Summary of Quantum Computering Study (量子電腦學教材摘要)

Contents

Ch 1. Fundamental Theorem of Quantum 量子學的基礎定理

1.1　State superposition and collapse ---quoted to ---> Consciousness is the foundation of quantum mechanics　（態疊加與塌縮 ---引證到---> 意識是量子力學的基礎）

1.2　Single superposition state ---Experiment is---> Schrodinger's cat ---Quoted to---> Consciousness is the foundation of the material world　（單體的疊加態 ---實驗為---> 薛定諤的貓 ---引證到---> 意識是物質世界的基礎）

1.3　Superposition state of many bodies ---quoted to ---> quantum entanglement　（多體的疊加態 ---引證到---> 量子糾纏）

1.4　Examples of development and application of quantum science　量子學的發展應用例子
Quantum teleportation, instantaneous transmission, past, present, and future entanglement, thoughts change reality...　（量子隱形傳輸、瞬間傳輸、過去、現在、未來的糾纏、念頭改變實相…）

1.5　Next wave of quantum computer science industry

-　RSA can be cracked

-　Protein folding can be simulated

-　New medicines and treatments

-　By 2023, quantum computers will become mainstream

    ■　Not the usual CPU + Memory model

    ■　Quantum Bits (qubits) are superposed & entangled

Ch 2. Quantum Crytography

- Non-Cloning Theorem

Photon Polarization Experience

BB84 protocol   (eavesdropper will corrupt data)

Ch 3. Foundation of Complex Number

Probability   P(A or B) = P(A)   +   P(B)   -   P(A and B)

Assume X is intermediate value, X^2 = -1

Imaginary quantity   i = $\sqrt{-1}$

Complex Number   a + bi   (real part + imaginary part)

Complex conjugate (a + bi)   = (a+bi)*   =   a – bi   (複合共軛物)

(a + bi)(a-bi) = square magnitude(a+bi) = $a^2 + b^2$

$| 3 + 2i |^2$ = $3^2 + 2^2$ = 9 + 4 = 13   (Vertical bars mean magnitude)

(-1/2   - $\sqrt{3}$ i/2)^2 = $1^2/2^2$ + $3i^2/4$ + $2\sqrt{3}$ i/4 = 1/4 – 3/4 + $\sqrt{3}i/2$ = -1/2 + $\sqrt{3}i/2$

$| 3 – 4i |$ = $\sqrt{}$ $3*3 + 4*4$ = $\sqrt{25}$ = 5

(-1/2   - $\sqrt{3}$ i/2)^3 = 1

Matrix Algebra (Linear Algebra /) ,   Transpose matrix [   ]^T

Transponse of Complex conjuage []$^+$ is   []^{*T} $^+$means Adjoint

[A][B] <> [B][A],   Identity Matrices

Asymmetric Algebra multiplication [3 by 3] x [3 by 1] = [3 by 1];

                                [1 by 3] x [3 by 1] = [1 by 1]

AND/OR/NOT Gates & Circuits

NAND

Inputs	Output
0 0	1
01	1
10	1
11	0

XOR

Inputs	Output
0 0	0
01	1
10	1
11	0

Ch 4. Quantum Physic Math Model

Square roots of negative probabilities will be complex numbers

Qubits can be OFF or ON or a superposition of OFF and ON

$\begin{bmatrix} p \\ q \\ r \end{bmatrix}$ represent superposition where p, q, r are complex number

$|\ |^2$ represent probabilities where $|p|^2 + |q|^2 + |r|^2 = 1$

Spin States

$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ $^<$ This is square root of probability of being measured in the pin Down State or "Qubit OFF" or

"Qubit 0"

$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $_<$ This is square root of probability of being measured in Spin Up State or "Qubit ON" or "Qubit 1"

$\begin{bmatrix} p \\ q \end{bmatrix}$ p and q are complex, if p = a+bi, then $|p|^2 = a^2 + b^2$

Since $|p|^2$ and $|q|^2$ are probabilities, we have $|p|^2 + |q|^2 = 1$

$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ have deterministric behavior only when measuring apparatus is aligned in the "up" or "down"

direction; otherwise, result will be probabilistic

Subsequent measurements with apparatus are deterministic; result of subsequent measurement is the same as the first measurement

