Simulation-Based Assessment of Quantum Circuit Reliability

Oana Boncalo
Computer Science Department
Politehnica University of Timisoara

Advisor: Prof. Mircea Vlăduțiu

PhD Project

September 2007

Abstract

This report aims at presenting the research program and to review the literature describing quantum computing, quantum fault tolerance issues and classical fault injection. Thus, previous work is presented together with future research directions. The proposed PhD research program aims at developing a methodology for simulation based fault injection for quantum circuits. Through simulation, I obtain a statistically significant sample in order to provide a quantitative means to assess quantum circuit reliability. A desired feature of the fault injection methodology is scalability. Nevertheless, quantum circuit simulation requires simulation resources that grow exponentially with the number of simulated qubits. Although, simulation techniques that improve the simulation cost and/or time were developed, still for complex and large structures it is not feasible to apply fault injection. In order to overcome this obstacle and obtain scalability, I propose a logic partitioning. The complex and large quantum circuit is decomposed into less complex parts. Fault injection is used to evaluate quantum reliability for these parts. The reliability for the larger circuit is assessed by means of a reliability graph by taking into account the reliability of the smaller parts.

Contents

List of Figures

Chapter 1

Introduction

Computing machines evolved tremendously during the last decades. There is an increasing need for new reliable devices with less power consumption which are smaller in size. As nowadays the dimension of semiconductors is required to become smaller and smaller, it will eventually reach the nanometer scale. The laws that must be obeyed on the nanometer scale are those of quantum physics [3]. Quantum Computation (QC) promises tremendous computational power for efficiently solving some of the most difficult problems in computational science, such as integer factorization, discrete logarithms, and quantum simulation an modeling that are intractable on present and even future conventional computational devices [1].

If classical computers are highly reliable, the quantum elements are more fragile. One of the greatest challenges for building quantum devices is decoherence, mainly the distortion of the quantum state due to the impossibility of perfectly isolating the quantum system from its environment [4][5]. In addition, the quantum elementary operations (called gates) suffer from inaccuracies [6][7][8]. The errors that accumulate ruin the quantum computation; hence, a way to overcome the effect of quantum noise is necessary. Optimistic signs for overcoming this drawback were given by the discovery of the quantum error detecting and correcting codes such as [9][10][11][12]. However, there are still some obstacles to overcome especially from the technological perspective [1][2]. It is necessary to develop significantly more complex quantum-information processing capabilities before quantum computer science issues can begin to be experimentally studied. The desired 2007 and 2012 high-level goals for QC as they are stated in the roadmap for QC are [1]:

- By the year 2007, to:

- to encode a qubit into the state of a logical qubit made of several physical qubits
- perform repetitive error correction on the encoded qubit
- transfer the state of the logical qubit into the state of another set of physical qubits with high fidelity.

- By the year 2012, to
  - implement a concatenated error correcting code.

The ways in which the nanotechnologies are suppose to evolve according to the International Technology Roadmap on Semiconductors (ITRS), the Emerging devices document is depicted in Fig. 1.1 [2].

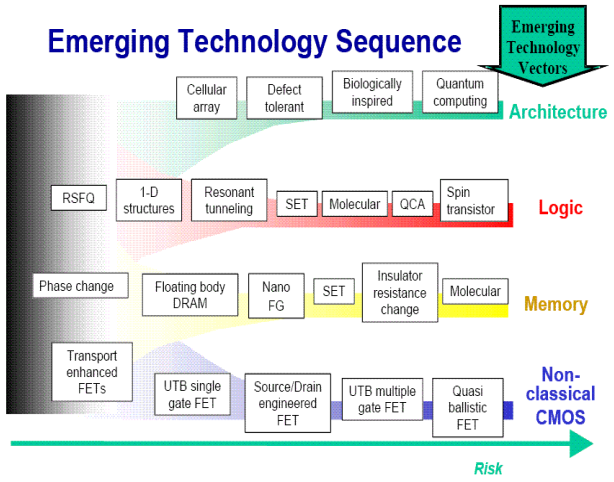

Figure 1.1: Emerging technology sequence [2]

Meeting this goals require both experimental and theoretical advances. Because device simulation and simulate fault injection proved to be a helpful tool to understand, explore and evaluate new hardware designs for classical circuits [15][16][17], I intend to take advantage of the rich classical theory of fault injection methodologies and to extend and adapt it for QC. In the proposed thesis I intend on focusing my attention on the modeling of quantum faults and error models in order to provide realistically simulation results. Thus, the core of my work will deal with simulated fault injection for quantum circuits - a tool to quantitatively evaluate quantum circuits fault tolerance.

This report is organized as follows: Section 2 titled “Previous work” is concerned with establishing the scientific domain of the proposed thesis.

Furthermore, the information presented in the section is divided in two scientific directions: on the one hand classical FI with existing methodologies is depicted, and on the other hand a review on the existing FI for quantum circuits and also QC related information is presented.

Section 3 – "Thesis overview" is concerned with describing the thesis goals and proposed outline. The timelines and the activities involved are also presented, together with the potential contribution of the proposed thesis.

The last Section 4 – "Conclusions" summarizes the repost and outlines the potential research directions and contributions.

Chapter 2

Previous work

2.1 Scientific domain

During the 1970s the field of digital circuit testing suffered a tremendous revolution. Before that, testing was strongly related to manufacturing, almost completely isolated from design – "we design it, we build and test it". After that time, more and more companies left the entire responsibility for testing in the hands of designers. The test technology that has evolved along with the digital system technology involves three distinct but interrelated areas: test hardware, test software and test theory. Test hardware (test systems) gives excellence performance but, they are often too expensive and they tend to be self-obsolete. Test software refers to two groups of applications. On the one hand, there is the software that runs the test systems and, on the other hand there are the application programs that deal with the automation of portions of the design process. Automatic test generation for combinational networks, fault simulation, and checking of design rules are among the programs that support test [14].

In the design phase, computed-aided design (CAD) environments are used to evaluate the design via simulation, included simulated fault injection. The simulation based fault injection is used to test the effectiveness of fault-tolerant mechanisms and evaluates the dependability, providing feedback to system designers. Simulation, however, needs accurate input parameters and also, the validation of the results is needed [13]. Central to this is the model of the faults and also the fault patterns. Faults can occur singly or there can also be multiple faults. A large portion of literature and research in this area deals with singly occurring faults. While this simplifies the analysis, the occurrence of multiple faults cannot be excluded [14]. In the design phase, simulated fault injection can be applied at various levels of abstraction for the classical circuits. However, as the circuits became larger and more complex, the simulation and analysis of various

parameters became more difficult. Thus, in order to avoid the growth of resources needed for the simulation, methods for dividing the circuit into logically independent parts, which can be processed separately, were designed. These methods are collectively called partitioning [14].

