Recent Trends and Perspectives on Defect-Oriented Testing

Recent Trends and Perspectives on Defect-Oriented Testing

P. Bernardi[§], R. Cantoro[§], A. Coyette[*], W. Dobbeleare[*], M. Fieback[¶], A. Floridia[‡], G. Gielen[‖], J. Gomez[‖], M. Grosso[‡], A. M. Guerriero[†], I. Guglielminetti[§], S. Hamdioui[¶], G. Insinga[§], N. Mautone[†], N. Mirabella[‡§], S. Sartoni[§], M. Sonza Reorda[§], R. Ullmann[†], R. Vanhooren[*], N. Xama[‖], L. Wu[¶]
[*]onsemi, BE, [†]Infineon Technologies, DE, [‡]STMicroelectronics, IT, [§]Politecnico di Torino, IT,
[¶]Technische Universiteit Delft, NL, [‖]Katholieke Universiteit Leuven, BE

***Abstract*—Electronics employed in modern safety-critical systems require severe qualification during the manufacturing process and in the field, to prevent fault effects from manifesting themselves as critical failures during mission operations. Traditional fault models are not sufficient anymore to guarantee the required quality levels for chips utilized in mission-critical applications. The research community and industry have been investigating new test approaches such as device-aware test, cell-aware test, path-delay test, and even test methodologies based on the analysis of manufacturing data to move the scope from 0PPM to 0PPB. This special session presents four contributions, from academic researchers and industry professionals, to enable better chip quality. We present results on various activities towards this objective, including device-aware test, software-based self-test, and memory test.**

***Index Terms*—cell-aware test, DPPM, DPPB, device-aware test, emerging technologies, Flash, non-volatile memories, data analytics, visual inspection**

I. Introduction

Conventional testing approaches based on traditional fault models (e.g., stuck-at and transition delay) for digital logic or March tests for memories fail to detect significant defectivity, which results in customer returns or may cause critical misbehaviors in the final application. As a result, testing techniques based on those approaches are no longer sufficient to guarantee the quality levels required during manufacturing and in the field.

This paper collects four contributions from industrial and academic researchers tackling defect-oriented testing methodologies, ranging from functional testing of digital logic (Section II) to memory testing (Sections III and IV) and data analytics of visual inspection on wafers (Section V).

Riccardo Cantoro, Andrea Floridia, Michelangelo Grosso, Iacopo Guglielminetti, Nunzio Mirabella, Sandro Sartoni and Matteo Sonza Reorda, from Politecnico di Torino and STMicroelectronics, are the authors of Section II.

Moritz Fieback, Said Hamdioui and Lizhou Wu, from Technische Universiteit Delft, are the authors of Section III.

Paolo Bernardi, Augusto Maria Guerriero, Giorgio Insinga, Nellina Mautone and Rudolf Ullmann, from Politecnico di Torino and Infineon Technologies, are the authors of Section IV.

Anthony Coyette, Wim Dobbelaere, Georges Gielen, Jhon Gomez, Ronny Vanhooren and Nektar Xama, from Katholieke Universiteit Leuven and onsemi, are the authors of Section V.

978-1-6654-7355-2/22/$31.00 ©2022 IEEE

Many published industrial studies have shown that the new cell-aware test (CAT) methodology can significantly reduce defect levels from scan testing for digital logic. In Section II, we approach CAT using the software-based self-test (SBST) methodology. SBST is extensively used in safety-critical domains (e.g., automotive) as an in-field safety mechanism when hardware approaches are missing or cannot be used. However, the fault grading is typically done on stuck-at and, eventually, on transition delay faults. The work shows, for the first time, CAT fault coverage values that can be reached using test programs developed for other fault models. We collected results on two open-source microcontrollers for automotive, showing that it is possible to achieve coverage values suitable for functional safety requirements.

Section III discusses the conventional memory testing approach and introduces Device-Aware Test (DAT). DAT is a new test approach that incorporates the impact of the physical defect into the technology parameters of the device and, after that, in its electrical parameters. Once the defective electrical model is defined, a systematic fault analysis derives appropriate fault models and test solutions. In this work, we apply DAT to a unique defect in STT-MRAM and a unique defect in RRAM to show the superiority of DAT as compared with the conventional approach.

Process variations during manufacturing are one of the biggest challenges for automotive system-on-chip (SoC) manufacturers. With devices performing slightly differently than their ideal behavior, a complete set of characterization steps must be performed to ensure that each commercialized device is within the specs advertised by its manufacturer. Section IV presents multiple characterization steps performed on an SoC's embedded Flash memory with experimental results from an Aurix 3G™ device made by Infineon™.

The integrated circuit (IC) manufacturing process contains a wide range of possible data collection points. Visual data on the wafer is collected between the different wafer fabrication steps and used to monitor the process. This source of information also presents possibilities to extend their possible use as it includes information for each IC as well. Section V details the new possible insights that become achievable once this data is leveraged. We show that by reshaping the visual data, we can see which IC areas are critical and more prone to cause defective behavior. We will also correlate this to measurements that show an irregular trend.

Table I
OPENMSP40_SAF_STL1: RESULTS ON SUB-MODULES

Module	Silvaco Nangate 45nm library								STMicroelectronics' proprietary 130nm technology							
	SAF		TDF		Stat CAT		Dyn CAT		SAF		TDF		Stat CAT		Dyn CAT	
	Faults	FC%	Faults	FC%	Faults	FC%	Faults	FC%	Faults	FC%	Faults	FC%	Faults	FC%	Faults	FC%
ALU	3,700	98.95	3,700	92.21	5,816	46.99	6,030	35.70	6,158	91.89	6,158	84.55	10,619	44.96	10,293	32.24
Register File	9,538	95.56	8,622	82.43	18,148	39.70	17,288	15.42	10,656	95.02	9,712	82.96	24,901	28.40	29,467	13.12
Frontend	5,616	89.13	5,136	80.14	9,572	63.61	7,307	44.47	6,598	89.09	6,118	78.50	11,281	52.50	11,122	36.24
Memory Backbone	2,210	96.02	2,058	92.17	2,692	62.97	2,435	48.95	2,690	95.97	2,542	90.45	4,025	66.59	3,753	57.80
Multiplier	8,842	94.56	8,558	79.32	16,547	52.56	16,188	2.03	11,928	92.84	11,640	78.21	21,934	48.22	26,582	2.44
(Whole DUT)	33,982	94.67	31,874	83.26	58,190	49.86	54,062	20.65	42,660	93.14	40,540	81.86	80,220	42.10	88,471	18.10

II. ARE STLS SUITABLE FOR DEFECT-ORIENTED FAULT MODELS?

Functional test of integrated circuits is becoming more and more relevant in sectors where in-field test of safety-critical devices is required, e.g., the automotive sector. This is typically carried out through a Software-Based Self-Test (SBST) approach [1], [2], where Self-Test Libraries (STLs), i.e., a suite of test programs, are periodically executed by the device under test (DUT). SBST has been proven effective for both processors [3]–[6] and peripherals [7]–[11]. Since companies developing STLs to comply with functional safety standards like ISO26262 [12] currently focus on the stuck-at fault model, with few academic works focusing on delay faults, there is no evidence about their effectiveness on defect-oriented fault models like cell-aware testing (CAT).

Several articles focusing on cell-aware testing can be found in literature [13]–[16]. These works focus on either proposing new algorithms to optimize cell-aware ATPG performances as in [13], [15], exploiting switch-level ATPGs to generate test patterns for CAT together with algorithms to automatically inject defects into the network to be tested [16], or focusing on intra-cell defects to improve cell-aware test methodologies, such as intra-cell bridging faults [14]. In-field testing of such defects through SBST, however, has never been investigated.

