Matrics, Computational Model and Circuits

Notion of matrix inverse

A matrix A is called invertible whenever there exists a matrix A^{-1} such that

$$A \cdot A^{-1} = Id = A^{-1} \cdot A$$

For instance : $Id^{-1} = Id$

What about : $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}^{-1}$?

Sends $|0\rangle \mapsto 0 \cdot |0\rangle + 1 \cdot |1\rangle = |1\rangle$ et $|1\rangle \mapsto |0\rangle$

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

Unitary matrix

In quantum computation, the operators on the quantum memory are **unitary** operations/matrices

Def of unitary matrix :
→ its inverse is equal to its conjugate transpose

If $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$, the transpose of A (denoted with A^T) is $A^T = \begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{pmatrix}$,
symmetry wrt the diagonal

The conjugate transpose A^* (or $A^\dagger$) is the transpose followed with the complex conjugate:

$$A^* = \begin{pmatrix} \overline{a_{11}} & \overline{a_{21}} \\ \overline{a_{12}} & \overline{a_{22}} \end{pmatrix}$$

→ **property** : a unitary map is precisely an orthonormal basis change

For instance

H (or Had) $= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ is equal to its conjugate transpose

Do we have $H \cdot H = Id$? $= \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} = Id$ (answer: yes)

H sends the basis $|0\rangle, |1\rangle$ to $|+\rangle, |-\rangle$

Other examples : Pauli matrices

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\ Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},\ Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, Id = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- All equal to their conjugate transpose
- Moreover $X \cdot X = Id, \quad Y \cdot Y = Id, \quad Z \cdot Z = Id$

Other standard unitary matrices

$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \qquad T^* = \begin{pmatrix} 1 & 0 \\ 0 & e^{-i\pi/4} \end{pmatrix}$ and therefore $T \cdot T^* = Id$

since $e^{i\pi/4}e^{-i\pi/4} = e^{i(\pi/4-\pi/4)} = e^0 = 1$

$S = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ (Note : $T^2 = S$ and $S^2 = Z$)

Parametrization of 1-qubit gates

The canonical form of a unitary map on 1 qubit is parametrized by 3 angles:

$$\begin{pmatrix} \cos(\theta/2) & -e^{i\lambda}\sin(\theta/2) \\ e^{i\phi}\sin(\theta/2) & e^{i\lambda+i\phi}\cos(\theta/2) \end{pmatrix}$$

Note : this parametrization is **modulo a global phase**

Exercice : some computations

- Compute HZH
- We saw that $\sqrt{Z} = S$. What could be $\sqrt{X}$?

Answers:

$$HZH = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$
$$= \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 0 & 2 \\ 2 & 0 \end{pmatrix} = X$$

Corollary : $HXH = HHZHH = Z$

$\sqrt{X} = HSH$ car $\begin{array}{l}(HSH)(HSH) = HS(HH)SH = HSSH \\ = HZH = X\end{array}$

On 2 qubits

SWAP : $|xy\rangle \mapsto |yx\rangle$ has matrix $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ $\begin{matrix}|00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle\end{matrix}$

$|00\rangle|01\rangle|10\rangle|11\rangle$

Unitary matrix, is its own inverse.

On $n+1$ qubits

If A is a unitary on n qubits, one can build a new unitary map called **controlled operation**

Pick $A : \mathcal{H}^{\otimes n} \to \mathcal{H}^{\otimes n}$
Build
$C-A : \mathcal{H} \otimes \mathcal{H}^{\otimes n} \to \mathcal{H} \otimes \mathcal{H}^{\otimes n}$
as follows

$$C-A : |0\rangle \otimes |x\rangle \mapsto |0\rangle \otimes |x\rangle$$

$$|1\rangle \otimes |x\rangle \;\mapsto |1\rangle \otimes \left(A \cdot |x\rangle\right)$$

$$C-A \;=\; \begin{array}{c} \begin{pmatrix} Id & 0 \\ 0 & A \end{pmatrix} \begin{array}{l} |0x\rangle \\ |1x\rangle \end{array} \\ |0x\rangle \; |1x\rangle \end{array}$$

