

Analog Boundary-Scan Description Language (ABSDL) for Mixed-Signal Board Test

Bambang Suparjo¹, Adam Ley², Adam Cron³ and Heiko Ehrenberg⁴

¹Mentor Graphics Corp., ²ASSET InterTech, Inc., ³Synopsys, Inc., ⁴GOEPEL Electronics LLC

Abstract

The IEEE Standard 1149.4 has been ratified and available for some time, now. However, describing the architectural content of an analog boundary-scan register has not yet been standardized. The work of the IEEE Std 1149.4 Working Group over the past several years has been to define and codify an extension to IEEE Std 1149.1's BSDL. In this paper, a language to describe the boundary-scan implementation in a mixed-signal device is proposed. The language is called Analog Boundary-Scan Description Language (ABSDL) and it is compatible with the existing IEEE Std 1149.1 BSDL. Using this language, test generation automation can proceed, and interconnect test on a mixed-signal board using analog boundary-scan cells can be performed to test both simple wires and discrete components between packaged devices with IEEE Std 1149.1 and 1149.4 infrastructure and access. Although not yet promulgated, the syntax definitions proposed in this paper are indicative of the current state of the Working Group's attempts at this effort. Along with the generation of the syntax for describing IEEE Std 1149.4 structures, the semantic checks for such an ABSDL file are presented.

1 Introduction

The IEEE Standard for a Mixed-Signal Test Bus (1149.4) was approved in 1999. It contains detailed descriptions for the test bus architecture including the test access port, test bus interface circuits and analog boundary modules, and also the control protocol and rules for test application. The development of ABSDL was initiated in 2000 and became the 1149.4 Working Group focus project in 2005 [3]. Section 2 of this paper provides a brief overview of the IEEE 1149.4 Standard. Section 3 describes ABSDL and the development approach used to arrive at this stage. Section 4 provides application examples of ABSDL. The future plan for continuing development is discussed in Section 5 and finally the conclusion is in Section 6.

2 IEEE Std 1149.4 Overview

The Mixed-Signal Test Bus Standard was designed to augment the digital Test Access Port and Boundary-Scan Standard by adding provisions to measure discrete analog components connected between 1149.4 compliant devices. Figure 1 shows a chip-level view of the 1149.4

architecture. The Analog Boundary Modules (ABM) allow testing at analog pins via the Analog Test Access Port (ATAP).

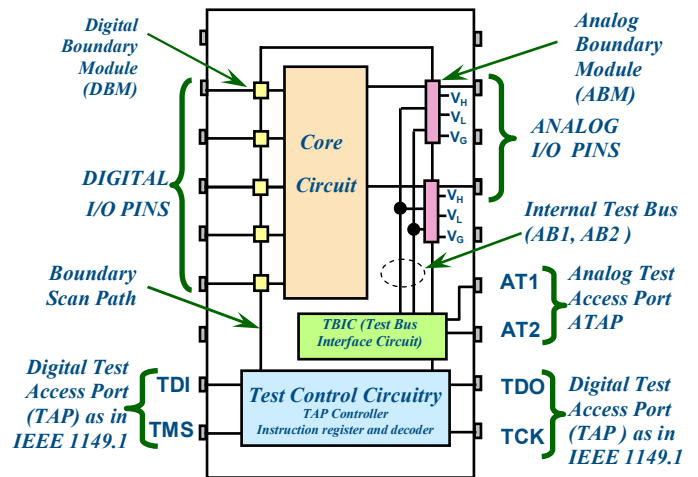


Figure 1: Top-Level 1149.4 Architecture

Zooming in on the ABM structure, Figure 2 shows the detailed architecture of an Analog Boundary Module. Analog switches associated with each pin, and controlled by the 1149.1 infrastructure, allow connection of the pin to various static reference sources and internal analog busses, which in turn connect to the ATAP pins of the device. [4], [5] and [6] explain how these features aid board-level test.

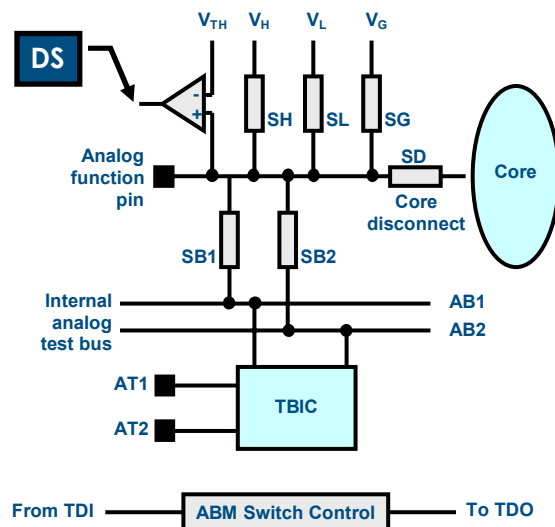


Figure 2: Analog Boundary Module Detail

Once designed into a chip and mounted on a board, ABSDL aids in automating the test development process.

