2007

Search for Universal Ternary Quantum Gate Sets with Exact Minimum Costs

Marek Perkowski
Portland State University, marek.perkowski@pdx.edu

Normen Giesecke

Dong Hwa Kim

Sazzad Hossain

Search for Universal Ternary Quantum Gate Sets with Exact Minimum Costs

Normen Giesecke, Dong Hwa Kim*, Sazzad Hossain and Marek Perkowski,
Department of Electrical Engineering, Portland State University,
FAB 160-05, 1900 SW Fourth Avenue,
Portland, Oregon, USA, E-mail: **mperkows@ee.pdx.edu**
* *Dept. of Instrumentation and Control Engn., Hanbat National University,*
16-1 San Duckmyong-Dong Yuseong-Gu, Daejon, Korea, 305-719.E-mail:
kimdh@hanbat.ac.kr

Abstract

The choice of the best set of universal ternary gates for quantum circuits is an open problem. We create exact minimum cost ternary reversible gates with quantum multiplexers using the method of iterative deepening depth-first search (IDDFS) [25]. Such search is better for small problems than evolutionary algorithms or other search methods. Several new gates that are provably exact minimum cost have been discovered. These gates are next used as library building blocks in the minimization of larger ternary quantum circuits like highly testable GFSOP cascades [15,16] (that generalize ESOP) as well as the wave cascades [24] generalized to ternary logic. They are useful to design oracles for multi-valued algorithms such as Deutsch-Jozsa [26] and Grover.

1. Quantum Gates and Circuits built from cascaded ternary quantum Multiplexers

The research on designing best gates for ternary quantum circuits is very new and no efficient synthesis methods exist. To discover good sets of universal gates, several approaches were used, such as evolutionary algorithms, but they did not lead to satisfactory results. Practically, the evolutionary algorithms, even working for a long time, do not give warranty that the solution is exact minimum. We used also other well-known methods for solving combinatorial problems: a) simulated annealing, b) bacteria foraging, c) probabilistic generation, d) A* search and similar search algorithms, e) scattered search, f) Tabu Search, g) Particle Swarm Optimization (PSO) [23], genetic algorithms [22], but so far with no particular successes over methods presented here, so that this paper focuses on exhaustive search. Let us also observe, that the search for the best gates is performed only once in order to create a gate that is next used repeatedly in libraries.

We can allow thus our computer to spend much time, even days and weeks, to find the exact minimum solution. Exhaustive search [3,20] has been already used before in reversible logic design, but there are many ways how the exhaustive search can be organized, and they differ in time and memory usage. We investigated several types of exhaustive search strategies to particular quantum circuit structures. We found that for this kind of problems the A* algorithm known from AI operates very similarly to breadth first search. Our IDDFS search is similar but is easy to program and uses less memory, thus allowing to minimize larger circuits.

Ternary quantum macros – conceptual gates can be implemented using quantum multiplexers [21] as primitives, which themselves are composed from Muthukrishnan-Stroud (M-S) gates [2]. The quantum multiplexer concept [21] (called also the mux), used also by several other authors, is a convenient intermediate notation to synthesize both binary and multiple-valued (mv) quantum circuits. Therefore, this synthesis will be performed in terms of quantum multiplexers and their argument single qudit functions. (Qudits are quantum bits with radices higher than 2. Qutrit is a qudit used in ternary logic.) There exist several different more or less regular structures that describe how these gates can be cascaded. We will find exact minimum solutions to some well-known operators and also to new gates in order to form libraries of universal gates for mv quantum circuit synthesis. The exhaustive search creates the gate as a cascade starting from input signals of the function and next adds sequentially quantum mux after quantum mux to create the logic outputs of the cascade. The first practical goal of the exhaustive search approach proposed here is to find the realizations of all 2-qudit-

gates and determine their minimum costs and the best efficiencies. Efficiency can be defined in terms of how many ancilla qudits are used to realize a function. The optimally designed gates presented in this paper are next used as the building blocks of larger gates in systematic synthesis methods which are extensions and generalizations of the previous logic synthesis methods used in reversible and quantum circuits [1,3,4,6,12,13,14,15,16,17,19,24]. Alternately, one can use the exhaustive method to synthesize small circuits. Exhaustive search can be also used as a part of more sophisticated hybrid synthesizers [20]. The method searches exhaustively until the given circuit is found for which is next proven that within given constraints (like size and gate types) it is not possible to find a better realization of the given function *F*. In addition, in another variant, we used adaptations of exhaustive methods to find useful solutions with no assurance of circuit's minimality, but with taking into account important practical constraints such as a user-specified limited number of ancilla bits.

2. Quantum Ternary Gates and Structures

Figures 2.1, 2.4 and 2.8 show different cascade implementations with two by two ternary quantum multiplexers. The small boxes at the left of mux symbol taken from classical logic represent arbitrary single qutrit unitary operators, but in this paper these operators are in addition permutative. The quantum multiplexer operates as follows: depending on a value 0, 1, and 2 of the control qutrit, the respective input with number 0, 1 and 2, respectively (counted from top) is selected and sent to the output. Thus respective operator f_i is executed on the controlled qutrit. Fig. 2.1a is a cascade of two multiplexers where *A* is the controlling qutrit and is passed through without any change. *B* is the controlled data qutrit on which the functions are applied. In the case of the Fig. 2.1a, controlled qutrit *B* would be manipulated always in pairs of f_0 and f_3, f_1 and f_4 or f_2 and f_5. Therefore this implementation is not very useful, because the manipulations could be restricted to one multiplexer; e.g. assuming f_0 = +1 and f_3 = +1 could be realized by only using f_0 = +2. Operations +1 and +2 are implemented as cyclical shifts by 1 and by 2, respectively in one-qutrit operations. In Fig. 2.1b let us look at the first multiplexer. On the second multiplexer bit *A* is used as data input and bit *B* is the control input. Assume functions f_0, f_1, f_2, f_3, f_4 and f_5 to be defined as f_0 = +1, f_1 = 01, f_2 = +2, f_3 = 02, f_4 = +2, f_5 = 12. The circuit and the resulting ternary map are shown in Fig. 2.2. Operation 01 is a permutation of values 0 and 1 in single qutrit, operations 12 and 02 are implemented analogously. (These operations are realized internally by combinations of X, Y and Z rotation operators [27,7] in data inputs of M-S gates [2]). It should be noticed that the ratio of symbols "0", "1" and "2" in both Karnaugh maps are 1:1:1. If we would continue the cascade using this structure we would only get result with always the same amounts of symbols zero, one and two. Therefore this structure is also not sufficient to scale up to an arbitrary quantum circuit. The SWAP gate was proven to be in the minimum set of universal gates [11] and can be easily build from other sets of universal gates. It is the hypothetical crossing of wires that allows one to realize many small gates that are difficult to realize without it. A realization was shown in [9] and is represented in Fig. 2.3. Each intermediate function is shown by a ternary map with A as rows and B as columns. This gate has high importance in ternary quantum circuit synthesis which has been not yet recognized by the published synthesis papers. This gate is necessary for mapping to ion trap technology quantum circuits and other technologies with linear layout of qudits (with every qudit having at most two neighbors). We can explain this property on an example as follows.

