

Machine Learning based Electromigration-aware Scheduler for Multi-core Processors

Jagadeesh Kumar P, Mini M G
Department of Electronics Engineering
Government Model Engineering College
Thrikkakara, Cochin 21
India

Abstract—The rising performance demands in modern technology devices see the need to pack more functionality per area and are made possible with the advent of technology scaling. The extremely down-scaled, high-density processors used in such technology devices functioning at high frequencies and greater temperatures expedite various aging effects which impact the reliable lifetime of computing systems. Electromigration is considered to be an important intrinsic aging effect that reduces the useful lifetime of modern microprocessors. The objective of this work is to use machine learning methods to develop an electromigration-aware scheduler for assigning workloads to cores based on reliability and performance requirements. Aging estimation of the processor cores is performed based on the proposed computationally efficient and accurate regression-based thermal prediction models. According to experimental findings, the suggested technique can significantly extend the lifetime of multi-core architectures while allowing performance to degrade gracefully. The maximum error in the prediction of the lifetime of the cores using the proposed methodology is estimated to be 2.85%.

Keywords—*Electromigration aware scheduler; useful lifetime; multi-core processor reliability; machine learning model*

I. INTRODUCTION

The computing requirements of modern embedded systems in application areas including automotive, storage, networking, and 5G, among others, necessitate the use of high-performance processors. To deliver higher functionality per area demand for such applications, manufacturers create dense integrated multi-core processor chips operating at higher speeds [1]. Increased power density and operating temperature of processor cores can expedite aging phenomena including electromigration, lowering the processor's quality and reliability [2]. Runtime task execution management is a pressing research topic to minimize such aging effects in processor cores [2], [3], [4]. A rise of 10 - 15°C in the operating temperature could reduce the processor lifespan to half [5]. The lifetime of multi-core processors can be improved by estimating the aging effects of tasks that are ready for execution. The workload assignment to cores and frequency adjustments can be done at runtime to minimize the effect of aging of the cores. Such strategies need to be computationally precise and quick for real-time implementation.

Transistor aging is considered an important phenomenon that affects the reliability and lifetime of modern CMOS integrated circuits [6]. Time-Dependent Dielectric Breakdown

(TDDB), Hot Carrier Injection (HCI), and Negative Bias Temperature Instability (NBTI) are the key aging effects that challenge the reliable operation of modern integrated circuits. The intrinsic effects, Electromigration (EM) and Stress migration (SM), which are due to the interconnect aging, are also significant in present CMOS technology devices [7]. New methods for investigating and predicting degradation effects are important for enhancing the lifetime reliability of modern CMOS technology devices.

The suggested method to increase the multi-core processors' lifespan reliability is presented in two sections. In the first section, machine learning models based on regression to predict the processor cores' steady state temperature for the incoming jobs are proposed. A fine-grained approach is followed in this work for predicting the thermal properties of the cores' processing components. The fine-grained approach is more suitable for understanding the localized characteristics of the cores. In the second section, a scheme for mitigating the aging effects of processor cores by implementing a runtime frequency control policy is presented, which takes the temperature estimates of the proposed machine learning models as input. The operating speeds of the cores are decided to take into account the aging effects and the performance requirements. Our experimental results show that the proposed aging-aware scheduler, which is guided by the developed machine learning models could predict the core's lifetime with a maximum error of 2.85 %. The average error in temperature estimation is assessed, for each of the proposed models, and the maximum value is observed as 0.492 °C.

The rest of the paper's contents are arranged in the order mentioned below. A number of the significant efforts on thermal prediction and the regulation of thermal challenges of the processor cores are discussed in Section II. Linear regression and polynomial regression schemes for the estimation of the core temperature and the concept of the proposed aging-aware scheduler for enhancing processor core lifetime reliability are presented in Section III. An analysis of the results obtained for the proposed schemes are mentioned in Section IV. Finally, Section V concludes the work and mentions the possibilities for further research in the field.

II. LITERATURE REVIEW

The increasing performance demands of current technology gadgets have prompted various studies to concentrate on optimal job scheduling methods in multi-core systems

operating under real-time constraints. A significant body of published works focuses on techniques for a better performance-power trade-off of multi-core processors [8], [9], [10]. The effect of using different styles of coding for balancing between performance and energy consumption of the processing cores is presented in [11]. A core allocation technique to lower the energy usage of mobile devices by engaging the LITTLE cores to a maximum extent and ensuring the performance of the device is proposed in [12]. The execution time of complex programs can be minimized and thus the performance of the applications can be improved by the use of today's high-performance computing systems employing multi-core processors. Parallelization schemes [13] can be employed for converting the serial execution of programs into a hybrid parallel mode to take the advantage of the processing capability of multi-core processors. Reliability-aware scheduling techniques [14] can be used to reduce soft errors of heterogeneous chip-multiprocessors. The performance and power efficiency in heterogeneous multi-core CPUs can be enhanced with smart workload schedulers [15].

An aging-aware scheduler needs an accurate thermal estimate of the various logical components to make run-time decisions when a workload gets executed in a processor core. Several research works have looked into the feasibility of using model-based methodologies to determine the CPU core temperature characteristics [16], [17], [18]. An architectural level thermal behavioral modeling technique known as Thermsid [19] builds temperature models from the observed temperature and power statistics. Multiple scheduling schemes based on the efficient and simple thermal model are used [20] for managing the operations of homogeneous processor platforms. Thermal Estimation Accelerator (TEA) [21], a processing element level monitoring scheme for the temperature at runtime using hardware accelerators can serve as a benchmark for Dynamic Thermal Management (DTM) methods.

For the development of thermal models, it is required to estimate thermal profiles of the benchmark tasks executing on selected cores at defined operating frequencies. HotSpot [22] is a popular tool used in the temperature estimation of processor cores and is helpful in architectural studies. Various logical components in the processor architecture are represented as its equivalent thermal resistor and capacitor values along with the thermal package information [23]. Information about the floor design and power estimations for the logical components are supplied to HotSpot to determine the temperature profile. The power estimation of the logical components can be done with the tool McPAT (Multi-core Power, Area, and Timing) [24]. Based on high-level data, such as the frequency of operation of the core, McPAT can predict the architectural level power usage of the processor core containing caches and memory controllers. A McPAT-monolithic framework is presented in [25] for the architecture modeling of 3-D hybrid monolithic multi-core systems. The work in [26] proposes a micro-architectural framework to estimate the performance and energy consumption of cores in a multi-core processor. A detailed validation of McPAT's power models done with the help of a toolchain used in industrial practice is presented in [27]. In this study, McPAT is utilized to calculate the dynamic

power of the core's logical subsystems. McPAT requires the operation statistics of the applications and micro-architectural characteristics as its inputs to calculate the power consumption of every system component. The operation statistics of the tasks can be evaluated using the gem5 simulator [28]. CPU models with different types of memory configurations and cache organizations can be defined for the analysis. The current generation of widely used commercial Instruction Set Architectures (ISAs) including ARM and x86 are supported by gem5. A significant number of these simulators are utilized in the relevant fields of research since they are useful in analyzing the performance and power consumption of various processor models and can be used to validate various design options. A study of the basics of several multiprocessor simulation methodologies and a summary of the correctness of six architecture simulators including gem5 are presented in [29]. Gem5-X, a framework for system-level simulation based on gem5 [30] may be employed to assess the potential benefits of the architectural extensions for many image processing-related applications.

