Mixed level and mixed signal simulation with PSpice A/D and VHDL

***Abstract*— PSpice A/D currently lacks the ability to simulate analog components connected to digital circuits that are modeled using Hardware Descriptive Languages (HDLs). This paper presents a design methodology and tools that enable simulation of synthesizable Very High Speed Integrated Circuit HDL (VHDL) models in PSpice A/D. Simulating of mixed signal Printed Circuit Boards (PCBs) with Programmable Logic Devices (PLDs) connected to discrete analog components is a typical design example, where the proposed design methodology and tools can be applied. A mixed signal design from NASA Goddard Space Flight Center for a brushless three phase motor that runs a space application is implemented by following the proposed design methodology.**

I. Introduction

Increasing design complexity and necessity to reduce design and verification times, drives the circuit design methodology and the choice of Electronic Design Automation (EDA) tools. Design methodologies are typically classified into:

- Top down design methodology, that enables designers to refine an abstract idea progressively as the design process continues. The design process begins with a very high level behaviorial definition of the system and then it can get down to finer details with Register Transfer Level (RTL) and gate level descriptions, as the design progresses. This methodology is popular with digital circuit designs with the advent of HDLs, PLDs and logic synthesis tools [1].
- Traditional or bottom up design methodology, that allows designers to pick components individually (from a set of libraries) and build the design by connecting them appropriately. This methodology is popular in the PCB design flow [2].

Large and complex systems are usually broken into smaller units that can be designed with different EDA tools and by following different methodologies. This curtails the ability to verify the functionality of the whole system, which is a potential cause for design failures [4] [5]. Moreover a majority of today's designs are mixed signal (analog and digital) circuits. A typical example of such system would be a PCB which has PLDs along with other discrete analog components. In awake of such scenarios, rises a need to have EDA tools that are capable of simulating mixed signal designs as well as designs designed using different methodologies. Such simulations are called mixed level and mixed signal simulations. PSpice A/D supports mixed signal simulation using traditional design methodology, however lacks the ability to simulate digital designs modeled using HDLs such as VHDL, Verilog etc. [3].

By enabling simulation of VHDL models in PSpice A/D it is possible to realize a mixed level simulator from a mixed signal simulator. This integrates traditional designing methodology with top down design methodology.

II. Proposed design methodology

The proposed design methodology is presented in figure 1. Mixed signal design is divided into two sections, namely analog and digital. While the analog circuit is designed by following the traditional design methodology in PSpice A/D schematic editor, the digital portion in VHDL follows the top down design methodology. Finally, the interfacing software (see section II-B) abridges the two design methodologies by enabling functional verification of the mixed signal mixed level design in PSpice A/D.

To achieve this goal, the proposed methodology employs a logic synthesis tool (e.g. Synplify), which translates RTL VHDL into VHDL gate level description [7]. This represents the digital system in terms of logic gates that are specific to a target technology that was chosen during the logic synthesis process - FPGA or CPLD from vendors like Xilinx, Altera, Actel etc. In order to simulate the gate level VHDL description in PSpice A/D, it needs to be in a format that is understood by the PSpice simulation engine. In other words, the gate level VHDL netlist requires to be translated into a PSpice subcircuit definition. Besides this requirement, simulation of technology-specific gate level VHDL description in PSpice also requires the need for appropriate digital device models within PSpice model libraries. The choice of the technology during logic synthesis determines the ease of translation and the ability to avail or create digital device models in PSpice. Typically, architectures of CPLDs are simpler when compared with FPGAs. Considering this factor the digital logic described in VHDL is synthesized by targeting at Lattice MACH 111 family of CPLDs.

The detailed design flow of the digital portion in this methodology is as follow:

- The digital circuit is described in VHDL and simulated to verify its functionality (top down design methodology)
- The RTL VHDL code is synthesized in Synplify using Lattice MACH 111 as the target technology. The gate level VHDL description (after synthesis) is once again verified by simulation [8].
- The gate level netlist is now converted into a PSpice circuit file using the interfacing software which was developed as a part of this work.

