

System Level Power Modeling and Simulation of High-End Industrial Network-on-Chip

Andrea Bona Vittorio Zaccaria Roberto Zafalon

STMicroelectronics, Advanced System Technology - R&I
Via Olivetti 2, 20041 Agrate Brianza, Italy

Abstract

Today's System on Chip (SoC) technology can achieve unprecedented computing speed that is shifting the IC design bottleneck from computation capacity to communication bandwidth and flexibility. This paper presents an innovative methodology for automatically generating the energy models of a versatile and parametric on-chip communication IP (STBus). Eventually, those models are linked to a standard SystemC simulator, running at BCA and TLM abstraction level. To make the system power simulation fast and effective, we enhanced the STBus class library with a new set of power profiling features ("Power API"), allowing to perform power analysis either statically (i.e.: total avg. power) or at simulation runtime (i.e.: dynamic profiling). In addition to random patterns, our methodology has been extensively benchmarked with the high-level SystemC simulation of a real world multi-processor platform (MP-ARM). It consists of four ARM7TDMI processors accessing a number of peripheral targets (including several banks of SRAMs, Interrupt's slaves and ROMs) through the STBus communication infrastructure. A remarkable amount of SW layers are executed on top of MP-ARM platform, including a distributed real-time operating system (RTEMS) and a set of multi-tasking DSP applications. The power analysis of the benchmark platform proves to be effective and highly correlated, with an average error of 9% and a RMS of 0.015 mW vs. the reference (i.e. gate level) power figures.

Keywords: Network-on-Chip power analysis, communication based low power design, system-level energy optimization.

1. Introduction

Embedded computing systems are on the way to provide a number of new services that will arguably become common practice in the next few years. The most important of these are (i) multimedia (audio/video streaming) capabilities in personal communicators, (ii) huge computing power (especially from clusters of processors) and storage size, (iii) high rate accessibility from mobile terminals.

Today's System on Chip (SoC) technology can achieve unprecedented computing speed that is shifting the IC design bottleneck from computation capacity to communication bandwidth and flexibility.

- SoC's designers need to leverage on pre-validated components and IPs such as processor cores, controllers and memory arrays. Design methodology will further support IP

re-use in a plug-and-play fashion, including buses and hierarchical interconnection infrastructures.

- SoCs will have to provide a functionally correct, reliable operation under data uncertainty and noisy signaling. The on-chip physical interconnection will be a limiting factor for both performance and energy consumption, also because the demand for component interfaces will steadily scale-up in size and complexity.

In this paper, we will present a thorough methodology for automatically building the energy model of a Network-on-Chip (NoC) IP at the BCA/Transaction level, in order to allow power profiling of an entire platform since the very early stages of the system design, often when only a software model of the system does exist.

The paper is organized as follows: Section 2 introduces a short background on Network-on-Chip. Section 3 illustrates the STBus versatile interconnect IP as an industrial example of NoC infrastructure. Section 4 introduces the overall NoC power characterization and estimation framework while Section 5 goes into details about our NoC's energy model. Section 6 presents the Design of Experiment policy and Section 7 reports a significant set of figures about the model validation and the experimental results, including a real-world platform simulation case.

2. Background

Although the main concepts and the terminology of **Network-on-Chip** design has been introduced quite recently [1][2][3], both the industrial and research communities have been starting to realize the strategic importance of shifting the design paradigm of high-end digital IC from a deterministic, wire-based interconnection of individual blocks and IPs, to a thorough communication-based design methodology [4][7][9], aiming to face with data packetization and non-deterministic communication protocols in next generation's SoCs.

With the advent of 90nm and 65nm CMOS technology, the challenges to fix the Network-on-Chip (NoC) issue "by design", will need:

- To provide a functionally-correct, reliable operation of the interconnected components by exploiting appropriate network infrastructure and protocols, i.e. interconnections to be intended as "on chip micro-network" [5][6][7], which is an adaptation of the OSI protocol stack [18].
- To achieve a fluid "flexibility vs. energy-efficiency" system exploration, allowing an effective network centric power management [8][11][12]. Unlike computation energy in fact,

the energy for global communication does not scale down with technology shrinking [3][4]. This makes energy more and more dominant in communications.

