ISSCI031

Main Menu

World Automation Congress

Fifth International Symposium on Soft Computing for Industry

Seville, Spain
June 28th-July 1st, 2004

Hardware Implementation Of Fast Quantum Searching Algorithms And Its Application In Quantum Soft Computing And Intelligent Control

D. M. Porto, S.V. Ulyanov and S. A. Panfilov

HARDWARE IMPLEMENTATION OF FAST QUANTUM SEARCHING ALGORITHMS AND ITS APPLICATION IN QUANTUM SOFT COMPUTING AND INTELLIGENT CONTROL

D. M. PORTO, STMicroelectronics Srl., Italy, massimo.porto@st.com
S.V. ULYANOV, YAMAHA Motor Europe N.V. R&D Office, Italy, ulyanov@tin.it
S. A. PANFILOV, YAMAHA Motor Europe N.V. R&D Office, Italy, sergueip@tin.it

ABSTRACT

The general approach for quantum algorithm (QA) simulation on classical computer is introduced. Efficient fast algorithm and corresponding software (SW) for simulation of Grover's quantum search algorithm (QSA) in large unsorted database is presented. Comparison with common QA simulation approach is demonstrated. Hardware (HW) design method of main quantum operators that are used in simulation of QA is described. Grover's QSA as benchmark of HW design method application is presented. This approach demonstrates the possibility of classical efficient simulation of quantum algorithm gates (QAG). Demo of HW-prototype for 3-qubit Grover' quantum searching algorithm is presented.

Keywords: Quantum algorithm gate, classical efficient simulation, fast algorithm, hardware implementation

1. INTRODUCTION

We describe simulation and design method of main quantum operators, and HW implementation of QAG for fast search in large database and related topics concerning the control of a process, including search-of-minimal intelligent operations. This method is very useful for minimal efforts of searching among a set of values and in particular is the first step for the realization of a HW control systems exploiting artificial intelligence in order to control a non-linear process in a robust way or in order to search in a database efficiently. The presented HW performs all the functional steps of the Grover's QSA. By suitable changes of traditional matricial approach, a modular *n*-qubit-hybrid structure is realized in order to prove the usefulness of iterations of the gate, which provide a higher probability of exact solution finding. A minimum-entropy based method is adopted as a termination condition criterion and realized in a digital part together with display output.

The possibility of providing an external clock signal for iteration management allows us to implement a very fast Grover's QSA, many times faster than the corresponding software (SW) realization [1], and less sensitive to qubits increment. In general, any QA can be represented as a circuit of smaller quantum gates as it is demonstrated on the Figure 1a [2]. Using transformation rules described in [2] the circuit can be transformed into the corresponding quantum gate (Figure 1b). After applying the transformation procedures we can design hardware structure Figure 1c and corresponding electronic circuit presented in Figure 1d. Figure 1e shows the computer assembly simulation process of the Grover's quantum gate using the result presented in Figure 1a and 1b for different combinations of input functions.

2. HW IMPLEMENTATION OF MAIN QUANTUM ALGORITHM OPERATORS

It has been found [3 - 5] a new method and circuit that implements the operations performed in second and third step of a quantum algorithm (the so-called entanglement and interference operators), able to perform Grover interference without products. The proposed circuit, which is one of the first hardware realization of a QA, is also the first one not based on matrices products but on functional relation between input and output vectors.