3 ABSDL Development

The purpose of ABSDL is to provide the boundary-scan description of a mixed-signal device designed to be compliant with the IEEE Std 1149.4. The description includes information about the interconnection between cells in the device's boundary-scan chain and the device pins. It also defines some aspects about the data to control the Test Bus Interface Circuit (TBIC) and Analog Boundary Modules (ABM). This information can then be used to generate patterns for interconnect and discrete component testing at the board level. Since ABSDL is compatible with 1149.1 BSDL, both digital and mixed signal interconnect could be tested simultaneously.

3.1 Development Approach

The development of the ABSDL language needs to meet the rules and requirements of IEEE Std 1149.4. There are two key modules in the 1149.4 architecture: the TBIC and ABM. The Standard document shows a sample implementation for each. The document then uses the sample implementation to help define the required functions of each part.

The TBIC description in ABSDL follows the operational requirements shown in Table 1 and Table 2 in the IEEE Std 1149.4 document:

- Table 1 – Switching patterns for TBIC
- Table 2 – TBIC switching assignments for defined instructions

Similarly, the ABM description in ABSDL follows the operations shown in Table 6, Table 7 and Table 8 in the IEEE Std 1149.4 document:

- Table 6 – Switch patterns for ABM
- Table 7 – Functions of ABM switching patterns
- Table 8 – Switching pattern requirements for ABM (considering defined instructions)

An example of an 1149.4 device is shown in Figure 3. This virtual device has two single-ended digital pins (A and B), two digital differential pin pairs (D1 and D1N, and D2 and D2N), two analog pins (W and Y) and two analog differential pin pairs (X1 and X1N, and X2 and X2N). The 1149.1 TAP interface pins are TDI, TDO, TMS, and TCK. The analog 1149.4 ATAP pins are actually a differential pair set (AT1 and AT1N, and AT2 and AT2N). NC1 and NC2 are “no-connect” pins, while VCC and GND are power and ground, respectively. The examples given in this paper are referenced against this fake device with a possible internal test structure as shown in Figure 4.

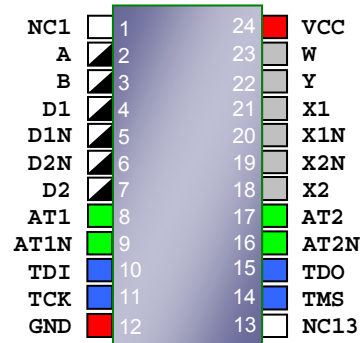


Figure 3: Example 1149.4 Device

A real device was also used during the development of ABSDL. Detailed in [7], the device datasheet can be viewed at [8]. This device is actually one of the first production devices with 1149.4. This device was also used as a test-bed for ABSDL development. If the Working Group could not define this simple device using ABSDL, then we were not making adequate progress to address the industry needs.

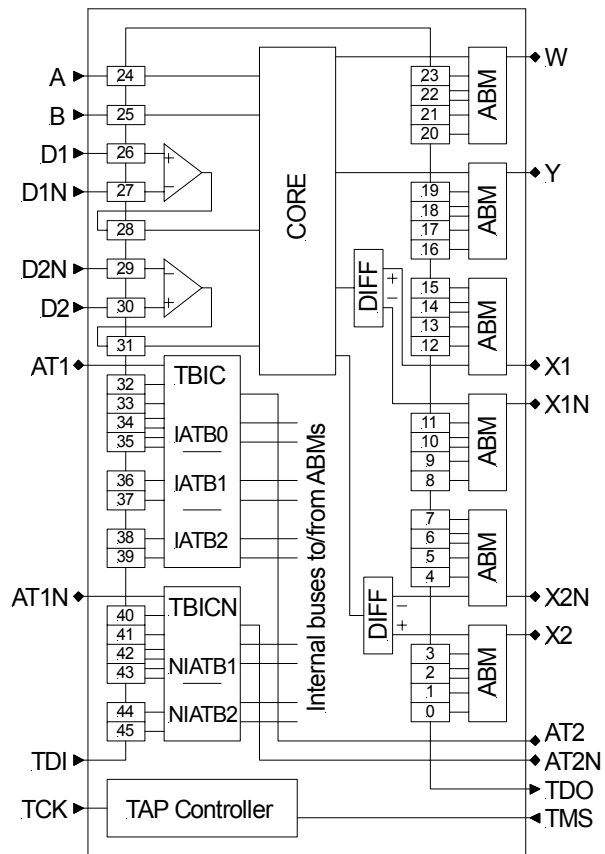


Figure 4: Example of 1149.4 test structure.

