BDD-based Synthesis for All-optical Mach-Zehnder Interferometer Circuits

Eleonora Schönborn[1] Kamalika Datta[2]
Robert Wille[1,3] Indranil Sengupta[4] Hafizur Rahaman[5] Rolf Drechsler[1,3]
[1]Institute of Computer Science, University of Bremen, 28359 Bremen, Germany
[2]National Institute of Technology, Meghalaya, Shillong 793003, India
[3]Cyber-Physical Systems, DFKI GmbH, 28359 Bremen, Germany
[4]Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur 721301, India
[5]Department of Information Technology, Bengal Engineering & Science University, Shibpur, Howrah 711103, India
{eleonora,rwille,drechsle}@informatik.uni-bremen.de, kdatta@nitm.ac.in, isg@iitkgp.ac.in, rahaman_h@it.becs.ac.in

***Abstract*—With the advancements in fabrication technology and the emergence of very high performance systems in VLSI, the interest for optical interconnects and optical functional on-chip units increased significantly. *Mach-Zehnder Interferometer* (MZI) switches based on *Semiconductor Optical Amplifiers* (SOAs) have been used as optical building blocks and allowed the synthesis of important Boolean functions such as multiplexers or adders. However, no automatic synthesis approach for arbitrary Boolean functions has been proposed yet. In this work, we introduce such a scheme. For this purpose, we make use of *Binary Decision Diagrams* (BDDs). A technology library is proposed where all possible BDD node configurations are identified and associated with corresponding all-optical sub-circuits. This library is utilized in order to map a BDD representing an arbitrary function into an all-optical circuit using a linear-time algorithm. Experimental evaluations confirm that this leads to an efficient realization of the considered functions.**

I. Introduction

The last decades have witnessed impressive developments of the semiconductor technology resulting in smaller and faster devices which allow Moore's law [2] to remain valid at least until the near future. However, with developments into deep submicron technologies and the ever-increasing emphasis on low-power designs, alternative directions are additionally investigated. In this regard, optical interconnects and optical functional on-chip units received significant interest in the recent years [3]. They allow for ultra-high-speed networks while having beneficial low-power properties. VLSI chips with ultra-fast optical interconnects have already been announced and will enter the market soon [4].

In such systems, the respective signals need to be transformed from the electrical domain to the optical domain and vice versa at every interconnect interface. This causes additional overhead and costs to be avoided. Hence, researchers and engineers are aiming for performing as many computations as possible (particularly signal coding but also various other functionality) within the optical domain. This motivated the design of logic sub-systems using all-optical technologies [5], [6], [7]. Here, optical circuits composed of *Mach-Zehnder Interferometer* (MZI) together with *Semiconductor Optical Amplifiers* (SOAs) have been proven to be a suitable implementation of such functionality [8].

Consequently, significant works on the logic design of optical circuits for important Boolean functions such as multiplexers, adders, universal logic blocks, etc. have been conducted in the recent past (see e.g. [9], [8], [10], [11], [12], [13], [6]). However, all these are ad hoc solutions in the sense that each of them

- has mainly been derived manually for a special purpose,
- addresses a very specific type of functional block,
- is often not extendable to arbitrary functions, and
- is typically suitable for small functions only, i.e. is limited with respect to its scalability.

In fact, no automatic and scalable synthesis approach for arbitrary Boolean functions has been proposed yet.

In this work, we aim for advancing the logic design of all-optical circuits by addressing this problem. We propose an automatic synthesis approach that can be applied to arbitrary functions and is scalable with respect to the number of inputs. The method generates a *Binary Decision Diagram* (BDD) [1] which efficiently represents the function to be synthesized. Then, by utilizing a technology library which provides a corresponding sub-circuit for each node of the BDD, a mapping scheme is applied which transfers the BDD into an all-optical circuit composed of SOA-based MZI switches.

By this, circuits for this technology can be generated automatically for the first time. A discussion and an experimental evaluation provide further insights into the properties of the resulting circuits. In fact, the size of the obtained circuits is bounded by the BDD size which has been heavily investigated from a theoretical perspective. Using the proposed approach, these findings can be transferred to the domain of optical circuits. The experimental results allow for a further evaluation of the benefits and drawbacks of the proposed scheme.

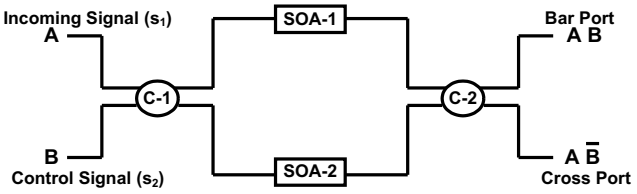
The rest of the paper is organized as follows. Section II provides the background on all-optical technologies with special emphasis on SOA-based MZI implementations, the logic design of these circuits, and BDDs as the applied data structure. Section III introduces and describes the proposed synthesis approach in detail, while Section IV discusses the resulting scheme from a theoretical point of view. Finally, Section V summarizes the results of our experimental evaluation and Section VI concludes the paper.