The rich techniques and methodologies developed for the classical circuits [13][15][17] inspire the development of similar tools and environments for the emerging technologies. Among these, QC faces the challenge of improving the quantum circuit reliability by means of fault tolerance mechanisms (such as error detection and correction codes [8][9]). However, these proposed mechanisms need to be evaluated. Because simulation of assessing fault tolerance through simulated based fault injection is a much cheaper mean than the hardware based solution [16] for classical circuits, I expect the same outcome for the use of such techniques for quantum circuits (especially because the hardware is less accessible than for classical systems).

For the proposed research, there are some domains that are connected. In order to extend the benefits of simulation based fault injection techniques for reliability parameters computation for quantum circuits, a bridge between quantum circuit simulation, quantum fault modeling and also the existing classical fault injection methodologies needs to be establishes. The merge between the above mentioned domains is needed in order to develop a viable methodology for quantum circuits fault injection.

2.2 Simulated FI for classical circuits

Fault injection techniques can be classified in three main categories: physical (or hardware implemented fault injection), software implemented and simulation-based fault injection [18]. Simulation-based fault injection techniques based on hardware description languages (HDL, especially VHDL), offer important advantages with regard to other fault injection techniques. First, they can be applied early in the design process, thus, reducing time-to- market. Furthermore, early diagnosis of design errors reduces costs [16][18]. Second, this types of techniques present high controllability and reachability Two main trends characterize recent work on fault injection (FI): first, to apply fault injection as early as possible in the design process of fault-tolerant systems, i.e., into the simulated design models of the fault tolerant systems; and second, when dealing

with the implementation of the target fault tolerant system, favor software-implemented fault injection [15][16].

2.2.1 Methodologies and tools

A very attractive group of fault injection techniques are those based on VHDL as a modeling language. These techniques are widely applied because they are offered the advantage of a standard description language [19]. A classification of these techniques is shown by Figure 2.1. As illustrated two categories of fault injection techniques are identified: one that demands the modification of the VHDL model, and a second technique which makes use of the simulator commands.

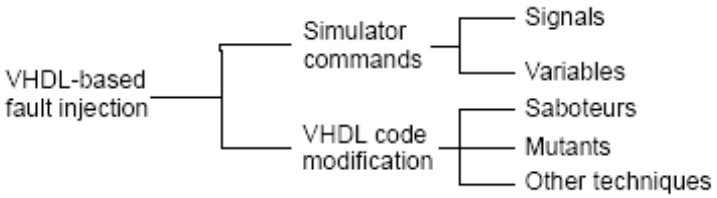

Figure 2.1: VHDL-based techniques for simulation-based fault injection [18]

The techniques based on simulator commands deal with signal and/or variable manipulation. When using signal manipulation, the correct value of the signals in the VHDL model is altered by disconnecting the signal from its driver(s) and forcing it to a new value. Variable manipulation, on the other hand, is useful for the behavioral VHDL models, and implies the altering of the variables present in the VHDL code [15].

For the second category there are as shown by Figure 2.1 two representatives: saboteurs and mutants [16]. A *saboteur* is a VHDL component that alters the value or timing characteristics of one or several signals when activated. There are two possible saboteur architectures presented in Figure 2.1. These saboteurs break the path between a drive and its corresponding receiver [15]. A *mutant* is a component description that replaces another component description. When inactive, it behaves as the component description it replaces, and when activated it presents the components behavior in the presence of faults. Mutants take advantage of the VHDL configuration mechanism in order to make the replacement of the correct component with its mutations [18].

Saboteurs fall into 2 categories serial and parallel, and can be simple or complex depending on the fault pattern that is being modeled (see Figure 2.2) [16][20]. A serial saboteur, on the one hand, breaks the signal path between a

driver output and its corresponding receiver input, while a parallel saboteur is usually added as an additional driver for a resolved signal for the receiver [16].

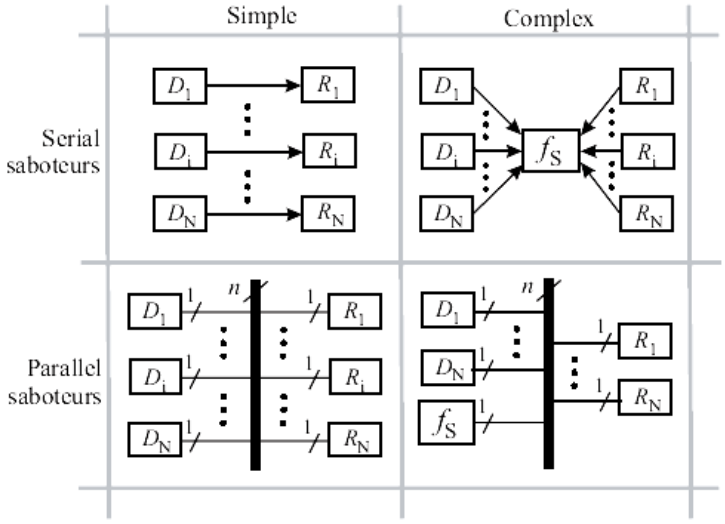

Figure 2.2: **Serial/parallel insertion on simple/complex saboteurs. $D_1 \ldots D_N$ are the drivers, $R_1 \ldots R_N$ the signal receivers, while f_S stands for the function required by complex saboteur computation [20]**

There are several ways for generating mutations (obtaining mutant descriptions of a correct component) [16][18]:

- adding saboteur(s) to an existing structural or behavioral VHDL component description,
- recursive mutations of a component by replacing subcomponents (e.g. replacing an AND gate by a NAND gate),
- by modifying statements in behavioral component description (this can support automatic generation of mutants).

To sum up, a comparison of the fault injection techniques presented above is required, based on the observations and simulation results from the work of [15][16][17][18]. Although at first glance, the techniques based on simulator

commands seam to be favored because they don't require code modification, these techniques present the disadvantage of being highly dependent on the VHDL simulator capabilities and the functionalities of their commands. Mutants offer the highest fault modeling capability and they use the full strength of the VHDL language by making use of the configuration mechanism. Saboteurs are generally used for less complex faults, but, there can also be complex saboteurs by incorporating finite state machines in them.
Based on the presented techniques a number of tools for simulated fault injection for classical circuits were developed such as the MEFISTO tool [15][16], the VFIT tool [18]. An overview of the VFIT tool is presented in Figure 2.3.
The most common features of these tools are:

- automation at different degrees of the fault injection process. This is accomplished by means of dedicated software modules for setting up and running the simulation (e.g.: for the setup part there are some automated code mutation – mutant generation; also, there are scripts designed to run the simulation fault injection campaigns).
- extraction of various error related information and also its processing;
- the injection experiment consists of 3 phases.

