

# NEM Relay Design with Biconditional Binary Decision Diagrams

Winston Haaswijk, Luca Amarú, Pierre-Emmanuel Gaillardon, Giovanni De Micheli  
Integrated Systems Laboratory (LSI), EPFL, Switzerland.  
Email: winston.haaswijk@epfl.ch

**Abstract**— In this paper, we present an improved design flow for nanoelectromechanical (NEM) relay-based combinational logic circuits. Six-terminal NEM relays can be programmed to act as 2-to-1 multiplexers. We can therefore use NEM relays to implement arbitrary combinational logic circuits. Previously, traditional logic synthesis techniques based on Binary Decision Diagrams (BDDs) have been used to map arbitrary logic functions to NEM relays. We improve this approach by showing how six-terminal relays can also be viewed as 2-to-1 multiplexers fed by comparators. This allows us to create a mapping from Biconditional BDDs (BBDDs) to NEM relays. We then show how it is possible to improve the BDD-based design flow, by presenting a methodology based on BBDD logic synthesis techniques. Experimental results show that our BBDD-based design flow reduces the average number of relays by 24% and the average critical path length by 12%. Considering an  $8 \times 8$  array multiplier with different mechanical delay implementations, we show a 33% average relay count reduction.

## I. INTRODUCTION

Nanoelectromechanical (NEM) relays are electrostatically actuated mechanical switches [1]. NEM relays have a number of desirable properties, such as a very low on-state intrinsic resistance ( $0.5\Omega$ ), and virtually infinitely large off-state resistance. On the other hand, they have several drawbacks as well. For example, they have a long switching time (hundreds of nanoseconds), poor device lifetime ( $10^8$  switching cycles), and limited scalability of the minimum feature size [2], [3]. NEM relays can be fabricated by top-down approaches using conventional lithography techniques or bottom-up approaches using carbon nanotubes or nanowire beams [3].

Different NEM relay structures for logic have been proposed [4], [5]. Most of them are based on electrostatic actuation and implement different logic functions depending on the number of terminals and the device geometry. Mechanical contacts (connections) are enforced via electric fields between the various terminals. Two-terminal and three-terminal NEM relays are simple devices that can be used to solve preliminary process challenges. Four-terminal and six-terminal NEM relays are more complex devices that trade simplicity for functionality. Despite this complexity, the use of multi-terminal devices does not have much impact on area footprint. Hence, they are desirable for compact logic implementations.

The ideal zero leakage current of NEM relays make them a promising alternative to CMOS for ultra low-power systems where low leakage current is a key feature to be harnessed [2], [3], [6]–[8]. Furthermore, multi-terminal NEM relays have increased logic expressivity as compared to traditional MOS

devices. Complete designs that demonstrate the advantages of NEM relays have already been proposed and fully implemented [9]. If some of the technological challenges are overcome, NEM relays have the potential to be superior devices for next-generation ultra low-power systems.

Lee *et al.* [5] realize a six-terminal NEM relay with two body contacts and two source contacts. The two body contacts are designed to be biased by opposite voltages. They show how such a NEM relay can be viewed as a 2-to-1 multiplexer. This is the basis for their design flow based on *Binary Decision Diagrams* (BDDs). BDDs can be implemented in hardware by mapping their nodes to 2-to-1 multiplexers. In [5], circuit designs are broken up into blocks which can be represented by BDDs. The BDDs are then mapped to 2-to-1 multiplexers that are implemented by NEM relays. Since BDDs can represent arbitrary logic functions, this BDD-based methodology can be used to implement arbitrary combinational logic functions using six-terminal NEM relays.

In this paper, we show that viewing six-terminal NEM relays as 2-to-1 multiplexers does not take full advantage of their expressivity. We propose a novel configuration, showing that they can indeed be interpreted as 2-to-1 multiplexers fed by comparators. We use this insight to show that *Biconditional BDDs* (BBDDs) are a native model for six-terminal NEM relays. BBDDs are a compact and canonical alternative to BDDs that can be used to represent arbitrary Boolean functions [10]. Analogous to the mapping in the BDD design flow, BBDD nodes can be mapped to 2-to-1 multiplexers fed by comparators. Our contribution is to show how we can leverage a BBDD-based design flow to obtain more compact circuits using NEM relays. For the MCNC benchmark suite our design flow reduces the average device count and average number of relays on the critical path by 24% and 12%, respectively. Additionally, we reduce the average relay count for different implementations of an  $8 \times 8$  array multiplier by 33%.

