A Fast and Low Cost Embedded Test Solution for CMOS Image Sensors

Julia Lefèvre, Philippe Debaud, Patrick Girard, Arnaud Virazel

▶ **To cite this version:**

Julia Lefèvre, Philippe Debaud, Patrick Girard, Arnaud Virazel. A Fast and Low Cost Embedded Test Solution for CMOS Image Sensors. ITC 2021 - IEEE International Test Conference, Oct 2021, Online, United States. pp.1-9, 10.1109/ITC50571.2021.00007 . lirmm-03377562

HAL Id: lirmm-03377562

https://hal-lirmm.ccsd.cnrs.fr/lirmm-03377562v1

Submitted on 14 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Fast and Low Cost Embedded Test Solution for CMOS Image Sensors

J. Lefevre [1, 2] P. Debaud [1] P. Girard [2] A. Virazel [2]

[1] STMicroelectronics
Imaging division, DFT
Grenoble, France
julia.lefevre@st.com, philippe.debaud@st.com

[2] LIRMM
University of Montpellier / CNRS
Montpellier, France
<lastname>@lirmm.fr

Abstract—This paper presents a novel test solution directly embedded inside CMOS Image Sensors (CIS) to sort out PASS and FAIL dies during production test. The solution aims at reducing test time, which can represent up to 30% of the final product cost. By simplifying the way optical tests are usually applied with an ATE, the proposed Built-In Self-Test (BIST) solution overcomes the drawbacks of long test time and huge amount of test data storage. We experimented our solution by considering that roughly half of the tests usually performed with an ATE can be embedded and applied using the proposed fast and low cost BIST engine. Results obtained on more than 2,400 sensors have shown that our solution reduces test time by about 30% without impacting the defect coverage. The area cost of our solution is about 1% of the digital part of the sensor, i.e., approximately 0.25% of the total sensor area. The proposed embedded CIS test solution outperforms existing solutions in terms of area overhead and test time saving, thus encouraging its future implementation in an industrial production flow.

Keywords—CMOS image sensor, BIST, optical test, pixel array.

I. Introduction

CMOS **I**mage **S**ensors (CISs) are widely used in many electronic devices for various industrial applications, such as smartphones, autonomous vehicles, night vision systems, medical imaging equipments, etc., as well as applications for the space conquest, such as cameras used by the Perseverance Mars rover in 2021. Although the Charge Coupled Device (CDD) technology was initially dominant due to superior sensitivity and picture quality, various improvements in CISs have led them to surpass CCD sensors in shipment volume since about 15 years. In particular, they can benefit from their low fabrication cost, low power consumption, as well as the possibility of combining analog and digital functions [1, 2].

As any other electrical sensor, a CIS translates physical information into electrical information [3]. Its architecture is divided into two parts: the optical part contains an array of pixels (picture elements) that capture photons (light information) and the electrical part converts this light information into electrical signals that are transmitted outside the sensor to display an image composed of pixel values.

Irrespective of the application field, verifying and ensuring that a CIS, especially the pixel array, will be able to operate correctly during in-field application is mandatory. To this purpose, test at the end of the manufacturing process is a crucial step to exhibit potential failures and check various functional, optical and electrical parameters.

Various solutions for manufacturing test of CIS have been proposed so far. A brief summary of these solutions is given in Section III.B. In the meantime, industrial practices for CIS testing are mainly based on using external test approaches based on the use of an ATE (**A**utomatic **T**est **E**quipment). Unfortunately, these practices have a significant impact on the final cost of the product [4]. Depending on the targeted application, the test cost for a complex IC (**I**ntegrated **C**ircuit) like a CIS may represent up to 30% of the final product cost, which is now considered as prohibitive [5].

In this paper, we present a fast and low cost BIST solution for CIS that aims at verifying the good functionality of the optical part (pixel array) inside the sensor. The main objective is to reduce the significant CIS test time required by typical external test procedures at a low cost. Our strategy consists of using the proposed BIST solution to screen out local defects (singlets, couplets, clusters) and to use ATE-based tests to target global defects. To deal with output images provided by a CIS, the BIST solution reuses the way pixels are scanned in conventional image processing algorithms (convolution and median filtering) applied with an ATE. By transforming two-dimensional computations into one-dimensional computations realized inside the sensor, test complexity and test data storage requirements are limited, and test time is saved.

