

Proposed Architecture for Conventional Computer with Co-Quantum Processor

E. M. Ameen, H. A. Ali, A. Osman

Abstract—This paper presents a proposed architecture for constructing a conventional-quantum computing machines from the hardware point of view. The proposed architecture will help in improving the performance of the conventional computer by building a new processor; such processor combines both a classical processor in conjunction with a Co-quantum processor. The proposed architecture classifies the operations of the whole central processing unit into two categories; the first for Co-quantum processor to execute the complicated quantum computations. The second, for the classical processor which plays the role of performing conventional tasks such as network communication process, storing data tasks and the main task of controlling all system operations. Discussion and analysis of the proposed architecture, which is based on quantum parallelism in computations, shows that; the proposed architecture will improve classical computer performance. In addition; the proposed architecture is capable of performing any arbitrary quantum computation as well as conventional computations.

Index Terms— Architecture, conventional processor, central processing unit, quantum processor, synchronization.

1 INTRODUCTION

TODAY computer hardware technology is based on high performance technological trend in the semiconductor industry continue to tolerate by Moore's Law, in contrast continually shrinking size of circuitry packed onto silicon chips would eventually reach a point where individual elements would be no larger than a few atoms and eventually the conventional wisdom of transistor cramming for progress must be abandoned. So as we are getting close to the end of this roadmap, the idea to utilize quantum physics for solving difficult computational tasks has a lot of interest among both physicists and computer scientists. Once such a quantum processor is successfully built, it could carry out factoring large numbers, searching databases, and simulating physical systems exponentially faster than any computer used today. All the aspects of quantum computing are currently under intensive research and, as a result, the first quantum algorithms appeared before the actual fabrication of the first quantum gate. Using solid-state devices as the basic building blocks in the design of a quantum processor is particularly interesting, since these devices can be fabricated with a technology already prevalent in current computer chip fabrication [1-6].

This paper is organized as follow; in the next section we will discuss the concept of quantum information and quantum computations, in section 3 we will explore the important requirements for building a practical quantum computer. Finally, we will introduce a proposed architecture for using the computational power of quantum computing to improve the capabilities of the conventional computer.

2 QUANTUM INFORMATION AND QUANTUM COMPUTATION

Quantum computation and quantum information can be defined as the study of information processing tasks accomplished using quantum mechanical systems. Quantum computation and quantum information has many important aspects and principles which has any similarity with classical computing paradigm. Similar to classical complexity theory, quantum complexity theory is concerned with classifying the difficulty of various computational problems, grouping them into complexity classes according to the memory and time resources needed by a quantum computer to solve those problems. The relative power of quantum computers with respect to classical ones can be couched in the relationships between classical and quantum complexity classes. The Relationships between quantum and classical complexity classes are shown in Fig. 1.

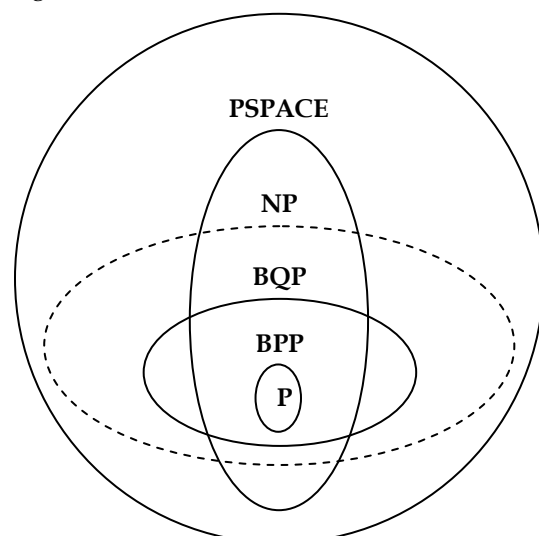


Figure 1: Relationships between quantum and classical complexity classes

- E. M. Ameen is with Department of Computers and Systems, Faculty of Engineering, Mansoura University, Mansoura, Egypt. E-mail: mahdy_maree@yahoo.com.
- H. A. Ali is with Department of Computers and Systems, Faculty of Engineering, Mansoura University, Mansoura, Egypt. E-mail: k-hesham77@yahoo.com.
- A. Osman is with Department of Computers and Systems, Faculty of Engineering, Mansoura University, Mansoura, Egypt. E-mail: m_elshobaky@yahoo.com.

