

# QMDD-based One-pass Design of Reversible Logic: Exploring the Available Degree of Freedom (Work-in-Progress Report)

Alwin Zulehner and Robert Wille

Institute for Integrated Circuits, Johannes Kepler University Linz, Austria  
alwin.zulehner@jku.at   robert.wille@jku.at

**Abstract.** Research on synthesis of reversible circuits has found substantial consideration in the past. Corresponding methods can be categorized into functional approaches (which often require a prior embedding step) and structural ones (which are often based on mapping). While functional approaches are less scalable and yield circuits with significantly larger costs, structural approaches typically yield circuits where the number of circuit lines is magnitudes above the minimum. Recently, also the idea of a one-pass design flow has been proposed, which aims to overcome the contradictory shortcomings of both approaches by combining the embedding and the synthesis step of the functional design flow. While this yields further opportunities for a more efficient synthesis, the actually available degree of freedom has not fully been explored yet—not to mention fully exploited. In this work-in-progress-report, we are discussing this issue and explore in detail the potential offered by the one-pass design flow. To this end, we consider the implementation of this flow using QMDD-based synthesis as a representative. The conducted investigations provide a more detailed understanding of this recently proposed flow and demonstrate its potential to be exploited in future work.

## 1 Introduction: QMDD-based One-pass Design

The general idea of one-pass design of reversible logic as proposed in [8] is to inherently conduct the embedding during functional synthesis. This way, a certain degree of freedom can be exploited and also the representation may remain more compact. In this work-in-progress-report, we focus on the solution, which yields circuits, where the number of circuit lines is the minimum (denoted as exact solution in [8]). Furthermore, we use QMDD-based synthesis (originally proposed in [4] and recently improved in [6]) as a representative implementation due to its benefits with respect to scalability (despite that, the one-pass design flow can also be realized using any other functional synthesis approach). In this section, we provide a rough overview of the resulting synthesis methodology which is sufficient to follow the discussions conducted in this work-in-progress report (we refer to [8] for a more detailed description). Based on that, the next section eventually describes and illustrates the proposed ideas on how to exploit the degree of freedom provided by this design flow.

In QMDD-based synthesis, a (non-)reversible function  $f : \mathbb{B}^n \rightarrow \mathbb{B}^m$  is represented by means of a  $2^{\max(n,m)} \times 2^{\max(n,m)}$  dimensional permutation matrix (a function matrix in the non-reversible case), which is composed of zeros and ones only. Within this matrix, a 1-entry indicates that an input (column) is mapped to an output (row).

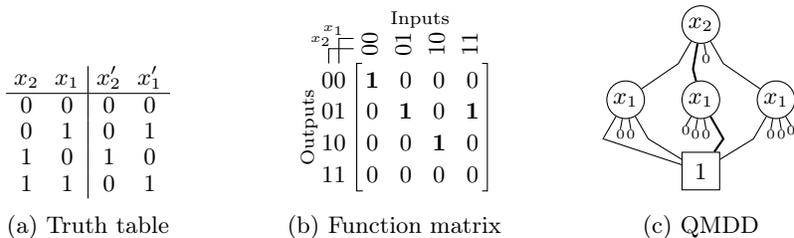


Fig. 1: Representations for a Boolean function

**Example 1** Consider the non-reversible function  $f$  shown in Fig. 1a. The corresponding function matrix is provided by means of Fig. 1b. The 1-entry in the fourth column of the matrix indicates that  $f$  maps input 11 to output 01.

To gain a more compact representation of a function matrix  $M$ , *Quantum Multiple-valued Decision Diagrams* (QMDDs [3]) are utilized. The general idea of QMDDs is to decompose  $M$  variable-wise into sub-matrices.<sup>1</sup> Considering the most significant variable, there are four possible mappings, from 0 to 0 (i.e. the top-left sub-matrix), from 1 to 0 (i.e. the top-right sub-matrix), from 0 to 1 (i.e. the bottom-left sub-matrix), and from 1 to 1 (i.e. the bottom-right sub-matrix). These four mappings provide the basis for a decomposition which is represented by a decision diagram node with four successors (denoting—from left to right—the four sub-matrices outlined above). The decomposition is applied recursively until a single value is reached (represented by a so-called terminal). Since some of the sub-matrices occur frequently, sharing is possible—resulting in a rather compact (non-exponential in most practically relevant cases) representation. For a more detailed description of QMDDs and how to efficiently construct them, we refer to [3].

