Abstract. A programmable quantum processor is a fixed quantum gate array that has two quantum inputs – data and a program. A quantum program specifies a transformation that is applied to data. In this paper, we formally define equivalence of deterministic, probabilistic and approximative programmable quantum processors. The condition for equivalence for different types and classes of processors is discussed.

1 Introduction

Quantum computing [1] has become a challenging and very important branch of physics and computer science. The power of quantum computing lies in employing puzzling quantum phenomena and laws for performing computations. The laws of microscopic quantum world are different from the laws that rule the classical world. Quantum computers are intrinsically parallel. In only one computational step, a quantum computer can perform exponentially many calculations. Quantum systems can be entangled and entangled systems can instantly influence each other no matter how distant these systems are. There are other phenomena that can be utilized, in particular, for cryptographic tasks.

The most important achievements in the field of quantum computing are quantum polynomial-time algorithms for factorization of integers and computation of discrete logarithms [2] . These algorithms allow exponential speedup compared to the best-known classical algorithms. This could be a serious threat for many contemporary cryptosystems (*ie.* RSA). Quantum cryptography attempts to find more secure solutions. Several schemes for secure quantum key generation were developed [3].Security of these schemes does not rely on unproven computational assumptions as in the case of many classical cryptosystems; it relies on the laws of the Nature. Another interesting application is quantum teleportation of an unknown quantum state .

A programmable quantum processor could be the heart of a quantum computer. Nowadays, quantum circuits are usually designed to perform only one specific task, but it would be useful to have circuits that are more flexible and can perform a variety of functions. Devices are usually controlled externally using classical parameters. Contrary to that, a programmable quantum processor is a fixed device and it is programmable by states of a quantum system. It is a

quantum gate array which has two quantum inputs – data and a program. The program specifies a transformation which should be applied to data.

Programmable quantum processors can be also utilized as programmable measurement devices [4] or programmable state discriminators [5]. For a survey on programmable quantum processors, see [6].

This paper deals with equivalence of particular types and classes of programmable quantum processors and the corresponding equivalence conditions. Equivalence relates processors with respect to operations they implement. In case of probabilistic and approximative processors, success probability, respectively a precision parameter, has to be taken into account.

The paper is organized as follows. In Section 2, a basic model of deterministic processors is introduced. Universality is discussed and other types of processors are described in Section 3. In Section 4, we deal with equivalence, its formal definition and equivalence conditions. Finally, in Section 5, we give a short conclusion and discuss future work.

2 Programmable Quantum Processors

A programmable quantum processor [7] has two input registers – the data register and the program register. The state of the program register specifies an operation which we want to perform on the state stored in the data register. We define a programmable quantum processor as follows.

Definition 1. *Let $\mathcal{H}_d$ and $\mathcal{H}_p$ be two Hilbert spaces. A programmable quantum processor is a unitary operation G acting on the joint data-program space $\mathcal{H}_d \otimes \mathcal{H}_p$.*

Remark: the basic type of processors is usually referred to as deterministic.

As an example of a programmable quantum processor, let us consider a controlled-U gate (see Fig. 1). The control system is the program and the target is the data. Assuming both data and program register are 1-qubit registers, if the control qubit is in the state $|0\rangle$, nothing is applied to the data state. If the control qubit is in the state $|1\rangle$, a unitary operation U is applied to the data state. If the control qubit is in the superposition state $\alpha|0\rangle + \beta|1\rangle$, the following quantum map

$$\mathcal{E}(\rho) = |\alpha|^2\rho + |\beta|^2 U\rho U^\dagger$$

is applied to the data state (*eg.* we obtain a CNOT gate in case $U = \sigma_x$).

