

Extending the Digital Core-Based Test Methodology to Support Mixed-Signal

Geert Seuren, Tom Waayers
Philips Research, Electronic Design & Tools
Prof. Holstlaan 4, NL-5656 AA Eindhoven
The Netherlands
Geert.Seuren@philips.com, Tom.Waayers@philips.com

Abstract

This paper presents an extension to a digital core-based test architecture to support testing of mixed-signal cores in a system-on-chip. It also presents a new mixed-signal test development flow that comprises a test library based approach to ease mixed-signal test development. The new flow was realized and experiments show clear advantages.

1 Introduction

As the manufacturing and process technology evolve very rapidly the design and test of modern integrated circuits is a challenging task. These new technologies enable a high level of integration making it possible that complete systems are integrated into one single chip whereas until recently these systems consisted out of several chips on a board. Typical examples of these so-called systems on a chip contain a mix of digital, memory and analogue/mixed-signal cores. Due to high time-to-market demands nowadays design of systems on a chip is often done by integrating several cores. These cores are often already pre-defined and pre-verified modules.

Though it has many advantages to include digital logic, memory and even analogue into one single chip, it has a huge impact on the testing complexity of these chips. The methods for testing analogue and digital are quite different and even the test methods of memories and digital logic differ. For memories march test and BIST methodologies are commonplace. Testing of digital logic already has a long history in scan- and Iddq-testing and also logic-BIST is becoming more popular. Furthermore there are a lot of EDA tools on the market to support test and verification of digital circuits and memories. Analogue on the other hand still heavily relies on the traditional methods of functional testing as was laid down in [1].

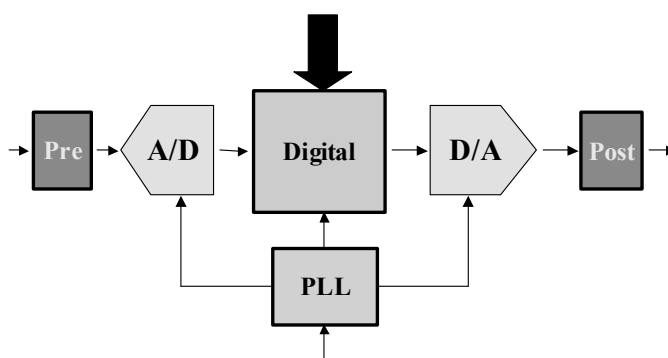


Figure 1 Content of modern SOC

Modern SOC's in the area of telecommunications, computer peripherals and consumer electronics can be classified as 'Big D – Small A'. This means one or more digital core's (MIPS, DSP, Memory) surrounded by ADC's and DAC's for analogue signal conversion, see Figure 1. Clock generation in such SOC's is done by one or more PLLs. In some cases ADCs are preceded and/or DACs are followed by signal conditioning stages. These stages are often analogue circuits, whereas the ADC and DAC are real mixed signal circuits.

Testing of analogue modules is often done implicitly by testing the whole signal path, from the analogue input up to the converter or visa versa. This does require that embedded digital terminals are accessible during these tests. To enable structured mixed-signal test, a DfT solution has been developed [2] that standardizes the access to the embedded digital terminals of mixed-signal cores. This standard approach enables the set-up of a standard flow for mixed-signal DfT verification and test vector generation, which is the subject of this paper.

The sequel of this paper is organized as follows. Section 2 gives a overview of prior work done in the domain of core-based testing and mixed-signal test development. Requirements for extending the digital core-based test

methodology to support mixed-signal are given in section 3. In Section 4 the mixed-signal extension to a digital core-based test architecture is shown. Section 5 discusses tool support for DfT generation and verification. In Section 6 the new mixed-signal test development flow is presented. Experimental results of the new flow are subject matter of Section 7 and Section 8 concludes this paper.

2 Prior work

Following the core-based design philosophy a core-based test strategy is a logical way of dealing with test. The IEEE P1500 Working Group [3] is defining a standard for embedded core test. This standard defines an architecture fulfilling the test access needs for digital cores. However, the requirements for mixed-signal are different which means that an extension is needed.

In [4] the authors describe a general purpose DfT methodology at the analogue-digital boundary. It describes a standard access architecture for embedded m/s building blocks, especially targeting AD-DA converters. However, this architecture does not fully support core-based test because it is not possible to test a building block in isolation and interconnection test cannot be performed. Others [5] come with proposals to extend the P1500 standard with analogue DfT alike IEEE 1149.4 standard.