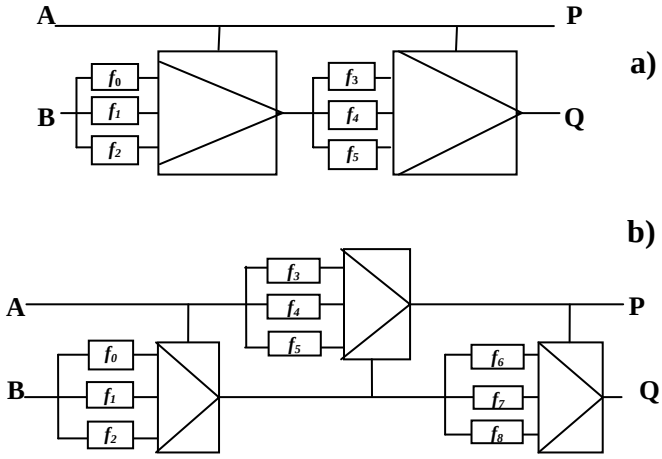
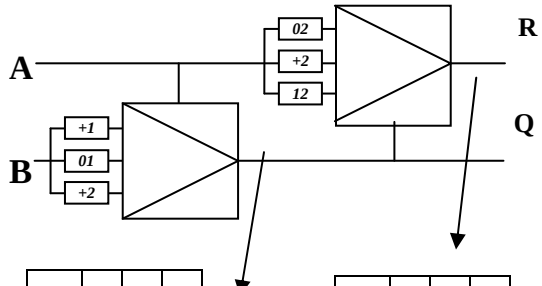


Fig. 2.1: First two structures based on cascaded quantum multiplexers

Assume that in a quantum array a wire in qudit 2 from top goes through Feynman gate in which qudit 1 controls qudit 3. This is not realizable in these technologies and requires two SWAP gates to be added between qudits 1 and 2, before and after the Feynman gate. Thus, instead of synthesizing without taking care of the no-crossing condition and next adding SWAP gates, another method can be invented where the no-crossing condition is build into the cost function and thus the number of crossing wires is reduced from the very design principle [23]. However,

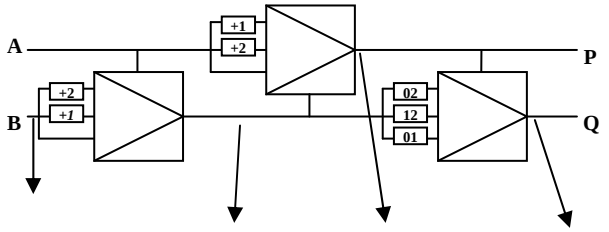
complete removal of SWAP gates is in general not possible. Various variants of SWAP gate (see below) are therefore used in our improved ternary MMD-like synthesizer, a successor to the program presented in [9].



A\B	0	1	2
0	1	2	0
1	1	0	2
2	2	0	1

A\B	0	1	2
0	2	0	2
1	0	1	2
2	1	0	1

Figure 2.2: Example of calculating intermediate ternary maps for part of the circuit from Fig. 2.1 b



A \B	0	1	2
0	0	1	2
1	0	1	2
2	0	1	2

B

A \B	0	1	2
0	2	0	1
1	1	2	0
2	0	1	2

B

A \B	0	1	2
0	0	1	2
1	0	1	2
2	0	1	2

A

A \B	0	1	2
0	0	0	0
1	1	1	1
2	2	2	2

B

Fig. 2.3. The structure of the Ternary SWAP gate

The next realization of a quantum circuit using the structure in Fig. 2.4a is more promising than the previous two from Fig. 2.1. Now we have two control qudits *A* and *B* and an additional static input (ancilla qudit) with the state |0⟩. The control qudits alternate in their application on the static input qudit. However, in this structure, constant ancilla inputs are introduced, and the number of ancilla qudits depends on the number of variables N. Ancilla qudits do not influence the logic of the output, but only facilitate the reversible implementation. These ancilla bits are, for the realization of reversible functions, unnecessary, and introduce extra qudit implementation overhead. The problems of minimizing the number of ancilla bits have been discussed in [19].

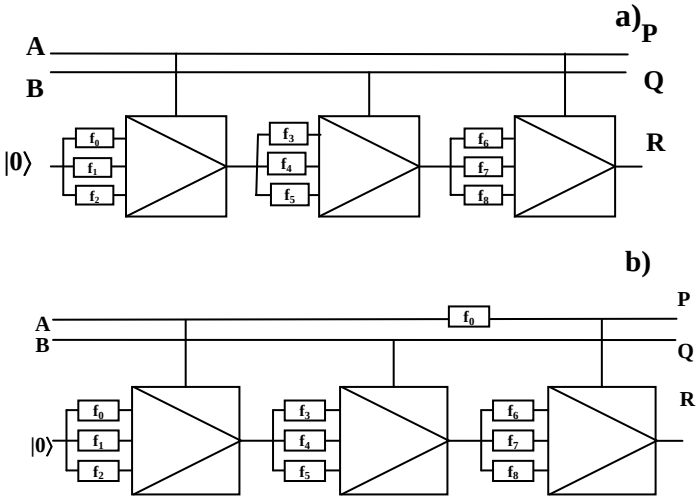


Figure 2.4: Second two structures based on cascaded quantum multiplexers.

