Design of quantum logic circuits for a three-qubit refined Deutsch-Jozsa algorithm with optically controlled, solid-state quantum logic gates

A Del Duce and P Bayvel†

† University College London

Abstract: Using a genetic programming approach we design quantum logic circuits suitable for the experimental demonstration of a three-qubit quantum computation prototype based on optically controlled, solid-state quantum logic gates.

1 Introduction.

Quantum computation is a computational model based on the interaction of two-level quantum mechanical systems typically termed qubits which has attracted great interest for its potential for increased computational power[1]. However, because of the demanding technological challenges, (particularly the issue of decoherence, i.e. the unwanted perturbation of the qubits by the environment which causes a loss of the stored quantum information), large-scale quantum computation is still far from the implementation stage and a number of physical systems have been proposed in the past aiming at the developing scalable systems[2]. One of these proposals is based on the optical control of the interactions between qubits stored in the electron spin of donors in a solid-state (possibly silicon) substrate[3]. This model, sometimes referred to as *SFG* model[4], aims at achieving large-scale quantum computation by exploiting the good tolerance towards decoherence of donor electron spin qubits in semiconductors[5] in conjunction with the optical control which allows to avoid the complicated fabrication processes for the exact placement of control electrodes used in other implementations[3]. The SFG model has been studied intensively during the last years and the implementation of a small-scale quantum computation prototype represents an important next step towards testing the potential of this technology.

In this paper, we work towards this aim by developing quantum circuits suitable for the experimental demonstration of a three-qubit quantum computation prototype based on the SFG model. We use the genetic programming approach adapted to quantum circuit design proposed by Williams and Gray[6] for designing quantum logic circuits, specifically tailored for SFG quantum logic gates, which implement a three-qubit refined Deutsch-Jozsa (DJ) algorithm[7]. First, we develop circuits based on ideal and technology independent quantum gates, then, we observe how the circuits' performance changes when the ideal gates are approximated by quantum gates produced within the SFG model, aiming at identifying solutions characterized by short computational time in order to protect the circuit from unpredictable errors which may perturb the system during an experimental implementation.

2. Quantum circuit design

In quantum computation, the implementation of an algorithm requires to apply a unitary transformation U_{comp} to a quantum register comprising N qubits. However, the transformation U_{comp} is an abstract object which, when implementing it experimentally on a quantum computer, is typically decomposed into a sequence of single qubit operations (i.e. quantum gates which alter the state of a single qubit) and two-qubit gates (quantum gates which lead to the interaction of two qubits)[1]. The specific types of gates available for the sequence depend on the particular technology used for the quantum computational system under study. Hence, given a quantum register of N qubits, the design of a quantum logic circuit implementing a specific algorithm corresponds to finding a decomposition into a well defined sequence of single- and two-qubit operations of the unitary $2^N x 2^N$ matrix U_{comp} which describes the algorithm[1]:

$$U_{comp} = U_j \cdot U_m \cdot U_k \cdot U_j \cdot U_m \cdot U_j \qquad (1)$$

The operators on the right-hand side of (1) are unitary operators U_α describing single- or two-qubit operations which are taken out of an ensemble $\alpha=\{j,k,l,m,n...\}$ comprising all the gates allowed by the

given technology. Here, to perform the decomposition given in (1), we use the genetic programming approach developed by Wiliams and Gray[6]: first, a population of random circuits is generated. Each circuit U_{circ} is represented by a random sequence of operators from the available set of gates. The fitness, which defines a comparison between the ideal transformation U_{comp} and how well a synthesized circuit U_{circ} actually implements it, is then evaluated for each element of the population. As a fitness parameter we use the average fidelity AF(see [8], for example):

$$AF=|Tr(U^{\dagger}_{comp}U_{circ})/2^N|^2 \quad (2)$$

AF returns 1 when U_{circ} implements U_{comp} exactly (up to an irrelevant phase difference) and decreases in case of imperfect implementation. Once the initial population has been created, the algorithm repeats following steps: a new generation of circuits is bred by randomly picking elements out of the previous generation, with circuits with higher fitness having a higher probability of being selected, and forming new circuits through crossover or mutation. In crossover, random fractions of two selected circuits are combined to form the new element while mutation forms a new circuit by adding, deleting or perturbing a random gate in the selected circuit. Once the new generation has been bred, the fitness of all circuits is evaluated. The algorithm continues breeding new generations until at least one element in the population has fitness higher than a desired threshold value (or if a maximum number of allowed iterations has been reached). The idea behind the genetic programming approach is that, mimicking the evolutionary process of nature, by composing or mutating well-performing circuits, generations of better-performing circuits might subsequently be built.

