Optimizing the Number of Gates Representing an FTA k/n Operator on a Quantum Computer

Reni Banov
Zagreb University of Applied Sciences
Vrbik 8, Zagreb, Croatia
reni.banov@tvz.hr

ABSTRACT

The classic static fault tree analysis is a deductive method based on Boolean logic widely used for reliability and safety analysis. The method is based on the application of logical operators to show the relationship between component failures that lead to intermediate or top-level adverse events. At the lowest level, it is assumed that the events are basic and that the probability of their occurrence is fixed according to the Bernoulli distribution, for instance, the probability of a failure occurring for a component. Recently, it has been shown that such events can be successfully simulated by means of qubits and that events on middle and highest levels can be written by using quantum gates that simulate logic operators, i.e., can be written utilizing a quantum fault tree. The quantitative and qualitative analysis of the static fault tree can then be simulated on that record aided by a quantum computer. In this article, in addition to the standard set of logical operators (and, or, not), we further explore the static fault tree record that contains the commonly used k/n logical operator. Although the k/n operator can be written employing a combination of a standard set of logical operators, however, such notation is usually not optimal for implementation on a quantum computer, except for the simplest cases. Therefore, in this paper we present a model for optimizing the number of quantum gates in a quantum circuit record for events that are described in a static fault tree by the k/n logic operator. In addition, in the given example of such a static fault tree consisting of six basic events and four derived events we closely present the calculation of the top event probability and the determination of all state vectors in the quantum fault tree that represent minimal cut-sets of the latter. Eventually, we analyse the usage and scaling of resources in a quantum computer and additionally provide a summary of practical limitations in the application of a hybrid static fault tree recording strategy on potential quantum computers in the near future.

Keywords: *Probabilistic Safety Assessment (PSA), Fault Tree Analysis (FTA), Quantum Fault Tree (QFT)*

1 INTRODUCTION

The classical definition of a static fault tree is based on the application of a deductive procedure founded on the laws of Boolean algebra. It is often used in engineering to assess the reliability of components and subsystems and to assess the risk of failures in technical systems. It basically consists of a model of a combination of basic and intermediate events associated with logic gates (AND, OR, etc.) and their impact on the functional state of the system. A special role is assigned to combinations of basic events (the so-called minimal cut-sets) being sufficient to achieve an undesirable event modelled by the fault tree. Due to the combinatorial complexity, finding a part or an entire set of minimal cut-sets is altogether demanding for the classical approach to the fault tree analysis and is often carried out by using approximate methods. Nevertheless, a static fault tree is undoubtably a widely used technique for risk and reliability assessment in many engineering disciplines, such as nuclear energy, aerospace, automotive and many other fields.

More recently, the classical fault tree has been expanded by the idea of applying quantum-mechanical techniques to analyse it. The idea is to represent basic events with quantum bits (qubits) and to record logic gates using quantum circuits [1]. Basically, by running a quantum circuit on a quantum computer, the probabilistic behaviour of a combination of events in a system modelled by a fault tree is simulated. Multiple execution of a quantum circuit achieves a realistic simulation of the probability of occurrence of an adverse event modelled by a fault tree. Along with that, the authors showed in their new paper [2] how the application of the Grover form of quantum amplitude amplification can more efficiently determine a set of minimum cut-sets for large fault trees. Based on a similar idea of applying the Grover's search, a process for finding a set of minimum cut-sets using quantum annealing was patented several years ago [3]. In essence, according to the aforementioned authors the idea of recording a fault tree on a quantum computer and applying quantum algorithms can lead to faster calculations or new possibilities for probabilistic analysis of large fault trees.

This article is organized into several sections. In the second section, a brief review of the literature is given and the record of the static fault tree is described by the quantum fault tree model. The third section presents the main result where a combined approach to the analysis of a static fault tree containing voting gates is described. In this part, a quantum mechanical model after the application of an algorithm for the voting gate decomposition was presented on the sample static fault tree. From the obtained model, after simulation of execution on a quantum computer the method of determining minimal cut-sets on a classical computer is given. Finally, in the last part, conclusions and comments are given, including restrictions and opportunities for further research.

2 QUANTUM FAULT TREE

The basic element of the static fault tree record is represented by elementary events that are sampled by a random variable with Bernoulli distribution. Bernoulli's distribution describes simple events that are distinguished at the elementary level of occurrence or non-occurrence. In other words, we observe a simple distribution of random events where the probability of occurrence of an event is given by value p, while the probability of non-occurrence is determined by the complementary value q=1-p, or vice versa. Due to this property of the complementarity of the probability of occurrence/non-occurrence of an event according to the Bernoulli distribution, it is to represent events efficiently with quantum bits (qubits) on a quantum computer. Qubits in quantum computing are the basic units of information for implementation of probabilistic algorithms on a quantum computer.

2.1 Qubits

A qubit in quantum computing is the simplest description of a quantum mechanical system with two basic states, that is, the orthonormal basis vectors from a two-dimensional unitary complex vector space (2-dimensional Hilbert space). The basis vectors are most frequently denoted by $|0\rangle$ and $|1\rangle$ in Dirac notations [4]. Basically, a qubit uses the quantum mechanical property of superposition to represent a state that corresponds to a linear combination of these two vectors.

