SystemC-AMS Models for Low-Power Heterogeneous Designs: Application to a WSN for the Detection of Seismic Perturbations.

Antoine Leveque, Francois Pecheux, Marie-Minerve Louerat, Hassan Aboushady, Michel Vasilevski
University Pierre & Marie Curie LIP6 Laboratory, 75252 Paris, France
{antoine.leveque,francois.pecheux,marie-minerve.louerat,hassan.aboushady,michel.vasilevski}@lip6.fr

Abstract

The paper presents a system-level approach for the modeling and simulation of a genuine heterogeneous system composed of individually powered Wireless Sensor Network nodes. The models are written in SystemC-AMS, an open-source C++ extension to the OSCI SystemC Standard dedicated to the description of AMS designs containing digital, analog, RF hardware as well as physical, optical or chemical IPs.
The paper is composed of two parts. The first part details the study case, a system of WSN nodes that can monitor a physical seismic perturbation, transmit information on this perturbation to other nodes by means of 2.4 GHz RF communication links, and finally compute the epicenter of the perturbation by asking the 32-bits processor embedded in a node to solve the system of nonlinear equations relative to the triangulation algorithm. Each node is powered by an autonomous kinetic battery model.
The second part presents the corresponding implementation in SystemC and SystemC-AMS, and gives an insight on how all the disciplines are elegantly intertwined, with an optimal model of computation associated to each hardware component of the simulated system. This part proves that the joint use of the Timed-DataFlow (TDF) Model of Computation for AMS parts, RF baseband equivalent for RF parts, and Communicating Synchronous Finite State Machines (CSFSM) for digital parts significantly reduces the simulation time while keeping excellent accuracy and code readability. After some results, the paper concludes on the possibilities offered by this approach in terms of validation and optimization of heterogeneous systems using an open-source simulation framework.

1 Introduction

One of the great challenges of the next decade consists undoubtly in the successful design of heterogeneous systems that mix digital, analog electronics and various domains like physics, biology or chemistry at a nanometric scale on a single die. As time-to-market periods becomes shorter, the ability to model and simulate these multidiscipline systems as early as possible in the design cycle reduces architecture exploration issues and design errors.

However, major industrial and academic players in their relative domains are today still confronted with the lack of a unified design framework that can be used efficiently by system designers with various educational backgrounds to conceive and produce such state-of-the-art systems. This paper is an attempt to demonstrate how two complementary technologies can profitably be used together, namely SoCLib, a library of interoperable models dedicated to the modeling of digital System On Chip components and SystemC-AMS, the analog mixed-signal extension to SystemC. Figure 1 shows the retained layered approach. Digital IPs and AMS IPs synchronize by means of the "synchronization" layer provided by SystemC-AMS.

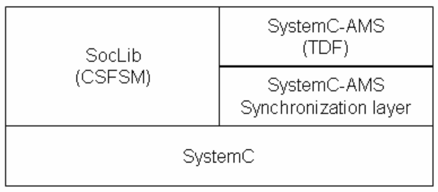

Figure 1: Different technologies cooperate for heterogeneous design.

1.1 The SocLib environment

SocLib is a library of digital models that clearly addresses the modeling of MultiProcessor Systems on Chips (MPSoC). As 80% to 90% of such designs is purely digital. and because the number of cores in a single die keeps increasing, the models have been developed bearing in mind the need for simulation performance. The models in SocLib are BCA-compliant (Bit-Cycle Accurate), respect the SystemC description rules [1] and are interoperable, i.e. they strictly follow the standard VCI/OCP protocol that allow any digital IP of the library to be interconnected with others through the use of a Network on Chip (Noc). To achieve this performance, all the models have been written as finite state machines (FSM). A given component may contain several Moore FSMs, and as such, can be fully represented as a set of two combinatorial functions (the transition function, and the Moore output generation function) and a register containing the component current state. The simulation of a whole SoC merely consists, during elaboration of the simulator, in setting up a network of Communicating Synchronous FSMs (CSFSM). On the rising edge of clock, all the transition functions of all the models in the digital design are called and, on the falling edge of clock, all the Moore output generation functions are called. The simulation cycle of such a simulation is therefore quite canonical and the discrete event algorithm performed by the SystemC simulation kernel has very low overhead because the event list is reduced to the rising and falling clock events. Processors are implemented as Instruction Set Simulators (ISS), that can simply be interconnected to the memory hierarchy (cache L1, cache L2 and main memory). Processor models include MIPS32, PowerPC405, ARM7, Sparc V8, Nios, etc. Furthermore, the SocLib library is freely available [2].

