Leakage-Aware Predictive Thermal Management for Multicore Systems Using Echo State Network

Hai Wang, Xingxing Guo, Sheldon X.-D. Tan, *Senior Member, IEEE*, Chi Zhang, He Tang, *Member, IEEE*, and Yuan Yuan

***Abstract*—Leakage power is becoming significant in new generation IC chips. As leakage power is nonlinearly related to temperature, it is challenging to manage the thermal behavior of today's multicore systems, since thermal management becomes a nonlinear control problem. In this paper, a new predictive dynamic thermal management (DTM) method with neural network thermal model is proposed to naturally consider the inherent nonlinearity between leakage and temperature. We start with analyzing the problems of using recurrent neural network (RNN) to build the nonlinear thermal model, and point out that there is exploding gradient induced long-term dependencies problem, leading to large model prediction errors. Based on this analysis, we further propose to use echo state network (ESN), which is a special type of RNN, as the leakage-aware nonlinear thermal model. We theoretically and experimentally show that ESN achieves much higher accuracy by completely avoiding the long-term dependencies problem. On top of this nonlinear ESN thermal model, we propose a novel model predictive control (MPC) scheme called ESN MPC, which uses iterative steps to find the optimal future power recommendations for thermal management. Being able to consider the leakage-temperature nonlinear effects and equipped with advanced control technique, the new method achieves an overall high quality temperature management with smooth and accurate temperature tracking. The experimental results show the new method outperforms the state-of-the-art leakage-aware multicore DTM method in both temperature management quality and computing overhead.**

***Index Terms*—Dynamic thermal management (DTM), echo state network (ESN), leakage power, model predictive control (MPC), multicore.**

I. Introduction

POWER density keeps increasing with technology scaling, causing severe thermal related problems in high performance multicore systems, including system reliability and performance degradations [1]–[4]. In order to find the economical and efficient methods to solve the high temperature issue and improve both system performance and reliability, researchers have proposed dynamic thermal management (DTM) methods, which control the thermal behavior of multicore systems by management actions, including task migration [5]–[10], dynamic voltage and frequency scaling (DVFS) [11]–[15], etc. To guide these management actions, modern DTM methods are employed with advanced control schemes. For example, model predictive control (MPC) using linear thermal models was proposed to provide system power recommendation [16]–[19]. With the help of MPC, the management actions, such as DVFS and task migration can be correctly executed.

However, most DTM methods do not consider leakage power properly, resulting in less accurate thermal management [20]. For current high performance systems manufactured using new technology, leakage power, which even accounts for over 50% of the total power consumption, cannot be neglected anymore [21]. To make matters worse, leakage power depends on temperature exponentially [22], [23], forming a positive feedback between power and temperature: temperature rise will cause the leakage power increase, and will in turn cause the temperature to rise further, which may lead to thermal runaway in the worst case. Therefore, the leakage power induced thermal problem has already become one of the most important limiting factors of IC system performance today.

The major challenge of considering leakage power in DTM lies in building a leakage-aware thermal model which is accurate and works well with DTM methods. It comes from the fact that most DTM methods require linear thermal models. However, the accurate leakage-aware thermal model is inherently nonlinear as aforementioned. To mitigate this problem, some approximation-based thermal models considering leakage power were proposed, including explicit linear approximation models [24]–[28], implicit linear approximation models by system identification [29]–[32], piecewise linear approximation models [33], [34], and polynomial approximation models [20]. However, all these models are problematic when integrated into DTM as will be discussed in Section II.

Manuscript received June 29, 2018; revised December 14, 2018 and February 22, 2019; accepted April 28, 2019. Date of publication May 7, 2020; date of current version June 18, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61404024, in part by the Fundamental Research Funds for the Central Universities under Grant ZYGX2016J043, and in part by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry. This paper was recommended by Associate Editor A. K. Coskun. *(Corresponding author: Hai Wang.)*

H. Wang, X. Guo, C. Zhang, and H. Tang are with the State Key Laboratory of Electronic Thin Films and Integrated Devices, University of Electronic Science and Technology of China, Chengdu 610054, China, and also with the School of Electronic Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China (e-mail: wanghai@uestc.edu.cn).

S. X.-D. Tan is with the Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA 92521 USA.

Y. Yuan is with the School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China.

Digital Object Identifier 10.1109/TCAD.2019.2915316

In this paper, we propose a new leakage-aware DTM method. In order to handle the nonlinear dependency between leakage power and temperature accurately, we propose to use neural network-based control scheme for DTM. After analyzing the problems in applying recurrent neural network (RNN) to the leakage-aware DTM, we find echo state network (ESN) not only considers the inherent nonlinearity between leakage and temperature but also avoids the long-term dependencies problem in normal RNN. Then, MPC specially designed for this ESN thermal model is introduced to calculate the future power recommendations for the thermal management. Being able to consider the leakage-temperature nonlinear effects and equipped with advanced control technique, the new method is able to achieve an overall high quality temperature management with smooth and accurate temperature tracking.

The remaining parts of this paper are organized as follows. In Section II, we review some relevant researches in DTM, and present the motivation and major contributions of this paper. In Section III, we introduce the leakage power modeling and thermal modeling techniques, which serve as the basic knowledge of leakage-aware DTM. Then, we demonstrate the new leakage-aware DTM using neural network-based control in Section IV. The experimental results showing the performance of the new method are presented in Section V. Finally, Section VI concludes this paper.

II. Related Work and New Contributions

In this section, we briefly review some relevant researches in DTM, leakage-aware thermal modeling, and leakage-aware DTM for multi/many-core microprocessors.

On the DTM side, many methods have been proposed to improve the thermal related performance of multi/many-core systems [35], [36], and data centers [37]. DTM method targeting CPU-GPU co-optimization was introduced in [38] to improve mobile gaming performance. Palomino *et al.* proposed adaptive temperature optimization [39] and approximate computing [40] based DTM methods for video coding process. Machine learning-based DTM method for HEVC encoding was introduced in [41]. DTM method which improves both the performance and reliability of the 3-D ICs was recently proposed in [42].

The DTM methods above are usually combined with management actions like task migration and DVFS. Task migration switches tasks among different cores in multi/many-core systems to lower the peak temperature of the chip [5]–[9], and can also be used to lower the energy consumption in heterogeneous multicore systems [10]. DVFS controls voltage and operating frequency to adjust the heat dissipation of the chip [11], [12]. It was also applied to dark silicon systems to determine the voltage or frequency levels of the active cores [13], [14], which was further improved by introducing the thermal safe power budget (TSP) [43] and dynamic power budget (GDP) [44]. Works combining task migration and DVFS were presented in [15], [19], [45], and [46].

The above management actions should be guided by control schemes. As a result, advanced control methods using MPC with linear models were proposed to improve the management quality [16]–[19], but they failed to consider the nonlinearity between leakage and temperature.

In order to handle the nonlinearity between leakage and temperature in DTM and thermal simulation, some leakage-aware thermal models have been proposed. These models can be basically classified into three categories: 1) linear approximation models; 2) piecewise linear approximation models; and 3) polynomial approximation models. Linear models which approximate the nonlinearity between leakage and temperature linearly were presented in [24]–[28]. System identification-based linear thermal models were also proposed in [29]–[32], which implicitly linearize the leakage. However, these linear models suffer from low accuracy issue caused by the large linear approximation error. Piecewise linear approximation models were proposed to improve the accuracy of the linear models [33], [34]. However, they can be hardly integrated into an advanced control scheme due to their complex structures for implementation, so no piecewise linear approximation model-based DTM has ever been proposed. Some researchers proposed polynomial-based models to approximate the nonlinearity between leakage and temperature [20]. Although this complicated model improves accuracy, it can only be applied to thermal management for single-core systems, because its polynomial is scalar function-based [20]. In recent years, some learning-based thermal modeling approaches have been proposed [47], [48], which have potential in leakage consideration.

There are very few existing leakage-aware DTM methods based on the leakage-aware thermal models. The method in [25] minimizes the maximum temperature for periodic hard real-time systems using linear leakage-aware thermal model. A similar linear model is also used in [45], but without advanced control scheme. In addition to the DTM methods mentioned above with linear model, a DTM method with quadratic polynomial-based leakage-aware thermal model was introduced in [20]. However, as mentioned before, this DTM method can only be used for single-core systems instead of multicore systems.

The discussions above reveal that it is difficult to design an accurate leakage-aware DTM method for multicore systems. In this paper, we resolve this problem by proposing a novel leakage-aware DTM with neural network-based nonlinear thermal model. The major contributions of this paper are summarized as follows.

1) In order to handle the nonlinearity between leakage and temperature, we propose to build an RNN-based thermal model for the multicore system. Since RNN is a nonlinear model itself, the leakage induced nonlinearity can be accurately modeled with proper RNN model structure and training process.
2) We analyze the problems of using RNN-based thermal model in leakage-aware DTM. Specifically, with both theoretical analysis and experimental evidence, we reveal that there is significant exploding gradient induced long-term dependencies problem for normal RNN model in this application. Because of this, normal RNN model shows large temperature estimation error, thus cannot be used for leakage-aware DTM.
3) Based on the analysis above, we propose to use ESN, which is a special type of RNN, for leakage-aware

thermal modeling. We show theoretically and experimentally that ESN is able to avoid the exploding gradient induced long-term dependencies problem and thus enables the new leakage-aware nonlinear DTM.

4) We specially designed an ESN-based MPC framework called ESN MPC for the leakage-aware DTM problem. It contains additional leaky units to better deal with the long-term dependencies problem [49], [50] and ignores the high order Taylor expansion terms to reduce computing overhead compared with the method in [51]. The detailed steps of integrating the ESN-based leakage-aware thermal model into the specially designed MPC is demonstrated. The ESN MPC framework is able to provide accurate dynamic power adjustment recommendations for the multicore systems.
5) We have experimentally compared the ESN-based thermal model with the recently proposed artificial neural network (ANN)-based thermal model. Our numerical results show that the ESN-based thermal model is more accurate than the ANN-based thermal model, because of its superior ability in dynamic system modeling thanks to its recurrent structure.
6) We have also experimentally compared the ESN MPC-based DTM method with one state-of-the-art linear leakage model-based multicore DTM method. Our numerical results show the new method outperforms the state-of-the-art leakage-aware multicore DTM method in both management quality and computing overhead, because it handles nonlinearity in a natural and efficient way. Furthermore, compared with the existing polynomial model-based method, the new method can easily handle multicore systems without restriction.

III. Background

In this section, the leakage power model used in this paper will be introduced first. After that, we briefly review the traditional leakage-aware thermal modeling techniques, and show their problems for DTM application.