The remainder of this paper is organized as follows. In Section II we provide a background on NEM relays and their logic abstraction. Section III gives an overview of the previous work that has been done on BDD-based design flows for NEM relays. In Section IV we show how BBDDs are related to NEM relays and how to use them as the basis for a design flow. Section V contains our experimental results, confirming that BBDDs can be leveraged to design more compact circuits with NEM relays. Finally, Section VI sums up our contributions and reflects on the theoretical and experimental results.

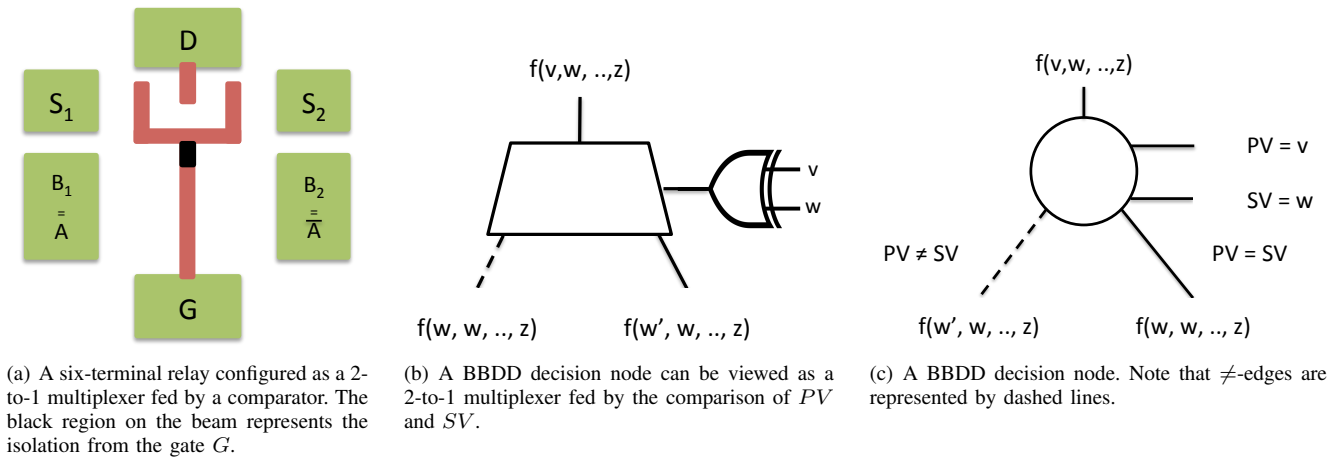


Fig. 1. A six-terminal NEM relay can be configured as a 2-to-1 multiplexer fed by a comparator, enabling a mapping from BBDD nodes to NEM relays.

## II. BACKGROUND

We cover the background on six-terminal NEM relay devices. Other types of NEM relays exist, but they are not as relevant to our extension of logic synthesis for NEM relays. Similarly, we will not cover the fabrication process. More information on different types of relays and fabrication can be found elsewhere [4], [5].

### A. Six-terminal NEM Relays

Fig. 1(a) shows a top-down schematic of a six-terminal relay. It consists of a gate ( $G$ ), a drain ( $D$ ), two bodies ( $B_1$ ,  $B_2$ ) and two sources ( $S_1$ ,  $S_2$ ). The movable part of the relay is the Y-shaped cantilever beam which can move in-plane to either side. By applying voltages to  $G$ ,  $B_1$ , and  $B_2$ , the beam can be actuated bidirectionally to  $B_1$  and  $B_2$ .

The operation of the six-terminal relay can be understood as follows [5]. Assume that the  $G$  and  $B_2$  terminals are grounded. If the  $B_1$  voltage is zero, then the beam remains stationary since it is not attracted to either side. As a result, in this case  $S_1$  and  $S_2$  are isolated from the drain  $D$ . As the  $B_1$  voltage increases, the beam moves toward  $B_1$  due to the increase in the electrostatic field. If the  $B_1$  voltage increases beyond a certain point (the *pull-in voltage*), the electrostatic force overcomes the elastic force that holds the beam into place. This causes the beam to collapse into  $S_1$  and  $D$ , thus connecting them. Similarly, the beam can also be made to connect  $S_2$  and  $D$ .

Note that the beam only moves when the voltages applied to  $B_1$  and  $B_2$  have opposite polarity. If we allow the voltages to be the equal, the beam may either remain stationary or swing randomly. In the first case,  $D$  is disconnected from the other terminals, providing a high-impedance (tristate) output.

