Quantum Circuits: From a Network to a One-Way Model

Larisse D. Voufo * †
Institute for Scientific Interchange Foundation, 10133 Torino, Italy

Gerardo Ortiz ‡
Department of Physics, Indiana University, Bloomington, IN 47405

Amr Sabry §
Department of Computer Science, Indiana University, Bloomington, IN 47405

ABSTRACT

We present elements of quantum circuits translations from the standard network (or circuit) model to the one-way one. We present a general translation scheme, give an account of currently existing tools to apply the scheme, and propose an extension of those tools into a complete translation calculus. We analyze the set of difficulties incurred from such work, and show an engendered opening to new sets of discussions and ideas. Among others, this paper extends the findings to the notions of graphical concatenation, graph state reduction (GSR) and graph state extension (GSE) passes. Further, it proposes an algorithm for running the (extended) measurement calculus with acceptable efficiency.

1. INTRODUCTION

As interests and need for experiments in one-way quantum computation rise, the need for a systematic way to translate already existing circuits, especially those that derive from the standard model, into circuits obeying the one-way model, rises equivalently. Some work has already been done by Schlingemann [10] that systematically translates circuits from a one-way to a circuit model, but that does not address the question of optimization or efficiency of the translation. Other works implicitly provide methods of circuit optimizations in a particular model through an account of possible circuit transformations and equivalence classes [9, 16]. However, to the best of our knowledge, there does not exist a straightforward and systematic way to transform circuits from one particular model to another, and surely not the most efficiently or in a way that would produce the best translated circuit possible. This poses some cumbersome limits on the possibilities of implementing and testing simulations in and across given models.

*Also affiliated with the Department of Computer Science at Indiana University, Bloomington, IN 47405

†email: voufo@isi.it

‡email: ortizg@indiana.edu

§email: sabry@cs.indiana.edu

DCM-QPL'08, July 12-13, 2008, Reykjavik, Iceland

In this paper, we address the problem of translating a circuit from a standard network to a one-way model. We give an account of elements that such a task would normally involve, reviewing current works and extending them to more effective components such as a graphical concatenation for circuits in the one-way model. Further, we approach the question of translations' efficiency and propose an extension of the measurement calculus [8] to include optimization passes.

In the following, we assume a basic familiarity with notions of quantum mechanics and computation, fundamental differences between the circuit and the one-way models, as well as with the relation of one-way realizations to graphs. In addition, for simplicity, we will refer to the Pauli matrices σ_x, σ_y and σ_z as X, Y and Z, respectively. We will also precede any controlled operation with the letter "C". The longer version of this paper, will elaborate more on concepts introduced or used here.

A general translation scheme

In a first time, the translation can follow the structure of the circuit, decomposing it into several levels of subdivisions (in a tree-like recursive fashion) until we reach the smallest ones. The chosen one-way realizations for those smallest subdivisions will form a set that we refer to as *universal sets*. Then, at each level, associated universal sets will need to be combined accordingly based on a choice from various combination methods that we call *concatenations*. This will produce a first realization that will need to be improved based on some particular needs that we refer to as *realization needs*. We call this step *optimization*.

Alternatively, the translation can be done using the *phase map decomposition*, which bypasses completely any reference to the circuit-model representation. Instead, it analyses the possible inputs and outputs of the circuit, and generates alternative realizations for the universal set.

All combinations of these alternatives describe the possible translation paths of the circuit, which can in turn be classified in terms of the realization needs that they would meet best. Indeed, this suggests choosing the right set of unitaries and concatenation methods (plus optimizations) to start with as a detrimental factor of translation. In any events, experiments will depend heavily on the efficiency and flexibility of the associated concatenation method.

Related work

Raussendorf *et al.* have given a detailed account of one-way (or measurement-based) quantum computation on cluster states that provides a first universal set for clusters as well as a concatenation method through by-product operators [5]. Alternatively, Danos *et al.* have defined a more robust and parsimonious universal set [7], as well as a more standardized method of concatenation called the measurement calculus [8]. Additionally, they have also defined the phase-map decomposition [6]. Several others, such as Hein *et al.* [16] and Schlingemann [10], have studied graph state properties and their classifications into some equivalence classes that could be helpful in the process of graph state optimization.

We review those concatenation methods, analysing their respective limitations, and derive an alternative scheme based on observing relationships between the graphical representations of to-be-composed one-way realizations: a *graphical concatenation*. In the process, we extend the measurement calculus to include more standardization passes as well as optimization passes derived from recent studies. We then present an example of practical interest with the translation of Cuccaro *et al.*'s quantum ripple carry addition circuit [1] and analyze the effects of the translation on the optimizations performed in the circuit model. As a result, it will be possible to implement a complete translation calculus and the process will give rise to a new thinking process and range of interesting discussions.

2. BUILDING UP A TRANSLATION PATH

2.1 Example: Building the Identity operation

Let's start with the composition of two Hadamard (H) operations into the Identity (I). A single-wire one-way realization of H entangles a logical input qubit with a logical output one, and then measures the input in X. With respect to the randomness of measurement results, this realizes the unitary $X_2^{s_1} H_{[1]}$ and the associated command sequence (from its pattern definition [1]) is $X_2^{s_1} M_1^x E_{12}$. The composition will have the output of the first application become the input of the second. Thus, the final realization of I will constitute of three qubits 1, 2, and 3, where qubit 2 is entangled with the remaining two and qubits 1 and 2 are measured in X.