Computer architecture simulators can be used to closely examine the execution properties of applications that run in a processing core. Regression-based models with high computational efficiency and accuracy may well be constructed by relating the thermal figures estimated with HotSpot to the application characteristics estimated using gem5. Representative applications in open-source benchmark MiBench [31] can be used to develop and validate thermal models of typical workloads running in embedded processors. MiBench presents a collection of 35 embedded programs organized in six categories, each of which focuses on a specific segment of the embedded market. Workloads belonging to the application areas: network, security, telecommunication, and consumer from MiBench suite are used in this work. The characteristics of the tasks jpeg encode/decode, 32-bit Cyclic Redundancy Check (CRC), Dijkstra, and Secure Hash Algorithm (SHA), representing the mentioned application areas, running on x86 architecture-based processing cores are analyzed using gem5 and are used to train the regression models. The MiBench suite's consumer device benchmarks are designed to simulate the consumer device applications found in products including Personal Digital Assistants (PDAs), scanners, and digital cameras. This category largely focuses on image processing, and one of the representative image compression and decompression technologies is JPEG encoding/decoding. The telecommunications category of applications stands close to consumer applications because of the increased demand for consumer devices with wireless communication capability. Cyclic redundancy checks (CRC) are often used in data transmission for the detection of errors. A 32-bit cyclic redundancy check is performed on a sound file from the adaptive differential pulse code modulation benchmark as part of the CRC32 test.

Devices such as switches and routers have embedded applications that fall under the network category. Finding the shortest path through a graph is one of the methods used to illustrate the networking category. The Dijkstra benchmark determines the shortest route across each set of nodes of a graph by repeatedly applying Dijkstra's algorithm. In

applications related to e-commerce, data security is a key factor. Security applications frequently employ a variety of hashing, encryption, and decryption algorithms. The Secure Hash Algorithm (SHA) is frequently used to create digital signatures and transfer cryptographic keys in a secure manner. The secure hashing method, SHA benchmark in MiBench, generates a 160-bit code for an input.

Estimating the temperature and power at runtime using the tools HotSpot and McPAT is computationally expensive and limits their implementation in real-time schedulers. A regression-based model can be developed based on the thermal and power profiles of the workloads estimated using HotSpot and McPAT. Such trained and created regression-based models can quickly and accurately estimate the temperature of the logical components, and they are suitable for the successful application of aging-aware scheduling methods. A method for constructing a compact machine learning-based thermal prediction system appropriate for fast decision-making is presented in [32]. A temperature control strategy based on machine learning to determine the appropriate core frequency and encoder configuration for High-Efficiency Video Coding (HEVC) is proposed in [33]. A machine learning and simulation-based approach can be employed [34] to estimate the temperature map of a chip using the power consumption, utilization of the core, and recorded sensor temperatures. As discussed, a significant amount of the related research works published in recent years propose different approaches for determining the critical parameters associated with the workloads using software tools. Many of these works emphasize the increasing need for runtime techniques to regulate the processor temperature by changing its operating behavior. Such recent research works propose many techniques for better performance-power trade-offs. But, in our understanding, the use of scheduling techniques to reduce aging with the help of computationally efficient runtime temperature estimation methods for high-performance applications running on multi-core processors is a largely unexplored topic.

III. PROPOSED WORK

The first part of the proposed work deals with thermal profile modeling problems. Here, a regression-based model is employed to predict the steady-state temperature of the processing elements in a multi-core architecture. The temperature estimation model is driven by the properties of the workload. In the second part, an aging-aware scheduler is presented, whose scheduling activities are based on the thermal estimates of the regression model proposed in the first part of the work. The scheduler utilizes the thermal estimations and the performance need of the workloads taking into account the operating speed of the cores and trying to minimize the effect of processor aging.

A. Development of Regression-based Models

Regression analysis is one of the powerful multivariate statistical techniques to infer and form a functional relationship in a population [35]. In this work, regression analysis is used to relate the workload characteristics with the thermal effects of the processing elements. Workloads in embedded applications are suitable for implementing the suggested scheme because

they typically have high levels of predictability in their execution characteristics, such as the instruction execution behavior, memory operations, and the kind of information processing. The execution patterns of the MiBench applications running on a Hexa-deca homogenous multi-core architecture are analyzed using the gem5 simulator. The gem5 can be set up to operate in either Syscall Emulation (SE) mode or Full System (FS) mode. In SE mode, gem5 can imitate system calls made by applications. When configured in the FS mode, gem5 creates a bare-metal context for executing an operating system. In this work, gem5 is configured in the SE mode for analyzing the patterns of execution of the benchmark applications. The cores of the multi-core processor are selected as having x86 architecture. The cache hierarchy is defined to be of two levels, with level 1 private cache and level 2 shared cache. A subset j of the workload parameters is employed in this study, which directly affects the thermal profiles of the various logical components of the core. i.e., $j \in \{W\}$, where W represents the complete set of workload characteristics analyzed using gem5.

The above-mentioned workloads from the MiBench suite, which cover several embedded application areas, are used for model development and analysis. The selected applications are analyzed using the gem5 simulation tool and the characteristics are determined. The power usage of the various functional parts of the CPU architecture is calculated using McPAT. The thermal model of HotSpot is driven by the estimated power traces, and the chip and package characteristics. The configuration file specifies the parameters of the processor core for the HotSpot tool, which are shown in Table I.

Linear Regression (LR) and Polynomial Regression (PR) are the two regression models used in this work for estimating the thermal values of the functional elements of the CPU. In the Linear Regression (LR) model, the steady-state temperature of a functional element is represented as a weighted sum of the selected workload characteristics. In the LR model, the predicted temperature is represented as in (1).

$$\hat{y}(w, x) = w_1 x_1 + \dots + w_p x_p + b \quad (1)$$

TABLE I. HOTSPOT CONFIGURATION PARAMETERS

HotSpot Parameters	Value
Thickness of the chip (in meters)	0.00015
Specific heat of Silicon (in J/(m ³ -K))	1.75 x 10e6
Thermal conductivity of Silicon (in W/(m-K))	100.0
Resistance (Convection) (in K/W)	0.1
Capacitance (Convection) in J/K (Heat sink)	140.4
Thickness (Heatsink) (in meters)	0.0069
Heatsink side (in meters)	0.06
Thermal conductivity of Heatsink (in W/(m-K))	400
Specific heat (Heatsink) (in J/(m ³ -K))	3.55 x 10e6
Side (Heat spreader) (in meters)	0.03
Thickness (Heat spreader) (in meters)	0.001
Thermal conductivity (Heat spreader) in W/(m-K)	400

where $X = (x_1, x_2, \dots, x_p)$ represents the features used to train the models, w_1, w_2, \dots, w_p are the coefficients and b represents the bias. To reduce the sum of the squared estimate of errors between the measured values of the data, linear regression fits a linear model using weights $W = (w_1, w_2, \dots, w_p)$. The loss function which indicates the adequacy of the fit is given by (2).

