Simulation of the superposition of multiple temporal gates orders in quantum circuits

Océane Koska

May 2021

Abstract

These last decades a huge interest has grown in quantum computers as they might provide important advantages in computation, especially because they exploit quantum superposition. Recently, the question of superposing not only quantum systems but temporal orders of gates in quantum circuits has been raised. In particular it might bring an advantage in query complexity (i.e. number of queries to gates). Recently a general definition of Quantum Circuits with Quantum Control of causal order (QC-QCs) was given. However, it is not yet known what kind of advantages QC-QCs can provide. To address this, we look at how to simulate this new kind of circuit with standard fixed order circuits to determine either they provide an interesting advantage in query complexity or not. We show that QC-QCs can be simulated with quantum circuits with fixed order of gates and we give a general method to find a such simulation. Moreover we apply this method to the $N$-Switch which provides a superposition of permutations of $N$ gates orders. The query complexity with the simulating circuit is $O(N^2)$ while the QC-QCs lead to a query complexity in $O(N)$.

1 Introduction

Quantum computing is a new approach to computation making use of quantum mechanical systems and quantum properties such as superposition and entanglement. It promises many possible advantages in computation. Quantum computers would be able to solve several computational problem drastically faster than classical ones. The idea of quantum computing appears in the 1980’s with physicist Richard Feynman. He raised the hypothesis to create a computer governed by quantum laws \cite{9}. Then in 1985 David Deutsch theoretically showed the possibility to create a Turing machine using quantum systems: quantum computing was born \cite{7}. In the following decades many physicists found algorithms exploiting quantum computers properties \cite{10}. For instance Peter Shor demonstrated the existence of a quantum algorithm to find the prime factors of an integer in a more efficient way than any known classical algorithm \cite{15}.

The standard model of quantum computing generalises the classical circuit model of computation, applying quantum gates in a circuit structure to quantum bits (qubits) \cite{18}. Recently, there has been interest in considering new types of quantum computations in which the order quantum gates are applied can be placed in quantum superpositions \cite{3, 8, 14}. Thus, it is not only a superposition of states but a superposition of circuits we can control. it is known that this can provide some advantages in specific information-processing tasks, like in phase-estimation problems \cite{2, 17}, but more general results and bounds are not yet known. Here we study this question by looking at how to simulate a class of such circuits – Quantum Circuits with Quantum Control of causal order (QC-QCs) \cite{19} –
with standard quantum circuits. The goal is to bound the number of queries in a QC-QC simulation with a standard quantum circuit without superpositions of gate orders.

2 Preliminaries

2.1 Quantum computing basics

In a classical computer there are bits. These bits are in a classical state, either 0 or 1. In a quantum computer there are also bits but they are in a quantum state, they are called qubits. A qubit, as a quantum system, can be in a superposition of \(|0\rangle\) and \(|1\rangle\) (this notation with “ket” |·⟩ is used for quantum states vectors):

\[ |\psi⟩ = \alpha |0⟩ + \beta |1⟩, \]

where \(\alpha, \beta \in \mathbb{C}, |\alpha|^2 + |\beta|^2 = 1.\) In this formula \(|\alpha|^2\) is the probability to see the qubit projected in the state \(|0⟩\) and \(|\beta|^2\) for \(|1⟩\) after measurement [12].

However, to perform computation we need of more than one qubit. Let us take two arbitrary Hilbert spaces \(H^A\) and \(H^B\) with respective given basis \((|i⟩^A)\) and \((|j⟩^B)\). Any state in \(H^A \otimes H^B\) can be written:

\[ |\psi⟩^{AB} = \sum_{i,j} \psi_{i,j} (|i⟩^A \otimes |j⟩^B). \]

Nevertheless, if there exist vectors \((\psi_i)\) and \((\psi_j)\) such that \(\forall i,j, \psi_{i,j} = \psi_i \psi_j\), then we can separate the state of the composite system in \(|\psi⟩^A \otimes |\psi⟩^B\) where \(|\psi⟩^A = \sum_i \psi_i |i⟩^A\) and \(|\psi⟩^B = \sum_j \psi_j |j⟩^A\). Here, states are separable, we call them product states. If this kind of separation is not possible the state is called entangled. When many qubits are entangled it is not possible anymore to consider them individually; we have to consider the whole entangled system [16].

In order to compute with qubits we have to perform operations on them. As we do in the classical computation, we can build quantum circuits with quantum gates. An important difference between a classical gate and a quantum gate is that the quantum gate has to be unitary [18]. An operator \(U\) is said unitary if \(UU^\dagger = U^\dagger U = I,\) where \(U^\dagger\) is the adjoint of \(U\) (in a matrix representation the adjoint of a matrix is the conjugate transpose of this matrix). For example a quantum NOT gate exists and this is its matrix representation:

\[ NOT = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \]

If this gates is applied to a qubit in the state \(\alpha |0⟩ + \beta |1⟩\) then the output will be \(\beta |0⟩ + \alpha |1⟩\). There also exist multi-qubits gates which are applied to several qubits. For example we can build the controlled NOT gate and applied it to a system of 2 qubits. The NOT operation is applied to the second qubit only if the first qubit is in the state \(|1⟩\):

\[ CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \]

We can generalise all this information to higher dimensional systems and create circuits by assembling several gates [12] [11]. An example of a simple quantum circuit is shown in Figure 1.
2.2 Quantum circuits with quantum control of causal order

Quantum circuits with quantum control of causal order can be seen as an extension of the concept of quantum circuits. In a standard quantum circuit the order of gates is totally fixed like in a classical circuit. The main property of QC-QCs is the superposition of different orders of gates; or more generally, their order can be controlled by another quantum system. As a result, the “causal order” between gates might be indefinite, just as a qubit in a superposition is in an indefinite classical state. A such superposition of gates order might be able to provide many advantages in computation; this question is actively being explored. In particular, it might reduce the query complexity (i.e. number of queries to some particular gates or “oracles”) in comparison with usual quantum circuits with fixed structures performing the same computation. For example, we could get a quadratic advantage in solving phase-estimation problem [2, 17].