When confining to the 'Big D- Small A' type of SOC's the analogue signals are already at the SOC boundary, making it justifiable to restrict the DfT architecture to the digital terminals of a mixed-signal module. The test requirements for the digital terminals of a mixed-signal core, however, are different than for a digital core. Therefore an extension of a digital core-based test standard was developed [2] to enable mixed-signal core-based test.

In [11] the authors show that significant improvements in time to market can be achieved by using a virtual test (VT) environment to simulate the whole mixed-signal test before tape-out. Besides a coupling to the test program the VT approach is very much dependent on the availability and correctness of device-under-test (DUT), test interface board and tester instrument models. These models are made in VHDL language. Despite being a digital description language it proves to be sufficient for modelling the mixed-signal blocks. Only test related properties need to be modelled, which implies that for mixed-signal test simulation we only need to have an abstraction of the access mechanism to the mixed-signal core and a model of the mixed-signal core itself.

Within our company we follow a core based test strategy already for quite some years now[6][7]. A test flow based on test (protocol) expansion [8][9] has been established and used on many designs. Test patterns delivered with or developed for a core can be expanded to a next (chip) level

in the SOC's hierarchy. The necessary test control is also expanded as is described in [10]. Advantage of this approach is that in an early stage of the design flow the DfT can be verified without simulation of design validation patterns.

We propose to extend the capabilities of the expansion tool to expand hardware descriptions from core level to higher and in the end chip level. By means of this expansion both the digital and mixed-signal DfT hardware is verified resulting in a correct description of the complete chip-level DfT architecture. The chip-level DfT hardware view generated during expansion nicely fits the need of Virtual Test and is used in the next steps of the test development: test bench and test pattern generation.

3 Requirements

Based on the above and experiences in testing analogue we derive requirements for the extension of the digital core-based test methodology towards the mixed-signal domain.

- Support core internal scan test as well as interconnect test for both digital and mixed-signal cores.
- Fit to digital core based test strategy.
- Easy integration into existing test standards (IEEE 1500, IEEE 1149.1).
- Enable analogue specification-based measurements.
- Enable re-use of test hardware and test programs.
- Enable behavioural simulation of tests.
- Enable flexible, user-driven and tester independent test pattern generation for mixed-signal cores.
- Test vector description must be human readable.

4 DfT architecture

Due to the additional requirements for mixed-signal testing a new access and control mechanism is developed which supplements the digital approach. In fact it is an extension to the digital core based test architecture. Alike that approach a mixed-signal core has a wrapper called TestShell that contains the DfT hardware.

As can be seen in Figure 2 the architecture only provides test access and control to the digital signals and not to the analogue signals. There are two reasons for this: firstly, as discussed in section 2, in a lot of m/s chips the analogue signals to be measured are already at the chip-level pins (e.g. output of a DAC). Secondly, the analogue blocks are in case of 'Big-D Small-A' SOC's not tested separately.

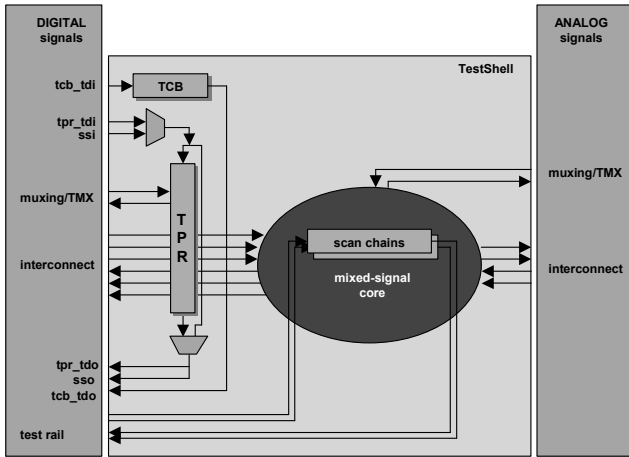


Figure 2 Mixed-signal DfT architecture

The complete analogue function is tested at once making access to terminals in between not necessary. In case embedded analogue access is required elements of the IEEE 1149.4 [12] can be considered to solve this need. The latter fits very well with the architecture described in this paper because it is also tailored towards IEEE 1149.1 JTAG control.

