On the Skipped Variables of Quantum Multiple-valued Decision Diagrams

David Y. Feinstein
Innoventions, Inc.
10425 Bissonnet Street
Houston, TX, USA
david@innoventions.com

Mitchell A. Thornton
Dept. of Computer Science and Engineering
Dept. of Electrical Engineering
Southern Methodist University
Dallas, TX, USA
mitch@engr.smu.edu

Abstract

The data structure referred to as quantum multiple-valued decision diagrams (QMDD) is used to efficiently represent the unitary matrices describing reversible and quantum circuits. This paper investigates the conditions that cause skipped variables to appear in the QMDD of some binary and ternary quantum circuits. We have found that a unitary matrix that produces a skipped variable in a QMDD is likely to cause a specific anomaly when it is decomposed into a cascade of two-level unitary matrices by the Beck-Zeilinger-Bernstein-Bertani algorithm.

1. INTRODUCTION

The potential exponential speed up for some intractable problems such as prime factoring [1] and searching [2] promoted extensive research interest in quantum computing. Quantum circuits are logically and physically reversible and are fully described by a unitary transformation matrix. Interest in unitary matrices as building blocks for quantum circuits creates the need for efficient metrics and analysis methods [3]. Earlier work by Beck, Zeilinger, Bernstein, and Bertani shows that any unitary matrix may be synthesized as a quantum circuit in the laboratory using beam splitters, thus establishing the beam splitter as a universal gate [4]. Large numbers of quantum, binary and MVL gates have been developed in the hope of achieving efficient synthesis of quantum circuits [5]. Various CAD tools have been developed to assist these efforts, including the *Quantum Information Decision Diagram* (QuIDD) [6] and the Quantum Multiple-valued Decision Diagrams (QMDD)[7] packages. In earlier work, we have used QMDD for simulation, synthesis, and investigation of redundancy in reversible circuits [8,9,10].

A crucial issue for quantum circuit simulation is whether we should constrain the control lines to take only the "binary" values $|0\rangle$ and $|1\rangle$ or to allow the control lines the freedom of using superposition values. In their paper on quantum synthesis using the *quantum decision diagram* (QDD) structure, Abdollahi and Pedram dubbed this issue as the *binary control signal constraint* [11]. They pointed out that researchers often adopt this constraint in quantum logic synthesis without explicit notice. In practice, various quantum circuits (components of the Grover search [2], fault tolerance stabilizer circuits [12], etc.) allow the control inputs to be fed with values in superposition [11]. We found that such quantum circuits may cause skipped variables in QMDD-based simulations [8]. Skipped variables refer to the existence of a path in a QMDD where a particular variable does not appear when compared to the sequence of variables in other QMDD paths. Skipped variables are important when the QMDD (or other ordered decision diagram structures) are manipulated or used for the purpose of quantum circuit synthesis. For this reason, it is important to understand why variables are skipped and what characteristics the underlying circuits must have that result in skipped variables. In this paper we investigate skipped variables in binary and ternary QMDD, and we found a correlating property that relates the appearance of skipped variable in a quantum circuit with a certain anomaly observed during the BZBB decomposition of the circuit into beam splitters.

The paper is organized as follows. In Section 2 we discuss binary and MVL reversible circuits, the QMDD data structure, and we detail the BZBB decomposition process. Theoretical analysis of skipped variables in binary and ternary circuits is outlined in Section 3. In Section 4 we discuss our preliminary experimental results in comparing quantum circuits with skipped variables to their corresponding BZBB decompositions. Conclusions and suggestions for further research appear in Section 5.

2. PRELIMINARIES

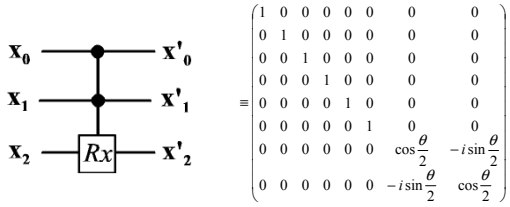
2.1. Reversible Circuits

Definition 1: A binary or MVL gate/circuit is logically *reversible* if it maps each input pattern to a unique output pattern. This mapping is defined by the *transformation matrix* of the circuit. □

For binary reversible logic, the transformation matrix is of the form of a permutation matrix. For quantum circuits, the transformation matrix is a unitary matrix with complex-valued elements. An $n \times n$ reversible circuit with n inputs and n outputs requires a $r^n \times r^n$ transformation matrix, where r is the radix.

