

Logic Synthesis for Energy-Efficient Photonic Integrated Circuits

Zheng Zhao¹, Zheng Wang², Zhoufeng Ying¹, Shounak Dhar¹, Ray T. Chen^{1,2}, and David Z. Pan¹

¹Department of Electrical and Computer Engineering, University of Texas at Austin

²Department of Materials Science and Engineering, University of Texas at Austin

zhengzhao@utexas.edu, wangzheng@utexas.edu, zfyang@utexas.edu, shounak.dhar@utmail.utexas.edu, chenrt@austin.utexas.edu dpan@ece.utexas.edu

ABSTRACT

The development of photonic integrated circuits (PICs) has made it possible to accomplish on-chip optical interconnects and computations. As a promising alternative to traditional CMOS circuits, optics has demonstrated the ability to realize ultra-high speed and low-power information processing and communications. In this work, we propose a logic synthesis methodology for PICs. For the first time, practical issues including the insertion losses from optical combiners and switches are considered. Two optimization techniques based on binary decision diagram, combiner elimination and coupler assignment, are proposed to improve the power efficiency for PICs. Experimental results of MCNC and IWLS combinational benchmarks showed our method could efficiently generate quality PICs with a 27.02X better optical power efficiency on average, and greatly reduce the optical power depletion and facilitate large-scale on-chip optical computation.

1. INTRODUCTION

As Moore's law is approaching the limits, photonic integrated circuits (PICs) has received increasing attention of high-speed and energy-efficient on-chip optical interconnects and computing. Among all platforms of building PICs, silicon-on-insulator (SOI) is the most promising due to the CMOS-compatible process enabled low-cost and large-volume manufacturing and the ability of monolithic integration of electronics and photonics. System advantages in the photonic schemes over the conventional electronic schemes include but not limited to: (1) significant reduction of gate latency, (2) ultra-low energy consumption per bit, and (3) simplified layout architecture for many complex computation structures [1, 2]. Compared with optical computing, optical interconnects have been more intensively investigated approving the advantage over metal interconnects especially in intra- and inter-chip communications [3–5]. To catch up with the advancement in optical interconnects, previous works on optical computing have demonstrated basic Boolean operations such as (N)AND, (N)OR and X(N)OR gates [6, 7] and more complex functionalities such as 1-bit half and full adders [2, 8]. Most recently, an ultra-fast low-power deep learning network has been implemented by optical Mach-Zehnder interferometers (MZIs), and has reignited the interest of computing with optics [9].

In order to implement general and large-scale PICs and pave the way for design-space exploration, a synthesis scheme based on virtual gates (VGs) and optical splitters are proposed in [10]. A virtual gate is a 2×2 crossbar that could be switched by a function, not necessarily a primary input. In a VG-based architecture, each literal is implemented by a VG. While the concept of such virtual gates is worthwhile, the proposed method usually results in a large number of op-

tical components (VGs and splitters). More importantly, as is well noted by the authors, VG-based architecture contains a great number of optical splitters, each resulting in a -3dB loss, and the loss cascaded inevitably leads to an extremely weak output signal indistinguishable from noises. In the same paper, the idea of using binary decision diagrams (BDD) is also briefly mentioned. The idea is simply to replace each BDD node by an optical crossbar which is solely controlled by the primary inputs. Splitters and waveguides are used to connect between crossbars. However, it has been discarded as an option due to the garbage outputs related to the architecture, which the authors claim complicates routing. In more recent technology, this problem can be easily resolved by adding an optical terminator to each garbage output. The overhead in terms of loss, area, stability, and manufacturability of the terminator is very negligible.

A more recent work [11] introduced a reverse-BDD scheme. Different from the idea mentioned in [10], the light streams from the top of the BDD and the photodetector receives the light at the bottom. Each splitter in the previous architecture is thus replaced by an optical combiner, a structural complement of the splitter. The reverse-BDD mainly targets at minimizing the splitter loss. However, the authors fail to consider the combiner loss. As shown in the power distribution based on the simulation and experimental results in Figure 1, the Y-branch takes light from the two ports on the right side and pass light to the left. In the first figure, if two input ports have light, the output power doubles each of the input power and there is almost zero loss. In the second and third figure, if only one input has light, due to the mode mismatch, half of the light escape from the waveguide to the free space. Therefore, there will be a -3dB (50%) power loss at the output.

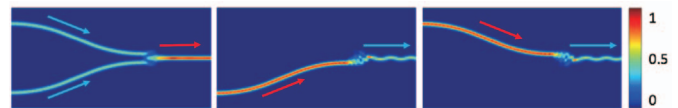


Figure 1: Power distribution of a typical optical combiner.

In this work, we resolve the aforementioned problems by a new set of optical synthesis techniques. We introduce optical switching components such as microresonator-based optical switches and directional couplers. Compared with MZIs used in the previous work, microresonators such as microrings and microdisks, have much a smaller footprint ($< 100 \text{ } \mu\text{m}^2$). The microresonators are used in microresonator-based optical switches that could be driven by a CMOS-compatible voltage ($< 1 \text{ Vpp}$) with much lower energy consumption ($< 50 \text{ fJ/bit}$) [12, 13]. Our experiments are based on up-to-dated experimental data from microdisk-based optical switches, which realizes the lowest real estate for conventional guided wave devices and therefore maximize the packaging density for PICs.