The motivation for the approach presented here is the realization of an arbitrary logic function through a series of cascaded gates, with the goal of minimizing the number of ancilla lines, while introducing greater freedom in the number of necessary stages. The amount of ancilla lines can only be hypothesized and depends always on the function to be realized. Muthukrishnan and Stroud [2] formulated a relation between the amount of data qudits and number of ancilla qudits for M-S gates only. To reuse the ancilla bits one implements mirror gates. The implementation of mirror gates has the goal to restore the ancilla qudit to the initial state, 0, 1 or 2. The mirror gates use always the inverse operations. The Table 2.1 shows the corresponding inverse functions to the single-qutrit functions. Mirrors are used in mv wave cascades and quantum realizations of GFSOP with various kinds of Toffoli- and Feynman-like gates [15,16,17]. They allow to fold wires that start and end (thanks to mirror) with constants. Mirror quantum multiplexers X and Y have inverse singl-qudit functions f_i in all their data inputs.

Table 2.1: Inverse functions for each single qutrit function

single qudit functions		inverse single qudit functions
+1	-->	+2
+2	-->	+1
01	-->	01
02	-->	02
12	-->	12

An example of the structure shown in Fig. 2.4a by assuming data input = *0* is shown in Fig. 2.5.

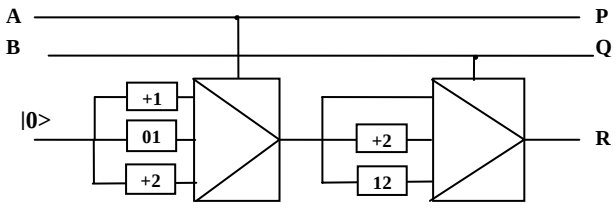


Figure 2.5.: Example of a promising implementation with ternary multiplexers

The circuit as in Fig. 2.5 is not limited to the ratio 1:1:1 of symbols zero, one and two anymore. Therefore, functions that are not balanced [8] can be also built. The maps of intermediate functions in Fig. 2.5 are not balanced. The last map has two*"0"*, four*"1"* and three *"2"*. This led to our conjecture that all combinations of the 2-qudits gates can be realized using this structure, given a finite number of multiplexers. The generalization of this structure to a given number (>2) of qudits and ancilla qudits means that it has a potential to feasibly realize all n-qudit functions. But the proof of the last assumption exceeds the scope of this paper. The reader is referred to [11,19] for more information.

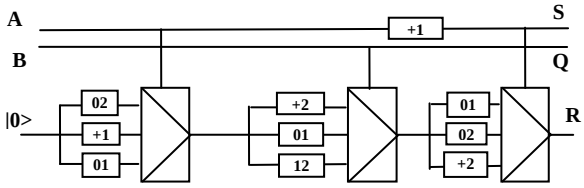


Figure 2.6: Example of Figure 2.4 b

At a closer look, the single-qudit-gate *+1* applied to input *A* shifts down the functions *01, 02* and *+2* by one at the last multiplexer in the circuit. The same result can be achieved by selecting the order *+2, 01* and *02.* Therefore the single qudit function, *+1* in this case, can be left out. The resulting circuit is in Fig. 2.7.

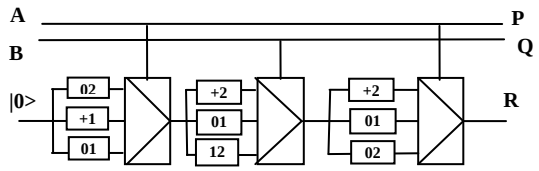


Figure 2.7: A realization of the example of Figure 2.4 b with fewer gates

The structure given in Fig. 2.4b contains only a little change to the structure of Fig. 2.4a. Now the control inputs and the static data input can be manipulated by applying single-qudit-functions like +1, +2, 01, 02 and 12. The application of these functions is only favorable when there are more than two multiplexers in the series (example in Fig. 2.6).

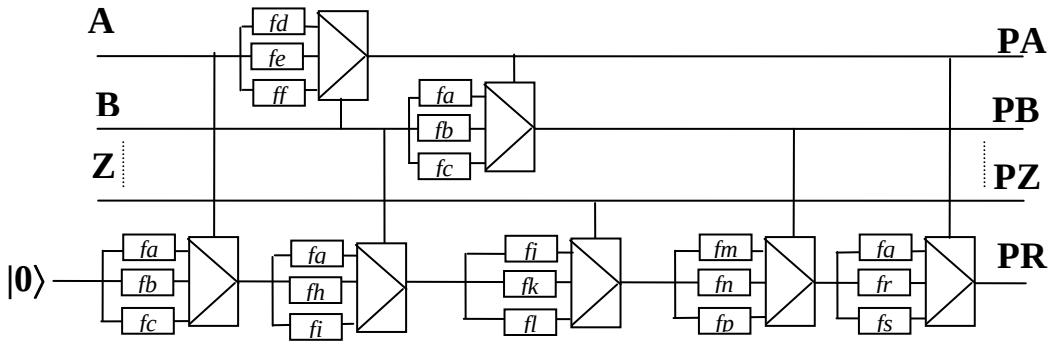


Figure 2.8: One more structure based on cascaded quantum multiplexers

The last type of circuit structure is introduced in Fig. 2.8. This structure is derived from the structure in Fig. 2.1b and is the most universal one out of the five general structures depicted in Figures 2.1, 2.4 and 2.8. The number of qudits is free to choose, as well as the number of ancilla qudits. Inputs *A* to *Z* are qudits that