A. Modeling of the Leakage Power

It is well known that, the total power of chip, denoted as p, is composed of dynamic power and leakage power (which is also called static power). The dynamic power, denoted as p_d, depends on the activity of the chip, and thus can be easily estimated by performance counter-based methods [52]–[54]. Unlike dynamic power, leakage power p_s is independent of the chip's activity and caused by leakage current I_{leak} as

$$p_s = V_{\text{dd}} I_{\text{leak}}. \tag{1}$$

The values of leakage power are harder to obtain than dynamic power, mainly because of the special temperature sensitivity caused by leakage current. IC leakage current has various components, including subthreshold current, gate current, reverse-biased junction leakage current, and so on. Among these components, subthreshold current I_{sub} and gate leakage current I_{gate} are the dominant parts. As a result, we can ignore other parts of leakage and get the leakage current approximation [55]–[57] as

$$I_{\text{leak}} = I_{\text{sub}} + I_{\text{gate}}. \tag{2}$$

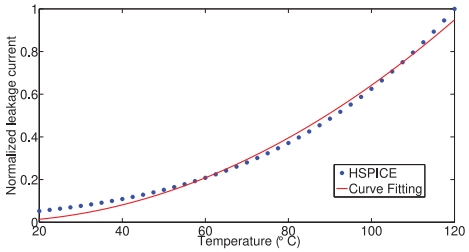


Fig. 1. Comparison of leakage of a PTM-MG 7-nm FinFET from HSPICE simulation with its curve fitting result using (3).

The subthreshold current is modeled in the commonly accepted MOSFET transistor model BSIM 4 [58] as (also apply $V_{\text{DS}} \gg v_T$ [56])

$$I_{\text{sub}} = K {v_T}^2 e^{\frac{V_{\text{GS}} - V_{\text{th}}}{\eta v_T}} \left(1 - e^{\frac{-V_{\text{DS}}}{v_T}}\right) \approx K {v_T}^2 e^{\frac{V_{\text{GS}} - V_{\text{th}}}{\eta v_T}} \tag{3}$$

where $v_T = (kT_p/q)$ is the thermal voltage and T_p is a scalar representing temperature at one place,[1] K and η are process related parameters, and V_{th} is the threshold voltage.

While the subthreshold current is highly related to temperature, the gate current I_{gate}, which results from tunneling between the gate terminal and the other three terminals, does not depend on temperature and can be considered as a technology-dependent constant.

Apparently, the leakage current has a complex relationship with temperature. In this paper, we use (1)–(3) to model the leakage power considering such relationship. The parameters of leakage current can be obtained by curve fitting using HSPICE simulation data. In order to see the accuracy of the model used, Fig. 1 shows an HSPICE simulation result of leakage using 7-nm PTM-MG FinFET models for high-performance applications (7-nm PTM-MG HP nMOS and HP pMOS) provided online at [59], and its curve fitting result using approximate leakage model. From the figure, we can see that the leakage power model has high accuracy for all common temperatures of IC chips.

We conclude that the leakage power distribution depends mainly on the temperature distribution for a certain chip with constant physical parameters. Since temperature also depends on power, in order to view the whole picture, traditional thermal model of IC chip is used to describe temperature's dependency on power as shown next.

B. Traditional Leakage-Aware Thermal Modeling and Its Problems

In order to calculate the full-chip temperature distribution, a thermal model that links the power and temperature is needed. Traditionally, to perform thermal analysis for an IC chip, the chip and its package are divided into multiple blocks called thermal nodes, with the partition granularity determined by accuracy requirements. Then the thermal resistances and capacitances among these thermal nodes are computed, which model the thermal transport and power response behaviors.

[1] T_t introduced latter in (4) is a vector representing temperatures at multiple positions.

For example, for a n-core system with m total thermal nodes, we can generate its thermal model as

$$GT_t(t) + C\frac{dT_t(t)}{dt} = BP(T_t, t)$$
$$Y(t) = LT_t(t) \tag{4}$$

where $T_t(t) \in \mathbb{R}^m$ is the temperature vector (distinguished from T_p, which is a scalar representing temperature at only one place), representing temperatures at m places of the chip and package; $G \in \mathbb{R}^{m\times m}$ and $C \in \mathbb{R}^{m\times m}$ contain equivalent thermal resistance and capacitance information, respectively; $B \in \mathbb{R}^{m\times n}$ contains the power injection topology information; $P(T_t, t) \in \mathbb{R}^n$ is the power vector with power dissipations of n cores, including both dynamic power vector P_d and leakage power vector P_s, i.e., $P(T_t, t) = P_s(T_t, t) + P_d(t)$, reminding that leakage power $P_s(T_t, t)$ is actually a function of temperature T_t modeled using (1)–(3); $Y(t) \in \mathbb{R}^n$ is the output temperature vector of n cores; $L \in \mathbb{R}^{n\times m}$ is the corresponding output selection matrix which selects the n core temperatures from $T_t(t)$. For the detailed structure of the thermal model, please refer to [45] and [60].

The leakage power $P_s(T_t, t + h)$ is a nonlinear function of current temperature $T_t(t + h)$, leading to the fact that we need $T_t(t + h)$ to compute $P_s(T_t, t + h)$ while we also need $P_s(T_t, t + h)$ to compute $T_t(t + h)$, similar to the famous chicken or the egg causality dilemma. As a result, $T_t(t + h)$ cannot be calculated directly.

Iteration-based method has been proposed to compute the temperature and leakage power by providing an initial guess [34], [61], [62]. Although this method is pretty accurate, it cannot be used in DTM because it is extremely time consuming. In this paper, we use the iteration-based thermal estimation method as the accuracy golden baseline (called "golden" in short) and it also serves as the multicore system plant. Detailed steps of the iteration-based method are discussed in our previous work [34].

In order to find a practical leakage-aware thermal model for DTM, researchers proposed to approximate the nonlinear function (3) using linear function, piecewise linear function, or simple polynomials. But all these methods show limitations in DTM as discussed in Section II.

IV. Leakage-Aware DTM Using Echo State Network-Based Predictive Control

As discussed in the previous sections, there are very few leakage-aware DTM works for multicore systems. In this section, we present a new leakage-aware DTM method using a neural network-based nonlinear thermal model and nonlinear MPC.

This section is organized as follows. First, in Section IV-A, we analyze the performance of the general RNN structure-based thermal model, and point out it does not work well for leakage-aware DTM because of the exploding gradient induced long-term dependencies problem. Then, in order to avoid such problem, we propose to use ESN model, which is an RNN with special structure, for leakage-aware DTM. The structure and training of ESN for thermal management application are presented in Section IV-B. Finally, in Section IV-C, we demonstrate the detailed steps of integrating the ESN-based thermal model into the new nonlinear ESN MPC framework to perform leakage-aware DTM.

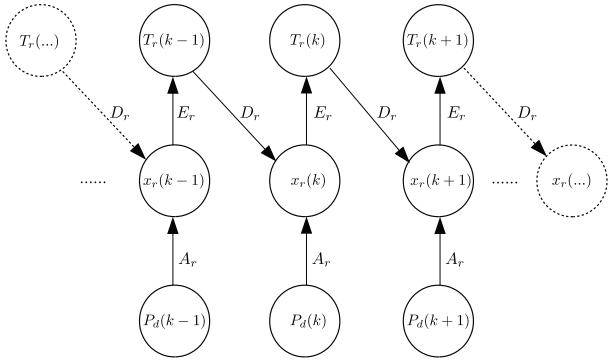


Fig. 2. Simple RNN architecture, whose recurrence is the feedback connection from the output to the hidden layer. It has the problem of learning long-term dependencies when it is used as the thermal model for leakage-aware DTM.

A. Leakage-Aware Thermal Modeling Using RNN and Its Long-Term Dependencies Problem

1) RNN-Based Leakage-Aware Thermal Model: RNN is a deep network specialized in sequence modeling. It is invented to deal with data in vector sequence form by the machine learning community [50]. Because dynamic systems produce the output vector sequence from a given input vector sequence, RNN also can be used as a black box model for dynamic systems, especially for nonlinear dynamic systems [63]. In addition, RNN has a simple structure, which makes it easier to be integrated into an advanced control framework than some other complex neural networks.

In order to improve DTM quality of multicore systems by accurately considering the nonlinearity between the leakage current and temperature, it is natural to think of using RNN as the leakage-aware thermal model. Although RNN is powerful in many applications, we show in this paper that it is difficult to train the normal RNN for leakage-aware DTM problem because of its problem of learning long-term dependencies in the training process [50], [64]. With the problem of learning long-term dependencies, the accuracy of the RNN model will suffer, especially for an RNN that requires a long sequence to train as in leakage-aware DTM.

Here, we use a simple RNN shown in Fig. 2 as an example to demonstrate this problem. Because RNN can naturally consider the nonlinearity between leakage power and temperature, we just need dynamic power $P_d(k)$ as the input and leakage power $P_s(k)$ should be handled automatically inside RNN. $T_r(k)$ is the output temperature of RNN, containing the on-chip temperatures only.[2] $x_r(k)$ is the state, which is also called the hidden unit. In addition, there are matrices A_r, D_r, and E_r, representing the weighted connections between input-to-hidden, output-to-hidden, and hidden-to-output, respectively, which are called weight matrices. This RNN outputs the on-chip temperatures $T_r(k)$ at each time step, and has recurrent connections from the output at one time step to the hidden units at the next

[2]We do not need the explicit package temperatures in most applications. If certain package temperatures are explicitly needed, we can simply add them to $T_r(k)$.

time step. Please note that we can put more than one hidden unit at each time step, in order to increase the model capacity.

Assume the multicore system has n cores ($T_r(k) \in \mathbb{R}^n$), n power sources ($P_d(k) \in \mathbb{R}^n$), and there are q hidden units ($x_r(k) \in \mathbb{R}^q$) used at each time step, then this simple RNN architecture can be written as

$$\begin{aligned} x_r(k) &= f(A_r P_d(k) + D_r T_r(k-1) + \alpha) \\ T_r(k) &= E_r x_r(k) + \beta \end{aligned} \tag{5}$$

where $A_r \in \mathbb{R}^{q\times n}$ is specifically called input weight matrix, $D_r \in \mathbb{R}^{q\times n}$ is called recurrent weight matrix, and $E_r \in \mathbb{R}^{n\times q}$ is called output weight matrix. The activation function f is an element wise nonlinear function. Usually, f is chosen as logistic sigmoid function $f(k) = [e^k/(e^k+1)]$ or hyperbolic tangent function $f(k) = \tanh(k)$ in RNN. $\alpha \in \mathbb{R}^q$ and $\beta \in \mathbb{R}^n$ are the bias vectors.

2) Long-Term Dependencies Problem in RNN-Based Leakage-Aware Thermal Model: The RNN model has to be trained before usage, i.e., the proper values of its weight matrices (A_r, D_r, E_r), which lead to an accurate RNN for the specific application, need to be determined in the training process. Assume we have a training set comprises input (dynamic power vector trace) and output (system temperature vector trace) samples of n_k time steps obtained using the slow but accurate golden iteration-based leakage-aware thermal estimation method [61], [62]. Let us denote $T_{\text{tr}}(k)$ as the output temperature from training samples and $T_r(k)$ as the output temperature from RNN model at time k. In order to get an accurate RNN model, we need to make the output temperature $T_r(k)$ of RNN as close as possible to the training temperature data $T_{\text{tr}}(k)$, by tuning the RNN weight matrices. As a result, the goal of the training process is to minimize the following cost function:

$$\mathcal{J} = \sum_{1\le k\le n_k} \|T_{\text{tr}}(k) - T_r(k)\|_2. \tag{6}$$

To minimize the cost function $\mathcal{J}$, our task is to search for the weight matrices (A_r, D_r, E_r) which reduce the cost function gradient $\nabla\mathcal{J}$ to zero in an iterative way. However, long-term dependencies problem may occur during the gradients computation process, leading to RNN model accuracy degradation, as explained in the following.

