# Introducing Energy and Area Estimation in HW/SW Design Flow Based on Transaction Level Modeling

Muhammad Omer Cheema, Omar Hammami
ENSTA
32 Boulevard Victor
75739 Paris, France
. {cheema,hammami}@ensta.fr

***Abstract*— Transaction Level Modeling (TLM) facilitates the system designer in decision making at early phases of electronic product development. Executable specifications obtained from TLM models are used for the exploration of various architectural parameters and configurations possible for a system. However, traditional design flows based on TLM don't take into account the area and energy consumption of the system which are inevitably the most important constraints in modern embedded system designs. Traditionally, area and energy estimation is incorporated in system design at RTL (Register Transfer Level) which comes later in system design cycles and most of the crucial design decisions for the system has already been taken at that stage. In this paper, we propose a methodology to incorporate area and energy estimation in TLM based system modeling. This methodology allows a system designer take system level design decisions in very early stages of system design hence avoiding redesign efforts and performance bottlenecks in advanced stages of a project. Results obtained by applying our methodology on an image processing application show the robustness of our approach.**

***Index Terms*— Hardware/Software Codesign, Behavioral Synthesis, Transaction Level Modeling, Area and Energy Estimation**

## I. INTRODUCTION

SystemC Transaction level modeling has emerged as a key modeling approach for ESL (Electronic System Level) design [1,2]. TLM's wide acceptance as a preferred modeling technique for electronic systems is driven by a lot of potential benefits obtained because of its introduction in a system design flow. These benefits include:

- Faster system design space exploration
- Easier system modeling
- Faster simulation speeds
- Early start of software development
- Avoidance of late bugs in a project.
- Hence: Shorter Times to Market for a product.

With the evolution of modern embedded systems in general and wireless embedded systems in particular, it is becoming more and more important to take into account the area and energy consumption issues in consideration during system design at early stages. Importance of energy consumption in modern embedded systems can be understood by the trends predicted by Gelsinger's law. According to it , New generation micro architectures use twice as many transistors for a 40 percent increase in performance which in other words predict that [1] "By 2010 high-end processors would radiate the same heat volume per square centimeter as a rocket nozzle, by 2015 – as the Sun's surface". While dealing with the heating issues is considered one of the bottlenecks in future embedded systems, we believe that it is very important to enhance the traditional system level design methodologies by incorporating energy estimation methodologies to get full potential benefits offered by TLM. Behavioral Synthesis techniques based on C++/SystemC provide us an opportunity to estimate area/energy of various components of the system without implementing them at RTL level (in VHDL/Verilog). A few tools have been commercially proposed that allow the behavioral synthesis of C++/ SystemC description of hardware modules. In this paper, we propose an extension to TLM based design methodologies by incorporating these C based behavioral synthesis techniques in the design flow which allows us to take into account the crucial factors of area and energy estimation at the early stage of system design which in returns increases the productivity, reduces the time to market and avoids the energy/area based bottlenecks at the later stages in system design.

Rest of the paper is organized as follows: Section II presents related work. Section III explains our System design methodology and section IV describe the experiment environment and results. Section V presents conclusions.

## II. RELATED WORK

Work on power estimation has been done at various layers of abstraction. Most of the work has been done in RTL and lower layers of abstraction. However a little work has been done on power estimation at TLM level of abstraction. Existing work as presented in [3-8] emphasizes on TLM modeling, TLM simulation speedup over VHDL, TLM-VHDL co-simulation. Work done on power [9-12] is instruction based, function based and state based power analysis. However these power analysis techniques are based

1-4244-0765-6/06/$20.00 ©2006 IEEE

on post partitioning architectural analysis of the system where partitioning decisions of the system has already been taken and energy is being estimated for communication taking place between various components. These techniques only estimate the power consumption of communication part of the system and hence these approaches can't prove to be useful for system partitioning decisions. Our approach emphasizes on area and energy estimation for various IP models in a system before the partitioning decisions has been taken. Instead of proposing lower level (instruction or transaction level) power and area estimation, we perform IP level area and energy estimation using existing work on behavioral synthesis of system level design languages. A few commercial tools [13,16]are available for area estimation for behavioral modules. Synopsys SystemC compiler was perhaps the first commercial efforts to synthesize behavioral code written in ESL languages. Design space exploration have been achieved with this tool [14,15] Celoxica's Agility can synthesize SystemC behavioral description of hardware modules and give the area estimation requirements for various ASIC and FPGA based architectures. Orinoco Dale [17] is another tool that estimates area and energy for C description of an application. We perform area and energy estimates of various IP components in the system before actually modeling the system in TLM. The area and energy estimation information is fed into the TLM model of the system where we partition the system by automatically exploring the system design space based on the given information. To our knowledge, we are the first ones to propose a design flow that uses TLM modeling for system design space exploration and includes area and energy estimation information during the partitioning process.

