Cross-Layer Virtual Observers for Embedded Multiprocessor System-On-Chip (MPSoC)

Santanu Sarma, Nikil Dutt, and Nalini Venkatasubramanian

Abstract—**There is a growing concern about the increasing vulnerability of future computing systems to errors in the underlying hardware due to manufacturing process variability, exponentially increasing power dissipation and heating, as well as drastic and harsh environments such systems may have to operate in. This research proposes the concept of cross-layer virtual observers and actuations with the aim of achieving improved reliability, performance, thermal stability, and reduced power and energy consumption applied across different layers of system stack. Cross-layer resilient systems, which distribute the responsibility for tolerating errors, device variation, and aging across the system stack, have the potential to provide the resilience required to implement reliable, high-performance, low-power systems in future fabrication processes at signicantly lower cost. By using redundant, complementary, or more timely information from multiple sensors at different layers, virtual observers can provide more reliable and accurate information, specific inferences, context, and conditions as well as accurate assessment of the surrounding environment, while identifying malfunction and dangers (e.g thermal overheating or hotspot) in diverse kind of system including emerging Cyber-physical and Multiprocessor system-on-chips (MPSoCs). Virtual observer enabled self-awareness allows a system to observe it's own internal behaviors as well as external systems it interacts with such that it is capable of making judicious decision to optimize performance and other quality of service (QoS) metrics. With the ability to discover potential present action and predict future actions as well as evaluate past actions and behaviors, these computer systems will be capable of adapting their behavior and resources to automatically find the best way to accomplish a given goal despite changing environmental conditions and demands. We demonstrate the effectiveness and applicability of these concepts specifically overcoming the vulnerabilities introduced by faults and process variability using two case studies: one using virtual observer to estimate temperature of unmeasured core and the other to predict the failure rate of the unmeasured core using the estimated temperature using concepts drawn from embedded multiprocessor systems on a single chip (MPSoC) respectively.**

Index Terms—**Reflective Middleware, Cross-Layer Design Approach, Observers, Estimators, Multi-Processor Systems-on-Chip, Cyber Physical Systems.**

I. Introduction

FUTURE computing systems, from Warehouse size computers to many/multi-core processors, shares a growing concern about the increasing vulnerability to errors in the underlying hardware due to manufacturing process variations [3], [12]and other types of faults[2] at various levels of abstraction. To understand and characterize these effects, risk, as well as dangers, extremely expensive test structures or sensors must be deployed (both in temporal and spatial dimensions) at multiple points. However, due to severe resource constraints, placement restrictions, complex machinery, and inaccessibility, often physical deployment of sensors and test structures to characterize and validate system behaviors, properties and parameters (e.g. fault behaviors, manufacturing process variation parameters), is often not possible with in an reasonable cost [20].

In this paper, we argue/conjecture that novel approaches are required to support scalable design and reliable execution of multicore embedded MPSoC systems. Specifically, we aim to show how a reflective approach enables resilient systems at both the infrastructure and application levels for multicore embedded systems Our proposed design philosophy has two key elements:

- Cross-layer design of embedded MPSoC systems that encompasses needs and constraints from multiple levels of the execution and system stack
- A reflective (observe-analyze-adapt) architecture embodied via closed loop control cycle in the system that observes application and the underlying complex embedded architecture,projects potential future errors and adapts the system to meet application needs and architectural constraints.

We propose such a design framework with cross-layer observations and actuations that enable adaptation across different layers of the system stack **with the aim of achieving improved reliability, performance, thermal stability, and reduced power and energy consumption**. We illustrate the value of the proposed design philosophy via a detailed case study from emerging nanoscale multicore processor system-on- chips (MPSoCs) executing over a broad range of synthetic workloads. We demonstrate how the ability to adapt execution at various layers can overcome the vulnerabilities introduced by faults and process variability activate efficient operating points and consequently more resilient system execution.

Cross-layer Design: As discussed earlier, accurate parameter observation at system levels is expensive and inefficient. We propose techniques to physically and virtually observe a range of immeasurable parameters at different points in the system stack. A key novelty of our approach is that we aim to develop techniques to expand limited physical observations by using software "sensors" as opposed to a pure physical or hardware based approach. We believe that such hardware-software techniques for observation is a powerful solution that combines information from a set of homogenous or heterogenous sensors to compute an abstract measurement placed at same or different abstract layers (cross-layers) of the system stack. A key benefit of such an approach is to enable individual system stacks being made self-aware, an ability to observer their own behavior and make judicious decisions, while optimize performance and other quality of service (QoS) metrics by evaluating present beneficial actions and discovering or predicting potential future actions. These ability enables the systems to adapt their behavior and resources to automatically find the best way to accomplish a given goal despite changing environmental conditions and demands. It also enables the distribution of responsibility for tolerating errors, device variation, and aging across the system stack, with the potential to provide the resilience required to implement reliable, high-performance, low-power systems in future fabrication processes at signicantly lower cost.

