Extended Abstract: CNOT circuit synthesis for topologically-constrained quantum memories

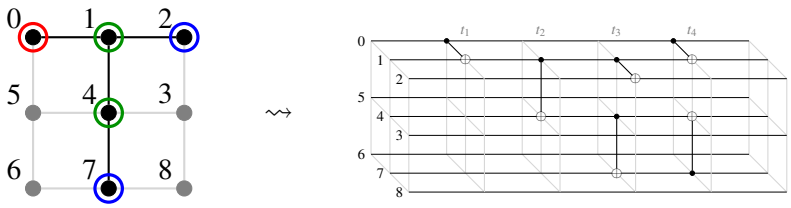
Aleks Kissinger
iCIS, Radboud University
aleks@cs.ru.nl

Arianne Meijer-van de Griend
Radboud University
ariannemeijer@gmail.com

This is an extended abstract. The full version of the paper is available online at:
https://arxiv.org/abs/1904.00633

Quantum circuits give a *de facto* standard for representing quantum computations at a low level. They consist of sequences of primitive operations, called quantum gates, applied to a register of quantum bits, or qubits. Increasingly, noisy intermediate-scale quantum (NISQ) computers with 10-80 qubits are becoming a reality. Popular physical realisations such as superconducting quantum circuits [12, 7, 14] and ion traps [2, 1, 5, 6] consist of qubits stored in the physical states of systems arranged in space, where two-qubit operations are typically only possible between pairs of systems that are adjacent in the 'coupling graph' of the architecture. Hence, when it comes to actually running a quantum computation on these architectures, logical qubits must be mapped to physical memory locations, and the circuit must be modified to only consist of 2-qubit operations between adjacent qubits in the physical architecture. Naïvely, this can be achieved by simply inserting swap gates to move a pair of qubits next to each other before each 2-qubit operation. However, this approach comes with an enormous overhead in terms of 2-qubit operations, each of which introduces a great deal more noise than a single qubit operation on most realistic architectures [1]. More sophisticated approaches have been proposed using SMT solvers and temporal planners [9, 13] as well as certain heuristic approximations thereof [8, 10, 3, 15]. Nevertheless, these are simply refinements of the basic 'search and swap' technique. As such, these approaches only take the topological structure of the circuit into account (i.e. which qubits are being acted upon) rather than semantic structure (i.e. the unitary being implemented), and hence miss out on opportunities for more efficient circuit mapping.

We present a new approach to quantum circuit mapping based on constrained Gaussian elimination, and apply it in the simplest case of mapping CNOT circuits. We modify a familiar technique based on a Gaussian elimination in such that a way that primitive row operations (implemented by CNOT gates) are only allowed between certain rows corresponding to neighbouring qubits. Hence, non-local row operations must be propagated through intermediate rows. We then give a simple strategy for identifying and using appropriate intermediate rows based on *Steiner trees*. For a subset S of the vertices of a graph, a Steiner tree is a minimal tree that covers all of the vertices in S. For our purposes, the subset S represents all of the rows that need to be reduced by primitive row operations (e.g. the rows containing non-zero entries below a pivot) whereas the tree itself gives us an efficient strategy for performing those reductions via nearest-neighbour CNOTs:



Submitted to:

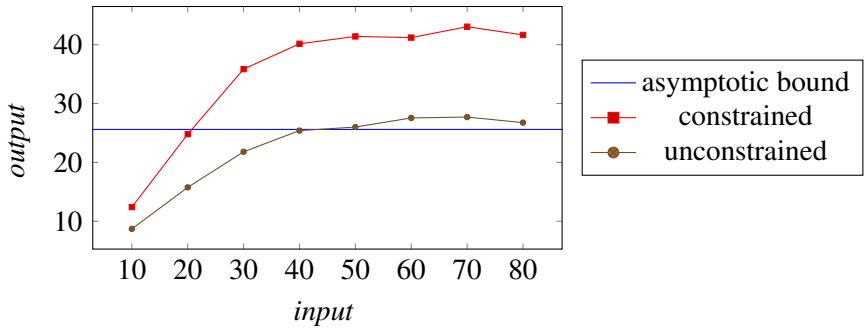


Figure 1: A plot of input vs. output CNOT gates when mapping on to a 9-qubit square grid, when computing and re-synthesising the circuit, both in the unconstrained case using the method described in [11] and respecting nearest-neighbour constraints using our method.

In particular, we use CNOTs to propagate row operations down toward the leaves of the tree then ultimately back up toward the root. The end result is CNOT circuit realising a given parity map involving only nearest-neighbour interactions. Much like other approaches, the size of the final circuit is very sensitive to the initial layout of the logical qubits within the physical architecture, especially for relatively small CNOT circuits. For us, a good choice of qubit positions means smaller Steiner trees, which in turn means fewer extra CNOTs getting applied. Thus, we perform a simple genetic algorithm-based optimisation procedure, with a cost function given as the total resulting CNOT count, to find a good choice of initial qubit locations.

