

To appear in:

Proceedings of the 6th Workshop on Design, Modeling and Evaluation of Cyber-Physical Systems (CyPhy'17), Seoul, Republic of Korea, October 19, 2017

An Integrated Simulation Tool for Computer Architecture and Cyber-Physical Systems

Hokeun Kim¹, Armin Wasicek², and Edward A. Lee¹

¹ University of California, Berkeley,

hokeunkim@eecs.berkeley.edu, eal@eecs.berkeley.edu

² Technical University Vienna, armin@vmars.tuwien.ac.at

Abstract. Simulating computer architecture as a cyber-physical system has many potential use cases including simulation of side channels and software-in-the-loop modeling and simulation. This paper presents an integrated simulation tool using a computer architecture simulator, gem5 and Ptolemy II. As a case study of this tool, we build a power and thermal model for a DRAM using the proposed tool integration approach where architectural aspects are modeled in gem5 and physical aspects are modeled in Ptolemy II. We also demonstrate simulation results of power and temperature of a DRAM with software benchmarks.

Keywords: tool integration, architectural simulation, cyber-physical systems, DRAM thermal modeling

1 Introduction

Ptolemy II [17] is a powerful framework, where multiple models of computation can be explored for actor-based design of cyber-physical systems [8]. For many applications, it is important to model details of the computer architecture for a candidate design. Consequently, the Ptolemy II framework can significantly benefit from the integration of architecture models. In this paper, we propose a tool integration of the gem5 computer architecture simulator [2] and Ptolemy II. For a specific computer architecture, gem5 generates execution information that is used to build a more fine-grained system model in Ptolemy II.

This integration supports many usage scenarios including:

- Simulation of side channels: Side-channel attacks target primarily the physical implementation of a computer system. Unlike traditional computer systems, embedded systems are particularly vulnerable to this class of attacks, because they are often accessible in untrusted environments [11]. An example of a side channel attack is a cold boot attack on DRAM memories [10], where an attacker obtains a memory dump after a cold restart to read out sensitive information like cryptographic keys.
- Software-in-the-Loop modeling and simulation: In this scenario the embedded processor, sensors, and actuators are modeled with gem5 and the physical environment is modeled in Ptolemy II. This could support, for example, automated grading of embedded systems lab exercises in massively online open courses (MOOCs) [18]. For example, this would be useful for the

EECS149.1x cyber-physical systems [13] course at UC Berkeley. In the labs of this class, students develop programs for an iRobot.

We demonstrate the integration of both tools by modeling power and temperature of a DRAM in a computer architecture. To simulate behavior of the processor including memory accesses, we use the gem5 simulator. A Ptolemy II model performs power and thermal modeling, using discrete-event and continuous time models. Experimental results show how a computer architecture and workloads affect power and the temperature of a DRAM.

2 Related work

Currently, Ptolemy II offers the inclusion of an execution environment’s characteristics through a modeling method called Aspect-Oriented Modeling (AOM) [1]. For instance, an execution aspect can model execution times of a processor [5]. Metro II [7] provides an environment for platform-based design, where functional aspects and architectural aspects are modeled separately. Kim *et al.* [12] propose a tool integration approach where execution times on given architectures are modeled in SystemC, and integrated into Ptolemy II using Metro II. This approach has more flexibility in architectures, whereas our approach provides higher accuracy in architectural models.

The gem5 architecture simulator [2] is one of the most popular and widely used architecture simulators in academia and industry. It started as a merger of the General Execution-driven Multiprocessor Simulator (GEMS) [16] and the M5 simulator [3]. The gem5 simulator takes advantage of memory systems simulation features from GEMS, while it benefits from multiple ISAs and diverse CPU models supported by M5.

The gem5 simulator is object-oriented and based on the discrete-event model of computation. It also provides modular and interchangeable computer architecture components such as CPUs, memories, buses and interconnects. This architectural simulator is also flexible in terms of accuracy and simulation time providing multiple levels of accuracy, such as more accurate but slower simulation models and faster but less accurate simulation models [4].