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad \alpha, \beta \in \mathbb{C} \tag{1}$$

Since one qubit represents a quantum mechanical system with two possible basic states, the complex values from (1) additionally meet the normalization condition for the probability amplitude value. A linear combination of two basis vectors can represent any relationship of these basic states with a certain probability of occurrence of a state denoted by $|0\rangle$ or a state denoted by $|1\rangle$.

$$|\alpha|^2 + |\beta|^2 = 1 \tag{2}$$

The basis vectors from (1) are usually represented by vectors from a standard computation base that simplifies the recording of the state of a multi-qubit quantum mechanical system. For example, in a standard computation base, the notation of basis vectors in single qubit system is

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix},\ |1\rangle = \begin{bmatrix}0\\1\end{bmatrix} \tag{3}$$

while the state of a two-qubit quantum mechanical system is depicted by using basis vectors

$$|00\rangle = \begin{bmatrix}1\\0\\0\\0\end{bmatrix},\ |01\rangle = \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, |10\rangle = \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, |11\rangle = \begin{bmatrix}0\\0\\0\\1\end{bmatrix} \tag{4}$$

which represent the four basic states of that system. In the computation base, changes in the state of a quantum mechanical system are described by unitary operators that are presented using quantum gates.

2.2 Quantum gates

Basically, we write every unitary operator using a matrix determined by the action of that operator over the basis vectors from the notation of the quantum mechanical system. In doing so, we distinguish operators according to the dimension of the complex unitary vector space over which the operator is applied, i.e. the number of basis vectors. For example, the unitary operator represented by a 2x2 matrix defines a change in the state of a one qubit quantum mechanical system since it is described by two basis vectors from a two-dimensional complex unitary vector space. In practical application, to describe the change in the state of a quantum mechanical system we most frequently use a standard set of operators (quantum gates)

- 1-qubit: Pauli X, Y, Z, and Hadamard (H)
- 2-qubit: controlled negation (CNOT)
- 3-qubit: Toffoli (CCNOT)

There are numerous unitary operators that can be defined over quantum mechanical systems with multiple qubits [5], however, as shown in [2], only a fraction of them is important for the implementation of gates from static fault trees. More complex unitary operators acting on multi-qubit quantum mechanical systems are defined using the Kronecker product of matrix representation of simpler unitary operators. For example, on a 2-qubit quantum mechanical system, let's apply the Pauli-X (NOT) gate on the first qubit, and the Hadamard (H) gate on the second. The unitary operator describing the operation of such quantum gates is determined by the Kronecker product

$$X|\Psi_1\rangle \otimes H|\Psi_2\rangle = (X \otimes H)(|\Psi_1\rangle \otimes |\Psi_2\rangle) \tag{5}$$

which can be written in matrix form of the computation base

$$X \otimes H = \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix} \otimes \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix} \otimes \begin{bmatrix}1 & 1\\1 & -1\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}0 & 0 & 1 & 1\\0 & 0 & 1 & -1\\1 & 1 & 0 & 0\\1 & -1 & 0 & 0\end{bmatrix} \tag{6}$$

2.3 Quantum circuit

Probabilistic algorithms for execution on a quantum computer are standardly described by quantum circuits. In the quantum circuit model, the lines represent the qubits while the quantum gates represent the unitary operators applied to those qubits. For example, Figure 1 shows the probabilistic algorithm described by the action of the unitary operator from equation (6)

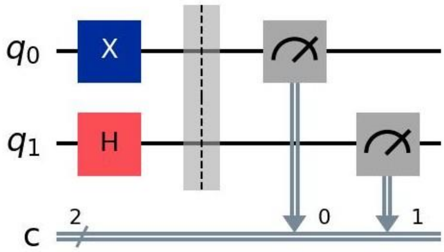

Figure 1: Two-qubit system with Pauli-X and Hadamard gate

In this figure, the first qubit is represented by a horizontal line named q_0, while the horizontal line named q_1 represents the second qubit. The application of unitary operators on qubits is indicated by quantum gates marked X and H. In addition to that, the figure shows horizontal double line that represent a classical two-bit register for storing the binary value obtained after the application of the measurement. It is important to note that according to the laws of quantum mechanics; after applying a measurement, qubits change their state into the measured state recorded in the classical register.