For both MEFISTO [16] and VFIT [18] there are 3 phases for simulation-based fault injection:

- A *setup phase* where the simulation parameters are tuned. These parameters may refer to the fault model and fault occurrence pattern, the number of simulations to be performed, the data which is fed to the simulated circuits, data which is to be collected during the simulation campaigns is decided.
- The *simulation phase* consists of the actual simulation of the VHDL circuit under test, and also, during this phase information is collected to be later analyzed during the last phase.
- The *analysis phase(results processing phase)* the faulty trace is compared to that of a golden run (a simulation result obtained from a correct functioning system), extracting different dependability and simulation related parameters.

With the inspiration drawn from the classical hardware HDL-based fault injection techniques, I aim at extending this knowledge to the quantum circuits. However, the classical fault injection methodologies presented above cannot be mapped without intervention for quantum computation due to its specific features.

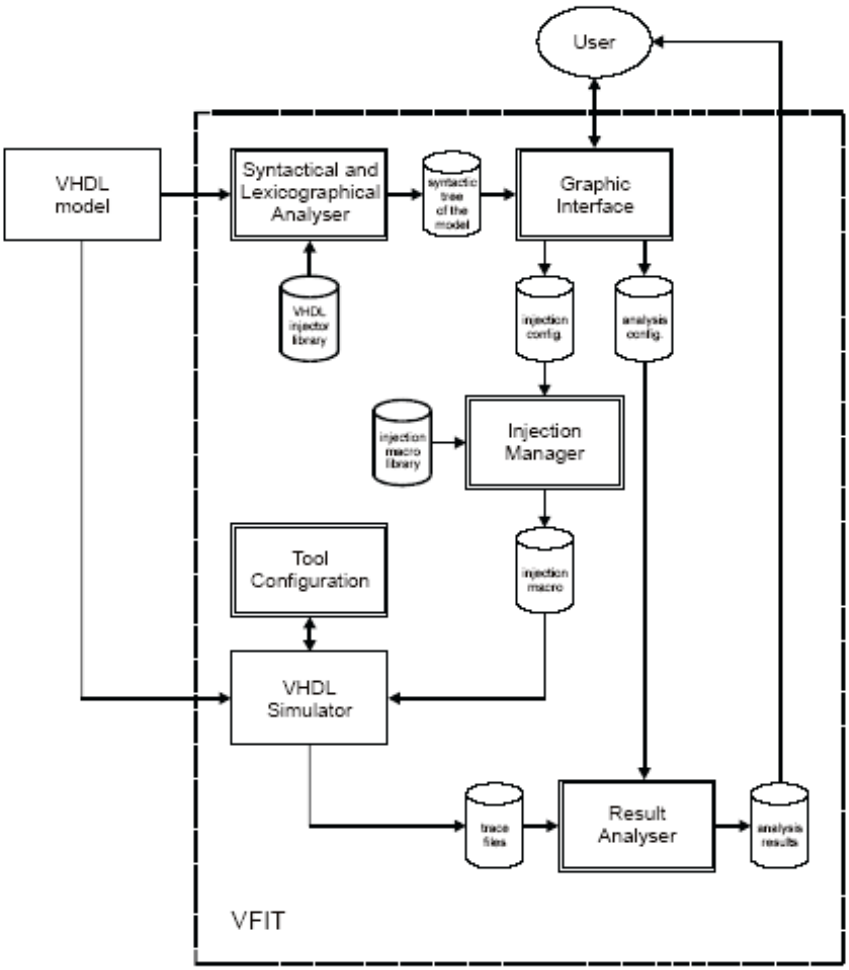

Figure 2.3: **VFIT block diagram [18]**

2.2.2 Logic partitioning

A fault-tolerant system must function correctly even after some of its elements failed. In order to estimate the reliability of a system as a whole we make use of reliability models. All reliability models start with assumptions regarding the rate at which various system elements fail. Fault tolerant systems require complex reliability models in order to predict overall system reliability. These models fall into one of the two cases: combinational or Markov [13]. Combinational models attempt to categorize the set of operational states in a way that the probabilities of each of these states can be determined by combinational means [13]. Markov models, on the other hand, concentrate on transitions and on the rate at which these take place. This information is used to determine the probabilities that the system is each of these states at some given time [13]. In general a structure can be represented by a Markov model if it's possible to characterize it in terms of states.

When dealing with large complex systems, testing techniques require large computational resources. A group of methods dedicated for dividing the circuit into logical independent parts, each of which can be processed separately was designed. These methods are collectively called partitioning (or divide and conquer) [14]. There are two possible reasons for partitioning:

- The network/system is to big and complex for the tools,
- The network/system has too many inputs for exhaustive testing.

2.3 Quantum circuits' reliability

2.3.1 Quantum circuits: background

In modern computers (referred to as *classical* to distinguish them from their quantum counterparts) binary information is stored in a bit. For the quantum domain, binary information is stored in a quantum bit called qubit. Thus, the qubit can be regarded as an extension of the classical notion of bit, where the besides the classical states denoted by denoted by $\{|0\rangle, |1\rangle\}$ we an also have a superposition of these two states in the form of a unitary vector $a|0\rangle + b|1\rangle$ (see bra/ket notation invented by Dirac [26]), where *a*, *b* are complex numbers called quantum amplitudes with $|a|^2+|b|^2=1$. If such a superposition is measured with respect to the

basis $\{|0\rangle, |1\rangle\}$ (any orthogonal unit vectors can be considered for the basis, as long as the notations are consistent) then, the probability that the measured value is $|1\rangle$ is $|b|^2$, and the probability that the measured value is $|0\rangle$ is $|a|^2$. Even if the quantum bit can be put in an infinitely many superpositions of states, it is only possible to extract one classical bit. The reason is that the information can only be obtained by measurement, which is an irreversible operation. When a measured is done, it changes the state to one of the basis states [5].

