INTEGRATION the VLSI journal 59 (2017) 42–51

Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

Synthesis of optical circuits using binary decision diagrams

Arighna Deb[a,b,*], Robert Wille[c,d], Oliver Keszöcze[a,d], Saeideh Shirinzadeh[a], Rolf Drechsler[a,d]

[a] Institute of Computer Science, University of Bremen, Bremen, Germany
[b] Computer Science and Engineering, Jadavpur University, Kolkata, India
[c] Institute for Integrated Circuits, Johannes Kepler University, Linz, Austria
[d] Cyber-Physical Systems, DFKI GmbH, Bremen, Germany

ARTICLE INFO

Keywords:
Optical circuits
Synthesis
Optimization
Binary decision diagrams

ABSTRACT

The advances in silicon photonics motivated the consideration of optical circuits as a new and emerging circuit technology. In particular for ultra-fast interconnects, optical circuits may provide a suitable alternative since it avoids the conversion of signals from the optical to the electrical domain. Accordingly, design automation of this kind of circuits received significant attention. In this work, we consider synthesis of optical circuits based on *Binary Decision Diagrams* (BDDs). Although BDDs allow for a direct mapping of the function representation to an optical circuit (and, hence, a scalable synthesis), they have their shortcomings with respect to dedicated cost metrics. In this work, we investigate this issue and provide an overview of the BDD-based synthesis schemes which are available thus far. Afterwards, we propose new solutions based on a dedicated BDD optimization which aim for addressing the known shortcomings. Experimental results confirm the benefits of the proposed approach.

1. Introduction

The scaling of integrated circuits driven by Moore's law [1] has dominated the semiconductor industry in last decades. But conventional technologies will reach to their physical limits in the near future. Besides that, the increasing emphasis on high speed and low power designs changes the need and requirements of modern circuits and systems. As a consequence, alternatives to conventional computing systems are currently being investigated.

Due to advances in *silicon photonics* [2], optical computing can be considered as such an alternative. So far, optical technology is used as ultra-fast interconnects [3,4] in electronic digital systems. But here, a conversion between the electrical and optical domain needs to be carried out at every interconnect interface. This requires corresponding hardware and introduces a significant overhead. The conversion, however, can easily be avoided if the underlying systems are realized by optical technologies. This motivates research in the area of designing optical circuits.

Optical technology has already found wide applications in communication networks [5] and signal processing systems [6]. The large sized optical circuits and systems used in communication networks are not suitable for chip based miniaturized computing devices. Consequently, researchers are working towards developing new miniaturized all-optical devices that can be used to design full scale all-optical computing systems. Recently, all-optical transistors [7,8] and an all-optical memory unit [9] have been demonstrated. While the suitable hardware aspects for all-optical circuits are being investigated, the software challenges i.e. the systematic design of all-optical circuits constitutes an important challenge in this field.

In this regard, design automation is one of the key driving forces behind the reduction of the circuit design time and the optimization of the circuit quality. Thus far, *Electronic Design Automation* (EDA) has played a crucial role in the development of complex conventional circuits [10]. The major design step usually starts with a logic level abstraction and, afterwards, moves down to the physical realization, where the desired circuit is refined with respect to the respective technological constraints. Although the physical constraints (such as size, cascadibility, power loss, fan-out capability or gain, etc.) in optical technology are not entirely solved yet, initial models and corresponding gate libraries already exist for the purpose of logic synthesis and optimization. At this stage, considering logic synthesis and optimization is motivated by the fact that EDA for conventional systems started with logic design automation before physical design automation was actually developed [11].

The initial work on logic design of optical circuits was primarily involved in realizing basic Boolean operations [12–14]. Later, some

[*] Corresponding author at: Computer Science & Engineering, Jadavpur University, Kolkata, India.
E-mail addresses: arighna.deb@gmail.com (A. Deb), robert.wille@jku.at (R. Wille), keszocze@informatik.uni-bremen.de (O. Keszöcze), saeideh@informatik.uni-bremen.de (S. Shirinzadeh), drechsler@uni-bremen.de (R. Drechsler).

http://dx.doi.org/10.1016/j.vlsi.2017.05.001
Received 30 May 2016; Received in revised form 14 April 2017; Accepted 5 May 2017
Available online 11 May 2017

important arithmetic units such as adder, comparator, encoder, divider, etc. have been demonstrated in optical domain (for e.g., see [15–17]). Although, this shows a progress in the development of optical logic circuits, the designs are limited to single applications only. Therefore, synthesis approaches are required to efficiently generate any Boolean function – an initial step towards design automation.

A particular consideration has been put on synthesis and optimization using *Binary Decision Diagrams* (BDDs) [18,19]. BDDs are a data-structure for the efficient representation of large Boolean functions [20] and allow for scalable circuit synthesis. Consequently, two approaches for the synthesis of optical circuits based on BDDs have been proposed: one employs a direct mapping scheme [18], where each node of the BDD is substituted by an optical gate. Furthermore, the sharing of a node is realized by a splitter which divides the optical signal into many. As this causes huge fractions of signal strength, an alternative has been proposed in [19] employing a reverse scheme avoiding splitting of signals and, hence, signal fractions at all (both approaches are later reviewed in more detail in Section 3).