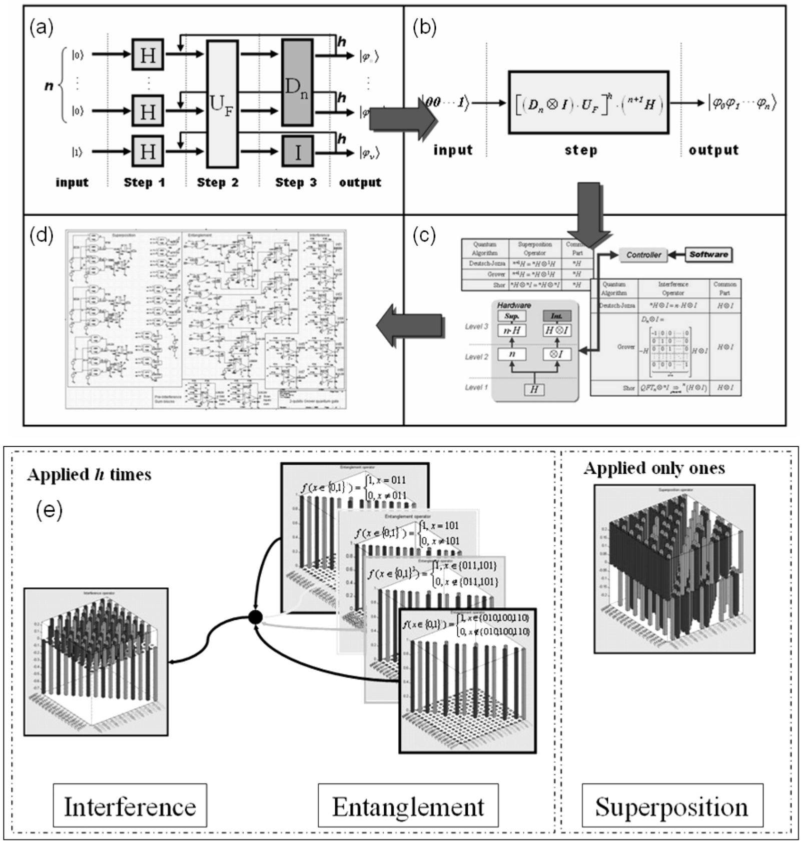


Figure 1: Circuit representation and computer assembly of QA gate and corresponding hardware realization

A general form of the entanglement output vector $U_F = G$ in can be the following:

$$G = [g_1, g_2, \ldots, g_i, \ldots, g_{2^{n+1}}] \tag{1}$$

where $g_i = y_i \oplus f_{1+\frac{INT(i-1)}{2}}$ and y_i is the general term of superposition transformed in a suitable binary value. The so-called superposition vector is fixed if we choose as input the canonical base. In order to find a suitable input-output relation, some particular properties of matrix $D_n \otimes I$ have to be taken in consideration .The generic element v_i of V can be written as follows in function of g_i :

$$v_i = \begin{cases} \dfrac{1}{2^{n-1}} \displaystyle\sum_{j=1}^{2^n} g_{2j-1} - g_i \quad , \quad for \quad i \quad odd \\ \dfrac{1}{2^{n-1}} \displaystyle\sum_{j=1}^{2^n} g_{2j} - g_i \quad , \quad for \quad i \quad even \end{cases} \tag{2}$$

This fact allows a great reduction of the number of operation (and therefore of electronics components) and consequently a significant increase of computational speed. According to the proposed high-level scheme [2, 3], our circuit realization can be divided into two main parts:

Part I: *Base module.* It implements a 3-qubits system and it performs step-by-step calculation of output values. This part is divided in the following subparts:

a: *Entanglement*	**c**: *Interference*
b: *Pre-Interference*	**d**: *Modular interface*

Part II: *Control module.* It performs entropy evaluation in [4], vector storing for iterations and output visualization. This part also provides initial superposition of basis vectors |0> and |1>. Entanglement operator composed by eight driven switches (see Figure 2).