Here, we illustrate such long-term dependencies problem by computing the derivative of the cost $\psi(k) := T_{\text{tr}}(k) - T_r(k) \in \mathbb{R}^n$ at time k in (6) with respect to a weight w_{ij} in the weight matrices as an example

$$\frac{\partial\psi(k)}{\partial w_{ij}} = \sum_{1\le l\le k} \left(\frac{\partial\psi(k)}{\partial x_r(k)} \frac{\partial x_r(k)}{\partial T_r(l)} \frac{\partial T_r^+(l)}{\partial w_{ij}} \right) \tag{7}$$

where $([\partial\psi(k)]/[\partial x_r(k)])([\partial x_r(k)]/[\partial T_r(l)])([\partial T_r^+(l)]/[\partial w_{ij}])$ measures how w_{ij} at time l affects the $\psi(k)$ at time k, $([\partial T_r^+(l)]/[\partial w_{ij}])$ is the "immediate" partial derivative by taking $T_r(l-1)$ as a constant, and

$$\begin{aligned} \frac{\partial x_r(k)}{\partial T_r(l)} &= \frac{\partial x_r(k)}{\partial T_r(k-1)} \left(\prod_{l+2\le i\le k} \frac{\partial T_r(i-1)}{\partial x_r(i-1)} \frac{\partial x_r(i-1)}{\partial T_r(i-2)} \right) \\ &= \left(\prod_{l+2\le i\le k} \text{diag}\big(f'(z_r(i))\big) D_r E_r \right) \\ &\quad \times \text{diag}\big(f'(z_r(l+1))\big) D_r \end{aligned} \tag{8}$$

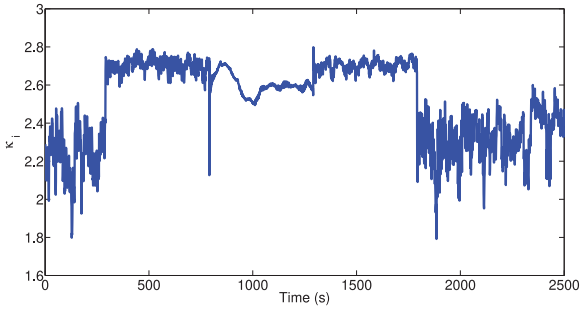


Fig. 3. Largest singular value κ_i of $\text{diag}(f'(z_r(i)))D_rE_r$, showing exploding gradient problem because $\kappa_i > 1$. This RNN model has three hidden layers with ten neurons in each layer. All other RNN models show similar results.

where $z_r(i)$ is defined as $z_r(i) = A_rP_d(i) + D_rT_r(i-1) + \alpha$ only to simplify notation and diag is an operator which converts a vector into a diagonal matrix.

The problem of learning long-term dependencies can be induced by either vanishing gradient or exploding gradient. In order to analyze the long-term dependencies problem and distinguish its cause, we mainly focus on the multiplication $\prod_{l+2\le i\le k} \text{diag}(f'(z_r(i)))D_rE_r$ in (8). Let us define $\kappa_i = \|\text{diag}(f'(z_r(i)))D_rE_r\|_2$, which is also the largest singular value of $\text{diag}(f'(z_r(i)))D_rE_r$. Then, if $\kappa_i < 1$ and $k \gg l$, the value of $\|\prod_{l+2\le i\le k} \text{diag}(f'(z_r(i)))D_rE_r\|_2$ will go to 0, indicating the vanishing gradient induced long-term dependencies problem. Similarly, the exploding gradient induced long-term dependencies problem may happen when $\kappa_i > 1$ and $k \gg l$. More discussions on the difficulty of learning long-term dependencies can be found in [50] and [64]–[66].

When encountering exploding gradient or vanishing gradient problems, it is difficult for RNN to learn the weights in the training process, which will lead to a large model error. Unfortunately, in the leakage-aware thermal modeling, there is a severe exploding gradient problem. We can see this by observing the value of κ_i shown in Fig. 3 for one RNN example where there are three hidden layers with ten neurons in each layer. In the figure, κ_i is larger than 1 for all training time k, indicating exploding gradient problem in this case. We remark that similar results are observed in all other tested RNN models with different sizes and configurations.

To see the disastrous results of this exploding gradient induced long-term dependencies problem, we built leakage-aware RNN thermal models with different sizes and hidden layer configurations using 10 000 samples obtained from the golden iteration-based method with sampling interval to be 1 s. Then, we use other 7000 samples to verify the accuracy of this model. The training and validation accuracy results are collected in Table I. Results shown in the table reveal that no matter how we adjust the model sizes and hidden layer configurations, RNN models have relatively large training error and validation error. Even the smallest average training and validation errors are larger than 6 °C and 8 °C, respectively. This means that normal RNN model is not suitable for building leakage-aware thermal model due to the exploding gradient induced long-term dependencies problem in the training process. In the next part, we will show this problem can be solved by using ESN, which has a special RNN structure.

TABLE I
ABSOLUTE TRAINING AND VALIDATION ERRORS (IN °C) OF NORMAL RNN-BASED LEAKAGE-AWARE THERMAL MODEL. ERRORS ARE LARGE FOR ALL RNNS WITH DIFFERENT CONFIGURATIONS, DUE TO THE EXPLODING GRADIENT INDUCED LONG-TERM DEPENDENCIES PROBLEM

Neuron # in layer				Train err		Val err	
l1	l2	l3	l4	max	avg	max	avg
10	0	0	0	49.2	22.4	65.2	42.6
20	0	0	0	37.5	15.7	43.4	21.3
5	5	0	0	23.2	10.9	30.5	13.1
10	10	0	0	20.6	9.0	22.7	10.3
20	20	0	0	18.3	7.4	19.4	8.6
5	5	5	0	19.5	7.9	20.2	9.5
10	10	10	0	17.5	7.2	18.2	8.1
15	15	15	0	17.3	6.7	19.3	8.7
20	20	20	0	17.5	6.5	19.7	9.3
5	5	5	5	17.9	7.4	19.5	8.5
10	10	10	10	17.1	6.4	20.4	9.7

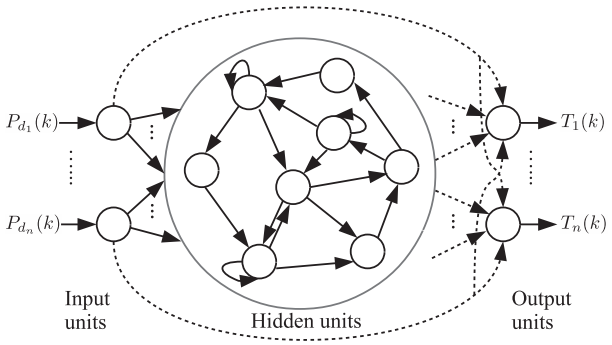


Fig. 4. ESN architecture of an n-core system. Arrows with solid lines: fixed weights which are created randomly; arrows with dashed lines: output weights which need to be trained. $P_{d_i}(k)$ is the dynamic power of the ith core and $T_i(k)$ is the temperature of the ith core.

B. ESN-Based Leakage-Aware Thermal Model for Multicore Systems

From Section IV-A2, we know that normal RNN has difficulty in learning long-term dependencies to build an accurate leakage-aware thermal model for DTM due to the exploding gradient problem in training process. In this section, we show that ESN [49], [67], [68], which is an RNN with special structure, is able to avoid this problem.

1) RNN Structure Selection for Leakage-Aware Thermal Modeling: By analyzing the difficulty in learning long-term dependencies in Section IV-A2, we know the cause of such difficulty is that the gradients [like the one in (7)], which propagate over many stages through time, tend to either vanish or explode when we train the recurrent weight matrix. Specifically, for the application of leakage-aware thermal modeling, there is severe exploding gradient induced long-term dependencies problem as shown in Section IV-A2.

In order to avoid the long-term dependencies problem in RNN, many variants of RNN were proposed with different structures. One famous variant is call the long-short term memory (LSTM) network [69], [70]. However, LSTM has a complex LSTM structure, which makes it difficult to be integrated into the DTM framework. Furthermore, LSTM was proposed to mitigate the vanishing gradient induced long-term dependencies problem, so it does not address the exploding gradient induced problem [64], which happens in leakage-aware thermal modeling as shown in Fig. 3.

On the other hand, ESN can avoid both vanishing gradient and exploding gradient induced long-term dependencies problems by learning only the output weight matrix in training. Because the long-term dependencies problems happen when we train the weights among hidden neurons using backpropagation, which causes gradients to propagate over many stages (as shown in Section IV-A2). ESN prevents this problem by avoiding the backpropagation-based training of the weights among hidden neurons. To be specific, the input and recurrent weight matrices (which contain weights among hidden neurons) of ESN are created randomly and fixed, meaning they are not trained using backpropagation. Instead, only the output weight matrix needs to be trained using simple linear regression as will be shown later. Since there is no backpropagation needed in training (but only a linear regression), there is no gradient propagation and vanishing/exploding gradient induced long-term dependencies problem in ESN. As a result, we can use ESN as the leakage-aware thermal model, which should be able to achieve high thermal prediction accuracy in DTM without the difficulty in learning long-term dependencies.

2) ESN Architecture for Leakage-Ware Thermal Modeling: The ESN architecture used for our thermal modeling is shown in Fig. 4. In the figure, $P_d(k) = [P_{d_1}(k), P_{d_2}(k), \ldots, P_{d_n}(k)]^T$ is the vector of dynamic power injections of the multicore system, and $T(k) = [T_1(k), T_2(k), \ldots, T_n(k)]^T$ contains the output on-chip temperatures. All recurrent connections of ESN are located between hidden units. The weights of the input-to-hidden units connections and hidden-to-hidden units connections are randomly assigned and fixed, which are shown as arrows with solid lines in Fig. 4. The weights of hidden-to-output units connections and input-to-output units connections should be determined in the training process, which are shown as arrows with dashed lines in Fig. 4.

ESN shown in Fig. 4 can be also written into the state space like formulation similar to the normal RNN in (5). Assume the multicore system has n cores ($T(k) \in \mathbb{R}^n$), n dynamic power sources ($P_d(k) \in \mathbb{R}^n$), and there are q hidden units ($x(k) \in \mathbb{R}^q$) in the ESN, then the ESN-based leakage-aware thermal model can be written as

$$\begin{aligned} x(k) &= (1-\gamma)x(k-1) + \gamma f(AP_d(k) + Dx(k-1)) \\ T(k) &= Ex(k) + HP_d(k) \end{aligned} \tag{9}$$

where γ is the parameter of the linear self-connection from hidden units $x(k-1)$ to $x(k)$ (such hidden units are called leaky units). When γ is close to 0, the information for a long time in the past can be remembered by ESN. When γ approaches 1, the past information is quickly discarded [50]. This is a simple and quite effective strategy used in ESN to deal with long-term dependencies problem [49]. Input matrix $A \in \mathbb{R}^{q\times n}$ and recurrent connection matrix $D \in \mathbb{R}^{q\times q}$ are randomly generated and cannot be changed in the training process. Matrices $E \in \mathbb{R}^{n\times q}$ and $H \in \mathbb{R}^{n\times n}$ represent the weighted connections between hidden-to-output and input-to-output, respectively, whose values will be learned in the training process presented next.

