Multi-Level Simulation Platform of Network-on-Chip for Design Space Exploration

Kuei-Chung Chang and Chien-Hao Chen

***Abstract*—As multi-core architectures scale in size, on-chip networks (NoC) have become the main communication architecture, replacing dedicated interconnections and shared buses. NoC architectures have to deliver good latency-throughput performance in the face of very tight power and area budgets. Live power measurement is necessary for both hardware and software designer, but it requires too much time for simulation, especially for embedded systems. The major contribution of this paper is to present a simple method for rapidly estimating power consumption and find the hotspots in the network-on-chip(NoC) at two different resolutions. In addition to the low-level cycle-accurate simulation, we also build a high-level NoC simulation platform for multi-core SoCs, called TLM-PVT level. The platform, implemented by SystemC, allows early exploration of the performance and power consumption of NoC, which is able to handle arbitrary topologies and routing schemes. In the experiments, we compare the implemented high-level simulation platform with cycle-accurate simulator. The results show that the TLM-PVT simulator gives a high simulation speedup factor with a negligible performance estimation error margin.**

***Index Terms* — Multi-Core SoC; Network-on-Chip; Low-Power Design; Simulation**

I. Introduction

As technology scaling enables the integration of billions of transistors on a chip, economies of scale are prompting the move toward parallel chip architectures with application-specific systems-on-a-chip (SoC) leveraging multiple specific purpose cores on a single chip for better performance at manageable design costs. As these parallel chip architectures scale in size, on-chip networks have become the main communication architecture, replacing dedicated interconnections and shared buses. NoC architectures have to deliver good latency-throughput performance in the face of very tight power and area budgets. Interconnection networks consume 20%–36% of total system power in many large SoCs [1].

High-level simulators work well at the behavioral and architectural levels, but they are useful only in determining the functional correctness of a system. When the aim is to evaluate the performance or power consumption, simulations at low level are needed, but they fail in delivering fast results. Models of systems can be very accurate, but designers also need feasible simulation times to validate their hypothesis and verify the design. With the introduction of transaction-level modeling (TLM), simulation times have shortened considerably while keeping an acceptable level of accuracy, raising the abstraction level that can be used for effective performance simulation.

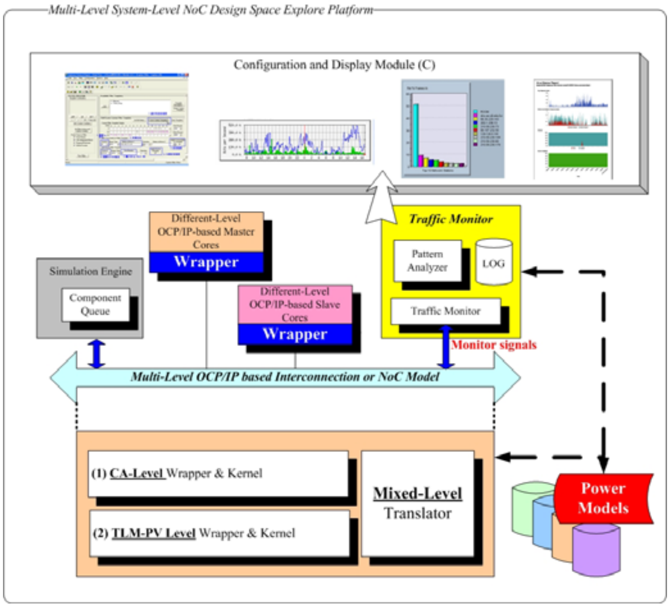

Multi-Level System-Level NoC Design Space Explore Platform
Configuration and Display Module (C)
Different-Level OCP/IP-based Master Cores
Wrapper
Simulation Engine
Component Queue
Different-Level OCP/IP-based Slave Cores
Wrapper
Traffic Monitor
Pattern Analyzer
LOG
Traffic Monitor
Monitor signals
Multi-Level OCP/IP based Interconnection or NoC Model
(1) CA-Level Wrapper & Kernel
(2) TLM-PV Level Wrapper & Kernel
Mixed-Level Translator
Power Models

Fig. 1. Basic concept of multi-level NoC simulation.

