Task Mapping-Assisted Laser Power Scaling for Optical Network-on-Chips

Yuyang Wang
Dept. of Electrical and Computer Engineering
University of California, Santa Barbara
California, U.S.A.
wyy@ece.ucsb.edu

Kwang-Ting Cheng
School of Engineering
Hong Kong University of Science and Technology
Clearwater Bay, Hong Kong
timcheng@ust.hk

***Abstract*—Energy efficiency of an optical network-on-chip (ONoC) largely relies on an effective laser power management strategy. Addressing the limitations of existing techniques, we propose a Task Mapping-Assisted Laser Power Scaling (TMALPS) framework to optimize the energy consumption and the application execution time of an ONoC. Through the combination of task mapping exploration and runtime laser power reconfiguration applied to a wide range of application benchmarks, our TMALPS framework achieves an average of 66% saving of the energy-delay product, compared to a baseline scenario where the optimization techniques are not applied. Significant improvement over existing techniques was also observed. The hardware overhead required to support our TMALPS framework is minimal with intelligent reuse of existing on-chip hardware resource.**

I. Introduction

The multi-processor system-on-chip (MPSoC) has been widely deployed in high-performance computing systems to accommodate traffic-intensive applications [1]. With the network complexity in modern MPSoCs ever growing, the optical network-on-chip (ONoC) was proposed to provide better bandwidth and energy efficiency than that of all-electrical implementations [2]. However, such energy efficiency can be compromised if the laser power is not well managed. The authors of [3] proposed to provide just enough laser power for each optical channel depending on its quality. Nevertheless, their solution lacks a reconfiguration mechanism under dynamic workloads. Recent advances in on-chip laser designs, and thus the improved laser power controllability [4], bring about traffic-adaptive reconfiguration strategies that switch the lasers *on* and *off* at application runtime [5]–[7]. Some of these strategies were reviewed in [8] and categorized as *coarse-grained (ON-OFF)* strategies since the lasers have solely two configurable states. Due to the large turn-on delay of the lasers [9], the energy saving obtained from the off state may be counteracted by the energy wasted when switching the lasers back on, and the application execution time may also be prolonged. To tackle this issue, a *fine-grained* strategy was proposed in [8], a.k.a. Dynamic Laser Power Scaling (DLPS), which further introduced a *standby* state and an *intermediate* state to the lasers in order to leverage a smaller switching delay. However, despite some improvement over the ON-OFF strategies observed for a few application benchmarks, the effectiveness of the DLPS strategy was minimal for many other application benchmarks since they do not have favorable traffic patterns. This fundamental limitation of the DLPS strategy, to be further elaborated in Section II-B, motivates us to explore adjustments to the traffic patterns so that laser power reconfiguration techniques can be maximally exploited.

In this study, we propose a Task Mapping-Assisted Laser Power Scaling (TMALPS) framework to jointly optimize the energy consumption and the application execution time of an ONoC. Task mapping is the process of assigning tasks to the processing units available on the MPSoC, so different mapping schemes result in different traffic patterns during application execution. Though well-studied for electrical network-on-chips [10]–[12], task mapping has only started to draw attention from ONoC researchers in recent years. The authors of [13] proposed an ONoC task mapping algorithm that optimizes the worst case crosstalk noise and optical loss and based on which, further optimizes the average laser power budget [14]. However, a minimum laser power budget does not necessarily lead to a minimum energy consumption if the application execution time is lengthy under that particular mapping scheme. Their energy computation method in [14] based on hop counting is over-simplified. In this study, we employed validated simulators which resolve task dependencies under arbitrary mapping schemes, and compute the energy consumption and latency of the optical network during application execution. The simulation results indicate that our TMALPS framework helps eliminate unfavorable traffic patterns through task mapping exploration, and thus significantly improves the effectiveness of runtime laser power reconfiguration techniques. By jointly optimizing the energy consumption and the application execution time of the ONoC, an average of 66% saving of the energy-delay product was observed for a wide range of application benchmarks, compared to a baseline scenario where the optimization techniques are not applied. Our TMALPS framework also demonstrated better scalability than previous techniques across the evaluated application benchmarks.

II. Background

A. Optical Network-on-Chip

An ONoC is a collection of optical interconnects that perform on-chip data communication between the processing units of an MPSoC. Various ONoC architectures have been proposed [15], many of which incorporated sophisticated topologies to address specific design considerations, such as higher link utilization, lower crosstalk noise, etc. In this study, we target an ONoC architecture based on a generic optical ring bus [16] to demonstrate our TMALPS framework.