2.2.1 The first historical example: The 2-switch

The 2-Switch is an easy example of QC-QC to start with. It was the first example of a circuit with indefinite causal order, before the notion of a QC-QC was properly formalised. Indeed, it produced only a superposition of orders of 2 gates [5]. Let us take 2 arbitrary quantum gates A and B.
The 2-switch uses two quantum systems: a target system in a Hilbert space $\mathcal{H}_t$, which the gates are applied to, and a control qubit in $\mathcal{H}_c$ that is used to control the order of the gates (see Figure 2). The aim is to create a quantum circuit producing the superposition of the paths AB and BA. Therefore the target system would simultaneously undergo both of the permutations of A and B depending on a control system. For example, if the control system is in the state $|0\rangle_c$ one can arbitrary decide that the chosen path is A then B and, on the contrary, if the control system is in the state $|1\rangle_c$ then it is B then A. Therefore, with a given target system state $|\psi\rangle_t$ and a given control system state $\alpha|0\rangle_c + \beta|1\rangle_c$, the 2-switch leads to the following output:

$$\alpha BA |\psi\rangle_t |0\rangle_c + \beta AB |\psi\rangle_t |1\rangle_c.$$

(1)

Here the superposition is obtained with only 2 queries to the gates A and B, one query to each gate \[3, 6\]. With a quantum circuit with fixed order of gates it is not possible to obtain the same output as in Eq. (1) with only two queries \[6\]. Instead, one needs at least 3 queries; a circuit simulating the 2-switch with 3 queries is shown in Figure 3.

Figure 3: Circuit simulating a 2-switch with fixed order of gates – Here there are 3 systems: the target system $t$, the control system $c$ and a buffer system called $tmp$. To perform all the possible permutations of the gates A and B (i.e. AB and BA) we can use a chain of gates containing all of them (here we choose ABA). Then we use controlled swaps which swap the both systems (with the crosses) if the control qubit (represented by the black dot) is in the state $|1\rangle_c$. This controlled swaps allow the target and the buffer system to be interchanged at the correct moment to apply the given permutation to the target.

For example, it was shown to provide an advantage in determining whether two operators commute or anti-commute \[4\]. In fact, with a 2-switch it is possible to solve this problem in only one query to each gate. This was experimentally implemented with an average success rate of 97.30% \[14\]. This raises hope in future applications of superposition of causal order of gates in quantum circuits with higher dimensions and higher numbers of gates.

2.2.2 The $N$-switch

The $N$-switch is a generalisation of the 2-switch to $N$ arbitrary quantum gates. The aim is to use a quantum control to control between any of the $N!$ permutations \[13\]. Let us consider a set $\mathcal{U} = \{U_0, \ldots, U_{N-1}\}$ of $N$ unitary operators. We denote $\Pi = \{\pi_1, \ldots, \pi_{N!}\}$ the set containing all the possible permutations of gates. We have a target system $t$ on $\mathcal{H}_t$ and an $N!$ dimensional control system $c$ on $\mathcal{H}_c$ that specifies the permutations to apply. The $N$-switch is technically a higher-order transformation: it maps the unitaries into a new unitary, $S_N(\mathcal{U})$:

$$S_N(\mathcal{U}) |\pi_i\rangle_c |\psi\rangle_t = |\pi_i\rangle_c U_{\pi_i(N-1)} \cdots U_{\pi_i(0)} |\psi\rangle_t,$$

(2)

where $|\psi\rangle_t$ is an arbitrary state in $\mathcal{H}_t$. The $N$-switch can be use to solve several problem like the Fourier and Hadamard promise problems introduced in \[2, 17\]. To apply a $N$-switch, we only use $N$ total queries to the unitaries while to simulate it and perform the transformation with a fixed order
quantum circuit one need $O(N^2)$ queries, so quadratically higher \cite{8}. This advantage in queries has led to an increased interest in the superposition of causal orders in quantum circuits and new types of quantum circuits with these properties have been defined.

### 2.2.3 QC-QCs

In \cite{19} a generalisation was proposed in an effort to try and formalise the most general form of quantum control of gate order. Indeed, it is possible to create quantum circuits which provide a superposition of causal gates order and performs intermediate other operations between gates.

First, let us give a definition of a QC-QC. A QC-QC is a quantum circuit with quantum control of causal order, that means a quantum system control the gates order and the intermediate operation applied to the target system and other internal systems. We can partition the circuit in several operative parts as shown in Figure 4.

![Figure 4: Schema of a QC-QC. The input of the circuit is the quantum system P (P for Past) and the output is the system F (F for future). In the circuit, new systems are added: $C_n$ is the control system which encodes the next gate chosen and the unordered set of gates already applied, $\alpha$ is an ancillary system that provides a quantum memory for the circuit. At each step of the circuit, the system encounters a $V_i$ block. These $V_i$ blocks consist of operations $V^{k_{n+1}}_{K_{n-1}, k_n}$ that are controlled by the control system. $V^{k_{n+1}}_{K_{n-1}, k_n}$ depends on the previously applied gates $K_{n-1}$ and just applied gate $k_n$, as well as the operation to be applied next, $k_{n+1}$ (which may be in a superposition). Moreover the control $c$ is updated to contain the previous gates and the next gate applied (i.e. $k_{n+1}$). Then the target system undergoes the gate given by the control system in the block $A$; since the control system will generally be in a superposition of different states, a superposition of different operations will generally be applied.](image)