### B. Logic Abstraction

We derive the logic behavior corresponding to the operation of a NEM relay. We will see that the output  $D$  is a biconditional function. That is, the output of a NEM relay depends on two variables. In Section IV-B we show how this corresponds to the biconditional expansion in Biconditional BDDs.

In our derivation, we consider only the voltages  $V_{BB}$  and GND, since these are the only voltages required to actuate the relay. Additionally, we only consider the case in which  $B_1 \neq B_2$ , since we are interested NEM relays as devices that implement combinational logic gates. The voltages  $V_{BB}$  and GND correspond to Boolean logic value 1 and 0, respectively. In the remainder of this section, we describe voltages in terms of their corresponding Boolean values.

Suppose that we have logic values  $A$  and  $\bar{A}$ . We can apply these to body terminals  $B_1$  and  $B_2$ , respectively. This configuration is illustrated in Fig. 1(a).

Let us now consider what happens when a voltage is applied to the gate  $G$ . If  $G \neq A$ , the cantilever beam is pulled towards  $B_1$ , thus connecting  $D$  to  $S_1$ . To see why, let us consider the two possibilities. Suppose that  $A = 1$ . Then  $G = \bar{A} = 0$ . Therefore, since  $A$  is larger than the pull-in voltage the beam will collapse into  $B_1$ . This connects the output  $D$  to  $S_1$ . In the other case  $A = 0$ . Then,  $G = \bar{A} = 1$ . Again, in this case the beam will collapse into  $B_1$  due to the electrostatic force overcoming the elastic force. Hence,  $G \neq A$  implies  $D = S_1$ . Using an analogous argument we can show that  $G = A$  implies  $D = S_2$ . The truth table for this configuration is illustrated in Table I.

TABLE I  
TRUTH TABLE OF A SIX-TERMINAL RELAY WHERE  $B_1 = A$  AND  $B_2 = \bar{A}$ .  
IT ACTS AS A 2-TO-1 MULTIPLEXER FED BY A COMPARATOR.

$A$	$G$	$G \odot A$	$D$
0	0	1	$S_2$
0	1	0	$S_1$
1	0	0	$S_1$
1	1	1	$S_2$

The output  $D$  now depends on the equality (or inequality) of  $A$  and  $G$ . By assigning  $B_1 = A$  and  $B_2 = \bar{A}$  to a six-terminal relay, its output equation becomes:

$$D = (G \oplus A) \cdot S_1 + (G \odot A) \cdot S_2 \quad (1)$$

where  $\oplus$  and  $\odot$  represent the XOR and XNOR operations, respectively. Thus, the output  $D$  is biconditional in  $A$  and  $G$ .

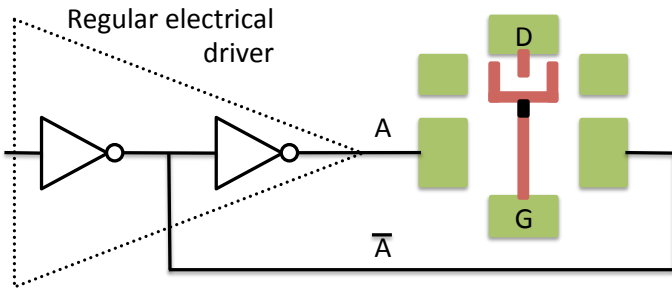


Fig. 2. We use the the electrical drivers of the primary inputs to generate the inverted signal required by the NEM relays.

Hence, when we set the two body terminals  $B_1$  and  $B_2$  to voltage  $A$  and  $\bar{A}$  respectively, we create a comparator between the gate voltage  $G$  and voltage  $A$ . In other words, we now have a 2-to-1 multiplexer that is fed by the comparison of two voltages. This is illustrated in Fig. 1(b). The behavior of such a multiplexer is modeled by a BBDD decision node, pictured in Fig. 1(c). This correspondence between six-terminal NEM relays and BBDD nodes is the inspiration for our BBDD-based design flow.

A caveat to this configuration of NEM relays is that we require the inverted signal for  $\bar{A}$ . With NEM relays, the primary inputs are electrically buffered. We can therefore use the inverter cascade feeding the buffers to generate  $\bar{A}$ , as shown in Fig 2.

Note that NEM relay logic does not require the insertion of a signal buffers between due to the low on resistance of NEM relays. This enables the realization of monolithic logic functions with high performance.