Deutsch and Jozsa were the first to propose the use of quantum complexity classes to capture the difficulty of solving specific problems on quantum models of computation. Thus, in analogy with the classical classes Polynomial time (P), Zero error Probability in Polynomial time (ZPP), and Bounded error Probability in Polynomial time (BPP) we have the quantum classes QP, ZQP and BQP. The class BQP attracts most interest, being considered, even more so than QP, the class of all computational problems which can be solved efficiently on a quantum computer. Shor's algorithms, for instance, belong to BQP, while it is not known whether they are also in BPP or not. So, exactly where BQP fits with respect to P, BPP, NP and PSPACE is as yet unknown. What is known is that BPP_BQP and BQP_PSPACE. Consequently, from P_BPP_BQP_PSPACE we can see that BQP lies somewhere between P and PSPACE as shown in Fig. 1. Thus, we know for sure that BQP contains all of P and BPP, but whether it also contains some problems in PSPACE that are not in NP, for example, remains an open question [7-9].

3 REQUIREMENTS FOR QUANTUM ARCHITECTURES

To build a quantum processor of practical computational value, we must be capable of storing and processing a system of tens of millions qubits. While the work in physics and device development has made significant progress, such quantum processor main requirements are five criteria [10-13];

1. Quantum bits (qubit): finite Hilbert space quantum systems which can be composed in direct product form. A qubit is a unit vector in a two dimensional state space, for which a particular orthonormal basis, denoted by $\{|0\rangle, |1\rangle\}$ has been fixed. An arbitrary qubit $|\Psi\rangle$ can be written as a linear combination of the computational basis as follow:

$$|\Psi\rangle = \alpha |0\rangle + \beta |1\rangle \quad (1)$$

Where α and β are complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$. The qubit $|\Psi\rangle$ in (1) is in a superposition of $|0\rangle$ and $|1\rangle$, a state in which it is not possible to say that the qubit is definitely in the state $|0\rangle$, or definitely in the state $|1\rangle$. In general, an n-qubit quantum system may represent 2^n bit strings distinguished by 2^n complex-valued coefficients:

$$|\Psi\rangle = \sum_{i=0}^{2^n-1} c_i |x_i\rangle, \text{ such that } \sum_{i=0}^{2^n-1} |c_i|^2 = 1 \quad (2)$$

where each x_i represents the i^{th} bit string from 0 to 2^n-1 in (2) [10,14].

2. Long coherence: high isolation of the system from the environment, such that quantum superposition states have long lifetimes because they are the heart of any quantum algorithm.
3. Universal control: ability to subject the quantum system to a controllable sequence of unitary transforms which can approximate an arbitrary operator.
4. Initialization: ability to prepare the quantum systems in a fiducially input state that used to evaluate a function $f(x)$ for exponentially many different values of x in the time it takes a classi-

cal computer to evaluate the function for just one value. This is possible by loading the memory register with a superposition of all possible input values x in the initialization process [5], [16], [17].

5. Measurement: the capability to measure the final quantum state. The amount of information that can be stored in a qubit and, respectively, retrieved from a qubit can be extracted using any point on the Bloch sphere. When measuring a qubit, the only possible results of a measurement are the eigenvalues of the unitary operator describing this measurement [4-6, 9, 16]. According to this, when we measure a qubit $|\Psi\rangle = \alpha |0\rangle + \beta |1\rangle$ with respect to the standard basis for quantum computation $|0\rangle$ and $|1\rangle$ we get either the result 0 with probability $|\alpha|^2$ or the result 1 with probability $|\beta|^2$. Furthermore, measurement alters the state of a qubit, collapsing it from its superposition of $|0\rangle$ and $|1\rangle$ to the specific state consistent with the measurement result [18], [19].

