IEEE 1149.4 Mixed-Signal Test Bus Working Group Meeting Minutes

for
January 18th, 2005

9:00AM-11:00AM

Meeting Agenda:

Time	Topic	Responsibility
9:00 AM	Introduction	Bambang Suparjo
9:05-11:00 AM	1. Review of Strawdog Proposal. 2. ABSDL Validation Progress. 3. Identification of Critical Issues for Next Actions.	All members
11:00 AM	Summary and Adjourn	Bambang Suparjo

Meeting Attendees:

Name	Company
Pete Collins	JTAG Technology
Anthony Sparks	JTAG Technology
Adam Ley	Asset-Intertech
Keith Lofstrom	Keith Lofstrom Integrated Circuits
Heiko Ehrenberg	Goepel
Karla May	Corelis
Ken Parker	Agilent Technologies
Adam Cron	Synopsys
Bambang Suparjo	Mentor Graphics

Sending Regret:

- Adam Osseiran
- Steve Sunter

1. Introduction

Bambang moderated the teleconference. The meeting was based on topic "Language Definition" in meeting minutes May 5th 2000 available at http://grouper.ieee.org/groups/1149/4/min0500.html.

2. Review of Strawdog Proposal and ABSDL Validation Progress

Ken Parker suggested to review the email from Keith Lofstrom (see Appendix A) to start the discussion. Keith provided a brief explanation on his email and suggested that the BSDL_extension should not include the detail properties of analog model/switch but should allow cross-reference to a separate file that contains the detail of analog model/switch with the assumption that analog model/switch properties may be different from vendor to vendor. With this approach, it can simplify the BSDL_extension description and provides accurate circuit analysis. Heiko agreed with this suggestion. However, Anthony informed the group that using multiple files may complicate the reading and processing of BSDL file.

Adam Ley suggested to the group to review the individual requirements for ABM and TBIC. To help the discussion Bambang suggested to the group to refer the BSDL example in Straw Dog Proposal (see Appendix B). For ABM case, the operation should follow Table 6 (Switch Patterns for Sample ABM) and Table 8 (Switching Pattern Requirements for Sample ABM) in the 1149.4 standard. Table 6 in the documentation provides 16 patterns that control the switch conditions to set the pin states. The associate data of C, D, B1 and B2 for each pattern is shown in Table 8. Based on those tables (where the switch conditions have been set by the patterns) and since they are all set by the standard, and after long discussion with Ken Parker and other members, the SW_list attribute as shown in Straw Dog Proposal is not needed and can be removed. Hence the Par_list can also be removed.

Original Description:

```
attribute ABM_List of Straw_Dog : entity is

-- ABM(num), Port, Bus 1, Bus 2, TBIC Partition, Switches
"ABM(1) ( W, 12, 13, TBIC(1), SW_List(1)), "&
"ABM(2) ( Y, 8, 9, TBIC(1), SW_List(1)) ";

attribute Switch_List of Straw_Dog : entity is
"SW_List(1) ("&
-- Param, V_threshold, Tolerance, Reference_port
"Vth, ( 2.5, 10%, Vss), "&

-- pathway from_port to_port Par_List
"SH ( Vdd, ABM, Par_List(1)), "&
"SL ( ABM, Vss, Par_List(1)), "&
"SG ( ABM, Vss, Par_List(1)), "&
"S5+SB1 ( AT1, ABM, Par_List(2)), "&
"S6+SB2 ( ABM, AB2, Par_List(2)), "&
") ";
```

Proposed Description:

```
attribute ABM_List of Straw_Dog : entity is
```

```
-- ABM(num), Port, Bus 1, Bus 2, TBIC Partition
"ABM(1) ( W, 12, 13, TBIC(1)), "&
"ABM(2) ( Y, 8, 9, TBIC(1)) ";
```

For TBIC case, the attribute will be reviewed by Adam Ley since it has been found that there is an issue related to TBIC calibration after reviewing Figure 16 (Control of a partitioned bus structure) in the 1149.4 standard.

3. Identification of Critical Issues for Next Actions.

Adam Ley will review the TBIC description that will take into consideration on calibration issue.

4. Summary and Adjourn

Bambang summarized the meeting; remove the SW_list in ABM attribute and Adam Ley will review the TBIC description. The tentative date for the next meeting will be on February 15th 2005 at 9:00 AM. The teleconference was adjourned at 11:00 AM.

Appendix A – Email from Keith Lofstrom

Date: Thu, 28 Oct 2004 08:17:42 -0700
From: Keith Lofstrom <keithl@kl-ic.com>
To: dot4 <stds-1149-4wg@mail.ieee.org>
Subject: ABSDL and AMODEL - split the problem up!

Thanks for a lively Dot4 meeting! I thought it over, and talked with the boss (when you run a small business, that's your spouse), and decided to contribute to the ABSDL discussion.

This problem might be a lot easier if we divide it into two parts, a "structural" ABSDL file, and a "measurement" AMODEL file.

AMODEL:

The switch and wire models do not need to know about structure; they may be passive or active, lumped or distributed, simple or fiendishly elaborate. The models may vary by application, frequency range, temperature, process, manufacturer, even by customer (some may get more information than others). These models are used in a larger model to generate the expected test result out of the analog measurement gear.