The main contribution of this paper is that we design a two-level design space exploration (DSE) platform based on SystemC [2] for NoC designs, shown as Fig. 1. The first level is the cycle-accurate level, which simulate the NoC architectures in low-level cycle-accurate designs to provide more accurate results. The second level is the programmer view with time (PVT-TLM), which is based on high-level TLM model to simulate the NoC behaviors to provide high-speed simulation results. The advantages of using system-level design are high performance and easy to explore optimal designs in large design space. It can also provide easy and fast simulation environments for modules to trace and estimate their designs. Clearly, high-level NoC simulator ignores many detailed activities of the component power; however, the proposed approach enables an easy analysis framework that is much faster than cycle-accurate simulations. The feature can help designers to design an optimal NoC architecture from large design space quickly and easily.

The rest of the paper is organized as follows. We discuss some related work in Section II. We describe the proposed

Manuscript received October 18, 2011, revised October 24, 2011. This work was supported by National Science Council under Grant No. NSC98-2220-E-035 -001.

Kuei-Chung Chang is with the Department of Information Science and Computer Engineering, Feng Chia University, Taichung, Taiwan. (e-mail:changkc@fcu.edu.tw).

Chien-Hao Chen, was with the Department of Information Science and Computer Engineering, Feng Chia University, Taichung, Taiwan.

multi-level power estimation approach for NoC in Section III. The experimental results and case studies are shown in Section IV. Finally, we summarize our findings in Section V.

II. Related Work

Designing cost-sensitive embedded products such as smart phones and portable media player requires maximizing a platform's performance while minimizing energy use. The more efficient version will result in a more cost-effective product. With power dissipation becoming an increasingly vexing problem across many classes of computer systems, measuring power dissipation of real, running systems has become crucial for hardware and software system research and design. Live power measurements are imperative for studies requiring execution times too long for simulation. Especially for embedded systems, there is a high demand for optimization techniques that enable energy reduction for software, since an increasing number of applications are powered by batteries. Therefore, recent studies have been focusing on developing techniques to reduce the energy consumption at various levels, including program optimization for low power [3-7].

There are two traditional methods used to acquire energy consumption information: simulations or measurements. Programmers find simulation-based energy estimation techniques convenient if appropriate simulation models are available [8-10]. For low-power software development, instruction- or architecture-level energy simulators such as Wattch [8] and SimplePower [10] might be better solutions. However, those cycle-accurate simulators have a reputation for being slow. At present, power measurement tools are available for only the lower levels of the design - at the circuit level and the gate level. These are very slow and impractical to use to evaluate the power consumption of software, and often cannot even be applied due to lack of availability of circuit and gate level information of the embedded processors.

Deep-submicron technologies have clearly had a big impact on capacity and what can be designed on a single system-on-chip (SoC). With increased functionality, however, comes increased complexity for the design and verification process. Simultaneously, the industry has been looking at ways to improve engineering productivity by offering improved register-transfer-level (RTL) verification tools with advanced features, such as constrained-random test generation, functional coverage metrics and assertions made available through such languages as SystemVerilog. Along with those trends, the industry has introduced design and verification tools that operate at higher levels of abstraction, such as the electronic system level (ESL), supported through languages such as SystemC [11-13].

III. Design of Multi-level NoC simulator

A. System Architecture

Fig. 2 shows the architecture of the proposed multi-level NoC simulation platform, which consists of several components, and each component is responsible to execute several functionalities. In the platform, we attempt to design an electronic system-level NoC simulation framework to provide early and fast exploration of the system information.

The traffic generator can inject traffic flow into the NoC for simulation. Network traffic can be characterized and constructed temporal characteristics, spatial distribution, and data size. The temporal characteristics describe the data generation probability during the simulation. The spatial distribution gives the communication partnership between sources and destinations. The data size defines the length of communicated data.

