PARS: A Power-Aware and Reliable Control Plane for Silicon Photonic Switch Fabrics

Mohammad Amin Mahdian, Ebadollah Taheri, and Mahdi Nikdast

Department of Electrical and Computer Engineering, Colorado State University, USA

Abstract—Fabrication-process variations and run-time thermal variations in silicon photonic (SiPh) switching devices present inherent uncertainties, diminishing the device and switch fabric performance. Such variations are often alleviated by re-tuning SiPh devices at the cost of significant power consumption. In this paper, we present PARS, a Power-Aware and Reliable control plane for SiPh Switch fabrics. By implementing efficient adaptive strategies, PARS minimizes the power necessary for the calibration and re-tuning of switching devices. We synthesized PARS's hardware and modeled our PARS controller using Beneš switches of different radices. Results show that PARS achieves at least more than 25% in power saving with less than 1% area and power overhead, and a maximum latency of 44 ps, without any effect on the operation frequency of the switch fabric.

I. INTRODUCTION

Silicon photonics (SiPh) has enabled a paradigm shift in high-performance computing (HPC) and data center systems by providing higher bandwidth and throughput while enhancing energy efficiency [1]. SiPh switch fabrics, as a stepping stone of this shift, play a vital role in facilitating low-latency and high-bandwidth communication within these systems, making the realization of high-radix SiPh switch fabrics imperative to address the growing demands in HPC and data centers. However, the realization of high-radix SiPh switch fabrics has been slowed down by several challenges in their design, fabrication [2], and reconfiguration (i.e., control) [3]. Moreover, some of these challenges (e.g., process and thermal variations) are inherent and cannot be solely addressed by device engineers, motivating the need for co-design solutions that incorporate device-level insights into the system-level designs (e.g., a variation-aware control plane for SiPh switches).

SiPh switch fabrics are inherently prone to fabrication-process variations (FPVs) and thermal variations (TVs) [2], [4], [5]. These variations impose scalability limitations, power losses, and crosstalk noise in SiPh switch fabrics. Several techniques have been proposed to alleviate the impact of such variations in SiPh switches, including the design of variation-tolerant devices [2] and different tuning mechanisms [4]. However, existing solutions lack scalability and often impose high power overhead when applied to high-radix SiPh switch fabrics. In this paper, our main objective is to tackle the aforementioned issues through the lens of the control plane. The control plane serves as the brain of the switch, responsible for crucial tasks like scheduling, switch (re)configuration, discovering network topology, and calculating routing paths
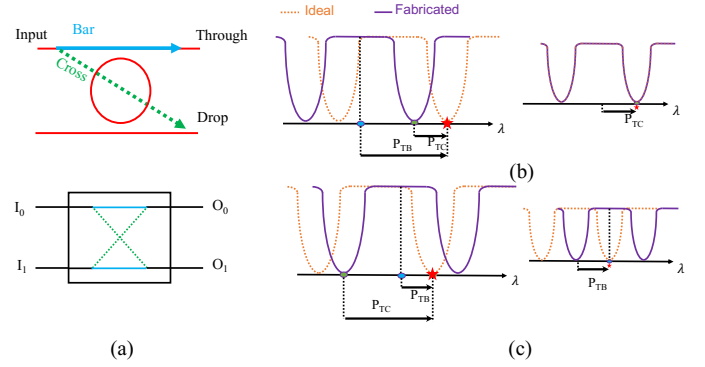
Fig. 1. (a) An MRR switching element. (b) The effect of variations on the through-port response of the switching element. The ideal response is the orange-dotted line whereas the solid-purple line shows the actual response experiencing the variations when P_{TC} <P_{TB} and cross default state consumes less power. (c) Showcasing the condition when P_{TB} <P_{TC} and Bar default state consumes less power. The star shows the wavelength of operation. Subfigures on the right-hand side show the response after trimming.

and protocols. Our focus lies in highlighting the significance of determining the optimal (e.g., power efficient) routing path while considering variations. We design a control plane, called PARS, using the proposed methodology and subsequently discuss the results through a case study of a Beneš network, demonstrating the potential advantages of our innovative approach through simulations and hardware synthesis.