2.4 Quantum circuit encoding a Fault Tree

A static fault tree basically represents a record of causal relationships between events using a directed acyclic graph structure. Although numerous logical relationships can appear in the Fault Tree (AND, OR, XOR, NOT, voting, …), the authors in the paper [2] presented an algorithm for determining a quantum circuit that represents a Fault tree based only on AND/OR logical operators. Their algorithm can also be applied to fault trees that contain voting gates (k/n) after expansion of these gates into a combination of AND/OR gates with a variable number (k) of inputs. For example, a voting gate G2 in a simple fault tree from Figure 2 can be written after expansion in the following form

$$G_2 = (G_3 \wedge E_2) \vee (G_4 \wedge E_2) \vee (G_5 \wedge E_2) \vee (G_3 \wedge G_4) \vee (G_3 \wedge G_5) \vee (G_4 \wedge G_5) \qquad (7)$$

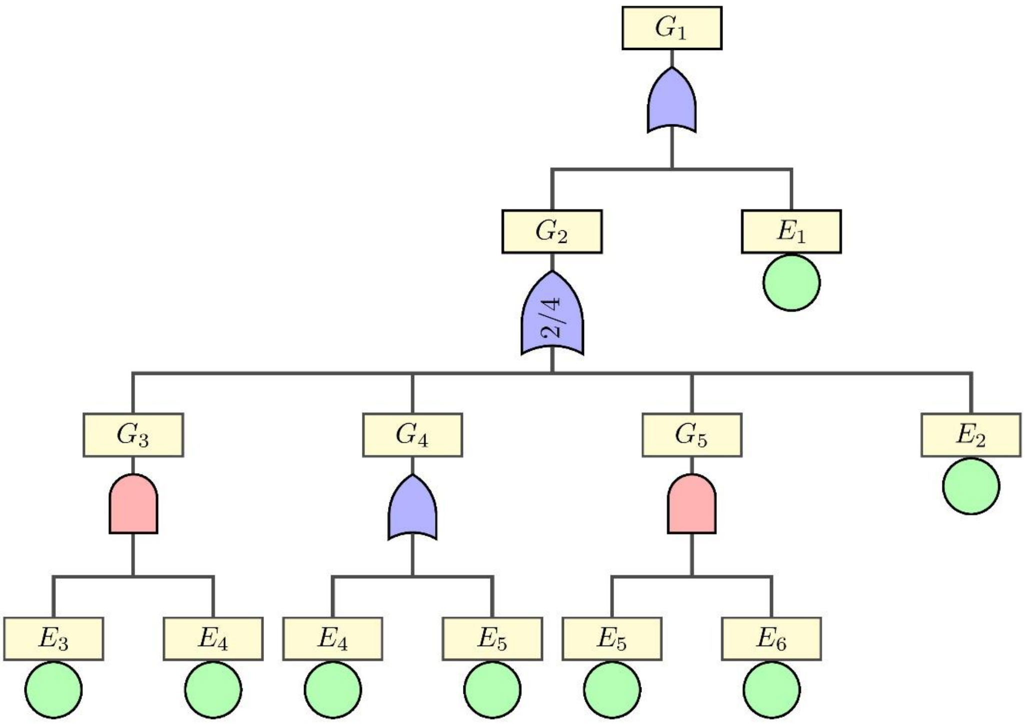

Figure 2: Fault Tree example with single voting (2/4) gate

A quantum circuit that implements a probabilistic algorithm to simulate this fault tree is shown in Figure 3. The figure highlights two blocks of the quantum circuit important for the quantitative analysis of the fault tree. The first highlighted part ("*Init part*") represents the initialization of six qubits for the basic events according to the Bernoulli random variables. In other words, the qubits are placed in a superposition state that corresponds to the probability of the occurrence of the basic event.

