# CyberPhysical-System-On-Chip (CPSoC): A Self-Aware MPSoC Paradigm with Cross-Layer Virtual Sensing and Actuation

S.Sarma, N.Dutt, P.Gupta†, N. Venkatasubramanian and A. Nicolau
Department of Computer Science, University of California Irvine, CA, USA.
Email:{santanus,dutt,nalini,nicolau}@ics.uci.edu
†Department of Electrical Engineering, University of California Los Angeles, CA, USA.
Email: puneet@ee.ucla.edu

***Abstract*—Cyber–physical systems (CPSs) are physical and engineered systems whose operations are monitored, coordinated, controlled, and integrated by a computing, control, and communication core. We propose Cyberphysical-System-on-Chip (CPSoC), a new class of sensor and actuator-rich multiprocessor systems-on-chip (MPSoCs), that augment MPSoCs with additional on-chip and cross-layer sensing and actuation capabilities to enable self-awareness within the observe-decide-act (ODA) paradigm. Unlike traditional MPSoC designs, CPSoC differs primarily on the co-design of computing-communication-control (C3) systems that interacts with the physical environment in real-time in order to adapt system behavior so as to dynamically react to environmental changes while achieving overall design goals. We illustrate CPSoC's potential through a virtual sensor network that accurately estimates run-time power for variability affected subsystems using noisy thermal sensors in improving system goals and Quality-of-Service (QoS).**

***Index Terms*—Cyber Physical Systems, Cross-Layer Approach, Self-Aware Computing, Adaptive Computing , MPSoC, CyberPhysical-System-On-Chip (CPSoC).**

## I. Introduction

DUE to increased demand of high performance, higher power/energy efficiency, and expanding functionality, prevailing MPSoCs are moving towards unprecedented parallelism and heterogeneous many/multicore architectures with several hundreds or even thousands of cores that need to deal with a diverse and rapid stream of dynamically changing applications with competing and conflicting demands/goals. Furthermore, MPSoCs need to deal with dramatic manufacturing process variability (as semiconductor technology dives deeper into the nanometer era), and increased vulnerability to environmental and aging effects that induce errors and subsequent faults and failures. In addition, MPSoCs face vexing thermal and heating hazards, creating drastic and harsh environments (e.g., hotspots), that further aggravate aging and wear-out phenomena (e.g., NBTI, HCI, TDDB, Electromigration etc. [1]) resulting in increased susceptibility to errors with the immediate consequence of diminishing yield, reliability and reduced usage lifetime [1], [2].

Platforms with new levels of heterogeneity in interconnected cores result in challenging coupled/coordinated interactions, and hard to fine-tune scores of runtime parameters for sustained efficiency. Consequently, there is a need for improved abstraction to manage the complexity, synergistic cross-layer cooperation and adaptations to effectively manage the on-chip resources, and new means of actuations and actions to meet the aggressive and competing demands/goals. Additionally, MPSoCs need to sense many more physical phenomena and system states across multiple abstraction levels in order to exploit workload and process variabilities [1], find root causes of faults and failures, as well as identify vulnerabilities (e.g. thermal hotspots, malicious attacks) to take proactive actions.

This work was partially supported by the NSF Variability Expedition award CCF-1029783 (Variability Expedition) and CNS-1063596 (Cypress).