## III. PROPOSED METHODOLOGY

Our target platform for the synthesized system is shown in Fig. 1. Starting from an application written in C/C++, we plan to have a system built around a general purpose processor (in our case IBM PowerPC 405) and extended by appropriate hardware accelerators that provide good system performance improvement without requiring too much increase in area and energy consumption.

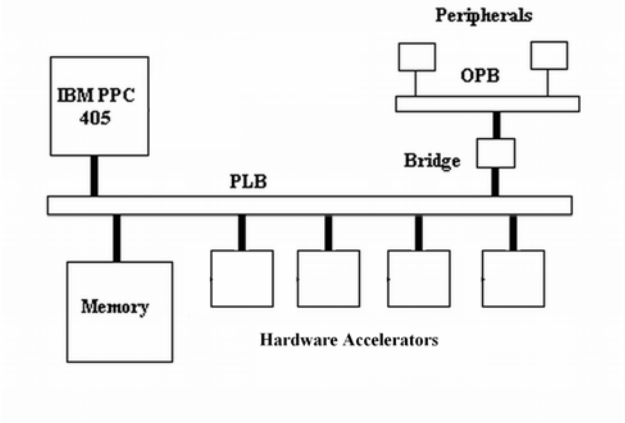


**Fig 1: Target Platform Built using IBM TLM**

The design flow is shown in Fig. 2. At the first stage, application is written in C/C++. This application description can be run over a general purpose processor describing the system without hardware accelerators. That means, at this stage all the functionality is implemented in software part.

Then this application is forwarded to a C/C++ synthesis tool. We use Orinoco dale as a tool of our choice for the estimation of area and energy consumption. Dale provides the statistics for area/energy which are stored in a database. On the other hand, task graph is extracted from the software that is extended with the information received from dale. This extended task graph allows the design space exploration engine to provide the information about various tasks and their hardware implementation requirements.

Our design space exploration engine then generates various configurations by shifting a software function to hardware function. It should be noticed that shifting functionality from hardware to software or software to hardware is a straightforward task as both hardware and software (hence the complete system) is described in C and only work required to move a functionality from software to hardware is to replace computation with a data transfer function in software and adding computation into hardware along with synthesis information obtained from dale. Similarly, for area and energy estimation, no alteration is required for original application description as dale produces area and energy estimation requirement from the C description. Hence, no major work is required for translating C to hardware description languages or vice versa during the complete design flow.

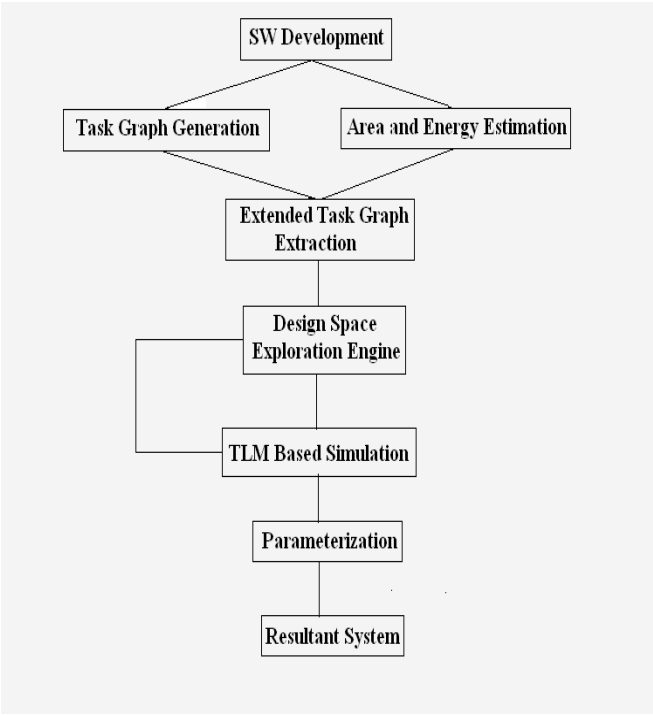


**Fig. 2 HW/SW Codesign Flow**

In the last step, tuning of some soft parameters is performed to improve the application performance and resource usage. Examples of such soft parameters include interrupt and arbitration priorities and bus width.