In this work, we aim at understanding if STLs written for other fault models can be effectively reused, given some necessary modifications, for CAT. In the following subsections, we briefly describe the process for the characterization of the CAT fault lists, and experimental results carried on two case studies.

A. Cell-aware characterization flow

The CAT characterization flow aims at identifying intra-cell defects such as shorts, opens and transistor-opens/shorts defects. It is a one-time task performed for each cell of the library to create the cell-aware models. As inputs, the flow needs: the timing information file provided by the technology library; the SPICE models of the library cells; the SPICE netlist of each cell. The flow starts with an analog simulation of the netlist; then, a pattern set is provided to detect the list of the cell-aware faults. Each fault is classified as *static CAT* fault – where the test vector detects a voltage difference during the simulation and it can be seen as a conditional stuck-at fault (SAF) by ATPG-tools – or *dynamic CAT* fault – where pairs of test vectors detect a voltage difference in a specific time delay during the simulation and it can be seen as a conditional transition-delay fault (TDF). As the last step, a cell-aware fault model synthesis is performed to create a cell-aware defect matrix for ATPG and fault-simulation tools. The obtained output is a logic model for CAT.

B. Experimental results

We experimented the effectiveness on CAT of various STLs developed for other fault models. For our analysis, we considered two open-source case studies: the CPU of the 16-bit microcontroller openMSP430 [17], and the execution unit of a 32-bit 4-stage RISC-V processor [18]. We synthesized both cores using the open-source technology library Silvaco Nangate 45nm [19]. Moreover, to study the impact on the fault coverage of the library, we also synthesized the openMSP430 with an STMicroelectronics' proprietary 130nm technology for power applications (hereinafter referred to as ST library).

We extracted the CAT faults using Synopsys CMGen. The experiments required two days for the characterization of 115 out of the 134 standard cells of the Nangate library, and ten days for about 500 cells of the ST library.

The openMSP430's area is 9,304 NAND2 equivalent gates. Table I reports the list of stuck-at faults (SAFs), transition-delay faults (TDFs), and static and dynamic CAT faults present in the main sub-modules; in total, the CPU accounts for 33,982 SAFs in the Nangate netlist and 42,660 in the ST netlist. The RISC-V execution unit's area is 18,904 NAND2 equivalent gates, roughly one forth of the whole CPU area; in terms of faults, it accounts for 72,788 SAFs and TDFs, 177,796 static CAT, and 179,151 dynamic CAT faults.

Tables I and II summarize the results gathered on open-MSP430. Since the 39 STLs were developed to cover stuck-at faults, Table II is ordered by SAF coverage, which goes from around 95% of STL1 to 83% of STL39; the TDF coverage shows a similar decreasing trend, from 83% to 63%, with few exceptions in the monotony. For CAT faults, the coverage values oscillate in the range of 40-50% for static and 10-20% for dynamic CAT, with only STL2 able to reach 27% of dynamic CAT coverage. Interestingly, STL9, which is significantly longer than the others, was the only one able to reach 53% of coverage on static CAT. The last row of the table shows the cumulative fault coverage, when all 39 fault lists are combined; we can cover only 64% and 39% of the modeled static and dynamic CAT faults, respectively.

Table II
RESULTS OF STLS DEVELOPED FOR OPENMSP430 CPU TARGETING SAF

STL	Clock cycles	SAF FC%	TDF FC%	Stat CAT FC%	Dyn CAT FC%
openMSP40_SAF_STL1	118,450	94.67	83.26	49.86	20.65
openMSP40_SAF_STL2	119,402	94.54	82.24	50.09	26.67
openMSP40_SAF_STL3	33,496	93.64	80.00	49.18	18.58
openMSP40_SAF_STL4	19,716	92.84	76.74	45.60	22.62
openMSP40_SAF_STL5	78,612	92.23	77.09	42.80	17.58
openMSP40_SAF_STL6	13,810	91.82	73.48	42.67	15.42
openMSP40_SAF_STL7	2,406	91.47	71.68	37.89	12.67
openMSP40_SAF_STL8	14,642	91.44	77.15	48.86	15.17
openMSP40_SAF_STL9	1,950,510	91.10	79.20	53.39	19.07
openMSP40_SAF_STL10	13,658	91.08	77.86	40.71	16.03
openMSP40_SAF_STL11	10,474	90.83	78.80	43.00	16.60
openMSP40_SAF_STL12	18,992	90.72	74.06	41.62	14.48
openMSP40_SAF_STL13	29,420	90.67	75.78	49.06	16.34
openMSP40_SAF_STL14	30,810	90.53	76.50	48.91	17.17
openMSP40_SAF_STL15	18,430	90.42	73.65	47.26	13.22
openMSP40_SAF_STL16	58,900	90.22	75.77	47.37	14.58
openMSP40_SAF_STL17	37,302	90.13	76.96	37.49	14.58
openMSP40_SAF_STL18	30,316	90.02	74.50	45.74	13.30
openMSP40_SAF_STL19	31,902	89.75	72.41	40.19	13.49
openMSP40_SAF_STL20	16,022	89.67	73.43	40.64	13.78
openMSP40_SAF_STL21	19,378	89.62	75.27	40.80	15.24
openMSP40_SAF_STL22	26,650	89.58	71.79	40.03	13.44
openMSP40_SAF_STL23	56,814	89.53	73.16	43.90	20.51
openMSP40_SAF_STL24	14,864	89.14	69.41	42.95	13.89
openMSP40_SAF_STL25	13,138	88.92	74.01	43.93	15.21
openMSP40_SAF_STL26	18,220	88.06	69.15	42.87	14.77
openMSP40_SAF_STL27	8,788	87.59	69.91	41.66	13.00
openMSP40_SAF_STL28	42,778	87.50	71.52	45.86	15.80
openMSP40_SAF_STL29	17,610	87.46	70.87	49.70	14.32
openMSP40_SAF_STL30	10,134	87.11	72.01	44.78	13.58
openMSP40_SAF_STL31	17,318	85.51	65.67	41.33	14.54
openMSP40_SAF_STL32	13,044	85.29	66.36	43.43	12.88
openMSP40_SAF_STL33	33,178	84.66	66.07	42.97	14.94
openMSP40_SAF_STL34	2,888	84.61	68.92	35.47	11.73
openMSP40_SAF_STL35	5,920	84.46	67.81	43.55	11.88
openMSP40_SAF_STL36	31,828	84.15	64.33	37.68	14.92
openMSP40_SAF_STL37	34,554	83.97	63.49	36.01	14.37
openMSP40_SAF_STL38	5,656	83.81	65.74	36.68	12.30
openMSP40_SAF_STL39	23,498	83.39	63.41	41.52	14.17
Cumulative fault coverage		96.78	90.20	63.92	38.38

Table III
RESULTS OF STLS DEVELOPED FOR RISC-V EXECUTION UNIT TARGETING SAF

STL	Clock cycles	SAF FC%	TDF FC%	Stat CAT FC%	Dyn CAT FC%
riscv_SAF_STL1	98,878	96.15	87.98	59.86	50.92
riscv_SAF_STL2	64,527	95.91	69.32	37.47	26.50
riscv_SAF_STL3	24,164	95.89	83.45	36.68	30.70
riscv_SAF_STL4	308,992	95.87	89.29	55.86	50.35
riscv_SAF_STL5	112,250	95.81	85.22	52.95	46.16
riscv_SAF_STL6	110,604	95.72	85.32	53.68	46.28
riscv_SAF_STL7	118,123	93.99	71.95	44.80	31.08
riscv_SAF_STL8	780,465	93.92	86.09	49.62	46.20
riscv_SAF_STL9	80,441	93.23	77.14	29.89	20.46
riscv_SAF_STL10	17,294	90.79	76.98	38.02	25.51
riscv_SAF_STL11	43,583	89.52	68.84	32.06	22.75
riscv_SAF_STL12	36,419	87.04	30.58	49.12	13.73
Cumulative fault coverage		98.53	95.31	65.58	58.99

