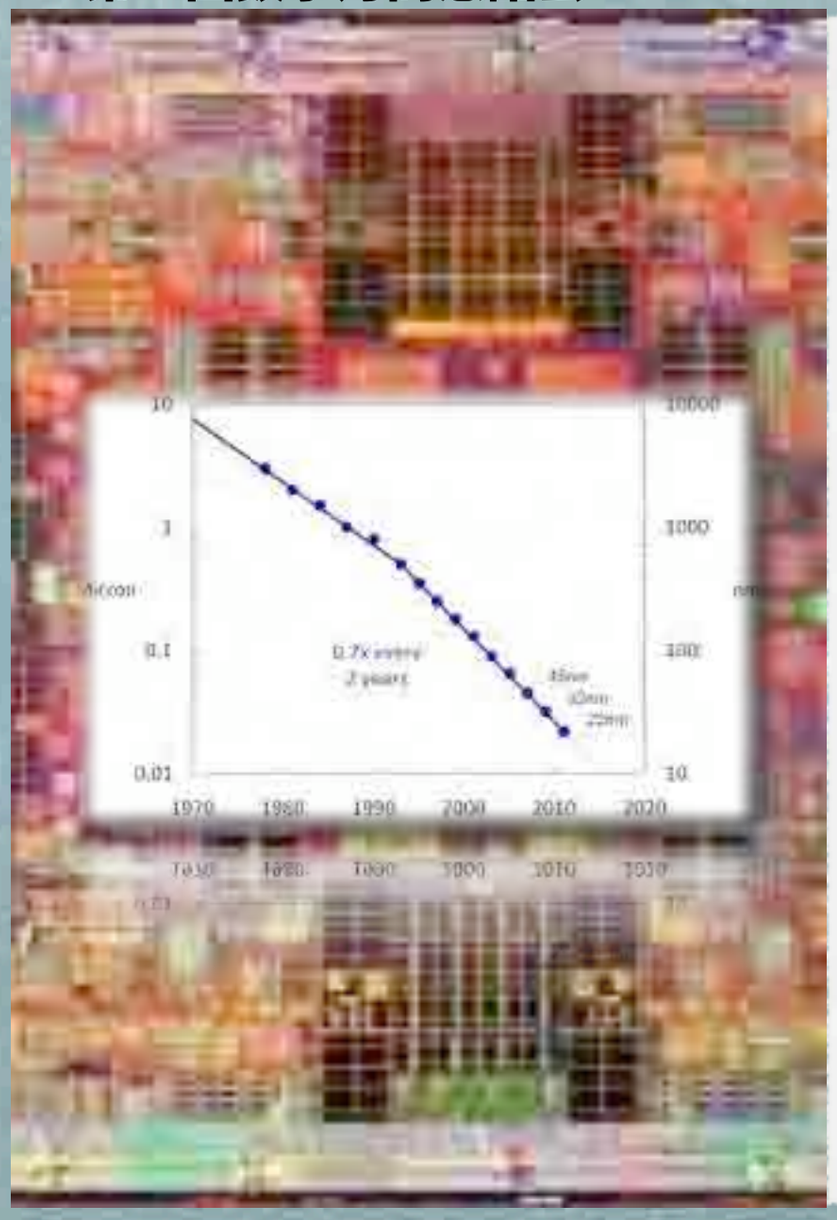
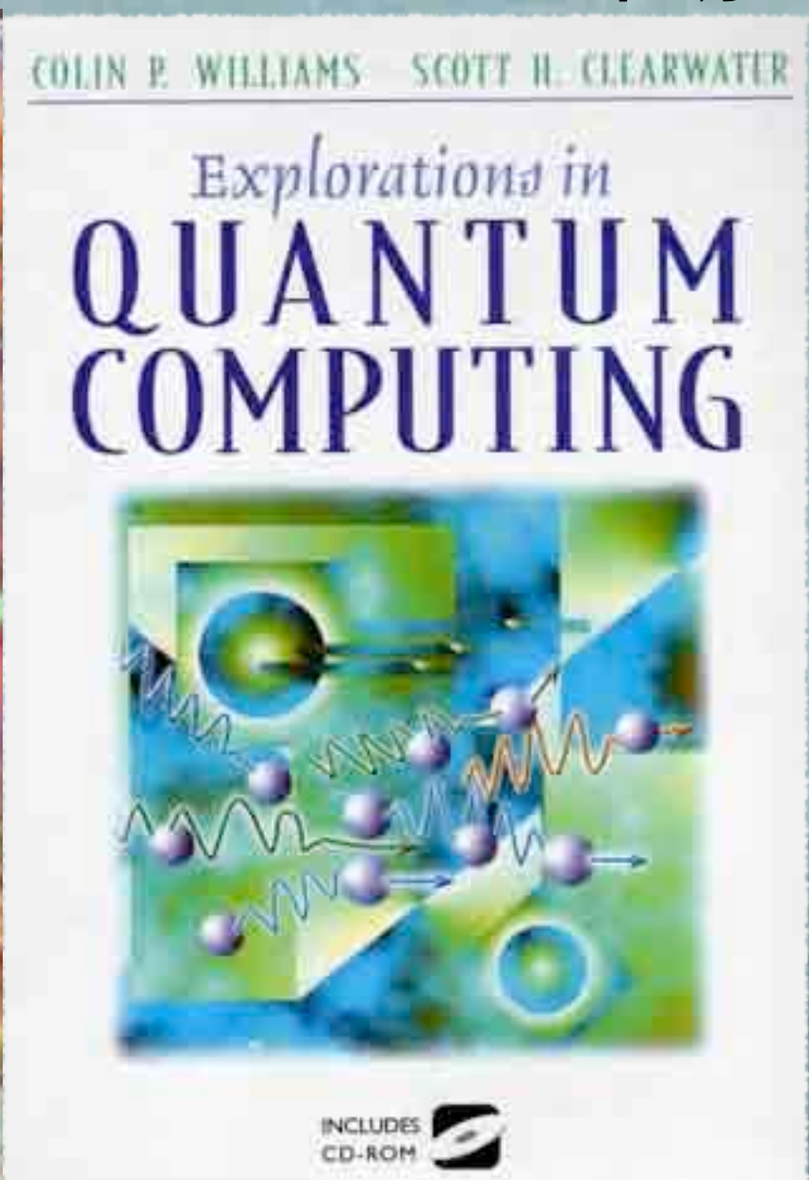
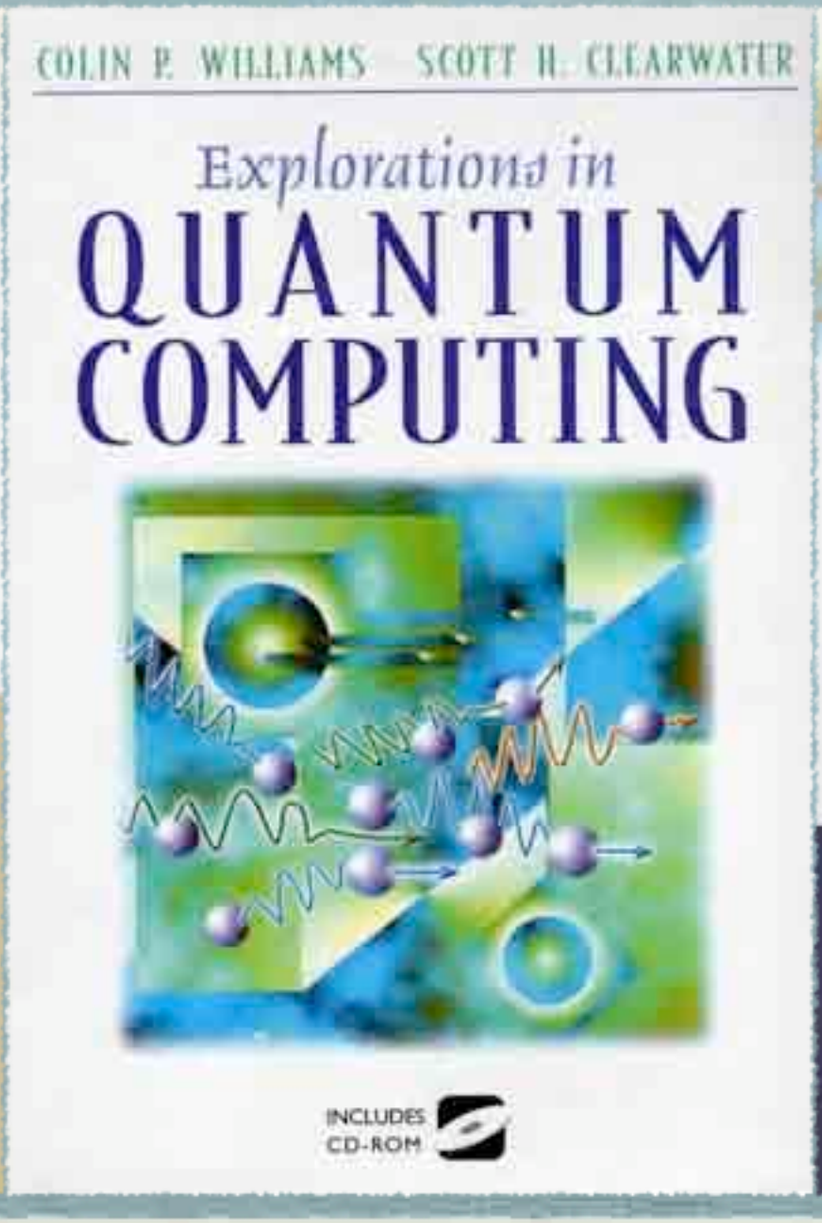
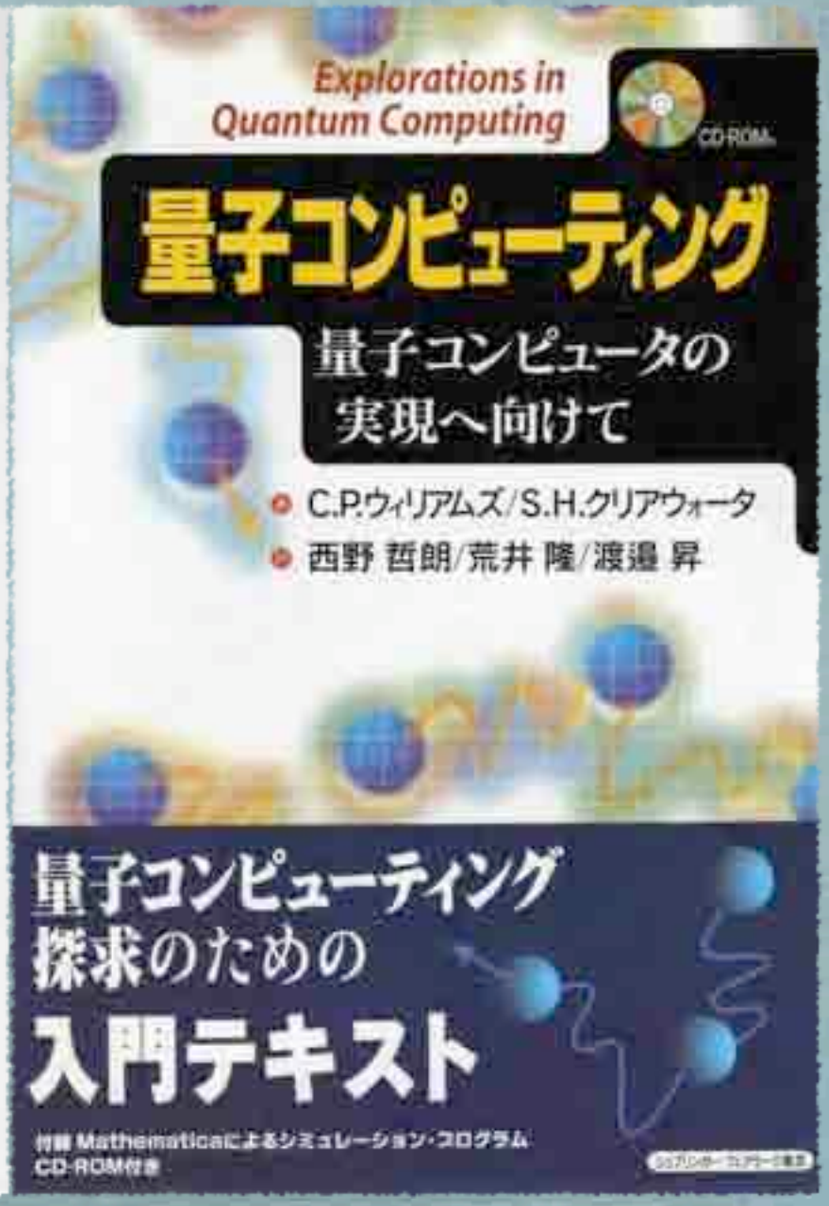