4.1 Test Point Register (TPR)

To be able to cope with the analogue specification based measurements of a mixed-signal core two types of test data are considered:

Static:

This is data that occasionally changes during a test and has therefore low speed requirements, which means that the test data can be transported indirect by means of a scan chain like architecture. Often settling time of analogue blocks is several times longer than a clock

period of the corresponding digital control part, which fits the low speed requirements of the test data.

Dynamic:

This is data that switches often during a test and/or needs precise timing and therefore needs to be accessed directly from chip-level pins. This can be achieved by means of (chip-level) multiplexing.

Besides that during testing the mixed-signal core must be in isolation of it's surrounding, it must be possible to perform chip level interconnection test.

The necessary DfT enabling all above mentioned functionality is implemented in the so-called Test Point Register (TPR). For each core input/output a DfT cell, a so-called Test Point (TP) slice, is implemented, which together form a chain as can be seen from Figure 3.

The TPR supports multiple configurations:

- Static:**
 The TPR is configured as a shift and update register providing serial access to the digital input and output terminals of the mixed-signal core through tpr_tdo / tpr_tdi. All TP slices form one chain.
- Dynamic:**
 The TPR is configured for dynamic access, enabling direct connection to the dynamic input and output terminals of the mixed signal core through extra terminals at the dynamic slices, see Figure 3. These dynamic TP slices are left out of the chain formed by the TP slices via the tpr_dyn_en control signal. This gives more efficient access to terminals controlled by static TP slices. Access to the chain is through tpr_tdi

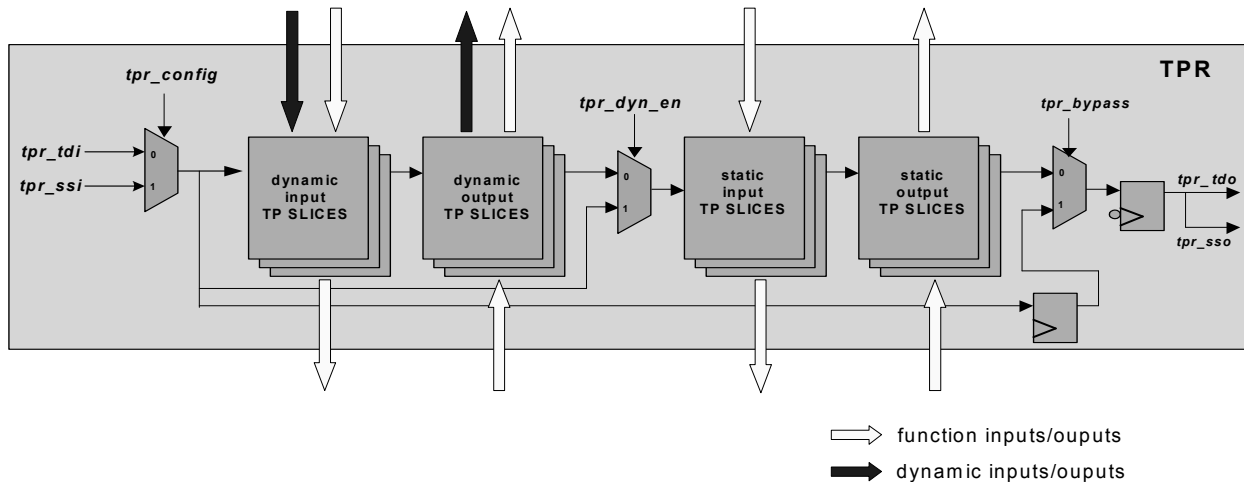


Figure 3

Test Point Register

and tpr_tdo. It should be noted that only those terminals that need dynamic access are provided with dynamic TP slices.

- Scan-test:**
 In this configuration the TP slices form one scan-chain connected between input tpr_ssi and output tpr_sso, enabling standard digital ATPG to test the digital.
- Bypass:**
 A bypass of the TPR is provided to allow optimization of test time when the TPR is chained to other TPRs after integrating several mixed-signal cores.

Both the static and dynamic configuration are used for mixed-signal test purposes; the TPR is accessible through tpr_tdi/tpr_tdo, which are connected at IC-level to an 1149.1 TAP controller, as will be discussed in section 4.4. The scan-test configuration supports interconnection test and the core's internal digital test. In this case the TPR acts as a wrapper and is connected to the chip-level scan chains (TestRail) through tpr_ssi / tpr_sso. When the mixed-signal core itself contains scan-chains they are connected to the chip level TestRail as well. In this way the requirement to fit to the digital DfT approach is fulfilled and the existing digital tool flow can cope with TPRs for digital test development.