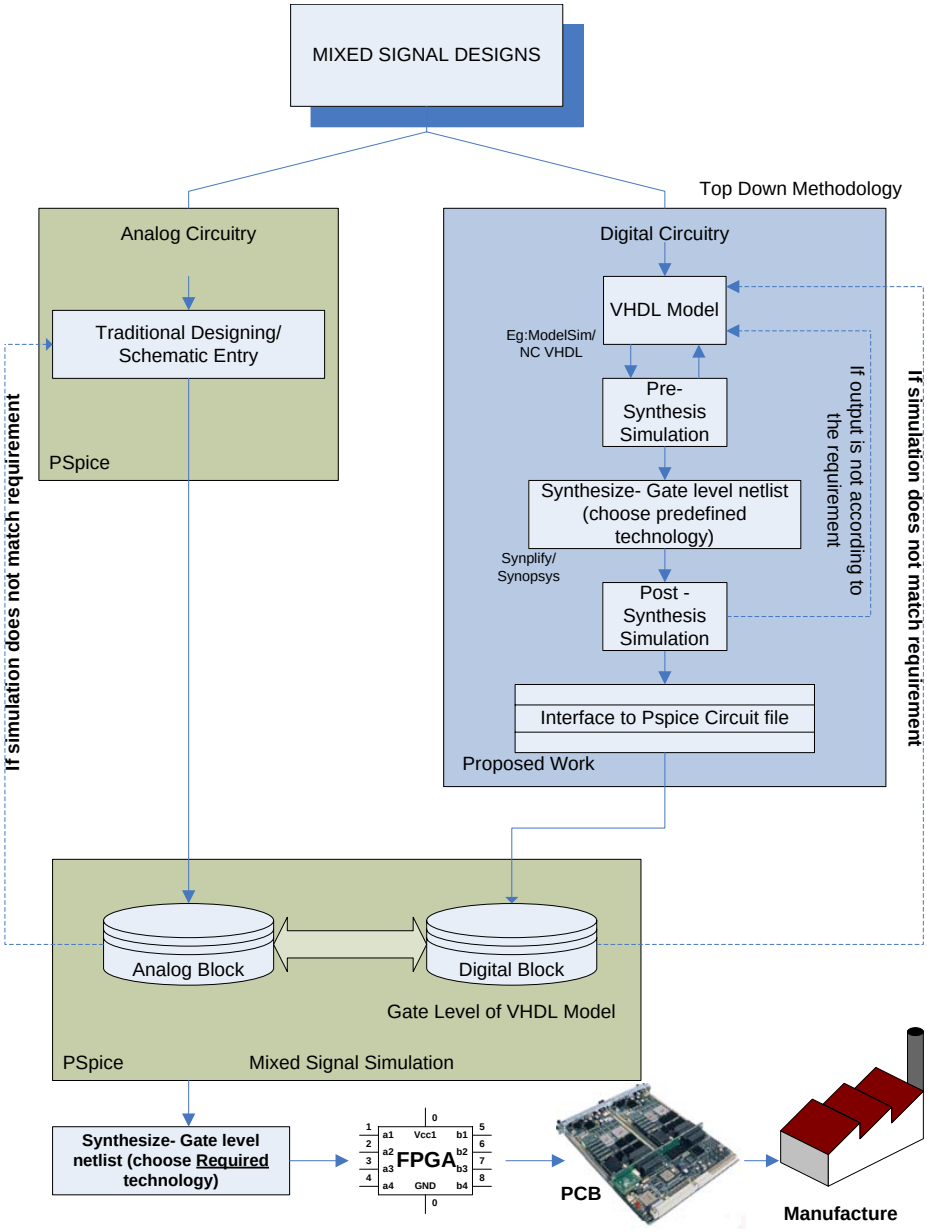

Fig. 1. Proposed design methodology

- The circuit file is converted into a schematic symbol and placed in OrCAD Capture schematic editor along with other analog components. The complete mixed signal design is verified by simulating in PSpice A/D.

The translation of the gate level VHDL netlist into its equivalent PSpice circuit file requires:

- A library of PSpice models for Lattice MACH 111 components.
- An Interfacing software that utilizes components from this library and create a PSpice subcircuit file from the gate level VHDL netlist.

