A Plug and Play Digital ABIST Controller for Analog Sensors in Secure Devices

Sébastien Lapeyre, Nicolas Valette, Marc Merandat, Marie-Lise Flottes, Bruno Rouzeyre, Arnaud Virazel

A Plug and Play Digital ABIST Controller for Analog Sensors in Secure Devices

S. Lapeyre N. Valette M. Merandat
INVIA
Meyreuil, France
{firstname.lastname}@invia.fr

M.-L. Flottes B. Rouzeyre A. Virazel
LIRMM - University of Montpellier / CNRS
Montpellier, France
{firstname.lastname}@lirmm.fr

***Abstract*—Secure devices embed analog sensors in order to measure some physical/environmental parameters which can alter its behavior such as temperature, voltage and electromagnetic field. To ensure the device security all along its lifetime, it is necessary to rely on those analog sensors. To achieve it, test solutions must be designed and proceed at each step of the system life cycle, taking into account inherent constraints of each cycle, i.e., absence or defective software in the chip or chip in user hand for example. In this paper, we present a plug and play digital ABIST controller which allows to run external or internal autonomous test phases on a temperature sensor used as case study. The external test mode is fully compliant with the IEEE Std. 1149.1 while the internal test one is controlled by the embedded CPU through a system bus.**

Keywords—Sensor test, ABIST, JTAG, digital control.

I. INTRODUCTION

The need for secure devices is typically present in many different fields of application such as medical, automotive [1] or military fields. In addition, connected "objects", smartphones, smart cards, RFID devices, are nowadays used by a large part of the population for private and professional usage. These objects also need to embed secure devices for privacy and personal data protection from potential cyber-attack or any form of data leakage. Forecasts indicate that the secure device market will reach USD 2036.2 million in 2022 [1][2]. In parallel, the number and the quality of the cyberattacks is increasing. Cybercrime damages will cost at least USD 6 trillion annually [3]. The profile and the target of the attackers are various. They range from basic users to nation state-attackers, focusing on software and/or hardware vulnerabilities of secure devices [4].

Hardware attacks are very diverse in terms of skill, strategy, tools and time requirements. A side-channel power analysis was used to retrieve a safe lock key value [6]. Reset glitch attack was used to skip instruction execution of a gaming console CPU. As a result, the attacker succeeded to skip the instruction in charge of ROM signature check, enabling malicious software to be loaded [7]. Cooling attacks allowed to retain data for dangerous period in a SRAM after power off [8]. Even combined attacks are realized, such as proposed in [9] where it is shown that heating the circuits to 100°C make it easier to realize a clock glitch attack.

As mentioned in NIST documentation on security requirements for cryptographic modules [10], sensors are part of the countermeasure portfolio for preventing hardware attacks. Their diversity (glitch detectors [11], temperature sensors [12][13], laser injection detector [14][15], voltage droop detectors [16]) allows to match with a range of possible perturbation attacks on hardware. While integrated in larger systems, their status can be checked continuously for triggering alarms when needed. They are thus expected to be reliable.

While cyber and hardware attacks have raised the need for secured device, the test of each silicon part in secure application is a must. Manufacturing quality, reliability and resiliency indeed relies on test technologies deployed for detection of faults or errors at post-production level and at mission time as well. Tests allow to detect manufacturing defects, failure or, potentially intentional misuses of the devices [17]. As first lines of defense, sensors for security must be tested as well and a secure test solution must be developed for that. As an example, the first ABIST (Analog Built-In-Self-Test) was presented 30 years ago in [18].

Test resources must include infrastructures for test application and test data storage. With respect to the infrastructure, two different approaches exist for analog Design-for-Test (DfT). First, parallel digital access, present in IC's, have been used to control analog commands. The IEEE Std. 1149.4 [19] attempts to standardize analog test access. However, the necessity of two extra pins compared to IEEE Std. 1149.1 [20] may be too costly, especially in smart-card domain where just five pins are available. The second approach for analog DfT is the serial digital access. The major advantage is to reduce area cost related to the bus width, but this solution increases the test time due to the serial/shift process. This solution needs Serial/Parallel conversion solution to convert commands and capturing results from analog buses [21]. An IEEE working group plan to extend digital-focused IEEE Std. 1687 [22] to include tools for analog ICs tests. The infrastructure is expected to be compatible for both analog and digital IPs. This solution allows parallel or/and serial canal to bring data in test instruments. It could be used to reduce the BIST cost area offloading computation traditionally done by the hardware components [23]. This standard introduces the IJTAG with the Reconfigurable Scan Network which can be used to facilitate the deployment of test instruments in analog sensors.

