



SIEMENS

Ingenuity for life

Siemens Digital Industries Software

Tessent IJTAG

Automated test creation for mixed-signal IP

Executive summary

The creation of test patterns for mixed signal IP has been, to a large extent, a manual effort. The IEEE 1687-2014 standard, also called IJTAG (internal JTAG), was defined to improve the process used to test, access, and control embedded IP. IEEE 1687 enables the industry to develop test patterns for IPs on the IP level without having to know how the IP will be embedded within different designs.

Siemens EDA and NXP Semiconductors (NXP) worked together to implement 1687 on mixed-signal IPs in a 65 nm automotive design. The results demonstrate the significant advantages of 1687 over the IEEE 1149.1-2001 (JTAG) test methodology, both in automating test pattern development and in reducing test setup data volume by more than 50%.

Martin Keim and Fredrich Hapke, Siemens EDA
Tom Waayers and Richard Morren, NXP Semiconductors

Introduction

The creation of test patterns for mixed signal IP has been, to a large extent, a manual effort. To improve the process used to test, access, and control embedded IP, the IEEE 1687-2014 standard¹ was defined by a broad coalition of IP vendors, IP users, major ATE companies, and all three major EDA vendors. This standard, also called IJTAG (internal JTAG), is rapidly and widely adopted by the semiconductor industry.^{2, 3, 4}

The 1687 standard enables the industry to develop test patterns for IPs on the IP level without having to know how the IP will be embedded within different designs. Engineers from Tessent and NXP Semiconductors (NXP) worked together to implement 1687 on mixed-signal IPs in a 65 nm automotive design. The results demonstrate the significant advantages of 1687 over the current IEEE 1149.1-2001 (JTAG) test methodology⁵, both in automating the test pattern development and in reducing test setup data volume by more than 50%.

Overview of IEEE 1687-2014

The IEEE 1687-2014 standard builds upon the popular IEEE 1149.1-2001 board and card-level test access standard with a set of uniform methods to describe chip internal IP blocks, which are called “instruments.” It allows the design, test, and system engineers to easily incorporate IP into different designs, and to reuse the IP’s instrument initialization, programming, and test procedures.

The 1687 description of an instrument includes the necessary I/O ports and register details, as well as the specific procedures of how to control and access the instrument. Both parts are described through new languages defined by the standard: Instrument Connectivity Language (ICL), and Procedural Description Language (PDL).

PDL describes the operation of the instrument through commands writing to or reading from the instrument’s ports, scan chains, or data registers. These operations are described with respect to the boundaries of the instrument. PDL is compatible with the popular Tcl scripting language.

ICL provides the abstract, hierarchical definition of I/Os, registers, and scan chains that are necessary to control the instrument. The ICL does not include the details of the inner workings of instruments, but only the I/O ports, registers, bit fields, and enumerations of data elements that are necessary to carry out the operation declared in the PDL. ICL also describes the network connecting the different instruments, although the standard only intends to describe the access to the instruments, and not the details of the instruments themselves. Therefore, ICL does not necessarily directly correspond to the physical implementation.

With the combination of the ICL description and PDL operation for an instrument, the instruments can be easily ported between designs. The standard ensures that a PDL sequence of operations written and verified at the instrument module-level can be used without modification after that instrument has been inserted inside a design. However, it is the process of retargeting that translates this PDL sequence from the module’s instance level to the top level of the design (or any other hierarchy level in between).

Figure 1 shows an example of different blocks of test IP that can be described in IEEE 1687. In this example, the three blocks of instruments inside Core1 are connected to a chip level 1149.1-based TAP controller via three Segment Insertion Bit (SIB) elements.

A SIB is a one-bit shift register with a shadowed update register that controls a scan chain multiplexer. It is this SIB construct that allows a scan chain sub-region to be easily switched in and out of the full scan chain. SIBs are not part of IEEE 1687, but their usage is recommended.

IJTAG automation

To fully enable and automate the use of IJTAG, EDA software is needed for generating the PDL at the top level of the design. Tessent has developed a tool that performs four tasks:

- Reads all the ICL and PDL files for a design.
- Performs design rule checks to validate that the instruments and other 1687 components (SIBs, etc.) are properly connected up to the chip level of the design.
- Retargets the PDL description of an instrument to the chip's top level.
- Translates the resulting retargeted 1687 PDL into IEEE Std 1364 Verilog test bench and standard test vector formats, like Waveform Generation Language (WGL), Standard Test Interface Language (STIL), or Serial Vector Format (SVF).