$$L(\hat{y}, t) = \frac{1}{2}(\hat{y} - t)^2 \quad (2)$$

where \hat{y} , t , and $(\hat{y} - t)$ represent the prediction, target, and residual values respectively. The coefficients w_1, w_2, \dots, w_p and b are selected in a manner to reduce the loss function as represented in (3) and (4).

$$E(w_1, w_2, \dots, w_p, b) = \frac{1}{N} \sum_{i=1}^N L(y^i, t^i) \quad (3)$$

$$= \frac{1}{2N} \sum_{i=1}^N (\sum_j w_j x_j^i + b - t^i)^2 \quad (4)$$

In this study, the Python-based Scikit-learn machine learning package [36] is used, which is regarded as an effective and reliable tool for predictive data analysis.

In the development of the Polynomial Regression (PR) models of steady state temperature, polynomial regression needs to be performed on the data set, which are the workload characteristics, to fit a polynomial equation to it. This work extends linear regression by building polynomial features from the coefficients. For instance, the features in the second-order polynomials are utilized to fit a paraboloid to the data rather than a plane, giving the model represented in (5):

$$\hat{y}(w, x) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2 \quad (5)$$

The above model can be considered as a linear model creating a set of features given in (6).

$$z = [x_1, x_2, x_1 x_2, x_1^2, x_2^2] \quad (6)$$

This renaming of the data allows for the formulation of the problem as in (7).

$$\hat{y}(w, x) = w_0 + w_1 z_1 + w_2 z_2 + w_3 z_3 + w_4 z_4 + w_5 z_5 \quad (7)$$

The derived polynomial regression belongs to a similar category of linear models as those previously mentioned, which can be evaluated using the same methods. The model is flexible enough to accommodate a broader range of data by taking into account the linear fits in a higher-dimensional space constructed with these basis functions. To transform an input data matrix into a new data matrix of a specific degree, the polynomial properties converter in the Scikit-learn Python machine learning toolkit is used. The parameters of X have been converted from $[x_1, x_2, \dots]$ to $[x_1, x_2, x_1 x_2, x_1^2, x_2^2, \dots]$ and are now applicable to any linear model.

Linear and polynomial regression are used to model the temperature profiles of the processing units of the core. Fig. 1 represents the architecture of the processor core considered in this work. The models thus developed are used to predict the steady-state temperature of the various processing elements in the architecture. The regression models presented in this work are most appropriate for multi-core systems executing embedded tasks since their task characteristics are often highly predictable. The proposed model-based prediction logic is

computationally efficient and is more suitable for real-time thermal estimation. The estimated data can be used by an aging-aware scheduler to determine the best course of action for controlling the temperature below threshold levels while maintaining performance goals and extending the useful lifetime of processor cores.

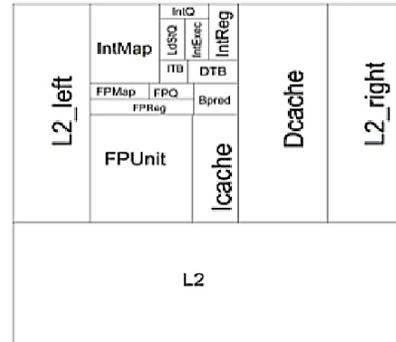


Fig. 1. Core Architecture.

B. The Aging-aware Scheduler Design

With the advancements in integrated circuit design and fabrication technology, more transistors can now fit into a square millimeter of a silicon wafer. When clocked at higher speeds to address the execution constraints, such densely integrated processors have increased power and heat dissipation, which has a negative impact on lifetime dependability. The major challenge is to develop algorithms that can forecast device-level degradation behavior based on the application features. This work presents a strategy for extending the useful lifetime of multi-core processors. A fine-grained approach is followed for estimating the aging effects of the various processing components of the multi-core processor. The data available for the recent industrial-grade embedded processors manufactured by Texas Instruments [37] is taken as the reference. Referring to [37], the operational lifespan of the semiconductor core is taken as ten years, when the junction temperature T_j is 105°C . The crucial factor affecting silicon lifespan is the junction temperature T_j when the circuit is functioning within the limits of voltage and frequency stated in the data sheet.

Due to continual operation at high temperatures, wear-out processes begin to develop in semiconductor products during their useful life. The wear-out processes commonly considered in the design of integrated circuits include Gate Oxide Integrity (GOI) [38], Electromigration (EM) [39], [40], and Time-Dependent Di-electric Breakdown (TDDB) [41]. Additionally, the lifespan of the present semiconductor devices is affected by processes such as Negative Bias Temperature Instability (NBTI) [42] and Channel Hot Carriers (CHC) [43]. Among these, electromigration is a major aging effect in present integrated circuits. The primary factor that influences electromigration is the junction temperature T_j . The junction temperature T_j is thus the critical factor affecting silicon lifespan under electrical bias when the chip is operating within the prescribed data sheet conditions, and the lifetime can be represented using an Acceleration Factor (AF). The Arrhenius equation, which links the chemical reaction rate to temperature,

can be used to analyze the damage that occurs in electronic devices over time for various working temperatures. The accelerating factor (AF) [37] can be represented as in (8).

$$AF = \exp\left(\frac{E_a}{K} \left(\frac{1}{T_{use}} - \frac{1}{T_{stress}}\right)\right) \quad (8)$$

where AF represents the Acceleration Factor, E_a is the Activation energy in eV, K is the Boltzmann's constant (8.63×10^{-5} eV/K), T_{use} is the use temperature in Kelvin and T_{stress} is the stress temperature in Kelvin.

This work proposes a methodology for improving the useful lifetime of the processor cores by considering electromigration as the primary failure mechanism. The aging-aware scheduler estimates the temperature of the processing elements of the core when a job is ready, by using the workload characteristics as input to the developed models. Our goal is to allocate the tasks to cores based on the aforementioned inputs to maximize chip lifetime while meeting the performance requirement bounds. For the reliability-aware scheduler design, two types of frequency adaptations are proposed in this work, where the processor clock can either be controlled in discrete values or in a continuous manner. Based on the workload characteristics, the performance requirement, and the end system reliability requirement, the aging-aware scheduler selects a core from the pool of feasible cores and decides its frequency of operation. The architecture of the proposed aging-aware scheduler is represented in Fig. 2.

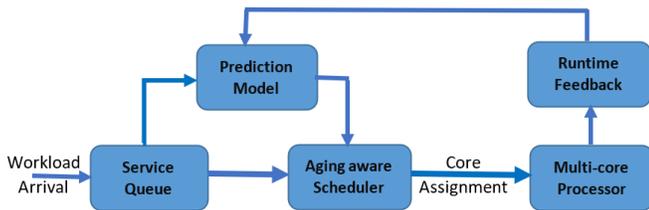


Fig. 2. The Aging-aware Scheduler.

