Subproject A2: Examination of the Applicability of SystemC-AMS for the Description of MEMS

Markert, Erik; Schlegel, Michael; Herrmann, Göran; Müller, Dietmar
TU Chemnitz, Faculty of Electrical Engineering and Information Technology,
Professorship Circuit and System Design
e-mail: erik.markert@etit.tu-chemnitz.de

1 Introduction

SystemC [1] provides elaborate methods for the description of digital hardware/software systems from functional downto register-transfer level. But today's systems usually also contain analog parts, they are heterogeneous. So an extension of the language as well as of the Models of Computation (MoC) is necessary. A SystemC Study Group currently develops the analog SystemC extension library SystemC-AMS. Now a first version is released by Fraunhofer Gesellschaft EAS/IIS Dresden, including the algorithms planned for phase 1 of the White Paper [2], [3]. Scope of application of this version are communication systems.

This paper discusses whether this version is already qualified for the description of MEMS although this is still planned by SystemC-AMS developers for future versions.

With SystemC-AMS 0.12, a vibration sensor array shall be described. In VHDL-AMS already exists a verified model and was presented in the last Annual Report [4]. There a brief system description can be found.

2 Modeling with SystemC-AMS

In the used SystemC-AMS version two possibilities for describing analog parts exist: conservative networks and static dataflow (SDF) networks.

The single elements of SDF networks communicate via directed data streams. All ports are to be declared as inputs or outputs. In the present version, all feedbacks have to be decoupled by a delay component. Communication systems are usually strong oversampled so on this field of application the delay does not provoke a significant error. It is to be explored if this will be the case in MEMS Design. For the description of transfer functions and differential equations, Laplace Transfer Functions can be used.

Beside SDF-modeling, simple conservative networks may also be described. They can consist of R, L, C, current and voltage sources. For connection to SDF networks, SDF-signal-driven sources and voltmeter may be used.

Mixed forms with digital SystemC 2.0 and hierarchical dispositions are possible.

3 Implementation

3.1 Overall System

SystemC-AMS is an extension of SystemC and therewith of C++, too. So only digital components can be described in SystemC. There the use of edge-sensitive processes offers a gain of simulation time and description effort.

Therefore in the vibration sensor array the fuzzy-pattern-classifier is described using digital SystemC while sensor and selector are pure analog SystemC-AMS components. Hierarchical modules are used for detector and microcontroller. Figure 1 shows the system architecture. In the following the analog module sensor is specified to show the new capabilities of SystemC-AMS.

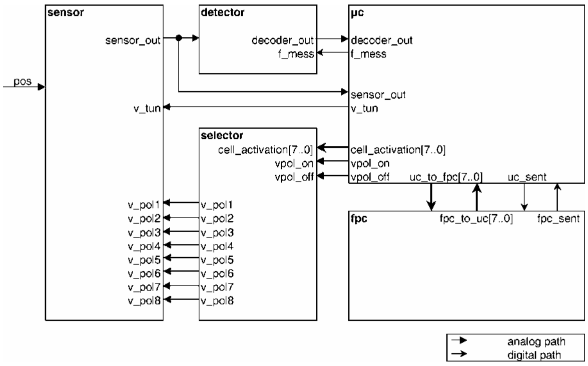

Figure 1: System Overview[5]

3.2 Module Sensor

The sensor converts mechanical signals to electrical ones. For this purpose, it includes a number of spring-mass-resonators which may be activated seperately. The output u_out (SDF-type double) provides a voltage for the following component.

In the used version of SystemC-AMS, it was impossible to describe user-specific conservative behaviour models. This is planned for one of the next language versions. Currently, only linear conservative behaviour in terms of the basic elements resistor, capacitor and inductivity is implementable. By using of C++-heredity other domains can be derivated.

The sensor consists of springs, masses and damps. These mechanical elements (v, F, k, m, c) are converted to electrical values (U, I, L, C, R). The sensor's input is a displacement. But in analogy, the current conforms to the velocity so the input signal must be derivated. The component sensor is structured as shown in figure 2.

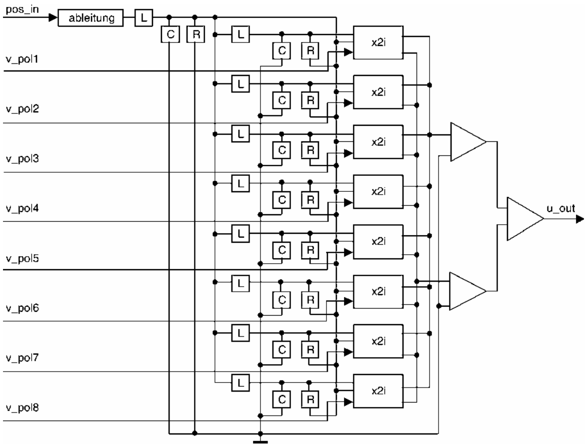

Figure 2: Structure of Sensor Module [5]

Listing 1 shows the description of a spring-mass-attenuator-component in SystemC-AMS.

```
ELSDF_MODULE(fmd) {
  // ----- ports -----
  elec_port in;
  elec_port out;
  elec_port gnd;
  sdf_inport<double> v_tun; //not used
  // ------ parameter ------
  sdf_para<double> m,d,k;
  // ----- components -----
  R* r1; L* l1; C* c1;
  // ----- constructor -----
  ELSDF_CTOR(fmd)
  {
    l1 = new L(this->k); //spring
    l1->a(pos_m);
    l1->b(pos_in);
    c1 = new C(this->m); //mass
    c1->a(pos_m);
    c1->b(gndm);
    r1 = new R(this->d); //attenuator
    r1->a(pos_m);
    r1->b(pos_in);
  };
};
```

Listing 1: Description of a part of the sensor module

4 Simulation Results

For result valuation, it is important to point out again that the used SystemC-AMS version is made for communication systems only, micromechanical problems will be supported in a future version.