Like the $N$-switch, QC-QCs are higher order operations that transform the gates $\{A_1, \ldots, A_N\}$ to some unitary $M(A_1, \ldots, A_N)$. They are a generalisation of quantum circuits transforming $N$ unknown operations $A_i$ into a new one. To simplify we assume that the $A_k$ input spaces are all isomorphic and likewise for the output spaces (without loss of generality). For all $0 \leq n \leq N - 1$,

$$A^I_n : \mathcal{H}^{A^I_n} \text{ (input space of the gate } A_n)$$

$$A^O_n : \mathcal{H}^{A^O_n} \text{ (output space of the gate } A_n).$$

As a consequence of this, the target system always has the same dimension, the one given by the gates. Finally, there are the ancillary systems. We can assume without loss of generality that they all have the same dimension too. For all $0 \leq n \leq N - 1$, at the $t_n$ time slot,

$$\alpha_n : \mathcal{H}^{\alpha_n}.$$
Then, in a general QC-QC there is a control system in which we encode the set of gates already applied, \( K_{n-1} \), which is a subset of \( n - 1 \) elements of \( \mathcal{N} = \{0, \ldots, N - 1\} \) and the current one, \( k_n \). This control system after \( V_n \) is a state in the Hilbert space \( \mathcal{H}^{C_{n}} \) with basis states \( |K_{n-1}, k_n\rangle_{C_{n}} \). It thus stores the unordered set of previously applied operations, and the operation to be applied at time step \( n \). This is important since, to form a valid circuit, every operation should be applied exactly once at the end of the computation.

The QC-QC is thus composed of operations \( V_i \), which are formally isometries, from the space \( \mathcal{H}^{A_{n-1}}_{\alpha_{i-1}} \odot H_{C_{i-1}} \) to \( \mathcal{H}^{A_{n}}_{\alpha_i} \odot H_{C_{i}} \), except for the first one (the input space is \( \mathcal{H}^{F} \)) and the last one (the output space is \( \mathcal{H}^{F} \otimes \mathcal{H}^{\alpha_F} \)).

\[
V_1 = \sum_{k_1} V_{\emptyset, k_1}^{\to k_1} \otimes |k_1\rangle_{C_1} : \mathcal{H}^{F} \rightarrow \mathcal{H}^{A_1}_{\alpha_1} \odot H_{C_1} \tag{3}
\]

\[
V_{n+1} = \sum_{K_{n-1}, k_n, k_{n+1}} V_{K_{n-1}, k_n}^{\rightarrow k_{n+1}} \otimes |K_{n-1} \cup k_n, k_{n+1}\rangle_{C_{n+1}} \langle K_{n-1}, k_n|_{C_n} \quad : \mathcal{H}^{A_n}_{\alpha_n} \otimes H_{C_n} \rightarrow \mathcal{H}^{A_{n+1}}_{\alpha_{n+1}} \odot H_{C_{n+1}} \tag{4}
\]

\[
V_{N+1} = \sum_{K_{N-1}, k_N} V_{K_{N-1}, k_N}^{\rightarrow F} \otimes |K_{N-1}, k_N|_{C_N} \quad : \mathcal{H}^{A_N}_{\alpha_N} \otimes H_{C_N} \rightarrow \mathcal{H}^{F}_{\alpha_F} \tag{5}
\]

When a QC-QC is applied to an initial state \( |\psi\rangle_{F} \), the output is the following state in the spaces \( \mathcal{H}^{F} \otimes \mathcal{H}^{\alpha_F} \):

\[
\sum_{(k_1, \ldots, k_N)} V_{\{k_1, \ldots, k_{N-1}\}, k_N}^{\rightarrow F} (A_{k_N} \otimes 1^{\alpha_N}) \cdots V_{\emptyset, k_1}^{\rightarrow k_1} (A_{k_1} \otimes 1^{\alpha_1}) V_{\emptyset, \emptyset}^{\rightarrow k_1} |\psi\rangle \tag{6}
\]

3 Simulating QC-QCs

3.1 The simulation model

To bound how much of an advantage we can hope to have with a QC-QC we need to simulate them with usual quantum circuits with fixed order. The aim of a such simulation is to find out if a quantum circuit without superposition of gates orders reaches the efficiency of a QC-QC. Thus, we have to "translate" the QC-QC model into a circuit with fixed order of gates one that performs exactly the same operation.

To simulate a QC-QC, we need to describe a quantum circuit, not necessarily using exactly the same Hilbert spaces, that reproduces the action of the QC-QC. To do this, we will show how, from a given QC-QC, to define a circuit simulating it. Given a QC-QC, our circuit will be defined on the following systems:

- A target system \( \mathcal{H}^t \) isomorphic to all the \( A_k^t \) and \( A_{k}^O \) of the QC-QC;
- \( N \) additional working target systems \( \mathcal{H}^{t_0}, \ldots, \mathcal{H}^{t_{N-1}} \) each isomorphic to \( \mathcal{H}^t \);
- An ancillary system \( \mathcal{H}^{\alpha} \) that is isomorphic to all the \( \mathcal{H}^{\alpha_n} \) of the QC-QC;
- A control system \( |K\rangle_{C_H} \otimes |k\rangle_c \), where \( \mathcal{H}^c \) is \( N \)-dimensional, and \( \mathcal{H}^{C_H} := \mathcal{H}^{C_1} \otimes \cdots \otimes \mathcal{H}^{C_N} \) (where all the \( C_i \) spaces are qubits) so that the basis states \( |K\rangle_{C_H} \) of \( C_H \) can be encoded in binary as \( |K\rangle = |K^{(1)}\rangle \otimes \cdots \otimes |K^{(N)}\rangle \) and \( |K^{(i)}\rangle = |1\rangle \) if \( k_i \in K \) and \( |0\rangle \) otherwise.

6
Figure 5: Schema of a simulated QC-QC. In the simulation there are several inputs in the circuits: the target system \( t \), additional working target systems \( t_i \), the ancillary \( \alpha \), the control history \( C_H \) which keeps in memory the gates already applied and the control system \( c \) which gives the following gate to apply. There are still different blocks: each \( V_i \) is replaced by a unitary operator \( U_i \). The \( \tilde{A}' \) parts are composed of controlled swap which enables the target system to undergo the gate given by the control system.