As will be shown, most of the proposed synthesis techniques

are also applicable to optical switches built with other structures, such as MZIs and Multimode Interferences (MMIs). The main contributions of this paper are as follows.

- We pinpoint and study the critical problem of optical power depletion, which has long been neglected in previous PIC design.
- We develop a synthesis method for optical circuits. Realistic issues of power loss are modeled and considered for the first time in the synthesis of PICs.
- We propose two optimization techniques that significantly reduce the optical power loss which assures a better signal-to-noise ratio (SNR) for practical PIC designs.

Note that the proposed techniques can be applicable to PICs built with *any switching components*, such as MZIs, microresonators, electro-absorption modulator, etc., and it is also not restricted to the choice of platforms.

The remainder of this paper is organized as follows. Section 2 introduces the background of optical computing devices used in this work, followed by the data structure for optical logic synthesis. Section 3 presents the synthesis techniques. Section 4 reports the experimental results, followed by the conclusion in Section 5.

2. BACKGROUND AND DEVICE MODELS

In this section, we introduce the basic optical components and their working principles. Then we briefly discuss the primary data structure for optical synthesis.

2.1 Optical Building Blocks

2.1.1 Microresonator-based Optical Switches

One of the fundamental optical computational units is optical switches. In this work, we focus on microresonator based switches for the advantages mentioned in Section 1. Note that the high-level switch model is also general enough to represent MZI-based switches. Figure 2a shows a schematic of a typical 1×2 microresonator-based optical switch using microrings, which has one light input and two outputs: the through output and drop output. A continuous wave (CW) light is fed into the switch from the input. Part of the light will be coupled into the microresonator, and then coupled back to the two bus waveguides with certain phase shifts. The phase shifts are highly related to the wavelength of the CW light, which result in a wavelength selective behavior shown in the black curves. One could introduce the refractive index perturbation by applying an electrical signal to shift the resonance peaks to switch the light, which is illustrated with the red and blue curves in Figure 2c.

By using both the through and drop ports of the switch, one can build 1×2 and 1×1 optical switches controlled by electrical signals. Figure 2b are the high-level notation for 1×2 and 1×1 optical switches, respectively. The light passes or terminates depending on the controlling electrical signal and the configuration. Following the convention of electronics, we define that if the controlling variable of the switch is 1, the light is passed to the 1 output port, otherwise to the 0 output. The 1×1 optical switch could be realized by adding an optical terminator at one of the ports¹. Generally, there will be some loss caused by switching (switching loss), especially when light comes from the drop port. In the spectrum, we can see the drop port has a power loss of 20% (i.e., the efficiency factor $\gamma_{drop} = 0.8$) for a non-ideal microresonator.

¹Strictly speaking, the 1×1 component can not be called a switch, but as it is very similar to the 1×2 switch, we will follow the same term.

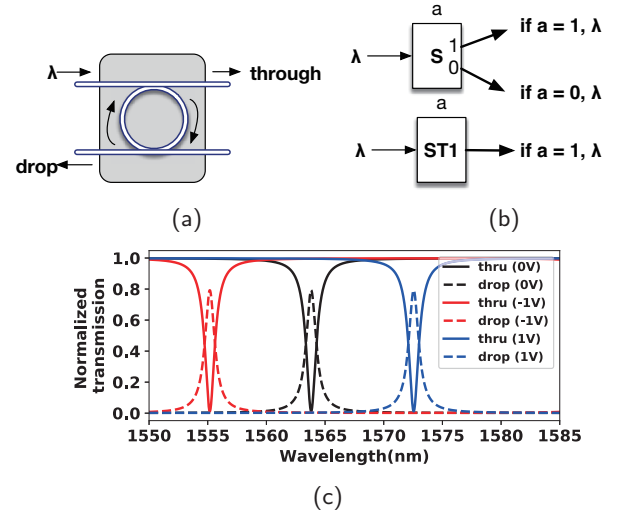


Figure 2: (a) schematic diagram of a microresonator (b) 1×2 and 1×1 switch notations (c) optical transmission spectra measured at through and drop ports under various bias.

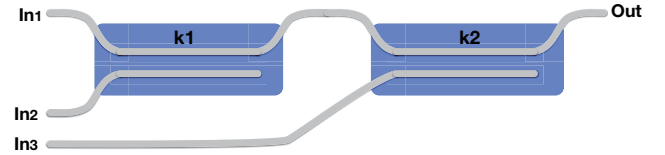


Figure 3: 3×1 directional coupler.

2.1.2 Y-branch Combiner and Directional Coupler

A typical 2×1 combiner can be implemented by a Y-branch silicon waveguide with a minimal footprint among all possible approaches. In Figure 1, we can see when there is only one light input (λ), we have the worst-case loss and the power efficiency factor is 0.5. An $N \times 1$ combiner can be implemented by connecting an array of 2×1 combiners. The efficiency factor for each input port to the output is the product of the efficiency factor of the 2×1 combiners on the path from this port to the output. The efficiency factor is dependent on the exact configuration of the 2×1 combiner array. In the following, we will assume that the efficiency factor is $1/N$ as in the mathematical expectation.