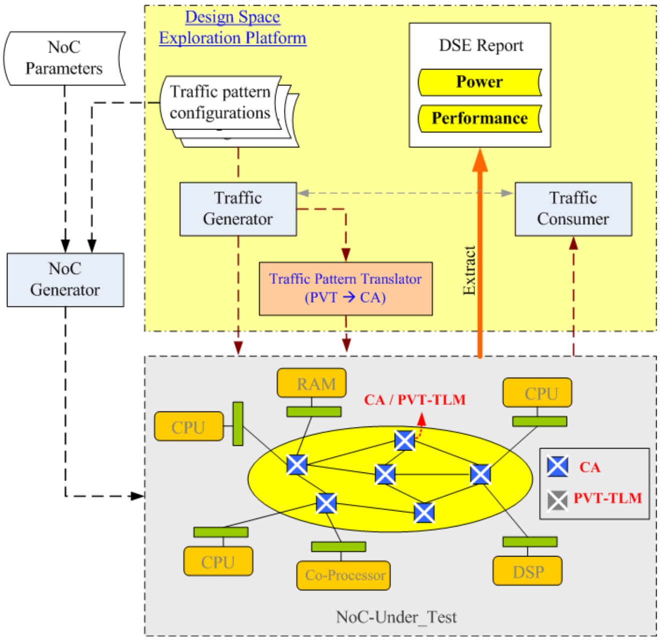


Fig. 2. NoC simulator design diagram

To model temporal characteristics we provide constant injection rate, probabilistic functions and trace based. Constant injection rate models generate traffic at a fixed rate, namely temporal uniform distribution. In fact, sending intervals between basic units of traffic are random. And these random intervals usually present a certain degree of regularity, satisfying some probabilistic distribution. This type of traffic should be modeled by probabilistic functions, such as Poisson and Self-similar distribution. The proposed multi-level simulation platform can generate Uniform, Poisson and Self-similar traffic, and it supports trace based traffic to model many real applications.

Spatial distribution can be fixed or random. Fixed distribution indicates that the communication partnership between sources and destinations is fixed through the simulation, which is applied with trace based traffic to characterize specific application in this platform. Random distribution can be divided into Uniform and Non-uniform distribution. Uniform distribution indicates that traffic is uniformly distributed to nodes with a different distance to model the localization feature.

We also provide a monitoring component, named traffic monitoring component (Traffic consumer), to monitor and gather communication traffic from the simulated interconnect architecture and either to display the information at run-time or store the information in files or database to provide off-line analysis and display. The major contribution we provided is to present a method for rapidly estimating power consumption.

The platform will support tracing for debugging purposes on all its elements. In addition, it will also support logging of bus traffic for purposes of estimations. We designed a subsystem, named power information analysis component (DSE Report) to provide on-line and off-line analysis for the simulation. In addition, we also want to build a GUI-based display component, named power information display component (PIDC) to provide different displays to provide designers different views for their designs.

B. *ESL-based NoC Design Flow*

The NoC is a structured interconnection architecture such that it can be integrated into a design flow easily, as shown in Fig. 3. First, the communication characteristics among partitioned cores can be derived by profiling embedded applications. Then, we can construct suitable communication topologies according to the profiling results and specific purposes, such as power and performance constraints. Using topology construction tools or topology templates in the library we can decide which cores should be connected in the same router such that the power consumption of communications can be minimized. After constructing the interconnection topology, we can apply other optimization mechanisms according to the application traffic characteristics and the interconnection architecture.