A variety of approaches have been studied for power and thermal modeling of DRAMs. Lin *et al.* [14] suggest a model to compute power and the temperature of a DRAM based on throughput information, while Liu *et al.* [15] propose a power and thermal model based on RC circuit models. In this paper, we choose the model used by Lin *et al.* [14] Heat dissipation from DRAM devices is based on a device’s power which is almost proportional to memory throughput. Thus, knowing a memory’s read and write throughput (in GB/s), the temperature can be derived. In addition to the current flowing through the DRAM, its temperature is also affected by cooling air flow and the physical structure of DIMM (Dual In-line Memory Module). Fig. 1 depicts their model of DIMM structure and the temperature. The Advanced Memory Buffer (AMB) stores and transfers data between the different DRAM channels. The AMB is also a major source of heat in their model, therefore, they also consider the data throughput across

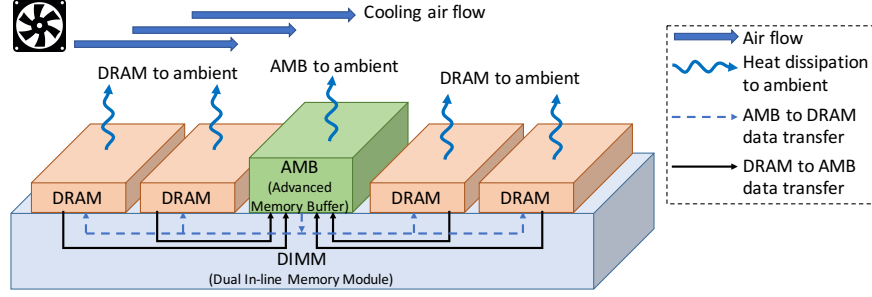


Fig. 1. Heat dissipation of DIMM. (Redrawn from the figure given by Lin *et al.* [14] and included here by permission of the publisher.)

DRAM channels. An ambient temperature refers to the temperature of the device’s environment and is in the most cases the room temperature.

There have been some approaches including DRAMPower [6] for simulating power and energy of a DRAM on a specific computer architecture. However, to the best of our knowledge, our case study is the first attempt to simulate heat and temperature of a DRAM by integrating a thermal model with a real-time computer architecture simulator, gem5.

3 Approach

In this section, we illustrate the integrated simulator design and the power and thermal model of a DRAM. For accessibility of our tool, we made all the working source code and experimental models available on-line. Configurations for the gem5 simulator and benchmark programs can be found at our GitHub repository (<https://github.com/gem5-ptolemy/gem5-ptolemy/>) and Ptolemy II can be downloaded from its homepage (<http://ptolemy.org>). An experimental model is included under “ptolemy/actor/lib/gem5/demo/DramThermalModel”, in Ptolemy II Version 11.0 (developer’s version).

3.1 Configuring the gem5 simulator

To integrate gem5 into Ptolemy II, we modify some configurations and source code of the latest stable version of the gem5 simulator. We modify some components so that they can generate information we need. We also configure the execution flow of the simulator so that it can run interactively by stopping and resuming the simulation when we want. In gem5, the main components such as CPUs and memory models are implemented in C++ for high performance, while connection between components and execution of components are implemented in Python so that the configurations are easily changed.

For power and thermal modeling, we modify C++ source codes associated with the DRAM memory controller model in gem5 to generate memory access

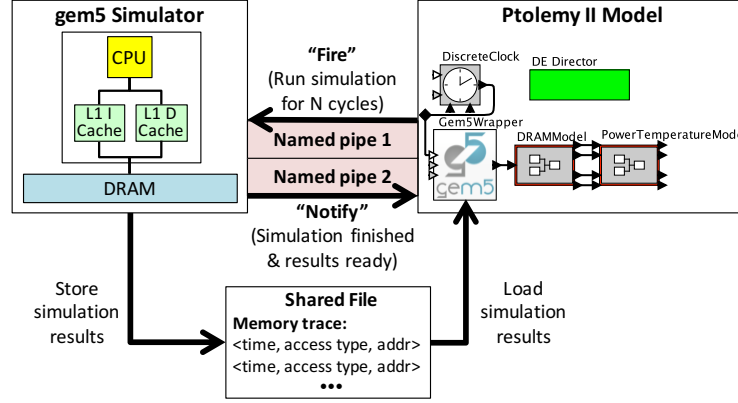


Fig. 2. An overview of gem5 and Ptolemy II integration

traces. We obtain extra information for power and thermal modeling by adding debug print functions defined in the gem5 simulator (*DPRINTF*) for recording memory access commands. For interactive simulation, we modify python scripts to call *Simulate* function iteratively with specified execution cycles.