** The models do NOT affect which bits are sent to the boundary **
** scan chain, so they do NOT need to be in the ABSDL file!!! **

The easiest way to make these models is just to do a pin-to-pin black box SPICE model for each of the two paths of every ABM pin to the associated TBIC pins. Want to connect two ABMs to the same bus? Sorry, no model, we don't normally support that. We might need a few extra models for calibration and characterization paths, but no analog test modeling tool should have to deal with the general case of all pins to all pins. Anyone is free to build special cases, of course.

The analog test modeling tool can just follow some simple rules to select the appropriate pin-to-pin model out of a (very large) AMODEL file. And each pin-to-pin model in the very large AMODEL file can be verified, path by path, with ATE and network analysis applied to the pins. How large a file? Tens of megabytes possibly. And there will be many files for one chip, depending on the circumstances listed in the third paragraph above. So what? Wouldn't you rather have the manufacturer do this, and generate the models by whatever means at their disposal, and not expect an ATPG program to figure it out?

The analog test modeling tool will just pick a test (structurally), yank models, append them to a SPICE file, run the simulation, and generate an expected value distribution for the measurement unit. A lot of very quick SPICE runs, hopefully. The SPICE simulator can come from a SPICE vendor. The analog test modeling tool just needs to be able to grep or perl the SPICE output and calculate parametrics from that.

I will follow up with some suggestions about the model file format, to help a parser pick its way through it, but in general the idea is to specify a way of constructing "info lines" (A better name is needed). An info line is a column one asterisk (SPICE comment) followed by a keyword followed by "useful information" about the SPICE-formatted models, global and local. Which SPICE simulator was used to make the model? Check the info line. Which temperature? Check an info line. Etc. The actual SPICE models might all have some incomprensible 30 character name, but an info line in front will tell us which pins and what conditions. Enough information so a text parser can decide which chunks of SPICE model text to pull and append to a SPICE file. Note that SPICE has an "include" function and a "library" function, and permits hierarchical nesting of models, so elaborate models can be imbedded without any extra complexity for the analog test modeling tool.

We don't have to always use SPICE, of course. The same technique could easily imbed models from other simulators (AHDL or verilog-A or Spectre) and even mix them, if the board designer has enough simulators around. The suggested default should be standard Berkeley SPICE (version 3F5 ?), available for free everywhere. But - I repeat - the analog test modeling tool does NOT need to understand these models, just patch them into a file and call the simulator to deal with them.

ABSDL:

The structural ABSDL model is entirely independent of this. If

you want to get to a particular ABM from the TBIC, there is only
one way to get there. You can number the buses and the switches
along the way, and use those to choose the bits for a hierarchical
route, but all the digital test pattern generator needs to know
about is the structure of the path, not resistance or capacitance
or leakage. Maybe voltage limits, but probably not even that;
that will be a board design issue instead. The ATPG will need
to know "force a current here", "measure a voltage there", but
the actual current needed and the voltage expected will come
from the analog test modeling tool, a different chunk of software,
very likely from a different vendor than the ATPG vendor. Who
merges the test patterns with the parametrics, and how? That is
probably something we can punt to the test description language
folks.

And in the beginning, with just an ABSDL file, we may not need
any models at all. Just stick a golden board into the test socket,
make the measurements, and log them. With a structure-only ABSDL
file, we can get started making measurements now, and learn to
predict them with elaborate SPICE models later. That might help
us choose modeling methods and accuracy before we get too far down
the AMODEL path.

- - - - -

This approach divides the problem cleanly in half, and allows a
simple and logical [;-)] extension of BSDL into ABSDL. By
standarizing on a pin-to-pin SPICE model, we can make those models
structurally independent, and not the worry of the bit jockeys.

Keith

--
Keith Lofstrom keithl@keithl.com Voice (503)-520-1993
KLIC --- Keith Lofstrom Integrated Circuits --- "Your Ideas in Silicon"
Design Contracting in Bipolar and CMOS - Analog, Digital, and Scan ICs