Reflective Middleware Approach: We argue that the middleware layer is the appropriate level at which to implement the cross-layer architecture due to its ability to interface with intra-system components (onboard processors, sensors, memory, storage, network)

The authors are with the Department of Computer Science, University of California, Irvine, USA, e-mail: santanus, dutt, nalini@ics.uci.edu.

ARM '12, Montreal, Quebec, Canada.

as well as devices, networks and applications external to the MPSoC in the overall distributed ecosystem. The designed middleware must be capable of observing multiple factors and adapting system execution intelligently to address the reliability needs and challenges. Along these lines, we explore the design of a Reflective Middleware Architecture that enables

- efficient and effective observation of architectural factors (thermal properties, power consumption) and variabilities through limited onboard sensing, observation of application activity behavior
- analysis of observed cross-layer features to estimate projected system and application behavior both spatially (across the embedded multicore platform) and temporally.
- software controlled adaptation of multiple layers to ensure real-time needs of applications while optimizing the thermal/power distributions in the underlying MPSoC .

In other words, the reflective middleware layer observes cross-layer features of the multicore platform, projects/analyses the current behavior to determine future (potentially problematic) situations and adapts multiple system layers to improve overall resilience and performance of the execution. To lend focus to our work, the case study in this paper focuses on the intra-system observation and adaptation (within the MPSoC); however, we see much potential for our approach as the number of devices, networks and applications scale. This is indeed the direction of our current and future work.

II. A Reflective Cross-Layer Design Framework for Embedded MPSoC

Computational reflection embodies the principles of self-awareness, i.e. the ability of the system to observe it's own internal behaviors as well as external systems it interacts with such that it is capable of making judicious decision that optimize performance and other quality of service (QoS) metrics (e.g., energy and power consumption, temperature, system reliability and resilience etc.). The concept of self-awareness is not new to hardware; it has been extensively developed in classical adaptive control systems [15] as well as variants of autonomous computing [9], invasive computing [18], evolvable and adaptive hardware [14] in a cross-layer design framework.

The cross-layer design paradigm proposed in this research is based on the classical control centric notion of feedback loops - the observe-decide-act nature of this loop meshes with with the reflective middleware approach described earlier. In the following section, we illustrate the importance of cross-layer observations, how to obtain such observations in a cost-effective manner and use them to design adaptations (or interventions) to support resilient execution of applications in embedded MPSoC platforms.

A. Efficient Cross-Layer Virtual Observation

As discussed earlier, accurate hardware-based monitoring of each parameter of interest in a cross-layer systems (e.g. temperature on each core of a multicore system, failure rate of each architectural component) is expensive and sometimes infeasible. Redundant, complementary, or more timely information from multiple sensors can provide more reliable and accurate information, specific inferences, context, and conditions as well as accurate assessment of the surrounding environment, threats and dangers including malfunction.

In our approach, effective observation is achieved through a hierarchical composition of hardware and software entities at multiple layers; this approach allows us to extrapolate physical measurements from one region to corresponding values in another region; it allows us to calculate immeasurable parameters using software functions. Such virtual observation entities, aka virtual observers, provide indirect measurement of abstract conditions, contexts, inferences or estimates by processing (e.g. combining, aggregating, or predicting) sensed data from either a set of homogeneous or heterogenous sensors. A virtual observer is specified by a set of input data types, a generic processing (or an aggregating function), set of output data types obtained by processing the inputs, and the processing rate or frequency of aggregation operation. We apply the notion of virtual observation at different layers of system stack of computing system as shown in Fig. 1.

One key advantage of the virtual observation technique using the cross-layer approach is that the most common source of sensing errors due to sensor calibration can be reduced or intelligently eliminated by consensus among several sensors. In addition to masking or virtualizing the data sources to the intended users [13], the virtual observers provide added advantage of reusability across different platforms and cores. In most general cases, different techniques such as estimation (e.g. non-parametric Weiner filter, parametric recursive least square RLS , Kalman filter), feature extractions and classifications (e.g. k-means clustering), inference methods, as well as artificial intelligence techniques including neural networks and fuzzy schemes can be adopted in the design and composition of the virtual sensors. The combination of these techniques that are often adopted to fuse information from different sensors are referred to as sensor fusion in literature [11] and is depicted in Fig. 2. A specific example illustrating virtual observation principle is shown in Fig.3 and Fig. 4. We illustrate the the concept with concrete example in section III.

