Real Time Emulations: Foundation and Applications

Azalia Mirhoseini
ECE Department,
Rice University
azalia@rice.edu

Yousra Alkabani
CS Department,
Rice University
yousra@rice.edu

Farinaz Koushanfar
ECE Department.,
Rice University
farinaz@rice.edu

ABSTRACT

The mesoscopic properties of the state-of-the-art nanoscale devices and the emerging petascale computing and storage systems have one thing in common: they function at scales that are orders of magnitude larger than what can be simulated in standard industry and academic laboratory settings. For many decades, CAD and verification communities have successfully developed and used *emulations* to overcome and complement the shortcomings of simulations for logic verification. Physical prototyping and 2D/3D silicon emulation of the increasingly complex systems holds a significant promise to overcome the limitations of computer modeling and simulations. While the potential opportunities are plenty, much research is required for prototyping and building effective, relevant and indicative emulation platforms.

Categories and Subject Descriptors

B.8.2 [**Performance and Reliability**]: Performance Analysis and Design Aids; B.7.3 [**Integrated Circuits**]: Reliability and Testing

General Terms

Measurement, Performance

Keywords

Real-time emultion, Thermal modeling

1. INTRODUCTION AND MOTIVATION

At large vineyards, there are often several rose bushes planted throughout. The reason is not the romantic nature of vineyard owners or the aroma of flowers, but sheer practicality: roses suffer of very similar sense of diseases as grapes but are much more sensitive and less expensive. When vineyard owners recognize the sick roses, they get an early warning to plan the required defense for the nearby vines. Our idea is to apply this simple but clever and efficient paradigm to ICs. Vineyards are like the big expensive ICs, e.g., Intel multicore processor or even multi-million data centers, and roses are scaled down ICs subject to identical conditions. Scaling factors may differ by orders of magnitude and can be used to explore the accuracy and latency of prediction versus cost.

The advantages of such real-time emulation systems are plenty including low energy consumption, rapid prototyping, real-time response, flexible and easy input manipulation, and a high degree of predictability. We show two examples of how this rapid prototyping in 2D and 3D can be used for processor and data center thermal modeling. Granular thermal modeling is complex since heat transfer modeling includes solving many partial differential equations that are costly and time-consuming to compute. Furthermore, the simulators do not scale well and are inefficient for larger systems. Abstract models are likely to miss important interrelationships among the granular components at scale. Note that the benefits of emulations are not limited to heat modeling and include predictions and actuation [4]. Also, it has been demonstrated that integration of simulation and emulation can simulatenously provide fast execution and complete observability and controllability [5].

2. CASE STUDY-1: PROCESSOR THERMAL MODELING

The exponential increase in CMOS power densities is a result of continuous scaling that in turn creates thermal hotspots. The resultant high temperautres not only affect performance and power consumption, but also tamper the system reliability. The huge scale of processors today has made it almost impossible to accurately model on-chip temperature, cooling system, package and environment all together. Using numerical analysis such as finite element methods is time-consuming and cost inefficient and not applicable to larger designs.

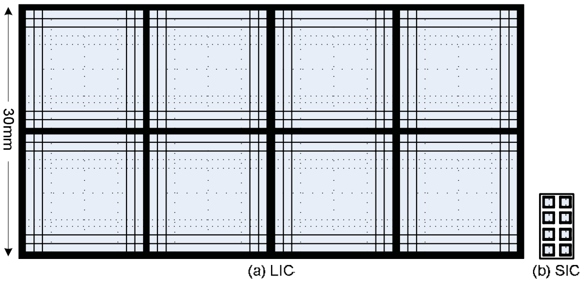

Figure 1: (a) Large IC (LIC) with 8 cores; (b) Emulated small IC (SIC).

Let us make an example: last year at ISSCC, Intel introduced its next-generation server processor Xeon which consists of eight 64b Nehalem cores and a shared L3 cache [1]. The processor is implemented in 45nm CMOS technology and has 2.3B transistors. We call this processor a large IC (LIC). One can partition the LIC into grids so that each

DAC'10, June 13–18, 2010, Anaheim, California, USA.
Copyright 2010 ACM ACM 978-1-4503-0002-5/10/06 ...$10.00

field consists of say, 100,000 transistors to construct a device that has 20600 fields. We place a temperature sensor in each field of LIC. Temperature sensors are also placed at each transistor on a small IC (SIC). Figure 1 demonstrates an example. We measure the temperature at each LIC and apply the input patterns required to bring the SIC's transistors to the same temperature. For example, transistors can serve as temperature sensors since we could indirectly measure their temperatures by measuring the speed, or even better, by finding their leakage which is exponentially sensitive to temperature changes.