3.2 ABSDL Components

Like BSDL, ABSDL uses VHDL constructs to define the required components. ABSDL adds onto BSDL. To make a complete ABSDL description file, BSDL and BSDL extensions come together to make ABSDL.

The ABSDL components are:

- Port Description Statements
- Standard and Optional Use Statements
- Component Conformance Statements
- Device Package Pin Mapping
- Grouped Port Identification
- Scan Port Identification
- Instruction Register Description
- Optional Register Description
- Register Access Description
- Boundary-Scan Register Description
- ATAP Identification Statements
- TBIC Statements
- ABM Pin Statements
- Differential Pin Statements

Sections 3.2.1 to 3.2.10 describe the basic ABSDL components and sections 3.2.11 to 3.2.14 describe the extended components. The semantic checks described in this section are for components that are relevant to 1149.4, only. Examples shown with each component description relate back to the internal test structure shown in Figure 4.

3.2.1 Port Description Statements

The pin types in the port description statements for different sorts of pins found on an 1149.4 compliant component are shown in Table 1.

Pin	in	out	inout	linkage
TAP	X	X		
ATAP (Single Ended)			X	
ATAP (Differential)			X	
Digital (Single Ended)	X	X	X	
Digital (Differential)	X	X	X	
Analog (Single Ended)			X	
Power				X
NC (no connection)				X

Table 1: Pin Types

In order to ensure the description statements are written correctly, semantic checks need to be performed. The semantic checks for the port description statements are:

- Port identifiers for the two ATAP pins (AT1 and AT2) shall be specified with pin type **inout** assigned to them

- If the device has a differential ATAP, port identifiers for the additional ATAP pins (AT1N and AT2N) shall be specified with pin type **inout** assigned to them
- Analog ports with 1149.4 test resources shall be specified with pin type **inout**
- Digital ports with 1149.4 test resources shall be specified with pin type **inout**
- Analog differential ports with 1149.4 test resources shall be specified with pin type **inout** (both the positive and negative ports of the differential pair)
- Digital differential ports with 1149.4 test resources shall be specified with pin type **inout** (both the positive and negative ports of the differential pair)
- Linkage ports with 1149.4 test resources shall be specified with pin type LINKAGE

An example ABSDL listing of a port description is shown below:

```
Port (
    TCK, TDI, TMS: in bit;
    TDO: out bit;
    A, B: in bit;
    D1, D1N: in bit;
    D2, D2N: in bit;
    AT1, AT2: inout bit;
    AT1N, AT2N: inout bit;
    W, Y: inout bit;
    X1, X1N: inout bit;
    X2, X2N: inout bit;
    NC1, NC13: linkage bit;
    GND, VCC: linkage bit);
```

3.2.2 Standard and Optional Use Statements

The standard **use** statement for 1149.1 is:

```
use STD_1149_1_2001.all;
```

This statement allows ABSDL to refer to the attributes and definitions in the 1149.1-2001 Standard, for example.

The optional **use** statement for 1149.4 is:

```
use STD_1149_4_version.all;
```

Similarly, this statement will allow ABSDL to refer to the attributes and definitions of 1149.4-version, for example. The word “version” is referring to the next version of 1149.4 standard that will release the Mixed Signal Test (MST) attributes and definitions.

The semantic check for standard and optional **use** statements is:

- A **use** statement shall exist for a standard 1149.4 package file to obtain the Mixed Signal Test (MST) attribute definitions from the 1149.4 standard package file string, **STD_1149_4_version**

3.2.3 Component Conformance Statements

Due to the fact that Standards documents get released, and then re-released with revisions, a mechanism to define to which revision a device conforms is required. The component conformance statement provides this mechanism in the ABSDL. For 1149.1 conformance, the following would be added:

```
attribute COMPONENT_CONFORMANCE of component_name
: entity is "STD_1149_1_2001";
```

For 1149.4, the component conformance statement is:

```
attribute MST_Component_Conformance of
component_name : entity is
"STD_1149_4_1999";
```

The semantic checks for the component conformance statement are:

- Component conformance shall be specified with the **MST_Component_Conformance** attribute
- The only valid conformance string is **STD_1149_4_1999**

3.2.4 Device Package Pin Mappings

An example of pin mapping information is shown below:

```
constant device_package_name : PIN_MAP_STRING :=
"TCK:11, TDI:10, TMS:14, " &
"TDO:15, " &
"AT1:8, AT2:17, " &
"AT1N:9, AT2N:16, " &
"A:2, B:3, " &
"D1:4, D1N:5, " &
"D2:7, D2N:6, " &
"W:23, Y:22, " &
"X1:21, X1N:20, " &
"X2:18, X2N:19, " &
"NC1:1, NC13:13, " &
"GND:12, VCC:24";
```

This example is referring to the 1149.4 device shown in Figure 3. Each pin is mapped to its associated pin number defined for the package.

