Center for Embedded Computer Systems
University of California, Irvine

Cyberphysical-System-On-Chip (CPSoC): A Self-Aware Design Paradigm with Cross-Layer Virtual Sensors and Actuators

Santanu Sarma, N, Dutt, N. Venkatasubramanian, A. Nicolau, P. Gupta†

Center for Embedded Computer Systems
University of California, Irvine
Irvine, CA 92697-2620, USA

†Department of Electrical Engineering
University of California Los Angeles
{santanus, dutt, nalini, nicolau}@uci.edu, puneet@ee.ucla.edu

CECS Technical Report No: CECS TR-13-06

May 30, 2013

Cyberphysical-System-On-Chip (CPSoC): A Self-Aware Design Paradigm with Cross-Layer Virtual Sensors and Actuators

S. Sarma, N. Dutt, N. Venkatasubramanian, A. Nicolau, P. Gupta*

***Abstract*— Cyberphysical 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-chips (CPSoC), a new class of sensor-actuator rich MPSoC, that additionally couple on-chip and cross-layer sensing and actuation to enable self-awareness within the observe-decide-act (ODA) paradigm. Unlike tradition MPSoC designs, CPSoC differs primarily on the co-design of control, communication, and computing system that interacts with the physical environment in real-time in order to modify their behavior so as to adaptively achieve certain objectives and Quality-of-Service (QoS) illustrated with a cogent use case.**

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

I. INTRODUCTION

CYBERPHYSICAL systems (CPSs)[1] are physical and engineered systems, whose operations are monitored, coordinated, controlled, and integrated by a computing, control, and communication core. This intimate coupling between the cyber and physical spheres manifests in many large-scale wide-area systems-of-systems (e.g. aerospace systems like spacecraft, transportation vehicles and intelligent highways, smart grids) into control-communication-computing (C3) codesign resulting in improved system efficiency and performance boost. Since such large computer based controlled systems interacts with the physical world, they must operate dependably, safely, securely, efficiently, and in real-time. As the multi-objective C3 codesign philosophy can address many challenges of large scale area-wide physical and cyber (information) systems as well as the physical manifestation of computation on the system, we propose and extend this design paradigm for nano-scale system, especially for emerging Multiprocessor System-On-Chip (MPSoC) containing thousands of cores.

MPSoC systems have been widely adopted for embedded signal processing and multimedia computing. Even though they have long history of use in embedded computing, recently they emerged as a viable platform for server/desktop computing with key architectural driving factors such as scalability, multitasking, programmability, real-time performance, and low-power operations. With increasing number of cores and large on-chip resources, the multi-core paradigm has increased the system complexity and added burden of parallelization for developers. Additionally, diverse applications with highly dynamic characteristics and unknown workload distributions (at compile time), that are unaware of resources and the states of system, lack the ability to economically use the available resources and hence fail to maintain a sustained predictable level of performance. While phenomenal scaling of semiconductors continue to enable Moories-law driven exponential growth and cost reduction, there is also a growing concern of increasing vulnerability to errors in the underlying hardware due to manufacturing process variability (in performance, reliability) due to deteriorating effects of material properties on power and die yields. Exponentially increasing power density and heating, creating drastic and harsh environments (e.g. hotspots), manifest as aging as well as wear-out phenomena (e.g. NBTI, HCI, TIDB, Electromigration etc.) that further increases susceptibility to errors in time over its usage life. The variability plagued semiconductor results in substantial fluctuations in critical device/circuit parameters over time [2] as well as in both with in-die and die-to-die variations with the immediate consequence of diminishing yield and reliability. To mass this variability and enhance yield and reliability, a large guard band with worst-case design margins (more than 50 % (3 sigma) [2]) is often used. Such worst-case designs lack the ability to extract untapped performance and energy potential in conservative designs to relax variation-induced guard-bands at different layers of the system stack.

To combat these overheads and exploit dynamic opportunities, we propose Cyberphysical-System-on-Chip (CPSoC), that comprise a combination of sensor-actuator-rich, self-aware computation core and adaptive communication fabric along with 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. With cross-layer sensor enabled self-awareness and the ability to discover potential present action and predict future actions as well as evaluate past actions and behaviors, these CPSoCs 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.