On the other hand, test resources availability for data storage depends on the step of the life cycle during which the test has to be run. Right after manufacturing for instance, the chip does not embed any software and all memories are empty. Conversely, after deployment, software and memory resources can be used for BIST purpose. However, because the chip could be in an unknown status with possible defects on software or memory, other resources must be provided for failure analysis in that case.

In this paper, we present a low-cost area solution to manage tests on analog sensors embedded in secure devices. Our goal is to design the simplest architecture handling both BIST mode and external test mode through JTAG, i.e., a common-used test canal in the industry. The proposed digital controller allows operating tests during the entire lifetime. To achieve it, it needs to supervise both the two test modes. The external test mode uses a serial digital access fully compliant with IEEE Std. 1149.1. A parallel digital access controlled by the CPU through the system bus is used to operate the BIST mode. A post-processing delivers a Pass/Fail test response to the CPU by the end of the BIST procedure. Those hardware features reduce the test timing and CPU calculation which is critical during boot (when test may be launched). Conversely, for external test, there is no post-processing on the test result.

It permits to give the full information to the tester and simplify our system by reducing the area cost. Full test result availability is an asset in critical analysis step such as post-manufacture and failure analysis.

Overview of various analog sensor with their own properties lead us to design a very adaptive controller. The analog designer set the test delay according to its design and the time needed to get stable test results. He/she defines the bit-size of input/output test data and provide a structural digital access to the ABIST infrastructure in order to connect the device under test to the digital controller. The proposed plug-and-play ABIST controller design allows to implement these designer-defined tests for one or various analog sensors within the same system.

For a better understanding of our solution, we designed an ABIST temperature sensor as case study in order to illustrate the test process. The temperature sensor is commonly used in secure devices to survey check whether the system is used in correct operational range temperature. To detect a misbehavior, we add several test signals in charge of verifying the different voltage references embedded in the sensor as well as stuck-at fault issues.

The paper is organized as follows. A digitally controlled Analog Built In Self-Test (ABIST) solution for temperature sensor is presented in Section II. In Section III, a digital controller allowing to manage autonomous and nonautonomous tests is described. Section IV presents a generalization of the proposed architecture with several sensors in order to optimize test operations and area cost. Section V concludes the paper and gives future perspectives.

II. ABIST ON TEMPERATURE SENSOR

A. *Temperature Sensor Description*

Each device operates effectively in a specified temperature range depending on manufacturer designs. Outside this range, faults may appear. In the context of secure devices, any kind of misbehavior is forbidden because it can generate security issues. A temperature sensor is commonly used to acknowledge that the current temperature is in the permitted range. It consists in raising an alarm when the temperature is out of the range specified by the manufacturer, -20°C to 120°C for smart card [24] and up to -55°C to 125°C in military fields [25]. It is a main asset to put the system in safe mode, to prevent misconduct due to processing in an extreme environment but also to detect cooling or heating attacks such as heating faults [26] or cold boot [27].

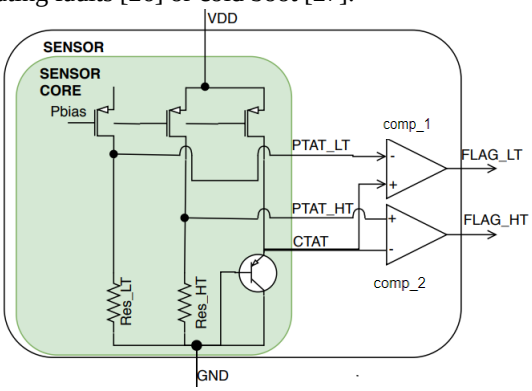

Figure 1: Temperature sensor

The temperature sensor implementation considered as case study is depicted in Figure 1. It embeds a stable voltage reference and a resistor network to generate voltage references, called PTAT_LT (Proportional To Absolute Temperature_Low Temperature) and PTAT_HT (Proportional To Absolute Temperature_High Temperature). Those stable voltages are compared with CTAT (Complementary To Absolute Temperature), which vary according to the current temperature, to generate, digital flags FLAG_LT and FLAG_HT respectively, as shown in Figure 2.

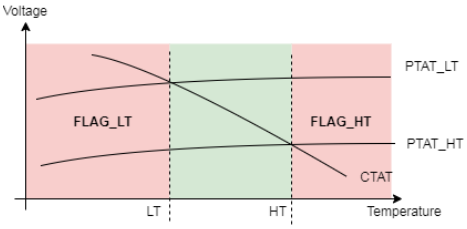

Figure 2: Temperature sensor functioning

B. *Adding DFT in the Temperature Sensor*

In the following, we use as an illustration the temperature sensor to describe how an analog sensor can be modified in order to manage test with the proposed digital ABIST controller. In this example, we did not target the best test coverage, but we show that with only a 3-bit digital signal commands it is possible to detect defects on voltage reference, PTAT_HT and PTAT_LT, and stuck-at fault on the comparator (comp_1 and comp_2). Those information are crucial all along the lifecycle. It allows detecting defaults in manufacturing, degradations of the analog sensors in fields, due to the aging or an attack, and permits a sharper failure analysis.