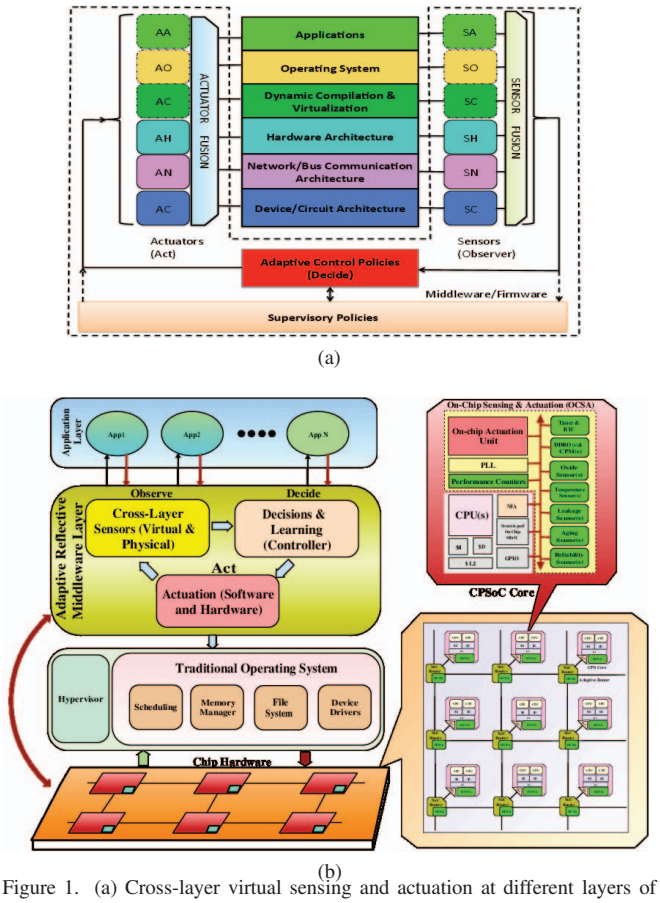


Figure 1. (a) Cross-layer virtual sensing and actuation at different layers of CPSoC (b) CPSoC architecture with adaptive Core, NoC, and the ODA Loop as Adaptive, Reflexive Middleware [3].

In this paper, we propose CyberPhysical-system-on-chips (CPSoC), a novel architecture and design paradigm that combines a sensor-actuator-rich self-aware computing-communication-control (C3) centric platform. CPSoC deploys an adaptive & reflective middleware (a flexible hardware-software stack and interface between the application and OS layer) to control the manifestations of computations (e.g., aging, overheating, parameter variability etc.) on the physical characteristics of the chip itself and the outside interacting environment. Inspired by the C3 paradigm of CPSs [4] and adaptive and learning abilities of autonomous computing [5], CPSoC provides a computing framework that addresses and assures the dependability of the cyber/information processing (i.e., the cyber aspects such as integrity, correctness, accuracy, timing, reliability and security) while simultaneously addressing the physical manifestations (in performance, power, thermal, aging, wear-out, material degradation, and reliability and dependability) of the information processing on the underlying computing platform. CPSoC aims to coalesce these two traditionally disjoint aspects/abstractions of cyber/information world and the underlying physical computing worlds into a unified abstraction of computing by using cross-layer virtual/physical sensing and actuation to enable a C3 centric self-aware computing platform.

## II. Architecture of CPSoC

The CPSoC architecture consists of a combination of sensor-actuator-rich computation platform supported by adaptive NoCs (cNoC, sNoC), Introspective Sentient Units (ISU), and an adaptive & reflective middleware to manage and control both the cyber/information and physical environment and characteristics of the chip. The CPSoC architecture is broadly divided into several layers of abstraction, for example, applications, operating system, network and bus communication, hardware, and the circuit / device layers. CPSoC inherits most features of MPSoC in addition to on-chip sensing and actuation to enable the ODA paradigm. Unlike traditional MPSoC, each layer of the CPSoC can be made self-aware and adaptive, by a combination of software and physical sensors and actuators as shown in Fig. 1a. These layer specific feedback loops are integrated into a flexible stack which can be implemented either as firmware or middleware as shown by the dotted line in Fig. 1a.

CPSoC distinctly differs from a traditional MPSoC in several ways. Traditional MPSoC paradigms lack the ability to sense the system states and behaviors across layers of system stack due to lack of architectural support; they are incapable of exploiting and exposing process and workload variations due to lack of suitable abstractions at multiple layers. Furthermore, they sacrifice usable performance and energy opportunities by adopting worst case design (guard-bands), and lack support for multi-level actuation mechanisms and adaptations to aggressively meet competing and conflicting demands. Moreover, traditional MPSoCs lack self-learning mechanisms that can anticipate failures and predict vulnerabilities. CPSoC overcomes these limitations as detailed below.