For instance :
the gate CNOT , or $C-X$: acts on 2 qubits

$$\begin{aligned} C-X : |0\rangle \otimes |b\rangle \;&\mapsto |0\rangle \otimes |b\rangle \\ |1\rangle \otimes |b\rangle \;&\mapsto |1\rangle \otimes |\neg b\rangle \end{aligned}$$

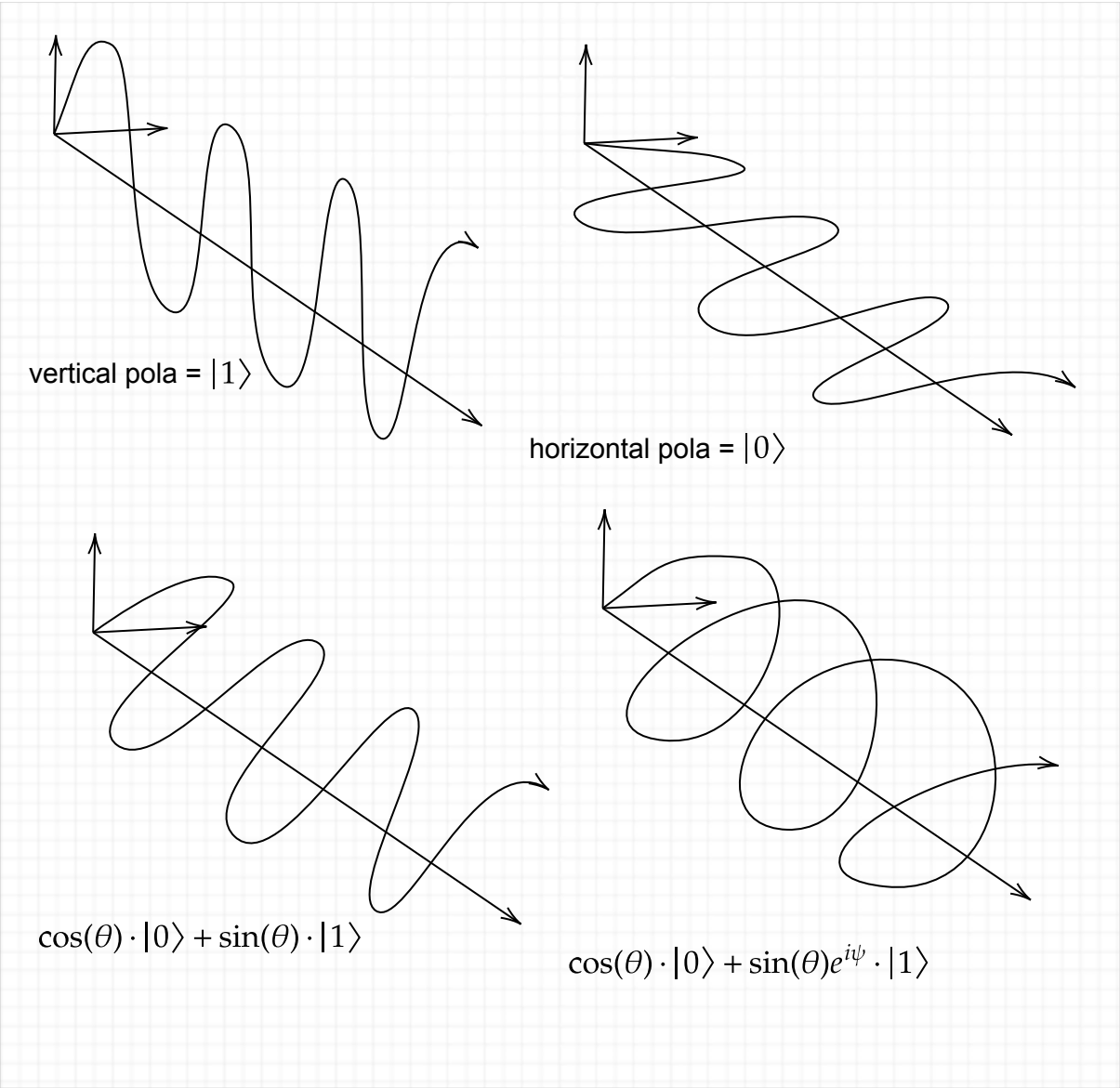
$$\begin{array}{c} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{array}{l} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{array} \\ |00\rangle |01\rangle |10\rangle |11\rangle \end{array}$$