The qubits can be organized in linear structures called *quantum registers*, encoding a superposition of all possible states of the corresponding classical registers. For a *n*-qubit quantum register, its corresponding state is a normalized vector in the $\mathcal{H}^2$ Hilbert space, $|\Psi\rangle = \sum_{i=0}^{2^n-1} a_i |i\rangle$, where $\sum_{i=0}^{2^{n-1}} |a_i|^2 = 1$ and $|i\rangle$ is one of the superposed states of the register.

e.g.:

In the case where the individual qubit states are known

$|\Psi_A\rangle = a_0|0\rangle + a_1|1\rangle$ and $|\Psi_B\rangle = a_2|0\rangle + a_3|1\rangle$,

the tensor product will give the overall state:

$$|\Psi_A\rangle \otimes |\Psi_B\rangle = a_0a_2|00\rangle + a_0a_3|01\rangle + a_1a_2|10\rangle + a_1a_3|11\rangle .$$

Also, in matrix representation we have:

$$\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \otimes \begin{bmatrix} a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} a_0a_2 & a_0a_3 & a_1a_2 & a_1a_3 \end{bmatrix} .$$

A quantum state affected by entanglement cannot be efficiently represented as a tensor product of its parts [21]. Two examples depicting a non entangled state and a quantum state affected by entanglement are shown. For the non-entangled state the representation of the state as a tensor product of its qubits is showed. Furthermore, the second example illustrates a quantum state affected by entanglement and also the impossibility in a straightforward representation of the entangled state as a tensor product of its qubits.

e.g.1:

$|\Psi_1\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |01\rangle)$, as tensor product: $|\Psi_1\rangle = |0\rangle \otimes \left[\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right]$;

e.g.2:

$|\Psi_2\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, for $|\Psi_2\rangle$, there aren't 2 vectors that verify the condition:

$$\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \otimes \begin{bmatrix} a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} a_0a_2 & a_0a_3 & a_1a_2 & a_1a_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}.$$

In matrix representation, a quantum state corresponding to *N* qubits affected by entanglement requires 2^N quantum amplitudes (i.e. matrix elements) to be stored, whereas a non-entangled quantum state (i.e. which can be represented by a tensor product) requires *N* (2×1)-size matrixes (therefore, 2*N* matrix elements). This means that when entanglement is present the resources required for simulation grow exponentially with the number of qubits, unless an optimized methodology for storing the qubits is used [25].

The circuit model of quantum computation consists of the quantum *gate array*, a formalism introduced by Deutsch [28]. It is an acyclic combinational logic circuit which is in fact made of quantum gates interconnected without fan-out or feedback by quantum wires [27]. A schematic representation of the quantum circuit is shown in Figure 2.4. Each rectangle represents a level of the quantum network that is being computed at a certain moment in time. Also, each level of gates has the number of inputs equal with the number of outputting wires.

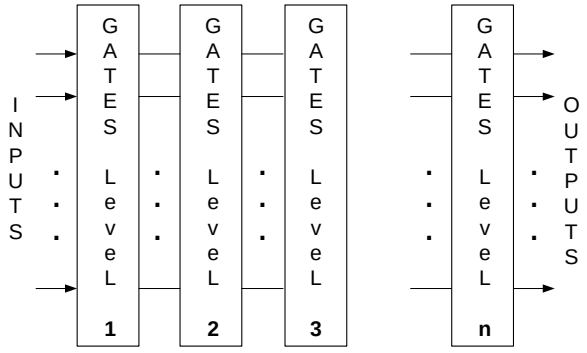

Figure 2.4: A schematic representation of a quantum network

Analogous to the way a classic computer is made of an electronic circuit containing wires and logic gates, a quantum computer is built from a quantum circuit containing quantum gates to manipulate the quantum information. Next, some simple quantum gates are presented. A quantum gate is described by a quantum transformation – most commonly in matrix form – with a single constraint: *unitarity* [21]. A matrix U is unitary (describes a unitary transformation) if $UU^* = I$. Unitary transformations can be regarded as rotations of a complex vector space [5]. A very important consequence is that quantum transformations are reversible. Reversible computation is especially attractive because of its relation to the energy of computation and information.

Some important gates that act upon one qubit (single qubit) gates are: the Z gate, the X gate, the Y gate, and the H gate (also known as Hadamard gate). These transformations are described below in both BraKet notation and matrix form (for more details see [5][21]). Also, the symbols most commonly used to represent them are depicted.

- The X gate (negation gate) is also known as bit-flip transformation:

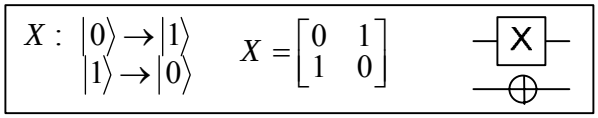

Figure 2.5: The X gate: BraKet notation, matrix representation, symbol

- The Y gate performs both phase shift and bit-flip $Y=ZX$:

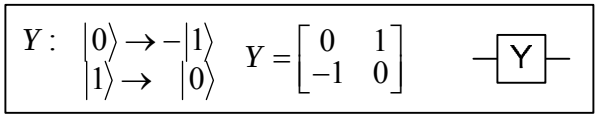

Figure 2.6: The Y gate: BraKet notation, matrix representation, symbol

- The Z gate which performs phase shift:

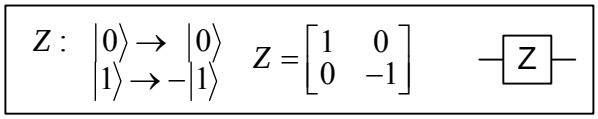

Figure 2.7: The Z gate: BraKet notation, matrix representation, symbol

- The H gate (Hadamard gate) is one of the most useful quantum gates.

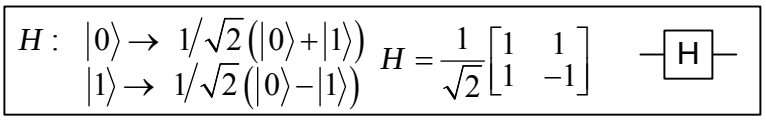

Figure 2.8: The Hadamard gate: BraKet notation, matrix representation, symbol

The Hadamard transformation is used in many applications. It is very important because it transforms $|0\rangle$ (or $|1\rangle$), a classical state into a superposition of states $1/\sqrt{2}(|0\rangle + |1\rangle)$ (or $|1\rangle \rightarrow 1/\sqrt{2}(|0\rangle - |1\rangle)$). If it is applied to *n* qubits individually it can generate a superposition of 2^n classical states.

$$\begin{aligned} &(H \otimes H \otimes \ldots \otimes H)|00\ldots0\rangle \\ &= 1/\sqrt{2^n}(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \otimes \ldots \otimes (|0\rangle + |1\rangle) \\ &= \sum_{i=0}^{2^n-1} |i\rangle \end{aligned} \qquad (2.1)$$

There are also gates acting on multiple qubits. A very relevant quantum gate acting on 2 qubits is the controlled-not gate or in short CNOT gate. This gate flips the controlled qubit if the controlling qubit is 1 as shown below:

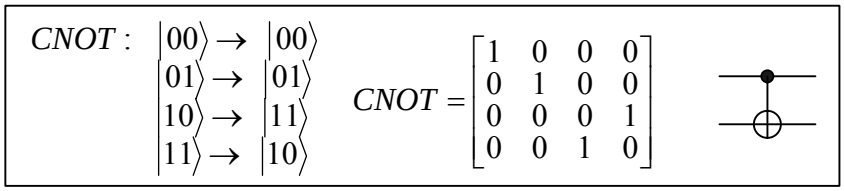

Figure 2.9: The CNOT (controlled-not) gate: BraKet notation, matrix representation, symbol

For 3 qubits there is the controlled-controlled-not gate also known as the Toffoli gate. This gate flips the controlled qubit if the 2 controlling qubits are 1. As a generalization of this 2 gates there is the controlled-controlled-…-controlled-U gate, where U is a unitary transformation that acts on one qubit. For the 2 gates (CNOT, Toffoli) mentioned above the single qubit transformation is $U=X$.

2.3.2 Quantum fault simulation

Maintaining a coherent, accurate quantum computation is not an easy task; there are some quantum specific issues one needs to deal with:

- *Phase errors.* The classical encoding provides no protection against phase shift errors.
- *Small errors. D*ue to quantum amplitudes, quantum information is not entirely digital therefore, an error may affect the amplitudes by a small amount of order ε, and these small errors can accumulate over time.
- *Measurement destroys superposed state*. In the classical error correcting schemes, one needs to measure the bits in order to detect and correct the errors. However, any measurement of a quantum state irreversibly disturbs it in quantum computing [21].
- *No cloning.* In the classical encoding, the information is protected by making extra copies of it. This is not possible in quantum computing since arbitrary quantum bits cannot be copied with perfect fidelity [5].
- *Decoherence* represents the distortion of the quantum state due to interactions with the environment [4].

Solutions for improving the quantum circuit reliability are mostly based on the classical theory of error detection and correction codes [8][9][10][11][12]. However, there are some which take advantage of new approaches such as reconfigurable gate arrays [31]. One of the simplest examples of a quantum error-correcting code is Shor's 3 qubit repetition code [12]. This code is based on the mechanism of majority voting and uses the following qubit encoding:

$$|0\rangle \rightarrow |000\rangle, \qquad |1\rangle \rightarrow |111\rangle \tag{2.2}$$

Shor's 3 qubit repetition code is capable of detecting and correcting a single error in the encoded block.

A very important quantum parameter is the accuracy *threshold*. The *accuracy threshold* is the physical gate error probability for which an arbitrary long quantum computation is possible with a given error probability [8]. Thus, all the error detection and correction codes aim at permitting an accuracy threshold which is as high as possible.

At this moment, it is too difficult to conduct a realistic assessment of quantum fault tolerance that does not depend on the details of the chosen fault and error model. Therefore, the fault and error models play an essential role in the

simulated fault injection process. The most widely accepted quantum faults are modeled by the Pauli operators [5][29]. These faults are: *bit-flip* (negation, modeled by *X* operator – Figure 2.5), *phase-shift* (modeled by the *Z* operator – Figure 2.6) and *both bit-flip and phase-shift* (modeled by the *Y* operator – Figure 2.7), and identity *I* as depicted bellow.

$\begin{matrix}\|0\rangle \to \|0\rangle \\ \|1\rangle \to \|1\rangle\end{matrix}$ $\sigma_I = \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$ Identity	$\begin{matrix}\|0\rangle \to \|1\rangle \\ \|1\rangle \to \|0\rangle\end{matrix}$ $\sigma_X = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$ Negation	$\begin{matrix}\|0\rangle \to -\|1\rangle \\ \|1\rangle \to \|0\rangle\end{matrix}$ $\sigma_Y = \begin{pmatrix}0 & 1\\ -1 & 0\end{pmatrix}$ Both Negation and Phase Shift	$\begin{matrix}\|0\rangle \to \|0\rangle \\ \|1\rangle \to -\|1\rangle\end{matrix}$ $\sigma_Z = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$ Phase Shift

Figure 2.10: The Pauli operators

Besides the Pauli operators, there are some other quantum fault models, as those described in reference [6]. These fault models proved to be very effective for quantum circuit testing [7].

- *Gate insertion:* in this case, we assume that the Pauli operators are inserted in the designated fault locations.
- *Gate removal*: any single gate can be removed from the system.

The most widely used error model is - *independent stochastic errors* [12][29][30][32][34] – the error model consists of having independent probabilistic errors represented by Pauli operators acting at error location. Their overall effect is estimated by means of classical error probability.

There are several approaches to quantum error simulation. The simulator constructed after Cirac and Zoller scheme of ion trap quantum computer [24], simulates errors by introducing operational errors and errors resulting from decoherence. The simulator implements gates as sequences of laser transformation. The operational errors are a result of altering these transformations. The drawback of such an approach is the fact that is a physical dependent implementation, which is relevant only for the ion trap computer.

A different approach uses C based programming languages (ANSI C, C++) in order to perform a Monte Carlo simulation [30][35]. The quantum computation is run several times and errors are introduced at each gate with some probability. The state of the quantum circuit is not stored. The only stored information for each

qubit is whether an error has occurred or not [35]. Thus, in the case of the Monte Carlo simulation the track of the fault propagation is kept rather than the evolution of the complete quantum state. Another approach consists of simulating quantum faults by means of a HDL-based tool - the QUantum ERror Injection Simulation Tool (QUERIST), an ongoing project with guidelines set in [33].

Chapter 3

Thesis overview

Writing a thesis involves activities such as the study and understanding of the state-of-the-art of the domains related to the scientific topic chosen. Furthermore, a very important aspect is related to the innovations and contributions of the research work. Such an activity can only take place in an academic environment where a group of people work to add up the pieces to a wider puzzle in order to successfully finish a complex research project. The proposed research direction is part of a complex framework QUERIST. The guidelines for this framework were proposed by dr. ing. Mihai Udrescu. He is a co-advisor for the proposed PhD thesis and one of the founders of the ACSA research group, the group that supports my research activity.