4.2 Direct access

Dynamic test data needs be provided directly from chip-level pins, because analogue tests often are specification-based measurements that require high-speed access or precise timing. To enable this the TPR has to be set into the dynamic configuration providing direct access to the core terminals, meanwhile isolating it from the surrounding. It should be noted that only those core terminals that really need direct access are supplied with it and all other terminals continue to have static access in this dynamic configuration. The core dynamic terminals need to be multiplexed to chip-level pins during the chip-level dynamic test, this has to be done by the chip designer during integration of the cores.

4.3 TPR controlled by a TCB

A so-called Test Control Block (TCB), the standard control mechanism for the digital cores, determines the configuration of the TPR. Like a P1500 WIR, a TCB is a shift and update kind of register in which control bits can be loaded serially through tcb_tdi and tcb_tdo (see Figure 4). The core-level and chip-level TCBs are configured as chains that are addressed by 1149.1 TAP (JTAG) controller instructions. Details on the implementation can be found in [10].

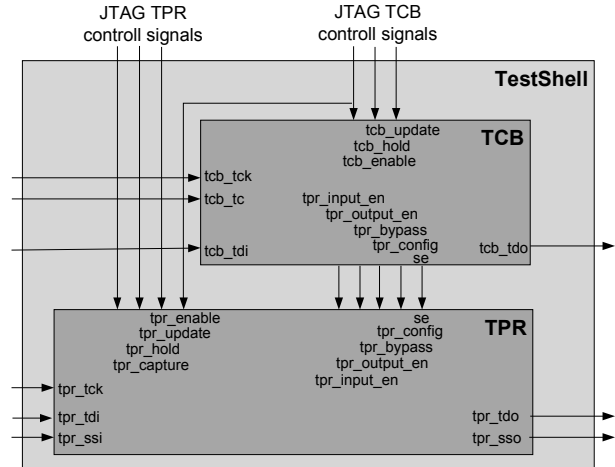


Figure 4 TCB controlling TPR

Next to setting the configuration of the TPR the TCB is used to set possible test control signals of the mixed-signal core itself. The TCB test operation and control, is done by an 1149.1 TAP controller, similar to the TPR.

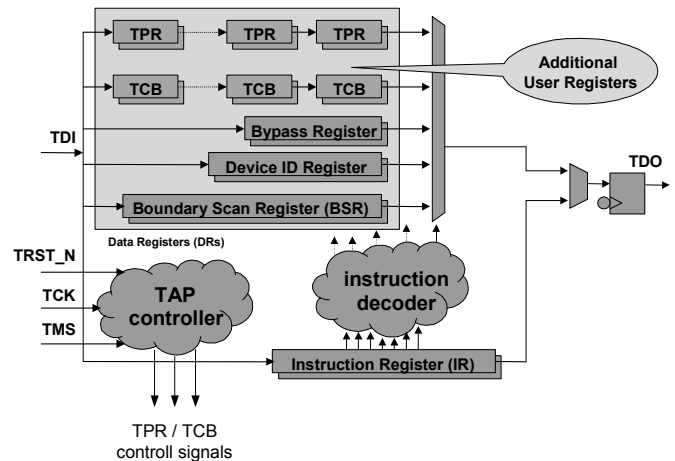


Figure 5 TCB and TPR integration into 1149.1 architecture.

4.4 Integration into 1149.1 Boundary Scan architecture

An 1149.1-TAP controller is used to control the chip-level test operation. As there is a lot of similarity between a TCB and a TPR the control is quite the same. All TCBs are chained together and connected to the TAP controller to form one or more so-called user register(s) in the 1149.1 architecture, as can be seen in Figure 5. The same holds for the TPRs: all core level TPRs are chained together through the tpr_tdi and tpr_tdo signals. The TCB and TPR registers are addressed by IEEE-1149.1 User-Defined instructions.