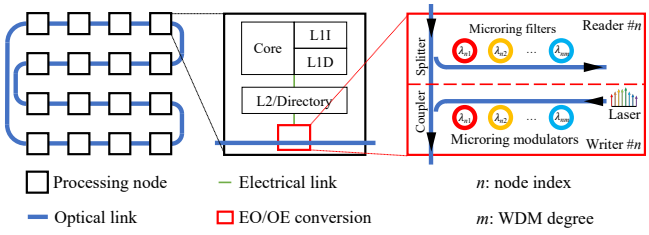


Fig. 1. Illustration of an MWMR ONoC architecture.

978-1-7281-2350-9/19/$31.00 ©2019 IEEE

As illustrated in Fig. 1, the processing nodes of the MPSoC are connected to an optical ring bus with a multiple-reader-multiple-writer (MWMR) architecture. Each node has both write and read access to the optical bus, achieved by silicon photonics microring modulators and filters [17], respectively. Wavelength division multiplexing (WDM) is enabled at each node. Therefore, for a network with n nodes, each with m multiplexed channels, the total number of microring modulators/filters required is $n \cdot m$.

In a typical MWMR configuration, all processing nodes are allowed to utilize all available wavelengths in order to increase the utilization. A wavelength allocation scheme is thus required to ensure mutual exclusion of the selected wavelengths. In this study, however, we assume that mutual exclusion is always satisfied, and focus on the joint impact of task mapping and runtime laser power reconfiguration. The models and assumptions for each network component will be detailed in Section IV.

B. Limitations of Existing Strategies

An ON-OFF strategy such as [5] switches the lasers off when there is no pending transmission. However, a non-trivial turn-on delay will be introduced when the laser is switched back on. As per [9], it can take as long as $\sim$10 ns for the laser to stabilize before any data transmission should take place, deteriorating both the energy consumption and the application execution time of the ONoC. DLPS was proposed in [8] as an extension to the ON-OFF strategy, where the laser can operate in one of the four modes, namely *off*, *standby*, *intermediate*, and *full-on*. In the standby mode, the laser is biased slightly above the threshold current to reduce the energy consumption yet maintain the capability of a fast turn-on (assumed 1 ns in [8]). In the intermediate mode, the laser provides just enough optical power (between zero and the maximum supported) to accommodate the requested data rate. The switching principle of the DLPS strategy is as follows:

- When traffic presents, the laser switches to either the intermediate mode or the full-on mode.
- When no traffic presents, the laser switches to the standby mode and stays for at most t_{idle} clock cycles before switching off, where t_{idle} is a tunable parameter for different applications.

Fig. 2 shows an example of the DLPS switching operations with t_{idle} set to 5 clock cycles. Under the DLPS strategy, the laser will not be switched off if the interval between two transmissions is smaller than t_{idle} cycles, and thus the second transmission can start with a faster switching.

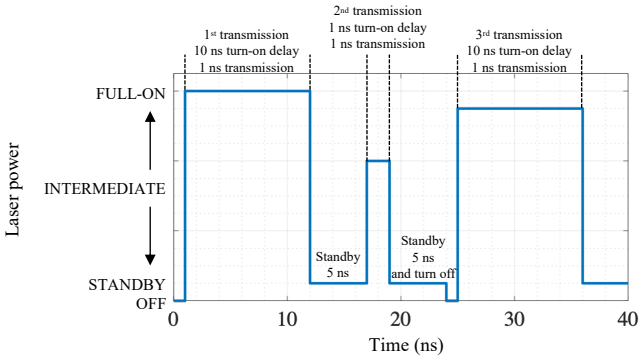


Fig. 2. An example of DLPS [8] with $t_{\text{idle}} = 5$.

While a larger t_{idle} reduces the occurrences of on-off switching, it also consumes energy during the standby period. A straightforward upper bound for t_{idle} can be derived as:

$$t_{\text{idle}} \leqslant t_{\text{max}} = \frac{\Delta_{\text{off}} \cdot P_{\text{on}} - \Delta_{\text{standby}} \cdot P_{\text{on}}}{P_{\text{standby}}} \tag{1}$$

where Δ_{off} and Δ_{standby} are the laser turn-on delay from the off mode and the standby mode, respectively; P_{on} and P_{standby} are the power consumption of the full-on mode and the standby mode, respectively. This inequality ensures that the energy consumption of the standby period is smaller than that of an off-on switching. The authors of [8] noticed that the energy saving of the DLPS strategy was no better than that of the ON-OFF strategy for many application benchmarks, due to that most of the transmission intervals in these applications are greater than t_{max}. In other words, it is simply better to switch the laser off immediately after each transmission, rather than wait in the standby mode for any period of time.

The lack of access to adjusting the traffic patterns limits the effectiveness of the DLPS strategy, which motivates our study to incorporate task mapping exploration to help resolve this fundamental limitation.