In the previous section we give a general overview of the important requirements for building a practical quantum computer. Next we will give a description to the proposed architecture

4 PROPOSED ARCHITECTURE FOR CONVENTIONAL COMPUTER WITH CO-QUANTUM PROCESSOR

In this paper, the high parallelism of quantum computer forces us to design an architecture aimed at improving the performance of the conventional computer by building an enhanced processor; this processor combines both a classical processor in conjunction with a co-quantum processor. Our proposed architecture classify the operation of the whole system into two categories; the first for co-quantum processor to execute the complicated quantum computations. The second category for the classical processor which plays the role of the coordinator to all system operations also it performs the other network communication and data manipulation tasks. We believe that our proposed architecture has the following contributions;

- 1) Improve classical computer performance using the quantum parallelism in computations.
- 2) It will be a good start towards a non homogeneous networks of conventional and quantum computers in the near future.

This architecture is capable of performing any arbitrary quantum computation as well as conventional computations. A schematic of the proposed conventional-quantum architecture is shown in Fig. 2. The architecture consists of a classical microprocessor with a co-quantum processor connected together with a memory unit through a common bus managed by a bus controller. All units are controlled from classical front-end software that determines the program to be run and the instructions to be executed by each processor. The following subsection gives an overview of the system components and their functions.