5 DfT generation and verification

The DfT architecture presented in section 4 is an extended version of the test control architecture presented in [10]. From a IEEE-1149.1 point of view, only User Registers have been added to control and observe embedded mixed signal blocks. The most important property of these user registers, implemented by a concatenation of TPRs, is the adaptive register length of configurations that depend on the controlling TCB register contents. Since the latter is also a User Register, a two layer or parent-child approach is introduced for test development using the standard IEEE-1149.1 protocol. For testing digital logic using internal scan chains we already saw a similar approach. Test modes must be initialized prior to testing. Before testing a mixed signal module using its TPR, core and chip level TCBs must be loaded with the appropriate test modes to set the required environmental conditions for the module under test and to configure efficient access.

In core-based design-for-test flows we want to benefit from the fact that for both test data access and test control, standardized structures are used. To efficiently make use of a test control architecture in different design stages, we already identified that tool support is needed to:

- *Implement* the architecture: Standard structures like TPR, TCB and TAP controller can be generated and inserted in core and chip level modules. This results in a faster and less error-prone design process.
- *Validate* the architecture: Check the architecture by exploring a design netlist. This enables outgoing and incoming inspection during core and chip design.
- *Use* the architecture: Support the test architecture at higher abstraction level, e.g. group test control bits of TCBs and TPRs in easy to reference test modes and configurations. This results in a faster and less error-prone test development and diagnosis process.

Implementing standard structures like TPR, TCB and TAP controller can easily be done by tools. In our core based design flow, the core packager sub-flow starts with the tool that generates standard TCB and TPR hardware. Besides the hardware, the tool also generates output in our own test data format to store abstractions that will be used for both DfT architecture validation and test development. Multiple TCB and TPRs can be generated and inserted in a core netlist. The resulting core netlist and all test data files are input to the architecture validation process that makes use of expansion [10]. If no errors found, all abstractions are expanded/translated to core level and a core DfT view is generated. The test control section of this view describes core level initialization protocols that cover all TCBs embedded in the core.

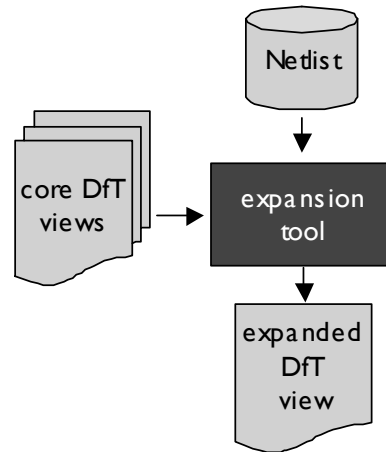


Figure 6 DfT verification flow

To support validation of a DfT architecture that uses TPRs and to enable mixed/signal test development near to a tester (without using a netlist) the proprietary test data format needs to be extended with a TPR construct. During analysis of the validation and test development processes the following requirements were found for the TPR construct:

- TPR interface: configuration, control, data, direct access and clock terminals.
- TPR configurations: static, dynamic, bypass:
 - Mode or signal conditions enabling the configuration
 - Positions of test points / analog module terminals in the resulting register
- TPR terminal conditions controlling the TPR modes of operation (capture/shift/hold/update).
- The order of TPRs in their chain.

Note that the scantest configuration is not part of the TPR description. In scantest configuration, the sequential elements are part of a standard scan chain description as used for digital test development.

Building a mixed-signal SOC is done using our core integrator flow that starts with the tool that generates standard IEEE 1149.1 hardware. The chip level test control architecture is defined using this tool, i.e the amount of TCB chains, TPR chains and TAP instructions that address the chains. This tool also generates output in our own test data format besides the hardware output. The chip netlist after insertion of cores and 1149.1 hardware, the TAP controller view and all core DfT views are again input to the architecture validation process. If no errors found, an expanded DfT view at chip level is generated, see Figure 6. This view defines the chip level initialization protocol and

opcodes for chip test modes that are translated to initialization vectors during test assembly. It also defines TPRs with initialization dependent configurations, standard settings and TPR chains, linking TPRs to positions in user registers.