Under the by-product approach [5]. The composition so described is justified by the fact that realizations meet the requirement of having their logical inputs measured in X. It can then be verified by composing the realized unitaries or by analysing the resulting state's characteristic eigenvalue equations; either way generating the associated by-product operators. So, the final unitary it realizes will be $X_3^{s_2} H_{[2]} \cdot X_2^{s_1} H_{[1]}$, and then $X_3^{s_2} Z_2^{s_1} \cdot H_{[2]} \cdot H_{[1]} = X_3^{s_2} Z_2^{s_1} \cdot I_{[1]}$ after propagation of the by-product operators accross the composing unitaries.

The propagation is rather abstract, as it deals with unitaries rather than automated entities, and does not guarantee the conservation of the parameters associated with unitaries that are not in the Clifford group [2] (although it does conserve the individual sets of measurements from each sub-circuit) — so the parameters will have to be redefined accordingly. Further, the task of modifying a measurement pattern for improvement, while maintaining the meaning of the realization and updating the associated by-product operator accordingly, is not trivial. Among others, one will have to reconsider the new input state's correlation operators, regenerating and re-evaluating its characteristic eigenvalue equations. This is without mentioning the fact that it is practically impossible to apply this scheme to arbitrary realizations for which the realized unitary and by-product operators have yet to be determined.

The phase-map decomposition handles the latest limitation and allows for trial-and-error experimentations that will uncover the realized unitary and associated by-product operator. All other limitations are addressed by the measurement calculus approach.

1. $E_{ij} X_i^s$	$\longrightarrow$	$X_i^s Z_j^s E_{ij}$
2. $E_{ij} Z_i^s$	$\longrightarrow$	$Z_i^s E_{ij}$
3. $E_{ij} A_{\vec{k}}$	$\longrightarrow$	$A_{\vec{k}} E_{ij}$, with $A \neq E$
4. ${}^t[M_i^\alpha]^s X_i^r$	$\longrightarrow$	${}^t[M_i^\alpha]^{s+r}$
5. ${}^t[M_i^\alpha]^s Z_i^r$	$\longrightarrow$	${}^{t+r}[M_i^\alpha]^s$
6. $A_{\vec{k}} X_i^s$	$\longrightarrow$	$X_i^s A_{\vec{k}}$, with $A \neq X$ and $A \neq Z$
7. $A_{\vec{k}} Z_i^s$	$\longrightarrow$	$Z_i^s A_{\vec{k}}$, with $A \neq X$ and $A \neq Z$
8. ${}^t[M_i^\alpha]^s$	$\longrightarrow$	$S_i^t\,[M_i^\alpha]^s$
9. $X_j^s S_i^t$	$\longrightarrow$	$S_i^t X_j^{s[t+s_i/s_i]}$
10. $Z_j^s S_i^t$	$\longrightarrow$	$S_i^t Z_j^{s[t+s_i/s_i]}$
11. ${}^t[M_j^\alpha]^s S_i^r$	$\longrightarrow$	$S_i^r\, {}^{t[r+s_i/s_i]}[M_j^\alpha]^{s[r+s_i/s_i]}$
12. $\perp S$	$\longrightarrow$	$\perp$

Figure 1: Standardization passes.

Under the measurement calculus approach [8]. Logical inputs can be measured in the X-Y plane of the Bloch sphere. Further, the approach is more generalized and systematic, propagating by-product operators — now Pauli correction commands — over measurement commands, rather than unitaries. Thus, the earlier composition is now performed through the more automated concatenation and standardization (C&S) of patterns.

Using the symmetric transformations provided by the work of Danos *et al.* [8] and listed in Figure 1, we can standardize the composition $X_3^{s_1} M_2^x E_{23} \cdot X_2^{s_1} M_1^x E_{12}$ by applying the sequence of passes 1, 3, 4, and 7 onto it. This will result in the following pattern for I:

$$\mathscr{I}(1) = (\{1,2,3\}, \{1\}, \{3\}, X_3^{s_2} Z_3^{s_1} M_2^x M_1^x E_{23} E_{12}) .$$

In addition, we can even derive a more general realization of I by composing arbitrary J unitaries[3], rather than H ones. Thus, the final pattern will be $\mathscr{I}(1) = \mathscr{J}_\beta(2) \cdot \mathscr{J}_\alpha(1) = (\{1,2,3\}, \{1\}, \{3\}, X_3^{s_2} M_2^{-\beta} E_{23} \cdot X_2^{s_1} M_1^{-\alpha} E_{12})$, and then

$$\mathscr{I}(1) = (\{1,2,3\}, \{1\}, \{3\}, X_3^{s_2} Z_3^{s_1} \left[M_2^{-\beta}\right]^{s_1} M_1^{-\alpha} E_{23} E_{12})$$

after running the same sequence of passes as before.

The transformations used here ensure the conservation of the meaning of realizations, while automatically updating the resulting by-product operators accordingly. Hence, there no longer is a

[1]The pattern is defined according to this tuple: (*Vertices*, *Logical Inputs*, *Logical Outputs*, *Command Sequence*).