Table IV
RESULTS OF STLS DEVELOPED FOR RISC-V EXECUTION UNIT TARGETING TDF

STL	Clock cycles	SAF FC%	TDF FC%	Stat CAT FC%	Dyn CAT FC%
riscv_TDF_STL1	439,962	95.80	92.94	74.62	81.66
riscv_TDF_STL2	40,766	95.40	92.54	53.95	47.38
riscv_TDF_STL3	32,931	93.12	88.91	66.13	59.99
riscv_TDF_STL4	79,160	91.64	87.20	73.14	66.80
riscv_TDF_STL5	28,173	91.43	86.68	40.67	37.01
riscv_TDF_STL6	23,709	92.35	86.05	31.52	26.04
riscv_TDF_STL7	14,057	90.22	84.20	65.47	52.83
riscv_TDF_STL8	13,278	91.06	83.42	55.29	47.07
riscv_TDF_STL9	42,849	90.85	82.08	34.81	29.08
riscv_TDF_STL10	12,633	90.86	81.67	33.01	27.91
riscv_TDF_STL11	8,596	88.11	80.83	70.08	61.66
riscv_TDF_STL12	38,178	93.71	80.44	50.94	38.24
riscv_TDF_STL13	12,031	90.01	80.20	33.52	28.44
riscv_TDF_STL14	47,480	91.58	79.54	36.39	26.14
riscv_TDF_STL15	573,169	95.01	70.82	85.79	69.55
Cumulative fault coverage		96.76	95.07	87.24	85.08

When focusing on the sub-modules (see Table I), one can notice that the modules Frontend and Memory Backbone present a lower CAT coverage loss – with respect to SAF and TDF – than the register file and ALU/multiplier. We observed a similar behavior in all the available STLs. To show that the behavior is not library-dependent, the table also reports the results gathered on the netlist based on the ST library.

We conducted further experiments on the RISC-V execution unit to verify whether or not the results are specific of the openMSP430 core (see Tables III and IV). In the same test conditions (i.e., using various STLs developed for SAF, and able to cover more than 80% of the target faults), the CAT coverage values oscillate from a minimum of 32% up to 60% for static, and from 14% up to 51% for dynamic CAT. These values are slightly better compared to openMSP430; however, such a coverage is still low in absolute value. Finally, we run another set of fault simulations using a second set of STLs, developed for TDFs. Interestingly, the CAT coverage values are better compared to the ones discussed above (see Table IV, where STLs are ordered by TDF coverage). In one case (i.e., STL15), the static CAT coverage is above 85%, while the dynamic CAT coverage is around 82% (reached by STL1). When combining all of the STLs in the table, more than 85% of both static and dynamic CAT faults is covered, compared to 66% and 59% using all SAF-oriented STLs combined.

C. Final remarks

This paper reported for the first time an experimental evaluation aimed at assessing the effectiveness in detecting cell-aware faults via Self-Test Libraries. We have shown that STLs able to reach more than 90% SAF coverage do not easily cover more than 60% of the static (and even less of the dynamic) CAT faults. However, when focusing on maximizing the TDF coverage, there is a significant gain in the CAT coverage, with more than 85% of CAT faults covered.

Future works will include the evaluation of functionally untestable CAT faults, and the definition of STL refinement strategies to improve CAT coverage.

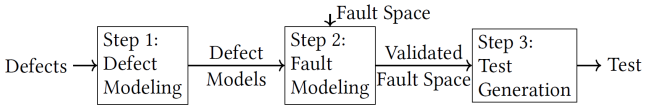


Figure 1. Device-Aware Test flow [25].

III. Device-Aware Test: A Means to Win the War Against Unmodelled Defects

Testing memories went through a long revolution. The early tests (typically before 1980) can be classified as ad-hoc tests due to the absence of formal models and proofs [20]. During the early 1980s, many memory fault models have been introduced, allowing the fault coverage of test algorithms (typically march tests) to be evaluated while the test time is usually linear with the size of the memory; these fault models were not based on any actual memory design. In the late 1990s, experimental results based on results of a large number of tests applied to a large number of memory chips indicated that many detected faults cannot be explained with the well-known fault models [21], which suggested the existence of additional faults. This stimulated the introduction of new fault models (both static and dynamic) based on linear resistor defect injection and SPICE simulation [22]. This *conventional memory test approach* (actually also logic) assumes that physical defects in devices can be be modeled as *linear resistors*. Although it can be convincing for modeling opens and shorts in interconnects, this assumption has never been validated for devices. Moreover, emerging non-volatile devices such as STT-MRAM and RRAM are non-linear analog devices for which modeling a device defect with just a linear resistor is far from reality; not to mention the fact that they are facing specific test challenges [23]. This was the inspiration behind the introduction of a new approach called *Device-Aware Test (DAT)* [24].

Next we will briefly explain the fundamentals of DAT, thereafter apply it to a unique defect in STT-MRAM and a unique defect in RRAM to show the superiority of DAT as compared with the conventional approach.

A. Fundamentals of Device-Aware Test

Fig. 1 shows the DAT approach that consists of three steps [24], [25]. **1) Defect Modeling:** This step is the core of the DAT approach. Here, the *actual* physics of a defect are modeled in order to estimate its effects on the technology parameters of a device (e.g., length and width) . Subsequently, the effective technology parameters are included in an electrical compact model and their impact on the electrical parameters is derived. After calibration, the model can be used in circuit simulations. Because the real physics of the defect are included in the defect modeling, subsequent fault modeling and analysis will result in realistic faults and thus in accurate test solutions. **2) Fault Modeling:** Here, the defect models are used to perform fault modeling to validate the fault space and to obtain a list of faults that can actually occur in the circuit. Faults in a memory are typically described using the fault primitive (FP) concept as $\langle \mathrm{S/F/R} \rangle$ [25], [26]. Here, S denotes the sensitizing sequence that is applied, e.g., $0\mathrm{w}1$, or $1\mathrm{r}1$. F denotes the state of the cell after S is completed. In emerging memories, there exist more states than just '1' and '0' [25]. Therefore, $\mathrm{F} \in \{1, 0, \mathrm{U}, \mathrm{L}, \mathrm{H}\}$. R denotes the output of the sense amplifiers, if the last operation in S was a read operation. R can be 0, 1, or ? when the output is uncertain and randomly 0 or 1. When the last operation is a write operation $\mathrm{R} = -$. Faults can be classified as easy-to-detect (EtD) if they are guaranteed to be detected by regular memory operations (e.g., $\langle 0\mathrm{w}1/0/- \rangle$), or hard-to-detect (HtD) if this guarantee is not possible (e.g., $\langle 0\mathrm{r}0/\mathrm{U}/0 \rangle$). **3) Test Development:** Here, a test that detects the faults in the validated fault space is developed. This can be a march test, but also other specialized tests are possible, e.g., stress tests to detect the HtD faults.

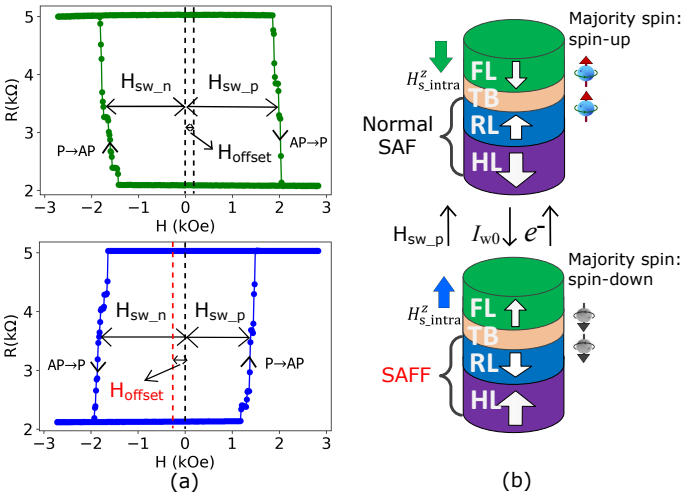


Figure 2. Defect-free MTJ (upper) versus a SAFF-defective MTJ (lower) with ⌀=55 nm: (a) R-H loop, (b) schematic of AP state.