To measure the effectiveness of our approach, we produce many random CNOT circuits on 9, 16, and 20 qubits, containing between 3 and 256 CNOT gates, and map them onto 5 different graph topologies: 3×3 and 4×4 square lattices, 16-qubit architectures of the IBM QX-5 and Rigetti Aspen devices, and the 20-qubit IBM Q20 Tokyo architecture. To compare the performance of our technique to general-purpose mapping techniques, we also map these CNOT circuits with the Rigetti QuilC compiler and $\mathsf{t}|\mathsf{ket}\rangle$ by Cambridge Quantum Computing. Using these as a baseline, we find an average savings in 2-qubit gates of 48% over QuilC and 36% over $\mathsf{t}|\mathsf{ket}\rangle$.[1] These results appear in Table 1.

For very deep CNOT circuits, the overheads of our mapping process are often negative. This is because the process we use computes the parity map associated to a CNOT circuit and re-synthesises the circuit using Gaussian elimination. In the unconstrained case, Patel *et al.* [11] gave a heuristic for (re-)synthesising generic CNOT circuits on n qubits with $O(n^2/\log(n))$ CNOTs. Indeed, using their algorithm for 9-qubit circuits, we see in Fig. 1 that the average CNOT count stabilises around the asymptotic bound of $9^2/\log_2(9)\approx25.6$. Our approach also stabilises, but at a higher number of CNOTs (~42 for 9 qubits). Interestingly, this is still well below the complexity of naïve unconstrained CNOT synthesis, which is $O(n^2)$.

Finally, we note that, while we applied these techniques to CNOT circuits, they extend straightforwardly to broader families of circuits where sythesis involves some form of Gaussian elimination. In particular, circuits consisting of CNOTs and arbitrary Z-phase rotations (i.e. *phase polynomial* circuits) can be synthesised using a version of this technique, and arbitrary Clifford+T circuits can be handled by modifying the circuit extraction procedure described by one of the authers in [4].

[1] All circuits can be found in QASM format at: https://github.com/Quantomatic/pyzx/tree/steiner_decomp/circuits/steiner

Architecture	# CNOTs	QuilC		t|ket⟩		Steiner	
9q-square	3	3.8	(26.67%)	3.6	(20.0%)	3.0	(**0.0**%)
9q-square	5	10.82	(116.4%)	6.4	(28.0%)	5.2	(4.0%)
9q-square	10	20.08	(100.8%)	16.95	(69.5%)	11.6	(16.0%)
9q-square	20	46.24	(131.2%)	40.75	(103.75%)	25.85	(29.25%)
9q-square	30	72.89	(142.97%)	66.15	(120.5%)	35.55	(18.5%)
16q-square	4	6.14	(53.5%)	5.8	(45.0%)	4.44	(11.09%)
16q-square	8	19.68	(146.0%)	12.95	(61.87%)	12.41	(55.14%)
16q-square	16	48.13	(200.81%)	36.2	(126.25%)	33.08	(106.78%)
16q-square	32	106.75	(233.59%)	94.45	(195.16%)	82.95	(159.22%)
16q-square	64	225.69	(252.64%)	203.75	(218.36%)	147.38	(130.28%)
16q-square	128	457.35	(257.3%)	436.25	(240.82%)	168.12	(31.34%)
16q-square	256	925.85	(261.66%)	922.65	(260.41%)	169.28	(**-33.88**%)
rigetti-16q-aspen	4	7.05	(76.25%)	7.15	(78.75%)	4.15	(3.75%)
rigetti-16q-aspen	8	28.2	(252.5%)	17.2	(115.0%)	11.22	(40.28%)
rigetti-16q-aspen	16	69.15	(332.19%)	52.0	(225.0%)	33.95	(112.19%)
rigetti-16q-aspen	32	147.3	(360.31%)	-	(-%)	101.75	(217.97%)
rigetti-16q-aspen	64	324.6	(407.19%)	-	(-%)	189.15	(195.55%)
rigetti-16q-aspen	128	664.65	(419.26%)	-	(-%)	220.75	(72.46%)
rigetti-16q-aspen	256	1367.89	(434.33%)	-	(-%)	222.15	(**-13.22**%)
ibm-qx5	4	6.75	(68.75%)	4.3	(7.5%)	4.0	(**0.0**%)
ibm-qx5	8	23.7	(196.25%)	14.75	(84.38%)	8.95	(11.87%)
ibm-qx5	16	60.5	(278.12%)	47.5	(196.88%)	26.55	(65.94%)
ibm-qx5	32	140.05	(337.66%)	122.95	(284.22%)	84.4	(163.75%)
ibm-qx5	64	301.05	(370.39%)	278.7	(335.47%)	152.65	(138.52%)
ibm-qx5	128	600.9	(369.45%)	597.65	(366.91%)	188.25	(47.07%)
ibm-qx5	256	1247.8	(387.42%)	1258.8	(391.72%)	193.8	(**-24.3**%)
ibm-q20-tokyo	4	5.5	(37.5%)	6.05	(51.25%)	4.0	(**0.0**%)
ibm-q20-tokyo	8	17.3	(116.25%)	12.0	(50.0%)	7.69	(**-3.82**%)
ibm-q20-tokyo	16	43.83	(173.96%)	29.05	(81.56%)	20.44	(27.74%)
ibm-q20-tokyo	32	93.58	(192.43%)	78.15	(144.22%)	66.93	(109.15%)
ibm-q20-tokyo	64	215.9	(237.34%)	181.25	(183.2%)	165.6	(158.75%)
ibm-q20-tokyo	128	432.65	(238.01%)	391.85	(206.13%)	237.64	(85.66%)
ibm-q20-tokyo	256	860.74	(236.23%)	789.3	(208.32%)	245.84	(**-3.97**%)