One can also build $C-C-X$, or **Toffoli**
$|xy\rangle \otimes |z\rangle \;\mapsto |xy\rangle \otimes |(x \cdot y) \oplus z\rangle$

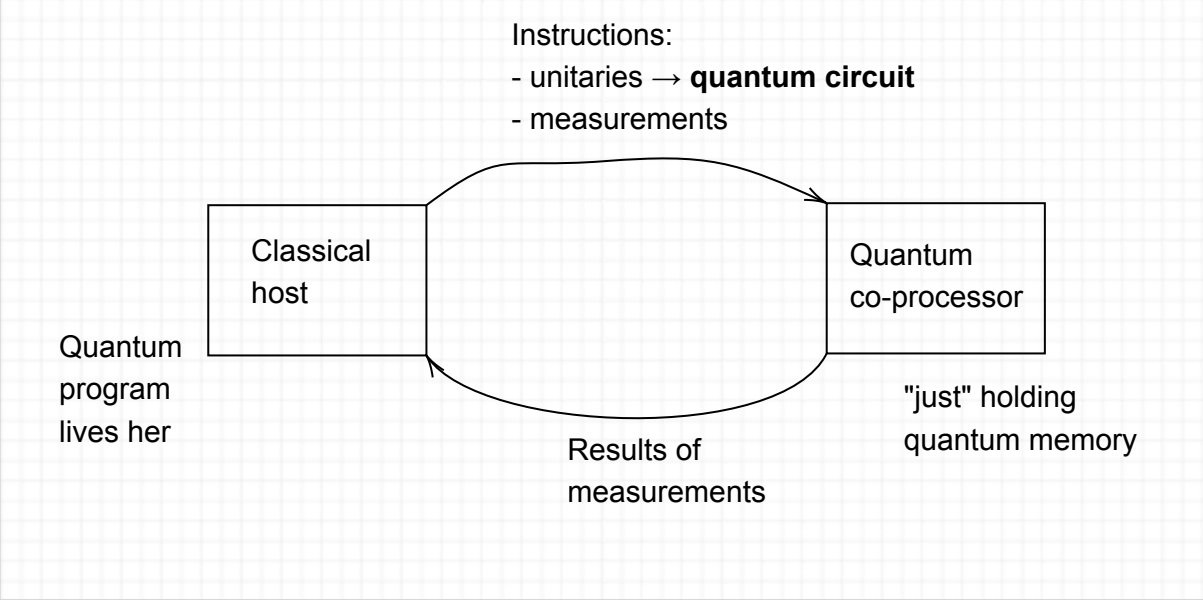
Note : $C-X : |x\rangle \otimes |z\rangle \;\mapsto |x\rangle \otimes |x \oplus z\rangle$

Example of a physical encoding of qubit

Polarization of a photon



The quantum co-processor model



The quantum co-processor holds an array of **individually addressable** quantum bits. It comes with a set of pre-defined unitaries ---typically on 1 or 2 qubits--- that it can perfom on the memory. In general, we aim for a **universal set of unitaries,** that is, a set from which one can realize any global unitary on the memory, using sequential composition and tensoring of gates.

Terminology: a **quantum gate** is a native elementary unitary operation of the quantum co-processor

Typical quantum algorithm

A quantum algorithm is meant to solve a **classical problem:**

- Factoring a number
- Finding an element in a graph
- ...

Any (classical or quantum) algorithm starts with a **problem instance**: the number to factor, the structure of the graph, etc.