Performances of our solution have been evaluated owing to a software emulation of the developed BIST engine. Validation of the solution has been done through experiments performed on output images coming from 2,400 CMOS image sensors, showing that test time can be reduced by roughly 30% without impacting the defect coverage when compared to a full ATE-based test solution. Moreover, the silicon area estimated to implement the BIST engine is about 1% of the digital part of the sensor, i.e., approximately 0.25% of the total area of a CIS.

The rest of this paper is organized as follows. Section II presents the general architecture of a CIS and details the

conversion flow from the physical information to the output image. A brief summary of the state of the art in the field is also proposed. Section III provides a taxonomy of defects in a CIS and shows how they can be detected by the proposed test solution. Section IV presents the novel BIST solution and its hardware implementation. Section V summarizes the results achieved and compared to optical tests usually performed with an ATE. Section VI concludes the paper.

II. Background on CIS Testing

A. CMOS image sensor overview

Thanks to his optical and electrical blocks, a CIS is able to transform the light information into electrical information through the following conversion flow. The light arrives directly on a micro lens located on top of the pixel so as to do not miss any light rays and to focus them on the photosensitive element. This element inside the pixel, generally a photodiode, converts photons in electrons owing to the photoelectric effect [6]. The general anatomy of a pixel in the array of a CIS is shown in Fig. 1.

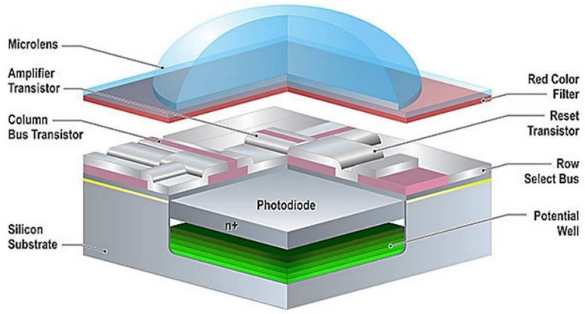

Figure 1: General anatomy of an active pixel in a CIS [6]

To deal with the color aspect, a color filter organized in a Bayer pattern (Red, Green and Blue) is located on the pixel array, bellow each micro lens (cf. red color filter as example in Fig. 1), so that one pixel contains one color information. Note that this color filter contains more green filtering patterns than red or blue patterns to mimic the human eye features with the dominance of green color.

A pixel can be characterized as passive or active. A passive pixel contains only the photosensitive element and a single transistor to select and read the information (electrons) from the photosensitive element. An active pixel has a quite similar architecture with the addition of a reset transistor and an amplifier transistor (cf. Fig. 1). The reset transistor is used to completely discharge the photodiode and hence be sure that there are no parasitic electrons from the previous capture that could contaminate the pixel output [7]. The amplifier transistor allows to verify the electron-to-voltage amplification directly inside the pixel. Due to the addition of transistors in an active pixel, a drawback is that the area occupied by the photosensitive element inside the pixel is reduced. The advantage is that the signal degradation is lower since the parasitic capacitance from the direct connection between the photodiode and the column bus is reduced.

At the high level, row select bus and column bus transistors of each pixel are addressed by row and column decoders to scan and to read out the whole pixel array **row by row**. These blocks are depicted in Fig. 2. They are managed by a sequencer block which controls the system in conjunction with an external clock. When the pixel is addressed by the sequencer, the output voltage of the pixel is converted into a digital signal thanks to an **A**nalog-to-**D**igital **C**onverter (ADC). ADC settings, e.g. gain of the ramp used to sample and to quantify the analog signal, allows to define the digital value of the pixel encoded in a bit word.