Definition 2: An $n \times n$ unitary matrix ($n \geq 2$) that transforms only two or fewer vector components is referred to as a two-level unitary matrix. □

Many binary and MVL reversible gates have been proposed [5,13]. Fig. 1 illustrates a quantum two-level unitary matrix which is the transformation matrix of the Deutsch-Toffoli universal gate. The control lines of each gate are denoted by filled circles while the target line is marked by a square with a function name (*Rx* in this case, and *U* in the general case). A common binary reversible gate is the Control-Not or Toffoli gate where the unitary function (the Not operation) is represented by a circle [14].



$$\equiv \begin{pmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&\cos\frac{\theta}{2}&-i\sin\frac{\theta}{2}\\ 0&0&0&0&0&0&-i\sin\frac{\theta}{2}&\cos\frac{\theta}{2} \end{pmatrix}$$

Figure 1. The Deutsch-Toffoli Universal Gate

Definition 3: An *n*-variable reversible gate cascade is a circuit composed of adjacent reversible gates that operate on the same *n* variables represented by horizontal lines across the circuit. Each gate may be connected to one or more of the lines and must be extended via the tensor product operator to affect all *n* lines. □

Fig. 2 illustrates 4-variable quantum cascade C that preserves the *binary control signal constraint.* It includes gates G1 (Toffoli), G2 (Deutsch-Toffoli), G3 (Not) and G4 (general unitary).

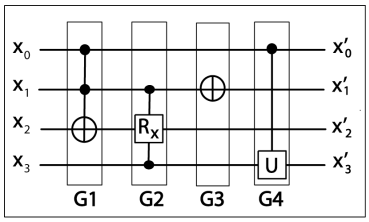


Figure 2. 4-variable quantum cascade

The transformation matrix of a circuit cascade is computed by multiplying the transformation matrices of the gates, starting with the rightmost gate.

$$\mathbf{C} = G4 \times G3 \times G2 \times G1 \qquad (1)$$

The unitary transformation matrix **C** maps the vector $\{x_3,x_2,x_1,x_0\}$ to vector $\{x_3',x_2',x_1',x_0'\}$ and vice versa.

2.2. Quantum Multiple-valued Decision Diagrams

The transformation matrix *M* of dimension $r^n \times r^n$ representing an MVL reversible/quantum logic circuit *C* with radix *r* can be partitioned as

$$M = \begin{bmatrix} M_0 & M_1 & \cdots & M_{r-1} \\ M_r & M_{r+1} & \cdots & M_{2r-1} \\ \vdots & \vdots & \ddots & \\ M_{r^2-r} & M_{r^2-r+1} & \cdots & M_{r^2-1} \end{bmatrix}$$

where each M_i element is a submatrix of dimension $r^{n-1} \times r^{n-1}$. This partitioning is exploited by the QMDD structure and is used to specify the circuit *C* in a compact form [7]. In a manner similar to a reduced ordered binary decision diagram (*ROBDD*) [15], a QMDD adheres to a fixed variable ordering and common substructures (representing submatrices) are shared. A QMDD has a single terminal vertex with value 1, and each edge in the QMDD, including the edge pointing to the start vertex, has an associated multiplicative complex-valued weight.

Theorem 1: *An $r^n \times r^n$ complex-valued matrix M representing a reversible or quantum circuit has a unique (up to variable reordering or relabeling) QMDD representation.*

Proof: A proof by induction based on the normalization of edge weights that may be performed during the construction of a QMDD is detailed in [7,8]. □

2.3. The BZBB Decomposition of a Unitary Matrix

The function of a beam splitter, which is a common component for quantum optical experiments, can be represented by a two-level unitary matrix. Beck et al. showed that a beam splitter is essentially a universal gate, capable of implementing any arbitrary finite unitary matrix [4]. More specifically, they proved that:

Theorem 2: An arbitrary $n \times n$ unity matrix **U** can be decomposed into a product of two-level unitary matrices so that $\mathbf{U}=\mathbf{V}_1\mathbf{V}_2....\mathbf{V}_k$ *and* $k \le n(n-1)/2$.