A bird-eye view of the typical quantum algorithm is as follows:

Start with a problem instance, then:

1. Generate a sequence of quantum gates (a.k.a. a circuit -- see below)
2. Initialize the quantum memory
3. Apply the sequence of gates
4. Measure the quantum memory (see below for the behavior) to get a bitstring
5. Post-process: can we infer a result to the problem?
    (a) If yes: exit with the result
    (b) If no: start again at (1) if the circuit needs to be updated, or at (2) if we can

keep the same one.

There is therefore a global loop. The measurement being a probabilistic operation (see below), we eventually exit the loop.

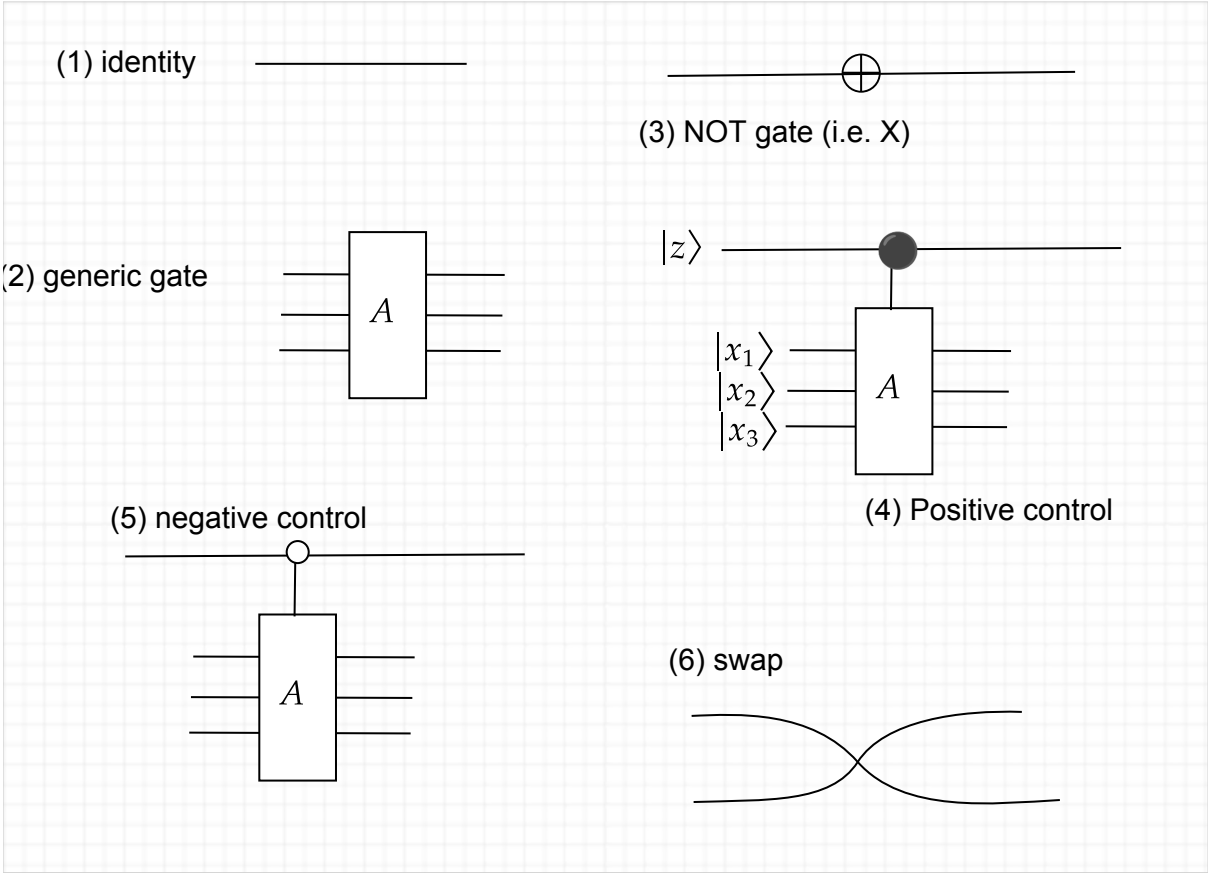
All the game in the design of a quantum algorithm is to make sure that realizing this sequence of steps is indeed more efficient than any other classical algorithms. There are two approaches for this:

- Theoretical complexity analysis: how does the run-time of the algorithm behaves as the size of the instance grow towards infinity ?
- Practical considerations: What is the concrete list of gates, and does it have any chance to be indeed useable in practice, for a concrete problem?

