2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems

SPECTRA: A Framework for Thermal Reliability Management in Silicon-Photonic Networks-on-Chip

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

***Abstract*—*Silicon nanophotonics 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 temperature variations that frequently occur on a chip. These variations can create significant reliability issues for PNoCs. For example, a microring resonator (MR) may resonate at another wavelength instead of its designated wavelength due to thermal variations, which can lead to bandwidth wastage and data corruption in PNoCs. This paper proposes a novel run-time framework called SPECTRA to overcome temperature-induced reliability issues in PNoCs. The framework consists of (i) a device-level reactive MR assignment mechanism that dynamically assigns a group of MRs to reliably modulate/receive data in a waveguide based on the chip thermal profile; and (ii) a system-level proactive thread migration technique to avoid on-chip thermal threshold violations and reduce MR tuning/trimming power by dynamically migrating threads between cores. Experimental results indicate that SPECTRA can satisfy on-chip thermal thresholds and maintain high NoC bandwidth while reducing total power by up to 61%, and thermal tuning/trimming power by up to 71% over state-of-the-art thermal management solutions.*

I. Introduction

Recent developments in the area of silicon nanophotonics have enabled their integration with CMOS circuits [1]. On-chip photonic links provide several prolific advantages over their traditional metallic counterparts, including near light speed transfers, high bandwidth density, and low power dissipation [2]. Moreover, photonic links 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 nanophotonics is thus becoming an exciting new option for on-chip communication, and has catalyzed much research in the area of high performance photonic NoCs (PNoCs) for emerging multicore systems [3], [4], [5], [6].

Typical PNoC architectures utilize several photonic devices such as multi-wavelength lasers, microring resonators (MRs), waveguides, and splitters. A laser source generates light of multiple wavelengths that simultaneously traverse a photonic waveguide to support parallel data transfer, e.g., 64 wavelengths can transfer 64 bits in parallel, which is an example of dense wavelength division multiplexing (DWDM). MRs are used to couple electrical signals with these wavelengths at a source node for data transmission in the waveguide (data-modulation phase). MRs are also used to detect light-modulated data from the waveguide at the destination node (data-detection phase) and subsequently generate proportional electrical signals. At any point in time in a photonic-waveguide, MRs can be either in-resonance or out-of-resonance with respect to the incident wavelengths. In the resonance-mode, an MR couples light of a wavelength from the waveguide when its circumference is an integer multiple of that wavelength. Different-sized MRs are thus essential to simultaneously modulate data on the available wavelengths in a DWDM-waveguide.

Unfortunately, MR devices are highly sensitive to temperature fluctuations. With increase in temperature, the refractive index of an MR device changes, causing a change in its resonance wavelength that has been statically assigned at design-time [7]. As a result of this variation in resonance wavelength, an MR may be unable to write or read data in the waveguide at higher temperatures. For example, consider figure 1 that 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 also leads to wastage of available bandwidth, e.g., MRs in figure 1 are unable to read or write to wavelength λ_n at temperature T_2.

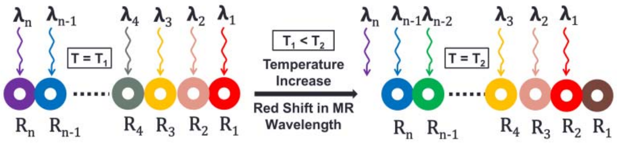


Figure 1: Impact of temperature increase in DWDM based PNoCs

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 induces a blue shift in the resonance wavelengths of MRs using carrier injection into MRs, whereas thermal tuning induces a red shift in the resonance wavelengths of MRs through heating of MRs using ring heaters. 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.*

Our goal in this work is to minimize the need for localized thermal tuning and trimming in PNoCs, thereby reducing key overheads and ultimately easing the adoption of PNoCs for future multicore systems. We propose a novel thermal reliability-aware run-time management framework called SPECTRA that integrates adaptive MR assignment at the device-level and dynamic thread migration at the system-level for emerging PNoC-based multicore systems. Our novel contributions as part of the SPECTRA framework are summarized below:

- We design a novel adaptive MR assignment (AMA) mechanism at the device-level that dynamically assigns a set of MRs at run-time to enable reliable modulation and reception of data from a photonic waveguide within a specific temperature range while maintaining maximum bandwidth;

978-1-4673-8700-2/16 $31.00 © 2016 IEEE
DOI 10.1109/VLSID.2016.12

- We propose a novel anti wavelength-drift dynamic thermal management (AWDTM) mechanism at the system-level that uses support vector regression (SVR) based temperature prediction and dynamic thread migration, to avoid on-chip thermal threshold violations and also reduce trimming/tuning power for MRs;
- We evaluate our proposed framework on a 64-core platform, comparing it with two state-of-the-art thermal management solutions: a microring aware thermal management (RATM) framework [10] and a predictive dynamic thermal management (PDTM) framework [17], and show significant reduction in maximum temperature and trimming/tuning power consumption compared to these existing solutions.