量子コンピューティングの考え方

第７回数学月間懇話会　2011.7.22　東大駒場キャンパス（数理科学研究科棟056教室）
防衛大学校　応用物理学科　荒井　隆





量子コンピューティングは数学，情報，物理学にまたがった**学際的な分野**

それぞれの学問分野を時間的**広範囲にわたって知っている**必要がある

今日の話の流れ

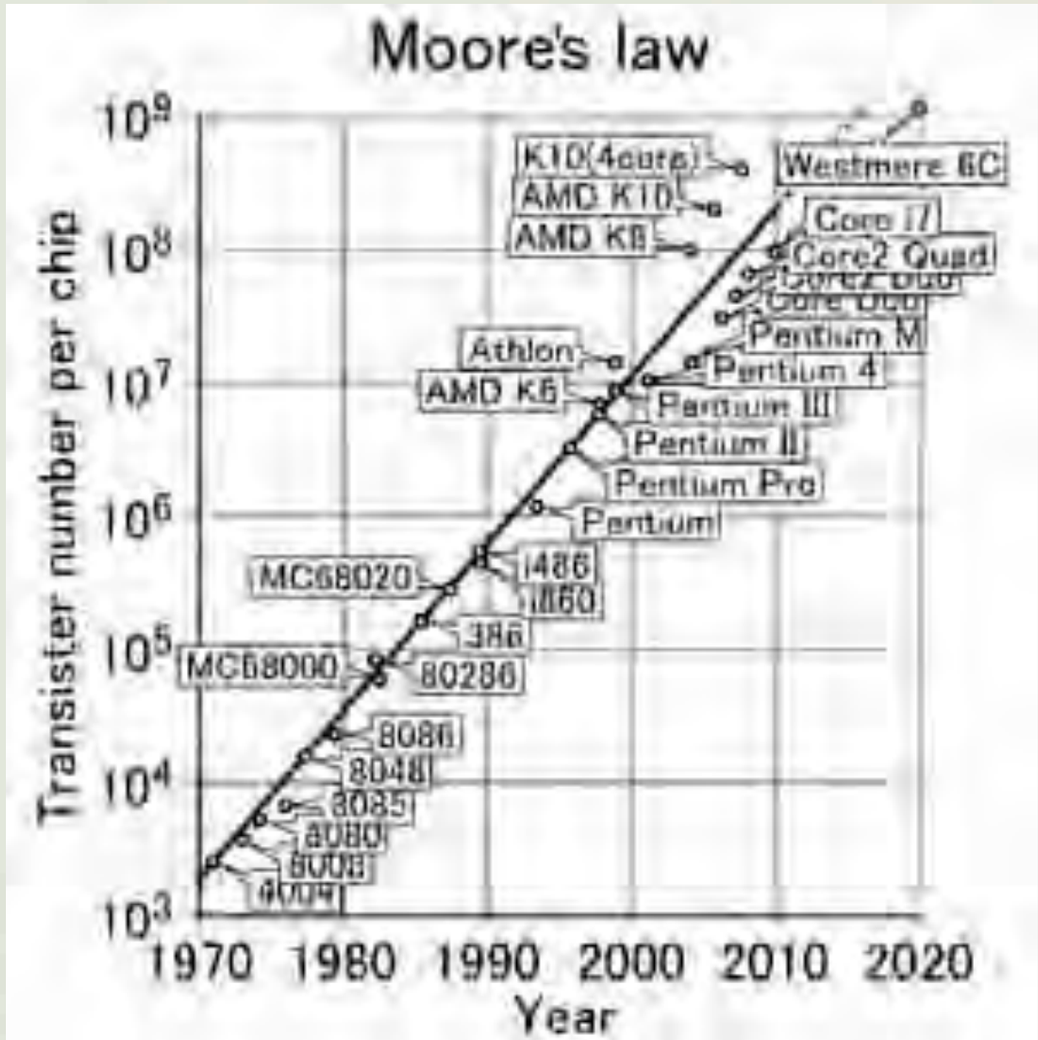
- ムーアの法則：２０２０年にはどうなるのか？
- ブール代数：古典となった論理ゲート
- 量子論理ゲート
- 量子干渉
- 量子並列
- いくつかの量子アルゴリズムの簡単な紹介

