Fourth LACCEI International Latin American and Caribbean Conference for Engineering and Technology (LACCET'2006)
"Breaking Frontiers and Barriers in Engineering: Education, Research and Practice"
21-23 June 2006, Mayagüez, Puerto Rico.

SystemC Implementation of VLSI Embedded Systems for MEMS Application

Junfang Yao, Ph.D Candidate
Research Associate, Department of Electrical and Computer Engineering, Florida International University, Miami, FL 33174; contact: jyao001@fiu.edu

Subbarao V. Wunnava, Ph.D., P.E
Professor; Director, VLSI and ISD Laboratories; Department of Electrical and Computer Engineering, Florida International University, Miami, FL 33174; contact: subbarao@fiu.edu

Abstract
With the increasing complexity of today's Integrated Circuits (IC) system and rapidly technology advancing, a single chip with smaller and smaller size can integrate more and more functionality. A high level of abstraction for modeling IC designs is required before hardware and software components in a single chip are implemented. SystemC is such a system level modeling language useful for Very Large Scale Integration (VLSI) design. With various features SystemC can perform system level modeling and simulation, which are missing in the generic Hardware Description Language (HDL)'s such as Very High Speed Integrated Circuits Hardware Description Language (VHDL) and Verilog (Sirpatil, 2002). In this paper the benefit of the SystemC technology is introduced, and also the design innovations of the SystemC Hardware Description Language was presented. Some interactive case studies of the Digital System Designs, such as Counters, Shift Registers, and Arithmetic Logic Units (ALU) were studied.

Keywords
SystemC; System Level Modeling; Counters, Microcontroller

1. Introduction

Today, more advanced process technologies and sophisticated design methods are used in VLSI system, along with rising gate counts and small chip size, software content is playing a much more important role in system hardware states. They are driving a need for more effective system-level methods.

SystemC is an open C++ class library used for hardware system design and verification. The SystemC class libraries add hardware attributes to the C++ language (Yarom et al., 2004). It is becoming widely accepted as a platform for modeling systems consisting of both hardware and software components. SystemC can be seen as a Hardware Description Language. However, unlike VHDL or Verilog, SystemC provides sophisticated mechanism that offer high abstraction levels on components interfaces. One of the major advantages of SystemC is that it can be used to describe a system at several levels of abstraction, starting at a very high level of functional description and down to synthesizable Register Transfer Logic (RTL) style (Yarom et al., 2004).

SystemC can model concurrent system in C++, and provides an event driven simulations environment. Due to these natures, SystemC can be used to descript and integrate complex Hardware and software design (Esperan, 2005).

Based on these advantages, along with the increasing trend towards the use of C/C++ languages as a unified modeling tool, SystemC will gradually replace other HDLs such as VHDL or Verilog to implement VLSI system in industry or academia.

2. Design Method

The flow used to design digital systems is described in figure 1. It consists of the following steps:

- The required system circuit is specified by users.
- A system level model and test bench is written in SystemC, and the model is tested with test bench. With the delivered test bench, the verification step is necessary (Vaumorin et al., 2004).
- Use VHDL or Verilog to write the equivalent system level model and test bench Simulation, Synthesis etc.

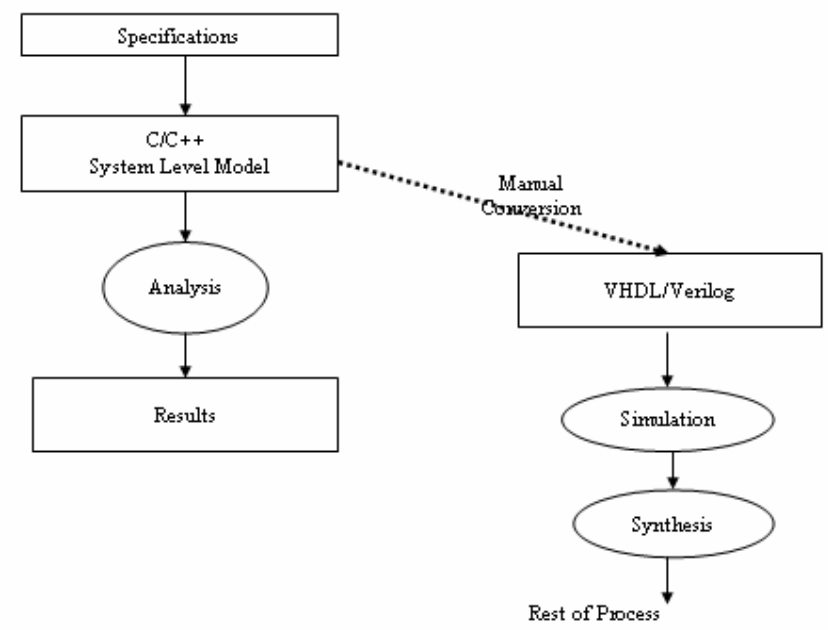

Figure 1: Digital system design flow

3. Case Study

3.1 4-bit Up-Down Counter

Counters and shift registers are basic circuit cells used in many digital circuit systems. In the following sections explanation for designing of a 4-bit Up-Down counter in SystemC is given. It is a 4-bit synchronous binary up and down counter with pause function and asynchronous reset. The counter loads a 4-bit data in and outputs the data on each positive edge of the clock. Input UPS decides the up or down count function and input PAUSE will temporally cease the counting work until it is released (Figure 2, Figure 3).