Reaching those goals will be crucial to the whole semiconductor industry in the next future, in order to face with the escalating range of signal integrity and physical wiring issues, who are making the target IC reliability harder and exponentially expensive to achieve. As of today, there is a limited availability of tools able to consistently support this emerging design methodology. Indeed, some high level models for functional/performance system simulations (i.e. Bus Cycle Accurate and Transaction) are smoothly coming up [13] across the design community. However, power predictability of NoCs still remains an open issue.

Although NoC's power estimation has been partially addressed in [10], its low level modeling (i.e. gate and device level) and the extremely slow simulation (i.e. 1000 cycle/s) makes it definitely unsuitable to face with any system level SW/HW exploration task, which might easily need for simulation speeds larger than 100 Kcycle/s.

3. On chip network: STBus Interconnect

STBus is versatile, high performances interconnect IP allowing to specify the communication infrastructure in terms of protocol, interface and parametric architectures [14][15]. It comes with an automated environment (*STBus generation kit*) suitable to support the whole design flow, starting from the system-level parametric network specification, all the way down to the mapped design and global interconnect floor-plan [16]. The protocol modes supported by STBus are compliant with VSIA standard [19]. In fact, they can scale up from *Peripheral*, to *Basic* and to *Advanced* mode, conventionally named Type-1, Type-2 and Type-3, respectively. In this work, we focus on the last 2 protocols (i.e. Type-2 and Type-3) since they better fit with the high demanding communication resources required by modern SoCs. More specifically, Type-2 supports *pipelined split transactions*, where each transaction is composed by a pair of *send* and *receive packets* (packet: a sequence of atomic messages called *cells*). On top of the above features, Type-3 allows to manage *out-of-order* packet delivery. The datapath's width can range between 32, 64 and 128 bits.

The STBus architecture builds upon the *node* module, configurable switch fabrics who can be instantiated multiple times to create a hierarchical interconnect structure. The topology of the switch fabric can be selected by choosing the number of resources dedicated to the request and the response packets; for example a shared bus interconnect has only 1 request and 1 response resources at a time, while a full crossbar has as many request and response resources as the number of initiators and targets connected to the node. Eventually, *type converter* and *size converter* modules can be adopted to interface heterogeneous network domains working under different protocols (i.e. Type-1, 2 and 3) and/or different datapath widths.

4. Enabling Energy Exploration for NoC

When dealing with multi-processors embedded systems, characterized by tens of masters and slaves connected through a complex communication infrastructure, energy estimation and optimization become of utmost importance. As a matter of fact, although more effective than traditional buses, NoCs are expected to make a relevant contribution to the area budget, due to the growing complexity of packet routing and

transaction management policies affecting the interconnection's control-path, and to the switch fabric in charge of supporting the high speed data packet delivery.

Such a complexity has a cost in terms of energy consumption that should be traded-off with the performance benefits. Network structures achieving lower packet's congestion (i.e. higher performance), are usually characterized by larger data-path complexity in terms of number of simultaneous routing resources available for packet broadcasting. For example, a shared bus communication node can be slower (i.e. higher congestion), yet less power consuming than a full crossbar switch-box, or, the *slot-reservation* arbitration policy may overcome the limitation of *Time Division Multiple Access (TDMA)* policy in case of asymmetric workloads in a multi-processors platform. These questions need to account for energy metric during the design exploration in order to find out the optimal platform configuration to meet the performance constraints at minimum energy.

Exploration and optimization for SoC design are rapidly evolving towards the analysis of abstract description models that mimic the main operations of the system under analysis, including speed and power behavior. According to the SystemC modeling scenario depicted in [13], the abstraction levels that can be used to model the function/power/performance of a communication-based system are the Functional untimed level, the Transaction level (TLM), the Bus Cycle Accurate level (BCA) and the Pin Accurate – Cycle Accurate level (PA-CA). In short, while the Functional level does not give any insight on the timing figures of the system, the Transaction level only gives coarse time hints (e.g. total read/write time slot), with no structural information on actual wires or pins. The BCA level achieves cycle-accurate timing estimates, yet functionally accurate at the boundaries, while the PA-CA goes down to a clock cycle timing with structural pin-accurate description, at the expense of a much slower simulation. In this paper we introduce a consistent methodology for automatic energy model's building to fit most of the above abstraction levels (i.e. Transaction, BCA, PA-CA), suitable to support the NoC's power estimation since the very early stages of the design exploration, when only a C/C++ model of the system is usually available.