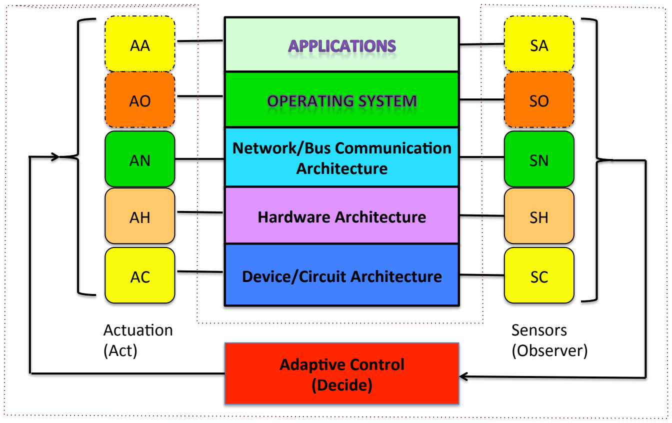


Figure 1. Cross-layer virtual sensing and actuation at different layers of system stack. Dotted line portion indicates reflective middleware.

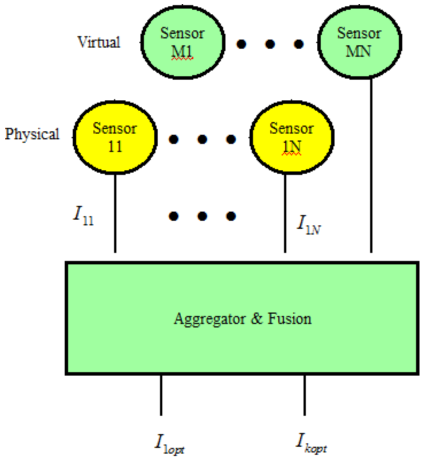


Figure 2. Virtual observer data fusion.

B. Exploiting Cross-Layer Virtual Observation to Achieve Cross-Layer Resilience

There is a growing concern about the increasing vulnerability of future computing systems to errors in the underlying hardware.

Table I
EXAMPLES OF VIRTUAL/PHYSICAL SENSING AND ACTUATIONS AT DIFFERENT LAYERS OF SYSTEM STACK

Layer	Virtual Observation/Sensor	Type	Virtual/Actuation	Type
Device/Circuit	Temperature; Circuit Delay; Aging	SC	VFS, DVFS, Adaptive Body Biasing, Multi-gate thresholds	AC
Hardware Architecture	Cache Misses; IPC; CPI; MLP	SH	Cache Sizing; Reconfiguration	AH
Network/Bus Communication	Bandwidth; Packet Utility; Congestion	SN	Adaptive Routing; Dynamic Bandwidth Allocation	AN
Operating System	System Utilization, Peripheral States	SO	Task Allocation/Migration; Scheduling	AO
Application	Workload; Power Consumption	SA	Loop perforation; Algorithmic choice; Obfuscation	AA

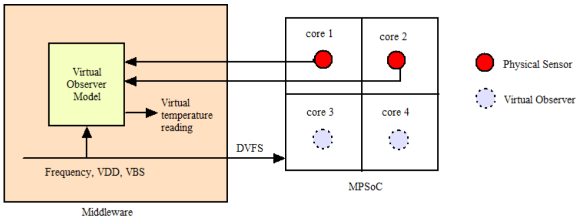


Figure 3. Example of virtual observer concept where two sensors are physically placed (red) and two are virtually constructed (using observers or estimators).

Traditional hardware redundancy techniques are expensive for designing energy-efficient systems that are resilient to high error rates. Implementing reliability within a single layer can simplify system architecture and design; however, this severely limits opportunities for optimized operation of the overall system. In contrast, cross-layer resilient systems, which distribute the responsibility for tolerating errors, device variation, and aging across the system stack, have the potential to provide the resilience required to implement reliable, high-performance, low-power systems in future fabrication processes at signicantly lower cost [4]. These systems can implement less-frequent resilience tasks in software to save power and chip area, can tune their reliability guarantees to the need of applications, and can use the information available at each level in the system stack to optimize performance and power consumption.

In this paper, we view resilience as a set of cross-layer tasks that systems must perform in order to detect and tolerate errors and variation. The detection of errors (current or future) is aided by the underlying hardware observation capabilities of the system (e.g. a few temperature sensors on the multicore chipset). We use the dynamical relationship of the thermal dissipation phenomena is a chip as modeled using a equivalent RC network [16] and exploit that dynamical relationship to compute states that are directly not measured by a sensor. Temperature of the chip significantly effect the failure rate of the chip and with out the ability to observe the unmeasured temperature will result either failure oblivious suboptimal operation or a greater danger of reduced mean time to failure (MTTF).

Virtual observation can also enable the prediction of circuit failures in MPSOC systems. Here, circuit failure prediction prognosticate the occurrence of a circuit failure before errors actually appear in system data and states. This is in contrast to classical error detection where a failure is detected after errors appear in system data and states. Circuit failure prediction is performed during system operation by analyzing the data collected by sensors inserted at various locations inside a chip. The same concept is extended to hardware architecture layer, where it is opportunistically applied to components that impact the system operation most. Agarwal et al. [1] demonstrate this concept of circuit failure prediction for a dominant PMOS aging mechanism induced by Negative Bias Temperature Instability (NBTI). NBTI-induced PMOS aging slows down PMOS transistors over time. As a result, the speed of a chip can significantly degrade over time and can result in delay faults. The traditional practice is to incorporate worst-case speed margins to prevent delay faults during system operation due to NBTI aging. This circuit level sensing integrated inside a flip-flop enables efficient circuit failure prediction at a low cost and demonstrate that this technique can significantly improve system performance by enabling close to best-case design instead of traditional worst-case design.