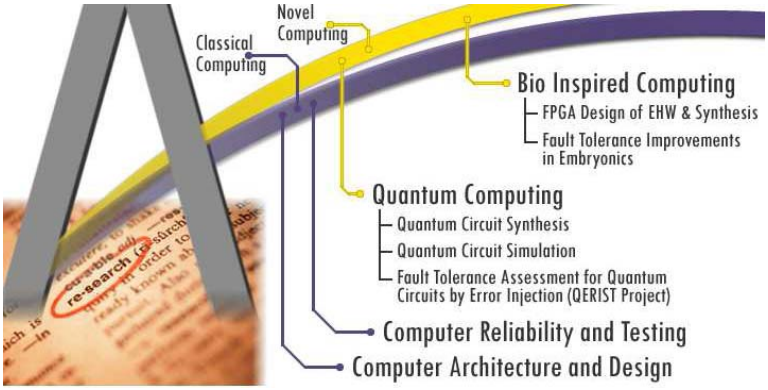

Figure 3.1: ACSA group overview

3.1 Proposed name

The title for my research project is: *"Simulation-Based Assesment of Quantum Circuit Reliability"*. The main target is that of finding an adequate methodology for assessment of quantum reliability by means of simulation. The starting point for this research is the classical theory of circuit reliability analysis and simulated based fault injection as a mean of computing different reliability parameters.

3.2 Thesis goals

The main objectives of the proposed research project, as well as the activities needed to successfully achieve them, are depicted in Table 3.1.

Table 3.1: Objectives of proposed research project

Year	Objectives	Associated activities
2008	1. The development of a theoretical basis for simulated based fault injection using HDL languages for quantum circuits.	1. Study of existing implementations and methodologies for simulated based fault injection for classical circuits and quantum circuits.
		2. The study of the quantum noise, quantum fault models. Study of existing error models.Review of literature.
		3. The VHDL modeling of quantum errors and faults.
	2. The implemetation of the QUERIST (Quantum Error Injection Simulation Tool) by adding the simulated based fault injection techniques	1. Refining the implementation of the proposed fault injection techniques.
		2. Driving complex simulated based fault injection campaigns in order to evaluate the effectiveness of the proposed techniques.
		3. Comparisson of the results with the ones claimed by the most recent and relevant publications.
	3. Dissemination – writting a PhD report entitled: *Simulation based fault injection techniques for quantum*	1. Gathering the theoretical knowledge and simulation results into a PhD report.
		2. Report presentation.

	circuits.	
	4. Development of the theoretical bases for quantum circuit logic partitioning	1. The study of the classical circuit logic partitioning techniques for combinational circuits.
		2. The developement of the mathematical model for the computation of the cuantum circuit reliability based on the reliability of its subcomponents. For this purpose the use of Markov chains or reliability graphs is considered.
2009	1. Developing a partitioning algorithm for quantum circuits.	1. The development of a logic partitioning algorithm for cuantum circuits.
		2. The study of the elaborated algoritm based on simulation campaigns.
		3. The writting of an article based on quantum circuit partitioning and then submitting it to a ISI indexed conference. Also, the participation to the conference.
	2. Dissemination – writting a PhD report entitled: *A logic partitioning based technique for quantum circuit reliability assessment*.	1. Gathering the theoretical knowledge and simulation results into a PhD report.
		2. Report presentation.
	3. Dissemination – the PhD thesis elaboration.	1. Results processing and comparisson with the latest reported results from this research field.
		2. Refining of PhD thesis and defending thethesis.
		3. Book contracting at Politehnica University of Timisoara in order to include the PhD thesis in the PhD series edited by The University Politehnica of Timisoara.

3.3 Thesis outline

In Table 3.2 a proposed thesis layout is presented. Of course, it is still early, and future results may overrule the information presented bellow. This can be considered today's image of how my PhD thesis will be structured. The second column from the table bellow briefly presents the intended contents of the thesis chapters.

Table 3.2: Proposed thesis layout

Thesis chapter	Comments
1. Introduction	Contains the problem statement. Also, the thesis structure and main contributions are stated.
2. Quantum background 2.1 Fault tolerance problems 2.2 Quantum noise 2.2.1 Fault model 2.2.2 Error model 2.2.3 Discussion 2.3 Quantum simulation 2.4 Quantum reliability: background	Presents some common knowledge regarding quantum computing. Then, it focuses on the quantum fault models and also the pattern on which these faults manifest. Furthermore a brief discussion about fault tolerance problems is also considered useful. Next, follow the quantum simulation drawbacks. Last but not least, the reliability metrics and parameters need to be presented.
3. Classical fault injection 3.1 Techniques and tools 3.2 Classical mutants 3.3 Classical saboteurs 3.4 Technique comparison	This chapter of the thesis is meant to deal with the techniques and tools developed for classical computation which are a useful source of knowledge and experience for the development of fault injection methodology for quantum circuits.
4. Quantum fault injection 4.1 Previous work 4.2 QUERIST tool 4.3 Simulator commands 4.4 Quantum mutants 4.5 Quantum saboteurs	Quantum fault injection techniques are presented in chapter 4. These techniques are presented, analyzed and compared from the performance and simulation resources required perspective, on the one hand. Also, on the other hand,

4.6 Comparison	parameters such as effort for setting up a simulation campaign, capacity of modeling, capacity of extracting the data concerning fault propagation are being tackled.
5. Simulation campaigns 5.1 First circuit 5.1.1 Simulation scenario 5.1.2 Fault injection entities 5.1.3 Results 5.2 Quantum double redundancy 5.2.1 Simulated circuit 5.2.2 Campaign description 5.2.3 Fault injection entities 5.2.4 Simulation results 5.3 Bell's circuit 5.3.1 Campaign description 5.3.2 Fault injection entities 5.3.3 Campaign results 5.4 Concluding remarks	This chapter is dedicated to the presentation of several simulation campaigns: the campaign's scenarios, the fault injection entities used described in detail, and also the assumptions considered for the simulations performed, the obtained results. A summary of the chapter as well as the conclusions drawn from the performed simulation experiments.
6. Logic partitioning 6.1 Background 6.2 Proposed algorithm 6.3 Experimental results	A review of literature concerning the partitioning algorithms and methods is presented in chapter 6. Furthermore, a solution for quantum circuit logic partitioning is proposed. Lastly, the experimental results are depicted.
7. Conclusions and future work	Provides a summary of the thesis, and proposed future research directions based on the results.
Bibliography Appendix	

3.4 Timelines

According to the PhD program, the second phase of the program should last maximum 2 years. Two PhD reports are needed and of course a PhD thesis must be

defended. In order to fulfill these requirements a list of objectives and associate activities was developed and described in Table 3.1.