Eventually, the system simulation (developed in SystemC, in our case) will rely on high-level profiling statistics to figure out the energy cost, by means of an appropriate library of energy views and a dedicated API. In the following, we will explain how the STBus energy models are based on a set of parametric, analytic equations that are individually accessed by the simulator to compute the eventual energy figures (either statically or at simulation runtime).

4.1 Energy Characterization Flow

The energy macro-model of the whole STBus interconnection is partitioned into sub-components, corresponding to each micro-architectural block of the interconnection fabrics that are *node*, *type-converter* and *size-converter*. For sake of simplicity, in this paper we will show the results of the *node* component. However, the same automatic flow is currently applied to all of the components of STBus architecture. The proposed model relies on the bus utilization rate, i.e. the number of cells traveling through the bus, as well as on the interconnection topology (i.e. the number of masters/targets), which need to be pre-characterized, once and

for all, through an accurate gate-level simulation for each target technology. The power characterization flow consists of 4 major steps depicted in Figure 1.

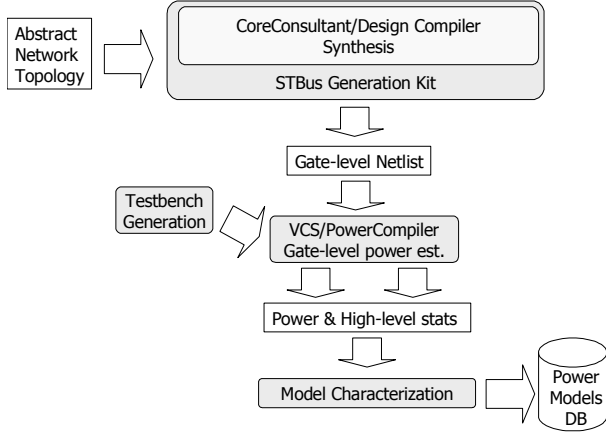


Figure 1. STBus power characterization flow.

As already mentioned in section 3, the *STBus generation kit* allows the designer to automatically synthesize a gate-level netlist starting from a system-level parametric network specification. This is done by inferring the corresponding RTL code and, then, synthesizing all the way down to the mapped design [16]. Thus, an extensive set of gate-level power simulations (VCS/PowerCompiler) is launched within a Testbench Generation suite, specifically tuned to fulfill the many requirements imposed by the STBus protocols and, at the same time, to sensitize the node under a wide range of traffic workloads. Specifically, the test-benches can be configured in terms of average latency per master request and slave response and type of operations to be performed on the bus. The operations can be splitted in two categories (load and store) as they can play with different operand sizes (from 1 to 32 bytes).

The last step of the flow in Figure 1 is the Model Characterization, where each of the coefficients is computed to fit the high-level model (ref. to next section 5 for details). The final models (one for each component and target technology) are stored into a centralized *Power Model Database*. Sure enough, the choice of experiments, the length of each simulation and the test-benches adopted during the characterization campaign are crucial knobs to be optimized before running the characterization flow, by means of a suitable *Design of Experiments* (DoE: see section 6).

4.2 Hooking the Energy Models to the System Simulator

The *STBus Generation Kit* supports the generation, among the others, of the SystemC model of each component, ready to be plugged into the target SystemC simulation platform. The current release of the *STBus Generation Kit* is compliant with BCA SystemC v2.0 descriptions [13]. In evolution, the support for TLM is planned soon, according to the STBus roadmap. The overall SystemC power estimation flow is outlined in Figure 2. To make the system simulation environment fast an effective, an ad-hoc API has been developed (*SystemC Power API*), together with a consistent library of functions allowing to enhance the basic SystemC capabilities with a *power profiling* feature, providing power analysis either statically (i.e.: total avg. power) or at simulation runtime (i.e.: dynamic profiling). The latter is done by computing a moving average on a given time window (e.g. ten clock cycles).