[2]For example, a general rotation U_{rot} will undergo modifications according to the following relations:

$$\begin{cases} U_{rot}(\alpha,\beta,\zeta) X & = X U_{rot}(\alpha,-\beta,\zeta) \\ U_{rot}(\alpha,\beta,\zeta) Z & = Z U_{rot}(-\alpha,\beta,-\zeta) \end{cases} .$$

[3]

$$\mathscr{J}_\alpha(1) = (\{1,2\}, \{1\}, \{2\}, X_2^{s_1} M_1^{-\alpha} E_{12}) .$$

Algorithm 1 A simple C&S Algorithm.

1. *Propagate E_2 backwards, across C_1* $\longrightarrow E = E_2.E_1$
2. *Propagate M_2 backwards, across C_1* $\longrightarrow M = M_2.M_1$ *and* $C = C_2.C_1$
3. *For each M*
   (a) *Introduce Shift*
   (b) *Propagate the Shift forward, across M and C, dropping it at the end of the command sequence.*

E_x, M_x, and C_x respectively represent the sequence of entanglement, measurement and Pauli correction operations of two standard sub-patterns 1 and 2, with x indicating the sub-pattern acted upon.

1. ${}^{t}\left[M_i^{-\frac{\pi}{2}}\right]^{s} = {}^{t}\left[M_i^{\frac{\pi}{2}}\right]^{s+1} = {}^{t+s+1}\left[M_i^{y}\right] = S_i^{t+s+1} M_i^{y}$
2. ${}^{t}\left[M_i^{\frac{\pi}{2}}\right]^{s} = {}^{t+s}\left[M_i^{y}\right] = S_i^{t+s} M_i^{y}$
3. ${}^{t}\left[M_i^{0}\right]^{s} = {}^{t}\left[M_i^{0}\right] = S_i^{t} M_i^{x}$
4. ${}^{t}\left[M_i^{3\frac{\pi}{4}}\right]^{s} = {}^{t+1}\left[M_i^{\frac{\pi}{4}}\right]^{s+1} = S_i^{t+1}\left[M_i^{\frac{\pi}{4}}\right]^{s+1} = S_i^{t+1}\left[M_i^{-\frac{\pi}{4}}\right]^{s}$

Figure 2: Extending the standardization passes.

striking need for verification and non-Clifford unitaries are handled inherently. This considerably improves the computation time, and an efficient algorithm for their execution is provided in Algorithm 1. The algorithm is defined for the composition of two circuits and can be applied several times for more composite circuits. It succeeds at skipping the unnecessary passes 3 and 12 and implicitly transfers dependencies induced by Z-actions into introduced and propagated *Shifts*.

So, where is the problem?. The situation gets tedious as the size of circuits increases. Among others, patterns become more complex and less human-readable, while the C&S algorithm becomes more time-consuming. We address these problems in the following section.

2.2 Extending The Measurement Calculus

We extend the previous list of standardization passes by noticing the equalities in Figure 2. Because of their simplicity, they can be incorporated into the C&S algorithm implicitly and hence improve its efficiency further. In addition, we simplify our pattern definition so that it only pays attention to ancilla qubits as necessary, while showing a direct relationship with the corresponding network model representation.

A monadic pattern representation. Vizzotto et al. [11] earlier related "unusual" features of quantum computing (quantum parallelism and measurement) to the semantic constructions of *monads* and *arrows* from the theory of programming languages. Using their abstractions, a quantum computation expressed in the circuit model could be elegantly expressed as a computation in the monad of quantum computation.

Take, for example, the decomposition of a CCZ operation in terms of controlled rotations. Based on its circuit-model representation, we can express it as in Algorithm 2. Here, an expression $F\,(i1, i2, ...)$ $>>=$ $\backslash(o1, o2, ...)$ represents the application of some function F on some input qubits $(i1, i2, ...)$, producing the outputs $(o1, o2, ...)$. In addition, the arrow $\rightarrow$ specifies the sequence of operations, while *return* specifies the final logical output qubits of the pattern. $>>=$ and *return* are monadic operations that hide all the quantum magic, allowing for easy translations into the previous patterns structure or a valid graphical representation. Remarkably, the one-way realization of the same circuit can be expressed in the exact same notation by varying the underlying monad. Figure 7 in the Appendix shows an example of the hidden computation.

Algorithm 2 Monadic pattern for CCZ.

$$\begin{aligned}
&CCZ\,(A0,B0,C0) \;=\\
&C\frac{\pi}{2}Phase\,(B0,C0) \;>>=\; \backslash(B1,C1) \;\rightarrow\\
&CNot\,(A0,B1) \;>>=\; \backslash(A0,B2) \;\rightarrow\\
&C(-\frac{\pi}{2})Phase\,(B2,C1) \;>>=\; \backslash(B3,C2) \;\rightarrow\\
&CNot\,(A0,B3) \;>>=\; \backslash(A0,B4) \;\rightarrow\\
&C\frac{\pi}{2}Phase\,(B4,C2) \;>>=\; \backslash(B5,C3) \;\rightarrow\\
&return\,(A0,B5,C3)
\end{aligned} \tag{1}$$

2.2.1 The Graphical Concatenation

At times, a graphical structure, for a one-way realization, representing only the entanglement relationships between the qubits as well as each one-qubit measurement can be enough for a user. Thus, we derive a concatenation scheme based on graphical representations of one-way realizations of sub-circuits.