6 Test development

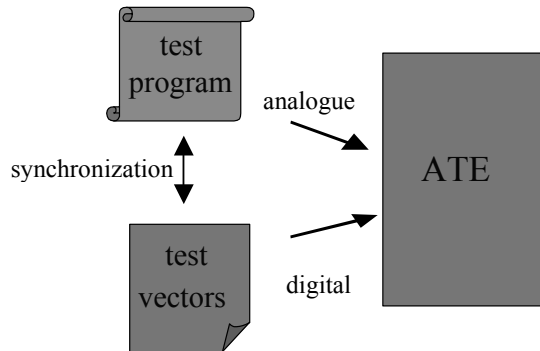


Figure 7 Mixed-signal stimulus/response

Specification based testing still is the dominant method for testing a mixed-signal device. In fact a mixed-signal test is a measurement of a parameter (e.g. signal to noise ratio, SNR) followed by the test, which is comparing the measured parameter against a limit [1]. The measurement is a procedure in which stimulus is applied to the device and the response is captured. The captured data is analysed in a post-processing stage to extract the parameter that we want to test for. This stimulus and response can have the form of analogue and/or digital. In case of analogue stimuli the test program controls the instrument(s) that drive or capture the analogue signals. The digital stimuli, referred to as test vectors (Figure 7), are driven by the digital subsystem of the tester.

The synchronization between the test program and the test vectors plays an important role in the measurement. On most mixed-signal testers the test vectors can contain instruction op-codes to perform basic control, like start and stop, of the analogue instruments in the test system. In this way the test in terms of test time can be optimised because the analogue instruments are directly controlled via the op-codes of the test vectors. The communication between the vectors and the test program and the development of the test program itself is beyond the scope of this paper. We will concentrate on generation and verification of the test vectors. The test vectors contain the digital control and test data for the mixed-signal measurement.

The kind of data in the test vectors is dependent on the measurement to be performed. This type of test vector development is mainly user driven, this in contrast to fault model based automatic test vector generation (ATPG) in

the digital domain. It is thus for mixed-signal test vector generation very important that the user can easily define test vectors.

In our case the existing digital test development flow is build around a test assembler that is controlled by a test description language. This propriety test description language (TDL) is in use already for years in the digital test development flow to generate test vectors in a tester independent way. The test assembler translates the TDL description into tester specific test vectors by using an extensive library of tester drivers. Next to that the assembler has drivers to generate test benches in either Verilog or VHDL language for simulation purposes. The test bench in fact is a generic model of a digital tester.

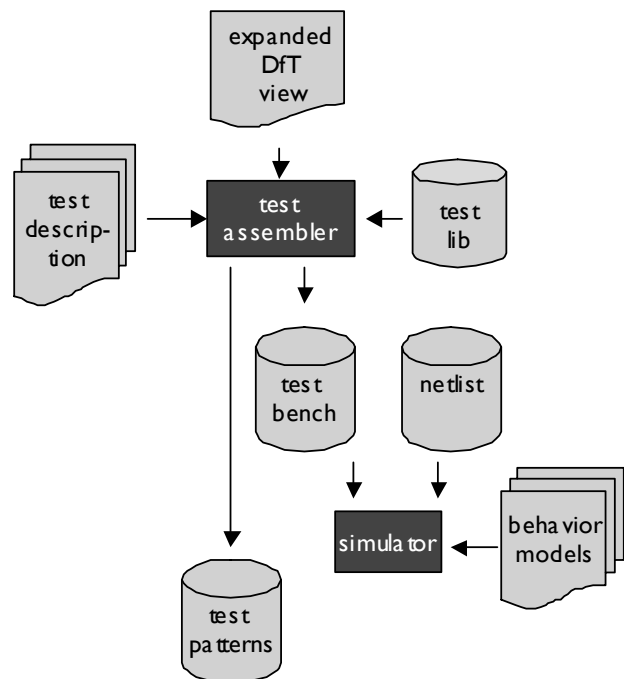


Figure 8 Test development flow

This test assembler controlled by the description language forms the basis of the new mixed-signal test development flow, which is shown in Figure 8. Starting point is the expanded DfT view, generated during the validation step discussed in section 5. This makes it possible to describe the mixed-signal test vectors at core level as if the core is tested stand-alone. Due to the fact that the expanded DfT view is the result of a validation step it is correct by construction. As a result the user can concentrate on the actual measurement to be performed. The assembly tool translates the core test to the higher (chip) level terminals using the expanded DfT view only. This makes the core

test description re-usable, because only the expanded DfT view needs to be re-generated for the next chip.