## III. Attributes and Features of CPSoC

The CPSoC framework supports four key ideas: 1) physical and virtual sensing and actuation 2) self-awareness and adaptation 3) multi or cross-layer interactions and interventions 4) predictive modeling and learning. We briefly describe these below. (A detailed description is in our Technical Report[6].)

### A. Cross-Layer Virtual and Physical Sensing & Actuation

CPSoCs are sensor-actuator-rich MPSoCs that include several on-chip physical sensors (e.g., aging, oxide breakdown, leakage, reliability, temperature, performance counters, as well as voltage, current, and power sensors [6]) on the lower three layers as shown by the on-chip-sensing-and-actuation block (OCSN) in Fig. 1b and tabulated in Table I. On the other hand, **virtual sensing** is a physical-sensor-less sensing of immeasurable parameters using indirect computation. It's a software sensor that provides 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 heterogeneous sensors. It's a computational technique that enhances and/or adds sensing capability, introduces sensing options, increases sensitivity, enables efficient sensor resource uses, and overcomes physical placement and cost restrictions. When combined with different kinds of sensors, virtual sensing enables consensus to resolve faults and errors while providing a test bed for on-chip sensor fusion. The need for such an over-provisioned sensing architecture for MPSoCs has also been identified by Intel [7].

Table I
Virtual/Physical Sensing and Actuations Across Layers

| Layers | Virtual/Physical Sensors | Virtual/Physical Actuators |
|---|---|---|
| **Application** | Workload type, code features, func. complexity (cyclomatic), power, energy, Execution time, | Loop perforation, memoization Algorithmic Choice, Degree of Parallelism, code redundancy |
| **Operating System** | System Utilization, Epoch length, Context Switch Counter, Thread load , History | Task Allocation, Scheduling, Migration, Offloading, Duty Cycling, Mem. Page Allocation |
| **Dynamic Compilation & Virtualization** | Code profiling stats, Instruction Groups, program phases, access patterns, critical instructions, | Data and code transformation (equivalence, multiplicity, rel. position, stack &heap size) |
| **Hardware Architecture** | Cache misses, Miss rate; access rate; branch miss , TLB misses, IPC, ILP/MLP, Core asymmetry | Cache Sizing; Reconfiguration, Resource Provisioning, Static/Dynamic Redundancy |
| **Network/Bus Communication** | Bandwidth; Packet/Flit status; Channel Status, Congestion, Latency, bus / router power | Adaptive Routing Dynamic Bandwidth Allocation Ch. no and direction |
| **Circuit/Device** | Ckt Delay, Aging, leakage, Power, Temperature, oxide breakdown, Reliability | DVFS, ABB, Reverse Biasing, Clock& power-gating, Multi-gate thresholding, |

Similarly, we define **virtual actuations** [6](e.g., application duty cycling, algorithmic choice, checkpointing) that are software/hardware interventions that can predictively influence system design objectives such as performance, power, and reliability. Virtual actuations can be combined with physical actuation mechanisms commonly adopted in modern chips (e.g., DVFS and adaptive body biasing (ABB) to control the chip performance, power, and parametric variations); the notion of actuator fusion in CPSoC represents virtual and physical actuations that are combined across different layers of abstraction [6].

### B. Self-Awareness and Adaptation

Self-awareness is used to describe the ability of the CPSoC to observe its own internal behaviors as well as external systems it interacts with such that it is capable of making judicious decisions that optimize performance and other quality of service (QoS) metrics [5]. Self-aware 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. A self-aware system must be able to monitor its behavior to update one or more of its components (hardware architecture, operating system and running applications), to achieve its goals. Similar efforts on self-awareness and adaptation are being pursued by other researchers including [8], [9], [10].