II. PARS: PROPOSED POWER-AWARE AND RELIABLE CONTROL PLANE

SiPh Microring Resonators (MRRs) based switching elements (see Fig. 1(a))—considered as an example in this paper—offer a compact footprint to realize area-efficient switch fabrics. However, MRRs are susceptible to FPVs and TVs, as discussed in [2], [4]. For example, it was shown in [6] that a miniature change of one nanometer in an MRR's waveguide thickness can cause approximately a two-nanometer shift in the resonant wavelength of the MRR. Such a resonant wavelength shift can introduce significant crosstalk noise and power losses in the SiPh switch fabric, ultimately posing reliability concerns and scalability limitations.

The impact of FPVs and TVs (referred to as variations hereafter) on the response of an MRR-based switching element is shown in two different scenarios in Figs. 1(b) and 1(c). The ideal response represents the design-time response. However, due to variations, the response of the MRR suffers from a blue shift and a red shift, as it is shown in Figs. 1(b) and 1(c), respectively. To address this issue, the resonant wavelength of the MRR can be readjusted by applying a tuning mechanism
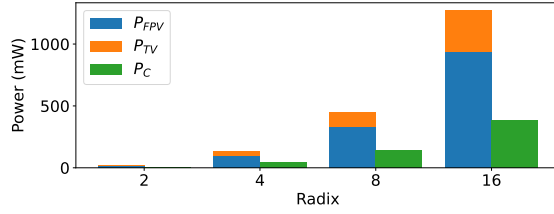
Fig. 2. Power consumption to compensate for variations (P_{FPV} and P_{TV}) and switching power (P_C) in a Beneš switch with different sizes.

Fig. 3. An example of power-efficient configuration in a 4×4 Beneš switch.

(e.g., thermal tuning) and consuming tuning power—a.k.a. trimming power—to readjust the displaced response to be at the wavelength of operation. We consider P_{TB} (P_{TC}) to be the trimming power required to readjust the MRR's response to be at the through (drop) port, a.k.a. the Bar (Cross) state.

Note that $P_{TB} \neq P_{TC}$ in MRR-based switching elements. Also, the process of switching the signal to the drop port consumes power (considered as P_C), causing an imbalance in the switching power between the Bar and Cross states. Due to variations, it is possible that P_{TB} and P_{TC} exceed P_C (i.e., the trimming power exceeds the switching power). Also, due to variations, each switching element in a switch fabric may consume different P_{TB}, P_{TC}, and P_C. Such a disparity among these power consumption values highlights an opportunity to design a switch control plane that can optimize the overall power consumption. By considering the power of each state, the controller can minimize power consumption by keeping the switching elements in the state that consumes minimum trimming and switching power. As a result, certain switching elements may have a default Bar state, while others may be in the Cross state by default.

To illustrate the impact of power penalty resulting from neutralizing variations in comparison to switching power consumption, we consider an example of a Beneš network as a case study in this paper. Fig. 2 presents a comparison between trimming power due to variations (P_{FPV} and P_{TV}) and switching power (P_C) in the Beneš switch fabric with different sizes. The total number of MRR switching elements in a Beneš network is determined using $N \log_2 N - \frac{1}{2} N$, where N denotes the switch radix. As the number of switching elements increases, higher power consumption is required for switching and trimming under variations. Fig. 2 depicts three different sources of power consumption comprising the total power consumption in each fabric. The highest power is allocated to compensate for variations. Conversely, on average, the power consumption for routing (i.e., switching) is relatively lower. Note that the impact of FPVs on the resonant wavelength of MRR switching elements is derived from [2], while the power consumption for mitigating thermal variations is based on [5]. Considering Fig. 2, we can observe an exponential rise in the fabric power consumption as the switch fabric scales up. This highlights the importance of developing a control plane capable of making more efficient power-aware decisions. In the upcoming subsections, we present the proposed configuration- and trimming-aware routing mechanisms used in our PARS control plane to improve power efficiency.
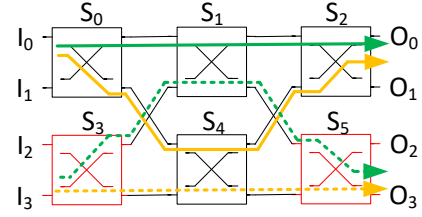
A. Configuration-Aware Routing Mechanism

In multistage networks, e.g. Beneš, there exist multiple routes between an input and an output port. Fig. 3 depicts two distinct routes leading to output O_0 from input I_0: the green and yellow routes. The green route does not require any switching power (when no variations are considered), as it can be achieved by maintaining S_0, S_1, and S_2 in the Bar state. On the other hand, the yellow route requires S_0 and S_2 to be in the Cross state, which consumes more power. The configuration-aware mechanism aims at selecting a route that minimizes the number of switching elements set to the state with a higher power consumption (Cross state in this example). This assumption is based on the initial design (i.e., default state) of the switching elements. However, in some cases, certain switching elements may be considered to have a Cross default state to reduce trimming power in PARS's trimming-aware mechanism, as discussed next.