As discussed in section 4.4 the mixed-signal test to be developed is mapped to the 1149.1 TAP controller. Therefore the sequence of operation during chip level test is as follows:

1. Put the chip in test mode by initiating the IEEE 1149.1 JTAG controller.
2. Select the instruction that puts the TCB chain in between TDI and TDO (PROGRAM_TCB).
3. Initialize the test environment by programming the TCB chain with the appropriate settings. The TPRs, clocks, muxing, power down etc. will be set into the right configuration.
4. Select the instruction which puts the TPR chain in between TDI and TDO (PROGRAM_TPR_AMS)
5. Provide the test data by loading the TPR chain with the appropriate test data. Simultaneously the TPR captures data on the core outputs.
6. The mixed-signal test can be carried out.
7. Step 5 and 6 can be repeated until the test is finished.
8. Next test starts when new TCB data is loaded.

The above is a sequence of standard procedures and therefore made available as predefined IEEE 1149.1 control functions in the test library. Next to these there are standard functions for generating waveforms like sine waves, bit-streams and multi-tones. The use of the functions is best illustrated by an example of a test vector description for an embedded 10 bits digital to analogue converter (DAC), which comprises a 5 bits gain-stage:

```

1 reset_jtag;
2 set_tcb_mode( 'dac_tcb', 'dynamic');
3 jtag_program_tcb;
4 set_tpr('dac_tpr');
5 set_tp('gain[0]')=&1;
6 jtag_program_tpr;
7 drive_sine_dynamic('da_in',M,N);

```

The commands *reset_jtag*, *jtag_program_tcb* and *jtag_program_tpr* in line 1, 3 and 6 are functions controlling the 1149.1-TAP controller. The first resets the controller and brings the chip into test mode. The second selects the TCB register as user register and shifts the defined mode into the register. The bit-values of the test mode are stored in the expanded DfT view. The last command shifts data into the TPR, based on the values set in the lines 4 and 5. These two 'set' functions select the TPR (line 4) and set the specified TPR test-point 'gain[0]' to logical value '1' (line5). All other bit values are in a default state, which can be set up-front (not shown in this

example). The *set_tcb_mode* in line 2 sets the TPR of the DAC in dynamic configuration via the TCB 'dac_tcb', all other TCB are in a default state. Because the DAC is in a dynamic configuration the data input bus 'da_in' is available at chip-level. Via this bus a digitized sine wave is driven into the DAC through the command *drive_sine_dynamic*, which is also available in the test library. The digitized sine wave is parameterized with M (number of sine cycles) and N (number of vectors).

In the background the test assembler will perform several checks to prevent the user of making mistakes like e.g. driving on the 'da_in' bus while it is not initialized.



Figure 9 Example simulation result

As one can see the test pattern description is written at core-level. The test assembler will retrieve the chip-level information from the expanded DfT view. The assembler will generate a test bench either in Verilog or VHDL. Despite the fact that these are digital description languages they can perfectly be used for high-level mixed-signal behaviour modelling. These models are delivered with the mixed-signal core, developed by the designer of the module. The analogue signals are represented by REAL variables. This way within a simulation waveform viewer the analogue signals can be viewed. When in above example the lines 5, 6, 7 are repeated for all gain bits of the DAC the simulation result will be as shown in Figure 9. For this DAC example one can visually check the expected functional behavior of the DAC by looking at the sine wave:

- Due to the setting of the 5 gain bits it must grow in amplitude.
- If the data inputs are connected wrong the waveform would not look like a sine wave, if some bits are connected wrong it immediately shows up.

As shown the test vector description can be developed at a high abstraction level. The pattern description can easily be parameterized to build a template. This template can be stored into a library making it possible to automate standard measurement procedures.

7 Experimental results

The new test development flow has been used on a design project. The design is a digital base-band processor and controlling IC for mobile communications. The IC is a typical ‘Big-D Small-A’ chip: the digital part consisting of a few big digital cores like system controller, DSP and several memories. The mixed-signal part consists of 3 PLLs and one oscillator for clock generation. Chip level test control is done with the standard IEEE 1149.1 Boundary Scan controller. The digital part of the design is equipped with the standard core based DfT.

The predecessor of this design did not have a structured mixed-signal DfT approach and mixed-signal test development flow. It was done partially manual, partially with the existing digital tool flow. The result is a manual and error prone process of mixed-signal test development. Next to that re-use of test is not possible. To improve this the DfT architecture discussed in this paper has been utilized and the mixed-signal flow has been set-up and used. Through expansion the DfT architecture is analysed before test development takes place, very helpful when debugging a test: one can concentrate on the test instead of checking whether the DfT is correct. The expansion step results in a chip-level DfT view. The big advantage of the automatic view generation is that the information it contains is the result of analyzing the design netlist and is correct. Compare this to the situation that one has to get this information by manually going through the netlist or get the information from the designer.