The proposed aging-aware scheduler supports two types of clocking schemes for the processor core, i) cores whose operating frequencies can be selected from a discrete set of values and ii) cores whose frequency of operation can be varied continuously within a defined range. For the discrete frequency selection scheme, the scheduler will select the maximum possible operating frequency from a set of possible values, such that the lifetime reliability requirement can be met. In the continuous frequency selection scheme, the scheduler will have fine control of the operating frequency. In this case, the scheduler will use interpolate functions to estimate the closest frequency of operation that meets the lifetime reliability requirement with graceful performance degradation. For the specified workload characteristics, the data set includes known frequency and related temperature values of the logical components of the core.

Linear Interpolation (LI) and Spline Interpolation (SI) are the two types of interpolation schemes attempted in this work. To build a function utilizing fixed frequency datasets for linear interpolation, the `interp1d` class from the "scipy.interpolate" package is utilized. SciPy [44] is a free and open-source

Python library used for scientific and technical computing. When using linear interpolation for curve fitting, additional data sets are generated inside the boundaries of a finite collection of existing datasets using linear polynomials. In applications where smoothing is necessary, the piecewise-defined spline function is employed instead of polynomial interpolation because it produces good results for low-degree polynomials while minimizing Runge's phenomena for higher degrees.

Algorithm I illustrate the aging-aware scheduler's pseudo code. The scheduler, based on the workload characteristics, forms a feasible set of cores $\{C_1, \dots, C_m\}$, from the set of available cores during the scheduling interval. The temperature patterns of the logical components are predicted using the prediction models developed in the first part of this work, and using these parameters, the aging factor AF of each core C_i related to the lifetime reliability is determined. Based on the workload's performance requirements, the lifetime reliability requirement of the cores, and the type of frequency control supported by the architecture, i.e., either a discrete frequency control or fine frequency control, the frequency of operation of the core is determined.

Algorithm I. Aging Aware Core Selection

Inputs: workload characteristics, performance constraints, lifetime reliability requirement

1. **while** (true) {
2. **for** each schedule window T_s , perform {
3. **for** each task $\{T_1, T_2, \dots, T_n\}$ in the process queue Q perform {
- a. analyze the characteristics of T_i and form the feasible set of cores $\{C_1, \dots, C_m\}$.
- b. estimate temperature characteristics of the processing elements $\{L_1, \dots, L_p\}$ of the feasible cores C_i .
- c. estimate aging factor AF of each core C_i related to the lifetime reliability.
- d. select a core based on the lifetime reliability and performance requirements.
- e. determine the core's operating frequency f :
 - i. if discrete frequency control
 $f = f_c$ where $f_c \in \{f_1, \dots, f_s\}$
 - ii. else $f = f_i$ where $f_{min} \leq f_i \leq f_{max}$
4. **//end** for each task
5. **//end** for each schedule window
6. **//end** while

Outputs: i) Mapping of the tasks $\{T_1, \dots, T_n\}$ to cores $\{C_1, \dots, C_m\}$ if $n \leq m$; stall the remaining $n-m$ tasks if $n > m$, ii) frequency of operation of the selected cores.

The regression models for predictive modeling need to be updated if the error in prediction is more than a threshold value because of the change in the data. The accuracy of the prediction logic is verified periodically with an updating interval. The updating interval, a customizable parameter, is kept substantially greater than the scheduling interval to reduce the computational overhead for the assessment of the actual temperature levels. The model updating process is represented in Algorithm II.

Algorithm II. Model Update

Inputs: workload characteristics, core id, maximum allowable error in prediction (threshold)

1. **while** (true) {
2. **for** each updating interval, T_u do {
 - a. **for** each core $\{C_1, \dots, C_m\}$ do {
 - compute the actual temperature of the processing elements (with on-chip sensors and/or software tools).
 - b. **}//end** for each core
 - c. **if** (prediction error > threshold)
 - update the prediction models.
 - d. **end if**
3. **}//end** for each updating interval ...
4. **}//end** while

Output: updated prediction models.

IV. RESULTS AND DISCUSSION

The experiments of this research work were conducted using the application benchmarks belonging to consumer, telecommunications, network, and security categories taken from the well-known MiBench suite as represented in Table II.

Multiple instances created by altering the data set handled by the tasks are used to model the temperature characteristics of the logical components. As a result, n versions of a task w are created and evaluated its execution on m number of cores of the multi-core processor. Using new instances of the workloads derived from MiBench, the per-logical unit temperature is estimated to evaluate the developed models. The integer ALU, integer register file, floating-point unit, floating point register file, Data Translation Lookaside Buffer (DTLB), Instruction Translation Lookaside Buffer (ITLB), and load/store queue are characterized as the key power-consuming processing elements of the cores. Fig. 3 illustrates the validation of the developed models for the task CRC. The Steady State Temperature (SST) of the logical elements which are having significant power consumption is evaluated using the developed LR and PR models and is compared with the values estimated using the tool HotSpot. The operating frequency of the core is defined as 3400 MHz. The difference in estimation using the two methods is represented as a percentage error and is shown in Fig. 4. With a maximum prediction error of 0.008 percent for linear regression and 0.826 percent for polynomial regression-based models, the suggested regression-based models exhibit good consistency with HotSpot.

TABLE II. REPRESENTATIVE BENCHMARKS

Category	MiBench Benchmark
Consumer	JPEG encoding/decoding - (cjpeg /djpeg)
Telecom	Cyclic Redundancy Checks (CRC)
Network	Dijkstra
Security	Secure Hash Algorithm (SHA)

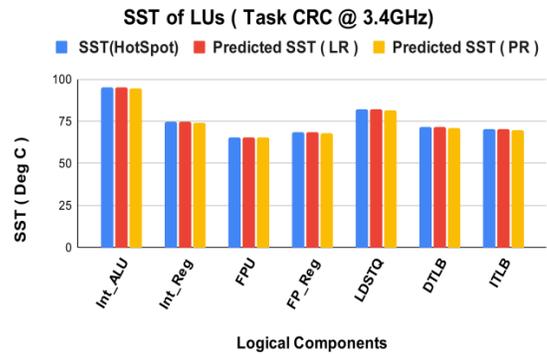


Fig. 3. SST of the Processing Elements Estimated using HotSpot, Linear Regression, and Polynomial Regression Models.

% error in the estimation of SST (Task CRC @ 3.4GHz)

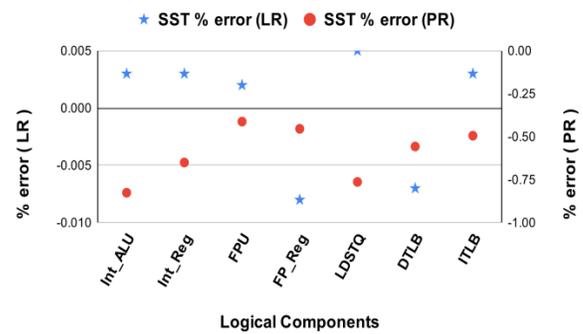


Fig. 4. Differences in the Estimation of SST of the Processing Elements are represented as a Percentage Error.