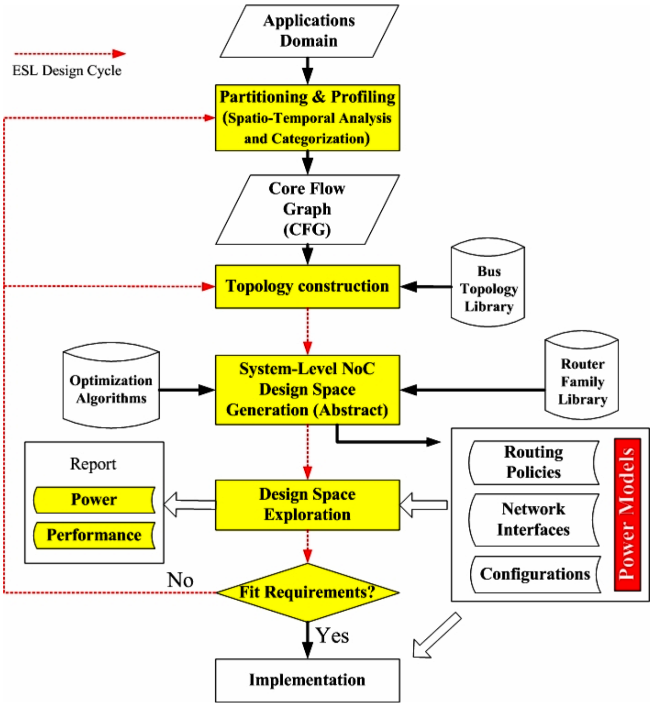
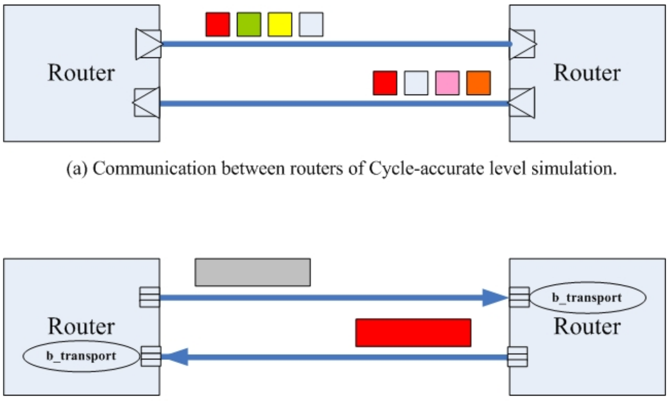


Fig. 3. ESL Design flow for NoCs

C. *Design of Multi-Level Simulator*

Fig. 4 shows the communication concepts of cycle-accurate level and TLM-PVT level. In cycle-accurate level, the detail design of the router is simulated, including the buffer, the routing policy, and switching behavior. The transmitted data size depends on the channel width. We can account correct switching factors of each component in a router because the communication behavior is bit-accurate. The sequence of the communication traffic is simulated according to the traffic flits one by one, and the contention status in each router can be simulated in the level.

The TLM-PVT level simplifies the description of inter-module communication transactions using objects and channels between router modules. In TLM-PVT level, the detail information for the sequence of flits is omitted. Transactions are performed through channels instead of signals as shown in Fig. 6(b). The channels implement one or several interfaces, and each interface has a set of read or write functions are instantiated by masters and sent through the port to the channel interface. At the level of slaves, the transaction will be recovered to execute the corresponding routing methods and to transport the message to the next node. In TLM-PVT level, a timing model is defined to approximate the execution time. The timing is estimated according to the design of the router, including switch structure, routing policy, and the buffer size.



(a) Communication between routers of Cycle-accurate level simulation.

(b) Communication between routers of TLM level simulation.

Fig. 4. CA vs. TLM communication behaviors.

D. *Power Estimation Approach*

The power estimation approach in this platform is the two-phase power estimation approach. For high-level fast simulation we have to measure the power consumption of the key components in the network router in phase I. We store these power models in power models, as shown in Fig. 1, for later simulations. In phase 2, we will analysis the traffic characteristics in the interconnect architecture and gather access counts and bit switching activities. With the gathered information we can estimate the rough power consumption of each router.