Proof: A proof by construction is detailed in [4] and useful examples appear in [5]. We outline the proof's construction here since we use the BZBB decomposition in Section 4. The main idea is to create a cascade of two-level matrices to transform U into a diagonal matrix. A first set of *n* (or less) two-level matrices transforms U into an intermediate matrix with a 1 in the first diagonal element and 0s elsewhere in the first column and first row. Another set of *n*-1 (or less) two-level matrices transforms the last intermediate matrix into another with a '1' in the second diagonal element and 0s elsewhere on the second column and second row. The process continues recursively until U is fully transformed into a diagonal matrix, so that $\mathbf{V}_1\mathbf{V}_2...\mathbf{V}_k\mathbf{U}=\mathbf{I}$. It is easy to see that $k \le (n-1)+(n-2)+...+1 = n(n-1)/2$. Since all $\mathbf{V}_i$ are unitary, $\mathbf{U} = \mathbf{V}_1^+\mathbf{V}_2^+...\mathbf{V}_k^+$. □

In Table 1 we illustrate the BZBB decomposition of two different 4×4 unitary matrices (each matrix is decomposed step-by-step along two columns of the table). Both examples require six two-level unitary matrices, resulting in the cascade

$$\mathbf{U} = \mathbf{V}_1^+\mathbf{V}_2^+\mathbf{V}_3^+\mathbf{V}_4^+\mathbf{V}_5^+\mathbf{V}_6^+ \qquad (2)$$

3. QMDD SKIPPED VARIABLES

3.1. Conditions for Skipped Variables

The QMDD points to any sub-matrix having all zero elements directly to the terminal node with *zero weighted edges*. Since quantum circuits are represented by unitary matrices that are typically sparse, QMDD achieve efficient sizes by exploiting the large number of zero weighted edges. However, like any other decision diagram, size explosion is still a potential limitation when working with QMDDs. A sifting technique to minimize the structure that employs local QMDD swap operations was investigated in [8].

Definition 4: A QMDD variable i is *skipped* if a non-terminal vertex representing decision variable i+1 has a non-zero weighted edge that points to a non-terminal vertex of decision variable i-1, thus *skipping the intermediate variable i in the overall ordering.* □

Sifting based on local swap operations can be made very efficient when the QMDD does not have skipped variables. Therefore, it is helpful to determine when quantum circuits may exhibit skipped variables [16].

Lemma 1: In order for an intermediate vertex representing decision variable i to be skipped in a QMDD, at least one of the $r \times r$ decomposed sub-matrices for the decision variable i+1 matrix must be further decomposed by r^2 identical sub-matrices.

Proof: The $r^i \times r^i$ sub-matrix of decision variable i+1 with r^2 identical sub-matrix decompositions represents a vertex that has all outgoing edges pointing to identical subtrees. By Theorem 1, a non-terminal vertex is redundant if all r^2 edges point to the same vertex with the same weight. It is easy to see that the condition in Lemma 1 causes the intermediate vertex i to be redundant, since the identical matrices pointed to by all edges are represented by the same vertex due to the uniqueness of all the QMDD vertices. This same vertex represents a matrix of size $r^{i-1} \times r^{i-1}$, which represents decision variable i+1. We thus show that the condition in Lemma 1 causes a vertex of decision variable i+1 to skip directly to a vertex of decision variable i-1. □

Fig. 3 shows the occurrence of a skipped variable in a *binary* QMDD in view of Lemma 1. The transformation matrix for vertex V1 at variable i+1 is decomposed into four sub-matrices that are subsequently decomposed at variable i. Note that variable numbering decreases from the root vertex to the terminal node T. To avoid cluttering of the illustrations, zero weighted edges are shown as truncated stubs with 0. The 4th sub-matrix is decomposed into four identical sub-matrices, and in accordance with Lemma 1 it causes a skipped variable to occur. Vertex V4 at variable $\underline{i}$ is eliminated by an edge that points directly to V7 at variable i-1.