III. Problem Formulation

A. Mapping Scheme Encoding

The execution of an application on multiple processing units involves several preliminary steps, as illustrated in Fig. 3. *Task partition* divides the application into a finite number of tasks. An optional *task grouping* step rejoins certain tasks into task groups, followed by *task mapping* which assigns each task/task group to a processing unit. Each of these steps affects the communication among tasks/task groups, and thus results in different traffic patterns during application execution. Note that in this study, we treat task grouping as a special case of task mapping, where multiple tasks are mapped onto the same processing unit.

For an application with T tasks, we encode its mapping scheme into a vector:

$$\mathbf{M}_{1\times T} = [p_1, p_2, \ldots, p_T] \tag{2}$$

where the integer $p_i \in [1, P]$ is the processing unit to which task $\#i$ is assigned, and P is the total number of processing units. Consecutive tasks are allowed to be assigned to the same processing unit. In such cases, the communication between the two tasks becomes internal to the processing unit and is not carried out by the optical network.

B. Optimization Framework

In order to explore the impact of task mapping on the effectiveness of runtime laser power reconfiguration strategies, we propose our TMALPS framework as illustrated in Fig. 4. Our framework takes the application-specific task mapping schemes as the input population. The figure of merit for each mapping scheme is evaluated by two metrics, namely the system energy consumption and the application

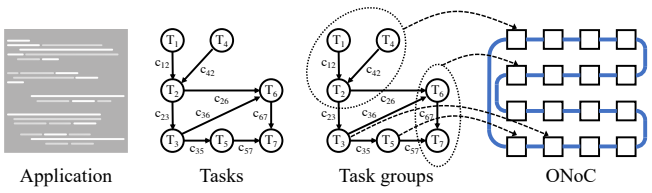


Fig. 3. Task partition, task grouping, and task mapping.

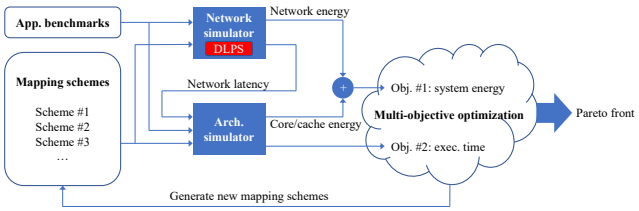


Fig. 4. TMALPS optimization framework.

execution time. Here, instead of having closed-form objective functions, both metrics are computed by simulating the application under the mapping scheme being evaluated. Specifically, a network simulator integrated with laser power reconfiguration features computes the network energy consumption and the latency. The latency information is then fed into an architectural simulator which computes the energy consumption of the cores/cache hierarchies and the overall application execution time. Finally, the overall system energy consumption and the application execution time are the two objectives both seeking to be optimized. The choice of simulators and some implementation considerations will be introduced in Section IV-C.

The fact that the input mapping schemes are vectors with all integer elements renders our TMALPS a multi-objective integer programming problem. In this study, we modified and employed a multi-objective particle swarm optimization (MOPSO) solver [18] to identify a Pareto front of the two objectives, where improvement in either one will require sacrificing the other. The solver also generates new mapping schemes for the next iteration of optimization, where the best schemes in both the current and the past iterations affect how far the new scheme deviates from the current one. Implementation details of the MOPSO solver can be found in [18] and will not be further elaborated here.

IV. Simulation Setup

A. Laser Turn-on Delay

In order to model the laser turn-on delay for various initial and final power states, we built a SPICE-compatible [19] laser transient model based on coupled rate equations [9] and calibrated it against the measured result of a fabricated laser diode reported in [20]. Fig. 5(a) shows a good match of the simulated and the measured laser output. Fig. 5(b) illustrates the laser turn-on delay for different initial and final power levels. It can be observed that the turn-on delay from the standby state to various final states can be limited within sub-nanosecond, while the turn-on delay from zero bias can take up to ∼10 ns. Inverse proportionality is observed between the turn-on delay and the final power level, which is also consistent with what reported in [9]. It is noteworthy that the actual delay incurred on the data transmission also depends on the clock rate of the system. For example, the granularity of the control logics under a 1 GHz clock is 1 ns, which means that the data transmission must be delayed by a full nanosecond even if the laser turn-on delay can be smaller than that.

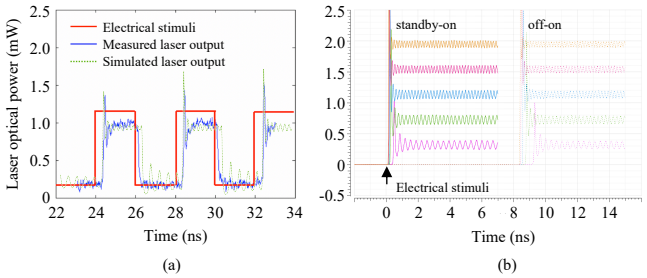


Fig. 5. (a) Laser transient model, and (b) turn-on delay simulation where different colors correspond to different final states.