In the platform we have to provide two different power models for high and low simulation levels. For low-level power model, we can estimate the power consumption of each component in the NoC by following formulation.

$$P(Ci) = (C_{af}*AF + C_{cf}*CF) * V^2_{dd} * f + P_{leak\,;}$$

$$E(Ci) = Ui * P(Ci) ;$$

where P(Ci) is the power consumption of component C_i per access with different switching activity factor(AF) and coupling factor(CF). U_i is the activity/utilization of component C_i. C_{af} represents the capacitance related to switching activity, and C_{cf} means the capacitance related to coupling activity. P_{leak} means the leakage power of the component. These characteristics can be measured by low-level power measurement tools off-line, such as PSPICE and Nanosim. They can precisely predict the timing, power consumption, and functionality of their designs.

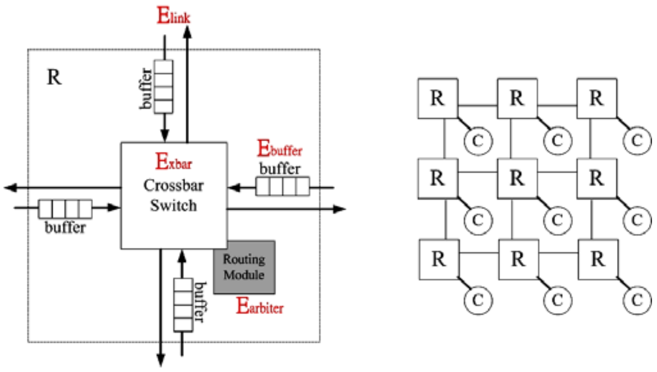
Switching activity and coupling activity cause dynamic energy consumption of CMOS circuits. The switching activity is largely dependent on the Hamming distance of data between current and previous clock cycles. The switching activity happens when the data bit is from 0 to 1 or from 1 to 0. Thus, it can be expected that the actual energy cost of executing a program may be different from the component's data inputs. The more bit switches, the more power consumed. We will keep the state of input data and control signal every cycle. We need to compare the data of current cycle with the data of previous cycle, so we can get the amount of bit switches. Coupling activity is determined by averaging the coupling between adjacent lines for a execution trace of a benchmark.

For high-level power model the real switching factor and coupling factor cannot be calculated accurately. In this model, we will assign a reasonable probability and weight to the AF and CF in previous power formula for each application. We will also monitor the access count of each component of the router during execution. We embed a counter outside the component to record the total access count. The value of the counter will be accumulated if the input data and the control signal of the component changed.

The energy model of the NoC router consists of four parts including E_{buffer}, E_{xbar}, E_{link}, and $E_{arbiter}$, shown as Fig. 5. We can get the total energy consumption E_{total} of the communication architecture by following formulation.

$$E_{packet} = E_{buf\_r} + E_{buf\_w} + E_{xbar} + E_{link} + E_{arbiter} + E_{base}\ ;$$

where E_{packet} is the total energy consumption during the communication, the E_{base} is the basic energy consumption except the power consumed by the accessed components. The E_{buf_r} means the energy cost reading packet from the buffer, and the E_{buf_w} means the energy cost writing packet in the buffer. The main purpose of the proposed simulator is to provide a flexible high-level simulation platform to tune the communication characteristics quickly, such as topology, mapping, buffer size, buffer count, etc. It's hard to achieve high accuracy for the power model; however, the relative power consumption in the router can be measured by low-level power simulators with high accuracy. The relative power consumption of each component in the router can be used in the simulator to find hot spots in the network efficiently.



(a) Network-on-Chip router (b) Network-on-Chip Model

Fig. 5. Power model of router in the NoC.

IV. Experimental Results

We design an evaluation flow as shown in Fig. 6 to capture the traffic traces from real applications. It starts from application specifications, continues through the topology construction of the application. At first we use the simplescalar to simulate applications and collect the data flow. Then the Read/Write analyzer will analyze the communication behaviors between the writer cores and reader cores, and it will generate a core flow graph. The test pattern generator will analyze the communication statistics between writers and readers, and then generates the simulation workloads for final simulations. The workload content includes access address, write data, read data, and request, etc. After profiling, we get the communication status of cores, and then generate the simulation patterns for Verilog simulator. Finally, we can get power and performance results to evaluate our design.