Figures 3 and 4 show the voltage at sensor output in SystemC-AMS and VHDL-AMS. Input waveform is a step, all simulation parameters are identical (step height 1 µm, step width 10 ns, integration algorithm trapezoidal).

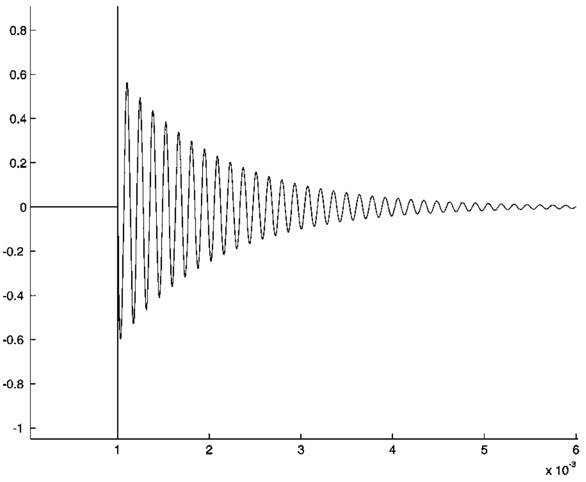

Figure 3: Step response in SystemC-AMS

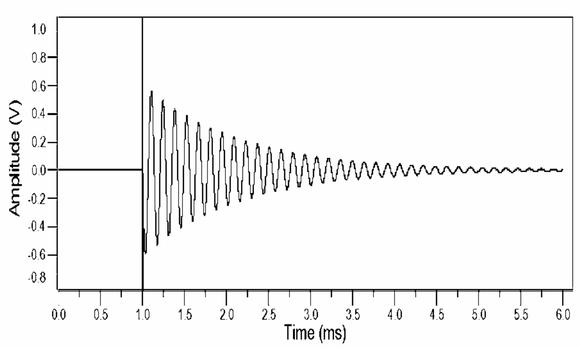

Figure 4: Step response in VHDL-AMS

Both pictures show similar behaviour. But the oscillation amplitudes differ up to 2 %. Reason for this deviation is the one cycle delay in the derivation component in the SystemC-AMS description while VHDL-AMS solves the non-linear equation system. The SystemC-AMS-implementation claims much lesser calculation time (2 min 24 s) than VHDL-AMS (24 min 2 s) on a SunBlade 2900.

The result of the digital fuzzy-pattern-classification is nearly the same. Table 1 shows the affiliation to the attribute classes.

Table 1: classification results

	VHDL-AMS	SystemC-AMS	Δ
Class 1	0.0070686	0.0070696	0.01 %
Class 2	0.0344848	0.0345094	0.07 %
Time	59 min 5 s	8 min 23 s	

Differences between simulation results are negligible small.

5 Conclusion

In this paper the modeling of a vibration sensor array in SystemC-AMS was presented. Field of application of this language’s first version are communication systems, but as shown in this paper the language already can be used for MEMS simulation.

The SystemC-AMS simulation results differ only a little from the VHDL-AMS simulation. If further solvers [3] will be included then this gap will diminish. The calculation time is obviously reduced. The results for system-level simulation are identical.

SystemC-AMS offers support for all SystemC- and C++-language elements and concepts. So it is easy to describe digital components with only few description effort. Additionally, software parts can be included directly. The modeling of analog behaviour is still target of language development. At the moment the description effort for MEMS is higher than in VHDL-AMS.

6 References

[1] Open SystemC Initiative (OSCI): *SystemC 2.0.1 Language Reference Manual*. 2003, http://www.systemc.org

[2] Einwich, K. et. al.: *White Paper SystemC-AMS Study Group.* http://www.ti.cs.unifrankfurt.de/systemc-ams/

[3] Einwich, K.: *SystemC-AMS Steps towards an Implementation*. Proceedings FDL ‘03, Frankfurt/Main 2003, ISSN 1636-9874

[4] Schlegel, M.; Bennini, F.; Mehner, J.; Herrmann, G.; Müller, D.; Dötzel, W.: *Analysis and simulation MEMS in VHDL-AMS Based on Reduced Order FE-Models.* Center for Microtechnologies Annual Report 2003, Chemnitz 2004

[5] Michel, M.: *Untersuchung der Möglichkeiten zur Beschreibung und Simulation heterogener Systeme unter SystemC-AMS im Vergleich zu VHDL-AMS.* Diploma Thesis, TU Chemnitz, June 2004