The next step was test development, which consisted of automated test bench generation and test vector development. By using behavioral models of the mixed-signal cores in the test bench the developed test vectors can be simulated at a high abstraction level, which enables verification of the DfT. After tape-out the test vectors for production test were developed and verified on a production tester. This resulted in a failing PLL test on the tester. The PLL did start-up in the opposite phase then simulated with the behavioral model. After changing the phase in the test description and re-generating the (expected) test vectors the test passed. The failure is due to limitations in the behavioral model, which is not strange because exact timing and start-up behaviour of the PLL is difficult if not impossible to model. In general behavioral models are not suited for mixed-signal measurement verification. This still has to be done on a tester.

In the end due to the use of standard functions and a correct by construction access description in the expanded view, the new DfT approach and test development flow realized a significant reduction in mixed-signal test development time for this chip, roughly from a week to a day.

Another aspect of the new flow is re-use. This was experienced on a next version of this chip for which the analogue part of the previous version of the chip was copied. The already developed test descriptions could be re-used. By re-running the flow, within half a day, the newly generated chip level test vectors on this new design could be verified.

8 Conclusion

In this paper we presented a new mixed-signal test development flow and a standard DfT architecture for mixed-signal cores that is supported by this flow. Key elements of the flow are DfT architecture validation and DfT view generation by means of expansion and test development through a library based approach. Practical experiments using the flow in a real silicon project clearly show the effectiveness of the approach. Reduced time was spend on debugging and developing mixed-signal tests for a complex mixed-signal system-on-chip and its derivate.

9 Acknowledgements

The authors would like to thank the members of the Philips CTAG.AMS work group for the definition of the mixed-signal design for test standard and test development flow. Also thanks to our colleagues Frank van der Heyden, Mario Konijnenburg and Krijn Kuiper for their fruitful discussions.

This work was partially funded through the European MEDEA+ program.

10 References

- [1] M. Mahoney. DSP-based Testing of Analog and Mixed-Signal Integrated Circuits. IEEE Computer Society Press, ISBN 0-8186-0785-8, 1987.
- [2] G. Seuren, C. Feige, “Extending core test methodology to the analog/mixed-signal domain”, Informal Proceedings European Test Workshop, Stockholm 2001.
- [3] IEEE P1500, “Standard for Embedded Core Test (SECT)”, homepage, <http://grouper.ieee.org/groups/1500/>.
- [4] J. Verfaillie, D. Haspelslagh, “A General Purpose Design-for-Test Methodology at the Analog-Digital Boundary of Mixed-Signal VLSI”, Journal of Electronic Testing: Theory and Application, Vol 9, 1996, pp 109-115.

- [5] M. Stancic, L.Fang, H.G. Kerkhoff, "Mixed Signal Core Based Testing", Proceedings of European Test Workshop (ETW), Cascais, pp. 279-280, 2000
- [6] Beenker et al., "A testability strategy for silicon compilers", Test Conference, 1989. Proceedings. 'Meeting the Tests of Time'. International, Page(s): 660 -669, 29-31 Aug 1989.
- [7] Marinissen et al., "A Structured and Scalable Mechanism for Test Access to Embedded Reusable Cores", Proceedings International Test Conference, IEEE Computer Society Press, Washington, DC, pp. 284-293, October 1998.
- [8] Marinissen et al., "Testability and Test Protocol expansion in hierarchical macro testing", European Test Conference, Rotterdam, The Netherlands, 1993, pp. 28-36
- [9] Marinissen, Lousberg, "The Role of Test Protocols in Testing Embedded-Core-Based System Ics", European Test Workshop, Constance, May 25-28 1999
- [10] T. Waayers, "An improved Test Control Architecture and Test Control Expansion for Core-Based System Chips", Proceedings International Test Conference, IEEE Computer Society Press, Charlotte, 2003, pp 1145-1154.
- [11] G. Krampfl, M. Rona, H. Tauber, "Test Setup Simulation- a High Performance VHDL-based Virtual Test Solution Meeting Industrial Requirements", Proceedings of International Test Conference, Baltimore, pp 870-878, October 2002.
- [12] IEEE std 1149.4-1999, 'IEEE Standard for a Mixed Signal Test Bus', IEEE Computer Society, March 2000.