

Integrating WAVES into an Existing ASIC Design Environment

Michael F. Sullivan
Michael A. Vincze
Integrated Circuits and Computers Department
Defense Systems and Electronics Group
Texas Instruments, Incorporated
8505 Forest Lane, M/S 3145
Dallas, Texas 75243
sullivan@ap040.csc.ti.com
mav@asd470.dseg.ti.com
(214) 480-1345 (Sullivan)
(214) 575-2723 (Vincze)

Abstract

This presentation relates our experience of integrating WAVES into an existing ASIC design environment. Presented is an introduction to WAVES, our ASIC methodology as related to simulation, the methodology of WAVES integration including its automation, and obstacles encountered.

The need for WAVES is two-fold. One need is from the designer's perspective, where the designer can use a common medium between the VHDL simulator and automatic tester environments. The other need is from the government contractor's perspective, where contractual obligations must be met with WAVES.

Test benches rely on stimulus/response data obtained from VHDL and non-VHDL environments. Our goals were to integrate WAVES with minimal interruption to our existing VHDL environment and to automate as much of the process as possible.

Some problems were encountered with the integration. For bi-directional pins, VHDL can not readily distinguish whether a value is a stimulus or response value. Adherence to the WAVES LRM was difficult as we did not have a WAVES analyzer. Writing the WAVES dataset in a form to simplify automation was another problem. The solutions to these problems and other obstacles are discussed.



Integrating WAVES into an ASIC Simulation Environment

Michael F. Sullivan
Texas Instruments, Inc.
Defense Systems and Electronics Group
Integrated Circuits and Computers Department
POB 660246
MS 3145
Dallas, Texas 75266
(214) 480-1345
sullivan@ap040.csc.ti.com

Michael A. Vincze
Texas Instruments, Inc.
Defense Systems and Electronics Group
Engineering Services Department
POB 869305
MS 8408
Plano, Texas 75986
(214) 575-2723
mav@asd470.dseg.ti.com

Michael Sullivan and Michael Vincze

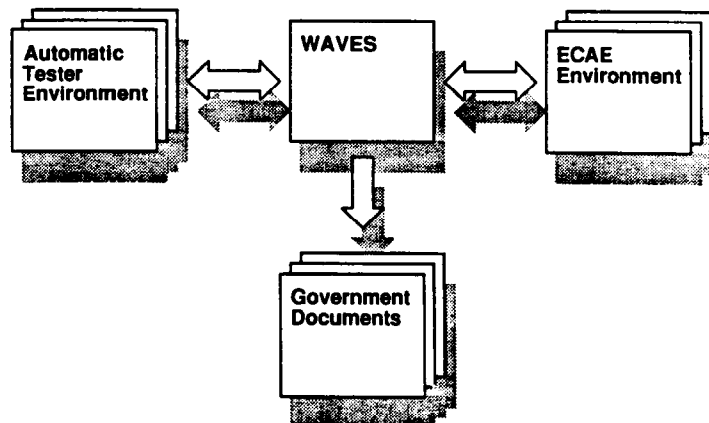
Defense Systems & Electronics Group



Integrating WAVES into an ASIC Simulation Environment

• **WAVES Goals**

- Provide a wave form and vector exchange mechanism between and among ECAE and Automatic tester environments.
- Support requirement 64 of MIL-STD 454.



Michael Sullivan and Michael Vincze

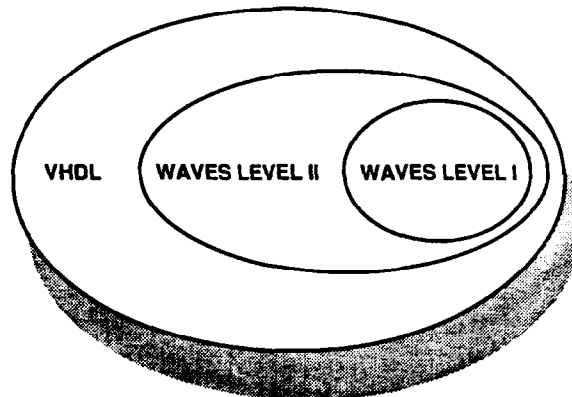
Defense Systems & Electronics Group



Integrating WAVES into an ASIC Simulation Environment

• WAVES Introduction

- WAVES is a subset of VHDL.
- WAVES has two levels of abstraction: Level I and Level II.
 - Level I has fewer language constructs than Level II.
 - Level I has a pre-defined external file format.