3 Universality and Nonideal Types of Processors

One of the most important results related to programmable quantum processors is the no-go theorem about universality of deterministic processors [7]. A processor that enables us to implement an arbitrary unitary transformation of a state stored in the data register must have infinite size of the program register. Therefore, even for a subset of all quantum operations, there is no processor of a

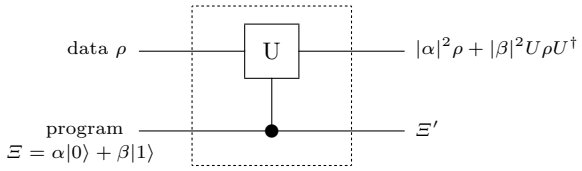

Fig. 1. A programmable quantum processor performing a controlled-U operation.

finite size. This justifies the study of different classes of programmable quantum processors and relations between them. Universality is possible only if we relax some of ideal conditions, *ie.* the desired data transformations are always realized perfectly. We have two basic options: probabilistic realization of quantum operations [7–9] or approximative realization of quantum operations [10, 11]:

Probabilistic programmable quantum processors implement the desired map only with a certain probability. Measurement of the program register can tell us whether the requested map was actually applied to data. A *probabilistic programmable quantum processor* is a programmable quantum processor G together with a quantum measurement M performed on the program register at the output of the processor G. This measurement can be

(a) fixed (measurement-assisted processors), or

(b) the choice of a measurement is part of quantum programming.

In the following, we consider only the latter case .

In the approximative case, desired maps are not performed perfectly, only up to some level of approximation. Indeed, *approximative quantum processors* are deterministic processors, however, we study similarity between the desired operations and the actually implemented maps. The process fidelity [12] can be used to measure the distance between two operations. Completely positive maps are compared using the density matrices which are associated to these maps via the Jamiolkowski isomorphism. For two general maps $\mathcal{E}$ and $\mathcal{F}$ and the corresponding density matrices $\rho_\mathcal{E}$ and $\rho_\mathcal{F}$ obtained by this isomorphism, the process fidelity is given as

$$F(\rho_\mathcal{E}, \rho_\mathcal{F}) = \left[\mathrm{Tr}\sqrt{\sqrt{\rho_\mathcal{E}}\rho_\mathcal{F}\sqrt{\rho_\mathcal{E}}}\right]^2.$$

Distance between quantum operations $\mathcal{E}$ and $\mathcal{F}$ is $D(\mathcal{E}, \mathcal{F}) = 1 - F(\rho_\mathcal{E}, \rho_\mathcal{F})$.

4 Equivalence of Processors

In this section, we introduce the notion of equivalence of programmable quantum processors for all three types of processors – deterministic, probabilistic and approximative ones.

4.1 Deterministic Programmable Quantum Processors

We are interested in the transformation applied to data which depends on the state of the program register:

$$\mathcal{E}_\xi[\rho] = \mathrm{Tr}_p\, G(\rho \otimes \xi)G^\dagger.$$

The model of programmable quantum processors is based on the formalism of a general quantum operation [1]. Given a pure program state $|\Xi\rangle_p$, a single data transformation can be described in operator-sum representation as

$$\mathcal{E}(\rho)_{|\Xi\rangle_p} = \sum_{i=1}^{N} A_i \rho A_i^\dagger,$$

where $\sum_{i=1}^{N} A_i^\dagger A_i = I_d$, N is the dimension of the program register, $A_i = {}_p\langle i|G|\Xi\rangle_p$ are Kraus operators and $\{|i\rangle_p\}$ forms an orthonormal basis of $\mathcal{H}_p$.

A processor G implements the following set of quantum operations:

$$\mathcal{C}_G = \{\mathcal{E}_\xi \mid \xi \in S(\mathcal{H}_p)\}.$$

Here follows the basic definition of equivalence of processors that relates processors that are able to implement the same set of operations.

Definition 2. *Two processors G_1, G_2 are* equivalent *if $\mathcal{C}_{G_1} = \mathcal{C}_{G_2}$. We denote two equivalent processors with $G_1 \equiv G_2$.*

We can define a weaker version – equivalence with respect to a given set of quantum operations. That is, such equivalent processors are able to implement each operation from this set but can differ regarding the realization of operations which do not belong to this set.

Definition 3. *Let S be a set of quantum operations acting on $\mathcal{H}_d$.*

Two processors G_1, G_2 are equivalent with respect to the set S *if*

$$S \subseteq \mathcal{C}_{G_1} \cap \mathcal{C}_{G_2} \qquad (\textit{denoted with } G_1 \equiv^S G_2).$$

It is clear that two processors that are equivalent in the strict sense are also equivalent with respect to the set S (this holds also for other types of processors.)