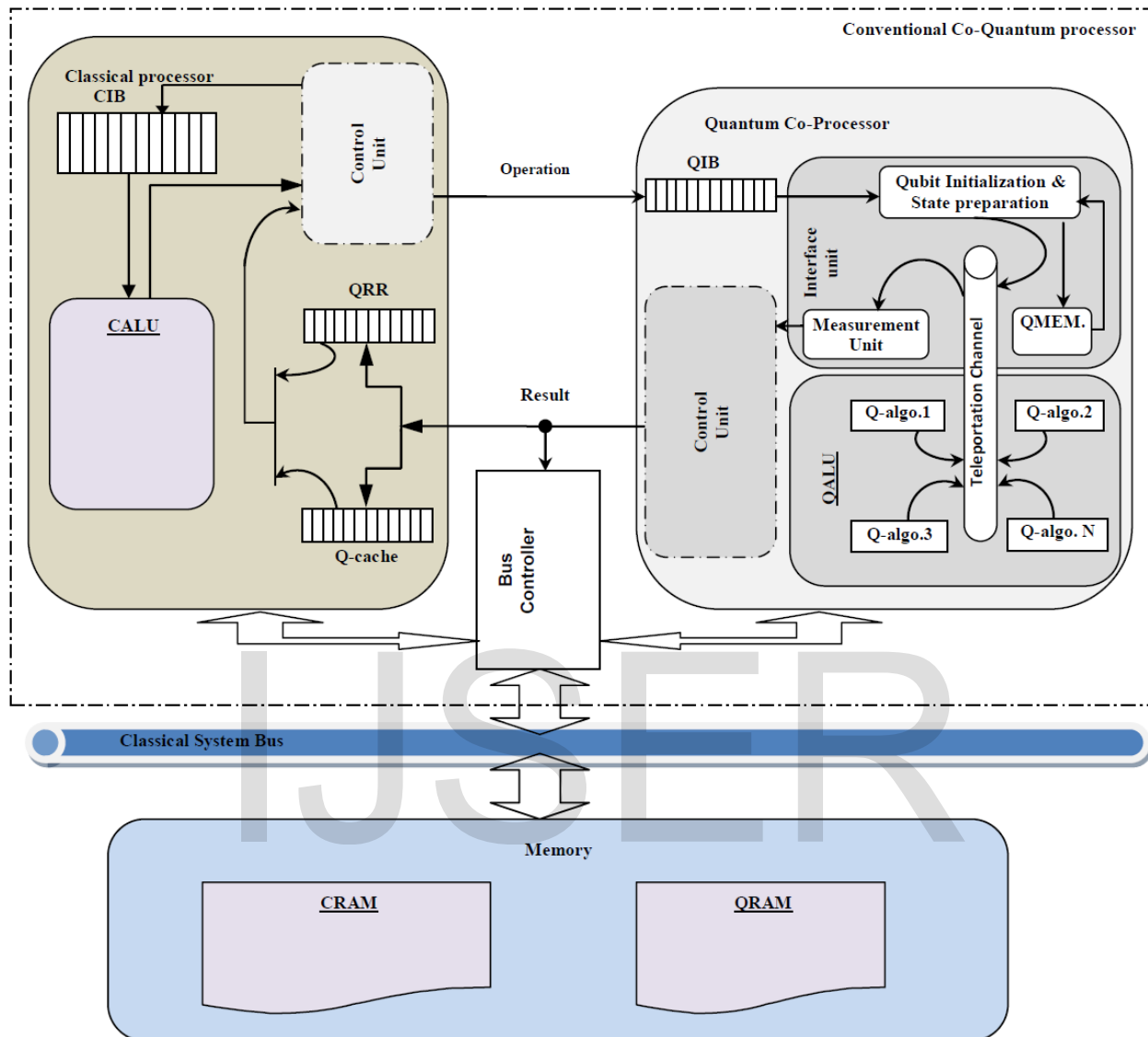


Figure 2: A schematic of the proposed conventional-quantum architecture

4.1 Conventional processor

As shown in the Fig. 2, it consists of classical arithmetic and logic unit (CALU) the main part of this processor [20], which is responsible for executing the instructions in the execution phase delivered by the control unit through conventional instruction buffer (CIB). Additionally, the control unit which is the brain of our architecture and its great mission of orchestrating all parts of this architecture starting from fetching the instruction from the memory, decoding it, preparing and fetching the operands, delivering it to the CIB to be queued until execution in the CALU and finally collects all results from the system. In this processor the quantum result registers (QRR) will be used to hold the obtained results from the quantum processor, but if the QRR is full or the quantum results are large it will be buffered in the quantum results cache (QR-cache).

4.2 Co-Quantum processor

As shown in the Fig. 2, it consists. First of all, these arrived quantum instructions to the co-quantum processor must be queued in the quantum instruction buffer (QIB) to be initialized in a quantum state. The qubit initialization and state preparation unit is the first physical quantum unit of the co-quantum processor that receives the decoded instruction from the QIB, and then prepares the operands in a quantum initial state to be processed by a specified processing unit according to the decoded quantum unitary operation. When new instruction received the qubit initialization and state preparation unit searches for operands in the quantum memory (QMEM), if cache hit occurs then it will deliver them immediately to the specific processing unit, but if cache miss occurs, it will prepare the qubits and push them to the specified processing unit. The co-quantum processor has quantum arithmetic and logic unit (QALU) which is composed of number of

processing units, each of them composed of a number of quantum circuits. The overall function of the sequence of operations in an entire n-qubit quantum circuit divided into K time steps and the collective action of the sequence is the product of all individual operators for each time step. In each time step, a single gate in a quantum circuit with one or more input qubits in the initial state $|\Psi\rangle$ transforms the state to a different state $|\Psi'\rangle$ by changing all probability amplitudes that describe the state vector $[c_0, c_1, \dots, c_{n-1}]^T$. Finally, the result from any quantum processing unit will be obtained by measurement [13, 18, 21-23]. Measurement collapses all obtained superposition states to only one output state that can be treated classically once again in the classical processor to finish the PREPARE-EVOLVE-MEASURE instruction execution cycle in the co-quantum processor. All co-quantum processor units are connected together through the teleportation channel that constructs the transportation means inside the co-quantum processor.

4.3 Bus controller and memory

Memory space is partitioned into two spaces, one for the conventional program requirements from storing the compiled program code till results be obtained. The second space for intermediate quantum results obtained through the execution cycle of the program. In fact the bus controller part of the system is an extension to the control unit functions; it controls the access to the memory spaces in the read and write operations.

After the previous outlines of the system components and their functions the following explanation will explore our proposed scenario of interactions between system components.

