System-Level Power Estimation Methodology using Cycle- and Bit-Accurate TLM

Miltos D. Grammatikakis[1 3], Stratos Politis[1], Jean-Pierre Schoellkopf[2] and Constantine Papadas[1]
[1] ISD S.A, Kifisias 32, Athens GR-15165, {mdgramma, politis, papadas}@isd.gr
[2] ASTUS-SA, 31 rue Gustave Eiffel, F-38000 GRENOBLE, jean-pierre.schoellkopf@astus-sa.com
[3] General Sciences Dept, TEI of Crete, Stavromenos, Heraklion, Crete, GR-71004

Abstract—**We propose a new system-level methodology for relative power estimation, which is independent of register transfer level models. Our methodology monitors the number of bit transitions for all input/output gate signals on a bit- and cycle-accurate SystemC virtual platform model. For absolute results and reliable technology-based predictions of system power and speed (e.g. in future 32/22nm technology nodes and variations), relative metrics can be multiplied with bit energy coefficients provided by semiconductor technology datasheets and device models.**

Keywords-design methodology; multicore; network-on-chip; SystemC; system-on-chip; TLM; component;

I. Introduction

Register Transfer Level (RTL) has been traditionally considered as the entry point in power-efficient design flow. However, the increase in SoC integration and design complexity implies that we must raise the abstraction level and address power consumption and performance at system-level [1]. In fact, both performance and power estimation can be examined at different abstraction levels, from the most abstract, such as system-level or behavioral, down to the most concrete transistor-level. This choice affects *power estimation accuracy*, *simulation efficiency*, and *time to develop the model*. Although low-level power estimation is more accurate, the scope for design space exploration is reduced, since significant changes are very costly to implement and reuse principles can not be easily applied.

Transactional Level Modeling (TLM) has recently gained momentum in the multicore SoC design industry, as the new entry point. Cycle-accurate TLM is fit for design space exploration, since it combines near RTL accuracy, with high simulation speed; it is usually an order of magnitude faster to implement and several orders faster to simulate than a corresponding RTL model. In addition, open TLM standards reduce vendor dependency and increase IP reuse, they are supported by many IP providers.

Existing system-level power estimation tools include power state models integrated within instruction set simulators, such as Hype, Jouletrack, SoftExplorer, Simunic, Avalance, Lajolo and Bulldast's PowerChecker [3]. In addition, advanced system-level design frameworks, such as ChipVision's Orinoco and Synopsys' Innovator introduce proprietary data flow abstractions for power estimation. Finally, power instrumentation tools, such as BlueSpec, PowerSC and Power-Kernel, build object-oriented C++ classes of hardware modules on top of the SystemC library. Among these three tools, only Power-Kernel (PK) is open source. PK allows simple introduction of a SystemC power macro model (focusing on signal activity) at RT-level.

II. New System-Level Power Estimation

All existing system-level tools discussed above rely on the use of spreadsheets or back annotation from RTL models which is often not available during system-level design space exploration.

To avoid this restriction, we propose a new system-level power estimation methodology based on simulating an early stage, bit- and cycle-accurate transaction-level model and computing the number of bit transitions at the interface of all subsystems. Notice that bit transitions (and no-transitions) reflect *relative power dissipation*, while traditional high-level synthesis can help achieve more detailed, accurate power estimation. *Absolute power estimation* is often not as important as relative accuracy, since qualitative power metrics correlate with the final implementation, ensuring that early design decisions are appropriate. More specifically, during each clock cycle, we compute the number of transactions and bit transitions experienced by each system component, i.e. interface signals, memory/registers/FIFOs, finite state machines (FSMs) and data path. While transactions are abstract macro-operations applied to a component, bit transitions refer to individual bit changes that take place as a consequence of these macro-operations. Thus, for the register and interface components, we compute the hamming distance of the current versus the previous data payload. Similarly, for control and data path components we consider bit changes at the input and output signals, as well as in representative data path computations depending on the macro-operations performed on internal system variables.

III. Multimedia Multicore SoC Virtual Platform

The above methodology is currently been applied to early-stage design space exploration of a NoC-based homogeneous multicore SoC prototype infrastructure, addressing the need for power-efficient multimedia applications requiring large-scale data parallel (SIMD) array processing. The implemented, bit- and cycle-accurate SystemC VP follows transaction-level modeling (TLM), without providing full-scale implementation details [2]. It connects together abstract processing models (PEs) and external storage elements (SEs) through network adapters to a hypercube-based on-chip interconnect.