Michael Sullivan and Michael Vincze

Defense Systems & Electronics Group



Integrating WAVES into an ASIC Simulation Environment

• WAVES Introduction (Continued)

- WAVES allows wave form descriptions in tabular format:

```
C 0 000 000 : 100 ns ;  
C 1 000 001 : 100 ns ;  
C 1 000 010 : 100 ns ;
```

- or structured format:

```
for i in 0 to 15 loop  
  to_string (i, address);  
  write_we (connect, tsets, address, data);  
  read (connect, tsets, address, data);  
end loop;
```

- Wave forms may be based on common tester formats:

```
constant tset2 : wave_timing := (200 ns, new_time_data (  
  new_frame_set_array (NRET (0 ns), address_pins) +  
  new_frame_set_array (ROFF (0 ns, 30 ns), data_in_pins) +  
  new_frame_set_array (RONE (10 ns, 110 ns), we_pin) +  
  new_frame_set_array (NRET (0 ns), cs_pin) +  
  new_frame_set_array (NRET (0 ns), oe_pin) +  
  new_frame_set_array (STROBE (160 ns, 199 ns), data_out_pins));
```

- Wave forms are created by one or more wave form generators.
- A collection of files that describe the wave form(s) is known as a WAVES dataset.

Michael Sullivan and Michael Vincze

Defense Systems & Electronics Group

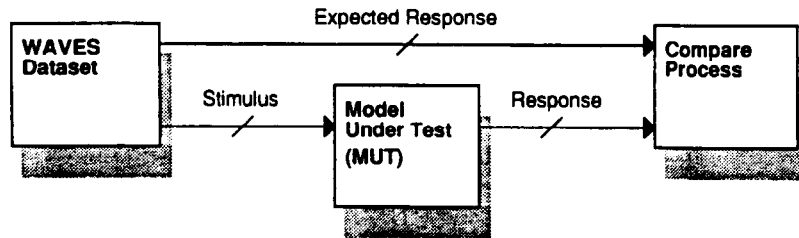


Integrating WAVES into an ASIC Simulation Environment

- **WAVES Test Bench**

- A WAVES dataset/wave form generator 's instantiation in a test bench is accomplished via a concurrent procedure and a process for each pin.
- The compare process and rest of test bench does not have to be WAVES compliant.

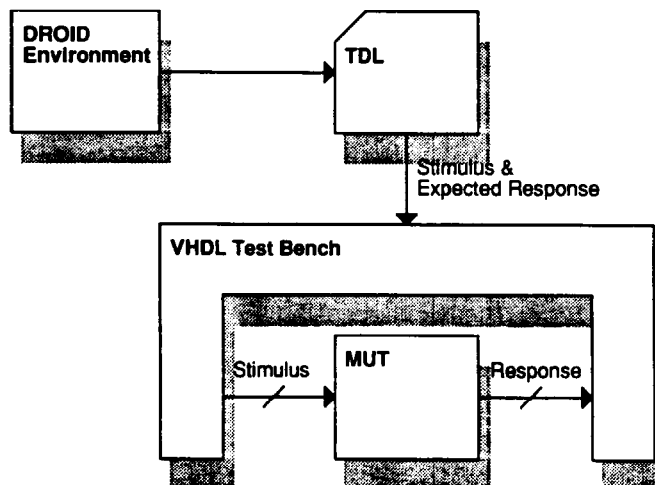
VHDL/WAVES Test Bench



Integrating WAVES into an ASIC Simulation Environment

- **Pre-existing ASIC Simulation Environment**

- A non-WAVES test bench reads test vectors from a TI vector file format (TDL.)
- The TDL file is generated from a proprietary TI design environment (DROID.)