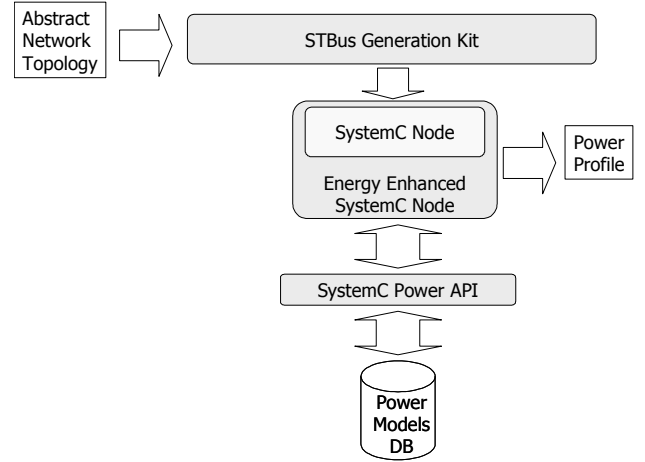


Figure 2. Power enhanced SystemC simulation.

Deriving the SystemC node classes and hooking them up to the specific SystemC *Power API* achieve the energy enhancement. As a matter of fact, *energy-enhanced* SystemC nodes provide an extremely fast procedural interface to retrieve each set of model's coefficients out from the *power model database* as well as to handle the power analysis during the actual SystemC simulation.

5. STBus Energy Model

In this section, we introduce the power model for a generic configuration $\mathbf{n}$ of a node. The configuration of an STBus node identifies a specific instance out from the design space S :

$$S = \{ \mathbf{n} \mid \mathbf{n} = \langle i, t, rqr, rpr, p, C_L, dps, Type \rangle \} \quad (1)$$

where i is the number of initiators, t is the number of targets, rqr is the number of request resources, rpr is the number of response resources, p is the type of arbitration policy (STBus has 7 arbitration policies), C_L is the output pin capacitance (range: $C_{Lmin} = 4$ Standard Loads ; $C_{Lmax} = 1$ pF), dps is the data-path size (range: 32, 64 and 128 bit) and $Type$ is the protocol mode (Type-2 and 3, in this case).

Based on an extensive experimental background, we recognize a fairly linear relationship between node energy and the rate of sent and received packet cells across all of the interconnection node's ports. Such a behavior matches with a set of random configuration samples across the entire design space and it has been confirmed during the model validation phase (see section 7).

The energy model for a generic configuration $\mathbf{n}$ of the STBus node is the following:

$$E(\mathbf{n}) = P(\mathbf{n}) \cdot C \cdot T_{clk} \quad (2)$$

where $P(\mathbf{n})$ is the average power consumption of the node during a simulation of C clock cycles, with a clock period of T_{clk} . The power consumption $P(\mathbf{n})$ is a linear combination of three contributions, according to the following equation:

$$P(\mathbf{n}) = B(\mathbf{n}) + P_{sent}(\mathbf{n}) \cdot \frac{r_s}{C} + P_{rec}(\mathbf{n}) \cdot \frac{r_r}{C} \quad (3)$$

where $B(\mathbf{n})$ is the average base cost depending on the specific configuration $\mathbf{n}$ of the node, $P_{sent}(\mathbf{n})$ is the additive power cost due to cell sent from the masters to the slaves and r_s is the total number of cells sent, $P_{rec}(\mathbf{n})$ is the power cost due to each packet cells received by the masters, r_r is the total number of cells received by the masters and C is the number of clock

cycles. In essence, the power model characterization consists in determining the value of the coefficients $B(n)$, $P_{sent}(n)$ and $P_{req}(n)$ for each specific configuration n of the node. As formerly mentioned, this task is performed by means of a polynomial regression over the set of experiments given by DoE (see section 6). So far, linear regression is successfully used to build the coefficients of the model but higher order models can be also used if accuracy has to be increased. The experimental setup is generated with the goal of properly stressing r_s and r_r over the whole range of variation. The total avg. switching activity coming out from the Test-benches is kept at 0.5. As far as the interconnection capacitive load “ C_L ” is concerned, our model supports a linear interpolation between C_{Lmin} and C_{Lmax} in order to provide a quite accurate estimation of the switching power under the specific load of the current instance.