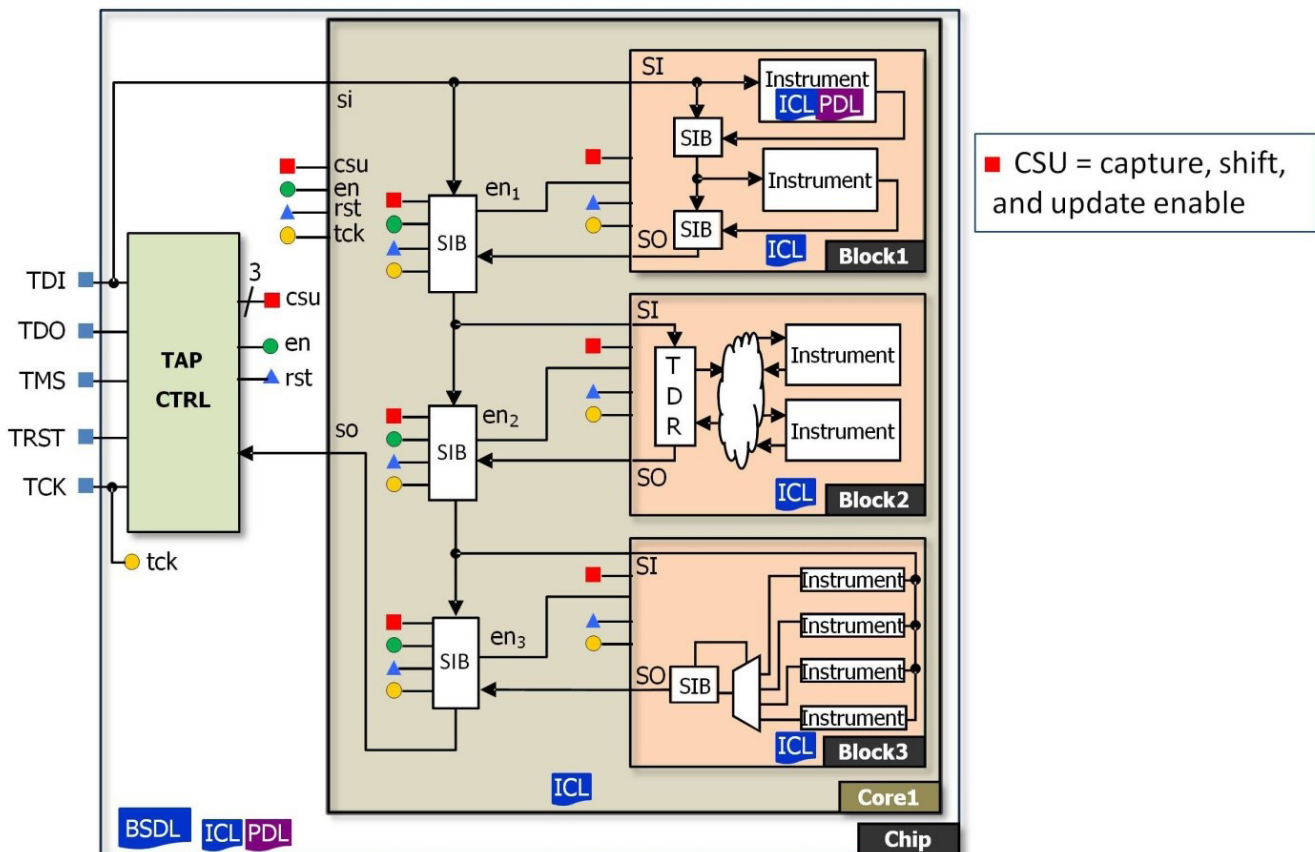


Figure 1. Example use of 1687 ICL and PDL.

IJTAG Example Application

The Tessent group at Siemens EDA worked with NXP to use IJTAG in a production design. The design is a 65 nm single-chip car radio IC with integrated tuners and digital IF DSP. It includes radio frequency (RF) tuners, channel processing, radio signal processing, audio processing, and input/output selection. Digital standards of HD Radio, Digital Audio Broadcasting (DAB), and Digital Radio Mondiale (DRM) are supported in combination with NXP's terrestrial digital radio processors.