A more sophisticated guided wave component for merging is the directional optical coupler. The coupling efficiency is adjustable in the range of 0 to 100% by changing the coupling length of the directional coupler. For a 2×1 coupler with a coupling constant k , the output power $P_{out} = k \cdot P_{in1}$, if the light comes from the first input port $in1$, and $P_{out} = (1 - k) \cdot P_{in2}$, if the light is from the second input port $in2$. We will assume at most one of input ports has light. As will be shown, this assumption is valid due to a special property of BDD. Similarly, an $N \times 1$ coupler can be obtained by cascading $(N - 1)$ 2×1 couplers, and achieves an arbitrary coupling efficiency to each input. Figure 3 shows an example of 3×1 coupler. Suppose the efficiency factors we want to achieve are γ_1 , γ_2 and γ_3 for the three inputs to the output, respectively. The coupling constant k_1 , k_2 and k_3 can be figured out by the following relationship

$$\gamma_1 = k_1 k_2, \gamma_2 = (1 - k_1) k_2, \gamma_3 = 1 - k_2 \quad (1)$$

From the last two equations, we have

$$k_2 = 1 - \gamma_3, k_1 = (1 - \gamma_2 - \gamma_3) / (1 - \gamma_3)$$

The rule of energy conservation demands that $\gamma_1 + \gamma_2 + \gamma_3 = 1$.

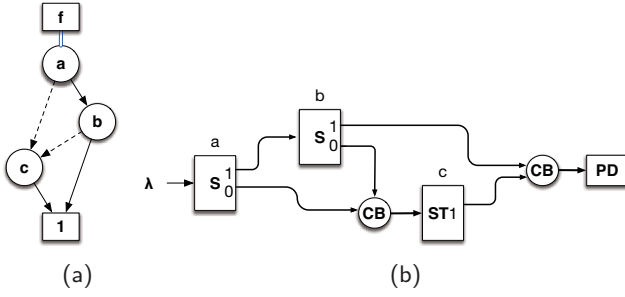


Figure 4: (a) 1-terminal BDD and (b) synthesis method in [11].

So $k_1 = \gamma_1/(1 - \gamma_3)$. Now we look at the first equation in Eq. 1. The product $k_1 k_2$ correctly satisfies the relationship. The general case can be proved by induction on the number of inputs and is spared due to the page limits.

The size of a typical coupler is almost two times the size of a typical micro-resonator-based switch (e.g., mirroring resonator-based switch). The trade-off of the flexibility of reassigning the efficiency factor and the induced area overhead is tackled in our synthesis flow.

2.2 Data Structures for Optical Synthesis

The primary data structure for optical logic synthesis is reduced ordered binary decision diagram (ROBDD) [14]. A BDD is a directed acyclic graph that can represent a Boolean function. There are two types of nodes in a BDD, terminal nodes and decision nodes. A terminal node has a value 1 or 0, representing the functional output evaluation. As will be explained, in optical synthesis, we will only keep the 1-terminal, as shown in Figure 4a. A decision node is functionally a 1×2 crossbar switch, which is controlled by a decision variable. The outputs of a decision node are called its children, the *low child* and the *high child*. The high (low) child are connected to the parent node through a *high edge* (low edge), marked by a solid (dashed) line.

Given a fixed variable order, a BDD can be reduced to become an ROBDD, where there is no redundant nodes that implement the same functionality. ROBDD is a more compact representation. In the following discussion, we will use BDD and ROBDD interchangeably.

BDD has an important *single-path property* which says

Claim 1. *For any input assignment, there is at most one path from any decision node to the 1-terminal.*

To prove it, suppose there are two paths from a decision node to the 1-terminal, which correspond to the same input assignment. As there are two paths, there must exist a decision node where the two paths diverge, one from its high edge and the other from its low edge. The two edges correspond to different input assignment, which contradicts the assumption.

The synthesis method proposed in [11] is demonstrated in Figure 4b. The light (λ) from a laser source (or from the output of the previous optical network) is streamed from the BDD top node to the 1-terminal, where a photodetector (PD) (or optical amplifier to the next computation stage) is located. The synthesis replaces each BDD node by an optical 1×2 crossbar, each controlled by a primary input. Waveguides and combiners are used to connect the crossbars. When there are multiple inputs to a crossbar, optical combiners (CB) discussed in Section 2.1.2 are used to merge the inputs.

A critical observation here is that the combiners always result in the worst-case loss; to achieve better-than-worst-case power efficiency, there has to be multiple lights streaming to the combiner at the same time, but this cannot happen due to the single-path property. From this configuration, we can

see the output of the optical network is a logical 1, if the PD can detect light at the 1-terminal, otherwise it is a logical 0.

3. SYNTHESIS ALGORITHMS

In this section, we will discuss the proposed synthesis techniques in detail. We start from an arbitrary logic function and build the initial BDD. As discussed in the previous section, the initial BDD can be trivially mapped to an optical network of optical switches and combiners. However, this direct implementation suffers from cascading optical power loss as the single input caused intrinsic combiner loss and is thus prohibited from building larger-scale functionalities.