B. DAT for STT-MRAMs

With comprehensive characterization on MTJs (STT-MRAM data-storing devices) with diameters ranging from 35 nm to 175 nm on different wafers, we observed that some defective devices have horizontally flipped R-H loops (see Figure 2a) but normal R-V loops. This suggests that the polarity of the stray field ($H^z_{\mathrm{s\_intra}}$) at the free layer of MTJ reverses when compared to defect-free devices. We attribute the root cause to the flip of magnetization in both hard layer and reference layer (see Figure 2b), which we name as *synthetic anti-ferromagnet flip* (SAFF) defects [27]. Next, we apply the three-step DAT approach to the SAFF defects and compare the results with that of conventional resistor-based approach.

1) Device-Aware Defect Modeling for SAFF Defects: We first physically model the effect of SAFF defect on the intra- and inter-cell stray fields at the free layer of the defective cell within a memory array. Thereafter, its impact is incorporated into two electrical parameters: the critical switching current I_{c} and the average switching time t_{w}. Finally, we calibrate the SAFF-defective MTJ compact model with measured data.

2) Device-Aware Fault Modeling for SAFF Defects: We use our SAFF-defective MTJ model and linear resistor model to

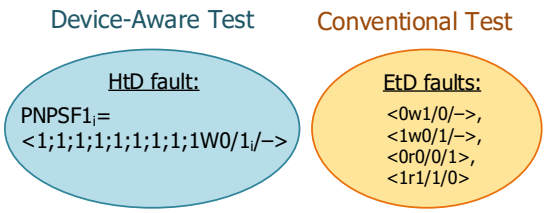


Figure 3. Comparison of sensitized FPs due to SAFF defects: device-aware test vs. conventional test based on linear resistors.

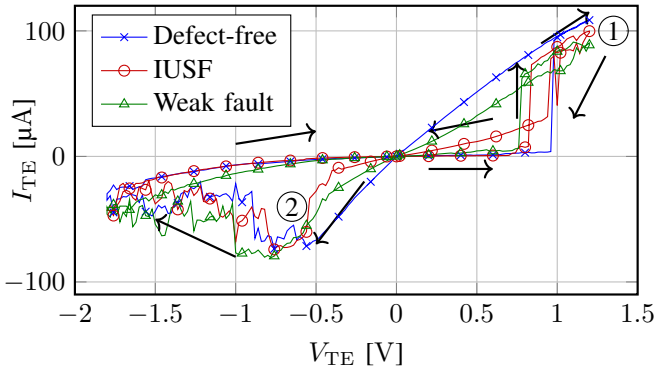


Figure 4. Defect-free, faulty, and weak devices [29].

perform circuit simulations and fault analyses; the results are shown in Figure 3. It can be seen that a SAFF defect leads to a HtD fault: *intermittent passive neighborhood pattern sensitive fault* ($\mathrm{PNPSF1_i}$) using our DAT approach. This cannot be obtained by the conventional fault modeling approach where a linear resistor is injected in parallel with or in series with an ideal defect-free MTJ device. In contrast, the conventional approach results in four EtD faults, as shown in the figure. This indicates that these four faults are not qualified to cover SAFF defects in STT-MRAMs. Accordingly, the March tests targeting these four EtD faults obviously cannot guarantee the defection of SAFF defects.

3) Device-Aware Test Development for SAFF Defects: Based on the fault modeling results, we propose a magnetic march test, which is the first in the test community. It aims at guaranteeing the detection by incorporating *magnetic* write operations in the March test [28]:

$$\{\Updownarrow(\mathrm{w0_H});\Updownarrow(\mathrm{r0})\} \text{ or } \{\Updownarrow(\mathrm{w1_H});\Updownarrow(\mathrm{r1})\}.$$

Here, the first element $\mathrm{w0_H}$ ($\mathrm{w1_H}$) indicates a magnetic write '0' ('1') operation; i.e., an external field H_{ext} is applied to switch the MTJ state rather than driving an electric current through the MTJ device.

C. DAT for RRAMs

We measured the electrical characteristics of 1T1R RRAM devices manufactured at ST Microelectronics. An illustrative measured I-V graph of a defect-free device is shown in Fig. 4 (blue line) [29]. The switching in a nominal defect-free device is *bipolar*, where logic '1' is represented by the SET state, and logic '0' by the RESET state. The range between '1'and

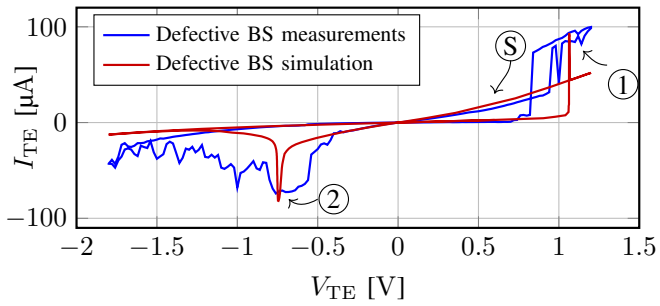


Figure 5. Calibrated IUSF defect model [29].

'0' is called an *undefined state* ('U'). Some devices show a strange switching characteristic. After a number of cycles, the switching resembles complementary switching (CS) instead of bipolar switching (BS) for a couple of up to $1.068\,\%$ of the cycles, i.e., the fault is intermittent. The observed undesired CS in all faulty devices can be classified into two groups: *faulty*, where the device switches into 'U', as shown in Fig. 4 (red line), and *weak* devices, where the device is in a disturbed '1' state, as shown in Fig. 4 (green line). We identified this fault as *intermittent undefined state fault* (IUSF) [29]. The cause of the faulty behavior is attributed to a reduced binding capacity of the capping layer for oxygen ions, which can be caused by over-forming or low doping of the capping layer [29]–[31].

1) Device-Aware Defect Modeling: In this step, we develop a compact defect model that accurately describes the physics of the fault. We modify the physics-based $\mathrm{HfO_x}$ RRAM model for CS, JART VCM v2, from [32], which describes CS as the exchange of oxygen vacancies between two regions in the oxide. We adapt the model so that region 1 becomes the BS filament, and that the switching depends solely on this region. Then, we include the unwanted CS by changing the initial oxygen vacancy concentration in region 1 (parameter N_{init1}), and the maximal vacancy concentration in the oxide (parameter N_{max}). The ratio $N_{\mathrm{init1}}/N_{\mathrm{max}}$ determines the number of vacancies that can be in region 2; a lower ratio leads to a stronger CS effect. Next, we include the parameters in the model and calibrate them to match the measurement data, as shown in Fig. 5.

2) Device-Aware Fault Modeling: We include the obtained defect model in a circuit simulation and perform fault analysis. Depending on the ratio $N_{\mathrm{init1}}/N_{\mathrm{max}}$, there are two faults sensitized: the strong HtD <0w1/U/-> fault, and the weak HtD disturbed '1' fault. The traditional defect modeling approach, where linear resistors are put in series or in parallel with a RRAM device, is unable to show the dynamics of the IUSF, and thus, a test based on these models will fail to detect it.