From a global viewpoint, the characterization campaign of STBus across the whole design space may easily become a huge computing task. The computational effort to power characterize STBus is similar or even larger than the characterization of an industrial size ASIC library. The whole comprehensive STBus design-space, in fact, would lead to more than $3.4 \cdot 10^5$ individual configurations to be characterized (i.e. RTL synthesis + gate-level simulation + power measure). Such a number comes out from the product of all the possible combinations of the STBus design subspaces (i.e. 8 initiators, 8 targets, 8 request and 8 response resources, 7 arbitration policies, 2 load capacitances, 3 data path sizes and 2 types of protocols). Running an exhaustive characterization is far to be feasible in a reasonable time, even by leveraging on distributed computers. We decided to adopt a *response surface method* approach to solve this problem. In this approach, only a selected set of configurations are synthesized and characterized, making the remaining set of coefficients derivable by accessing an appropriate set of models (either analytic or look-up table) obtained through response surface methods. Although this approach may lead to some inaccuracy with the energy estimation process, the global accuracy can be taken well under control while allowing a remarkable drop in characterization effort.

6. Optimal Design of Experiments - DoE

The fundamental theory on statistical design has been largely consolidated during the last twenty years or so, for a wide variety of applications [20]. In this context, the Design of Experiments is based on the *convergence analysis* of some peculiar quality figures such as *average power* and *average prediction error*. Converging on the *average power* figure let us to identify the minimum length necessary for each simulation, by considering when the power consumption gets close to a steady value, given an arbitrary acceptance threshold (see the power-time curve of Figure 3). On the other hand, the minimum number of experiments (i.e.: synthesis + simulations) needed to safely probe the design space and characterize the specific model, strongly depends on the target accuracy (i.e. max prediction error) as well as on the acceptable characterization effort. The regression analysis to fit the model’s coefficients is performed on the raw characterization data. Therefore, the QoR can be analytically measured by the prediction correlation coefficients (R and R^2) and the Root Mean Square error (RMS). Eventually, the minimum number of experiments is identified by considering both the RMS

steady state and the absolute error, over a set of significant benchmarks.

6.1 Convergence analysis: average power

The minimum simulations length necessary for the model characterization has to be identified through a *convergence analysis*. While the minimum simulations length of the testbench would not affect the actual power consumption, it is crucial to make sure that the circuit under analysis can always reach a steady state functional activity before measuring the avg. power consumption. To identify the correct simulation length, we minimized a cost-function that is a product of the simulation time and a measure of the derivative of the power consumption. The cost function is the following:

$$C(t) = t^2 \frac{\Delta P(t)}{P(t)} \quad (4)$$

where t is the simulation time, $P(t)$ is the power consumption measured at time t and $\Delta P(t)$ is the difference between $P(t)$ and $P(t-1)$. Figure 3 shows the average behavior of the cost function for all the possible configurations of shared bus, Type-3, 32 bit width nodes. As can be seen, after 5000 ns the difference between power values does not pay for the increased simulation time. Thus, 5000 [ns] has been selected as the simulation length for all the characterization experiments.

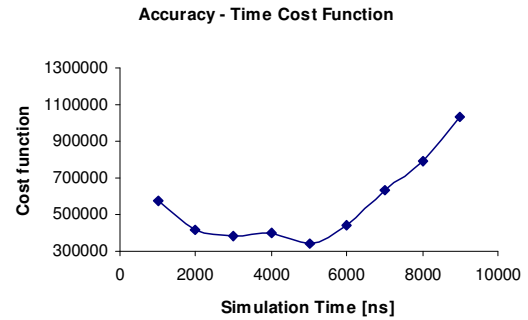


Figure 3. Avg. power vs. Simulation-time convergence analysis for a given STBus node’s configuration.

The derivative (i.e. differential ratio) has been sampled every 1000 ns and, then, normalized to the related power values in order to give a percentage variation.

6.2 Convergence analysis: model accuracy

According to previous section 5, the model’s coefficients are resulting from the polynomial regression over a given set of experiments. Those experiments are generated according to the DoE’s policy, by stochastically changing the number of data packets sent/transmitted across the bus and the operation modes. The goal is to find out the minimum number of experiments necessary to meet the required accuracy. Given a set of representative STBus nodes, we perform their characterization with an increasing number i of experiments. For each set of i calibration experiments, the Root Mean Square error (RMS) is evaluated. Figure 4 shows that, for $i > 160$, the RMS for all the configurations of the design space gets close to the respective asymptotic values, with a maximum value bounded to 0.012 mW. The minimum number of experiments to proceed with the characterization of the STBus nodes has been defined accordingly.