4.2 Probabilistic Programmable Quantum Processors

In this section, we formally define equivalence of probabilistic programmable quantum processors. First, we define the set of implementable operations with respect to a given probability.

Definition 4. *Let G be a probabilistic programmable quantum processor. The set $\mathcal{C}_G^{\mathsf{prob}\geq p}$ denotes the set of operations implementable on G with probability at least p, $0 \leq p \leq 1$:*

$$\mathcal{C}_G^{\mathsf{prob}\geq p} = \{\mathcal{T} \mid P_{success}(\mathcal{T}) \geq p\},$$

where $P_{success}(\mathcal{T}) = \max_\xi\{q \mid \mathcal{E}_\xi = q\mathcal{T} + (1-q)\mathcal{N}$ is a realizable decomposition$\}$, $\mathcal{E}_\xi[\rho] = \mathrm{Tr}_p\, G(\rho \otimes \xi)G^\dagger$, $\xi \in S(\mathcal{H}_p)$; ie. there exists a measurement such that $\mathcal{T}$ is realized with the probability q.

Now, similarly to the deterministic case, we define two versions of equivalence. The first one is the strict equivalence, the second one is a weaker version that considers equivalence only for the specified set of quantum operations.

Definition 5. *Let G_1, G_2 be probabilistic programmable quantum processors and S be a set of quantum operations acting on $\mathcal{H}_d$.*

G_1 and G_2 are probabilistically equivalent with respect to a probability p *if*

$$\mathcal{C}_{G_1}^{\mathsf{prob}\geq p} = \mathcal{C}_{G_2}^{\mathsf{prob}\geq p} \quad (G_1 \equiv_{\mathsf{prob}\geq p} G_2).$$

G_1 and G_2 are probabilistically equivalent with respect to a probability p and the set S *if*

$$S \subseteq \mathcal{C}_{G_1}^{\mathsf{prob}\geq p} \cap \mathcal{C}_{G_2}^{\mathsf{prob}\geq p} \quad (G_1 \equiv^{S}_{\mathsf{prob}\geq p} G_2).$$

4.3 Approximative Programmable Quantum Processors

We formally define equivalence of approximative programmable quantum processors. First, we define the set of implementable operations with respect to a given precision.

Definition 6. *Let G be an approximative programmable quantum processor. The set $\mathcal{C}_G^{\mathsf{appr}\leq\epsilon}$ denotes the set of operations implementable on G with error at most ϵ, $\epsilon \geq 0$:*

$$\mathcal{C}_G^{\mathsf{appr}\leq\epsilon} = \{\mathcal{T} : \epsilon(\mathcal{T}) \leq \epsilon\},$$

where $\epsilon(\mathcal{T}) = \min_{\xi}\{D(\mathcal{T}, \mathcal{E}_\xi)\}$, $\mathcal{E}_\xi[\rho] = \mathrm{Tr}_p\, G(\rho \otimes \xi)G^\dagger$, $\xi \in S(\mathcal{H}_p)$, $D = 1 - F(\mathcal{T}, \mathcal{E}_\xi)$, F is the process fidelity.

The strict and weak equivalence of approximative programmable quantum processors is defined as follows:

Definition 7. *Let G_1, G_2 be programmable quantum processors and S be a set of quantum operations acting on $\mathcal{H}_d$.*

G_1 and G_2 are approximatively equivalent with respect to an error ϵ *if*

$$\mathcal{C}_{G_1}^{\mathsf{appr}\leq\epsilon} = \mathcal{C}_{G_2}^{\mathsf{appr}\leq\epsilon} \quad (G_1 \equiv_{\mathsf{appr}\leq\epsilon} G_2).$$

G_1 and G_2 are approximatively equivalent with respect to an error ϵ and the set S *if*

$$S \subseteq \mathcal{C}_{G_1}^{\mathsf{appr}\leq\epsilon} \cap \mathcal{C}_{G_2}^{\mathsf{appr}\leq\epsilon} \quad (G_1 \equiv^{S}_{\mathsf{appr}\leq\epsilon} G_2).$$

Example: Using results from [13], we obtain the following relation between d-dimensional versions of QID (quantum information distributor) and SWAP gate:

$$\mathsf{QID}_d \equiv^{\mathcal{U}_d}_{\mathsf{appr}\leq 1-\frac{1}{d^2}} \mathsf{SWAP}_d,$$

where $\mathcal{U}_d$ denotes the set of all unitary transformations on $\mathcal{H}_d$.