3) Device-Aware Test Development: Due to the nature of the faults being intermittent and causing the cell to switch into 'U', a regular march test cannot guarantee the detection of such a fault. However, it is possible that it will *probabilistically* detect the fault; reading the cell in the 'U' state will sometimes result in '1' and sometimes in '0'. The following march algorithm can be used:

$$\text{March-IUSF} = \left\{\Updownarrow(\mathrm{w0},\mathrm{w1},\mathrm{r1})^k\right\},$$

where k indicates the number of times the sequence is applied. If we assume that reading a cell in 'U' state results the same probability of getting '1' or '0' (i.e., $50\,\%$), and that the probability of IUSF is P_{IUSF}, then the detection probability is: P_{d}=$1-(1-P_{\mathrm{IUSF}}\cdot 50\,\%)^{k}$. When P_{IUSF}=$1.068\,\%$ and $k=560$, then the fault coverage (FC) FC=$95\,\%$, and when $k=1291$, then FC=$99.9\,\%$. Hence, realizing high FC needs a long test time and is detrimental to the device endurance. The detection capabilities can be improved, e.g., by changing the SA reference so that it can distinguish between '1' and 'U', rather than between '1' and '0'.

IV. Read DfT Pre-Characterization/centering for non-volatile memories SoC

Automotive Systems-on-Chip (SoCs) Manufacturers are constantly developing new and advanced embedded memory products to improve their lineup. In this contest, the rising densities of non-volatile memories (NVM) bring severe concerns about the robustness and centering of memory cells belonging to vast arrays.

Due to process variations, each component of an SoC may perform slightly differently from its ideal behavior. For this reason designers take countermeasures to ensure that their product works as expected, at least in the vast majority of cases. Such countermeasure can be "chronologically" divided as follows:

During the design phase:

- multiple simulations are run to understand the behavior of the circuits in case of slight deviation in the component characteristics such as slower transistors, different values for capacitors, and so on [33];
- additional circuitry may be included to compensate for faulty circuit logic or storage, e.g., duplication, triplication techniques, or Error Correction Codes (ECC).

After production and during the testing phase:

- it is possible to fine-tune analog components such as Voltage Controlled Oscillator, Voltage regulators, and reference current sources;
- embedded memories are repaired with the help of spare redundancy elements.

Apart from the simulation steps performed during the project's early development, a complete set of test suites is necessary to ensure that each produced device complies with the marketed specifications and works at the optimal operating point.

When a novel technology is used to manufacture a chip, old tests developed for older devices cannot be used as plug-and-play solutions. Moreover, manufacturers should also be aware of the risks of "over-testing" and "under-testing", as testing the devices while imposing too weak or too tight testing constraints may cause a correctly working product to be marked as defective and to be discarded accordingly or vice-versa. As the manufacturers' goal is to maximize yield, it is essential to perform multiple characterization studies on a representative number of devices to evaluate the performance and endurance of every component of the new products.

This work is focused on one of the crucial components of an SoC, the embedded eFlash memory (eFlash), for which we describe some characterization efforts devised to assess their testing flow as also seen in [34]. In particular, we illustrate how we collected measures about Erased Cells' current distribution and read access time, which is helpful to characterize the memory cell limits to be considered along the testing phases of the product.

The presented strategies and results refer to devices of the Aurix 3G™ family made by Infineon™.

A. Background

An eFlash is made of a matrix of cells, each containing a single bit. This matrix is organized in columns named bitlines and rows called wordlines that are further divided into pages. From a user perspective, pages represent the minimum granularity of the memory: to access a single bit the user has to access the entire page it belongs to.

Testing Flash memories is particularly time-consuming because Flash storage is highly dense, abundant, and amounts to a large area of the die. The typical testing flow checks the hardware's analog behavior (voltages, currents, oscillators' frequency) and logic functionality.

B. Erased cells current distribution

For each bitline in the memory, a sense amplifier performs a comparison between a given reference current and the current generated by the selected bit cell. If the current generated by the selected bit cell is higher than the reference one, then the bit is decoded as a 0. Vice-versa, if this current is lower, it is decoded as a 1. This mechanism is explained in Fig. 6. This current measure may vary from cell to cell. Therefore, performing and analyzing the overall cell matrix values and distribution is essential.

Identifying the most appropriate reference current to use during volume testing is one of the characterization steps performed on the eFlash. The test of the current observed from each memory cell after an erase or a program operation provides valuable information to set the reference current limit for an erased or programmed cell.

The result of the current tests are collected through stacked bitmaps classically collected through bitmapping techniques [35] or with more advanced shape recognition techniques [36]. Fig. 7 shows an example of a stacked bitmap, showing the current distribution in an erased bank.

As it can be seen in Fig. 7, the current distribution is strongly position-dependent. With "weaker" cells concentrated in specific areas such as the middle part of the top half of Fig. 7. Manufacturers can then use the information coming from these stacked bitmaps to set the lower limit for their sense amplifier reference currents. Erased cells that exhibit a current lower than this limit will be marked as defective.

C. Read Access time characterization

Another crucial aspect for manufacturers is the read access time for the eFlash memories. The faster it is, the higher

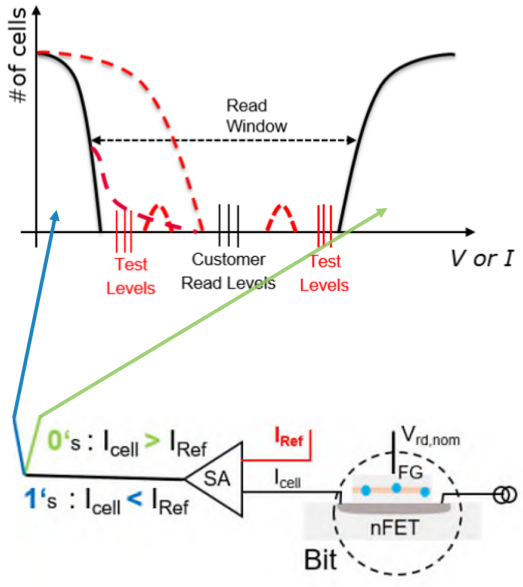


Figure 6. Bit cell decoding mechanism; a sense amplifier

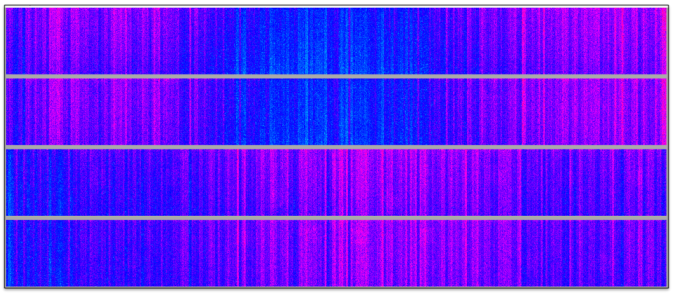


Figure 7. Stacked bitmap showing the current distribution of erased cells in a bank. The more violet the color, the higher the current of the cell

the performance of the SoC they are part of. The lower limit for the access time is established by multiple factors such as wires and sense amplifiers delays. However, also these parameters show some process variations together with temperature and voltage dependence and need to be assessed during the characterization of the devices.

To perform this characterization, the manufacturers can act on different aspects of their memory read, such as:

- the supply voltage of the eFlash modules;
- the temperature at which the characterization is performed;
- the system frequency of the SoC they are part of.

The following example in Fig. 8 shows the effect of changing the system frequency and the reference voltage on the eFlash behavior when operating at room temperature. Along with these eFlash tests, performed by stepping through growing voltage and frequency values, the interesting parameter is the point at which faults start to appear.

In Fig. 8 the horizontal axis represents the frequency at which the SoC is working: faster frequency means less time for the memories to return their read. The vertical axis represents the percentage of failings over the total amount of tested bits.