A. PSpice library of Lattice devices

The gate level VHDL netlist generated by synthesis tool contains components specific to Lattice MACH 111 technology. The following set of combinational and sequential logic elements are utilized by the interfacing software during the translation process:

- Combinational logic elements:
  1) IBUF - Input Buffer
  2) OBUF - Output Buffer
  3) INV - Logic inverter
  4) OR2 - 2 Input logic OR
  5) XOR2 - 2 Input logic XOR
  6) AND2 - 2 Input logic AND

- Sequential logic elements:
  1) MACHDFF - Reset predominant D flip flop with low preset and reset
  2) DFFRH - Reset predominant D Flip flop with preset remaining HIGH all times
  3) DFFSH - Reset predominant D Flip flop with reset remaining HIGH all times
  4) DFF - Reset predominant D Flip flop

For all these elements PSpice models have been created and put in a library.

B. Interfacing software

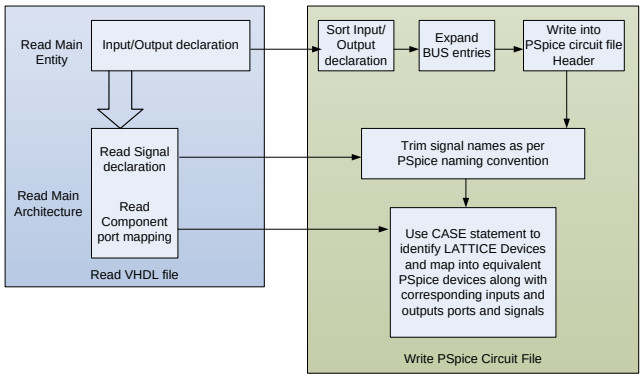

Fig. 2. Flow chart for VHDL-PSpice Conversion program

Next, the gate level VHDL netlist needs to be translated into a PSpice subcircuit file. A software program was developed to perform this task.

Figure 2 presents steps necessary to translate gate level VHDL description into a PSpice circuit file. The gate level VHDL netlist follows a typical pattern of structured VHDL logic description (component declaration and definition followed by the main entity and architecture). For every component defined in the gate level VHDL netlist, there exists an equivalent PSpice model.

The interfacing program reads through the gate level VHDL netlist, identifies a Lattice MACH device and replaces it with its equivalent PSpice model in the subcircuit file which it writes simultaneously. The following procedure is followed by the interfacing software

- Parse through the gate level VHDL netlist and skip until the main entity within the file is reached.
- Within the main entity, extract the name of the inputs and outputs. If the inputs/outputs are declared as a bus, elaborate the bus entries and assign individual net names for each one of the bus entries. PSpice digital device modeling language does not permit bus declaration.
- Using the input and output names obtained, define the subcircuit header in the PSpice circuit file by following the PSpice modeling language syntax.
- Continue to parse the VHDL file and skip until the architecture of the entity is reached.
- Within the architecture, skip the section where internal signal and component names are declared.
- Scan the architectural definition and identify the Lattice MACH device that is being ”port - mapped”
- Map the identified component with its equivalent PSpice model.
- Scan the ”port - mapping” definition to identify the input and output net names and assign them to appropriate PSpice model terminals.
- Loop until the end of architecture section is reached.

III. Case study: A Mixed Signal design from NASA Goddard Space Flight Center

Motion control and positioning of many space mechanisms are accomplished by using brushless motors. They are simple yet robust, offer high power to weight ratio, low inertia and optimal performances at high and low speed. Typical applications are scan mirror, thrust vector control actuators, fuel value control actuators, solar array deployment, control moment gyroscopes, applications requiring light weight, low thermal emission, high and low rotation per minute. [6]. An overview of a mixed signal design involving an Actel Flash FPGA for controlling the motion of a brushless motor that runs a gas compressor for a space application at NASA Goddard Space Flight Center is shown in figure 3. This design provides a fully synchronous system with single clock domain, high flexibility due to the possibility of modifying or updating, low chance of ground bounce or system noise due to reduction in the simultaneously switching outputs, and high reliability.