### III. BDD-BASED DESIGN FLOW

We present the previous work on a BDD-based design flow [5]. We first give a brief overview of BDDs in Section III-A. Then, in Section III-B, we show how NEM relays can be configured as 2-to-1 multiplexers. This configuration enables the mapping of BDD nodes to NEM relays, which is the basis for the BDD-based design flow described in Section III-C.

#### A. Binary Decision Diagrams

*Binary Decision Diagrams* (BDDs) are data structures that can be used to represent Boolean functions. The concept of BDDs was first introduced by Lee [11] and Akers [12]. They were later extended by Bryant [13] who showed that, when the ordering and reduction rules are applied, BDDs are a canonical representation for Boolean functions. In other words, if two Boolean functions are equivalent, then they will be represented by exactly the same BDD.

A BDD is a rooted acyclic graph consisting of *decision nodes* and *terminal nodes*. The terminal nodes represent the Boolean values 0 and 1, respectively. Each decision node  $N$  corresponds to a Boolean variable  $\text{Var}_N$  of the Boolean function represented by the BDD. Decision nodes have two children called the *1-child* and the *0-child*. The edge from a node  $N$  to its 0-child represents the assignment of  $\text{Var}_N$  to 0.

TABLE II  
TRUTH TABLE OF A SIX-TERMINAL RELAY WHERE  $B_1 = V_{BB}$  AND  $B_2 = \text{GND}$ . IT ACTS AS A 2-TO-1 MULTIPLEXER.

$G$	$D$
0	$S_1$
1	$S_2$

Similarly, the edge to the 1-child represents the assignment of  $\text{Var}_N$  to 1.

In a BDD, each decision node represents an instance of *Shannon's expansion*:

$$f(v, w, \dots, z) = x \cdot f(1, w, \dots, z) + x' \cdot f(0, w, \dots, z)$$

BDDs can be implemented in hardware by mapping nodes to 2-to-1 multiplexers. To see why, suppose that we have a BDD representing some Boolean function  $f(v, w, \dots, z)$ . The decision node for variable  $v$  can be implemented by a 2-to-1 multiplexer with decision variable  $v$ . The multiplexer's output is connected to  $f(1, w, \dots, z)$  or to  $f(0, w, \dots, z)$ , if  $v = 1$  or  $v = 0$ , respectively. This is illustrated by Fig 3(c). In practice, BDDs are often represented by graphs with 1 terminal node and complemented edges, as this is a more compact representation [14].

#### B. NEM Relays As 2-to-1 Multiplexers

Lee *et al.* [5] show how a six-terminal relay can be used to build a 2-to-1 multiplexer. Suppose that body electrodes  $B_1$  and  $B_2$  are connected to  $V_{BB}$  and GND, respectively, where  $V_{BB}$  is larger than the pull-in voltage. This is illustrated by the schematic in Fig. 3(a). When voltage on gate  $G$  is GND (logic 0), the beam is pulled towards  $B_1$ , because the voltage on  $B_1$  is larger than the pull-in voltage. This connects the output  $D$  to  $S_1$ . Similarly, when the voltage on  $G$  is  $V_{BB}$  (logic 1),  $D$  is connected to  $S_2$ . In other words, the gate  $G$  selects which of the source inputs  $S_1$  and  $S_2$  is connected to output  $D$ . We can therefore express the output  $D$  as the Boolean function:

$$D = \bar{G}S_1 + GS_2. \quad (2)$$

Note that Eq. 2 is the special case of Eq. 1 where  $A = 1$ . In this configuration, the NEM relay no longer corresponds to a biconditional function.

Hence, a six-terminal relay can be viewed as a multiplexer, where  $G$  is the selector input. This is illustrated by Fig.3(b). Table II shows the truth table for this configuration.

#### C. Design Flow

The configuration of NEM relays as 2-to-1 multiplexers enables a mapping from BDD nodes to NEM relays. This approach is used to demonstrate BDD based logic synthesis using NEM relays [5]. Designs are broken up into blocks for which the corresponding BDDs are then created. For each BDD a one-to-one mapping strategy generates a netlist of nanorelays implementing the target logic function.