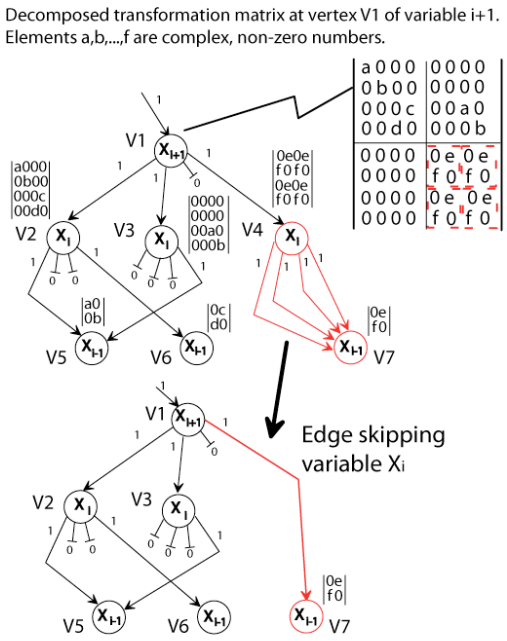


Figure 3. Skipped Variable in a Binary QMDD

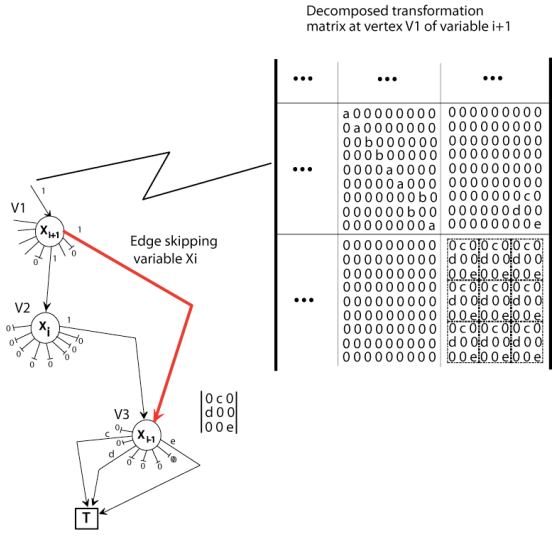


Figure 4. Skipped Variable in a Ternary QMDD

Fig. 4 shows the occurrence of skipped variable in a ternary QMDD. Notice that the transformation matrix at each vertex is decomposed into 9 sub-matrices, and each vertex has 9 edges. The last sub-matrix at vertex V1 is

decomposed into *identical* 9 sub-matrices, thus causing an edge that skips variable *i* in accordance with Lemma 1.

3.2. Quantum Circuits with Skipped Variables

In our previous work we have shown that binary reversible circuits cannot result in QMDD representations containing skipped variables because their transformation matrices are necessarily of the form of a permutation matrix [8]. A permutation matrix only contains a single '1' value in each column and row, thus the condition of Lemma 1 cannot be met. In this work we investigate unitary matrices that have many non-zero elements that may produce skipped variables.

Definition 5: We will refer to an $N\times N$ matrix D with all diagonal elements equal to $\frac{2}{N}-1$ and all the remaining elements equal to $\frac{2}{N}$ as $D_I(N)$ matrices. □

$$D_I(N)=\begin{pmatrix} \frac{2}{N}-1 & \frac{2}{N} & \frac{2}{N} & \dots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N}-1 & \frac{2}{N} & \dots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} & \frac{2}{N}-1 & \dots & \frac{2}{N} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{2}{N} & \frac{2}{N} & \frac{2}{N} & \dots & \frac{2}{N}-1 \end{pmatrix} \quad (3)$$

Lemma 2: If N is an integer such that $N>1$, then the $D_I(N)$ matrix is unitary. If m is an integer such that $m>1$, then a QMDD representing $D_I(2^m)$ must have skipped variables. Furthermore, the QMDD must have at least one vertex with a non-zero weight that skips m-1 variables.