Figure 2: 4-bit Up-Down counter Architecture Description

Pin #	Name	Bits	I/O	Function Descript
1	clk	1	I	Clock signal
2	n_rst	1	I	Asynchronous reset, valid on negative
3	load	1	I	Load data in
4	pause	1	I	Pause the counting work
5	ups	1	I	Switch on up or down count, ups=1 for up count Ups=0 for down count
6	data_in	4	I	Data need to be loaded in
7	data_out	4	O	Data count for output

Figure 3: 4-bit Adder Pin Description

3.2 Design Procedure

3.2.1 Setting up SystemC

In order to build SystemC environment, the following tools are required: the SystemC workspace and class libraries, which can be downloaded from www.systemc.org, and Visual C++.

I. Open Microsoft Visual C++

II. Port SystemC libraries to Microsoft Visual C++:

a) "Project" -> "Settings", select the "C/C++ tab", then the "C++ Language" category. Make sure that the "Enable Run Time Type Information (RTTI)" checkbox is checked.

b) Also "C/C++ tab" and select the "Preprocessor" item, typing "C:\SystemC\systemc-2.0.1\src" in the text entry field labeled "Additional include directories".

c) Next click on the "Link" tab, typing "C:\SystemC\systemc-2.0.1\msvc60\systemc\Debug" in the text entry field labeled "Additional library path".

d) Add the SystemC object files: "Project" -> "Add to Project" -> "Files". In the File Browser navigate to the "C:\SystemC\systemc-2.0.1\msvc60\systemc\Debug" directory. In the text entry field labeled "File Name", add all "*.obj" files.

e) Delete the file "sc_isdb_trace.obj" in your workspace window under the "File View" Tab.

3.2.2 Generate 4-bit adder code in SystemC and verify code.

3.2.3 Convert SystemC code to VHDL/Verilog.

Mentor Graphic's software is selected to convert SystemC code and simulate it. ModelSim is

a simulation and verification tool for VHDL, Verilog, SystemC, System Verilog and mixed-language designs. SystemC design can be loaded in ModelSim with some modifications (Mentor 2006).

3.2.4 Simulation and synthesis.

ModelSim Simulation (Figure 4):

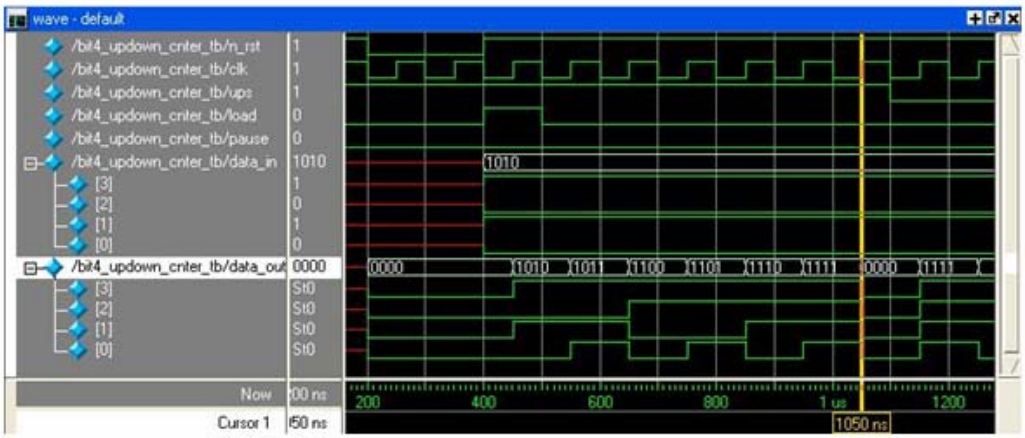

Figure 4: ModelSim Simulation Waveform

Synthesis with Leonardo on ASIC platform (Figure 5):