In addition to the functional communication channels, an on-chip test infrastructure was implemented to serve silicon characterization, rapid silicon bring-up, and production test. This infrastructure is standard within NXP. For scan testing of the digital logic and memory instances, it uses a single TAM daisy-chain.⁶

Test access to and from analog IP uses a combination of analog and digital test busses, and registers controlled via an IEEE 1149.1 TAP controller. Production test development is based on abstract behavior views per IP. Translation to chip-level behavior is automated in the NXP internal core-based design flow.^{7, 8} The test infrastructure uses two types of sequential control structures that can be classified and described as instruments in the 1687 domain.

JTAG Test Infrastructure

The design fully complies with IEEE 1149.1. The on-chip test infrastructure uses a JTAG TAP as a master controller during test. The device's multiple on-chip instruments are daisy-chained in JTAG user data registers (figure 2).

There are two types of instruments. The Test Control Block (TCB) is the standard control mechanism for digital cores in NXP. Like a P1500 Wrapper Instruction Register (WIR), a TCB consists of an update register and a shift register, into which control bits can be loaded serially. The register length of a TCB is fixed in the sense that all shift register elements are on the shift path during a shift event.

The second instrument is a Test Point Register (TPR). The TPR is NXP's standard test access mechanism for digital signals on the interface of analog IP. During scan testing of digital logic, all TPRs are configured as part of the chip scan infrastructure, such that full coverage can be achieved on the functional logic, as well as on the digital test logic that surrounds the analog IP. In its mission mode, the TPR supports three different shift configurations to provide efficient register access during Analog-Mixed-Signal (AMS) and RF test. These configurations of a TPR are controlled from a single TCB. In the NXP design, some TPRs share a single TCB to control their configuration.

The design contains 167 TCBs configured in a single user data register. The TCBs control BIST instruments, on-chip sensors, I/O pad controls, test multiplexer (TMX) controls, oscillators (OSC), and settings of integrated subsystems like the Audio Base-Band (ABB) processor, Radio Front End (RFE), and other sub-systems. The total number of control bits in TCBs for this device is 2641. The TPRs are distributed over two user data registers that do not share power domains. The first data register consists of 188 TPRs and has 4258 shift register stages for control and observation of chip internal nodes. The second register consists of 5 TPRs, and has 1282 shift register stages.

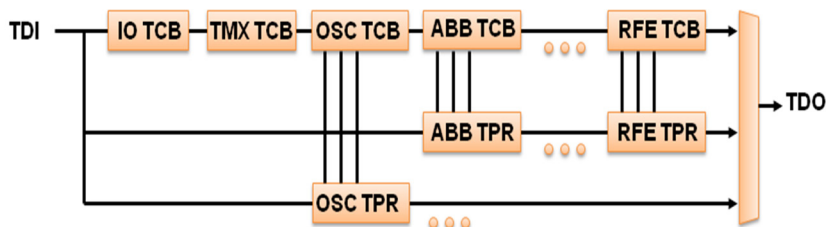


Figure 2. JTAG user data registers.

Problems with Existing Test Development

For production test development, data that initializes the environment for a specific test to be executed on a specific module is controlled via TCB. Data that is involved in the test execution is controlled via TPR. This design employs a wide variety of tests, each requiring a specific set of resources. In general, the resource scheduling is a manual task, but the translation to read and write bit sequences for the JTAG test infrastructure with TCB and TPR registers is fully automated. The automation uses an abstraction of the infrastructure that is derived from the design netlist and expressed in an NXP internal Test Data (TD) format.

For the automation of production test development, NXP introduced the concept of a test macro, shown in figure 3. The test macro defines the resource requirements for test initialization and execution of a specific test for a specific module. The main resources to be controlled for a test macro are IO pads, supplies, clocks, analog test bus, digital test bus, TCB registers, and TPR registers. Time is not explicitly modeled as a resource, which hampers a fully automated test development flow.

The data that is stored on the ATE is simple in nature because of the separation between TCB and TPR data. On the other hand, the ATE does not provide the minimum amount of digital tester data volume that is required for a specific test. Compared to the time spent in analog measurements, the test time inefficiency of the accompanying digital data transport can mostly be neglected. However the inefficient use of ATE vector memory is often of more concern.