II. Background

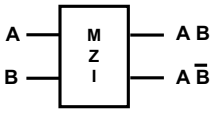
This section briefly reviews the background on optical circuits composed of SOA-based MZI gates and provides an overview of previously conducted logic design for this kind of circuits. Besides that, the basics on BDDs, the main data structure for the proposed synthesis approach, are reviewed.

A. Optical Circuits Composed of SOA-based MZI Switches

A *Mach-Zehnder Interferometer* (MZI) switch based on *Semiconductor Optical Amplifiers* (SOAs) is a technologically improved optical device which has the ability to identify the relative phase-shift difference between light rays and, hence, can be analyzed for various operations based on amplitude as well as phase. It is one of the vital configurable instruments in optical computing [13], [7]. In high capacity optical networks,



(a) Semiconductor Optical Amplifier based MZI



(b) Functional Behavior of MZI Switch

Fig. 1: SOA-based MZI Switch

SOA-based MZI switches have shown their potential in amplifying signals. They further have inherent advantages with respect to fabrication, compressed size, high quality, thermal stability, and fast switching time.

Two SOAs and two couplers are required to construct an all-optical MZI switch. Due to optical pumping, SOAs typically have a variation in carrier density which in turn changes the refractive index. This results in a cross-phase modulation. A coupler on the other hand is a passive optical component which can perform two basic operations, either splitting a signal or combining multiple signals depending on the desired requirements.

The schematic diagram of an MZI switch is shown in Fig. 1 (providing both, a technological as well as a functional description). Each MZI switch has two input ports and two output ports. At the input ports, the optical signal at port A is termed as incoming signal s_1 and the optical signal at port B is called the control signal s_2. The output ports are known as bar port and cross port. A band pass filter is placed at the output which rejects the control signal and allows to pass the wavelength s_1.

By this, the MZI switch works as follows:

- When there is a presence of an incoming signal s_1 at the input port A and a control signal s_2 at port B, then we get the presence of light at the output bar port and no light at the output cross port.
- When the control signal s_2 is absent at port B and the incoming signal s_1 is present at port A, then we observe the presence of light at the output cross port and no light at the output bar port.

Presence and absence of light is represented by binary 1 and 0, respectively. Hence, the functional behavior of the MZI switch can be expressed in terms of Boolean algebra as

$$\text{Bar port} = AB, \text{ and} \quad (1)$$

$$\text{Cross port} = A\overline{B} \quad (2)$$

MZI gates as introduced above represent a universal gate library. That is, any Boolean function can be realized by a circuit composed of these gates. Accordingly, researchers and engineers started to investigate how to design practically relevant functions using this technology.

B. Logic Design for MZI Circuits

Various works have been reported in the literature on building all-optical logic gates (see e.g. [9], [8], [14], [15], [16], [17], [6]) as well as all-optical functional blocks like adders, comparators, multiplexers, flip-flops, etc. (see e.g. [18], [19], [20], [21], [5], [10], [11], [12], [13]).

More precisely, an all-optical XOR gate has been proposed and experimentally validated using SOA-based MZI switches in [6]. The result of the XOR operation is numerically analyzed by solving the rate equation of the SOA. In [8], all-optical implementations of reversible gates using MZI switches are proposed, i.e. so-called CNOT and Toffoli gates are constructed. In [14], an all-optical implementation of a reversible NOR gate is proposed. Larger building blocks have also been considered. For example, [18], [19], [20] provide proposals for all-optical designs of adders. In [21] and [5], a multiplexer design and a flip-flop realization based on MZI switches are proposed, respectively.

However, while all these investigations yielded building blocks which are essential for the design of practically relevant functions, no automatic and scalable synthesis approach for arbitrary Boolean functions has been proposed yet. In this work, this problem is addressed. For this purpose, binary decision diagrams are applied which are briefly reviewed next.

C. Binary Decision Diagrams

A Boolean function $f : \mathbb{B}^n \rightarrow \mathbb{B}$ can be represented by a *Binary Decision Diagram* (BDD) [1]. A BDD is a directed acyclic graph $G = (V, E)$ where a Shannon decomposition

$$f = \overline{x}_i f_{x_i=0} + x_i f_{x_i=1} \quad (1 \leq i \leq n)$$

is carried out in each node $v \in V$. The functions $f_{x_i=0}$ and $f_{x_i=1}$ are the *co-factors* of f obtained by assigning x_i to 0 or 1, respectively. Other decompositions might be applied additionally for a more compact function representation; e.g. positive Davio and negative Davio in a *Kronecker Functional Decision Diagram* (KFDD) [22]. In the proposed approach, KFDDs are used. However, for the sake of simplicity, all descriptions and discussions will focus on BDDs.