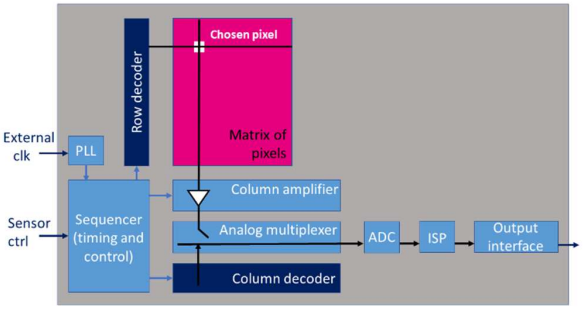

Figure 2: General architecture of a CIS

From the image containing the digital values of all pixels, various operations are carried out by the **I**mage **S**ignal **P**rocessor (ISP) such as white balancing to calibrate and correct the color dominance according to the ambient light, noise reduction to avoid "pepper and salt" aspect, sharpening to ensure good contrast in the image, etc. [8]. The ISP also performs an interpolation on the output RGB image between pixels of the same type (Red, Green or Blue) to rebuild the CIS image on the common three-channels RGB. This principle, also called demosaicing, is illustrated in Fig. 3. The goal of the demosaicing operation is to fill the white boxes as shown in Fig. 3 with values computed from the same-colored neighbors. The output image of CIS is a superposition of the three-channel images and can be displayed on a display module thanks to the interface block (cf. Fig. 2).

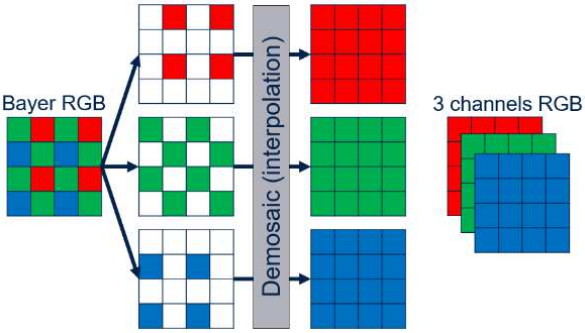

Figure 3: From Bayer RGB to 3 channels RGB image by the ISP demosaicing operation

The output image of a CIS does not reflect exactly the physical scene in front of the sensor but rather is an approximation. There are sources of inaccuracy such as the gap, called *pitch*, between pixels that creates a loss of information, the potential loss of light rays on the photosensitive element (even if the micro lens catches the majority of rays), or the light level definition. Note that, a light level is the interpretation of the brightness. Naturally, the light level seen by a human eye is proportional to the arithmetical value of the pixel, i.e., the lighter the pixel, the higher is its value.

B. State of the art in CIS testing

CIS testing is done at various stages of a sensor life, from characterization of the die on the wafer to online test during sensor lifetime. The work presented in this paper in intended to be used at a production test level in order to sort out PASS and FAIL sensor dies. Test of a CIS comprises electrical and optical tests, all usually performed on an ATE, in order to test all the elements inside the sensor. When one test fails (optical or electrical), the sensor is declared as defective and is withdrawn from the production line.

Electrical tests can be functional, structural, or parametric depending on the part of the circuit (analog, digital or mix) which is under test [9]. Electrical tests include continuity test, leakage control, IDDQ test, conventional stuck-at fault tests, as well as test for memories, ADC, PLL, etc. [10].

Optical tests are performed on the output images coming from a CIS and use image processing algorithms like defect tracking, dark current capture, signal imbalance verification, gain verification, etc. [11]. Results of these optical tests allow to check if output images are correct according to predefined image quality criteria [12]. During optical testing, the CIS under test is put in two main states (dark or light) to avoid error sources such as non-uniformity and presence of a scene or colours. Moreover, there is not only "light on" and "light off" extremes cases but a panel of settings variations (different converter gains, integration time, etc.) [13]. Note that it is possible to detect a physical defect on the pixel array of a CIS in dark *and* in light images or independently in dark *or* in light images.

Optical tests are performed thanks to image processing algorithms applied on the output images of the CIS under test. Usually, it is necessary to wait for several images to be captured entirely and stored by the ATE before computing metrics and determine if the sensor images pass the optical tests or not. This mean that test time and data storage dedicated to test may become prohibitive, e.g. test time can go from 10 seconds to 1 minute [14].