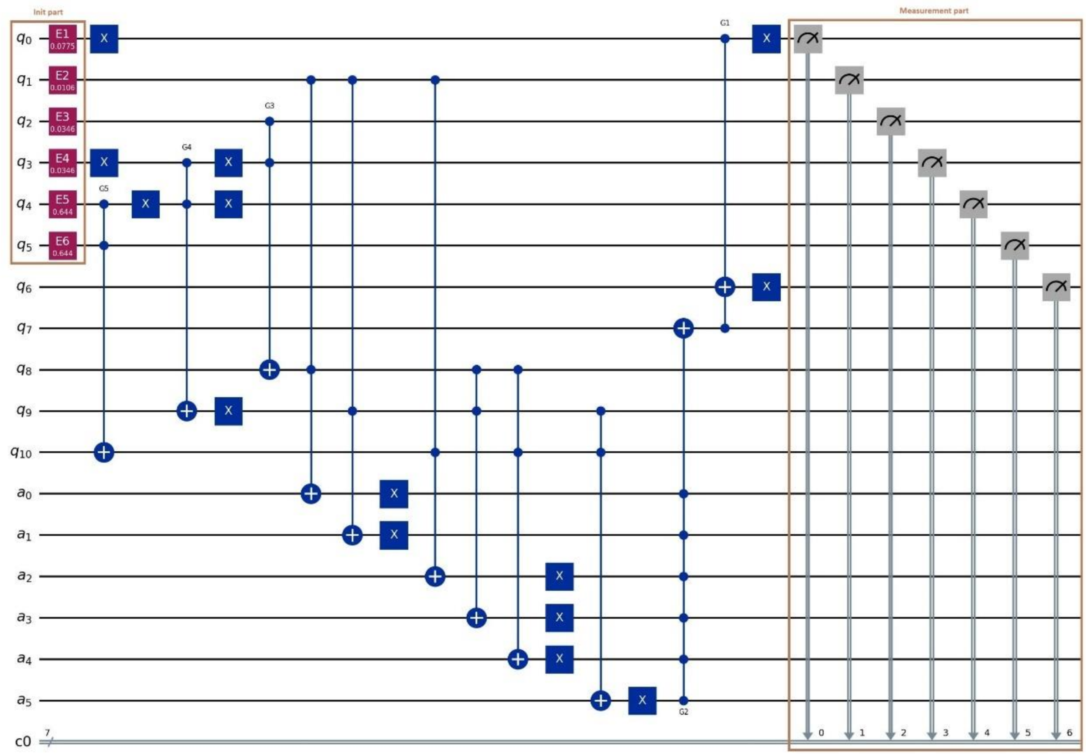

Figure 3: Quantum circuit encoding a sample Fault Tree from Figure 2

The probability of measuring qubits in one of the two basic states can be represented by the position vector on the Bloch sphere (Figure 4). Namely, the position vector on the Bloch sphere in a most general form represents the following qubit state

$$|\Psi\rangle = cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\varphi}\, sin\left(\frac{\theta}{2}\right)|1\rangle \tag{8}$$

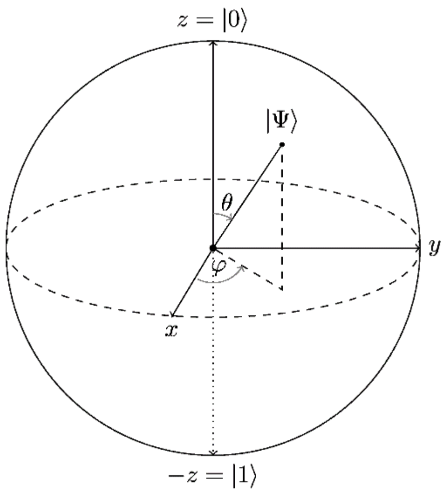

Figure 4: Qubit state vector $|\Psi\rangle$ as a position vector on Bloch sphere

The angle θ between the position vector and the z-axis on the Bloch sphere determines the alignment of the state vector with the basic states of one qubit. Therefore, the probability of the occurrence of each basic event can be given using the rotation operator applied to the basic state $|0\rangle$. According to the authors [2], by applying the rotation Y-operator to an angle from the equation (9) the qubit is placed in a state that corresponds to the probability of measuring qubit in the state $|1\rangle$, i.e., the probability p of the occurrence of the basic event.

$$\theta = 2\tan^{-1}\sqrt{\frac{p}{1-p}} \tag{9}$$

The second highlighted part (*"Measurement part"*) in Figure 3 shows the part of the quantum circuit intended to determine the classical state of a quantum mechanical system at the time of measurement. Namely, by executing the circuit multiple times on a quantum computer, we collect the frequency of occurrence of the classical state of qubits in the system. For example, for the fault tree in Figure 2, a frequency is collected for the qubit state $(q_0, \ldots, q_6)$ that corresponds to the combination of the basic events $(E_1, \ldots, E_6)$ and the resulting top event state. By subsequent classical processing of the frequencies from the obtained classical states we can determine the minimal cut-sets that lead to the top event.

The rest of the quantum circuit between the mentioned two barriers represents the implementation of the logical connections defined by the fault tree. Logical AND/OR gates in the fault tree are implemented using reversible Toffoli quantum gates for quantum AND/OR operators. The quantum AND operator are implemented by a Toffoli gate that stores the result of a logical AND operation without changing the state of the input qubits. The quantum OR operator uses De Morgan's rule for implementation, according to which the logical OR operator corresponds to the inverse of the result of the logical AND operator applied to the inverted inputs. Both quantum gates store the result in auxiliary qubits, which are later used in the circuit as inputs to other gates.

3 QFT APPROACH ON FAULT TREES WITH VOTING GATES

In the previous section, on the example of a simple fault tree with a voting gate G_2, the traditional expansion into the combination of AND/OR logical gates have been applied. Such expansion resulted in the quantum circuit implementation of six AND combined with five OR logic gates shown in Figure 3. The resulting record in the quantum circuit is implemented by six Toffoli gates combined with single multi-controlled (6-inputs) Pauli X quantum gate and six additional qubits $(a_0, \ldots, a_5)$ to store the intermediate results. In addition to the ancilla qubits, for simulation on a quantum computer these multi-input gates are converted into a combination of elementary (1,2,3-qubit) gates. Such a gate conversion process commonly results in an increase of the quantum circuit depth, and as a consequence increased the execution time. Therefore, it is advisable to expand the voting gate with the smallest number of logical AND/OR gates before implementation on quantum circuit. The algorithm from [6] developed for the purpose of static fault tree analysis using conventional methods proves to be useful for such expansion. By applying this algorithm, we can write G_2 voting gate in a simpler form that requires a smaller number of logical AND/OR gates than the traditional expansion.

$$G_2 = ((G_3 \vee G_4) \wedge (G_5 \vee E_2)) \vee (G_3 \wedge G_4) \vee (G_5 \wedge E_2) \tag{10}$$