B. Optical Link Power

We consider an ONoC based on MWMR optical links introduced in Section II-A. We further assume 5-channel WDM for each node, and a maximum data rate of 20 Gbps per channel yielding a total of 100 Gbps supported by each node. When a transmission is requested, a 5-channel optical link is activated between a writer and a reader, each channel with a power consumption of:

$$P_{\text{channel}} = P_{\text{laser}} + P_{\text{mod}} + 2P_{\text{tuning}} + P_{\text{TIA}} + P_{\text{SerDes}} \tag{3}$$

Here, P_{mod}, P_{TIA}, and P_{SerDes} are the power consumptions of the modulator driver, the receiver transimpedance amplifier, and the serialization/deserialization circuitry, which largely depend on the data rate. A decent analysis of such dependency is given in [21]. P_{tuning} is required for two microrings (a modulator and a filter) to align their resonance wavelengths to the laser wavelength. Low-power tuning techniques are proposed in [23]–[27], while a recent method reported in [23], evaluated on measured process variations, is adopted for the computation of P_{tuning} in this study.

P_{laser} is the electrical power consumed by the laser diode. It subjects to a wall-plug efficiency (WPE) to convert the electrical power into the optical power, which in turn must be high enough to overcome the power losses (PL) along the optical link, and eventually suffice the receiver sensitivity:

$$P_{\text{optical}} \cdot \prod_i \text{PL}_i \geqslant P_{\text{sensitivity}} \tag{4}$$

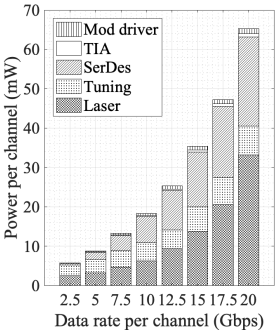
Here, $P_{\text{optical}} = P_{\text{laser}} \cdot \text{WPE}$. $P_{\text{sensitivity}}$ is another data rate-dependent term, for which a reasonable model is provided in [22] with respect to various bit error rate (BER) requirements. A summary of the power models featured in this study for optical components is given in Fig. 6(a), and based on which, the computed power consumption per channel w.r.t. different data rates are plotted in Fig. 6(b).

C. Simulator Choice

In our study, the laser turn-on delay is variable throughout the application execution. To the best of our knowledge, such feature is not supported by any readily available network simulators. Therefore, we implemented our in-house network simulator to compute the energy

Data rate-dependent	
P_{mod}, P_{TIA}, P_{SerDes}	[21]
$P_{\text{sensitivity}}$	[22]
Microring	[23]
Passing loss	0.2 dB
Insertion loss	0.5 dB
Drop-port loss	1 dB
Tuning efficiency	0.15 nm/mW
Laser	
Wall-plug efficiency	20% [20]
Other	
WG propagation loss	1 dB/cm [22]
Targeted BER	10^{-12}

(a)



(b)

Fig. 6. Power models for an optical channel.

and the latency of the ONoC. For a fair comparison, we validated our network simulator with parameters identical to those reported in [8], and was able to match their results. We then configured our network simulator with the parameters listed in Section IV-B and interfaced the network latency result to the architectural simulator. We employed JADE [28] as our architectural simulator since it supports customized task mapping. Table I lists its configurations.

V. Evaluation

We evaluated our TMALPS framework on a 64-node ONoC based on the MWMR architecture introduced in Section II-A. Twelve application benchmarks were obtained from the COSMIC Multiprocessor Benchmark Suite [29]. The applications are already partitioned into fine-grained tasks, as summarized in Table II. Each application comes with a default task mapping scheme based on load balancing. Detailed descriptions of the applications can be found in [29].

The solutions given by our TMALPS framework are compared to three other strategies. The baseline strategy does not include runtime laser power reconfiguration, and uses the default task mapping scheme. A simple ON-OFF strategy and the DLPS strategy [8] were also included for comparison, both using the default task mapping scheme as well.

A. Case Study on RS-encoder

We first show a case study on the RS-encoder application to demonstrate the interpretation of the simulation results. The application task graph is profiled under the ARM instruction set.