5 INTERACTIONS BETWEEN CONVENTIONAL-QUANTUM ARCHITECTURE COMPONENTS

A computer system cannot be designed without an understanding of its target workload and expected performance. Above, we suggested the goals of the architecture and now we will get a brief description of the work flow of this architecture. The proposed architecture consists mainly of two processors, as previously stated the co-quantum processor composed of a number of quantum processing units that communicates through teleportation but the only way of communication between the classical and quantum hardware is through measurement unit. The communication and synchronization of the two processors is handled by the control unit according to a pre-compiled program with the help of specific communication and synchronization protocol. The following subsections will illustrate the suggested methodology used in the proposed communication and synchronization protocol.

5.1 Execution cycle and Communication protocol scenario

In the proposed architecture quantum instructions are architected as an extension to the existing instructions set through the use of a special opcode which indicates that the rest of the instruction should be treated as quantum instruction. Also to guaranteed high performance, the proposed protocol must preserve the pipelining execution technique for the instructions execution, as a result a special instruction pre-fetch buffer will be used in both processors; one for the classical proces-

sor and called CIB and the other is called QIB as previously stated in section 4.1. This allows instruction to be fetched simultaneously while another instruction is being executed.

Achieving high degree of concurrent instruction execution could be facilitated by the compiler, which should have the freedom to fully classify instructions and computation in order to optimize the usage of quantum data and make much parallelism between the units as possible. In this system the proposed communication protocol characteristics depends on the following factors;

- 1) The way instruction is passed to the co-quantum processor.
- 2) The way instruction is queued into the existing instructions queues in both processors.
- 3) The amount of out of sequence and concurrent execution between the two processors.

As stated before the control unit will take the total control through the instruction execution cycle in both processors. On the instruction fetch, the classical processor receives the incoming instruction, they are encoded in a way which make easy to distinguish classical instruction from quantum one. Therefore, incoming instruction is pre-decoded and buffered into the CIB which used to buffer the two types of instruction, but if the pre-decoded instruction is quantum one it will be buffered in both buffers CIB and QIB. Fig. 3 illustrates the instruction execution cycle where, QALU executes quantum instructions as they received one by one from the QIB then, send results to the QRR or to the QR-cache in the classical processor. If both QRR and QR-cache are full, then data will be written back into the quantum memory space for later use. Finally, to preserve synchronization under branching, accessing memory and interrupt call all current execution information should be saved to help pointing to the exact point in the program where the exception occurred. The next subsection will give explanation of how the proposed communication protocol will preserve high synchronization between the two processors as well as high data manipulation rates.

5.2 Synchronization and out of sequence execution

Synchronization here means that the two processors must be at the same point in the program flow. Ideally both processors will be running concurrently executing their instructions in parallel, additionally they will execute their instructions as they arrive from the instruction queues therefore allowing out of sequence execution. Under these circumstances the maximal throughput would be achieved, and the overall system performance is determined by the individual performance of the processors as well as the performance of instructions mix. But branches in program and data dependencies between classical and quantum processes do not make these circumstances possible all the time. So, to assure consistency of the operations, the two processors must be synchronized under the following conditions;