After applying the algorithm from [6], the resulting equation (10) requires the implementation of only three AND with four OR logic gates in the quantum circuit. Due to the smaller number of AND/OR logic gates, less additional qubits are required for the implementation in a quantum circuit (Figure 5). For example, in the quantum circuit from Figure 5, five additional qubits $(a_0, \ldots, a_4)$ are needed instead of six as in the implementation of traditional expansion of that voting gate. In addition, five Toffoli gates and one multi-controlled (3-inputs) Pauli X gates are required to implement logic gates in a quantum circuit. Thus, we can conclude from the example above, that the application of the algorithm from [6] resulted in multiple reductions in the implementation of the voting gate with a quantum circuit. First of all, the number of additional qubits and Toffoli gates has been reduced by one and the number of inputs for multi-controlled Pauli X gate has been reduced from six to three.

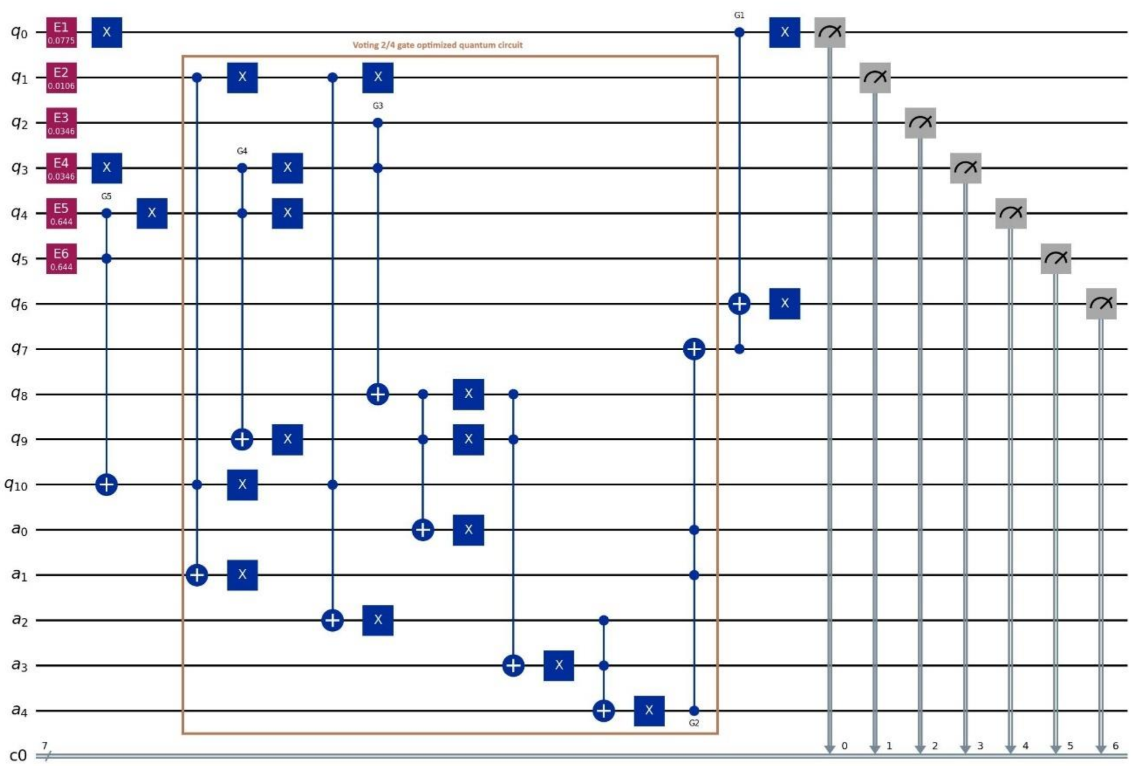

Figure 5: Quantum circuit for encoding of the optimized FT from Figure 2

3.1 Minimal cut-sets identification

According to the measurement postulate of quantum mechanics [5], by multiple execution of quantum circuit and by measuring the state, we can determine only the relative frequencies of the basic states for the selected set of qubits. In other words, if the state of a k-qubit system is described by a unitary vector space spanned by a set of 2^k basis vectors $\{|\psi_i\rangle\}$, then from relative frequencies we can only estimate the probability of basic states, i.e. the following equation is valid

$$P(|\psi_i\rangle) \approx \frac{n_i}{n} \qquad (11)$$

where n is the number of executions (or simulations) of the quantum circuit, while n_i is the number of occurrences of the *i*-th basic state (basis vector). For example, for a 2-qubit quantum system, we can determine only relative frequencies and estimate the probability of basic states represented by basis vectors $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ from the computation base. Accordingly, for the quantum circuit in Figure 5, we will measure the relative frequencies and estimate the probability for the basic states defined by the state of basic events (qubits $q_0, \ldots, q_5$) and the state of the top event (qubit q_6). Since the measurement after the simulation yields only one state, the minimum probability for any state (the cut-off probability) is limited by the number of simulations. Figure 6 shows the relative frequencies for the basic states of a quantum circuit from Figure 5 after $n = 10^6$ simulations of execution on a quantum computer, i.e., cut-off probability is *1E-6*.