The path efficiency is the product of all the components efficiency on the path. It is not difficult to imagine that the optical signal degrades very fast from the input to the output. We attempt to solve this problem by two techniques: combiner elimination and coupler assignment. The two techniques have their own strengths and limitations and will be exploited together in the synthesis flow.

The efficiency factor γ is an important concept in this work. For an optical network, simple or complex, the efficiency factor measures the fraction of the input optical power can be passed to the output, i.e., $Power_{out} = \gamma \cdot Power_{in}$. For multi-input or multi-output optical network, the concept can be used to measure the portion of a specific input to a specific output, or the worst-case/best-case scenario for any of the inputs to any of the outputs. In the following discussion, the efficiency factor means the worst-case efficiency factor, unless it is specified. The reader could derive the meaning from the context.

In a BDD-based optical network, the efficiency factor can be defined for the node as an optical switch (γ_v), edge as an input branch of a combiner/coupler ($\gamma_{(u,v)}$), path as a sub-network ($\gamma_{v \rightarrow u}$), or the whole network (γ_{net}). The path efficiency is calculated by multiplying the efficiency factors of all the components including combiners, couplers and switches, along the path. The network efficiency is defined as the minimal efficiency of all the paths from the network input to the network output. Our goal of the synthesis is to improve the *worst-case* network efficiency factor under a reasonable overhead and computational budget. We consider to optimize the worst-case efficiency for this determines the minimum required detection threshold of the photodetector; if optical amplifiers are used, the worst-case efficiency also determines the number of the optical amplifiers needed.

3.1 Combiner Elimination

Combiner elimination is a direct method to transform a combiner-intensive path which diminishes the light fast. As the combiner efficiency is inversely proportional to the number of combiner inputs, the idea of combiner elimination is to reconnect part of the inputs to a functionally equivalent copy of the original combiner output. The new copy is obtained by duplicating the BDD node (and its fan-out cone) that is connected to the combiner output. An example is shown in Figure 5, where node c is copied and the combiner connected to the input of c can be eliminated. It is not difficult to see that the single-path property is preserved after combiner elimination following the argument in Section 2.

Combiner elimination can be performed on any *internal* multi-input combiners to mitigate the decomposing number. We do not remove the combiners connected to the 1-terminal by copying the 1-terminal, as each separate terminal copy still has to connect to the very photodetector of the network, which has no effect on the efficiency factor.

Suppose the terminal combiner has $nTerm$ inputs. For some path p_i , let the original efficiency be $\gamma_{org}/nTerm$, where

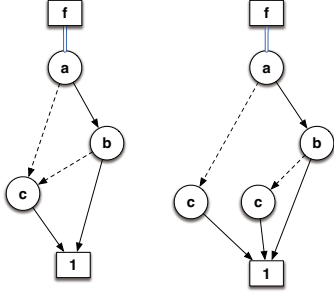


Figure 5: Example of combiner elimination.

γ_{org} is the worst-case efficiency factor from the top node to the node before reaching the terminal combiner. Given an nIn -input node u on the path, with $nCornTerm$ of the terminal combiner inputs belonging to its fan-out corn, we create $nCopy$ copies of this node. The overhead of each copy is the number of the node in the fan-out corn (including the root of the corn), denoted by $nCornSize$. The new efficiency factor can be estimated as

$$\frac{nIn}{nTerm + nCopy \cdot nCornTerm} \cdot \gamma_{org} \quad (2)$$

The ratio r over the original efficiency is thus

$$r(p_i, u) = \frac{nIn}{1 + nCopy \cdot nCornTerm / nTerm} \quad (3)$$

In the example of Figure 5, suppose 0 switching loss for the simplicity of this demonstration only. Node c and its fanout cone is copied. Therefore, the number of inputs for node c $nIn = 2$; the number of inputs to the terminal $nTerm = 2$; the number of inputs to the terminal that belongs to the copied fanout corn $nCornTerm = 1$; the number of copies $nCopy = 1$; and finally the overhead, i.e., the total number of the copied nodes, can be calculated to be the fanout corn size including the corn root node but excluding the terminal node $nCornSize = 1$. By Eq. 3, the the ratio over original is $4/3$. It can be verified that the original BDD has an efficiency factor of $1/4$ (for path $a \rightarrow c \rightarrow 1$ and $a \rightarrow b \rightarrow c \rightarrow 1$). The efficiency after this duplication becomes $1/3$ (for all three paths from a to 1). The ratio correctly reflects the effect of the duplication.

We want to make sure the ratio is at least greater than 1. At the same time, we want to limit the increased overhead with duplicated nodes. The idea is to evaluate nodes that are closer to the bottom, which generally have smaller fanout corn, then moving upwards, until the overhead budget is reached. Given an initial BDD, the expected minimum benefit ratio *Benefit* upon one copy, and the area overhead budget *Budget*, the algorithm is shown in Algorithm 1.