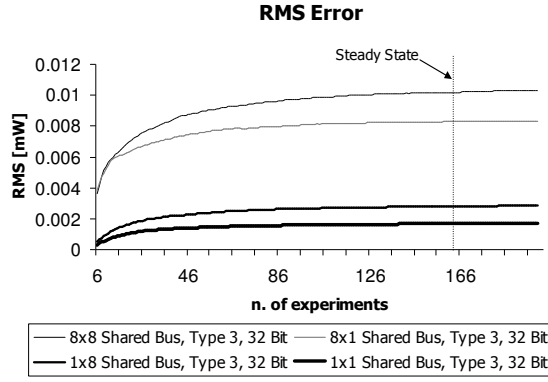


Figure 4. Power model's RMS Error vs. Number of calibration experiments, under four different initiators/targets configurations.

7. STBus Power Model Validation and Experimental Results

We present hereafter the results obtained from the validation phase of the proposed power macro modeling. In addition to the validation carried out by applying an extensive set of synthetic test-benches, we extended the test to a realistic application, featuring mission-mode SystemC simulations of a multi-processors platform. All the characterization and experimental results presented in this paper are targeted to STMicroelectronics's HCMOS9 ASIC library, featuring 8 metal layers and 0.13 μm MOS channel length, operating at 1.2V nominal supply voltage.

7.1 Random pattern validation

We carried out a synthetic validation by applying a uniform set of stochastically generated Verilog test-benches, similar to those used during the calibration phase (section 6.2). In Figure 5 we illustrate the scatter plot between the model estimation and the reference power measurement (coming from detailed gate-level power analysis). The **average error** is 1% with a correlation **R** of 96%.

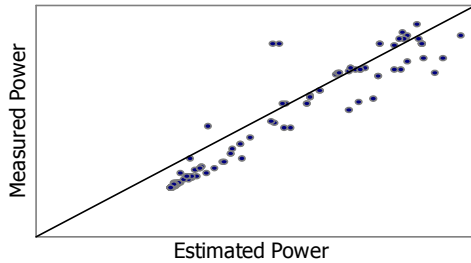


Figure 5. Scatter plot of Measured vs. Estimated power consumption, for a set of synthetic benchmarks.

7.2 Mission mode validation through SystemC co-simulation

To extensively validate our methodology into a real world simulation platform, we decided to assess the robustness of the power model by correlating the power estimation coming from a high-level SystemC simulation with respect to the gate-level power measure of the synthesized STBus node subject to the input stream generated at runtime by SystemC. The multi-processor platform is outlined in Figure 6. The architecture consists of four ARM7TDMI processors accessing a number of

targets (including several banks of SRAMs, Interrupt's slaves and ROMs) through the STBus communication infrastructure, configured as a 4 initiators, 3 targets, Type-3, shared bus, 32 bit, fixed priority request arbitration policy, dynamic priority response arbitration policy.

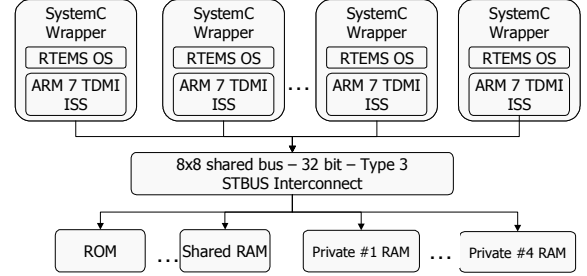


Figure 6. Multi-processor platform including four ARM7TDMI processors connected through STBus.

Indeed, a remarkable amount of SW layers are intended to be executed on top of this HW platform, including a distributed real-time operating system who runs on each individual processor (RTEMS), and a class of multi-tasking DSP applications, featuring intensive integer matrix computations.