As emerging from Fig. 8, the voltage has a clear effect on the performance of the eFlash. The lower the voltage, the sooner the faults start to appear. On the other hand, by focusing on the frequency, the figure shows that at some critical points, the eFlash has not enough time to decode the bits correctly, and failures start to appear.

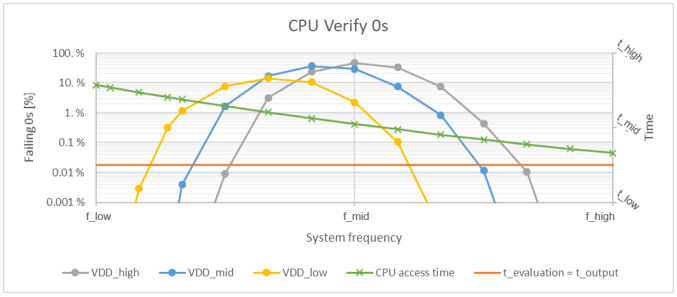


Figure 8. Example of eFlash characterization tests performed at room temperature

Most manufacturers also include additional hardware components called Memory Built-in-Self-Test (MBIST) in their eFlash to speed up the testing procedures. Nevertheless, also the performance of these devices has to be characterized. Fig. 9 shows the results of these tests in case of a verify 0s on a memory composed of all erased cells. Similarly, Fig. 10 shows a verify 1s over a memory composed of all programmed cells.

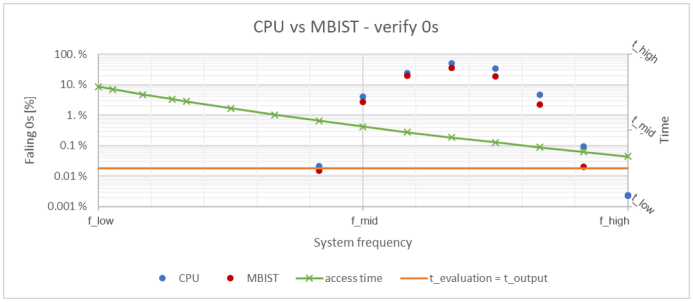


Figure 9. Test to check the effectiveness of the integrated MBIST against the CPU in case of a verify 0s

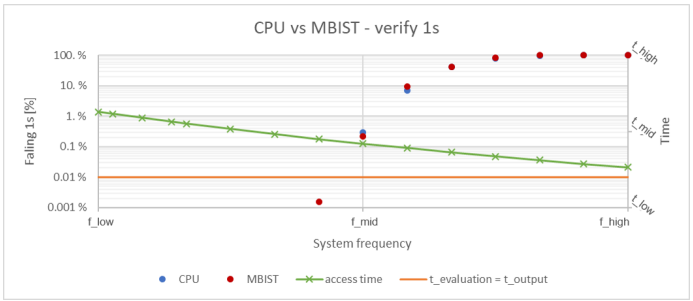


Figure 10. Test to check the effectiveness of the integrated MBIST against the CPU in case of a verify 1s

The last two figures show that the CPU and MBIST results are almost identical. The comparison between CPU and MBIST shows that the specialized hardware can be reliably operated in the testing environment, with all the advantages in test time it brings.

D. Conclusions

In conclusion, this work showed a brief overview of the characterization steps that SoC manufacturers perform on testing their embedded memories with actual test results coming from devices made by Infineon™.

V. Exploiting visual data to boost testing methodologies

As automotive testing requirements continue to become more stringent, novel viewpoints and additional sources of information are needed to further improve automotive IC reliability. The use of visual inspection information from wafers obtained during fabrication between individual process steps opens new possibilities to improve testing methodologies. The data can lead to more efficient and focused test development, as well as open up new ways to detect defects that threaten reliability. Using such visual inspection data, a novel analysis technique is proposed that searches for peaks in defect densities at different sites of ICs and that links this to measurements in order to determine which tests to enhance or which functional block on the IC to test more. From this analysis and depending on the IC that is analyzed, it is possible to show which tests are most important and even to generate a first-order prediction for measurements exhibiting anomalous behavior.

A. Defects on Wafers

The demands on IC testing in the automotive industry continue to tighten even more. Therefore, continuously extra techniques are required to further establish and enhance IC testing proficiency moving forward. The further increase in the number of ICs used in everyday vehicles, results in more stringent specifications to ensure safe daily operations. At the same time, cost limitations limit how far and how thorough testing can be used and adjusted. Ensuring correct IC operation within specification during the targeted lifetime is becoming an important requirement that needs to be explicitly verified for safety-critical applications. ISO26262 incorporates this in some way, but a more comprehensive methodology needs to be developed to ensure safe operation during lifetime [12].

In this paper we will introduce a new source of data that can aid greatly towards developing improved testing schemes. Using the additional data source of visual/optical inspection data taken from the wafers at different stages (layers) of the production process, it is possible to develop novel approaches to test development.

B. Restructuring the Visual Inspection Data to Extract all Knowledge

Figure 11 shows a defect wafer map as captured by a surfscan equipment in the wafer fab. Even though defects seem to be randomly located, which most of them are, it is possible to link certain defects to certain areas on the die of the ICs. The data are gathered using specific machines in between the different processing steps during IC production. Their usual function is to keep track of the process itself, but data linkage tools make it possible to use that information for much broader purposes, in this case towards improving testing methodologies.

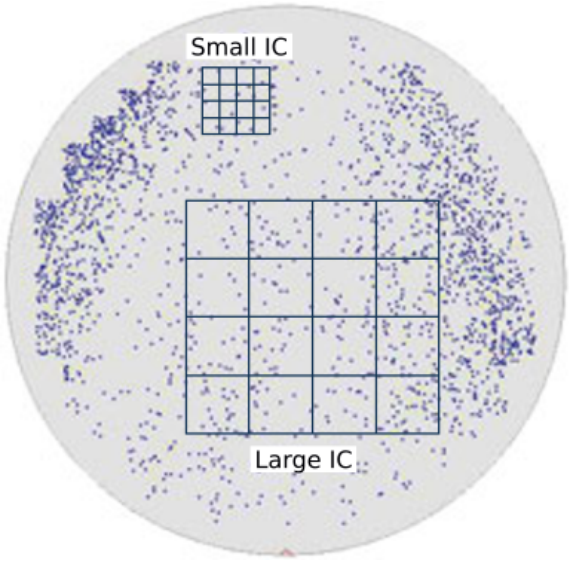


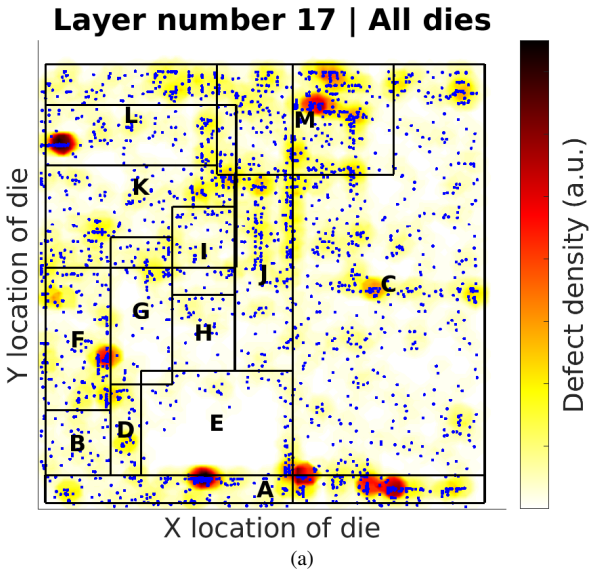
Figure 11. Example of wafer with ICs of different size and their impact with regards to defect locations. Taken from [37].

