Simulation of Quantum Computers 

H. De Raedt[1], K. Michielsen[2], A.H. Hams[1], S. Miyashita[3], and K. Saito[3]

[1] Institute for Theoretical Physics and Materials Science Centre, University of Groningen, Nijenborgh 4, NL-9747 AG Groningen, The Netherlands
[2] Laboratory for Biophysical Chemistry, University of Groningen, Nijenborgh 4, NL-9747 AG Groningen, The Netherlands
[3] Department of Applied Physics, School of Engineering, University of Tokyo, Bunkyo-ku, Tokyo 113, Japan

Abstract. We describe a simulation approach to study the functioning of Quantum Computer hardware. The latter is modeled by a collection of interacting spin-1/2 objects. The time evolution of this spin system maps one-to-one to a quantum program carried out by the Quantum Computer. Our simulation software consists of code that solves the time-dependent Schrödinger equation for a sequence of time-dependent Hamiltonians and a graphical user interface to program and control the solver. We present results of simulations on a 4-qubit Quantum Computer executing Grover's database search algorithm.

1 Introduction

Recent theoretical work has shown that a Quantum Computer (QC) has the potential of solving certain computationally hard problems such as factoring integers [1,2] and searching databases much faster than a conventional computer [3]. In all this work the operation of a QC is described in terms of highly idealized transformations on the qubits [1–4] . In conventional digital circuits (which may be build using e.g. semiconductor devices) the internal working of each basic unit is irrelevant for the logical operation of the whole machine (but extremely relevant for the speed of operation and the cost of the machine). This is not the case for a QC: The internal quantum dynamics of each elementary constituent is a key ingredient of the QC itself. In a QC the logical operation and the physical realization of the qubits are intimately related and cannot be disentangled from each other. There is ample evidence that simulation is an essential part of the design process of each new generation of microprocessors. This will even be more so in the case of QC's. Software to simulate physical models representing hardware implementations of a QC will be essential for the design and operation of QC's. The present paper serves to illustrate the use of such a piece of software.

2 Physical model of a Quantum Computer

Primitive 2-qubit QC's have been implemented using nuclear magnetic resonance (NMR) [5–8]. Future technologies may use Josephson junctions [9].

Physically all this QC hardware can be modeled in terms of S=1/2 spins (the qubits) that evolve in time according to the time-dependent Schrödinger equation (TDSE) $i\partial/\partial t|\Phi(t)\rangle = H(t)|\Phi(t)\rangle$, where $|\Phi(t)\rangle$ describes the state of the whole QC at time t. We use units such that $\hbar = 1$. The time-dependent Hamiltonian $H(t)$ takes the form

$$H(t) = -\sum_{j,k=1}^{L}\sum_{\alpha=x,y,z} J_{j,k,\alpha}(t) S_j^\alpha S_k^\alpha \\ -\sum_{j=1}^{L}\sum_{\alpha=x,y,z} \left(h_{j,\alpha,0}(t) + h_{j,\alpha,1}(t)\sin(f_{j,\alpha}t + \varphi_{j,\alpha})\right) S_j^\alpha, \quad (1)$$

where the first sum runs over all pairs P of spins, S_j^α denotes the α-th component of the spin-1/2 operator representing the j-th qubit, $J_{j,k,\alpha}(t)$ determines the strength of the interaction between the qubits labeled j and k, $h_{j,\alpha,0}(t)$ and $h_{j,\alpha,1}(t)$ are the static (magnetic) and oscillating field acting on the j-th spin respectively. The frequency and phase of the periodic field are denoted by $f_{j,\alpha}$ and $\varphi_{j,\alpha}$. The number of qubits is L and the dimension of the Hilbert space $D = 2^L$. Hamiltonian (1) captures the physics of most candidate technologies for building QC's.