control the multiplexer at the static qudit, and can also be manipulated by other qudits. The structures shown in Fig. 2.4a and Fig. 2.8 are the most desirable and capable to build quantum circuits based on quantum multiplexers. The drawback may be the need to employ additional ancilla qudits, but the ancilla bits allow to realize initially non-reversible functions and decrease the number of muxes so they provide a worthy design trade-off. In the two variable case it is possible to create every possible logic function with at most one constant-input $|0\rangle$, $|1\rangle$ or $|2\rangle$ [6]. To reuse the ancilla qudit, a mirror circuit with inverse operations is required. The structure in Fig. 2.4b is capable to create every two-qudit ternary gate. Building upon this, we can generalize the structure to use more than two-qudit circuit, e.g. to use three variables. The extension to three variables can be explained as shown in Fig. 2.9. The first ternary map is a two ternary qudit map with 9 ($=3^2$) cells presenting all combinations of two ternary variables. By converting it to three ternary variables the ternary map is extended to 27 ($=3^3$) cells. The contents of the first map with 9 cells is now the contents of the first column in the second map. One can see that if we want to change the cell containing 9, the function should only operate on *a=2* and *b=2*, and will impact only the values in the last row. These properties are utilized in the new heuristic ternary minimizer.

a\b	0	1	2
0	***1***	***2***	***3***
1	***4***	***5***	***6***
2	***7***	***8***	***9***

ab\c	0	1	2
00	***1***		
01	***2***		
02	***3***		
10	***4***		
11	***5***		
12	***6***		
20	***7***		
21	***8***		
22	***9***		

Fig. 2.9: Explanation of the convergence from two to three variables

3. Experimental Results of the search

The goal of the exhaustive Iterative Deepening Depth-First (IDDFS) search [25] was to find a set of the best realizations for 2-qudit-gates by using the multiplexer implementation shown in the previous section. As an example we present synthesis of cascades as in Fig. 3.1. The new interface [23] allows for a larger variety of segments and theoretically unlimited number of qudits and muxes. At the beginning, the number of multiplexers in series was set to one. For one multiplexer the algorithm found only 27 unique results. This is clearly not enough. This was, in fact, expected because only 216 = $(6)^3$ functions can be realized with one multiplexer.

The second search was made with two ternary multiplexers in series. There are 3 out of 6 possible operations on each multiplexer. This results in 216 ($=6^3$) combinations per multiplexer. By connecting 2 multiplexers in sequence, 46,656 ($=216^2$) results can be calculated. The exhaustive search algorithm permuted all operations (+0, +1, +2, 01, 12, 20) on two multiplexers. The control variables were chosen to be *"A"* for the first and *"B"* for the second multiplexer. The result showed that only 891 unique results were found.

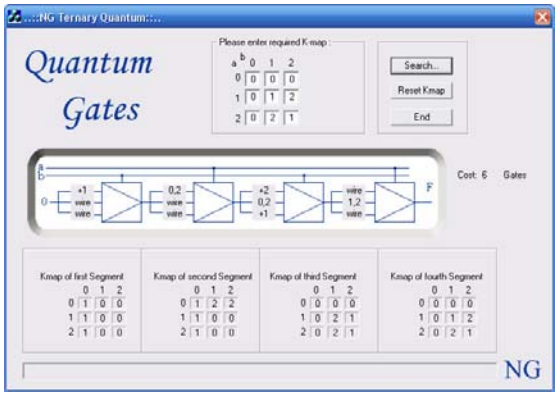


Figure 3.1: The Gate-Viewer Front End

The second bar in the diagram in Fig. 3.2 shows that it is possible to find only 891 out of 19,683 = 3^9 (all ways of placing three symbols in nine cells of ternary map) possible unique solutions. The detailed distribution can be found in [22]. To find more solutions, the number of multiplexers was set to three and the order of control variables was *"A","B" and "A"*. We will denote this order as ABA. A total number of 10,077,696 = 216^3 results could be calculated but it was not sufficient to generate all 19683 unique solutions. The total number of unique results found was 14,013.

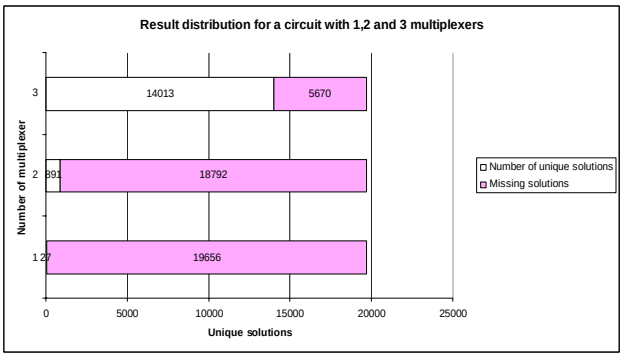


Fig. 3.2: Diagrams of unique solutions for 1, 2 and 3 multiplexers in series. Only 5670 functions cannot be realized with 3 multiplexers.

The third bar in the diagram in Fig. 3.2 reveals that a very large number of functions can be found using three multiplexers in sequence. But still realizations of 5670 functions are missing. Therefore the circuit was extended by another multiplexer. The structure was again chosen to be similar to Fig. 2.4a and the order of the control variables is of type *"ABAB"*. The structure in Fig. 2.4a with four multiplexers is capable to provide all unique solutions for any 2-qutrit ternary operator.

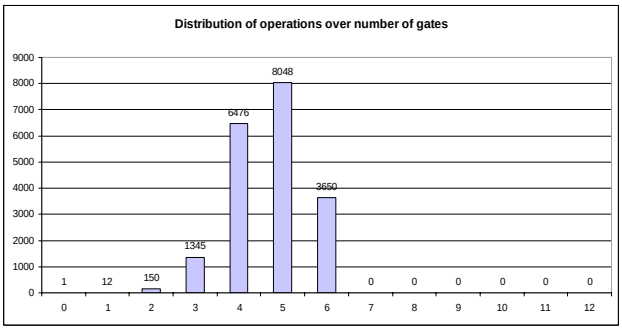


Fig. 3.3: Histogram distribution of operations (+1, +2, 01, 02 and 12) used to implement two-input ternary operator gates.