In the following, the node representing $f_{x_i=0}$ ($f_{x_i=1}$) is denoted by $low(v)$ ($high(v)$) while x_i is called the *select variable*. A BDD is called free if each variable is encountered at most once on each path from the root to a terminal node. A BDD is called ordered if all variables are encountered in the same order on all such paths. In the following, ordered binary decision diagrams are called BDD for brevity. The *size* k of a BDD is defined by the number of nodes.

In the past, several techniques to optimize the size of BDDs have been developed. Especially *shared nodes* [1], i.e. nodes employing more than one predecessor, allow significant reductions. In particular, functions $f : \mathbb{B}^n \rightarrow \mathbb{B}^m$ (i.e. functions with more than one output) can be represented more compactly using shared nodes. Further reduction can be achieved if *complement edges* [23] are applied. This enables the representation of a function as well as of its negation by a single node only. Furthermore, the size of a BDD significantly depends on the chosen ordering of its input variables [1].

Example 1: Fig. 2 shows a BDD representing the function $f = \overline{x}_1\overline{x}_2\overline{x}_3x_4 + \overline{x}_1x_2x_3\overline{x}_4 + x_1\overline{x}_2x_3\overline{x}_4 + x_1x_2\overline{x}_3x_4$ as well as the respective co-factors resulting from the application of the Shannon decomposition.

III. Proposed Synthesis Approach

In this section, we introduce the proposed synthesis approach which generates an all-optical circuit from a given functional specification. First, the general idea and the overall synthesis flow are sketched. Afterwards, details on the implementation are provided.

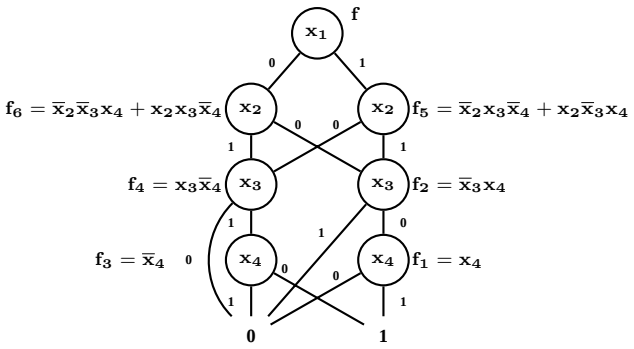


Fig. 2: Binary Decision Diagram (BDD)

A. General Idea

The aim of the proposed approach is to efficiently determine an optical circuit composed of SOA-based MZI switches for an arbitrary Boolean function. It is well known that Boolean functions can efficiently be represented by BDDs [1]. Hence, we propose to use this data structure for the purpose of synthesis. The general idea here is as follows[1]: Given a BDD $G = (V, E)$ representing the function f to be synthesized, an optical circuit can be derived by traversing the decision diagram and substituting each node $v \in V$ with a corresponding sub-circuit. By properly connecting the obtained sub-circuits, the desired circuit realizing f results.

However, in order to employ this scheme, the following issues need to be addressed:

- A BDD representing the desired function f has to be generated. This can be done efficiently using state-of-the-art BDD packages (e.g. CUDD [26]).
- MZI realizations for all possible configurations of the BDD nodes must be available. This requires a corresponding technology library.
- A mapping scheme is required which properly maps each node to the corresponding sub-circuit.

This eventually leads to a synthesis flow as shown in Fig. 3. In the following, the two main steps of this flow, i.e. the generation of the technology library as well as the mapping procedure, are described in detail.

B. Technology Library

The technology library provides sub-circuits composed of SOA-based MZI switches representing the functional behavior of the possible BDD node configurations and, by this, provides building blocks for the proposed synthesis scheme. Determining the library is crucial as it significantly affects the quality of the obtained results. Moreover, the library must provide building blocks for *all* possible BDD node configurations. Hence, completeness and compactness are the most important considerations when creating the library.

A BDD node $v \in V$ may occur in different configurations, i.e. with and without terminal successors as well as with and without complement edges. Considering all possibilities, 13 distinct cases have been identified (15 cases for KFDD nodes). The most important ones are shown in the first column of Table I[2]. For all of these cases, MZI sub-circuits as show in the second column of Table I have been generated.

[1]A similar idea has been applied before e.g. for the synthesis of conventional as well as reversible circuits in [24] and [25], respectively.

[2]The remaining cases only differ slightly from the configurations already listed in Table I and are not relevant for the understanding of the proposed synthesis scheme. For the sake of clarity, they are omitted.

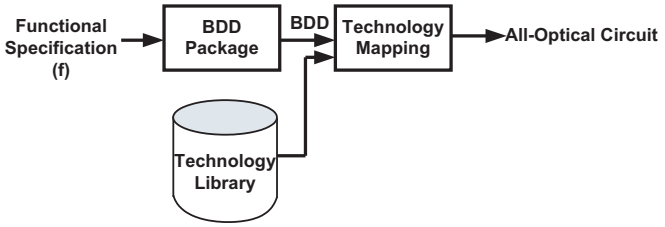


Fig. 3: Overall Synthesis Flow

Besides MZI switches, *Beam Splitters* (BS; denoted by black dots) are applied to branch (fan-out) optical signals which are needed more than once. *Beam Combiners* (BC; denoted by white dots) are applied to combine optical signals, corresponding to a logic disjunction. Overall, this provides all building blocks needed to realize the proposed synthesis scheme.

