

[28] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, R. Cleve and I. L. Chuang, "Experimental Realization of an Order-Finding Algorithm with an NMR quantum Computer", Phys. Rev. Lett. **85**, 5422 (2000).

[29] W. S. Warren, Science **277**, 1688 (1997).

[30] *International Technology Roadmap for Semiconductors, San Jose, California, 1999* (Semiconductor Industry Association, 1999), http://public.itrs.net/Files/2000UpdateFinal/2kUdFinal.htm

[31] J. Makino, T. Fukushige and M. Koga, "A 1.349 Tflops Simulation of Black Holes in a Galactic Center on GRAPE-6", in *Proc. IEEE/ACM SC2000 Conf.,* Dollas, Texas USA, Sponserd by the IEEE Computer Society and ACM SIGARCH, (2000), http://grape.astron.s.u-tokyo.ac.jp/.

[32] J. Makino, "Grape Project: Special-Purpose Computers for Many-Body Simulations", Computer Physics Communications **139**, 45 (2001)

謝辞

本研究を進める上で, 大勢の方々に御指導, 御協力を頂きました.

まず第一に筆者の指導教官である鳳紘一郎教授並びに藤島実助教授の両先生に心より感謝致します. 本研究は, 集積回路と量子コンピュータの境界領域に集積回路の立場から取り組むという斬新な研究であり, 大変難しいテーマでありましたが, 適切な御指導を賜わり, また励ましを頂き, 本論文の提出に導いて頂きました.

また, 本研究は科学技術振興事業団 菅野卓雄教授のプロジェクトに参加して行なわれたものでありましたが, 同教授をはじめ, 本研究の属する鳳チームの柴田直教授, 平本俊郎助教授の諸先生方ならびに経済産業省産業総合研究所の伊藤順司氏に大変貴重な討論とコメントを頂きました.

本研究は, 鳳・藤島研究室の千葉智子さん (現松下電器産業株式会社), 並びに杉浦学, 小野内雅文, 斉藤康祐, 安藤雅享, 鈴木康文, 吉本晴洋諸氏との共同作業によって行なわれたものです. 計算手法の立案や回路の試作に関する膨大な討論と共同で行なった作業は, いうまでもなく本論文の根幹をなしています.

この他, 鳳・藤島研究室の北沢清子助手をはじめ, 多くの研究室の方々, 並びに科学技術振興事業団の方々に支えて頂きました. また, 研究室外部の方々に多数の討論とコメントを頂きました.

本論文を提出するにあたって, 全ての方々に心より感謝致します.