### C. Multi or Cross-Layer Interactions and Interventions

Two key attributes of the self-aware CPSoC are adaptation of each layer and multiple cooperative ODA loops. As an example, the unification of an adaptive computing platform (with combined DVFS, ABB, and other actuation means) along with a bandwidth adaptive NoC [6] offers a completely different approach (extra dimensions of control) and solutions in comparison to traditional MPSoC architecture. These cooperative and hierarchical control loops –e.g., the combination of traditional control loop (dotted lower box in Fig. 2) together with virtual sensing enabled optimized loop (upper

loop in Fig. 2) – effectively translate user goals or QoS into one or more multiple design objectives [6].

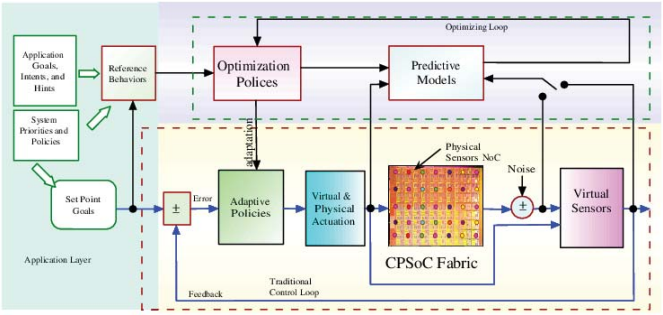


Figure 2. Adaptation using predictive control model and policies in CPSoC.

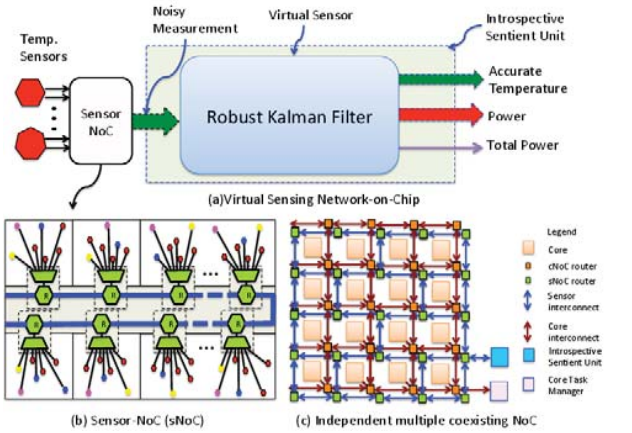


Figure 3. (a)Virtual Sensor Network-on-Chip Architecture (b) Sensor NoC (sNoC) (c) Independent multiple coexisting NoCs in CPSoC.

### D. Predictive Modeling and Learning

Predictive modeling and learning abilities of the system behavior as well as internal and external (environmental) states provide self-modeling abilities in the CPSoC paradigm. The system behavior and states can be built using on-line or off-line linear or non-linear models in time or frequency domains [11]. We specifically use statistical and neural network approaches [12], [13] such that the model accuracy can be traded-off for model computational complexity. We use regression based linear predictors and nonlinear neural predictors to build models of the system performance, power and energy consumption using the cross-layer events, hardware counters, and on-chip sensor data [6]. In addition, use of coupling parameters (a metric that quantifies the interactions between layers) helps to develop application and cross-layer interaction models for nominal and abnormal operations. We use the predictive and learning abilities of CPSoC to improve autonomy in managing the system resources and assisting proactive resource utilization in the run-time system [6].

## IV. CPSoC Applications and Use Cases

We demonstrate the applicability of CPSoC using several examples in [6]. Here we illustrate one such use case for accurate run-time power and temperature estimation and prediction using noisy thermal sensors placed at the suitable locations in the CPSoC fabric.