Mooreの法則

- **1965年**にインテル社の共同創業者であるゴードン・ムーアが論文上に示した予測，**「決まった大きさのチップに組み込まれるトランジスターの数（性能）は約２年で２倍になるだろう」**は４５年後でもその予測から外れず，現在でも生き続けている。

式で表現すれば、n年後の倍率 p は、

$$p = 2^{(n / 2)}$$

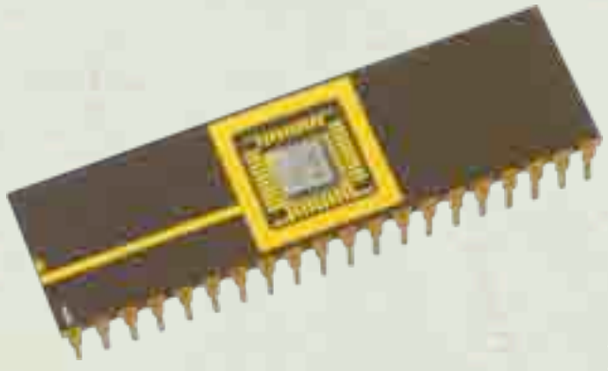


Mooreの法則

- MIPS（１秒間に行なわれる演算回数の百万倍）という量でコンピュータの CPU の性能を評価するが、これは、 CPUを構成しているトランジスター数に比例するし，CPUのクロック数にも比例する。

Texas Instruments, SN7400,
8 transistors, 1966

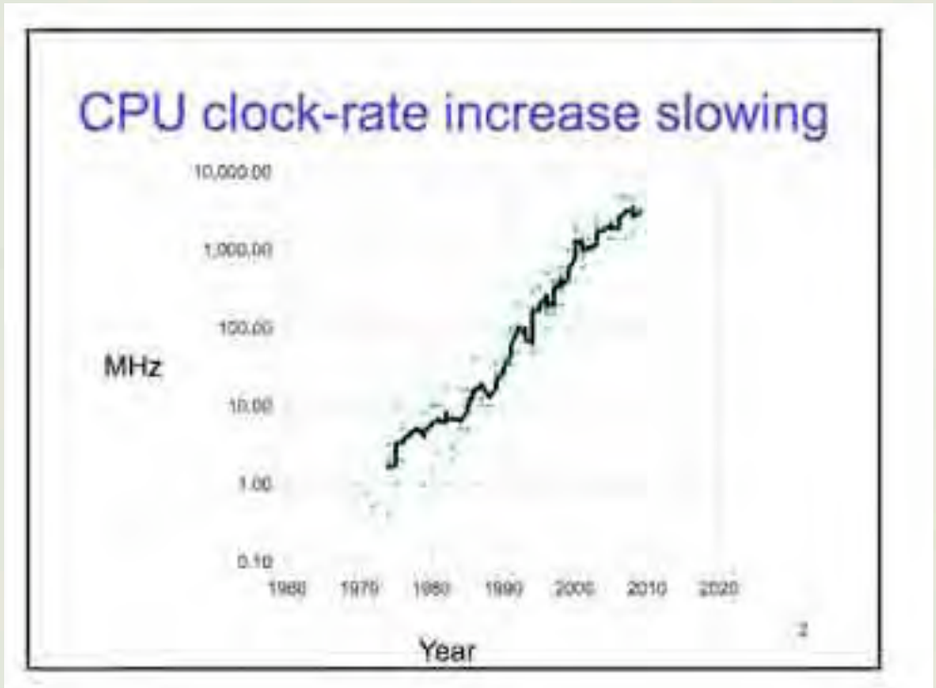
トランジスタ



Motorola MC6800 1MHz 8bit
6800 transistors,1974

Intel Clarkdale Core i5, 3.3GHz 64bit 2 core + GPU

383,000,000 transistors,2010



Mooreの法則

ムーアの法則はさらに１トランジスタあたりの製作コストと消費電力は同じように減少している
よってウェハーあたりのコスト、チップあたりの消費電力は一定のまま変わらないというものであるという予測も含んでいる

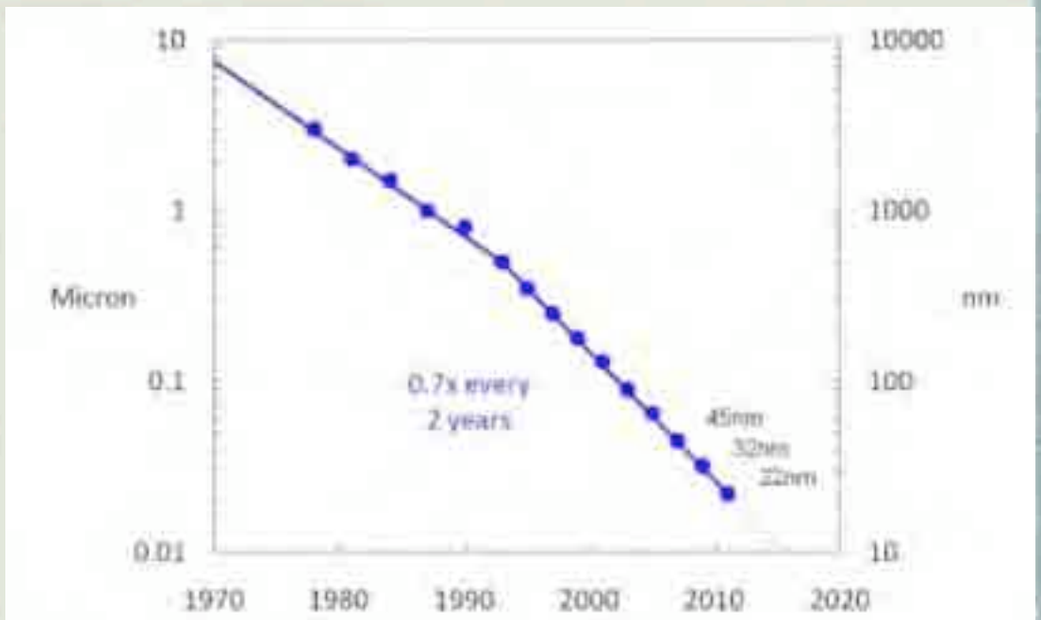
Mooreの法則による２０２０年問題

ムーアの法則が2020年まで継続することにより、
将来たった**原子３個分しかない幅のトランジスタ**になる。

また、エネルギーE は振動数νに比例する（ E = hν ）ため
動作**クロック数は消費電力に比例し、**
ゆえにクロックの上昇は発熱上昇をもたらす。

ある程度以上の高速化されたCPUでは発熱量の増加が上回り、
放熱問題に直面して、動作クロックの高速化は現実的でなくなる

4.3GHz以上の速度で高信頼性のCPUを提供するのはほとんど無理。



Caption: Transistor dimensions scale to improve performance, reduce power and reduce cost per transistor.