With this 3-bit signal, the tester can operate 3 different tests described below:

- TEST 1: Verify that both comparators (comp_1 and comp_2 are not stuck and verify that CTAT is between PTAT_LT and PTAT_HT (PTAT_LT > CTAT > PTAT_HT). This is implemented by crossing PTAT_LT and PTAT_HT in the scheme and verifying that both output flags are raised.
- TEST 2: Verify that PTAT_LT is above PTAT_HT. This is implemented by crossing CTAT and PTAT_HT and verifying that both output FLAGS are Low.
- TEST 3: Verify the resistance network, by chunking a part of it. The resistance network for PTAT_LT, correspond to an equivalence resistance RES_LT=RES_HT+RES_DIFF. Chunking this resistance network to a value just under RES_DIFF (noted RES_DIFF$^-$ on Figure 3), verify that PTAT_LT is above PTAT_HT multiplied by a given ratio (PTAT_LT >= K x PTAT_HT with K=RES_LT/RES_HT).

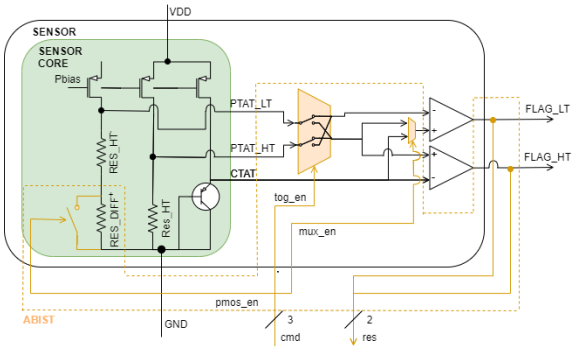

Figure 3: Temperature sensor with DFT

Figure 3 depicts the modification done to the considered temperature sensor to perform those tests. It illustrates the 3-bit digital signal command, named pmos_en, tog_en and mux_en. Test results are observed on flag functional signal FLAG_LT and FLAG_HT.

In TABLE I. , the commands corresponding to the tests and the expected results for a pass test are resumed.

TABLE I. TEST COMMANDS AND EXPECTED RESULTS

Function tested	Commands values			Expected result	
	pmos_en	*tog_en*	*mux_en*	*FLAG_LT*	*FLAG_HT*
TEST 1	0	1	0	1	1
TEST 2	0	0	1	0	0
TEST 3	1	0	1	1	0

In this part, we introduced how a sensor could be modified to implement an ABIST with the example of a temperature sensor. In the next part, we present how tests are executed and controlled by a digital controller with the ABIST Inputs/Outputs (cmd and res in the case study).

III. ABIST CONTROLLER

In this section we describe the ABIST controller with its different blocks, how commands access the sensor and the complete process of test and finally some synthesis results.

A. System overview

To manage sensors tests, we developed a digital system, called ABIST Controller, in charge of transmitting the test commands, and managing test results to and from the sensor. The main goal is to process tests all along the life cycle of the secure device. To achieve it, the developed ABIST controller uses two modes: the BIST mode and the external test mode. The BIST mode is controlled by the CPU embedded in the system and the external mode is supervised by a JTAG access. The BIST mode is used during operation (during the boot phase for example), external test mode through JTAG is used in test labs, when software has not been uploaded yet or to analyze a defective product.

In both modes, a command is passed to ABIST inputs. Then, the result is captured after a "test delay" specified by the ABIST designer. This "test delay" can be different for each test and sensor. It depends on the time needed to have a stable result.

The system embedding the ABIST Controller must have a signal acknowledging the current test mode. It is important that both modes do not interfere with each other's because they use the same ABIST Inputs/Outputs to control test signals (command and result). In the system implementation, we consider that this signal sel_test_mode is generated by another entity, out of our IP, which is in in charge to manage the selected test mode.

Also, a standardized TAP (Test Access Port) is recommended to perform external test. For internal test, we arbitrary chose an APB (Advanced Peripheral Bus) bus [28] to communicate with the embedded CPU, but any other data bus can be chosen.

The controller itself is composed of four sub-modules (as depicted in Figure 4.