3) Training of the Leakage-Aware ESN Thermal Model: In this part, we introduce the process of training the ESN-based

thermal model of multicore systems. ESN training is relatively simple: we only need to train the output matrix, denoted here as $W_{\text{out}} = [E, H] \in \mathbb{R}^{n\times(q+n)}$, using linear regression as shown below.

Assume we have a training set with training input series $P_{\text{tr}}(k)$ and training output series $T_{\text{tr}}(k)$, where $k = 1, 2, \ldots, n_k$. By injecting the power input data $P_{\text{tr}}(k)$ into the ESN model (9), we can compute the state series $x(k)$, $k = 1, 2, \ldots, n_k$ easily because both A and D are known constant matrices.

Then, we collect the state series and training input series as state collection matrix $S \in \mathbb{R}^{n_k\times(q+n)}$

$$S = \begin{bmatrix} x(1), x(2), \ldots, x(n_k) \\ P_{\text{tr}}(1), P_{\text{tr}}(2), \ldots, P_{\text{tr}}(n_k) \end{bmatrix}^T.$$

Similarly, we collect training output series $T_{\text{tr}}(k)$ as output collection matrix $O \in \mathbb{R}^{n_k\times n}$

$$O = [T_{\text{tr}}(1), T_{\text{tr}}(2), \ldots, T_{\text{tr}}(n_k)]^T.$$

From (9), we have $O^T = W_{\text{out}}S^T$, which is a linear function. As a result, the trained output matrix W_{out} can be easily computed as

$$W_{\text{out}} = \left(S^{\dagger}O\right)^T \tag{10}$$

where $S^{\dagger}$ represents the pseudo-inverse of S.

Since we get the trained ESN model without using gradient propagation (which may cause the gradient to vanish or explode), the training of ESN successfully avoids the long-term dependencies problem. In this way, we obtain a trained ESN-based leakage-aware thermal model, which should be accurate and can be integrated into MPC for DTM as shown next in Section IV-C.

C. Leakage-Aware DTM With ESN MPC for Multicore Systems

MPC has a long history in the process industrial field. In recent years, MPC has been used for DTM of multicore systems [17]–[19]. However, these methods are unable to consider the nonlinearity between leakage and temperature, resulting in significant management error for systems with high leakage ratio. In Section IV-B, we have shown the new ESN-based compact thermal model, which is capable of handling the leakage induced nonlinearity. Although building and training the ESN-based thermal model are not difficult, it is not straightforward to integrate such model into the MPC-based DTM framework to compute the proper future dynamic power recommendations, because existing MPC-based DTM methods require compact linear thermal models [17]–[19]. In this section, we present a newly designed DTM framework: *ESN MPC*. In this framework, the MPC flow is specially modified to adapt the ESN-based nonlinear thermal model, and is able to provide the leakage-aware power adjustment for multicore systems.

The framework of the new ESN MPC-based leakage-aware DTM method for multicore systems is shown in Fig. 5. The basic task of ESN MPC is to calculate the input dynamic power recommendation $P_d(k+1)$, such that the future plant temperature will track a given temperature target. In order to do that, the ESN MPC predicts the future temperature $T(k+i|k)$ using the ESN thermal model (presented in Section IV-B) with current state estimation $x(k)$. Then, the proper $P_d(k+1)$ is solved from an optimization problem (represented by the "optimization" block in Fig. 5) which minimizes the difference between the predicted temperature $T(k+i|k)$ and the target temperature. Note that current state $x(k)$ is not directly available. It should be estimated using extended Kalman filter [67] with sensor temperature information $T(k)$ from the multicore system plant.

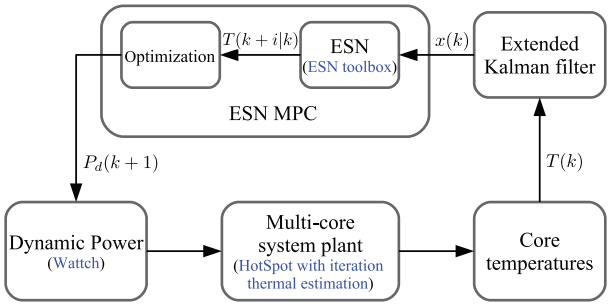


Fig. 5. Framework of ESN MPC-based leakage-aware DTM for multicore systems. Extended Kalman filter is used for state estimation. The blue phrases in parentheses are the tools used to implement the specific blocks in our experiment which will be presented in Section V.

The challenge in the ESN MPC-based DTM is how to handle the nonlinearity of the ESN thermal model properly in the power recommendation computing process. Now, we present detailed steps of the ESN MPC-based DTM.

First, at current time (assume we are at time k), we denote the future input dynamic power trajectory (which is unknown and needs to be computed in the end) into the future N_c steps (where N_c is called the *control horizon* in MPC) as

$$\mathcal{P}_d = \left[P_d(k+1)^T, P_d(k+2)^T, \ldots, P_d(k+N_c)^T\right]^T$$

and the future input dynamic power difference trajectory as

$$\Delta\mathcal{P}_d = \left[\Delta P_d(k+1)^T, \Delta P_d(k+2)^T, \ldots, \Delta P_d(k+N_c)^T\right]^T$$

where $\Delta P_d(k+i) = P_d(k+i) - P_d(k+i-1) \in \mathbb{R}^n$, $\mathcal{P}_d \in \mathbb{R}^{N_c n}$, $\Delta\mathcal{P}_d \in \mathbb{R}^{N_c n}$.

Then, the temperature predictions from current time k into the future N_p steps (where N_p is called the *prediction horizon* in MPC), denoted as $T(k+i|k)$, $i = 1, 2, \ldots, N_p$, can be expressed as a function of $\mathcal{P}_d$, using the ESN thermal model (9) and current temperature information $T(k)$ read from thermal sensors in the multicore system. These temperature predictions are written in vector trajectory $\mathcal{T} \in \mathbb{R}^{N_p n}$ as

$$\mathcal{T} = \left[T(k+1|k)^T, T(k+2|k)^T, \ldots, T\big(k+N_p|k\big)^T\right]^T$$

where $T(k+i|k)^T$ is the predicted temperatures at time $k+i$ using information of current time k.

Similarly, the target temperature vector T_{tg} is written in a vector trajectory $\mathcal{T}_{tg} \in \mathbb{R}^{N_p n}$ as

$$\mathcal{T}_{tg} = \left[T_{tg}^T, T_{tg}^T, \ldots, T_{tg}^T\right]^T.$$

Next, we will introduce the optimization process in ESN MPC, which is represented by the optimization block in Fig. 5.

As briefly mentioned before, the objective of the MPC-based DTM is to compute the proper power recommendation which brings the predicted output temperature $\mathcal{T}$ as close as possible to the target temperature $\mathcal{T}_{tg}$. This control problem is transformed into the following optimization problem:

$$\text{minimize} \quad \left\|\mathcal{T}_{tg} - \mathcal{T}\right\|_2. \tag{11}$$

Note that $\mathcal{T}$ is a function of the input power trajectory $\mathcal{P}_d$, so this optimization problem looks for the optimal future power trajectory $\mathcal{P}_d$ (power recommendation) which minimizes $\|\mathcal{T}_{tg} - \mathcal{T}\|_2$.

For practical usage, a regulation term $r_w\|\Delta\mathcal{P}_d\|_2$ may be added to the original cost function in the optimization problem (11), to form the new regulated optimization problem [71], [72]

$$\text{minimize} \quad \left\|\mathcal{T}_{tg} - \mathcal{T}\right\|_2 + r_w\|\Delta\mathcal{P}_d\|_2 \tag{12}$$

where r_w is a tuning parameter. In order to facilitate presentation, we can rewrite optimization problem (12) as

$$\text{minimize} \quad \mathcal{F}(\mathcal{P}_d) = \Psi^T\Psi + \Delta\mathcal{P}_d^T R_w \Delta\mathcal{P}_d \tag{13}$$

where $\Psi = \mathcal{T}_{tg} - \mathcal{T} \in \mathbb{R}^{N_p n}$. $R_w = r_w I \in \mathbb{R}^{N_c n \times N_c n}$ is a diagonal matrix and $I \in \mathbb{R}^{N_c n \times N_c n}$ is identity matrix.

Then, the remaining steps focus on how to compute the power recommendation trajectory to minimize $\mathcal{F}(\mathcal{P}_d)$ in (13). In order to find the $\mathcal{P}_d$ which minimizes the nonlinear function $\mathcal{F}$, the procedure is to compute the gradient of $\mathcal{F}$ against $\mathcal{P}_d$, and search the solution $\mathcal{P}_d$ along the direction where the gradient of $\mathcal{F}$ decreases in an iterative way. In this paper, we use Levenberg–Marquardt (LM) algorithm [73] for the solution search.

LM algorithm uses continuous iterations to search for the optimal solution. In each iteration, it will compute a search offset $\Delta\varepsilon$, and update the solution $\mathcal{P}_d$ as

$$\mathcal{P}_d \leftarrow \Delta\varepsilon + \mathcal{P}_d. \tag{14}$$

The problem now is how to calculate the search offset $\Delta\varepsilon$. Let us denote the Jacobian matrices of Ψ and $\Delta\mathcal{P}_d$ as F_1 and F_2, respectively

$$F_1 = \frac{\partial\Psi}{\partial\mathcal{P}_d}, \quad F_2 = \frac{\partial\Delta\mathcal{P}_d}{\partial\mathcal{P}_d}.$$

Then the offset $\Delta\varepsilon$ in LM algorithm is calculated as

$$\Delta\varepsilon = -(M + \tau\,\text{diag}(M))^{-1}\left(F_1^T\Psi + F_2^T R_w \Delta\mathcal{P}_d\right) \tag{15}$$

where

$$M = F_1^T F_1 + F_2^T R_w F_2$$

and τ is the (non-negative) damping factor that is adjusted at each iteration. If the value of $\mathcal{F}$ decreases after an iteration, divide τ by v, where v is set by experience. Inversely, if the value of $\mathcal{F}$ increases after an iteration, multiply τ by v. Please note that $\Delta\varepsilon$ in (15) can be solved using Gaussian elimination efficiently without computing $(M + \tau\,\text{diag}(M))^{-1}$ explicitly.

In order to calculate $\Delta\varepsilon$ using (15), we still need to compute the Jacobian matrices F_1 and F_2.