In the course of concatenating two circuits 1 and 2, with some outputs of circuit 1 (O_1) becoming inputs of circuit 2 (I_2) — we call them *intersecting qubits*, let's explore the standardization passes and the C&S algorithm further. For each interleaving qubit, let's examine the relationships between the measurement in I_2 and the associated Pauli correction in O_1, with respect to the corresponding measurement in the resulting pattern. This yields the following conclusion, keeping in mind that each introduced *Shift* is dependant on the corrections signals of its associated qubit in O_1:

1. All X-correction will always introduce *Shifts* on the neighbors (within circuit 2) of the intersecting qubit. Meanwhile, the measurement
   (a) will either simply remain unchanged if it is in the X-observable,
   (b) or remain unchanged and accompanied with a *Shift*, if it is in the Y-observable,
   (c) or become dependant on the correction's signals otherwise.
2. Meanwhile, All Z-correction on the intersecting qubit will always leave its measurement unchanged and be accompanied with a *Shift*.

Therefore, the concatenation procedure will consist of, first, determining whether the condition *1c* above is satisfied and, second, blindly converting the representation of each intersecting qubit from input- or output- qubit representations to regular-qubit (ancilla) ones, while conserving their respective derived measurement angle.

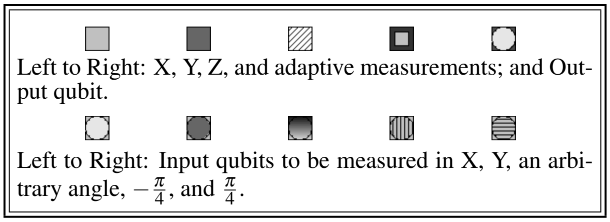

Figure 3: Graph state representation legend.

A reduced graphical concatenation scheme. In this paper's graphical representation, we are only concerned with whether measurements are adaptive (i.e. dependant on previous measurements) and with the particular measurements in X, Y, and $\pm\frac{\pi}{4}$ when they are not. Thus, intersecting qubits with adaptive measurements need no further analysis. This is particularly helpful in situations where computing the exact Pauli corrections is not as much a priority as determining the entanglement and measurement patterns of the final circuit.

Further, this scheme can be used to approximate concatenations of circuits for which information about signals and Pauli corrections are missing. One simply needs to assume that intersecting qubits are X-corrected and that the *Shifts*' propagation will never cancel signals — and hence neither measurements, nor corrections.

As an illustration of the simplicity and convenience of this scheme, we derive a one-way realization of the CCZ's pattern from Equation 1 in Figure 4. The graph is defined according to the following legend.

Graph legend. To easily recognize how close a graph state is to a cluster one, we borrow Raussendorf *et al.*'s 2-dimensional lattice grid design [5], and extend the design to include representations of logical input states that will be measured in observables different from X and Y. In relation to general graph states, we add entanglement "wires" (simple lines) for cases where maintaining the rectangular grid representation is not possible. These "wires" will hence constitute the characteristic indication that the state represented is a general graph state and not a cluster one. Finally, for step-by-step illustrations of concatenations, we also use arrows to represent which logical output qubit becomes which logical input one. Figure 3 presents the complete legend for qubits in use.

A general graphical concatenation scheme. Further observations yield to the realization that if we extend the graphical representation to include the specification of more particular measurement angles, the type of corrections (X or Z) that are performed on each output qubit, as well as the existing signals on each measurement and correction, then we will be able to easily introduce and propagate the *Shifts* accordingly. Further, we will know the exact final pattern.

Furthermore, by noticing that the *Shifts*' propagations affect only those operations (measurements and corrections) that have explicit signals (different from the number 1), we then improve the C&S algorithm even further by skipping non-adaptive measurements during the *Shifts*' propagation step.

2.2.2 Optimizing the results

We classify optimizations defined in recent works into GSR and GSE transformations. We also re-express them as symmetric transformation passes to be added to previous list. Indeed, in order to maximize the transformations' effectiveness, by finding the smallest graph state that satisfies one's realization needs, it is preferable to run the GSR transformations prior to the GSE ones. In addition, the execution of this step will require that measurements be allowed to be in Z, in addition to the previous limitation in the X-Y plane.

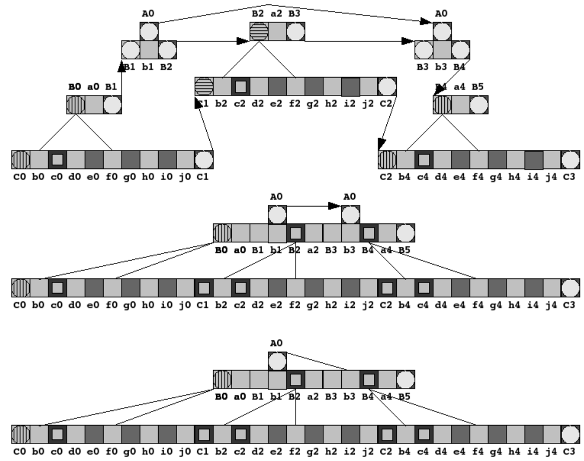

Figure 4: Building a CCZ gate from controlled rotations (graph) .