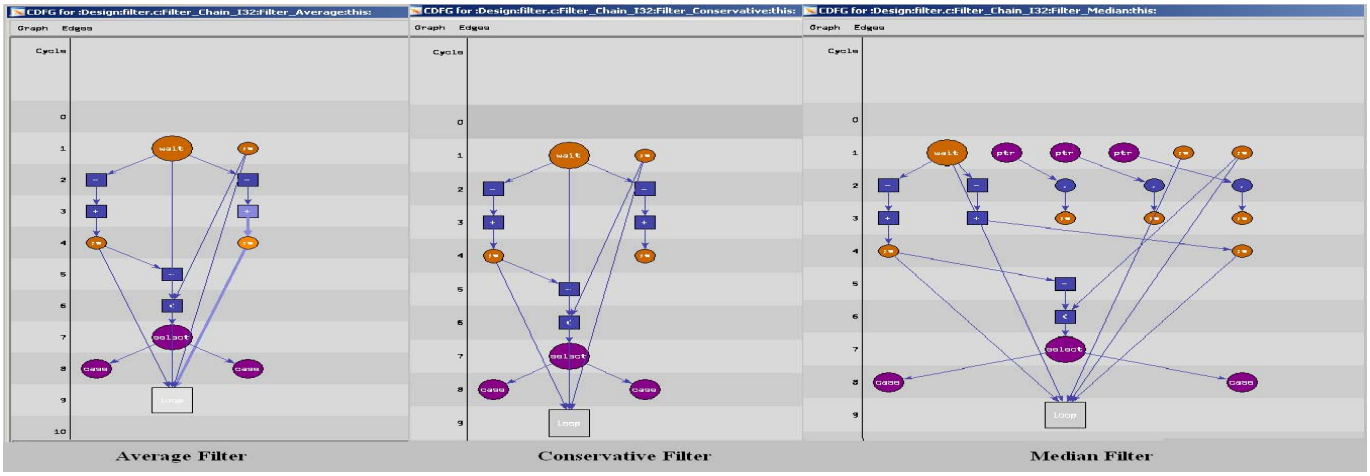


**Fig 3: Control Data Flow Graph of Filters in the Application**

## IV. Experiment Environment and Results

We have tested over approach using IBM's PowerPC 405 Evaluation Kit (PEK) [9] that allows designers to evaluate, build, and verify SoC designs using Transaction level modeling. As mentioned above, our target is to synthesize a system based on a general purpose processor (in our case, IBM PowerPC 405) and extended with the help of suitable hardware accelerators to improve the system performance significantly. A gcc based cross compiler for PowerPC405 was used to compile the software while SystemC compiler was used to compiler hardware modules. In our application, we had three components in an image processing chain: median, conservative and average filters. For our simulation, our general purpose processor (PowerPC 405) was running at 333 MHz with 16K of Data and instruction caches.

Figure 3 shows the Control data flow graph (CDFG) generated by dale for each of the filters while Fig.4, Fig. 5 and Fig. 6 show the area, energy and execution cycles taken by the filters if synthesized using 180 nm process technology.

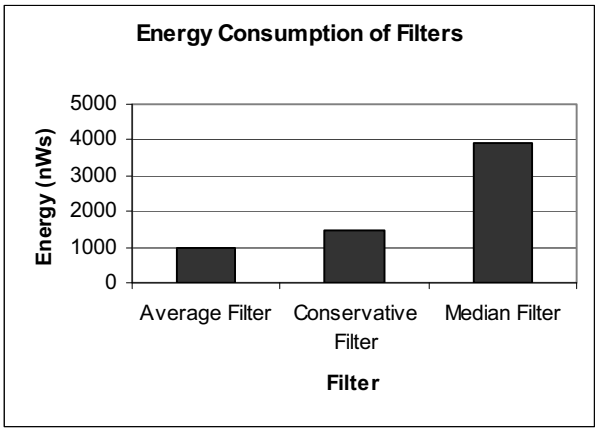


**Fig 4: Energy consumption requirements for filters**

It should be noted that median filter is more expensive in terms of number of cycles and energy consumption while conservative filter is slightly more expensive in terms of area requirements. Results obtained from these tables are then forwarded to design space exploration engine which adds the computation time estimations in TLM description of these' components and performs the simulation of various configurations.