For the first Jacobian matrix F_1, we can write it in the following form as:

$$F_1 = -\begin{bmatrix} \frac{\partial T(k+1|k)}{\partial P_d(k+1)} & \frac{\partial T(k+1|k)}{\partial P_d(k+2)} & \cdots & \frac{\partial T(k+1|k)}{\partial P_d(k+Nc)} \\ \frac{\partial T(k+2|k)}{\partial P_d(k+1)} & \frac{\partial T(k+2|k)}{\partial P_d(k+2)} & \cdots & \frac{\partial T(k+2|k)}{\partial P_d(k+Nc)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial T(k+Np|k)}{\partial P_d(k+1)} & \frac{\partial T(k+Np|k)}{\partial P_d(k+2)} & \cdots & \frac{\partial T(k+Np|k)}{\partial P_d(k+Nc)} \end{bmatrix} \tag{16}$$

where $([\partial T(k+i|k)]/[\partial P_d(k+j)]) \in \mathbb{R}^{n\times n}$ and $F_1 \in \mathbb{R}^{N_p n \times N_c n}$.

Using the ESN (9), $([\partial T(k+i|k)]/[\partial P_d(k+j)])$ can be easily computed in the following way.

1) For all $i < j$, since the future inputs do not affect current outputs, we get
$$\frac{\partial T(k+i|k)}{\partial P_d(k+j)} = \mathbf{0} \tag{17}$$
where $\mathbf{0}$ is a $n \times n$ zero matrix.
2) For all $i = j$, we obtain
$$\frac{\partial T(k+i|k)}{\partial P_d(k+j)} = E\frac{\partial x(k+i|k)}{\partial P_d(k+j)} + H \tag{18}$$
where
$$\frac{\partial x(k+i|k)}{\partial P_d(k+j)} = \gamma\,\text{diag}\big(f'(z(k+i|k))\big)A$$
and $z(k+i|k) = AP_d(k+i) + Dx(k+i-1|k)$.
3) For all $i > j$, there is
$$\frac{\partial T(k+i|k)}{\partial P_d(k+j)} = E\frac{\partial x(k+i|k)}{\partial P_d(k+j)} \tag{19}$$
where
$$\begin{aligned}\frac{\partial x(k+i|k)}{\partial P_d(k+j)} &= (1-\gamma)\frac{\partial x(k+i-1|k)}{\partial P_d(k+j)} \\ &\quad + \gamma\,\text{diag}\big(f'(z(k+i|k))\big) \\ &\quad \times \left(D\frac{\partial x(k+i-1|k)}{\partial P_d(k+j)}\right).\end{aligned}$$

Finally, F_1 can be computed by using the formulas above.

Because $\Delta\mathcal{P}_d$ has a linear relationship with $\mathcal{P}_d$ [specifically, there is $\Delta P_d(k+i+1) = P_d(k+i+1) - P_d(k+i)$], the second Jacobian matrix F_2 is easy to compute as shown below.

1) For all $i = j$, we have
$$\frac{\partial\Delta P_d(k+i)}{\partial P_d(k+j)} = I \tag{20}$$
where $([\partial\Delta P_d(k+i)]/[\partial P_d(k+j)]) \in \mathbb{R}^{n\times n}$ for all cases.
2) For all $i = j+1$, there is
$$\frac{\partial\Delta P_d(k+i)}{\partial P_d(k+j)} = -I. \tag{21}$$
3) For all other cases, we have
$$\frac{\partial\Delta P_d(k+i)}{\partial P_d(k+j)} = \mathbf{0}. \tag{22}$$

In summary, F_2 can be simply written in the following form:

$$F_2 = \begin{bmatrix} I & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ -I & I & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & -I & I \end{bmatrix}. \tag{23}$$

C11	C12	C13	C14
C21	C22	C23	C24
C31	C32	C33	C34
C41	C42	C43	C44

Fig. 6. Configuration of the 16-core chip used in the experiment.

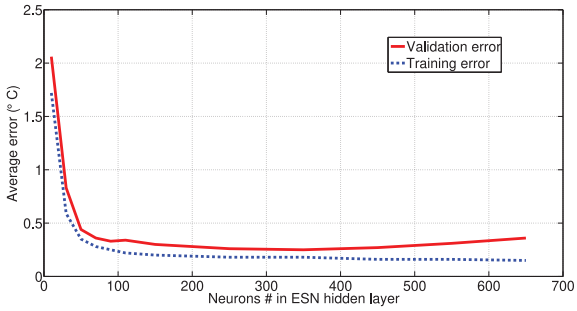


Fig. 7. Average absolute errors of ESN model with different numbers of neurons in the hidden layer. Validation error reaches minimum at around 350 neurons in the hidden layer.

Now, for each iteration, we have Jacobian matrices F_1 and F_2. Then, we can compute the search offset using (15), and update the solution for the next iteration using (14). Such iteration will continue until convergence is reached.

Note that we get a solution $\mathcal{P}_d$ by iteration at each time step, but only its first element $P_d(k+1)$ will be outputted as the power recommendation for thermal management. With the guidance of the power recommendation $P_d(k+1)$, thermal management actions will be performed to make the output temperatures T track the target temperature T_{tg} (or simply lower than T_{tg} if the system task loads are light). For a simple thermal management scheme, we can just lower the frequency of a heavy loaded core to make its dynamic power equal to the recommended dynamic power. Advanced thermal management actions based on power recommendation is also viable, such as the one presented in our previous work [19].

V. Experimental Results and Discussions

In this section, we evaluate the performance of the newly proposed leakage-aware DTM method. The experiment is performed using a 16-core system plant with chip configuration shown in Fig. 6. We place one thermal sensor for each core, which provides the on-chip temperature information to the extended Kalman filter in the DTM process as shown in Fig. 5. The ambient temperature is set to be 25 °C, and the target temperature in thermal management is set as 85 °C. By using the PTM-MG 7-nm FinFET model for high-performance applications [59], the leakage power is set to be around 40% of total power at 85 °C according to [74]. The ESN model is built by using the ESN toolbox provided online [75]. All the experiments are performed in MATLAB, including the building and training of the ESN-based thermal model. All the results are obtained on a PC with Intel Core i5-2400 CPU and 4-GB memory.

In order to show the accuracy of the ESN model, we compare it with the recently proposed ANNs-based method [48]. Then, to illustrate the advantages of our control method (ESN-based MPC), we compare it with the state-of-the-art leakage-aware DTM method for multicore systems called MAGMA [45]. We use the MAGMA open source code provided in [45], and link it to the same multicore system plant used for our new method with the same settings. Since MAGMA is based on linear approximation thermal model to handle the nonlinear leakage temperature dependency, its linear approximation range is set to a practical range of 35 °C–115 °C.

A. Accuracy Analysis of the ESN-Based Leakage-Aware Thermal Model

For model-based control method, it is critical to build a thermal model which is accurate across the temperature control range. As a result, before testing the new leakage-aware DTM method, we first verify the accuracy of the ESN-based thermal model, which naturally considers the nonlinearity between leakage and temperature. We also use thermal model built by ANN [48] for comparison.

In this experiment, we define the *golden* temperature data as the most accurate leakage-aware chip temperature data we can get, which is obtained by using iteration-based leakage-aware thermal simulation method (which is very accurate but time consuming as briefly introduced in Section III-B and discussed in details in our previous work [34]) with thermal model extracted from HotSpot [76]. The simulation time step is set as 0.01 s to ensure accuracy.

By using the golden temperature data, we can acquire the training and validation data to build the thermal models. Since we set the thermal management cycle in DTM to be 1 s, we use the power input averaged every 100 simulation steps (each simulation step is 0.01 s) and the golden temperature data at the end of each 100 simulation steps as one data sample for the training and validation of the ESN model. We collected 65 000 samples in total, and out of which, we use 38 000 samples for training and 27 000 samples for validation. It is well known that the accuracy of the neural network is highly related to the training samples. As a result, in addition to creating a wide range of output temperatures (for example, from 0 to 2000 s in Fig. 8), we also manually create some output temperature samples around the management target temperature 85 °C (for example, from 2000 to 3000 s in Fig. 8) to enhance/verify the thermal model accuracy in the DTM process. Both ESN-based and ANN-based thermal models share the same training and validation data for a fair comparison.

In order to see the model accuracy and computing overhead with different model sizes, we test ESN models with different neuron numbers in the hidden layer. The average errors for training and validation of the ESN model with different sizes are shown in Fig. 7, where the value of γ in (9) is 0.2 for all cases and the error is defined as the difference between the golden temperature and the output temperature of ESN model. From Fig. 7, we can see that the training error decreases as the ESN model size increases, and gets saturated at around

TABLE II
RUNTIME (TIME), MEMORY COST (MEM), PREDICTION DIFFERENCE (PRED DIFF), AND TRACKING DIFFERENCE (TRACK DIFF) RESULTS OF THE NEW ESN MPC-BASED DTM METHOD. RUNTIME IS RECORDED AS THE AVERAGE COMPUTING TIME FOR EACH THERMAL MANAGEMENT ACTION (EVERY 1 s). PREDICTION DIFFERENCE IS THE TEMPERATURE DIFFERENCE BETWEEN THE TARGET TEMPERATURE AND THE TEMPERATURE PREDICTION IN THE ESN MPC USING THE COMPUTED POWER RECOMMENDATION. TRACKING DIFFERENCE IS THE TEMPERATURE DIFFERENCE BETWEEN THE TARGET TEMPERATURE AND THE ACTUAL PLANT TEMPERATURE WITH ESN MPC. THE PREDICTION DIFFERENCE AND THE TRACKING DIFFERENCE ARE IN °C

Neuron # in ESN hidden layer	$N_c = 1, N_p = 1$						$N_c = 1, N_p = 2$						$N_c = 1, N_p = 3$						$N_c = 2, N_p = 3$					
	time	mem	pred diff		track diff		time	mem	pred diff		track diff		time	mem	pred diff		track diff		time	mem	pred diff		track diff	
	(ms)	(KB)	max	avg	max	avg	(ms)	(KB)	max	avg	max	avg	(ms)	(KB)	max	avg	max	avg	(ms)	(KB)	max	avg	max	avg
10	7	4	1.17	0.68	7.95	2.79	10	6	1.26	0.75	7.72	2.82	12	9	1.22	0.72	7.68	2.81	15	14	1.24	0.73	8.03	2.84
50	14	23	1.10	0.63	5.76	1.60	18	23	0.82	0.42	5.75	1.59	22	24	1.23	0.75	5.92	1.60	23	26	1.29	0.82	5.86	1.62
150	28	106	1.16	0.66	5.52	1.51	38	106	0.99	0.55	5.55	1.49	46	106	0.84	0.49	5.59	1.45	55	106	1.05	0.72	5.62	1.49
250	37	300	1.18	0.71	5.26	1.43	64	300	0.87	0.59	5.15	1.40	91	300	0.94	0.65	5.25	1.42	110	300	1.01	0.67	5.24	1.44
350	68	960	1.20	0.74	4.92	1.33	121	960	0.85	0.46	4.87	1.32	170	960	0.90	0.48	4.90	1.34	201	960	1.19	0.78	5.03	1.39
450	102	1584	1.01	0.64	5.31	1.44	239	1584	1.09	0.65	5.22	1.40	375	1584	1.05	0.63	5.12	1.37	502	1584	1.07	0.67	5.13	1.42
550	146	2364	0.98	0.58	5.81	1.51	340	2364	1.00	0.51	5.83	1.50	552	2364	0.86	0.41	5.86	1.52	792	2364	1.12	0.75	5.87	1.53

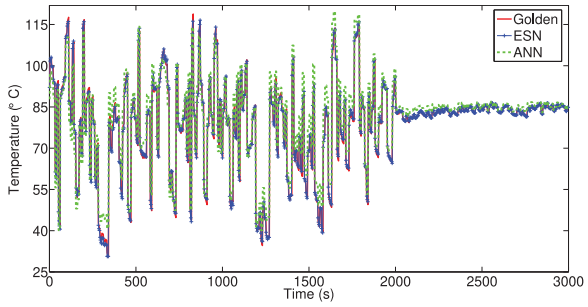


Fig. 8. Output temperatures of core C32 in the validation test using ESN model and ANN model [48], where the other cores have similar comparison results. We deliberately created temperature data close to the target temperature 85 °C from 2000 to 3000 s, in order to verify the thermal management performance around the target temperature.