Proof: It is easy to show that $D_I(2^m)\ D_I(2^m)^*=I$, and therefore, $D_I(2^m)$ is unitary and represents a valid quantum state transformation. When $n=2^m$, where m is an integer and $m>1$, $D_I(2^m)$ can be represented by a binary (radix r = 2) QMDD. To prove that $D_I(2^m)$ contains a skipped variable, consider the decomposition of the matrix into 4 sub-matrices at each decision variable during the construction of the QMDD. For $m=2$, the $D_I(4)$ matrix is of dimension 4×4 and is unitary.

$$D_I(4)=\begin{matrix} -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \end{matrix} \quad (4)$$

The first decision variable x_1 decomposes $D_I(4)$ into four 2×2 sub-matrices. Clearly these sub-matrices are not equal. The next and final decision variable x_0 further decomposes each of the four 2×2 sub-matrices into four 1×1 sub-matrices. With this example, it is easy to show that the condition of Lemma 1 exists within the second and third sub-matrices that are

$$S_1=S_2=\begin{matrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{matrix} \quad (5)$$

It is easy to show by induction that matrix $D_I(2^m)$ where $m>1$ results in a QMDD containing skipped variables that actually skip over m-1 variables. This proves the case for $m=2$ that there is a non-zero weighted edge that skips m-1=1 variables.

For the m+1 case, we note that increasing m by 1 adds one decision variable and quadruples the size of the transformation matrix from that of m. Since all elements except the diagonal elements are the same, we have a skipped variable which skips one more variable than the case for m. □

Fig. 5 illustrates the skipped variables that arise in the $D_I(8)$ unitary matrix.

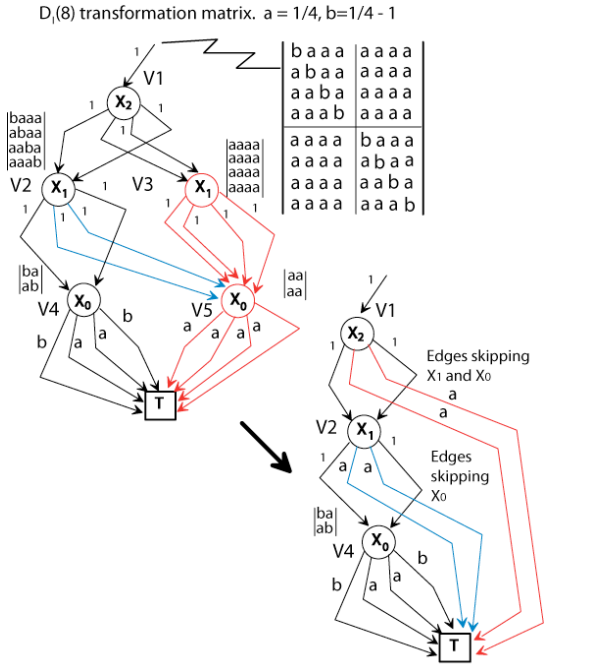


Figure 5. Skipped Variable in QMDD of $D_I(8)$

The $D_I(N)$ matrices, which represent quantum circuits that are used by the Grover search algorithm clearly require the control inputs to have superposition values, thus relaxing the binary control signal constraint [11].

4. EXPERIMENTAL RESULTS

In the first phase of our experimentation we considered several published quantum circuits in our QMDD simulator and we noted that the occurrence of skipped variables is indeed a rare phenomenon among this test set. Except for the $D_I(N)$ matrices that were discussed in Section 3.2 (or their tensor products when

they were incorporated as part of larger quantum circuits) we have not observed any other quantum circuits that produce skipped variables. In addition, we have generated numerous "random" unitary matrices using the common Scilab mathematical software, and found that none of examples exhibited skipped variables.

In the second phase of our experimentation, we investigated the $D_I(N)$ matrices to determine if the fact that they cause QMDD skipped variables correlates with any anomaly metrics associated with other processes that may be performed on unitary matrices. We found an interesting comparison with the BZBB decomposition. As explained in Section 2.3, this decomposition of an arbitrary $n \times n$ unitary matrix creates a cascade of up to $n(n\text{-}1)/2$ two-level unitary matrices. Clearly this process is not efficient in comparison with other synthesis methods. However, the two-level unitary matrices employed by the BZBB decomposition are the lowest complexity "universal gates", and as such we expected them to provide some fundamental insight into the metrics of the decomposed matrix.