$A^+E = [0\ 1] \times \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 1$ where A = Apparatus, E = Electron Spin; A^+ - Transpose of the complex conjugate of

the apparatus vector;   E – Vector for electron spin

Dirac Bra-Ket Notation

< A | E > = < A | x | E >   where   <A| is transpose of complex conjugate of A and is ajjoint of A (i.e. <A| = A^+

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

Irreversible Transformation measurement

Multi-Qubit Systems

2 Qubit state:

$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ i.e. $\begin{bmatrix} 00 \\ 01 \\ 10 \\ 11 \end{bmatrix}$ => Both Qubits are 0. |00>

$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$ i.e. $\begin{bmatrix} 00 \\ 01 \\ 10 \\ 11 \end{bmatrix}$ => First qubit is 0, second qubit is 1. |01>

$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$ i.e. $\begin{bmatrix} 00 \\ 01 \\ 10 \\ 11 \end{bmatrix}$ => First qubit is 1, second qubit is 0. |10>

$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ i.e. $\begin{bmatrix} 00 \\ 01 \\ 10 \\ 11 \end{bmatrix}$ => Both Qubits are 1. |11>

Superposition of 2 Qubit state

$\begin{bmatrix} 1/\sqrt{2} \\ 0 \\ 0 \\ 1/\sqrt{2} \end{bmatrix}$ In Dirac Notation $1/\sqrt{2}$ (|00> + |11>)

Entanglement

$\begin{bmatrix} 1/\sqrt{2} \\ 0 \\ 0 \\ 1/\sqrt{2} \end{bmatrix}$ $= 1/\sqrt{2}$ (|00> + |11>) $\begin{bmatrix} P(00) = 0.5 \\ P(01) = 0 \\ P(10) = 0 \\ P(11) = 0.5 \end{bmatrix}$

Quantum Computer Model

Ancilla   (reference - 附屬物)

Fanout gate to copy the state of 0 or 1, but cannot copy superposition and entanglement.

Hadamard H   (interchange of superposition)

$1/\sqrt{2}$ $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

H | 0> = $1/\sqrt{2}$ (|0> + |1>)

H | 1> = $1/\sqrt{2}$ (|0> - |1>)

CNOT gate (Controlled NOT)

x XOR y   (x controlled on y)   CNOT gate is own reversible

CCNOT

z XOR (x AND y) XOR (x AND y) = z

8 Qubit = 8 element in matrix

Ch 5. Q# sample program

```
namespace Quantum.QB4
{
    open Microsoft.Quantum.Intrinsic;
    open Microsoft.Quantum.Canon;
     open Microsoft.Quantum.Convert;

     @EntryPoint()
     operation QB4Run () : Unit
    {

         mutable num0000 = 0;
         mutable num0001 = 0;
         mutable num0010 = 0;
```

