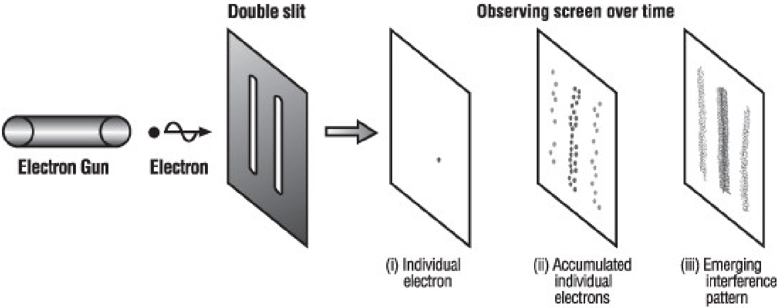
An introduction

Oswaldo Gomez

AI Engineering

January 21, 2020

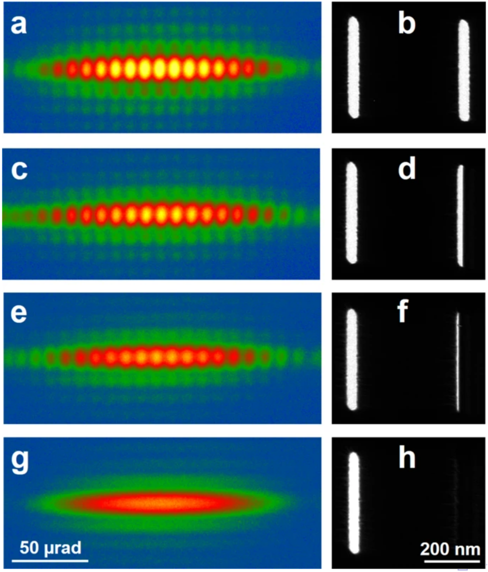


Double-slit apparatus showing the pattern of electron hits on the observing screen building up over time.

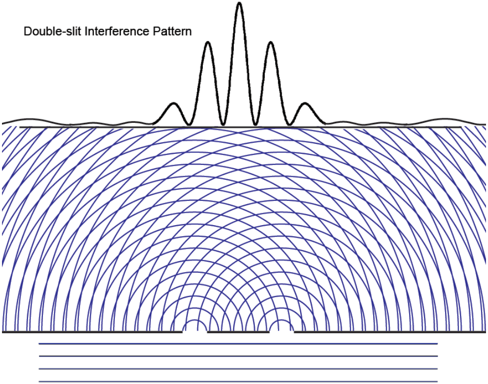
Figure: Image credit: ©2012 Perimeter Institute for Theoretical Physics, via https://www.perimeterinstitute.ca/research/research-areas/quantum-foundations/more-quantum-foundations.

Quantum world if fascinating

Particles behave like waves

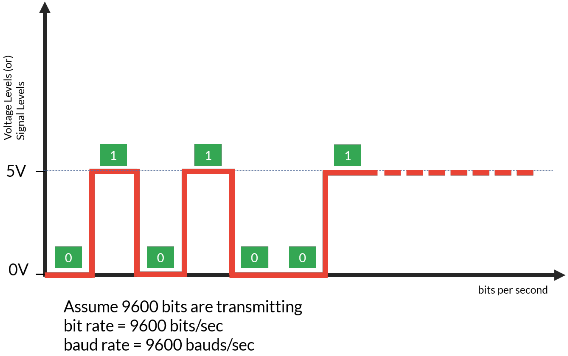


The state of a particle after passing through either one of the slits can be described as a *wave* function (probability distribution) namely $\Psi = (\alpha_0\psi_0 + \alpha_1\psi_1)$ with $\{\alpha_0, \alpha_1\} \in \mathbb{C}$



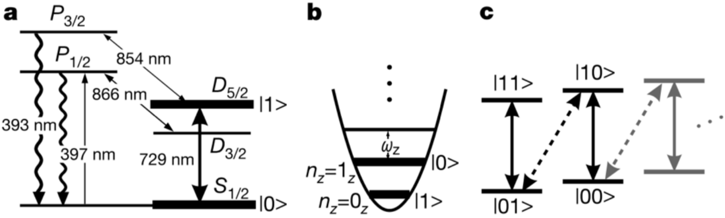
Basic Unit of information: Bits

Traditional computation works with 0 and 1 as basic units of information. A physical realization of this is voltage from 0V to 5V



Basic Unit of information: Qubits

Quantum computation works with $|0\rangle$ and $|1\rangle$ as basic units of information. A physical realization of this would be a spin $1/2$ particle.



Computational basis states

Qubits can be in different states *other* than $|0\rangle$ or $|1\rangle$. It is possible to form *linear combinations* of states, called superpositions:

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$$

The numbers α_0 and α_1 are complex numbers and $|\alpha_0|^2 + |\alpha_1|^2 = 1$.

