Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 93 (2016) 153 – 160

Procedia Computer Science

6th International Conference On Advances In Computing & Communications, ICACC 2016, 6-8 September 2016, Cochin, India

Realization Of Ternary Reversible Circuits Using Improved Gate Library

P. Mercy Nesa Rani[a,*], Abhoy Kole[b], Kamalika Datta[a], Alok Chakrabarty[a]

[a]*Department of computer Science& Engineering, National Institute of Technology, Shillong 793003, India*
[b]*Department of Computer Application, B. P. Poddar Institute of Management & Technology, Kolkata 700052, India*

Abstract

Ternary logic has some distinct advantage over binary logic. In this paper we propose a synthesis approach for ternary reversible circuits using ternary reversible gates. Our method takes a boolean function as input. The input is provided as .pla file. The .pla file is first converted into ternary logic function, which can be represented as permutation. The gate library used for synthesis is Ternary Not, Ternary Toffoli and Ternary Toffoli$^+$ (N_T,T_T,$T_T{}^+$). The proposed constructive method, generates 3-cycles from the permutation, and then each 3-cycle is mapped to (N_T,T_T,$T_T{}^+$) gate library. Experimental results show that the method generates lesser number of gates for some circuits compared to previously reported works.

Peer-review under responsibility of the Organizing Committee of ICACC 2016

Keywords: Ternary Reversible Gate; (N_T,T_T,$T_T{}^+$) gate library; Quantum Computing; Qutrit;3-cycles permutation.

1. Introduction

Multi-valued logic (MVL) has got several advantages over the binary logic in quantum computation *viz.,* better security for quantum cryptography and more power for quantum information processing. Among the MVL realization of quantum circuits, three-valued logic (ternary logic) plays an important role and is termed as qutrit (quantum ternary digit). The amount of information that can be stored per bit for ternary logic is more as compared to the binary logic. It is expected that qutrit-based quantum information processing will be more powerful than qubit (two-valued) implementation.[13] As a result, many researchers have proposed ternary quantum logic synthesis approaches.[8 5 12] There are two approaches for ternary reversible circuit synthesis which includes group theory based approach and Genetic Algorithm (GA) based approach.There exists some group theoretic approaches in literature for synthesis of ternary reversible circuits[18 15 17] which synthesizes the circuits using different gate libraries. Li et al.[10] proposed the synthesis of ternary non-reversible circuits using the ternary Swap, Not and Toffoli (SNT) gates. They converted the non-reversible ternary logic circuits into ternary reversible logic circuits and obtained 3-cycles from the given boolean function which are decomposed into the product of neighbouring 3-cycles. Then the neighbouring 3-cycles

* P.Mercy Nesa Rani, Tel.: 91-0364-2501294 Fax.: 91-0364-2501113
E-mail address: mercyrani@nitm.ac.in

Peer-review under responsibility of the Organizing Committee of ICACC 2016
doi:10.1016/j.procs.2016.07.195

are synthesized using SNT gate library which increases the gate count to synthesize the circuits but reduces the number of ancilla input and garbage output bits. GA based synthesis of ternary reversible circuits have been reported in[2 11 9] that has a large search space for solving problems in reversible circuits and the successful selection of fitness function is very important to achieve the convergence for search solution.
In this paper, we develop a group theory based ternary reversible synthesis method, which require the ternary Toffoli, ternary toffoli$^+$ and ternary Not gates to synthesize the ternary reversible circuits. The use of (N_T,T_T,$T_T{}^+$) gate library helps in the reduction of gate count as compared to existing work[10]. The rest of the paper is organized as follows: Section II presents the basic concepts of ternary reversible gates. The proposed approach with an example to realize the ternary reversible circuits are given in section III. Section IV summarizes the experimental results for some benchmarks followed by conclusion in section V.

2. Basic Concepts and Ternary Reversible Gates

In this section we discuss about ternary reversible gates and circuits. In a general form a ternary reversible gate can be defined in the following way:

Definition 1. *A ternary reversible gate with n-inputs represent a bijection of:* $T^n \rightarrow T^n$ *where T denote the set of ternary logic values* $\{0, 1, 2\}$.