3.2 Communication between gem5 and Ptolemy II

Fig. 2 illustrates an overview of gem5 and Ptolemy II integration. The gem5 simulator and a *Gem5Wrapper* actor in a Ptolemy II model interact with each other. The *Gem5Wrapper* actor is a Java actor in Ptolemy II model. It communicates with gem5 through named pipes and a shared file. When the *Gem5Wrapper* is initialized in the Ptolemy II, it fires the gem5 simulator by writing on the named pipe where the gem5 simulator is blocked on read. The *Gem5Wrapper* actor also gets blocked on read on another named pipe in its *fire()* method. The gem5 simulator runs for the specified number of cycles. While running, the gem5 simulator records execution information such as a memory trace on the shared file. When the simulation is finished, gem5 notifies *Gem5Wrapper* by writing on another named pipe where *Gem5Wrapper* is blocked. Then, *Gem5Wrapper* resumes in its *fire()* and reads execution information from the shared file. *Gem5Wrapper* fires gem5 again in its *postfire()* and this pattern is repeated.

Simulation results are transferred to *Gem5Wrapper* through the shared file and used for DRAM power and thermal modeling. The results include DRAM memory access events. Each access events is composed of the time when the event occurred, an access type (e.g. read/write) and a memory address (e.g. bank and channel numbers).

3.3 DRAM behavioral model in Ptolemy II

The Ptolemy II model for the overall system consists of two main parts. DRAM's behavior is modeled in the first part, and power and the temperature of the

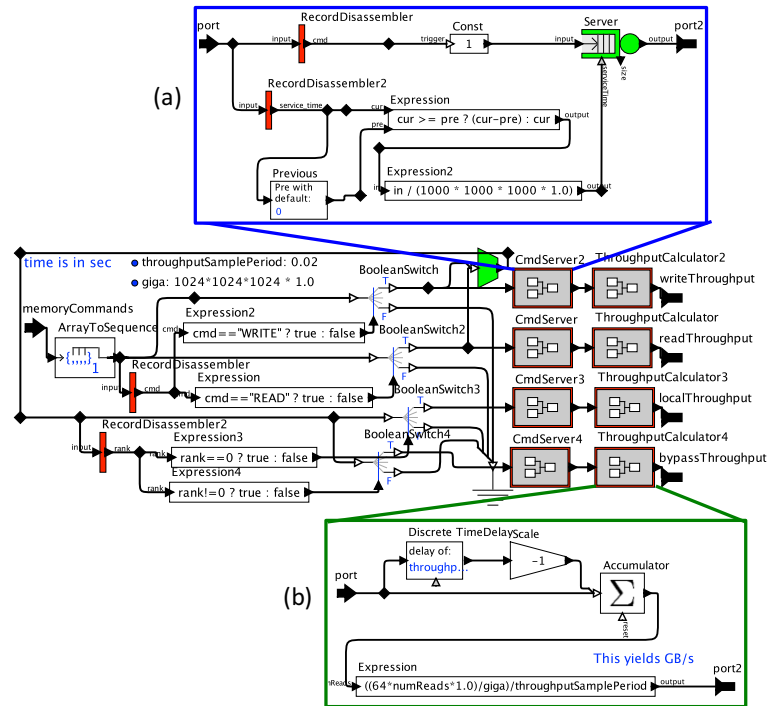


Fig. 3. Ptolemy II DRAM model overview (*DRAMModel*). (a) command server actor (*CmdServer*) (b) throughput calculator (*ThroughputCalculator*)

DRAM is modeled in the second part. In the Ptolemy II model, Gem5Wrapper is triggered periodically by a DiscreteClock actor. When Gem5Wrapper receives simulation results from gem5, it stores result data as an array type defined in Ptolemy II. Then, Gem5Wrapper sends the data array to a composite actor called *DRAMModel* shown in the middle of Fig. 3.