Fig. 5 illustrates the validation of the thermal models of the logical component integer ALU. The tasks used in the analysis are executed in cores set to operate at a clock frequency of 3400 MHz. The percentage error in the estimation of temperature is shown in Fig. 6. Simulation results show that the model is comparable to the HotSpot model in estimating the thermal profile of the logical components of the processor core.

SST of int_ALU for different tasks (core @ 3.4GHz)

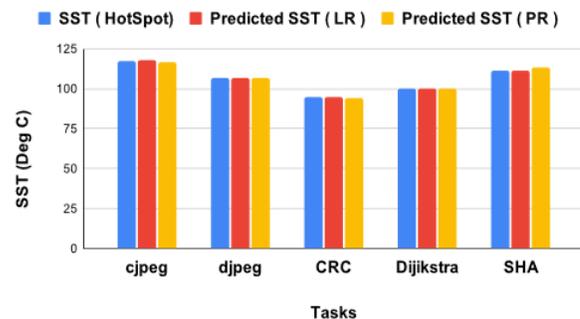


Fig. 5. Steady State Temperature of Integer ALU Estimated using HotSpot, Linear Regression, and Polynomial Regression Models.

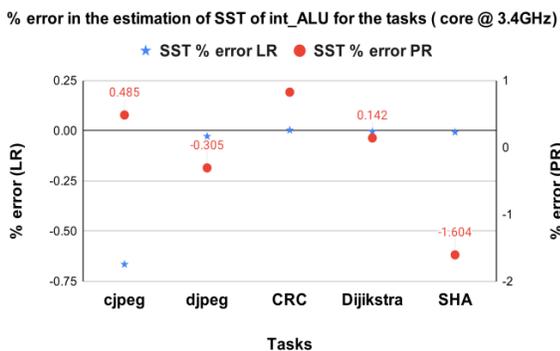


Fig. 6. Differences in the Estimation of Steady-state Temperature of Integer ALU.

The proposed aging-aware scheduler uses predicted SST of the logical components to estimate the degradation in the lifetime of the processor core during the scheduling of workloads. The lifetime of the critical components is represented using the Acceleration Factor (AF) while considering the junction temperature (T_j) of silicon as the primary variable impacting the lifetime of the cores. Fig. 7 shows the validation of AFs of the principal power-consuming processing elements of the core for the task djpeg where the AFs are computed using the SST values estimated using HotSpot, LR, and PR models. Fig. 8 represents the validation of the AF estimation of int_ALU for the different tasks. The processor core is set to operate at a frequency of 3400 MHz.

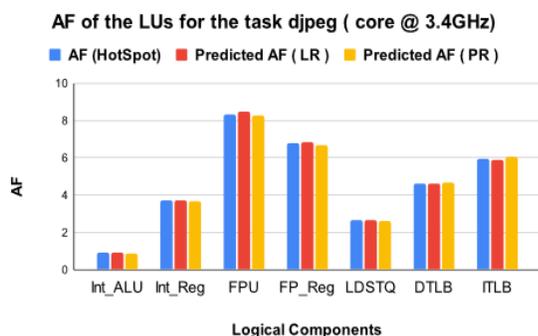


Fig. 7. Acceleration Factor of the Processing Elements Estimated using HotSpot, Linear Regression, and Polynomial Regression Models.

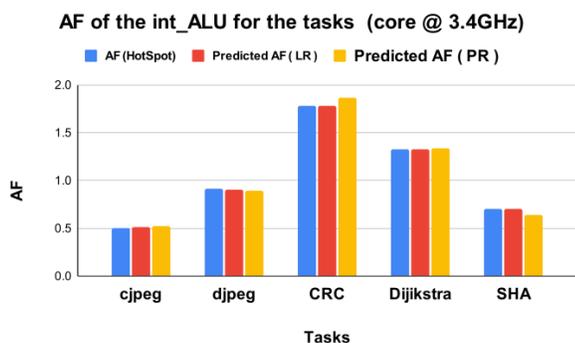


Fig. 8. Acceleration Factor of the Tasks Estimated using HotSpot, Linear Regression, and Polynomial Regression Models.

The proposed aging-aware scheduler estimates the temperature profiles of the logical components of the feasible cores based on the characteristics of the task in the service queue and decides the appropriate frequency of operation of the core. The lifetime reliability requirement of the core and the workload's performance requirements are used to determine the core operating frequency. In this work, the performance of the workloads running on a core clocked at 4000 MHz is taken as the reference, and the performance of the cores executing workloads with different operating conditions is represented relative to the reference performance. Fig. 9 represents the lifetime improvement of the cores and the corresponding relative performance degradation of the tasks when implementing a discrete frequency control scheme. In this case, the lifetime reliability requirement is taken as ten years, corresponding to an AF of 1. The core operating frequency is selected from the set {4GHz, 3.4GHz, 2.4GHz, 1.4GHz} based on the lifetime reliability requirement. Fig. 10 shows the corresponding values for AF = 1.5. The scheduler uses the thermal profile of the logical component having the highest value, for the estimation of AF of the core. This work assumes that the CPU core is designed to function for ten years when the junction temperature is at 105°C.

When the reliability-aware scheduler functions in the continuous frequency mode, the core operating frequency can assume a value within the defined range of 1400MHz to 4000 MHz. Linear Interpolation (LI) and Spline Interpolation (SI) are the two interpolation schemes employed, for estimating the closest frequency required for meeting the reliability and performance requirements. The frequency values for interpolation are determined based on the estimated temperature of the cores. Thermal estimation is performed using HotSpot, the standard method, and with the proposed LR and PR methods. The frequencies estimated by the scheduler for these temperature values are shown in Table III. The frequencies are determined for meeting the lifetime reliability requirement of ten years (corresponding AF = 1). Lifetime reliability of ten years corresponds to a threshold value of the silicon junction temperature, T_{qual} of 105 Degree Celsius.

Useful lifetime improvement and relative performance of tasks for AF = 1

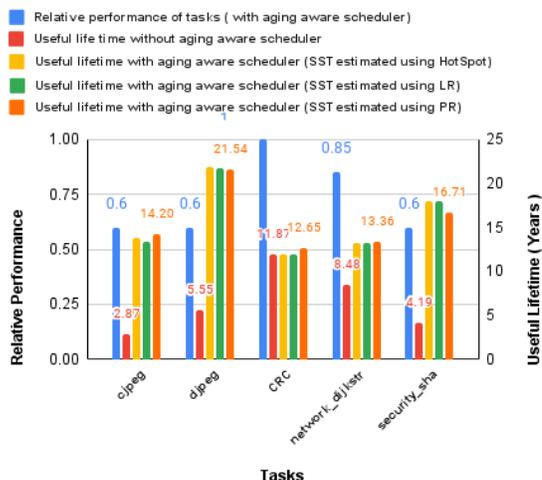


Fig. 9. Useful Lifetime Improvement and Relative Performance of Tasks (Discrete Frequency Control with AF = 1).