Algorithm 1 Combiner Elimination

```

1: Overhead  $\leftarrow$  0
2: for each critical path  $p_i$  starting from the weakest do
3:   for each edge  $(u, v)$  on  $p_i$ , bottom to top do
4:     Compute estimated benefit  $r(p_i, v)$ 
5:     Compute overhead  $nCornSize(v)$ 
6:     if Overhead +  $nCornSize > Budget$  then
7:       continue ▷ try next
8:     else if  $r(p_i, v) > Benefit$  then
9:        $v' \leftarrow$  Copy  $v$  and its fanout corn.
10:      Reconnect  $u$  to  $v'$ .
11:      Overhead  $\leftarrow$  Overhead +  $nCornSize$ 

```

To make sure the efficiency for the critical paths (or weak paths) crossing a certain node can be improved using combiner elimination, we need to consider all the critical paths in the network.

Critical paths are the paths that contribute to the worst-case power reduction, which can be calculated in the similar manner as the shortest paths for directed acyclic diagrams (DAGs) and the only difference is the recurrence relation [15, 16]. We note that the worst-case efficiency cascaded from the top till reaching a node v is equal to the minimum of the worst-case power efficiency of its parent node times the switching efficiency and combiner efficiency related to node v . Formally, given a BDD (V, E) , the recurrence of the efficiency of a BDD node v is

$$\gamma(v) = \min_{(u,v) \in E} \{ \gamma(u) \cdot \gamma_{sw}(v) \cdot \gamma_{cb}(v) \} \quad (4)$$

where $\gamma_{sw}(v)$ and $\gamma_{cb}(v)$ is the switching efficiency and combiner/coupler efficiency related to node v , respectively. The base case is $\gamma(v) = 1$, for the top node v . During this procedure, the combiner/coupler efficiency is estimated to be the efficiency in the mathematical expectation, i.e., 1 over the number of parents of v , hence $\gamma_{cb} = 1/|(u, v) \in E|$. A second approach, coupler assignment, will handle the nuance more directly.

3.2 Coupler Assignment

The second technique is coupler assignment, which improves the power efficiency by redistributing the optical power directly. Basically, we assign the efficiency factors of the edges feeding into a multi-input combiner by using the couplers we discussed in Section 2. The general coupler assignment problem can be formulated as a polynomial programming problem as follows

$$\text{Maximize } \gamma_{net} = \min_i \{ \gamma_i \cdot \prod_{j: e_j \in p_i} x_j \} \quad (5)$$

$$\text{s.t. } \sum_{i \in In(n)} x_i = 1, \forall n \quad (6)$$

$$0 \leq x_i \leq 1, \forall i \quad (7)$$

The objective is the efficiency factor of the whole PIC network, which is defined by the minimum of all the path efficiency factor. Each variable x_j represents an assignment of coupling efficiency for the edge e_j on path p_i . The path efficiency of p_j is calculated as the product of the coupling efficiency x_j 's times γ_i , the efficiency factor related to the other parts, e.g., the switch drop port efficiency. Eq. 6 represent the rule of power conservation; i.e., the sum of the efficiency factor of the input ports to a coupler is 1, which we call the *node constraint*. Eq. 7 simply states the efficiency factor should be positive and smaller than 1. The max-min objective can be transformed by adding a dummy variable f and add the constraints that

$$f \leq \gamma_i \cdot \prod x_j, \forall i \quad (8)$$

We call it the *path constraint*. Note that the polynomial programming problem can be solved by semidefinite programming (SDP) relaxation [17], but when the variable number is non-trivial, it usually takes long time to solve. Here we propose a much more scalable solution by using quadratically constrained programming (QCP), so that an efficient second-order cone programming solver, e.g., QCP solver in Gurobi [18], can be used. In QCP, we limit the number of variables in each path constraint to be less than 2, so each constraint is quadratic. The QCP optimization is run for a number of iterations until we cannot further optimize or have reached the overhead limit. This also allows us to have a prior control of the overhead related to the couplers in each iteration, as the overhead is proportional to the number of variables.

The selection of x_i 's is performed in an iterative manner. In each iteration, we choose a small set of x_i 's to optimize. The

algorithm for each iteration is shown in Algorithm 2. Similar to combiner elimination, we will start with the critical paths, from the most critical to the least critical. For each critical path, we evaluate each multi-input node v on the path, with the maximum and the minimum efficiency factors from the top node till reaching v . We name the ratio as the divergence factor $div(v)$

$$div(v) = \max_{\substack{(u_i, v), (u_j, v) \in E, \\ u_i \neq u_j}} \frac{\gamma_{top \rightarrow u_i}}{\gamma_{top \rightarrow u_j}}, \quad (9)$$

where $\gamma_{top \rightarrow u}$ denotes the worst-case efficiency factor from the top to node u . Intuitively, $div(u)$ estimates the potential of the improvement of reassigning the coupling efficiency of the inputs. If there are two paths which correspond to two of the inputs of v , one having a large (worst-case) efficiency factor and one a small (worst-case) efficiency factor, we can redistribute their power efficiency, so that the weaker path can become stronger.