III. CASE STUDY: TUNING/ESTIMATING MPSOC THERMAL, DELAY, AND FAILURE PROPERTIES

This section illustrates the concept of cross-layer virtual observers with two specific examples and simulations. We consider the case of estimating the unmeasured temperature of many/multi-core systems using measurements from few sensors. In the second case we estimate/ compute the dynamic variations in circuit delay (and an indicator of eventual failure) as well as the mean time to failure (MTTF) (and as an guage for the failure rate) with dynamic variations in temperature, voltage and frequency (DVFS), as well as other cross-layer parameters.

A. Virtual Observation of Temperature and Thermal Profile Prediction

Virtual observation of temperature presented in this example is based on the principles of state estimation [15] and widely popular a discrete Kalman filter [21] in control system literature that uses a state space model of the thermal dissipation as in Hotspot [17], [16] and Wang et al. [19] at block and core level granularity, respectively. Using an analogy from electrical networks, HotSpot represents the thermal characteristics and interactions between various thermal blocks as an RC network, with power inputs modeled as current sources, while heat spreading and storing capacities modeled through resistors and capacitances respectively. The thermal dissipation model as in Wang et al [19], [17], [16] describes the temperature dynamics using a linear time-invariant (LTI) model as function of input power or operating frequency as:

$$\mathbf{T}[k+1] = \mathbf{A_t}\mathbf{T}[k] + \mathbf{B_t}\mathbf{P}[k] \qquad (1)$$

where $\mathbf{A_t}$,$\mathbf{B_t}$ are system matrices generated by the HotSpot or estimated on-line, $\mathbf{T}[k], \mathbf{P}[k] = \begin{bmatrix} P_1[k] & P_2[k] & \dots & P_{n-1}[k] & P_n[k] \end{bmatrix}$ are the temperature and power consumption vector at the k^{th} time instant respectively. The measurement of the temperature states is given by

$$\mathbf{T_m}[k] = \mathbf{C_t}\mathbf{T}[k] + \mathbf{W}[k] \qquad (2)$$

where $\mathbf{T_m}[k]$ is the measured output, $\mathbf{W}[k]$ is the measurement noise, and $\mathbf{C_t}$ is the measurement matrix in the measurement equation (2).

Total active power P_j consumed by a core j is sum of the dynamic power P_{AC_j} and the leakage power (also know as static power) P_{DC_j} for a given supply voltage V_{D_j} and consumes no power when power gated or in sleep mode i.e. V_{D_j} is zero. Thus,

$$P_j = P_{AC_j} + P_{DC_j} = A_i.F_j.V_{D_j}^2 + P_{DC_j} \qquad (3)$$

where A_i is the circuit activity (or activity factor) when executing the application a_i on processor p_j, F_j is the operating frequency, and V_{D_j} is the supply voltage. The leakage power is assumed to be constant with in the sampling time (between two consecutive samples). Thus, the power consumption of a core, when the supply voltage and circuit activity is constant, can be represented as [19]

$$P_j = K_j.F + C_j \tag{4}$$

On the other hand, if the core is operated with fixed supply voltage and frequency (without DVFS), the power consumption is mainly influenced by the circuit activity (or activity factor) of the executing application. Often, application exhibits phased behaviors [7] or different execution modes where the activity factor changes with time resulting in the following power relationship:

$$P_j = K_j.A_i + C_j \tag{5}$$

We remove the constant in (5) by taking difference between two successive samples as in [19] to obtain a state equation relating the change in temperature with change in application activity as:

$$\Delta\mathbf{T}[k] = \mathbf{A}_f\Delta\mathbf{T}[k-1] + \mathbf{B_f}\Delta\mathbf{A_i}[k-1] \tag{6}$$

where $\triangle\mathbf{A_i}[k-1]$ is the change in the activity between successive samples. We construct now a Kalman Filter similar to the one presented in [21] but with the difference that the temperature are estimated for change in the activity instead of power to emphasize cross-layer approach. As the activity changes in the cores, the temperatures varies according to the dynamics in (6). Fig. 5 shows an example with two cores, with a sensor based measurement of the core-1 and that of the virtually observed temperature at core-2 estimated by the the Kalman Filter.

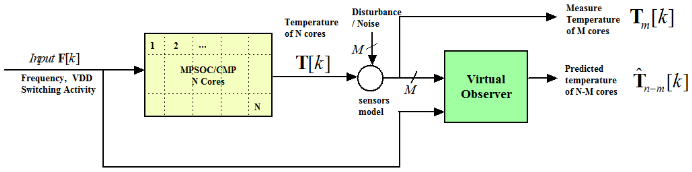


Figure 4. Block diagram representation of virtual sensing scheme.