The detailed distributions of single-qubit operations for cascades with three and four multiplexers were analyzed in [22]. The total number of single-qudit-operations on all multiplexers was counted and the solution with the minimum amount of single qudit operations was saved. The maximum number of operations used was found to be 6 for all 2-qudit gates (Fig. 3.3). Only the operation functions (+1, +2, 01, 02 and 12) were taken into account. The "wire" or "+0" gate does not produce any cost. Observe that there were only 3650 functions with minimal solutions of 6 multiplexers. More statistical results are in [22].

4. The exact minimum ternary gates

Besides the statistics, the explorations made with the exhaustive algorithm are even more useful. The building blocks of a quantum circuit are quantum gates. Therefore, a few interesting gates to build a quantum circuit will be presented and their properties will be discussed here. Every quantum circuit is reversible and made of reversible quantum gates. A reversible circuit is one that has the same number of inputs and outputs and is a one-to-one mapping, or a so called onto function (bijection), between vectors of input and output values. Such circuits are operationally the same as "permutative" quantum circuits; every quantum circuit is reversible [14], and subclasses of quantum circuits are binary and ternary reversible circuits. Another general property of some quantum gates is universality [18], which means that in order to build an arbitrary quantum circuit, the used gates need to be universal. An example of a universal binary gate set are Toffoli, NOT and Controlled-NOT. We design their ternary counterparts. It was recently shown in [11] that the following ternary gates: SWAP, NOT and 1-Controlled-NOT are universal for the realization of arbitrary ternary n-qudit reversible circuits without ancilla qudits. This result is interesting as it is different than for binary case where there are no universal one-qubit and two-qubit sets of permutative gates and one has to resort to truly quantum gate primitives such as CV [7] to build universal gates such as the Toffoli gate. This result points also to the usefulness of the exact minimum realizations found here for ternary quantum synthesis.

Min, Max and 2-input ternary operations. In classical binary logic AND and OR are well known gates. As a standard, in multiple-valued logic domain AND is replaced by the MIN gate and OR is replaced by the MAX gate. The MIN gate is the arithmetic minimum and the MAX gate is the arithmetic maximum of integers being their arguments. When the respective maps were entered into the exhaustive search algorithm the results were that both quantum gates need 6 single-qudit operations. The algorithm found these solutions with the order "DCDC" of the control variables. An interesting fact to notice is that the order can also be switched to "CDCD" order and no changes are made on the single-qudit operations and the gate outputs the same result. This results from the symmetry of maps (order of arguments can be changed) and from the reversibility (see Figs 4.1, 4.2 and 4.3).

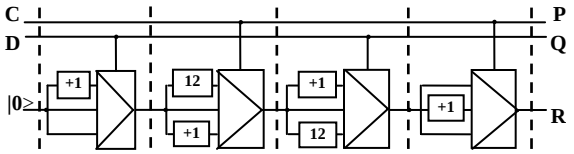


Fig. 4.1: Realization of the Ternary Min gate with control order of "DCDC"

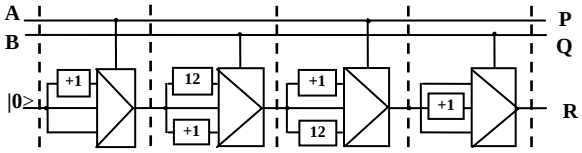


Fig. 4.2.: Realization of the Ternary Min gate with control order of "ABAB"

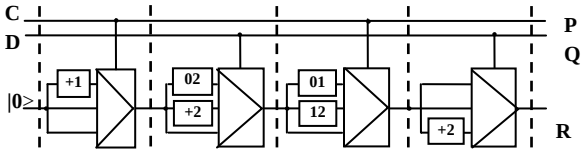


Fig. 4.3: Realization of the Ternary Max gate with control order of "CDCD"

Any unitary matrix represents a quantum gate. If a unitary matrix has only one *"1"* in every column and the remaining elements are *"0"*'s, then such a matrix is called a permutation matrix. The set of output vectors of such a permutation gate is simply certain permutation of the set of input vectors. A practical realization of mv Feynman gate was shown in [2]. The exhaustive algorithm found first the minimal realization based on the multiplexer structure for 2-qudits (+ 1 ancilla qudit) gates. The cost function for mux cascades was defined by the total number of single-qubit operations used. For the ternary Feynman realization only two multiplexers and 4 single-qudit-functions are needed and one ancilla qudit. Since the Feynman gate is a permutation gate, based on results from [11] the algorithm was used to search for a solution without any ancilla qudits. Fig. 4.4 illustrates such a solution. Instead of using 2 multiplexers and 4 single-qudit operations, only one mux and 2 single-qudit operations (*+1 and +2*) are required. The mux number is one and the cost is 2 operations. Interestingly, the number of muxes of the ternary Feynman gate, which is one, is the same as for binary Feynman gate. Some gates and their costs are summarized in Table 4.1.

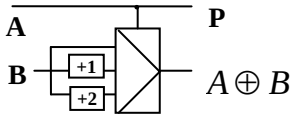


Fig. 4.4: Ternary Feynman gate realization using 1 quantum multiplexer and 2 single qudit operations

Table 4.1: Overview of Ternary Quantum gates realizations, number of multiplexers and their costs

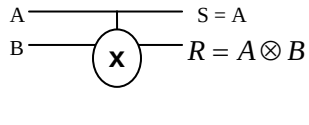
Gate	Multiplexer	Cost
Min	4	6
Max	4	6
Feynman	1	2
Galois multiplicati	4	6

Galois Gates. The ternary Feynman gate can be viewed as Galois Field 3 Addition whereas both inputs *A* and *B* are added up. Ternary Galois Field (TGF, or GF(3)) consists of the set of elements $T = \{0, 1, 2\}$ and two basic operations – **addition** (denoted by +) and **multiplication** (denoted by · or absence of any operator). Muthukrishnan and Stroud [2] found a relation between the number of qudits and the number of required ancilla qudits; this formula is based in the M-S-Gates, which are different from the universal multiplexer used in the present research, $r = [(n-2)/(d-2)] \quad (d > 2)$. The number of ancilla qudits is *r*, whereas *n* is the number of data qudits and *d* is the radix of the logic. For ternary logic *d* is 3. For example a system with 7 qudits needs 5 ancilla qudits to perform its logic function. In terms of the MIN/MAX architecture, the approach needed 1 ancilla qudit. The formula by M-S would result with r = 0, means no ancilla qudits are needed for a 2 qudit gate realization. The formula proposed by M-S is only valid for an n-qudit circuit using Muthukrishnan and Stroud gates. General formulae of how many ancilla qudits are needed are not known. This is always dependent on the functions that are being realized.