これは2020年には**サイズ，高速化の両面で実際上の物理的限界が来る**ことを意味している。

現在，アメリカ合衆国の電力消費の５％がコンピュータによって消費されている。ムーアの予測によれば2020年には40GHz相当で動作するコンピュータ（総電力４０W）となり、今の電力の１０倍以上にコンピュータに電力を費やすことになると見積もられている。

Mooreの法則と量子コンピュータの必要性

節電ブームで，パラダイムシフトが更に加速するか？

これまでこの法則は明らかに克服できないように見える障害にしばしば直面したが、
すぐにこれらを乗り越えていった。

例１）バイポーラートランジスタからMOSFETトランジスタへ

例２）計算能力を向上させる方法は、単一の命令ストリームを1つの演算部で可能な限り早く処理するだけとは限らず、遅い動作クロックであっても複数の演算部で並列的に処理することでも計算能力を向上できる。（CPUのCore分割＋GPU）

考え方

CMOSに取って代わる新しい計算機構と論理デバイスが必要になる。

そのデバイスは，原子１個あたり１演算を行なえるような量子力学に基づいた原理で働き，
非常に高い並列度で計算可能な仕組みをとるはずである。
そして発熱による素子の劣化を防ぐ仕組みも持っているはずである。

ブール代数：古典論理ゲート

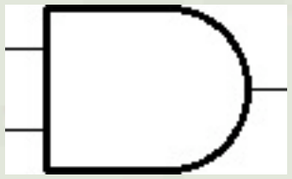
トランジスターの役割

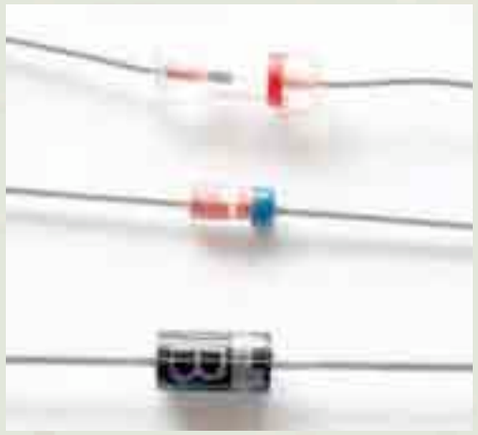
- コンピュータの動作原理は，ブール代数にある。
- ブール代数は論理回路で物理的に実現可能である。
- 否定，論理和、論理積，排他的論理和等の論理回路から加算器を作成し，ラッチ，フリップフロップなど順序回路を使ってカウンタ，メモリ，レジスタを作成し、これらを組み合わせて数列の和をもとめるという簡単な計算をする回路を作成できる。（コンピュータの基礎）