The power models for the arbiter, buffer, crossbar, and the wire have been calibrated with Nanosim simulations of these components over different technologies. We also compare the speed and the accuracy between these two simulation levels.

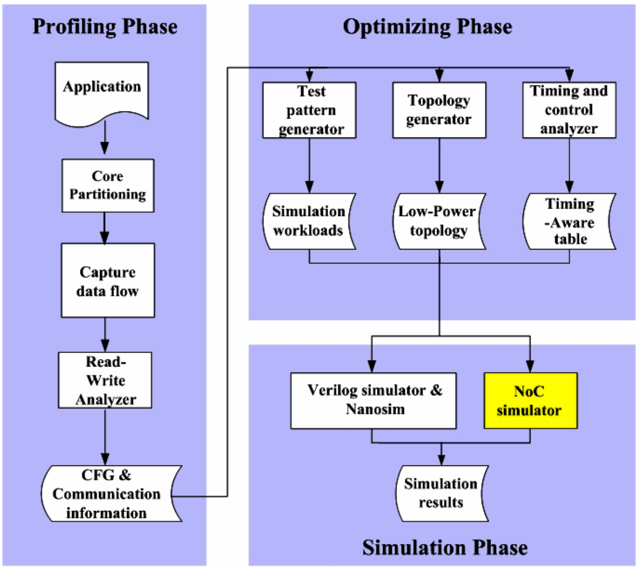
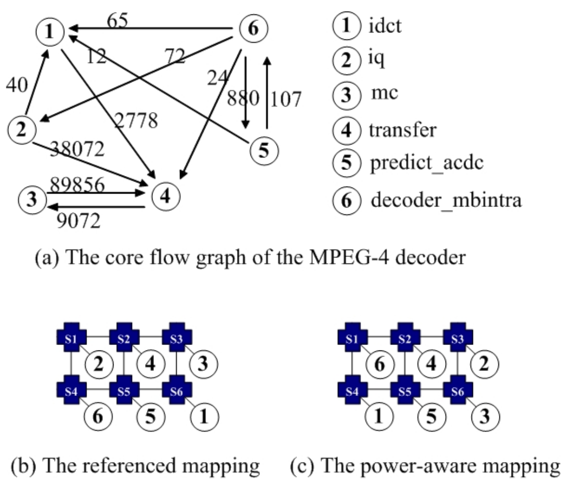


Fig. 6. The evaluation flow of the experiments.

A. Case Study – MPEG-4 Decoder



(a) The core flow graph of the MPEG-4 decoder

(b) The referenced mapping (c) The power-aware mapping

Fig. 7. MPEG-4 decoder mapping topologies.

We take the MPEG-4 decoder as our case study, and Fig. 7 shows the profiled core flow graph of MPEG-4 decoder and the experimental interconnection architectures with power-aware mapping described in [14]. Fig. 10(b) is the compared topology, and the power-aware topology generated by the proposed tool is shown as Fig. 10(c). In the experiments, the ratio of the power saving approximates to 35% of the NoC compared to the referenced topology. The power saving measured by low-level power measurement in [14] approximates to 30%. From the results we can find that the high-level simulator can identify the optimal communication architecture.

B. Comparisons between CA and TLM-PVT Levels

This experiment shows the power consumption of a 3x3 NoC. The count of simulated packets is about 1,000,000 flits, and the injection rate is about 0.1 (flit/cycle/node). The traffic is uniformly generated to random destinations.

Fig. 8 shows the simulation results measured by CA level NoC simulator. The top of the figure shows the average latency of a flit traversed from the source to the destination. The down of the figure shows the average dynamic power consumption of each router. Fig. 9 shows the simulation results by TLM-PVT level simulator for the same traffic flow. From the two figures we can find that the difference of the relative power consumption between these two levels is small.

Table I shows the comparison results between cycle-accurate level and TLM-PVT simulators. The speedup of the TLM-PVT level simulation is about nine compared to the CA level simulation. The error rate of the performance is about 5%, and average dynamic power consumption is about 5.6%. The error rate of the confliction ratio is large, about 58%. This is because the sequence of the communication patterns in PVT-level simulation cannot be determined precisely. However, the simulation time of TLM-PVT simulation is 5 times of CA simulation.