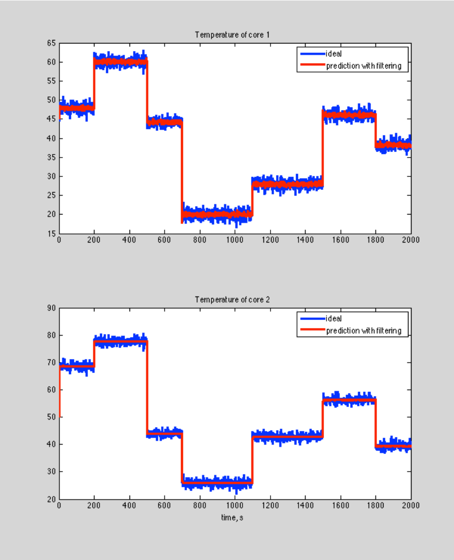


Figure 5. Estimation of the unmeasured temperature of the core 2 from the measured temperature of core 1 while filtering the measurement noise.

B. System Reliability Prediction

The relationship between the circuit delay, threshold voltage and supply voltage is related by the alpha power law [10]. However, as temperature effects circuit delay by effecting carrier mobility and threshold voltage [5], the delay model is modified to reflect the temperature dependence as [10]

$$delay = \tau = K_D\frac{V_D T^{\mu}}{(V_D - V_{TH})^{\lambda}} \tag{7}$$

where K_D is technology specific constant,$\mu = 1.19$, and $\lambda = 1.2$ are empirical constants for 65-nm technology[10].

The circuit delay prediction as above can be used to define a yield function such a way that the circuit delay beyond a threshed value in any critical path would result in a failure. With the ability to virtually observe conditions of each core /block, the failure of each core/ block can quickly be predicted to take corrective steps and avoid such failures. Beside, the delay prediction can be combined with the mean time to failure (MTTF) of the circuit blocks to achieve a consistent and reliable prediction. The mean time to failure (MTTF) of metal wire interconnects in chips when subjected to electromigration [8] follows an exponential relationship with temperature as in (8)

$$MTTF_{EM} = \frac{1}{\lambda_{EM}} = K_{EM}.\frac{A_c}{J^n}.exp\left(\frac{E_a}{k.T}\right) \tag{8}$$

where λ_{EM} is the failure rate due to electromigration, K_{EM} is a technology dependent constant, A_c is the cross-sectional area, J is the current density, n is the scaling factor usually (1.1 to 2), E_a is the activation energy , k is the Boltzmann constant, and T is the absolute temperature in degree Kelvin. The current density of the core J can be related to the switching activity of the line as [6]:

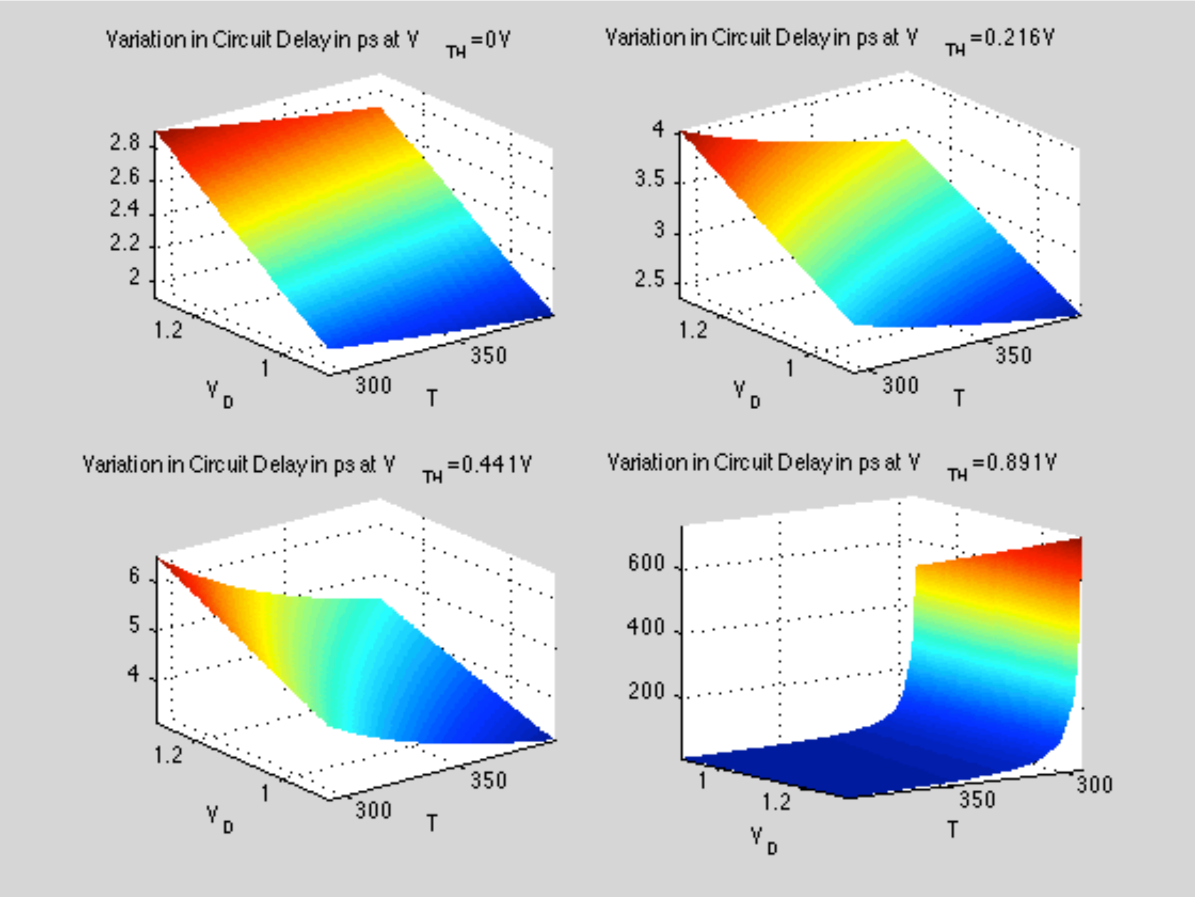
$$J = K_J\frac{CV_{D_j}}{WH}.F_j.A_i \tag{9}$$