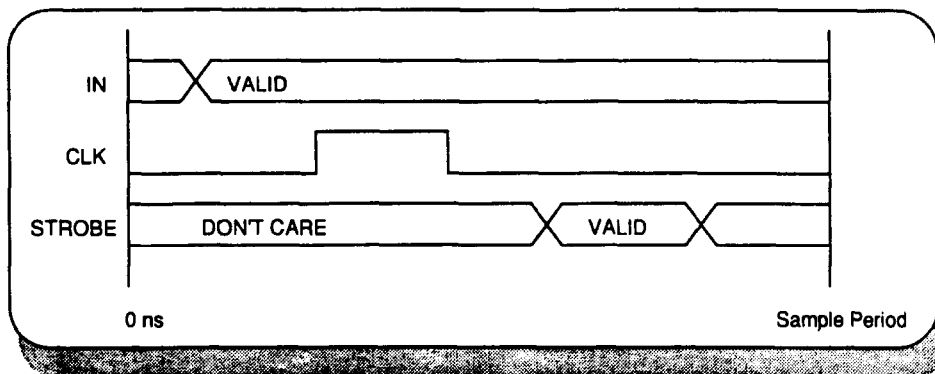
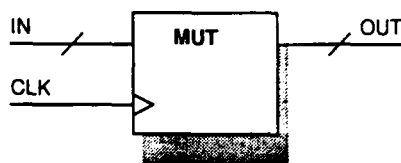




Integrating WAVES into an ASIC Simulation Environment

• ASIC Design

- All ASICs designed with a single continuous clock.
- All test vectors applied uniformly before clock edge.
- All outputs sampled (strobed) at a constant periodic rate (same rate as clock.)



Michael Sullivan and Michael Vincze

Defense Systems & Electronics Group

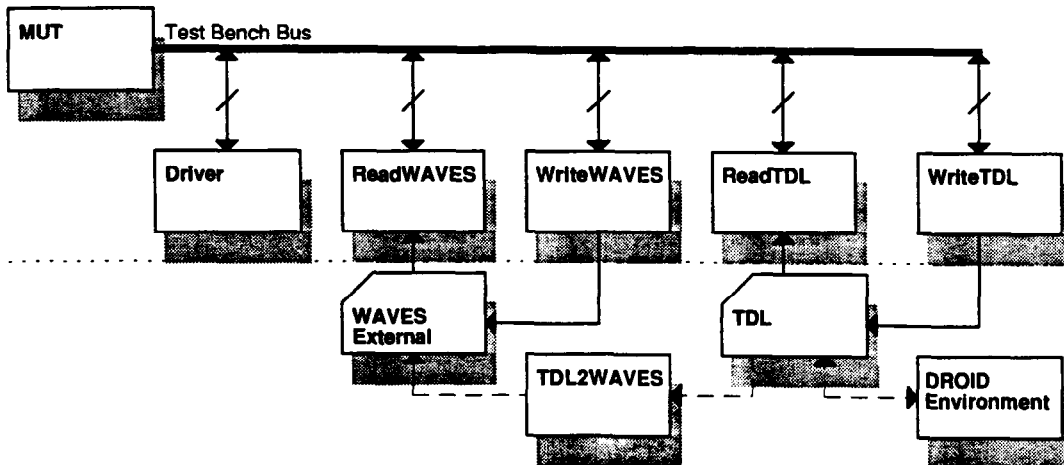


Integrating WAVES into an ASIC Simulation Environment

• New ASIC Simulation Environment with WAVES

- The parent test bench simulates using several child test benches.
- The child test benches can read and write vectors for TDL and WAVES formats.
- The new environment satisfies WAVES requirements and allows ties to and from the old environment.
- ReadWAVES, WriteWAVES, ReadTDL, and WriteTDL are automatically generated from programs written at TI.