### A. Virtual Run-time Power Sensing of Subsystems Using Noisy Thermal Sensors

This scenario illustrates the virtual sensing capability of CPSoC in jointly estimating and predicting the run-time transient power consumption and accurate temperature of each subsystem unit as well as accurate temperatures from noisy temperature sensors readings using the observe-decide-act (ODA) paradigm. The virtual approach (Fig.3(a)) combines a sensor-network-on-chip (sNoC) and a robust Kalman filter [14] to enable in-situ, on-the-fly run-time estimation of the power and temperature of each block/unit from on-chip noisy thermal measurements. Unlike traditional MPSoCs, the specialized sensor network (Fig.3(b)) can coexist independently or combined with the core-to-core communication network (cNoC) as shown in Fig.3(c). Such an architecture decouples the sensing and computation concerns and allows flexibility to independently perform on-chip sensing without interfering or burdening the core-to-core computation network.

Temperature values at different locations on the die depend on various factors such as power consumptions of functional units, layout of the chip and the package characteristics. The differential equations describing the heat flow have a form dual to that of electrical current, represented using lumped values of thermal R and Cs, and form the basis for commonly used micro-architectural thermal models in state space form [15][16].

Virtual sensing of the run-time transient power requires the thermal model which represents the relation between subsystem power and the temperature. However, to address the emerging problem of semiconductor process variations in deep sub-micron technologies and sensor noise [17], the processor thermal dynamics is augmented with the process and measurement noise in the standard LTI model as:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}_k\mathbf{x}_k + \mathbf{E}_k\mathbf{d}_k + \mathbf{w}_k \\ \mathbf{y}_k &= \mathbf{H}_k\mathbf{x}_k + \eta_k \end{aligned} \tag{1}$$

where $\mathbf{x_k} \in \mathbb{R}^{\mathbf{n}}$ are the system states (i.e., each block's temperature), $\mathbf{d_k} \in \mathbb{R}^{\mathbf{p}}$ are the unknown inputs (i.e., each block's power), $\mathbf{y_k} \in \mathbb{R}^{\mathbf{m}}$ are the measurements (i.e., the temperature sensor measurement), $\mathbf{A}_k$,$\mathbf{E}_k$, $\mathbf{H_k}$ are system matrices of appropriate dimensions. The process noise $\mathbf{w}_k$ and measurement noise $\eta_\mathbf{k}$ are assumed to be mutually uncorrelated, zero-mean, white random signals with known covariance matrices, $\mathbf{Q_k} = \mathbf{E}\left[\mathbf{w_k}\mathbf{w_k^T}\right]$ and $\mathbf{R_k} = \mathbf{E}\left[\eta_\mathbf{k}\eta_\mathbf{k}^\mathbf{T}\right]$ respectively.

To effectively construct the virtual sensing mechanism for the variability and noise corrupted dynamic system in (1), we use an optimal Kalman filter to provide robust estimates. The objective of the Kalman filter is to jointly estimate the unknown power input $\mathbf{d_k}$ as well as the temperature states $\mathbf{x_k}$ from noisy temperature measurements in the presence of both process and measurement noise as described in Fig. 4. Robust Kalman filters [14] have been shown to provide unbiased estimates of the states and unknown inputs and guarantee global optimality and minimum variance of the estimates [14]. Consequently, virtual sensors constructed using such filters will provide the statistically best solution for the thermal and power estimation of the subsystems.

To validate our approach we created a CPSoC simulation platform called CPSoCSim [18]. We used the Alpha 31386 processor and its compilation tools that have been extensively used in earlier research work [15], [16]. We use the SPEC and PARSEC benchmarks and their corresponding power traces to generate the thermal profile using the Hotspot simulator [16]. We assume that the temperature of the blocks are measured with noisy sensors and are collected by the sNoC so that these can be processed to produce thermal and the power estimates at each subsystem unit of the processor. Figure 5 shows the actual temperature obtained from Hotspot [16] versus the temperature obtained from the virtual sensor. The estimation of the power of each units and run-time tracking of total power consumed by the processor are shown in Fig. 6(a) and (b) respectively. Although the estimation error is a function of process and measurement noise, for typical scenarios the temperature and power estimate errors are less than 1% and 5% respectively. Note that no power sensors were used in the whole process of power estimation of the subsystems; results were generated indirectly by computational means using virtual sensing.