Graph State Reduction (GSR) transformations. These concern realizations that need to use as few qubits as possible, due to either restrictions on physical space, or needs to maintain coherence.

- The removal of redundant qubits [5]

  Type-lifting the notations from the measurement calculus, M^z_{Q/Q_N} and s_{Q/Q_N} respectively represent the sets of all measurements in Z and of all resulting signals. The transformation pass is then

  $$C_{Q_N} M_{Q_N} E_Q \longrightarrow C_{Q_N} M_{Q_N} Z_{Q_N}^{s_{Q/Q_N}} M^z_{Q/Q_N} E_Q \ ,$$

  which will result in the pattern $C_{Q_N} M_{Q_N} E_{Q_N}$. Thus $E_{Q_N} = Z_{Q_N}^{s_{Q/Q_N}} M^z_{Q/Q_N} E_Q$.

- The removal of unnecessary measurements [5]

  The transformation pass for this is:

  $$C_Q M_{Q_N} M^x_{Q/Q_N} E_Q \longrightarrow C_{Q_N} M_{Q_N} E_{Q_N} \ ,$$

  where M^x_{Q/Q_N} represents the set of measurements in X that can be removed as specified by Raussendorf *et al.* [5], and thus so that the size of Q/Q_N is even. Also, the resulting pattern is modified by removing all reference (e.g. entanglement edges and measurement outcomes) to the qubits that the removed measurements were acting upon.

Graph State Extension (GSR) transformations. Among realization needs satisfied by these transformations, are the preferences for either cluster or two-colorable states. In any event, the particular passes in the GSE case can be generalized as converses of GSR transformations.

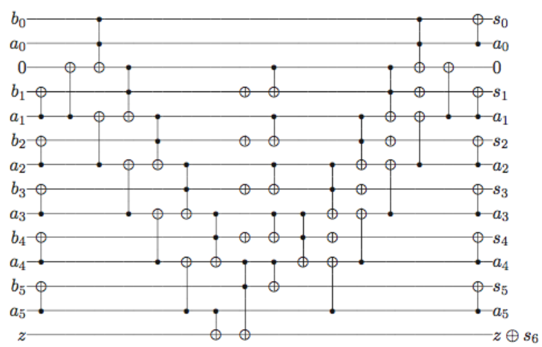

Figure 5: 6-qubit addition circuit: high parallelism version.
[Picture from [1]]

- The introduction of Z measurements

  We derive the following transformation pass:

$$C_{Q_N} M_{Q_N} E_{Q_N} \longrightarrow C_{Q_N} M_{Q_N} Z_{Q_N}^{s_{Q/Q_N}} M^z_{Q/Q_N} E_Q \ ,$$

  which will eventually produce a $C_Q M_Q E_Q$ pattern after propagation of the $Z_{Q_N}^{s_{Q/Q_N}}$ corrections, and introduced *Shift* commands, across the M_{Q_N} measurements, and the C_{Q_N} corrections.

- The identity propagation

  This is simply a double-Hadamard transformation and corresponds to this pass:

$$C_{Q_N} M_{Q_N} E_{Q_N} \longrightarrow C_{Q_N} M_{Q_N} C_{Q_N}^{s_{Q/Q_N}} M^x_{Q/Q_N} E_Q \ ,$$

  where, again, the size of Q/Q_n is even.

It is important to clarify that these are not the only existing transformations. In addition to the Hadamard transformation itself, Schlingemann studies the removal of the Clifford parts of a given graph state in depth, and presents additional transformations [9].

3. PRACTICAL EXAMPLE: A (NEW) ONE-WAY QUANTUM RIPPLE CARRY ADDITION CIRCUIT

We take a special look at Cuccaro *et al.*'s "new quantum ripple-carry addition circuit" [1] and generate clear translations in both graph and cluster states; illustrating the applicability of the optimization passes defined earlier (especially the introduction of Z measurements). Among others, the circuit from Figure 5 results in the realization in Figure 6, of the high parallelism version of their addition.

4. RESULTS ANALYSES: NO OPTIMIZATION TRANSLATION

We review the needs and quantifiable parameters for algorithm analysis across models of computations, analyse the efficiency of our translations, and notice that optimizations are not always conserved across different models of computations. In fact, whether we translate the original addition circuit, or its high parallelism version, the resulting one-way realizations have a low constant bounded depth and the difference in extra ancillae is negligible. This suggests that circuit optimizations performed under the network model cannot be expected to make much of a difference under the one-way one. Indeed, optimizations that manipulate gates in the Clifford group can only affect the spacial and operational resources, while those involving appropriate changes in non-Clifford operations affect the final depth.