1.2 SystemC-AMS

The new SystemC-AMS library of C++ classes for modeling "More than Moore" Systems is particularly well suited for modeling such heterogeneous designs. The work presented herein uses the experimental version of SystemC-AMS, starting point for the standardization to come [3] [4] [5]. The SystemC-AMS library is an extension of SystemC that soundly manages several Modeling Formalisms and Models of Computation (MoC). Parts of the system can be modeled using appropriate means such as Synchronous Dataflow (SDF), Linear Networks (LN), and can interact together or with other MoCs (i.e. Discrete Event of SystemC) through the SystemC-AMS synchronization layer. Using this layered structure, interacting parts of a complex heterogeneous system can be modeled and simulated within their optimal methodology and solving algorithm. From a system engineer viewpoint, this approach allows for a higher modeling efficiency and faster simulations by several orders of magnitude.
The available SystemC-AMS prototype currently integrates two MoCs. SystemC-AMS Synchronous DataFlow, is also known as Timed DataFlow (TDF), and can be used to describe complex non-conservative (signalflow) behaviors. The second domain, not described in the paper, is the conservative description, which can be used for Electrical Linear Networks (ELN).
Unlike the untimed synchronous model inherited from [6], TDF is a discrete-time modeling style which considers data as sampled signals. These signals are updated at discrete points in time and carry discrete or continuous values, like amplitudes.
The set of connected TDF modules forms a directed graph, called a TDF cluster. TDF modules are the vertices of the graph, and TDF signals correspond to edges. Each TDF Module involved in the cluster contains a C++ member method, named **processing()**, that computes a mathematical function that depends only on its direct inputs. The overall equation computed by the cluster is therefore defined as the mathematical composition of the functions of the involved TDF modules in the appropriate order.
A TDF module may have several input and output ports. TDF signals are used to connect ports of different modules together.
A given TDF module function is calculated (or "fired" according to the SDF formalism) if and only if there are enough data samples available at the input ports. In this case, the samples computed by a TDF module are written to the appropriate output ports. In SDF, the number of samples read from or written to the module ports is constant. In TDF, a timestamp is associated to each sample data, and a TDF module can produce more than one data sample.
Provided the associations performed on the ports or the modules of a TDF graph are compatible, the order and number of samples (sampling rate) in a TDF cluster is known for each calculation. This order can be statically determined before the simulation starts and corresponds to a static schedule of the TDF cluster. This static schedule, computed once for all during the simulator elaboration is at the origin of the simulation speedup over traditional simulation kernels. Thus, and more formally, a TDF cluster can be defined as the set of connected TDF modules which belong to the same static schedule.

2 Detection of seismic perturbations

The modeled WSN system is presented in Figure 2. It aims at determining the two-dimensional coordinates of the epicenter of a seismic perturbation on the earth soil. The WSN consists of four independent nodes *N0*, *N1*, *N2* and *N3* located at well known positions on the grid matrix representing the monitored ground surface. The nodes continuously monitor seismic activity with their builtin acceleration sensor. The seismic perturbation is modeled as a gaussian-like pulse with an initial amplitude on the grid with constant radial velocity. The initial pulse is then propagated both in time and space through the use of the wave equation. Once the acceleration sensor of a node detects the presence of the seismic perturbation, the node propagates its identifier as well as its internal timestamp to other nodes thanks to its 2.4 GHz RF transceiver and a noisy communication channel. Each node thus gathers information on all the consecutive timestamps of seismic perturbations. Thanks to a triangulation algorithm, each node is able to compute the epicenter. Nodes are totally equivalent from the hardware and software viewpoints, have an internal real-time clock initialized with the very same initial timestamp 0 (quite an unrealistic assumption) and are powered by a battery source modeled as a kinetic battery. In this preliminary model, power estimation is coarse grain and only two power activities are in fact distinguished: RF transceiver emitter enabled and not enabled. At last, the protocol retained to prevent collisions of RF messages is TDMA (Time Division Multiple Access).