4

```
        mutable num0011 = 0;
        mutable num0100 = 0;
        mutable num0101 = 0;
        mutable num0110 = 0;
        mutable num0111 = 0;
        mutable num1000 = 0;
        mutable num1001 = 0;
        mutable num1010 = 0;
        mutable num1011 = 0;
        mutable num1100 = 0;
        mutable num1101 = 0;
        mutable num1110 = 0;
        mutable num1111 = 0;

        using (qubits = Qubit[4])     \\allocate 4 qubit
        {
            for (test in 1..10000)
            {
                Set (Zero, qubits[0]);
                Set (Zero, qubits[1]);
                Set (Zero, qubits[2]);
                Set (Zero, qubits[3]);

                //Demo of pass through (NOP)
                //Set (Zero, qubits[0]);
                //Set (Zero, qubits[1]);
                //Set (Zero, qubits[2]);
                //Set (Zero, qubits[3]);

                //Demo of NOT gate
                //Set (Zero, qubits[0]);
                //X(qubits[0]);

                //Demo of Hadamard
                //Set (Zero, qubits[0]);
                //H(qubits[0]);

                //Demo of SWAP
                //Set (One, qubits[0]);
                //SWAP (qubits[0],qubits[1]);

                //Demo of Controlled-NOT   ( CNOT )
                //Set (One, qubits[0]);
                //Set (Zero, qubits[1]);
                //CNOT(qubits[0],qubits[1]);

                //Demo of Bell State
                //Set (Zero, qubits[0]);
                //Set (Zero, qubits[1]);
                //H(qubits[0]);
                //CNOT(qubits[0],qubits[1]);

                //Demo of Toffoli Gate in general configuration
                //Set (One, qubits[0]);
                //Set (One, qubits[1]);
                //Set (One, qubits[2]);
                //CCNOT(qubits[0],qubits[1],qubits[2]);

                //Demo of Toffoli Gate in NOT configuration
                //Set (One, qubits[0]); // Always 1
                //Set (One, qubits[1]); // Always 1
                //Set (One, qubits[2]); // Input of NOT gate. Gets inverted by NOT
                //CCNOT(qubits[0],qubits[1],qubits[2]);

                //Demo of Toffoli Gate in AND configuration
                //Set (One, qubits[0]); // Input 1 of AND
                //Set (One, qubits[1]); // Input 2 of AND
                //Set (Zero, qubits[2]);// Initialized to 0. Gets the result of AND
                //CCNOT(qubits[0],qubits[1],qubits[2]);

                //Demo of Toffoli Gate in FANOUT configuration
                //Set (One, qubits[0]); // Always 1
                //Set (One, qubits[1]); // Input & Pass through.
                //Set (Zero, qubits[2]); //Initialized to 0. Gets a result of the fanout from
qubits[1]
```

```
                //CCNOT(qubits[0],qubits[1],qubits[2]);

                let res0 = M (qubits[0]);
                let res1 = M (qubits[1]);
                let res2 = M (qubits[2]);
                let res3 = M (qubits[3]);

                if(res3 == Zero and res2 == Zero and res1 == Zero and res0 == Zero){set num0000 =
num0000 + 1;}
                if(res3 == Zero and res2 == Zero and res1 == Zero and res0 == One ){set num0001 =
num0001 + 1;}