Efficient methods for representing and minimizing Ternary Galois Field Sum of Products (TGFSOP) expressions are very important. Such expressions can be either realized directly in quantum cascades or become a starting point of factorization processes leading to factorized cascades and wave cascades [24,15]. These methods are not a subject of this paper and can be studied for example in [16,4]. It should be

stressed however, that the results of this paper contribute the cheapest gates to be used in Galois Field Sum of Product (GFSOP) and factorized GFSOP cascades.

The other GF(3) operation is the Galois Multiplication. Where Galois addition replaces EXOR, the Galois multiplication can replace Boolean AND in the multivalued domain [9] and is therefore a very important approach to quantum circuit design but it is only one method of AND replacement and not a unique ternary gate that can be used for this task.



A\B	0	1	2
0	*0*	*0*	*0*
1	*0*	*1*	*2*
2	*0*	*2*	*1*

R

Fig. 4.5.: Galois multiplication; a) symbol. b) ternary map

If input $A = 2$ and $B = 2$ then the output R yields to R = 2 ⊗ 2 = 1 as the result of modulo 3 product. The realization of the Galois multiplication in our approach is in Fig. 4.6.

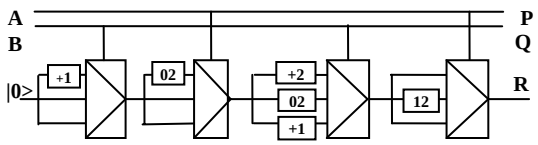


Fig. 4.6: Realization of Galois field multiplication

For the realization of the Galois Field multiplication operation 4 multiplexers with 6 single-qudit operations are needed.

SWAP gates. Another challenging quantum gate is the SWAP gate. It is not a single-output function like *min* and is thus more difficult to find. The 2-qudit SWAP gate exchanges two qudits. It has no counterpart in the classic electrical domain, because it is simply done by crossing wires within two layers of metalization. In the domain of quantum computing a "crossing" of wires is not possible. If the prototype of a SWAP gate for 2 qudits is developed, it can be used for circuit with *n*-qudits. The swapping is then just performed between 2 qudits within this circuit. A general approach to swap a given number of qudits might be interesting. There are, for example, 6 possible input/output combinations for a 3-qudit SWAP. Only 2 combinations are real 3-qudit SWAP gates and all swaps can be build with only 2 swap gates.

A new gate was discovered by our program, we call it the 2-qutrit Inverse SWAP gate. For the 2-qudit SWAP gate, the exhaustive search found a realization that requires 3-multiplexers and 7 single qudit operations (Fig. 2.3). Therefore we expected to find a solution using 3 multiplexers as well. The realization found by the algorithm is illustrated in Fig. 4.7. It is the minimum cost solution of the 2-qutrit Inverse SWAP gate. More realizations of the SWAP gate can be found in [22]. Khan *et al.* found a solution for the 2- qudit SWAP gate in [17], our solution made with the exhaustive search algorithm is the same and could not be improved. Interestingly, both in binary and ternary cases the SWAP gate has three muxes.

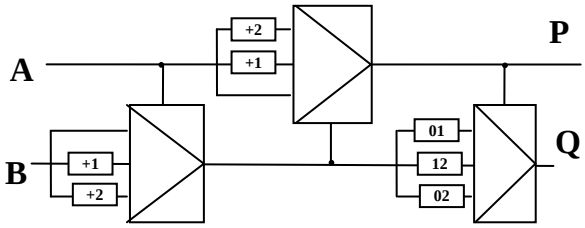


Fig. 4.7: Realization of the 2-qudit ternary Inverse SWAP gate using 3 multiplexers

We have found an efficient way to realize and implement all two-qudit ternary quantum gates by having at most one ancilla qudit. The ancilla qudit is the only drawback of this type of synthesis. On some functions that are not reversible the ancilla qudit is needed in any case, and thus should not be viewed as a weakness of our approach.

Toffoli Gates. One could derive from the structure of the Toffoli gate that might not require an ancilla qudit. With our algorithm an implementation of the Toffoli as Controlled-Controlled-NOT gate with 3 data qudits and without any ancilla qutrits is possible. Toffoli is known as universal [11,1] and is therefore another important gate. Yang *et al.* [11] show that SWAP, Not and Controlled-Not are universal for the realization of arbitrary ternary *n*-qudit reversible circuits without an ancilla qudit. From the Toffoli gate, which is a 2-Controlled-Not, it is possible to build up an n-qudit Controlled-Not. The Toffoli gate is a *controlled-controlled-NOT* gate and its diagrammatic representation is as in Fig. 4.8:

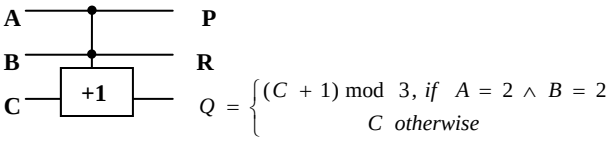


Fig.4.8: Symbol of Ternary Toffoli gate and its function

This gate flips the qudit *C,* if and only if the qudits *A* and *B* are both *"2"* and leaves *C* alone if they aren't. The controlling qudits "*A*" and "*B*" remain unchanged and are simply mapped to the output. The algorithm found 7128 ways to realize the Toffoli gate. One of the minimal solutions is shown in Fig. 4.9. The cost of this realization is 4; meaning only four single qudit operations are needed. Interestingly, in terms of the number of quantum multiplexers this gate needs only 4 muxes, while the binary Toffoli gate needs 2 Feynman gates, 2 Controlled-V and one Controlled-V^+ for a total of 5 muxes (Controlled-V is also called Controlled-Square-Root-of-NOT, Controlled-V^+ is its hermitian [7,27]). Another interesting Toffoli-like ternary gate is in Galois Logic [15,16,21] and uses Galois multiplication and Galois addition.