Appendix B – BSDL Example in Straw Dog Proposal

```
-- Straw Dog 0-- Straw Dog 1.0 BSDL description of a simple .4 device
-- This is an 1149.1-compatible BSDL that does encode 1149.4 features.
-- Still suitable for use with 1149.1 Interconnect Testing software.


entity Straw_Dog is
generic (PHYSICAL_PIN_MAP : string := "dip");

port(TCK, TDI, TMS: in bit;
A, B: in bit; -- Digital pins
W, Y: inout bit; -- Analog pins
TDO: out bit;
AT1, AT2: inout bit; -- ATAP pins (bidir)
GND, VCC:linkage bit);
```

```
use STD_1149_1_1994.all; -- Get Std 1149.1-1990 attributes and
definitions
use STD_1149_4_2000.all; -- Get .4 attributes and definitions

attribute COMPONENT_CONFORMANCE of Straw_Dog : entity is
"STD_1149_1_1993";

attribute PIN_MAP of Straw_Dog : entity is PHYSICAL_PIN_MAP;

constant dip:PIN_MAP_STRING:="A:7, B:8, W:9, Y:12, AT1:10, AT2:11, " &
"GND:2, VCC:1, TDO:6, TMS:3, TCK:4, TDI:5";

attribute TAP_SCAN_IN of TDI : signal is true;
attribute TAP_SCAN_MODE of TMS : signal is true;
attribute TAP_SCAN_OUT of TDO : signal is true;
attribute TAP_SCAN_CLOCK of TCK : signal is (20.0e6, BOTH);

attribute INSTRUCTION_LENGTH of Straw_Dog : entity is 2;

attribute INSTRUCTION_OPCODE of Straw_Dog : entity is
"BYPASS (11)," &
"EXTEST (00)," &
"SAMPLE (01)," &
"PROBE (10)";

attribute INSTRUCTION_CAPTURE of Straw_Dog : entity is "01";

attribute BOUNDARY_LENGTH of Straw_Dog : entity is 14;

attribute BOUNDARY_REGISTER of Straw_Dog : entity is
-- num cell port function safe [ccell disval rslt]
"0 (BC_1, A, input, x), " &
"1 (BC_1, B, input, x), " &

-- The following cells are TBIC controls
-- num cell port function safe [ccell disval rslt]
"2 (BC_7, AT1, bidir, 0, 4, 0, Z), " &
"3 (BC_7, AT2, bidir, 0, 4, 0, Z), " &
"4 (BC_1, *, control, 0), " & -- required safe bits
"5 (BC_1, *, internal, 0), " & -- required safe bits




-- The following cells control the Y analog signal

"6 (BC_7, Y, bidir, 0, 7, 0, Z), " &
"7 (BC_1, *, control, 0), " & -- required safe bits
"8 (BC_1, *, internal, 0), " & -- required safe bits
"9 (BC_1, *, internal, 0), " & -- required safe bits


-- The following cells control the W analog signal

"10 (BC_7, W, bidir, 0, 11, 0, Z), " &
"11 (BC_1, *, control, 0), " & -- required safe bits
"12 (BC_1, *, internal, 0), " & -- required safe bits
```

```
"13 (BC_1, *, internal, 0) " ; -- required safe bits


-- Now add extensions for 1149.4. The declarations given here could be
-- included in a user package named "STD_1149_4_200x instead of being
listed
-- here.


-- Extension declarations
attribute ABM_List : BSDL_Extension;
attribute TBIC_List : BSDL Extension
attribute SWITCH_List : BSDL_Extension;;
attribute Parametrics_List : BSDL_Extension;
attribute Residual_Table : BSDL_Extension;

-- Extension definitions
attribute ABM_List of Straw_Dog : entity is

-- ABM(num), Port, Bus 1, Bus 2, TBIC Partition, Switches

"ABM(1) ( W, 12, 13, TBIC(1), SW_List(1)), "&
"ABM(2) ( Y, 8, 9, TBIC(1), SW_List(1)) ";

attribute TBIC_List of Straw_Dog : entity is
-- TBIC, AT1 Port, AT2 Port, Char_cell, Switches
"TBIC(1) ( AT1, AT2, 5, SW_List(2)) ";

attribute Switch_List of Straw_Dog : entity is
"SW_List(1) ("&
-- Param, V_threshold, Tolerance, Reference_port
"Vth, ( 2.5, 10%, Vss), "&

-- pathway from_port to_port Par_List
"SH ( Vdd, ABM, Par_List(1)), "&
"SL ( ABM, Vss, Par_List(1)), "&
"SG ( ABM, Vss, Par_List(1)), "&
"S5+SB1 ( AT1, ABM, Par_List(2)), "&
"S6+SB2 ( ABM, AB2, Par_List(2)), "&
") ";


"SW_List(2) ("&
-- Param, V_threshold, Tolerance, Reference_port
"Vth, ( 2.5, 10%, Vss), "&

-- pathway from_port to_port Par_List
"S1 ( Vdd, AT1, Par_List(3)), "&
"S3 ( AT1, Vss, Par_List(3)), "&
"S2 ( Vdd, AT2, Par_List(3)), "&
"S4 ( AT2, Vss, Par_List(3)), "&

") ";


attribute Parametrics_List of Straw_Dog : entity is
```

```
-- Table Min_Z, Max_Z, Min_off, Max_off, Min_I, Max_I, Min_V, Max_V
"Par_List(1) (400, 600, 0, 0, 0, 0, 0, 5) ";
"Par_List(2) (250, 400, 0, 0, 0, 0, 0, 5) ";
"Par_List(3) (175, 225, 0, 0, 0, 0, 0, 5) ";

attribute Residual_Table of Straw_Dog : entity is

-- Residual Elem, Units, From, To, Min, Max, Match/Correl
"Z1 (Ohms, ABM(1), Internal N, 50, 60, *), "&
"Z2 (Ohms, Internal N, W, 110, 120, Z1/0.95), "&
"Z3 (Ohms, Internal N, ABM(2), 40, 45, *), "&
"W1 (*, ABM(2), Y, *, *, *) ";


end Straw_Dog;
```