Now, as the input of the circuit is given in \( t \); we choose the other systems to have the given initial states so that the overall initial state is the following:

\[
\forall \psi, |\psi(0)\rangle = |\psi\rangle^t \otimes |0\rangle^\alpha \otimes |0\rangle^C_H \otimes |0\rangle^N \otimes |0\rangle^{t_i}. \tag{7}
\]

Then, since the \( V_i \) operations are isometries and defined on the spaces of a QC-QC and not our simulation, we cannot directly use them in our simulation. Given a QC-QC with operations \( V_i \) as defined in Eqs. \( \text{(3), (4) and (5)} \) we will define new unitary operations \( U_i \) used in our simulation in two steps. First, we translate the QC-QC isometries into isometries \( \tilde{V}_1, \tilde{V}_{n+1}, \tilde{V}_{N+1} \) on the correct spaces (and with the correct encoding) we defined. Then, we take a unitary extension of these isometries that gives the correct action when the extra spaces \( t_i \) and \( \alpha \) have the correct inputs we specified above.

For the first step, we take the QC-QC isometries and redefine them replacing \( \alpha_n \) by \( \alpha \) and \( A_I^n, A_O^n \) by \( t \). This is possible since all these spaces can, as discussed above, be assumed to be isomorphic. More formally, we define \( \tilde{V}_1 \) from \( V_1 \):

\[
\tilde{V}_1 = \sum_{k_1} \tilde{V}_{0,0}^{-k_1} \otimes |0\rangle^{C_H} \otimes |k_1\rangle^c : \mathcal{H}^P \to \mathcal{H}^{t_\alpha C_H c},
\]

where \( \tilde{V}_{0,0}^{-k_1} : \mathcal{H}^P \to \mathcal{H}^{t\alpha} \) is the same operation as \( V_{0,0}^{-k_1} \) just with the appropriate substitutions on the Hilbert spaces (this can be formally defined following the article \[19\]).
Then we define $\tilde{V}_{n+1}$ from $V_{n+1}$ for all $1 \leq n \leq N - 1$:

$$\tilde{V}_{n+1} = \sum_{K_{n-1}, k_n} \tilde{V}_{K_{n-1},k_n}^{\rightarrow k_{n+1}} \otimes |K_{n-1} \cup k_n \rangle \langle K_{n-1}|^{CH} \otimes |k_{n+1}\rangle \langle k_n|^{c}$$

$$: \mathcal{H}^{t\alpha CHc} \rightarrow \mathcal{H}^{t\alpha CHc},$$

where $\tilde{V}_{K_{n-1},k_n}^{\rightarrow k_{n+1}} : \mathcal{H}^{t\alpha} \rightarrow \mathcal{H}^{t\alpha}$ is the same operation as $V_{K_{n-1},k_n}^{\rightarrow k_{n+1}}$ just with the appropriate substitutions on the Hilbert spaces. The index expression bellow the sum should be understood as $K_{n-1} = \{k_1, \cdots, k_{n-1}\}$, $k_n, k_{n+1} \notin K_{n-1}$ and $k_n \neq k_{n+1}$ (here and throughout). Finally there is $\tilde{V}_{N+1}$ that have to be defined from $V_{N+1}$:

$$\tilde{V}_{N+1} = \sum_{K_{N-1}, k_N} \tilde{V}_{K_{N-1},k_N}^{\rightarrow F} \otimes |K_{N-1} \cup k_N \rangle \langle K_{N-1}|^{CH} \otimes |0\rangle \langle 0|^{c}$$

$$: \mathcal{H}^{t\alpha CHc} \rightarrow \mathcal{H}^{F\alpha F},$$

where $\tilde{V}_{K_{N-1},k_N}^{\rightarrow F} : \mathcal{H}^{t\alpha} \rightarrow \mathcal{H}^{F}$ is the same operation as $V_{K_{N-1},k_N}^{\rightarrow F}$ just with the appropriate substitutions on the Hilbert spaces. We notice that $K_{N-1} \cup k_N = \{1, \ldots, N\}$, since the notation above assumes $k_N \notin K_{N-1}$.

We now need to complete these isometries into unitaries to form a valid quantum circuit. However, it is well known that any isometry permits a (generally not unique) unitary extension. We thus take $U_i$ to be such an extension satisfying Eqs. (8), (9) and (10), which indeed is always possible. Since the spaces $\alpha, CH, c$ are initialised in these states in our circuit (7), this ensures the unitary extension has the same action as the isometry for any input state $|\psi\rangle$ as the QC-QC it simulates.

$$U_1 : U_1(|\psi\rangle^t \otimes |0\rangle^\alpha \otimes |0\rangle^{CH} \otimes |0\rangle^c) = \tilde{V}_1|\psi\rangle^t. \quad (8)$$

Then, for all $1 \leq n \leq N$, keeping the idea that $K_{n-1} = \{k_1, \cdots, k_{n-1}\}, k_n \notin K_{n-1}$, we can take such a unitary $U_{n+1}$ from $\tilde{V}_{n+1}$:

$$U_{n+1} : U_{n+1}(|\psi\rangle^{t\alpha} \otimes |K_{n-1}|^{CH} \otimes |k_n\rangle^c) = \tilde{V}_{n+1}(|\psi\rangle^{t\alpha} \otimes |K_{n-1}|^{CH} \otimes |k_n\rangle^c) \quad (9)$$

Finally there is $U_{N+1}$:

$$U_{N+1} : U_{N+1}(|\psi\rangle^{t\alpha} \otimes |K_{N-1}|^{CH} \otimes |k_N\rangle^c) = \tilde{V}_{N+1}(|\psi\rangle^{t\alpha} \otimes |K_{N-1}|^{CH} \otimes |k_N\rangle^c) \quad (10)$$

So after choosing the $U_i$, we need to define the $\tilde{A}'$ part in the simulation. At each time step, the $\tilde{A}'$ swaps the target system with the correct $t_i$ and apply the correct gate $A_i$, which is given by the control system, to the target system. But before defining $\tilde{A}'$ we need to introduce a new unitary operator. This operator is a generalisation for higher dimensions of the 2-dimensional SWAP gate which swaps the states of two $d$-dimensional quantum systems:

$$SWAP(a, b) = \sum_{i,j=0}^{d-1} |i\rangle^{a} \otimes |j\rangle^{b} \otimes |i\rangle^{a} \otimes |j\rangle^{b}.$$

