An Efficient Design Representation of Conservative Reversible Logic Gates

Tyler McCormick, J.W. Bruce, Ph.D.
Department of Electrical and Computer Engineering

1. INTRODUCTION

Computers, which have become a ubiquitous staple of modern society, consume nearly 10% of all the energy produced worldwide. While computing implementation technology has been made more energy efficient over the years, the energy required to operate a gate logically has become an increasingly large proportion of the total energy required. A systematic improvement of this power use would result in significant power savings, which would grow even more appreciable as overall efficiency improves. One method for achieving these power savings would be the use of conservative reversible logic (CRL) gates for system design. However, to date, only a few designs using these types of gates have been developed. This sporadic development is primarily due to the lack of a systematic method for representing CRL gates.

2. OBJECTIVES

- Present two methods of adapting πDDs to represent CRL gates
- Demonstrate examples of design manipulation and analysis

3. BACKGROUND

A Binary Decision Diagram (BDD) is a systematic method for representing digital circuits using a directed acyclic graph [1]. In a typical BDD, a truth table is used to create the diagram, as demonstrated in Fig. 1.

A	B	A•B
0	0	0
0	1	0
1	0	0
1	1	1

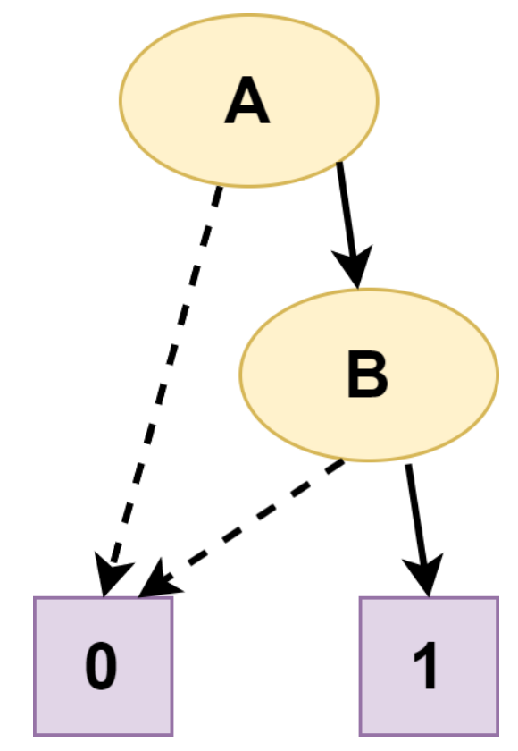

Fig. 1 Truth Table and BDD for AND2 operation

In the BDD, dashed arrows indicates the input it originates from is a '0', while a solid arrow indicates a '1'. Starting at the first input 'A', the arrows are followed depending on the value of the input, and ultimately terminate at either a '0' or a '1', which indicates the output.

While CRL gates can be represented using traditional BDDs, their truth tables grow exponentially with the number of inputs. A CRL gate is ultimately just a permutation of its inputs.

Due to this, the representation of these gates can be condensed using a modified BDD, designed specifically for permutations, known as a πDD [2]. In a πDD, any permutation can be written as an ordered list of two-element swaps on certain items.

For example, if an initial ordered list of items is written as:

$$\pi_e = \{1,2,3\}$$

and a permutation of that list of items is written as:

$$\{(1,3,2)\}$$

The permutation can be as the original list of items followed by a swap of the second and third items, which yields the πDD pictured in Fig. 2.

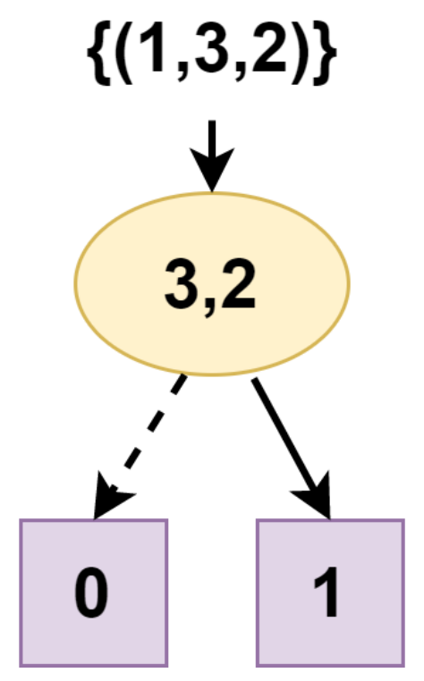

Fig. 2: Example πDD

In a πDD, a solid arrow indicates the swap occurs and the dashed arrow indicates it does not, and the '0' and '1' termination blocks indicate whether the resulting permutation is in the desired list of permutations or not.