Useful lifetime improvement and relative performance of tasks for AF = 1.5

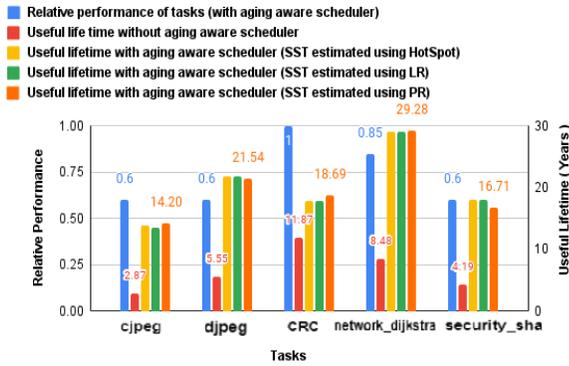


Fig. 10. Useful Lifetime Improvement and Relative Performance of Tasks (Discrete Frequency Control with AF = 1.5).

TABLE III. CORE OPERATING FREQUENCIES FOR AF=1

Fine Control of Operating Frequency (AF = 1 T _{qual} = 105 Degree Celsius)						
Case	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Temp Estimation scheme	Hot-Spot	Hot-Spot	LR	LR	PR	PR
Freq estimation scheme	LI	SI	LI	SI	LI	SI
Tasks	Estimated Frequencies of the cores (in GHz)					
cjpeg	2.7127	2.7128	2.7357	2.7356	2.7381	2.7377
djpeg	3.2871	3.2872	3.2852	3.2851	3.2663	3.2663
CRC	4.0	4.0	4.0	4.0	4.0	4.0
Dijkstra	3.7752	3.7752	3.7750	3.7749	3.7876	3.7876
SHA	3.0191	3.0195	3.0192	3.0191	2.9254	2.9253

Validation of the proposed scheme is carried out for the case where there is a lifetime reliability requirement of ten years. In this case, the aging-aware scheduler will adjust the operating frequency to limit the core temperature to 105°C. The actual temperature of the cores for the frequencies of operation mentioned in Table III is computed using HotSpot and is shown in Table IV. It can be seen that the proposed algorithm is adjusting the core frequencies in such a way that the temperature of operation of the cores is very close to the required value of 105°C. The average prediction error of the proposed scheme in the estimation of the core temperature is compared with the results reported in recent publications and is shown in Table V.

The lifetime of the cores, when operating with the temperatures shown in Table IV, is computed using (8). The theoretical lifetime corresponding to AF=1 is ten years. Table VI shows the lifetime of the cores corresponding to the steady state temperature values mentioned in Table IV. In this case, the algorithm is driving the operating frequency of the cores in such a way as to meet the lifetime requirement of ten years. The lifetime of the cores when operating with the frequencies estimated by the algorithm is having a maximum deviation of 2.85 % from the required lifetime.

TABLE IV. VALIDATION OF THE CORE TEMPERATURES FOR AF=1

Tasks	Temperature of Cores (in Degree Celsius) - Validation					
	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
cjpeg	104.99	104.99	105.4	105.4	105.46	105.44
djpeg	105.01	105.01	104.98	104.98	104.69	104.69
CRC	105.0	105.0	105.0	105.0	105.0	105.0
Dijkstra	105.0	105.0	104.98	104.98	105.15	105.15
SHA	105.0	105.0	105.0	105.0	103.46	103.46

TABLE V. COMPARISON OF THE PREDICTION ERROR OF CORE TEMPERATURE WITH PROPOSED METHODS IN THE LITERATURE

Scenarios	Average Prediction Error
Case 1	0.004 °C
Case 2	0.004 °C
Case 3	0.088 °C
Case 4	0.088 °C
Case 5	0.492 °C
Case 6	0.488 °C
Alzemiro et al. ^[21]	0.020 °C
Kaicheng Zhang et al. ^[32]	2.900 °C
Carlton Knox et al. ^[34]	1.390 °C

TABLE VI. THE ESTIMATED LIFETIME OF THE CORES

Tasks	Estimated Lifetime of Cores (in years) for AF=1					
	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
cjpeg	10.006	10.006	9.776	9.776	9.7426	9.7426
djpeg	9.9943	9.9943	10.011	10.011	10.178	10.178
CRC	10.0	10.0	10.0	10.0	10.0	10.0
Dijkstra	10.0	10.0	10.011	10.011	9.9152	9.9152
SHA	10.0	10.0	10.0	10.0	10.917	10.917

Using the methodology proposed, embedded application developers can perform a fast design space exploration between the lifetime reliability of the processor cores and the performance requirement of the tasks. Fig. 11 depicts the compromise between AF and the performance of the benchmark applications when linear regression is employed for temperature estimation along with linear interpolation for frequency estimation. At higher values of AF, processing cores have better lifetime reliability but at the expense of application execution performance. Fig. 12 represents the corresponding trade-off when polynomial regression is employed with continuous frequency assignment.

The concept suggested in this paper, where the aging-aware scheduler selects a task from a predetermined list of tasks to assign to a core, works well enough to extend the lifespan of multi-core systems running embedded workloads. A task might run at multiple scheduling points with a varying level of computational load because the complexity of the jobs that get executed on the cores may change over time. It is possible to account for these diverse computational costs at various execution times by building the regression model utilizing the heat profiles of logical units carrying out activities of varied computational costs at various execution times.

REFERENCES

[1] V. Rajaraman, "Multi-core microprocessors," Resonance 22, no. 12, pp. 1175–1192, 2017, <https://doi.org/10.1007/s12045-017-0580-0>.

[2] B. Wang, "Task Parallel Scheduling over Multi-core System," in M. G. Jaatun, G. Zhao, C. Rong, (eds), Cloud Computing, CloudCom 2009, Lecture Notes in Computer Science, vol 5931, Springer, Berlin, Heidelberg, https://doi.org/10.1007/978-3-642-10665-1_39.

[3] S. Deniziak and A. Dzitkowski, "Scheduling of Distributed Algorithms for Low Power Embedded Systems," International Journal of Advanced Computer Science and Applications (IJACSA), 7(12), 2016. <http://dx.doi.org/10.14569/IJACSA.2016.071205>.

[4] B. Ranjbar, T. D. A. Nguyen, A. Ejlali, and A. Kumar, "Power-Aware Run-Time Scheduler for Mixed-Criticality Systems on Multi-Core Platform," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, pp. 2009–2023, no. 10, 2020.

[5] R. Viswanath, V. Wakharkar, A. Watwe, and V. Lebonheur, "Thermal Performance Challenges from Silicon to Systems", Intel Technology Journal, Q3 2000, pp. 1–16.