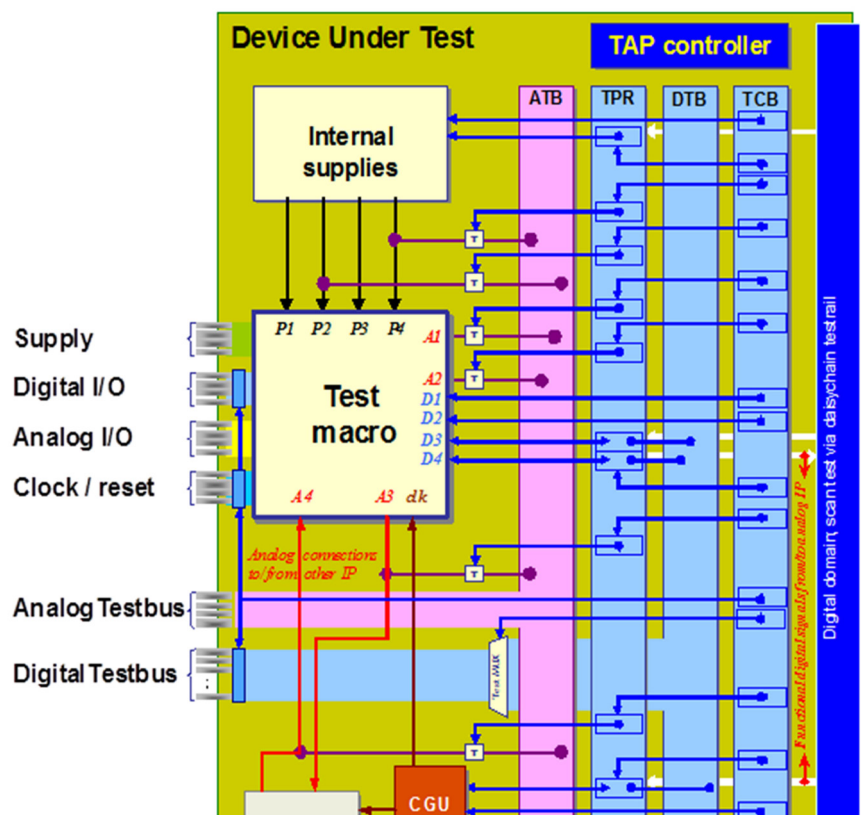


Figure 3. A test macro in test infrastructure.

The Experiments

The Tessent team at Siemens EDA and NXP partnered to explore the capabilities of the automation provided by the Tessent™ IJTAG tool⁹, which lets IEEE 1687 calculate instrument settings based on the minimum requirements for a specific test macro. The ultimate goal is to limit the user-defined input to the bare minimum.

Another goal of the experiments is to explore the impact of using the IJTAG network structure that is recommended by IEEE 1687. For this goal, we use one SIB per TPR, and one SIB per TCB, to implement a bypass per instrument controlled from the shift data. Overall, the analysis is limited to the amount of digital data that is transported via JTAG to and from TCB and TPR instruments.

MODELING IN ICL

The first step is to create ICL descriptions for all the instruments that are embedded in the design, and their top-level integration. The instrument descriptions define the register details of the TCBs, TPRs, and test mode settings. This information is extracted from the NXP TD database with a script.

Figure 4 shows the schematic of how the TCB and TPR structures are modeled in ICL. It is important to model the dependencies between the structures, because we want Tessent IJTAG to calculate the TCB content based on access requirements for user-defined TPR read and write operations.

The TPR has four configurations that can be enabled via three signals—`tpr_config`, `tpr_dynamic_en`, and `tpr_bypass`-- that cross between the TCB and the TPR. The signal `tpr_config` defines whether the structure is in its mission mode, with a shift register between `tpr_tdi` and `tpr_tdo`, or in scan test mode, in which all shift and update register elements are configured in a scan chain driven from `tpr_si`. In our experiments, we only use the mission mode of TPRs.

The selection between static and bypass mode of a TPR is now implemented by the SIB structure that is always on the shift path. When the update stage of a SIB is loaded with a logic high value, the accompanying TCB or TPR is made available on the shift path.