The device package pin mappings can be described as:

```
attribute PIN_MAP of component_name : entity is
PHYSICAL_PIN_MAP;
```

Once this statement is given, details of the port-to-pin mapping are provided by the pin map string.

3.2.5 Grouped Port Identification

As in 1149.1, grouped port identification is also required for 1149.4 devices. The digital and analog differential (voltage or current) grouped ports can be identified by issuing statements such as the one shown below:

```
attribute PORT_GROUPING of component_name :
entity is
"Differential_Voltage ((D1, D1N), (X1,X1N)), "&
"Differential_Current ((D2, D2N), (X2,X2N));"
```

D1, D1N, D2 and D2N are digital differential pins and X1, X1N, X2 and X2N are analog differential pins.

3.2.6 Scan Port Identification

Since 1149.4 is also using the 1149.1 scan ports, the ports' identification for an 1149.4 device are shown below:

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

The data path for test control and observe data is the same as that used for 1149.1. The protocol has been maintained so that interoperability of both 1149.1 and 1149.4 devices interconnected at the board level is guaranteed.

3.2.7 Instruction Register Description

In 1149.4, the PROBE instruction is a mandatory instruction. Therefore it needs to be defined in the instruction register description. An example of an instruction register description is shown below:

```
attribute INSTRUCTION_LENGTH of component_name :
entity is 4;
```

```
attribute INSTRUCTION_OPCODE of component_name :
entity is
-- mandatory instructions (1149.1)
"EXTEST (1000), " &
"PRELOAD (0010), " &
"SAMPLE (0010), " &
"BYPASS (1111), " &
-- mandatory instruction (1149.4)
"PROBE (1011), " &
-- optional instructions (1149.1)
"CLAMP (0100), " &
"HIGHZ (1101), " &
"INTEST (1110), " &
"IDCODE (0001)";
```

```
attribute INSTRUCTION_CAPTURE of component_name :
entity is "0001";
```

The semantic checks for the instruction register description are:

- PROBE is a mandatory instruction for an 1149.4 compliant device and therefore shall be defined in the **INSTRUCTION_OPCODE** attribute
- The opcode value assigned to the PROBE instruction shall be unique
- The opcode for PROBE can be any value except the all 1s value

3.2.8 Optional Register Description

The optional register descriptions are identical to that of 1149.1. Optional registers discussed in the 1149.1 document include IDCODE and USERCODE. Also like 1149.1, users can address other internal data registers with either public or private instruction decodings.

3.2.9 Register Access Description

As in 1149.1, all instructions place a test data register between TDI and TDO. Therefore, the PROBE instruction used in 1149.4 needs to be included in the register access description. An example of a register access description is shown below:

```
attribute REGISTER_ACCESS of component_name:
entity is
-- access to mandatory registers
-- (1149.1)
"BOUNDARY (EXTEST, PRELOAD, SAMPLE)," &
"BYPASS (BYPASS), " &
"BOUNDARY (INTEST), " &
-- access to mandatory registers
-- (1149.4)
"BOUNDARY (PROBE), " &
-- access to optional register
-- (1149.1)
"DEVICE_ID (IDCODE)";
```

The semantic checks for the register access description are:

- The **REGISTER_ACCESS** attribute for the mandatory instruction PROBE shall be specified
- PROBE shall access the Boundary-Scan register

3.2.10 Boundary-Scan Register Description

The boundary-scan register description consists of information about the boundary-scan chain including the length, the description of each boundary-scan cell, its function or application, and its position in the boundary-scan register. The description of the boundary-scan chain length is defined with an attribute statement similar to the following:

```
attribute BOUNDARY_LENGTH of component_name:
entity is 46;
```

The following boundary-scan cell descriptions refer to the example shown in Figure 3 and Figure 4. Before describing each cell, the statement, below, needs to be issued:

```
attribute BOUNDARY_REGISTER of component_name :
entity is
```

After this pre-able, the complete description of each boundary-scan register cell is given. For the example chip, there are descriptions for cells attached to digital, analog, and ATAP pins in the design. The examples which follow

detail how these descriptions might be composed for the sample design.

a. Digital Pins

The boundary-scan cell description for cells associated with digital pins is identical to the IEEE Std 1149.1 BSDL syntax. An example of the description is shown below:

```
-- following 2 cells are for
-- digital pins
-- num cell port function safe
-- [ccell disval rslt]
"24 (BC_1, A, input, x), " &
"25 (BC_1, B, input, x), " &

-- following 6 cells are for
-- digital differential pins
-- num cell port function safe
-- [ccell disval rslt]
-- D1:D1N Single Ended:
"26 (BC_1, D1, input, x), " &
"27 (BC_1, D1N, input, x), " &
"28 (BC_1, *, internal, x), " &
-- D2:D2N Single Ended:
"29 (BC_1, D2, input, x), " &
"30 (BC_1, D2N, input, x), " &
"31 (BC_1, *, internal, x), " &
```

The number associated with each cell refers to the test structure example shown in Figure 4. For port A, the cell number is 24, the boundary scan cell type is BC_1, the cell function supports an input, and the cell safe value is X (don't care).