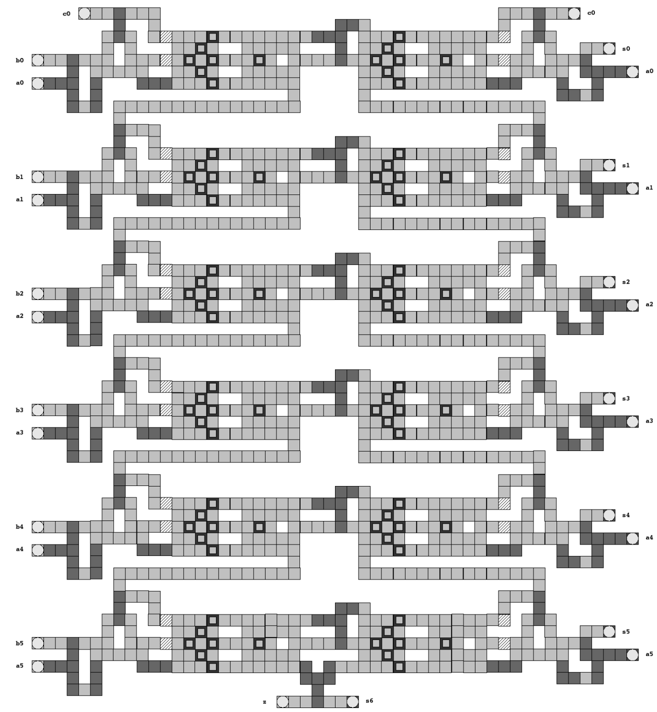

Figure 6: One-way translation of Fig. 5 (cluster).

5. FUTURE: A COMPLETE TRANSLATION CALCULUS

Based on the information gathered from our translations, and the analysis of their results, we notice a number of limitations on the currently available translation tools: (1) measurement angles, in the measurement calculus, are limited to the X-Y plane of the Bloch sphere, (2) algorithms for running the measurement calculus passes have not been designed, and nor has their efficiency been discussed, and (3) we still need to actually define complete sets of optimization passes. We have addressed those limitations and proposed extensions to the measurement calculus.

Attempts at resolving the said limitations call for the definition of a complete translation calculus. Indeed, this is very possible and will broaden our experimental range by allowing for flexible navigations between different sub-categories of graph states (e.g. clusters, 2-colorable, etc...). In fact, it could even be extended to allow navigations and analysis across different models of computation. The idea of implementing such a calculus, then running and analysing simulations raises a range of interesting discussions that would be handled better once the complete translation calculus is defined and analysed:

- Which combination of methods is best?
- Are results from phase map decompositions optimal?

- How well can one efficiently perform a systematic circuit's optimization?

Meanwhile, defining the calculus raises another set of discussions related to possible future experiments:

- When implementing the measurement calculus, could a graph theoretical approach be more effective?
- How effective would conversions of results from a graphical concatenation, to and from their pattern representations, be?
- How open would the design be to trial-and-error approaches?

Acknowledgements

We would like to thank all direct and indirect contributors to this work. We especially appreciated fruitful inputs from a number of people including Dirk Schlingemann and the rest of the Quantum Information Unit group at the Institute for Scientific Interchange.

6. REFERENCES

[1] Cuccaro, S., T. Draper, S. Kutin, D. Moulton. 2006. *A new quantum ripple-carry addition circuit*. arxiv:quant-ph/0410184 v1 22 Oct 2004.

[2] Browne, D., H. Briegel. 2006. *One-way Quantum Computation*. arxiv:quant-ph/0603226 v2 3 Oct 2006.

[3] Raussendorf, R., H. Briegel, 2000. *A One-Way Quantum Computer*. DOI: 10.1103/PhysRevLett.86.5188.

[4] Browne, D., R. Raussendorf, H. Briegel. 2000-01. *The One-Way Quantum Computer: Computation from Correlations*. [www.qunat.org/quoxic/talks/danbrowne.ppt]

[5] Raussendorf, R., D. Browne, H. Briegel. 2006. *Measurement-based quantum computation on cluster states*. arxiv:quant-ph/0301052 v2 14 Jan 2003.

[6] Beaudrap, N., Danos V., E. Kashefi. *Phase-map decomposition for unitaries.* arXiv:quant-ph/0603266 v1 29 Mar 2006.

[7] Danos, V., E. Kashefi, P. Panangaden. 2005. *Parsimonious and robust representation of unitaries in the one-way model.*

[8] Danos, V., E. Kashefi, P. Panangaden.*The measurement Calculus.* arXiv:quant-ph/0412135 v1 17 Dec 2004.

[9] Schlingemann, D. *Cluster states, algorithms and graphs*. arXiv:quant-ph/0305170 v2 4 Aug 2003.

[10] Schlingemann, D. *Logical network implementation for cluster states and graph codes*. arXiv:quant-ph/0202007 v1 1 Feb 2002.

[11] Vizzotto, J. K., T. Altenkirch, A. Sabry. 2007. *Structuring quantum effects: superoperators as arrows*. arXiv:quant-ph/0501151 v1 25 Jan 2005.

[12] Vedral., V., A. Barenco, A. Ekert. *Quantum Networks for Elementary Arithmetic Operations*. arXiv:quant-ph/9511018 v1 16 Nov 1995.

[13] Draper, T. G., S. A. Kutin, E. M. Rains, K. M. Svore. 2006. *A Logarithmic-Depth Quantum Carry-Lookahead Adder*. arXiv:quant-ph/0406142 v1 20 Jun 2004.

[14] Drapper, T.G. 1998. *Addition on a Quantum Computer*. arXiv:quant-ph/0008033 v1 7 Aug 2000.