Indeed we can use a multiple controlled swap which swaps the target system $t$ with the $t_i$ system corresponding to the gate to be applied according to the control system $c$:

$$\tilde{A}' = \left( \sum_{k=0}^{N-1} |k\rangle^c \otimes SWAP(t, t_k) \right) \left( \sum_{k=0}^{N-1} A_k \otimes |k\rangle^c \otimes SWAP(t, t_k) \right). \quad (11)$$

To have a global vision of the simulation model of a QC-QC we can sum up the previous ideas in a schema like the one given in the figure\[2\]. To summarise, the operations in the simulated circuit are
defined as follows:

\[
U_1 : U_1(|\psi\rangle^t \otimes |0\rangle^a \otimes |0\rangle^{C_H} \otimes |0\rangle^c) = \tilde{V}_1|\psi\rangle^t
\]

\[
\forall 1 \leq n \leq N - 1, U_{n+1} : U_{n+1}(|\psi\rangle^{t_n} \otimes |K_{n-1}\rangle^{C_H} \otimes |k_n\rangle^c) = \tilde{V}_{n+1}(|\psi\rangle^{t_n} \otimes |K_{n-1}\rangle^{C_H} \otimes |k_n\rangle^c)
\]

\[
U_{N+1} : U_{N+1}(|\psi\rangle^{t_N} \otimes |K_{N-1}\rangle^{C_H} \otimes |k_N\rangle^c) = \tilde{V}_{N+1}(|\psi\rangle^{t_N} \otimes |K_{N-1}\rangle^{C_H} \otimes |k_N\rangle^c)
\]

\[
\tilde{A}' = \left( \sum_{k=0}^{N-1} |k\rangle \langle k|^c \otimes SWAP(t, t_k) \right) \left( \sum_{k=0}^{N-1} |k\rangle \langle k|^c \otimes SWAP(t, t_k) \right).
\]

### 3.2 Correctness of the simulation model

Now we have all defined the simulation circuit, we need to show that the simulation is indeed correct. To do this, we will show that the final state of the target and ancillary systems is the same as in the QC-QC circuit simulates. Firstly:

- we apply \( U_1 \) on \(|\psi(0)\rangle\) to obtain \(|\psi(1)\rangle = U_1|\psi(0)\rangle\):

\[
U_1|\psi(0)\rangle = \tilde{V}_1|\psi\rangle^t \otimes |0\rangle^t_1 = \sum_{k_1} V^{t_1}_{0,0} |\psi\rangle^t \otimes |0\rangle^{C_H} \otimes |k_1\rangle^c \otimes |0\rangle^t_1 = |\psi(1)\rangle;
\]

- we apply \( \tilde{A}' \) on \(|\psi(1)\rangle\)

\[
\tilde{A}'|\psi(1)\rangle = \sum_{k_1} \left( (A_{k_1} \otimes I^a) V^{t_1}_{0,0} |\psi\rangle^t \otimes |0\rangle^{C_H} \otimes |k_1\rangle^c \otimes |0\rangle^t_1 \otimes \bigotimes_{i \neq k_1} A_i|0\rangle^{t_i} \right).
\]

where \( V^{t_1}_{0,0} |\psi\rangle^t \in H^{t_1} \). Then, by recurrence on \( n \):

- we apply \( U_{n+1} \) on \( \tilde{A}'|\psi(n)\rangle \) to obtain \(|\psi(n+1)\rangle = U_{n+1}\tilde{A}'|\psi(n)\rangle\)

\[
U_{n+1}\tilde{A}'|\psi(n)\rangle = \sum_{K_{n-1}, k_n, k_{n+1}} \left( V^{t_{n+1}}_{K_{n-1}, k_n} (A_{k_n} \otimes I^a) \cdots V^{t_1}_{0,0} |\psi\rangle^t \right)
\]

\[
\otimes |K_{n-1} \cup \{k_n\}\rangle^{C_H} \otimes |k_{n+1}\rangle^c \bigotimes_{i \in (K_{n-1} \cup \{k_n\})} A_i^n|0\rangle^{t_i} \bigotimes_{i \notin (K_{n-1} \cup \{k_n\})} A_i^n|0\rangle^{t_i};
\]

- we apply \( \tilde{A}' \) on \(|\psi(n+1)\rangle\)

\[
\tilde{A}'|\psi(n+1)\rangle = \sum_{K_{n-1}, k_n, k_{n+1}} \sum_{(k_1, \ldots, k_{n-1})} \left( (A_{k_{n+1}} \otimes I^a) V^{t_{n+1}}_{K_{n-1}, k_n} (A_{k_n} \otimes I^a) \cdots V^{t_1}_{0,0} |\psi\rangle^t \right)
\]

\[
\otimes |K_{n-1} \cup \{k_n\}\rangle^{C_H} \otimes |k_{n+1}\rangle^c \bigotimes_{i \in (K_{n-1} \cup \{k_n, k_{n+1}\})} A_i^n|0\rangle^{t_i} \bigotimes_{i \notin (K_{n-1} \cup \{k_n, k_{n+1}\})} A_i^n|0\rangle^{t_i}.
\]

And finally, for the last step:
we apply $U_{N+1}$ on $\tilde{A}\ket{\psi(N)}$

$$U_{N+1}\tilde{A}\ket{\psi(n)} = \left( \sum_{K_{N-1},k_N} \sum_{(k_1,\ldots,k_{N-1})} V^{\rightarrow F}_{K_{N-1},k_N} (A_{k_N} \otimes I^c) \cdots V^{\rightarrow k_1}_{0,0} \ket{\psi}^c \right)$$

$$\otimes \{|k_1,\ldots,k_N\}\rangle^C \otimes |0\rangle^c \bigotimes_{i=0}^{N-1} A_i^{N-1} |0\rangle^{t_i}.$$