An interesting sample of our investigation is detailed in Table 1. In the 1st and 2nd columns we show the step-by-step BZBB decomposition of a 4×4 unitary matrix that **does not exhibit skipped variables**. The 3rd and 4th columns of Table 1 show the BZBB decomposition of the $D_I(4)$ unitary matrix that does exhibit skipped variables. In fact, this matrix represents the well known HCH quantum circuit shown in Fig. 6 that is used to represent the superimposed result of a CNOT operation.

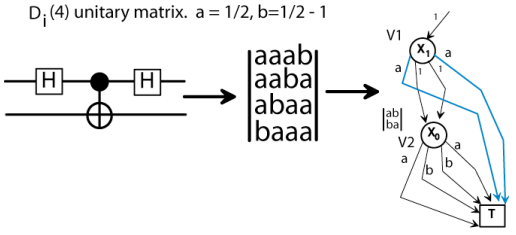


Figure 6. Skipped Variable in QMDD of HCH

In the case of the unitary matrix without skipped variables, we saw that the intermediate matrices (in 2nd column) never exhibit 0 in the diagonal. In contrast, the $D_I(4)$ matrix with the skipped variables exhibits a 0 in the diagonal of the intermediate matrix (in 4th column). This occurrence must be followed by a permutation matrix (V_4 in the 3rd column) to rotate a non-zero number into the diagonal. Similar behavior was observed in other $D_I(N)$ matrices.

The QMDD requires that any sub-matrix that has all 0 elements point directly to the terminal vertex with zero weight. It should be noted that in Definition 4 we made a distinction that edges that skip variables must have a non-zero weight. Without this distinction, it can be shown that every sparse unitary matrix exhibits skipped variables. Naturally, the BZBB decomposition does not make this distinction and it responds with a permutation matrix even when variables are skipped with zero-weight edges. As a result, 0 elements do appear in the diagonal of the intermediate matrices in the decomposition of sparse unitary matrices (e.g. permutation matrices representing binary reversible circuits).

5. CONCLUSIONS AND FUTURE WORK

This paper has considered the phenomenon of skipped variables in QMDD. This phenomenon was found to occur rarely; however, its occurrence may reduce the overall efficiency of sifting-based QMDD minimization.. The results are used to determine the conditions that cause skipped variables and several useful circuits that are known to exhibit skipped variables are reviewed. We investigated the correlation between circuits that have skipped variables in their QMDD representation and the appearance of an anomaly in their BZBB decomposition.

In future work we will investigate other types of unitary decompositions that may exhibit various anomalies with circuits that exhibit skipped variables. It will be interesting to see if the skipped variable phenomenon has bearing to other Quantum decision diagram packages. We will also consider the potential to use the existence of skipped variables in the QMDD of a unitary matrix as a basis for metrics that may aid the synthesis of the circuit represented by the unitary matrix.

ACKNOWLEDGEMENT

The authors would like to acknowledge Dr. Eike Rietsch for developing and providing the Scilab script for the BZBB decompositions used in this work.

REFERENCES

[1] P. W. Shor, "Algorithms for quantum computation: Discrete log and factoring", In *Proc. of the 35th Annual Symposium on Foundations of Computer Science*, pp. 124-134, 1994.

[2] L. Grover, "A Fast Quantum Mechanical Algorithm for Database Search", In *Proc. ACM Symp. on Theory of Computing*, pp. 212-219, 1996.

[3] H. F. Chau, "Metrics On Unitary Matrices, Bounds On Eigenvalues Of Product Of Unitary Matrices, And Measures Of Non-commutativity Between Two Unitary Matrices", In *arXiv:1006.3614v1 [quant-ph]*, 2010.

[4] M. Beck, A. Zeilinger, H.J. Bernstein, and P. Bertani, Experimental realization of any discrete unitary operator. In *Phys. Rev. Let.*, 73(1), pp. 58-61, 1994.