                if(res3 == Zero and res2 == Zero and res1 == One  and res0 == Zero){set num0010 =
num0010 + 1;}
                if(res3 == Zero and res2 == Zero and res1 == One  and res0 == One ){set num0011 =
num0011 + 1;}
                if(res3 == Zero and res2 == One  and res1 == Zero and res0 == Zero){set num0100 =
num0100 + 1;}
                if(res3 == Zero and res2 == One  and res1 == Zero and res0 == One ){set num0101 =
num0101 + 1;}
                if(res3 == Zero and res2 == One  and res1 == One  and res0 == Zero){set num0110 =
num0110 + 1;}
                if(res3 == Zero and res2 == One  and res1 == One  and res0 == One ){set num0111 =
num0111 + 1;}
                if(res3 == One  and res2 == Zero and res1 == Zero and res0 == Zero){set num1000 =
num1000 + 1;}
                if(res3 == One  and res2 == Zero and res1 == Zero and res0 == One ){set num1001 =
num1001 + 1;}
                if(res3 == One  and res2 == Zero and res1 == One  and res0 == Zero){set num1010 =
num1010 + 1;}
                if(res3 == One  and res2 == Zero and res1 == One  and res0 == One ){set num1011 =
num1011 + 1;}
                if(res3 == One  and res2 == One  and res1 == Zero and res0 == Zero){set num1100 =
num1100 + 1;}
                if(res3 == One  and res2 == One  and res1 == Zero and res0 == One ){set num1101 =
num1101 + 1;}
                if(res3 == One  and res2 == One  and res1 == One  and res0 == Zero){set num1110 =
num1110 + 1;}
                if(res3 == One  and res2 == One  and res1 == One  and res0 == One ){set num1111 =
num1111 + 1;}
            }
            Set(Zero, qubits[0]);
            Set(Zero, qubits[1]);
            Set(Zero, qubits[2]);
            Set(Zero, qubits[3]);
        }
        Message("0000 -> " + DoubleAsString(IntAsDouble(num0000)/10000.0));
        Message("0001 -> " + DoubleAsString(IntAsDouble(num0001)/10000.0));
        Message("0010 -> " + DoubleAsString(IntAsDouble(num0010)/10000.0));
        Message("0011 -> " + DoubleAsString(IntAsDouble(num0011)/10000.0));
        Message("0100 -> " + DoubleAsString(IntAsDouble(num0100)/10000.0));
        Message("0101 -> " + DoubleAsString(IntAsDouble(num0101)/10000.0));
        Message("0110 -> " + DoubleAsString(IntAsDouble(num0110)/10000.0));
        Message("0111 -> " + DoubleAsString(IntAsDouble(num0111)/10000.0));
        Message("1000 -> " + DoubleAsString(IntAsDouble(num1000)/10000.0));
        Message("1001 -> " + DoubleAsString(IntAsDouble(num1001)/10000.0));
        Message("1010 -> " + DoubleAsString(IntAsDouble(num1010)/10000.0));
        Message("1011 -> " + DoubleAsString(IntAsDouble(num1011)/10000.0));
        Message("1100 -> " + DoubleAsString(IntAsDouble(num1100)/10000.0));
        Message("1101 -> " + DoubleAsString(IntAsDouble(num1101)/10000.0));
        Message("1110 -> " + DoubleAsString(IntAsDouble(num1110)/10000.0));
        Message("1111 -> " + DoubleAsString(IntAsDouble(num1111)/10000.0));
    }

    operation Set (desired: Result, q1: Qubit) : Unit
    {
        let current = M(q1);                \\measure of Qubit
        if (desired != current)
        {
            X(q1);
        }
    }

}
```
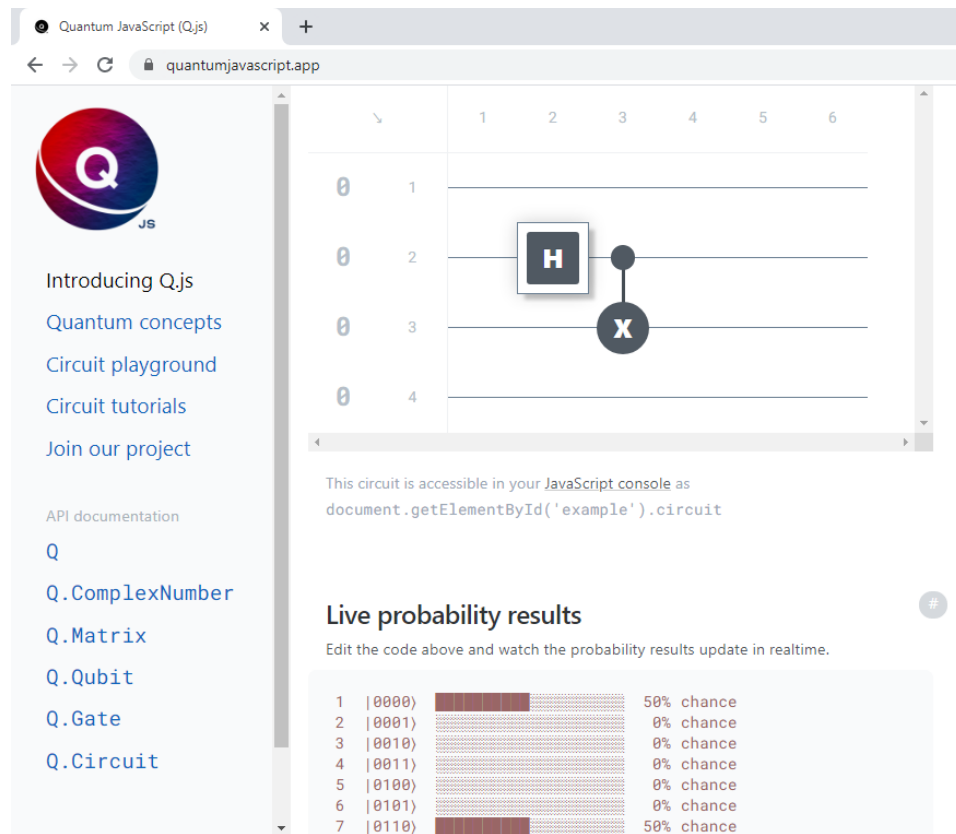
Ch 6. Quantum JavaScript (Q.js)