100 neuron in the hidden layer. On the other hand, the validation error decreases first as the model size increases. It starts to increase later, after reaching its smallest value 0.19 °C at neuron number 350, which clearly indicates overfitting. We remark that the optimal number of neurons in the hidden layer should be determined by trading-off the validation error and runtime computing overhead (shown later in Table II) according to different application requirements, and also by avoiding the overfitting problem [77]–[79].

Since the ESN model will be used in DTM at runtime, in this paper, we use a moderate sized ESN model with 50 neurons in the hidden layer (with average validation error of 0.40 °C), which balances the computing overhead and accuracy. We plot the transient validation error of this ESN model in Fig. 8. Note that in order to verify the thermal management performance around the target temperature (which is 85 °C in this experiment), we deliberately make the temperature data to be close to 85 °C from 2000 to 3000 s.

For comparison, we also plot the results of the recently proposed ANN-based thermal model [48] in Fig. 8. This ANN model has three hidden layers with 15 neurons in each layer, which is the ANN configuration with the smallest error in our test. As seen from the figure, the ANN model has significantly larger error (with average validation error of 3.15 °C) than the new ESN-based model (with average validation error of 0.40 °C). The reason is that ESN method has the recurrent structure which makes it more suitable for dealing with time series problems and modeling dynamic systems than ANN without recurrent structure [50], [63].

B. Performance Evaluation of Leakage-Aware DTM With ESN MPC

After analyzing the accuracy of the ESN-based leakage-aware thermal model, we evaluate the performance of the new leakage-aware DTM method with ESN MPC.

The experimental flow diagram for the performance evaluation of the ESN MPC-based DTM is given previously in Fig. 5, where the blue phrases in parentheses are the tools used to implement the specific blocks in our experiment. Power estimator Wattch [80] is used to generate the dynamic power by running the standard SPEC benchmarks. The different power traces from SPEC benchmarks are randomly assigned to different cores of the multicore system. Leakage power of the multicore system plant is obtained by using the iteration-based leakage-aware thermal simulation method with simulation step 0.01 s.

In order to choose a better ESN MPC configurations, we test DTM methods with different model sizes for ESN MPC (by changing the number of neurons in the hidden layer). In addition, we also test different MPC prediction horizon lengths (N_p) and control horizon lengths (N_c). In the experiment, we set the purpose of the leakage-aware DTM as making the output temperature to track the target temperature 85 °C. r_w in MPC is chosen as 0.1 by trial and error.

The results of ESN MPC method are shown in Table II. We mainly focus on two DTM performances. The first is the temperature tracking difference between the actual plant temperature and the target temperature, which represents the effectiveness and accuracy of the management. The second is the overhead [computing overhead (runtime) and memory cost] of the DTM, with respect to different ESN model sizes as well as different N_p and N_c.

From Table II, we can see that the average plant temperature tracking difference against the target is smaller than 3 °C for all cases. For the best case in this test (with 350 neurons in the hidden layer and $N_c = 1$ and $N_P = 2$), the average tracking difference is only 1.32 °C. However, the memory cost is 960 KB and the runtime for this case is greater than 120 ms for each thermal management frame of 1 s, which is an unacceptable computing overhead as an on-line algorithm. By balancing the tracking difference and overhead, we choose the DTM configuration as: ESN model which has 50 neurons in the hidden layer with $N_c = 1$ and $N_P = 2$. In this case, the average tracking difference is 1.59 °C, the memory cost is 23 KB and the runtime is only 18 ms for the 16-core system. Because such

computation is performed on only one out of the 16 cores, the throughput degradation is estimated to be only around 0.1% at runtime (assuming there are no synchronization problems in parallel computing) or can be avoided by implementing the algorithm in low power coprocessor or FPGA. Generally, the time and memory costs grow linearly with model size, but there is an optimal model size for accuracy because overfitting may happen if the model is too large.

We also record the prediction difference of ESN MPC, which stands for the temperature difference between the target temperature and the temperature prediction in the ESN MPC using the power recommendation. The average prediction difference is within 1 °C for all cases, and this difference is caused by the regulation term in (12).

In order to show the advantage of the new method, we compare the new method (50 neurons in the hidden layer with $N_P = 2$ and $N_c = 1$) with the state-of-the-art leakage-aware DTM method MAGMA [45], which uses linear approximation to deal with nonlinearity between leakage and temperature. In MAGMA, each core is divided into 25 thermal blocks to ensure accuracy. For a fair comparison, we integrated the open source MAGMA program into the same multicore system plant as our new method. In order to do this, we also added Kalman filter to MAGMA for state estimation, because the multicore system plant can only provide the on-chip temperature through thermal sensors.

We have also performed another comparison to provide direct evidence that the new ESN-based DTM has good performance in leakage power consideration. We use the golden linear thermal model (the same as in the plant) with linear approximation-based leakage model (linearized at target temperature 85 °C), and integrate it into the MPC framework. The thermal management result using this new setting is called "golden thermal model with linear leakage." There is no thermal model error in this new setting, which is an ideal assumption and actually takes advantage over all other methods including ESN MPC (there are both thermal model error and leakage model error mixed inside ESN MPC).

In addition, we have also implemented a leakage-aware DTM using iteration-based thermal prediction, which works as the DTM baseline for management accuracy (called "baseline" in short). Since the DTM baseline is based on the golden thermal model (4) and uses iterative method to deal with nonlinear relationship between leakage and temperature, it avoids errors in both thermal modeling and leakage modeling. Note that this method is only used to provide the DTM accuracy baseline, because its golden thermal model assumption is unrealistic and its computing overhead is far too large for practical runtime usage.

The plant temperature comparison results of core C32 using ESN MPC-based method, MAGMA [45], golden thermal model with linear leakage, and the baseline are shown in Fig. 9(a). The corresponding frequencies of C32 under control are given in Fig. 9(b) with the base frequency as 2 GHz. Due to page limitation, we do not plot the results of other cores which have similar results. Once the dynamic power is supplied to the cores starting from 5 s, we can see all DTM controlled temperatures rise from a low temperature (idle temperature with only leakage power) to the target immediately upon activation with different overshoots. Then, the temperatures oscillate

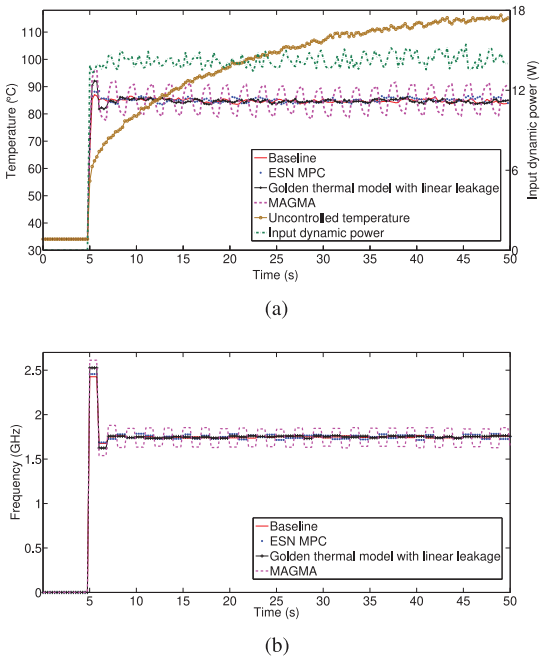


Fig. 9. Comparison results of different thermal management methods, including ESN MPC, MAGMA [45], golden thermal model with linear leakage, and baseline method. The dynamic power is supplied to the cores starting from 5 s. The tracking temperature target is set as 85 °C. (a) Temperatures of C32 using different thermal management methods. The uncontrolled input dynamic power and its corresponding temperature of C32 are also shown. (b) Frequencies of C32 using different thermal management methods.

around the target temperature because the SPEC power inputs are regulated by thermal management. For example, when the temperature is higher than the target, thermal management will lower the temperature in the next management cycle (by lowering the power input). But if over adjustment is caused due to the inaccuracy of management, the temperature will be raised (by increasing power input) in the next management cycle. From the figure, we observe that the average temperature tracking difference of baseline is less than 0.75 °C. Such tracking difference is caused by the fast power variations between two thermal management actions (with duration of 1 s in this test), which is unavoidable for any DTM methods.

From Fig. 9(a), it is clear that the temperature controlled by MAGMA [45] shows large temperature tracking difference (with average tracking difference 4.89 °C) against the target temperature. In fact, the reason that MAGMA shows large error is mostly two folds. First, MAGMA uses a simple linear model to approximate the nonlinear relationship between leakage current and temperature, which is not very accurate and leads to control accuracy loss. Second, MAGMA ignores the heat exchange among cores in the multicore system [45], which will cause error in control decision.

The new ESN-based DTM method shows good temperature tracking results in Fig. 9(a). The temperature controlled by the new method is very close to the target temperature (with 1.59 °C average tracking difference), which means the new method even performs very close to the baseline (within 0.75 °C as shown before) in temperature tracking accuracy. The reason is that the new method avoids the two

problems in MAGMA as explained here. First, the new method uses the ESN-based leakage-aware thermal model. Since ESN is a nonlinear model, it is able to accurately model the nonlinearity between leakage and temperature. This is further supported by the comparison results between ESN MPC and the golden thermal model with linear leakage: ESN MPC has much smaller temperature control overshoot than the latter method (from 5 s to around 8 s), even with the fact that the latter method has the ideal golden thermal model.

Second, the ESN-based thermal model is a multiple-input and multiple-output model which considers the core to core heat exchange (as well as core to package heat exchange). Furthermore, the MPC framework is improved in this paper to be compatible with this ESN-based thermal model, such that the future power recommendation computed by the ESN MPC fully considers the heat exchange among cores. From the observations and discussions above, we can see that leakage-aware DTM with ESN MPC method outperforms MAGMA in thermal management quality for multicore systems.