1) The Pareto front: The output from our TMALPS framework is a Pareto front of the system energy consumption and the application execution time. In Fig. 7, we normalize both metrics by the baseline strategy to show the improvement/overhead of various solutions. For RS-encoder, the simple ON-OFF strategy results in a large execution time overhead without providing any energy saving, due to the frequent on-off switching of the lasers. The DLPS strategy achieves considerable energy saving at the cost of a small execution time overhead by leveraging the standby mode. Our TMALPS framework identifies a Pareto front (black dots) among all evaluated mapping schemes (blue shade). Since the DLPS solution with the default mapping scheme is among the initial population fed into our optimizer, any new solution, if adopted by the Pareto front, is guaranteed to outperform the DLPS strategy in at least one of the two metrics.

2) Impact of task mapping exploration: Among the TMALPS solutions in Fig. 7, three special cases are highlighted. Solution ① achieves the highest energy saving (69%) at the cost of 34% longer execution time. Solution ② reduces the execution time by 16% while achieving less energy saving, nevertheless, still better than the baseline strategy in both metrics. Solution ③ is one of our TMALPS solutions that outperform the DLPS strategy in both metrics.

In Fig. 8(a), the system energy consumption and the execution time of these solutions are plotted w.r.t. different t_{idle} values. It can be observed that the curves experience several abrupt drops at $t_{\text{idle}} = 1, 9, 17$, and 25, for both the DLPS and our TMALPS solutions. By referencing Fig. 8(b), we noticed that these values correspond to the major bars in the histogram of possible transmission intervals for the RS-encoder application. As soon as t_{idle} reaches one of these critical values, those transmissions with the corresponding interval begin to take advantage of the standby mode without the need to switch off the laser.

The circled bar in Fig. 8(b) indicates that there are transmissions with an interval of 36 clock cycles in the DLPS solution. However, the DLPS energy curve does not drop at $t_{\text{idle}} = 36$. This means that the energy consumption of 36 standby cycles already exceeds that of an off-on switching. While the optimal t_{idle} is still 25 cycles for the RS-encoder, these transmissions with an interval of 36 cycles are always suffering from the off-on switching penalty, and thus counteract the effectiveness of the DLPS strategy. By changing the task mapping scheme of the RS-encoder application, our TMALPS solution ③ was able to eliminate such unfavorable transmissions, and thus provided a better optimization result compared to the DLPS strategy in both optimization targets. Our TMALPS framework also explores other task mapping schemes and provides a set of solutions with different emphasis. A solution with either lower energy consumption or shorter execution time can be identified based on the specific priority of a user.

TABLE I
Architectural Configurations.

Instruction set architecture	ARM	alpha
CPU frequency	1 GHz	1 GHz
L1 I/D cache size	32 kB/core	64 kB/core
L2 cache size	512 kB/core	2 MB/core
Cache line size	32 b	64 b
Cache coherency protocol	Directory-based MSI_MOSI	
Technology node (proc. die/mem. die)	40 nm/40 nm	180 nm/90 nm

TABLE II
Application Benchmarks Used in the Evaluation.

Synthetic application benchmarks		
Type	**# of tasks**	**# of communications**
Uniform	128	496
Hotspot	128	960
Realistic application benchmarks		
Name	**# of tasks**	**# of communications**
Cifar-10	33	50
FaceRecgonition	33	50
RS-encoder	141	140
RS-decoder	526	789
HPCG	2912	18323
Snap	2299	20844
FFT	15360	24064
Ultrasound	567	44746
RayTracing	25576	45468
MolecularDynamics	4334	194419

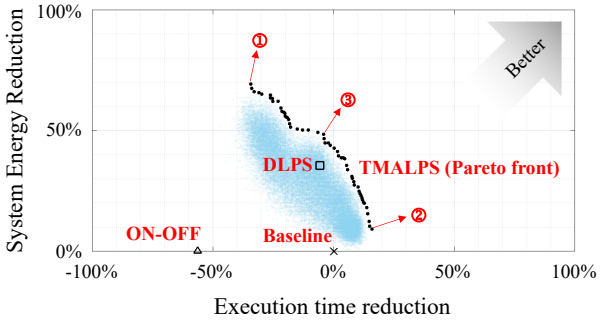


Fig. 7. Pareto front of system energy reduction and execution time reduction for RS-encoder given by our TMALPS framework.

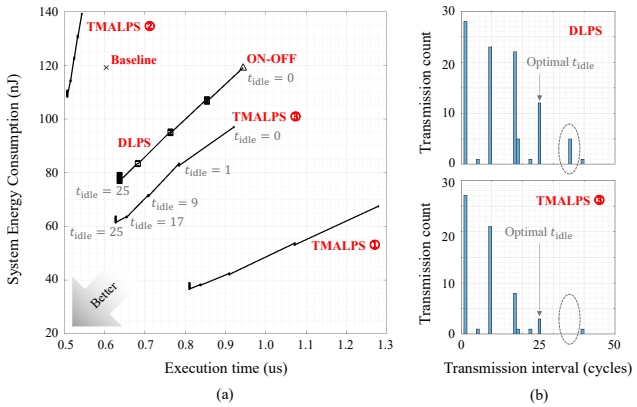


Fig. 8. Task mapping exploration improves DLPS effectiveness by eliminating unfavorable transmissions.