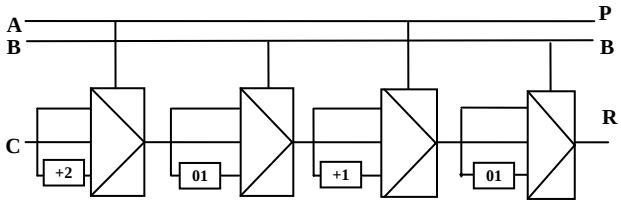


Fig. 4.9.: Ternary Toffoli (2-controlled-NOT) gate; minimal solution using quantum multiplexers

For the ternary 2-qudit logic, Kerntopf et al. [18] found that there are 1680 reversible functions. The rest (18,003) out of the 19683 functions are non reversible. This means that for the majority of the 2-qudit gates the realizations that use an additional ancilla qudit are needed. Our algorithm calculates and displays all possible 2-qudit quantum gates with minimum cost and can therefore be a foundation on which to build larger quantum circuits. The exhaustive algorithm produced a library containing all 2-qudit gates, their structure and the single qudit operations that are needed. Realization of ternary Controlled-NOT gate is shown in Fig. 4.10.

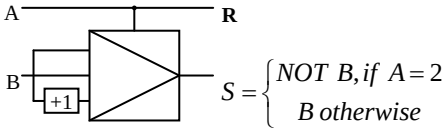


Figure 4.10: Realization of the Ternary Controlled-NOT gate

M.H.A. Khan and M. Perkowski proposed a ternary 3*3 Toffoli gate in [10]. Their realization needed 5 multiplexers and one ancilla qudit to perform the ternary Toffoli. The realization that was found by the exhaustive algorithm needs only 4 multiplexers and no ancilla qudit is needed. Thus garbage is avoided and the cost is minimized using this version of the Toffoli gate. The realization is illustrated in Fig. 4.9. This result alone shows that an exhaustive program can be truly useful in research as it found a better solution than that found by "hand and eye" method by two experienced researchers in the field and published in an international conference and next a journal. Denler et al. [19] found 12 ternary gates to be sufficient to build up every quantum gate. Those gates and all 5 single qudit gates where used by the GA.

The border line of this approach is the number of multiplexers. For a given quantum gate structure with 4 multiplexers the algorithm needs around eight hours to compute all different combinations ($216^4 = 2{,}176{,}782{,}336$) using a computer with a 2,4 Ghz CPU. Adding more and more multiplexers, the time just increases exponentially to compute the quantum gate. A five multiplexer quantum gate search needs around 4 days to compute only 6% of all solutions. This is not acceptable anymore. At this point the Genetic Algorithm, similar approaches, and probabilistic adaptations of our IDDFS search start to be used [22,23,25,7]. The drawback is only, that for these approaches there is neither a guarantee for a solution nor a guarantee for the minimal solution.

5. Conclusion and Future Research

Designing a cascade of reversible ("quantum permutative") gates is one of the fundamental problems in quantum computing, as such cascades are used in logic blocks in oracles of Grover algorithm, in arithmetic part of Shor algorithm and in other quantum circuits and algorithms. Various Ternary permutative gates and synthesis methods can be used to create such cascades. The choice of gate types and their realization with quantum-realizable primitives are thus of basic importance to multi-valued quantum logic synthesis

algorithms. In this paper exact minimum gates to be used in various types of ternary reversible cascades were found for the first time. Some statistics have been also found that will be useful in next algorithms creation.

We showed that exhaustive search makes it possible to generate all possible ternary two-variable output functions, using, at worst, one ancilla qudit and only four quantum multiplexers. Exact minimum solutions have been found in particular for the ternary MIN, MAX, Feynman, Galois addition and Galois multiplication, Toffoli, and swap gates, including the new Inverse Swap Gate. Many more such generalizations are possible [22,23]. We found two ternary generalizations of Feynman and two generalizations of Toffoli, both useful as building blocks for various types of quantum cascades. Using this method, the ternary Toffoli gate has been realized for the first time with 4 quantum muxes equipped with 4 single qudit operations. The program proved also that all 2-qudits gates can be realized within at most 4 quantum muxes and only one ancilla qudit. The method allows to investigate trade-offs between the number of gates and ancilla bits. For instance, a circuit without ancilla bits may be theoretically realizable but would likely be much longer than a circuit with one ancilla bit. This question is a difficult one and open to future research. For instance, in [11] we proved that ternary SWAP, NOT and 1-Controlled-NOT gates are universal for realization of arbitrary ternary *n*-qudit reversible circuits without ancilla qudits. We also demonstrated that all even ternary n-qudit reversible circuits can be constructed by ternary NOT and ternary 1-Controlled-NOT. Moreover the method is constructive, which means that it can be programmed to obtain the circuit for any number of qudits. However, the circuits according to [11] seem to be unnecessarily long. Our new method will allow to compare circuits with the minimum number of ancilla bits with those that have few more ancilla bits. Other approaches, not presented here for a lack of space were based on various evolutionary and Nature-mimicking paradigms [23,7]. Although these methods found several large circuits as well as circuits presented here, they were not able to find any new realization of a universal quantum gate of the smallest cost. For instance, none of the methods did deliver results for 3 qudit gates like the 3-qudit SWAP and ternary Fredkin gates yet. The iterative deepening depth-first search method is more practical for these tasks than biology-mimicking methods and its potential has been not yet recognized in quantum circuits community. It can be combined with A* search algorithm by adding a heuristic evaluation function, which is one of our current goals.