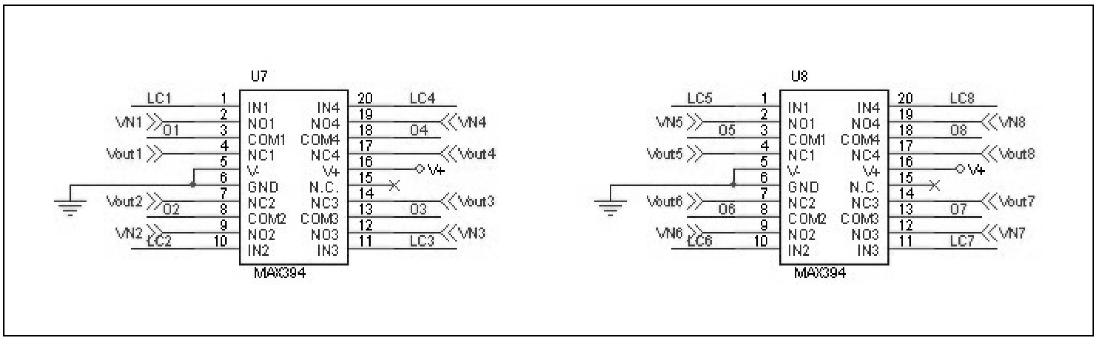


Figure 2: Entanglement circuit

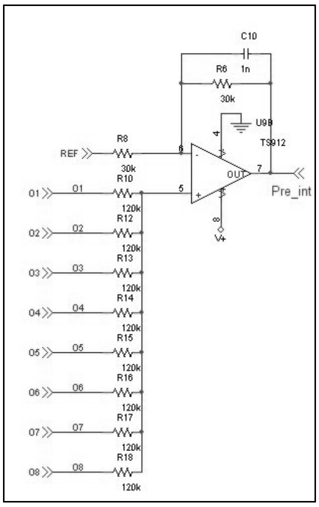


Figure 3: Pre-interference circuit

Referring to Figure 4, the switches (MAX394) present in the circuit are only the odd ones. They receive the elements of initial superposed vector in couples (Vout1 and VN1, Vout2 and VN2…) and perform the exchange according to the signal coming from the encoder.

The output signals (O1,…, O8) are the odd values of the entangled vector (even values are correspondent opposite value). These values are summed and scaled (the scaling factor is ¼ in the case of three qubits) by the OPAMP (see Figure 3), which constitutes the pre-interference step. The differences among this sum and each one of the elements are performed by the Interference block, whose structure is reported in Figure 4.

3. MODULAR SYSTEM

In order to realize modular system, some devices has been introduced. First and more important is operational amplifier (see Figure 5).

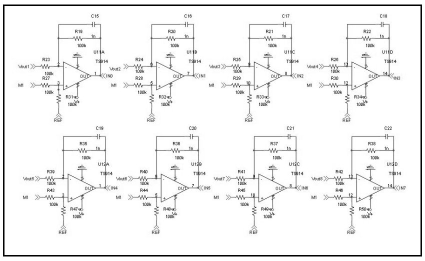


Figure 4: Interference circuit

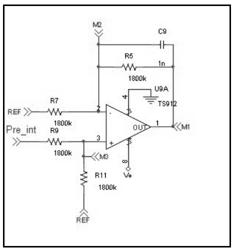
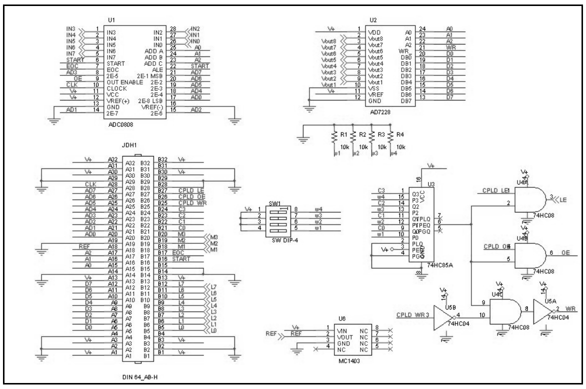


Figure 5: Modular Interface, Increasing qubit



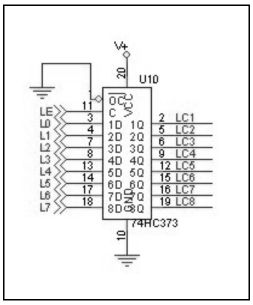


Figure 6: Modular Interface, Module selection and data conversion

Figure 7: Modular Interface, Latch

Labels M1, M2, M3 in Figure 5 are joined with corresponding others of different modules, performing parallel configuration. By this way output of two modules was summed and divided by 2, which is the result we wish to obtain in order to realize Grover algorithm [2,3] for $n + 1$ qubits. In fact each module performs three qubits QSA and by adding a second module we can realize four qubits QSA.Each module must be unequivocally identified through his address (a selector assign this address on each one), so the control module can send information to a specified module. The control module [3] send bit stream containing address and data to bus. On each module there is a device (74HC85A in Figure 6) that compare address sent by CPLD with the label of module and, if these are equal, allow it to process data.