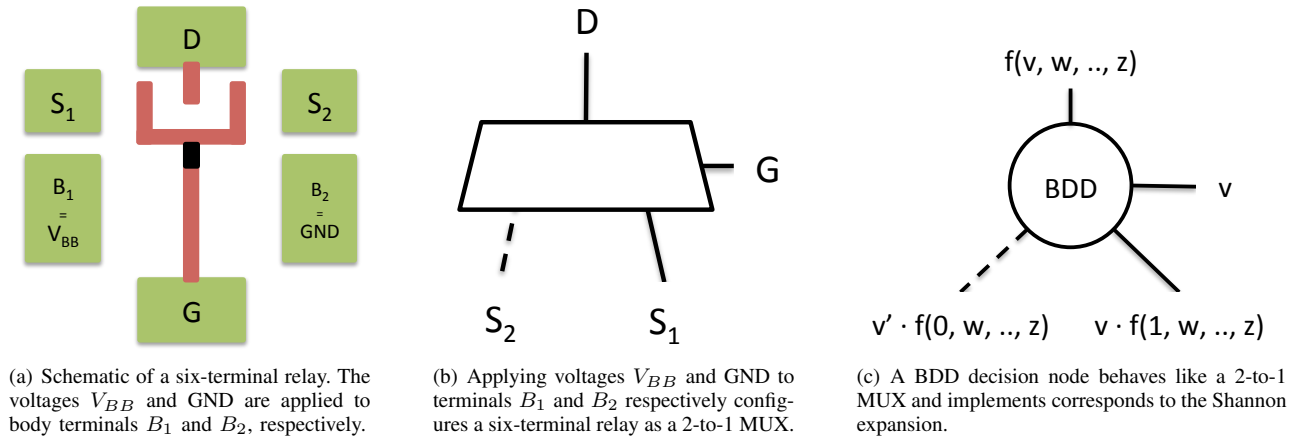


Fig. 3. Six-terminal NEM relays can be configured as 2-to-1 multiplexers. This allows us to map BDD decision nodes to NEM relays.

#### IV. BBDD-BASED DESIGN FLOW

We explain here how BBDD-based logic synthesis techniques can be used to improve on existing design flows for NEM nanorelays. Although the BDD mapping presented above is valid, we know that NEM relays can implement more expressive logic and thus lead to more compact combinatorial circuits. In particular, we have seen that they act as 2-to-1 multiplexers driven by comparators. This makes BBDDs their native model. Therefore, in this section we propose an alternative design flow based on BBDDs.

In Section IV-A we present a short introduction to BBDDs. A more extensive introduction to BBDDs can be found in [10]. Section IV-B then describes our design flow, based on the mapping from BBDD nodes to NEM relays.

##### A. Biconditional BDDs

We have seen how decision nodes in BDDs correspond to the Shannon expansion. In BBDDs, the Shannon expansion is replaced by the biconditional expansion:

$$f(v, w, \dots, z) = (v \oplus w) \cdot f(w', w, \dots, z) + (v \odot w) \cdot f(w, w, \dots, z)$$

Conceptually, BBDDs operate similarly to BDDs. The difference is in the expansion. The branching condition for decision nodes in a BBDD relates to the equality or inequality of two variables. We call these variables the *primary variable* (PV) and the *secondary variable* (SV). Fig. 1(c) shows an example of a BBDD decision node. We refer to  $PV \neq SV$  and  $PV = SV$  edges simply as  $\neq$ -edges and  $=$ -edges.

Recall from section III-A that BDD nodes can be viewed as 2-to-1 multiplexers. In a similar way, BBDD nodes can be viewed as 2-to-1 multiplexers that are fed by comparators. BBDD nodes are indeed the native Boolean logic representation for NEM relays. Conceptually, the output selected by the multiplexer depends on equality of PV and SV. If they are equal, the  $=$ -edge is selected by the multiplexer. Otherwise, the multiplexer selects the  $\neq$ -edge. A schematic of this interpretation is shown in Fig. 1(b).

Previous work has shown that BBDDs are often superior to BDDs as data structures for logic synthesis [15]. They have

been shown to improve synthesis of logic circuits, providing up to 4.4x speed-ups in traditional decision diagram applications, and decreasing the critical path to about 32% of to state-of-the-art BDD based synthesis [10]. These results suggest that a BBDD-based design flow presents opportunities to further exploit the capabilities of NEM relays.

##### B. Design Flow

Our design flow is inspired by the design flow presented in [5]. We break designs up into blocks and represent each block by a BBDD. We can do this since arbitrary logic functions can be represented by a BBDDs. As we have seen above, NEM relays can be programmed to behave as BBDD nodes. This enables us to implement any logic function with six-terminal NEM relays, by mapping the corresponding BBDDs to networks of NEM relays.