A quantum algorithm consists of a sequence of elementary operations that change the state $|\Psi\rangle$ of the quantum processor according to the TDSE, i.e. by (a product of) unitary transformations. Each elementary operation is completely specified by its Hamiltonian (1), i.e. the values of all the J's and h's (which are kept constant during the operation) and the time τ it takes to complete the operation. During this time interval the only time-dependence of $H(t)$ is through the (sinusoidal) modulation of the fields on the spins.

Formally the solution of the TDSE can be expressed in terms of the unitary transformation $U(t+\tau,t) \equiv \exp_+(-i\int_t^{t+\tau} H(u)du)$, where $\exp_+$ denotes the time-ordered exponential function. To solve the TDSE we use $U(t+\tau,t) = U(t+m\delta,t) = U(t+m\delta,t+(m-1)\delta)\cdots U(t+2\delta,t+\delta)U(t+\delta,t)$ and replace each $U(t+(n+1)\delta,t+n\delta)$ by a Suzuki product-formula [10–16]. For all practical purposes, the results obtained by this technique are indistinguishable from the exact solution of the TDSE.

3 Grover's database search algorithm

Finding the needle in a haystack of N elements requires $\mathcal{O}(N)$ queries on a conventional computer [17]. Grover has shown that a QC can find the needle using only $\mathcal{O}(\sqrt{N})$ attempts [3]. This algorithm has been implemented on a 2-qubit NMR QC for the case of a database containing four items [6,8]. Assuming a uniform probability distribution for the needle, for $N = 4$ the average number of queries required by a conventional algorithm is 9/4 [8,17]. With Grover's QA the correct answer can be found in a single query [6,8].

For the 2-qubit QC, Grover's algorithm may be implemented by applying the sequences of pulses [8,15]

$$U_0 = X_1\bar{Y}_1X_2\bar{Y}_2I(\pi)X_1\bar{Y}_1X_2\bar{Y}_2I(\pi) \ , \tag{2a}$$
$$U_1 = X_1\bar{Y}_1X_2\bar{Y}_2I(\pi)X_1\bar{Y}_1\bar{X}_2\bar{Y}_2I(\pi) \ , \tag{2b}$$
$$U_2 = X_1\bar{Y}_1X_2\bar{Y}_2I(\pi)\bar{X}_1\bar{Y}_1X_2\bar{Y}_2I(\pi) \ , \tag{2c}$$
$$U_3 = X_1\bar{Y}_1X_2\bar{Y}_2I(\pi)\bar{X}_1\bar{Y}_1\bar{X}_2\bar{Y}_2I(\pi) \ , \tag{2d}$$

where the U_n correspond to the case where the needle is in position n, X_i (Y_i) denotes a clock-wise rotation of spin i about $\pi/2$ along the $x(y)$-axis, and $\bar{X}_i$ ($\bar{Y}_i$) stands for the inverse operation. The two qubits "communicate" with each other through the operation $I(a) \equiv e^{-iaS_1^zS_2^z}$. These sequences are to be applied on the uniform superposition state which is obtained by letting the sequence $\bar{X}_2\bar{X}_2\bar{Y}_2\bar{X}_1\bar{X}_1\bar{Y}_1$ act on the state with all spins up [8,15]. Experiments [6,8] and simulations [15,16] using sequences (2) show that the NMR QC returns the correct answer.

The implementation described above uses the two qubits, e.g. the two nuclear spins of the molecules, for two different tasks. First they are used to store the information contained in the database and second they are used to carry out Grover's algorithm to query the very same two qubits. Although this is sufficient to demonstrate the realization of a 2-qubit QC, to be useful in practice the database and quantum processor should have their own qubits. This obviously requires at least 4 qubits, two to hold the database information (in superposition state) and two to process the query. Furthermore we require an operation to copy (not clone) the state of the database into the quantum processor. This operation should transfer the amplitudes of an arbitrary linear combination of spin-up and spin-down of e.g. spin 1 to e.g. spin 3, initially in a state of spin up. More specifically we want $C_{1,3}(a|\uparrow\rangle_1 + b|\downarrow\rangle_1)|\uparrow\rangle_3 = |\uparrow\rangle_1(a|\uparrow\rangle_3 + b|\downarrow\rangle_3)$. In principle this can be done by a network of Controlled-NOT gates [4] but we have chosen to adopt another approach. It is easy to see that a spin Hamiltonian of the XY type performs the required transfer of amplitudes, up to some phase factors that can be removed by single-spin rotations. A short calculation shows that the operation

$$C_{1,3} = Y_3\bar{X}_3\bar{Y}_3e^{-i\tau J(S_1^xS_3^x+S_1^yS_3^y)} \ , \tag{3}$$

with $\tau J = \pi$ performs the desired task.