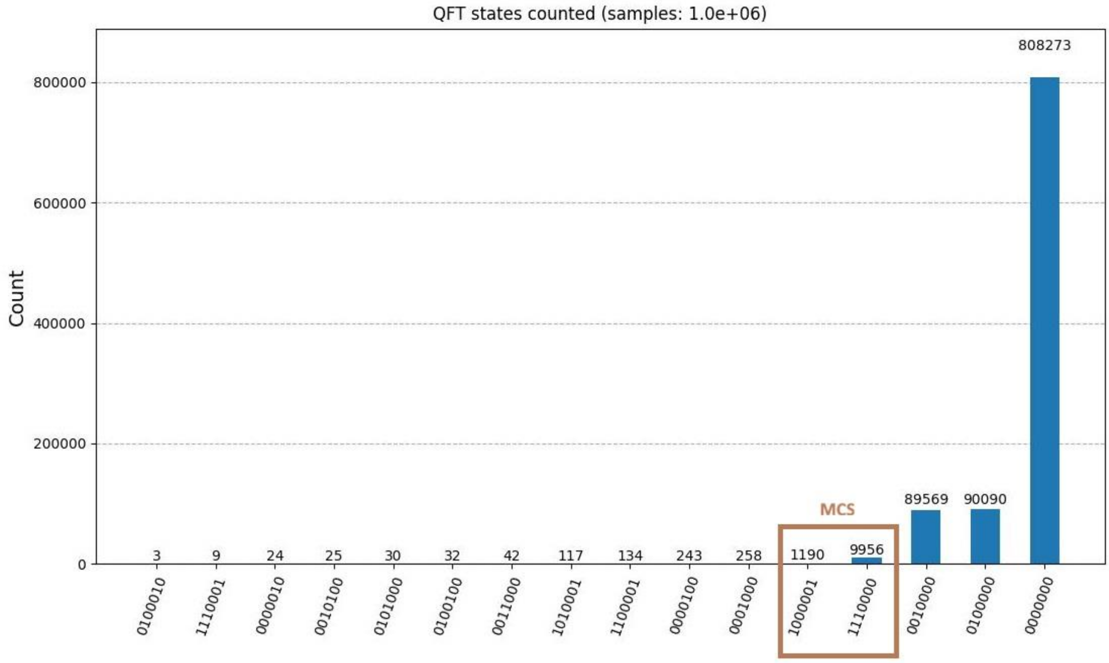

Figure 6: Collected state vector frequencies for the quantum circuit from the Figure 5

From the obtained set of the states and their relative frequencies, we can determine the set of minimal cut-sets with the following classical algorithm.

Algorithm MCS

Minimal cut-sets identification from the quantum circuit execution (or simulation)

Inputs: Set of state vectors $S = \{|\psi\rangle_i : (q_k, \ldots, q_0)_i\}$ with their relative frequencies n_i
Number of executions *n*

Outputs: Set of minimal cut-sets with their estimated probabilities

- Select the set of cut-sets from the state vectors, $CS = \left\{c_i : \left(1, q_{k-1}, \ldots, q_0\right)_i\right\} \subset S$
- Sort lexicographically a CS set
- Initialize the set of minimal cut-sets to empty $MCS = \{\emptyset\}$
- For each cut-set c_i from *CS*:
  - If $((c_i \wedge m_j) \neq m_j)\ \forall m_j \in MCS$ then
    - $MCS = MCS \cup c_i$
    - Estimate probability of the cut set $P(c_i) = \frac{n_i}{n}$

Let us apply the MCS algorithm on the set of state vectors and their relative frequencies from Figure 6. In the description of the execution of the MCS algorithm on these results, we assume the following order of qubits: q_6 represents the state of the top event, while $q_5, \ldots, q_0$ represents the state of the basic events $E_6, \ldots, E_1$ respectively. In the first step, we create a set of cut-sets from the state vectors by selecting those state vectors for which the value of q_6 qubit is equal to 1, i.e. we select state vectors only for combinations of the basic events that cause the top event occurrence. After the ascending sort in the second step, the set of cut-sets is

$$CS = \{1000001, 1010001, 1100001, 1110000, 1110001\}$$

In the third step, we initialize an empty set of minimal cut-sets. Next, we process all five elements of the *CS* set:

1. *MCS* set is empty, we add the first cut-set 1000001 from the *CS*, and we estimate the probability of that cut-set as

$$P(1000001) \approx \frac{1190}{1E6} = 0.00119$$

2. For the next cut-set 1010001, condition $(1010001 \land 1000001) \neq 1000001$ is not fulfilled therefore it is not minimal.
3. In the same way, for the cut set 1100001 condition $(1100001 \land 1000001) \neq 1000001$ is not fulfilled, therefore, it is not minimal either.
4. For the next cut-set 1110000 condition $(1110000 \land 1000001) \neq 1000001$ is fulfilled, therefore it is minimal so we add it to the *MCS* set and we estimate probability as
$$P(1110000) \approx \frac{9956}{1E6} = 0.009956$$
5. For the last cut-set 1110001 the condition $(1110001 \land 1000001) \neq 1000001$ is not fulfilled therefore it is not minimal.