In the literature, some solutions have been proposed to embed test facilities inside a CIS and hence save test time. Authors in [14-16] and in [17-18] propose to reduce the amount of optical tests and proceed with a fully electrical test set so as to avoid precise and restrictive settings needed in optical tests. Authors in [14], [15] and [16] propose a solution based on the generation of a pulse (voltage stimuli) which is sent on the anode of the photodiode inside the pixel to simulate the light illumination. Authors indicate that the implementation of this method is very dependent on the pixel architecture. This structural-based test is done on the pixels of the array independently of each other, without any management of clustering pixels. Experimental results demonstrate a relatively good correlation in dark conditions with respect to tests performed with an ATE, but the comparison is unreliable in the light conditions due to the use of a light source not stable enough.

The work presented in [17] and [18] proposes a BIST framework to detect defective lines used to read out or addressing pixels, thanks to the extraction of the impedance of lines. Authors use two solutions for the detection of continuous and partial defective lines. One solution deals with continuous defective lines by reusing the ADC circuitry to measure resistances. The other solution uses a vector network analyzer equipment to drive the RF impedance used to detect partial defective lines. Despite its efficiency, this solution needs additionnal equipment (line delay detector, delay controler, pulse detector, etc.) to detect defective lines.

Note that all these BIST solutions in [14-18] aimed at reducing optical test time, a common objective with the BIST solution proposed in this paper.

III. Defective Pixel Definition and Determination

A defect is an unwanted particularity of a device that can create a failure. It may come from particle deposition (e.g., bridging between two metal lines, bad contact definition) or process deviation (e.g., diffusion depth issue, misalignment of masks) during manufacturing.

Regarding CIS, defects can often be seen after reading of the pixel array, when the output images are available. Figure 4 shows some examples of image defects. Note that A*A is a Kernel containing A^2 pixels, with A being a positive odd integer. A Kernel is used to stake out an area in an image that allows to classify a defective pixel with respect to its neighbors. The choice of A depends on the sensor resolution and on the customer specifications.

Several categories of defects have been defined [6]:

- **A singlet** is a defective pixel with no immediate defective neighbors and located at the center of a Kernel of A*A pixels, as shown in Fig. 4;
- **A couplet** is a defective pixel with only one immediate defective neighbor in a Kernel of A*A pixels. One of the defective pixels is at the center of the Kernel;
- **A cluster** is a group of more than two defective pixels in a Kernel of A*A pixels. A defective row or a defective column (continuous, partial, dotted, etc.) are two particular cases of clusters.

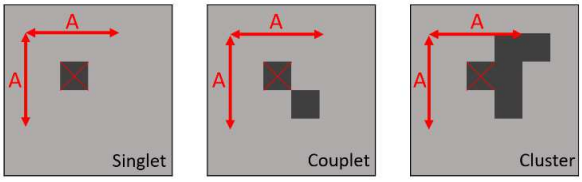

Figure 4: Examples of possible image defects [18]

The classification illustrated in Fig. 4 is based on the defective pixel definition and each of these categories can reflect a physical defect. In order to determine whether or not a pixel is defective, its light level and arithmetic value need to be considered. In a fault-free case, the light level of the pixel needs to be in accordance with the light level of its neighbors so as to ensure uniformity. The arithmetic value of the pixel is used to evaluate its lightness level with respect to a predefined range of values. This range depends on the sensor features. For example, if the pixel value is encoded on 8 bits, the range of pixel values will be [0; 255].

There exist three different types of defective pixels: hot, dead and weak. A *hot* pixel is a pixel with the highest arithmetic value and a *dead* pixel is a pixel with the lowest value [19][20]. A hot pixel can be seen as a pixel stuck at '1' and a dead pixel can be seen as a pixel stuck at '0'. A *weak* pixel is a pixel with a lower arithmetic value when compared to a threshold computed with the values of its neighbors [21].

To detect a defective pixel, a local average has to be determined to check if the pixel value is in accordance to the values of its neighbors. Several local averages are calculated by using different mask sizes. A mask is composed of a group of several neighboring pixels located on the same row. For example, a local average can be computed with a 1x5 size mask (i.e., a 1D array of five pixel values) and is computed by excluding the pixel at the center, i.e. the pixel under test. A defective pixel has a pixel value which is outside a range defined by its higher and lower bounds, called "thresholds" in the sequel. To implement the proposed BIST solution, these thresholds need to be unified, i.e., they need to be calculated in the same way by using constant parameters, named a_{min} and b_{min} for the lower threshold, and a_{max} and b_{max} for the higher threshold. As can be seen in Equations (1), these thresholds are represented by affine functions in which constants are defined depending on whether the CIS operates in a dark or in a light environment (cf. Fig. 5).