論理積の物理的実現

- 論理積（AND）は、ダイオードだけで次のように作成できる

（物理的実現）



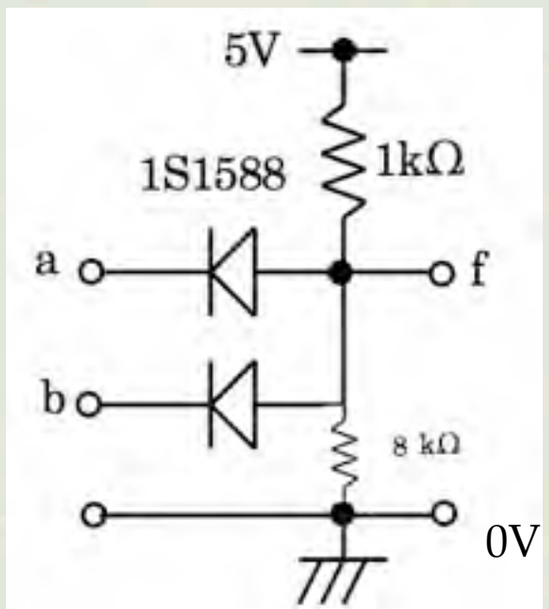


$$a \otimes b = f$$

論理積の真理表

入力 a	入力 b	出力 f
0	0	0
0	1	0
1	0	0
1	1	1

0V	5V
0	1



- 否定（NOT/インバーター）

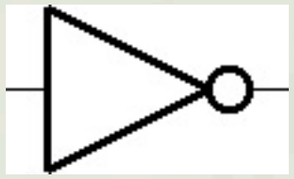
否定の真理表

入力１	出力１
0	1
1	0

ダオードロジックでは実現できない

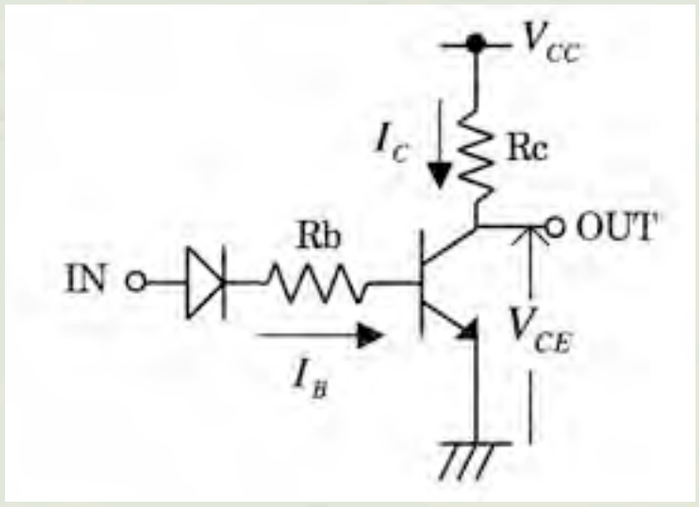
否定の物理的実現

- 否定（NOT/インバーター）　$\overline{x}$



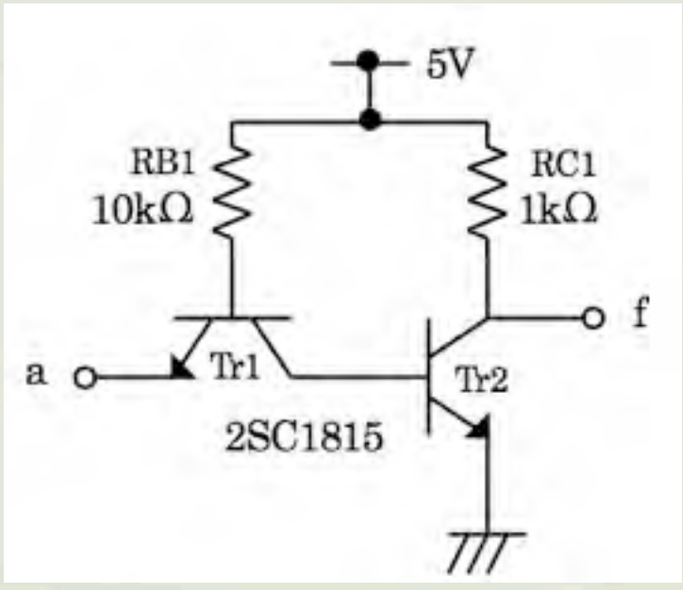
否定の真理表

入力１	出力１
0	1
1	0



ダイオード–トランジスタ ロジック

トランジスタ

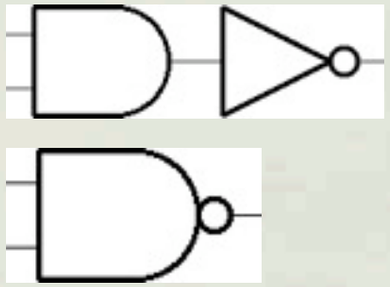


トランジスタ–トランジスタ ロジック

防衛大学校　応用物理学科　　荒井　隆

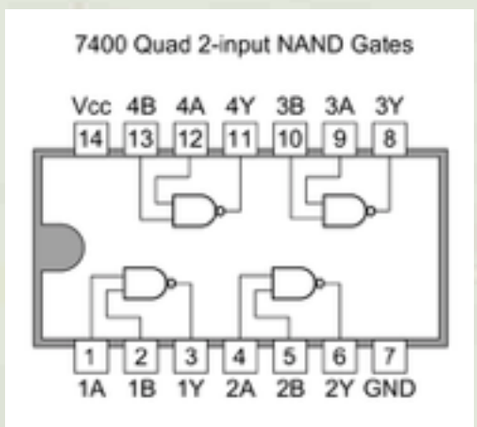
NAND回路　万能ゲート

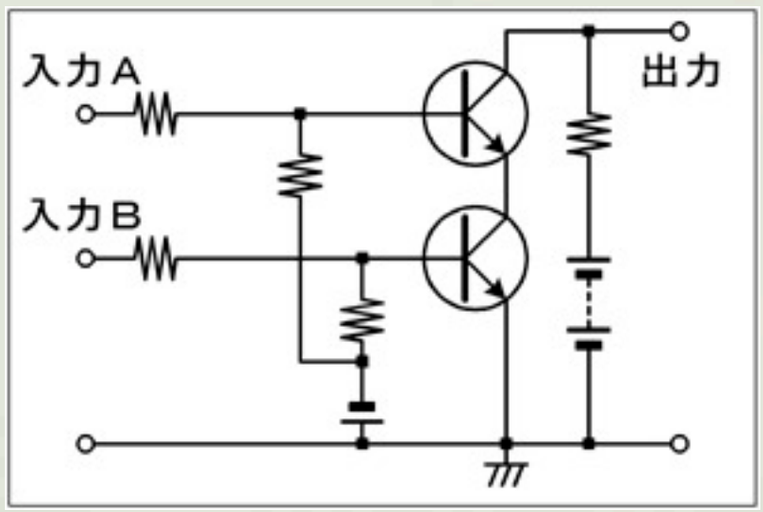
- 論理積の否定（NAND）　$\overline{a \otimes b}$

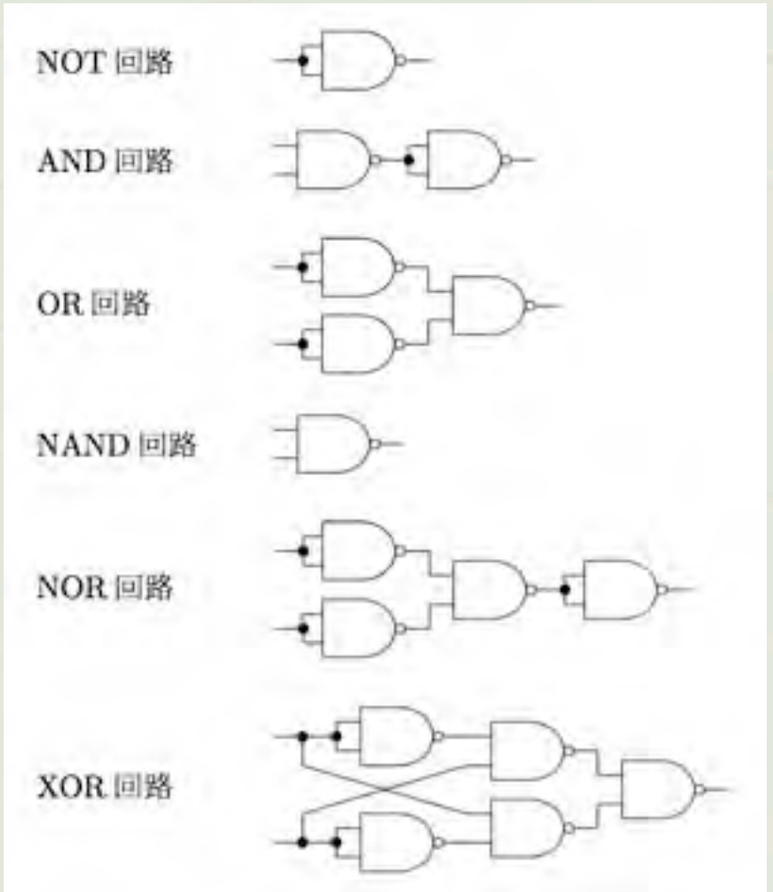


NANDの真理表

入力 A	入力 B	出力
0	0	1
0	1	1
1	0	1
1	1	0







XORの真理表

入力 A	入力 B	出力
0	0	0
0	1	1
1	0	1
1	1	0

加算器の例

コンピュータでの計算（コンピュタ上のソフトの全ての動作）は２進数の演算で行なわれる。

加算（足し算）は最も基本的な動作である

２進数６桁の足し算回路

１０進数：　　23　＋　13　＝　36

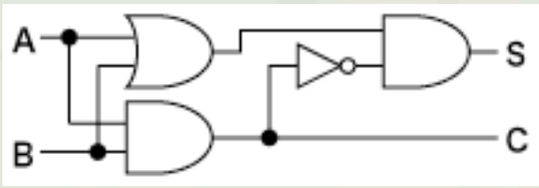
２進数：　　010111　＋　001101　＝　100100

A　　B　　S

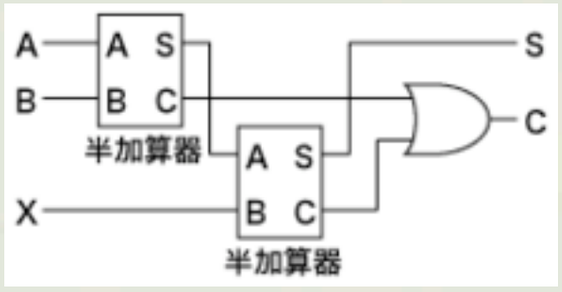
半加算器の真理表

入力 A	入力 B	出力S	出力C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

和　繰り上げ

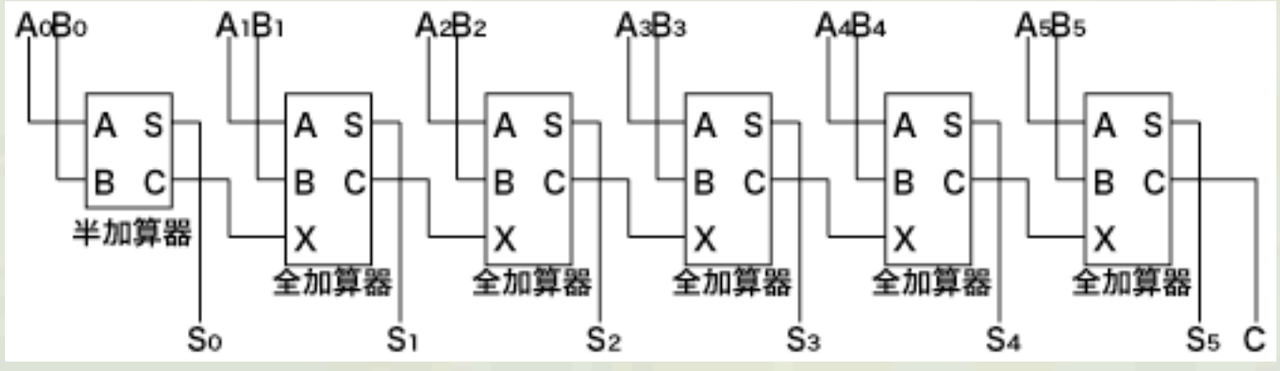


半加算器



全加算器

A,Bが入力値



Sが足した結果の値

防衛大学校　応用物理学科　　荒井　隆

Mooreの法則からの研究成果

可逆計算可能なコンピュータは発熱ゼロよって電力消費ゼロ

- **「論理的に可逆」と「熱力学的に可逆」の間のいくつかの命題が証明された**
  - NOTゲートは１入力１出力であり，論理的に可逆である。
  - ANDゲートは論理的に不可逆である。（２入力１出力）
  - 論理的に可逆なゲートは熱力学的にも可逆でありよって発熱しない（C. Bennett, Scientific American,1987）。
  - 量子力学的な考えに基づいた数学モデルを使うと論理的にエネルギーを散逸しない計算機を設計できる（P. Benioff, Phys.Rev.Lett.,1982）。

