Cross-Layer Thermal Reliability Management in Silicon Photonic Networks-on-Chip

Sudeep Pasricha, Sai Vineel Reddy Chittamuru, Ishan G. Thakkar
Department of Electrical and Computer Engineering
Colorado State University, Fort Collins, CO, U.S.A.
{sudeep, sai.chittamuru, ishan.thakkar}@colostate.edu

ABSTRACT

Silicon photonics technology is being considered for future networks-on-chip (NoCs) as it can enable high bandwidth density and lower latency with traversal of data at the speed of light. But the operation of photonic NoCs (PNoCs) is very sensitive to on-chip temperature variations. These variations can create significant reliability issues for PNoCs. This paper presents a run-time cross-layer framework to overcome temperature variation-induced reliability issues in PNoCs. The framework consists of a device-level reactive mechanism and a system-level proactive technique to avoid on-chip thermal threshold violations and mitigate thermal reliability issues. Our analysis indicates that this framework can reliably satisfy on-chip thermal thresholds and maintain high network bandwidth while reducing power dissipation over state-of-the-art solutions.

KEYWORDS

Photonic network on chip; thermal management; cross-layer design

ACM Reference Format:
Sudeep Pasricha, Sai Vineel Reddy Chittamuru, Ishan G. Thakkar. 2018. Cross-Layer Thermal Reliability Management in Silicon Photonic Networks-on-Chip. In GLSVLSI '18: 2018 Great Lakes Symposium on VLSI, May 23–25, 2018, Chicago, IL, USA. ACM, NY, NY, USA, 6 pages.
https://doi.org/10.1145/3194554.3194608

1. INTRODUCTION

Recent advances in the area of silicon photonics have enabled their integration with CMOS circuits [1]. On-chip photonic links provide several significant advantages over their traditional metallic counterparts, including near light speed transfers, high bandwidth density, and low power dissipation [2]. Photonic links also have several times lower data-dependent energy consumption for global on-chip transfers than electrical wires, enabling the design of high-radix networks that are easier to program [3], [4]. Silicon photonics is thus becoming an exciting new option for on-chip communication, and has catalyzed the design of high performance photonic NoCs (PNoCs) for multicore systems [3]-[6].

Typical PNoC architectures employ several photonic devices such as photonic waveguides, couplers, splitters, and multi-wavelength laser sources, along with microring resonators (MRs) as modulators, detectors, and switches. In a typical high bandwidth PNoC, an off-chip laser source generates multi-wavelength light, which is coupled by an optical coupler to an on-chip photonic waveguide. This waveguide guides the input optical power of multiple dense-wavelength-division-multiplexed (DWDM) wavelengths, via a series of optical power splitters, to the individual nodes (e.g., processing cores) on the chip. Each node in the PNoC can connect and communicate to multiple other nodes through such photonic waveguides that can guide the utilized DWDM wavelengths. A wavelength serves as a carrier to a data signal. Typically, multiple data signals are generated at a source node in the electrical domain as sequences of logical 1 and 0 voltage levels. These input electrical data signals are modulated onto the DWDM wavelengths using a bank of modulator MRs. These data-modulated carrier wavelengths traverse a waveguide to a destination node, where an array of detector MR filters remove them and drop them on adjacent photodetectors to regenerate electrical data signals. In general, each node in a PNoC should be able to send and receive data in the optical domain on all of the utilized carrier wavelengths. Therefore, each node has a bank of modulator MRs and a bank of detector MRs. Each MR in a bank is structurally itself a looped waveguide with a small and unique circumference, which makes the MR resonate with and operate on a specific carrier wavelength referred to as the MR's assigned carrier wavelength. Thus, the excellent wavelength selectivity of MRs and DWDM capability of waveguides enable high bandwidth data transfers in PNoCs, under ideal conditions.

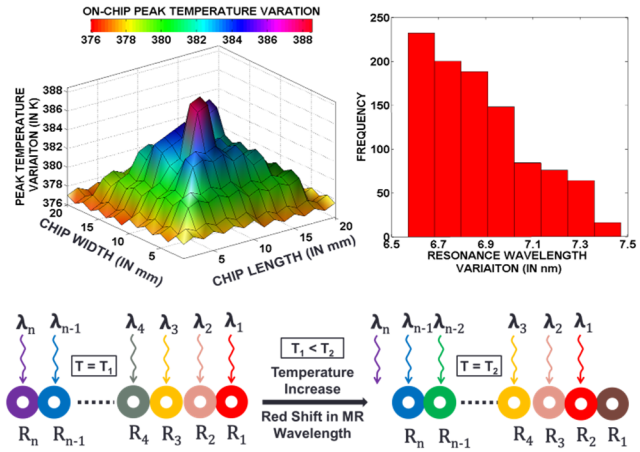

Fig. 1: Impact of temperature increase in DWDM based PNoCs

Unfortunately, MR devices are highly sensitive to temperature fluctuations. As an example, consider the top left of Fig. 1 which shows a plot of the maximum peak temperature of each part of the photonic die (stacked on top of the logic die) for various PARSEC and SPLASH-2 applications executed on a 64-core chip. The main reason for the higher temperature at the center of the chip is the inefficiency of the heat sink to remove heat from the center of the chip. With increase in temperature on the die, the refractive index of an MR device changes, causing a change in its resonance wavelength that has been statically assigned at design-time [7]. The

GLSVLSI'18, May 23-25, 2018, Chicago, IL, USA.

ACM ISBN 978-1-4503-5724-1/18/05...$15.00.
DOI: *https://doi.org/10.1145/3194554.3194608*

resonance wavelength shifts because of the peak temperature-rises can be as high as 7.4nm, as shown in the histogram at the top right of Fig. 1. This has important implications for communication reliability. The bottom of Fig. 1 depicts MRs R_1-R_n that have been manufactured to resonate on wavelengths λ_1-λ_n respectively at temperature T_1. As the temperature increases, due to the resulting variations in refractive index, each MR now resonates with a different wavelength towards the red end of the visible spectrum (i.e., *red-shift*). This red-shift is shown in the figure where, at temperature T_2, MR R_i will now be in resonance with λ_{i-1}. This phenomenon reduces transmission reliability and wastes bandwidth, e.g., MRs in Fig. 1 are unable to read or write to wavelength λ_n at temperature T_2.

In a typical multicore processor chip, it is quite common to observe on-chip maximum temperature as large as 90^{0}C [10], which can easily result in a mismatch of the resonant wavelengths of MRs and lead to inefficient data transmission. Recent work has proposed localized trimming [9] and thermal tuning [7] mechanisms to re-align the resonant wavelengths of MRs. Trimming alters the free-carrier concentration in an MR core, whereas thermal tuning uses integrated micro-heaters to alter local temperatures at MRs. However, these mechanisms come with considerable power and performance overhead. *Hence, it is essential to intelligently manage temperatures in PNoC-based multicore systems, to achieve reliable communication with minimal local trimming and tuning costs.*

In this work, we aim to minimize the need for (and overheads of) localized thermal tuning and trimming in PNoCs while coping with thermal variations, thereby easing the adoption of PNoCs in future multicore systems. We propose a cross-layer thermal reliability- management framework (SPECTRA) that integrates adaptive MR assignment at the device-level and dynamic thread migration at the system-level for emerging PNoC-based multicore systems. We compare our framework with other thermal management solutions and show significant reduction in maximum temperature and trimming/tuning power consumption compared to the state-of-the-art.

2. RELATED WORK

Traditional electrical NoC communication fabrics are projected to suffer from high power dissipation and severely reduced performance in future multicore systems [1]. The higher bandwidth density and lower power dissipation possible with silicon-photonic links, compared to electrical wires, has made them an attractive option for emerging multicore platforms. Recent research has thus focused on exploring a wide spectrum of network topologies and protocols to enable efficient PNoC architectures [3]-[6], [26]-[32].

One of the key challenges for the widespread adoption of PNoC architectures is thermal management of the silicon-photonic links. Several techniques exist to reduce thermal hotspots and gradients using DVFS [11], workload migration [17] and liquid cooling [12], but these techniques do not consider the unique challenges (e.g., MR resonance wavelength shifts) and constraints (e.g., wavelength match between sender and receiver MR pairs) that exist in PNoCs.

A few prior works have explored thermal management in PNoCs at either the device-level or the system level. The device-level efforts have mainly proposed various athermal photonic devices to reduce the localized tuning/trimming power in MRs. These design time solutions include using various materials such as cladding to reduce thermal sensitivity [13] and using heaters as well as temperature sensors for thermal control [14]. *These device-level techniques either possess high power overhead or require costly changes in the manufacturing process (e.g., larger device areas) that decrease network bandwidth density and area efficiency.*

At the system-level, the overhead associated with localized tuning of MRs is reduced in [7] using the group drift property of co-located MRs as part of a method to trim a group of rings at the same time. A reliability-aware design flow to address variation induced reliability issues is proposed in [15], which uses athermal coating at fabrication-level, voltage tuning at device-level, as well as channel hopping at the system architecture level. In [10] a ring aware thread scheduling policy is proposed to reduce on-chip thermal gradients in a PNoC. To enhance transmission reliability, two encoding techniques PCTM5B and PCTM6B are presented in [8] and a wavelength spacing (WSP) technique is presented in [16] which aim to improve SNR in DWDM-based crossbar PNoC architectures. *None of these system-level solutions for PNoCs consider the impact of run-time workload variations or the relationship between thermal hotspots and transmission reliability.*

To overcome the shortcomings of prior work at the device-level and system-level, we devise a cross-layer thermal management framework for PNoCs that exploits the synergy between new enhancements at both levels to outperform state-of-the-art solutions.

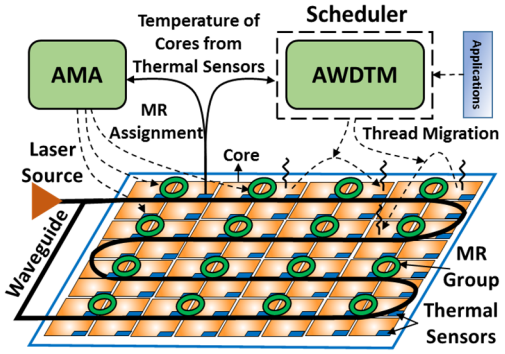

Fig. 2: Overview of SPECTRA, that integrates a device-level adaptive microring assignment mechanism (AMA) and a system-level anti wavelength-drift dynamic thermal management (AWDTM) technique.

3. SPECTRA FRAMEWORK OVERVIEW

Our cross-layer SPECTRA framework enables thermal reliability-aware run-time PNoC management by integrating device-level and system-level enhancements. Fig. 2 gives a high-level overview of our framework. The adaptive microring assignment (AMA) mechanism dynamically assigns a set of MRs for reliable modulation and reception of data from a photonic waveguide in a specific temperature range. This device-level technique aims to adapt to the changing on-chip thermal profile and maintain maximum bandwidth while minimizing trimming and tuning power in the PNoC. However, AMA cannot control maximum on-chip temperature, whose control is critical to further minimize MR trimming and tuning power. Thus, to control maximum on-chip temperature we devise an anti-wavelength-drift dynamic thermal management (AWDTM) scheme that uses support vector regression (SVR) based temperature prediction and dynamic thread migration, to avoid on-chip thermal threshold violations, minimize on-chip thermal hotspots, and reduce thermal tuning power for MRs. The next two sections present details of the two schemes.

4. DEVICE-LEVEL OPTIMIZATION

As discussed earlier, with increase in chip temperature, MR resonance wavelengths change and result in MRs being unable to read or write data to their design-time resonance wavelengths in the waveguide. Fortunately there is a linear dependency between temperature increase and resonance wavelength shift [15], which

we exploit to propose a device-level adaptive microring assignment (AMA) mechanism that dynamically assigns a set of MRs to modulate and detect data in a particular temperature range.

Fig. 3 shows how at temperatures T_i and T_{i+1} ($T_{i+1} > T_i$), an MR resonance is in exact alignment with the available wavelengths λ_k and λ_{k+1}, respectively. These temperatures are called ideal resonant temperatures (IRTs). When the MR temperature is in between IRTs T_i and T_{i+1}, as shown in Fig.3, the MR needs to be either *trimmed* to resonate to λ_k (which is the resonance wavelength of an MR at temperature T_i) or thermally *tuned* to resonate to λ_{k+1} (which is the resonance wavelength of an MR at temperature T_{i+1}). To adaptively choose the least power consuming method from trimming and thermal tuning, we divide the temperature range between IRTs T_i and T_{i+1} into two parts: trimming temperature range (Δ_{tr}) and tuning temperature range (Δ_{tu}). For an MR at temperature T, if $(T_i + \Delta_{tr}) > T > T_i$ we perform trimming as it takes the least power, else if $(T_i + \Delta_{tr}) < T < T_{i+1}$ we perform tuning as it takes the least power. At the boundary of the trimming and tuning temperature ranges, where $T_{i+1}-\Delta_{tu} = T_i+\Delta_{tr}$, both trimming and tuning consume equal power, and hence, an MR can be either trimmed or tuned. This temperature is called the boundary temperature (BT_i). It has been shown that for a small resonance wavelength shift (<1nm), thermal tuning power is higher compared to trimming power to mitigate the same amount of TV-induced shift [7]. Thus, our *AMA* approach considers a higher trimming temperature range compared to tuning temperature range ($\Delta_{tr}>\Delta_{tu}$), to minimize total trimming and tuning power.

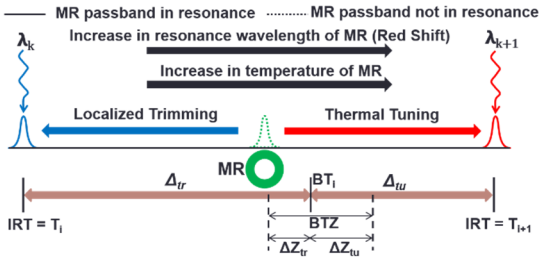

Fig. 3: Red shift of MR with increase in temp. from IRTs T_i to T_{i+1} with trimming and tuning range of temperatures between these IRTs.

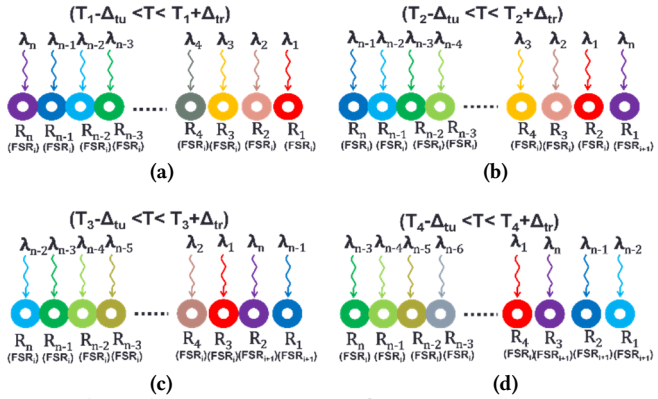

Fig. 4: Thermal aware assignment of microrings (R_{1-n}) to wavelengths (λ_{1-n}) at four successive IRTs T_1, T_2, T_3, and T_4 in *AMA* mechanism.

In *AMA*, MRs are dynamically shifted (trimmed or tuned) to an appropriate IRT for correct operation based on their current temperature. Fig. 4(a)-4(d) show four different MR wavelength assignment configurations at successive IRTs T_1, T_2, T_3, and T_4, where $T_4>T_3>T_2>T_1$. If the MR group temperature T is such that $(T_1 - \Delta_{tu}) < T < (T_1+\Delta_{tr})$ then the assignment in Fig. 4(a) is chosen, otherwise if $(T_2 - \Delta_{tu}) < T < (T_2+\Delta_{tr})$, $(T_3 - \Delta_{tu}) < T < (T_3+\Delta_{tr})$, *or* $(T_4 - \Delta_{tu}) < T < (T_4+\Delta_{tr})$ then the assignment in Fig. 4(b), 4(c), or 4(d) is chosen, respectively. One critical observation in the assignment shown in Fig. 4(a) is that MRs R_1-R_n are in resonance with λ_1- λ_n within the same Free Spectral Range (FSR_i), whereas, in Fig. 4(b) at IRT T_2, MRs R_2-R_n are in resonance with λ_1-λ_{n-1}, respectively in FSR_i and MR R_1 is in resonance with λ_n of the next FSR (i.e., FSR_{i+1}). In this assignment and the ones shown in Fig. 4(c) and 4(d), there is a need to reposition bits in the electrical domain using backend barrel-shifters or pipelined shift registers. The assignments shown in Fig. 4(b), 4(c), and 4(d) require one, two, and three bit shifts, respectively, to retrieve the original data.

AMA represents a powerful reactive technique to adapt to on-die thermal variations with low overhead while ensuring reliable and high-bandwidth communication in MR based PNoCs. But there is scope for two further enhancements. First, there is a need to proactively control the peak on-chip temperature to reduce the range of on-chip temperature swings, which ultimately limits the number of required bit shifts (this work caps the number bit shifts to three as shown Fig. 4(d)) and reduces the latency to retrieve the original data. Second, at the BT temperature (Fig. 3), maximum trimming or tuning power is required to realign the MR resonances to their nearest carrier wavelengths. Thus, avoiding BT temperatures at MRs can reduce trimming and tuning power overhead. As shown in Fig. 3, we define a boundary temperature zone (BTZ) around each BT_i. This zone includes temperatures T such that $BT_i-\Delta Z_{tr} < T < BT_i+\Delta Z_{tu}$ where ΔZ_{tr} and ΔZ_{tu} are designer specified parameters. Cores with corresponding MR bank temperatures that are within BTZs are called boundary temperature cores (BTCs). As BTCs possess the highest trimming and tuning power overhead for their corresponding MR bank, a mechanism that reduces the number of BTCs can save trimming and tuning power. Section 5 describes such a mechanism, which also controls the range of on-chip temperature swings within allowable limits.

5. SYSTEM-LEVEL OPTIMIZATION

To proactively reduce thermal hotspots (which will reduce instances of 'irrecoverable drift') and control on-die temperature (to reduce the number of BTCs), we propose a system-level anti wavelength-drift dynamic thermal management (AWDTM) technique. The primary goals with AWDTM is to maintain the temperature of all the cores on a die below a specified thermal threshold, i.e., for all cores $1 \le i \le N$, $T_i < T_t$ where T_i is the temperature of core i and T_t is threshold temperature. We utilize support vector based regression (SVR) to predict the future temperature of a core. The predicted temperature is compared with a thermal threshold to determine the potential for a thermal emergency. If such a potential exists, threads are migrated to available BTCs. This step has a twofold benefit. First by moving the thread away from a core that could suffer a thermal emergency, we avoid instances of irrecoverable drift in the MR groups of that core. Second, by moving the thread to a BTC, the temperature of the BTC will increase resulting in that core no longer being a BTC (consequently the temperature of the core's MR groups will also increase, taking them outside of their BTZ and closer to IRTs, which will reduce trimming/tuning power).

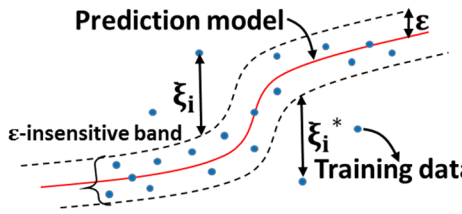

Fig. 5: Non-linear support vector based regression prediction model

5.1 Thermal Prediction

A typical SVR relies on a prediction model that ignores errors that are situated within ε distance of the true value. This type of a prediction model is called an ε-insensitive prediction model. Fig. 5 shows an example of a one-dimensional non-linear SVR based prediction model with an ε-insensitive band. The variables *(ξ and ε)* measure the cost of the errors on the training points. These are zero for all points that are inside the ε-insensitive band. SVR is primarily designed to perform linear regression. To handle non-linearity in data, SVR first maps the input x_i onto an m-dimensional space using some fixed (non-linear) mapping notated as Φ, and then a linear model is constructed in this high-dimensional space. Thus, it overcomes drawbacks of linear and logistic regression towards handling non-linearity in data. This class of SVRs is called *kernel based SVRs.* As on-chip temperature variation data is non-linear in the original space, our SVR model employs a kernel based regression which uses a Radial Basis Function (RBF) Gaussian kernel. The RBF kernel improves the accuracy of SVR when data has non-linearity in the original space.

Our SVR predictor accepts input parameters reflecting the workload for the core under consideration, in terms of instructions per cycle (IPC_i), as well as the current core temperature (CT_i) and temperatures of surrounding cores (TN_i). Once these parameter values are obtained at run-time, the future temperature for the core can be predicted. We trained our SVR model using a set of multi-threaded applications from the SPLASH-2 [22] and PARSEC [18] benchmark suites, specifically: *blackscholes (BS), bodytrack (BT), vips (VI), facesim (FS), fluidanimate (FA), swaptions (SW), barnes (BA), fft (FFT), radix (RX), radiosity (RD)* and *raytrace (RT)* with different thread counts: 2, 4 and 8. We considered different combinations of thread mappings on a 9-core (3×3) floorplan, to train our predictor to determine the temperature of the center (target) core. As the future temperature of a target core is dependent on the average temperature of its immediate neighboring cores, we trained our SVR model with temperature inputs from the target core running a single thread, as well as its surrounding cores running a variable number of threads. Simulations for each of these floorplans allowed us to obtain data to train our SVR model. This data included temperature for the target core and its neighboring core temperatures, as well as instructions per cycle *(IPC)* for the target core. IPC is very useful to determine if there is a phase change in an application and plays a crucial role in maintaining future temperature prediction accuracy, especially when temperatures of a target core and its neighbors are similar at a given time. Our training algorithm involved an iterative process that adjusts the weights and bias values in the SVR predictor to fit the training set.

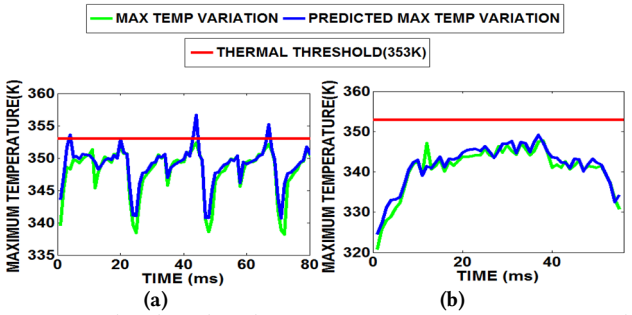

Fig. 6: Actual and predicted maximum temperature variation with execution time for (a) fluidanimate *(FA)* (b) radiosity *(RD)* benchmark applications executed on 64-core platform executing 32 threads.

We verified the accuracy of our SVR model for multi-threaded benchmark workloads (we considered 6000 floorplans, with 70% of input data for training and 30% for testing) and found that it has an accuracy of over 95%. Fig. 6(a) and 6(b) show actual and predicted on-chip temperature variations for a 64-core platform executing 32 threads of the FA and RD benchmarks, respectively. From these figures it can be seen that our temperature predictor tracks temperature quite accurately. When predicted temperature exceeds the thermal threshold our thread migration mechanism (which is discussed next) migrates threads from hotter cores to cooler cores to keep overall maximum temperature below the threshold.

5.2 Thermal-Aware Thread Migration

Algorithm 1 shows the pseudo-code for the AWDTM thread migration procedure. For each core, we periodically monitor the IPC value from performance counters and temperature from on-chip thermal sensors. If a thermal emergency is predicted for a core by the SVR predictor, then AWDTM initiates a thread migration procedure, otherwise no action is taken. Firstly, future temperature (PT_i) of the i^{th} core is predicted using the SVR based predictor with inputs: core temperature (T_i), core IPC (IPC_i), and temperature of neighboring cores (TN_i) in steps **1-3**. The list of available BTCs (i.e., those that are not currently executing any thread) and available NBTCs is obtained in steps **4-10**. In steps **11-12**, a loop iterates over all cores and checks for possible thread migration conditions (i.e., thermal emergency cases where current core predicted temperature (PT_i) is greater than thermal threshold (T_t)). If a thread migration is required, then in steps **13-21**, we check for free BTCs, and if they are available then we migrate the thread from current core to the BTC with lowest temperature, else we migrate the thread to a free NBTC with lowest temperature. This AWDTM thread migration is invoked at every epoch (1ms).

Algorithm 1: AWDTM thread migration algorithm

Inputs: Current core temperature (T_i), average neighboring core temperature (TN_i), current core IPC (IPC_i)

```
1:  for each core i do  // Loop that predicts future temperature
2:      PTi = SVR_predict_future_temperature (Ti, TNi, IPCi)
3:  end for
4:  for each core i do // Loop that checks for free BTCs and NBTCs
5:      if Ti in BTZ and IPCi == 0 then
6:          List_BTC = Push i  //add core to BTC list
7:      else if IPCi == 0 then
8:          List_NBTC = Push i //add core to NBTC list
9:      end if
10: end for
11: for each core i do   // Loop that performs thread migration
12:     if PTi ≥ Tt then
13:         if List_BTC ≠ {} then
14:             Migrated_core = Find_min_temperature_core(List_BTC)
15:             Do_thread_migration( core_i → Migrated_core)
16:             List_BTC = Pop i
17:         else if List_NBTC ≠ {} then
18:             Migrated_core = Find_min_temperature_core(List_NBTC)
19:             Do_thread_migration( core_i → Migrated_core)
20:             List_NBTC = Pop i
21:         end if
22:     end if
23: end for
```

Output: Thread migration to BTC or NBTC cores

6. EXPERIMENTS

6.1 Experimental Setup

We target a 64-core multicore system for evaluation of our SPECTRA (AMA+AWDTM) framework. Each core has a Nehalem x86 [19] micro-architecture with 32KB L1 instruction and data caches and a 256 KB L2 cache, at 32nm and running at 5GHz. We evaluate our framework on two well-known PNoC architectures: Corona [3] and Flexishare [4]. Corona uses a 64×64 multiple write

single read (MWSR) crossbar with token slot arbitration. Flexishare uses 32 multiple write multiple read (MWMR) waveguide groups with a 2-pass token stream arbitration. Each MWSR waveguide in Corona and each MWMR waveguide in Flexishare is capable of transferring 512 bits of data from a source to a destination node.

We modeled and simulated these architectures with the SPECTRA framework for multi-threaded applications from the PARSEC [22] and SPLASH-2 [18] benchmark suites (Section 6.2). Simulations were performed with a "warm-up" period of 100-million instructions and execution period of one billion cycles. Power and instruction traces for the benchmark applications were generated using the Sniper 6.0 [19] simulator and McPAT [20]. We used the 3D-ICE tool [21] for thermal analysis. The ambient temperature was set to 303K and the thermal threshold (T_t) was set to 353K. We considered a three layered 3D-stacked multicore system as advocated in existing PNoC architectures [3]-[4] with a die area footprint of 400mm^2, where the top layer is the core-cache layer, the middle layer is the analog to digital (A/D) and digital to analog (D/A) conversion layer, and the bottom layer is the photonic layer with MRs, waveguides, ring heaters, and ring trimmers for carrier injection. Some of the materials used in the construction of the 3D-stack in the 3D-ICE tool and their properties are shown in Table 1.

The MR thermal sensitivity was assumed to be 0.11nm/°C [15]. For PNoCs, we considered 64 DWDM waveguides sharing the working band 1530–1625 nm with a wavelength channel width of 1.48 nm. The MR trimming power is set to 130µW/nm [9] for current injection (blue shift) and tuning power is set to 240µW/nm [7] for heating (red shift). To compute laser power, we considered detector responsivity as 0.8 A/W [25], MR through loss as 0.02 dB, waveguide propagation loss as 1 dB/cm, waveguide bending loss as 0.005 dB/90^0, and waveguide coupler/splitter loss as 0.5 dB [25]. We calculated photonic loss in components using these values, which sets the photonic laser power budget and the electrical laser power. For energy consumption of photonic devices, we adapt parameters from [25], with 0.42pJ/bit for every modulation and detection event, and 0.18pJ/bit for the driver circuits of MRs.

As presented in section 4, to minimize trimming and tuning power consumption, trimming temperature range (Δ_{tr}) and tuning temperature range (Δ_{tu}) for AMA are calculated as 8.73K and 4.72K respectively. We also set ΔZ_{tr} and ΔZ_{tu} as 2K and 3.5K respectively. Based on our sensitive analysis we get the best accuracy for our SVR-based temperature predictor when parameters C and γ are set to 1000 and 0.1 respectively.

Table 1: Properties of materials used by 3D-ICE tool [21], [23]

Material	Thermal Conductivity	Volumetric Heat Capacity
Silicon	1.30e-4 W/µm K	1.628e-12 J/µm^3 K
Silicon dioxide	1.46e-6 W/µm K	1.628e-12 J/µm^3 K
BEOL	2.25e-6 W/µm K	2.175e-12 J/µm^3 K
Copper	5.85e-4 W/µm K	3.45e-12 J/µm^3 K

6.2 Experiment Results

We compared the performance of our SPECTRA framework with two prior works on multicore thermal management: a ring aware policy (RATM) [10] and a predictive dynamic thermal management (PDTM) framework [17]. RATM distributes threads uniformly across cores that are closer to PNoC nodes first and then distributes the remaining threads in a regular pattern from outer cores to inner cores. PDTM uses a recursive least square based temperature predictor to determine if the predicted temperature of a core exceeds a thermal threshold, and if so then thread migration is done from that core to the coolest core not executing any threads.

Fig. 7 (a)-(b) shows the maximum temperature obtained with the three frameworks across eleven applications from the PARSEC and SPLASH-2 benchmarks suites with 48 and 32 thread counts executed on a 64-core system with the Corona PNoC [3] architecture. From Fig. 7(a) it can be observed that some applications (e.g., FA, SW) with 48 threads exceed the threshold (353K) as there are insufficient number of free cores on the chip whose temperature is below the thermal threshold to migrate threads. However in Fig. 7(b), our SPECTRA framework avoids violating thermal thresholds for all the benchmark applications with 32 threads. On average, SPECTRA has 13.2K and 14.5K lower maximum temperature compared to the RATM policy for 48 and 32 threads, respectively. SPECTRA migrates threads from hotter cores to cooler cores to control maximum temperature, whereas RATM does a simple thread allocation that is unable to appropriately control maximum temperature. For most of the cases, maximum temperatures with PDTM and SPECTRA are below the thermal threshold. As SPECTRA prefers to migrate threads to BTCs instead of to the coolest cores as done in PDTM, maximum temperatures with SPECTRA are sometimes higher compared to maximum temperatures with PDTM, as BTCs have higher initial temperature compared to the coolest cores on the chip. However, despite the higher maximum temperatures, SPECTRA still saves more power than PDTM (and also RATM) as discussed next.

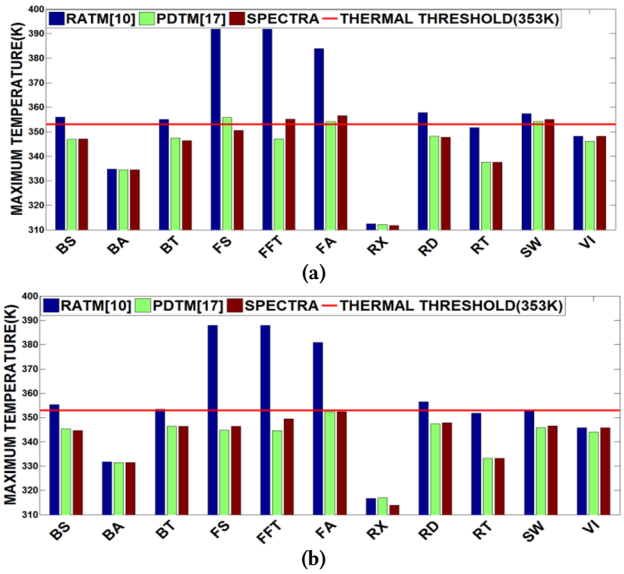

Fig. 7: Maximum temp. comparison of SPECTRA with RATM [10] and PDTM [17] for (a) 48 (b) 32 threaded PARSEC and SPLASH-2 benchmarks executed on 64-core system with Corona PNoC.

Fig. 8 shows the power dissipation comparison for the three frameworks across multiple 48-threaded applications for the Corona [3] and Flexishare [4] PNoC architectures, respectively. One of the main reasons why SPECTRA has lower power dissipation than RATM and PDTM is that it more aggressively reduces trimming and tuning power in both Corona and Flexishare PNoCs. As can be seen in Fig. 8(a), SPECTRA has 73.3% and 72.5% lower trimming and tuning power on an average compared to RATM and PDTM for the Corona PNoC architecture, respectively. Furthermore, as seen in Fig. 8(b), SPECTRA has 72.3% and 71.5% lower trimming and tuning power on average compared to RATM and PDTM for the Flexishare PNoC architecture, respectively. The AMA technique in SPECTRA intelligently conserves trimming and tuning power compared to RATM and PDTM by performing MR reassignment with increase in temperature, while our AWDTM further improves trimming and tuning power savings with its intelligent thread migration to BTCs. In addition, SPECTRA saves considerable trimming/tuning power to ultimately achieve overall power reduction. From the power analysis in Fig.

8(a), it can be observed that SPECTRA with Corona has 45.4% and 44.5% lower total power consumption compared to Corona with RATM and PDTM respectively. Further from Fig. 8(b), it can be seen that Flexishare with SPECTRA has 63.4% and 62.6% lower power dissipation compared to RATM and PDTM.

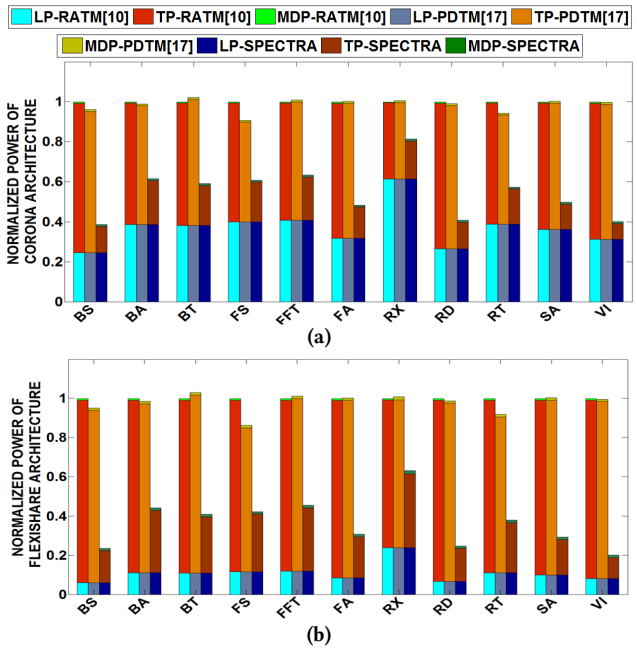

Fig. 8: Normalized power (Laser Power (LP), Trimming and tuning power (TP) and modulating and detecting Power (MDP)) comparison of SPECTRA with RATM [10] and PDTM [17] for 48 threaded applications of PARSEC and SPLASH-2 suites executed on (a) Corona (b) Flexishare PNoC architectures for a 64-core multicore system. Results shown are normalized w.r.t RATM.

In summary, from the above results, it is apparent that our proposed cross-layer SPECTRA framework outperforms previously proposed approaches for thermal management in multicore systems with PNoCs by combining a novel reactive device-level technique (AMA) that improves waveguide channel utilization with a novel system-level proactive thread migration technique (AWDTM). The excellent power savings compared to previous approaches strongly motivate the use of cross-layer techniques in general, and more specifically our proposed cross-layer thermal management framework in future PNoC based multicore architectures.

7. CONCLUSION

We have presented the cross-layer SPECTRA framework that combines two dynamic thermal management mechanisms for the reduction of maximum on-chip temperature and to conserve trimming and tuning power of MRs in DWDM-based PNoC architectures. These techniques (AMA at the device-level, AWDTM at the system-level) constitute a hybrid, cross-layer, reactive-proactive management framework that demonstrated interesting trade-offs between performance and power across two different state-of-the-art crossbar-based PNoC architectures. Our experimental analysis on the well-known Corona and Flexishare PNoC architectures has shown that SPECTRA can notably conserve total power by up to 63.4% and trimming and tuning power by up to 73.3%.

REFERENCES

[1] Y. Xu, S. Pasricha, "Silicon Nanophotonics for Future Multicore Architectures: Opportunities and Challenges", IEEE D&T, 2014.
[2] J. D. Owens, et al., "Research challenges for on-chip interconnection networks," in IEEE Micro, Sep-Oct 2007.
[3] D. Vantrease et al., "Light speed arbitration and flow control for nanophotonic interconnects," in *IEEE/ACM MICRO*, Dec. 2009.
[4] Y. Pan, J. Kim, G. Memik, "Flexishare: Channel sharing for an energy efficient nanophotonic crossbar," in Proc. HPCA, 2010.
[5] S. V. R. Chittamuru, S. Desai, S. Pasricha "A reconfigurable silicon-photonic network with improved channel sharing for multicore architectures," in ACM GLSVLSI, May 2015.
[6] S. Pasricha, S. Bahirat, "OPAL: A Multi-Layer Hybrid Photonic NoC for 3D ICs," in ASPDAC, Yokohama, Japan, Jan 2011.
[7] C. Nitta, M. Farrens, and V. Akella, "Addressing system-level trimming issues in on-chip nanophotonic networks," in Proc. HPCA, 2011.
[8] S. V. R. Chittamuru, S. Pasricha "Crosstalk Mitigation for High-Radix and Low-Diameter Photonic NoC Architectures", IEEE D&T, 2015.
[9] J. Ahn et al., "Devices and architectures for photonic chip-scale integration", Applied Physics A: MSP, 95:989–997, June 2009.
[10] T. Zhang et al., "Thermal management of manycore systems with silicon-photonic networks," in Proc. DATE, 2014.
[11] A. K. Coskun, T. S. Rosing, K. A. Whisnant, and K. C. Gross, "Static and dynamic temperature-aware scheduling for multiprocessor SoCs," Proc. IEEE Trans. on VLSI, vol. 16, no. 9, pp. 1127–1140, 2008.
[12] A. K. Coskun, J. L. Ayala, D. Atienza, and T. S. Rosing, "Modeling and dynamic management of 3D multicore systems with liquid cooling," in IFIP/IEEE Int'l Conference on VLSI-SoC, 2009, pp. 35–40.
[13] Z. Li et al., "Reliability modeling and management of nanophotonic on-chip networks," IEEE TVLSI, vol. 20, no. 1, pp. 98–111, 2012.
[14] C. T. DeRose et al., "Silicon microring modulator with integrated heater and temperature sensor for thermal control," in CLEO, 2010.
[15] M. Mohamed et al., "Reliability-Aware Design Flow for Silicon Photonics On-Chip Interconnect," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 22(8), pp. 1763–1776, 2014.
[16] S. V. R. Chittamuru, S. Pasricha, "Improving Crosstalk Resilience with Wavelength Spacing in Photonic Crossbar-based Network-on-Chip Architectures", in IEEE MWSCAS, 2015.
[17] I. Yeo, C. Liu, and E. Kim, "Predictive dynamic thermal management for multicore systems," in Proc. DAC, 2008, pp. 734–739.
[18] S. C. Woo et al., "The SPLASH-2 programs: characterization and methodological considerations," in ISCA, 1995, pp. 24–36.
[19] T. E. Carlson et al., "Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulations," in SC, 2011.
[20] S. Li et al., "McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures," MICRO, 2009.
[21] A. Sridhar et al., "3D-ICE: Fast compact transient thermal modeling for 3d ICs with inter-tier liquid cooling," in Proc. ICCAD, Nov. 2010.
[22] C. Bienia et al., "The PARSEC benchmark suite: Characterization and Architectural Implications," in Proc. PACT, 2008, pp. 72–81.
[23] http://www.mt-berlin.com/frames_cryst/descriptions/quartz.htm
[24] D. Harris et al., "Support Vector Regression Machines", in Advances in Neural Information Processing Systems NIPS, 1996, MIT Press.
[25] P. Grani, S. Bartolini, "Design Options for Optical Ring Interconnect in Future Client Devices," JETC, 2014.
[26] S. Bahirat, S. Pasricha, "METEOR: Hybrid Photonic Ring-Mesh Network-on-Chip for Multicore Architectures", ACM Trans. on Embedded Computing Systems (TECS), 13(3):116:1-116:33, Mar 2014.
[27] S. Bahirat, S. Pasricha, "A Particle Swarm Optimization Approach for Synthesizing Application-specific Hybrid Photonic Networks-on-Chip", IEEE ISQED, Mar. 2012.
[28] S. Bahirat, S. Pasricha, "UC-PHOTON: A Novel Hybrid Photonic Network-on-Chip for Multiple Use-Case Applications", ISQED 2010.
[29] S. V. R. Chittamuru, S. Desai, S. Pasricha, "SWIFTNoC: A Reconfigurable Silicon-Photonic Network with Multicast Enabled Channel Sharing for Multicore Architectures", ACM Journal on Emerging Technologies in Computing Systems (JETC), Jun 2017.
[30] I. Thakkar, S. V. R. Chittamuru, S. Pasricha, "Improving the Reliability and Energy-Efficiency of High-Bandwidth Photonic NoC Architectures with Multilevel Signaling," IEEE/ACM NOCS, Oct 2017.
[31] I. Thakkar, S. V. R. Chittamuru, S. Pasricha, "A Comparative Analysis of Front-End and Back-End Compatible Silicon Photonic On-Chip Interconnects," ACM/IEEE SLIP, Jun 2016.
[32] I. Thakkar, S. V. R. Chittamuru, S. Pasricha, "Run-Time Laser Power Management in Photonic NoCs with On-Chip Semiconductor Optical Amplifiers," IEEE/ACM International Symposium on Networks-on-Chip (NOCS), Aug 2016.