4.4 Probabilistic Processors Used as Approximative Ones

A probabilistic processor without measurement can be used as an approximative processor [8]. Instead of measuring the output of the program register, we discard it, *ie.* we trace over it. The transformation can be expressed as $\mathcal{E}_\xi[\rho] = p_{success}\mathcal{T}[\rho] + p_{error}\mathcal{O}[\rho]$, where $\mathcal{T}$ is the map we want to perform and both p_{error} and $p_{success}$ are independent of the data state ρ. Due to properties of the process fidelity F one finds that $p_{success} \leq 1 - D(\mathcal{E}_\xi, \mathcal{T}) \leq F(\mathcal{E}_\xi, \mathcal{T})$. The accuracy of the approximation is therefore bounded from below by the success probability. This inequality implies the following relation between sets of operations implementable on probabilistic and approximative processors and the corresponding equivalence:

$$\mathcal{C}_G^{\mathsf{prob}\geq p} \subseteq C_G^{\mathsf{appr}\leq 1-p},$$

$$G_1 \equiv_{\mathsf{prob}\geq p} G_2 \Rightarrow G_1 \equiv_{\mathsf{appr}\leq 1-p} G_2.$$

4.5 Equivalence Condition for Deterministic Processors

The sufficient condition for the equivalence of basic deterministic type of processors was given in [14].

Theorem 1. *Two (deterministic) processors G_1, G_2 are equivalent if they are related by the following equation:*

$$G_1 = (I \otimes U_2)\, G_2\, (I \otimes U_1).$$

Two processors are equivalent if one processor can be converted to another by inserting two fixed unitary gates, one at the input and one at the output of the program register. This condition is sufficient for the equivalence of two processors but it is not proven that it is necessary as well: $G_1 = (I \otimes U_2)\ G_2\ (I \otimes U_1) \stackrel{?}{\Leftrightarrow} G_1 \equiv G_2$.

4.6 Equivalence Condition Inspected for Other Types

The condition for equivalence given in the previous section is sufficient for each type of programmable processors.

$$\begin{aligned} G_1 = (I \otimes U_2)\ G_2\ (I \otimes U_1) &\Rightarrow G_1 \equiv G_2 \\ &\Rightarrow G_1 \equiv_{\mathsf{prob}\geq p} G_2 \\ &\Rightarrow G_1 \equiv_{\mathsf{appr}\leq \epsilon} G_2 \end{aligned}$$

These relations hold for any $0 \leq p \leq 1$ and $\epsilon \geq 0$. If processors implement deterministically the equal set of quantum operations then a measurement at the output of the program register does not change the set of implementable operations with respect to the probability p on these processors. Similarly, for approximative equivalence, both processors implement perfectly the equal set of operations and therefore results concerning approximations of operations have to be equal as well.

Example: Let us consider the following example of equivalent probabilistic processors:

$$G_1 = \sum_{i=1}^{d^2} \sigma_i \otimes |i\rangle\langle i| \qquad G_2 = \sum_{j=1}^{d^2} U_j \otimes |j\rangle\langle j|,$$

where $\{\sigma_i\}$ is the set of generalized Pauli matrices, U_j forms a different basis of unitary operations, *ie.* U_j are unitary operators satisfying the orthogonality relation $\mathrm{Tr}\, U_k^\dagger U_l = d\delta_{kl}$, d is the dimension of the data register.

With both processors G_1 and G_2, we can implement an arbitrary unitary transformation with probability $1/d^2$. In this sense, both processors are universal: processors are equivalent with respect to probability $1/d^2$ and the set $\mathcal{U}_d$ of all unitary transformations on $\mathcal{H}_d$:

$$\mathcal{U}_d \subset \mathcal{C}_{G_1}^{\mathsf{prob}\geq 1/d^2} \cap \mathcal{C}_{G_2}^{\mathsf{prob}\geq 1/d^2}.$$

However, the equivalence condition for basic deterministic processors is not necessary in this case because processors G_1 and G_2 are not related by the unitary maps applied only to the program register, $G_1 \neq (I \otimes U_2)\ G_2\ (I \otimes U_1)$.