where K_j is constant, $C, W,$ and H are the capacitance, thickness, and width of the line respectively.

By estimating the temperature of the unmeasured core (or block), we predict the MTTF and the failure rate of the core as shown in Fig. 7. The predicted values enables to take proactive and corrective actions in a closed loop using suitable adaptation and control techniques [15] as discussed in the next section.

C. Closed-Loop Adaptation and Control

The adaptation is key part of the reflective middleware which can be achieved by classical controller like proportional (P), proportional-integral (PI), and proportional-integral-derivative (PID) as used in Skadron at el [17], [16] to control the temperature of the cores. In this classical temperature control approach, a reference temperature for each core is defined (that is with in the safe limit) and the error signal between the desired reference and the predicted temperature is fed in to the classical controller to compute an actuation signal (in terms of change in supply voltage and operating frequency as in dynamic voltage and frequency scaling (DVFS)) to reduce and compensate the error between the desired reference and the predicted temperature as depicted in Fig.(8). The DVFS changes the power consumption of the core which directly control the core temperature as described by the dynamics in (1). In our specific example case study, the controller output controls the chip activity to compensate for the error, for instance, by reducing the frame rate in video and multimedia processing, or by migrating relatively hot tasks to colder cores and vice versa. As shown in (3) power consumed can be controlled by the supply voltage, operating frequency, as well the chip activity, all the three control knobs are orthogonal to each other and applied simultaneously to achieve aggressive results. The same approach is extended for the control of circuit delay and failure rate. It may be noted that the current density J in (9) (and hence switching activity) has similar impact on the failure rate (and MTTF) as that of the temperature. Consequently, the control of the switching activity would be an equally effective control mechanism as that of the temperature for controlling the failure rate due to electromigration.


Variation in Circuit Delay in ps at V_{TH} =0V

Variation in Circuit Delay in ps at V_{TH} =0.216V

Variation in Circuit Delay in ps at V_{TH} =0.441V

Variation in Circuit Delay in ps at V_{TH} =0.891V

Figure 6. Impact of supply voltage and temperature variations in circuit delay.

IV. Conclusion

Increasing error, variation, and aging rates in semiconductor systems are making it more and more costly to tolerate all of the possible non-ideal device behaviors in one or two layers of the system stack. Distributing resilience and reliability across the system stack can improve performance and reduce power and area costs by taking advantage of the strengths of each layer and exploiting the characteristics of individual applications. In this research, we propose a design framework with cross-layer virtual sensing and actuations that enable adaptation across different system stacks with the aim of achieving improved reliability, performance, thermal stability, while reducing power and energy consumption. We demonstrate the effectiveness and applicability of these concepts specifically overcoming the vulnerabilities introduced by faults and process variability using two case studies: one using virtual observer to estimate temperature of unmeasured core and the other to predict the failure rate of the cores using the estimated temperature using concepts drawn from embedded multiprocessor systems on a single chip (MPSoC) respectively.

Acknowledgement

This material is based on work supported by the NSF under awards CCF-1029783 (Variability Expedition) and CNS-1063596. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] M. Agarwal, B.C. Paul, Ming Zhang, and S. Mitra. Circuit failure prediction and its application to transistor aging. In *VLSI Test Symposium, 2007. 25th IEEE*, pages 277 –286, may 2007.