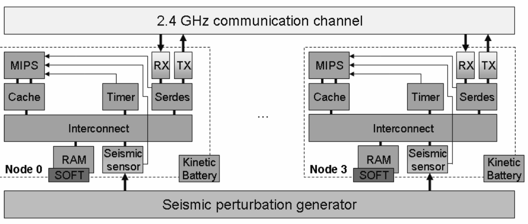

Figure 2: The complete WSN, consisting of four communicating nodes N0 to N3.

2.1 Generation of seismic stimulus

The perturbation is modeled as a seismic pulse with an amplitude f that propagates to all points of the ground surface by means of the wave equation. The following partial differential equation can be used to describe any physical phenomena involving the propagation of a wave and describes how the amplitude of a wave change based on conditions in its immediate neighborhood:

$$c^2 . \frac{d^2 f}{dt^2} = \frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2} \tag{1}$$

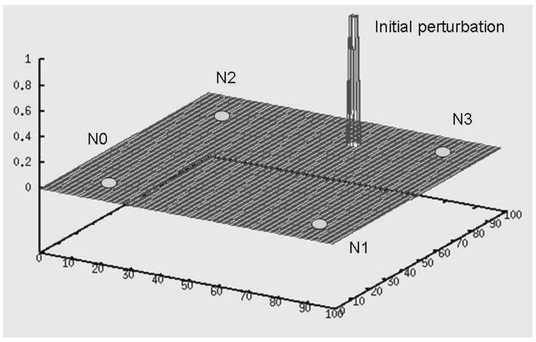

Figure 3: The surface grid, with initial seismic perturbation, and nodes locations.

The equation applies to every point of the ground surface so it can be used to model the dynamic behavior of a wave, the term on the left represents the rate at which the amplitude is accelerating up or down at a given point, while the right term sums acceleration across each spatial dimension at the same point. Instead of defining the value of this equation everywhere (continuously), only selected points are considered. The more closely these points are spaced the more accurate an approximation to the continuous case and the more time consuming the computation. Figure 3 shows the initial perturbation, as well as the locations of the 4 sensors on a 100 by 100 grid.

2.2 Seismic Sensor

In the present paper, the seismic sensor is trivial unlike [7], and the read amplitude value is directly converted into its digital equivalent by means of a second order sigma-delta 1-bit modulator with return-to-zero feedback [8], and a decimator using a third order FIR2 [9], that can be parameterized to generate a n-bit word, as shown in Figure 4 [10].

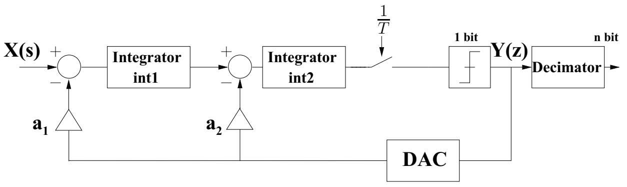

Figure 4: Second order sigma-delta continuous-time modulator and decimator.

2.3 Microcontroller, system-on-chip

The SocLib library has been used to completely design the digital part. This part is composed of a 32-bit MIPS32 processor and its associated instruction and data L1 caches, a RAM containing the embedded application code and data, a timer used to generate TDMA interrupts, the digital part of the seismic sensor, and the component responsible for the serialization/deserialzation (serdes) of the RF data. The MIPS32 may receive three interrupts, from the timer (to identify if the node is in its TDMA timeslot), serdes (when RF data has been received) and seismic sensor (when a given amplitude threshold has been reached, indicating the presence of the perturbation at the sensor location) respectively.

2.4 2.4 Ghz QPSK RF transceiver

The RF transceiver is responsible for converting the digital bitstream emitted by the serdes into RF information (RF transmitter) and vice versa (RF receiver). The RF model uses a coherent QPSK (Quadrature Phase Shift Keying) transmission scheme, as explained in [11] and shown in figures 5 and 6 with a f_c =2.4 GHz carrier frequency and a f_b =2.4 MHz data frequency. AWGN (Additive White Gaussian Noise) allows to take into account channel noise in the modeling of the RF communication channel and is necessary for calculating the fundamental RF characteristic BER (Bit Error Rate) with respect to SNR (Signal-to-Noise Ratio). In this nearly ideal modeling, the power amplifier (PA) and the low noise amplifier (LNA) models are very straightforward. Likely, the communication channel is considered as an ideal gain block with an additive white Gaussian noise (AWGN).