Where $|0\rangle$ and $|1\rangle$ are vectors $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ in $\mathbb{C}^2$.

A superposition state is a linear combination $\psi = \alpha_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \alpha_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

Quantum NOT gate

Classical computer circuits consists of wires and logic gates. E.g. the NOT gate which has a truth table $0 \to 1$ and $1 \to 0$.

The analogous quantum operation would take

$$\alpha_0 |0\rangle + \alpha_1 |1\rangle$$

to

$$\alpha_0 |1\rangle + \alpha_1 |0\rangle$$

Since a quantum state can be represented as a vector, we are looking for a matrix such that:

$$\mathrm{X} \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_0 \end{bmatrix}$$

Quantum Gates

Quantum gates are represented by matrices applied on our vectors (qubits). Are all matrices valid Quantum Gates?...no.

Recall that $|\alpha_0|^2 + |\alpha_1|^2 = 1$ for a quantum state

$$\alpha_0 |0\rangle + \alpha_1 |1\rangle$$

. This must also hold for

$$\alpha_0^{'} |0\rangle + \alpha_1^{'} |1\rangle$$

after the gate has acted. It turns out that the appropriate condition on the matrix representing the gate is tha the matrix U be unitary. That is (with $U^\dagger$ is the adjoint of U)

$$U^\dagger U = I$$

Quantum Gates

In classical computers we only have one non-trivial gate for one bit (NOT gate). In the case of Quantum computers we have many!

For example the NOT gate:

$$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

The *Haddamard* gate

$$H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Quantum Gates

Some single qubit matrices are used quite frequently and are very useful.

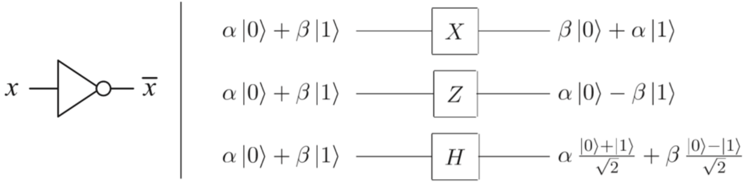


Figure: Single bit (left) and qubit (right) logic gates.

Balanced or constant

We are task with identifying fair vs unfair coins

We represent the four possible outputs of flicking the coin as four functions f that map one input bit (a=0,1 representing "heads" or "tails") onto one output bit (f(a)=0,1 standing for "head or tail").

Table 1 **Truth table for the four possible functions**

	Constant functions		Balanced functions	
	Case 1	Case 2	Case 3	Case 4
$f(0)$	0	1	0	1
$f(1)$	0	1	1	0
$w \oplus f(a)$	ID	NOT	CNOT	Z-CNOT

Figure: The third line is the effect of the logic U_{f_n} on the bottom qubit

Deutsch's algorithm

Quantum circuits can outperform classical ones. Deutsch's algorithm combines quantum parallelism with quantum interference. In the previous slide, we would need two evaluations of $f(x)$ in classical computation, but only one using this Quantum algorithm.

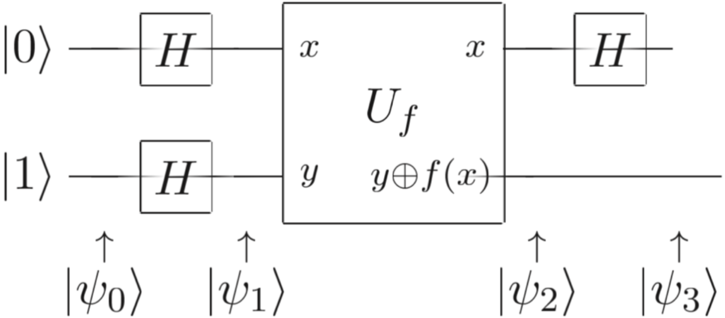


Figure: Quantum circuit implementing Deutsch's algorithm

Bibliografía

- What is bit rate and baud rate with examples – bytesofgigabytes.com. (2019). Retrieved 20 January 2020, from http://www.bytesofgigabytes.com/embedded/bit-rate-and-baud-rate/

- Nielsen, M., & Chuang, I. (2010). Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511976667

- Gulde, S., Riebe, M., Lancaster, G. et al. Implementation of the Deutsch–Jozsa algorithm on an ion-trap quantum computer. Nature 421, 48–50 (2003) doi:10.1038/nature01336
- Tavabi, A., Boothroyd, C., Yücelen, E., Frabboni, S., Gazzadi, G., Dunin-Borkowski, R., & Pozzi, G. (2019). The Young-Feynman controlled double-slit electron interference experiment. Scientific Reports, 9(1). doi: