On the Data Structure Metrics of Quantum Multiple-valued Decision Diagrams

David Y. Feinstein and Mitchell A. Thornton
Department of Computer Science and Engineering
Southern Methodist University
Dallas, TX, USA
dfeinste,mitch@engr.smu.edu

D. Michael Miller
Department of Computer Science
University of Victoria
Victoria, BC, Canada
mmiller@cs.uvic.ca

Abstract

This paper describes new metrics for the data structure referred to as quantum multiple-valued decision diagrams (QMDD) which are used to represent the matrices describing reversible and quantum gates and circuits. These metrics provide information about QMDD that allows for improvement of minimization techniques. We explore metrics related to the frequency of edges with non-zero weight for the entire QMDD data structure and their histograms with respect to each variable. We observe some unique regularity particular to the methodology of the QMDD. We develop new heuristics for QMDD dynamic variable ordering (DVO) that are guided by the proposed metrics. Experimental results show the effectiveness of the proposed techniques.

1. Introduction

The *quantum multiple-valued decision diagram* (QMDD) was proposed for the efficient specification and simulation of reversible and quantum circuits [9,10]. The QMDD data structure has been successfully used for reversible and quantum circuit simulations, equivalence checking, and other applications [4]. In this paper we develop and investigate the use of data structure metrics for the analysis and minimization of QMDD.

Unlike the *binary decision diagram* (BDD) which allows only two possible transition edges from each vertex, and which is used in the QuIDDPro package [15], a multiple-valued QMDD has r^2 transition edges from each vertex (where r is the radix). Since the transformation matrices representing quantum circuits are often sparse, many QMDD edges point directly to the terminal node with zero weight. This creates an interesting and complex inter-connectivity among the QMDD vertices, that allows QMDD to represent relatively complex circuits with a small number of vertices.

The data structure metrics proposed in this research are used to better understand this inter-connectivity. They include the ratio of non-zero weight edges to the number of vertices for the entire QMDD as well as a histogram of the similar ratio measured for each variable. We propose a set of heuristics based on the values of these metrics to guide *dynamic variable ordering* (DVO) resulting in an improvement in minimization for many benchmark circuits.

In our previous work, we have developed a DVO sifting algorithm that utilized efficient adjacent variable interchange [8]. Our sifting algorithm, adapted for QMDD from Rudell's binary ROBDD sifting approach [13], achieved significant size reductions on some benchmarks of reversible/quantum circuits of n-variables while considering only $O(n^2)$ of the $n!$ possible variable orderings.

In this work we offer additional heuristic algorithms for QMDD minimization. The new heuristics achieve improved minimization for some benchmarks over our previous sifting algorithm. Since such DVO minimization efforts consume processing time, it is beneficial to be able to predict in advance whether a certain QMDD structure can be minimized [16]. We describe heuristics, based on our new data structure metrics, that provide effective prediction in this regard for most of the benchmarks we tested.

The paper is organized as follows. In Section 2 we briefly discuss reversible logic, the QMDD structure, and our previous work on QMDD minimization. The proposed QMDD data structure metrics are described in Section 3. In Section 4 we discuss the application of the proposed new metrics in QMDD minimization heuristics, and in Section 5 we discuss our preliminary experimental results. Conclusions and suggestions for further research appear in Section 6.

2. Preliminaries

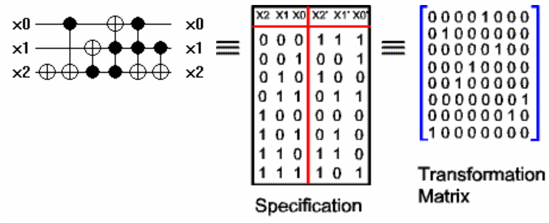
2.1. Reversible Logic and Quantum Circuits

In this section, we briefly introduce the basic concepts of reversible and quantum circuits. More extensive background is available in [11].

Definition 1: A gate / circuit is logically *reversible* if it maps each input pattern to a unique output pattern. For classical reversible logic, the mapping is a permutation matrix. For quantum circuits, the gate / circuit operation can be described by a unitary transformation matrix.

Bennet [1] showed that reversible gates can theoretically result in binary circuits that are completely free from energy loss. The concept of reversibility has been extended to MVL circuits [7]. Binary quantum logic gates and circuits are inherently both logically and physically reversible [11]. Non-binary quantum logic circuits are logically reversible.

Fig. 1 shows a 3 line and 6 gate binary reversible circuit "c3_17.nct" from D. Maslov's benchmarks [6]. The symbol ⊕ denotes the NOT operation. For each gate, the NOT operates on the *target* line if every *control* line (lines with a black circle) has the value 1. Otherwise the target line is unchanged. Control and unconnected lines pass through the gate unchanged. A gate with no controls is a conventional *NOT* gate. One with a single control is termed a *controlled-NOT* and gates with more than one control are referred to as *Toffoli* gates [11].



X2	X1	X0	X2'	X1'	X0'
0	0	0	1	1	1
0	0	1	0	0	1
0	1	0	1	0	0
0	1	1	0	1	1
1	0	0	0	0	0
1	0	1	0	1	0
1	1	0	1	1	0
1	1	1	1	0	1

$$\begin{bmatrix} 0&0&0&0&1&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&1&0 \\ 1&0&0&0&0&0&0&0 \end{bmatrix}$$

Figure 1. 3-variable benchmark C3_17.nct

Fig. 1 shows the specification and the transformation matrix for the benchmark, as captured by the QMDD tool. In general, an r-valued reversible circuit with n lines (that is, a circuit with n inputs and n outputs) requires a transformation matrix of dimension $[r^n \times r^n]$, where r is the radix.

2.2 Quantum Multiple-valued Decision Diagrams

The QMDD structure was proposed to simulate and specify reversible and quantum logic circuits in a compact form [9]. A matrix of dimension $r^n \times r^n$ can be partitioned as:

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

where each M_i element is a matrix of dimension $r^{n-1} \times r^{n-1}$. A QMDD applies this partitioning analogous to how a reduced ordered binary decision diagram (*ROBDD*) [3] recursively applies Shannon decompositions. In a similar manner to ROBDDs, a QMDD adheres to a fixed variable ordering and common substructures (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 complex-valued weight. Also similar to ROBDDs, the QMDD representation is very useful due to its property of canonicity.

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 iterative construction of a QMDD and the normalization of edge weights that is performed during that construction is detailed in [10].

We show in Fig. 2 the QMDD structure of the binary benchmark "c3_17.nct" given in Fig. 1, as it is captured in our QMDD tool. The full details on the construction and properties of the QMDD are described in [9,10]. Nine nonterminal vertices are required in addition to the terminal vertex. To make the figure more readable, segmented edges marked with 0 are used to denote edges with zero weight that point to the terminal node.

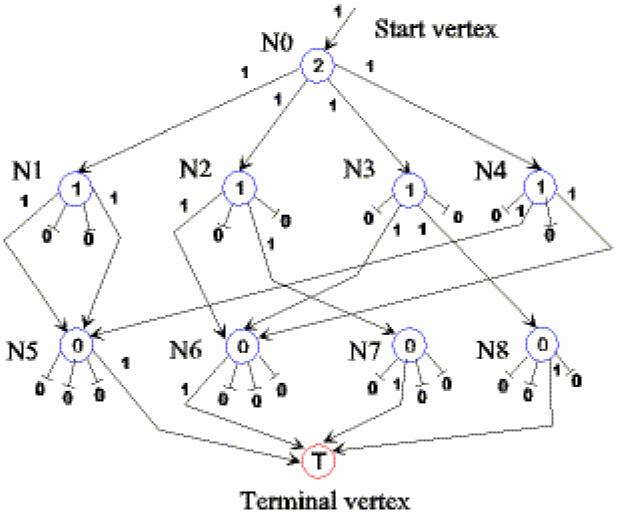


Figure 2. The QMDD for benchmark C3_17.nct

We note that for binary (r=2) reversible / quantum gates / circuits the required matrices can be represented using BDD, e.g. QuIDDPro [15] which employs the very efficient CUDD package [14]. The QMDD approach is quite different even in the binary case since each nonterminal vertex has four outgoing edges rather than two. This makes each vertex more complex but fewer (about half) vertices are required. In addition, the diagram has half the depth. QMDD are also applicable to multiple-valued situations.

One can verify that this QMDD represents the transformation matrix of Fig. 1, by traversing each path from the start vertex to the terminal vertex and multiplying the weights of the edges. While binary reversible circuits like "c3_17.nct" have only 1 and 0 weights, quantum circuits have complex numbers as edge weights.

2.3 Related work on QMDD Variable Sifting

In our previous work we demonstrated that local adjacent variable interchange can be efficiently implemented within the QMDD package [8]. We then showed that a minimizing algorithm similar to Rudell's binary ROBDD sifting algorithm [13] is well suited to reduce the number of nodes in a QMDD representation. Our sifting algorithm employs a greedy selection rule that iteratively picks the variable with the largest number of associated vertices for sifting. In the event of a tie, the variable closest to the terminal node is selected.

Our QMDD sifting algorithm produced minimization levels that varied significantly from example to example. On some benchmarks (e.g. "rd84d1" and "cycle17_3r") we achieved spectacular size reductions (82% and 92.64% respectively). A low improvement on other benchmarks can be the result of having started from what is already a good ordering, or due to the possibility that the function's QMDD representation is insensitive to variable ordering. Since the sifting heuristic visits only a small fraction ($n^2 / n!$) of all possible variable orderings, we are motivated in this paper to explore additional minimization heuristics influenced by the data structure metrics of the QMDD.

3. QMDD Data Structure Metrics

The unique table used by the QMDD software continuously maintains and tracks Active[i], the number of active vertices for each variable *i, 0≤i<n*, where *n* is the number of variables. This is an important QMDD data structure metric that we have used extensively in our previous work. We define a new QMDD data structure metric as follows:

Definition 2: Let *α* be the ratio of the total number of edges with non-zero weight to the total number of vertices in the entire QMDD. Similarly, let α[*i*] be the ratio of the number of edges with non-zero weight that emerge from all the vertices of variable *i* to the number of vertices of the same variable. It follows that,

$$\alpha = \frac{\sum_0^{n-1} Active[i] \times \alpha[i]}{\sum_0^{n-1} Active[i]}$$

Since each nonterminal QMDD vertex has r^2 edges, $1 \le \alpha \le r^2$ and $1 \le \alpha[i] \le r^2$, where r is the QMDD radix. For example, computing $\alpha[1]$ for the QMDD of Fig. 2, we have a total of four vertices and eight non-zero weight edges, thus $\alpha[1]$ = 8/4=2.00. We observe that while vertex *N1* has two edges with non-zero weight, they both connect *N1* to *N5*, essentially connecting the vertex to only one *unique* vertex. We therefore distinguish between the number of edges with non-zero weight and the number of unique *connections* that reach different vertices as follows:

Definition 3: Let *β* be the ratio of the total number of *unique connections with non-zero weight* to the total number of vertices in the entire QMDD. Similarly, let $\beta[i]$ be the ratio of the number of unique connections with non-zero weight edges that emerge from all the vertices of variable *i* to the number of vertices of the same variable. As above, $1 \le \beta \le r^2$ and $1 \le \beta[i] \le r^2$. It follows that

$$\beta = \frac{\sum_0^{n-1} Active[i] \times \beta[i]}{\sum_0^{n-1} Active[i]}$$

For variable 1 in Fig. 2, we have a total of four vertices and seven unique connections with non-zero weight, thus $\beta[1]$ = 7/4=1.75. The relation between *α* and *β* is given by the following lemma.

Lemma 1 *Let* $\alpha, \beta, \alpha[i], and\, \beta[i]$ *be the data structure metrics of a QMDD with n variables as defined above. Hence, (i)* $1 \le \beta[i] \le \alpha[i] \le r^2$ *for all i, 0≤i<n, and (ii)* $1 \le \beta \le \alpha \le r^2$.

Proof: Each vertex of variable *i* must have at least one edge with non-zero weight, since a QMDD vertex with all edges having zero weights is redundant. This edge creates a unique connection to a nonterminal vertex, thus $1 \le \beta[i]$ must be true. We observe that the number of unique connections for the vertices of variable *i* cannot exceed the number of edges with non-zero weight by Definition 3. Thus $\beta[i] \le \alpha[i]$ must be true.

Let *Active[i]* denote the number of active vertices for variable *i*, and *E* the total number of the QMDD's edges with non-zero weight, and *C* the total number of unique connections. Then

$$E = \sum_0^{n-1} Active[i] \times \alpha[i]$$

and

$$C = \sum_0^{n-1} Active[i] \times \beta[i]$$

Since we have proven that $\beta[i] \leq \alpha[i]$, it follows that $C \leq E$ and

$$\beta = \frac{C}{\sum_0^{n-1} Active[i]} \leq \frac{E}{\sum_0^{n-1} Active[i]} = \alpha.$$

Clearly, $\alpha[i] \leq r^2$, with the equality occurring when all edges have non-zero weights. The proof for (ii) follows similar arguments. □

Listing the metrics for QMDD metrics in tabular form provides a histogram-like display of the structure of the QMDD which we will for convenience refer to as a histogram. Table 1 shows the histogram for the QMDD in Fig. 2.

The bottom line of the histogram provides the overall α, β and number of vertices for the entire QMDD. Note that the sum of *Active[i]* does not include the terminal node. The histogram lists the variables according to the *current variable order* of the QMDD. In Table 1, variable 2 labels the start node and variable 0 is the closest variable to the terminal node. As a result, variable *n-1*, the start node, always has *Active[n-1]=1.*

Table 1: Histogram for "3_17.nct"

variable	Active[i]	$\alpha[i]$	$\beta[i]$
2	1	4.00	4.00
1	4	2.00	1.75
0	4	1.00	1.00
total	9	1.78	1.67

BDD researchers have observed that the numbers of vertices for each variable (arranged by the variable order) in general exhibit a pear shape pattern [12]. We have observed that while most QMDD exhibit similar pear shape histograms, many histograms exhibit quite a flat pattern, where most variables have roughly the same *Active[i]*. This allows us to classify QMDD histograms as "FLAT" or "PEAR", as demonstrated in the two histograms of Table 2. The left histogram of "hwb12" exhibits the typical PEAR like pattern commonly observed in classical BDDs. The right histogram of benchmark "cycle10_2*" is of type FLAT as variable 7 to 2 have similar numbers of vertices. We later show that different minimization heuristics apply based on this classification.

Table 2: Histograms for "hwb12" and "cycle10_2*"

var	act[i]	$\alpha[i]$	$\beta[i]$	var	act[i]	α [i]	$\beta[i]$
11	1	4.00	4.00	11	1	2.00	2.00
10	4	4.00	4.00	10	2	4.00	3.00
9	16	4.00	4.00	9	6	1.67	1.67
8	64	4.00	4.00	8	10	2.00	1.60
7	256	3.91	3.91	7	16	1.44	1.19
6	990	2.84	2.84	6	16	1.44	1.19
5	2258	1.37	1.37	5	16	2.00	1.56
4	1174	1.17	1.17	4	18	1.50	1.17
3	304	1.16	1.16	3	18	2.00	1.00
2	76	1.16	1.16	2	15	1.20	1.00
1	19	1.21	1.21	1	9	1.33	1.00
0	4	1.00	1.00	0	3	1.33	1.00
Total	5167	1.76	1.76		131	1.65	1.24

The $\alpha[i]$ and $\beta[i]$ histograms provide additional insight that is explored in the following Section. It should be noted that the QMDD unique table provides efficient means to compute the Active[i] as well as the $\alpha[i]$ and $\beta[i]$ histograms without much processing overhead.

4. Improving QMDD Minimization with the Data Structure Metrics

The inability of our sifting algorithm (as well as any other similar heuristic algorithm) to achieve consistent positive results with all the benchmarks, is of course, due to the fact that it examines only $O(n^2)$ ordering possibilities out of $n!$ possible orderings. To achieve improved minimization, common binary BDD packages offer a rich choice of reordering heuristics. These heuristics can be grouped into sifting algorithms (that include also group sifting and symmetric sifting), random algorithms, window algorithms, and other special algorithms (including simulated annealing and genetic evolutionary algorithms) [16].

While all of these heuristics can be extended for QMDD, we concentrate here on heuristics primarily based on sifting and random approaches. Our experimentation with the group sifting approach that exploits the affinity among variables achieved limited success. One reason is that benchmarks with sufficiently large numbers of variables are currently not available. Furthermore, it seems that since reversible circuits are maximally connected [1,4], they tend to display a strong group behavior as a whole, making attempts to identify subgroups more difficult or redundant.

Our random algorithm performs a set of random variable position changes without regard to structure size between repeated applications of the sifting algorithm. This process breaks apart the initial grouping of variables allowing the sifting algorithm to find a better ordering. A set of heuristics inferred from the data structure metrics is then used to determine when the process should stop.

QMDD Minimization Procedure:

i) Apply the sifting algorithm for initial reordering.

ii) Use metrics-based heuristics to predict if a better variable ordering is likely. If prediction is negative – stop the procedure.

iii) Perform a random set (or a specially designed fixed set) of variable rearrangements *without regard* to structure size.

iv) Repeat steps (i) to (iii). Stop the process if this step is repeated more than *k* times.

We present a sample set of the data structure metrics heuristics that have been investigated. Like all heuristics, they obtain different levels of success as described in the following section, hence the *k* limit in step iv).

Sample Heuristics:

Heuristic 1: For PEAR type benchmarks, if the sifting algorithm provides moderate or no improvement and if the pattern of the histogram of $\alpha[i]$ and $\beta[i]$ histogram is not changed by the sifting, then no significant improvement is likely to be achieved.

Heuristic 2: For FLAT type benchmarks, no significant improvement is likely if the first half of the variables exhibit $\alpha[i] \approx 2.00$.

Heuristic 3: Appearance of $\alpha[i]$ or $\beta[i]$ equal to 2 in the first two variables (of the variable ordering) of a PEAR type benchmark suggests benefit for further reduction attempts, as the QMDD may become FLAT.

Heuristic 4: Benchmarks with $\alpha[i]$ that is not monotonically decreasing (along variable order from the start vertex) resist minimization.

Heuristic 5: If reordering of a FLAT Benchmark changes its histogram shape to PEAR, the total number of vertices will usually increase.

5. Experimental Results

We summarize our result for a broad set of benchmarks in Table 3. The second column indicates the histogram type (PEAR or FLAT) as described in Section 3. The last column indicates the improvement (if any) of the new algorithm over our previous sifting algorithm.

We have seen significant improvements with the benchmarks “9symd2”, “cycle10_2”, “ham7”, “ham15m” and “rd73d2”. We recorded the QMDD histogram during the entire minimization process. An interesting observation for most benchmarks is that the initial QMDD buildup process produces a monotonically decreasing $\alpha[i]$ histogram. On the other hand, the $\beta[i]$ histogram may exhibit fluctuations, although the values are generally decreasing in view of Lemma 1. The only observed violation of the monotonic decrease of $\alpha[i]$ appears with benchmark “mod5adders”. Since this benchmark seems to resists minimization, we use this finding in Heuristics 4.

Table 3: Improvements in QMDD Minimization

Name	Hist type	vars	gates	Initial QMDD vertices	Previous sifting reduction	New algorithm reduction	Improve-ment
5mod5	Pear	6	17	28	42.86%	46.43%	8.33%
6symd2	Pear	10	20	247	50.20%	56.68%	12.91%
9symd2	Pear	12	28	229	19.65%	48.19%	145.24%
mod5adders	Pear	6	21	38	0.00%	0.00%	0.00%
0406142c2	Flat	35	116	150	9.33%	9.33%	0.00%
0410184	Flat	14	46	39	15.38%	15.38%	0.00%
0410184.qc	Flat	14	74	39	15.38%	15.38%	0.00%
cycle17_3	Flat	20	48	236	57.20%	82.20%	43.71%
cycle17_3r	Flat	20	48	236	82.20%	82.20%	0.00%
cycle10_2	Flat	12	19	67	40.30%	62.69%	55.56%
ham7	Pear	7	23	130	3.08%	24.62%	699.35%
ham15	Pear	15	132	4521	49.56%	72.62%	46.53%
ham15m	Pear	15	132	1774	19.17%	36.30%	89.36%
ham15r	Pear	15	132	4522	60.59%	73.33%	21.03%
hwb7	Pear	7	289	179	13.41%	13.41%	0.00%
hwb8	Pear	8	614	343	18.37%	18.37%	0.00%
hwb9	Pear	9	1541	683	23.87%	23.87%	0.00%
hwb10	Pear	10	3631	1331	27.87%	29.30%	5.13%
hwb11	Pear	11	9314	2639	34.44%	34.44%	0.00%
hwb12	Pear	12	18393	5167	38.36%	39.40%	2.71%
rd84d1	Pear	15	28	3588	92.64%	92.64%	0.00%
rd73d2	Pear	15	28	3588	14.00%	40.76%	191.14%
rd53d1	Pear	7	12	26	19.32%	19.32%	0.00%

The “hidden bit” benchmarks (“hwb7”, “hwb8”, “hwb9”, “hwb10”, “hwb11”, and “hwb12”) worked very well with Heuristic 1. In its reversed application, it showed particularly good improvements for “rd73d2”, “ham15r”, “ham15”.

Heuristic 2 seems to work fine but we do not have a large choice of FLAT type benchmarks to verify it more extensively. A similar situation occurs with Heuristic 3.

To further demonstrate Heuristics 5, we subjected the benchmark “cycle10_2” to several random variable reordering, and captured histograms we marked as random0, random1 and random2. We compared the active[i] of these histogram with the minimized histograms obtained by our sifting algorithm and the new algorithm of this paper. We show our results in Fig. 3, where we also include the original ordering histogram. Each series in the graph illustrates the active[i] (number of vertices) for each variable. The variable number is assigned from the start node (1) to the last variable (12) as traversed along the QMDD variable order.

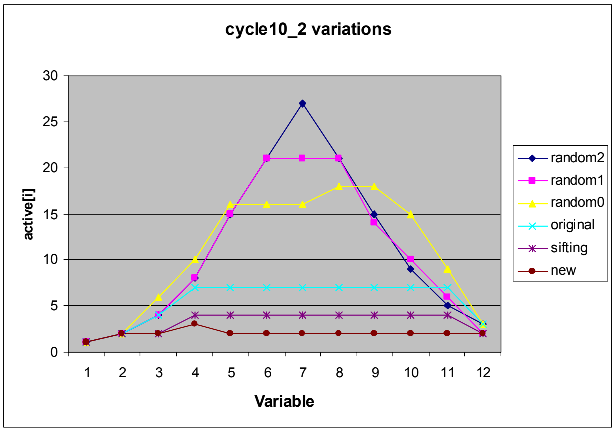


Figure 3. Shape changes from FLAT to PEAR for benchmark Cycle10_2.nct

The total number of vertices for random2, random1, and random0 are 132, 126 and 131 respectively. From Table 3, the total number for the original variable order is 67, and for the minimized sifting and new algorithm it is 40 and 25 respectively. We can see that the shape changes from FLAT to PEAR for random2 and random1. Clearly, these variable orders cause the QMDD to use a much higher number of vertices than in our minimized variable orders. Therefore, a minimized flat benchmark must remain flat during further minimizations.

6. Conclusions

This paper proposes new data structure metrics that provide insight on the inter-connectivity within a QMDD. Heuristics based on these metrics have been developed and used to guide an iterative minimization algorithm. Our preliminary minimization results show some significant improvement versus our sifting algorithm, although no improvement was achieved for some benchmarks.

Considerable work is still needed with the metrics based heuristics, although their potential has been demonstrated to some extent in this paper. We recognize that the heuristics that perform well with our current benchmarks may err with future benchmarks.

In future work we plan to test the application of this data structure metrics for QMDD with a radix larger than 2, where the importance of the inner-connectivity among the vertices is likely to be higher.

References

[1] V. D. Agrawal, "An information theoretic approach to digital fault testing,", In *IEEE Trans. Computers*, Vol. 30, pp. 582-587, Aug. 1981.

[2] C.H. Bennett, "Logical Reversibility of Computation,"*IBM, J. Res. Dev.*, Vol. 17, No. 6, pp. 525-532, 1973.

[3] R.E. Bryant. "*Graph-Based Algorithms for Boolean Function Manipulation"*. IEEE Transactions on Computers, Vol C-35 Issue 8, August 1986, pp. 677-691.

[4] D. Y. Feinstein, M. A. Thornton, and D. M. Miller, "Partially Redundant Logic Detection Using Symbolic Equivalence Checking in Reversible and Irreversible Logic Circuits", to appear in *DATE'08*.

[5] D.C. Marinescu and G.M. Marinescu. *Approaching Quantum Computing*. Pearson Prentice Hall, 2005

[6] D. Maslov. Reversible logic synthesis benchmarks page. http://www.cs.uvic.ca/~dmaslov/, Nov. 15, 2005.

[7] D. M. Miller, G. Dueck, and D. Maslov, "A Synthesis Method for MVL Reversible Logic," *Proc. 2004 Int. Symposium on Multiple-Valued Logic*, Toronto, Canada, May 2004, pp. 74-80.

[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*. 2007. pp. 537-552.

[9] D. M. Miller and M. A. Thornton, "QMDD: A Decision Diagram Structure for Reversible and Quantum Circuits", *Proc. IEEE International Symposium on Multiple-Valued Logic (ISMVL),* on CD, May 17-20, 2006.

[10] D. M. Miller, M. A. Thornton, and D. Goodman, "A Decision Diagram Package for Reversible and Quantum Circuits", *Proc. IEEE World Congress on Computational Intelligence*, on CD, July 2006.

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

[12] S. Panda and F. Somenzi, "Who are the variables in your neighborhood", In *ICCAD' 95*, December 1995, pp. 1-4.

[13] R. Rudell, "Dynamic variable ordering for ordered binary decision diagrams", In *Proceedings of the International Conference on Computer-Aided Design*, Santa Clara, CA, November 1993, pp. 42-47.

[14] F. Somenzi, *"The CUDD Package"*, University of Colorado at Boulder, 1995. Version 2.4.0 available at: http://vlsi.colorado.edu/~fabio/.

[15] G. F. Viamontes, I. L. Markov, and J. P. Hayes, "QuIDDPro: High-Performance Quantum Circuit Simulation", http://vlsicad.eecs.umich.edu/Quantum/qp/, Oct. 20, 2006.

[16] S. N. Yanushkevitch, D. M. Miller, V. P. Shmerko and R. S. Stankovic, *Decision Diagram Techniques for Micro- and Nanoelectronic Design*, CRC Taylor and Francis, 2006..