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

[6] G.F. Viamontes, M. Rajagopalan, I.L. Markov, and J.P. Hayes, "Gate-level simulation of quantum circuits", In *ASP-DAC 2003,* pp. 295–301, 2003.

[7] D.M. Miller and M.A. Thornton, "QMDD: A Decision Diagram Structure for Reversible and Quantum Circuits," In *ISMVL'06*, on CD, 2006.

[8] D. M. Miller, D. Y. Feinstein, and M. A. Thornton, "QMDD Minimization using Sifting for Variable Reordering", In *Journal of*

Multiple-valued Logic and Soft Computing, Vol. 13, no. 4-6, pp. 537-552, 2007.

[9] D. Y. Feinstein and M. A. Thornton, "On the Guidance of Reversible Logic Synthesis by Dynamic Variable Ordering", In *ISMVL'09,* pp. 132-138, 2009.

[10] D. Y. Feinstein, M. A. Thornton, and D. M. Miller, "Partially Redundant Logic Detection Using Symbolic Equivalence Checking in Reversible and Irreversible Logic Circuits", In *DATE'08* pp 1378-1381, 2008.

[11] A. Abdollahi and M. Pedram, "Efficient synthesis of quantum logic circuits by rotation-based quantum operators and unitary functional bi-decomposition", In *IWSL'05,* on CD ROM, 2005.

[12] D. Gottesman, "A theory of fault-tolerant quantum omputation" arXiv:quant-ph/9702029V2, 1997.

[13] D. M. Miller, G. Dueck, and D. Maslov, "A Synthesis Method for MVL Reversible Logic," In *ISMVL'04*, pp. 74-80, 2004.

[14] T. Toffoloi, "Reversible Computing," In *Automata, Languages, and Programming*, Springer Verlag, pp. 632-644, 1980.

[15] R.E. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation*"*, In *IEEE Trans. Computers*, C-35(8):677-691, 1986.

[16] D. Y. Feinstein, *Computer-Aided-Design Methods for Emerging Quantum Computing Technologies*, Ph.D. Thesis, Southern Methodist University, 2008.