1) *The Manager (ABIST_CTRL_MGR) is:*
   - responsible of internal test management,
   - running under internal clock domain (pclk or system clock),
   - interfacing with the APB to communicate with the CPU,
   - controlling the JTAG command module reset.
2) *The JTAG Command Manager (JTAG_CMD) is:*
   - responsible of external test management,
   - running under external clock domain (TCK or test clock),
   - interfacing with the TAP.
3) *The synchronization block (synchro_sck_tck):*
   - synchronizes control signal from system clock to test clock.

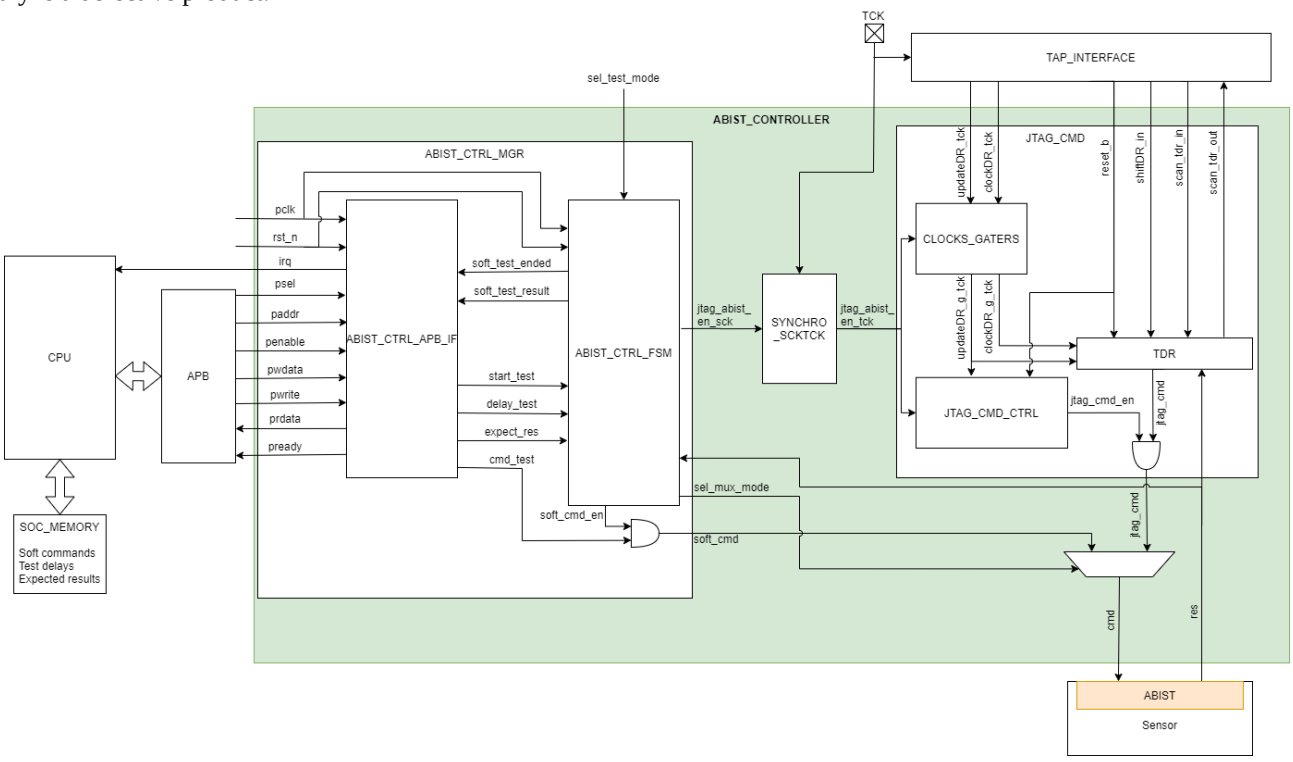

Figure 4: ABIST controller architecture

4) The multiplexer:

- selects command from JTAG or SOFT mode in order to avoid interferences between the two blocks

B. Control command

The ABIST Controller guarantee two following important properties for the test command accessing the analog part:

- Property_1: A command from the non-selected test mode cannot interfere with the running test.
- Property_2: If a test is not running, the command sent to the sensor must be equal to the "null command". The "null command" corresponds to the vector that disables all DfT functions ("000" for the temperature sensor described above).

To satisfy Property_1, a multiplexer handles which command signals, jtag_cmd or soft_cmd, are redirected to the sensor. This multiplexer is controlled by the signal sel_mux_mode generated by ABIST_CTRL_FSM.

Property_2 is satisfied thanks to gating modules embedded in JTAG_CMD and ABIST_CTRL_MGR respectively controlled by jtag_cmd_en and soft_cmd_en signals. Their behavior are presented below.