II. Related Work

Traditional electrical NoC communication fabrics are projected to suffer from cripplingly 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], [4], [5], [6].

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]. *While these device-level techniques are promising, they either possess a high power overhead or require costly changes in the manufacturing process (e.g., much larger device areas) that would 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.*

In view of the shortcomings of prior work, in this paper we aim to devise a novel thermal management framework for PNoCs that overcomes limitations at both the device-level and system-level. Sections III-V describe this proposed framework which is then evaluated in Section VI against prior work.

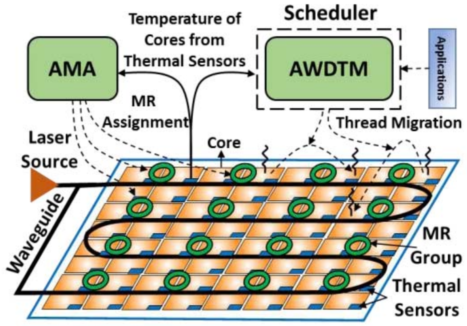


Figure 2: Overview of SPECTRA framework that integrates a device-level adaptive microring assignment mechanism (AMA) and a system-level anti wavelength-drift dynamic thermal management (AWDTM) technique

III. SPECTRA Framework Overview

Our SPECTRA framework enables thermal reliability-aware run-time PNoC management by integrating device-level and system-level enhancements. Figure 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.

IV. Adaptive Microring Assignment (AMA)

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.

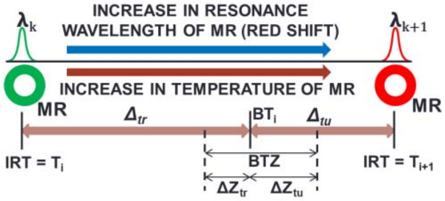
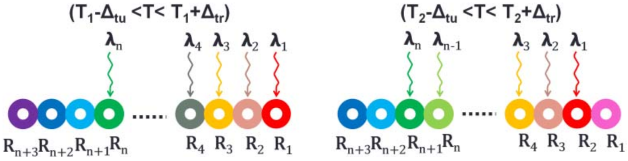


Figure 3: Red shift of MR with increase in temperature from IRTs T_i to T_{i+1} with trimming and tuning range of temperatures between these IRTs.

Figure 3 shows how at temperature T_i and T_{i+1}, an MR is in exact resonance with available wavelengths λ_k and λ_{k+1} respectively, in the waveguide. These temperatures are called ideal resonant temperatures (IRTs). When the MR temperature is in between IRTs T_i and T_{i+1} then the MR needs to be either *trimmed* to resonate at λ_k or thermally *tuned* to resonate at λ_{k+1}. To enable energy efficient trimming and thermal tuning of MRs, in AMA 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, else if $(T_i + \Delta_{tr}) < T < T_{i+1}$ we perform tuning (Figure 4). It has been shown that for a small resonance wavelength drift (<1nm), tuning power to mitigate temperature induced drift is higher compared to trimming power to mitigate the same amount of drift [7]. Thus AMA considers a higher trimming temperature range compared to tuning temperature range $(\Delta_{tr}>\Delta_{tu})$, to conserve total trimming and tuning power. At the boundary of trimming and tuning temperature range where $T_{i+1}-\Delta_{tu} = T_i+\Delta_{tr}$, an MR can be either trimmed or tuned, and this temperature is called the boundary temperature (BT_i).


