On-Chip Self-Awareness Using Cyberphysical-Systems-On-Chip (CPSoC)

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

The increased demands of high performance, higher power/energy efficiency, and expanding functionality are moving traditional MultiProcessor Systems-on-Chips (MPSoCs) towards heterogenous many-core architectures with hundreds to thousands of cores that need to deal with a diverse and rapid stream of dynamically changing applications with competing and conflicting demands and goals. Furthermore, MPSoCs face 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].

These new demands on MPSoC platforms with increased 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 critical 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.

To address these challenges, we present CyberPhysical-Systems-on-Chip (CPSoC) [3], a new class of sensor-actuator rich many-core computing platforms that intrinsically couples on-chip and cross-layer sensing and actuation to enable self-awareness (Figure 1). Unlike traditional MPSoC designs, the CPSoC paradigm co-designs the control, communication, and computing (C3) system that interacts with the physical environment in real-time to modify the system's behavior [4] so as to adaptively achieve desired objectives and Quality-of-Service (QoS). The CPSoC design paradigm enables self-awareness [5] (i.e., the ability of the system to observe its own internal and external behaviors such that it is capable of making judicious decisions) and (opportunistic) adaptation using the concept of cross-layer physical and virtual sensing and actuations applied across different layers of the hardware/software system stack (Fig. 1a). CPSoC deploys an adaptive & reflective middleware (a flexible hardware-software stack and interface between the application and OS layer) for the observe-decide-act (ODA) loop that controls the manifestations of computations (e.g., aging, overheating, parameter variability etc.) on the physical characteristics of the chip itself and the outside interacting environment (Fig. 1b).

ESWEEK'14, October 12 - 17 2014, New Delhi, India

ACM 978-1-4503-3051-0/14/10...$15.00.
http://dx.doi.org/10.1145/2656075.2661648

CPSoC coalesces the two traditionally disjoint aspects/abstractions of the cyber/information world and the underlying physical computing worlds into an unified abstraction of computing by using cross-layer virtual/physical sensing and actuations to enable a C3 centric self-aware computing platform. learning abilities of CPSoC provide a unified interface API for sensor and actuator fusion along with the ability to improve autonomy in system management.

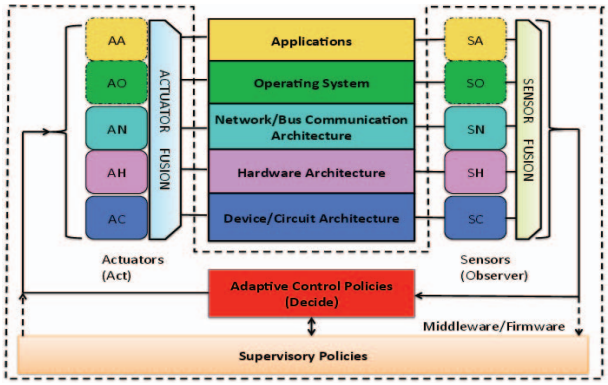
CPSoC Organization

The CPSoC architecture consist of a combination of sensor-actuator-rich computation platform supported by adaptive Networks-on-Chip (cNoC, sNoC), Introspective Sentient Units (ISU) [3], and an adaptive & reflective middleware to manage and control both the cyber/information and physical environment and characteristics of the chip as shown in Fig. 1b. The CPSoC architecture is broadly divided into several layers of abstractions, 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 a firmware or middleware as shown by the dotted line in Fig. 1a.