C. Mapping BDD to MZI Circuit

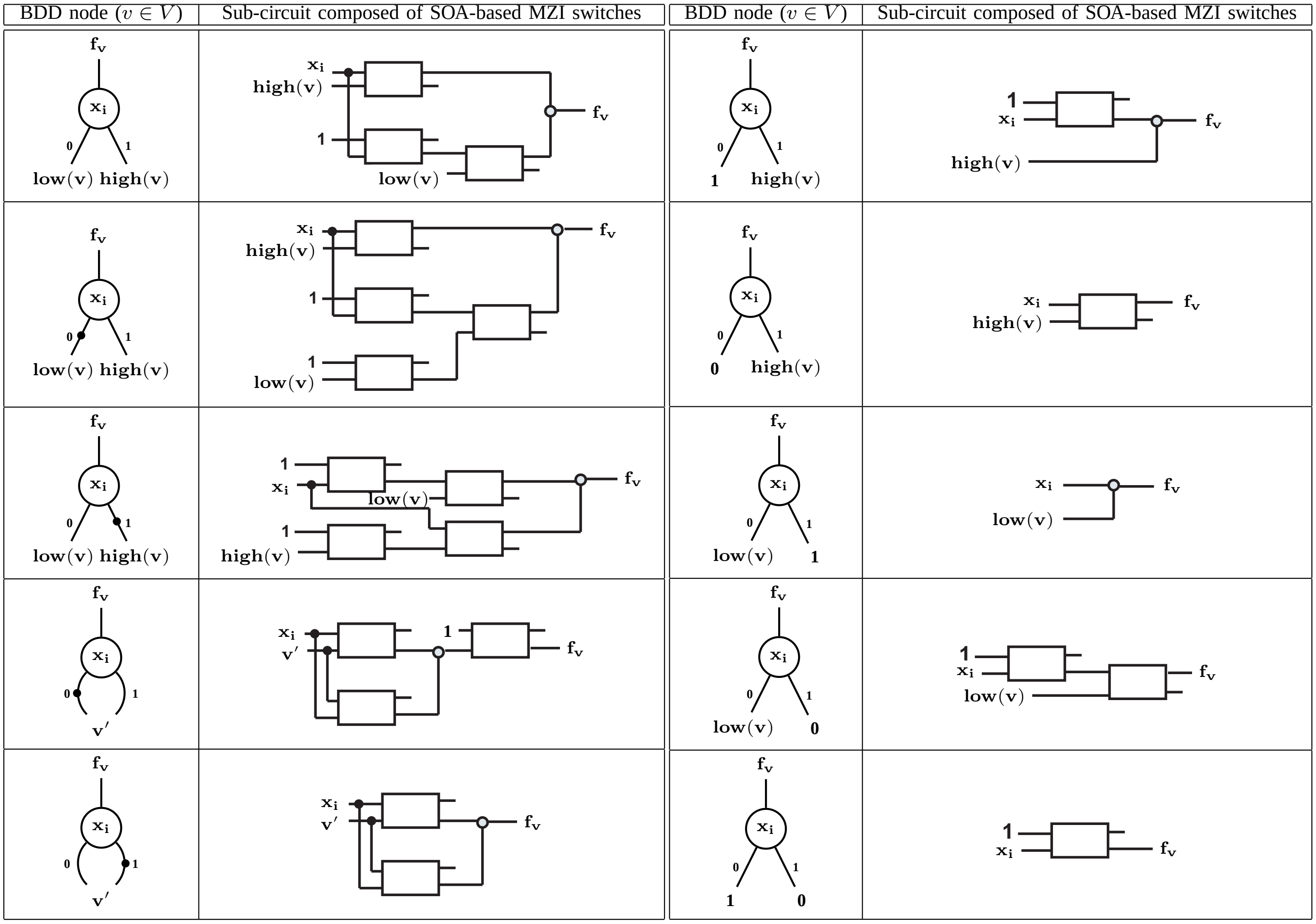
Using the technology library introduced above, a BDD $G = (V, E)$ representing the function f to be synthesized is mapped to a corresponding MZI circuit by performing the following steps:

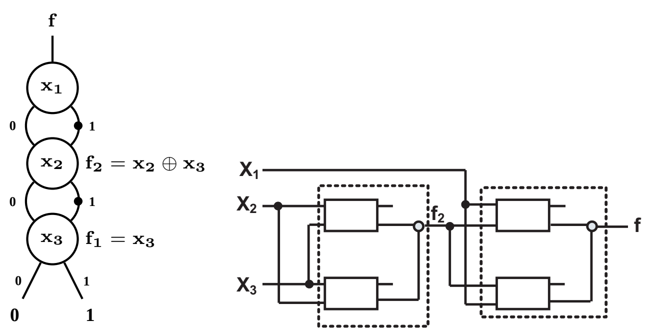
1) Traverse G in a depth-first manner.
2) For each node $v \in V$:
   a) Add the building block from the technology library realizing the respective configuration of v to the circuit to be synthesized.
   b) Connect the incoming signals of the building block accordingly to either the primary inputs of the overall circuit or to the corresponding output signals of the previously traversed nodes of G.
   c) If v is a root node of G, connect the respective outgoing signal of the building block to the corresponding primary output.