$(T_1-\Delta_{tu} <T< T_1+\Delta_{tr})$
λ_n
λ_4 λ_3 λ_2 λ_1
$R_{n+3}R_{n+2}R_{n+1}R_n$
R_4 R_3 R_2 R_1
$(T_2-\Delta_{tu} <T< T_2+\Delta_{tr})$
λ_n λ_{n-1}
λ_3 λ_2 λ_1
$R_{n+3}R_{n+2}R_{n+1}R_n$
R_4 R_3 R_2 R_1

(a) (b)

Figure 4: Assignment of microring ($R_{1-(n+3)}$) to wavelength (λ_{1-n}) at two successive IRTs T_1 and T_2, in adaptive microring assignment (AMA)

In AMA, MRs are dynamically shifted (trimmed or tuned) to an appropriate IRT for correct operation based on their current temperature. Figures 4(a)-(b) show two different MR wavelength assignment configurations, at successive IRTs T_1 and T_2 where $T_2 > T_1$. If the MR group temperature T is such that $(T_1 - \Delta_{tu}) < T < (T_1+ \Delta_{tr})$ then the assignment in Figure 4(a) is chosen, otherwise if $(T_2 - \Delta_{tu}) < T < (T_2+ \Delta_{tr})$ then the assignment in Figure 4(b) is chosen. AMA avoids bandwidth wastage at higher temperatures by using additional MRs (R_{n+1}, R_{n+2}, and R_{n+3} in figure 4). These additional MRs are designed to be in resonance with available wavelengths in the waveguide at higher temperatures. For example, figure 4(b) shows how an additional MR R_{n+1} is in resonance with wavelengths λ_n, allowing full utilization of available bandwidth even at the higher temperature T_2. Similarly, if temperature increases beyond $(T_2+ \Delta_{tr})$ then additional MRs R_{n+2} and R_{n+3} can be used to avoid bandwidth wastage. The additional MRs used in AMA have a minimal photonic area overhead of 4.5% for a high-bandwidth 64-DWDM waveguide. The additional MRs do increase waveguide through losses which in turn increases PNoC laser power. This overhead is quantified in Section VI.

AMA represents a powerful reactive technique to adapt modulating and detecting MRs in PNoCs to ensure reliable and high bandwidth communication, in the presence of on-die thermal variations. But there is scope for two further enhancements. First, there is a need to proactively control maximum on-chip temperature so as to prevent the resonance wavelength of the last MR R_{n+3} from drifting away from the ultimate wavelength λ_n (figure 4(b)), which we refer to as 'irrecoverable drift'. Second, at the BT temperature (figure 3), maximum trimming or tuning power is required to align MRs to resonate at their corresponding wavelengths, thus avoiding BT temperatures can reduce trimming and tuning power overhead. As shown in figure 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_{tr} are designer specified parameters. Cores whose corresponding modulating and detecting MR group temperatures are within BTZs are called boundary temperature cores (BTCs). As BTCs possess the highest trimming and tuning power overhead for their corresponding MR groups, a mechanism that reduces the number of BTCs can save trimming and tuning power. The next section describes such a mechanism.

Table 1: List of AWDTM parameters and their definitions

Symbol	Definition
IPC_i	Instructions per cycle of i^{th} core
T_i	Current temperature of i^{th} core
TN_i	Average temperature of immediate neighboring cores of i^{th} core; if this core is on chip periphery and missing neighbors, then we consider virtual neighbor cores at ambient temperature in lieu of the missing cores
PT_i	Predicted temperature of i^{th} core
T_t	Thermal threshold
BTCs	Boundary temperature cores
NBTCs	Non-boundary temperature cores
C	Regularization parameter
W	Weight vector for regression
x_i *and* y_i	Input and outputs in training and test data
ξ_i	Slack variables
E	Error function
B	Bias for cost function

V. Anti Wavelength-Drift Dynamic Thermal Management (AWDTM)

To proactively reduce thermal hotspots (which will reduce instances of 'irrecoverable drift') and control on-die temperature (to reduce BTCs), we propose a system-level anti wavelength-drift dynamic thermal management (AWDTM) technique.

A. Objective

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. This 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). The parameters used to describe AWDTM in the remainder of this section are shown in Table 1.

B. Temperature Prediction Model

We designed a support vector regression (SVR) based temperature predictor that accepts input parameters reflecting the workload for the core under consideration, in terms of instructions per cycle (IPC_i), current core temperature (T_i), and surrounding core temperatures (TN_i), and predicts the future core temperature.