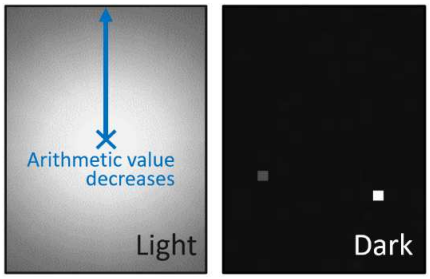

Figure 5: Example of light and dark images of a CIS [19]

In a dark environment, the distribution of pixel values for a given mask should be uniform in a fault-free case. So, we just need to **add a fix number** to the local average (b_{min} and b_{max} in Equations (1)) to compute the thresholds and determine whether the pixel is defective or not. In a light environment, a lens is used to diffuse while focusing the light on the sensor, thus provoking a shaded aspect (cf. example of a light image presented in Fig. 5) on the output image [22]. Due to the difference of light repartition, a fixed value cannot be used to compute the local average. Indeed, a pixel at the center of the image has a high arithmetic value due to the light concentration. Therefore, it could be falsely declared as defective if the computation would be done in the same way as for pixels located at the borders of the pixel array, which have a lower arithmetic value due to the darkness. To compute the thresholds ("Thr"), an alternative is to **use a percentage** of the local average (a_{min} and a_{max}) based on the values of the neighbors of the pixel:

$$\begin{cases} Thr_{min} = a_{min} \times local_{average} + b_{min} \\ Thr_{max} = a_{max} \times local_{average} + b_{max} \end{cases} \quad (1)$$

Equations in (1) are defined in a generic manner to allow their use irrespective of the CIS under test, considering that the threshold values may change from one CIS to another with the modification of the pixel architecture, the gain of the ADC, the number of pixels in the array, etc. This generic formulation also allows the computation of thresholds values irrespective of the detection environment (light or dark). Note that the constant parameter values in (1) are chosen **empirically** based on the acceptable rate of defective pixels in CIS output images.

IV. New Built-In Self-Test Solution

A. Principle of the proposed test approach

CIS testing consists of electrical and optical testing phases. A CIS can pass electrical tests and fail optical tests, or vice versa. In both cases, the CIS is declared as FAIL if at least one test reports a failure. However, electrical tests are significantly faster than optical tests, thus motivating the need to minimize the duration of optical tests for CIS.

In order to save test time of a CIS, a solution consists in using an embedded test solution (BIST) to screen out local defects (singlets, couplets, clusters) and to use ATE-based tests to target global defects, such as pixel values uniformity along the array or noises like **F**ixed **P**attern **N**oise (FPN) [1], which represents the variation of pixel values in the array. BIST allows to test each pixel on-the-fly without the need to store a full image before testing as is the case with ATE-based testing. Moreover, BIST is performed at-speed so that test time can be greatly reduced compared with ATE-based test.

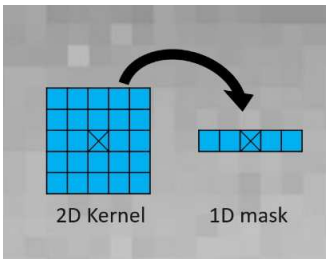

Figure 6: From a 2D Kernel to a 1D mask on a pixel under test

Of course, the area overhead required by the BIST hardware limits the amount and complexity of possible computations, so that embedding conventional image processing algorithms used for optical test in a BIST engine can be too costly. Indeed, convolution and filtering methods used in image processing algorithms are generally based on two-dimensional data that are available when an entire image has been stored. However, when only a flow of pixels coming from a row (i.e. a mask) of the pixel array is available, using such methods becomes impossible. To alleviate this issue, we simply reuse the way pixels are scanned in 2D convolution and 2D median filtering methods to deal with 1D masks of pixels (cf. Fig. 6). With this modification, we avoid the need of additional hardware to store data during the reading of the array, and we limit the complexity of operations that are specific to 2D data management, thus reducing test time.