B. Evaluation on Other Benchmarks

We evaluated our TMALPS framework across a wide range of application models that are profiled under two instruction set architectures (ISA), namely ARM and alpha. For each application, we report the normalized energy-delay products (EDP) of different strategies, which are computed as the product of the system energy consumption and the application execution time normalized by that of the baseline strategy. Among the multiple Pareto solutions given by our TMALPS framework, the one with the smallest EDP is chosen to be the representative solution.

1) Overall EDP comparison: Fig. 9 summarizes the EDP comparison between our TMALPS framework and other strategies. Despite slight differences in some numbers, identical patterns were observed for both ISAs, that our TMALPS solution yields the lowest EDP in 23 out of 24 configurations. Overall, an average of 66% reduction of EDP can be achieved by our TMALPS framework for the twelve application benchmarks, compared to the baseline strategy where neither task mapping exploration nor laser power reconfiguration is involved.

2) Complementary behavior of ON-OFF and DLPS: An interesting observation arises when analyzing the second and the third bar of each group. For illustration purpose, we reordered the application benchmarks in Fig. 9 in a way that for the first half of the twelve benchmarks, the ON-OFF strategy performs worse than the baseline. As previously discussed, these applications are traffic intensive for which too frequently switching the lasers off could compromise the energy efficiency and the application execution time. The DLPS strategy, on the other hand, performs reasonably well on these applications, since bursty transmissions benefit from the standby mode. For the second half of the benchmarks, the simple ON-OFF strategy works considerably well, which indicates longer intervals between transmissions. Consequently, the improvement of the DLPS strategy over the ON-OFF strategy is quite limited, especially for RS-decoder, HPCG, Snap, and RayTracing. That being said, our TMALPS framework is scalable enough to further optimize the EDP in both scenarios.

3) Curse of dimensionality: For an ONoC with P processing units and an application with T tasks, Eqn. (2) of Section III suggests an exponential searching space of $O(P^T)$. Although our TMALPS framework employs a metaheuristic optimizer which does not require an exhaustive search, an adequate number of iterations should the optimizer run, otherwise, the optimality of the solutions can be impaired. However, the two objectives in our TMALPS framework are evaluated by the execution of multiple simulators, which renders the problem evaluation-expensive for applications with a large number of communications. This limitation starts to surface when evaluated on the MolecularDynamics application, where our TMALPS framework failed to bring improvement over the DLPS strategy under the alpha ISA. In future work, other optimizers should be explored to tackle high-dimensional, evaluation-expensive problems. A hybrid of multiple optimizers may help further enhance the scalability of our TMALPS framework. Alternatively, an online task mapping algorithm for ONoCs is also worth investigation.

C. Hardware Overhead

The hardware overhead required to support our TMALPS framework mainly comes from idle time tracking. Fortunately, devices on modern MPSoCs are typically controlled by finite state machines, which means that a multiple-bit STATE register already exists and can be reused to indicate the device idle time. The STATE register itself only reflects the instantaneous state of the device. In order to extract time information, the authors of [30] proposed a simple hardware modification to map the multiple-bit STATE register into a single-bit busy/idle indicator. The indicator is periodically polled by the processor running a power management software. The indicator sets to *busy* whenever there are pending tasks; it resets to *idle* only when the STATE register shows idle AND a polling signal has been received. Under such logic, if polling of the indicator returns *idle*, the device must have stayed in the idle state for at least one polling period. The implementation of the busy/idle indicator only requires several tens of logic gates, as reported in [30]. By setting the polling period to t_{idle} in this study, we are able to keep track of the transmitters that have been idle for threshold cycles and trigger the turn-off signal.

VI. Conclusion

In this study, we propose TMALPS, a Task Mapping-Assisted Laser Power Scaling framework for optical network-on-chips (ONoC). Our TMALPS framework combines task mapping exploration and runtime laser power reconfiguration to optimize the energy consumption and the application execution time of the ONoC. Simulation results across various application benchmarks show that our TMALPS framework is able to achieve an average of 66% saving of the system energy-delay product. With the assistance from task mapping exploration, our TMALPS framework scales well to many situations where previous techniques fail to bring improvement. The hardware overhead required to support our TMALPS framework can be well-controlled with intelligent use of software-level techniques and existing registers accessible to on-chip devices.

Acknowledgment

The authors would like to acknowledge the sponsorship of Research Grants Council of Hong Kong. The work described in this paper was partially supported by an RGC grant, GRF-16207917.

References

[1] W. Wolf *et al.*, "Multiprocessor system-on-chip (mpsoc) technology," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 27, no. 10, pp. 1701–1713, Oct. 2008.

[2] R. G. Beausoleil *et al.*, "Photonic architectures for high-performance data centers," *IEEE J. Sel. Top. Quantum Electron.*, vol. 19, no. 2, pp. 3 700 109–3 700 109, Mar. 2013.

[3] R. Wu *et al.*, "Variation-aware adaptive tuning for nanophotonic interconnects," in *2015 IEEE/ACM Int. Conf. Comput. Des.* IEEE, Nov. 2015, pp. 487–493.

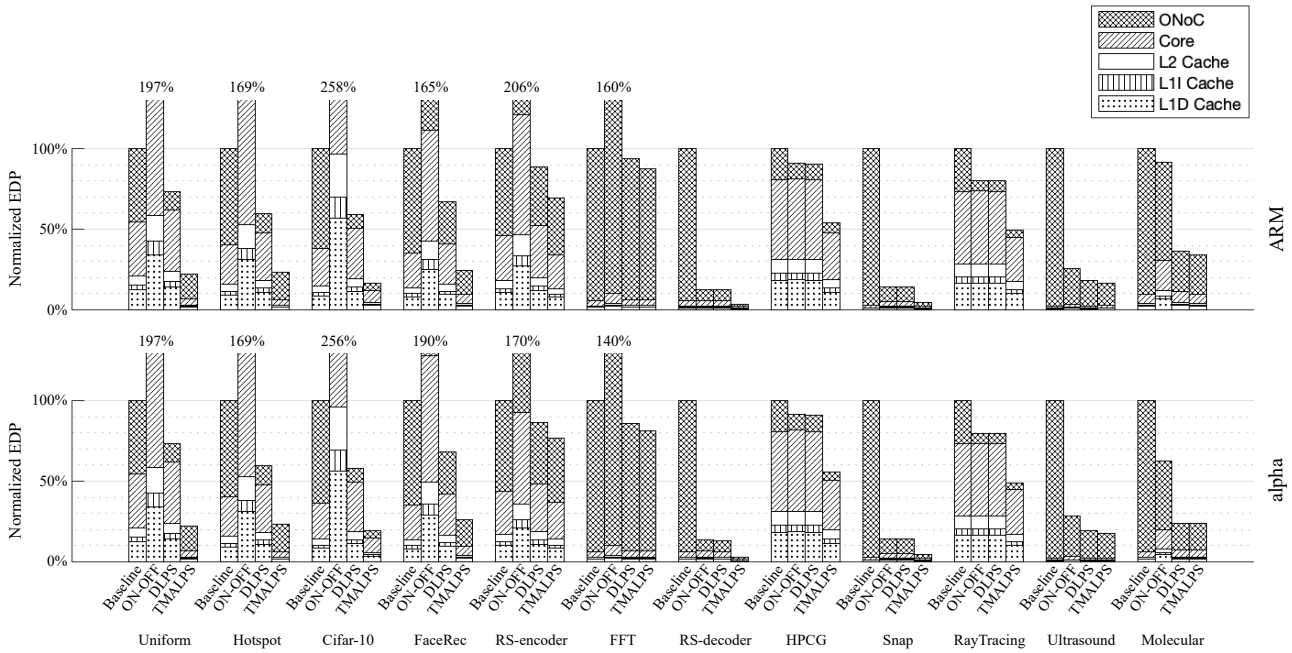


Fig. 9. Evaluation of our TMALPS framework on 12 application benchmarks profiled under 2 ISAs.

[4] M. J. Heck *et al.*, "Energy efficient and energy proportional optical interconnects for multi-core processors: Driving the need for on-chip sources," *IEEE J. Sel. Top. Quantum Electron.*, vol. 20, no. 4, 2014.

[5] C. Chen *et al.*, "Runtime management of laser power in silicon-photonic multibus noc architecture," *IEEE J. Sel. Top. Quantum Electron.*, vol. 19, no. 2, 2013.

[6] Y. Demir *et al.*, "Ecolaser: An adaptive laser control for energy-efficient on-chip photonic interconnects," in *Proc. 2014 Int. Symp. Low power Electron. Des. - ISLPED '14*, no. Figure 2. ACM Press, 2014, pp. 3–8.

[7] C. Chen *et al.*, "Managing laser power in silicon-photonic noc through cache and noc reconfiguration," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 34, no. 6, pp. 972–985, Jun. 2015.

[8] F. Lan *et al.*, "Dlps: Dynamic laser power scaling for optical network-on-chip," in *2017 22nd Asia South Pacific Des. Autom. Conf.* IEEE, Jan. 2017, pp. 726–731.