In synthesizing the circuits, gates are added to the circuit in order to realize the desired functionality. Like binary reversible circuit, a ternary reversible circuit is defined as:

Definition 2. *A ternary reversible circuit represents a cascade of ternary reversible gates, i.e.,*

$$C = T_1 T_2 \dots T_{\mathfrak{R}} = \bigcap_{i=1}^{\mathfrak{R}} T_i$$

without any feedback or fanout.

The outputs of a ternary reversible function is a *permutation* of its inputs.

Definition 3. *A permutation on* $M = \{d_1, d_2, \dots, d_n\}$ *is a bijection of M onto itself,* $M \rightarrow M$.

The set of all permutations on inputs, forms a group under composition of *mapping*, called symmetric group S_k[3]. A permutation group is simply a subgroup of a symmetric group. A mapping $S : M \rightarrow M$ can be written as a product of disjoint cycles.

Example 1. *For the inputs* $\{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9\}$ *one possible mapping is*

$$mapping = \begin{pmatrix} d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 & d_8 & d_9 \\ d_1 & d_4 & d_7 & d_2 & d_5 & d_8 & d_3 & d_6 & d_9 \end{pmatrix}$$

which can also be written as a composition *of* (d_2, d_4), (d_3, d_7) *and* (d_6, d_8).

Definition 4. *For a given group of symbols* $d_1, d_2, \dots, d_n \in S_n$ *a mapping* $d_{i1} \rightarrow d_{i2} \cdots \rightarrow d_{ik} \rightarrow d_{i1}$, *where* $k \leq n$ *and* $1 \leq i1, i2, \dots, ik \leq n$ *is called a k-cycle denoted as* $(d_{i1}, d_{i2}, \dots, d_{ik})$.

In realizing the reversible circuits, Toffoli gate is used extensively in binary domain. In ternary domain the funtionality of the gate is extended and the gate is termed as Ternary Toffoli gate.

Definition 5. *A ternary Toffoli gate* $T(\{B_2, B_3\}; B_1)$ *is defined such that if the state of control qutrits* $B_2, B_3 \in \{1, 2\}$ *and* $B_2 = B_3$, *then the state of target qutrit* B_1 *is realized as* $P_1 = B_1 \oplus_3 1$, *where* $\oplus_3$ *stands for addition modulo 3; otherwise,* $P_1 = B_1$, *whereas* $P_i = B_i$, *for* $i \neq 1$.

In other words, (B_1, B_2, B_3) maps to $(B_1 \oplus 1, B_2, B_3)$ using modulo 3 addition when $B_2, B_3 \in \{1, 2\}$ and $B_2 = B_3$. Fig. 1 shows a Ternary Toffoli gate and its corresponding truth table is presented in Table 1 where $X \in \{0, 1, 2\}$.

The operation of a ternary reversible gate can be realized using a cascade of elementary multi-valued M-S gates that can be implemented using ion-trap technology[14].

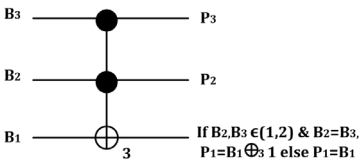

Fig. 1: Ternary Toffoli gate

Table 1: Truth table of Ternary Toffoli Gate

B_3	B_2	B_1	P_3	P_2	P_1
0	0	X	0	0	X
0	1	X	0	1	X
0	2	X	0	2	X
1	0	X	1	0	X
1	1	0	1	1	1
1	1	1	1	1	2
1	1	2	1	1	0
1	2	X	1	2	X
2	0	X	2	0	X
2	1	X	2	1	X
2	2	0	2	2	1
2	2	1	2	2	2
2	2	2	2	2	0