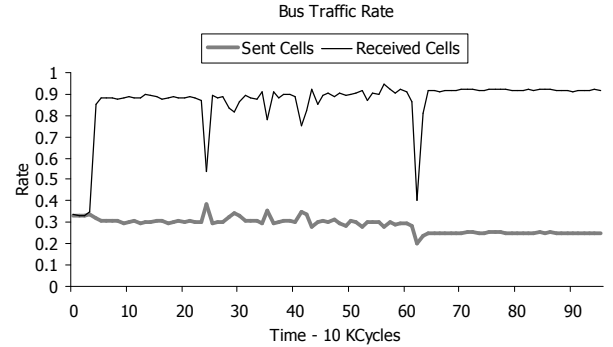


Figure 7. Data packet rate monitored across the STBus on the target multiprocessor platform.

As far as the simulation framework is concerned, each processor's ISS has been encapsulated with a SystemC wrapper, in charge of managing the interface protocol with the STBus communication node. The whole SW benchmark has total execution duration of 1 Million clock cycles, including the RT-OS booth strap (the initial 200 Kcycles) and the execution of the DSP application SW. In Figure 7, the data cell's statistics (i.e. rate of cells sent/received per unit of time) across the STBus is reported. The overall SystemC/Verilog co-simulation flow is depicted in Figure 8.

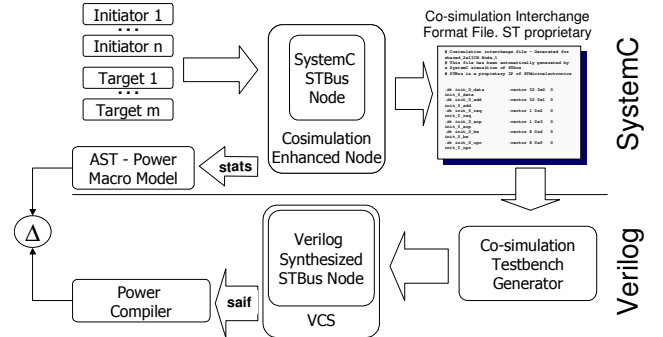


Figure 8. SystemC/Verilog co-simulation flow.

During the SystemC simulation, initiators and targets generate a trace of "mission mode" transactions, monitored through a

specific feature of the STBus node. In fact, the node has been enhanced in order to gather the full signals stream out from the SystemC simulation session. The eventual trace file carries comprehensive print-on-change informations, sampled on a clock cycle basis. The co-simulation file is then applied to drive the gate-level Verilog simulation (VCS [16]) and, then, feeding the detailed power analysis of the mapped netlist (PowerCompiler [16]).

In Figure 9 we compare the power predicted by SystemC when running the system simulation (Pestimated) vs. the reference power measured by Power Compiler at gate-level (Pmeasured). Please notice that absolute power numbers are hidden for technology confidentiality.

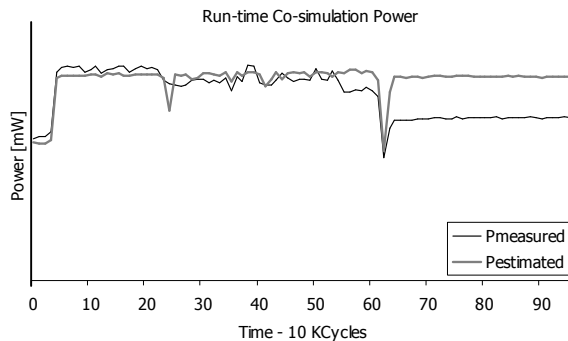


Figure 9. Estimated vs. Measured average power in STBus.

The system level estimation proves to be highly correlated to the reference power figure with an average error of 9% and a RMS of 0.015 mW. Note, however, that in the last 200K cycles the power consumption is overestimated. This is due to a high number of cells transiting through the node with near-zero switching activity.

8. Conclusion

An innovative methodology for automatically generating the energy models of a versatile and parametric on-chip communication infrastructure (STBus) has been presented in this paper. The methodology aggressively targets correlated power estimation with efficient SystemC simulation, running at BCA and TLM abstraction level. Among other synthetic benchmarks, the NoC's power models validation has been extensively addressing the high-level SystemC simulation of a real world multi-processor platform (MP-ARM), which includes four ARM7TDMI processors accessing a number of peripheral targets (including several banks of SRAMs, Interrupt's slaves and ROMs) through the STBus communication infrastructure. All the characterization and experimental results presented in this paper are targeted to STMicroelectronics's HCMOS9 ASIC library, featuring 8 metal layers and 0.13 μm MOS channel length, operating at 1.2V nominal supply voltage.