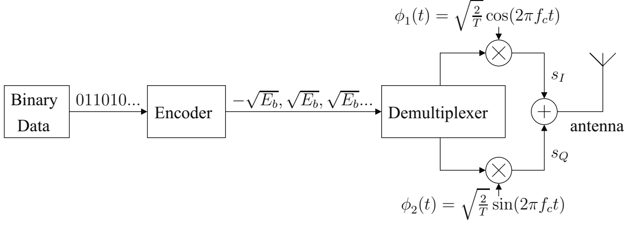
$\phi_1(t) = \sqrt{\frac{2}{T}}\cos(2\pi f_c t)$
Binary Data
011010...
Encoder
$-\sqrt{E_b}, \sqrt{E_b}, \sqrt{E_b}...$
Demultiplexer
s_I
s_Q
antenna
$\phi_2(t) = \sqrt{\frac{2}{T}}\sin(2\pi f_c t)$

Figure 5: QPSK RF transmitter.

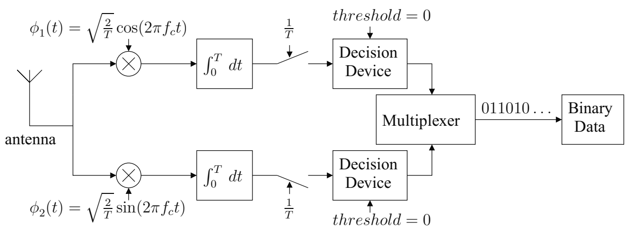

Figure 6: QPSK RF receiver.

To prevent this simulation time to become too prohibitive, and hence to validate and optimize parts of the WSN, a common technique [12] is to define an equivalence between the initial 2.4 GHz RF signal and its baseband representation to remove the carrier from the RF signal expression :

$$\begin{aligned} x(t) = DC + I_1\cos(\omega t) + I_2\cos(2\omega t) + I_3\cos(3\omega t) \\ +Q_1\sin(\omega t) + Q_2\sin(2\omega t) + Q_3\sin(3\omega t) \end{aligned} \quad (2)$$

In the baseband equivalent transmission scheme, the only data actually transmitted over the RF channel are the 7 coefficients of equation 2 that represent signal harmonics and their associated 2nd and 3rd order distortions, at a rate that is ten thousand times smaller than in a direct/naive simulation.

2.5 Battery modeling

The application uses the intuitive Kinetic Battery Model (KiBaM) from Manwell and McGowan [13]. it is called kinetic because it relies on the use of chemical kinetic differential equations. As shown in Figure 7, the battery charge is distributed over two wells: the available-charge well and the bound-charge well.

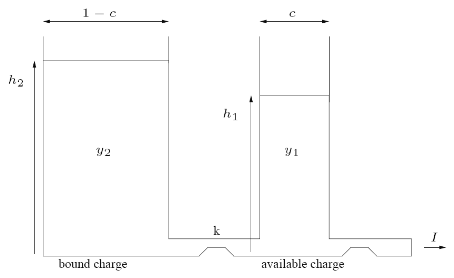

Figure 7: The battery model from Manwell and McGowan, taken from [14], and built upon the available-charge and bound-charge wells.

The available charge well supplies electrons directly to the load, the WSN node, while the bound-charge well supplies electrons only to the available-charge well. The rate at which charge flows between the wells depends on the difference in heights of the two wells, and on a parameter *k*. The parameter *c* gives the fraction of the total charge in the battery that is part of the available-charge well. The change of the charge in both wells is given by the following system of differential equations:

$$\begin{cases} \frac{dy_1}{dt} = -I + k.(h_2 - h_1), \\ \frac{dy_2}{dt} = -k.(h_2 - h_1) \end{cases} \quad (3)$$

with initial conditions $y_1(0) = c.C$ and $y_2(0) = (1-c).C$, where *C* is the total battery capacity. For *h1* and *h2* we have: $h1 = y1/c$ and $h2 = y2/(1-c)$. When a load *I* is applied to the battery, the available charge reduces, and the difference in heights between the two wells grows. When the load is removed, charge flows from the bounded-charge well to the available-charge well until *h1* and *h2* are equal again. So, during an idle period, more charge becomes available and the battery lasts longer than when the load is applied continuously.

3 SystemC-AMS implementation

The presented system has been implemented in SystemC and SystemC-AMS and is composed of approximately 100 C++ and include files. Interesting results concerning some parts of the design have already been published [15]. This section shows how the battery model and the wave equation can be coded in SystemC-AMS, and illustrates the SystemC-AMS constructs used to interface the digital and AMS worlds. The structure of the embedded software is also given.