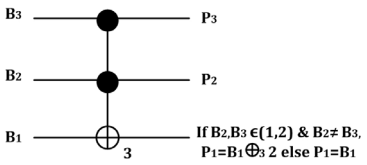

Fig. 2: Ternary Toffoli$^+$ gate

Table 2: Truth table of Ternary Toffoli$^+$ Gate

B_3	B_2	B_1	P_3	P_2	P_1
0	0	X	0	0	X
0	1	X	0	1	X
0	2	X	0	2	X
1	0	X	1	0	X
1	1	X	1	1	X
1	2	0	1	2	2
1	2	1	1	2	0
1	2	2	1	2	1
2	0	X	2	0	X
2	1	0	2	1	2
2	1	1	2	1	0
2	1	2	2	1	1
2	2	X	2	2	X

Definition 6. *Muthukrishnan-Stroud (M-S) gate $G(A;B)$ is defined as the input value A controls the output value Q, which is the Z-transformation of input B whenever $A = 2$, where $Z = \{+1, +2, 12, 01, 02\}$.*

The M-S gate is considered to be an elementary quantum building block which has a cost of 1[7]. Fig. 3 shows the graphical representation of a M-S gate. The operation of a M-S gate can be realized by Eqn. (1).

$$\text{M-S}(A,B) = \begin{cases} B\ Shift\ by\ Z & \text{if } A = 2 \\ B, & \text{otherwise} \end{cases} \tag{1}$$

where $Z = \{+1, +2, 01, 02, 12\}$. The Z-transformation is also presented in Table 3.

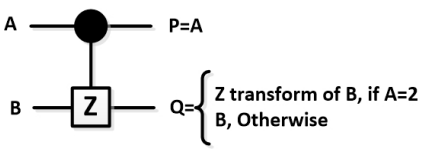

Fig. 3: M-S gate

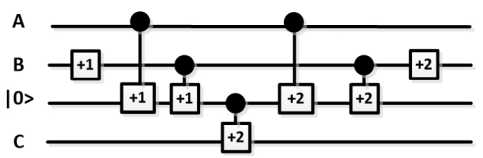

Fig. 4: Illustrative ternary Toffoli$^+$ gate realization using M-S gate

The quantum realization of ternary Toffoli gate requires 5 M-S gates operating in a sequence when both control qutrits are in state 2 [6] and the realization requires 4 additional 1-qutrit elementary gates when both control qutrits are in state 1 are as shown in Fig. 5.

An extension to Ternary Toffoli gate, Ternary Toffoli$^+$ is also considered in the synthesis process.

Table 3: Truth Table of Z transformation

Input B	Output Q				
	$Z(+1)$	$Z(+2)$	$Z(12)$	$Z(01)$	$Z(02)$
0	1	2	0	1	2
1	2	0	2	0	1
2	0	1	1	2	0

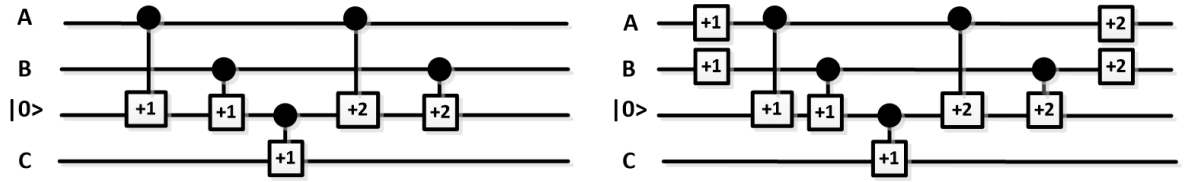

Fig. 5: (a) Realization with both controls in state 2 ; (b) Realization with both controls in state 1

Fig. 6: (a) Illustration of a ternary Not gate; (b) Equivalent elementary gate realization

Definition 7. *A ternary Toffoli$^+$ gate $T(\{B_2, B_3\}; B_1)$ is defined such that if the state of control qutrits $B_2, B_3 \in \{1, 2\}$ and $B_2 \neq B_3$, then the state of target qutrit B_1 is realized as $P_1 = B_1 \oplus_3 2$, where $\oplus_3$ stands for addition modulo 3; otherwise, $P_1 = B_1$, whereas $P_i = B_i$, for $i \neq 1$.*

Fig. 2 shows a ternary Toffoli$^+$ gate and its corresponding truth table is presented in Table 2 where $X \in \{0, 1, 2\}$. The M-S gate realization cost of a Ternary Toffoli$^+$ gate is 7. Fig. 4 shows the quantum implementaton of a ternary Toffoli$^+$ gate operating on target qutrit C when control qutrits A and B are in state 2 and 1 respectively.
In order to make the library universal, a Ternary NOT gate is also included in the library.