Cell 28 is for the internal single ended resolution point of the differential pins D1 and D1N, while cell 31 is for the internal single ended resolution point of the differential pins D2 and D2N. An asterisk (*) is used as the internal cell port name.

b. TBIC Control

The description of cells controlling the TBIC is shown below:

```
-- following 4 cells are TBIC Controls
-- num cell port function safe
-- [ccell disval rslt]
"32 (BC_1, *, internal, 0), " & -- Ca
"33 (BC_1, *, control, 0), " & -- Co
"34 (BC_7, AT1, bidir, 0, " &
"33, 0, Z), " & -- D1
"35 (BC_7, AT2, bidir, 0, " &
"33, 0, Z), " & -- D2
```

The safe value for each cell is set to 0. This implies that when the TBIC in a device is not in use during a specific test application, the internal analog test bus lines will be disconnected from the ATAP pins, AT1 and AT2, as defined in Table 1 and Table 2 in the standard documentation. Both cells 34 and 35 use the BC_7 boundary scan cell type because they are not only supplying control logic values to D1 and D2 but also allow monitoring of the one-bit digital representation on each

ATAP pin (during simple interconnect testing) as specified in rule 6.2.1.1(I) in the 1149.4 standard documentation.

The semantic checks for the TBIC Control cells refer to section 6.3 in the standard documentation:

- Boundary-Scan cells shall be specified for TBIC Control as follows
 - Calibrate cell, Ca (**internal**)
 - Control cell, Co (**control**), shall be listed in the **disable specification** for the TBIC Base Partition's D1 and D2 cells
 - D1 cell, AT1 pin (**bidir**), **disable specification** shall be defined
 - D2 cell, AT2 pin (**bidir**), **disable specification** shall be defined

c. TBICN Control

For differential applications, boundary-scan cells in the TBIC are needed to control the negative legs of the ATAP (ATAPN). In order to meet ABSDL syntax requirements, TBICN is used instead of TBIC as the naming nomenclature. An example of TBICN Control is shown below:

```
-- following 4 cells are TBICN Controls
-- num cell port function safe
--
--      [ccell disval rslt]
"40 (BC_1, *, internal, 0), " & -- Ca
"41 (BC_1, *, control, 0), " & -- Co
"42 (BC_7, AT1N, bidir, 0, " &
      "41, 0, Z), " & -- D1
"43 (BC_7, AT2N, bidir, 0, " &
      "41, 0, Z), " & -- D2
```

The determination of the safe values and purposes for the BC_7 usage on bits 42 and 43 are identical to the TBIC Control case.

The semantic checks for TBICN Control refer to section 6.4 in the 1149.4 standard documentation:

- If TBICN ports are defined in the **port description** then four boundary-scan cells shall be defined for the TBICN Control, as follows:
 - Calibrate cell, Ca (**internal**)
 - Control cell, Co (**control**), shall be listed in the **disable specification** for the TBICN Base Partition's D1 and D2
 - D1 cell, AT1N pin (**bidir**), **disable specification** shall be defined
 - D2 cell, AT2N pin (**bidir**), **disable specification** shall be defined

d. TBIC and TBICN Extension Controls

Inside an 1149.4 device, the analog busses may be subdivided such that sections of the bus may be enabled or disabled from connecting to the ATAP pins. These

sections are called partitions. They exist to allow better isolation for noise immunity, to lower bus capacitance, or other design and test reasons.

If there are additional internal test busses, TBIC and TBICN Extension Controls are needed. An example description of TBIC and TBICN Extension Controls is shown below:

```
-- following 2 cells are TBIC
-- Extension Controls
-- num cell port function safe
"36 (BC_1, *, internal, 0), " & -- D1
"37 (BC_1, *, internal, 0), " & -- D2
-- following 2 cells are TBIC
-- Extension Controls
-- num cell port function safe
"38 (BC_1, *, internal, 0), " & -- D1
"39 (BC_1, *, internal, 0), " & -- D2
-- following 2 cells are TBICN
-- Extension Controls
-- num cell port function safe
"44 (BC_1, *, internal, 0), " & -- D1
"45 (BC_1, *, internal, 0), " & -- D2
```

The semantic checks for TBIC and TBICN Extension Control descriptions are shown below, and refer to the requirements described in section 6.5 in the 1149.4 standard documentation:

- Boundary-scan cells must be specified for TBIC Extension Control, if it exists
 - D1 cell (**internal**)
 - D2 cell (**internal**)
- Boundary-scan cells must be specified for TBICN Extension Control, if it exists
 - D1 cell (**internal**)
 - D2 cell (**internal**)

e. Cells Controlling ABMs

The description of cells controlling an ABM for a single ended analog pin is shown below:

```
-- following 4 cells control the
-- analog signal W
-- num cell port function safe
--
--      [ccell disval rslt]
"23 (BC_1, *, control, 0), " & -- C
"22 (BC_7, W, bidir, 0, " &
      "23, 0, Z), " & -- D
"21 (BC_1, *, internal, 0), " & -- B1
"20 (BC_1, *, internal, 0), " & -- B2
```

The safe value of 0 in each cell will isolate the analog pin both from the core and from all the test circuitry, as defined in Table 6 and Table 7 in the 1149.4 standard documentation. Similar in fashion to the case for TBIC Control, a BC_7 cell is used for bit 22 to provide a control logic value to the D pin and to monitor the one-bit digital representation at pin W during simple interconnect test. The one-bit digital representation can be derived by a conceptual comparator in the ABM structure shown in Figure 2.

For differential analog pins, the description of cells controlling their associated ABMs is shown below:

```
-- following 4 cells control the analog
-- differential signal X1
-- num cell port function safe
--           [cell disval rs1t]
"15 (BC_1, *, control, 0), " & -- C
"14 (BC_7, X1, bidir, 0, " &
           "15, 0, Z), " & -- D
"13 (BC_1, *, internal, 0), " & -- B1
"12 (BC_1, *, internal, 0), " & -- B2

-- following 4 cells control the
-- analog differential signal X1N
-- num cell port function safe
--           [cell disval rs1t]
"11 (BC_1, *, control, 0), " & -- C
"10 (BC_7, X1N, bidir, 0, " &
           "11, 0, Z), " & -- D
"9 (BC_1, *, internal, 0), " & -- B1
"8 (BC_1, *, internal, 0), " & -- B2
```

The semantic checks for both single ended and differential pins are shown below:

- Boundary-scan cells shall be specified for analog I/O pins (single-ended and differential), as described in section 7.3.5 in the 1149.4 standard documentation
 - Control cell (**control**)
 - Data cell, analog I/O pin (**bidir**), **disable specification** shall be defined
 - B1 cell, AB1 (**internal**)
 - B2 cell, AB2 (**internal**)

Below are the semantic checks that apply to all boundary-scan registers:

- If there are no TBICN ports defined in the **port description** then there shall be no boundary-scan cells defined for TBICN Control
- Safe values shall be defined for all TBIC, TBICN, TBIC Extension, TBICN Extension, and ABM boundary-scan cells
- The disable result for all **bidir** cells shall be Z
- Cell assignments shall be unique; boundary-scan cells shall not be shared for multiple ABM or TBIC/TBICN Control or TBIC/TBICN Extension Control and shall not be used for multiple functions

3.2.11 ATAP Identification Statements

The ATAP pin identification statements are shown below:

```
-- ATAP port identification (1149.4)
attribute MST_AT1 of component_name : entity is
"AT1";
attribute MST_AT2 of component_name : entity is
"AT2";

-- ATAPN port identification
-- for differential ATAP (1149.4)
attribute MST_AT1N of component_name : entity is
"AT1N";
```

```
attribute MST_AT2N of component_name : entity is
"AT2N";
```

The semantic checks for the ATAP pin identification statements are:

- The ATAP ports, AT1 and AT2, shall be identified in the **MST_AT1** and **MST_AT2** attributes, respectively
- The pin type for the two ATAP ports shall be defined in the **port description** as **inout**
- For a differential ATAP, the additional ATAP ports, AT1N and AT2N, shall be identified in the **MST_AT1N** and **MST_AT2N** attributes, respectively; otherwise these attributes shall not be present
- The pin type for the two additional ATAP ports of a differential ATAP interface shall be defined in the **port description** as **inout**

3.2.12 TBIC Statements

TBIC statements specify the cells that have been assigned to control the TBIC. This includes each TBIC Partition as well. An example definition of a TBIC Partition is shown below:

```
-- TBIC register (1149.4)
attribute MST_TBIC of
component_name : entity is
-- Ca_num Co_num
"32, 33 : " &
-- {partition_name D1_num D2_num}
"IATB0 (34, 35), " &
"IATB1 (36, 37), " &
"IATB2 (38, 39)";
-- TBICN register (1149.4)
attribute MST_TBICN of
component_name : entity is
-- Ca_num Co_num
"40, 41 : " &
-- {Npartition_name D1_num D2_num}
"NIATB1 (42, 43), " &
"NIATB2 (44, 45)";
```

The cell numbers refer to the previous descriptions that relate to TBIC Controls described in sections 3.2.10(b), 3.2.10(c) and 3.2.10(d).