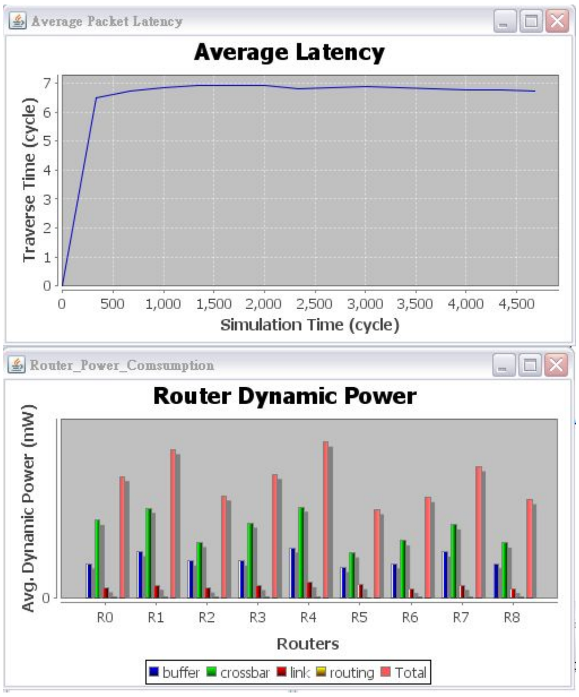


Fig. 8. The results of CA level.

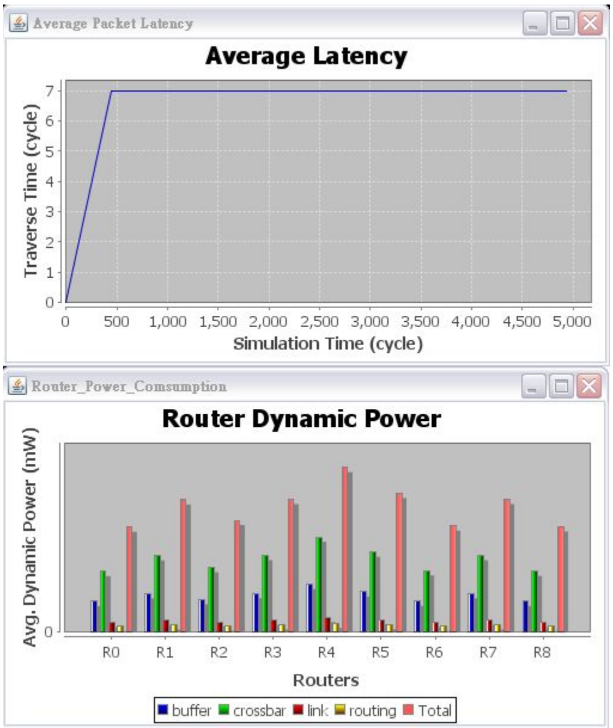


Fig. 9. The results of PVT level.

TABLE I: THE COMPARISONS BETWEEN TWO LEVELS.

	Sim. Cycles	Avg. Latency	Conf. Ratio	Avg. Power(mW)
CA	18703	6.66	12%	0.757
TLM-PVT	19704	7	5%	0.802
Error rate	5%	4.8%	58%	5.6%

C. Discussions

The software simulator must use high level language to model the behavior of the core, components and the monitoring circuit, so the simulation speed depends on the speed of the computer. The proposed method has less accuracy than gate level simulation. Also, the accuracy of the proposed TLM-PVT level is less than the proposed cycle-accurate simulator. However, our goal is not to get precise power consumption of each component, and we just want to get relative power consumption of each accessed component in a NoC. By this way, programmers can remove hotspots that could consume the maximum power between pairs of modules. Due to the flexibility of the ESL modeling, it makes the application to be simulated and tuned in a reasonable and realistic way. Clearly, high-level NoC power ignores many detailed activities of the component power; however, the proposed approach enables an easy analysis framework that is much faster than low-level power simulations.