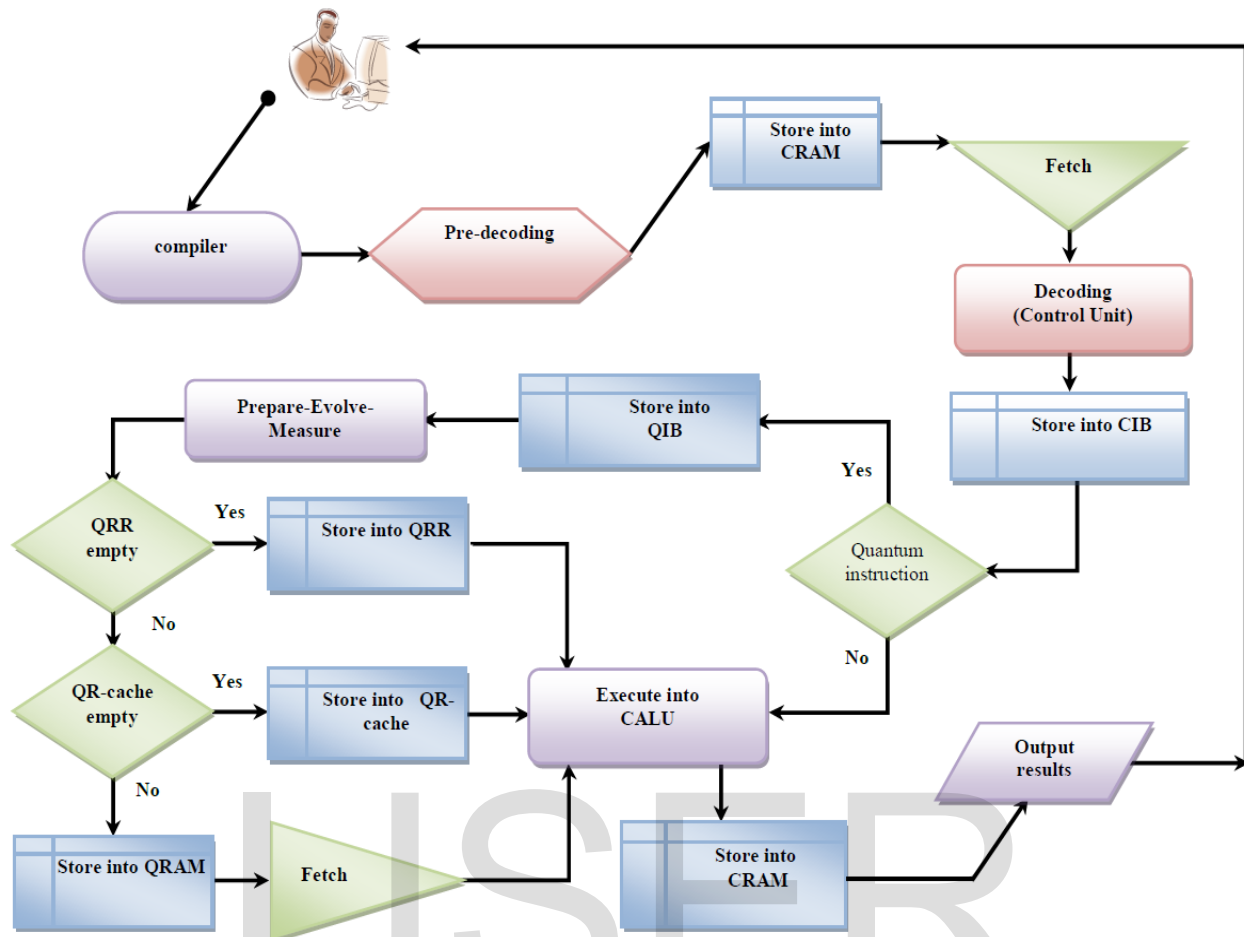


Figure 3: The instruction execution cycle

- 1) Branches in the program (conditional and unconditional).
- 2) Exchange of data (either through the memory or exchange of register contents).
- 3) Interrupts and procedure calls.

Branches in the program causes the processor to suddenly changing the order of instructions execution, so the QIB contents must be changed to hold the new instructions set for the new address of the branch instruction. When the branch instruction is decoded the classical processor must wait until QIB complete signal arrived from quantum processor. At this point both processors are synchronized and branch can take place.

Any memory access is a reason for synchronization. If data is to be written to memory it must be assured that the data at the location to be overwritten is not longer needed by the other processor (classical or quantum). Similarly if there is an access to memory location and data in this location is a result of an instruction currently executed by the other processor, then this instruction must be completed. Also if the data is already available, the other processor should not be allowed to proceed beyond this point if there is possibility that execution of the next instruction might alter the data. When interrupts and

procedure calls are presents the processors must achieve synchronization before the status of the calling process is saved. In all cases, when an exception is detected the status of the processors must be saved in the way that makes it possible to restart the process to this saved point in the program.

6 CONCLUSION

In this paper we proposed an architecture for constructing a conventional-quantum computing machines from the hardware point of view. Discussion and analysis emphasis that our proposed architecture will improve the performance of the conventional computer by building a new processor; this processor combines both a classical processor in conjunction with a Co-quantum processor. Complete information of this hybrid machine will be presented in our next work; we will give the system structure in more details.