**Example 1 (continued)** Fig. 1c shows the QMDD-representation of the matrix shown in Fig. 1b. The path highlighted in bold represents the mapping from  $x_2x_1 = 11$  to  $10$ , since it traverses the second edge of the node labeled  $x_2$  (representing the mapping from 1 to 0) and the fourth edge of the node labeled  $x_1$  (representing a mapping from 1 to 1). Note that zero matrices (i.e. matrices composed of zeros only) are visualized by stubs in order to increase readability.

Having a compact function representation (by using QMDDs) allows for a scalable synthesis. By following the exact one-pass design flow (i.e. the flow that yield a circuit where the number of circuit lines is the minimum, cf. [8]), we first determine how many additional variables (i.e. garbage outputs and ancillary inputs) are required.<sup>2</sup> From that we can conclude how many placeholders have to be added such that the function can be realized in a reversible fashion.

From a matrix perspective, it is rather simple to insert placeholders. In fact, assuming that  $k$  variables are added to a matrix  $M$ , all that has to be done is forming the Kronecker product  $M \otimes G$ , where  $G$  is a  $2^k \times 2^k$  matrix with a single 1-entry in its top-left corner. By this, a matrix results where each column that represents an input with any of the ancillary inputs set to 1, contains only

<sup>1</sup> In the following we denote a mapping from input  $x_i$  to output  $x'_i$  by a variable  $x_i$ .

<sup>2</sup> How this can be done efficiently is e.g. discussed in [7].

0-entries—encoding that we actually don't care about the output and ensuring that synthesis can (almost) be conducted as usual.

Using QMDDs, the matrix  $G$  can easily be constructed, since it only contains a single decision diagram node for each additional variable where all edges except the first end in a 0-stub. Moreover, forming the Kronecker product can be formed efficiently by exchanging the terminal node of  $M$  with the root node of  $G$ .

**Example 1 (continued)** *Since the most frequently occurring output pattern (i.e. 01) occurs twice, a single additional variable  $g_1$  is required. The matrix  $G$  as well as the resulting QMDD after forming the Kronecker product are shown in Fig. 2.*

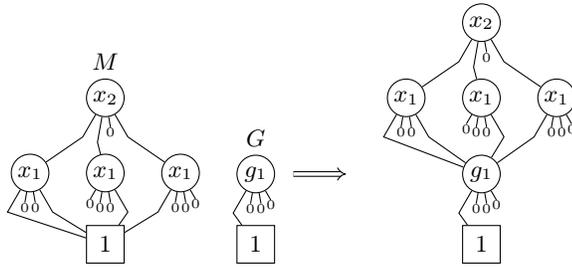


Fig. 2: Insert placeholders

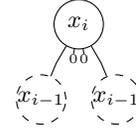


Fig. 3: Identity structure

After extending the function matrix with placeholders, QMDD-based synthesis can (almost) be conducted as usual. The key idea here is to traverse the QMDD in breadth-first manner and thereby transform each visited node to the identity structure shown in Fig. 3 (in which only mappings from 0 to 0 and from 1 to 1 occur). This can easily be accomplished by applying reversible gates. By this, the function to be realized is transformed towards the identity variable by variable.

More precisely, transforming a single QMDD-node to the identity requires the consideration of its successors. To this end, we determine all paths from the currently considered node to the 1-terminal (representing a 1-entry in the matrix) through the respective outgoing edges—resulting in the so-called sets of *1-paths*  $P_1$ ,  $P_2$ ,  $P_3$ , and  $P_4$ . Each path represents an input combination and, thus, contains one literal for each edge it traverses (excluding the edge leaving the currently considered node). Furthermore, we additionally, determine the sets of *0-paths* (i.e. paths ending in a 0-stub) through each outgoing edge, i.e.  $\bar{P}_1$ ,  $\bar{P}_2$ ,  $\bar{P}_3$ , and  $\bar{P}_4$ .