The jtag_cmd_en signal is generated by JTAG_CMD_CTRL entity described in Figure 5. A rising edge can be observed on this signal when the command is updated (TAP state equals Update_DR) in the TDR (Test Data Register). It asynchronously switches to '0' when jtag_abist_en_tck signal is disabled or when reset from TAP is active.

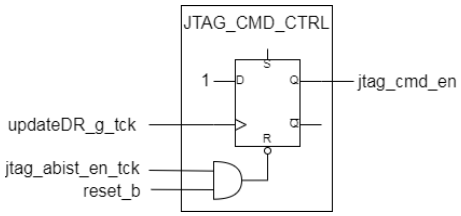

Figure 5 : JTAG_CMD_CTRL entity

The soft_cmd_en signal is generated by the ABIST_CTRL_FSM entity and it is controlled by the FSM (Finite State Machine) depicted in Figure 6. This signal rises when the start command has been sent by the CPU and falls when the test delay is passed.

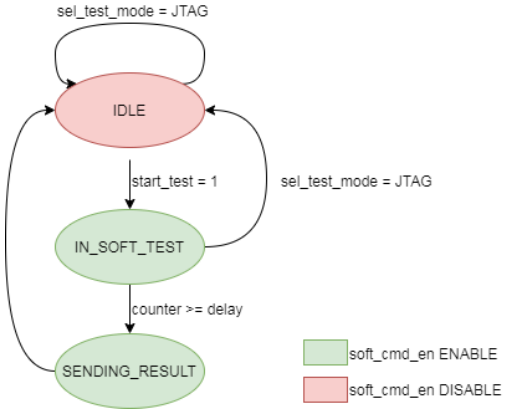

Figure 6 : ABIST_CTRL_FSM

C. External test mode

We consider JTAG components described in IEEE 1149.1 standards to manage external test. Only IEEE 1149.1 signals are necessary to communicate with JTAG_CMD sub-module.

The external test mode is set as the default mode to cover a software defect (i.e., during the failure analysis phase) or the absence of software (i.e., during the manufacturing test process). If it wasn't the case, test processing would be inaccessible in the case where the test mode manager would be defective. That hypothesis must be taken into account.

In external test mode, digital test commands and results are sent through the TDR. According to sensor design, test delays are managed by the external tester. It must include these delays in its test protocol. JTAG command, named jtag_cmd is gated by the signal jtag_cmd_en controlled by IEEE 1149.1 signals and jtag_abist_en_tck. This feature prevents unwanted commands triggering the sensor.

1) External test mode procedure

External test can be described with the seven steps presented below.

1. The JTAG test mode must be enabled (if not already enabled) thanks to the signal *sel_test_mode* from a global test manager
2. Clock gater then *jtag_cmd_en* are enabled thanks to the signal *jtag_abist_en_tck* issued from ABIST_CTRL_FSM.
3. ABIST TDR selected thanks to the TAP protocol, using the Test Data Instruction (TDI).
4. Shift command test in TDR shift register thanks to the TAP through JTAG.
5. Update command and send it to extra control inputs of the modified sensor (3-bit cmd signal in Figure 3). Tester has to wait for test delay to get the stable result. This test delay must be converted into external clock period.
6. Get result (2-bit *res* signal in Figure 3). Captured in the TDR and shifted out by the tester, through JTAG interface.
7. To avoid misbehavior, the tester has to shift the "null command" in TDR when getting last result to avoid new test sequence or go to Test-Logic-Reset TAP state which reset the TDR to the null command. If this step is not done, the sensor could stay in test mode with the last command entered, until a reset is done.

2) External test mode timing

The runtime depends on:

- the offset due to TAP Finite State Machine (equals 11 test clock cycles),
- the length of the TDI (Test Data Instruction) an IEEE 1149-1 component embedded in TAP_INTERFACE,
- the length of the TDR for control/observation of the proposed ABIST Controller solution. …equal to the maximum number of bits for control or observation), Here, 3 bits are necessary for controlling the ABIST controller implementation presented in Section II,
- the test delay,
- the period of the test clock (T_{TCLK}).

$$Runtime = T_{TCLK} \times (len(TDI) + 2 \times len(TDR)) + \max(test\ delay\ ; 2 \times T_{TCLK})$$

where the part depending on TDI length (len(TDI)) and TDR length (len(TDR)) corresponds to the time necessary to shift instruction, command and result through JTAG channel. If the test delay is less than 2 T_{TCLK}, we still have an overtime between update and capture step due to the TAP state machine.

