A HW-SW Design Tool for Sensor Nodes in Wireless Networks

Jingyao Zhang
ECE Department, Virginia Tech, Blacksburg, Virginia 24060
jingyao@vt.edu

Abstract

Existing simulators only support evaluations of protocols and software aspects of wireless sensor networks (sensornets) design. They cannot accurately capture the significant impacts of various hardware designs on sensornet performance. As a result, the performance benefits of customized hardware designs are difficult to be evaluated in sensornet research. To fill in this technical void, we developed SUNSHINE, a scalable hardware-software emulator for sensornet applications. SUNSHINE is the first sensornet simulator that effectively supports joint evaluation and design of sensor hardware and software performance. SUNSHINE captures the performance of network protocols, software and hardware up to cycle-level accuracy. Furthermore, SUNSHINE provides accurate power estimation for sensor nodes in wireless networks.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Performance attributes

General Terms

Design, Experimentation, Performance

Keywords

FPGA, Sensor Network, Simulator, Power

1 Introduction

Over the past few years, we have witnessed an impressive growth of sensornet applications, ranging from environmental monitoring, to health care and home entertainment. A remaining roadblock to the success of sensornets is the constrained processing-power and energy-budget of existing sensor platforms. This prevents many interesting candidate applications, whose software implementations are prohibitively slow and energy-wise impractical over these platforms. On the other hand, in the hardware community, it is well-known that the specialized hardware implementation of demanding sensor tasks can outperform equivalent software implementations by orders of magnitude. In addition, recent advances in low-power programmable hardware chips (FPGAs) have made flexible and efficient hardware implementations achievable for sensor node architectures. Hence, the joint software-hardware design of a sensornet application is a very appealing approach to support sensornets.

Unfortunately, joint software-hardware designs of sensornet applications remain largely unexplored since there is no effective simulation tool for these designs. Due to the distributed nature of sensornets, simulators are necessary tools to help sensornet researchers develop and analyze new designs. Developing hardware-software co-designed sensornet applications would have been an extremely difficult job without the help of a good simulation and analysis instrument. While a great effort has been invested in developing sensornet simulators, these existing sensornet simulators, such as TOSSIM [1], ATEMU [2] and Avrora [3] focus on evaluating the designs of communication protocols and application software. They all assume a fixed hardware platform and their inflexible models of hardware cannot accurately capture the impact of alternative hardware designs on the performance of network applications. As a result, sensornet researchers cannot easily configure and evaluate various joint software-hardware designs and are forced to fit into the constraints of existing fixed sensor hardware platforms. This lack of simulator support also makes it difficult for the sensornet research community to develop a clear direction on improving the sensor hardware platforms. The performance benefits that are available to the hardware community therefore remain hard to reach.

To address this problem, we developed a sensornet simulator, named SUNSHINE, to support hardware-software co-design in sensornets. SUNSHINE can simulate the impact of various hardware designs on sensornets at cycle-level accuracy. The performance of software network protocols and applications under realistic hardware constraints and network settings can be captured by SUNSHINE. In addition, SUNSHINE supports accurate power estimation tool (PowerSUNSHINE) for nodes in wireless networks.

2 Related Work

TOSSIM is an event-based network simulator that simulates wireless nodes at functional level. TOSSIM cannot

SenSys'11, November 1–4, 2011, Seattle, WA, USA.

capture the performance of various hardware designs or the software implementations of sensornet applications.

ATEMU [2] and Avrora [3] are the existing sensornet simulators that venture out of the event-based simulations in network domain. They provide cycle-accurate software domain simulation to evaluate the fine-grained behaviors of software over AVR microcontrollers of MICA2 sensor nodes. Though both of them are cycle-level sensornet simulators, they can only simulate MICA2 sensor nodes. They cannot accurately capture the impact of alternative hardware designs on the performance of sensornet applications. In other words, they do not support flexibility and extensibility in hardware beyond very simple parameter settings.

PowerTOSSIM [4], which is built on top of TOSSIM, can estimate power consumption for sensor nodes in wireless networks. The energy consumption of the network is profiled through the sum of each component's energy dissipation of sensor nodes. To estimate power consumption for microcontroller, one component of the sensor node, PowerTOSSIM has to estimate time duration of CPU instructions based on the assembly code generated by TinyOS applications in that TOSSIM cannot emulate CPU execution time. This estimation, however, may be fairly inaccurate.

A more accurate power estimation tool, AEON [5] is developed on top of AVRORA to profile Mica2's energy consumption. Since the CPU cycles are precisely counted by AVRORA, the microcontroller's energy consumption can be accurately captured in simulation. As a result, AEON can provide more accurate power estimation for wireless sensor networks. However, both of the power estimation tools are dedicated for fixed sensor nodes so that they cannot evaluate the power consumption of flexible sensor nodes.