Conversely, when the update stage of a SIB is loaded with a logic low value, the SIB structure does not enable the TCB or TPR, which results in a shift length of one cycle. As a result, TPRs that are not relevant in a test no longer need any specific bit loaded in the configuring TCB. A TCB or TPR that is not on the shift path holds its content.

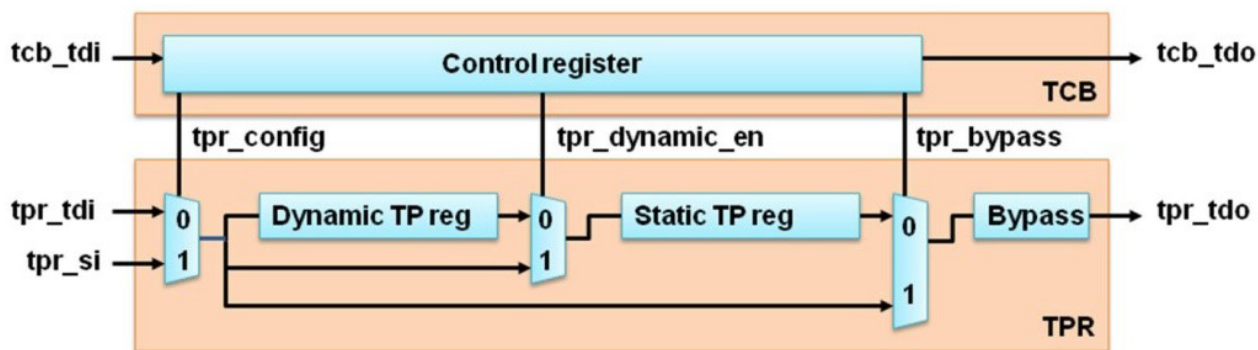


Figure 4. TCB/TPR modeling in ICL.

Two Test Groups

The production test set for NXP's automotive design can be divided in two groups of tests. One group uses the scan chain infrastructure during test execution, and includes mainly tests of the digital logic and memory instances. The second group uses the JTAG infrastructure during test execution, and targets mainly non-digital IP such as sensors, analog, RF, and mixed signal blocks.

Three Test Scenarios

Experiments based on three scenarios were performed, as illustrated in figure 5.

- The content for all TCBs is fully defined.
This represents the current situation in NXP.
- The content of all relevant TCBs is defined.
- TCBs that do not require specific content are assigned fill values upon vector creation. If SIBs are present, these TCBs can be bypassed instead.

The content of relevant TCBs that configure TPRs is calculated (i.e., not defined a priori).

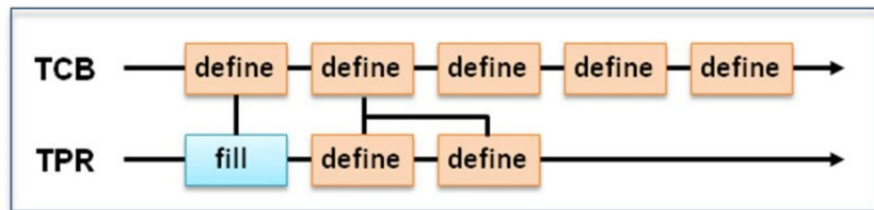
The configuration of TPRs that are not relevant in a specific test is calculated. The content of relevant TCBs that do not configure TPRs is defined.

The TCBs and TPRs that do not require specific content are assigned fill values upon vector creation. If SIBs are present, these TCBs and TPRs can be bypassed independently.

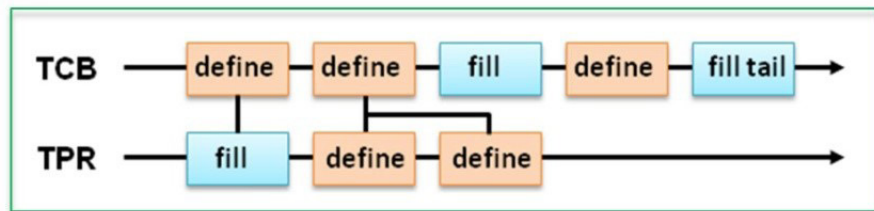
The following PDL template covers both groups of tests for the three experiment scenarios. Some tests require several consecutive test setup and test executions, as we will see in the experimental result section.

```
foreach test_setup {
  iWrite all TCB bits                (a)
  iWrite configuring TCB bits       (a)(b)
  iWrite relevant non-configuring TCB bits (a)(b)(c)
  iWrite relevant TPR bits          (a)(b)(c)
  iRead relevant TPR bits           (a)(b)(c)
  iApply                            (a)(b)(c)
}
```