V. Conclusion And Future Work

In this work we present a multi-level NoC performance and power estimation platform based ESL methodology. The high-level simulation can simulate the NoC design with high speed; the low-level simulation can simulate the NoC design with high accuracy. Designers can choice the simulation level depending on the design phase of the simulation. Because our method has per-component power consumptions, we can get unit-by-unit power estimates in the router. Furthermore, we can treat these component power estimates as a power signature that can effectively distinguish power phase behavior based on simple analysis. It is our hope that, in the future, hardware vendors will see the competitive advantage of providing customers with detailed power information about their products. In this way, our tool can get more accurate and different types of the power consumption about each application.

In the future, we have to solve two key issues of the platform. The first one is the simulation speed of the low-level simulation. We are going to enhance the low-level simulation with OpenMP multi-threading design on multi-core platform. Second, we have to enhance the precision of the high-level simulation, such as contention ratio of each router. Finally, we will build the integrated development environment to let programmers to develop power-aware multi-core applications easily.

References

[1] Soteriou, V. and PEH, L.-S. Design-space exploration of power-aware on/off interconnection networks. In Proceedings of the IEEE International Conference on Computer Design (ICCD). 510–517, 2004.

[2] SystemC , http://www.systemc.org/home/

[3] Naehyuck Chang, Kwanho Kim, and Hyung Gyu Lee. Cycle-accurate energy measurement and characterization with a case study of the arm7tdmi. IEEE Transactions on Very Scale Integration Systems, 10(2), April 2002.

[4] Robert P. Dick, Ganesh Lakshminarayana,Anand Raghunathan, and Niraj K. Jha. Analysis of power dissipation in embedded systems using real-time operating systems. IEEE Transaction on computer-aided design of integrated circuits and systems, 22(5), 2003.

[5] Canturk Isci and Margaret Martonosi. Run-time power monitoring in high-end processors: Methodology and empirical data. In Proceedings of the 36th International Symposium on Microarchitecture, December 2003.

[6] D. Shin, H. Shim, Y. Joo, H. Yun, J. Kim, and N Chang. Energy monitoring tool for low-power embedded programs. IEEE Design and Test of Computers, 19(4), July 2002.

[7] Greg Stitt, Frank Vahid, Tony Givargis, and Roman Lysecky. A _rst-step towards an architecture tuning methodology for low power. In Proceedings of the International conference on Compilers, architectures, and synthesis for embedded systems, pages 187-192, November 2000.

[8] Robert P. Dick, Ganesh Lakshminarayana, Anand Raghunathan, and Niraj K. Jha. Power analysis of embedded operating systems. In Proceedings of the 36th conference on Design automation conference, pages 312{315, June 2000.

[9] Tajana Simunic, Luca Benini, and Giovanni DeMicheli. Cycle-accurate simulation of energy consumption in embedded systems. In Proceedings of the 36th ACM/IEEE conference on Design automation conference, pages 867-872, June 1999.

[10] W. Ye, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin. The design and use of simplepower: a cycle-accurate energy estimation tool. In Proceedings of the 37th conference on Design automation, pages 340-345, June 2000.

[11] D. Gajski et al. “SpecC: Specification Language and Methodology,” Kluwer, Jan 2000.

[12] L. Cai and D. Gajski, “Transaction Level Modeling: An Overview,” Proc. CODES+ISSS, pp. 19-24, Newport Beach, CA, Oct. 2003.

[13] Chris Lennard and Nizar Romdhane, “Building a virtual-platform Design Ecosystem through Open System-level Integration Strategies”, Design Strategies and Methodologies, vol 5, no 4, pp. 52-57, 2006

[14] Kuei-Chung Chang, Jih-Sheng Shen and Tien-Fu Chen, “Tailoring Circuit-Switched Network-on-Chips to Application-Specific System-on-Chip by Two Optimization Schemes”, to appear in ACM Transaction on Design Automation of Electronic Systems, Vol. 13, No. 1, Article 12 , January 2008