[9] L. A. Coldren *et al.*, *Diode lasers and photonic integrated circuits*. John Wiley & Sons, 2012, vol. 218.

[10] S. Murali *et al.*, "Sunmap: A tool for automatic topology selection and generation for nocs," *Proc. - Des. Autom. Conf.*, pp. 914–919, 2004.

[11] S. Saeidi *et al.*, "Smap: An intelligent mapping tool for network on chip," in *2007 Int. Symp. Signals, Circuits Syst.*, vol. 1. IEEE, Jul. 2007, pp. 1–4.

[12] L. Bononi *et al.*, "Noc topologies exploration based on mapping and simulation models," in *10th Euromicro Conf. Digit. Syst. Des. Archit. Methods Tools (DSD 2007)*. IEEE, Aug. 2007, pp. 543–546.

[13] E. Fusella *et al.*, "Phonocmap: An application mapping tool for photonic networks-on-chip," *Des. Autom. Test Eur. Conf. Exhib. (DATE), 2016*, pp. 289–292, 2016.

[14] E. Fusella *et al.*, "Reducing power consumption of lasers in photonic nocs through application-specific mapping," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 14, no. 2, pp. 1–11, Jul. 2018.

[15] S. Werner *et al.*, "A survey on optical network-on-chip architectures," *ACM Comput. Surv.*, vol. 50, no. 6, pp. 1–37, Dec. 2017.

[16] P. Grani *et al.*, "Design options for optical ring interconnect in future client devices," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 10, no. 4, pp. 1–25, Jun. 2014.

[17] Q. Xu *et al.*, "Micrometre-scale silicon electro-optic modulator," *Nature*, vol. 435, no. 7040, pp. 325–327, May 2005.

[18] C. Coello Coello *et al.*, "Mopso: a proposal for multiple objective particle swarm optimization," in *Proc. 2002 Congr. Evol. Comput. CEC'02 (Cat. No.02TH8600)*, vol. 2. IEEE, 2002, pp. 1051–1056.

[19] L. W. Nagel *et al.*, "Spice (simulation program with integrated circuit emphasis)," EECS Department, University of California, Berkeley, Tech. Rep. UCB/ERL M382, Apr 1973.

[20] C. Zhang *et al.*, "Low threshold and high speed short cavity distributed feedback hybrid silicon lasers," *Opt. Express*, vol. 22, no. 9, p. 10202, May 2014.

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

[22] M. Bahadori *et al.*, "Energy-performance optimized design of silicon photonic interconnection networks for high-performance computing," in *Des. Autom. Test Eur. Conf. Exhib. (DATE), 2017*. IEEE, Mar. 2017, pp. 326–331.

[23] Y. Wang *et al.*, "Bidirectional tuning of microring-based silicon photonic transceivers for optimal energy efficiency," *Asia South Pacific Des. Autom. Conf.*, Jan. 2019.

[24] A. V. Krishnamoorthy *et al.*, "Exploiting cmos manufacturing to reduce tuning requirements for resonant optical devices," *IEEE Photonics J.*, vol. 3, no. 3, pp. 567–579, 2011.

[25] Y. Zheng *et al.*, "Power-efficient calibration and reconfiguration for optical network-on-chip," *J. Opt. Commun. Netw.*, vol. 4, no. 12, p. 955, Dec. 2012.

[26] R. Wu *et al.*, "Pairing of microring-based silicon photonic transceivers for tuning power optimization," in *2018 23rd Asia South Pacific Des. Autom. Conf.* IEEE, Jan. 2018, pp. 135–140.

[27] Y. Wang *et al.*, "Energy-efficient channel alignment of dwdm silicon photonic transceivers," *Des. Autom. Test Eur. Conf. Exhib. (DATE), 2018*, Mar. 2018.

[28] R. K. V. Maeda *et al.*, "Jade: a heterogeneous multiprocessor system simulation platform using recorded and statistical application models," in *Proc. 1st Int. Work. Adv. Interconnect Solut. Technol. Emerg. Comput. Syst. - AISTECS '16*. ACM Press, 2016, pp. 1–6.

[29] Z. Wang *et al.*, "A case study on the communication and computation behaviors of real applications in noc-based mpsocs," in *2014 IEEE Comput. Soc. Annu. Symp. VLSI*. IEEE, Jul. 2014, pp. 480–485.

[30] C. Xu *et al.*, "Automated os-level device runtime power management," in *Proc. Twent. Int. Conf. Archit. Support Program. Lang. Oper. Syst. - ASPLOS '15*, vol. 50, no. 4. ACM Press, Mar. 2015, pp. 239–252.