On the computing overhead side, we have tested the runtime of both the new method and MAGMA, recorded as the average computing time for each thermal management action (every 1 s). The new ESN MPC method has a runtime of 18 ms, which is much smaller than that of MAGMA which is 321 ms. In order to increase the speed of MAGMA, we reduce the resolution of MAGMA to be the same as the ESN-based method (with one thermal node for each core resulting in 46×46 sized system matrices for MAGMA). In this setting, the runtime of MAGMA is 101 ms, which is still larger than ESN-based method with 50 neurons in the hidden layer (18 ms runtime). The MAGMA accuracy becomes even worse in this resolution, with average temperature tracking difference increased to 9.74 °C. This tracking difference is significantly larger than MAGMA with higher resolution 814×814 sized system matrices (4.89 °C tracking difference) and the ESN-based method with 50 neurons (1.59 °C tracking difference).

From the observation above, we can see that even with a larger model size (814×814), the tracking accuracy of MAGMA (with 4.89 °C average tracking difference) is worse than the ESN MPC-based DTM method (with 1.59 °C average tracking difference). Reducing computing overhead by reducing the model size (46×46) brings even larger tracking difference (with 9.74 °C) for MAGMA.

In summary, the experimental results show that the ESN-based leakage-aware thermal model accurately considers the nonlinear effects between leakage and temperature. By integrating this ESN-based thermal model into MPC, the new method outperforms the state-of-the-art leakage-aware DTM method MAGMA in both accuracy and speed.

VI. Conclusion

In this paper, we proposed a new leakage-aware DTM method for multicore systems using neural network-based thermal models and improved nonlinear MPC. We show that ESN is better suited for the nonlinear leakage-aware thermal model than the normal RNN since it is able to avoid the exploding gradient induced long-term dependencies problem of RNN in leakage-aware DTM. Based on the new nonlinear thermal model, we further propose a new leakage-aware DTM method called ESN MPC, which integrates the ESN-based thermal model to provide the power adjustment recommendations for the multicore systems. The experimental results show that the new method outperforms the state-of-the-art leakage-aware multicore DTM method in both temperature management quality and computing overhead.

References

[1] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," *IEEE Micro*, vol. 32, no. 3, pp. 122–134, May/Jun. 2012.

[2] J. Henkel *et al.*, "Reliable on-chip systems in the nano-era: Lessons learnt and future trends," in *Proc. Design Autom. Conf. (DAC)*, 2013, pp. 1–10.

[3] A. K. Singh, M. Shafique, A. Kumar, and J. Henkel, "Mapping on multi/many-core systems: Survey of current and emerging trends," in *Proc. Design Autom. Conf. (DAC)*, 2013, pp. 1–10.

[4] M. Shafique, S. Garg, J. Henkel, and D. Marculescu, "The EDA challenges in the dark silicon era," in *Proc. Design Autom. Conf. (DAC)*, Jun. 2014, pp. 1–6.

[5] A. K. Coskun, T. S. Rosing, and K. Whisnant, "Temperature aware task scheduling in MPSoCs," in *Proc. Eur. Design Test Conf. (DATE)*, Apr. 2007, pp. 1–6.

[6] Y. Ge, P. Malani, and Q. Qiu, "Distributed task migration for thermal management in many-core systems," in *Proc. Design Autom. Conf. (DAC)*, Jun. 2010, pp. 579–584.

[7] T. Chantem, X. S. Hu, and R. P. Dick, "Temperature-aware scheduling and assignment for hard real-time applications on MPSoCs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 10, pp. 1884–1897, Oct. 2011.

[8] G. Liu, M. Fan, and G. Quan, "Neighbor-aware dynamic thermal management for multi-core platform," in *Proc. Eur. Design Test Conf. (DATE)*, Mar. 2012, pp. 187–192.

[9] Z. Liu, S. X.-D. Tan, X. Huang, and H. Wang, "Task migrations for distributed thermal management considering transient effects," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 2, pp. 397–401, Feb. 2015.

[10] J. Cong and B. Yuan, "Energy-efficient scheduling on heterogeneous multi-core architectures," in *Proc. Int. Symp. Low Power Electron. Design (ISLPED)*, 2012, pp. 345–350.

[11] R. Jayaseelan and T. Mitra, "A hybrid local-global approach for multi-core thermal management," in *Proc. Int. Conf. Comput.-Aided Design (ICCAD)*, 2009, pp. 314–320.

[12] A. Mutapcic *et al.*, "Processor speed control with thermal constraints," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 9, pp. 1994–2007, Sep. 2009.

[13] T. S. Muthukaruppan, M. Pricopi, V. Venkataramani, T. Mitra, and S. Vishin, "Hierarchical power management for asymmetric multi-core in dark silicon era," in *Proc. Design Autom. Conf. (DAC)*, May 2013, pp. 1–9.

[14] H. Khdr, S. Pagani, M. Shafique, and J. Henkel, "Thermal constrained resource management for mixed ILP-TLP workloads in dark silicon chips," in *Proc. Design Autom. Conf. (DAC)*, 2015, pp. 1–6.

[15] C. Tan, T. S. Muthukaruppan, T. Mitra, and L. Ju, "Approximation-aware scheduling on heterogeneous multi-core architectures," in *Proc. Asia South Pac. Design Autom. Conf. (ASP-DAC)*, 2015, pp. 618–623.

[16] F. Zanini, D. Atienza, L. Benini, and G. De Micheli, "Multicore thermal management with model predictive control," in *Proc. Eur. Conf. Circuit Theory Design*, Aug. 2009, pp. 711–714.

[17] Y. Wang, K. Ma, and X. Wang, "Temperature-constrained power control for chip multiprocessors with online model estimation," in *Proc. Int. Symp. Comput. Archit. (ISCA)*, 2009, pp. 314–324.

[18] A. Bartolini, M. Cacciari, A. Tilli, and L. Benini, "Thermal and energy management of high-performance multicores: Distributed and self-calibrating model-predictive controller," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 1, pp. 170–183, Jan. 2013.

[19] H. Wang *et al.*, "Hierarchical dynamic thermal management method for high-performance many-core microprocessors," *ACM Trans. Design Autom. Electron. Syst.*, vol. 22, no. 1, Jul. 2016, Art. no. 1.

[20] B. Shi and A. Srivastava, "Dynamic thermal management considering accurate temperature-leakage interdependency," in *Encyclopedia of Thermal Packaging: Thermal Packaging Tools*. Singapore: World Sci., 2015, pp. 39–60.

[21] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 5th ed. Waltham, MA, USA: Elsevier, 2012.

[22] D. Rossi *et al.*, "Energy-efficient near-threshold parallel computing: The PULPv2 cluster," *IEEE Micro*, vol. 37, no. 5, pp. 20–31, Sep./Oct. 2017.

[23] D. Rossi *et al.*, "A 60 GOPS/W, −1.8 V to 0.9 V body bias ULP cluster in 28 nm UTBB FD-SOI technology," *Solid-State Electron.*, vol. 117, pp. 170–184, Mar. 2016.

[24] G. Quan and Y. Zhang, "Leakage aware feasibility analysis for temperature-constrained hard real-time periodic tasks," in *Proc. Euromicro Conf. Real Time Syst.*, 2009, pp. 207–216.

[25] V. Chaturvedi, H. Huang, and G. Quan, "Leakage aware scheduling on maximum temperature minimization for periodic hard real-time systems," in *Proc. Int. Conf. Comput. Inf. Technol.*, 2010, pp. 1802–1809.

[26] S. R. Sarangi, G. Ananthanarayanan, and M. Balakrishnan, "LightSim: A leakage aware ultrafast temperature simulator," in *Proc. Asia South Pac. Design Autom. Conf. (ASP-DAC)*, 2014, pp. 855–860.

[27] M. Mohaqeqi, M. Kargahi, and A. Movaghar, "Analytical leakage-aware thermal modeling of a real-time system," *IEEE Trans. Comput.*, vol. 63, no. 6, pp. 1378–1392, Jun. 2014.

[28] H. Sultan and S. R. Sarangi, "A fast leakage aware thermal simulator for 3D chips," in *Proc. Eur. Design Test Conf. (DATE)*, Mar. 2017, pp. 1733–1738.

[29] R. Diversi, A. Tilli, A. Bartolini, F. Beneventi, and L. Benini, "Bias-compensated least squares identification of distributed thermal models for many-core systems-on-chip," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 9, pp. 2663–2676, Sep. 2014.

[30] R. Diversi, A. Bartolini, F. Beneventi, and L. Benini, "Thermal model identification of supercomputing nodes in production environment," in *Proc. IEEE Ind. Electron. Soc. Annu. Conf. (IECON)*, 2016, pp. 4838–4844.

[31] A. Bartolini, R. Diversi, D. Cesarini, and F. Beneventi, "Self-aware thermal management for high performance computing processors," *IEEE Design Test*, vol. 35, no. 5, pp. 28–35, Oct. 2018.

[32] S. Reda, K. Dev, and A. Belouchrani, "Blind identification of thermal models and power sources from thermal measurements," *IEEE Sensors J.*, vol. 18, no. 2, pp. 680–691, Jan. 2018.

[33] R. Rao and S. Vrudhula, "Performance optimal processor throttling under thermal constraints," in *Proc. Int. Conf. Compilers Archit. Synth. Embedded Syst.*, 2007, pp. 257–266.

[34] H. Wang *et al.*, "A fast leakage-aware full-chip transient thermal estimation method," *IEEE Trans. Comput.*, vol. 67, no. 5, pp. 617–630, May 2018.

[35] H. Khdr, T. Ebi, M. Shafique, H. Amrouch, and J. H. Karlsruhe, "mDTM: Multi-objective dynamic thermal management for on-chip systems," in *Proc. Eur. Design Test Conf. (DATE)*, 2014, pp. 1–6.

[36] P. Bogdan, P. P. Pande, H. Amrouch, and J. H. Shafique, "Power and thermal management in massive multicore chips: Theoretical foundation meets architectural innovation and resource allocation," in *Proc. Int. Conf. Compilers Archit. Synth. Embedded Syst.*, 2016, pp. 1–2.

[37] M. A. Oxley *et al.*, "Rate-based thermal, power, and co-location aware resource management for heterogeneous data centers," *J. Parallel Distrib. Comput.*, vol. 112, pp. 126–139, Feb. 2018.

[38] A. Prakash, H. Amrouch, M. Shafique, T. Mitra, and J. Henkel, "Improving mobile gaming performance through cooperative CPU-GPU thermal management," in *Proc. Design Autom. Conf. (DAC)*, 2016, pp. 1–6.

[39] D. Palomino, M. Shafique, A. Susin, and J. Henkel, "TONE: Adaptive temperature optimization for the next generation video encoders," in *Proc. Int. Symp. Low Power Electron. Design (ISLPED)*, 2014, pp. 33–38.

[40] D. Palomino, M. Shafique, A. Susin, and J. Henkel, "Thermal optimization using adaptive approximate computing for video coding," in *Proc. Eur. Design Test Conf. (DATE)*, 2016, pp. 1207–1212.

[41] A. Iranfar, M. Zapater, and D. Atienza, "Machine learning-based quality-aware power and thermal management of multistream HEVC encoding on multicore servers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 10, pp. 2268–2281, Oct. 2018.

[42] H. Wang *et al.*, "STREAM: Stress and thermal aware reliability management for 3-D ICs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, to be published.