Scenario *a*.
All TCBs defined



Scenario *b*. Required
TCBs defined



Scenario *c*. Configuring
TCBs calculated

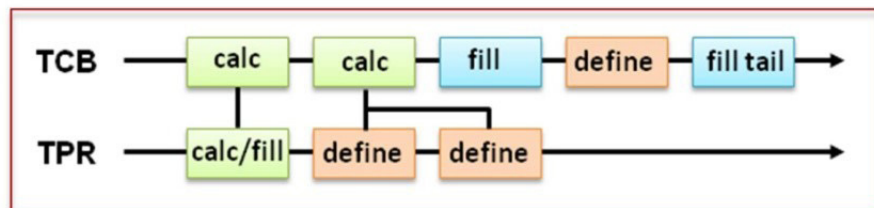


Figure 5. The three experiment scenarios: *a*, *b*, and *c*.

Empirical Results

Figure 6 shows a graphical overview of the results for all experiments. It shows the average number of cycles per test (y-axis) for all experiment scenarios, and for each required test setups (single or multiple), for non-digital tests as well as digital-only tests.

It is clear that the use of SIBs in combination with the 1687 tool (scenario c) calculates optimal data results in

the smallest number of cycles in all test cases. For the setup of digital test, a 34% reduction is achieved compared to the current test architecture and test development. For these tests, many TCBs have care bits defined, and thus cannot be bypassed. The number of cycles in the setup of tests for non-digital IP shows a reduction of 56%.

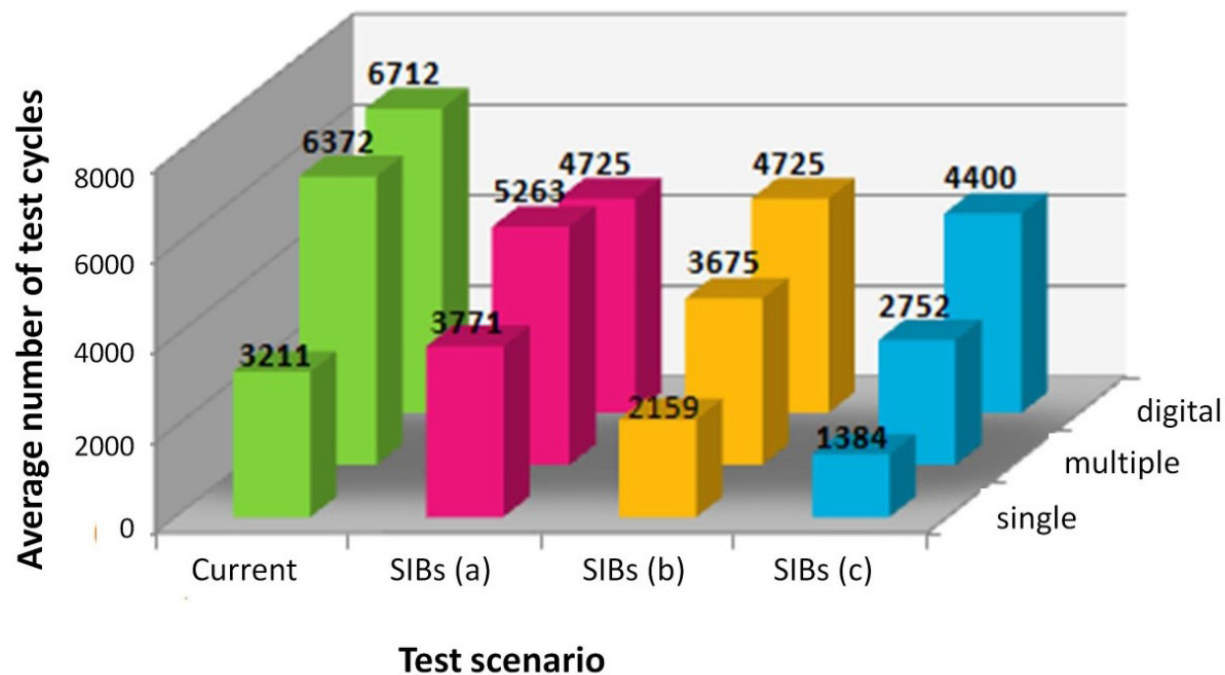


Figure 6. Overview of experimental results.

