Improving SoC Design Flow with Unified Modeling Language and HDL

Chang Hoon Kim, Sang Joon Hwang, Seung Woo Hong, Man Young Sung*
Korea Univ. Dpt. of Electrical engineering
*E-Mail : semicad@korea.ac.kr

UML과 HDL을 이용한 SoC 설계 개선

김창훈, 황상준, 홍승우, 성만영
고려대학교 전기공학과

Abstract : HDL(Hardware Description Language) is the most important modern tools used to describe hardware, and becomes important as we move to higher levels of abstraction. The HDL has been made brisk use of in analog design, MEMS device[1-2], process related field as well as digital design. The most important characteristics of HDL is Abstraction which is the strongest tool that extend greatly designer's design ability. In this paper by the Modelling Continuum with hierarchical structure of abstraction, we apply UML(Unified Modeling Language) to SoC Design with HDL. UML makes an easy and visual description of the various levels of abstraction, and gives designers good flexible modeling capabilty for SoC Design.

Key Words : HDL, AMS, Behavioral Modeling, UML, Abstraction, Modeling Continuum

1. Introduction

The past computer was only a convenient equipment or device that handles simplified and repeated work. Nowadays a computer is a tool that guides an information-oriented society as well as bands together organically with some social components, as that spread phenomenons that have remained in cyber circumstance over actually society. By existing design process and CAD tools or DEA tools it is necessary to produce, maintain, and develop technology and products stably that is done large scale and complex. Various methods has been proposed to solve these problem, a note worthy method among them, observing is to use Embodiment Hardware language such as HDL[3].

2. Consideration of The Existing Behavioral Modeling and Simulations

HDL is devided into Digital HDL, Analog HDL, AMS(Analog Mixed Signal) HDL greatly. There are other methods, such as SPICE, C/C++ and etc.

2.1 Analog HDL and Digital HDL[4-5]

VHDL and Verilog have become very common digital modeling design languages, but digital designs potentially suffer from analog effects, often due to deep submicron design - in the real SoC applications. Issues regarding timing, parasitics, long wire and transmission line effects, and other physical phenomena within a physical design can be handled easily by another effective method, such as Analog HDL. We can see a description domain which is dimidiated in shown Figure 1. as above things.

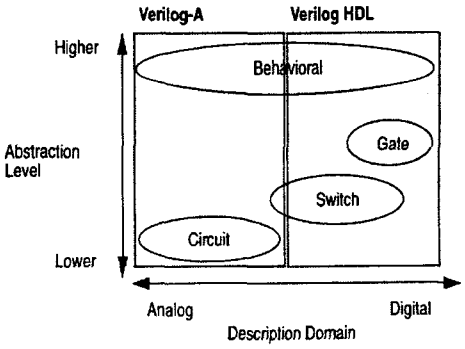


Figure 1. The description domain of Verilog HDL (for Digital) and Verilog-A (for Analog)

2.2 VHDL and Verilog AMS(Analog Mixed Signal)[6-7]

The appearance of VHDL-AMS and Verilog-AMS provides new capabilities to support the design of analog and analog mixed-signal systems. Figure 2 shows combined description domain which was divieded into two domains in Figure 1. Both VHDL-AMS and Verilog-AMS are all suitable for SoC Design. However, to address the interfacing issues for AHDL-AMS, Verilog-AMS supports the notion of connect modules. The use of connect modules for interfacing digital, analog, and mixed-sinal Verilog-AMS modules provides a general modeling capability more flexible than with VHDL-AMS, although at the possible cost of interface errors

not being identified[8]. So We suggest using Verilog-AMS for more delicate feature System Design

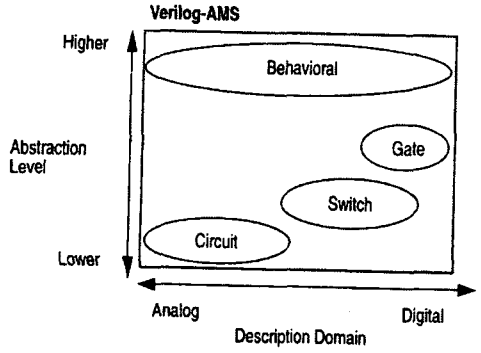


Figure 2. The scopes of the combined design descriptions domain targeted by Verilog-AMS(for mixed signal)

2.3 SPICE and C/C++

SPICE is a generally purposed analog circuit simulator and the most widely used analog simulator. Although SPICE does provide some capabilities for abstraction and refinement via model and subcircuit mechanisms, it has poor and limited functionality. Hence, behavioral modeling with SPICE is not suitable for SoC design by itself.

Recently, in particular, C/C++ have enjoyed significant interest as potential system level design languages their popularity with systems and software engineers. A number of system engineering tools can use the output of C/C++ or use them as intermediate data, the widespread use of C/C++ already exits. The definite language proposal, standardization of the packages and libraries, and user base are all lacking as well.

2.4 Abstraction and Modeling Continuum[5]

Abstractaction feature has various species levels and it is a very important property in design. It is a tool that will enable modeling languages, such as HDL or UML, to be extended with the modeling contiuum. On the other hand, Abstrcation is depicted and embodied in HDL. Hence Abstaction is inseparable from the description modeling languages like a HDL or UML.

