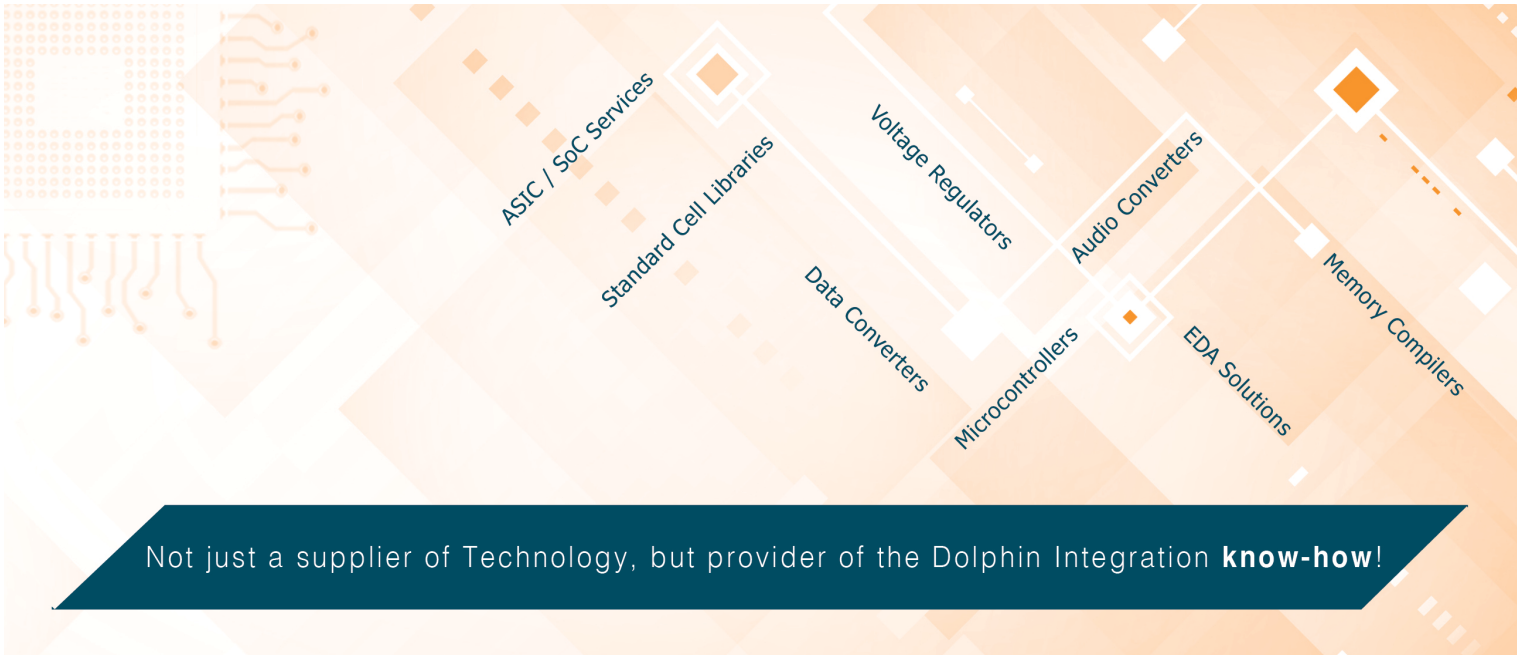
DOLPHIN INTEGRATION

IMPROVING LOADING TIME ON ADVANCED NODES WITH SMASH™ MIXED-SIGNAL SIMULATOR



Abstract

In the latest advanced nodes, getting a quick loading time for large analog circuits becomes a decisive stake. This paper explains how the latest SMASH™ revision allows to reach it. SMASH™ is a seamless IC-PCB mixed-signal simulator enabling the development and verification of analog and mixed-signal Silicon IPs and Integrated Circuits (IC) as well as the optimization of application schematics thanks to its unique multi-domain capabilities.

Introduction

As MOSFET size is constantly reducing, MOSFET model complexity is continuously increasing. This growth of complexity also involves a growth of parameter number in the MOSFET SPICE models. The number of model parameters was about ten with processes greater than hundreds of nanometers, and this number reaches thousands with processes less than twenty nanometers

This growth of parameters number has a significant impact in the loading time of SPICE netlist. In addition, improvements made on analog solvers and 64-bit platform allows loading of larger netlist. As a consequence, loading time becomes the main optimization concern in the latest MOSFET technologies for SMASH™.

The solution chosen for SMASH™ to address this issue will be presented in the following paragraphs.