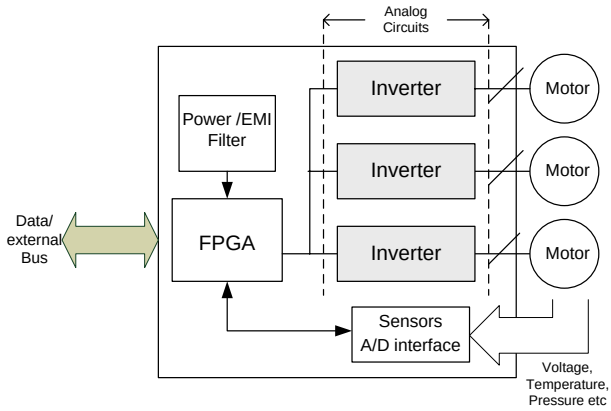

Fig. 3. Block diagram of the design from NASA

The motor is controlled and driven by digital circuits implemented on the FPGA that also communicates with the main spacecraft computer. The analog section is an inverter with half bridge drivers and transformers, that provide the required current conditioning and 120 degrees out of phase waveforms with required power levels to run the motor. The objective is to advance a Johnson counter according to a set of design rules and specifications. The output of the counter triggers a power MOSFET (within the analog circuit) at specific time intervals. Therefore, the arrival times of the pulses from Johnson counter are very important as they control the functioning of the motor.

The digital logic implemented on the FPGA is modeled in VHDL and simulated using a VHDL simulator. The analog circuits in the inverter section are simulated in PSpice A/D. The verification of the complete design (consisting of the digital circuits modeled in VHDL and the analog circuits in PSpice) is necessary to predict the behavior of the overall system.

The following sections will demonstrate the verification of a complete design by following proposed design methodology.

A. Simulation of VHDL model in PSpice A/D

The behavioral VHDL description of the digital logic is verified by simulation. It is then synthesized using Synplify by targeting at Lattice MACH 111 family of CPLDs. The gate level VHDL description generated by Synplify is once again verified by simulation.
Using the interfacing tool that was developed, the gate level VHDL description is converted into a PSpice subcircuit file. PSpice model editor is then used to create a circuit symbol which then can be placed on the schematic along with other analog or digital components. The PSpice model representing the gate level VHDL netlist is simulated (in PSpice) to verify its functionality. Figure 4 shows the results of simulation in PSpice and the waveforms at the “mosfet” pin and it matches the post synthesis simulation results of the VHDL description.

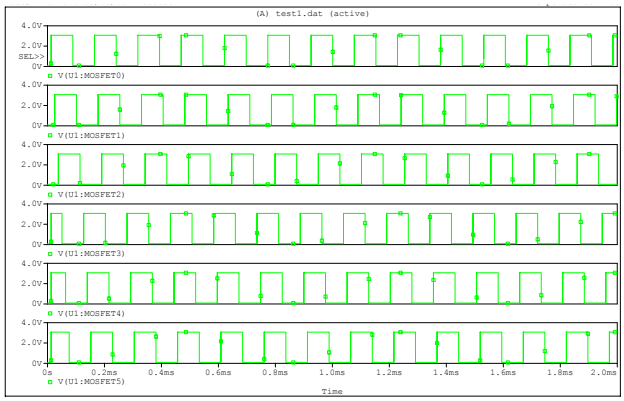

Fig. 4. Results after PSpice simulation of the VHDL (gatelevel) model

B. Integration of digital signals from the VHDL model with the analog circuit

The PSpice model representing the VHDL code is now ready for mixed signal simulation along with the discrete analog components. The outputs from the schematic symbol representing the VHDL model (in PSpice) is given as the input to the analog circuits. Section of this circuit is shown in figure 5. Figure 6 shows the simulation results of the complete mixed signal design (digital section modeled in VHDL along with analog circuits in PSpice). The result represents the output of the inverter which converts DC power to AC after receiving appropriate control signals from the VHDL model.