The modeling continuum is a defined as hierarchy of abstractions. It connects hierarchically low level model with high level model as shown in Figure 3. It is a more important element that enables data share between many designers and data extension through it.

There is no exact method for using the various simulation tools for design. In other words, as shown in Figure 3, more abstraction needs less simulation time while less abstraction needs more simulation time. It is difficult to decide an optimal trade off point between accuracy and time.

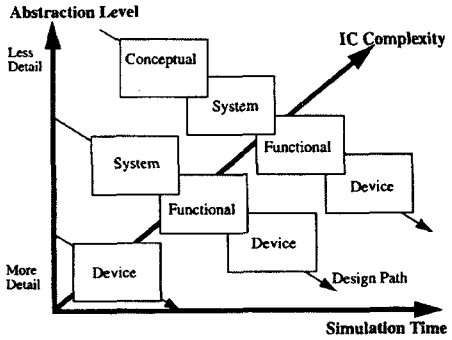


Figure 3. The Modeling Continuum

3. UML for SoC and Extentions to HDL

UML integrates OMT(Object Modelling Technology), OOSE(Object-Oriented Software Engineering) Methodology, and Booch Method, etc. in OMG(Object Management Group ; www.omg.org) that is one of object standardization organizations and announced in modelling language. Figure 4. shows a general constituent of UML.

The UML is the language which is made for specifying, visualizing, constructing, and Documenting of the Artifact such as software system or Business Modeling and other non-software system[9].

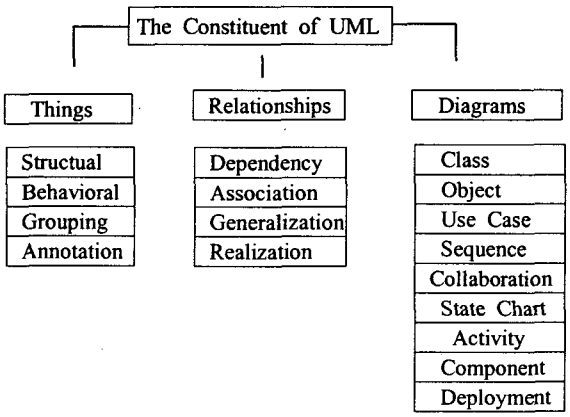


Figure 4. The constituent of UML

After all these characteristics of UML is just the object of using UML. As the complexity of system increases, an effective modeling technology to achieve good modelling becomes important. As a result, in order to construct object-oriented system and component basis system, it is necessary to select visual modelling language. UML that language do not include everything, but provide the modeling simple and standardization. These offer softness that UML can

be used in industrial circles design of various systems which are generally used. UML is open system to everybody without being monopolistic in the company. Therefore, it is expected that UML becomes the basis of a lot of tools for visual modeling and simulation, as development environment if see as present design circumstance[9]. As integration development environment is improved, embodiment of system on the basis of the UML will increase more gradually. Figure 5. shows the proposed concept diagram for SoC flexible Design with UML.

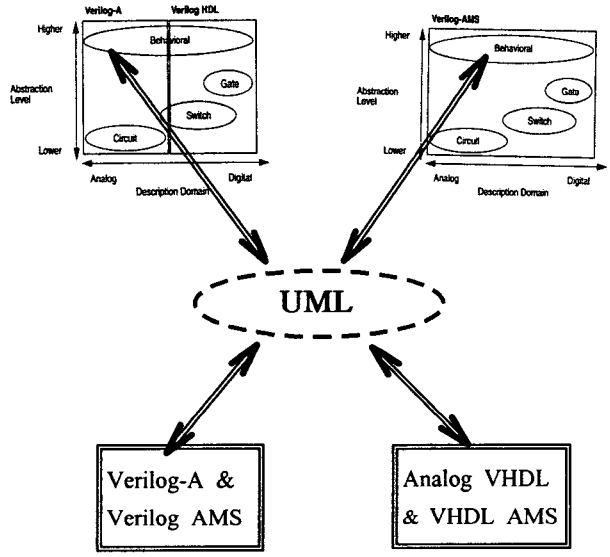


Figure 5. The propose concept diagram

In the design of the analog and mixed-signal, behavioral modeling at high abstraction level by using UML makes the system design rules more elastic. It can solve the problem oh the confusion linked up decision of description language when the beginning stage of system design as well as the easiness and visibility.

The basic structure of the primitive modeling shown in Figure 6(a) is normally composed of three sections and the simplest equivalent circuit structure model is shown in Figure 6(b).

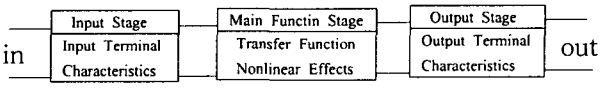


Figure 6(a). The basic sections of modeling diagram

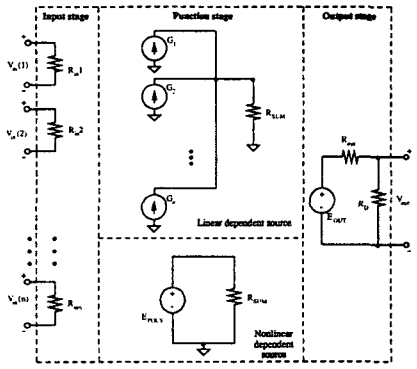


Figure 6(b). The equivalent circuit structure model (for Figure 6a.)