If a node v is selected, we pick the two input edges (u_i, v) and (u_j, v) , that correspond to the maximum and minimum worst-case $\gamma_{top \rightarrow u}$ in the computation of $div(v)$ for reassignment. In Algorithm 2, for each unprocessed critical path ordered from the most critical to the least critical, we select two candidate nodes with the highest divergence factors. We will try to optimize the efficiency factors of their input combiners by re-assigning the efficiency factors. In Line 10, The path expression $PathExpr_j$ (which defines the product of all the variables along the path) will be updated for each of the affected paths. If any of such expression becomes non-quadratic during the updating, the problem cannot be solved with QCP so the updates are reversed. We then continue to the next critical path. At the end of the first for loop, the variables to be optimized for this iteration are decided and the quadratic path expressions are built. In Line 14 and 17, the path constraints and node constraints are formulated and will be processed by the QCP solver. As the rest of inputs are kept unchanged Hence, the 1 constant in constraint Eq. 6 is modified to be the sum of their original efficiency. When we decide the two variables for the path in analysis, we can write the path constraint in Eq. 8. We also need to add the path constraints for all the paths that can be affected by the assignment of the two variables. The search continues as long as all the path constraints remain quadratic.

The computation of the max-efficiency paths that cross node u , as required in the calculation of div , can be performed similarly as the computation of the critical paths by replacing max by min in the recursion of Eq. 4. We start from node v , following the edges reversely to the top node and update the efficiency factor. Due to the quadratic constraint and the overhead budget, each iteration is very fast. We conduct the algorithm for a fixed number of iterations until there is no chance to improve, i.e., the overhead budget is reached or the problem is no longer quadratic.

To realize the reassignment of two of the edges of selected nodes, we exploit the 3×1 coupler, two inputs for the two edges and the other one for to the combiner output of the inputs which are not reassigned.

Using the concept of the divergence factor, we can also assign the output ports of a 1×2 switch. Define γ_{thru} and γ_{drop} as the through and drop port efficiency factors. Generally, $\gamma_{thru} > \gamma_{drop}$. In this case, $div(u)$ is evaluated by $\gamma_{(low-child) \rightarrow (1-term)}$ and $\gamma_{(high-child) \rightarrow (1-term)}$. We start from the critical paths, evaluate the div of each node, and flip the through and drop port if

$$\gamma_{(low-child) \rightarrow (1-term)} < \gamma_{(high-child) \rightarrow (1-term)}$$

The flip of the assignment can be trivially realized by inverting the controlling electrical signal of the switch.

Algorithm 2 Coupler Assignment QCP Formulation.

```

1:  $PathCons \leftarrow \emptyset, NodeCons \leftarrow \emptyset$ 
2:  $PathExpr_i \leftarrow 1, \forall i$   $\triangleright$  expression for each path constraint.
3: for each critical path  $p_i$  whose  $PathExpr_i$  is null do
4:   find two nodes  $v_1, v_2 \in p$  with the highest  $div$ 
5:    $u_a, u_b \leftarrow \operatorname{argmax}_{u_i, u_j} div(v_1)$   $\triangleright$  see Eq. 9
6:    $u_c, u_d \leftarrow \operatorname{argmax}_{u_i, u_j} div(v_2)$ 
7:   create four variables for  $(u_a, v_1), (u_b, v_1), (u_c, v_2), (u_d, v_2)$ 
8:   for each path  $p_j$  containing any of the four edges  $e_k$  do
9:      $x_k \leftarrow$  the variable to  $e_k$ 
10:     $PathExpr_j \leftarrow PathExpr_j \cdot x_k$ 
11:    if  $PathExpr_j$  is not quadratic then
12:      reverse all the changes related to the four variables
13:    break
14: for each path expression  $PathExpr_j$  do
15:    $PathCons \leftarrow PathCons \cup \{f \leq \gamma_j' \cdot PathExpr_j\}$ 
16: for each node  $v$  whose inputs are assigned  $x_i, x_j$  do
17:    $NodeCons \leftarrow NodeCons \cup \{x_i + x_j = x_{i,prev.val} + x_{j,prev.val}\}$ 
18: return  $PathCons, NodeCons$ 

```

4. EXPERIMENTAL RESULTS

We implemented the proposed methods in C++ with CUDD package [19], and tested it on an 8-core 3.4GHz Linux machine with a 32GB RAM. The following experiments were conducted on Microelectronics Center of North Carolina (MCNC) and International Workshop on Logic and Synthesis (IWLS) benchmarks [20]. Gurobi QCP solver [18] was used in coupler assignment. As different BDD-reordering heuristic may result in different initial BDD, to make a fair comparison with the state-of-the-art method [11], we applied a common BDD re-ordering heuristic CUDD_REORDER_SYMM_SIFT available in CUDD.

The two techniques, combiner elimination and coupler assignment (including switch output assignment) were performed iteratively for each benchmark and stop if cannot make any further improvement with the overhead budget. As the switching efficiency assignment has no overhead as coupler assignment and combiner elimination, it is performed at the end of each iteration of coupler assignment and combiner elimination. The best improvement along with the total time for the optimization is reported.

The optical performance in the experiments are all based on the models discussed in Sections 2. In Table 1, the first three columns summarized the benchmark information, the number of primary inputs (#PI) and number of primary outputs (#PO). The next four columns show the number of optical switches and worst-case network efficiency (net_eff) of the previous method [11] and our method. The number of switches also counts for the number of optical couplers used in our method. As noted in Sections 2, a 2×1 coupler is twice the size of an optical switch. The overhead percentage in terms of the increase of the number of optical switches (incl. the couplers) for each benchmark is summarized in Column 8. We limit the overhead to be less than 20%. As was calculated in the last row, the average overhead percentage is a small 7.63%.