With comparison, SUNSHINE is built to simulate both fixed and flexible sensor nodes in wireless networks. In addition, SUNSHINE provides a power profiling tool which aims to capture accurate power consumption.

3 Research Challenges

The challenges of developing SUNSHINE are due to the following design requirements:

Scalability: As a network simulator, SUNSHINE should support simulation of large scale networks.

Fast Prototyping: SUNSHINE should bridge the gap between design and deployment for applications of both fixed and flexible sensor nodes. TinyOS applications for fixed sensor nodes emulated by SUNSHINE can be directly loaded to actual sensor node. For flexible sensor nodes, the code running on FPGA should be synthesizable and should have the capability to be loaded and run on actual FPGA.

Reconfigurability: The architecture of sensor nodes simulated by SUNSHINE should be easily reconfigurable to evaluate different architectures of sensor nodes. Unlike other simulators that can only simulate a fixed architecture of sensor nodes, SUNSHINE should be able to design different hardware architectures of sensor nodes at configuration step before starting simulation. As a sequence, SUNSHINE should have the capability to compare performance among different hardware architectures of sensor nodes.

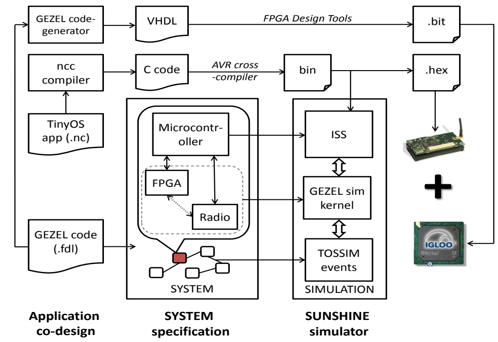

Figure 1. SUNSHINE Architecture

Flexibility: SUNSHINE should support simulation of flexible sensor nodes that have hardware coprocessors. Different from microcontrollers whose circuits are fixed, the circuitry of hardware coprocessor, i.e. FPGA, can be configured according to specific applications running on the FPGA. The flexibility of SUNSHINE means that SUNSHINE should not only support to design applications running on FPGA, but also support simulation of sensor nodes consisting of FPGAs in wireless networks.

4 SUNSHINE Architecture

Figure 1 presents the architecture of SUNSHINE, which aims to solve the above challenges. As shown in "SYSTEM specification" block, SUNSHINE can simulate a whole network, which can be composed of tens to hundreds of sensor nodes. The sensor nodes emulated by SUNSHINE can be either fixed or flexible which aims to solve the challenge of scalability. Take a flexible sensor node, which is the colored one in the figure, for example. To configure SUNSHINE simulation for the flexible sensor node, three input files are needed: network topology, which configures the topology of sensor nodes in simulation; TinyOS application, which is executed over microcontroller; and GEZEL code, which configures the hardware architecture of the sensor node as well as specifies behaviors of the coprocessor.

In simulation, the microcontroller is simulated by ISS, while the FPGA and the radio are emulated by GEZEL simulation kernel, and the wireless channel is simulated by TOSSIM. To be specific, ISS interprets binaries compiled from TinyOS applications, and interacts with GEZEL emulated FPGA. The emulated FPGA executes tasks specified by GEZEL code and sends the message back to the microcontroller. The microcontroller then puts the message to GEZEL emulated radio and let the radio send the data to other nodes according to TinyOS applications. The sensor node transmits or receives data to or from other nodes via wireless channel that is simulated by TOSSIM.

After getting satisfied simulation results, both TinyOS application and GEZEL code simulated in SUNSHINE can be compiled to binaries via code generation tool. These binaries can be loaded and run on actual hardware, which solves the fast prototyping challenge.

GEZEL, a cycle-accurate hardware description language is used to configure the hardware architecture of sensor nodes. The feature of GEZEL supports SUNSHINE to solve the challenges of reconfigurability and flexibility. In detail, the snippets of GEZEL code are listed as follows:

```
ipblock avr {    //specify microcontroller
  iptype "atm128core";
  ipparm "exec=app";
  ipparm "fcpuMhz=8";
  ipparm "asnyctimerkHz=32.768";}
ipblock m_cc2420(      //specify radio
    out fifo, fifop, cca, sfd  : ns(1);
    in  ssr,   sck,   mosi     : ns(1);
    out miso                   : ns(1)) {
    iptype "ipblockcc2420";
    ipparm "node_id = 1"; }
dp hw_top (      //configure FPGA
  in ss      : ns(1);
  in sck       : ns(1);
  in mosi      : ns(1);
  out miso     : ns(1)) {
       ...... } //codes running on FPGA
```

As shown in the snippets, three blocks are included. The first block "ipblock avr{}" specifies an 8 Mhz Atmega128 microcontroller with a 32.768Khz asynchronous timer that executes a binary "app". The second block "ipblock m_cc2420{}" specifies a CC2420 radio that uses SPI to communicate with the microcontroller. The last block "dp hw_top()" configures the applications running on FPGA and the communication protocol (SPI) used to interconnect the FPGA with the microcontroller. With the support of GEZEL, SUNSHINE can design and simulate flexible sensor nodes.