In this work, we provide an overview of previously proposed BDD-based synthesis approaches. Afterwards, we discuss their efficiency and, motivated by that, consider how BDD optimization may be put to an advantage for the synthesis of the corresponding circuits. We particularly observe how the number of shared nodes and the number of one-paths affect the results depending on the applied synthesis scheme. Based on this, we propose improvements which significantly address the drawbacks of the previously proposed solutions.

This way, this manuscript provides a comprehensive consideration of BDD-based synthesis of optical circuits covering previous work as well as new optimizations. Experimental evaluation shows the effect of the newly proposed optimization. Depending on the design objective either improvements in the number of gates of up to 33% on average or improvements of several orders of magnitudes in the worst case fraction can be reported. This eventually yields logic designs which fit much better to currently known and possible future technological constraints.

The remainder of the paper is structured as follows: Section 2 provides an overview on the optical logic gates as well as the basics of BDDs. In Section 3, both BDD-based synthesis approaches are briefly reviewed and a general idea of exploiting BDD optimization is discussed. Afterwards, in Section 4, a detailed discussion of new BDD optimization technique is provided along with the brief descriptions of existing BDD optimization techniques. Finally, the effect of these optimization techniques on the resulting circuits is experimentally evaluated in Section 5, while the paper is concluded in Section 6.

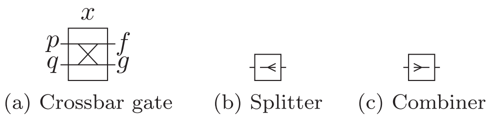
2. Preliminaries

To keep the paper self-contained, this section briefly reviews the logic models and cost metrics used in the domain of optical logic synthesis. We also describe the basics of BDDs which are used as the underlying data structure by the considered synthesis approaches.

2.1. Optical circuits

As discussed in the previous section, several technological realizations of optical circuits are currently considered. However, in order to ease the development of corresponding methods for the design automation of optical circuits, logic models, gate libraries, and cost metrics have been proposed which abstract from detailed physical and optical issues but discretely reflect the major constraints.

In this abstraction, Boolean functions can be realized by optical switching elements, namely *crossbar gates*, which route the optical signals between two parallel paths. The inputs of both paths can either be sourced by light (representing the logical value “1”) or darkness (representing the logical value “0”). Furthermore, the routing of both paths may be switched depending on the value of an electrical signal.



(a) Crossbar gate (b) Splitter (c) Combiner

Fig. 1. Optical gates.

The output of each optical signal can be read using optical receivers. Logically, this leads to the following definition of a crossbar gate.

Definition 1. A crossbar gate realizes a Boolean function $\mathbb{B}^3 \to \mathbb{B}^2$ composed of two optical inputs p and q, one electrical input x, and two optical outputs f and g. The signals p and q as well as f and g are connected by waveguides which, depending on the value of x, realize either the identity or a switch of the input values, i.e.

$$\bar{x} \Rightarrow (p \equiv f) \wedge (q \equiv g) \quad \textit{and} \quad x \Rightarrow (p \equiv g) \wedge (q \equiv f)$$

is realized. Fig. 1(a) provides the graphical representation of a crossbar gate.

Note that, except for the crossbar gate as defined above, an optical signal cannot directly modify the value of an electrical signal and vice-versa. For such purposes, an opto-electrical interface would be required which, however, is considered as expensive as well as slow. In this regard, we follow the established model from [18] which prohibits any interaction between electrical and optical signals except for crossbar gates.

In addition to crossbar gates, splitters and combiners are also utilized as optical logic elements in order to realize logic functions.

Definition 2. A splitter divides an optical signal into two signals – each with only half of the incoming signal power. In contrast, a *combiner* merges two optical signals into a single one and, by this, inherently realizes the OR-function. A splitter (combiner) may have more than two outputs (inputs). Then, in case of a splitter, the strength of the signal is divided by the number of outputs. Fig. 1(b) and (c) provide the graphical representation of both elements.

Using these logic elements, a technology library is formed that allows to realize all basic Boolean operations as well as arbitrary Boolean functions. As an example, Fig. 2 shows realizations for NOT, XOR, AND, and OR (as well as their inversions) – clearly representing a universal set of operations out of which any other function can be composed.

To measure the complexity of an optical circuit, several metrics got established in the literature [18,19]. More precisely:

- The number of crossbar gates (often also denoted as gate costs): This metric is motivated by the fact that each crossbar gate needs to be physically realized. Sometimes, also the number of splitter and the number of combiners are considered. However, as they are significantly easier to realize than the crossbar gates, they are often considered negligible.
- The worst case fraction: This metric is motivated by the fact that splitters in the circuit degrade the strength of the involved signal. Since simply counting the number of splitters provides an imprecise measure of the resulting effects (the impact of splitters will significantly differ if, e.g., ten splitters separately split ten signals or one signal ten times in a cascade), a more elaborate metric is considered. First, it is assumed that each splitter with n outputs produces an output signal of strength $^1/_n$. Then, the circuit's worst case fraction is determined by following all paths from the primary inputs to the primary outputs and multiplying all signal strengths produced by splitters visited on the current path. Afterwards, the maximum value of all paths is considered the worst case fraction.[1]

[1] To keep the model simple, combiners are assumed not to change the signal strength. This makes the worst case fraction a pessimistic approximation i.e. the actual signal strength may be better than the computed one.