Connecting the respective incoming and outgoing signals may require the addition of further beam splitters. For example, each primary input x_i is used as an input in every building block mapped from a node with the select variable x_i. Hence, if there are multiple nodes with the select variable x_i, a beam splitter is connected to the primary input x_i, providing each building block mapped from these nodes with the input value of x_i. Similarly, if a node v is a shared node, the output signal of the building block mapped from v is needed as an input in multiple other building blocks. A primary input or an output signal from a building block may also be needed more than once as an input in the building block for a single other node (see Table I). Beam splitters are added accordingly in all these cases.

Example 2: Fig. 4a shows a BDD representing the function $f = x_1 \oplus x_2 \oplus x_3$ as well as the respective co-factors resulting from the application of the Shannon decomposition. The co-factor f_1 can easily be represented by the primary input x_3. Having the value of f_1 available, the co-factor f_2 can be realized by two MZI switches and a beam combiner as depicted in Fig. 4b. Since both inputs x_2 and x_3 (f_1) are needed twice in this building block, beam splitters are added to the respective primary inputs. The overall function f can then be realized by another two MZI switches and a beam combiner. Here, also the primary input x_1 as well as the co-factor f_2 are needed twice in the building block, i.e. beam splitters are added accordingly. The resulting circuit is shown in Fig. 4b.

TABLE I: The Applied Technology Library





(a) BDD (b) Resulting circuit

Fig. 4: Synthesizing $f = x_1 \oplus x_2 \oplus x_3$

IV. Discussion

In the literature, the costs of an optical circuit composed of SOA-based MZI switches are mainly measured in terms of the following metrics [14], [18], [21]:

- The number of MZI gates (often denoted as optical costs): This metric is simply motivated by the fact that each MZI gate needs to be physically realized.
- The number of beam splitters: Each time a beam is split, its signal strength is decreased considerably. Hence, keeping the splitting on signals as small as possible is an important objective.
- Delay: This metric denotes the longest path of a signal (with respect to the number of MZI gates passed) from an incoming port to an outgoing port. Keeping the delay as small as possible is also a crucial design objective.

The number of beam splitters indicates how often beams are split, but not how often a *single beam* is split. For the physical implementation, it will make a difference if e.g. ten beams are split just one time, or one beam is split ten times. Because of this, a further metric is considered in the following:

- The worst case fraction of a single beam: This metric approximates into how many beams a single beam is split in the worst case.

If we have a beam splitter with j outputs, each output is assumed to have a signal strength of $1/j$ of the incoming signal. Accordingly, for each beam splitter a signal passes, its fraction is multiplied by the corresponding number of outputs. A beam combiner will not change the fraction, as the worst case is assumed, i.e. all other inputs of the BC are assumed to hold the value 0. For an MZI switch, the fraction of both output signals is assumed to be the maximal fraction of both input signals. While this metric is just an approximation that may not correspond perfectly to the physical implementation, it gives a better idea of the resulting signal strength than just counting the beam splitters.

In the past, properties and characteristics of BDDs have intensely been investigated (see e.g. [27], [28]). Since the size of the circuits obtained by the synthesis scheme proposed above is bounded by the BDD size, all these results can also be applied to the synthesized optical circuits composed of SOA-based MZI switches. This allows for a deduction of theoretical properties for these optical circuits with respect to the metrics reviewed above. While an elaborated consideration of theoretical bounds is left for future work, some possible results are sketched in the following.

Number of MZI Gates: Using the synthesis approach proposed above, optical circuits realizing a function f are generated whose total number of MZI gates is bounded by the number of nodes in the BDD representing f. More precisely, the resulting circuit is composed of *at most* $(4 \cdot k) + 1$ MZI gates, since, for each node, a sub-circuit with at most 4 gates is added according to the technology library from Table I. Another single MZI gate might be necessary if the root node includes an incoming complemented edge. From this, the following conclusions can be drawn:

- A BDD representing a single-output function has $2^n - 1$ nodes in the worst case. Thus, each function can be realized as an optical circuit with at most $(4 \cdot 2^n) - 3$ MZI gates.
- A BDD representing a symmetric function has $\frac{n \cdot (n+1)}{2}$ nodes in the worst case. Thus, each symmetric function can be realized as an optical circuit with a quadratic number of gates.
- A BDD representing specific functions, like AND, OR, XOR, has a linear size. Thus, there exists an optical circuit realizing these functions in linear size as well.
- A BDD representing an n-bit adder has linear size. Thus, there exists an optical circuit realizing addition with a linear number of MZI gates as well.