**Example 2** *Consider the root node of the QMDD shown on the right-hand side of Fig. 2. The corresponding sets of 1-paths are  $P_1 = \{\bar{x}_1\bar{g}_1, x_1\bar{g}_1\}$ ,  $P_2 = \{x_1\bar{g}_1\}$ ,  $P_3 = \emptyset$ , and  $P_4 = \{\bar{x}_1\bar{g}_1\}$ . Moreover the sets of 0-paths are  $\bar{P}_1 = \{\bar{x}_1g_1, x_1g_1\}$ ,  $\bar{P}_2 = \{\bar{x}_1\bar{g}_1, \bar{x}_1g_1, x_1g_1\}$ ,  $\bar{P}_3 = \{\bar{x}_1\bar{g}_1, \bar{x}_1g_1, x_1\bar{g}_1, x_1g_1\}$ , and  $\bar{P}_4 = \{\bar{x}_1g_1, x_1\bar{g}_1, x_1g_1\}$ .*

The goal is now to determine a sequence of reversible gates, which swaps the *1-paths* in  $P_2$  with 0-paths from  $\bar{P}_1$  while, at the same time, swapping the

1-paths from  $P_3$  with 0-paths from  $\overline{P}_4$ .<sup>3</sup> This eventually establishes the desired identity structure and, by this, realizes the function in terms of a reversible circuit. The correspondingly required sequence of reversible gates can easily be determined as described in [4] and [8]. However, note that for each gate of the sequence we additionally have to add control lines that represent the path from the root node to the currently considered node in order to avoid that other nodes (that might already establish the identity structure) are affected. Finally, note that the breadth-first traversal is only performed for variables holding a primary output, since the actual value of the garbage outputs does not matter as long as the resulting function is reversible. However, reversibility is always guaranteed since only reversible gates are applied to the circuit during synthesis.

## 2 The Available Degree of Freedom

The previous section reviewed a direct implementation of the one-pass design flow using QMDD-based synthesis. While this already yields substantial improvements with respect to scalability and costs of the resulting circuit (as evaluated in [8]), a significant degree of freedom has not been exploited yet. This is mainly because how to exploit the available degree of freedom in order to achieve the best results regarding these metrics is a complex task since often local optimizations may have global effects. In order to understand them better, this section provides an exploration of the available degree of freedom which serves as basis for future optimizations in the one-pass design flow. In the following, we conduct the explorations with respect to the exploitation of redundancies in paths as well as in nodes.

### 2.1 Exploiting Redundancies in Paths of QMDDs

Usually, there exist several paths from the root node to the currently considered node. The sequence of reversible gates (together with their according additional control lines) that has been determined in order to transform the currently considered node towards the identity structure has to be replicated once for each of these paths—significantly increasing the circuit’s costs. In order to reduce the number of repetitions (i.e. the number of paths) and the number of additionally required control lines (i.e. the number of literals in a path), one can form the *Exclusive Sum of Products* (ESoP) of all paths and apply optimizations such as proposed in [2].<sup>4</sup> This avoids redundancies (in particular sequences of almost identical gates which mainly cancel each other out) and significantly reduces the costs.

The one-pass design flow allows for some degree of freedom that can additionally be exploited here. In fact, the first or the fourth outgoing edge of a QMDD node might point to a 0-stub after establishing the identity structure (since some columns do not contain a 1-entry). Since a 0-stub represents a zero-matrix—indicating that we actually do not care about the outputs for all inputs—there are certain paths that can be considered as *don’t care* in the ESoP minimization algorithms (e.g. in those proposed in [1, 5]).

<sup>3</sup> Note that this is always possible since the additionally inserted variables ensure that  $|\overline{P}_1| \geq |P_2|$  and  $|\overline{P}_4| \geq |P_3|$ .

<sup>4</sup> Note that this is possible if all occurring gates are self-inverse (which is e.g. the case for Toffoli or Fredkin gates).

**Example 3** Consider the QMDD shown in Fig. 4 and assume that the node highlighted in blue is currently considered. This node is reached from the root node by the path  $p_1 = \bar{x}_3\bar{x}_2$ . Furthermore, note that the path  $x_3\bar{x}_2$  terminates in a 0-stub. Since applying any operation to this path does not affect the function represented by this QMDD, we can use the path as don't care condition when optimizing the ESoP. In this case, adding this path to the ESoP allows to reduce the ESoP to  $\bar{x}_3\bar{x}_2 \oplus x_3\bar{x}_2 = \bar{x}_2$ , which has one literal less than the original path to the currently considered node. In contrast, when considering the other node labeled  $x_1$ , it is not beneficial to include the don't care path in the ESoP, which can already be minimized to  $\bar{x}_3x_2 \oplus x_3x_2 = x_2$ .

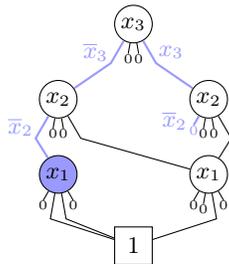


Fig. 4: Exploiting redundancies in paths

However, one shall take the sequence of gates into account that are required to transform the currently considered node to the identity, since this sequence also determines whether to optimize the number of products or the number of literals in the individual products during ESoP-minimization.