Experimental Results Details

Results of the experiment scenarios for digital logic and memory test are presented in table 1.

For each test, the number of setup sequences, TCB and TPR care bits, and resulting cycles are listed. This number of setup sequences equals the number of iApply commands. (An iApply is an IJTAG PDL command that tells Tessent IJTAG to retarget any PDL command sequences since the last iApply.) The results of the three scenarios based on the current test architecture of the automotive design are grouped in one column (abc).

Each of the three test scenarios has the same number of cycles. This confirms that the TCB care bits extracted from current automation in NXP define the optimal configuration for the TPRs in the at-speed tests. The TCB content that is calculated by the 1687 tool results

in the same TPR configuration, and thus the same number of cycles. The tests that use TCK do not require any TPR communication.

The results from the architecture that uses SIBs show a clear improvement in setup cycles compared to the incumbent architecture. In general, this is due to the TCBs that can now be bypassed via the SIB. Also, TPRs that are not relevant in a test no longer need any specific bit loaded in the configuring TCB. This explains the smaller amount of cycles in scenario c compared to scenario b. Note that the memory BIST tests do not show a difference between scenario a and b because the input data that was extracted from the NXP production test set is not the minimum set, but a fixed set that covers the control bits of all the BIST engines.

Table 1: Results for digital logic and memory test.

Architecture				Current	1687 SIBs		
Experimental scenario				abc	a	b	c
Test name	Setups	TCB bits	TPR bits	Cycles	Cycles	Cycles	Cycles
stuck-at, IDDQ on TCK	10	2641	0	2661	3000	3000	3000
at_speed_transition	3	3486	4120	14814	8175	8175	7200
at_speed_grp0_MBIST	3	3486	4120	14814	8175	8175	7200
at_speed_grp1_MBIST	3	3486	4120	14814	8175	8175	7200
at_speed_grp2_MBIST	3	3486	4120	14814	8175	8175	7200
at_speed_grp3_MBIST	3	3486	4120	14814	8175	8175	7200

Mixed-Signal Results

For non-digital IP, a total of 164 tests use a single setup sequence. Results of the experiment scenarios for these tests are presented in table 2. Multiple tests that target the same module and result in the same cycle count are grouped in a single table entry to present the data in a more compact format.

A first observation is that, for all tests, the number of cycles that result from scenario a in the case of the architecture with SIBs is larger than the number of cycles that result from the current architecture. This increase is caused by the shift overhead for the SIBs. After reset, the SIBs must be loaded to give access to

the relevant TCBs and TPRs. In scenario a, for both architectures, we force all TCBs to be loaded, which is the existing standard practice in NXP.

In the case of the architecture that uses SIBs in scenario b, the sequential loading of the non-relevant TCBs is no longer required, since they are no longer on the shift path after reset. Instead of the complete TCB register, the fill data now consists only of the accompanying SIB bit. This causes the cycle decrease for scenario b compared to scenario a. In the case where the 1687 tool is used to calculate the content of the TCBs that control

the TPRs in scenario c, there is a further reduction in cycles. In this scenario, the 1687 tool calculates sequences that make use of the initial state of SIB bits, as well as the initial state of *tpr_config* signals driven by TCBs when its active reset is de-asserted. Tessent IJTAG automatically figures out that a TCB does not require data to be loaded to guarantee the shift path of the TPRs it controls.

Table 2: Results for single setup test for non-digital IP.