For KFDDs, the same conclusions hold, but the worst case number of 4 gates in a sub-circuit is replaced by 5.

Worst Case Fraction of a Single Beam: Also the number of beam splits in the optical circuits generated with the proposed approach is bounded by the BDD. In the worst case, a select variable x_i is required as an input for an exponential number of nodes (the lowest level in a complete BDD has 2^{n-1} nodes). This leads to a beam with an exponential fraction. Such a beam might travel through n sub-circuits obtained from an entire BDD path, possibly leading to further splittings. Because of the BDD structure, the fraction will remain within $\mathcal{O}(2^n)$. This seems to promise rather poor results. However, as the experimental evaluation in Section V shows, the actual worst case fraction of a single beam is often considerably less than this theoretical value.

Delay: Finally, the delay of the optical circuits generated with the proposed approach is bounded by the depth of the BDD. More precisely, for each node, a sub-circuit with a delay of at most two MZI gates results (at most three gates for KFDD nodes). Let d be the depth of the considered BDD, then a signal never passes more than $2 \cdot d$ MZI gates from the primary incoming port to the primary outgoing port. In the worst case, $d = n$, i.e. the depth of the generated circuit is bounded by $2 \cdot n$. This depth might be increased by 1, if the root node includes an incoming complemented edge.

V. Experimental Evaluation

The synthesis approach presented in Section III has been implemented in C++ using the BDD-package CUDD [26]. In order to evaluate the approach, well known benchmark functions have been taken from libraries such as LGSynth and RevLib [29]. Since the size of the generated BDDs depends on the order of their input variables, a heuristic approach was applied to determine a good order efficiently. All experiments have been conducted on a 2.8 GHz Intel Core i7 processor with 7.8 GB of main memory.

Table II provides the obtained results. The name of each synthesized function is given in the first column (*Benchmark)*, followed by its number of primary inputs (*PI*) and primary outputs (*PO*). Column *k* denotes the number of nodes in the corresponding BDD. The following columns give the number of beam splitters (*BS*), MZI switches (*MZI*), and beam combiners (*BC*) in the obtained optical circuit. The worst case fraction of a single beam and the depth of the circuit are given in the Column *Fract.* and Column *d*, respectively. Finally, Column *Time* denotes the overall run-time of the synthesis process in CPU seconds.

The results confirm the applicability of the proposed approach: For the first time, optical circuits can automatically be generated for arbitrary Boolean functions. As can be seen, this process is done in negligible run-time for almost all considered benchmarks. Considering that, thus far, optical circuits composed of SOA-based MZI switches have mainly been derived manually, this is a significant achievement.

Besides that, the results also show that the obtained circuits are indeed bounded by the BDDs from which they have been derived (as discussed in Section IV). For example, the *urf*-functions are known to have no regular structure in their specification, which usually results in large BDDs. In contrast, functions like e.g. *ham15_30* can usually be represented very efficiently as BDDs. This is also reflected in the circuit sizes: Although *ham15_30* has more primary inputs, the obtained circuit is factors (for some metrics even magnitudes) smaller than the circuit e.g. obtained for *urf1_72*.

However, the need for splitters is a drawback of the proposed approach. But also here, the precise effect strongly depends on the considered function. Sometimes no or only very few splitters are required; sometimes a significant amount. Also the actual effect of the splitters is different (something which has not been considered at all in logic design of all-optical circuits yet): Sometimes a large number of beam splitters exist which, however, are distributed uniformly through all signals, i.e. the worst case fraction of a single beam remains moderate. In other cases, the fraction is very large. Optimizing this is an important issue for future work.

VI. Conclusion

In this work, we proposed an approach for the synthesis of all-optical circuits composed of SOA-based MZI switches. For this purpose, BDDs have been exploited. This allowed, for the first time, an automatic and scalable logic synthesis of arbitrary Boolean functions as optical circuits. At the same time, the approach allows to transfer theoretical results known from BDDs such as upper bounds into this domain. Experimental evaluations demonstrated the applicability of the synthesis scheme.