**Algorithm: Virtual Sensing of Subsystem Power and Temperature**

**Input**: Temperature sensor measurement from sensors $\mathbf{y}$, Thermal state space model $\mathbf{A_k}, \mathbf{E_k}, \mathbf{H_k}, \mathbf{Q_k}, \mathbf{R_k}$
**Output**: Subsystem Thermal Profiles, $\hat{\mathbf{x}}_\mathbf{k}$, and Power Profile, $\hat{\mathbf{d}}_k$, Total Power, $P_{total}$

Perform the following every sampling step:

1) Initialize the values of $\hat{\mathbf{x}}_\mathbf{k}, \hat{\mathbf{P}}_\mathbf{k}, \mathbf{A_k}, \mathbf{E_k}, \mathbf{H_k}, \mathbf{Q_k}, \mathbf{R_k}$
2) Robust Two-Stage Kalman Filter (RTSKF):
   a) $\bar{\mathbf{x}}_{\mathbf{k}|\mathbf{k-1}} = \mathbf{A_k}\bar{\mathbf{x}}_{\mathbf{k}|\mathbf{k-1}}$
   b) $\mathbf{P}^{\bar{\mathbf{x}}}_{\mathbf{k}|\mathbf{k-1}} = \mathbf{A}_{k-1}\hat{\mathbf{P}}_{\mathbf{k-1}|\mathbf{k-1}}\mathbf{A}^{\mathbf{T}}_{\mathbf{k-1}} + \mathbf{Q_{k-1}}$
   c) $\mathbf{C_k} = \mathbf{H_k}\mathbf{P}^{\bar{\mathbf{x}}}_{\mathbf{k}|\mathbf{k-1}}\mathbf{H}^{\mathbf{T}}_{\mathbf{k}} + \mathbf{R_k}$
   d) $\mathbf{P}^{\mathbf{d}}_{\mathbf{k}|\mathbf{k}} = \left\{\mathbf{E}^{\mathbf{T}}_{\mathbf{k-1}}\mathbf{H}^{\mathbf{T}}_{\mathbf{k}}\mathbf{C}^{-\mathbf{1}}_{\mathbf{k}}\mathbf{H_k}\mathbf{E_{k-1}}\right\}^{-1}$
   e) $\mathbf{K}^{\mathbf{d}}_{\mathbf{k}} = \mathbf{P}^{\mathbf{d}}_{\mathbf{k}|\mathbf{k}}\mathbf{E}^{\mathbf{T}}_{\mathbf{k-1}}\mathbf{H}^{\mathbf{T}}_{\mathbf{k}}\mathbf{C}^{-\mathbf{1}}_{\mathbf{k}}$
   f) $\mathbf{K}^{\bar{\mathbf{x}}}_{\mathbf{k}} = \mathbf{P}^{\bar{\mathbf{x}}}_{\mathbf{k}|\mathbf{k-1}}\mathbf{H}^{\mathbf{T}}_{\mathbf{k}}\mathbf{C}^{-\mathbf{1}}_{\mathbf{k}}$
   g) $\mathbf{V_k} = (\mathbf{I} - \mathbf{K}^{\bar{\mathbf{x}}}_{\mathbf{k}}\mathbf{H_k})\mathbf{E_{k-1}}$, $\mathbf{I}$ is identity matrix
   h) $\hat{\mathbf{d}}_{\mathbf{k}|\mathbf{k}} = \mathbf{K}^{\mathbf{d}}_{\mathbf{k}}\left(\mathbf{y_k} - \mathbf{H_k}\bar{\mathbf{x}}_{\mathbf{k}|\mathbf{k-1}}\right)$
   i) $\bar{\mathbf{x}}_{\mathbf{k}|\mathbf{k}} = \bar{\mathbf{x}}_{\mathbf{k}|\mathbf{k-1}} + \mathbf{K}^{\bar{\mathbf{x}}}_{\mathbf{k}}\left(\mathbf{y_k} - \mathbf{H_k}\bar{\mathbf{x}}_{\mathbf{k}|\mathbf{k-1}}\right)$
   j) $\hat{\mathbf{P}}^{\mathbf{x}}_{\mathbf{k}|\mathbf{k}} = \mathbf{P}^{\bar{\mathbf{x}}}_{\mathbf{k}|\mathbf{k}} + \mathbf{V_k}\mathbf{P}^{\mathbf{d}}_{\mathbf{k}|\mathbf{k}}\mathbf{V}^{\mathbf{T}}_{\mathbf{k}}$
   k) $\mathbf{P}^{\bar{\mathbf{x}}}_{\mathbf{k}|\mathbf{k}} = (\mathbf{I} - \mathbf{K}^{\bar{\mathbf{x}}}_{\mathbf{k}}\mathbf{H_k})\mathbf{P}^{\bar{\mathbf{x}}}_{\mathbf{k}|\mathbf{k-1}}$
   l) $\hat{\mathbf{x}}_{\mathbf{k}|\mathbf{k}} = \bar{\mathbf{x}}_{\mathbf{k}|\mathbf{k-1}} + \mathbf{V_k}\mathbf{d}_{\mathbf{k}|\mathbf{k}}$
3) Output the estimates: Power Estimates , $\hat{\mathbf{d}}$; Total Power, $P_{total}$; Temperature Estimates, $\hat{\mathbf{x}}$