4. METHODS

The simplest CRL gate, the Fredkin gate [3], has been well studied and used in large digital designs [4]. However, larger CRL gates have not been systematically examined.

Figs. 3 and 4 show two representative 4-input, 4-output CRL gates and their operation with respect to their "control" inputs, A.

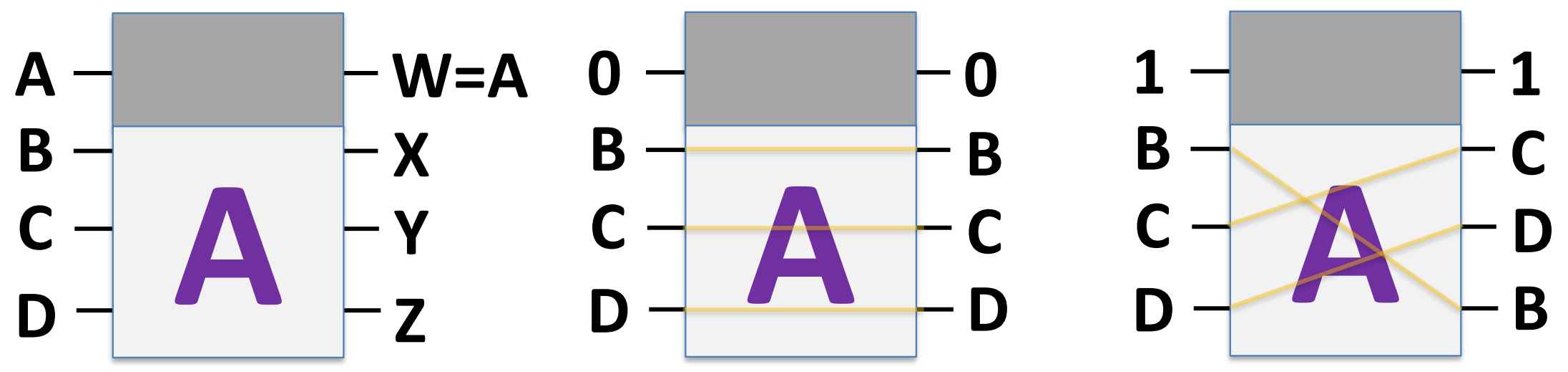

Fig. 3: A 4x4 CRL gate and its two operational states

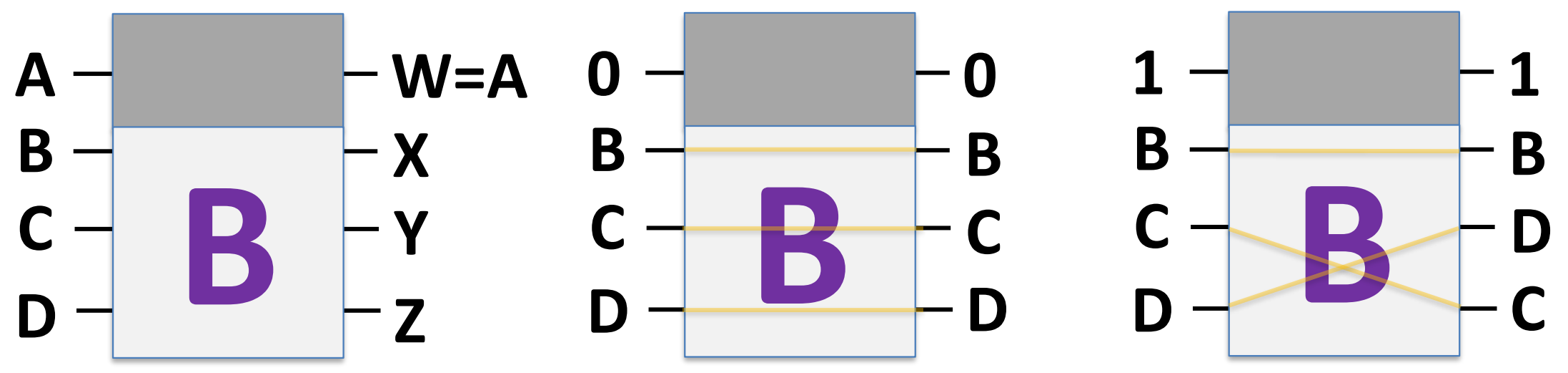

Fig. 4: A 4x4 CRL gate and its two operational states

A. Approach 1

For the first approach, each output column of the original truth table is mapped to a specific permutation, and each permutation is represented in the final πDD. First the control line, column A, is defined as an ordered list of unique items:

$$\pi_e = \{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16\}$$

where items 1-8 are defined as 0's and items 9-16 are defined as 1's. Therefore, the resulting πDD for CRL gate 'A' will contain the four distinct permutations:

$$\{(\pi_e),(1,2,3,4,9,10,13,14,5,6,11,12,7,8,15,16),\\(1,2,9,11,5,6,13,15,3,10,4,12,7,14,8,16),\\(1,9,3,10,5,11,7,12,2,4,6,8,13,14,15,16)\}$$

while CRL 'B' will contain the permutations:

$$\{(\pi_e),(1,2,3,4,9,10,11,12,5,6,7,8,13,14,15,16),\\(1,2,9,11,5,6,13,15,3,10,4,12,7,14,8,16),\\(1,9,3,10,5,13,7,14,2,4,11,12,6,8,15,16)\}$$

and the final πDDs for CRL gates 'A' and 'B', pictured in Fig. 5, can be constructed.