[2] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *Dependable and Secure Computing, IEEE Transactions on*, 1(1):11 – 33, jan.-march 2004.

[3] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De. Parameter variations and impact on circuits and microarchitecture. In *Design Automation Conference, 2003. Proceedings*, pages 338 – 342, june 2003.

[4] N.P. Carter, H. Naeimi, and D.S. Gardner. Design techniques for cross-layer resilience. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2010*, pages 1023 –1028, march 2010.

[5] R.S.C. Cobbold. Temperature effects in m.o.s. transistors. *Electronics Letters*, 2(6):190 –191, june 1966.

[6] A. Dasgupta and R. Karri. Electromigration reliability enhancement via bus activity distribution. In *Design Automation Conference Proceedings 1996, 33rd*, pages 353 –356, jun, 1996.

[7] C. Isci, A. Buyuktosunoglu, and M. Martonosi. Long-term workload phases: duration predictions and applications to dvfs. *Micro, IEEE*, 25(5):39 – 51, sept.-oct. 2005.

[8] JEDEC. *Failure Mechanisms and Models for Semiconductor Devices*. In JEDEC Publication JEP122-A, 2002.

[9] J.O. Kephart and D.M. Chess. The vision of autonomic computing. *Computer*, 36(1):41 – 50, jan 2003.

[10] Weiping Liao, Lei He, and K.M. Lepak. Temperature and supply voltage aware performance and power modeling at microarchitecture level. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 24(7):1042 – 1053, july 2005.

[11] R.C. Luo, Chih-Chen Yih, and Kuo Lan Su. Multisensor fusion and integration: approaches, applications, and future research directions. *Sensors Journal, IEEE*, 2(2):107 –119, apr 2002.

[12] Michael Orshansky, Linda Milor, Pinhong Chen, Kurt Keutzer, and Chenming Hu. Impact of systematic spatial intra-chip gate length variability on performance of high-speed digital circuits. In *Proceedings of the 2000 IEEE/ACM international conference on Computer-aided*