Definition 8. *A ternary NOT gate $T(\emptyset; B_i)$ is defined as $P_i = B_i \oplus_3 1$, where $\oplus_3$ stands for addition modulo 3.*

The pictorial representation of ternary NOT gate and its equivalent quantum implementation using ternary shift gate is shown in Fig. 6.

Table 4 summarizes the working of all the reversible gates from the library, termed as (N_T,T_T,$T_T{}^+$) that is used in proposed synthesis approach.

Table 4: Working of Ternary Reversible Gates

Toffoli		Toffoli$^+$		NOT
Controls	Change in Target	Controls	Change in Target	Change in Target
1 1	$0 \rightarrow 1$	1 2	$0 \rightarrow 2$	$0 \rightarrow 1$
2 2	$1 \rightarrow 2$	2 1	$2 \rightarrow 1$	$1 \rightarrow 2$
	$2 \rightarrow 0$		$1 \rightarrow 0$	$2 \rightarrow 0$

Table 5: Realization of all 3-cycles

3-cycles	$\#N$	$\#C_i11$	$\#C_i22$	$\#C_i12$
(3, 10, 5)	3	7	0	3
(4, 11, 8)	2	6	3	6
Total	5	13	3	9

The decimal encoding for the qutrit is different from that of qubit. The ternary value *abc* in decimal encoding is represented as $9a + 3b + c$ (i.e., $3^2 \times a + 3^1 \times b + 3^0 \times c$), e.g., the decimal value 3 in qutrit representation is 010.

3. Proposed Approach

The synthesis method takes a *.pla* file as input, performs required ternary transformation, generate all the 3-cycles and use the (N_T,T_T,$T_T{}^+$) library for synthesis. The schematic diagram for ternary reversible circuit synthesis is shown in Fig. 7. The *.pla* files in binary representation are taken from RevLib[16]. These *.pla* files are converted into

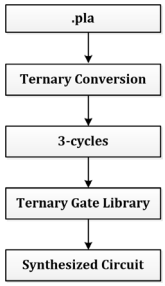

Fig. 7: Schematic Diagram for Ternary Reversible Circuit Synthesis

ternary representation by necessary transformation and the 3-cycles are generated from the ternary representation. The transformation approach is presented as below:

Step 1: Input is a .pla file.

Step 2: Make the number of input lines in the binary representation is equal to the number of input lines in ternary representation.

Step 3: If the output lines in the truth table is repeated for $\leq 3^n$ times, n garbage output lines are to be added in the output.

Step 4: Accordingly the number of input lines are added in ternary representation to make the truth table reversible.

Step 5: Apply the don't care assignment to map the truth table.

Step 6: Derive the cycles from the permutation.

Step 7: Decompose the derived larger cycles into 2-cycles and 3-cycles using the group theoretic rules[1].

Step 8: Even number of 2-cycles are combined together to form 3- cycles.

Step 9: Return the 3-cycles generated.

Then the 3-cycles are synthesized using the proposed algorithm with the help of the (N_T,T_T,$T_T{}^+$) library. The synthesis process started by transforming the matrix representing a 3-cycle. Initially, matrix generated from the 3-cycle is reduced to 3 column matrix by applying gates from the (N_T,T_T,$T_T{}^+$) library. Then the transformation of this 3 column matrix is carried out by applying gates from the (N_T,T_T,$T_T{}^+$) library. The complete synthesis approach is presented in the form of Algorithm 1 depicted below.

The complete synthesis process is illustrated by the following example.