[6] I. Hill, P. Chanawala, R. Singh, S. A. Sheikholeslam, and A. Ivanov, "CMOS Reliability From Past to Future: A Survey of Requirements, Trends, and Prediction Methods," in IEEE Transactions on Device and Materials Reliability, vol. 22, no. 1, pp. 1-18, March 2022, <https://doi.org/10.1109/TDMR.2021.3131345>.

[7] M. D. Shroff and A. L. Loke, "Design-technology co-optimization for reliability and quality in advanced nodes," Proc. SPIE 11614, in Design-Process-Technology Co-optimization XV, 1161403, February 2021, <https://doi.org/10.1117/12.2585220>.

[8] X. Yao, P. Geng, and X. Du, "A Task Scheduling Algorithm for Multi-core Processors," 2013 International Conference on Parallel and Distributed Computing, Applications and Technologies, pp. 259–264, 2013, <https://doi.org/10.1109/PDCAT.2013.47>.

[9] T. Zhang, X. Pan, W. Shu, and M. Y. Wu, "Asymmetry-Aware Scheduling in Heterogeneous Multi-core Architectures," in C. H. Hsu, X. Li, X. Shi, R. Zheng, (eds), Network and Parallel Computing, NPC 2013, Lecture Notes in Computer Science, vol 8147, Springer, Berlin, Heidelberg, https://doi.org/10.1007/978-3-642-40820-5_22.

[10] S. Holmbacka and J. Keller, "Workload type-aware scheduling on big.LITTLE platforms," in International Conference on Algorithms and Architectures for Parallel Processing, Springer, Cham, pp. 3-17, 2017, https://doi.org/10.1007/978-3-319-65482-9_1.

[11] H. H. Hassan, A. S. Moussa, and I. Farag, "Performance vs. Power and Energy Consumption: Impact of Coding Style and Compiler," International Journal of Advanced Computer Science and Applications IJACSA, 8(12), 2017, <http://dx.doi.org/10.14569/IJACSA.2017.081217>.

[12] D. Kim, Y. Ko, and S. Lim, "Energy-Efficient Real-Time Multi-Core Assignment Scheme for Asymmetric Multi-Core Mobile Devices," in IEEE Access, vol. 8, pp. 117324–117334, 2020, <https://doi.org/10.1109/ACCESS.2020.3005235>.

[13] A. Algarni, A. Alofi, and F. Eassa, "Parallelization Technique using Hybrid Programming Model," International Journal of Advanced Computer Science and Applications (IJACSA), 12(2), 2021. <http://dx.doi.org/10.14569/IJACSA.2021.0120285>.

[14] A. Naithani, S. Eyerman, and L. Eeckhout, "Reliability-Aware Scheduling on Heterogeneous Multicore Processors," 2017 IEEE International Symposium on High Performance Computer Architecture (HPCA), 2017, pp. 397–408, <https://doi.org/10.1109/HPCA.2017.12>.

[15] S. I. Kim and J. Kim, "A Method to Construct Task Scheduling Algorithms for Heterogeneous Multi-Core Systems," in IEEE Access, vol. 7, pp. 142640–142651, 2019, <https://doi.org/10.1109/ACCESS.2019.2944238>.

[16] Guoping Xu, "Thermal Modeling of Multi-Core Processors," Thermal and Thermomechanical Proceedings, 10th Intersociety Conference on Phenomena in Electronics Systems, 2006, IThERM 2006, pp. 96–100, <https://doi.org/10.1109/ITHERM.2006.1645327>.

[17] D. Jaeckle and A. Sikora, "Thermal modeling of homogeneous embedded multi-core processors," 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 588–593, 2014, <https://doi.org/10.1109/ICACCI.2014.6968448>.

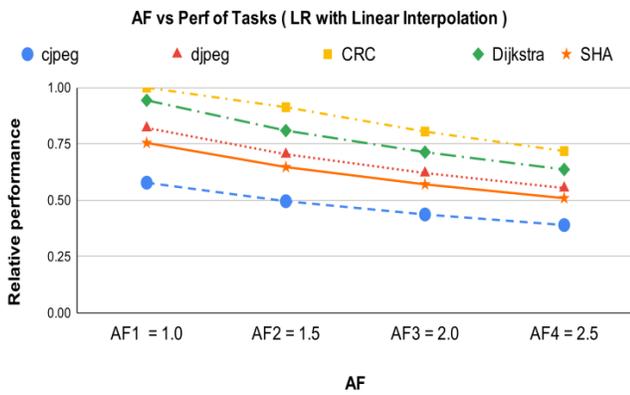


Fig. 11. Acceleration Factor (AF) - Performance Tradeoff (Linear Regression with Linear Interpolation).

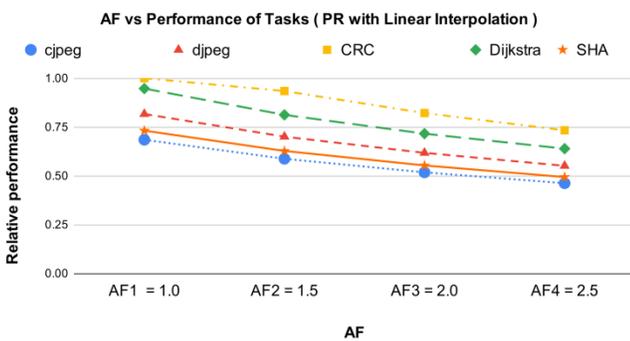


Fig. 12. Acceleration Factor (AF) - Performance Tradeoff (Polynomial Regression with Linear Interpolation).

V. CONCLUSION AND FUTURE WORK

The aging-aware scheduler proposed in this work uses the developed computationally efficient models to estimate the steady-state temperature of the processing elements in multi-core processor architecture. Temperature values estimated with the models are used to predict electromigration-induced aging. The scheduler performs an aging-aware application mapping strategy for enhancing the lifetime reliability of the cores. The suggested scheduler will estimate the operating frequency of the processing cores for satisfying the lifetime reliability constraints with a gentle decline of the performance as opposed to a no-aging aware scheduler, where the workloads are distributed to the cores based on the performance need. Results from simulations show that the suggested approach can increase the lifespan of the operation of multi-core processor systems.

The algorithm proposed in this work is extensible and configurable. The proposed framework is configurable, as it is possible to use on-chip thermal sensor data for estimating the temperature and aging effects of the logical components along with the temperature data computed using the software tools. In the future, the framework may be extended to take into account the impacts of aging brought on by Hot Carrier Injection (HCI), Positive-Bias Temperature Instability (PBTI), and Negative-Bias Temperature Instability (NBTI), along with electromigration.