4.7 Equivalence Condition for U processors

Processors from the class of U processors implement controlled unitary operations and have the following form:

$$G = \sum_{i=1}^{N} U_i \otimes |i\rangle\langle i|,$$

where U_i are unitary operators and $\{|i\rangle\}$ is a basis of the program space. It is the type of processors studied in [7].

If we restrict ourselves to the class of deterministic U processors, we have the following necessary and sufficient equivalence condition:

Theorem 2. *Let G_{U_1}, G_{U_2} be two U processors.*

$$G_{U_1} = (I \otimes U^\dagger)\ G_{U_2}\ (I \otimes U) \Leftrightarrow G_{U_1} \equiv G_{U_2}$$

Proof. Sufficiency of this condition is evident from Theorem 1.

To see the necessity of the equivalence condition, recall that a U processor has the specific form $\sum_{i=1}^{N} U_i \otimes |i\rangle\langle i|$.

The data part has to remain the same – $\sum_{i=1}^{N} U_i$. The program part could be modified. However, we still have to guarantee that it is of the form $|j\rangle\langle j|$ for some basis of the program register $\{|j\rangle\}$. Only this basis can be changed – by applying unitary transformation. Such operation cannot affect the set of implementable quantum operations applied to data.

Therefore, two equivalent U processors are related by a change of the basis of the program register.

5 Conclusion and Future Work

In this paper, we have formally defined the notion of equivalence for deterministic, probabilistic and approximative types of programmable quantum processors. Strict and weak versions of equivalence were given. We have inspected the equivalence condition given for deterministic processors and its applicability to other types of processors. There are still open questions regarding the equivalence conditions for each type. We can also consider the possibility of changing the notion of equivalence in the following broader sense. We can define two processors as being equivalent if the sets of operations they implement can be transformed into each other by fixed unitary operators.

Acknowledgements I would like to thank Mário Ziman and Mark Hillery for an invaluable discussion. This work is supported by MSM0021622419 and GAČR 201/07/0603.

References

1. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press (2000)
2. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings of the 35th Annual Symposium on Foundations of Computer Science. (1994)
3. Bennett, C.H., Brassard, G.: Quantum cryptography: public key distribution and coin tossing. In: Proceedings of IEEE International Conference on Computers Systems and Signal Processing, Bangalore, India (1984) 175–179
4. Ziman, M., Bužek, V.: Realization of POVMs using measurement-assisted programmable quantum processors. Physical Review A **72** (No. 2) (2005) 022343
5. Dušek, M., Bužek, V.: Quantum controlled measurement device for quantum-state discrimination. Physical Review A **66** (No. 2) (2002) 022121
6. Bužek, V., Hillery, M., Ziman, M., Roško, M.: Programmable quantum processors. Quantum Information Processing **5** (No. 5) (2006) 313–420
7. Nielsen, M.A., Chuang, I.L.: Programmable quantum gate arrays. Physical Review Letters **79** (No. 2) (1997) 321–324
8. Vidal, G., Masanes, L., Cirac, J.I.: Storing quantum dynamics in quantum states: a stochastic programmable gate. Physical Review Letters **88** (No. 4) (2002) 047905
9. Hillery, M., Bužek, V., Ziman, M.: Probabilistic implementation of universal quantum processors. Physical Review A **65** (No. 2) (2002) 022301
10. Vidal, G., Cirac, J.I.: Storage of quantum dynamics on quantum states: a quasi-perfect programmable quantum gate. quant-ph/0012067 (2000)
11. Hillery, M., Ziman, M., Bužek, V.: Approximate programmable quantum processors. Physical Review A **73** (No. 2) (2006) 022345
12. Gilchrist, A., Langford, N.K., Nielsen, M.A.: Distance measures to compare real and ideal quantum processes. Physical Review A **71** (No. 6) (2005) 062310
13. Ziman, M., Bužek, V.: Universality and optimality of programmable quantum processors. Acta Physica Hungarica B (No. 26) (2006) 277–291
14. Hillery, M., Bužek, V., Ziman, M.: Programmable quantum gate arrays. Fortschritte der Physik **49** (2001) 987