B. BIST workflow

To test the pixel array, the BIST engine needs to access each arithmetic pixel value inside the array. The array is streamed row by row at the pixel rate thanks to the sequencer block presented in Fig. 2. When a pixel is dealt with the BIST module, first, the pixel type (Red, Green or Blue) is determined from the pixel coordinates. Then, the arithmetic value of the pixel is compared with the two thresholds computed from a local average (cf. Equations (1)). If the pixel value is outside the defined range, the pixel is defective and its data (value and coordinates) are stored in a memory for the next phase. Otherwise, these data are not saved.

Once all the pixels of the array have been evaluated by the BIST, the data of each defective pixels stored inside a

memory is read by a software program which is able to count singlets, couplets and determine whether there is a cluster or not into the pixel array of the CIS under test.

C. General BIST architecture

In our work, CIS die sorting is done by resorting to two types of resources: the first one is a BIST infrastructure embedded in the CIS for test and defective pixel detection, and the second one is an external memory for test data storage. This distribution of tasks has been decided to achieve the best tradeoff between test time efficiency and CIS hardware overhead. An external CPU (**C**entral **P**rocessing **U**nit) is used for defect classification.

One important point is the location of the BIST module inside the CIS architecture. In order to avoid pre-processing on pixel values by the ISP, that could potentially correct and hide potential defects, pixel values must come directly from the pixel array without going through intermediate hardware modules like correction modules, filtering block, scaler block, etc. [8]. Fig. 7 shows the integration and interfacing of the BIST module inside a CIS.

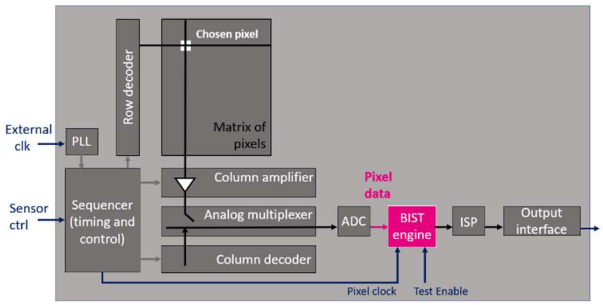

Figure 7: Insertion of the BIST module inside a CIS

The proposed BIST solution has been implemented in Verilog language. The general BIST architecture is depicted in Fig. 8 where each digital block is dedicated to one function.

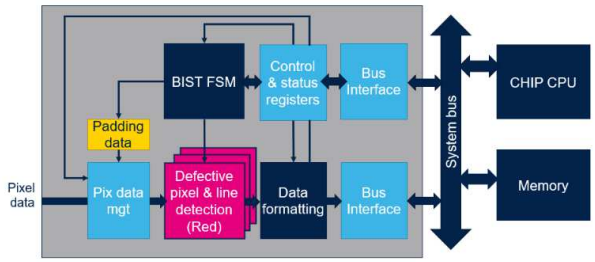

Figure 8: General BIST architecture

All the blocks detailed in this architecture are synchronized by a Finite State Machine (*BIST FSM*) which is used to manage the BIST process by launching appropriate signals and collecting required information. Communications between the various blocks is done owing to wires (1-bit signal carriers) or bus (word of several bits carrier).

In order to test all pixels in a CIS array, each pixel is selected based on its type (red, green or blue) since a pixel of one type can only be tagged as defective with respect to pixels of the same type. The block *Pixel data mgt* guides the pixel in dedicated accumulators located inside the *Defective pixel & line detection* block to store pixel values of the same type. This allows to do parallel computations on several accumulators to pipeline the detection on several pixel types.

If a pixel is located near the borders of the array, it is mandatory to compute a padding value to detect if the pixel is defective or not. This suggests a coordinate management inside the BIST module, that can be done with counters operating at the pixel rate and returning row and column relative positions in the array.

Then, the pixel is processed by the *Defective pixel & line detection* block. More precisely, the value of the pixel is compared to the thresholds (Thr_{min} & Thr_{max}) to determine if the pixel is correct or defective. The pixel data, i.e. the value of the pixel, its coordinates and its type, are stored in the memory of the sensor in case it is declared as defective.

The information "the pixel is defective" in the form of a boolean will launch the writing of the pixel data inside the external memory. Note that, to avoid test hardware overhead, the external memory is a memory already existing in the CIS but not used during the test phase and hence available to store pixel data of defective pixels. The coordinates and values of a defective pixel are encoded by the *Data formatting* block before storage in the memory.