The data array is decomposed into a sequence of memory access events inside the *DRAMModel*, and a sequence of memory access events are sent to the *CmdServer* actor in Fig. 3 (a). Each memory access event becomes a discrete event in *CmdServer* and is sent to the *ThroughputCalculator* actor in Fig. 3 (b), where the throughput results are computed. The types of throughput results include *read*, *write*, *local* (to a local DRAM channel) and *bypass* (to non-local DRAM channels). The throughput results are used for AMB/DRAM power estimation in the section below.

Power and the temperature of a DRAM is modeled in the second part of the Ptolemy II model within a composite actor called *PowerTemperatureModel* described in Fig. 4. This actor runs in the continuous-time domain, sampling

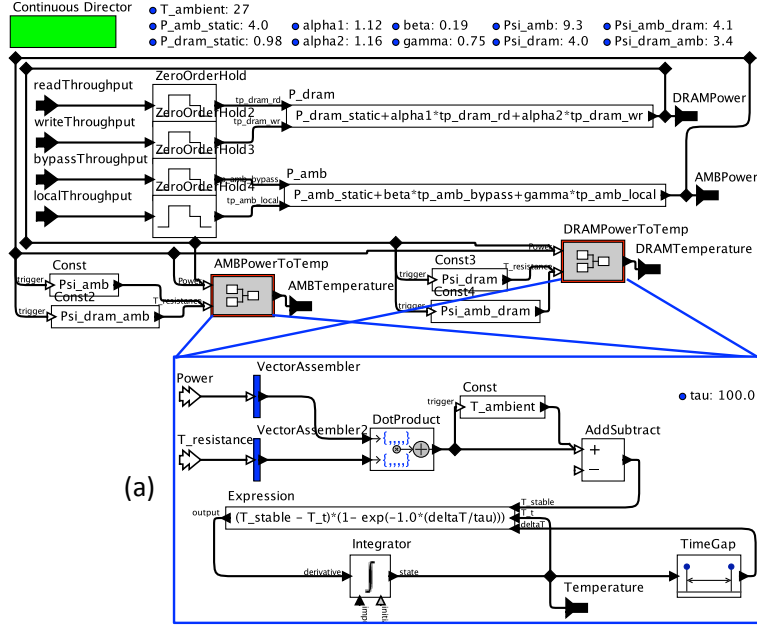


Fig. 4. Ptolemy II DRAM power and thermal model overview (*PowerTemperature-Model* actor). (a) *AMB/DRAMPowerToTemp* actor that estimates the temperature of an AMB/DRAM based on its power

throughput information from input ports. Power models for CMOS devices usually combine the static power of the device with its dynamic power. Static power is the power when transistors are not in the process of switching. Dynamic power occurs during switching operations:

$$P_{device} = P_{DRAM_static} + P_{DRAM_dynamic} \quad (1)$$

To compute power in the DRAM and AMB, we use the following equations introduced by Lin *et al.* [14] P_{DRAM} and P_{AMB} are total power in the DRAM and AMB, respectively. P_{DRAM_static} and P_{AMB_idle} denote static power of DRAM and AMB. α_1 , α_2 , β , and γ are coefficients measured in [14], and their units are Watt/(GB/s).

$$P_{DRAM} = P_{DRAM_static} + \alpha_1 \times Throughput_{read} + \alpha_2 \times Throughput_{write} \quad (2)$$

$$P_{AMB} = P_{AMB_idle} + \beta \times Throughput_{Bypass} + \gamma \times Throughput_{Local} \quad (3)$$