Table 1: The average number of CNOTs needed to map random CNOT circuits of various sizes. The first column shows the architecture mapped to and the second column the original number of CNOT gates. The remaining columns show the average 2-qubit gate count after mapping 20 random circuits. The percentages show mapping overheads (i.e. % of the original gate count that was added during mapping), where negative overheads indicate a mapped circuit that is smaller than the original.

References

[1] CJ Ballance, TP Harty, NM Linke, MA Sepiol & DM Lucas (2016): *High-fidelity quantum logic gates using trapped-ion hyperfine qubits*. Physical Review Letters 117(6), p. 060504.

[2] Joseph W Britton, Brian C Sawyer, Adam C Keith, C-C Joseph Wang, James K Freericks, Hermann Uys, Michael J Biercuk & John J Bollinger (2012): *Engineered two-dimensional Ising interactions in a trapped-ion quantum simulator with hundreds of spins*. Nature 484(7395), p. 489.

[3] Alexander Cowtan, Silas Dilkes, Ross Duncan, Alexandre Krajenbrink, Will Simmons & Seyon Sivarajah (2019): *On the qubit routing problem*. arXiv:1902.08091.

[4] Ross Duncan, Alex Kissinger, Simon Perdrix & John van de Wetering (2019): *Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus*. https://arxiv.org/abs/1902.03178.

[5] JP Gaebler, TR Tan, Y Lin, Y Wan, R Bowler, AC Keith, S Glancy, K Coakley, E Knill, D Leibfried et al. (2016): *High-Fidelity Universal Gate Set for Be 9+ Ion Qubits*. Physical Review Letters 117(6), p. 060505.

[6] WK Hensinger, S Olmschenk, D Stick, D Hucul, M Yeo, M Acton, L Deslauriers, C Monroe & J Rabchuk (2006): *T-junction ion trap array for two-dimensional ion shuttling, storage, and manipulation*. Applied Physics Letters 88(3), p. 034101.

[7] IBM (2019): *IBM Unveils World's First Integrated Quantum Computing System for Commercial Use*. https://newsroom.ibm.com/2019-01-08-IBM-Unveils-Worlds-First-Integrated-Quantum-Computing-System-for-Commercial-Use. Accessed: 2019-03-28.

[8] Gushu Li, Yufei Ding & Yuan Xie (2018): *Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices*. arXiv:1809.02573.

[9] Prakash Murali, Ali Javadi-Abhari, Frederic T Chong & Margaret Martonosi (2019): *Formal Constraint-based Compilation for Noisy Intermediate-Scale Quantum Systems*. Microprocessors and Microsystems.

[10] Alexandru Paler, Alwin Zulehner & Robert Wille (2018): *NISQ circuit compilers: search space structure and heuristics*. arXiv:1806.07241.

[11] Ketan N. Patel, Igor L. Markov & John P. Hayes (2008): *Optimal Synthesis of Linear Reversible Circuits*. Quantum Info. Comput. 8(3), pp. 282–294. Available at `http://dl.acm.org/citation.cfm?id=2011763.2011767`.

[12] Matthew Reagor, Christopher B Osborn, Nikolas Tezak, Alexa Staley, Guenevere Prawiroatmodjo, Michael Scheer, Nasser Alidoust, Eyob A Sete, Nicolas Didier, Marcus P da Silva et al. (2018): *Demonstration of universal parametric entangling gates on a multi-qubit lattice*. Science advances 4(2), p. eaao3603.

[13] Davide Venturelli, Minh Do, Eleanor Rieffel & Jeremy Frank (2018): *Compiling quantum circuits to realistic hardware architectures using temporal planners*. Quantum Science and Technology 3(2), p. 025004.

[14] R Versluis, S Poletto, N Khammassi, N Haider, DJ Michalak, A Bruno, K Bertels & L DiCarlo (2016): *Scalable quantum circuit and control for a superconducting surface code*. arXiv:1612.08208.

[15] Alwin Zulehner, Alexandru Paler & Robert Wille (2018): *An efficient methodology for mapping quantum circuits to the IBM QX architectures*. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.