Therefore these address indicate which modules must be in third state or which other must communicate through D/A and A/D converters with CPLD (see Figure 7). In order to provide the target value (element to be find) each base module has a latch able to store it (see Figure 7).

As previously reported the Control module performs entropy evaluation, vector storing for iterations and output visualization; however its main aim is to manage algorithm iterations.

Control module, that has been realized in digital way (CPLD programmable logic), is able to communicate with Base Modules through addressing system previously described (see in Figure 6 comparator '74HC85A') and D/A, A/D converters. Figure 8 shows the main board ($n = 3$) for

Grover's QSA that realized the modular structure. General methodology of quantum algorithm gate design and SW simulation results are described in [6-10].

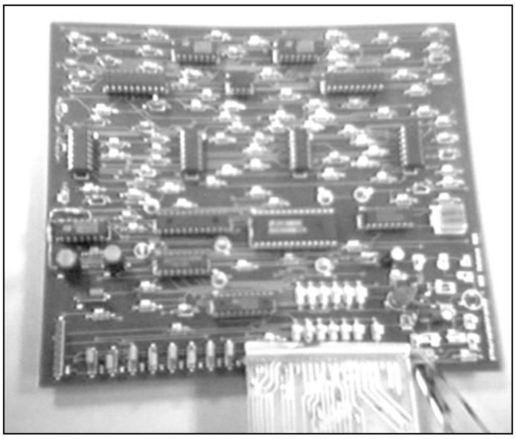


Figure 8: Main 3-qubits board with modular structure

4. SIMULATION RESULTS OF GROVER QSA

Figure 9 shows simulation results of Grover's QSA after application of each quantum operator. Initial state Figure 9a is transformed into superposition of all quantum states (Figure 9b). Entanglement operation marks the result by flipping corresponding probability amplitudes (Figure 9c) and interference extracts the requested state.

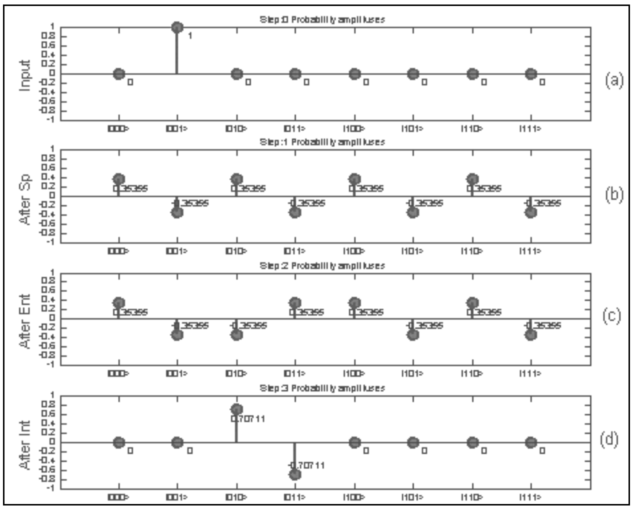


Figure 9: Simulation results of Grover's QSA

5. EXPERIMENTAL RESULTS OF GROVER QSA BOARD

Figure 10 demonstrates experimental results of the hardware simulation of the Grover's QSA. After few iterations the state of the quantum system indicates the necessary solution. As a termination condition we used minimum of Shannon information entropy criteria [4].

6. CONCLUSIONS

Hardware design of main quantum operators for quantum algorithm gates simulation on classical computer is developed. Hardware implementation for realization of information criteria as minimum Shannon entropy for quantum algorithm termination is demonstrated. Simulation and experimental results of Grover's QSA with classical hardware board are reported.