REFERENCES

- [1] Douglas J. Matzke, "Is Quantum Computing A Solution For Semiconductor Scaling", Dallas IEEE Computer Society Meeting, Jan 19, 2007.
- [2] Technical Report 2005, School of Computing, Queen's University, Kingston, Ontario, " On The Importance Of Parallelism For Quantum Computation And The Concept Of A Universal Computer", May 2005.
- [3] D. Kielpinski, Chris Monroe, and David J. Wineland, "Architecture for a large-scale ion-trap quantum computer", Nature, 417:709, June 13, 2002.

- [4] Marius Nagy and Selim G. Akl, "Quantum Computation And Quantum Information", Technical Report No. 496 School of Computing Queen's University Kingston, Canada, 2005.
- [5] Sofge D.A., "Survey of Quantum Programming Languages: History, Methods, and Tools", Quantum, Nano and Micro Technologies, IEEE Second International Conference on 2008.
- [6] Hassan S., Humaira and Asghar M., "Limitation of Silicon Based Computation and Future Prospects", Communication Software and Networks, ICCSN '10, Second International Conference on 2010.
- [7] Mohamed T., Badawy W. and Jullien G., "On using FPGAS to accelerate the emulation of quantum computing", Electrical and Computer Engineering, CCECE '09. Canadian Conference on 2009.
- [8] M. Oskin, F. Chong, and I. Chuang, "A Practical Architecture for Reliable Quantum Computers," IEEE Comput., Vol. 35, No. 1, p. 79-87, January 2002.
- [9] M. Nielsen, "Optical quantum computation using cluster states," Phys. Rev. Lett, Vol. 93 (040503), 2004
- [10] D. P. DiVincenzo, "The physical implementation of quantum computation," Fortschr. Phys., Vol. 48, pp. 771-783, 2000.
- [11] Lieven Vandersypen and Isaac Chuangy, "NMR Quantum Computing-Lessons For The Future", submitted to Rinton on March 28, 2001.
- [12] A. Ekert et al., "Practical quantum cryptography based on two-photon interferometry", Phys. Rev. Lett., Vol. 69, pp. 1293-1295, 1992.
- [13] Bokor J., Nadai L. and Rudas I., "Controllability of Quantum Bits from the von Neumann Architecture to Quantum Computing", Computational Intelligence and Intelligent Informatics, ISCIII '07. International Symposium on 2007.
- [14] M. A. Nielsen and I. L. Chuang, "Quantum Computation and Quantum Information", Cambridge Univ., Press, New York, NY, 2000.
- [15] L. M. Vandersypen et al., "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance," Nature, Vol. 414, p. 883, 2001.
- [16] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, "Quantum Computation by Adiabatic Evolution," 2000, quant-ph/0001106.
- [17] H. J. Briegel and R. Raussendorf, "Persistent entanglement in arrays of interacting particles," Phys. Rev. Lett, Vol. 86, pp. 910-913, 2001.
- [18] T. S. Metodi, D. D. Thaker, A. W. Cross, F. T. Chong, and I. L. Chuang, "A quantum logic array microarchitecture: scalable quantum data movement and computation," In Proc. 38th International Symposium on Microarchitecture, MICRO-38, 2005.
- [19] J. I. Cirac and P. Zoller, "Quantum computations with cold trapped ions," Phys. Rev. Lett, Vol. 74, pp. 4091-4094, 1995.
- [20] Xunying Zhang and Xubang Shen, "A Power-Efficient Floating-point Coprocessor Design", International Conference on Computer Science and Software Engineering, IEEE, 2008.
- [21] M. Oskin, F. T. Chong, J. Kubitowicz, and I. L. Chuang, "Building quantum wires: the long and the short of it," in ISCA-30, San Diego, CA, 2003.
- [22] Sofge D. A., "Toward a Framework for Quantum Evolutionary Computation", IEEE Conference Cybernetics and Intelligent Systems on 2006.
- [23] Boncalo O. and Amaricai A., "Reliability Analysis of Qubit Data Movement for Distributed Quantum Computation", Digital System Design, Architectures, Methods and Tools, 12th Euromicro Conference on 2009.