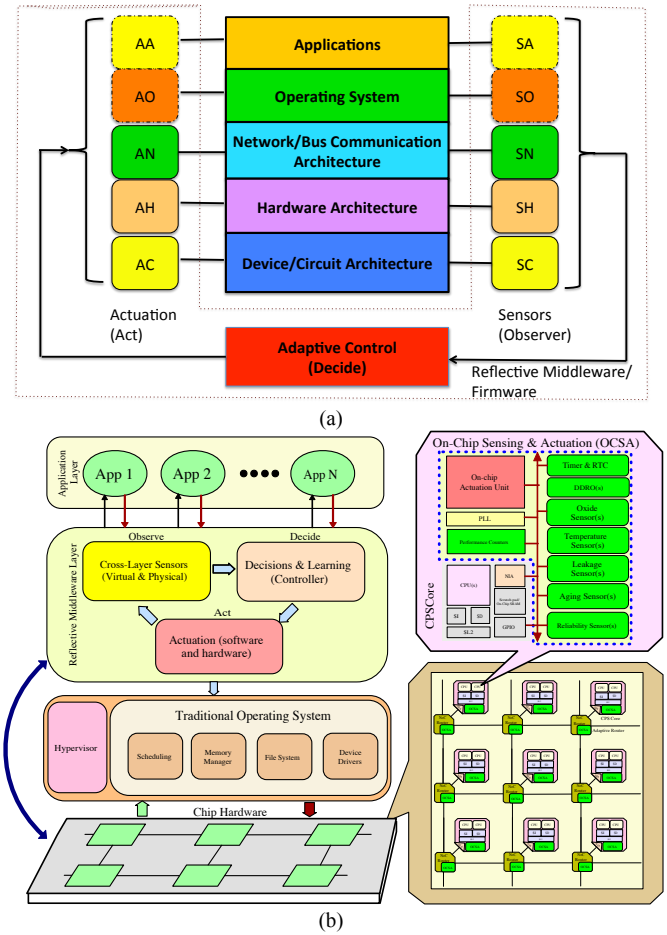

Figure 1: (a) Cross-layer virtual sensing and actuation at different layers of CPSoC (b) CPSoC architecture with adaptive Core, NoC, and the ODA Loop as Middleware.

II. ORGANIZATION OF CPSOC

The CPSoC architecture consist of a combination of sensor-actuator-rich computation core and an adaptive NoC along with an adaptive & reflective middleware to control, manage, and adapt both the internal and external environment and characteristics (physical and information) of the chip as shown in Fig. 1(a). 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 self-awareness within the observe-decide-act (ODA) paradigm. Unlike traditional MPSoC, each layer of the CPSoC can be made self-aware, although not mandatory, by a combination of software and physical sensors and actuators as shown in Fig. 1a. These layer specific feedback based closed loops are integrated into a flexible stack which can be implemented either as a firmware or a middleware as shown by the dotted line in Fig. 1b. The distinct differences and similarities of the proposed CPSoC with traditional MPSoC

Manuscript submitted on Dec 3, 2013. The authors are with University of California Irvine, and University of California Los Angeles*. The authors can be contacted at e-mail: {santanus, dutt, nalini, Nicolau}@ics.uci.edu and *{puneet@ee.ucla.edu}.

are tabulated in Table 1. Unlike traditional MPSoC, the CPSoC architecture is a computation-control-communication centric, rich in on-chip sensors and actuators (both physical and virtual), integrated into a feedback based closed-loop ODA middleware that enables adaptation in all the three dimension of computation, communication, and control in real-time.

Table 1: Traditional MPSoC Vs. CPSoC

Attributes	Traditional MPSoC	CPSoC
Objective	Efficient Resource Mangement for performance and power	Adaptive System /w Multi-Objective : (Reliability, Yield, Performance, Power/ Energy, Thermal, Security)
Design Paradigm	Computation-Communication Centric	Computation-Control-Communication Centric
Physical Sensing & Actuation	Nil or limited sensing ability	Sensor Rich /Many On-chip Sensors
Cross-layer Virtual Sensing & actuation	Not Applicable	Yes (at each layer)
Control System (ODA loop)	Open-loop response or actuation	Feedback based Closed-loop, Hierarchical, Distributed control
Operating System	Traditional Operating System	OS with Middleware (ODA control loop)
Communication Architecture	Traditional NoC	Adaptive NoC ((Bandwidth & Channel)
Resource Management	Ad-hoc or Limited	Full Resource-Awareness & Adaptive Resource Management
Scalability	User assisted parallel execution (mostly static)	Automatic parallelization of applications (dynamic)

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. These properties and features are briefly discussed in the subsequent sections.