Example 2. *Consider a reversible function, which has been specified using the following input-output permutation:*

$$f_r = (0, 1, 2, 10, 11, 3, 6, 7, 4, 9, 5, 8, 12, 13, 14, 15,$$
$$16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26).$$

Algorithm 1: Generation of Ternary 3-cycles

input : 3-cycles generated from transformation
output: Ternary reversible circuit

1 Generate matrix for each 3-cycle;
2 If the matrix has more than 3 columns;
3 Apply ternary gates if required to make the elements of a column identical;
4 Reduce the size of the matrix by eliminating column with identical element;
5 Repeat steps 2-4 until matrix is reduced to 3 columns matrix;
6 Check the value v_i (= 1 or 2) of the first two column that appears maximum number of times;
7 Apply ternary gates to set the values of the first two column of the matrix v_i;
8 Apply ternary gates if necessary to make the values of the last column as 0, 1 and 2 irrespective of their order;
9 Apply ternary Toffoli gate $T(\{c_1, c_2\}; t)$ with controls on the first two columns ($c_1 = c_2 = v_i$) and target t on the 3rd column;
10 Apply all the gates in the reverse order except the last one (applied in step 9);
11 Apply two identical gates if one such gate was applied earlier;
12 Otherwise apply one gate for each two previously applied identical gate;
13 Repeat steps 1-12 for all the 3-cycles;
14 Return the generated cascade of ternary reversible gates as the resulting circuit;

That is, input vector 0 *maps to output vector* 0, 1 *maps to* 1, 2 *maps to* 2 *etc. The given permutation can be expressed as a product of the following 3 cycles.*

a. $(3, 10, 5)$
b. $(4, 11, 8)$

i. For the 3*-cycle* $(3, 10, 5)$ *the matrix representation with the decimal encoding of the qutrit is*

$$\begin{pmatrix} 0,1,0 \\ 1,0,1 \\ 0,1,2 \end{pmatrix}$$

ii. The number of $1's$ *is greater than number of* $2's$ *in the matrix. So use Ternary Gates from* (N_T,T_T,$T_T{}^+$) *library to make the first and second column of the matrix as 1, i.e.,*

$$\begin{pmatrix} 0,1,0 \\ 1,0,1 \\ 0,1,2 \end{pmatrix} \xrightarrow{C_2 11} \begin{pmatrix} 0,1,0 \\ 1,1,1 \\ 0,1,2 \end{pmatrix} \xrightarrow{C_1^2 12} \begin{pmatrix} 0,1,0 \\ 1,1,1 \\ 1,1,2 \end{pmatrix} \xrightarrow{N_3} \begin{pmatrix} 0,1,1 \\ 1,1,2 \\ 1,1,0 \end{pmatrix} \xrightarrow{C_1 11} \begin{pmatrix} 1,1,1 \\ 1,1,2 \\ 1,1,0 \end{pmatrix}$$

Here matrix columns are indexed from left to right as 1-3. *The application of ternary Toffoli/Toffoli*$^+$ *gate is denoted by* $C_t^r S_i S_j$, *where t represents the target column index, r represents number of times the gate is applied, and* S_i *and* S_j, $i < j$ *represent the states of control qutrits on i-th and j-th columns. Similarly, the application of ternary NOT gate is denoted by* N_t^r, *where t represents the target column index, and r represents number of times the gate is applied.*

iii. Now the third column contains three different values. So rotate the third column values only once to get the matrix as follows using the Ternary Toffoli gate $C_3 11$.

$$\begin{pmatrix} 1,1,2 \\ 1,1,0 \\ 1,1,1 \end{pmatrix}$$

iv. Finally reverse all the gates applied from the step ii. to restore the states of qutrits as follows.