Q is a quantum circuit simulator, drag-and-drop circuit editor, and powerful JavaScript library that runs right here in your web browser. There's nothing to install and nothing to configure, so jump right in and experiment. → https://quantumjavascript.app/

Appendix 1. What is "Akashic"？ 什麼是"阿卡西"？

Akashic can be treated as the universal domain or unlimited domain (i.e. main server of the whole universal.

Akashic records Akashic (Akashic) is Sanskrit, which means "ether" or the primitive energy of the universe. The Akashic record represents the highest dimension of consciousness, a range of sacred mysteries. It is the "book of life" of the universe. It contains the history of the entire universe. It not only contains the life blueprint of the universe, but also records all the experiences and experiences of all living entities in different time periods according to this blueprint.

The Akashic record contains two parts. The first part is the blueprint of the divine life of the universe's life reality or source. This reality or blueprint is invisible, unchangeable, eternal, and without beginning and end. The other part is all the experiences and activities of Source as a life individual in different times and spaces. In such experiences and experiences, the source fully manifests and expresses the reality of life or the blueprint of the divine life (that is, the source itself) in a tangible way.

Akashic records neither talk about supernatural powers nor psychic. From the perspective of Akashic records, ascension is not an external event that occurs at a certain point in time, but a process that occurs within each of us. The Akashic record represents the highest frequency of consciousness in the universe,

leading us straight to the source. But it actually only focuses on one thing: taking the problems in human life as a breakthrough and entry point, leading you to see your life blueprint, enter the deepest reality of life, awaken your divine power, and then in the most ordinary Live out your highest spirituality in your real life.

The Akashic record contains such endless life, mystery and wisdom. In recent years, many people have found that using Akashic records can effectively assist their original work, such as therapeutic consultation, physical exercises, energy healing, artistic creation and even business.

阿卡西 (Akashic) 可被視為無量之網通或宇宙網絡（即整個宇宙的主服務器）。
阿卡西記錄（Akashic records） 阿卡西（阿卡夏）是梵文，它的意思是"以太"或者宇宙的原始能量。阿卡西記錄代表著最高的意識次元，是一個神聖奧秘的範圍。它是宇宙的"生命之書"，蘊含著整個宇宙的歷史，既包含宇宙的生命藍圖，也記載著所有生命體按照這個藍圖在不同時間段的所有經歷和體驗。

阿卡西記錄包含兩個部分。第一部分是宇宙的生命實相或者源頭的神聖生命藍圖。這個實相或藍圖無形無相，不可更改，永恆不變，無始也無終。另一部分是源頭作為生命個體在不同時間和空間裡進行的所有體驗和活動。在這樣的體驗和經歷中，源頭就把生命實相或神聖生命藍圖（也就是源頭自己）以有形有體的方式完全、豐滿地彰顯和表達出來。

阿卡西記錄不講神通，也不講通靈。從阿卡西記錄的視角來看，揚升不是某個時間點會發生的一個外在事件，而是發生在我們每個人內在的一個過程。阿卡西記錄代表宇宙的最高意識振頻，引領我們直達源頭。但它其實只關注一件事：以人性生活中的問題為突破口和切入點，引領你看到你的生命藍圖，進入最深層的生命實相，喚醒你的神性力量，從而在最平凡的現實生活中活出你的最高靈性。

阿卡西記錄蘊含著如此追測不盡的生命、奧秘和智慧。近些年來，許多人發現運用阿卡西記錄能夠有效地輔助他們原有的工作，比如治療性諮詢、身體練習、能量療愈、藝術創作甚至商業。

Appendix 2. How Healy Watch Determine the Body-Mind-Soul Balance (Healy 手錶如何確定身心平衡)

The Healy Watch makes Healy a closed feedback system for body and mind. It measures your bioenergetic state through an analysis of BRA (Bioenergetic Rhythm Analysis) and HRV (Heart Rate Variability). Quickly and easily.

I am investigating a quantum resonance analysis therapy device. I find it has much marketing potential, and have many areas to develop and invent. My investigation direction will be to use Quantum technology to link with this Healy device, and to develop function to check, predict, and improve healthy.

Feel free to contact https://Skywaycare.com or whatsapp 95878788 for more info about Healy Watch.

Healy Watch 使 Healy 成為針對身心的封閉反饋系統。 它通過對 BRA（生物能節奏分析）和 HRV（心率變異性）的分析來測量您的生物能狀態， 輕鬆快捷。

我正在研究一種量子共振分析治療儀。 我發現它具有很大的營銷潛力,並且有很多領域需要開發和發明。 我研究的方向是使用 Quantum 技術與該 Healy 設備鏈接,並開發檢查、預測和改善身體健康的功能。

歡迎隨時聯繫 https://Skywaycare.com 或 whatsapp 95878788,以獲取有關 Healy Watch 的更多信息。