International Conference on Network Applications, Protocols and Services 2008 (NetApps2008)
21 – 22 November 2008
Executive Development Center, Universiti Utara Malaysia

Estimation Transition Density in Wireless Sensor Network (WSN) with SpecC and SystemC languages

M. Rafiee
Shahed university, IRAN
rafiee@shahed.ac.ir

M. B. Ghaznavi-Ghoushchi
Shahed university, IRAN
ghaznavi@shahed.ac.ir

B. Seyfe
Shahed university, IRAN
seyfe@shahed.ac.ir

Abstract- Estimation energy consumption of sensor nodes is urgent work because life time of their battery is limited. This paper presents a replacement method to estimate power, which calculates transition density from SystemC and SpecC descriptions. Dynamic power is largest portion of the total power and in similar power supply and technology, transition density have important effect in power dissipation. The resulting transition density is very close to the results obtained using VCD and SAIF file formats.

I. INTRODUCTION

Networked sensor systems are seen by observers as an important technology that will experience major deployment in next few years for a plethora of applications. Typical applications include, but are not limited to, data collection, monitoring, surveillance, and medical telemetry [1]–[2]. Wireless Sensor Network consists of large number of sensor nodes. These tiny nodes consist of sensing, data processing, and communicating components. Sensor Network (SN) aim to provide an efficient and effective connection between the physical and computational worlds. At the same time, SN propose numerous new research and development challenges, including the need for the next generation low power, low cost, small size, error and fault resiliency, flexibility, conceptually new security and privacy mechanisms, and new types of input/output (I/O) operations [3]. The demand for wireless sensor is growing fast. New generation of these products includes more complexity, and thus the power dissipated by them.

For lower levels (gate and transistor-level), there are a vast set of CAD tools, e.g. SPICE [4], but estimation power at these levels of abstraction is very time-consuming. But estimation power at higher level of abstraction is very important.

Dynamic power consumption in circuits is due to the energy that is drawn from the power supply to charge parasitic capacitances, thus calculating transition density (transition from logic "0" to logic "1") is pointer of power consumption.

In this paper we explain structure WSN and estimation power at different abstraction level in section I. In Section II, a brief description of some of the existing techniques for high-level power estimation is given. System modeling language is explained in section III. Methods of Calculation Transition density in WSN are covered in sections IV. In section V WSN modeling is described. Method of gathering transition density in SystemC and SpecC languages and results are described in sections VI. Finally we conclude the paper in section VII.

II. PREVIOUS WORK

Early techniques of estimation power are associated with physical capacitance and activity of the circuit. In [11], a reference gate is chosen from a technology library, where the average power dissipated by this gate is pre-calculated. Then, to estimate the power consumed by a functional block, the number of gate equivalents to implement the block is extracted from the library and multiplied by the average power dissipated by the gate. There are also techniques based on macro-modeling, where measurements are made in existing implementations to produce a power model.

One macro-modeling technique which takes into account the data activity is [12]. During simulation, the activity in the control path and in the data path is monitored and then this activity information is used to feed power models that explicitly account for activity. A so-called equation-table method is proposed in [13] to reduce the time required during the characterization phase. The macro-modeling techniques can be divided into three categories: equation-based, table-based and hybrid. Most equation-based techniques make use of statistical methods for model building (e.g., regression analysis) [13]. Table-based technique was proposed in [14], consists of a three dimensional table, where the axes represent input/output signal statistics, and a table entry represents a power value.

III. SYSTEM MODELING LANGUAGES

System modeling in a large extent is a matter of handling abstract and possibly incomplete information and trying to evaluate different solutions based on the system model [5]. Mostly the easiness which a system can be modeled with a language depends on the semantics and syntax of the language used. At present, there is no complete language available for entire system modeling but there are some System-Level Design Language's (SLDL) being used extensively for following system-level design methodology.

Two major candidates for system modeling are SpecC and SystemC languages. SystemC is now based on IEEE-STD

1666-2005 [9]. SpecC and SystemC share many common features, such as dynamic sensitivity mechanism for dynamic scheduling of execution sequence. SpecC is better suited for the architecture exploration as compared to SystemC in terms of profiling and determination of execution sequence. The architecture refinement step involving allocation, partitioning and mapping is easier in SystemC compared to SpecC. SpecC and SystemC have similar capabilities in terms of support for transaction exploration, except for the determination of channel traffic which is much easily feasible using SpecC on account of its profiling capability. Both the SpecC and SystemC are equally capable for transaction refinement and the refinement process is quite similar for both of them. SpecC and SystemC have similar capabilities in terms of support for communication exploration. Both of them are equally capable for communication refinement and the refinement process is quite similar for both [6].

The SpecC language was specifically developed for the specification and design of digital embedded systems, including hardware and software portions. Built on the top of the ANSI-C programming language, the SpecC language supports concepts essential for embedded systems design, including behavioral and structural hierarchy, concurrency, communication, synchronization, state transitions, exception handling, and timing. Since SpecC is a true superset of ANSI-C, so every C program is also a SpecC program [7].