TABLE II: Experimental Evaluation

Benchmark	PI	PO	k	BS	MZI	BC	Fract.	d	Time
0410184_85	14	14	65	63	184	57	46080	16	0.10
4gt11_23	4	1	2	0	1	0	1	1	0.00
4gt12_24	4	1	4	0	2	1	1	2	0.00
4gt13_25	4	1	3	0	2	0	1	2	0.00
4mod5_8	4	1	4	5	8	2	8	4	0.01
4mod7_26	4	3	11	17	29	8	96	6	0.00
5xp1_90	7	10	29	35	68	22	38880	14	0.01
9symml_91	9	1	24	25	62	21	128	11	0.00
add6_92	12	7	52	52	140	47	55296	16	0.02
adr4_93	8	5	15	27	38	13	17496	13	0.00
alu_9	5	1	6	9	12	4	4	4	0.00
alu1_94	12	8	20	12	34	12	4	4	0.00
alu2_96	10	6	149	194	451	131	207360	19	0.13
alu3_97	10	8	75	68	197	66	100800	15	0.03
alu4_98	14	8	675	251	1823	639	2433024	18	1.91
apex2_101	39	3	496	260	1108	493	1583688704	34	1.21
bw_116	5	28	92	111	353	90	18144	12	0.06
cordic	23	2	35	43	103	28	150994944	32	0.02
decod24-enable_32	3	4	7	5	8	4	4	3	0.00
e64	65	65	198	118	128	196	1610612736	3	0.11
ex5p_154	8	63	235	80	430	214	168480	12	0.20
ham15_30	15	15	62	100	147	57	4096	11	0.31
ham3_28	3	3	5	4	15	4	15	4	0.00
hwb4_12	4	4	14	22	42	11	144	7	0.00
hwb5_13	5	5	37	14	99	31	96	7	0.01
hwb6_14	6	6	63	32	181	58	336	9	0.02
hwb8_64	8	8	162	79	512	156	2744	12	0.14
hwb9_65	9	9	245	108	791	236	8232	15	0.31
mod5d2_17	5	5	9	13	24	6	48	6	0.00
one-two-three_27	3	3	7	9	16	4	24	5	0.00
pdc	16	40	577	245	1099	575	2995200	18	1.19
plus127mod8192_78	13	13	24	36	51	19	590490	7	0.05
plus63mod4096_79	12	12	22	33	48	17	196830	7	0.03
plus63mod8192_80	13	13	24	36	53	18	590490	8	0.05
rd53_68	5	3	13	24	46	12	800	10	0.01
rd73_69	7	3	30	29	80	26	768	11	0.00
rd84_70	8	4	41	37	107	37	1536	12	0.01
sf_232	4	1	5	6	12	3	6	4	0.00
spla_202	16	46	570	198	1050	568	14288400	17	1.19
sym6_63	6	1	12	12	25	9	16	6	0.00
urf1_72	9	9	647	441	2171	621	515424	16	2.21
urf2_73	8	8	336	323	1154	325	231840	15	0.64
urf5_76	9	9	307	365	949	281	575168	19	0.52
xor5	5	1	5	8	12	4	16	5	0.00

Benchmark: Name of funct. PI/PO: Number of primary inp./outp.
k: Nodes in the BDD BS: Number of beam splitters
MZI: Number of MZI switches BC: Number of beam combiners
Fract.: Worst case fraction of a single beam
d: Depth of the circuit Time: Run-time in CPU seconds

The proposed approach provides the basis for further work in automatic synthesis of all-optical circuits. In particular, reducing the maximal number of times a beam is split remains an open issue. Modifying the costs function of the BDD package or applying post-synthesis optimization schemes are possible directions to address this problem. Also the approach of traversing the BDD in a reverse fashion as concurrently developed in [30] for crossbar gate-based circuits is a promising direction in this regard. Besides that, a detailed analysis of the theoretical results that can be obtained by the proposed approach is left for future work.

REFERENCES

[1] R. E. Bryant. Graph-based algorithms for Boolean function manipulation. *IEEE Trans. on Comp.*, 35(8):677–691, 1986.

[2] G.E. Moore. Cramming more components onto integrated circuits. *Journal of Electronics*, 38(8):183–191, 1965.

[3] A. Shacham, K. Bergman, and L. P. Carloni. Photonic network-on-chip for future generations of chip multi-processors. *IEEE Transactions on Computers*, 57(9):1246–1260, 2008.

[4] P.K. Kaliraj, P. Sieber, A. Ganguly, I. Datta, and D. Datta. Performance evaluation of reliability aware photonic network-on-chip architectures. In *Int'l Green Computing Conf.*, pages 1–6, 2012.

[5] R. Clavero, F. Ramos, J. M. Martinez, and J. Marti. All-optical flip-flop based on a single SOA-MZI. *IEEE Photonics Technology Letters*, 17(4):843–845, 2005.

[6] Q. Wang, G. Zhu, H. Chen, J. Jaques, J. Leuthold, A. B. Piccirilli, and N. K. Dutta. Study of all-optical XOR using Mach-Zehnder interferometer and differential scheme. *IEEE Journal of Selected Topics in Quantum Electronics*, 40(6):703–710, 2004.

[7] M. Zhang, Y. Zhao, L. Wang, J. Wang, and P. Ye. Design and analysis of all-optical XOR gate using SOA-based Mach-Zehnder interferometer. *Optical Communications*, 223:301–308, 2003.