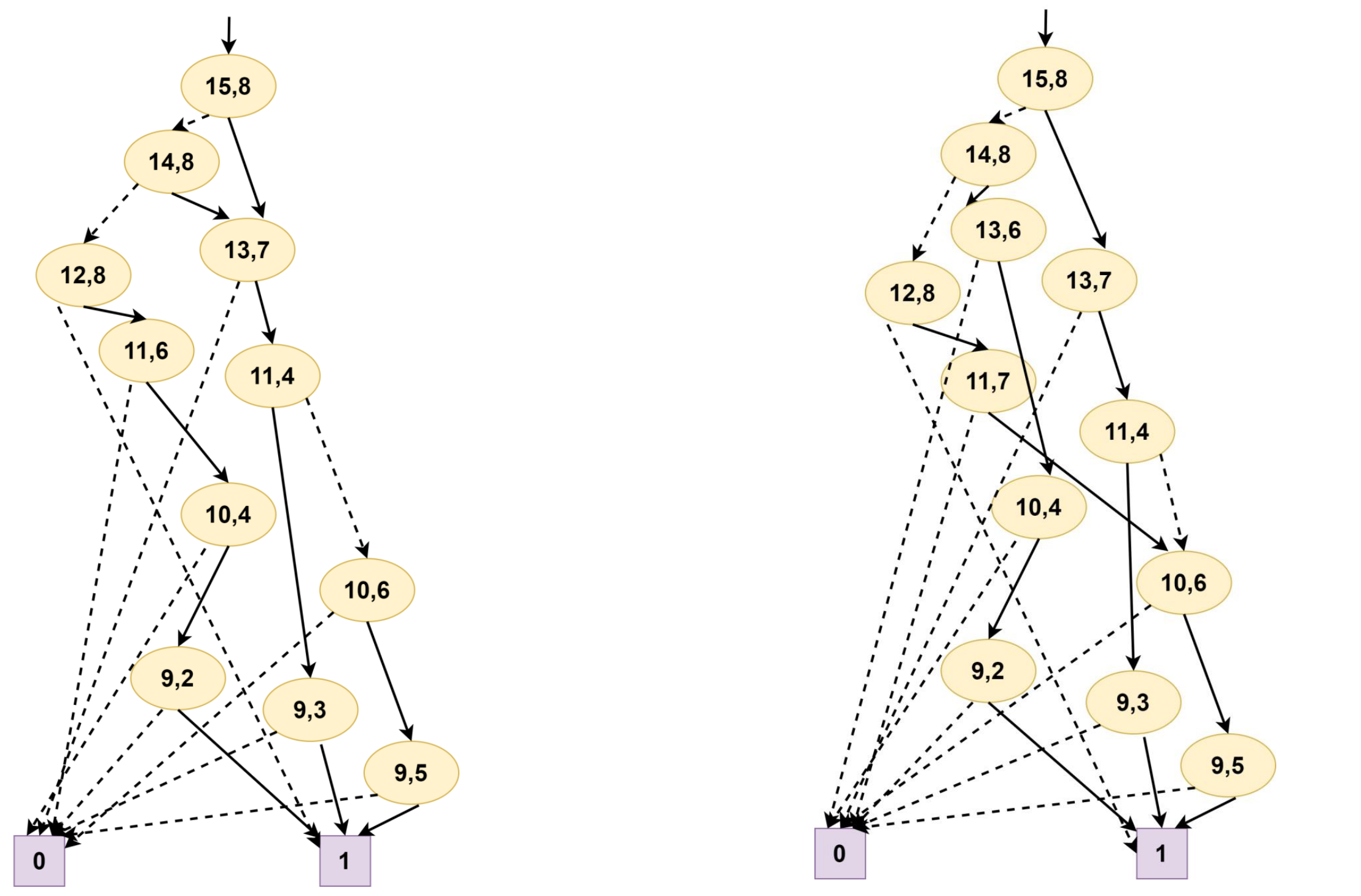

Fig. 5: Representation #1 of 4x4 CRL gates A (left) and B using πDD

B. Approach 2

For the second approach, the desired list of permutations is designated generally as the columns of the truth table, but none are mapped to specific values. Therefore the resulting list of desired permutations for CRL gates 'A' and 'B' respectively is:

$$\{(\pi_e),(1,3,4,2)\} \text{ and } \{(\pi_e),(1,2,4,3)\}$$

and the final πDDs, pictured in Fig. 6, can be constructed.