SystemC is a C++ based modeling platform supporting design abstractions at the register-transfer, behavioral, and system levels .Consisting of a class library and a simulation kernel, the language is an attempt at standardization of a C/C++ design methodology, and is supported by the Open SystemC Initiative (OSCI) [8]. IEEE-STD-1666-2005 is SystemC Language Reference Manual Standard. SystemC is an ANSI standard C++ class library for system and hardware design for use by designers and architects who need to address complex systems that are a hybrid between hardware and software. This standard provides a precise and complete definition of the SystemC class library so that a SystemC implementation can be developed with reference to this standard alone. The primary audiences for this standard are the implementers of the SystemC class library, the implementers of tools supporting the class library, and users of the class library [9].

Beside other features, SystemC models a system with logic threads running in parallel. However, its simulator does not take advantage of the parallelism. The entire design runs in a single process. SystemC is supported by OSCI. It also recently is used during direct model synthesis from system specification [15-17]. The above mentioned reasons and features are our motivation in choosing SystemC for modeling and simulation. Our future experiments show that the synthesis is also very uniform using this approach.

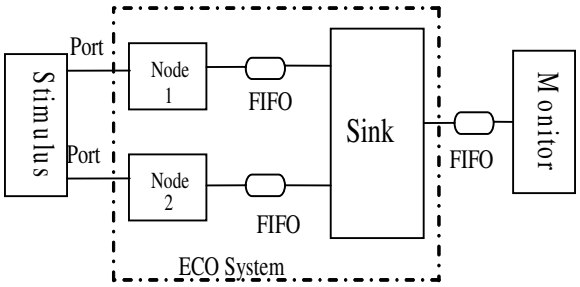

Figure 1. Overall ECO System

IV. Methods of Calculating Transition density in WSN

There are two main file formats that handle Transition density (toggle) information: VCD and SAIF. The VCD (Value Change Dump) file is generated over the simulation and provides toggle rates of the signals in the design. The SAIF (Switching Activity Interchange Format), was created by Synopsys to standardize a power format, is much more compact than the VCD, and it doesn't grow in size over the simulation. The format is mainly composed of the toggle count and the state probability.

V. WSN modeling

A sensor network is modeled as a set of heterogeneous entities. Sensor nodes deployed over the area of interest are triggered by a certain set of stimuli that eventually result in a sensor report that is transmitted to a remote base-station. Following this paradigm in a simulation environment, three main types of sensor nodes need to be created and supported: 1) target nodes that do stimulation of the sensors, 2) sensor nodes to monitor events and 3) user nodes that query the sensors and are the final destination of the target reports. To our knowledge the most interesting model is that of the sensor node in [5]. Fig. 1 shows the overall structure of WSN system.

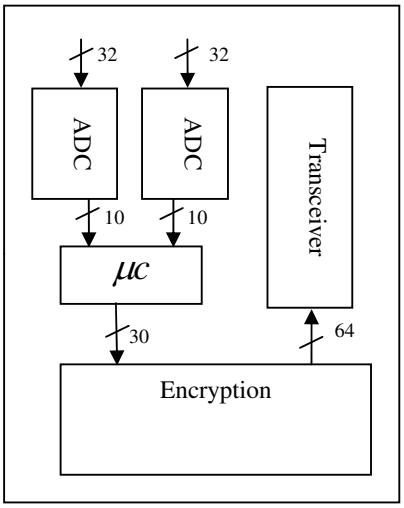

Figure 2. Node module

```
Transition Density (new_val, cur_val,) {
for (i=0; i<num_bit; i++){
if (new_val! = cur_val) {
Add to Counter_toggle of each bit
}
}
for (i=0; i<num_bit; i++){
Td [i] =TC/total_number_input_data
}
```

Figure 3. Pseudocode of the transition density calculating

We primarily used the approach proposed in [5] then we entirely start from scratch and write down a modular framework for simulation capable to include various protocols, topology, security and more features. Fig. 2 shows the structure of Node module in WSN.

VI. Methodology gathering transition density in SystemC and SpecC

In this method previous content and current content of inputs and outputs are used for calculating transition density. Transition in each bit of inputs or outputs from logic "0" to logic "1" or from logic "1" to logic "0" is registered in variables. The transition density is defined as the average number of transitions from 0->1 and 1->0 per unit time.

The pseudocode of the Transition Density calculating procedure is shown in Fig. 3.

Network is simulated with SystemC languages and comparison with network is simulated with SpecC language. Simulation time for system model is 3000 ms.

Table I shows toggle count gathered form SystemC language. Table II shows toggle count gathered form SpecC language. Table III shows toggle count gathered form VCD and SAIF file in ADC, Microcontroller and Encryption module. Table IV shows the difference between the results of SAIF (VCD) file and SystemC (SpecC) description regarding the toggle count

Calculated Transition density in all modules in SystemC and SpecC is similar. As we can see in these tables, the results obtained for the two experiments (modules ADC and Microcontroller) are almost equal for both the SAIF file format and SystemC description. In the third design (Encryption module), result of SAIF file is different with result of SystemC description. This difference is due to the unaccounted initial condition of inputs.

