

# Mixed-Mode Architecture Development under an Existing VHDL Environment

Michael F. Sullivan, Gary A. Ransom, Karl H. Renner

Microelectronics Technology Center  
Texas Instruments, Inc.  
Defense Systems and Electronics Group  
PO Box 660246, MS 3145  
Dallas, Texas 75266

## Abstract

This paper presents our experience for merging analog and digital architectures under an existing Very High Speed Integrated Circuit Hardware Description Language (VHDL) design environment. Three major areas are presented: a brief description of a mixed-mode architecture, the incorporation of analog into the traditional digital VHDL design flow, and the mixed-mode ASIC simulation. Finally, an overview is presented of the need for a standard hardware description language with modeling capabilities for the full analog domain.

## 1.0 Introduction

VHDL was developed to generally describe digital hardware components and systems. However, the applicability of VHDL can easily be extended into inclusion of the analog domain for modeling and simulation. Analog designs are solutions of simultaneous equations for networks and can be modeled with VHDL's sequential capabilities. In the Texas Instruments (TI) Microelectronics Technology Center, this extension for using VHDL has been applied successfully in the designs of mixed-mode architectures.

This paper will focus on an analog testability integrated circuit (ATIC) as shown in Figure 1. ATIC is designed to generate analog waveforms for test applications. The IEEE 1149.1 JTAG standard interface controls it which enables ease of use and minimizes the I/O pin count. ATIC contains a DAC, a EEPROM, an EPROM, and associated control logic. The JTAG

interface controls the boundary and internal scan paths, and the EEPROM controller. The EEPROM controller is a state machine which controls the EEPROM and a sequencer. The EEPROM is programmed with microcode which is read and executed by the sequencer. The sequencer generates addresses which are used to read data samples corresponding to a waveform from the EPROM. Each data sample is converted to an analog voltage by a DAC such that the desired waveform is approximated by analog voltage steps. This voltage is then routed to one of four outputs. An analog MUX selects either an input or the test waveform for each output.

## 2.0 Traditional VHDL Environment

The traditional digital VHDL design environment used at Microelectronics Technology Center is depicted in Figure 2. From the design specification of a digital ASIC, a behavioral VHDL model is developed describing the functionality. Under a VHDL test bench, simulations are performed to verify the functional behavior of the digital design and to capture test vector patterns for subsequent replay capability. Digital design synthesis is performed mapping the digital functionality to a digital gate-level VHDL representation. Gate-level VHDL simulations are performed to verify the test vector patterns generated against the behavioral VHDL model. The physical layout of the top-level digital design is then developed through place and route of the gate-level description. Back-annotation simulation is performed on the gate-level VHDL description verifying the digital design routed layout.



output of the array which supplies an analog voltage to the unity-gain voltage buffer. The output buffer filters any residual glitches and drives the capacitive load seen through the analog output MUXes. The simple behavioral VHDL representation which models the DAC's functionality is shown in Figure 3.

```

DAC_function:
process(DATA_REG)
variable RAIL_VOLTAGE:real:=2.5;
begin
DAC_OUT <= RAIL_VOLTAGE * ((2.0 * TO_REAL(DATA_REG) / 255.0) - 1.0);
end process DAC_function;

```

Figure 3. DAC's Behavioral VHDL Model.

Since the DAC is analog in nature, the analog signals are treated as real numbers. The 8-bit digital data sample in *DATA\_REG* is converted into a real number which ranges between -2.5 and +2.5 volts. The *RAIL\_VOLTAGE* is the power supply limit. The *TO\_REAL* function converts the digital data sample into a real number ranging from 0.0 to 255.0. The *DAC\_function* process returns this result as the DAC output voltage.

Special considerations were given in order to merge the analog functions into the traditional VHDL design environment. Figure 4 presents the VHDL design environment allowing for mixed-mode design development. From the analog design specification, a device description is schematically captured and Spice simulations are performed to verify the functionality. Custom physical layouts are then developed and again Spice simulations are performed. Once the analog blocks are completed and verified, they can be viewed as design macros.