The semantic checks for TBIC statements are:

- The cell assignment for TBIC Control cells shall be specified in the **MST_TBIC** attribute
- The first cell in this attribute, **Ca_num**, shall be the calibrate cell for the TBIC; the cell shall be listed in the boundary-scan register as an **internal** cell
- The second cell in this attribute, **Co_num**, shall be the control cell for the TBIC; the cell shall be defined in the boundary-scan register with function **control** and shall be listed in the **disable specification** for the ATAP ports
- A **partition_name** for the TBIC Base Partition shall be specified with its two data cells, **D1_num** and

- **D2_num**, and the data cells shall be defined in the boundary-scan register
- TBIC Extensions are specified with a **partition_name** and the assigned data cells; the data cells shall be defined in the boundary-scan register; the listing order for multiple TBIC Extensions is not relevant
- If a differential ATAP is implemented in the device, the cell assignment for TBICN Control shall be specified in the **MST_TBICN** attribute, otherwise this attribute shall not be present
- The first cell in this attribute, **Ca_num**, shall be the calibrate cell for the TBICN; the cell shall be listed in the boundary-scan register as an **internal** cell
- The second cell in this attribute, **Co_num**, shall be the control cell for the TBICN; the cell shall be defined in the boundary-scan register with function **control** and shall be listed in the **disable specification** for the additional ATAP ports
- An **Npartition_name** for the TBICN Base Partition shall be specified with its two data cells, **D1_num** and **D2_num**, and the data cells shall be defined in the boundary-scan register
- TBICN Extensions are specified with an **Npartition_name** and the assigned data cells; the data cells shall be defined in the boundary-scan register; the listing order for multiple TBICN Extensions is not relevant

3.2.13 ABM Statement

The ABMs with their associated control cells, pins and TBIC Partitions are specified in the ABM statement. An example ABM statement is shown below:

```
attribute MST_AB_M Pins of
component_name : entity is
-- ABMs (1149.4)
-- port partition_name
--
--      C   D   B1  B2
"W ( IATB0: 23, 22, 21, 20 ), " &
"Y ( IATB0: 19, 18, 17, 16 ), " &
"X1 ( IATB1: 15, 14, 13, 12 ), " &
"X1N ( NIATB1: 11, 10, 9, 8 ), " &
"X2 ( IATB2: 3, 2, 1, 0 ), " &
"X2N ( NIATB2: 7, 6, 5, 4 );
```

The cell numbers refer to the previous descriptions on cells controlling an ABM, described in section 3.2.10 (e). The ABM for port W is using the TBIC Partition IATB0. The TBIC Partition IATB0 is defined in section 3.2.12.

The semantic checks for the ABM statement are:

- Every ABM shall be listed in the **MST_AB_M_Pins** attribute
- Each ABM shall be listed with its port identifier, as defined in the **port description (port attribute)**, followed by the TBIC Partition it is connected to and the four boundary-scan cells assigned to the ABM control register

- The TBIC Partition name shall be defined in the **MST_TBICN** attribute
- If the ABM is connected to a TBICN Partition, then that partition name shall be defined in the **MST_TBICN** attribute
- The order of the boundary-scan cell assignment in the ABM pin table is the following: Control cell (**C_num**), Data cell (**D_num**), B1 cell (**B1_num**), and B2 cell (**B2_num**)

Note: The bit order in this attribute refers to the bit order in Table 8 in the 1149.4 standard documentation.

- The boundary-scan cell assigned to **C_num** shall be defined in **BOUNDARY_REGISTER** as a **control** cell
- The boundary-scan cell assigned to **D_num** shall be defined in **BOUNDARY_REGISTER** as a **bidir** cell and shall have a **disable specification** which specifies **C_num** as its control cell
- The boundary-scan cell assigned to **B1_num** shall be defined in **BOUNDARY_REGISTER** as an **internal** cell
- The boundary-scan cell assigned to **B2_num** shall be defined in **BOUNDARY_REGISTER** as an **internal** cell

3.2.14 Differential Pins Statement

The differential pin statement specifies the differential pair, polarity, and associated single ended connection point. An example of a differential pin statement is shown below:

```
attribute MST_Diff_Pins of
component_name : entity is
-- DBMs at single-ended side
-- of differential drivers/receivers
--
-- representative_port
--      associated_port
--      num
"D1 : D1N (28), " &
"D2 : D2N (31)";
```

In this example, the pin order defines the polarity. For example, D1 is positive and D1N is negative. Cell 28 is the single ended resolution point of the differential pins D1 and D1N.