TABLE I

Switching activtity (toggle count) gathered with SystemC language

Language \ Module name		ADC	Micro	Encryption
SystemC	Input	70	224	216
	Output	100	225	934
	Total	170	449	1150

TABLE II

Switching activtity (toggle count) gathered with SpecC language

Language \ Module name		ADC	Micro	Encryption
SpecC	Input	70	224	216
	Output	100	225	934
	Total	170	449	1150

TABLE III

Switching activtity (toggle count) gathered with VCD and SAIF file format

File format \ Module name		ADC	Micro	Encryption
VCD and SAIF	Input	70	224	216
	Output	100	224	892
	Total	170	448	1116

TABLE IV

Difference between the results of SAIF (VCD) file and SystemC (SpecC) description

Module name	ADC	Micro	Encryption
Error	0%	~0%	~3%

As we can see in these tables, the results obtained for the two experiments (modules ADC and Microcontroller) are almost equal for both the SAIF file format and SystemC description. In the third design (Encryption module), result of SAIF file is different with result of SystemC description. This difference is due to

Transition density total in input and output of ADC module is equal to 0.0754 and 0.3448 respectively.

Transition density total in input and output of Microcontroller module is equal to 0.3862 and 0.2586 respectively.

Transition density total in input and output of DES module is equal to 0.2482 and 0.5032 respectively.

VII. Conclusions

This paper presented a method for modeling and estimation processing power (Transition Density) of Wireless Sensor Networks with SystemC and SpecC languages in RTL descriptions. Result of Simulation shows that transition density and power consumption of each module strongly depends on the size of input of system. In the proposed approach the transition density in ADC, Microcontroller and DES modules that are consumers of processing power in WSN are calculated. The results

obtained using this method is very close to commercial tool and capturing Transition Density with more accurately. The main contribution of this work is to enable the fast gathering of transition density in a transparent way to the designer, with the minimum effort, and without including new tools in the design flow. But there are important gaps in this methodology, as for example, the absence of technology library information.

REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey",Computer Networks, 38(4), pp. 393–422, March 2002.

[2] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Incrementing the World with Wireless Sensor Networks", IEEE ICASSP 2001, 4, pp. 2033-2036, 2001.

[3] J. Feng, F. Koushanfar, M. Potkonjak. "Sensor Network Architecture", Book chapter, in: 'Handbook of Sensor Networks', I. Mahgoub and M. Ilyas (eds.), CRC press, Section III, no. 12, 2004.

[4] L. W. Nagel. Spice2: A computer program to simulate semiconductor circuits. Erlm520, Univ. California, Berkeley, 1975.

[5] G. Sachdeva, R.Doemer, and P. Chou, "System Modeling: A Case Study on A Wireless Sensor Network", Technical Report CECS-TR-05-12, University of California, June 15, 2005.

[6] L. Cai, S. Verma and D. D. Gajski, "Comparison of SpecC and SystemC Languages for System Design", Technical Report CECS-03-11, University of California, Irvine, May 15, 2003.

[7] D. D. Gajski, J. Zhu, R. Dömer, A. Gerstlauer, and S. Zhao, "SpecC: Specification Language and Design Methodology", Kluwer Academic Publishers, 2000.

[8] P. R. Panda, "SystemC a modeling platform supporting multiple design abstractions," in International Symposium on System Synthesis, pp. 75–80, Sept. 2001.

[9] "IEEE STD 1666 - 2005 IEEE Standard SystemC Language Reference Manual", IEEE, pp. 1- 423, 2006.

[10] R. G. Kammer, "DATA ENCRYPTION STANDARD (DES)", Federal Information Processing Standards Publication, 1999 October 25.

[11] K. D. M¨uller-Glaser, K. Hirsch, and K. Neusinger, "Estimating essential design characteristics to support project planning for ASIC design management", In Louise Goto, Satoshi; Trevillyan, editor, Proceedings of the IEEE International Conference on Computer-Aided Design, pages 148–151, Santa Clara, CA, November 1991. IEEE Computer Society Press.

[12] Paul E. Landman and Jan M. Rabaey, "Activity-sensitive architectural power analysis", In IEEE Transactions on Computer-Aided Design of Integrated Circuits, pages 571–587. IEEE Computer Society Press, June 1996.

[13] M. Anton, I. Colonescu, E. Macii, and M. Poncino, "Fast characterization of rtl power macromodels", In IEEE Proc. of ICECS 2001, pages 1591–1594, 2001.

[14] Subodh Gupta and Farid N. Najm, "Power macromodeling for high level power estimation", In Proc. of DAC, pages 365–370, 1997.

[15] "Agility Compiler manual", For Agility 1.2, Available: http://www.celoxica.com.

[16] "SystemCrafter SC User Manual", Version 3.0.0, Available: http://www.SystemCrafter.com.

[17] "Getting Started with CoCentric SystemC Studio", Version 2002.05-SP1, synopsys, U.S.A, September 2002.