# CPSC 4210 - Final Paper

R. Lowry, C. Rabl, R. Rana

## Abstract

Insert mathematical insight here.

# 1 Permutation Matrices

**Definition 1.** *A* **permutation matrix** $P$ *is an* $n \times n$ *matrix created by permuting the rows of the identity matrix* $I_n$*. It is the case that* $P * P^T = P^T * P = I$ *and that* $det(P) = 1$*.*

Rather than deal with extremely large and unwieldy unitary transformations all the time, we can use permutation matrices, as described by Williams [1999]. These are a powerful tool, since an $n \times n$ permutation matrix can be used to represent a $2^n \times 2^n$ unitary operation. Additionally, since permutation matrices are sparse, they can be computed with and stored more efficiently than full matrices. As an aside, some unitary matrices are also permutation matrices (such as the NOT gate), but the gates which are "true quantum primitives", as described in Lukac et al. [2003] are only unitary.

In brief, permutation matrices encode the rows of a circuit or gate's truth table. Given the truth table for a CNOT gate, for instance, it is quite simple to construct its permutation matrix: we begin by encoding the inputs and outputs of the gate as decimal numbers, and create a mapping between them. Then, we use this mapping to construct the permutation matrix, using the following rule:

$$P = [p_{ij}] \text{ where } p_{ij} = \begin{cases} 1 & \text{if } i = n \text{ and } j = M(n) \quad \forall n \in \mathbb{Z}_k \\ 0 & \text{otherwise} \end{cases}$$

In this case, $k = 2^w$ where $w$ is the "width" of the gate, or the number of inputs. Since CNOT has a width of 2, that means $k = 2^2 = 4$, in this case.

| a | b | a' | b' |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 |

$\rightarrow$

| $n$ | $M(n)$ |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 3 |
| 3 | 2 |

$\rightarrow \quad P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

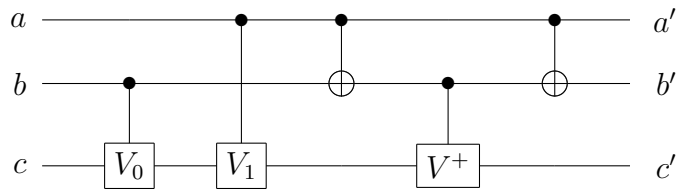
A useful property of permutation matrices is that they allow us to "compose" permutations. In order to do this, we use the following identity: $P_{\sigma \circ \pi} = P_\pi * P_\sigma$. Note that the order of the matrix multiplication matters, as matrices do not typically commute under multiplication. Having this composition operator makes it easy to represent cascades in a unique way. We can check that two cascades realize the same function if their output permutation matrices are identical. This provides circuit designers with an efficient way to "equate" cascades and determine which is "better" (although Rubin will explain this in his talk).

# 2 Quantum Cost

Since we can represent operations on qubits using unitary transformations (which conveniently correspond to exactly one quantum operation each), we can devise a metric called "quantum

cost" in order to determine whether the transformations we perform constitute an efficient synthesis of a given operation. In an NMR system, each electromagnetic pulse to which we subject a qubit has a cost: whether it is the amount of energy required to create the pulse, or the risk of the qubit decohering into a useless state (through vibrations, or other environmental perturbations), these factors may be treated as unitless "cost" variables which must be taken into account.

As quantum cost is a unitless quantity which corresponds directly to the number of unitary operations in a quantum circuit, it is a very useful metric for calculating the efficiency of an implementation of a circuit. In order to determine the quantum cost of a gate or cascade, we need to break it down into "quantum primitives" (unitary transformations). For instance, we can break down a 3-input Toffoli gate like so:




Of course, it is not immediately obvious why this construction gives us a Toffoli gate. Note that the $\sqrt{\text{NOT}}$ gates (and their Hermitian friend) do not get activated unless their control lines are 1.

So, if we pass $a = 0$ and $b = 0$ through our gate, $c$ remains unchanged, as do $a$ and $b$. If $a = 0$ and $b = 1$, then the gate that gets applied to $c$ will be $V_0 * V^+ = I$, which is the identity, so $c$ will be unchanged. If $a = 1$ and $b = 0$, then the gate that gets applied to $c$ will be $V_1 * V^+ = I$, so $c$ will be unchanged, and finally, if $a = 1$ and $b = 1$, $c$ will be inverted because the gate that gets applied will be $V_0 * V_1 = \text{NOT}$. And thus, we have shown that a 3-input Toffoli gate may be simulated by at least five quantum primitives, and so it has a quantum cost of 5. This result is due to DiVincenzo and Smolin [1994].

# References

David P. DiVincenzo and J. Smolin. Results on two-bit gate design for quantum computers. In *Physics and Computation, 1994. PhysComp '94, Proceedings., Workshop on*, pages 14–23, Nov 1994. doi: 10.1109/PHYCMP.1994.363704.

Martin Lukac, Marek Perkowski, Hilton Goi, Mikhail Pivtoraiko, Chung Hyo Yu, Kyusik Chung, Hyunkoo Jeech, Byung-Guk Kim, and Yong-Duk Kim. Evolutionary approach to quantum andreversible circuits synthesis. *Artif. Intell. Rev.*, 20(3-4):361–417, December 2003. ISSN 0269-2821. doi: 10.1023/B:AIRE.0000006605.86111.79. URL `http://dx.doi.org/10.1023/B:AIRE.0000006605.86111.79`.

C.P. Williams. *Quantum Computing and Quantum Communications: First NASA International Conference, QCQC '98, Palm Springs, California, USA, February 17-20, 1998, Selected Papers*. Lecture Notes in Computer Science. Springer, 1999. ISBN 9783540655145. URL `http://books.google.ca/books?id=4QuAhlwj2icC`.