Figure 7(a) and (b) show the architecture and behavioral model of an ideal op amp. Respectively a variety of modeling levels(see Figure 2.) can describe the op amp but in this paper a simple model for top level architecture is used.

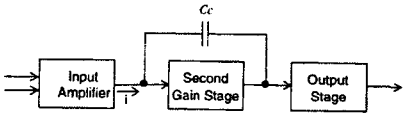


Figure 7(a). The basic architecture of an ideal op amp.

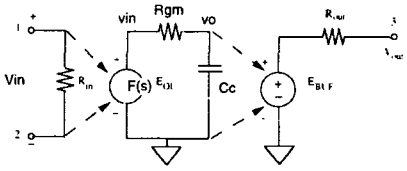


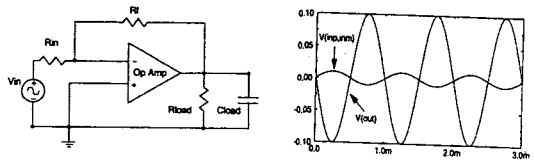
Figure 7(b). The behavioral mode for op amp(Figure 7a.)

We have applied UML model to basic architecture of an ideal op amp. The UML model takes place of Verilog or VHDL model as well as worked with HDL as subcircuit and model library.

Verilog Description	UML Digram
module ideal_opamp(inm, inp, out); **inout** inm, inp, out; electrical inm, inp, out; **parameter real** gain=250k **parameter real** rgm=1k **parameter real** cc=10p **parameter real** rin=1M electrical vin, vo; **analog begin** I(inp, inm) <+V(inp, inm)/rgm; V(vin) <+ **laplace_nd**(gain*V(inp, inm), {1.0}, {1.0, 5.0e-7}); I(vin, vo)<+ V(vin, vo)/rgm; I(vo) <+ **ddt**(cc*V(vo)); V(out) <+ V(vo); **end** **endmodule** * other level description * **module** ideal_opamp(pout, nout, pin, nin); **output** pout, nout; **input** pin, nin; electrical pin, nin, pout, nout; **branch**(pout, nout) out; **branch**(pin, nin) in; **analog begin** V(vo):V(vin)==0; **end** **endmodule**	InputStage parameterRin . . . vIn( ) GainStage parameterGian parameterRgm parameterCc . . . vin( ) vo( ) OutputStage1 parameterRout . . . vOut( ) rOut( ) able to be omitted

Table 1. Descripition of HDL and UML

Figure 8. shows transient analysis waveform and a simple test circuit with the ideal op amp mentioned above.



(a) (b)

Figure 8. The simple test circuit(a) and time domain response for behavioral model op amp(b)

4. Conclusion

When designing system and embodying it in a chip, designers make every endeavor to accomplish it notwithstanding many difficulties of the process, technique, and time(see Figure 3). UML which has the characteristics of easiness and visuality and others adds flexibility to these designer's capability of the modeling. Issues related to the embodiment of high abstraction level model by UML have been identified. While the accuracy at the high abstraction level and the embodiment of low abstraction level model by UML, two all are the method of Extentions to HDL for the more good SoC, we should solve it another problems before long. Though design with UML has some problems as ever, UML grow up to be a suitible tool for On Demand era.

References

[1] Dennis Gibson, Carla Purdy, "Analog Integrated Circuits and Signal Processing", vol. 20, p. 227-238, 1999.

[2] Gary K. Fedder, "Behavioral Modeling and Simulation", Proc. 2003 IEEE International Workshop, p. 64, 2003.

[3] A.Ferrari and A.Sangiovanni-vincentelli, "System Design : Traditional Concepts and New Paradigms", in Proceedings of International conference on computer Design, October 1999.

[4] Verilog AMS Language Reference Manual Version 2.1, Accellera, 2003.

[5] Paul A. Duran, "A Practical Guide to Analog Behavioral Modeling for IC System Design", Kluwer Academic Publishers, Boston, 1998.

[6] Ernst Christen, Kenneth Bakalar, Allen M. Dewey and Eduard Moser, 36th Design Automation Conference, New Orleans, 1999.

[7] Kundert Kenneth S, Zinke. Olaf, "The Designe's Guide to Verilog-AMS", Kluwer Academic Publisher, Boston, 2004.

[8] Peter Frey, Donald O'Riordan, "Verilog-AMS : Mixed-Signal Simulation and Cross Domain Connect Modules," IEEE/VIUF International Workshop on Behavioral Modeling and Simulation, October 2000.

[9] Luciano Lavagno, Grant Martin, and Bran Sellic, "UML for Real Desgin of Embedded Real-time Systems", Kluwer Academic Publishers, Boston, 2003.