Table 1. BZBB Decomposition of 4x4 Unitary Matrices

BZBB Decomposition of a Unitary Matrix without Skipped Variables		**BZBB Decomposition of a Unitary Matrix with Skipped Variables**	
2-Level Unitary Matrix	Intermediate Cascade Matrix	2-Level Unitary Matrix	Intermediate Cascade Matrix
	Decomposed Matrix $\mathbf{U}=\begin{bmatrix}0.5 & 0.5 & 0.5 & 0.5\\ 0.5 & 0.5i & -0.5 & -0.5i\\ 0.5 & -0.5 & 0.5 & -0.5\\ 0.5 & -0.5i & -0.5 & 0.5i\end{bmatrix}$		Decomposed Matrix $\mathbf{U}=\mathbf{HCH}=\begin{bmatrix}0.5 & 0.5 & 0.5 & -0.5\\ 0.5 & 0.5 & -0.5 & 0.5\\ 0.5 & -0.5 & 0.5 & 0.5\\ -0.5 & 0.5 & 0.5 & 0.5\end{bmatrix}$
$\mathbf{V}_1=\begin{bmatrix}0.7071 & 0.7071 & 0 & 0\\ 0.7071 & -0.7071 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$	$\mathbf{V_1U}=\begin{bmatrix}0.7071 & 0.3536+0.3536i & 0 & 0.3536-0.3536i\\ 0 & 0.3536-0.3536i & 0.7071 & 0.3536+0.3536i\\ 0.5 & -0.5 & 0.5 & -0.5\\ 0.5 & -0.5i & -0.5 & 0.5i\end{bmatrix}$	$\mathbf{V}_1=\begin{bmatrix}0.7071 & 0.7071 & 0 & 0\\ 0.7071 & -0.7071 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$	$\mathbf{V_1U}=\begin{bmatrix}0.7071 & 0.7071 & 0 & 0\\ 0 & 0 & 0.7071 & -0.7071\\ 0.5 & -0.5 & 0.5 & 0.5\\ -0.5 & 0.5 & 0.5 & 0.5\end{bmatrix}$
$\mathbf{V}_2=\begin{bmatrix}0.8165 & 0 & 0.5774 & 0\\ 0 & 1 & 0 & 0\\ 0.5774 & 0 & -0.8165 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$	$\mathbf{V_2V_1U}=\begin{bmatrix}0.866 & 0.2887i & 0.2887 & -0.2887i\\ 0 & 0.3536-0.3536i & 0.7071 & 0.3536+0.3536i\\ 0 & 0.6124+0.2041i & -.4083 & 0.6124-0.2041i\\ 0.5 & -0.5i & -0.5 & 0.5i\end{bmatrix}$	$\mathbf{V}_2=\begin{bmatrix}0.8165 & 0 & 0.5774 & 0\\ 0 & 1 & 0 & 0\\ 0.5774 & 0 & -.8165 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$	$\mathbf{V_2V_1U}=\begin{bmatrix}0.866 & 0.2887 & 0.2887 & 0.2887\\ 0 & 0 & 0.7071 & -0.7071\\ 0 & 0.8165 & -0.4082 & -0.4082\\ -0.5 & 0.5 & 0.5 & 0.5\end{bmatrix}$
$\mathbf{V}_3=\begin{bmatrix}0.866 & 0 & 0 & 0.5\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0.5 & 0 & 0 & 0.866\end{bmatrix}$	$\mathbf{V_3V_2V_1U}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0.3536-0.3536i & 0.7071 & 0.3536+0.3536i\\ 0 & 0.6124+0.2041i & -0.4083 & 0.6124-0.2041i\\ 0 & 0.5774i & 0.5774 & -0.5774i\end{bmatrix}$	$\mathbf{V}_3=\begin{bmatrix}0.866 & 0 & 0 & -0.5\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ -0.5 & 0 & 0 & -0.866\end{bmatrix}$	$\mathbf{V_3V_2V_1U}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 0.7071 & -0.7071\\ 0 & 0.8165 & -0.4082 & -0.4082\\ 0 & -0.5774 & -0.5774 & -0.5774\end{bmatrix}$
$\mathbf{V}_4=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0.433+0.433i & 0.75-0.25i & 0\\ 0 & 0.75+0.25i & 0.433+0.433i & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$	$\mathbf{V_4V_3V_2V_1U}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0.8165 & 0.4082i & 0.4082\\ 0 & 0 & 0.7071 & 0.7071i\\ 0 & 0.5774i & 0.5774 & -0.5774i\end{bmatrix}$	$\mathbf{V}_4=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$	$\mathbf{V_4V_3V_2V_1U}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0.8165 & -0.4082 & -0.4082\\ 0 & 0 & 0.7071 & -0.7071\\ 0 & -0.5774 & -0.5774 & -0.5774\end{bmatrix}$
$\mathbf{V}_5=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0.8165 & 0 & -0.5774i\\ 0 & 0 & 1 & 0\\ 0 & 0.5774i & 0 & 0.8165\end{bmatrix}$	$\mathbf{V_5V_4V_3V_2V_1U}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0.7071 & 0.7071i\\ 0 & 0 & -0.7071 & 0.7071i\end{bmatrix}$	$\mathbf{V}_5=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0.8165 & 0 & -0.5774\\ 0 & 0 & 1 & 0\\ 0 & -0.5774 & 0 & -0.8165\end{bmatrix}$	$\mathbf{V_5V_4V_3V_2V_1U}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0.7071 & -0.7071\\ 0 & 0 & 0.7071 & 0.7071\end{bmatrix}$
$\mathbf{V}_6=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0.7071 & -0.7071\\ 0 & 0 & -0.7071i & -0.7071i\end{bmatrix}$	**Finally -** $\mathbf{V_6V_5V_4V_3V_2V_1U}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$	$\mathbf{V}_6=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0.7071 & 0.7071\\ 0 & 0 & -0.7071 & 0.7071\end{bmatrix}$	**Finally -** $\mathbf{V_6V_5V_4V_3V_2V_1U}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$