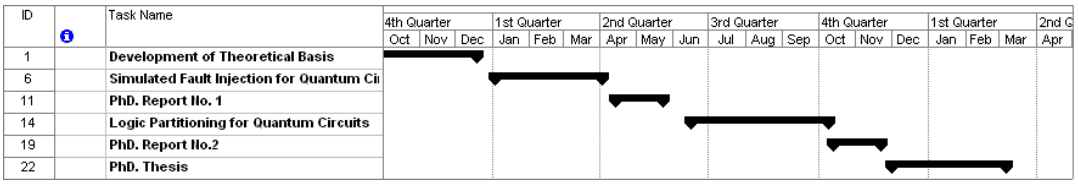

Figure 3.2: Schedule for the main objectives of the PhD thesis

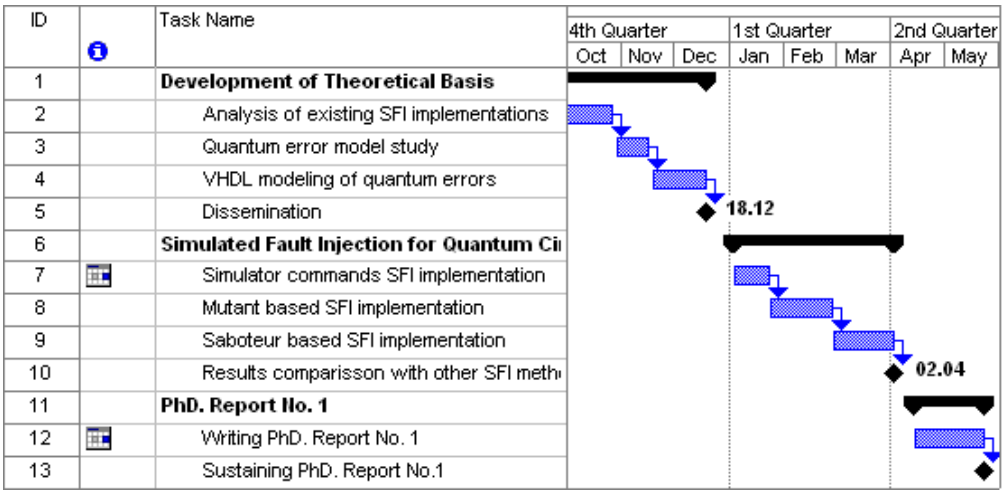

Figure 3.3: Schedule for objectives and associated activities for the academic year 2007-2008

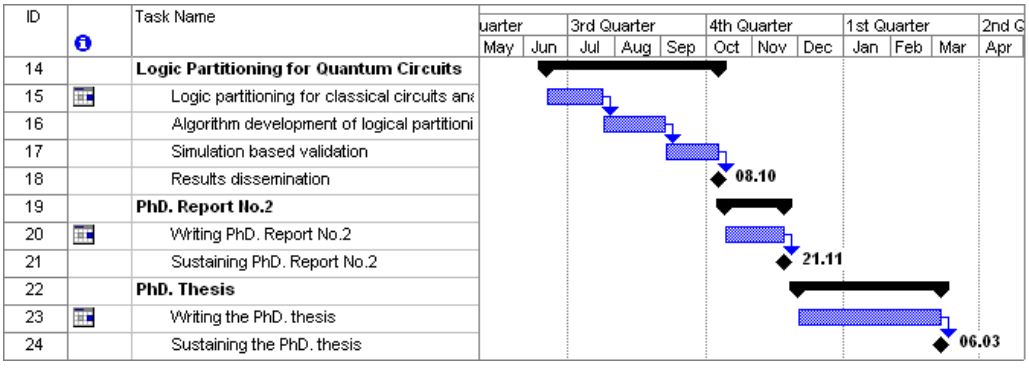

Figure 3.3: Schedule for objectives and associated activities for the academic years 2008-2009

The activities described in Table 3.1 are scheduled to take place according to the timelines depicted by Figure 3.3 for the academic year 2007-2008. Furthermore the tasks with the proposed deadlines for the academic year 2008-2009 are presented by Figure 3.4.

3.5 Dissemination

There are two dissemination directions:

- On the one hand, there are the two PhD reports:
    1. The first PhD report entitled *"Simulation based fault injection techniques for quantum circuits"*.
    2. And the second PhD report: *"A logic partitioning based technique for quantum circuit reliability assessment"*.

    And last, but not least the writting of the PhD thesis: *"Simulation-Based Assesment of Quantum Circuit Reliability"*.
- On the other hand there are the publications which are to be submitted at conferences, workshops and journals. For the conferences and workshops the candidates are: Annual Simulation Symposium (ANSS- ISI indexed), European Test Symposium (ETS- ISI indexed), Euromicro Conference on Digital System Design (DSD- ISI indexed). There are also a number of journals of interest: IEEE Design and Test of Computers, IEEE Transactions on Computer Aided Design, ACM Journal on Emerging Technologies in Computer Systems.

3.6 Potential contributions

The proposed PhD research program - *"Simulation-Based Assesment of Quantum Circuit Reliability"* - aims at developing a methodology for quantitatively assessing quantum circuit reliability. The main contributions of this research would be:

- A review of literature concerning: quantum noise, quantum circuit reliability, classical simulated based fault injection, classical logic decomposition.

- An as accurate as possible model of the quantum noise. The HDL modeling of existing error models.
- Developing a viable methodology for simulation based fault injection for quantum circuits.
- Finding a solution for making the developed reliability assessment techniques scale. A possible solution is logic partitioning. Thus, a potential contribution might be the development of an algorithm for quantum circuit partitioning.
- The implementation of the fault injection techniques and the partitioning algorithm in a tool. The process is intended to be automated, so that through simulation, a statistically significant sample is obtained in order to provide a quantitative means to assess quantum circuit reliability.

Chapter 4

Conclusions

In this report the research program and a review the of the literature describing quantum computing, quantum fault tolerance issues and classical fault injection were presented. Previous work is described in Chapter 2. There are two domains – on the one hand, there are is classical theory with fault injection techniques and logic partitioning and a rich experience with HDL modeling and tools; - and on the other hand there is the quantum domain plagued by quantum noise, which requires new solutions to improve reliability and also, there is the need to study for a better understanding of the effect of quantum errors.

The following chapter introduces the proposed research directions with refined steps o consider for reaching the goals depicted in Subsection 3.2. Furthermore, the timelines for the proposed activities together with the dissemination activities are presented. Last, but not least, the potential contributions of the research activities are illustrated by Subsection 3.2.