任意の有限な実現可能な物理系は有限の回数で演算する万能モデル・コンピュータによって完全に模倣できる（Deutschの提言）。

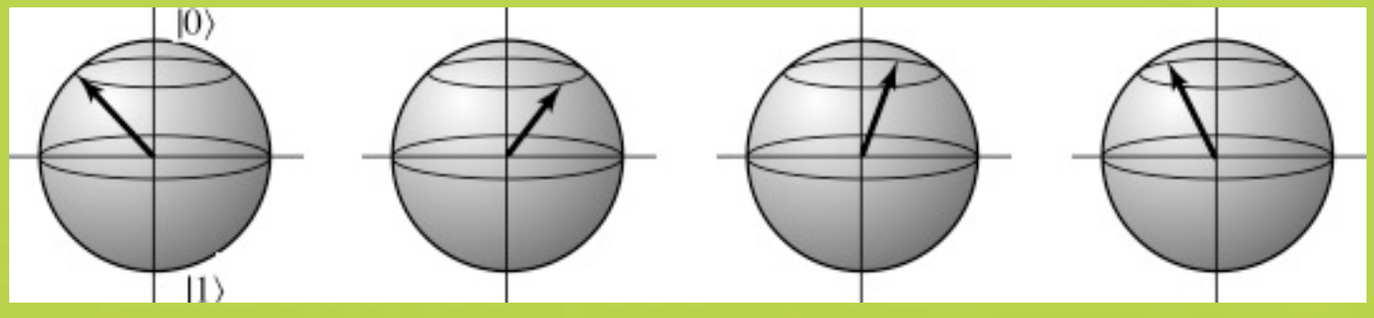
論理の量子化

- 古典物理学を量子化したのと全く同じ方法で**「計算の基本」や「計算すること」を量子化したもの**が量子コンピューティングと呼ばれているものである。

- 「真」、「偽」あるいは「１」、「０」を、｜１＞という状態と｜０＞という状態として量子力学に基づいて定義する。

状態：複素数の列ベクトル

- 全ての計算（状態）を｜１＞という状態と｜０＞という**状態を重ね合わせた状態**として扱う。



複素線形ベクトル空間

量子力学の公理

「量子コンピュータの基礎」by 細谷暁夫では,
A.Peres ("Quantum Theory:Concept and Methods")を引用して公理を述べてある

1. 重ね合わせの原理

状態｜a＞と｜b＞が可能な状態であれば，αとβを複素数として，その重ね合わせα｜a＞+β｜b＞も可能な状態である。

2.Schrödeinger方程式

状態の時間発展はシュレーディンガー方程式にしたがって，ユニタリな発展をする。

｜ψ＞　--->　U｜ψ＞

$$U(t) = e^{-iHt/\hbar}$$

時間発展の演算子 U はユニタリ演算子である。

3.波束の収縮と確率解釈

a、bをある物理量Qの固有値とし，それらの固有状態をおのおの｜a＞と｜b＞とするとき,

重ね合わせの状態α｜a＞＋β｜b＞にあるときに物理量Qを観測すると状態は,

波束の収縮をして状態｜a＞または｜b＞のどちらかの状態に移る。

各々の確率は，対応する確率振幅の2乗，すなわち，｜α｜^{2}及び｜β｜^{2}で与えられる.

状態を確率的に解釈する

- 任意の状態は「０」と「１」の重ね合わせ

$$\psi = \omega_0 \mid 0\rangle + \omega_1 \mid 1\rangle$$

２つの状態を次のように直交する固有状態として決めておく

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$|1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

- ω_0とω_1はそれぞれ、状態｜０＞と状態｜１＞の確率振幅

$$|\omega_0|^2 + |\omega_1|^2 = 1$$

- 確率振幅は，全ての２乗和が１になるように規格化する。

$$\psi = \omega_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \omega_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \omega_0 \\ \omega_1 \end{pmatrix}$$

- 観測をすると，確率｜ω_0｜^{2}で「１」となり、確率｜ω_1｜^{2}で「０」となる。

量子論理ゲート

- NOTは量子論理ゲートでは、ユニタリー行列　NOT= $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ で表される。
- 状態 $|0>$ は

$$\mathrm{NOT}|0> = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1>$$

- と状態 $|1>$ に遷移する。
- さらに、NOTを作用させると，　状態 $|0>$ に戻る

$$\mathrm{NOT}|1> = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = |0>$$

量子論理ゲート

これまでの論理回路では不可能な演算　$\sqrt{NOT}$

- 命題「ある等しい論理回路を2つ続けて作用させて，NOTと等価な回路を作ることができるか？」
- 現在の論理ゲートを使っては，上の命題は「偽」である。
  （ブール代数の性質にはない）
- 量子論理ゲートでは，「真」となる。

$$\mathrm{NOT}|1> = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |0>$$

- これを満たす，複素数の組，a,b,c,d は規格化条件を使えば簡単に求まり，例えば，

$$U = \begin{pmatrix} \frac{i+1}{2} & \frac{i-1}{2} \\ \frac{i-1}{2} & \frac{i+1}{2} \end{pmatrix} = \sqrt{NOT}$$

- となる。

AND 量子論理ゲート

- ANDゲートは２入力１出力であるため、可逆ではあり得ない。量子論理ゲートでは、入力状態をそのまま維持して出力し，それとは別に演算結果を３番目の信号として出力するような３入力３出力ゲートとして表現する必要がある。

- 真理表は右図のようになり、入力Cが「0」に対する出力CにANDの演算結果が与えられる。

量子ANDの真理表

入力 A	入力 B	入力C	出力A	出力B	出力C
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

この真理表を実現させるためのユニタリー行列U_{AND}は

$$U_{AND}=\begin{pmatrix}1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1\\0&0&0&0&0&0&1&0\end{pmatrix}$$

$$U_{AND}|000>=|000>$$
$$U_{AND}|001>=|001>$$
$$U_{AND}|010>=|010>$$
$$U_{AND}|011>=|011>$$
$$U_{AND}|100>=|100>$$
$$U_{AND}|101>=|101>$$
$$U_{AND}|110>=|111>$$
$$U_{AND}|111>=|110>$$

状態ベクトルは８要素

$$|000>=\begin{pmatrix}1\\0\\0\\0\\0\\0\\0\\0\end{pmatrix}\quad |001>=\begin{pmatrix}0\\1\\0\\0\\0\\0\\0\\0\end{pmatrix}$$

XOR 量子論理ゲート

- XORゲートは２入力１出力であるが、演算の性質から一方の入力さえ保持しておけば可逆となる。量子論理ゲートでは、１入力状態をそのまま維持して出力し，演算結果を２番目の出力信号とするような２入力２出力ゲートとして表現できる。

量子XORの真理表

入力 A	入力B	出力 A	出力C
0	0	0	0
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

- 真理表は右図のようになり、
- 出力CにXORの演算結果が与えられる。

この真理表を実現させるためのユニタリー行列U_{XOR}は

$$U_{XOR}=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}$$

$$U_{XOR}\,|00>=|00>$$
$$U_{XOR}\,|01>=|01>$$
$$U_{XOR}\,|10>=|01>$$
$$U_{XOR}\,|11>=|00>$$

状態ベクトルは4要素

$$|00>=\begin{pmatrix}1\\0\\0\\0\end{pmatrix}\qquad |01>=\begin{pmatrix}0\\1\\0\\0\end{pmatrix}$$

量子干渉

量子干渉は量子現象の典型的な性質である

- 目的を達成するのに異なる２つ以上の方法がある場合、必ず量子干渉が起こる。

- 例：ある病気の治療をするのに、２つの方法A、Bがあり，方法Aを選ぶと確率7/20で成功し、方法Bを用いると確率4/5で成功する。一般的な手順はまず、方法AかBを選択する。Aを選択する確率を p とすると方法Bを選択する確率は1–p である。次ぎに，選択した方法で治療を行なうことになる。数学的には成功する確率は，
$$\frac{7}{20}p+\frac{4}{5}(1-p)=\frac{4}{5}-\frac{9}{20}p$$