After the algorithm is completed, the *MCS* set contains two elements $\{1000001, 1110000\}$ that represent the binary notation of the minimal cut-sets $\{\{E_1\}, \{E_5, E_6\}\}$ and the assessment of their probability of occurrence. From the obtained set of minimal cut-sets, we approximately calculate the reliability parameters using the conventional approach [7,8]. For example, the acceptable value for the probability of a top event is estimated with minimal cut set upper bound equation

$$Q_{TOP} \leq 1 - \prod_{i=1}^{N_m} \left(1 - P(c_i)\right) \approx 0.0114 \qquad (12)$$

where N_m is the number of known minimal cut-sets. In addition, the importance measures can also be approximated with a conventional approach. It is important to note that the final number of obtained minimal cut-sets depends on the number of times the quantum circuit is executed on a quantum computer or its simulation. Namely, the number of executions of a quantum circuit determines the lower bounds for the probability of the minimal cut-set that can be detected by this approach.

4 COMMENTS

The presented approach to fault tree analysis raises some important practical questions for limitations on the application to which we can give only partial answers. The first question that arises is related to the generalization and scaling of the presented optimization according to different k/n combinations. As shown in the article [6], decomposition using their algorithm is applicable for different combinations of the number of inputs to the k/n gate with limitations determined by the available resources. It is important to point out that the application of traditional combinatorial development for voting gates results in the generation of C_k^n combinations of k-sized conjunctions (AND gates) connected with classical OR gates, while the application of the new algorithm results in a smaller number of classical AND, OR gates. By applying the algorithm from the article, the authors successfully demonstrated a significant reduction within the required resources (space, time) and its applicability for k/n gates that result in a combinatorial explosion of the traditional approach, for example, applicability on k values greater than 8 and a number of inputs greater than 16.

Another practical limitation for application is the depth of the quantum circuit and it is determined by technological properties of the currently available quantum gates. Typical values for the quantum circuit range depth are a few hundred quantum gates, and the primary technological limitation in this regard is the decoherence time and the occurrence of noise at quantum gates for two qubits. Namely, the appearance of even the slightest noise at the quantum gate requires the implementation of a logical qubit using several physical qubits, that is, it requires the implementation of quantum error correction codes. Therefore, the practical limitation on the depth of a quantum circuit is determined by the number of physical qubits and their topological organization due to the implementation of quantum error correction codes, and the time in which the qubit maintains a stable state. As can be seen from report [9], last year (2025), no technology for the implementation of quantum computers has reached a mature technological level that allows the implementation of a noiseless quantum computer for practical application.

Another limitation in application appears due to the physical laws of quantum mechanics on which quantum computers are based. Namely, according to the measurement axiom of quantum mechanics, after measuring the state of the qubit, the system is set to a classical measured state, meaning that after executing a quantum circuit that describes the fault tree and performing measurements, we will get only one combination of basic events that lead or do not lead to a peak event. This property of quantum circuits results in the cut-off limit being determined by the number of times the circuit is executed on a quantum computer or simulation. For example, if a quantum circuit is performed n times on a quantum computer, we can expect with moderate probability (~37%) that a state for which the probability of occurrence is less than $\frac{1}{n}$ will not be measured. To some extent, we can approach this problem by applying the QAOA algorithm improved by Grover's search [2] or by developing a new approach based on the application of QAOA to solve the problem of the satisfiability of Boolean functions [10]. For the large fault tree models, such an approach is fundamentally not practical due to the complexity of searching for minimal cut-sets and the order of complexity of Grover's search algorithm. Namely, the complexity of Grover's search algorithm is determined by the square root of the number of minimal cut-sets for the fault tree, so the applicability of such an approach is limited to fault trees with several dozen basic events. However, the combination of executing a quantum circuit on a quantum computer and of classical determining and processing of the minimal cut-sets from the obtained statistics of the system state vectors opens up the possibility of processing larger models with limited accuracy. Indeed, the limited accuracy is inherently determined by the number of state vectors collected, and the results obtained by this approach are comparable to the approximate results obtained by the conventional approach for the same cut-off threshold. The BDD fault tree analysis is carried out on the entire set of minimal cut-sets where the accuracy of the obtained results is not questionable, but the BDD approach requires the determination of the order of the basic events [11] with which it is possible to create a BDD for a fault tree within the available resources. Due to all of the above, at this point we can safely say that currently available quantum computers are not technologically developed enough for a practical application of the approach presented in this paper. Thus, conventional and BDD approaches are certainly the first choice for a fault tree analysis. However, with the development and technological progress of quantum computers, in the future we can expect the application of solutions based on a combination of quantum and classical algorithms for a fault tree analysis.