The power computed above is used to estimate temperatures in the AMB and DRAM. The composite actor shown in Fig. 4 (a) implements this thermal estimation. We use following equations introduced by Lin *et al.* [14] to calculate

temperatures of the AMB and DRAM. T_{AMB} and T_{DRAM} are stable temperatures of the AMB and DRAM, respectively. T_A stands for the ambient temperature explained in section 2. Parameters Ψ_{AMB} and Ψ_{DRAM} denote the thermal resistances of the AMB and DRAM. The thermal resistances are measured as the ratio of the change of the stable temperature over the change of power. The thermal resistances from AMB to DRAM and from DRAM to AMB are denoted as Ψ_{AMB_DRAM} and Ψ_{DRAM_AMB} , respectively.

$$T_{AMB} = T_A + P_{AMB} \times \Psi_{AMB} + P_{DRAM} \times \Psi_{DRAM_AMB} \quad (4)$$

$$T_{DRAM} = T_A + P_{AMB} \times \Psi_{AMB_DRAM} + P_{DRAM} \times \Psi_{DRAM} \quad (5)$$

The equation expressing the relation between the stable temperature and the actual temperature is as follows. $T(t)$ is the actual temperature at t and Δt denotes each time step. We use the τ value, which is the time for the temperature difference to be reduced to $1/e$, as measured in [14]. This equation is realized with the Integrator actor in Ptolemy II as illustrated in Fig. 4 (a).

$$T(t + \Delta t) - T(t) = (T_{stable} - T(t))(1 - e^{-\frac{\Delta t}{\tau}}) \quad (6)$$

4 Experiments and results

4.1 Experimental setup

The architectural configurations used for experiments are as follows. The CPU was based on ARM ISA, and the type of the CPU was *TimingSimpleCPU* defined in the gem5 simulator, which stalls on every load memory access. The clock rate of both the CPU and the overall system was 1GHz. The type of off-chip DRAM memory was DDR3 SDRAM with a data rate of 1600MHz and a bus width of 16 bits. We assumed the program and data exist in the DRAM before starting the execution. The size of cache blocks was 64 bytes.

We chose MiBench [9] as the benchmark for our experiments. Among MiBench programs executable in the gem5, top 5 programs with the highest memory intensity were chosen for our experiments. We defined the memory intensity as the number of memory accesses per instruction, and the memory intensity was computed by running each program for one million cycles in gem5. The benchmark programs used for our experiments are listed in Table 1.

4.2 Power and temperature results

Table 2 shows average power and the peak temperature of the DRAM and AMB for different cache configurations. The results were obtained by running the gem5 simulator and Ptolemy II DRAM power and thermal model together for 0.1 seconds in simulated time (100 million cycles). For this experiment, *cjpeg-large* in MiBench was used as a software workload. The temperature is expressed in

Table 1. List of benchmark programs used for example workloads

MiBench programs	Writes	Reads	Total instructions executed	Memory intensity (%)
consumer/cjpeg_large	6,183	74,966	1,000,000	8.11
security/rijndael_large	2,558	68,458	1,000,000	7.10
consumer/typeset_small	12,843	55,963	1,000,000	6.88
network/dijkstra_large	4,942	59,198	1,000,000	6.41
network/patricia_large	4,255	49,198	1,000,000	5.35

Table 2. Power and temperature results for different cache configurations for the workload *cjpeg_large*

Cache size options (KB)		Average power (mW)		Maximum temperature increase ($10^{-6} \text{ }^{\circ}\text{C}$)	
L1	L2	DRAM	AMB	DRAM	AMB
16	N/A	1,057	4,027	2.67	6.05
32	N/A	1,023	4,011	2.63	5.93
64	N/A	1,000	4,008	2.46	5.51
32	128	996	4,006	2.17	4.86
32	256	995	4,006	1.99	4.47

the difference between the highest temperature and the ambient temperature. We assumed the processor has two level-1 (L1) caches, each for instructions and data. Bigger caches led to less cache misses, and thus less DRAM accesses. Since the level-2 (L2) cache absorbed off-chip traffic from L1 caches, they reduced DRAM memory accesses. Therefore, we could see decrease in DRAM power and the peak temperature in the results shown in Table 2.