For instance, there might be some huge overhead that gets negligeable as the instance size grows large enough (e.g. the number of atoms in the universe). But this situation might never happen in practice, if the targetted instances never reach these sizes.

Now, let us see the two main components of quantum algorithms:
quantum circuits and measurements.

Notion of quantum circuit



For instance

(4) : Assume a gate A acting on 3 qubits
the gate $C-A$ acts on $|z\rangle \otimes |x_1x_2x_3\rangle$ and behaves as
$|0\rangle \otimes |x_1x_2x_3\rangle \mapsto |0\rangle \otimes |x_1x_2x_3\rangle$
and $|1\rangle \otimes |x_1x_2x_3\rangle \mapsto |1\rangle \otimes \left(A\,|x_1x_2x_3\rangle\right)$

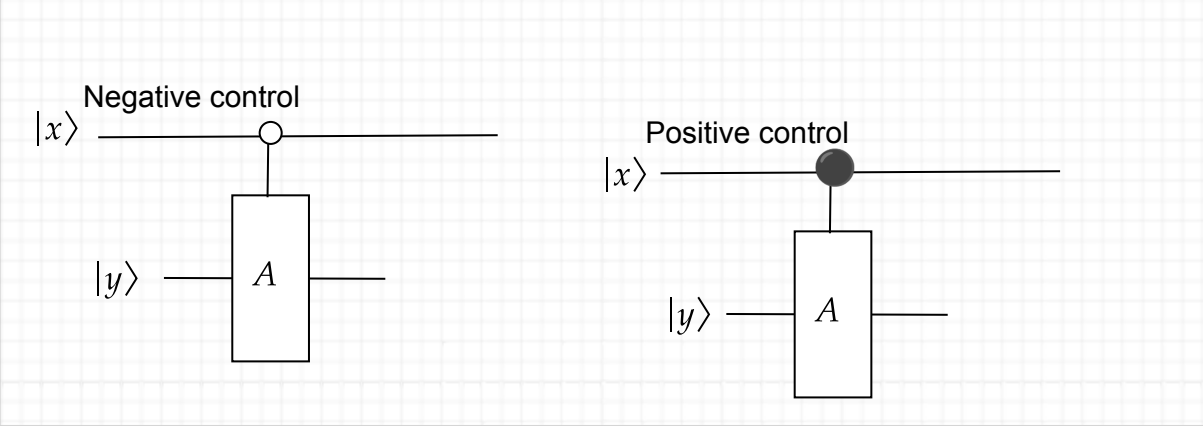
It is a linear operation, so :

$$\sum_{z,x_1,x_2,x_3} \alpha_{z,x_1,x_2,x_3} \cdot |z\rangle \otimes |x_1x_2x_3\rangle$$
$$=$$
$$\sum_{x_1,x_2,x_3} \alpha_{0,x_1,x_2,x_3} \cdot |0\rangle \otimes |x_1x_2x_3\rangle \;+$$
$$\sum_{x_1,x_2,x_3} \alpha_{1,x_1,x_2,x_3} \cdot |1\rangle \otimes |x_1x_2x_3\rangle$$

$$\mapsto$$

$$\sum_{x_1,x_2,x_3} \alpha_{0,x_1,x_2,x_3} \cdot |0\rangle \otimes |x_1x_2x_3\rangle \;+$$
$$\sum_{x_1,x_2,x_3} \alpha_{1,x_1,x_2,x_3} \cdot |1\rangle \otimes \left(A|x_1x_2x_3\rangle\right)$$

Controls:



Negative case : $|0\rangle \otimes |y\rangle \mapsto |0\rangle \otimes \left(A|y\rangle\right)$ et $|1\rangle \otimes |y\rangle \mapsto |1\rangle \otimes |y\rangle$
Positive case : $|1\rangle \otimes |y\rangle \mapsto |1\rangle \otimes \left(A|y\rangle\right)$ et $|0\rangle \otimes |y\rangle \mapsto |0\rangle \otimes |y\rangle$

Some remarks:

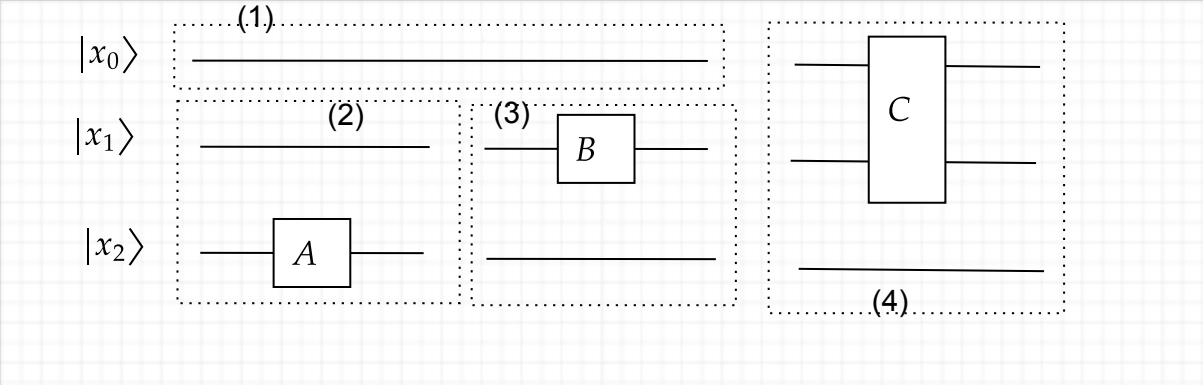
- vertical concatenation ≡ tensoring
- horizontal concatenation ≡ composition
- Wires are always horizontal: no branching, no loop

A circuit is a linear, sequential description of operations

Note : We need a convention when reading wires -- here we read from top to bottom

Parallelism

Consider the following circuit acting on $|x_0 x_1 x_2\rangle$



3 operations, ordered as follows: A on x_2 then B on x_1 then C on x_0 and x_1

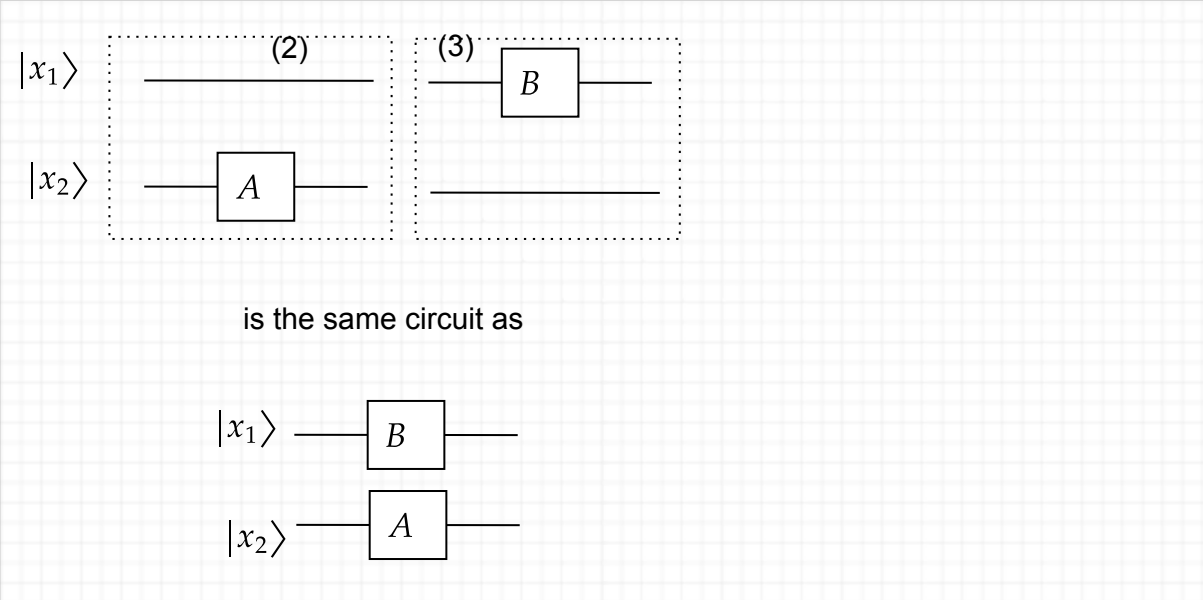
The circuit is built piecewise :
(1) : Identity on x_0 : Id_1
(2) : Operation on 2 qubits (x_1 and x_2), it is $Id_1 \otimes A$
(3) : Operation on 2 qubits (x_1 and x_2) : $B \otimes Id_1$
(4) : Operation on the 3 qubits : $C \otimes Id_1$