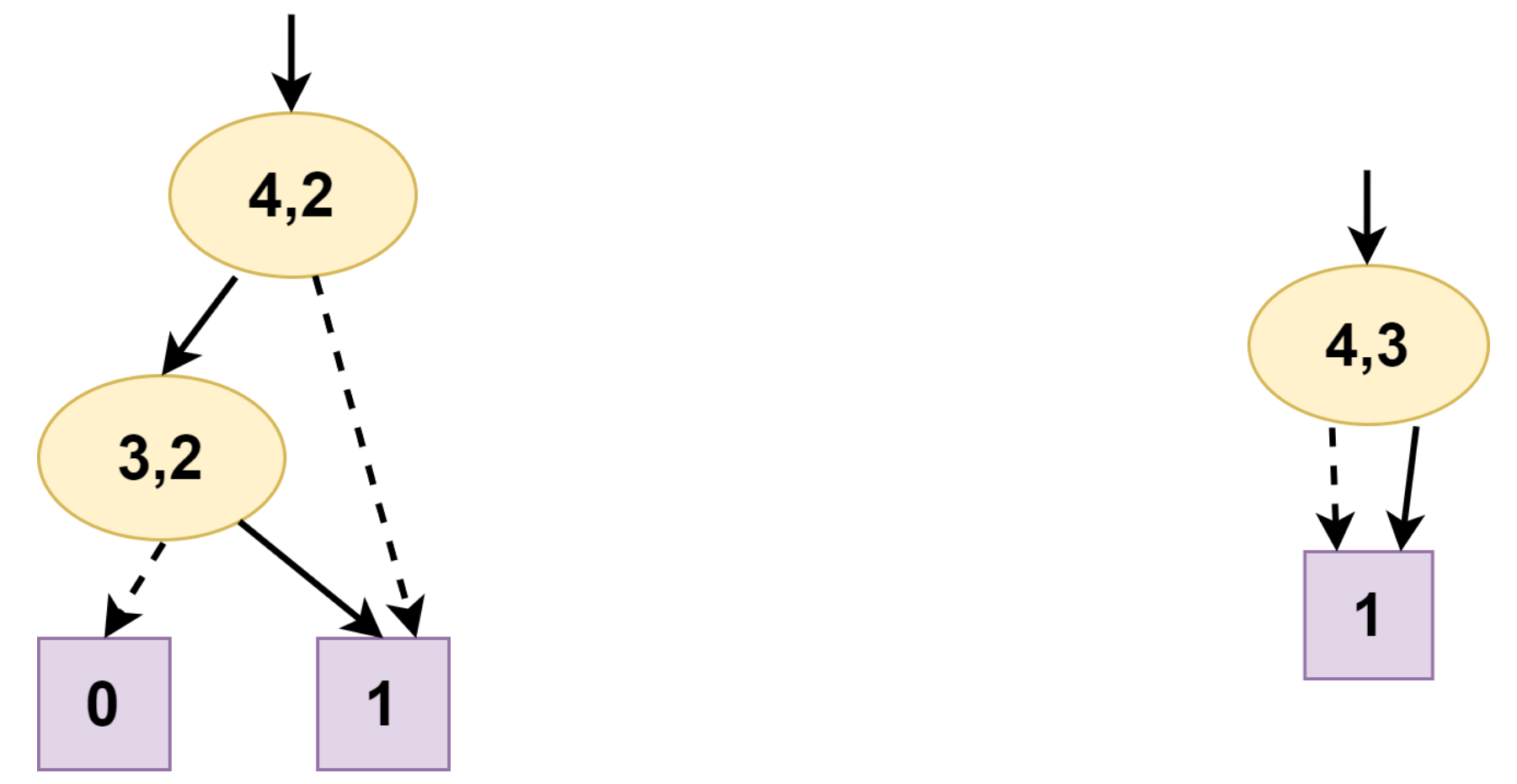

Fig. 6: Representation #2 of CRL gate A (left) and gate B using πDD

Here, the decision to swap the columns in the resulting πDD correlates directly to the value of the control line in the original truth table.

While both approaches efficiently represent CRL gates, there are benefits to both methods

- Approach 1 is a direct synthesis of the truth table in binary, and therefore directly accounts for all possibilities
- Approach 2 is more general, and therefore more efficient than the former, but would require further development to obtain the full truth table

In addition to representing CRL gates, the second πDD representation can represent combinations of connected CRL gates by combining the appropriate πDDs.

5. RESULTS

Using Approach 2 for depicting CRL gates using πDDs, CRL gates 'A' and 'B' are denoted with πDDs in Fig. 6. Using the precise element transposition notation for πDDs, the resulting permutation from 'A' followed by 'B' can be written as the transpositions of CRL gate 'A' followed by the transpositions of CRL gate 'B':

$$\tau(4,3)[\tau(3,2)\tau(4,2)]$$

This series of transpositions can then re-ordered using an algorithm for swapping transpositions [2]. First, the last two transpositions are compared:

$$[\tau(4,3)\tau(3,2)]\tau(4,2)$$

The first two transpositions violate the fixed order of a πDD, and so they are adjusted and swapped:

$$\tau(3,2)\tau(4,2)\tau(4,2)$$

Lastly, the last two transpositions can be eliminated because they are swapping the same two items, yielding the following remaining transposition:

$$\tau(3,2)$$

This remaining transposition, is then depicted as the πDD pictured in Fig. 8.