[43] S. Pagani *et al.*, "Thermal safe power (TSP): Efficient power budgeting for heterogeneous manycore systems in dark silicon," *IEEE Trans. Comput.*, vol. 66, no. 1, pp. 147–162, Jan. 2017.

[44] H. Wang *et al.*, "GDP: A greedy based dynamic power budgeting method for multi/many-core systems in dark silicon," *IEEE Trans. Comput.*, vol. 68, no. 4, pp. 526–541, Apr. 2019.

[45] V. Hanumaiah, S. Vrudhula, and K. S. Chatha, "Performance optimal online DVFS and task migration techniques for thermally constrained multi-core processors," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 11, pp. 1677–1690, Nov. 2011.

[46] V. Hanumaiah and S. Vrudhula, "Energy-efficient operation of multicore processors by DVFS, task migration, and active cooling," *IEEE Trans. Comput.*, vol. 63, no. 2, pp. 349–360, Feb. 2014.

[47] A. K. Singh, C. Leech, B. K. Reddy, A.-B. M. Hashimi, and G. V. Merrett, "Learning-based run-time power and energy management of multi/many-core systems: Current and future trends," *J. Low Power Electron.*, vol. 13, no. 3, pp. 310–325, 2017.

[48] K. Zhang *et al.*, "Machine learning-based temperature prediction for runtime thermal management across system components," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 2, pp. 405–419, Feb. 2018.

[49] H. Jaeger, M. Lukoševičius, P. Dan, and U. Siewert, "Optimization and applications of echo state networks with leaky-integrator neurons," *Neural Netw.*, vol. 20, no. 3, pp. 335–352, 2007.

[50] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, U.K.: MIT Press, 2016.

[51] Y. Pan and J. Wang, "Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks," *IEEE Trans. Ind. Electron.*, vol. 59, no. 8, pp. 3089–3101, Aug. 2012.

[52] W. Wu, L. Jin, J. Yang, P. Liu, and S. X.-D. Tan, "A systematic method for functional unit power estimation in microprocessors," in *Proc. Design Autom. Conf. (DAC)*, Jun. 2006, pp. 554–557.

[53] M. Powell *et al.*, "CAMP: A technique to estimate per-structure power at run-time using a few simple parameters," in *Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2009, pp. 289–300.

[54] H. Wang, S. X.-D. Tan, X.-X. Liu, and A. Gupta, "Runtime power estimator calibration for high-performance microprocessors," in *Proc. Eur. Design Test Conf. (DATE)*, Mar. 2012, pp. 352–357.

[55] A. Abdollahi, F. Fallah, and M. Pedram, "Leakage current reduction in CMOS VLSI circuits by input vector control," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 2, pp. 140–154, Feb. 2004.

[56] Y. Liu, R. Dick, L. Shang, and H. Yang, "Accurate temperature-dependent integrated circuit leakage power estimation is easy," in *Proc. Eur. Design Test Conf. (DATE)*, 2007, pp. 1–6.

[57] R. Shen, S. X.-D. Tan, H. Wang, and J. Xiong, "Fast statistical full-chip leakage analysis for nanometer VLSI systems," *ACM Trans. Design Autom. Electron. Syst.*, vol. 17, no. 4, p. 51, Oct. 2012.

[58] W. Liu, K. Cao, X. Jin, and C. Hu, "BSIM 4.0.0 technical notes," EECS Dept., Univ. California at Berkeley, Berkeley, CA, USA, Rep. UCB/ERL M00/39, 2000. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2000/3863.html

[59] *Predictive Technology Model*. Accessed: Mar. 27, 2019. [Online]. Available: http://ptm.asu.edu

[60] H. Wang, S. X.-D. Tan, D. Li, A. Gupta, and Y. Yuan, "Composable thermal modeling and simulation for architecture-level thermal designs of multicore microprocessors," *ACM Trans. Design Autom. Electron. Syst.*, vol. 18, no. 2, p. 28, Mar. 2013.

[61] L. He, W. Liao, and M. R. Stan, "System level leakage reduction considering the interdependence of temperature and leakage," in *Proc. Design Autom. Conf. (DAC)*, 2004, pp. 12–17.

[62] J. C. Ku, S. Ozdemir, G. Memik, and Y. Ismail, "Thermal management of on-chip caches through power density minimization," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 5, pp. 592–604, May 2007.

[63] A. P. Trischler and G. M. T. D'Eleuterio, "Synthesis of recurrent neural networks for dynamical system simulation," *Neural Netw.*, vol. 80, pp. 67–78, Aug. 2016.

[64] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2013, pp. 1310–1318.

[65] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[66] R. Pascanu, T. Mikolov, and Y. Bengio, "Understanding the exploding gradient problem," *CoRR*, vol. abs/1211.5063, Nov. 2012. [Online]. Available: https://pdfs.semanticscholar.org/728d/814b92a9d2c6118159bb7d9a4b3dc5eeaaeb.pdf

[67] H. Jaeger, "Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the echo state network approach," German Nat. Res. Center Inf. Technol., Sankt Augustin, Germany, Rep. 159, 2002.
[68] H. Jaeger, "Long short-term memory in echo state networks: Details of a simulation study," School Eng. Sci., Jacobs Univ. Bremen, Bremen, Germany, Rep. 27, 2012.
[69] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[70] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, 2000.
[71] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB*. London, U.K.: Springer-Verlag, 2009.
[72] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2006.
[73] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quart. Appl. Math.*, vol. 2, no. 2, pp. 164–168, 1944.
[74] Q. Xie *et al.*, "Performance comparisons between 7-nm FinFET and conventional bulk CMOS standard cell libraries," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 8, pp. 761–765, Aug. 2015.
[75] *Echo State Networks and Reservoir Computing*. Accessed: Mar. 27, 2019. [Online]. Available: http://minds.jacobs-university.de/research/esnresearch/
[76] W. Huang *et al.*, "HotSpot: A compact thermal modeling methodology for early-stage VLSI design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 5, pp. 501–513, May 2006.
[77] S. Xu and L. Chen, "A novel approach for determining the optimal number of hidden layer neurons for FNN's and its application in data mining," in *Proc. Int. Conf. Inf. Technol. Appl.*, 2008, pp. 683–686.
[78] F. S. Panchal and M. Panchal, "Review on methods of selecting number of hidden nodes in artificial neural network," *Int. J. Comput. Sci. Mobile Comput.*, vol. 3, no. 11, pp. 455–464, 2014.
[79] W. Liu *et al.*, "Thermal modeling for energy-efficient smart building with advanced overfitting mitigation technique," in *Proc. Asia South Pac. Design Autom. Conf. (ASP-DAC)*, Jan. 2016, pp. 417–422.
[80] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A framework for architectural-level power analysis and optimizations," in *Proc. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2000, pp. 83–94.

Hai Wang received the B.S. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2007, and the M.S. and Ph.D. degrees from the University of California at Riverside, Riverside, CA, USA, in 2008 and 2012, respectively.

He is currently an Associate Professor with the University of Electronic Science and Technology of China, Chengdu, China. His current research interests include modeling, optimization, and artificial intelligence assisted design automation of very large-scale integration circuits and systems.

Dr. Wang was a recipient of the Best Paper Award nomination from Asia and South Pacific Design Automation Conference (ASP-DAC) in 2019. He has served as a Technical Program Committee Member of several international conferences, including Design, Automation and Test in Europe, ASP-DAC, and International Symposium on Quality Electronic Design, and also served as a Reviewer of many journals, including the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, and *ACM Transactions on Design Automation of Electronic Systems*.

Xingxing Guo received the bachelor's degree from Anhui University, Hefei, China, in 2016. She is currently pursuing the master's degree with the University of Electronic Science and Technology of China, Chengdu, China.

Her current research interests include deep learning, thermal analysis, power analysis, and thermal management of integrated circuit and building systems.

Ms. Guo was a recipient of the Best Paper Award nomination from Asia and South Pacific Design Automation Conference in 2019.

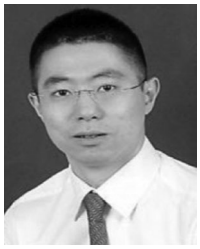
Sheldon X.-D. Tan (S'96–M'99–SM'06) received the B.S. and M.S. degrees in electrical engineering from Fudan University, Shanghai, China, in 1992 and 1995, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Iowa, Iowa City, IA, USA, in 1999.

He is a Professor with the Department of Electrical Engineering, University of California at Riverside, Riverside, CA, USA, where he also a Cooperative Faculty Member with the Department of Computer Science and Engineering. He was a Visiting Professor of Kyoto University, Kyoto, Japan, as a JSPS Fellow from 2017 to 2018. His current research interests include very large scale integration reliability modeling, optimization and management at circuit and system levels, hardware security, thermal modeling, optimization and dynamic thermal management for many-core processors, parallel computing, and adiabatic and Ising computing based on GPU and multicore systems. He has published over 290 technical papers and has coauthored 6 books in the above areas.

Dr. Tan was a recipient of the NSF CAREER Award in 2004, four Best Paper Awards from ICSICT'18, ASICON'17, ICCD'07, and DAC'09, and the Honorable Mention Best Paper Award from SMACD'18. He is serving as the TPC Vice Chair of ASPDAC 2019. He is currently serving as the Editor-in-Chief for *Integration, the VLSI Journal* (Elsevier), and an Associate Editor for two journals: *ACM Transaction on Design Automation of Electronic Systems* and *Microelectronics Reliability* (Elsevier).

Chi Zhang received the bachelor's degree from the Taiyuan University of Science and Technology, Taiyuan, China, in 1994, and the master's degree from the Microelectronics Research Institute, Chinese Academy of Sciences, Beijing, China, in 2003. He is currently pursuing the Ph.D. degree with the University of Electronic Science and technology of China, Chengdu, China.

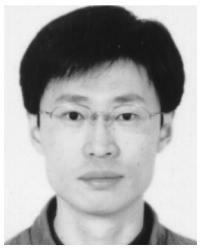
His current research interests include mixed-signal integrated circuit design, EDA technology, and multimode biometrics technology.



He Tang (M'09) received the B.S.E.E. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2005, the M.S. degree in electrical and computer engineering from the Illinois Institute of Technology, Chicago, IL, USA, in 2007, and the Ph.D. degree in electrical engineering from the University of California at Riverside, Riverside, CA, USA, in 2010.

From 2010 to 2012, he was with OmniVision Technologies Inc., Santa Clara, CA, USA, as an Analog IC Designer, where he researched on high-speed I/O interface. Since 2012, he has been an Associate Professor and, subsequently, a Professor with the University of Electronic Science and Technology of China. He has authored or coauthored over 40 papers. His current research interests include data converters and analog/mixed-signal IC designs. His past research includes high-speed high-resolution pipelined ADCs with digital calibration and high-performance ultralow-power SAR ADCs.

Dr. Tang has been serving on IEEE CAS Analog Signal Processing Technical Committee since 2013.



Yuan Yuan received the B.S. and M.S. degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 1992 and 2005, respectively.

He is currently an Associate Professor with the University of Electronic Science and Technology of China. He has published over 10 research papers in international conferences and journals. His current research interests include electronic measuring equipment design, computer-based measuring technology, and embedded system.