The proposed PhD research program - *"Simulation-Based Assesment of Quantum Circuit Reliability"* - aims at developing a methodology for simulation based fault injection for quantum circuits. A desired feature of the fault injection methodology is scalability. Nevertheless, quantum circuit simulation requires simulation resources that grow exponentially with the number of simulated qubits. Although, simulation techniques that improve the simulation cost and/or time were developed [22][23][25], still for complex and large structures it is not feasible to apply fault injection. A proposed solution is logic partitioning. The complex and large quantum circuit is decomposed into less complex parts. Fault injection is used to evaluate quantum reliability for these parts. The reliability for the larger circuit is assessed by means of a reliability graph by taking into account the reliability of the smaller parts. I also want to focus on automating as mush as possible of the process of creating, running and post-processing the results of a simulation campaign. In this manner, a statistically significant sample can be obtained in order to provide a quantitative means to assess quantum circuit reliability.

Bibliography

[1] ARDA. A quantum Information Science and Technology Roadmap. http://qist.lanl.gov.5

[2] ****, Emerging Research Devices with a New Section on Emerging Research Materials, International Technology Roadmap for Semiconductors (ITRS) Report (2005 Update), www.itrs.net/Common/2005Update/, (2005).International Technology Roadmap for Semiconductors. Emerging research devices. (2005)

[3] Centre for Quantum Computation web page. www.qubit.org.

[4] L. Spector, *Automatic Quantum Computer Programming. A Genetic Programming Approach*, Kluwer Academic Publishers, (2004)

[5] E. Rieffer, W. Polak, *An Introduction to Quantum Computing for Non-Physicists*, nline preprint quantum-ph/9809016, (1998)

[6] M. Perkowski, J. Biamonte, M.Lukac, *Test Generation and Fault Localization for Quantum Circuits*, Proceedings of the 35th International Symposium on Multiple-Valued Logic (ISMVL'05), (2005)

[7] J.P. Hayes, I. Polian, B. Becker, *Testing for Missing Gate Faults in Reversible Circuits*, Proceedings of the 13th Asian Test Symposium (ATS 2004), (2004)

[8] J. Preskill, *Fault-Tolerant Quantum Computation*, Online preprint Quant-ph/9712048, (1997)

[9] A.M. Steane, *Error correcting codes in quantum theory*, Phys. Rev. Lett. 77, 793 (1996)

[10] D. Gottesman, *Stabilizer Codes and Quantum Error Correction*, PHD Thesis, California Institute of Technology Pasadena, California, (1997)

[11] E. Knill, R. Laflamme, *Concatenated codes*, Online preprint quant-ph/9608012, (1996)

[12] E. Knill, R. Laflamme, A. Ashikhmin, H. Barnum, L. Viola, W.H. Zurek, *Introduction to Quantum Error Correction*, Online preprint Quant-ph/0207170, (2002)

[13] K. Pradhan, *Fault-Tolerant Computer System Design*, Prentice Hall, (1996)

[14] P. H. Bardell, W. H. McAnney, J. Savir, *Built-In Test for VLSI: Pseudorandom Techniques*, A Wiley-Interscience publication, (1989)

[15] M. Rimen, J. Ohlsson, J. Karlsson, E. Jenn, J. Arlat, *Validation of Fault Tolerance by Fault Injection in VHDL Simulation Models,* Raport LAAS No.92469, (1992)
[16] E. Jenn, J. Arlat, M. Rimen, J. Ohlsson, J. Karlsson, *Fault Injection into VHDL Models: The MEFISTO Tool*, 24th Annual International Symposium on Fault Tolerant Computing (FTCS-24), pp 66-75, (1994)
[17] T.A. DeLong, J.A.. Profeta III*, A Fault Injection technique for VHDL Behavioral-Level Models*, IEEE Design and Test of Computers, pp 24-33, winter, (1996)
[18] J.C. Baraza, J. Gracia, D. Gil, P.J. Gil, *Improvement of Fault injection Techniques Based on VHDL Code Modification,* IEEE International Design Validation and Test Workshop, pp 19-26, (2005).
[19] Institute od Electic and Electonic Engineers (IEEE). *IEEE Standard VHDL Language Reference Manual*, IEEE STD 1076-1993.
[20] M.Udrescu, PhD report No. 3, www.acsa.upt.ro, (2004)
[21] M.A. Nielsen, I.L. Chuang, Quantum *Computation and Quantum Information*, Cambridge University Press, (2000)
[22] G. F. Viamontes, I. L. Markov and J. P. Hayes, *Improving Gate-Level Simulation of Quantum Circuits*, Quantum Information Processing vol. 2(5), pp. 347-380, October, (2003)
[23] G. F. Viamontes, I. L. Markov and J. P. Hayes, *Graph-based Simulation of Quantum Computation in the Density Matrix Representation*, Quantum Information and Computation, vol.5, no.2 pp. 113-130, quant-ph/0403114, February, (2005)
[24] K. M. Obeland, A. M. Despain, *A parallel quantum computer simulator,* High performance computing, Online preprint quant-ph/9804039, (1998)
[25] M. Udrescu, L. Prodan, M. Vladutiu, *The Bubble Bit Technique as Improvement of HDL-Based Quantum Circuit Simulation*, Proceedings IEEE 38th Annual Simulation Symposium, pp. 217-224, (2005)
[26] P. Dirac, *The Principles of Quantum Mechanic (fourth edition)*, Oxford University Press, (1958)
[27] A. Barenco, C.H. Bennett, R. Cleve, D. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, H. Weinfurter, *Elementary gates for quantum computation*, Phys. Rev. A (52), 3457-3467, (1995)
[28] D. Deutsch, *Quantum computational networks*, Proc. Roy. Soc. Lon. A. 425, 73, (1989)
[29] E. Knill, E. Laflamme, W. H. Zurek, *Resilient Quantum Computation: Error Models and Thresholds*, Online preprint quant-ph/9702058, (1997)
[30] C. Zalka, *Threshold Estimate for Fault Tolerant Quantum Computing*, Online preprint quant-ph/9612028, (1997)
[31] M. Udrescu, L. Prodan, M. Vladutiu, *Improving Quantum Circuit Dependability with Reconfigurable Quantum Gate Arrays*, Proceedings ACM 2nd International Conference on Computing Frontiers (CF'05), Ischia, Italy, pp. 133-144, (2005)

[32] E. Knill, *Fault Tolerant Postselected Quantum Computation: Threshold Analysis*, Online preprint Quant-ph/0404104, (2004)
[33] M. Udrescu, L. Prodan, M. Vladutiu, *Simulated Fault Injection in Quantum Circuits with Bubble Bit Technique*, 7th International Conference on Adaptive and Natural Computing Algorithms (ICANNGA), Springer, pp 276-279, (2005)
[34] A.M. Steane *Space, time, parallelism and noise requirements for reliable quantum computing*, Online preprint quantum-ph/9708021, (1997)
[35] A.M. Steane *Overhead and noise threshold of fault tolerant quantum error corection*, Phys. Rev. A68, (2003)