## 2.2 Exploiting Redundancies in Nodes of QMDDs

Note that, in QMDD-based synthesis, the sequence of reversible gates that is required in order to transform the currently considered node to the identity is uniquely determined by the sets of 1-paths (and the sets of 0-paths). Moreover, as stated in [6], there might exist different QMDD nodes with equal sets of 1-paths and 0-paths, which allows to consider them jointly when transforming them to the identity. Since the cardinalities of the sets of 1-paths and 0-paths always match (i.e.  $|P_2| = |\bar{P}_1|$ ) for reversible functions, it is clear which nodes can be considered jointly.

In contrast, when using QMDD-based one-pass design, the cardinalities are related by a inequalities  $|P_2| \leq |\bar{P}_1|$  and  $|P_3| \leq |\bar{P}_4|$ . Consequently, one can consider two QMDD nodes jointly, whenever the following criteria are met:

$$\begin{aligned}
 (P_2 \cup P'_2) \cap (P_4 \cup P'_4) &= \emptyset \\
 (P_3 \cup P'_3) \cap (P_1 \cup P'_1) &= \emptyset \\
 |P_2 \cup P'_2| &\leq |\bar{P}_1 \cap \bar{P}'_1| \\
 |P_3 \cup P'_3| &\leq |\bar{P}_4 \cap \bar{P}'_4|
 \end{aligned}$$

However, determining whether it is a good choice to consider two nodes jointly is a complex task. In fact, considering nodes jointly typically requires a more complex sequence of gates to transform the nodes to the identity (since more 1-paths have to be considered). Moreover, considering several nodes jointly means that also the ESoP of all paths from the root node to the respective nodes has to be formed (cf. Section 2.1)—often leading to a more complex “minimized” ESoP. Hence, it might be cheaper in certain cases to treat two QMDD nodes separately—requiring dedicated heuristics to choose which nodes shall be considered jointly.

### 3 Conclusion

Within this work-in-progress-report, we explored the available degree of freedom in QMDD-based one-pass design of reversible logic. By this, we provide a better understanding on how this new design flow can fully be exploited and, by this, provide the basis for future work on how to further improve the one-pass design flow. In fact, initial tests provided in [8] already confirmed that this flow may yield reversible circuits which are significantly cheaper and can be realized in significantly less runtime. The next steps now of course involve implementing the proposed ideas to fully exploit the available degree of freedom as well as conducting thorough evaluations on the resulting methods.

### Acknowledgements

This work has partially been supported by the European Union through the COST Action IC1405.

### References

1. Kozłowski, T., Dagless, E.L., Saul, J.M.: An enhanced algorithm for the minimization of exclusive-or sum-of-products for incompletely specified functions. In: *Int’l Conf. on Comp. Design*. pp. 244–249 (1995)
2. Mishchenko, A., Perkowski, M.: Fast heuristic minimization of exclusive-sums-of-products. In: *Int’l Workshop on Applications of the Reed-Muller Expansion in Circuit Design*. pp. 242–250 (2001)
3. Niemann, P., Wille, R., Miller, D.M., Thornton, M.A., Drechsler, R.: QMDDs: Efficient quantum function representation and manipulation. *IEEE Trans. on CAD of Integrated Circuits and Systems* **35**(1), 86–99 (2016)
4. Soeken, M., Wille, R., Hilken, C., Przigoda, N., Drechsler, R.: Synthesis of reversible circuits with minimal lines for large functions. In: *Asia and South Pacific Design Automation Conf.* pp. 85–92 (2012)
5. Song, N., Perkowski, M.A.: Minimization of exclusive sum-of-products expressions for multiple-valued input, incompletely specified functions. *IEEE Trans. on CAD of Integrated Circuits and Systems* **15**(4), 385–395 (1996)
6. Zulehner, A., Wille, R.: Improving synthesis of reversible circuits: Exploiting redundancies in paths and nodes of QMDDs. In: *Int’l Conf. of Reversible Computation*. pp. 232–247 (2017)
7. Zulehner, A., Wille, R.: Make it reversible: Efficient embedding of non-reversible functions. In: *Design, Automation and Test in Europe*. pp. 458–463 (2017)
8. Zulehner, A., Wille, R.: One-pass design of reversible circuits: Combining embedding and synthesis for reversible logic. *IEEE Trans. on CAD of Integrated Circuits and Systems* **37**(5), 996–1008 (2018)