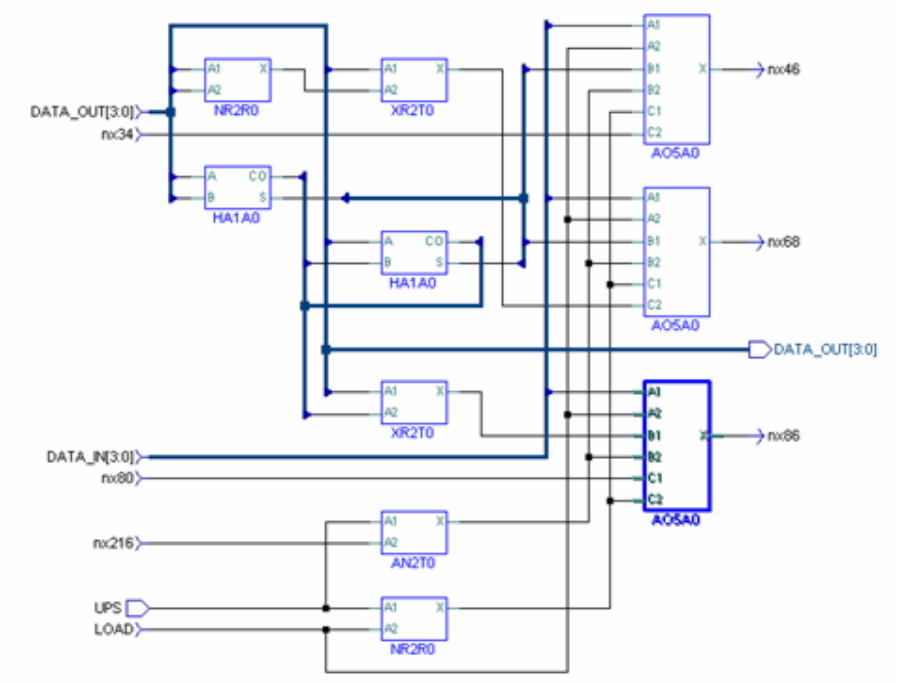

Figure 5: Synthesis with Leonardo

In SystemC after all files were compiled, the projects were built and executed. This will give

waveform value change dump (VCD) file. This VCD file include data that can be graphically displayed or analyzed with post processing tools. An example of such a tool is ModelSim vcd2wlf tool, which converts a VCD file to a WLF file that can be viewed in a ModelSim wave window.

- Input vcd2wlf *.vcd *.wlf at the ModelSim command prompt.
- In ModelSim, open the wave file *.wlf by entering the following command: vsim –view *.wlf.

The same waveform as shown in Fig. 4 was obtained.

4. Microcontroller

Every embedded system these days contain a microcontroller. These microcontrollers are an integrated chip which includes a central processing unit (CPU), Random Access Memory (RAM), Read Only Memory (ROM), Input-Output (I/O) ports, and timing devices. These embedded systems are small in size because of their functionality. They perform only one specific task and usually control only one system. Therefore, all the functionalities are included in a single chip; this concept is known as System on Chip (SoC).

Microcontrollers integrate all of the needed components onto a single chip, and generally these components are built from logic components. Because of their simplex functionality, microcontrollers do not require significant processing power (Blacharski).

Microcontrollers can be used in many fields: home appliances, communication and electronic control systems (Ferreira et al., 2005). Moreover, it is used in automobiles industry as well, and play a key role in robotics.

The 4-bit Counter design can be applied to design a microcontroller. Over the past years SystemC has been positioned as a suitable system level design language and framework to build models of working systems, such as embedded systems, microcontrollers. However, how to model various components of a microcontroller in the most appropriate model for the component’s functionality is a big issue, it can avoid unnecessary data cycles and improve simulation efficiency.

Conclusion

SystemC allows the development of an effect solution to design digital systems, because the entire system executes within a single simulation environment. It is demonstrated on how to design a 4-bit counter in SystemC, and this method can be applied to design microcontrollers, even more complex system (such as microprocessors).

Besides this, future work is to find an appropriate model to make each component of a microcontroller to reach its maximum function.

Reference

Brijesh Sirpatil (2002), "Software Synthesis of SystemC Models", Master thesis, Virginia Polytechnic Institute and State University, Virginia, USA.

Dan Blacharski, URL: www.wisegeek.com

Esperan (2005), "Introduction to SystemC Tutorial".

Emmanuel Vaumorin and Thierry Romanteau (2004), "From Behavioral to RTL model in Systemc", LLR-PROSILOG scientific collaboration.

Itai Yarom and Gabi Glasser (2004), "SystemC Opportunities in Chip Design Flow", *The 11th IEEE International Conference* on Electronic Circuits and Systems.

Luis F. Ferreira, Emilio L. Matos, Luis M. Menendez and Enrique Mandado (2005), "MILES: A Microcontroller Learning System combining Hardware and Software tools", *Frontiers in Education, Proceedings 35th Annual Conference.*

Mentor Graphics (2006), "ModelSim Tutorial", version 6.1e, http://www.mentor.com/

The Open SystemC Initiative (OSCI), URL: www.systemc.org

Authorization and Disclaimer

Authors authorize LACCEI to publish the papers in the conference proceedings. Neither LACCEI nor the editors are responsible either for the content or for the implications of what is expressed in the paper.