Note that the system in the brackets is in $H^{t_0}$. Furthermore we can simplify this equation by noticing that the double sum can be reduced to a single sum over all orders $(k_1,\ldots,k_N)$:

$$U_{N+1}\tilde{A}\ket{\psi(n)} = \left( \sum_{(k_1,\ldots,k_N)} V^{\rightarrow F}_{k_1,\ldots,k_{N-1}} (A_{k_N} \otimes I^c) \cdots V^{\rightarrow k_1}_{0,0} \ket{\psi}^c \right)$$

$$\otimes \{|k_1,\ldots,k_N\}\rangle^C \otimes |0\rangle^c \bigotimes_{i=0}^{N-1} A_i^{N-1} |0\rangle^{t_i}.$$

First, it is really important to notice that there is a factorization for $c$, $C_H$ and $t_i$ systems. Actually, for all given permutations of gates the control history system $C_H$ ends up in the state $\{|k_1,\ldots,k_N\}\rangle^C = |\{0,\ldots,N-1\}\rangle^C$, the control system in the state $|0\rangle^c$ and each $t_i$ is in the state $A_i^{N-1} |0\rangle^{t_i}$. Then at the end of the circuit the target/ancilla system’s state $t_0$ in the bracket has exactly the same form as the output of the QC-QC given by Eq. (6), which shows that the circuit indeed correctly simulates the QC-QC.

Therefore the simulation model works and we show that any QC-QC can be translated into a circuit without superposition of gates orders. To concretely realise this simulation, we need to find out a unitary extension of each part given in its definition. As there exist several unitary extensions of an isometry, we can create different circuits with fixed order of gates to simulate exactly the same QC-QC. With the method given here the query complexity of the simulating circuit is $O(N^2)$ if the number of gates we want to permute is $N$; while for a QC-QC this complexity is $O(N)$. Thus, the simulation has a quadratic overhead.

4 Application to the $N$-Switch

4.1 The $N$-switch as a QC-QC

The $N$-switch is the simplest non-trivial example with indefinite causal order we can imagine. It provides a superposition of several permutations of $N$ given gates. As an example, we will here describe it as a QC-QC and then show how the simulation procedure defined above applies to it. Let us take $2^{d_t}$-dimensional Hilbert spaces $H^{P_t}$ and $H^{F_t}$ representing the past and the future of the target system, $N!$-dimensional isomorphic Hilbert spaces $H^{P_c}$ and $H^{F_c}$ representing the past and the future of the control system and $H^{c_0}$ for the ancillary. Let’s consider the orthonormal bases $\Pi = \{ \ket{\pi} \}$ of the permutations $(\pi = (\pi(1),\ldots,\pi(N)))$ of the $N$ chosen gates in $A = \{A_0,\ldots,A_{N-1}\}$. It is possible to find a simulation of the $N$-switch using the general model.

Let us start by defining the first block $V_1$. In this block $P^t$ is directly transmitted to the first gate $A_{k_1}$, for all $k_1 \in \{0,\ldots,N-1\}$. As a reminder, the value of $k_1$ is given by the control system. Thus,
one has to consider all the permutation starting with the gate $A_{k_1}$ and the $V_1$’s expression is:

$$V_1 = \sum_{k_1=0}^{N-1} V_{0,0}^{-k_1} \otimes |\emptyset\rangle^{C_H} \otimes |k_1\rangle^c$$

$$= \sum_{k_1=0}^{N-1} \prod_{1 \leq i \leq k_1}^{A_{k_1}} \otimes \sum_{\pi(1)=k_1}^{\pi \text{ compatible}} \langle \pi |^{P_c} \otimes |\pi\rangle^{\alpha_1} \otimes |\emptyset\rangle^{C_H} \otimes |k_1\rangle^c. \tag{12}$$

Here, the only permutations kept have $A_{k_1}$ as the first gate. Then, there are intermediary blocks $V_2, \cdots, V_N$. In the same way, the target system is transmitted from gate to gate and only undergoes the correct gates given by the chosen permutations. However, it is really important to keep in memory the gates already chosen. Keeping in memory gates already chosen is necessary because to keep a quantum control of causal order one gates has to be applied once. Nevertheless it is not that simple. If one stores the gates after each $A$ block keeping the order then some interferences between paths will be lost. Thus, it is crucial to keep in memory the set of gates chosen but without storing their order. Let us write this set for the $V_n$ bloc $K_{n-1} = \{k_1, \cdots, k_{n-1}\}$. Thus for all $0 < n \leq N - 1$:

$$V_{n+1} = \sum_{K_{n-1},k_n,k_{n+1}} V_{K_{n-1},k_n}^{-k_{n+1}} \otimes |K_{n-1} \cup \{k_n\}|^{C_H} \otimes |k_{n+1}\rangle^{k_n^c}$$

$$= \sum_{K_{n-1},k_n,k_{n+1}} \prod_{1 \leq i \leq k_n}^{A_{k_n}} \otimes \sum_{\pi \text{ compatible}}^{\pi \text{ compatible}} \langle \pi |^{\alpha_n} \otimes |\pi\rangle^{\alpha_{n+1}} \otimes |K_{n-1} \cup \{k_n\}|^{C_H} \otimes |k_{n+1}\rangle^{k_n^c}, \tag{13}$$

where by compatible we mean that the permutation $\pi$ is compatible with the history and the given gates $A_{k_n}$ and $A_{k_{n+1}}$; that is to say a permutation is kept if its previous chosen gates constitute the set $K_{n-1}$, the current gate in the permutation is $A_{k_n}$ and the next one is $A_{k_{n+1}}$. Finally, there is the block $V_{N+1}$ that leads to the future $F$:

$$V_{N+1} = \sum_{K_{N-1},k_N} V_{K_{N-1},k_N}^{A_{k_N}} \otimes \langle K_{N-1}|^{C_H} \otimes \langle k_N|^{k_N^c}$$

$$= \sum_{K_{N-1},k_N} \prod_{1 \leq i \leq N}^{A_{k_N}} \otimes \sum_{\pi \text{ compatible}}^{\pi \text{ compatible}} \langle \pi |^{\alpha_N} \otimes |\pi\rangle^{F_c} \otimes \langle K_{N-1}|^{C_H} \otimes \langle k_N|^{k_N^c}. \tag{14}$$