D. BIST test mode

The BIST test mode running on an analog sensor, requires that test information (i.e., test procedure) is stored in the available memory. For the temperature sensor case study presented in Section II, this test information defines the 3-bits cmd signal for the sensor, the test delay and the expected test result to compare with the 2-bit sensor signal res.

The BIST test mode is launched by the embedded CPU and test vector is sent to the ABIST Controller bus interface through the data bus system (in our case an APB is used) managed by the ABIST_CTRL_APB_IF entity. The ABIST_CTRL_FSM entity is in charge of waiting the test execution, comparing the test result with the expected one and throwing an interruption to inform the CPU that the test is finished. It could then request for the result of the test, which is a Pass or Fail response.

1) BIST mode usage

BIST usage can be described with the six following steps:

1. Enable soft test: Test Manager selects internal test mode for ABIST controller Manager.
2. Signal jtag_abist_en_tck disabled: TCK clocks are gated and JTAG commands are disabled.
3. Test sequence initialization: Enable the end test interruption. CPU copies command, test delay and expected result from memory into ABIST controller manager registers, using APB interface.
4. Test triggered: CPU triggers sequence through APB interface.
5. Test processing: ABIST controller manager is autonomous. It sends command to sensor, waits for the test delay (converted in internal clock period) and captures the output of the sensor. Then, it compares it with the expected result and stores the status into a register (indicating Pass or Fail).
6. Test completion: When the processing is finished, the ABIST sets a flag in APB interface, indicating that the test is completed. An interrupt could be generated, if suitable for CPU. Then, CPU can read the result register.

2) Internal Test timing

Runtime depends only on test delay and system clock frequency. It needs 9 system clock cycles to initialize the test (enable the interruption, write the test configuration and start the test), a user defined test delay to trigger the interruption, and 3 more system clock cycles to read the result. We add 30 clock cycles off overtime to manage the interruption, which is the time needed by our CPU to save the context before handling an interruption.

$$Runtime = 42 \times T_{SCLK} + test\ delay$$

E. Synthesis result

The proposed ABIST Controller was designed in using a 55nm CMOS technology. Its area is directly related to the length of the control and observed data (cmd and res in our example). A number of digital ABIST Controller was synthetized with different command and result length. Thus, the area can be stated according to the following formula (in gate equivalent), with +/- 0.5% precision:

$$Area = 665 + 55 \times len(cmd) + 27 \times len(res)$$

The exact values are given in TABLE II. to compare, the area of the temperature sensor (designed with the same technology) is 15 000 µm². Adding an ABIST Controller to manage its test it corresponds to an overhead of about 6 %.

TABLE II. ABIST_CONTROLLER COMPLEXITY

cmd length (bits)	res length (bits)	Logic gate (GE)	Silicon area (µm²)
1	1	744	834.12
3	2	866	993.51
16	8	1760	1972.53

IV. TESTING MULTIPLE ANALOG DEVICES

A SoC (System on Chip) commonly embeds several analog sensors (temperature sensor, voltage class monitor, glitch detector …). Those sensors have different properties such as power consumption, design complexity, response time. Thanks to its flexibility, the proposed ABIST Controller can be set for different targets with specific response time, test delays and test data lengths (e.g., glitch detector and Physically Unclonable Function)..

This plug-and-play solution is illustrated on the example depicted in Figure 7. The target is a 2x2 temperature sensor matrix [29] embedded to fully cover the device area. In that case, the ABIST Controller has a 12 command signals (i.e., 3 signals by sensor) and 8 result signals (i.e., 2 signals by sensor). The test delay for each test can be approximated by 2µs. System clock frequency is 100 MHz and external test clock frequency is 10 MHz.

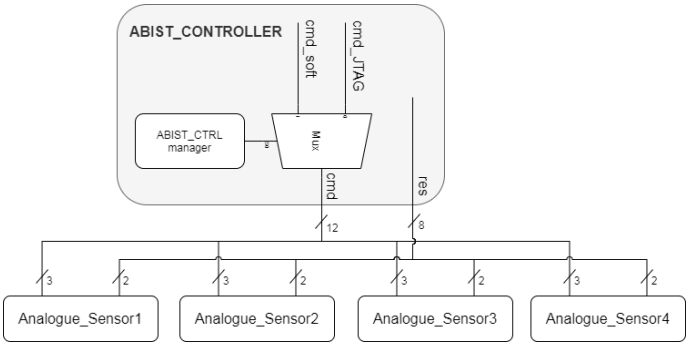

Figure 7: ABIST linked to four analog sensors

TABLE III. shows the difference of cost area and test time to implement the 3 tests described above on 4 sensors in a system using three different architectures: 1/ only one ABIST Controller for 4 sensors, 2/ one ABIST Controller per sensor and their TDRs on the same chain, 3/ one ABIST Controller per sensor and their TDRs on different chains. In all of them the TDI length is equal to 5. The interruption management allows the CPU to launch multiple auto test in parallel (the CPU doesn't have to wait for the end of the test to execute other operation). As a result, internal test timing doesn't differ for all implementations. But cost area and external test timing are significantly improved if the designer embeds a single ABIST Controller for four temperature sensors.