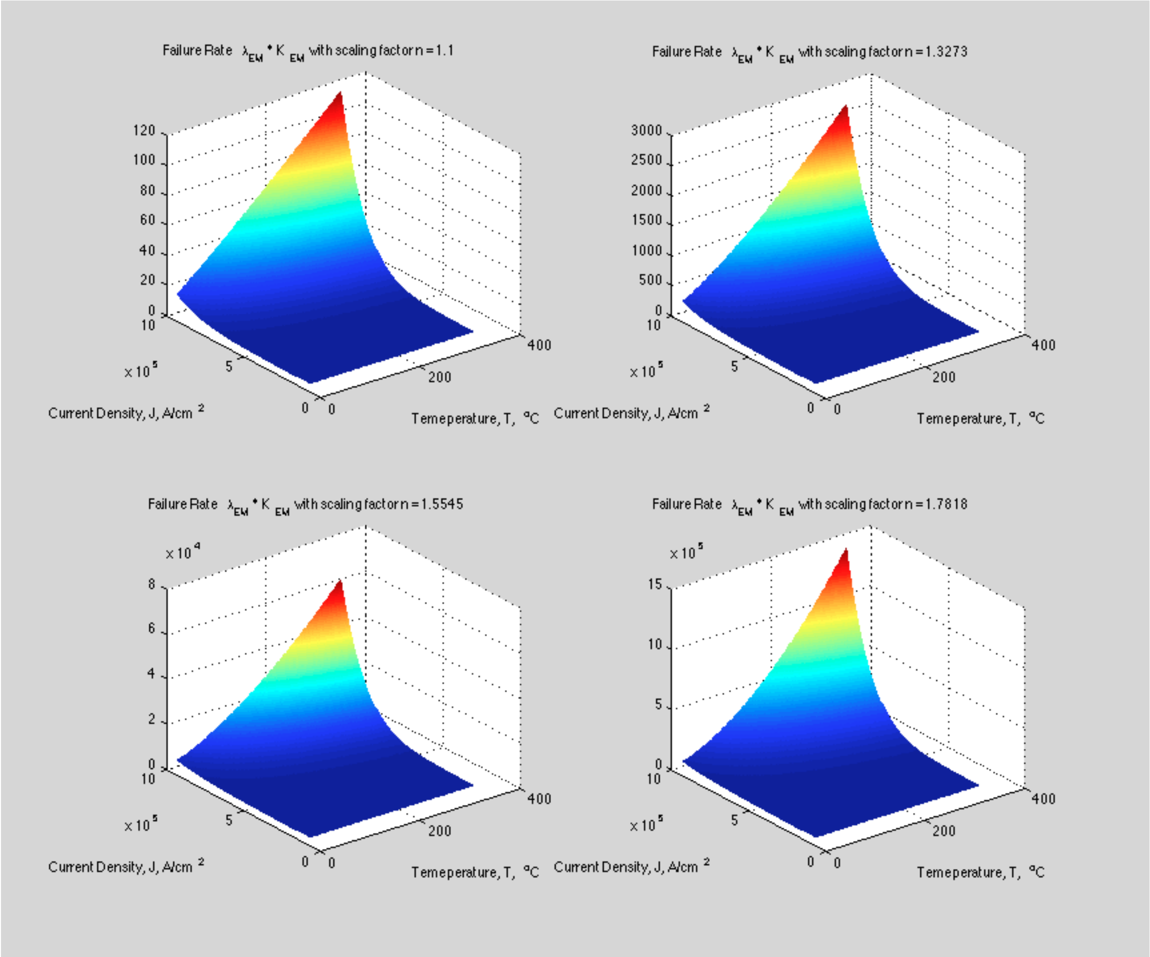


Figure 7. Variation in failure rate with current density and temperature due to Electromigration failure mechanism.

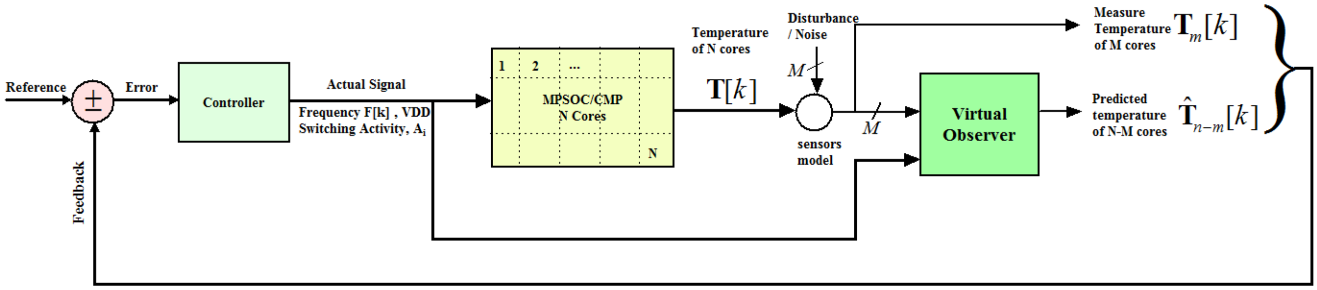


Figure 8. Closed-loop adaptation and control.

design, ICCAD '00, pages 62–67, Piscataway, NJ, USA, 2000. IEEE Press.

[13] N. Raveendranathan, S. Galzarano, V. Loseu, R. Gravina, R. Giannantonio, M. Sgroi, R. Jafari, and G. Fortino. From modeling to implementation of virtual sensors in body sensor networks. *Sensors Journal, IEEE*, 12(3):583 –593, march 2012.

[14] M.D. Santambrogio, H. Hoffmann, J. Eastep, and A. Agarwal. Enabling technologies for self-aware adaptive systems. In *Adaptive Hardware and Systems (AHS), 2010 NASA/ESA Conference on*, pages 149 –156, june 2010.

[15] Shankar Sastry and Marc Bodson. *Adaptive control: stability, convergence, and robustness*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[16] Kevin Skadron, Tarek Abdelzaher, and Mircea R. Stan. Control-theoretic techniques and thermal-rc modeling for accurate and localized dynamic thermal management. In *Proceedings of the 8th International Symposium on High-Performance Computer Architecture*, HPCA '02, pages 17–, Washington, DC, USA, 2002. IEEE Computer Society.

[17] Kevin Skadron, Mircea R. Stan, Karthik Sankaranarayanan, Wei Huang, Sivakumar Velusamy, and David Tarjan. Temperature-aware microarchitecture: Modeling and implementation. *ACM Trans. Archit. Code Optim.*, 1(1):94–125, March 2004.

[18] Jürgen Teich, Jörg Henkel, Andreas Herkersdorf, Doris Schmitt-Landsiedel, Wolfgang Schröder-Preikschat, and Gregor Snelting. Invasive computing: An overview. In M. Hübner and J. Becker, editors, *Multiprocessor System-on-Chip – Hardware Design and Tool Integration*, pages 241–268. Springer, Berlin, Heidelberg, 2011.

[19] Xiaorui Wang, Kai Ma, and Yefu Wang. Adaptive power control with online model estimation for chip multiprocessors. *Parallel and Distributed Systems, IEEE Transactions on*, 22(10):1681 –1696, oct.

2011.

[20] Wangyang Zhang, Xin Li, and Rob A. Rutenbar. Bayesian virtual probe: Minimizing variation characterization cost for nanoscale ic technologies via bayesian inference. In *Design Automation Conference (DAC), 2010 47th ACM/IEEE*, pages 262 –267, june 2010.

[21] Yufu Zhang and A. Srivastava. Leakage-aware kalman filter for accurate temperature tracking. In *Green Computing Conference and Workshops (IGCC), 2011 International*, pages 1 –7, july 2011.