### 4.2 Simulation of the N-switch

The aim is now to use the general model to simulate a specific circuit and see how to get from a QC-QC another circuit with a fixed order of gates. We choose to simulate a $N$-switch because it is the most simple QC-QC to understand. Indeed, as the only thing performed is applying a given permutation of several orders of given gates, the $N$-switch model as a QC-QC is not so difficult to define and to “translate” into a circuit with fixed order of gates. To make this simulation we need to specify precisely the unitary extensions of the isometries used. Then it is also important to show how to construct these unitaries explicitly, and ideally out of simple gates, to allows us to understand better the computation.

First we can remind the expression of $\hat{A}'$ given in Eq. [11] as this part is the same for every QC-QCs:

$$\hat{A}' = \left( \sum_{k=0}^{N-1} |k\rangle\langle k|^{c} \otimes SWAP(t,t_k) \right) \left( \bigotimes_{k=0}^{N-1} A_k \right) \left( \sum_{k=0}^{N-1} |k\rangle\langle k|^{c} \otimes SWAP(t,t_k) \right).$$

Now we have to define the $U_i$ blocks in the circuit. But before we need to introduce new unitaries we will use. Firstly, to update the control history system we need to perform a union with $K_{n-1}$ and
\( \{ k_n \} \). However the union is not a unitary operation thus we have to introduce another operator to make this update unitary and which reduces to the union on the initial states used in the simulation. We choose for this an exclusive union:

\[
ExUnion^{(K,k)} = \sum_K \sum_k |K \cup \{ k \} \rangle \langle K |^K \otimes |k\rangle^k,
\]

where \( K \cup \{ k \} = K \cup \{ k \} \) if \( \{ k \} \not\subset K \), else \( K \cup \{ k \} = K \setminus \{ k \} \). Then we need a unitary operator which copies the \( i^{th} \) element of the permutation \( \pi \) in the control system; principally to initialize the control system at the beginning and to clear it at the end. To keep the operator unitary we cannot just erase the control system information and copy the value into it. Therefore we perform a sum modulo \( N \) and if the control system contains \( 0 \) then it is equivalent to copy the value into it:

\[
\forall n \in \{ 1 \cdots N \}, \zeta^{(c,\alpha)}(n) = \sum_{\pi} |\pi\rangle |\pi\rangle^\alpha \otimes \sum_j |j + \pi(n)\rangle |j\rangle^c.
\]

We can use this operator to copy the given value and its adjoint to erase it (if we apply \( \zeta^{(c,\alpha)}(n) \) and then \( \zeta^{(c,\alpha)}(n)^\dagger \) we get the identity). Finally we need of an operator which enables to progress in the permutation from the \( i^{th} \) to the \( (i + 1)^{th} \) element:

\[
SHIFT^{(c,\alpha)} = \sum_{\pi} \sum_j |\pi\rangle \langle \pi|^{\alpha} \otimes |\pi(\pi^{-1}(j) + 1)\rangle |j\rangle^c,
\]

where \( \pi^{-1}(j) \) is equal to the index of \( j \in \{ 0, \ldots, N - 1 \} \) in the permutation \( \pi \).

Now we can define the \( U_i \) blocks as unitary extensions of the \( \tilde{V}_i \). Let us start with \( U_1 \). We have:

\[
\tilde{V}_1 = \sum_{\pi} 1^f \otimes |\pi\rangle \langle \pi|^{\alpha} \otimes 1^{CH} \otimes |\pi(1)\rangle^c
\]

from which \( U_1 \) is defined as:

\[
U_1 = 1^t \otimes 1^{CH} \otimes \left( \sum_{\pi} |\pi\rangle \langle \pi|^{\alpha} \sum_i |\pi(1) + i\rangle |i\rangle^c \right) = 1^{tCH} \otimes \zeta^{(c,\alpha)}(1).
\]

(15)

As we consider only \( |0\rangle^c \) to be compatible with the \( \tilde{V}_1 \) subset definition \( U_1 \) is clearly an extension of \( \tilde{V}_1 \). Indeed, the sum on \( i \) has only one term which is \( |\pi(0 + 1)\rangle |0\rangle^c = |\pi(1)\rangle |0\rangle^c \). Then for all \( 1 \leq n \leq N - 1 \), we have:

\[
\tilde{V}_{n+1} = \sum_{\pi} 1^f \otimes |\pi\rangle \langle \pi|^{\alpha} \otimes |\{ \pi(1), \ldots, \pi(n) \}\rangle \langle \{ \pi(1), \ldots, \pi(n - 1) \}|^{CH} \otimes |\pi(n + 1)\rangle |\pi(n)\rangle^c
\]

from which \( U_{n+1} \) is defined as:

\[
U_{n+1} = \left( 1^{tCH} \otimes \left( \sum_{j,\pi} |\pi\rangle \langle \pi|^{\alpha} \otimes |\pi(\pi^{-1}(j) + 1)\rangle |j\rangle^c \right) \right) \left( 1^{ta} \otimes \sum_i |i\rangle |i\rangle^c \otimes \sum_{K \cup \{ 1 \}} |K \rangle \langle K|^{CH} \right)
\]

\[
= \left( 1^{tCH} \otimes SHIFT^{(c,\alpha)} \right) \left( 1^{ta} \otimes ExUnion^{(CH,\alpha)} \right).
\]

(16)