5 SUMMARY AND CONCLUSION

In this paper, an approach based on the application of quantum circuits to perform a static fault tree analysis containing a voting gate has been demonstrated. The approach is based on the application of an algorithm for converting a voting gate into a record with a reduced number of AND/OR gates, followed by executing (or simulating) the corresponding quantum circuit. To illustrate the approach, an analysis has been carried out on the example of a coherent fault tree with six underlying events, four intermediate and one top event. Using the example of a fault tree, the voting gate was converted into an equivalent AND/OR record using the algorithm from [6]. The resulting fault tree record was converted into the corresponding quantum circuit by applying approach from [2]. A reduced number of AND/OR logic gates in the fault tree leads to a reduction in the number of additional qubits (quantum circuit width reduction) and a reduction in the number of Toffoli gates (quantum circuit depth reduction) for the implementation of multi-controlled X gates. For the quantum circuit, a simulation of multiple execution on a quantum computer was carried out and statistics for the relative frequencies of the measured states were collected. For the purpose of determining the set of minimal cut-sets from the obtained statistic, a novel algorithm was developed by which the minimal cut-sets were extracted that meet the condition of a probability higher than the cut-off value. The cut-off probability value is inherently determined by the number of times the circuit is executed on a quantum computer. From the obtained set of minimal cut-sets, a conservative upper bound for the probability of a top event has been determined, and it was noted that other reliability parameters can also be determined using the conventional approach.

There are a number of limitations and practical obstacles in this approach. The quantum hardware is still limited by noise, error rate, decoherence, and a limited number of qubits and limited connectivity. Quantum error correction and fault tolerance in quantum computing remain a major challenge. These hardware limitations constrain the size of the fault tree that we can meaningfully analyse with this approach. The encoding of a fault tree into a quantum circuit is a non-trivial decision. For example, it is necessary to make the right decision on how to correctly map basic events to qubits or logic gates to quantum sub-circuits. For large combinatorial problems such as searching for a set of minimal cut-sets, speeding-up is not necessarily exponential. Given all these pitfalls and problems, we might still question ourselves where and when can quantum fault trees be useful? One possibility is on very large, complex systems where conventional fault tree analysis becomes too expensive. For example, in cases where the determination of the minimal cut-sets is combinatorically demanding. Also, a quantum-assisted approach such as the one presented in this paper can be useful if we are looking for all minimal cut-sets up to a certain size. Also, in safety-critical areas where better risk and reliability modelling presents a real value. As quantum hardware matures, industries that want to be "ready for quantum advantage" could build procedures based on fault trees for which they can optionally offload computationally demanding parts to quantum solvers once they become available. In short – quantum fault trees are a legitimate new research topic that connects reliability engineering and quantum computing. They have yet to go mainstream, but they seem likely to be worth for further research, for example, in cases where a conventional approach based on approximate or BDD based calculations may be improved.

REFERENCES

[1] G.S.M. Silva, T. Parhizkar, E.L. Droguett, Quantum fault trees, *arXiv preprint arXiv:2204.10877*, 2022.

[2] G.S.M. Silva, E.L. Droguett, Quantum fault trees and minimal cut sets identification, *Reliability Engineering & System Safety*, DOI: https://doi.org/10.1016/j.ress.2025.111147., Vol. 262, 2025.

[3] R.E. Correll, A.O. Rodriguez, A.P. Ortiz (2020), Performing fault tree analysis on quantum computers (U.S. Patent No. 10,796,240). U.S. Patent and Trademark Office.

[4] P.A.M. Dirac, A new notation for quantum mechanics, Mathematical Proceedings of the Cambridge Philosophical Society, Vol. 35, pp. 416-418, 1939.

[5] P. Kaye, R. Laflamme, M. Mosca, An Introduction to Quantum Computing, Oxford University Press, Oxford, 2006.

[6] Wang J, Chen S, Wang F, Hu L. A new decomposition algorithm for complex voting gates processing in qualitative fault tree analysis. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. 2015;229(6):568-575. doi:10.1177/1748006X15594488

[7] J.B. Fussell, Fault tree analysis: concepts and techniques, American Society of Mechanical Engineers, USA, 1976.

[8] W.E. Vesely, F.F. Goldberg, N.H. Roberts, D.F. Haasl, Fault Tree Handbook, U.S. Nuclear Regulatory Commission, Washington D.C., 1981.

[9] Ruane, J., Kiesow, E., Galatsanos, J., Dukatz, C., Blomquist, E., Shukla, P., The Quantum Index Report 2025, MIT Initiative on the Digital Economy, Massachusetts Institute of Technology, Cambridge, MA, May 2025.

[10] Boulebnane, S., Montanero, A., Solving Boolean Satisfiability Problems with The Quantum Approximate Optimization Algorithm, *PRX Quantum*, American Physical Society, 2024, 5(3): 348-379. doi:10.1103/PRXQuantum.5.030348

[11] Banov R., Šimić Z., Grgić D. A new heuristic for the event ordering in binary decision diagram applied in fault tree analysis. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. 2020;234(2):397-406. doi:10.1177/1748006X19879305