Previous work has shown that, through statistical methods, it is possible to screen out latent defects using visual data [38]. The technique builds upon DPAT to enhance outlier detection. This new technique, called VEDPAT, uses statistical properties of the visual data to enhance DPAT and to flag latent defects, while having minimal false positive flagging of chips and, hence, low yield loss.

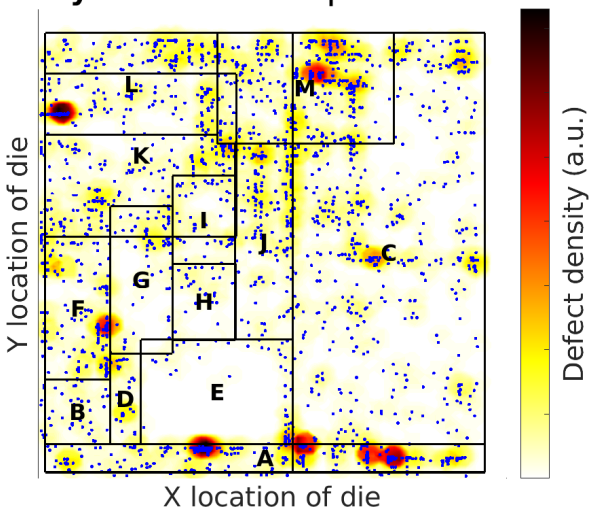
While the technique from [38] is using visual data to build general tests, it is also possible to analyse the data provided by visual inspection in order to improve testing of specific areas on the die of each IC. The proposed analysis uses heatmaps that show the density of spotted defects; examples are shown in Figures 12(a), 12(b) and 12(c). The first heatmap, Figure 12(a), shows an aggregation of spotted defects and their locations on the die for some layer (number 17 in our example case) of the process for 25 wafers. The die itself can be divided into several functional blocks (labeled from A to M) based on the layout. Here it can already be noted that certain locations form defect hot spots (darker color on the scale). For example, blocks A, C and M in Figure 12(a) show multiple locations with many possible defects.

To enable deeper and more extensive analysis, the data are split into dies that passed (Figure 12(b)) and failed (Figure 12(c)) the conventional electrical wafer tests. The good dies in Figure 12(b) show similar behavior, i.e. the hotspots remain consistent. For the bad dies, however, new hotspots appear in Figure 12(c). Blocks E, K and J contain these new hotspots for the failed dies, as can be seen, while some previous hotspots remain. For each hotspot, we can find all dies that pass all tests and check which of their measurements show the most anomalous behavior.

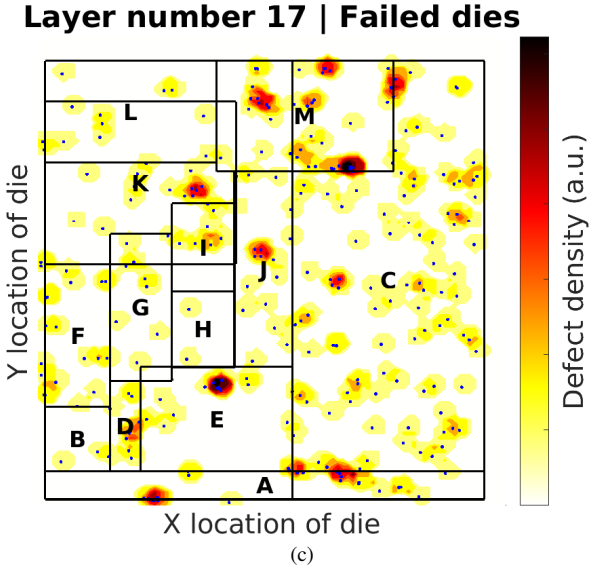
This analysis can be extended further by investigating the root cause of these hotspots and checking which structures



(a)



(b)



(c)

Figure 12. Heatmap of (a) each defect spotted after fabrication step 17 by visual inspection; (b) defects within dies that passed all electrical tests; and (c) defects from dies that failed during testing.

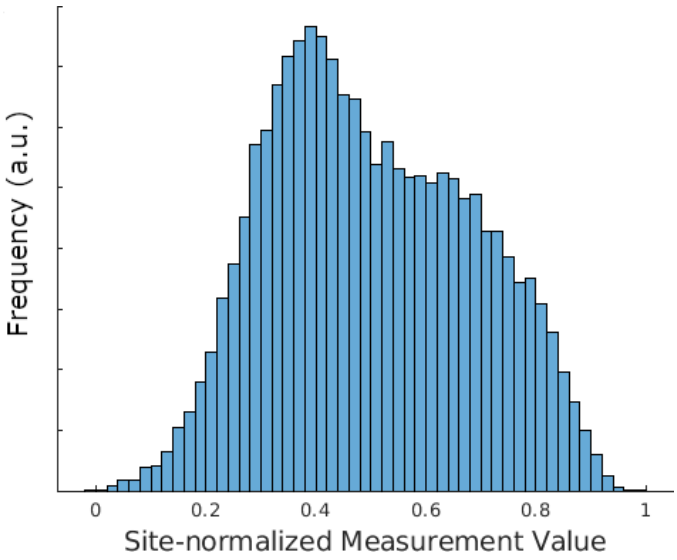


Figure 13. Histogram of the normalized measurement of the dies that passed all the tests but that have visual defects in hotspot areas. It shows that part of the measurements are skewed.

are present there for the considered layer and which test measurements show anomalous behavior. From this analysis, extra tests can be created and/or adapted to improve testing these structures. For example, if there are hotspots for a certain metal layer in a block, those connections can be stressed more during testing, when visual inspection shows as inclination for defects around those hotspots. By looking at Figure 12(c) we choose block E as a sensitive block and calculate which are the key measurements, for example by taking the mean of the measurements of the good dies that contain a failure in block E. The resulting histogram from one of these key measurements is shown in Figure 13, where the measurements have been normalized at the site level, removing any potential bias due to fabrication. Here, only dies with visual defects in hotspot areas (of bad dies) are shown. We can see that the distribution is not entirely symmetric, showing a skewed bias to the right. In this way, tests with higher potential importance can be singled out.

It is also possible to extend this analysis to include high-temperature and low-temperature testing stages that are performed after die packaging. Dies that have passed the initial test stage are first packaged before proceeding with temperature testing. When analyzing the results of tests in these later stages, we see that failed dies that contain some visual defect in the neighborhood of a defect hotspot (of bad dies), relative to the total population of failed dies occur three times more frequently than passed good dies with visual defects in these same areas (relative to the total population of good dies).

C. Visual Inspection Conclusions

Novel methods have been presented to use data from visual inspection taken in between IC processing steps beyond the initial use of process monitoring. These data can be used well towards test methodology development and improvement, aiming for improved defect detection. Firstly, the information

from visual inspection is used to enhance statistical methods currently used. In particular, VEDPAT has been developed to enhance DPAT with the new information. This results in the detection of latent defects at test time and, thus, the prevention of potential failures in the field. Secondly, using the visual pattern information defect hotspots that cause many ICs to fail the electrical tests have been identified. This has been extended to later stages of temperature test failures that show relatively higher occurrence of visual defects. Furthermore, key tests can be pinpointed that should be augmented or extended in order to improve the detectability of potential field failures at IC test time.

ACKNOWLEDGMENTS

The work has been supported by the Center for Automotive Research and Sustainable mobility@PoliTO (CARS).

REFERENCES

[1] M. Psarakis, D. Gizopoulos, M. Hatzimihail, A. Paschalis, A. Raghunathan, and S. Ravi, "Systematic software-based self-test for pipelined processors," in *2006 43rd ACM/IEEE Design Automation Conference*, 2006.

[2] M. Psarakis, D. Gizopoulos, E. Sanchez, and M. Sonza Reorda, "Microprocessor Software-Based Self-Testing," *IEEE Design & Test of Computers*, 2010.