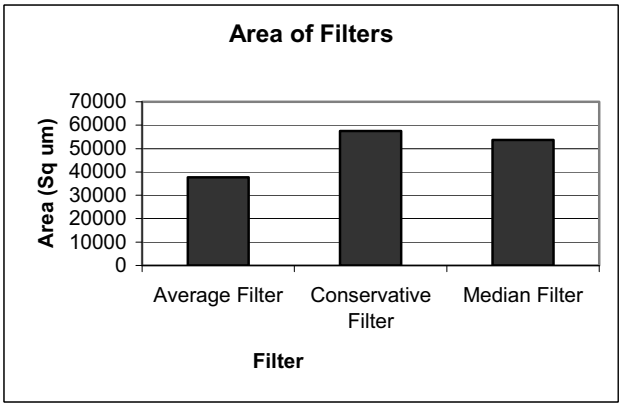


**Fig 5: Area requirements for filters**

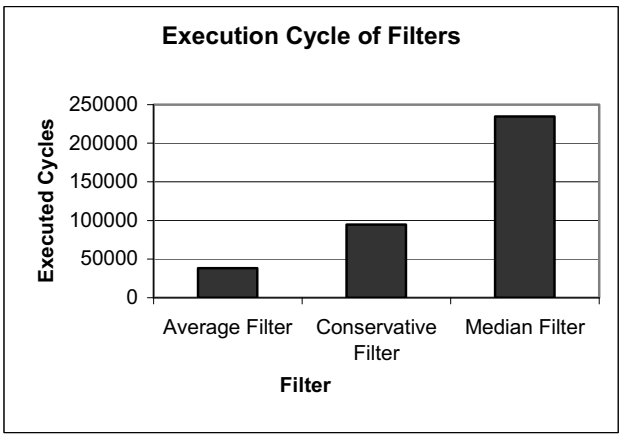


**Fig 6: Execution cycles taken by Filters**

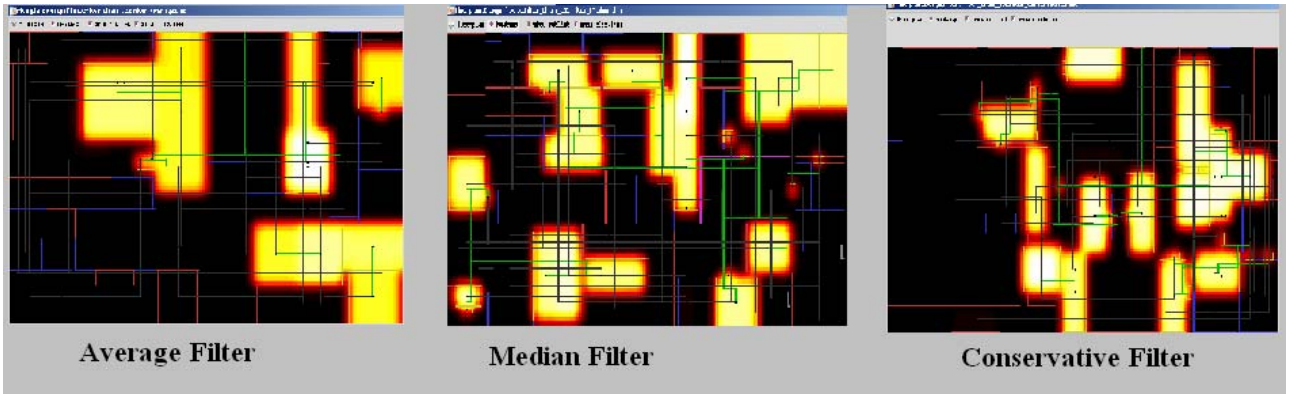


**Fig. 7 Heat Maps of Filters**

Table 1 shows the various configurations generated by the design space exploration engine and the number of PowerPC cycles taken by the system for each of the configuration. We can see that best speedup can be obtained if Median and Conservative filters are implemented as hardware. However, looking at the high area requirements of conservative filters, we can opt to keep the conservative filter in software and hence configuration 1 gives a solution that is optimal both for area and speedup requirements