The difference in our mapping and the one described in [5] is in how primary inputs are connected to the NEM relay terminals. In [5], only gate inputs  $G$  are connected to primary inputs. In order to program the relays to behave as comparator-fed multiplexers, we map  $B_1$  and  $B_2$  to primary inputs as well.

Suppose we have a BBDD for the logic function  $f(v, w, \dots, z)$ . We have seen that this function can alternatively be written as:

$$(v \oplus w) \cdot f(w', w, \dots, z) + (v \odot w) \cdot f(w, w, \dots, z)$$

This expansion can be mapped to a NEM relay as follows. Let  $PV = v$  and  $SV = w$ , as in a BBDD node. We connect the gate  $G$  to  $v$ . The body terminals  $B_1$  and  $B_2$  are connected to  $w$  and  $w'$ , respectively. To the output terminals  $S_1$  and  $S_2$ , we connect NEM relay outputs that represent  $f(w', w, \dots, z)$  and  $f(w, w, \dots, z)$ . This connection is made by recursively representing biconditional expansions of the function with different NEM relays. As we have seen above, by connecting the six-terminal relay in this way, its output  $D$  is now represented by the equation:

$$\begin{aligned} D &= (G \oplus A) \cdot S_1 + (G \odot A) \cdot S_2 \\ &= (v \oplus w) \cdot f(w', w, \dots, z) + (v \odot w) \cdot f(w, w, \dots, z) \end{aligned}$$

TABLE III

TOTAL NUMBER OF RELAYS, THE NUMBER OF RELAYS ON THE CRITICAL PATH, AND RATIOS COMPARED TO [5] (MCNC BENCHMARK CIRCUITS).

Circuit Name	Nr. of Relays	Nr. of Relays on the Critical Path	Ratio BBDD/BDD Nodes	Ratio (Critical Path)
alu4	599	14	0.77	1.00
apex4	992	8	0.90	0.89
des	3130	18	0.78	1.00
ex1010	1047	10	0.94	0.91
ex5p	283	8	0.92	1.00
misex3	846	14	1.29	1.00
pdc	865	14	0.35	0.88
spla	691	16	0.82	1.00
8-b adder	28	9	0.19	0.53
16-b adder	56	17	0.31	0.52
$8 \times 8$ multiplier	14094	16	1.05	1.00
Average	2057.36	13.09	0.76	0.88

Thus, we map the nodes in our BBDD representation to NEM relays.

## V. EXPERIMENTAL RESULTS

We demonstrate how BBDDs-based logic synthesis techniques can be used to improve combinational circuits using NEM relays.

### A. Methodology

Our BBDD design flow is described above. We apply it by synthesizing NEM relay circuits with an online BBDD package [16]. We modified the software package to use 0/1 terminal nodes instead of complemented edges. This gives us a similar construction procedure as [5]. The size of the BBDDs then tells us how many NEM relays would be necessary to implement the design. To evaluate the advantage of using BBDD-based synthesis for NEM relays, we compare our results with those obtained in [5].

### B. Comparison: MCNC Benchmark Circuits

We test our design flow against the MCNC benchmark suite. Table III shows the number of relays and the number of relays on the critical path. It compares these numbers with the corresponding numbers in [5] and shows the BBDD to BDD ratio for the different benchmark circuits. We also provide the ratios for the number of relays on the critical path.

Previous work has shown that BBDDs are generally more compact representations than BDDs [10]. Since our BBDDs representations are generally more compact, we expect that they can be mapped to a smaller number of NEM relays. Table III confirms that this is indeed the case.

Our design flow results in an average reduction in NEM relays of 24%. This is due to the compactness of the BBDD representation relative to the BDD representation. Since BBDDs require fewer nodes than BDDs, our circuits require fewer NEM relays.

Furthermore, our design flow enables us to obtain circuits with shorter critical paths. On average the critical path length

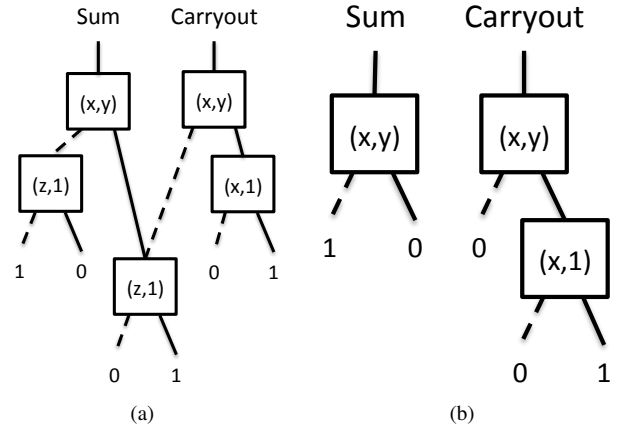


Fig. 4. BBDD representations for a full-adder and a half-adder are illustrated in (a) and (b), respectively. The primary inputs are  $x$ ,  $y$ , and  $z$ .

is reduced by 12%. This decrease in the critical paths is due to the BBDD reduction rules, which can be leveraged to decrease the height of BBDDs more than the reduction rules for their BDD counterparts.