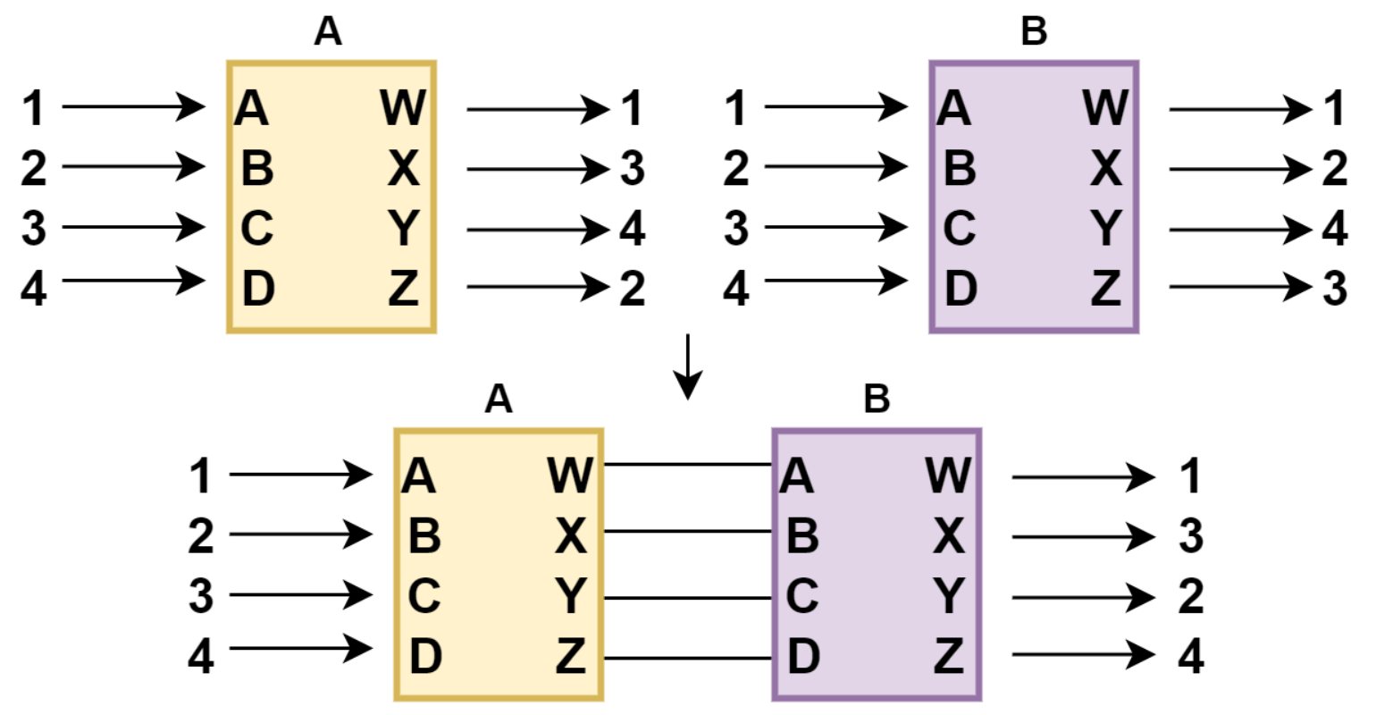

Fig. 7: The 4x4 CRL gate A followed by CRL gate B

This single πDD yields the output column permutation '1324', which is identical to the πDD created from the resulting permutation from cascading CRL gates 'A' and 'B', as depicted in Fig. 7.

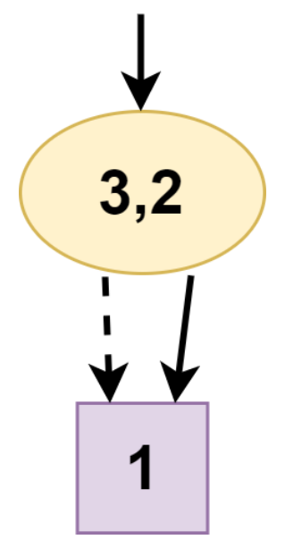

Fig. 8: πDD denoting the cascaded CRL gates in Fig. 6

6. CONCLUSION

Using a πDD, two different methods for representing CRL gates were created and then analyzed. Not only are the methods a compact and efficient way to represent CRL gates, they can also be used for designs consisting of cascaded CRL gates.

7. REFERENCES

[1] R. E. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation," IEEE Transactions on Computers, vol. C-35, no. 8, pp. 677–691, 1986.

[2] S.-I. Minato, "πDD: A New Decision Diagram for Efficient Problem Solving in Permutation Space," Theory and Applications of Satisfiability Testing - SAT 2011 Lecture Notes in Computer Science, pp. 90–104, 2011.

[3] E. Fredkin and T. Toffoli, "Conservative Logic", *Intl. J. Theoretical Physics*, v.21, n.3-4, pp. 219-253, 1982.

[4] J.W. Bruce et al, "Efficient Adder Circuits Based on a Conservative Reversible Logic Gate, *Proc. IEEE ISVLSI 2002*, pp.83-88, 2002.