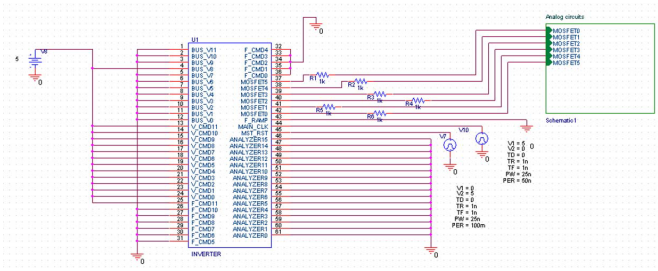

Fig. 5. Circuit representation of VHDL code during Co simulation

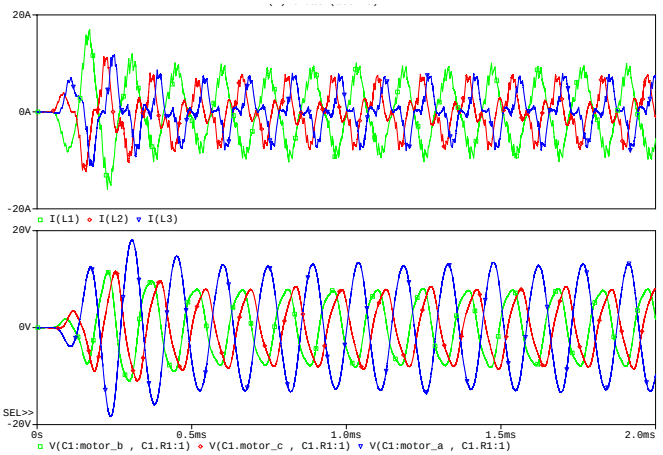

Fig. 6. Three phase current and voltage waveforms after mixed signal simulation with VHDL model in PSpice A/D

C. Summary

Prior to the proposed design methodology, it was necessary to duplicate the output signals from the FPGA with clock generator functions in PSpice, subjecting the entire design to various input conditions. The new design methodology allowed to precisely model the arrival times of input signals from the VHDL model in PSpice, which in turn produced accurate conversion of DC power to AC. Simulating the entire design at the PCB level allowed to study the behavior of the analog circuits when controlled by the FPGA.

IV. Conclusions

In this work the possibility of mixed signal mixed level simulation with PSpice A/D and VHDL was investigated. A new design methodology to simulate synthesizable VHDL models in PSpice A/D was proposed. It combines traditional and top down design methodologies in one mixed signal simulation environment. Interfacing software and PSpice device libraries were developed. Their functionality were utilized and verified in a practical application.

V. Future Work

The following areas were identified for further research

1) Increased simulation time. Using gate level VHDL netlist becomes a bottleneck if the number of gates in the netlist exceeds a few thousand. In order to reduce the simulation time, further investigation on behavioral

simulation methods of the VHDL code in PSpice is necessary.

2) Automatization and graphical user interface. The proposed methodology involves repeating a sequence of commands in each of the EDA tools, which can be scripted and automated. In doing so, a lot of details can be hidden from an end user and this would increase the appeal of the solution.
3) Development of libraries for different target technologies for more accurate timing simulations.

REFERENCES

[1] Peter Frey and Radharamanan Radhakrishnan. Parallel mixed-technology simulation. *Fourteenth Workshop on Parallel and Distributed Simulation*, pages 7 – 14, 2000.

[2] K. Kundert, H. Chang, D. Jefferies, G. Lamant, E. Malavasi, and F. Sendig. Design of mixed - signal systems - on- chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19:1561 – 1571, 2000.

[3] Bashir Al Hashimi. *The art of Simulation using Pspice Analog and Digital*. CRC Press, United States of America, 1995.

[4] N.L. Eastman. Considerations for Mixed analog/digital PCB design. *WESCON/96*, pages 297 – 301, 1996.

[5] Dave Brady and Tom Dewey. Are the benefits of using FPGA's consumed by the obstacles of integrating the FPGAs on Printed Circuit Board? *Mentor Graphics White Paper*, 2003.

[6] G.C. Caprini, F. Innocenti, L. Fanucci, S. Ricci, G. Taraschi, P. Terreni, M. Tonarelli, and L. Tosi. Embedded system for brushless motor control in space application. *MAPLD International Conference*, pages 151 – 156, 2004.

[7] B Fawcett. Synthesis for FPGAs: an overview. *WESCON/94. 'Idea/Microelectronics'. Conference Record*, pages 576 – 580, 1994.

[8] Synplicity. *Synplify Pro Reference Guide*, 2004. Version 7.5.1.