$$\begin{pmatrix}1,1,2\\1,1,0\\1,1,1\end{pmatrix} \xrightarrow{C_1^2 11} \begin{pmatrix}1,1,2\\1,1,0\\0,1,1\end{pmatrix} \xrightarrow{N_3^2} \begin{pmatrix}1,1,1\\1,1,2\\0,1,0\end{pmatrix} \xrightarrow{C_1 12} \begin{pmatrix}1,1,1\\0,1,2\\0,1,0\end{pmatrix} \xrightarrow{C_2^2 11} \begin{pmatrix}1,0,1\\0,1,2\\0,1,0\end{pmatrix}$$

Similarly, synthesis for the remaining 3*-cycle,* $(4,11,8)$ *is carried out by going through the steps i.-iv. The realization for all the* 3*-cycles are given in Table 5.*

4. Experimental Results

The method has been implemented using python and run on an Intel i3 based processor with 4 GB memory running Ubuntu 12.04. In order to analyze the efficiency and effectiveness of the proposed cycle-based synthesis method using

Table 6: Synthesis Results for Benchmark

Benchmark	l	N	T_{11}	T_{22}	T_{12}	n	qc
4_49_7	3	39	39	0	24	4	558
4gt11_23	5	18	23	8	18	6	391
4gt13_25	4	9	17	0	21	5	309
4gt5_21	6	30	88	24	84	7	1530
4mod5_8	5	18	17	14	24	6	409
5xp1_90	10	648	1208	595	1212	11	22979
alu_9	7	78	138	30	102	8	2184
graycode6_11	5	105	205	117	210	6	4005
half_adder	3	18	27	19	24	4	524
ham7_29	6	246	477	266	552	7	9733
max46_177	12	603	961	160	588	13	14168
mini-alu_84	5	60	111	42	81	6	1836
mod5adder_66	5	60	75	82	102	6	1859
mod5d1_16	4	42	16	110	69	5	1219
mod5d2_17	4	87	99	55	99	5	1946
one-two-three_27	3	24	27	26	24	4	565
radd_193	13	1296	2760	948	3036	14	52128
rd32_19	3	27	30	32	30	4	667
rd73_69	12	654	1101	533	1251	13	21985
root_197	12	1200	2543	957	2610	13	47142
sqn_203	10	384	767	212	762	11	13681
sqrt8_205	13	1599	3128	961	2586	14	52658

Table 7: Comparison of Implementation

Ternary Half Adder	Gate library	#ancilla	#garbage	#gates
[12]	$T_F, TC^2NOTGate, T_T$	2	2	4
[10]	N_T, T_T	1	1	122
Our	N_T, T_T, T_T^+	1	1	88

gates from ternary (N_T,T_T,$T_T{}^+$) library, we run the proposed method using benchmarks and presented the results in Table 6. The benchmarks are realized for the first time except ternary half adder and therefore, cannot be compared with other results. The first two columns represent name of the benchmarks and number of qutrits (l) required to realize the benchmark in ternary reversible domain. The next four columns shows in order the number of ternary NOT gates (N), ternary Toffoli gates with all controls in state 1(T_{11}), ternary Toffoli gates with all two controls in state 2 (T_{22}), and ternary Toffoli$^+$ gates with controls one in state 1 and another in state 2 (T_{12}/T_{21}) respectively. The last two columns represent number of qutrits to realize the benchmark when decomposed using M-S[4,14] gates and total number of M-S gates (i.e., $N*1+T_{11}*9+T_{22}*5+T_{12}*7$ gates where 1, 9, 5 and 7 represent the number of M-S gates required to realize the operation of N, T_{11}, T_{22} and T_{12} gate respectively) that are required for an equivalent quantum realization at logical level. To show improvements from running our method using (N_T,T_T,$T_T{}^+$) library on ternary half adder over previously reported results we have also presented our result in Table 7. As can be seen the implementation of ternary half adder require one less ancilla/garbage line than[12] and we have achieved 27% improvements over[10].

5. Conclusion

In this paper a ternary reversible logic synthesis approach is proposed using (N_T,T_T,$T_T{}^+$) gate library. Experimental evaluation shows that for some circuits our result performs better compared to existing ones. Results are reported for upto n=13 input size. This method is restricted to the use of 3 qutrit ternary reversible gates which in turn increases the gate count to synthesize the benchmarks. The introduction of n-bit ternary Toffoli gate and 3-cycles optimization are being investigated to reduce the gate count in benchmarks realization.