We have tested these ideas by simulating a 4-qubit NMR QC on our Quantum Computer Emulator (QCE) [16]. In Fig.1 we show a picture of the QCE window, for the case where the needle is in position 2, corresponding to the sequence U_2 in the 2-qubit case. The left most panel contains the elementary operations, called micro-instructions (MI's, e.g. "MI -X1", corresponding to $\bar{X}_1$), that completely specify the QC hardware (i.e. the spin model and the

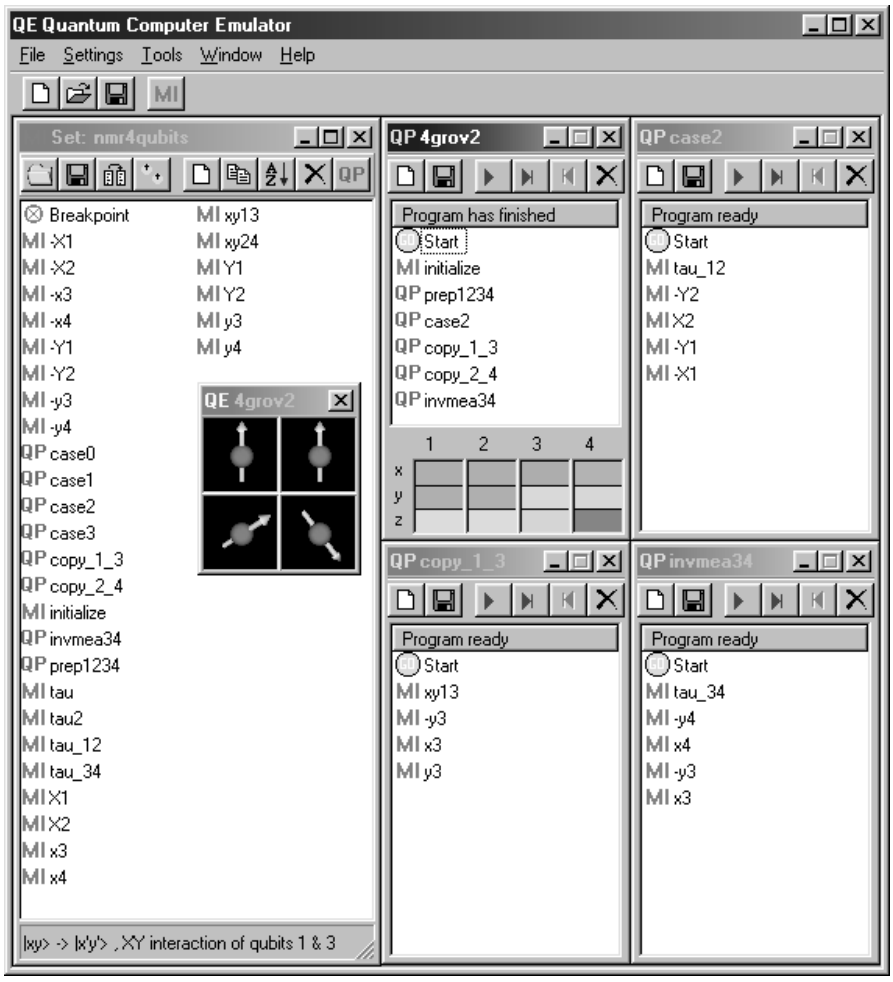

Fig. 1. Picture of the Quantum Computer Emulator running Grover's database search algorithm