3.1 Kinetic Battery model

The listing for the SystemC-AMS implementation of the battery model is given in Listing 1. The mathematical integrations expressed in the genuine battery model are computed using the sampling period or a multiple of this sampling period associated to the TDF cluster as the parameter **dt** of the ODEs (lines 22 to 26), The current *I* is declared as a TDF input at line 6. The value of I corresponds to the current evaluation that is performed at each cycle of the digital clock.

Listing 1: The Kinetic Battery Model in SystemC-AMS

```
 1 #ifndef KIBAM_H
 2 #define KIBAM_H
 3
 4 SCA_SDF_MODULE (kibam)
 5 {
 6    sca_sdf_in < double > in_i;
 7    sca_sdf_out < double > py1;
 8    sca_sdf_out < double > py2;
 9
10    double y1, y2;
11    double dy1, dy2;
12    double dt;  // time step
13
14    double C;
15    double k;
16    double c;
17    double current;
18
19    void sig_proc () {
20      current= in_i.read() ;
21
22      dy1 = (-current + k* (y2/(1-c)-y1/c)) * dt;
23      dy2 = (-k * (y2/(1-c)-y1/c) )* dt;
24      y1 += dy1;
25      y2 += dy2;
26          py1.write(y1);
27          py2.write(y2);
28    }
29
30    void init() {
31          dt=in_i.get_T().to_seconds();
32      }
33
34      SCA_CTOR (kibam) {
35            y1=4500.0 ;
36            y2=2700.0 ;
37            k=4.5e-5;
38            c=0.625;
39      }
40
41  };
42  #endif
```

3.2 Seismic stimulus generator

The seismic stimulus generator used to model the application environment is also a SystemC-AMS TDF module. It is connected to the seismic sensor of each node with SystemC-AMS ports *sca_sdf_out out_sensor* on line 3. The environment is represented by a 3-dimensionnal matrix (two dimensions for space and one for time). The amplitude of the wave at a given point is calculated thanks to the discretized version of equation 1:

$$f_{x,y,t+1} = c^2.(f_{x+1,y,t} + f_{x-1,y,t} + f_{x,y+1,t} + f_{x-1,y,t} \\ -4.f_{x,y,t}) - f_{x,y,t-1} + 2.f_{x,y,t} \tag{4}$$

This equation shows that it is possible to evaluate the amplitude of a point of the matrix at time t_{i+1} knowing the amplitude values of its neigbours at time t_i and the previous amplitude of the point at time t_{i-1}. Each time the TDF module is activated by the Systemc-AMS scheduler the **processing()** function as described in listing 2 is executed. First, all points of coordinates (x,y) on the grid are updated (lines 8 to 14). wave[x][y][1] and wave[x][y][0] represent the values at the previous iterations of the amplitude of point (x,y). After this update phase the value corresponding to the position of the sensor is written to each node connected (lines 13 to 16). Then the current value becomes a previous one (line 22 to 26).

Listing 2: The seismic stimulus generator in SystemC-AMS

```
1   SCA_SDF_MODULE (wavegen)
2   {
3           sca_sdf_out < double > out_sensor[4];
4   ...
5   void processing () {
6     int x,y,c;
7
8     for (x=1;x<WAVE_SIZE-1;x++) {
9       for (y=1;y<WAVE_SIZE-1;y++){
10        wave[x][y][2]=2.0*wave[x][y][1] -
11          wave[x][y][0] +
12          cd*cd*(wave[x+1][y][1] + wave[x-1][y][1] +
13                 wave[x][y+1][1] + wave[x][y-1][1] -
14                 4.0*wave[x][y][1]);
15      } }
16    ...
17    for (c=0 ; c < NB_SENSORS; c++)
18      out_sensor[c].write(
19           wave[pos_x_sensor[c]][pos_y_sensor[c]][2]
20                          );
21
22    for (x=0;x<WAVE_SIZE;x++) {
23      for (y=0;y<WAVE_SIZE;y++) {
24        wave[x][y][0]=wave[x][y][1];
25        wave[x][y][1]=wave[x][y][2];
26      } }
27  }
28  ...
29  }:
```