Loading performance issue

As explained above, the number of parameters per macro-model of MOSFET has changed considerably in the last process nodes. The Figure 1 below shows this evolution of the number of parameters per MOSFET through three process nodes (180nm, 55nm and 28nm). The number of parameters to be computed for each MOSFET is about 60 in 180nm process, and up to 1500 in 28nm process.

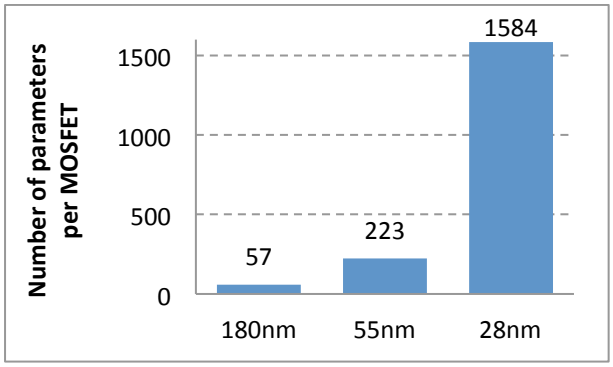


Figure 1 - Evolution of the number of parameter per MOSFET Vs. process node size

The Figure 2 illustrates the loading time issue. The test case used is a SPICE netlist containing about 5000 MOSFETs. Loading time is measured with SMASH™ 6.6.1 for three process nodes: 180nm, 55nm and 28nm. It shows that the loading time is about 5 times slower with the 28nm process node than with the 180nm one.

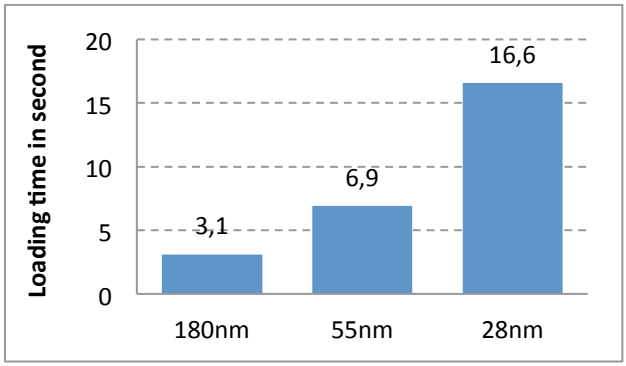


Figure 2 - Loading time Vs. process node size

Program profiling[1] reveals, for 28nm process node, that time is mainly spent in evaluation of SPICE parameters. Indeed, as shown in Figure 1, whereas at 180nm process node the number of parameters is to 57 per MOSFET, it reaches 1584 parameters for 28nm process node. As a consequence, instead of computing 285,000 parameters, 7,920,000 parameters are computed, which explains the performance issue in 28nm process node.

[1] Profiling consists in analyzing the performance of a software to get knowledge about its behavior during runtime.

Solution

Difference between process nodes results in the number of parameters that should be computed as shown in Figure 3. The MOSFET instantiation is encapsulated within a sub-circuit, which is then typically used as shown in Figure 4. Only few parameters (Between 10 to 50) are passed to the instantiation and it can be observed that several instantiations are using the same set of parameters with the same values. Based on this observation, the chosen solution is to create a unique identifier for a particular set of model, parameters and values.

```
.SUBCKT D G S B NMOS_28n
+ W=0 L=0
+ P001=2.1u … P0060=1.3n
+ P061=1.6u … P0220=5.2n
+ P221=1.1u … P1600=6.4n

M D G S B nfet W='P512' L='P1000'
.ENDS
```

[1] Profiling consists in analyzing the performance of a software to get knowledge about its behavior during runtime.

Figure 3 - SPICE model with MOSFET instantiation

```
X1 1 2 0 0 NMOS_28n W=4.9n L=20n
X2 3 2 0 0 NMOS_28n W=4.9n L=20n
X3 4 5 0 0 NMOS_28n W=3.9n L=20n
X4 5 7 0 0 NMOS_28n W=3.9n L=20n
```

Figure 4 - SPICE sub-circuit instantiation

As a consequence, for the example given in Figure 4, a unique identifier is computed for instance 'X1'. The parameter computation associated to this identifier is stored in a table. As instance 'X2' has the same set of model, parameter names and parameter values, the same identifier (computed for 'X1') is retrieved and sub-circuit parameter values are taken from the table instead of being computed again.

As the value of parameter 'W' is different from previous instance, instance 'X3' will give a new identifier. Hence the new parameter computation is stored in the table for this new identifier. Instance 'X4' will have the same identifier as the instance 'X3', and then sub-circuit parameters will be retrieved from the table.