5 Power Profiling

Figure 2 illustrates the block diagram of PowerSUNSHINE, a power profiling tool of SUNSHINE. PowerSUNSHINE is associated with cycle accurate sensor nodes emulated by SUNSHINE. When SUNSHINE is simulating applications of sensor nodes, PowerSUNSHINE breaks down sensor nodes into components, calculates power/energy consumption of each component, and then adds all the components power/energy consumption together.

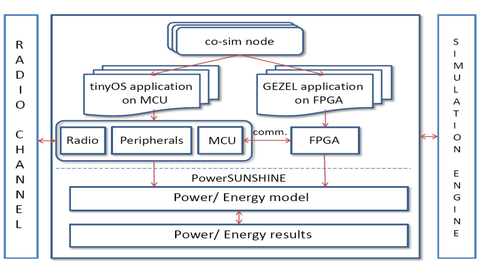

Figure 2. Block diagram of PowerSUNSHINE

To be specific, if PowerSUNSHINE is applied for fixed sensor nodes in simulation, it tracks cycle accurate activities of every component, and uses the power/energy model to calculate the total power/energy consumption of the nodes according to their component activities. Compared with fixed node, flexible node has an extra programmable FPGA. If PowerSUNSHINE is applied for the flexible nodes, the additional power/energy dissipation of FPGA should be considered. Therefore, the total power/energy profiling should contain the power/energy consumption of both fixed hardware components and the reconfigurable FPGA. By establishing a power/energy model for each hardware component, PowerSUNSHINE can estimate the power/energy consumption of arbitrary platform designs.

6 Ongoing and Future Work

The results are not presented due to the page limit. For more information, please check [7].

Currently, we focus on simulating the network, software and hardware behaviors of MICAz nodes. Based on the characteristics of GEZEL, different instruction-set simulators can be connected with GEZEL to emulate the microcontrollers' hardware and software performance. We have already interfaced GEZEL with ARM, 8051 microcontroller, etc. GEZEL is also able to connect to other instruction-set simulators to capture other existing sensor motes hardware and software behaviors in a networked context.

Different flexible nodes hardware architecture can be configured and emulated in SUNSHINE, such as using a different microcontroller, radio and FPGA, as well as different communication protocols among the components, etc. Currently, the communication protocol between microcontroller and FPGA is SPI, other protocols such as I^2C, UART, and parallel, can also be used in platform construction and be emulated by SUNSHINE.

We are designing a PCB that mainly contains a microcontroller, a radio and a low power FPGA. We will use PowerSUNSHINE to explore potential power savings from flexible sensor platforms and use the designed board to do validation.

Biographical sketch: Jingyao Zhang is a Ph.D. student in ECE department, Virginia Tech. Her advisor is Dr. Yaling Yang. She would submit her dissertation in May, 2013.

7 References

[1] P. Levis, N. Lee, M. Welsh, and D. Culler. Tossim: accurate and scalable simulation of entire tinyos applications. In *ACM Sensys*, 2003.

[2] J. Polley, D. Blazakis, J. McGee, D. Rusk, and J. Baras. Atemu: a fine-grained sensor network simulator. In *SECON*, pages 145–152, October 2004.

[3] B. L. Titzer, K. D. Lee, and J. Palsberg. Avrora: Scalable sensor network simulation with precise timing. In *IPSN*, 2005.

[4] V. Shnayder, M. Hempstead, B. Chen, G. W. Allen, and M. Welsh. Simulaitng the power consumption of large-scale sensor network applications. In *ACM SenSys*, 2004.

[5] O. Landsiedel, K. Wehrle, and S. Gotz. Accurate Prediction of Power Consumption in Sensor Networks. In *IEEE EmNets*, 2005.

[6] J. Zhang, Y. Tang, S. Hirve, S. Iyer, P. Schaumont, and Y. Yang. A Software-Hardware Emulator for Sensor Networks. In *IEEE SECON*, 2011.

[7] SUNSHINE webpage. http://rijndael.ece.vt.edu/sunshine/index.html