In parallel to the analog macro development, a behavioral VHDL model is created describing the analog-digital mixed-mode functionality. VHDL simulations are performed to verify functional behavior of the mixed-mode design. Design synthesis is performed mapping the digital functionality to a digital gate-level VHDL representation. The analog behavioral VHDL description is preserved through this process. During design synthesis, analog macros are black-boxed, viewed as design macros. The mixed-mode VHDL design, behavioral analog with gate-level digital, is then simulated to verify the test vector

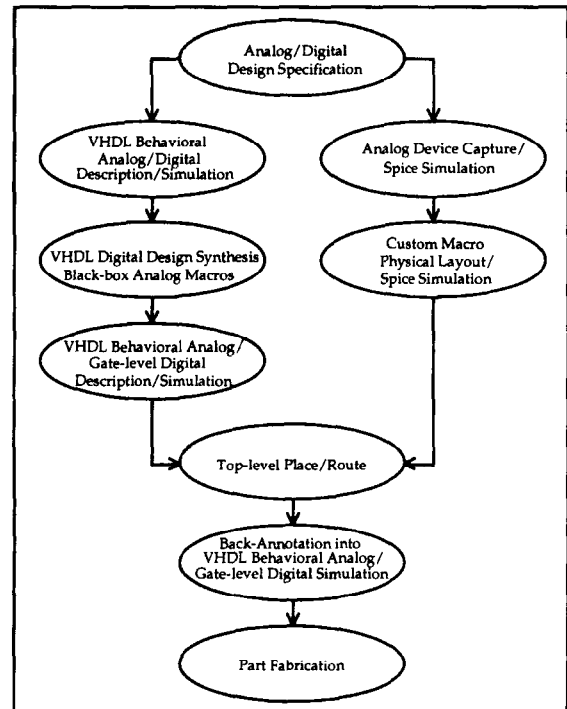


Figure 4. Mixed-Mode VHDL Design Environment.

patterns generated against the full behavioral VHDL model. The top-level mixed-mode design is then placed and routed generating the physical layout. The analog macro blocks are used in conjunction with the digital library cells during top-level physical layout generation. Back-annotation simulation is performed on the mixed-mode VHDL description, behavioral analog and gate-level digital, providing verification of the mixed-mode design routed layout.

### 3.1 Special Considerations

Detail was required to ensure the preservation of the analog functionality under a VHDL design environment. Without a strong mature mixed-mode simulator, Spice simulation on analog circuitry combined with VHDL simulation on digital circuitry, care had to be taken on critical timing paths. Timing analysis pre-layout and post-layout was essential in verifying the mixed-mode design.

### 4.0 Need for VHDL-A

The ATIC design contains analog blocks that can be described in VHDL by simple

equations. However, not all analog architectures can be described using VHDL. Within the analog domain, systems can also be described using nonlinear equations. These nonlinear equations can define systems such as mechanical and rotational systems. The analog domain can also be described in the frequency and continuous time domain. VHDL can only model time as discrete events. The need is highly evident for a standard hardware description language with modeling capabilities for the full analog domain.

VHDL-A is the VHDL extension for analog currently being developed by the IEEE 1076.1 Working Group. VHDL-A will be a superset of VHDL'93. For designs unlike ATIC that requires serious modeling of analog architectures and analog-digital mixed-mode architectures, especially at higher levels of abstraction, VHDL-A will provide a fully integrated hardware description language environment. VHDL-A will provide the capability to describe and simulate analog and analog-digital mixed-mode architectures and allow for the exchange of design information between companies and vendor tools.

## 5.0 Conclusions

In TI-Microelectronics Technology Center, integration of analog architectures into the traditional digital VHDL design environment has been employed successfully. With this defined VHDL design environment, most mixed-mode ASICs can be implemented in the traditional fashion. With the continued development of VHDL-A and its integration into today's design environment, all mixed-mode ASICs will be implemented in the same fashion.

## 6.0 References

1. *IEEE Standard VHDL Language Reference Manual*, IEEE Std 1076-1987, The Institute of Electrical Engineers, Inc., New York, NY, March, 1988.
2. M. Vincze, M. Sullivan, and D. Kopca, *Merging WAVES into an Existing VHDL Environment*, Texas Instruments Technical Journal, May-June, 1993, pp. 57-63.
3. M. Swenholt, *Specification for Integrated Circuit, Analog Test Waveform Generator*, Texas Instruments IDEA Program, Rev. A, October, 1993.