As a result, in the above example sub-circuit parameters have been computed only two times instead of four times as it could have been previously. As it has been the case on this simple example, the number of SMASH™ computations will be also drastically reduced in the case of real circuits, allowing a significant loading time reduction as disclosed in the next paragraph.

Results

The histogram in Figure 5 presents the loading time obtained with SMASH™ 7.0 after applying optimization on the same netlist as the one used for the Figure 2. It can be noticed that the loading time is improved for all the process nodes. As expected, it can be also observed that improvement is better for process nodes with many parameters such as 28nm.

Before optimization, loading time of 28nm process node was 5 times slower than loading time of 180nm one. Now the ratio has been shrunk to 1.9 times. This enhancement allows the latest revision of SMASH™ to address serenely the latest MOSFET process nodes.

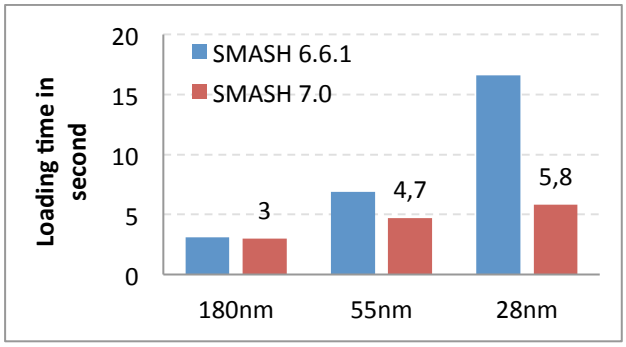


Figure 5 - Loading time Vs. mosfet node size after solution implementation

The chosen method of loading time optimization of course greatly depends on the number of MOSFETs generating the same identifier. Indeed, the more repetitive will be the MOSFET instantiations, the less parameter computations will be necessary and the better will be the loading time.

To illustrate this point, loading time of several test benches have been measured with SMASH™ 6.6.1 and SMASH™ 7.0. The ratios obtained between both SMASH™ revisions are listed in the Figure 6.

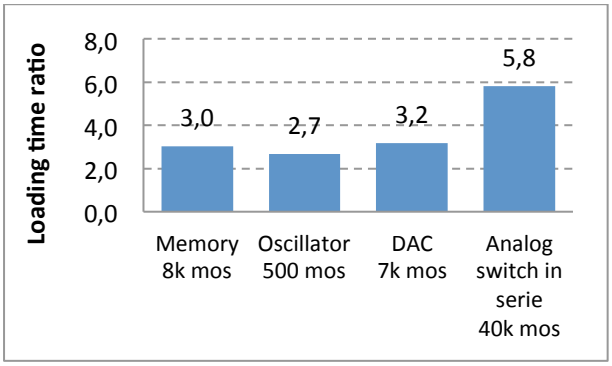


Figure 6 - Loading time ratio between SMASH™ 6.6.1 and SMASH™ 7.0

The ratios are ranging from 2.7 to 5.8 depending on the circuit. On real user cases (like the first three in Figure 6), loading time is divided by about 3 times. The last test case (Analog switch) is not a user bench but has been designed to show the maximum reachable gain. Indeed, in this netlist all MOSFETs generate the same identifier, explaining the ratio difference.

Conclusion

Loading time of complex SPICE analog circuit is a crucial issue for analog designers requiring latest MOSFET sub-micron technologies. The last revision of SMASH™ speeds up loading time by ration 3, on advanced nodes circuits as 28nm. This new optimization has shown that SMASH™ is able to answer efficiently to this concern and highlights a new SMASH™ as a serious and competitive analog and mixed-signal simulator.

About authors

Benoit Dumas received a master's degree in electrical engineering from Polytech'Grenoble, France, in 2007. He joined Dolphin Integration the same year. He was involved in the Software development of SMASH, the company's Mixed simulator for the last 10 years. He mainly contributed to the support of Verilog-AMS and VHDL-AMS languages and to improve performances of the analog engine of SMASH.

Cédric Valla received a master's degree in modelling and simulation of industrial systems from University Joseph Fourier Grenoble, France, in 2006. He joined Dolphin Integration in 2007. He is software developer and technical architect of analogue simulation development in mixed-signal simulator SMASH.

Florian Espalieu joined Dolphin in 2003 after a master's degree in electrical engineering from Phelma (ex-ENSERG), Polytechnic National Institute of Grenoble, France. He is in charge of Application Engineering for Dolphin EDA solutions. He has three years of experience as designer in the digital domain (development of SATA high speed data links) and four years in the mixed-signal domain where he is holder of a patent on an analog test device in the frame of noise propagation modeling in a SoC.