3.3 Synchronization between digital and RF parts

Because SystemC-AMS uses its own representation of the simulation time, it is not recommended to interface directly a SystemC-AMS TDF module to a SystemC module due to possible synchronization issues. This is why SystemC-AMS provides dedicated converter ports (*sca_scsdf_out* or *sca_scsdf_in*) which ensure that the value written or read will be the right one at the right time. As an example the listing 3 shows the TDF module for the the Power Amplifier pa, part of the RF transceiver. The RF transmission chain is modeled using baseband equivalent which is represented in this module by the template parameter <*BB*> (lines 4 and 5). The in port *en* line (3) is connected to a control port from a SystemC module which can (des)activate the RF tranmitter. Lines 7 to 14 show how the processing function of the module uses this enable flag to send a valuated BB sample (corresponding to the case the node is actually emitting) or a null BB (when the RF transmitter is deactivated). All the required synchronization is done by the *sca_scsdf_in* port so the use of digital ports is transparent from the designer viewpoint.

Listing 3: The pa TDF module

```
1   SCA_SDF_MODULE (pa)
2   {
3     sca_scsdf_in < bool > en;
4     sca_sdf_in < BB >in;
5     sca_sdf_out < BB >out;
6    ...
7    void processing () {
8      ...
9       BB input=in.read();
10      if (en.read()==true)
11        out.write (GAIN(input));
12      else
13        out.write (...);
14    }
15   ...
16  };
17  #endif
```

3.4 Embedded software

The embedded application, responsible for initializing peripherals and managing interrupts, is cross-compiled with GNU GCC for the MIPS32 processor. The application programmer can access the registers of the digital components (e.g. SERDES) directly with the dedicated set anf get function **soclib_io_set()** and **soclib_io_get()**. In this application the 4 nodes communicate using a simple TDMA protocol implemented in software and paced by an internal timer. One can examine the code of the main function given in listing 4. Each node has a temporal slot associated which can be used to broadcast a message to every other nodes. A timer is setup to raise an interrupt which indicates that a new temporal slot begins (lines 7). If this slot is the one associated to the node (line 10) then an RF message can be sent (lines 11 to 21). If it is not the present TDMA timeslot, the node is in a reception mode waiting for a message from the other nodes (lines 22 to 24). The message is a 32-bit word, 4-bit for the identification of the node sending it, and the others for data (lines 14 and 15). The data in this case is the time at wich the seismic sensor detects that the amplitude is over a given threshold. When an interrupt is generated by the seismic sensor of the node (line 11) or when a message from an other node is received, the corresponding information is saved in a sensor table by a call to the **fill_table(...)** function. As soon as all the sensors have transmitted their information (line 9), the node computes the epicenter (lines 26 to 29).

Listing 4: The main() function of the embedded software

```
1  int main(int argc, char**argv)
2  {
3    uint32_t data;
4    init();
5    printf("Sensor %d is monitoring...\n", id);
6    while (1) {
7      if (tdma_trigger==1) {
8        tdma_trigger=0;
9        if(!flag_calcul){
10         if (tdma_slot==id) {
11           if(has_value) {
12             soclib_io_set(
13                  base(UART), UART_CTRL, 2);
14             data=(id<<28) |
15                  (sensor_cpt & 0x0fffffff );
16             printf("sending data %x\n",data);
17             soclib_io_set(
18                  base(UART), UART_DATA, data);
19             fill_table(id,sensor_cpt);
20             has_value=0;
21           } }
22         else
23           soclib_io_set(
24                base(UART),UART_CTRL,5);
25       }
26       else {
27         irq_disable();
28         compute_epicentre();
29       } } }
30   return 0;
31 }
```

3.5 Simulation results

For a simulated microcontroller clock of 100 Mhz, the simulation of this four nodes WSN lasts approximately two minutes on a AMD X2 PC running Linux at 2.5 GHz with 1Gb of RAM. There are 3 TDF clusters, one for RF, one for the seismic perturbation, and one managing power. Figure 8 shows the evolution of the charge in each battery. As the current model is very simplified, the recovery effect of the battery modeled by the two wells can not be directly pointed out. The slopes of the plots correspond to the two cases, RF transmitter disabled (smooth slope) and enabled (steep slope).