Horizontal composition of (2) and (3) : $|x_1 x_2\rangle \mapsto (B \otimes Id)\big((Id \otimes A)|x_1 x_2\rangle\big)$
Let us simplify :

$$(B \otimes Id)\big((Id \otimes A)|x_1 x_2\rangle\big)$$
$$=$$
$$(B \otimes Id)\Big((Id \otimes A)\big(|x_1\rangle \otimes |x_2\rangle\big)\Big)$$
$=$ (distributivity between tensor and application)
$$(B \otimes Id)\Big(\big(Id|x_1\rangle\big) \otimes \big(A|x_2\rangle\big)\Big)$$
$$=$$
$$(B \otimes Id)\Big(|x_1\rangle \otimes \big(A|x_2\rangle\big)\Big)$$
$=$ (distributivity between tensor and application)

$$\left(B|x_1\rangle\right) \otimes \left(Id\left(A|x_2\rangle\right)\right)$$
$$=$$
$$\left(B|x_1\rangle\right) \otimes \left(A|x_2\rangle\right)$$
$$=$$
$$(A \otimes B)|x_1 x_2\rangle$$

So



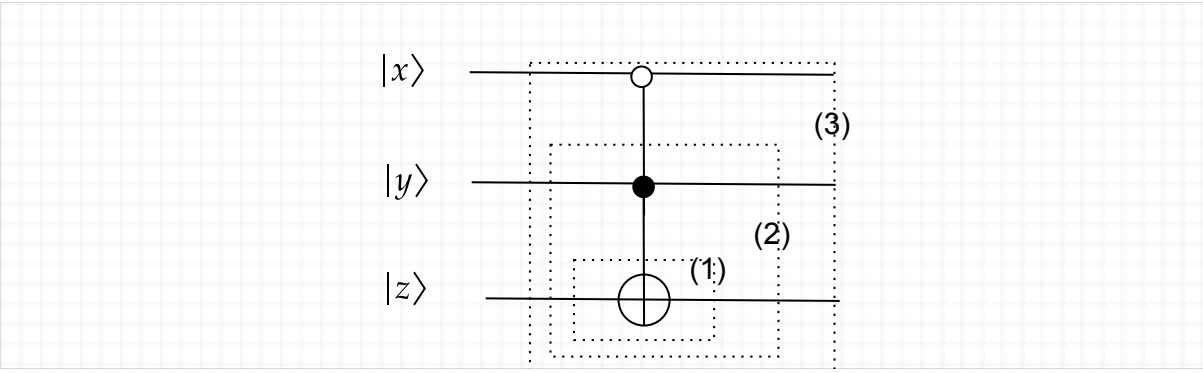
If we add (1), we get an operation on 3 qubits : $Id_1 \otimes A \otimes B$

And if we furthermore add (4) : $(C \otimes Id_1)(Id_1 \otimes A \otimes B)$

We cannot however swap C et $A \otimes B$ -- in any case, not in a generic manner

Examples

multi-control NOT.