[8] T. Chattopadhyay. All-optical symmetric ternary logic gate. *Optics and Laser Technology*, 42(6):1014–1021, 2010.

[9] A. Al-Zayed and A. Cherri. Improved all-optical modified signed-digit adders using semiconductor optical amplifier and Mach-Zehnder interferometer. *Optics and Laser Technology*, 42(5):810–818, 2010.

[10] D. K. Gayen and T. Chattopadhyay. Designing of optimized all-optical half adder circuit using single quantum-dot semiconductor optical amplifier assisted Mach-Zehnder interferometer. *Journal of Lightwave Technology*, 31(12):2029–2035, 2013.

[11] M. A. Ghasemi, R. Khodadadi, and H. A. Banaei. Design and simulation of all optical multiplexer based on one-dimensional photonic crystal for optical communications systems. *Int'l Journal of Engineering Research and Applications*, 2(6):960–968, 2012.

[12] J. W. M. Menezes, W. B. Fraga, A. C. Ferreira, G. F. Guimares, A. F. G. F. Filho, C. S. Sobrinho, and A. S. B. Sombra. All-optical half adder using all-optical XOR and AND gates for optical generation of sum and carry. *Fiber and Integrated Optics*, 29(4):254–271, 2010.

[13] M. Nady, K. F. A. Hussein, and A. E. A. Ammar. Ultrafast all-optical full adder using quantum-dot semiconductor optical amplifier-based Mach-Zehnder interferometer. *Progress In Electromagnetics Research B*, 54:69–88, 2013.

[14] S. Kotiyal, H. Thapliyal, and N. Ranganathan. Mach-Zehnder interferometer based all optical reversible NOR gate. In *IEEE Computer Society Annual Symposium on VLSI*, pages 207–212, 2012.

[15] T. Chattopadhyay. All-optical modified Fredkin gate. *IEEE Journal of Selected Topics in Quantum Electronics*, 18(2):585–592, 2012.

[16] C. Taraphdar, T. Chattopadhyay, and J.N. Roy. Mach-Zehnder interferometer based all-optical reversible logic gate. *Optics and Laser Technology*, 42(2):249–259, 2010.

[17] A.J. Poustie and K.J. Blow. Demonstration of an all-optical Fredkin gate. *Optics Communications*, 174:317–320, 2000.

[18] S. Kotiyal, H. Thapliyal, and N. Ranganathan. Mach-Zehnder interferometer based design of all optical reversible binary adder. In *Design, Automation and Test in Europe*, pages 721–726, 2012.

[19] Z. Chen. Simple novel all-optical half adder. *Optical Engineering*, 49(4):043201–1–043201–6, 2010.

[20] M. P. Scaffardi, P. Ghelfi, E. Lazzeri, L. Poti, and A. Bogoni. Photonic processing for digital comparison and full addition based on semiconductor optical amplifiers. *IEEE Journal of Selected Topics in Quantum Electronics*, 14(3):826–833, 2008.

[21] K. Datta and I. Sengupta. All optical reversible multiplexer design using Mach-Zehnder interferometer. In *Int'l Conf. on VLSI Design*, pages 539–544, January 2014.

[22] R. Drechsler, A. Sarabi, M. Theobald, B. Becker, and M. A. Perkowski. Efficient representation and manipulation of switching functions based on ordered Kronecker functional decision diagrams. In *Design Automation Conf.*, pages 415–419, 1994.

[23] K. S. Brace, R. L. Rudell, and R. E. Bryant. Efficient implementation of a BDD package. In *Design Automation Conf.*, pages 40–45, 1990.

[24] R. Drechsler, J. Shi, and G. Fey. MuTaTe: an efficient design for testability technique for multiplexor based circuits. In *Great Lakes Symposium on VLSI*, pages 80–83, 2003.

[25] R. Wille and R. Drechsler. BDD-based synthesis of reversible logic for large functions. In *Design Automation Conference*, pages 270–275, 2009.

[26] F. Somenzi. *CUDD: CU Decision Diagram package: release 2.5.0*. University of Colorado at Boulder, http://vlsi.colorado.edu/ fabio/CUDD/, 2012.

[27] I. Wegener. *Branching programs and binary decision diagrams: theory and applications*. Society for Industrial and Applied Mathematics, 2000.

[28] R. Drechsler and D. Sieling. Binary decision diagrams in theory and practice. *Software Tools for Technology Transfer*, 3:112–136, 2001.

[29] R. Wille, D. Große, L. Teuber, G. W. Dueck, and R. Drechsler. RevLib: An online resource for reversible functions and reversible circuits. In *Intl. Symposium on Multi-valued Logic (ISMVL)*, pages 220–225, 2008. RevLib is available at http://www.revlib.org.

[30] R. Wille, O. Keszocze, C. Hopfmuller, and R. Drechsler. Reverse BDD-based synthesis for splitter-free optical circuits. In *ASP Design Automation Conf.*, 2015.