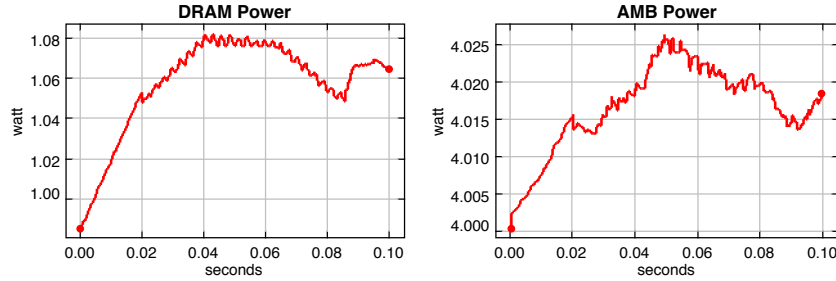
**Fig. 5.** DRAM and AMB power results in graphs for *cjpeg_large* with 16KB L1 caches

Fig. 5 illustrates DRAM and AMB power graphs for the workload *cjpeg_large* with 16KB L1 caches. *cjpeg_large* loads a 786KB Portable Pixel Map (PPM) file for a raw image and compresses it to a JPEG format. We could see DRAM power was affected by total read/write throughput while AMB power was related to cross-channel accesses. The power consumption for both DRAM and AMB steadily increases as the benchmark program initializes until around 0.02 seconds. The program shows heavy power consumption between 0.02 and 0.063 seconds while actively loading and compressing the raw image, followed by a

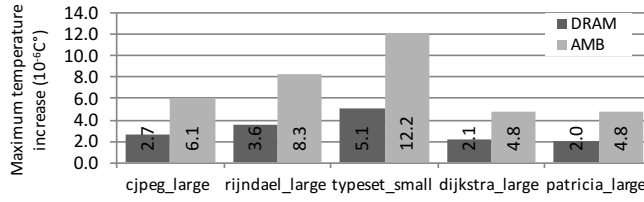


Fig. 6. Temperature results for different software workloads

slight decrease in power consumption after 0.063 seconds as the program wraps up. The total simulation time for 100 million cycles (0.1 seconds in simulated time) was ranging from 89 seconds (*cjpeg_large*) to 320 seconds (*patricia_large*) on a MacBook Pro laptop with 2.2GHz Intel Core i7 and 16GB DRAM.

Different workloads also led to change in the peak DRAM temperatures as illustrated in Fig. 6. For this experiment, we used 16KB L1 caches without an L2 cache. The results suggest that other aspects of workloads as well as the memory intensity can affect thermal behaviors of DRAMs. Specifically, *rijndael_large* and *typeset_small* had higher peak temperatures although they had lower memory intensity than *cjpeg_large*. This was because they had higher bypass throughput, which caused higher power in the AMB, thus resulting in higher peak temperatures both in the AMB and DRAM. Moreover, *typeset_small* showed the highest write throughput, also leading to the highest peak temperatures.

5 Conclusions

In this paper, we integrate the widely used gem5 architecture simulator into Ptolemy II to have a more accurate architectural model in Ptolemy II. Effectiveness and usefulness of this integration is demonstrated by constructing a power and thermal model of a DRAM in computer architecture. Execution information such as memory accesses on given architectures are modeled in gem5 whereas the power and temperature of a DRAM are modeled in the continuous time domain in Ptolemy II. The constructed model is used for experiments of simulating different architectural configurations and software workloads.

As future work, we can apply the proposed approach to more applications, for example, the two use cases suggested in section 1. Another possible extension is to use gem5 for aspect-oriented modeling in Ptolemy II. Specifically, execution aspect parameters such as execution time can be obtained dynamically through gem5 simulation for higher accuracy.