- 量子論理では，まず、Aを選択する確率 p を $p=\cos^2\Phi$ と変数を変えておき、成功する状態を｜０＞、失敗する状態を｜１＞と表すことにする。また、状態Aを｜A＞,状態Bを｜B＞と表しておく。量子論理では重ね合わせの状態をつくるので

量子干渉

量子干渉は量子現象の典型的な性質である

- 方法Aで治療が成功する確率は 7/20、 方法Bで治療が成功する確率は 4/5 なので

$$|\psi_A>=\sqrt{\frac{7}{20}}|0>+\sqrt{\frac{13}{20}}|1>$$

$$|\psi_B>=\sqrt{\frac{16}{20}}|0>+\sqrt{\frac{4}{20}}|1>$$

- 成功する全確率は，二つの状態の重ね合わせ

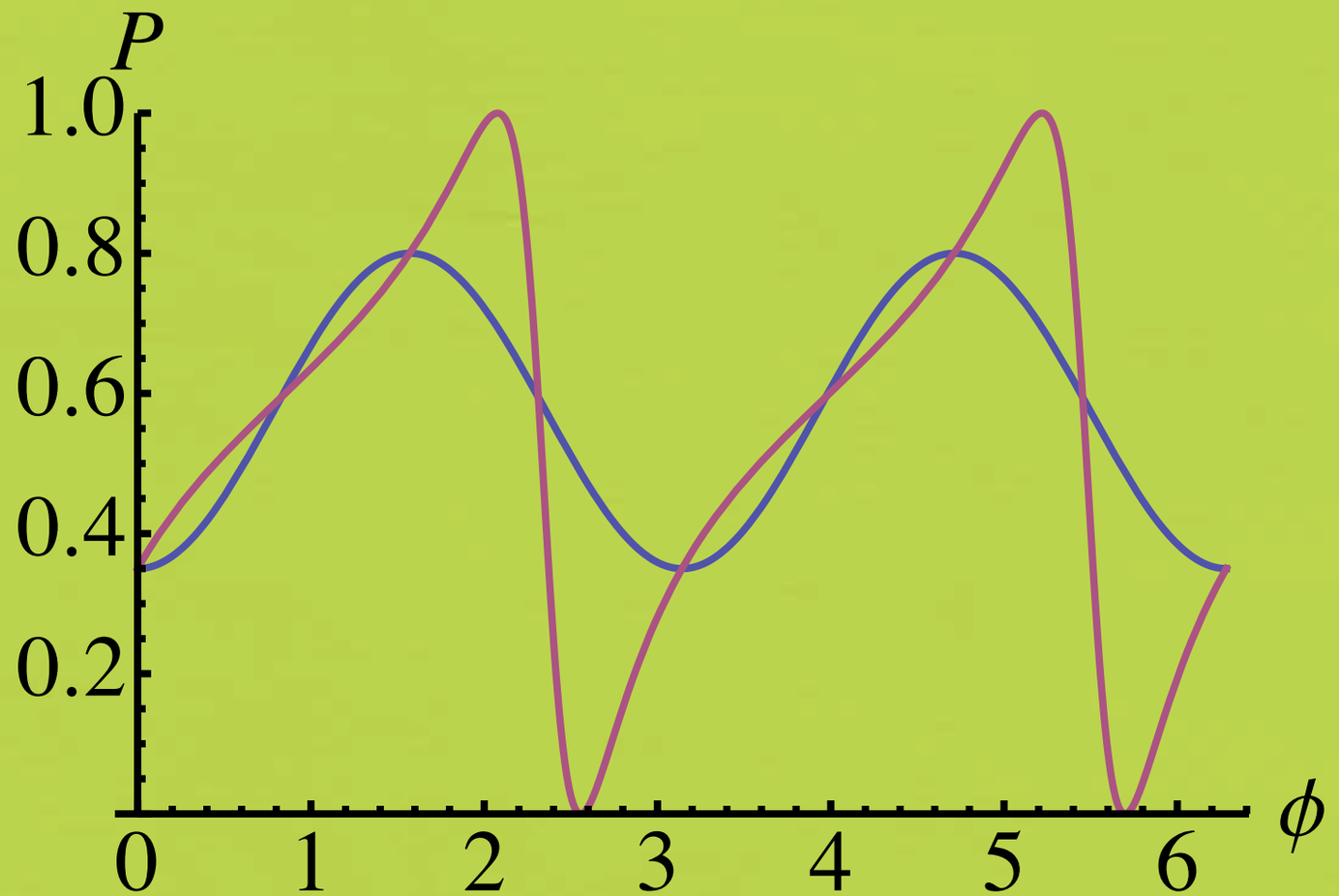
$$|\psi>=\cos\phi|\psi_A>+\sin\phi|\psi_B>$$
$$=\omega_0|0>+\omega_1|1>$$

- で表される。実際に方法A が選択される確率は $p=\cos^2\Phi$で方法Bが選択される確率は1–pとなっている。
- したがって、 $\dfrac{|\omega_0|^2}{|\omega_0|^2+|\omega_1|^2}$ として、成功確率を求めると

$$\frac{23-9\cos(2\phi)+8\sqrt{7}\sin(2\phi)}{40+4(2\sqrt{7}+\sqrt{13})\sin(2\phi)}$$

量子干渉

- 治療が成功する確率



青線：古典論理

赤線：量子論理

量子論理ではある特定な確率で方法Aを選択すれば治療が１００％成功する、あるいは１００％失敗することが起きる。

量子並列

- 簡単な計算　１＋１＝２　を量子論理ゲートで行なうことを考える
- ２進数１桁の足し算（＋繰り上げ）論理ゲート（半加算器）
- 和は XOR、繰り上げはANDのゲートが必要となる。

$$U_{HADD} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

- これを行なわせるにはU_{XOR}とU_{AND}の内積をとり、
- 真理表は右のようになり，
- 出力Aはそのまま，出力BはXOR（和）、
- 出力CはAND（繰り上げ）となっている。

量子HADDの真理表

入力 A	入力 B	入力C	出力A	出力B	出力C
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

すなわち，２進数１桁の全ての足し算が同時に得られる。

量子並列

- 現在のコンピュータでは，２x３＝６は１回の演算で実行できる。
- 量子コンピュータでは，１０進数１桁同士の乗算を行う回路をまず作成することになり、１０進数９は２進数で$(1001)_2$であるので２進数４桁同士の乗算回路を作成することになる。これには最低２５６x２５６の大きさのユニタリ行列の演算を行う必要があり，その結果，１回の演算で１０進数で２２５までの九九の表が完成することになる。

役に立つのか？

量子干渉と量子並列の性質を使ってこれまでにないパフォーマンスを出す

いくつかの量子アルゴリズム

- Groverのデータベース検索アルゴリズム
- 暗号解読のShorの因数分解アルゴリズム

Groverのデータベース検索アルゴリズム

ある病気の一因となっている遺伝子を解読したヒトゲノムのデータベースから検索する

- N個のデータベース情報の中から欲しい１個を検索する場合（必ずあるという前提で），データベースの並び順で逐次検索を行なうと，欲しい情報が最初に見つかれば１回の検索で終わるし，最後の１つにあれば，N–1回の検索が必要となり，情報がある場所が等しい確率で存在するならば，平均の検索回数は１からN–1までの和をN–1で割った，N/2回となる。Groverのデータベース検索量子アルゴリズムはこれを√N回で行なうことができる。
- ２進数で000～111の N=8個の中から，１つ（010）を検索することを考える
- 調べるデータベースの状態はそれぞれの状態|000>、|001>、|010>、|011>、|100>、|101>、|110>、|111>が等しい確率振幅で重ね合わせの状態 |s> にあるとする。

$$|s>=\frac{1}{\sqrt{8}}\left(|000>+|001>+|010>+|011>+|100>+|101>+|110>+|111>\right)$$