[15] Bennet, C. H. 1973. *Logical Reversibility of Computation*.

[16] Hein, M., J. Eisert, H. J. Briegel. 2006. *Multi-party entanglement in graph states*. arxiv:quant-ph/0307130 v7 9 Aug 2005

APPENDIX

Defining the pattern for a CCZ operation

Figure 7 illustrates the translation from the pattern definition in Algorithm 2 (monadic) to that of Danos *et al.* [8], making an explicitly use of the trivial pattern

$$\mathscr{T}(1) = (\{1\}, \{1\}, \{1\}, \perp)\,.$$

$$
\begin{aligned}
CCZ_{\{A0,B0,C0\}} &= (\mathscr{T}_{(A0)} \otimes C\frac{\pi}{2}Phase_{\{B4,C2\}})\,.\,(\wedge\mathscr{X}_{\{A0,B3\}} \otimes \mathscr{T}_{(C2)})\,.\,(\mathscr{T}_{(A0)} \otimes C(-\frac{\pi}{2})Phase_{\{B2,C1\}}) \\
&\quad .\,(\wedge\mathscr{X}_{\{A0,B1\}} \otimes \mathscr{T}_{(C1)})\,.\,(\mathscr{T}_{(A0)} \otimes C\frac{\pi}{2}Phase_{\{B0,C0\}}) \\
&= Z_{C3}^{s_{i4}+s_{g4}+s_{e4}+s_{c4}+s_{C2}+s_{h4}+s_{d4}+s_{b4}+1}\, X_{C3}^{s_{j4}+s_{h4}+s_{f4}+s_{d4}+s_{b4}}\, X_{B5}^{s_{a4}}\, Z_{B5}^{s_{B4}+s_{e4}+s_{c4}+s_{d4}+s_{b4}} \\
&\quad M_{a4}^{x} M_{B4}^{-\frac{\pi}{4}} M_{j4}^{x} M_{i4}^{y} M_{h4}^{x} M_{g4}^{y} M_{f4}^{x} M_{e4}^{y} M_{d4}^{x} \left[M_{c4}^{-\frac{\pi}{4}}\right]^{s_{b4}} M_{b4}^{x} M_{C2}^{-\frac{\pi}{4}} \\
&\quad E_{a4B5} E_{B4a4} E_{j4C3} E_{i4j4} E_{h4i4} E_{g4h4} E_{f4g4} E_{B4f4} E_{e4f4} E_{d4e4} E_{c4d4} E_{b4c4} E_{C2b4} E_{b4B4} \\
&\quad .\, X_{B4}^{s_{b3}} Z_{B4}^{s_{B3}} Z_{A0}^{s_{B3}} M_{b3}^{x} M_{B3}^{x} E_{A0b3} E_{B3b3} E_{b3B4}\,. \\
&\quad Z_{C2}^{s_{i2}+s_{g2}+s_{e2}+s_{c2}+s_{C1}+s_{h2}+s_{d2}+s_{b2}}\, X_{C2}^{s_{j2}+s_{h2}+s_{f2}+s_{d2}+s_{b2}}\, X_{B3}^{s_{a2}}\, Z_{B3}^{s_{B2}+s_{e2}+s_{c2}+s_{d2}+s_{b2}+1} \\
&\quad M_{a2}^{x} M_{B2}^{\frac{\pi}{4}} M_{j2}^{x} M_{i2}^{y} M_{h2}^{x} M_{g2}^{y} M_{f2}^{x} M_{e2}^{y} M_{d2}^{x} \left[M_{c2}^{\frac{\pi}{4}}\right]^{s_{b2}} M_{b2}^{x} M_{C1}^{\frac{\pi}{4}} \\
&\quad E_{a2B3} E_{B2a2} E_{j2C2} E_{i2j2} E_{h2i2} E_{g2h2} E_{f2g2} E_{B2f2} E_{e2f2} E_{d2e2} E_{c2d2} E_{b2c2} E_{C1b2} E_{B2b2} \\
&\quad .\, X_{B2}^{s_{b1}} Z_{B2}^{s_{B1}} Z_{A0}^{s_{B1}} M_{b1}^{x} M_{B1}^{x} E_{A0b1} E_{B1b1} E_{b1B2}\,. \\
&\quad Z_{C1}^{s_{i0}+s_{g0}+s_{e0}+s_{c0}+s_{C0}+s_{h0}+s_{d0}+s_{b0}+1}\, X_{C1}^{s_{j0}+s_{h0}+s_{f0}+s_{d0}+s_{b0}}\, X_{B1}^{s_{a0}}\, Z_{B1}^{s_{B0}+s_{e0}+s_{c0}+s_{d0}+s_{b0}} \\
&\quad M_{a0}^{x} M_{B0}^{-\frac{\pi}{4}} M_{j0}^{x} M_{i0}^{y} M_{h0}^{x} M_{g0}^{y} M_{f0}^{x} M_{e0}^{y} M_{d0}^{x} \left[M_{c0}^{-\frac{\pi}{4}}\right]^{s_{b0}} M_{b0}^{x} M_{C0}^{-\frac{\pi}{4}} \\
&\quad E_{a0B1} E_{B0a0} E_{j0C1} E_{i0j0} E_{h0i0} E_{g0h0} E_{f0g0} E_{B0f0} E_{e0f0} E_{d0e0} E_{c0d0} E_{b0c0} E_{C0b0} E_{b0B0} \\
&= Z_{C3}^{\xi_{C3}} X_{C3}^{\chi_{C3}} X_{B5}^{s_{a4}} Z_{B5}^{\xi_{B5}} Z_{A0}^{\xi_{A0}}\,. \\
&\quad M_{a4}^{x} \left[M_{B4}^{-\frac{\pi}{4}}\right]^{s_{b3}+s_{a2}+s_{b1}+s_{a0}} M_{j4}^{x} M_{i4}^{y} M_{h4}^{x} M_{g4}^{y} M_{f4}^{x} M_{e4}^{y} M_{d4}^{x}\,. \\
&\quad \left[M_{c4}^{-\frac{\pi}{4}}\right]^{\mu_{c4}} M_{b4}^{x} \left[M_{C2}^{-\frac{\pi}{4}}\right]^{\mu_{C2}} M_{b3}^{x} M_{B3}^{x}\,. \\
&\quad M_{a2}^{x} \left[M_{B2}^{\frac{\pi}{4}}\right]^{s_{b1}+s_{a0}} M_{j2}^{x} M_{i2}^{y} M_{h2}^{x} M_{g2}^{y} M_{f2}^{x} M_{e2}^{y} M_{d2}^{x}\,. \\
&\quad \left[M_{c2}^{\frac{\pi}{4}}\right]^{\mu_{c2}} M_{b2}^{x} \left[M_{C1}^{\frac{\pi}{4}}\right]^{\mu_{C1}} M_{b1}^{x} M_{B1}^{x}\,. \\
&\quad M_{a0}^{x} M_{B0}^{-\frac{\pi}{4}} M_{j0}^{x} M_{i0}^{y} M_{h0}^{x} M_{g0}^{y} M_{f0}^{x} M_{e0}^{y} M_{d0}^{x} \left[M_{c0}^{-\frac{\pi}{4}}\right]^{s_{b0}} M_{b0}^{x} M_{C0}^{-\frac{\pi}{4}}\,. \\
&\quad E_{a4B5} E_{B4a4} E_{j4C3} E_{i4j4} E_{h4i4} E_{g4h4} E_{f4g4} E_{B4f4} E_{e4f4} E_{d4e4} E_{c4d4} E_{b4c4} E_{C2b4} E_{b4B4}\,. \\
&\quad E_{A0b3} E_{B3b3} E_{b3B4}\,. \\
&\quad E_{a2B3} E_{B2a2} E_{j2C2} E_{i2j2} E_{h2i2} E_{g2h2} E_{f2g2} E_{B2f2} E_{e2f2} E_{d2e2} E_{c2d2} E_{b2c2} E_{C1b2} E_{B2b2}\,. \\
&\quad E_{A0b1} E_{B1b1} E_{b1B2}\,. \\
&\quad E_{a0B1} E_{B0a0} E_{j0C1} E_{i0j0} E_{h0i0} E_{g0h0} E_{f0g0} E_{B0f0} E_{e0f0} E_{d0e0} E_{c0d0} E_{b0c0} E_{C0b0} E_{b0B0}
\end{aligned}
$$