Architecture				Current	1687 SIBs		
Experimental scenario				abc	a	b	c
Test group	Single setups	TCB bits	TPR bits	Cycles	Cycles	Cycles	Cycles
RFE_T2	25	18	86	2629	3085	1686	941
RFE_T1	25	22	93	3130	3673	2007	1120
RFE_TOP_CLKSLKT	3	22	99	3130	3673	2007	1120
RFE_T1_TOPBB	5	22	99	3130	3673	2007	1120
RFE_T2_TOPBB	5	24	111	3130	3673	2012	1127
RFE_AM	11	28	118	3130	3673	2072	1209
RFE_FM1	9	2	12	3130	3673	2007	1120
RFE_IFTIA1	2	22	110	3130	3673	2007	1120
RFE_FM0	9	3	14	3154	3698	2032	1153
RFE_IFTIA0	2	24	125	3154	3698	2032	1153
RFE_DBDET	6	26	105	3162	3717	2120	1257
RFE_DAB3	10	24	120	3154	3698	2032	1153
RFE_DABL	10	24	120	3154	3698	2032	1153
RFE_AGCADC	1	10	31	2950	3513	1932	1011
RFE_DABDAC	1	14	42	3014	3563	1966	1053
RFE_ASEL00	7	78	847	3760	4331	2744	2235
RFE_ASEL01	7	78	847	3760	4331	2744	2235
RFE_ASEL10	7	78	847	3760	4331	2744	2235
RFE_ASEL11	7	78	847	3760	4331	2744	2235
RFE_ASEL20	6	78	847	3760	4331	2980	2308
RFE_ASEL21	6	78	847	3760	4331	2744	2235

Conclusion

Tight co-operation between NXP and Siemens EDA has demonstrated the benefits of using IEEE 1687 on a real industrial design manufactured at the 65 nm technology node. We created PDL files for the test setup of embedded mixed signal blocks on the instrument level. We also created the ICL descriptions of the instruments and their connections, up to the top level of the design.

These experiments used Tessent IJTAG to automatically retarget the PDL descriptions from instrument level to the top-level chip pins. Top-level test benches were created to validate the correctness of the mixed signal tests. Test vectors were created as well, to transfer the test patterns to a production test. The results clearly show that test setup length can be reduced by up to 56%.

Furthermore, we confirmed that implementing the IJTAG-based tests can be done with a high level of automation. This demonstrates that tests described by IP providers at the instrument level can be automatically retargeted to the top level of the chip with a minimum of user input.

References

1. IEEE Std 1687-2014, "IEEE Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device", IEEE, USA, 2014
2. IEEE Std 1149.1-2001, "IEEE Standard Test Access Port and Boundary-Scan Architecture", IEEE, USA, 2001
3. IEEE Design and Test, Issue 5, Volume 30, 2013
4. Crouch, M. Laisne, M. Keim, "Generalizing Access to Instrumentation Embedded in a Semiconductor Device", Computer Magazine, July 2017.
5. M. Portolan, J. Rearick, M. Keim, "Linking Chip, Board, and System Test via Standards", 2020 IEEE European Test Symposium (ETS), May 2020.
6. T. Waayers, et al., "Definition of a robust Modular SOC Test Architecture; Resurrection of the single TAM daisy-chain", IEEE International Test Conference, paper 25.3, 2005
7. T. Waayers, "An improved Test Control Architecture and Test Control Expansion for Core-Based System Chips," IEEE International Test Conference, pp. 1145-1154, 2003
8. G. Seuren, T. Waayers, "Extending Digital Core-Based Test Methodology to Support Mixed-Signal" IEEE International Test Conference, pp. 281-289, 2004
9. Tessent IJTAG is the 1687 EDA tool from Siemens EDA.

Siemens Digital Industries Software

Headquarters

Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas

Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe

Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific

Unit 901-902, 9/F
Tower B, Manulife Financial Centre
223-231 Wai Yip Street, Kwun Tong
Kowloon, Hong Kong
+852 2230 3333

About Siemens Digital Industries Software

Siemens Digital Industries Software is driving transformation to enable a digital enterprise where engineering, manufacturing and electronics design meet tomorrow. Our solutions help companies of all sizes create and leverage digital twins that provide organizations with new insights, opportunities and levels of automation to drive innovation. For more information on Siemens Digital Industries Software products and services, visit siemens.com/software or follow us on [LinkedIn](#), [Twitter](#), [Facebook](#) and [Instagram](#). Siemens Digital Industries Software – Where today meets tomorrow.

siemens.com/eda

© 2019 Siemens. A list of relevant Siemens trademarks can be found [here](#). Other trademarks belong to their respective owners.

83207-C3 01/21 BM