In our design process, we exploit for single-qubit operations the operator $R_z(\theta)$[1], shown in expression (3a). As described in [3],[8], techniques for implementing single-qubit operations compatible with the SFG model have been already demonstrated experimentally in the past and will therefore not be discussed here. In terms of two-qubit gates, we exploit two types of gates. We start using the ideal controlled-phase (CP) gate (expression (3b)), which is a gate commonly used in quantum computation[1]. Then, we repeat the design process after substituting the ideal CP gates with approximations obtained via the SFG model. In SFG quantum computation, interactions between two qubits carried by the electron spin of donors are mediated by a control particle placed in their proximity[3]. All the wavefunctions of this three-particle system are well separated and do not interact in their ground state. Instead, if the electron of the control particle is brought to an excited state through an optical pulse, its wavefunction spreads and overlaps with the adjacent wavefunctions of the qubit electrons, hence, leading to an effective interaction between them. The interaction is stopped by a second de-exciting pulse which returns the control electron to the ground state. For a discrete set of pulse interleave times T_i which can be shown to depend on two integers M and N, the qubits interact without losing information to the control particle and the two-qubit interaction is described by the matrix given in (3c) where J is the strength of the exchange interaction characterizing the qubits-control atom system, B is a static magnetic field term and $f=B/J$[8]. A variety of different gates can be produced within the SFG model, including, as shown in [8], approximations of CP gates. Expressions (3a), (3b) and (3c) summarise the gates used in our work:

$$R_z(\theta)=\begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \text{(3a)}; \qquad CP=\begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&-1 \end{bmatrix} \text{(3b)}; \qquad SFG(M,N)=e^{i(J-B)T}\begin{bmatrix} e^{-i[(3-f)J+2B]T} & 0 & 0 & 0 \\ 0 & \frac{\left[(-1)^M+e^{-i(1-f)JT}\right]}{2} & \frac{\left[(-1)^M-e^{-i(1-f)JT}\right]}{2} & 0 \\ 0 & \frac{\left[(-1)^M-e^{-i(1-f)JT}\right]}{2} & \frac{\left[(-1)^M+e^{-i(1-f)JT}\right]}{2} & 0 \\ 0 & 0 & 0 & e^{2iBT}(-1)^N \end{bmatrix} \text{(3c)};$$

We use the genetic programming approach to design circuits for the core operator U_f of the refined DJ algorithm, assuming a quantum register of three qubits as this represents the simplest structure with maximum functionality[7]. We label the 3 qubits as '0', '1' and '2', and assume that each can interact with the others. In terms of two-qubit interactions, we assume that, for each design procedure, three-two qubit gates are available: U_1, U_2 and U_3 (respectively between qubits 0 and 1, 1 and 2 and 0 and 2), which may either be ideal CP gates or SFG gates. The core operator U_f of a three-qubit refined DJ algorithm is (except for a constant case in which it corresponds to the identity matrix) an 8x8 diagonal

operator, with a balanced distribution of '1's and '-1's on its diagonal, which implements a function f chosen by an oracle used in the algorithm. There are 35 different balanced functions and, therefore, 35 different quantum circuits which implement them[9].

4.1 Quantum circuits exploiting ideal CP gates

We have run the genetic programming algorithm for all the 35 balanced functions using ideal CP gates for two-qubit interactions. The results are presented in Table 1, in which for each function f_{1J} the corresponding circuit sequence returned by the genetic programming algorithm is reported. The circuits have been split according to the number of two-qubit gates contained in the sequence and the subscripts of the quantum gates labels describe to which qubits in the quantum register the gates are applied. We have used the same hexadecimal codification of the functions as in the results presented by Kim et al where circuits for the same algorithm have been derived through a generator expansion technique for the case of a nuclear magnetic resonance quantum computer[9]. Here, we chose the genetic programming approach proposed by Williams and Gray[6] because it is more flexible in terms of the quantum gates which can be used in the decomposition process.