- [18] I. Takouna, W. Dawoud, and C. Meinel, "Accurate Multicore Processor Power Models for Power-Aware Resource Management," 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, pp. 419-426, 2011, <https://doi.org/10.1109/DASC.2011.85>.
- [19] T. J. A. Eguia, R. Shen, S. X. D. Tan, E. H. Pacheco, and M. Tirumala, "Architecture level thermal modeling for multi-core systems using subspace system method," IEEE 8th International Conference on ASIC, pp. 714-717, 2009, <https://doi.org/10.1109/ASICON.2009.5351305>.
- [20] J. P. Rodríguez and P. M. Yomi, "Work-in-Progress: Towards a fine-grain thermal model for uniform multi-core processors," 2020 IEEE Real-Time Systems Symposium (RTSS), pp. 403-406, 2020, <https://doi.org/10.1109/RTSS49844.2020.00049>.
- [21] A. L. da Silva, A. L. D. M. Martins, and F. G. Moraes, "Fine-grain temperature monitoring for many-core systems," in Proceedings of the 32nd Symposium on Integrated Circuits and Systems Design, pp.1-6, 2019, <https://doi.org/10.1145/3338852.3339841>.
- [22] R. Zhang, M. R. Stan, and K. Skadeon, "HotSpot6.0: Validation Acceleration and Extension," Tech. Report CS-2015-04.
- [23] M. R. Stan, K. Skadron, M. Barcella, W. Huang, K. Sankaranarayanan, et al., "HotSpot: a dynamic compact thermal model at the processor architecture level," Microelectronics Journal, Dec 2003, 34, (12), pp. 1153-1165, [https://doi.org/10.1016/S0026-2692\(03\)00206-4](https://doi.org/10.1016/S0026-2692(03)00206-4).
- [24] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, et al., "McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures," 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 12-16 Dec 2009, pp.469-480, <https://doi.org/10.1145/1669112.1669172>.
- [25] A. Guler and N. K. Jha, "McPAT-Monolithic: An Area/Power/Timing Architecture Modeling Framework for 3-D Hybrid Monolithic Multicore Systems," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, no. 10, pp. 2146-2156, Oct. 2020, <https://doi.org/10.1109/TVLSI.2020.3002723>.
- [26] F. A. Endo, D. Couroussé, and H. Charles, "Micro-architectural simulation of embedded core heterogeneity with gem5 and McPAT," Proceedings of the 2015 Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools, RAPIDO '15, Amsterdam, Holland, Jan 2015, Article No.7.
- [27] S. L. Xi, H. Jacobson, P. Bose, G. Wei, and D. C. Brooks, "Quantifying sources of error in McPAT and potential impacts on architectural studies", IEEE 21st International Symposium on High Performance Computer Architecture (HPCA), CA, USA, Feb 2015, pp.577-589.
- [28] N. L. Binkert, B. M. Beckmann, G. Black, S. K. Reinhardt, A. G. Sadi, et al., "The gem5 Simulator," ACM SIGARCH Computer Architecture News, May 2011, 39, (2), pp. 1-7, <https://doi.org/10.1145/2024716.2024718>.
- [29] A. Akram and L. Sawalha, "A Survey of Computer Architecture Simulation Techniques and Tools," in IEEE Access, vol. 7, pp. 78120-78145, 2019, <https://doi.org/10.1109/ACCESS.2019.2917698>.
- [30] Y. M. Qureshi, W. A. Simon, M. Zapater, D. Atienza, and K. Olcoz, "Gem5-X: A Gem5-Based System Level Simulation Framework to Optimize Many-Core Platforms," 2019 Spring Simulation Conference, pp.1-12, 2019, <https://doi.org/10.23919/SpringSim.2019.8732862>.
- [31] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, et al., "MiBench: A free, commercially representative embedded benchmark suite," Proceedings of the Fourth Annual IEEE International Workshop on Workload Characterization. WWC-4 (Cat. No.01EX538), 2001, pp. 3-14, <https://doi.org/10.1109/WWC.2001.990739>.
- [32] K. Zhang, A. Guliani, S. O. Memik, G. Memik, K. Yoshii, et al., "Machine Learning-Based Temperature Prediction for Runtime Thermal Management Across System Components," in IEEE Transactions on Parallel and Distributed Systems, vol. 29, no. 2, pp. 405-419, 2018, <https://doi.org/10.1109/TPDS.2017.2732951>.
- [33] A. Iranfar, M. Zapater, and D. Atienza, "Work-in-progress: a machine learning-based approach for power and thermal management of next-generation video coding on MPSoCs," 2017 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), pp. 1-2, 2017, <https://doi.org/10.1145/3125502.3125533>.
- [34] C. Knox, Z. Yuan, and A. K. Coskun, "Machine Learning and Simulation Based Temperature Prediction on High-performance Processors," in Proceedings of ASME International Technical Conference and Exhibition on Packaging and Integration of Electronic and Photonic Microsystems (InterPACK), July 2022.
- [35] M. Mahbobi and T. K. Tiemann, "Introductory Business Statistics with Interactive Spreadsheets," 1st Canadian Edition, BCcampus, December 7, 2015, ch.8, Ebook ISBN 978-1-77420-007-0.
- [36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, et al., "Scikit-learn: Machine Learning in Python," JMLR 12, pp. 2825-2830, 2011.
- [37] Allan Webber, "Calculating Useful Lifetimes of Embedded Processors," Texas Instruments Application Report, SPRABX4B, November 2014 Revised March 2020.
- [38] F. Bonoli, P. Godio, G. Borionetti, and R. Falster, "Gate oxide integrity dependence on substrate characteristics and SiO2 thickness," Materials Science in Semiconductor Processing, Volume 4, Issues 1-3, 2001, pp. 145-148, ISSN 1369-8001, [https://doi.org/10.1016/S1369-8001\(00\)00152-9](https://doi.org/10.1016/S1369-8001(00)00152-9).
- [39] J. R. Black, "Electromigration - A brief survey and some recent results," in IEEE Transactions on Electron Devices, vol. 16, no. 4, pp. 338-347, April 1969, <https://doi.org/10.1109/T-ED.1969.16754>.
- [40] D. G. Pierce and P. G. Brusius, "Electromigration: A review, Microelectronics Reliability," Volume 37, Issue 7, 1997, Pages 1053-1072,ISSN 0026-2714, [https://doi.org/10.1016/S0026-2714\(96\)00268-5](https://doi.org/10.1016/S0026-2714(96)00268-5).
- [41] J. F. Verweij and J. H. Klootwijk, "Dielectric breakdown: A review of oxide breakdown," Microelectronics Journal, Volume 27, Issue 7, 1996, pp. 611-622, ISSN 0026-2692, [https://doi.org/10.1016/0026-2692\(95\)00104-2](https://doi.org/10.1016/0026-2692(95)00104-2).
- [42] M. A. Alam and S. Mahapatra, "A comprehensive model of PMOS NBTI degradation," Microelectronics Reliability, 45:71-81, Aug. 2005.
- [43] Eiji Takeda, "Hot-carrier effects in scaled MOS devices," Microelectronics Reliability, Volume 33, Issues 11-12, 1993, Pages 1687-1711, ISSN 0026-2714, [https://doi.org/10.1016/0026-2714\(93\)90081-9](https://doi.org/10.1016/0026-2714(93)90081-9).
- [44] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, et al., "SciPy 1.0: fundamental algorithms for scientific computing in Python," Nat Methods 17, 261-272, 2020, <https://doi.org/10.1038/s41592-019-0686-2>.