B. Trimming-Aware Routing Mechanism

As we discussed earlier in this section, variations can lead to different trimming powers per switching element. In specific scenarios, it might be more power-efficient to alter the default state of the switching element, to the state that consumes less power to perform the switching. In the 4×4 Beneš example in Fig. 3, we assume that S_3 and S_5 (in red) suffer from a variation similar to the case shown in Fig. 1(a), where it is more power efficient to consider Cross to be their default state. In such a scenario, the dashed-green route requires lower trimming power than the dashed-yellow route for I_3 to O_3. This is because both S_3 and S_5 require more trimming power to be readjusted to be at the Bar state than the Cross state.

C. PARS's Implementation

Switch controllers can operate in two different modes: offline and online. In the offline approach, pre-calculated routes are stored in a lookup table (LUT), as introduced in [7]. This method minimizes run-time delay because there is no computation overhead for online routing. However, the LUT may face memory limitations when scaled up for high-radix switches. Alternatively, in the online approach, each route is calculated online. For instance, in [8], a Clos scheduler is designed in three pipeline stages including output-port allocation, Clos routing, and configuration update. However, this method is limited to a specific topology and lacks scalability. To address this, HyCo was proposed in [3] to reduce operation latency by employing a Bloom filter, enabling an online multi-agent controller that is topology independent.

PARS is compatible with any controller using either online or offline modes; however, the routing itself is not the focus of this paper. Therefore, we assume that all the possible routes are available to our PARS controller, allowing us to choose the optimized route among them. Algorithm 1 helps create a LUT through a search mechanism to find all the possible routes for each input-output combination.

Algorithm 1 Search configuration

R: $[s0, s1, s2, ...]$ Bar:0, Cross:1
for each c in all input-output combinations **do**
    Define a list: L_c
    for each R in all possible routes in this c (input-output scenario) **do**
        Add R to L
    Store L_c as possible route for c

In Algorithm 2, a search is performed for finding the optimal switching element configurations based on the default state of the switching elements among the available routes for each input-output combination, which are extracted from the LUT generated by Algorithm 1. Within this algorithm, PARS determines the most viable default state of the switch based on the variations affecting the switching elements. Subsequently, the route is determined in a way that it traverses through the switching elements with the default state consuming the lowest trimming power, denoted as R_O. Note that the decision is made by a simple XNOR operation (see Section III-B for hardware overhead). The complexity of Algorithm 2 depends on the number of routes for an input-output mapping.

Algorithm 2 PARS algorithm to find optimized configuration

D: displaced vector based on FPV and TV ($[d_0, d_1, d_2, ..., d_S]$)
R_o: optimized configuration
for s in all switch cells **do**
    if Displaced state is closer to Bar state **then**
        $d_s \leftarrow 0$
    else if Displaced state is closer to Cross state **then**
        $d_s \leftarrow 1$
for each R in L_c **do**
    $R_D \leftarrow R$ XNOR D
    if R_D includes the minimum number of "1"s so far **then**
        $R_o \leftarrow R_D$

III. RESULTS AND DISCUSSIONS

A. Switch Power Analysis

We consider a case study of a Beneš network (as an example only; PARS can be applied to any switch fabric) with different scales to show PARS's impact on switch power consumption under various random FPVs and thermal variations. Our power modeling to tolerate variations is based on [2] and [5]. The normalized results for the two routing mechanisms are compared with the baseline (i.e., without a power-aware routing) in Fig. 4 for three different switch fabric radices. In this comparison, the trimming ($P_T = P_{FPV} + P_{TV}$) and switching (P_C) power are separated using, respectively, a lighter and darker color on the same bar. The blue bars show the power consumption of the configuration-aware routing mechanism (CA in the figure)
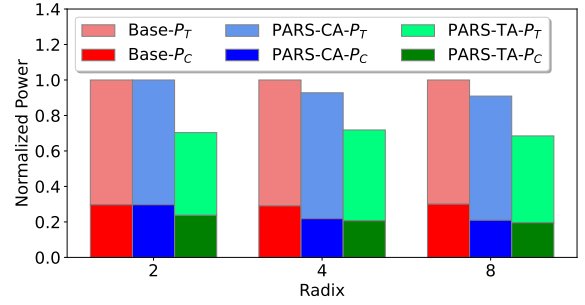

Fig. 4. Comparison of the normalized power consumption of our case study switch fabric (Beneš) with three different radices. The performance of the PARS controller can be seen in the green bar.