Note that, in this experiment, all designs are represented by monolithic decision diagrams. This implies that all designs have just one mechanical delay. By obtaining a shorter critical path, our design flow reduces the electrical delay, which is in this case the only way of reducing the overall delay. In general, we treat the mechanical delay of a design as part of its specification. Given a design with a specific mechanical delay, we are interested in minimizing the number of relays and the critical path. Our BBDD representation achieves both.

There are some results that warrant discussion. The *misex3* benchmark requires 29% more relays in our design flow than its counterpart in [5]. Similarly, the  $8 \times 8$  multiplier uses 5% more relays. The size of decision diagrams is significantly impacted by the input variable ordering. Therefore, we account for the increase in the number of relays with an unfavorable variable ordering in our experiments.

We do not give a comparison for all the run times for the various benchmarks, as we did not have access to the same hardware to run our experiments. We will just state our worst case runtime, which occurs with the  $8 \times 8$  multiplier. BBDD generation for that benchmark takes 1.09 seconds to generate a BBDD of 15447 nodes. Size optimization to 14094 nodes then takes an additional 6.64 seconds. We are able to handle circuits with more than 10k nodes in approximately the same time. This demonstrates the scalability of BBDD-based synthesis.

### C. Comparison: $8 \times 8$ Array Multiplier

We now compare our approach in the case of synthesizing an  $8 \times 8$  array multiplier. This multiplier, described in more detail in [5], is implemented using a carry-save adder tree followed by a ripple carry adder. We represent the same multiplier, but using BBDDs instead of BDDs. We show that this compact representation allows us to implement the multiplier using a smaller number of NEM relays.

TABLE IV  
COMPARISON OF BDD-BASED VS. BBDD-BASED SYNTHESIS OF AN  $8 \times 8$  ARRAY MULTIPLIER

Number of Mechanical Delays	Number of Six-Terminal Relays in BBDD-Based Implementation	Number of Six-Terminal Relays in BDD-Based Implementation	Ratio BBDD/BDD
1	8890	16352	0.54
2	2491	4129	0.60
3	1367	2186	0.63
4	647	875	0.74
5	434	590	0.74
6	407	533	0.76
Average	2372.67	4110.83	0.67

The carry tree is implemented by a tree of full-adders and half-adders represented by BDDs. Fig. 4 shows how we can represent full- and half-adders using BBDDs. We implement a BBDD full-adder using 5 BBDD nodes, and a half-adder using 3 nodes. In [5] equivalent adders are implemented with BDDs using 8 nodes and 4 nodes, respectively. This means that using a BBDD representation we save 3 nodes for every full-adder and 1 nodes for every half-adder. This results in an area reduction of 37.5% for full-adders and 25% for half-adders.

The various stages of the multiplier can be merged in order to decrease the mechanical delay. We merge the stages in a similar manner as in [5], but using BBDDs instead of BDDs. The relay count for the multiplier at various numbers of mechanical delay is shown in Table IV.

The results show that our compact BBDD representation indeed requires a smaller number of relays. Over the different mechanical delay targets, we achieve an average relay count reduction of 33%. Furthermore, as the number of mechanical delays decreases, so does the ratio of the number of relays required by the BBDD representation versus the BDD representation. This is due to the fact that the BBDD representation is able to more efficiently share sub-functions, as can be seen in the case of the full-adder implementation. It therefore grows more slowly as more functions are merged in.

Four-terminal relays can be biased to act as n-type and p-type switches. This is used in [5] to compare BDD-based designs to their corresponding CMOS-style designs. In particular, this is done for the  $8 \times 8$  multiplier, showing that a BDD-based implementation requires fewer NEM relays than a CMOS-style implementation. We use the same approach as [5] to compare our BBDD-based designs to their CMOS-style counterparts. On average, our BBDD implementations require 88% fewer devices than the CMOS-style implementations.