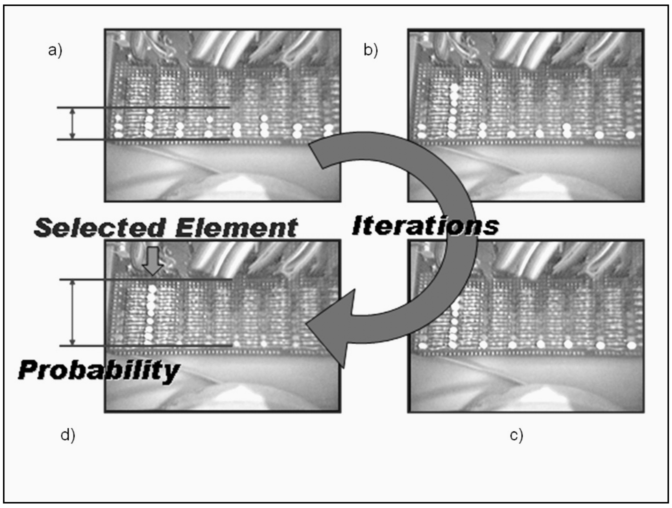


Figure 10: Experimental results of Grover's QSA board

6. REFERENCES

[1] J. Niwa, K. Matsumoto, H. Imai, "General-purpose parallel simulator for quantum computing", *Physical Review* A, 2002, Vol. 66, No 6.

[2] S.V. Ulyanov, F. Ghisi, S.A. Panfilov, I. Kurawaki and L.V. Litvintseva, *Simulation of quantum algorithms on classical computers*, Università degli Studi di Milano, Polo Didattico e di Ricerca di Crema Note del Polo Vol. 32, Crema, 1999.

[3] *PCT patent WO 01/67186 A1, 2000*, "Method and hardware architecture for controlling a process or for processing data based on quantum soft computing", (Inventors: Ulyanov S.V., Rizzotto G.G., Kurawaki I., Panfilov S.A., Amato P. and Porto D.)

[4] S.V. Ulyanov, S.A. Panfilov, I. Kurawaki, A.V. Yazenin, "Information analysis of quantum gates for simulation of quantum algorithms on classical computers," *Proc. QCM&C2000*, Kluwer Academic/Plenum Publ., 2001, pp.207-214.

[5] S. Panfilov, K. Takahashi, I. Ulyanov, "Fast quantum algorithms for quantum soft computing and simulation of robust intelligent robust control systems", *Proc. WAC 2004*, this issue.

[6] S.V. Ulyanov, K. Takahashi, G.G. Rizzotto, and I. Kurawaki, "Quantum soft computing: Quantum global optimization and quantum learning processes – Benchmarks of application in AI, informatics and intelligent control systems," *Proc. SCI' 2003*, July 27-30, Orlando, USA, 2003.

[7] S.V. Ulyanov, K. Takahashi, S.A. Panfilov, I.S. Ulyanov, P. Amato, D.M. Porto and G.G. Rizzotto, "Quantum soft computing via robust control: From Structure and HW implementation of quantum algorithm gates to Classical efficient simulation of Quantum Algorithm Gates and Control", *Proc. ICSCCW'2003*, (*Dedicated to Professor Lotfi Zadeh*), September 9-11, Antalya, Turkey, 2003.

[8] *PCT patent n. 01830383.4, 2001*, "Design of an analog circuit implementing the superposition operation in a quantum search algorithm", (Inventors: G.G Rizzotto, P. Amato and D. M. Porto).

[9] *PCT patent n. 02425447.6, 2002*, "A new method and circuit for implementing entanglement and interference operations in a quantum search algorithm", (Inventors: G.G Rizzotto, P. Amato and D.M. Porto).

[10] *US patent n. 6.678.018, 2003*, "System and method for control using quantum soft computing", (Inventor: S.V. Ulyanov).