0 Two-qubit gates		1 Two-qubit gate		2 Two-qubit gates		3 Two-qubit gates	
f_{0F}	$R_{2z}(\pi)$	f_{1E}	$R_{2z}(\pi)CP_{01}$	f_{1B}	$CP_{01}\ CP_{02}R_{2z}(\pi)$	f_{17}	$CP_{01}\ CP_{02}\ CP_{12}$
f_{33}	$R_{1z}(\pi)$	f_{2D}	$R_{0z}(-\pi)CP_{01}R_{2z}(\pi)$	f_{1D}	$R_{2z}(\pi)CP_{01}\ CP_{12}$	f_{2B}	$CP_{12}\ CP_{02}\ CP_{01}\ R_{1z}(-\pi)R_{2z}(\pi)$
f_{3C}	$R_{1z}(\pi)R_{2z}(\pi)$	f_{36}	$CP_{02}R_{1z}(\pi)$	f_{27}	$CP_{01}\ CP_{02}\ R_{1z}(-\pi)$	f_{4D}	$CP_{02}\ R_{2z}(\pi)CP_{01}CP_{12}R_{0z}(\pi)$
f_{55}	$R_{0z}(\pi)$	f_{39}	$CP_{02}R_{1z}(-\pi)R_{2z}(-\pi)$	f_{2E}	$R_{1z}(-\pi)R_{2z}(-\pi)CP_{12}\ CP_{01}$	f_{71}	$CP_{12}\ R_{1z}(-\pi)CP_{01}CP_{02}R_{0z}(\pi)$
f_{5A}	$R_{0z}(\pi)R_{2z}(\pi)$	f_{4B}	$CP_{01}R_{0z}(\pi)R_{2z}(-\pi)$	f_{35}	$R_{1z}(\pi)CP_{02}\ CP_{12}$		
f_{66}	$R_{0z}(\pi)R_{1z}(\pi)$	f_{56}	$CP_{12}R_{0z}(-\pi)$	f_{3A}	$CP_{12}\ R_{2z}(\pi)\ CP_{02}\ R_{1z}(\pi)$		
f_{69}	$R_{0z}(\pi)R_{1z}(\pi)\cdot$ $\cdot R_{2z}(\pi)$	f_{59}	$R_{2z}(\pi)CP_{12}R_{0z}(-\pi)$	f_{47}	$CP_{02}\ R_{1z}(\pi)CP_{12}$		
		f_{63}	$CP_{02}R_{0z}(\pi)R_{1z}(\pi)$	f_{4E}	$R_{2z}(\pi)CP_{01}\ CP_{02}\ R_{0z}(-\pi)$		
		f_{65}	$R_{0z}(\pi)R_{1z}(-\pi)CP_{12}$	f_{53}	$CP_{02}\ R_{0z}(\pi)CP_{12}$		
		f_{6A}	$R_{0z}(-\pi)R_{1z}(\pi)R_{2z}(-\pi)CP_{12}$	f_{5C}	$CP_{12}\ R_{0z}(-\pi)\ CP_{02}\ R_{2z}(-\pi)$		
		f_{6C}	$R_{1z}(-\pi)CP_{12}R_{0z}(-\pi)R_{2z}(\pi)$	f_{72}	$R_{1z}(-\pi)CP_{01}\ R_{0z}(-\pi)CP_{02}$		
		f_{78}	$R_{1z}(-\pi)R_{2z}(\pi)CP_{12}R_{1z}(-\pi)$	f_{74}	$R_{1z}(-\pi)CP_{01}\ CP_{12}\ R_{0z}(\pi)$		

Table 1: Quantum circuits for all 35 balanced functions of the refined DJ algorithm

All the reported circuits implement the corresponding operator exactly (average fidelity $AF=1$). Comparing Table 1 and the results presented in [9], it can be seen that the circuits obtained with the two different methods require the same number of two-qubit gates. In terms of single-qubit gates, the same length of circuits has been found for all functions belonging to the group requiring 0 two-qubit gates. For the remaining functions, we found that 2D, 39, 63, 59, 65, D8, AC, CA, 27, 47, 53, 1D, 35,17 designed with our genetic programming algorithm required 1 less single-qubit gate, functions 36 and 56 two less, while function 4D required one single-qubit gate more. However, these differences might not be caused by the different methods used for the decomposition, but could also have been induced by the different two-qubit gates used.

4.1 Quantum circuits exploiting SFG gates approximating CP gates

Focusing on the function f_{17} (whose operator U_{17} has the balanced string [1 1 1 -1 1 -1 -1 -1] on its diagonal and belongs to the class requiring 3 two-qubit gates, therefore representing one of the functions with maximum complexity) we ran the genetic programming algorithm using SFG gates approximating the CP gates. Aiming at obtaining circuits with short computational time in order to reduce the number of errors which may accumulate during computation, we tested two different sets of SFG gates. A first set (Set A), exploiting SFG gates which approximate CP gates with high precision (AF>0.999) but with long gate operation times (T_i>80ns), and a second set (B) which only approximated the CP gates with AF>0.99 but with gate operation time T_i <10ns. Choosing the same magnetic field term B of 0.136meV used in a recent study on SFG quantum gates[4] and using the equations given in [8], we obtained the gate parameters shown in Table 2 for the two sets of gates which returned, respectively, the two circuits shown in (4) for the operator corresponding to f_{17}:

Set A (AF>0.999)	Set B (AF>0.99)
U_{1A}=SFG(1595,2137), J=6.15GHz, T=157.95ns	U_{1B}=SFG(124,142), J=51.93GHz, T_1=2.63ns
U_{2A}=SFG(1584,2177), J=1.74GHz, T=81.17ns	U_{2B}=SFG(137,156), J=54.37GHz, T_2=2.77ns
U_{3A}=SFG(815,904), J=16.44GHz, T=100.84ns	U_{3B}=SFG(143,162), J=56.77GHz, T_3=2.77ns

Table 2: Parameters of two sets of SFG gates approximating CP gates with a different degree of precision

$$U_{17appA} = R_{z0}(-0.001)\, U_{1A}\, R_{z0}(0.008)\, U_{3A}\, R_{z1}(-0.011)\, R_{z2}(0.019)\, U_{2A}$$
$$U_{17appB} = U_{2B}\, R_{z1}(0.038)\, U_{1B}\, R_{z0}(0.059)\, U_{3B}\, R_{z2}(0.183) \quad (4)$$

Both circuits require extra single-qubit rotations which, as can be seen from Table 1, were not necessary in the ideal case. These compensate for part of the non-idealities introduced by the SFG quantum transformations as opposed to the ideal CP gates. In terms of performance, the first circuit, U_{17appA}, approximates the ideal transformation with an average fidelity AF>0.999. However, while U_{17appB} approximates the ideal circuit only with an average fidelity of ~0.988, it does so exploiting two-qubit quantum gates more than ~30 times faster. Hence, at the expenses of a slightly lower (yet still high) precision, it implements the desired transformation in less time and, therefore, better protecting the circuit from further, unpredictable errors (for example due to decoherence) which may perturb the system during an experimental demonstration.

4. Conclusions.

Using a genetic programming approach, we have developed quantum circuits for an SFG quantum computational system which implement a three-qubit refined Deutsch-Jozsa algorithm. Through a compromise between precision of the solution and length of computation time, we have identified a convenient solution which would be suitable for the experimental realization of a three-qubit SFG quantum computer.

Acknowledgments.

We are grateful to Prof. A.M.Stoneham, Dr.A.Harker, Dr.P.T.Greenland and Prof. A.J.Fisher (UCL) for valuable discussions throughout the development of this work. This work was supported by RCUK and EPSRC, through the Basic Technology Programme, and the Royal Society.

References.

[1] M.A. Nielsen, I.L.Chunag, *Quantum Computation and Quantum Information*, Cambridge: Cambridge University Press, 2003

[2] P. Zoller, T. Beth, D. Binosi, R. Blatt, H. Briegel, D.Bruss et al., "Quantum information processing and communication – Strategic report on current status, visions and goals for research in Europe", *European Physical Journal D*, vol. 36, Nov. 2005, pp. 203-228

[3] A.M. Stoneham, A.J. Fisher, P.T. Greenland, "Optically driven silicon-based quantum gates with potential for high-temperature operation", *J. Phys .:Condens. Matter*, vol. 15, Jul. 2003, pp.L447-L451

[4] A. Kerridge, A.H. Harker, A.M. Stoneham, "Electron dynamics in quantum gate operation",*J. Phys.: Condens. Matter*, vol. 19, Jul. 2007, article number 282201

[5] A.M. Tyryshkin, S.A.Lyon, A.V.Astashkin, A.M.Raitsimring, "Electron spin relaxation times of phosphorus donors in silicon", *Phys. Rev. B*, vol.68, Nov. 2003, Article number 193207

[6] C.P.Williams,A.G. Gray:"Automated design of quantum circuits",*in Quantum Computing and Quantum Communications,Book Series:Lecture Notes in Computer Science*,vol.1509,Jan.1999, pp.113-125

[7] D. Collins, K.W. Kim, W.C. Holton, "Deutsch-Jozsa algorithm as a test for quantum computation", *Phys. Rev. A*, vol. 58, Sept.1998, pp. R1633-R1636

[8] R. Rodriquez, A.J. Fisher, P.T.Greenland, A.M. Stoneham, "Avoiding entanglement loss when two-qubit quantum gates are controlled by electronic excitation", *J. Phys.: Condens. Matter*, vol.16, Apr. 2004, pp.2757-2772

[9] J. Kim, J.S. Lee, S. Lee, C. Cheong, "Implementation of the refined Deutsch-Jozsa algorithm on a three-bit NMR quantum computer", *Phys. Rev. A*, vol. 62, Aug. 2000, Article number 022312