The NoC itself is configured with parameterizable high performance, low latency routers. A network adapter

.978-3-9810801-7-9/DATE11/©2011 EDAA

component converts variable size transport-layer packets to 128-bit network-layer packets (with 32-bit headers) and vice-versa. Packet transmission supports recovery from loss via timeout & retransmission. The network adapter component integrates a configurable general interconnect controller which simultaneously supports both shared memory and message passing communication primitives, allowing a hybrid programming model to reap benefits of both approaches, e.g. for innovative multimedia application environments. More specifically, each transport-layer send/receive operation between PEs is converted to one or more fixed size network-layer packet(s) and vice-versa. Variations of transport-layer packets are considered (parameters omitted for convenience).

- `synchronous/asynchronous blocking send, and`
- `synchronous blocking/nonblocking receive.`

In addition, each transport-layer remote shared memory read/write is converted to one or more fixed size network-layer packet(s) and vice-versa. This includes strong and weak read-modify-write operations on shared data; the memory barrier synchronizes a set of memory modules accessed by a list of PEs. Notice that memory address mapping onto network node IDs is provided.

- `synchronous/asynchronous blocking read,`
- `synchronous/asynchronous blocking write,`
- `synchronous/asynchronous blocking copy,`
- `get & modify       (Get&Add/Get&Set)`
- `compare & set`
- `load linked/set conditional`
- `memory barrier`

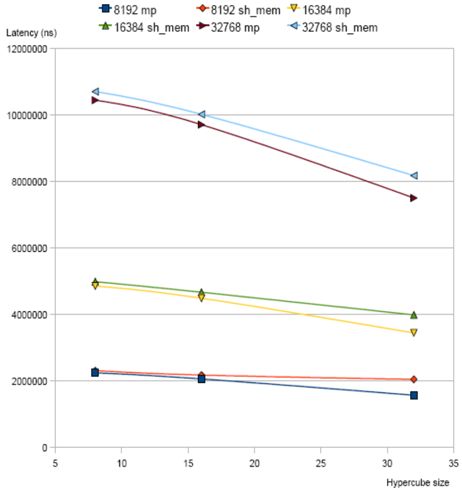

Figure 1. Latency of bitonic sorting for array sizes: 8192, 16384 and 32768

Using this VP, we examined shared memory (sh_mem) and message passing (mp) for ping-pong and parallel sorting kernels, as well as testbench emulations of software transactional memory locking using application traces obtained from Stanford STAMP vacation benchmark [4]. Preliminary simulation results (e.g. see Figure 1) indicate that power-efficiency and linear speedup can be obtained for both shared memory and message passing implementations, especially for compute-bound applications. Full-scale VP experimentation relates to a proprietary hypercube-based fault tolerant NoC and a Thales multicore SoC architecture with custom processor tiles supporting physical shared memory.

IV. Conclusions and Future Extensions

It is generally complex to accurately model performance and power consumption at system-level due to unavailable architecture/algorithm implementation characteristics and technology parameters coming from RTL synthesis. The proposed power estimation methodology focuses on state transitions and can be applied for early analysis and power optimization of any system-level IP, since it does not rely on an equivalent RT level implementation.

Using this methodology, performance and software power can be addressed early in the design, thus reaping important benefits from product differentiation. End users are able to explore innovative algorithmic, architectural and technology-related features to perform accurate and efficient variation-aware power analysis by focusing on the distribution of several key system metrics.

The methodology can be extended to provide absolute technology-related performance, power and variability prediction for future technology generations by annotating roadmap data (obtained from technology datasheets and CMOS device models, e.g. MASTAR, GTX or bsim3) directly into the bit- and cycle-accurate SystemC models. In this case, power consumption metrics are estimated as the product of the number of bit transitions and transactions for each component with an appropriate normalized bit energy technology coefficient which can be obtained by applying regression on semiconductor technology data, e.g. for a NAND2 gate.

Acknowledgment

Partially supported by EU ARTEMIS/SCALOPES – Ref n° 100029 (duration: 2009-2010), b) ENIAC/MODERN - Ref n° 120003 (duration 2009-2011), and Greek funding authorities.

References

[1] L. Benini and G. De Micheli, "System-level power optimization: techniques and tools", *ACM Trans. Design Automation Electr. Syst.*, **5 (2)**, 2000, pp. 115—192.

[2] L. Cai and D. Gajski, "Transaction level modeling in system level design," *Tech. Rep. 03-10*, CECS, UC-Irvine, 2003.

[3] M.D. Grammatikakis and M. Coppola, "Power-aware multicore SoC and NoC design", (invited chapter), in *Multiprocessor System-on-Chip: Current Trends and Future*, Ed. M. Huebner, Chapter 3, Springer, 30 pages, 2010.

[4] C.C. Minh, J.W. Chung, C. Kozyrakis and K. Olukotun, "STAMP: Stanford Transactional Applications for Multi-Processing", *in Proc. IEEE Int. Symp. on Workload Characterization*, 2008. Also see http://stamp.stanford.edu