TABLE I
AREA AND POWER ANALYSIS OF THE CONTROLLER LOGIC

	2×2	4×4	8×8
Area (μm^2)	1	40	8306
Normalized area (%)	0.00016	0.0011	0.046
Power (μW)	0.045	1.6	310
Normalized power (%)	0.000219	0.0013	0.075
Delay (ps)	4	16	44

while the green bars show the power consumption of the trimming-aware mechanism (TA in the figure). Overall, PARS-TA archives at least 28.1% power improvement for all the switch radices. Moreover, we observe that power efficiency becomes more prominent as the radix of the switch increases. This is because the number of switching elements whose state are optimized by PARS increases. Consequently, although configuration-aware routing has shown some improvements (e.g., 9.1% for the 8×8 switch), the trimming-aware routing results in greater cumulative power savings, leading to a significant reduction in the overall power consumption. These findings further affirm the effectiveness of the trimming-aware routing mechanism in a switch fabric while improving the fabric tolerance to variations.

B. PARS's Hardware Analysis

To assess the hardware implementation overhead of PARS, we implemented the PARS controller module in Verilog and utilized Cadence Genus under 15 nm technology to estimate its area, power, and latency overhead. The results are summarized in Table I, where the area and power of the PARS controller are normalized to the switch. Area normalization is based on [9] and power normalization is based on our power modeling discussed earlier. The area and power overhead are negligible (<1%) and the latency is smaller than 44 ps.

IV. CONCLUSION

In this paper, we presented two power-aware control mechanisms to alleviate the impact of variations in SiPh switch fabrics. By implementing these mechanisms, we achieved a significant power consumption improvement of 28.1% in an 8×8 Beneš switch fabric. Our work in this paper highlights the essential role of electronic controllers in improving the performance of SiPh switch fabrics.

REFERENCES

[1] E. Taheri *et al.*, "ReSiPI: A reconfigurable silicon-photonic 2.5 d chiplet network with pcms for energy-efficient interposer communication," in *ICCAD*, 2022, pp. 1–9.

[2] A. Mirza *et al.*, "Silicon photonic microring resonators: A comprehensive design-space exploration and optimization under fabrication-process variations," *IEEE TCAD*, vol. 41, no. 10, pp. 3359–3372, 2021.

[3] D. Magalhães *et al.*, "Hyco: A low-latency hybrid control plane for optical interconnection networks," in *IEEE RSP*. IEEE, 2021, pp. 50–56.

[4] X. Chen *et al.*, "Simultaneously tolerate thermal and process variations through indirect feedback tuning for silicon photonic networks," *IEEE TCAD*, vol. 40, no. 7, pp. 1409–1422, 2020.

[5] R. Polster *et al.*, "Efficiency optimization of silicon photonic links in 65-nm cmos and 28-nm fdsoi technology nodes," *IEEE TVLSI*, vol. 24, no. 12, pp. 3450–3459, 2016.

[6] Bogaerts *et al.*, "Layout-aware variability analysis, yield prediction, and optimization in photonic integrated circuits," *IEEE JSTQE*, vol. 25, no. 5, pp. 1–13, 2019.

[7] F. Gohring *et al.*, "Design and modelling of a low-latency centralized controller for optical integrated networks," *IEEE Commun. Lett.*, vol. 20, no. 3, pp. 462–465, 2016.

[8] P. Andreades *et al.*, "Low latency parallel schedulers for photonic integrated optical switch architectures in data centre networks," in *ECOC*, 2017, pp. 1–3.

[9] Q. Cheng *et al.*, "Ultralow-crosstalk, strictly non-blocking microring-based optical switch," *Photonics Research*, vol. 7, no. 2, pp. 155–161, 2019.