The synthetic validation between the model estimation and the reference power figures (i.e. gate-level power measure of the synthesized NoC) shows an **average error** of 1% and correlation **R** of 96%. The power analysis of the MP-ARM benchmark proves to be highly effective and correlated, with an **average error** of 9% and a **RMS** of 0.015 mW vs. the reference power.

Acknowledgements

The authors are grateful to dr. C.Pistritto and his CMG/OCCS team in Catania, for their valuable and synergic support to achieve the leap enhancement of making the STBus IP offering truly "power aware".

9. References

- [1] J.Duato, S.Yalamanchili, L. Ni, "Interconnection Networks: an Engineering Approach", IEEE Computer Society Press, 1997.
- [2] K. Lahiri, S.Dey et al., "Efficient Exploration of the SOC Communication Architecture Design Space", Proc. of ICCAD-2000, Nov. 2000, S.Jose, USA.
- [3] W. Dally, B. Toles, "Route Packets, not Wires: On-Chip Interconnection Network", Proceedings of 38th DAC 2001, June 2001, Las Vegas, USA.
- [4] A. Sangiovanni Vincentelli, J. Rabaey, K. Keutzer et al., "Addressing the System-on-a-Chip Interconnect Woes Through Communication-Based Design", Proceedings of 38th DAC 2001, June 2001, Las Vegas, USA.
- [5] F. Karim, A. Nguyen et al., "On Chip Communication Architecture for OC-768 Network Processors", Proceedings of 38th DAC 2001, June 2001, Las Vegas, USA.
- [6] K. Lahiri, S.Dey et al., "Evaluation of the Traffic Performance Characteristics of System-on-Chip Communication Architectures", Proc. 14th Int'l Conference on VLSI Design 2001, Los Alamitos, USA.
- [7] L. Benini, G. De Micheli, "Network on Chip: A New SoC Paradigm", IEEE Computer, January 2002.
- [8] T. Ye, L. Benini, G. De Micheli, "Analysis of power consumption on switch fabrics in network routers", Proceedings of 39th DAC 2002, June 2002, New Orleans, USA.
- [9] S. Kumar et al., "A network on chip architecture and design methodology", International Symposium on VLSI 2002.
- [10] H.-S. Wang, X. Zhu, L.-S. Peh, and S. Malik, "Orion: A Power-Performance Simulator for Interconnection Networks", International Symposium on Microarchitecture, MICRO-35, November 2002, Istanbul, Turkey.
- [11] T. Ye, G. De Micheli and L.Benini, "Packetized On-Chip Interconnect Communication Analysis for MPSoC", Proceedings of DATE-03, March 2003, Munich, Germany, pp. 344-349.
- [12] J.Hu and R. Marculescu, "Exploiting the Routing Flexibility for Energy/Performance Aware Mapping of Regular NoC Architectures", Proceedings of DATE-03, March 2003, Munich, Germany, pp. 688-693.
- [13] T. Grotker, S. Liao, G. Martin and S. Swan, "System Design with SystemC", Kluwer Academic Publishers, 2002.
- [14] "STBus Communication System: Concepts and Definitions", Reference Guide, STMicroelectronics, October 2002.
- [15] "STBus Functional Specs", STMicroelectronics, public web support site, http://www.stmcu.com/inchtml-pages-STBus_intro.html, STMicroelectronics, April 2003.
- [16] Synopsys Inc., "Core Consultant Reference Manual", "Power Compiler Reference Manual" and "VCS: Verilog Compiled Simulator Reference Manual", v2003.06, June 2003.
- [17] C. Patel, S. Chai, S. Yalamanchili, and D. Schimmel, "Power-constrained design of multiprocessor interconnection networks," in Proc. Int. Conf. Computer Design, pp. 408-416, Oct. 1997.
- [18] H.Zimmermann, "OSI Reference Model - The ISO model of architecture for Open System Interconnection", IEEE Trans. on Communication, n 4, April 1980.
- [19] VSI Alliance Standard, "System-Level Interface Behavioral Documentation Standard Version 1", Released March 2000.
- [20] Box, George E. P. and Draper Norman Richard. *Empirical model-building and response surfaces*, John Wiley & Sons New York, 1987