[3] R. Cantoro, S. Carbonara, A. Floridia, E. Sanchez, M. Sonza Reorda, and J. Mess, "An analysis of test solutions for COTS-based systems in space applications," in *2018 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, Oct. 2018.

[4] A. Riefert, R. Cantoro, M. Sauer, M. Sonza Reorda, and B. Becker, "A flexible framework for the automatic generation of sbst programs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2016.

[5] R. Cantoro, D. Foti, S. Sartoni, M. Sonza Reorda, L. Anghel, and M. Portolan, "New Perspectives on Core In-field Path Delay Test," in *IEEE ITC*, 2020.

[6] R. Cantoro, P. Girard, R. Masante, S. Sartoni, M. Sonza Reorda, and A. Virazel, "Self-test libraries analysis for pipelined processors transition fault coverage improvement," in *2021 IEEE 27th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, 2021.

[7] A. Apostolakis, D. Gizopoulos, M. Psarakis, D. Ravotto, and M. Sonza Reorda, "Test program generation for communication peripherals in processor-based soc devices," *IEEE Design Test of Computers*, Mar. 2009.

[8] R. Cantoro, S. Sartoni, and M. Sonza Reorda, "In-field functional test of can bus controllers," in *2020 IEEE 38th VLSI Test Symposium (VTS)*, 2020.

[9] L. Bolzani, E. Sanchez, M. Schillaci, M. Sonza Reorda, and G. Squillero, "An automated methodology for cogeneration of test blocks for peripheral cores," in *13th IEEE International On-Line Testing Symposium (IOLTS 2007)*, Jul. 2007.

[10] A. van de Goor, G. Gaydadjiev, and S. Hamdioui, "Memory testing with a risc microcontroller," in *2010 Design, Automation Test in Europe Conference Exhibition (DATE 2010)*, Mar. 2010.

[11] A. Cook, D. Ull, M. Elm, H. Wunderlich, H. Randoll, and S. Döhren, "Reuse of structural volume test methods for in-system testing of automotive asics," in *2012 IEEE 21st Asian Test Symposium*, 2012.

[12] "ISO 26262-1..12:2018:Road vehicles - Functional safety - Part 1-12," en, International Organization for Standardization, Standard, 2018. [Online]. Available: https://www.iso.org/standard/68383.html.

[13] Z. Gao, M.-C. Hu, J. Swenton, S. Malagi, J. Huisken, K. Goossens, and E. J. Marinissen, "Optimization of cell-aware atpg results by manipulating library cells' defect detection matrices," in *2019 IEEE International Test Conference in Asia (ITC-Asia)*, 2019.

[14] D.-Z. Lee, Y.-Y. Chen, K.-C. Wu, and M. C.-T. Chao, "Improving cell-aware test for intra-cell short defects," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2022.

[15] F. Hapke, R. Krenz-Baath, A. Glowatz, J. Schloeffel, H. Hashempour, S. Eichenberger, C. Hora, and D. Adolfsson, "Defect-oriented cell-aware atpg and fault simulation for industrial cell libraries and designs," in *2009 International Test Conference*, 2009.

[16] P.-Y. Chuang, C.-W. Wu, and H. H. Chen, "Cell-aware test generation time reduction by using switch-level atpg," in *2017 International Test Conference in Asia (ITC-Asia)*, 2017.

[17] O. Girard, *openMSP430*, 2009. [Online]. Available: https://opencores.org/projects/openmsp430 (visited on 08/01/2022).

[18] ETH Zurich and Università di Bologna, *PULP Platform*. [Online]. Available: https : / / github . com / pulp - platform / pulp/ (visited on 08/01/2022).

[19] Silvaco, *Silvaco 45nm FreePDK Library*. [Online]. Available: https://si2.org/open-cell-library/ (visited on 08/01/2022).

[20] M. A. Breuer, A. D. Friedman, *et al.*, *Diagnosis & reliable design of digital systems*. Springer, 1976.

[21] I. Schanstra and A. J. Van De Goor, "Industrial evaluation of stress combinations for march tests applied to srams," in *International Test Conference 1999. Proceedings*, IEEE, 1999.

[22] S. Hamdioui and A. J. Van De Goor, "An experimental analysis of spot defects in srams: Realistic fault models and tests," in *Proceedings of the Ninth Asian Test Symposium*, IEEE, 2000.

[23] E. I. Vatajelu, P. Prinetto, M. Taouil, and S. Hamdioui, "Challenges and solutions in emerging memory testing," *IEEE Transactions on Emerging Topics in Computing*, 2017.

[24] M. Taouil and S. Hamdioui, "Device aware test for memory units," European pat., Sep. 3, 2019.

[25] M. Fieback, L. Wu, G. C. Medeiros, H. Aziza, S. Rao, E. J. Marinissen, M. Taouil, and S. Hamdioui, "Device-aware test: A new test approach towards dppb level," in *2019 IEEE International Test Conference (ITC)*, 2019.

[26] A. J. van de Goor, *Testing Semiconductor Memories - Theory and Practice*. John Wiley & Sons, 1991.

[27] L. Wu, S. Rao, M. Taouil, E. J. Marinissen, G. S. Kar, and S. Hamdioui, "Characterization, modeling and test of synthetic anti-ferromagnet flip defect in STT-MRAMs," in *Int. Test Conf.*, Nov. 2020.

[28] L. Wu, S. Rao, M. Taouil, E. J. Marinissen, G. Sankar Kar, and S. Hamdioui, "Testing STT-MRAM: Manufacturing defects, fault models, and test solutions," in *Int. Test Conf.*, Oct. 2021.

[29] M. Fieback, G. C. Medeiros, A. Gebregiorgis, H. Aziza, M. Taouil, and S. Hamdioui, "Intermittent undefined state fault in rrams," in *2021 IEEE European Test Symposium (ETS)*, 2021.

[30] F. Nardi, S. Balatti, S. Larentis, D. C. Gilmer, and D. Ielmini, "Complementary switching in oxide-based bipolar resistive-switching random memory," *IEEE TED*, 2013.

[31] X. Chen, W. Hu, Y. Li, S. Wu, and D. Bao, "Complementary resistive switching behaviors evolved from bipolar TiN/HfO2/Pt device," *APL*, 2016. [Online]. Available: http : // aip . scitation . org / doi / 10 . 1063 / 1 . 4941287.

[32] C. La Torre, A. F. Zurhelle, and S. Menzel, "Compact modelling of resistive switching devices based on the valence change mechanism," in *SISPAD*, 2019.

[33] A. Michard, F. Cacho, D. Celeste, and X. Federspiel, "Global and local process variation simulations in design for reliability approach," in *2019 IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, 2019.

[34] L. D. Abbati *et al.*, "Industrial best practice: Cases of study by automotive chip- makers," in *2021 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, 2021.

[35] A. L. Landzberg and R. Van Nostrand, *Microelectronics Manufacturing Diagnostics Handbook*. 1993.

[36] P. Bernardi, G. Insinga, G. Paganini, R. Cantoro, P. Beer, M. Coppetta, N. Mautone, G. Carnevale, P. Scaramuzza, and R. Ullmann, "Optimized diagnostic strategy for embedded memories of automotive systems-on-chip," in *2022 IEEE European Test Symposium (ETS)*, 2022.

[37] W. Dobbelaere, *Lecture on Testing of Mixed-Signal Integrated Circuits*, Guest Lecture, Nov. 2020.

[38] A. Coyette, W. Dobbelaere, R. Vanhooren, N. Xama, J. Gomez, and G. Gielen, "Latent Defect Screening with Visually-Enhanced Dynamic Part Average Testing," in *2020 IEEE European Test Symposium (ETS)*, 2020.