The semantic checks for the differential pin statement are:

- Boundary-scan data cell numbers for DBMs on the single-ended side of differential drivers/receivers shall be assigned to the respective differential port pair defined in the **MST_DIFF_PINS** attribute
- First, the representative port identifier shall be listed, then the associated port identifier, followed by the boundary-scan cell number

- The two port identifiers shall be defined in the **port description** (port attribute)
- The DBM's boundary-scan cell shall be defined in the **BOUNDARY_REGISTER** attribute

3.3 Summary

In Section 3.2, the components of ABSDL have been described in detail. This includes attribute references, pin definitions, recommended cell settings for controlling TBIC circuitry and ABMs, and also the interconnection between cells and test circuitry.

4 Application Examples

Table 2 shows the values that need to be assigned to the relevant TBIC and ABM control cells based on the test structure in Figure 4 to perform the following applications:

- Application I - Applying stimulus to port W using AT1 and monitoring the signal at port W using AT2.
- Application II - Connecting port W to VG and monitoring the value using AT2.
- Application III - Applying differential signals to X1 and X1N. AT1 is connected to X1 and AT1N is connected to X1N.

The cells' data in each application, above, refers to Table 1 and Table 2 for the TBIC and Table 6 and Table 8 for the ABM, in the 1149.4 standard documentation. The information in the ABSDL file described in Section 3, above, should be able to provide appropriate data to control the TBIC and ABM for test applications.

TBIC/ABM	Code Bit	Cell No.	Applications		
			I	II	III
TBIC	Ca	32	0	0	0
	Co	33	0	0	0
IATB0	D1	34	1	0	0
	D2	35	1	1	0
IATB1	D1	36	0	0	1
	D2	37	0	0	0
TBICN	Ca	40	0	0	0
	Co	41	0	0	0
NIATB1	D1	42	0	0	1
	D2	43	0	0	0
ABM Port W	C	23	0	0	0
	D	22	0	1	0
	B1	21	1	0	0
	B2	20	1	1	0
ABM Port X1	C	15	0	0	0
	D	14	0	0	0
	B1	13	0	0	1
	B2	12	0	0	0
ABM Port X1N	C	11	0	0	0
	D	10	0	0	0
	B1	9	0	0	1
	B2	8	0	0	0

Table 2: Application examples

5 Future Plan

In order to ensure that ABSDL is capable of assisting EDA tools in generating test patterns for mixed-signal interconnect test, several verifications of its features need to be carried out. These include applying the ABSDL file to EDA tools, generating test patterns and applying the patterns using real mixed-signal devices.

As it stands today, the proposal covers enough content to complete simple wired interconnect testing. More content and syntax needs to be developed to handle analog component measurement. The Working Group has begun this task, but decided to scale back its efforts and stage the introduction of ABSDL.

6 Conclusion

A language that can describe the boundary-scan implementation in a mixed-signal device has been developed. The current ABSDL features should be able to provide adequate information for simple and extended interconnect tests in a mixed-signal board.

7 Acknowledgments

We would like to thank Keith Lofstrom (KLIC), Ken Parker (Agilent Tech.), Pete Collins (ASTER Ingenierie), Anthony Sparks (JTAG Tech.), Karla May (Corelis), Jake Karrfalt (ASC), Rohit Kapur (Synopsys) and Wim Driessen (JTAG Tech.) and the rest of the IEEE Std 1149.4 Working Group for their excellent contribution and participation in ABSDL development.

8 References

- [1] IEEE Standard Test Access Port and Boundary-Scan Architecture (IEEE Std. 1149.1 – 2001), Institute of Electrical and Electronics Engineers, Inc., July, 2001.
- [2] IEEE Standard for Mixed-Signal Test Bus (IEEE Std. 1149.4 – 1999), Institute of Electrical and Electronics Engineers, Inc., March, 2000.
- [3] <http://grouper.ieee.org/groups/1149/4/>, IEEE 1149.4 Working Group Web page, ABSDL Effort.
- [4] A. Cron, "IEEE P1149.4-almost a standard", Proceedings International Test Conference, pp 174-182, Nov 1997.
- [5] K. P. Parker, J. E. McDermid, S. Oresjo, "Structure and Metrology for an Analog Testability Bus", Proceedings, International Test Conference, pp 309-322, Oct 1993.
- [6] <http://www.national.com/an/AN/AN-1200.pdf>, "Mixed Signal Testing Using the IEEE 1149.4 STA400", Application Note 1200, June 2004.
- [7] S. Sunter, K. Filliter, W. Joe, P. McHugh, "A general purpose 1149.4 IC with HF analog test capabilities", Proceedings International Test Conference, pp 38-45, Oct 2001.
- [8] <http://www.national.com/pf/ST/STA400EP.html>, National Semiconductor STA400EP IEEE 1149.4 Analog Test Access Device Product Folder