Figure 4. Virtual Sensing of Core and Subsystem Temperature and Power.

The overall overhead of the CPSoC considering 1000 sensors (consisting of 5 types), the sNoC and the ISU is within 7.3150 % of the area and 0.6476 % of the power budget of an equivalent 16-core ARM Cortex A9 platform. Using virtual sensing of power the overall overhead is further reduced to 1.6834% of the area and 0.2529% of the power budget of the 16-core platform [6].

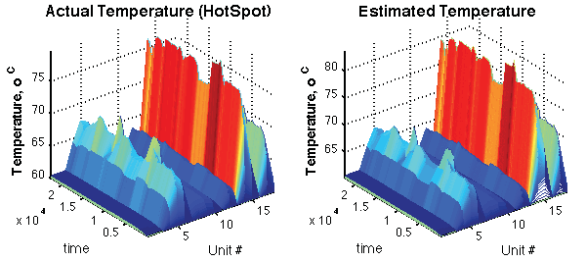


Figure 5. Run-time subsystem temperature estimation.

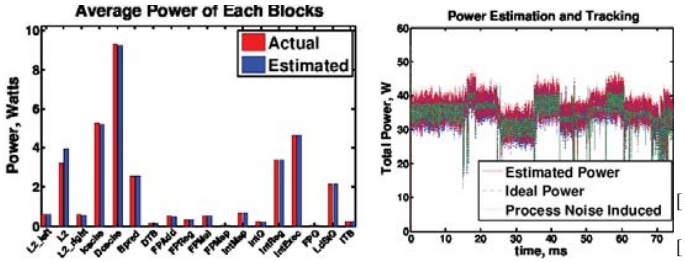


Figure 6. (a) Power estimation of subsystem units (b) Run-time total power estimation and tracking.

### *B. Improving System QoS through Cross-Layer Adaptation*

On-chip self-awareness with cross-layer virtual and physical sensing and actuation is a key enabling technology for efficient use of heterogeneous architectures, and applications with guaranteed run-time system goals and QoS (performance, reliability, power, thermal behavior) in a highly dynamic environment. Our Technical Report [6] contains several sample applications where self-awareness is used to improve energy efficiency, increase system lifetime by reducing aging effects and improve system performance under thermal constraints. For instance, we show that cross-layer virtual sensing and actuation can improve the sensing accuracy and reduce the sensing overhead of thermal estimation by an order of magnitude [6]. In another example we show how virtual sensing enables an adaptive scheduler (an actuation mechanism) to exploit heterogeneous architectures for energy efficiency; our experimental result shows over a 50 % improvement in energy efficiency for a quad-core system with much higher gains expected for larger systems. We also demonstrate a 4-year improvement in lifetime for a 20-core heterogeneous CPSoC compared to a traditional (area and power equivalent) homogenous MPSoC that has an average 28-year lifetime. We are currently investigating more aggressive cross layer sensing and actuation mechanisms to improve system resilience and energy efficiency using a FPGA emulation and prototyping platform [18].