.

sinusoidal pulses that may be applied). Also included in this panel are quantum programs (QP's, e.g. "QP case 2") that may be used as "subroutines". These quantum programs are build from MI's and QP's. Each MI and hence the full QCE is fully programmable. The QP *4grov2* is the 4-qubit quantum program to search for the needle in the database represented by qubits 1 and 2. Subroutines *prep1234* and *case2* store the information in the database(for case 2) and initialize the quantum processor (qubits 3 and 4). The query of the database consists of two steps: Copy the information from the database into qubits 3 and 4 (*copy_1_3* and *copy_2_4*) and determine the position of the

item in the database by inverting the state held in qubits 3 and 4 around the mean [3,8,16]. The final result of the query is shown in color-coded form in the grid at the bottom of the QP *4grov2* window. The QCE can also animate the motion of the spins as the execution of the quantum program proceeds. A snapshot of the 3D representation of the final state of the four qubits is also shown in Fig.1.

The expectation value of the z-components of spin 3 and 4 gives the position of the item in binary form. It is clear that spins 3 and 4 (bottom part of animation window) are not perfectly down and up. This is due to the fact that the pulses used in the NMR implementation do not exactly correspond to the ideal rotations assumed in the theoretical analysis (by changing the MI's the QCE also simulates an ideal QC). For 2-qubit NMR QC's we have shown that calculations with sequences obtained by interchanging logically commuting operations may yield unpredictable results [15]. Although the copy-qubit operation (3) does not seem to generate instabilities by itself the mere fact that additional elementary steps, and hence more non-commuting operations, are required enhances the intrinsic instabilities already observed in the 2-qubit case [15]. Research exploring the effect of these instabilities on the performance of QC hardware is in progress.

References

1. P. Shor: in *Proc. 35th Annu. Symp. Foundations of Computer Science*, S. Goldwasser ed., (IEEE Computer Soc., Los Alamitos CA, 1994) 124
2. I.L. Chuang, R. Laflamme, P.W. Shor, and W.H. Zurek: Science **230**, 1663 (1995)
3. L.K. Grover: in *Proc. of the 28th Annual ACM Symposium of Theory of Computing* (ACM, Philadelphia 1996)
4. A. Ekert, and R. Jozsa: Rev. Mod. Phys. **68**, 733 (1996)
5. J.A. Jones, and M. Mosca: J. Chem. Phys. **109**, 1648 (1998)
6. J.A. Jones, M. Mosca, and R.H. Hansen: Nature (London) **393**, 344 (1998)
7. I.L. Chuang, L.M.K. Vandersypen, Xinlan Zhou, D.W. Leung, and S. Lloyd: Nature **393**, 143 (1998)
8. I.L. Chuang, N. Gershenfeld, and M. Kubinec: Phys. Rev. Lett. **80**, 3408 (1998)
9. Y.Nakamura, Yu. A. Pashkin and J.S. Tsai: Nature **398**, 786 (1999)
10. M. Suzuki, S. Miyashita, and A. Kuroda: Prog. Theor. Phys. **58**, 1377 (1977)
11. H. De Raedt: Comp. Phys. Rep. **7**, 1 (1987)
12. M. Suzuki,: Proc. Japan Acad. **69**, Ser. B 161 (1993)
13. P. de Vries, and H. De Raedt: Phys. Rev. B **47**, 7929 (1993)
14. H. De Raedt and K. Michielsen: Comp. in Phys. **8**, 600 (1994)
15. H. De Raedt, A.H. Hams, K. Michielsen, S. Miyashita, and K. Saito: J. Phys. Soc. Jpn. (Supp.) **69**, 401 (2000)
16. H. De Raedt, A.H. Hams, K. Michielsen, and K. De Raedt: Comp. Phys. Comm. (in press) (quant-ph/9911041)
17. T.H. Cormen, C.E. Leiserson, R.L. Rivest: *Introduction to Algorithms* (MIT Press, Cambridge 1994)