(With basis vectors so that it is easy to write down)

Notations :

- $\otimes \equiv$ tensor of vector space $\equiv$ juxtaposition of circuits
- $\oplus \equiv$ boolean XOR : $0 \oplus 0 = 0\,,\, 1 \oplus 1 = 0\,,\, 0 \oplus 1 = 1\,,\, 1 \oplus 0 = 1$

(1) : $|z\rangle \mapsto |1 \oplus z\rangle$

In other words : $|0\rangle \mapsto |1\rangle$ et $|1\rangle \mapsto |0\rangle$

(2) :

$|0\rangle \otimes |z\rangle \mapsto |0\rangle \otimes |z\rangle$

(control qubit at 0 → no action)

$|1\rangle \otimes |z\rangle \mapsto |1\rangle \otimes |1 \oplus z\rangle$

(control qubit at 1 → apply (1) on z)

On a general basis vector z:

$|y\rangle \otimes |z\rangle \mapsto |y\rangle \otimes |y \oplus z\rangle$

Control-NOT is a standard operation. In the canonical basis, its matrix is

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix}$$

$$|00\rangle|01\rangle|10\rangle|11\rangle$$

(3) :

$|0\rangle \otimes |y\rangle \otimes |z\rangle \mapsto |0\rangle \otimes |y\rangle \otimes |y \oplus z\rangle$

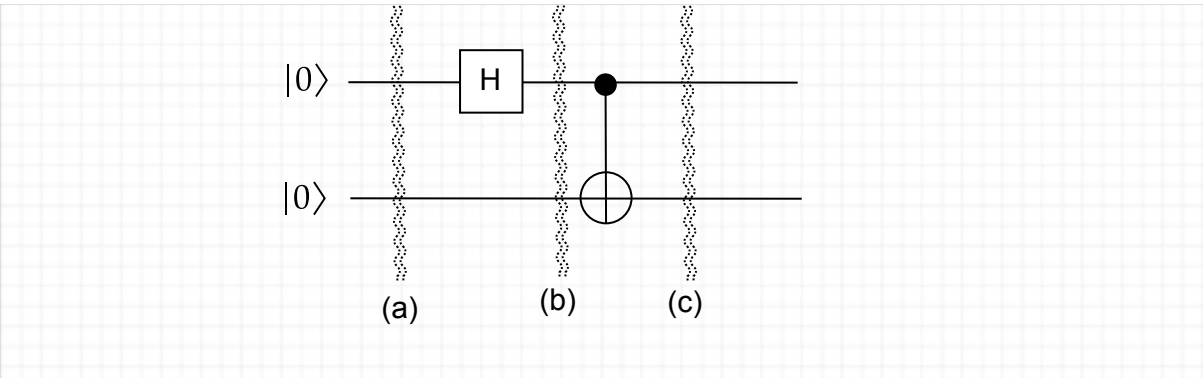
$|1\rangle \otimes |y\rangle \otimes |z\rangle \mapsto |1\rangle \otimes |y\rangle \otimes |z\rangle$

On a generic basis vector:

$|x\rangle \otimes |y\rangle \otimes |z\rangle \mapsto |x\rangle \otimes |y\rangle \otimes |((1 \oplus x) \cdot y) \oplus z\rangle$

Other example

Let us compute the result of the following circuit:



Recall that $H : |0\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ et $H : |1\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

So $H : |x\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle + (-1)^x \cdot |1\rangle)$

In (a) : $|0\rangle \otimes |0\rangle$

We apply $H \otimes Id_1$

So in (b) : $(H|0\rangle) \otimes (Id|0\rangle)$

$$= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle$$

$$= \frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |0\rangle)$$

The apply CNOT with 1st qubit as control qubit

So in (c) :

$$CNOT\left(\frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |0\rangle)\right)$$

$$= \frac{1}{\sqrt{2}}\left(CNOT(|0\rangle \otimes |0\rangle) + CNOT(|1\rangle \otimes |0\rangle)\right)$$

$$= \frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle)$$

$$= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Which is one of Bell's basis vector