- 状態を列ベクトルで表すと，それぞれ以下のようになる

$$|000>=\begin{pmatrix}1\\0\\0\\0\\0\\0\\0\\0\end{pmatrix}\quad |001>=\begin{pmatrix}0\\1\\0\\0\\0\\0\\0\\0\end{pmatrix}\quad |010>=\begin{pmatrix}0\\0\\1\\0\\0\\0\\0\\0\end{pmatrix}\quad |011>=\begin{pmatrix}0\\0\\0\\1\\0\\0\\0\\0\end{pmatrix}\quad |100>=\begin{pmatrix}0\\0\\0\\0\\1\\0\\0\\0\end{pmatrix}\quad |101>=\begin{pmatrix}0\\0\\0\\0\\0\\1\\0\\0\end{pmatrix}\quad |110>=\begin{pmatrix}0\\0\\0\\0\\0\\0\\1\\0\end{pmatrix}\quad |111>=\begin{pmatrix}0\\0\\0\\0\\0\\0\\0\\1\end{pmatrix}$$

GroverのデータベースⅠ検索アルゴリズム

ある病気の一因となっている遺伝子を解読したヒトゲノムのデータベースから検索する

検索する状態を $|w\rangle = |010\rangle$ とすると、Groverのアルゴリズムは，２つのユニタリ行列，

$$U_1 = I_8 - 2|w\rangle\langle w|$$

$$U_2 = 2|s\rangle\langle s| - I_8$$

を初期の状態 $|s\rangle$ に交互に作用させることを繰り返すというものである。

ただし、顔文字｜０＞＜０｜のような演算は、直積を表す。

今の場合，

$$|w\rangle = |010\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad \langle w| = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad |w\rangle\langle w| = \begin{pmatrix} 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \end{pmatrix}$$

また、I_8 は８x８の単位行列である。

$$I_8 = \begin{pmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \end{pmatrix}$$

$$|s\rangle\langle s| = \begin{pmatrix} \frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8} \\ \frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8} \\ \frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8} \\ \frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8} \\ \frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8} \\ \frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8} \\ \frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8} \\ \frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8}&\frac{1}{8} \end{pmatrix}$$

Groverのデータベース検索アルゴリズム

ある病気の一因となっている遺伝子を解読したヒトゲノムのデータベースから検索する

$U_1 = I_8 - 2|w><w|$

$U_2 = 2|s><s| - I_8$

を満たす U_1、U_2はそれぞれ、

となる。

$$U_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$U_2 = \begin{pmatrix} -\frac{6}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} \\ \frac{2}{8} & -\frac{6}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} \\ \frac{2}{8} & \frac{2}{8} & -\frac{6}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} \\ \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & -\frac{6}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} \\ \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & -\frac{6}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} \\ \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & -\frac{6}{8} & \frac{2}{8} & \frac{2}{8} \\ \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & -\frac{6}{8} & \frac{2}{8} \\ \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & -\frac{6}{8} \end{pmatrix}$$

まず、 U_1 に初期の状態｜s>を作用させて、

$$U_1|s> = \frac{1}{\sqrt{8}}\begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

続いて、U_2を作用させて、

$$U_2U_1|s> = \begin{pmatrix} -\frac{6}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} \\ \frac{2}{8} & -\frac{6}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} \\ \frac{2}{8} & \frac{2}{8} & -\frac{6}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} \\ \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & -\frac{6}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} \\ \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & -\frac{6}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} \\ \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & -\frac{6}{8} & \frac{2}{8} & \frac{2}{8} \\ \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & -\frac{6}{8} & \frac{2}{8} \\ \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & \frac{2}{8} & -\frac{6}{8} \end{pmatrix} \frac{1}{\sqrt{8}}\begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \frac{1}{8\sqrt{8}}\begin{pmatrix} -6+2\times6-2 \\ -6+2\times6-2 \\ -6+2\times7 \\ -6+2\times6-2 \\ -6+2\times6-2 \\ -6+2\times6-2 \\ -6+2\times6-2 \\ -6+2\times6-2 \end{pmatrix} = \frac{1}{2\sqrt{8}}\begin{pmatrix} 1 \\ 1 \\ 2 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Groverのデータベース検索アルゴリズム

ある病気の一因となっている遺伝子を解読したヒトゲノムのデータベースから検索する

U_2U_1のユニタリ行列を繰り返し作用させることで,

検索する状態の確率振幅だけが増大する。

これは、量子干渉によって生じていると考えられる。

高々, $\sqrt{2^n}$ 回で検索されたと見なせるに充分な確率振幅となる。

暗号解読のShorの因数分解アルゴリズム

日本：２＋３＝ロ

イギリス：◯+△＝５

6257493337 x 6356046119 は簡単に求まるが,

39772916239307209103を因数分解するのには時間がかかる

RSA公開鍵暗号法

2000年までは、RSA129(426ビット)が用いられていたが，現在512ビットの素因数分解がクラウドコンピューティングで可能になったのを受けて，RSA1024を使うように推奨されている。

Shorの因数分解の手順

因数分解すべき大きな整数nが与えられると、

nより小さな互いに素な数をxとして、

$$f_x(a) \equiv x^a \bmod n$$

によって定義される関数 $f(a)$ の周期性を見つけることにより、nの因数を何回かの試行で推定できる。

$f(a)=1$の場合を　$x^r \equiv 1 \bmod n$　と表し,

$$\boldsymbol{f(a) \equiv x^a \bmod n}$$

を満たす周期 r （偶数）を整数aの重ね合わせの状態を用いて指数倍高速に見つける。r　が求まれば、

$\gcd(x^{r/2\pm1}, n)$　**が因数となる。**

nの２進数桁数のオーダ（$\log_2 n$）の計算量

大きな2つの素数の組 $\{p, q\}$ を生成し、それらの積 $n\ (=pq)$ を求める。

$(p-1)(q-1)$と互いに素な整数 d を求める。$\{d\}$は秘密鍵となる。

$ed \equiv 1 \bmod (p-1)(q-1)$　より e を求める。

組$\{e, n\}$ を暗号化に使用する公開鍵として通知する。

文章$\{M\}$をn以下の整数列$\{M_i\}$に変換し,

$E_i = M_i^e \bmod n$　により暗号化する。

受信者は

$M_i = E_i^d \bmod n$　を用いて、 E_iを復号化する。

傍受者は公開鍵 $\{e, n\}$を使って、E_i列の復号化を試みる。

傍受者が秘密鍵$\{d\}$を発見するには n を因数分解した結果の組 $\{p, q\}$ を発見する必要がある。

組 $\{p, q\}$がわかれば,

$ed \equiv 1 \bmod (p-1)(q-1)$　より d が求まる。

nの桁数の指数関数の計算量

第７回数学月間懇話会　2011.7.22　東大駒場キャンパス(数理科学研究科棟056教室)

まとめ

防衛大学校　応用物理学科　荒井　隆

- **Mooreの法則**が２０２０年まで成り立つのか,
  また、**２０２０年**にはパソコンの性能や**消費電力**はどうなるのか、
  非常に興味深い。

- 量子コンピューティングの基本的な考え方は、「全ての状態は**確率振幅で重ね合わせた状態**として扱う」ということである。
  これにより、**量子干渉**，**量子並列**といったこれまでにない論理発展をする点が新しい結果を生む。

「量子コンピューティング」西野哲郎，荒井　隆，渡辺昇、シュプリンガー・フェアラーク東京，　2000
「量子コンピュータの基礎」細谷暁夫、1999
「ファインマン　計算機科学」原康夫，中山　健，松田和典，岩波書店，1999
「量子情報理論」佐川弘幸，吉田宣章、シュプリンガー・フェアラーク東京，　2003
「Lectures on Quantum Information」Ed. Dagmar Bruβ and Gerd Leuchs, WILEY-VCH Verlag GmnH & Co., 2006
「Quantum Computing」Mika Hirvensalo, Springer-Verlag Berlin, 2003