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

The CPSoCs are sensor-actuator-rich MPSoC that include several on-chip physical sensors (such as Aging, DDRO, Oxide Breakdown, Leakage, Reliability (error signatures), Temperature, Performance Counters, as well as Voltage, Current, and Power sensors) on the lower three layers as shown by the on-chip-sensing-and-actuation (OCSA) block in Fig. 1b and tabulated in Table 2. On the other hand, **virtual sensing** is a physical sensor-less sensing of immeasurable parameters using a software approach as opposed to physical or hardware based approach. 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 an information processing technique, a computational approach, to enhance and/or add sensing capability, introduce sensing options, increase sensitivity, enable efficient sensor resource uses, and overcome physical placement and cost restrictions. When combined and vetted with different kinds of sensors, virtual sensing technique enables consensus to resolve faults and calibration errors while providing a test bed for multisensor fusion [3].

Similarly, we define **software actuations** (e.g. application duty cycling, algorithmic choice, check pointing), which are purely software interventions that can predictively influence system design objectives like performance, power, and reliability. Physical actuation mechanisms such as DVFS and adaptive body biasing (ABB), commonly adopted in modern chip to control performance, power, and parametric variations, when combined with software actuations adopted in the upper layers of the stack introduces the notion of **actuator fusion** and possible unified API (application program interface) in the CPSoC framework.

Table 2: 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

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. Self-aware 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. 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. Two key attribute of the self-aware CPSoC (unlike autonomous computing [4] and invasive computing [5]) are **adaptation of the layers and multiple cooperative ODA loops**. As an example, the unification of an adaptive computing core (with combined DVFS, ABB, and variable size Caches) along with a bandwidth adaptive NoC (with variable frequency/data rate, dynamic direction and channel control) offers a completely different approach (extra dimensions of control) and solutions in comparison to MPSoC architecture. These cooperative control loops translate the user goals or quality-of-service (QoS) into one or multiple design objectives.

C. *Predictive Modeling & Learning*

Predictive modeling and learning abilities of the system performance, environment, and external and internal disturbances provides the intelligence in the CPSoC paradigm. Use of coupling parameters (a metric that quantify the interactions between the layers) helps to develop application and cross-layer interaction models for nominal and abnormal operations. The learning abilities of CPSoC improve autonomy in managing the system, sensor and actuator fusion, and proactive resource discovery.

IV. CPSoC Applications, Use Cases, and Challenges

CPSoCs can find numerous applications in embedded, mobile, and general purpose computing including desktops and servers. CPSoCs with their ability to adapt to changing conditions as well as predictively control and achieve system goals (e.g. precise control of performance, power, reliability, temperature, and security) in comparison to just mere open-loop influence of system goals as in existing MPSoC, will outperform in many use cases. To illustrate with an example, a CPSoC as a cross-layer resilient systems, which distributes 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 variability plagued unreliable fabrication processes at significantly lower cost, while existing MPSoCs certainly lack or fail to achieve that. Besides, CPSoC as an emerging technology needs to address several challenges such as CPSoC simulation framework, hardware implementation and evaluation platforms, and test and characterization methods to verify and validate design issues relating to adaptive resource provisioning along with the in-field performance estimation models.

V. Summary

In this paper, we argue that future MPSoC operation presents a situation that can be compared to a large scale area-wide CPS and propose the approach of computation-control-communication codesign of CPS along with feedback control based closed-loop cross-layer adaptations into a new design framework called CPSoC to achieve multiple design objectives (such as improved yield, reliability, performance, thermal stability, and reduced power and energy consumption). The proposed design 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 decision) and (opportunistic) adaptation using the concept of cross-layer physical and virtual sensing and actuations applied across different layers of system stack. The closed loop control used for adaptation to dynamic variation, which is commonly known as observer-decide-act (ODA) loop in computing literature, is implemented as a middleware. The learning abilities of CPSoC provide a unified interface API for sensor and actuator fusion along with the abilities of improve autonomy in system management.

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] E. Lee, "Cyber physical systems: Design challenges," in *ISORC'08*, May 2008, pp. 363 –369.

[2] Borkar, S. et al., "Parameter variations and impact on circuits and microarchitecture," *in DAC* '03, 2003.

[3] R. Luo et al., “Multisensor fusion and integration: approaches, applications, and future research directions,” *Sensors Journal*, IEEE, vol. 2, no. 2, pp. 107 –119, apr 2002.

[4] J. Kephart et al., “The vision of autonomic computing,” *Computer*, vol. 36, no. 1, pp. 41 – 50, jan 2003.

[5] J. Teich et al., “Invasive computing: An overview,” in *Multiprocessor System-on-Chip – Hardware Design and Tool Integration*, M. Hübner and J. Becker, Eds. Springer, Berlin, Heidelberg, 2011, pp. 241–268.