Architecture: A typical SVR [24] relies on defining a prediction model that ignores errors which are situated within the ε distance of the true value. This type of a prediction model is called a ε-insensitive prediction model. Two variables $(\xi$ *and* $\varepsilon)$ 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 (nonlinear) mapping notated as Φ, and then a linear model is constructed in this high-dimensional space as shown in equations (2) and (3) below. Thus it overcomes drawbacks of linear and logistic regression towards handling non-linearity in data. This class of SVRs is called kernel based SVRs which uses kernel κ as shown in equation (4) for implicit mapping of non-linear training data into a higher dimensional space.

$$CF = min\ \frac{1}{2}\ W^T.W + C\ \sum_{i=1}^{n}(\xi_i + \xi_i{}^*) \quad (1)$$

Subject to:

$$y_i - W^T\Phi(\ x_i) - b\ \leq \varepsilon +\ \xi_i\ \ (\xi_i \geq 0, i = 1, 2, \ldots, n) \quad (2)$$

$$W^T\Phi(\ x_i) + b -\ y_i \leq \varepsilon +\ \xi_i{}^*\ (\xi_i{}^* \geq 0, i = 1, 2, \ldots, n) \quad (3)$$

$$\kappa(\ x_i,\ x_j) = \Phi(x_i)^T\Phi(x_j) \quad (4)$$

SVR performs linear regression in this high-dimension space using ε-insensitive loss and, at the same time, tries to reduce model complexity by minimizing $W^T.W$. This can be described by introducing (non-negative) slack variables ξ_i and $\xi_i{}^*$ *(i = 1 to n)*, to measure the deviation of training samples outside the ε-insensitive band. Thus SVR is formulated as minimization of the cost function in equation (1) with constraints shown in equations (2) and (3).

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) [25] (Gaussian kernel) as shown in equation (5). The RBF kernel improves the accuracy of SVR when data has non-linearity in the original space. We performed a sensitivity analysis (SA) to determine regularization parameter (C) and 'gamma' (γ) values of the kernel based SVR (see Section VI.A for chosen values). This SA overcomes the possibility of over fitting of training data and improves accuracy.

$$\kappa(\ x_i,\ x_j) = exp\left(-\gamma\left|\ x_i -\ x_j\right|^2\right) \quad (5)$$

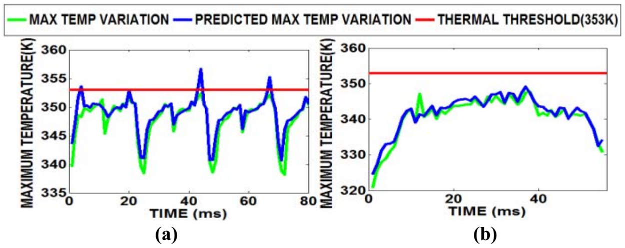


Figure 5: Actual and predicted maximum temperature variation with execution time for (a) fluidanimate (FA) (b) radiosity (RD) benchmarks executed on 64-core platform executing 32 threads

Training and Accuracy: We trained our SVR model using a set of multi-threaded applications from the PARSEC [22] and SPLASH-2 [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 shown in equations (1)-(3) to fit the training set.

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:        PT_i = SVR_predict_future_temperature (T_i, TN_i, IPC_i)
3:    end for
4:    for each core i do // Loop that checks for free BTCs and NBTCs
5:        if T_i in BTZ and IPC_i == 0 then
6:            List_BTC = Push i  //add core to BTC list
7:        else if IPC_i == 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 PT_i ≥ T_t 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

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%. Figure 5(a) and 5(b) show actual and predicted on-chip temperature variations for a 64-core platform executing 32 threads of the FA and RD benchmarks. 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.

C. *Thermal Management Framework*

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).

VI. EXPERIMENTS

A. *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 node 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 V.B). 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 planar 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 key materials used in the construction of the 3D-stack in the 3D-ICE tool and their properties are shown in Table 2.

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 III, 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.

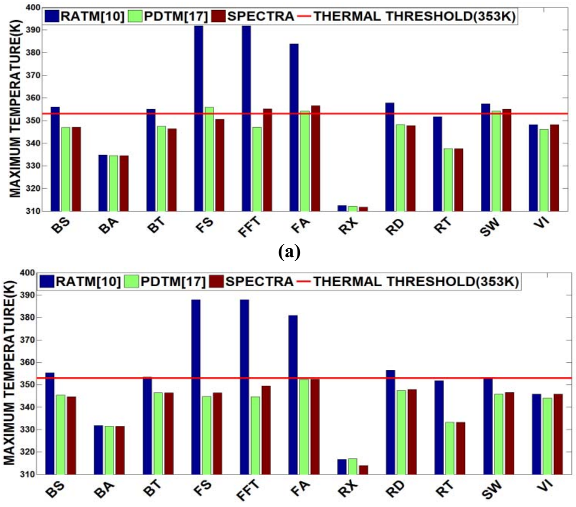
B. Experiment Results

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.