with

$$
\begin{cases}
\mu_{C1} = & s_{j0}+s_{h0}+s_{f0}+s_{d0}+s_{b0} \\
\mu_{c2} = & s_{b2}+s_{b1}+\mu_{C1}+s_{a0} \\
\mu_{C2} = & s_{j2}+s_{h2}+s_{f2}+s_{d2}+s_{b2}+\mu_{C1} \\
\mu_{c4} = & s_{b4}+s_{b3}+s_{j2}+s_{h2}+s_{f2}+s_{d2}+\mu_{c2}+s_{a2} = s_{b4}+s_{b3}+\mu_{C2}+s_{a2}+s_{b1}+s_{a0} \\
\xi_{A0} = & s_{B3}+s_{b3}+s_{B2}+s_{j2}+s_{h2}+s_{f2}+s_{e2}+s_{c2}+s_{a2}+1 \\
\xi_{B5} = & s_{B4}+s_{e4}+s_{c4}+s_{d4}+s_{b4}+s_{B1}+s_{B0}+s_{e0}+s_{d0}+s_{c0}+s_{b0}+\xi_{A0}+1 \\
\chi_{C3} = & s_{j4}+s_{h4}+s_{f4}+s_{d4}+s_{b4}+s_{j2}+s_{h2}+s_{f2}+s_{d2}+s_{j0}+s_{h0}+s_{f0}+s_{d0}+s_{b0} \\
\xi_{C3} = & s_{i4}+s_{g4}+s_{e4}+s_{c4}+s_{h4}+s_{d4}+s_{b4}+s_{b3}+s_{C2}+s_{j2}+s_{i2}+s_{g2}+s_{f2}+s_{e2} \\
& +s_{c2}+s_{a2}+s_{C1}+s_{C0}+s_{i0}+s_{h0}+s_{g0}+s_{e0}+s_{d0}+s_{c0}+s_{b0}
\end{cases}
$$

Figure 7: Working out the CCZ's pattern.