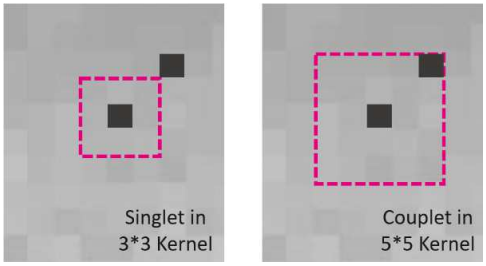

Figure 9: Singlet or couplet according to the size of the Kernel

When all the pixels in the array have been evaluated and have passed through the BIST engine, the memory fulfilled by all the defective pixels data is read by a python program embedded on the CPU of the system to classify the defective pixels. By using information about the Kernel size, this program is able to compare pixel coordinates to determine whether a defective pixel can be linked with neighboring pixels and placed in a given category. Let us consider the example in Fig. 9. Here, the program will count two singlets in the left image if the size of the Kernel is defined by 3*3 whereas it will count one couplet in the right image if the Kernel size is 5*5. The number of singlets, couplets and the presence of clusters are information that can make the die passing from PASS to FAIL. If the number of singlets or couplets in the dark or light images is too high considering a given limit, the CIS is tagged as FAIL. Note that the presence of only one cluster in dark or light conditions is a FAIL factor.

V. Validation of the BIST solution

In order to validate the proposed BIST solution, we performed experiments by using a dataset composed of images taken by CISs in dark and light environments. The output of the BIST engine is a PASS/FAIL information related to the CIS under test as well as the number of singlets, couplets and cluster associated to each image. The goal of our experiments is to demonstrate the efficiency of the BIST solution in reproducing part of optical tests for CIS as they

are usually applied with an ATE without the drawbacks of long-time duration and huge amount of data storage.

A. Experimental setup

A dataset composed of output images coming from more than 2,400 CMOS image sensors originating from the same packet were used in our experiments. Among these 2,400 CISs, a part of them were identified as FAIL by former ATE-based optical tests. Images coming from these CISs were selected so as to get a representative sample of various defect categories, i.e., singlets, couplets, defective columns, defective rows, clusters of various sizes, etc. For each CIS, we collected the **same number** of dark and light images so that the dataset was split into two equivalent sets of dark and light images as shown in Fig. 10.

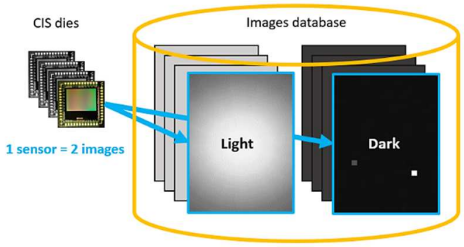

Figure 10: Dark & Light image distribution of the dataset [6]

The subset of images coming from the FAIL CISs was organized into several categories with respect to the defect type. For each image, the numbers of singlets and couplets are known, as well as the presence of cluster, thanks to the output data from former optical tests performed on ATE. Many other information coming from electrical tests formerly performed on these CISs are also available for each image. For this reason, there is a need to analyze all data related to a given image to sort out relevant information and to be sure that the FAIL classification of a given CIS came from an optical failure and not from an electrical failure. To this end, these information are stored in a csv file to be further used in a postprocessing stage and observe correlation between results from the BIST solution and those from the ATE-based tests.

B. Results and correlation

In order to evaluate our BIST solution, we define some **error metrics** to quantify the relative error between results obtained with the ATE-based solution, considered as a reference, and those produced by the emulated BIST solution. For each defect category, an error metric has been defined by Equations (2) as follows:

$$\begin{cases} error_{singlet}[i] = singlet_{BIST}[i] - singlet_{ATE}[i] \\ error_{couplet}[i] = couplet_{BIST}[i] - singlet_{ATE}[i] \\ presence_{cluster}[i] = (cluster_{BIST}[i] == cluster_{ATE}[i]) \end{cases} \quad (2)$$

where "i" is an integer in [0; N-1] associated to a given image, N being the total number of images in the dataset.