All the presented methods have been extended to quaternary logic. An interesting open problem is to extend them to arbitrary radix.

6. Acknowledgment.

The authors would like to thank referee A for very useful comments and ideas.

7. References

[1] **D. M. Miller, D. Maslov, and G. W. Dueck,** "Synthesis of Quantum Multiple-Valued Circuits", *J. MVL.*, Vol. 12, No. 5-6, 2006.

[2] **A. Muthukrishnan and C R. Stroud, Jr**., "Multivalued Logic Gates for Quantum Computation", *Physical Review A*, vol. 62, no. 5, 2000, pp. 052309/1-8

[3] **D. M. Miller, G. W. Dueck, and D. Maslov**, "A Synthesis Method for MVL Reversible Logic", *Proc. 34th ISMVL*, Toronto, Canada, 19-22 May 2004, pp. 74-80.

[4] **A. Al-Rabadi, and M.A. Perkowski**, "Multiple-Valued Galois Field S/D Trees for GFSOP Minimization and Their Complexity". *Proc. ISMVL 2001,* pp.159-166

[5] **U. Kalay, D. Hall, and M. Perkowski**,"A Minimal and Universal Test Set for Multiple-Valued Galois Field Sum-of-Products Circuits," *Proc. 7th ULSI*, pp. 50-51, Japan, May 1998

[6] **M.H.A Khan**, **M.A. Perkowski,** "Quantum Realization of Ternary Parallel Adder/Subtractor with Look-Ahead Carry", *Proc. RM,* Tokyo, 5-6 Sept. 2005, pp.15-22.

[7] **J. H. Bae, Ch. B. Bae, G. B. Lee, D. H. Kim, M.A. Perkowski, M.H.A. Khan** "Minimization of Ternary and Mixed Binary-Ternary Permutative Quantum Circuits". *Report PSU,* 2007.

[8] **E. Fredkin and T. Toffoli**, "Conservative logic", *Intern. J. Th. Physics*, 21, pp. 219-253, 1982.

[9] **E. Curtis, and M. Perkowski**, **"**A Transformation Based Algorithm for Ternary Reversible Logic Synthesis using Universally Controlled Ternary Gates", *Proc. IWLS* 2004, pp. 345 – 352.

[10] **M.H.A. Khan, and M. Perkowski,** "Quantum Realization of Ternary Encoder and Decoder," *Proc. RM.* 2005, pp. 23-27.

[11] **G. Yang, F. Xie, X. Song and M.A. Perkowski**: "Universality of two-qudit ternary reversible gates", *J. Phys. A: Math. Gen.*, 2006, Vol. **39,** pp. 7763-7773.

[12] **E. Dubrova, Y. Jiang, R. Brayton**, "Minimization of Multiple-Valued Functions in Post Algebra", *Proc. IWLS01,* pp. 132-138, June 2001.

[13] **A. N. Al-Rabadi**, "Novel methods for reversible logic synthesis and their application to quantum computing," *Ph.D. dissertation*, Electr. and Comp. Eng Dept, PSU, 2002

[14] **M. Lukac, M. Pivtoraiko, A. Mishchenko, and M. Perkowski**, "Automated Synthesis of Generalized Reversible Cascades using Genetic Algorithms", *Proc. 5th Boolean Problems*, pp. 33-45, Sept. 19-20 2002, Freiberg, Sachsen, Germany

[15] **M.H.A. Khan, M.A. Perkowski, and P. Kerntopf**, "Multi-Output Galois Field Sum of Products Synthesis with New Quantum Cascades", *Proc. 33rd ISMVL*, Tokyo, May 16-19, 2003, pp. 146-153.

[16] **M.H.A. Khan, M.A. Perkowski, M.R. Khan, and P.Kerntopf**: "Ternary GFSOP Minimization using Kronecker Decision Diagrams and Their Synthesis with Quantum Cascades"; *MVL Journal Special Issue.* Vol. 11. No. 5-6, 2005.

[17] **M.H.A. Khan and M. Perkowski, "**Evolutionary Algorithm Based Synthesis of Multi-Output Ternary Functions Using Quantum Cascade of Generalized Ternary Gates", *submitted to.MVL J.*

[18] **P. Kerntopf, M. Perkowski and M.H.A. Khan**, "On Universality of General Reversible Multiple-Valued Logic Gates" Proceedings of ISMVL 2004, pp. 68-73.

[19] **N. Denler, B. Yen, M. Perkowski, and P. Kerntopf,** "Minimization of Arbitrary Functions in a New Type of Reversible Cascade built from Quantum-Realizable "Generalized Multi-Valued Gates" , *Proc. IWLS 2004.* pp. 321 – 328.

[20] **M. Lukac, M. Perkowski,** "Combining Evolutionary and Exhaustive Search to Find the Least Expensive Quantum Circuits", *Proc. ULSI*, May, 2005.

[21] **M. Perkowski, A. Al-Rabadi, P. Kerntopf,** *"Multiple-Valued Quantum Logic Synthesis,"* Proc. Conf. on New Directions in VLSI Design, *Sendai, Japan, December 2002.*

[22] **N. Giesecke,** *Ternary Quantum Logic*, M.Sc. Thesis, PSU, 2006.

[23] **S. Hossain,** Ph.D. Thesis in preparation, PSU.

[24] **A. Mishchenko**, **M. Perkowski,** "Logic Synthesis for Reversible Wave Cascades", *Proc. IWLS 2002*, June 4-7, 2002.

[25] **W. Zhang,** *State Space Search. Algorithms, Complexity, Extensions, and Applications*, Springer, 1999.

[26] **Y. Fan, "**A Generalization of the Deutsch-Jozsa Algorithm to Multi-Valued Quantum Logic," *Proc. ISMVL* 2007.

[27] **M. Nielsen** and **I. Chuang**, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.