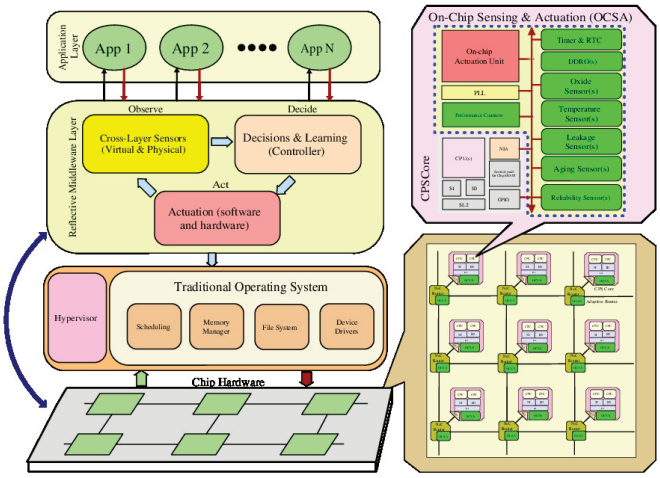
CPSoC distinctly differs from a traditional MPSoC in several ways. Traditional MPSoC paradigms lack the ability to sense the system states and behaviors across layers of system stack due to lack of architectural support; they are incapable of exploiting and exposing process and workload variations due to a lack of suitable abstractions at multiple layers. Furthermore, they sacrifice usable performance and energy potentials 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. The CPSoC framework overcomes

Table I: Adaptive Control Policies

Sensors Used	Diagnosis	Severity	Adaptive Control Policies	Selection of Actuation Mechanism
Delay Sensor and Temperature Sensor	Thermal induced short term delay	1	Change the frequency proportional to delay to avoid thermal emergency	DFS (frequency control)
Virtual workload sensor Temp. Sensor	Workload resulting in thermal and power emergency	2	Reduce power and computation resilient applications	loop perforation and approximate computation + DVFS
BTI sensor, delay sensor, Temp. Sensor	NBTI induced timing errors	3	Adjust frequency and voltage and migrate workloads	DVFS and Task Migrations
Aging/TDDB sensor, BTI and Temp. Sensor	Impending failure	4	Immediate rest for healing	clock gate and power gating of the block, Adaptive body biasing
BIST, Aging/TDDB BTI & Temp. Sensor	Permanent failure	5	Reconfigure avoid block/core	Reconfiguration



(a)



(b)

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 Middleware.

these limitations using some key ideas summarized below; our Technical Report[3] contains more details.

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 [3]) on the lower three layers as shown by the on-chip-sensing-and-actuation block (OCSN) in Fig. 1b and tabulated in Table II. On the other hand, **virtual sensing** [6] is a physical-sensorless sensing of immeasurable parameters using indirect computation. These can be viewed as **software sensors [6], [3]** that enable 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. 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**.

Similarly, we define **virtual actuations** [3](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 [3].

Table II: Virtual/Physical Sensing and Actuations Across Layers

Layers	Virtual/Physical Sensors	Virtual/Physical Actuators
Application	Workload; Power, Energy, Execution Time,	Loop perforation Algorithmic Choice
Operating System	System Utilization Peripheral States	Task Allocation, Scheduling, Migration, Duty Cycling
Network/Bus Communication	Bandwidth; Packet/Flit status; Channel Status, Congestion, Latency	Adaptive Routing Dynamic Bandwidth Allocation Ch. no and direction
Hardware Architecture	Cache misses, Miss rate; access rate; IPC, Throughput, ILP/MLP, Core asymmetry	Cache Sizing; Reconfiguration, Resource Provision Static/Dynamic Redundancy
Circuit/Device	Circuit Delay, Aging, leakage Temperature, oxide breakdown	DVFS, ABB, Multi-gate thresholding, Clock-gating

On-chip 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], [7]. 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. Fig. 2 shows a high-level abstraction of self-awareness in the CPSoC context, where the lower loop (in blue) performs simple adaptations by self-monitoring based based on fixed policies; and the upper loop (in red) achieves self-aware adaptation by building a reflexive model of the system and tuning, reconfiguring, and creating adaptive policies (e.g., Table I). Self-awareness can be used in several CPSoC contexts. For example,