Acknowledgments

This work was supported in part by the TerraSwarm Research Center, one of six centers supported by the STARnet phase of the Focus Center Research Program (FCRP) a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

References

1. Akkaya, I., Derler, P., Emoto, S., Lee, E.A.: Systems engineering for industrial cyber-physical systems using aspects. *Proc. of the IEEE* 104(5), 997–1012 (Mar 2016)
2. Binkert, N., et al.: The gem5 simulator. *SIGARCH Comput. Archit. News* 39(2), 1–7 (Aug 2011)
3. Binkert, N., Dreslinski, R., Hsu, L., Lim, K., Saidi, A., Reinhardt, S.: The M5 Simulator: Modeling Networked Systems. *IEEE Micro* 26(4), 52–60 (Jul 2006)
4. Butko, A., Garibotti, R., Ost, L., Sassatelli, G.: Accuracy evaluation of GEM5 simulator system. In: 2012 7th Int'l Workshop on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC). pp. 1–7 (Jul 2012)
5. Cardoso, J., Derler, P., Eidson, J.C., Lee, E.A., Matic, S., Zhao, Y., Zou, J.: Modeling timed systems. In: Ptolemaeus, C. (ed.) *System Design, Modeling, and Simulation using Ptolemy II*. Ptolemy.org (2014)
6. Chandrasekar, K., Weis, C., Li, Y., Goossens, S., Jung, M., Naji, O., Akesson, B., Wehn, N., Goossens, K.: DRAMPower: Open-source DRAM power & energy estimation tool (2012), <http://www.drampower.info>
7. Davare, A., Densmore, D., Guo, L., Passerone, R., Sangiovanni-Vincentelli, A.L., Simalatsar, A., Zhu, Q.: Metro II: a design environment for cyber-physical systems. *ACM Trans. Embed. Comput. Syst.* 12(1s), 49:1–49:31 (Mar 2013)
8. Derler, P., Lee, E.A., Vincentelli, A.S.: Modeling cyber-physical systems. *Proceedings of the IEEE* 100(1), 13–28 (Jan 2012)
9. Guthaus, M., Ringenberg, J., Ernst, D., Austin, T., Mudge, T., Brown, R.: MiBench: a free, commercially representative embedded benchmark suite. In: *IEEE Int'l Workshop on Workload Characterization, WWC-4*. pp. 3–14 (Dec 2001)
10. Halderman, J.A., et al.: Lest we remember: Cold-boot attacks on encryption keys. *Commun. ACM* 52(5), 91–98 (May 2009)
11. Hwang, D.D., Schaumont, P., Tiri, K., Verbaauwhede, I.: *Securing embedded systems*. IEEE Computer Society (2006)
12. Kim, H., Guo, L., Lee, E.A., Sangiovanni-Vincentelli, A.: A tool integration approach for architectural exploration of aircraft electric power systems. In: 2013 IEEE 1st Int'l Conf. on Cyber-Physical Systems, Networks, and Applications (CP-SNA). pp. 38–43 (Aug 2013)
13. Lee, E.A., Seshia, S., Jensen, J.: EECS149.1x, Cyber-Physical Systems (May 2014), EECS, University of California, Berkeley, <https://www.edx.org/course/cyber-physical-systems-uc-berkeleyx-eeecs149-1x>
14. Lin, J., Zheng, H., Zhu, Z., David, H., Zhang, Z.: Thermal modeling and management of DRAM memory systems. In: *Proc. of the 34th Annual Int'l Symp. on Computer Architecture*. pp. 312–322. ISCA '07, ACM, New York, NY, USA (2007)
15. Liu, S., Leung, B., Neckar, A., Memik, S., Memik, G., Hardavellas, N.: Hardware/software techniques for DRAM thermal management. In: *IEEE 17th Int'l Symp. on High Performance Comput. Archit. (HPCA)*. pp. 515–525 (Feb 2011)
16. Martin, M.M.K., et al.: Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset. *SIGARCH Comput. Archit. News* 33(4), 9299 (Nov 2005)
17. Ptolemaeus, C. (ed.): *System Design, Modeling, and Simulation using Ptolemy II*. Ptolemy.org (2014), <http://ptolemy.org/books/Systems>
18. Skiba, D.J.: Disruption in higher education: Massively open online courses (MOOCs). *Nursing Education Perspectives* 33(6), 416–417 (Nov 2012)