The ratio of the worst-case network efficiency is shown in Column 9. As can be seen, for all the benchmarks, the power efficiency has been improved. The average ratio for all the benchmarks is 27.02X. The CPU time of our method is shown in Column 10. The longest runtime is 14.5s and the average is 1.88s.

To take a closer look at the effects of our method, Figure 6 plots the number of the paths belonging to the following seven efficiency ranges, $\{0 \sim 10^{-6}, 10^{-6} \sim 10^{-5}, \dots, 10^{-1} \sim 1\}$. The curves show a fitted distribution of the bars. The results of previous method [11] is colored in red, the results of

Table 1: Experimental Results

benchmark	#PI	#PO	[11]		ours		comparison		time (s)	electrical power (uW)	optical power (uW)
			#SW	net_eff	#SW	net_eff	overhead(%)	net_ratio			
cps	24	109	2288	1.62E-07	2536	5.41E-06	10.84	33.44	1.90	2.95E+02	1.27E-04
spla	16	46	985	5.27E-05	1014	3.95E-04	2.94	7.50	0.35	2.90E+02	5.07E-05
pdc	16	40	981	8.09E-05	1049	6.83E-04	6.93	8.44	0.20	3.03E+02	5.25E-05
alu4	14	8	855	8.32E-07	968	6.35E-06	13.22	7.63	2.28	4.36E+02	4.84E-05
misex3c	14	14	872	1.11E-06	897	1.14E-05	2.87	10.25	0.90	2.55E+02	4.49E-05
table3	14	14	1680	1.47E-06	1816	2.25E-05	8.10	15.30	0.45	4.45E+02	9.08E-05
apex5	117	88	1431	2.50E-05	1658	2.95E-04	15.86	11.80	0.95	2.28E+02	8.29E-05
table5	17	15	1879	4.16E-07	1947	3.56E-05	3.62	85.45	0.60	3.43E+02	9.74E-05
vda	17	39	1152	6.21E-05	1198	8.89E-04	3.99	14.32	0.25	1.66E+02	5.99E-05
k2	45	45	2282	2.98E-06	2541	9.59E-05	11.35	32.19	0.61	1.99E+02	1.27E-04
seq	41	35	2452	1.67E-07	2572	1.27E-06	4.89	7.55	2.75	7.80E+02	1.29E-04
steppermotordrive	29	29	546	3.73E-05	586	4.12E-04	7.33	11.03	1.15	5.39E+01	2.93E-05
count	35	16	216	4.40E-04	216	1.25E-02	0.00	28.42	0.05	4.71E+01	1.08E-05
dalv	75	16	1692	6.38E-08	1832	3.62E-06	8.27	56.73	14.50	2.81E+02	9.16E-05
i2c	147	142	1850	1.83E-06	1992	2.41E-04	7.68	131.48	1.20	3.01E+02	9.96E-05
i8	133	81	2456	2.60E-05	2823	3.88E-04	14.94	14.91	1.75	4.18E+02	1.41E-04
simple_spi	148	144	1499	2.08E-04	1511	1.16E-03	0.80	5.59	1.25	3.61E+02	7.56E-05
stpmotor	29	29	546	3.73E-05	586	4.12E-04	7.33	11.03	1.05	5.39E+01	2.93E-05
frg2	143	139	2099	2.23E-06	2394	4.55E-05	14.05	20.39	3.55	3.18E+02	1.20E-04
average							7.63	27.02	1.88	2.93E+02	7.93E-05

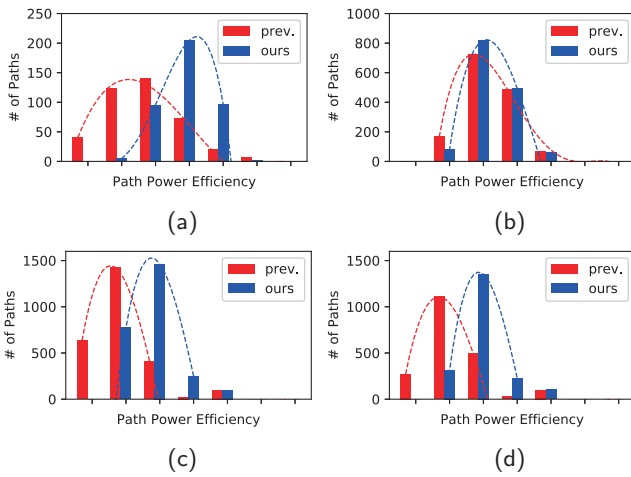


Figure 6: Comparison of the loss distribution of [11] and the proposed method. (a) benchmark cps (PO.27), (b) alu4 (PO.7) (c) dalu (PO.27), (d) dalu (PO.9)

our method is in blue. As can be observed, the distribution curves move from the lower efficiency zone to higher applying our method, which results in a more balanced distribution of power efficiency.