Parent Test Bench



Michael Sullivan and Michael Vincze

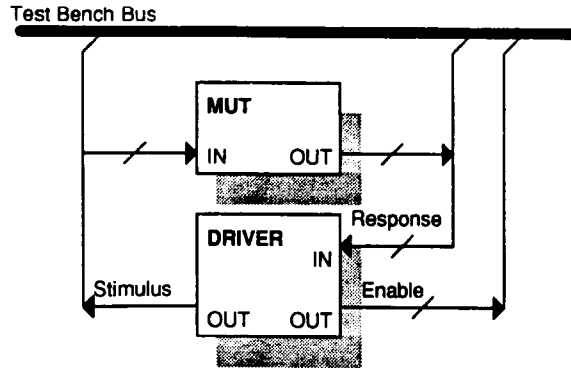
Defense Systems & Electronics Group



Integrating WAVES into an ASIC Simulation Environment

• Driver Child Test Bench

- Driver allows the designer to write test vectors in a typical VHDL test bench fashion (i.e.: by the use of assignment statements, wait statements, subprograms, etc.)
- Each bi-directional pin set requires an extra output signal specifying the bi-directional pin set's direction. The enable signal is used by WriteWAVES and WriteTDL to help determine if a signal from a bi-directional pin is a stimulus or a response.



Michael Sullivan and Michael Vincze

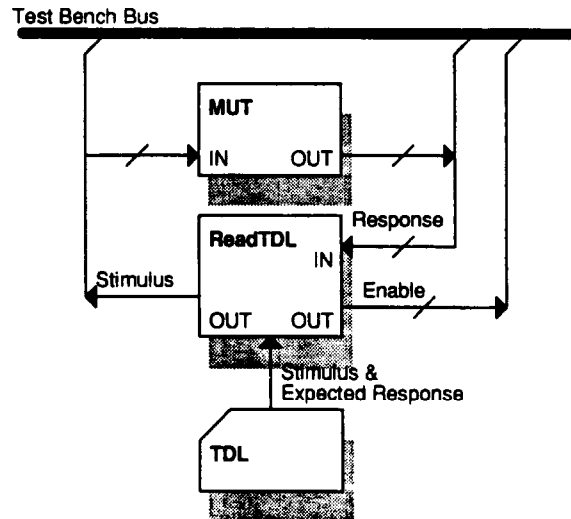
Defense Systems & Electronics Group



Integrating WAVES into an ASIC Simulation Environment

• ReadTDL Child Test Bench

- ReadTDL allows the designer to read a TDL file, apply stimulus, and compare the expected response with the actual response.
- Each bi-directional pin set requires an extra output signal specifying the bi-directional pin set's direction. The enable signal is used by WriteWAVES and WriteTDL to help determine if a signal from a bi-directional pin is a stimulus or a response.



Michael Sullivan and Michael Vincze

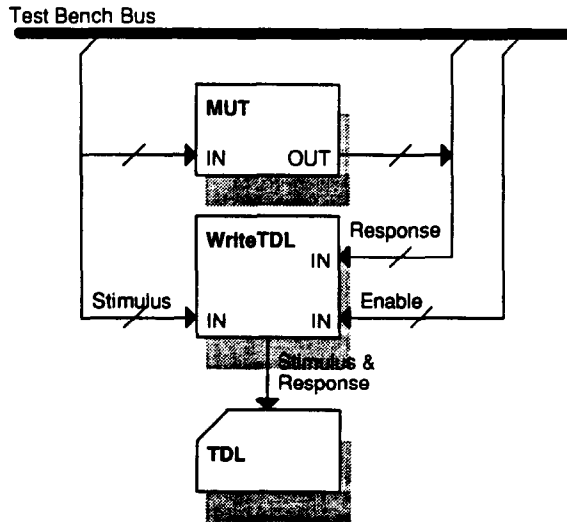
Defense Systems & Electronics Group



Integrating WAVES into an ASIC Simulation Environment

• WriteTDL Child Test Bench

- WriteTDL writes a TDL file based on the stimulus and the MUT's response.
- The enable signal is used to determine whether a signal on a bi-directional pin set is a stimulus or a response.