Pseudocode 1 - Tasks Flow for LIC emulation

```
Instrument LIC;
Create an SIC measurement network;
Create a measurement schedule;
Generate input vector;
Find the heat (Q) on LIC;
Create the relevant predictions;
Analyze the decisions on SIC;
Select the best decision(s) for LIC;
```

The steps for building an emulation platform for the LIC are shown in Pseudocode 1. In the first step, we instrument specific places on the LIC by sensors or other means of measurement/monitoring. The corresponding SIC measurement network would be built for the SIC. A schedule for taking the measurements would be devised and input vectors to stimulate the sensors would be set. The values would be read on the LIC but predictions cannot be actuated on LIC since it is working. Instead, the different options are examined on the SIC in a very fast manner and the best decision is sent to the LIC afterwards.

Note that there are several ways to speed up emulation with respect to the LIC. Among them, the conceptually simplest and often the most effective is to generate heat k times faster and to simultaneously use materials with a k times higher heat transfer. Therefore, the emulations will be k times faster than the actual system and can be used for creating effective predictions and cooling decisions. The measurement schedule may be periodic or adaptive (temperature dependent), i.e., measuring more often when the negative impact of hotspots and high leakages are more likely. By using Fourier equations of heat transfer and sensor measurements one can calculate the dissipated energy at each field of the LIC grid. Next, one can use a simple linear program (LP) to generate similar temperatures at the corresponding SIC sensors. A full emulation system for heat did not exist earlier, but a system for HW/SW co-simulation/emulation on FPGA for thermal predictions was reported [2].

Since the details of the above LIC are not available, we quantify the benefits of emulation versus simulation using a linear solver (lp_solve) as an example. We use the benchmark C880 with only 303 gates. A number of input vectors are given to the chip and the output is the overall chip leakage. We solve a system of linear equations for finding each gate's leakage [6]. The number of measurements and the corresponding number of variables and LP runtime are shown in Table 1. We see that even for this small circuit, increasing the number of measurements quadratically increases the runtime. A linear growth in the circuit size would result in an exponential runtime increase.

# of measurements	256	512	1024	2048	4096
# of variables	1663	2943	5503	10623	20863
LP running time (s)	30.7	100.6	367.1	1424.3	8936.3

Table 1: Changing the number of measurements versus the LP running time for C880.

3. CASE STUDY-2: DATA CENTER THERMAL MODELING

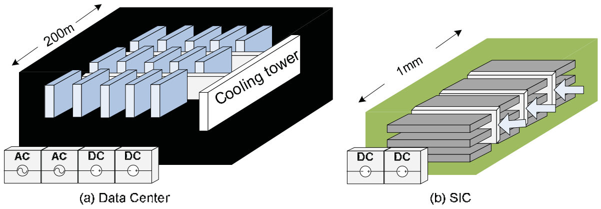

Figure 2: (a) Data center; (b) Small IC (SIC).

Our second example is a data center. The increasingly high power density of the existing and pending server generations requires detailed thermo-fluid analysis and solving a large system of partial differential equations. Additional interactions with the environment would make such simulations amazingly complex [3]. We propose using an emulated small IC with adjusted scales and ratios such that the differential equations for heat transfer on the SIC correspond to the principal components of the heat flow in the data center. Figure 2 shows an example where even the placement of cooling components and packages needs to be exactly emulated. Thus, if we equip the SIC with sensors because of the size and speed advantages, it would provide inexpensive and fast predictions that could be used for adaptive load balancing and server scheduling.

4. MERITS AND IMPACT

Emulation by 2D and 3D IC architectures holds a great promise in modeling and prediction of complex computational environments. The advantages of emulation have been previously shown for logic verification and for hw/sw simulations on FPGA. The benefits include fast simulation time, low energy consumption, flexible and easy manipulation, and predictability. Perhaps the greatest advantage of emulation versus simulation is the real-time nature of the emulation, making it *the only potential solution* for predicting and adaptive decision making in time-critical systems. While the opportunities are apparent, several open research questions are yet to be addressed.

5. REFERENCES

[1] Intel xeon processor 3400 series-based platforms. Product brief, 2009.

[2] D. Atienza, P. D. Valle, G. Paci, F. Poletti, L. Benini, G. D. Micheli, and J. Mendias. A fast HW/SW FPGA-based thermal emulation framework for multi-processor system-on-chip. In *DAC*, pages 618–623, 2006.

[3] A. Beitelmal and C. Patel. Thermo-fluids provisioning of a high performance high density data center. *Distributed and Parallel Databases*, 21(2-3):227–238, 2007.

[4] U. Khan, H. Owen, and J. Hughes. FPGA architectures for ASIC hardware emulators. In *ASIC*, pages 336–340, 1993.

[5] F. Koushanfar, D. Kirovski, and M. Potkonjak. Symbolic debugging scheme for optimized hardware and software. In *ICCAD*, pages 40–43, 2000.

[6] D. Shamsi, P. Boufounos, and F. Koushanfar. Noninvasive leakage power tomography of integrated circuits by compressive sensing. In *ISLPED*, pages 341–346, 2008.