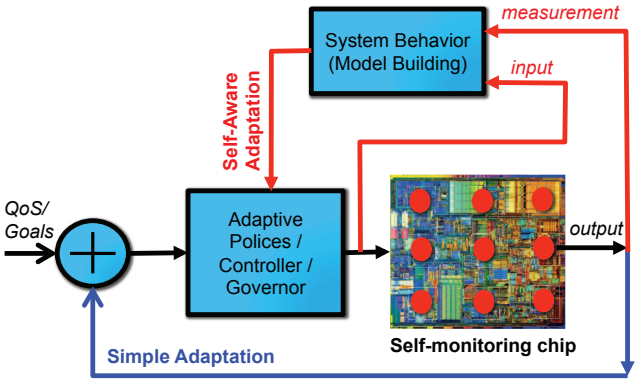


Figure 2: High-level abstraction CPSoC self-awareness.

on-chip self-awareness can be used to find or predict the the root cause of failures and respond just-in or ahead of time with adaptive policies as shown in Table I. For instance, in case of delay changes (either due to BTI or soft breakdown), the system should apply appropriate actuation mechanisms: for BTI-induced delay changes, both voltage scaling and frequency scaling can be applied; whereas for soft breakdown, the system should migrate the workload and avoid using the circuit with this impending failures. Similar efforts on self-awareness and adaptation are being pursued by other researchers including [8], [9], [10], [11].

Other key attributes of the self-aware CPSoC are adaptation within/across layers 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 [3] offers a completely different approach (extra dimensions of control) and solutions compared to a traditional MPSoC architecture. These cooperative and hierarchical control loops – e.g., the combination of traditional control loops together with virtual sensing enabled self-aware adaptation loops – effectively translates user goals or QoS into one or more design objectives [3].

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 [12]. We specifically used statistical and neural network approaches [13], [14] such that the model accuracy can be traded-off for model computational complexity. We used regression based linear predictors to build model of the system performance, power and energy consumption using the cross-layer events, hardware counters, and on-chip sensor data [15]. 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. [3].

Sample Applications

On-chip self-awareness with cross-layer virtual and physical sensing and actuations is a key enabling technology for efficient use of heterogeneous architectures, and applications with guarantee runtime system QoS (performance, reliability, power, thermal behavior) in a highly dynamic environment. Our Technical Report [3] 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 and power estimation by an order of magnitude [3]. In another example we show how virtual sensing enables an adaptive scheduler (an actuation mechanism) to exploit heterogenous architectures for energy efficiency; our preliminary result shows over a 30 % 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 [16].

Conclusion

We presented CPSoC, a self-aware 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 on-chip self-awareness (selective or opportunistic) adaptation using the concepts of cross-layer physical and virtual sensing and actuations. In [3] we illustrate CPSoC's potential for self-awareness and cross-layer adaptations using several examples and have developed an FPGA prototype to emulate a typical CPSoC.

Acknowledgment

This material is based on work supported by the NSF under awards CCF-1029783, CCF-1029030 (Variability Expedition) and CNS-1063596 (Cypress).

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.*, "Cyberphysical System-On-Chip (CPSoC): Sensor-actuator rich self-aware computational platform," University of California Irvine, Tech. Rep. CECS TR-13-06, May 2013.

[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.*, "Cross-layer virtual observers for embedded multiprocessor system-on-chip (MPSoC)," in *Proceedings of the 11th International Workshop on Adaptive and Reflective Middleware*, ser. ARM '12. New York, NY, USA: ACM, 2012, pp. 4:1–4:7.

[7] B. H. Cheng *et al.*, "Software engineering for self-adaptive systems: A research roadmap," in *Software engineering for self-adaptive systems*. Springer, 2009, pp. 1–26.

[8] H. Hoffmann *et al.*, "Self-aware computing in the Angstrom processor," in *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE*. 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] *Engineering Proprioception In Computing Systems*. Seventh Framework Programme (FP7), 2011. [Online]. Available: http://www.epics-project.eu/

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

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

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

[15] S. Sarma and N. Dutt, "Cross-layer design space exploration of heterogeneous multicore processors with predictive models." University of California Irvine, 2014, no. CECS TR 14-02.

[16] S.Sarma and N. Dutt, "FPGA emulation and prototyping of a CyberPhysical-System-On-Chip (CPSoC)," in *Proceedings of the International Symposium on Rapid System Prototyping (RSP'14)*, October 2014.