**Table 1: Various configurations and Speed ups for Filters**

| Config. No. | Hardware Implementation | Time (cycle) | Speedup Over Software Version |
|---|---|---|---|
| **1** | Median | 3897000 | 1.859 |
| **2** | Average | 7202000 | 1.006 |
| **3** | Conservative | 6630000 | 1.093 |
| **4** | Median + Average | 3493000 | 2.075 |
| **5** | Median+ Conservative | 2921000 | 2.481 |
| **6** | Average+ Conservative | 6028000 | 1.202 |
| **7** | Software Version | 7248000 | 0 |

## V. CONCLUSIONS

In this paper, we have proposed a methodology of HW/SW codesign that incorporates area and energy estimates in the existing TLM based system design flows. With the evolution of C based synthesis tools, it is becoming possible to enhance the existing methodologies based on transaction level modeling to include area and energy estimates as well. The design decisions taken at such higher levels provide a detailed insight of the system design space to the system designer which helps him take partitioning decisions while keeping in mind the factors of area and energy consumption as well. Our experimentation over an image processing application shows the usefulness of our approach.

## REFERENCES

[1] ITRS http://www.itrs.net/
[2] SystemC www.systemc.org
[3] S. Lee, S. Park, "Transaction Analysis of Multiprocessor Based Platform with Bus Matrix", (IWSOC'05) ,Banff, Canada
[4] Caldari, M.; Conti, M.; Coppola, M.; Curaba, S.; Pieralisi, L.; Turchetti, C.;, "Transaction-level models for AMBA bus architecture using SystemC 2.0", DATE, 2003
[5] Moussa, I.; Grellier, T.; Nguyen, G.;, "Exploring SW performance using SoC transaction-level modeling", DATE, 2003
[6] Bernstein, A.; Burton, M.; Ghenassia, F.;" How to bridge the abstraction gap in system level modeling and design", IEEE/ACM International Conference on Computer Aided Design, 2004. ICCAD-2004. 6-11 Nov. 2004
[7] Xu, S.; Pollitt-Smith, H.;, "A TLM platform for system-on-chip simulation and verification", (VLSI-TSA-DAT)27-29 April2005
[8] Portero, A.; Pla, R.; Rodriguez, A.; Carrabina, J.;" NoC Design of a Video Encoder in a Multiprocessor System on Chip Solution", The 17th International Conference on Microelectronics, 2005. ICM 2005, Islamabad Pakistan
[9] N. Dhanwada, I. Lin, V. Naraynan, "A Power Estimation Methodology for SystemC Transaction Level Models", CODES+ISSS'05, Sept. 19–21, 2005, Jersey City, New Jersey, USA.
[10] G. Qu, N. Kawabe, K, Usame, M. Potkonjak, "Function-level power estimation methodology for microprocessors", Proceedings of the 37th DAC, 2000
[11] R. Bergamaschi, Y.W. Jiang, "State-based power analysis for systems-onchip", Proceedings of the 40th Design Automation Conference, 2003.
[12] L. Zhong, S. Ravi, A. Raghunathan, N. K. Jha."Power Estimation for Cycle- Accurate Functional Descriptions of Hardware", ICCAD 2004.
[13] Synopsys. Behavioral Compiler User Guide Version 1999.10, 1999
**[14]** Chtourou, S.; Hammami, O., SystemC space exploration of behavioral synthesis options on area, performance and power consumption, ICM 2005, 13-15 Dec. 2005
[15] Chtourou, S.; Hammami, O., Design Space Exploration of SystemC SOM Implementation, ICICT 2005, Dec. 2005 Page(s):159 - 169
[16] Agility : http://www.celoxica.com/products/agility/default.asp
[17] Orinoco Dale http://www.chipvision.com/company/index.php
[18] http://www-128.ibm.com/developerworks/power/library/pa-pek/