## V. Conclusion

In this paper we presented CPSoC, a sensor-actuator-rich MPSoC platform that deploys the computation-communication-control co-design of CPS together with cross-layer adaptations to achieve multiple design objectives. The CPSoC paradigm enables self-awareness (i.e., the ability of the system to observe it's own internal and external behaviors such that it is capable of making judicious decisions) and (selective or opportunistic) adaptation using the concepts of cross-layer physical and virtual sensing and actuations. In [6] we illustrate CPSoC's potential for self-awareness and cross-layer adaptations using several examples, and in this paper we present the specific instance of estimating power and temperature accurately with low overhead at runtime.

## References


[1] P. Gupta *et al.*, "Underdesigned and opportunistic computing in presence of hardware variability," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Trans. on*, vol. 32, no. 1, pp. 8–23, 2013.

[2] S. Borkar *et al.*, "Parameter variations and impact on circuits and microarchitecture," in *DAC, 2003. Proc.*, June 2003, pp. 338 – 342.

[3] S. Sarma *et al.*, "On-chip self-awareness using CyberPhysical-Systems-on-Chip (CPSoC)," in *CODES+ISSS'14*. ACM, 2014, pp. 22:1–22:3.

[4] E. Lee, "Cyber physical systems: Design challenges," in *ISORC, 2008*, may 2008, pp. 363 –369.

[5] J. Kephart *et al.*, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41 – 50, Jan 2003.

[6] S.Sarma *et al.*, "CyberPhysical-System-On-Chip (CPSoC): Sensor-actuator rich self-aware computational platform," University of California Irvine, Tech. Rep. CECS-TR-13-06, 2013.

[7] S. Borkar, "Achieving energy efficiency by HW/SW co-design," in *3rd Berkeley Symposium on Energy Efficient Electronic Systems*, October 2013.

[8] H. Hoffmann *et al.*, "Self-aware computing in the Angstrom processor," in *DAC 2012*. IEEE, 2012, pp. 259–264.

[9] *Invasive Computing*. TCRC 89, 2014. [Online]. Available: http://invasic.informatik.uni-erlangen.de/en/index.php

[10] *Dependable Embedded Systems*. DFG SPP 1500, 2014. [Online]. Available: http://spp1500.itec.kit.edu/

[11] L. Ljung, *System identification*. Springer, 1998.

[12] S. S. Haykin *et al.*, *Neural networks and learning machines*. Pearson Education Upper Saddle River, 2009, vol. 3.

[13] L. V. Fausett, *Fundamentals of neural networks*. Prentice-Hall, 1994.

[14] C.-S. Hsieh, "Robust two-stage Kalman filters for systems with unknown inputs," *Automatic Control, IEEE Trans. on*, vol. 45, no. 12, pp. 2374–2378, 2000.

[15] K. Skadron *et al.*, "Temperature-aware microarchitecture," ser. ISCA '03. New York, NY, USA: ACM, 2003, pp. 2–13.

[16] W. Huang *et al.*, "Hotspot: a compact thermal modeling methodology for early-stage vlsi design," *VLSI Systems, IEEE Trans. on*, vol. 14, no. 5, pp. 501–513, 2006.

[17] Y. Zhang *et al.*, "Accurate temperature estimation using noisy thermal sensors for gaussian and non-gaussian cases," *VLSI Systems, IEEE Trans. on*, vol. 19, no. 9, pp. 1617–1626, 2011.

[18] S.Sarma and N. Dutt, "FPGA emulation and prototyping of a CyberPhysical-System-On-Chip (CPSoC)," in *RSP'14*, October 2014.