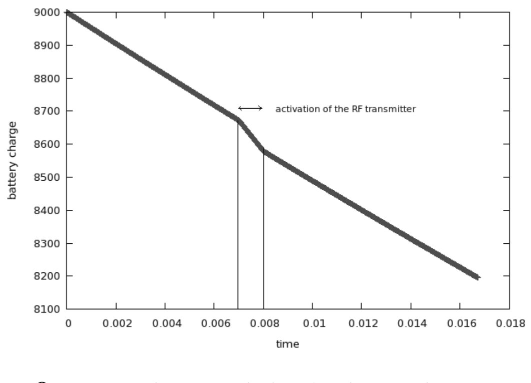

Figure 8: Battery charge evolution for the 4 nodes.

In the RF part, an analysis has been performed to display bit error rate according to SNR variation. A theorical BER has been computed from AWGN characteristics and has been successfully compared to simulation results. Figure 9 shows the perfect match between simulation and theorical results. This analysis has been extended in [10] to transceiver impairments.

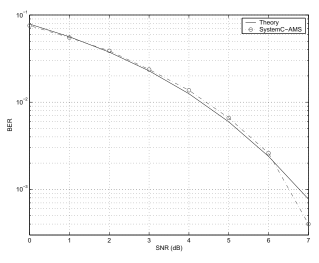

Figure 9: Bit error rate for QPSK transmission through an AWGN channel.

4 Conclusion

The paper shows that the system simulation of a complete WSN that encompasses several domains is actually possible with open-source tools, with excellent accuracy. Model interoperablity and performance are obtained through the use of C++, SystemC and SystemC-AMS, and simulation times (when using state of the art RF modeling techniques) seem extremely encouraging. Ongoing research focuses on adding a real communication protocol to handle asynchronicity between nodes. A much more accurate model for power estimation in a WSN is also being developed.

References

[1] "SystemC," http://www.systemc.org.

[2] "An open platform for virtual prototyping of multi-processors system on chip," http://www.soclib.fr/.

[3] "SystemC-AMS," http://www.systemc-ams.org.

[4] A. Vachoux, C. Grimm, and K. Einwich, "Towards Analog and Mixed-Signal SOC Design with SystemC-AMS," *IEEE International Workshop on Electronic Design, Test and Applications (DELTA)*, Jan. 2004.

[5] ——, "Analog and Mixed Signal Modelling with SystemC-AMS," *IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2003.

[6] "Ptolemy II," http://ptolemy.eecs.berkeley.edu/ptolemyII/.

[7] E. Markert, M. Dienel, G. Herrmann, D. Müller, and U. Heinkel, "Modeling of a new 2D Acceleration Sensor Array using SystemC-AMS," *Internationnal MEMS Conference (IMEMS)*, May 2006.

[8] H. Aboushady, F. Montaudon, F. Paillardet, and M. M. Louerat, "A 5mW, 100kHz Bandwidth, Current-Mode Continuous-Time Sigma-Delta Modulator with 84dB Dynamic Range," *IEEE European Solid-State Circuits Conference (ESSCIRC) Florence,Italy*, Sep. 2002.

[9] H. Aboushady, Y. Dumonteix, M. Louerat, and H. Mehrez, "Efficient Polyphase Decomposition of Comb Decimation Filters in Sigma-Delta Analog-to-Digital Converters," *IEEE Trandactions on Circuits and Systems-II (TCASII)*, Oct. 2001.

[10] M. Vasilevski, F. Pecheux, H. Aboushady, and L. de Lamarre, "Modeling Heterogeneous Systems Using SystemC-AMS, Case Study: A Wireless Sensor Network Node," *IEEE International Behavioral Modeling and Simulation Conference (BMAS)*, Sep. 2007.

[11] simon haykin, *communication systems, 3rd ed.* Wiley.

[12] D. G.-W. Yee, "A design methodology for highly-integrated low-power receivers for wireless communications," Ph.D. dissertation, University of California, Berkeley, 2001.

[13] J.Manwell and J.McGowan, "Lead acid battery storage model for hybrid energy systems," *Solar Energy, vol.50, pp.399-405*, Sep. 1993.

[14] M. Jongerden and B. Haverkort, "Battery modeling," Enschede, January 2008. [Online]. Available: http://doc.utwente.nl/64556/

[15] M. Vasilevski, F. Pecheux, N. Beilleau, H. Aboushady, and K. Einwich, "Modeling an Refining Heterogeneous Systems with SystemC-AMS: Application to WSN," *Proceedings IEEE Date 2008 Conference, Munich*, March 2008.