The second set of experiments compares the power consumption for the benchmark circuits implemented by the electronics and the optics. Note that the power for the optical implementation is estimated for switching the optical switches and is different from the *optical power* (laser power) we discussed previously. The electrical implementation was synthesized in Design Compiler in the gsc145nm library; the clock frequency was set to be 1GHz. The optical implementation is based on the models discussed in Sections 2. Each switch is implemented by a microring resonator, which consumes 50fJ for switching. It can be seen that the power consumption of the optics is several-magnitude smaller. It should be noted that for the power calculation, we did not consider the power consumption spent in aligning the microrings using heating, as in more advanced optical technology, the microrings would be able to become well-aligned during manufacturing.

5. CONCLUSION

In this work, we study the long-neglected optical power depletion problem in previous PIC synthesis, and propose two techniques, combiner elimination and coupler assignment, to

address this problem. The experiments where various sources of optical power depletion are considered, shows the efficacy of our method of generating optical-power efficient PICs, which also helps to build a much more robust and scalable integrated photonic system.

Acknowledgement

The authors acknowledge the Multidisciplinary University Research Initiative (MURI) program through the Air Force Office of Scientific Research (AFOSR), contract No.FA 9550-17-1-0071, monitored by Dr. Gernot S. Pomrenke.

6. REFERENCES

- [1] R. Wille, B. Li, U. Schlichtmann, and R. Drechsler, "From Biochips to Quantum Circuits: Computer-aided Design for Emerging Technologies," in *Proceedings of the 35th International Conference on Computer-Aided Design*. ACM, 2016.
- [2] Z. Ying, Z. Wang, S. Dhar, Z. Zhao, D. Z. Pan, and R. T. Chen, "On-chip microring resonator based electro-optic full adder for optical computing," in *CLEO: QELS-Fundamental Science*. Optical Society of America, 2017.
- [3] Y. Ye, J. Xu, B. Huang, X. Wu, W. Zhang, X. Wang, M. Nikdast, Z. Wang, W. Liu, and Z. Wang, "3-D Mesh-based Optical Network-on-Chip for Multiprocessor System-on-Chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2013.
- [4] D. Ding, B. Yu, and D. Z. Pan, "GLOW: A Global Router for Low-power Thermal-Reliable Interconnect Synthesis using Photonic Wavelength Multiplexing," in *Design Automation Conference (ASP-DAC)*, 2012 17th Asia and South Pacific. IEEE, 2012.
- [5] C. Sun, M. T. Wade, Y. Lee, J. S. Orcutt, L. Alloatti, M. S. Georgas, A. S. Waterman, J. M. Shainline, R. R. Avizienis, S. Lin *et al.*, "Single-chip microprocessor that communicates directly using light," *Nature*, 2015.
- [6] P. Zhou, L. Zhang, Y. Tian, and L. Yang, "10 ghz electro-optical or/nor directed logic device based on silicon micro-ring resonators," *Optics letters*, 2014.
- [7] L. Yang, L. Zhang, C. Guo, and J. Ding, "Xor and xnor operations at 12.5 gb/s using cascaded carrier-depletion microring resonators," *Optics express*, 2014.
- [8] Y. Tian, L. Zhang, J. Ding, and L. Yang, "Demonstration of electro-optic half-adder using silicon photonic integrated circuits," *Optics express*, 2014.
- [9] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund *et al.*, "Deep learning with coherent nanophotonic circuits," *Nature Photonics*, 2017.
- [10] C. Condrat, P. Kalla, and S. Blair, "Logic Synthesis for Integrated Optics," in *Proceedings of the 21st edition of the great lakes symposium on Great lakes symposium on VLSI*. ACM, 2011.
- [11] R. Wille, O. Keszcse, C. Hopfmuller, and R. Drechsler, "Reverse BDD-based Synthesis for Splitter-free Optical Circuits," in *Design Automation Conference (ASP-DAC)*, 2015 20th Asia and South Pacific. IEEE, 2015.
- [12] E. Timurdogan, C. M. Sorace-Agaskar, J. Sun, E. S. Hosseini, A. Biberman, and M. R. Watts, "An ultralow power athermal silicon modulator," *Nature communications*, 2014.
- [13] Z. Wang, X. Xu, D. Fan, Y. Wang, and R. T. Chen, "High quality factor subwavelength grating waveguide micro-ring resonator based on trapezoidal silicon pillars," *Optics letters*, 2016.
- [14] R. E. Bryant, "Symbolic Boolean Manipulation with Ordered Binary-Decision Diagrams," *ACM Computing Surveys (CSUR)*, 1992.
- [15] J. Y. Yen, "Finding the k shortest loopless paths in a network," *management Science*, 1971.
- [16] D. Eppstein, "Finding the k shortest paths," *SIAM Journal on computing*, 1998.
- [17] P. A. Parrilo and B. Sturmfels, "Minimizing polynomial functions," *Algorithmic and quantitative real algebraic geometry, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 2003.
- [18] G. Optimization, "Gurobi optimizer," <https://www.gurobi.com/downloads/gurobi-optimizer>, accessed: 2017-05-30.
- [19] F. Somenzi, "CUDD: Colorado university decision diagram package," <http://vlsi.colorado.edu/~fabio/CUDD/>, accessed: 2015-09-30.
- [20] "IWLS 2005 benchmarks," <http://iwls.org/iwls2005/benchmarks.html>, accessed: 2015-09-30.