References

1. K. Datta, I. Sengupta, and H. Rahaman. Group theory based reversible logic synthesis. In *5th International Conference on Computers and Devices for Communication (CODEC-2012)*, pages 365–374. IEEE, 2012.
2. V. G. Deibuk and A. V. Biloshytskyi. Design of a ternary reversible/quantum adder using genetic algorithm. *I.J. Information Technology and Computer Science*, 09:38–45, 2015.
3. J. D. Dixon and B. Mortimer. *Permutation Groups*. Springer. New York, 1996.
4. M. M. Hasan. A low-cost realization of quantum ternary adder using muthukrishnan-stroud gate. In *Proceedings of International Conference on Electrical and Computer Engineering(ICECE 2008)*, pages 732–734. IEEE, 2008.
5. M. H. Khan and M. A. Perkowski. Quantum ternary parallel adder/subtractor with partially-look-ahead carry. *Journal of Systems Architecture*, 53:453464, 2007.
6. M. H. A. Khan. Design of reversible/quantum ternary multiplexer and demultiplexer. *Engineering letters*, 13(2):65–69, 2006.
7. M. H. A. Khan. Quantum realization of multiple-valued feynman and toffoli gates without ancilla input. In *Multiple-Valued Logic, 2009. ISMVL '09. 39th International Symposium on*, pages 103–108, May 2009.
8. M. H. A. Khan, a. M. R. K. Marek A. Perkowski, and P. Kerntopf. Ternary gfsop minimization using kronecker decision diagrams and their synthesis with quantum cascades. *Journal of Multi Valued Logic and Soft Computing*, 11:567602, 2005.
9. R. Khanoma, T. Kamalb, and M. H. A. Khana. Genetic algorithm based synthesis of ternary reversible/quantum circuit. In *11th International Conference on Computer and Information Technology ICCIT 2008*, pages 270–275. IEEE, 2008.
10. X. Li, G. Yang, and D. Zheng. Logic synthesis of ternary quantum circuits with minimal qutrits. *Journal of Computers*, 8(3):1941–1946, December 2013.
11. M. Lukac, M. Perkowski, H. Goi, M. Pivtoraiko, C. H. Yu, K. Chung, H. Jee, B. G. Kim, and Y. D. Kim. *Artificial intelligence in logic design*. Springer Netherlands, 2004.
12. S. Mandal, A. Chakrabarti, and S. Sur-Kolay. Synthesis techniques for ternary quantum logic. In *Proceedings of International Symposium on Multiple-Valued Logic (ISMVL)*, pages 218–223. IEEE, 2011.
13. D. McHugh and J. Twamley. Trapped-ion qutrit spin molecule quantum computer. *New Journal of Physics*, 7:174/1–9, 2005.
14. A. Muthukrishnan and J. C. R. Stroud. Multivalued logic gates for quantum computation. *Phys. Rev. A.*, 62(5):052309/1–8, 2000.
15. M. Saeedi, M. S. Zamani, M. Sedighi, and Z. Sasanian. Reversible circuit synthesis using a cycle-based approach. *J. Emerg. Technol.Comput. Syst. ACM*, pages 1–13, 2010.
16. R. Wille, D. Große, L. Teuber, G. Dueck, and R. Drechsler. RevLib: An online resource for reversible functions and reversible circuits. In *International Symposium on Multi-Valued Logic*, pages 220–225, 2008. RevLib is available at http://www.revlib.org.
17. G. Yang, X. Song, M. Perkowski, and J. Wu. Realizing ternary quantum switching networks without ancilla bits. *Journal of Physics A: Mathematical and General*, 38:1–10, 2005.
18. G. Yang, F. Xie, X. Song, and M. Perkowski. Universality of 2-qudit ternary reversible gates. *Journal of Physics A: Mathematical and General*, 39:7763–7773, 2006.