## VI. CONCLUSION

In this paper, we have shown how BBDD-based logic synthesis techniques can be used to create more compact combinational circuits using six-terminal NEM relays. We have done so by showing that BBDDs nodes are a native model for NEM relays. This enables a mapping from BBDD nodes

to NEM relays. Generally, BBDDs can represent arbitrary logic functions more compactly than BDDs can. By mapping BBDDs to NEM relays we can use this compact representation to synthesize combinational circuits. This results in circuits that use smaller numbers of NEM relays as compared to BDD-based synthesis. For the MCNC benchmarks our BBDD-based design flow allows us to reduce the relay count by 24% on average. Furthermore, we reduce the average critical path length by 12%. Finally, for different mechanical delay targets, we reduce the average relay count for an  $8 \times 8$  array multiplier by 33%.

## ACKNOWLEDGMENT

This research was supported in part by SNSF-200021-146600 and ERC-2009-AdG-246810.

## REFERENCES

- [1] K. Akarvardar and H. Wong, "Nanoelectromechanical Logic and Memory Devices," *ECS transactions*, vol. 19, pp. 49–59, 2009.
- [2] O. Y. Loh and H. D. Espinosa, "Nanoelectromechanical contact switches," *Nature Nanotechnology*, vol. 7, no. 5, pp. 283–295, 2012.
- [3] D. Tsamados, A. M. Ionescu, K. Akarvardar, H.-S. Philip Wong, E. Alon, and T.-J. King Liui, "Nano-Electro-Mechanical Switches," pp. 1–16, 2008.
- [4] M. Spencer, F. Chen, C. C. Wang, R. Nathanael, H. Fariborzi, A. Gupta, H. Kam, V. Pott, J. Jeon, T. J. K. Liu, D. Marković, E. Alon, and V. Stojanović, "Demonstration of integrated micro-electro-mechanical relay circuits for VLSI applications," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 308–320, 2011.
- [5] D. Lee, W. S. Lee, C. Chen, F. Fallah, J. Provine, S. Chong, J. Watkins, R. T. Howe, H. S. P. Wong, and S. Mitra, "Combinational Logic Design Using Six-Terminal NEM Relays," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 5, pp. 653–666, 2013.
- [6] V. Pott, H. Kam, R. Nathanael, J. Jeon, E. Alon, and T.-J. K. Liu, "Mechanical Computing Redux: Relays for Integrated Circuit Applications," *Proceedings of the IEEE*, vol. 98, no. 12, pp. 2076–2094, 2010.
- [7] P. Sharma, J. Perruisseau-Carrier, C. Moldovan, and A. M. Ionescu, "Electromagnetic Performance of RF NEMS Graphene Capacitive Switches," *Nanotechnology, IEEE Transactions on*, vol. 13, no. 1, pp. 70–79, Jan. 2014.
- [8] D. Weinstein and S. a. Bhave, "The resonant body transistor," *Nano Letters*, vol. 10, no. 4, pp. 1234–1237, 2010.
- [9] H. Fariborzi, F. Chen, V. Stojanović, R. Nathanael, J. Jeon, and T. J. K. Liu, "Design and demonstration of micro-electro-mechanical relay multipliers," *Proceedings of Technical Papers: IEEE Asian Solid-State Circuits Conference*, pp. 117–120, 2011.
- [10] L. Amaru, P.-E. Gaillardon, and G. D. Micheli, "Biconditional Binary Decision Diagrams : A Novel," *IEEE Journal on Emerging and Selected Topics in Circuits And Systems (JETCAS)*, vol. 4, no. 4, pp. 487–500, 2014.
- [11] C. Lee, "Representation of Switching Circuits by Binary-Decision Programs," *Bell Systems Technical Journal*, vol. 38, pp. 989–999, 1959.
- [12] S. B. Akers, "Binary Decision Diagrams," *IEEE Transactions on Computers*, vol. C-27, no. 6, pp. 509–516, 1978.
- [13] R. E. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation," *IEEE Transactions on Computers*, vol. C-35, no. 8, pp. 677–691, 1986.
- [14] "CUDD Package." [Online]. Available: <http://vlsi.colorado.edu/fabio/CUDD/>
- [15] L. Amaru, P.-E. Gaillardon, and G. De Micheli, "Biconditional BDD: A Novel Canonical BDD for Logic Synthesis Targeting XOR-rich Circuits," *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2013*, no. c, pp. 1014–1017, 2013.
- [16] "BBDD Package." [Online]. Available: <http://lsi.epfl.ch/BBDD>