This time, we only consider cases with \( |\pi(n)\rangle \rangle \) and \( \pi(n) \not\subset K \) to fit the \( \tilde{V}_{n+1} \)’s subspace. By changing the previous equation following these conditions we notice that indeed \( U_{n+1} \) is an extension of \( \tilde{V}_{n+1} \). Finally we have to define \( U_{N+1} \) the last unitary of the circuit. We have:

\[
\tilde{V}_{N+1} = \sum_{\pi} 1^f \otimes |\pi\rangle \langle \pi|^{\alpha} \otimes |\{ \pi(1), \ldots, \pi(N) \}\rangle \langle \{ \pi(1), \ldots, \pi(N - 1) \}|^{CH} \otimes |0\rangle |\pi(N)\rangle^c.
\]
Then we defined $U_{N+1}$ from $\tilde{V}_{N+1}$ as:

$$U_{N+1} = \left(1^{tC_H} \otimes \left(\sum_{\pi} |\pi\rangle \langle \pi|^a \otimes |j\pi(N) + j|^c\right)\right) \left(1^{t\alpha} \otimes \sum_i |i\rangle \langle i|^c \otimes \sum_K |K\cap\{i\}\rangle \langle K|^C_H\right)$$

$$= \left(1^{tC_H} \otimes \zeta^{(\alpha,c)}(N)^t\right) \left(1^{t\alpha} \otimes ExUnion^{(C_H,c)}\right).$$  \hspace{1cm} (17)

This quantum circuit with fixed order of gates correctly simulates the $N$-switch since it is obtained from our general simulation method proven in the previous section by providing an explicit unitary extensions of the isometries used therein. In this simulation we can see that the number of queries to gates in $\mathcal{A}$ is $N^2$, while a $N$-switch with superposition of gate orders uses only $N$. Thus we find another way to simulate a $N$-switch with all these unitaries. Moreover, we can compare with some other simulations that have been given for the $N$-switch. For example, in [17] a simulation of a 4-switch is given with only 9 queries to gates. With our simulation the 4-switch requires 16 queries but we can adapt the simulation to any $N$ really easily while the other simulation requires to find a chain of gates containing all the possible permutations; so in this other simulation we need to reorganise everything from the gate disposition to the use of operators. Furthermore, it is important to see that in addition to the number of queries to the given gates we need to use several other operators to correctly control the $N$-switch.

\section{Conclusion and Discussion}

From the definition of a QC-QC we have created a method to simulate it with a fixed order of gates. It shows that any QC-QC can be simulated with a quantum circuit with fixed structure, which is not unique. With the method given here the query complexity of the simulating circuit is $O(N^2)$ for $N$ gates; while for a QC-QC this complexity is $O(N)$. Thus, the simulation has a quadratic overhead. This places an upper bound on any possible advantage using QC-QCs.

But we can ask if it is possible to simulate a QC-QC with less queries. In the method given here we arbitrarily decided that when we want to apply a gate to the target system all the other gates are also applied to buffer systems. Therefore, as we want to apply $N$ gates to the target system, the query complexity reaches $O(N^2)$. To reduce this overhead we might think that we can change the way we choose the gate applied to the target system. As we showed in figure 3 we can simulate a 2-switch with 3 gates instead of 4 by using a chain of gate containing all the possible permutations of the given gates (this type of chain is called super-permutation). We could try and adapt the simulation model to fit this idea by keeping only one buffer system and work with the smallest chain of gates containing all the permutations. However we would face two main issues. On one hand the control stores unordered sets instead of the chosen permutation; so it might be highly difficult to choose if the gate has to be applied or not at each step. On the other hand, the lower bounds of a super-permutation is $N! + (N-1)! + (N-2)! + N - 3$, so we keep an advantage only for $3 \leq N$ if we use super-permutations. Therefore it seems to not be possible to find a simulation circuit of a QC-QC providing a better query complexity than $O(N^2)$. Thus, given the current state of research the QC-QC provides a quadratic advantage in number of queries. However, it is important to contrast this advantage by reminding the creation of a superposition of quantum causal orders requires the quantum system dimension to grow exponentially with $N$, making it difficult to implement.

However we can ask if there is an interest in using QC-QCs instead of their simulation. The answer to this depends on whether we find interesting problems that saturate the bound, and thus get a quadratic speedup, or not. It has been shown that for problems such as phase estimation this bound can be saturated. If we consider the Hadamard promise problem, [17] which is a type of the phase estimation problem, for all the possible permutations of $N$ gates, the $N$-switch provides a solution in $N$ queries while the best known standard circuit requires $O(N^2)$ queries. [17]. Despite the fact that simulating the $N$-switch requires $\Omega(N^2)$ queries, it is not known that there is not another quantum
circuit that can obtain better asymptotic behaviour. The QC-QCs are interesting in solving problems
given that they provide a quadratic advantage in query complexity. Nevertheless, to implement QC-
QC in laboratory we need of quantum systems which grow exponentially with $N^{[17]}$; creating and
working with such big systems is really hard to reach.

6 Acknowledgements

Throughout the writing of this paper I have received a great deal of support and assistance.

I would first like to thank my supervisor, Alastair Abbott, for making this laboratory experience
possible. Your advice and valuable feedback helped me to discover the real work of a researcher.
Thank you.

Then I would thank Zoltán Szigeti for setting up this module. To me it was a huge opportunity
and an interesting introduction to laboratory research. Furthermore, thank you Karmijn and Marie
for sharing with me your own experiences in laboratory.

I would also like to thank my parents and my little brother for being there for me. It was warmly
appreciable to be taken home in car after my lab session while the snow was intensely falling.

In addition, I would like to thank my friends. Thank you Guillaume, Sylvain, Charlotte and
Solenne for always making me go forward even if you still try to figure out what I say when I talk
about quantum. Thank you Loïc for your sympathetic ear and your multiple proof-readings.

References

[1] Anonymous 4chan Poster, Robin Houston, Jay Pantone, and Vince Vatter. A lower bound on

[2] Mateus Araújo, Fabio Costa, and Časlav Brukner. Computational advantage from quantum-


computation with programmable connections between gates. Physics Letters A,

[7] David Deutsch. Quantum theory, the church–turing principle and the universal quantum com-
puter. Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences,

Lecture Notes in Computer Science Theory and Applications of Models of Computation, page