TABLE III. COST AREA AND TEST TIMING WITCH DIFFERENT ARCHITECTURE CHOICES

Architecture choice	Cost area (GE)	External test timing (µs)	Internal test timing (µs)
1 ABIST_CTRL for 4 sensors	1531	13.7	7.26
4 ABIST_CTRL for 4 sensors with TDR on the same scan chain	3536	13.7	7.26
4 ABIST_CTRL for 4 sensors with TDR on different scan chains	3536	25.2	7.26

V. SECURITY CONCERN

Our solution aims to improve the global security of a system embedding analog sensors dedicated to environmental parameters monitoring. In order to rely on those sensors during the entire product lifetime, we introduced ABIST facilities and infrastructures allowing to implement built-in self-tests or external tests when needed.

Design-for-testability, i.e., introduction of design features for improved testability, may however increase the system attack surface. As an example, a significantly number of attacks using the JTAG test channel has been described in [30][31][32]. It demonstrates that the security of test infrastructures is as much as important as the security of other components [33].

With respect to our proposal, we assume here that CPUs and memories used during BIST procedures are secure because embedded in a secure system. Tests using external data must however be secure too. Several solutions exist in the literature for secure access to the test infrastructures, from (dynamic) key-check systems to strong user authentication processes or test data encryption [34][35]. Those solutions are all compliant with our test scheme.

VI. CONCLUSION

Test management for sensors dedicated to security in integrated systems is essential for achieving reliability and security. Sensor accessibility and test procedures must thus be available during the entire system life time. In this paper we propose a test scheme for analog sensors including test procedure, sensor design-for-testability, ABIST controller for test in the field, and possible access to external test resources when needed. While the test procedure and the sensor level design-for-testability solution has been set up on one particular sensor dedicated to temperature sensing, the proposed low-cost ABIST controller is compliant with any test-dedicated digital interface on a sensor and can manage different test lengths, test delays and numbers of devices under test. The architecture is IEEE 1149.1 compliant for external access. It makes it easily integrable in an industrial process with suitable tools.

REFERENCES

[1] S. Katzenbeisser, I. Polian, F. Regazzoni and M. Stöttinger, "Security in Autonomous Systems", *2019 IEEE European Test Symposium (ETS)*, Baden-Baden, Germany, 2019, pp. 1-8

[2] Industry research, "Global hardware security modules (HSM) market 2020 by manufacturers, regions, type and application, forecast to 2025", August 2020, https://www.marketsandresearch.biz/report/98330/global-hardware-security-modules-hsm-market-2020-by-manufacturers-type-and-application-forecast-to-2025

[3] Morgan, S.,"Global Cybercrime Damages Predicted To Reach $6 Trillion Annually By 2021", 2018 Cybercrime Magazine.

[4] M. Rocchetto and N. O. Tippenhauer, "On attacker models and profiles for cyber-physical systems", *European Symposium on Research in Computer Security*, pp. 427-449, 2016.

[5] Christina Quast, "Common Attacks on IoT device", *Europe Open IoT Summit*, October 2018, [online] Available: https://elinux.org/images/f/f8/Common-Attacks-on-IoT-Devices-Christina-Quast.pdf.

[6] *Plore. Side-channel Attacks on High-security Electronic Safe Locks Def Con 24*, 2016, [online] Available: https://www.defcon.org/.

[7] E. DeBusschere and M. McCambridge, "Modern Game Console Exploitation", *Technical Report Department of Computer Science University of Arizona*, 2012.

[8] SP Skorobogatov, "Low temperature data remanence in static RAM", *Technical report UCAM-C L-TR-536*, June 2002.

[9] T. Korak, M. Hutter, B. Ege and L. Batina, "Clock glitch attacks in the presence of heating", *Proc. Workshop Fault Diagnosis Tolerance Cryptogr. (FDTC)*, pp. 104-114, Sep. 2014.

[10] NIST 2001. FIPS 140-2, "Security Requirements for Cryptographic Modules", Washington, US Government Printing OfficeQsd.

[11] A. G. Yanci, S. Pickles and T. Arslan, "Characterization of a Voltage Glitch Attack Detector for Secure Devices", 2009 Symposium on Bio-inspired Learning and Intelligent Systems for Security, Edinburgh, 2009, pp. 91-96.