Michael Sullivan and Michael Vincze

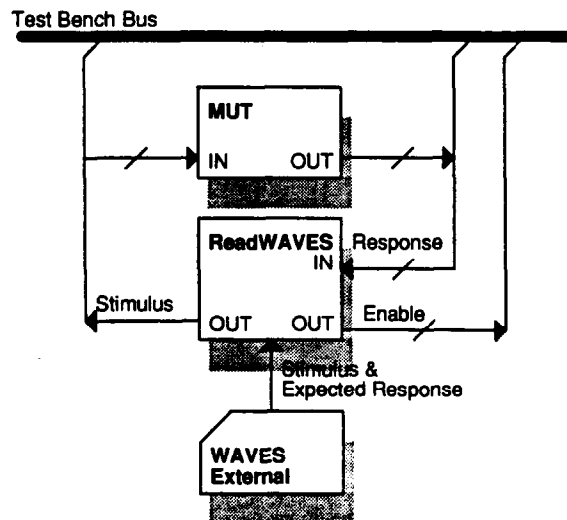
Defense Systems & Electronics Group



Integrating WAVES into an ASIC Simulation Environment

• ReadWAVES Child Test Bench

- ReadWAVES allows the designer to read a WAVES external file, apply stimulus, and compare the expected response with the received response.
- Each bi-directional pin set requires an extra output signal specifying the bi-directional pin set's direction. The enable signal is used by WriteWAVES and WriteTDL to help determine if a signal from a bi-directional pin is a stimulus or a response.



Michael Sullivan and Michael Vincze

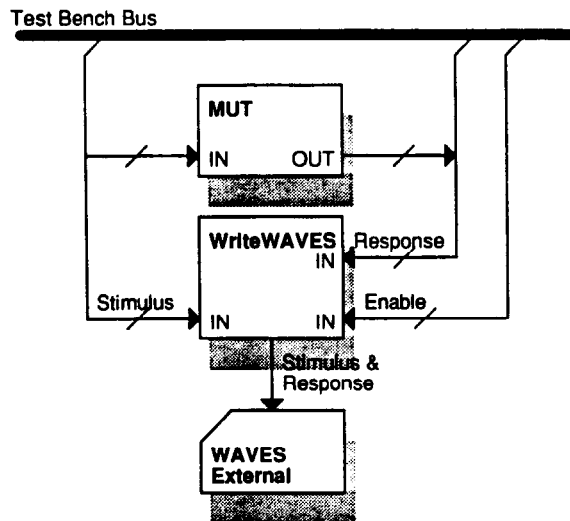
Defense Systems & Electronics Group



Integrating WAVES into an ASIC Simulation Environment

- **WriteWAVES Child Test Bench**

- WriteWAVES writes a WAVES external file based on the stimulus and the MUT's response.
- The enable signal is used to determine whether a signal on a bi-directional pin set is a stimulus or a response.



Michael Sullivan and Michael Vincze

Defense Systems & Electronics Group



Integrating WAVES into an ASIC Simulation Environment

- **Concerns**

- A WAVES analyzer was not readily available. Our WAVES dataset was checked for WAVES compliance by sending copies of the dataset to the WAVES Analysis and Standards Group.
- Bi-directional pins created problems in the test bench, as it was not readily known if its value was a stimulus or response. This was overcome by providing a bi-directional pin with two values (in the WAVES external file): one as an input value and the other as an output value.
- The need for having a process for each pin in the WAVES test bench created a lot of code. This was overcome by grouping pins according to their modes (i.e.: INs, OUTs, INOUTs, etc), and placing signals assigned to these pins in vectors. This allowed the use of generate statements.



Integrating WAVES into an ASIC Simulation Environment

- **Conclusion**

- WAVES can be successfully integrated into an ASIC simulation environment.
 - Integration successfully tested on a 15K integrated circuit design.
- The integration can allow for test vectors to be moved between the pre-existing and new design environments.
- Based on consistent ASIC designs, such as having a single continuous clock, test benches can be automatically generated.