Figure 6 (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 figure 6(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 figure 6(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.

Table 2: 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 di oxide	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

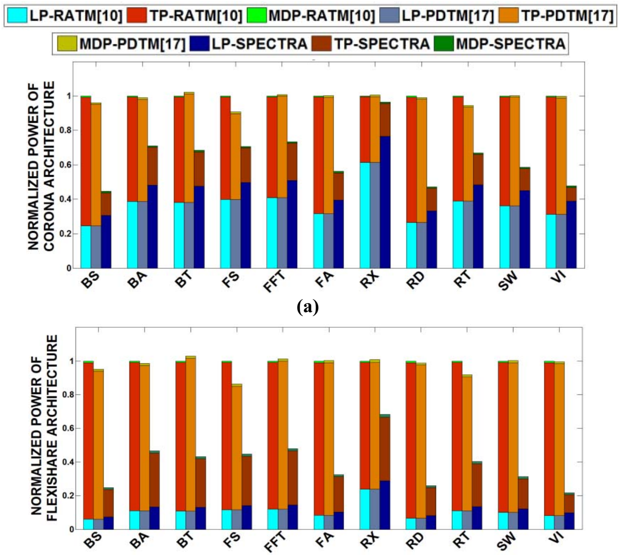


(a)

(b)

Figure 6: Maximum temperature comparison of SPECTRA with RATM [10] and PDTM [17] for (a) 48 (b) 32 threaded PARSEC and SPLASH-2 benchmarks executed on 64-core multicore system with Corona PNoC.

Figure 7 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 figures 9(a)-(b), SPECTRA has 71% and 70.2% lower trimming and tuning power on average compared to RATM and PDTM for these PNoC architectures. 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.



(a)

(b)

Figure 7: 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.

Figure 7 also shows the laser power overheads of the three frameworks for the Corona and Flexishare architectures. It can be observed that Corona and Flexishare with SPECTRA need 24.7% and 21.4% higher laser power compared to Corona and Flexishare architectures with RATM and PDTM. The main reason for this increase in the laser power is because our AMA technique in SPECTRA increases the number of MRs to modulate and receive data at higher temperatures, which in turn increase through losses in the waveguides of both Corona and Flexishare architectures, thus requiring additional laser power. Further from these results it can be summarized that the laser power overhead in Corona is higher than better performance optimized architecture of Flexishare. Despite the higher laser power overhead, SPECTRA saves considerable trimming/tuning power to ultimately achieve overall power reduction. From the power analysis in figure 7(a), it can be observed that SPECTRA with Corona has 35.2% and 34.8% lower total power consumption compared to Corona with RATM and PDTM respectively. Further from figure 7(b) it can be seen that Flexishare with SPECTRA has 61% and 60.5% lower power dissipation compared to RATM and PDTM.

In summary, from the above results, it is apparent that our proposed 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 and energy savings compared to previous approaches strongly motivate the use of our thermal management framework in future PNoC based multicore architectures.

VII. Conclusions

We have presented the SPECTRA framework that combines two novel dynamic thermal management mechanisms for the reduction of maximum on-chip temperature and 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 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 61% and trimming and tuning power by up to 71%.

VIII. Acknowledgments

This research is supported by grants from SRC, NSF (CCF-1252500, CCF-1302693), and AFOSR (FA9550-13-1-0110).

References

[1] Y. A. Vlasov et al., “Silicon CMOS-integrated nano-photonics for computer and data communications beyond 100G,” IEEE Comm. Mag., vol. 50, no. 2, pp. 67–72, Feb. 2012.

[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 et al., "A reconfigurable silicon-photonic network with improved channel sharing for multicore architectures," in ACM GLSVLSI, May 2015.

[6] S. Bahirat et al., "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”, in 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 International 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, pp.1–2.

[15] M. Mohamed et al., “Reliability-Aware Design Flow for Silicon Photonics On-Chip Interconnect,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 22, no. 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 Proc. SC, 2011.

[20] S. Li et al., “McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures,” in 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.