[12] L. Cartagena and S. Barbin, "Low power CMOS temperature protection sensor for smart cards", *2017 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC)*, Aguas de Lindoia, 2017, pp. 1-5,

[13] Y. Hashimoto, T. Toyoshima, S. Yogo, M. Koike, T. Hamaguchi, S. Jing, et al., "Safety securing approach against cyber-attacks for process control system", *Computers & Chemical Engineering*, vol. 57, pp. 181-186, 2013.

[14] F. Lu, G. Di Natale, M. Flottes and B. Rouzeyre, "Customized cell detector for laser-induced-fault detection", *2014 IEEE 20th International On-Line Testing Symposium (IOLTS)*, Platja d'Aro, Girona, 2014, pp. 37-42.

[15] O. Derouet, "Secure Smartcard Design against Laser Fault Injection", *FTDC*, 2007

[16] R. Petersen, "Voltage transient detection and induction for debug and test", *Proc. IEEE ITC*, 2009-N

[17] A. Antonopoulos et al., "Trusted analog/mixed- signal/RF ICs: A survey and a perspective", IEEE Design & Test, vol. 34, no. 6, pp. 63–76, 2017..

[18] L. Milor and V. Visvanathan, "Detection of catastrophic faults in analog integrated circuits", *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 8, no. 2, pp. 114–130, Feb. 1989.

[19] "IEEE standard for a mixed-signal test bus", IEEE Std 1149.4-2010 (Revision of IEEE Std 1149.4-1999), 2011.

[20] "IEEE Standard 1149.1: Standard Test Access Port and Boundary Scan", 1990.

[21] S. Sunter, J. -. Côté and J. Rearick, "Streaming Access to ADCs and DACs for Mixed-Signal ATPG", in *IEEE Design & Test*, vol. 33, no. 6, pp. 38-45, Dec. 2016

[22] "IEEE standard for access and control of instrumentation embedded within a semiconductor device", IEEE Std 1687-2014, 2014.

[23] M. M. Portolan, M. J. Barragan, R. Alhakim and S. Mir, "Mixed-signal BIST computation offloading using IEEE 1687", 2017 22nd IEEE European Test Symposium (ETS), Limassol, 2017, pp. 1-2.

[24] L. Cartagena and S. Barbin, "Low power CMOS temperature protection sensor for smart cards", 2017 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC), Aguas de Lindoia, 2017, pp. 1-5.

[25] Maxim Integrated,"Military and Aerospace Semiconductors", https://www.maximintegrated.com/en/markets/military-aerospace.html

[26] Hutter and J.-M. Schmidt, "The temperature side channel and heating fault attacks" in Smart Card Research and Advanced Applications—CARDIS 2013, Cham, Switzerland:Springer, pp. 219-235, 2013.

[27] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, et al., "Lest we remember: Cold boot attacks on encryption keys", *USENIX Security Symposium*, pp. 45-60, 2008.

[28] Specification, A. M. B. A. "Rev. 2.0." ARM Limited (1999).

[29] V. Szekely, M. Rencz and B. Courtois, "Integrating on-chip temperature sensors into DfT schemes and BIST architectures", *Proceedings. 15th IEEE VLSI Test Symposium (Cat. No.97TB100125)*, Monterey, CA, USA, 1997, pp. 440-445

[30] Amitabh Das, "Differential Scan-Based Side-Channel Attacks and Countermeasures (Differentiële scan-gebaseerde nevenkanaalaanvallen en tegenmaatregelen)", October 2013

[31] K. Rosenfeld and R. Karri, "Attacks and Defenses for JTAG", in IEEE Design & Test of Computers, vol. 27, no. 1, pp. 36-47, Jan.-Feb. 2010

[32] C. Clark, "Anti-tamper JTAG TAP design enables DRM to JTAG registers and P1687 on-chip instruments", 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), Anaheim, CA, 2010

[33] D. Hanford, "KeyNote: The Role of Security and Test in the New Era of Silicon Lifecycle Management",IEEE International Symosium on Hardware Oriented Security and Trust, 2020, [online] Available: http://www.hostsymposium.org/speakers/DeirdreHanford.php.

[34] M. Portolan, V. Reynaud, P. Maistri and R. Leveugle, "Dynamic Authentication-Based Secure Access to Test Infrastructure", 2020 IEEE European Test Symposium (ETS), Tallinn, Estonia, 2020, pp. 1-6

[35] R. Baranowski, M. A. Kochte and H. Wunderlich, "Fine-Grained Access Management in Reconfigurable Scan Networks", in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, no. 6, pp. 937-946, June 20.