For all experiments, errors were computed for dark and light images in the dataset. A negative error indicates that the BIST solution has identified a lower number of defects than the ATE-based solution. Note that applying such type of error computation for the cluster category is unfeasible as the presence of a cluster is represented by a Boolean indicator (0 or 1) and not by an integer value. So, in this case, we count the number of **common images** with a cluster reported by the BIST and by the ATE-based tests.

To quantify the efficiency of the proposed BIST solution, we have computed the mean of the relative error for all output images owing to Equations (3) shown below:

$$\begin{cases} Average_error_{singlet} = \frac{\sum_{i\in[0;N-1]} error_{singlet}[i]}{N} \\ Average_error_{couplet} = \frac{\sum_{i\in[0;N-1]} error_{couplet}[i]}{N} \end{cases} \quad (3)$$

To determine the threshold values as defined in Section III, we need to calculate the local average for each pixel under test and define values of constant parameters. The local average is calculated by considering a mask size and the values of pixels inside the mask. Several mask sizes (1x3, 1x5 and 1x7) have been considered based on experience. The values of constant parameters used to determine higher and lower thresholds were empirically chosen as shown in (4).

$$\text{Constants for dark img} : \begin{cases} a_{max_DARK} = 1; & b_{max_DARK} = +120 \\ a_{\min_DARK} = a_{max_DARK}; & b_{\min_DARK} = -\frac{b_{max_DARK}}{2} \end{cases}$$
$$\text{Constants for light img} : \begin{cases} a_{max_LIGHT} = 1 + 0.15; & b_{max_LIGHT} = 0 \\ a_{\min_LIGHT} = 1 - 0.15; & b_{min_LIGHT} = b_{max_LIGHT} \end{cases} \quad (4)$$

The two sets of constants given in (4) are chosen to detect defective pixels with values outside the range [local_avg-60; local_avg+120] for dark images ("img") and outside the range [local_avg*(100%-15%); local_avg*(100%+15%)] for light images. These values are very dependent on the type of CIS and have been chosen empirically.

TABLE I. SINGLET AND COUPLET MEAN ERROR (IN PIXEL) IN DARK AND LIGHT IMAGES DEPENDING ON THE MASK SIZE

	Mask size	1x3	1x5	1x7	3x3
Dark	**Singlet**	3.70	4.40	-66.20	4.80
	Couplet	-0.05	-0.08	-0.26	-0.08
Light	**Singlet**	-1.90	-2.10	30.00	-1.80
	Couplet	0.05	0.00	0.06	0.01

Table I reports the mean error calculated for singlets and couplets with respect to a given mask size. Three differents 1D mask sizes (1x3, 1x5 and 1x7) have been experimented. The selection of these mask sizes has been done so as to limit storage of pixel values used in the local average computation. The 3x3 mask size, which is based on a 2D computation performed by the ATE-based test, is used to compare results with a 1D computation. We can observe that the singlet mean error for dark and light images are acceptable for 1x3 and 1x5 mask sizes. Indeed, a mean error of 4.4 pixels with a 1x5 mask size for dark images represents a deviation of **0.0003%** with respect to the total number of pixels in the CIS array, which is a negligible error. Conversely, we can observe that the 1x7 mask size is not appropriate for the detection of singlets in dark or light images. Indeed, 66.2 defective pixels are missing with the BIST solution for the dark images and 30 additional pixels have been found for the light images, which is not acceptable. The detection of couplets in dark or

light images performs very well irrespective of the mask size (mean errors are around 0), thus demonstrating the efficiency of the BIST solution in detecting double defective pixels. From the results reported in Table I, we can conclude that the detection of singlets and couplets performs very well for 1x3 and 1x5 mask sizes.

Table II reports the number of images containing a cluster with respect to a given mask size. The 1x3 mask allows to detect a higher number of bad images with the BIST compared to the ATE-based solution (92 against 69) in a light environement. However, as only 63 images over the 92 are in common with the ATE-based solution, we can conclude that using a 1x3 mask size is not appropriate.

TABLE II. NUMBER OF IMAGES WITH A CLUSTER DEPENDING ON THE MASK SIZE

	Mask size	1x3	1x5	1x7	3x3
Dark	**Found by ATE-based test**	33			
	Common images	*32*	*33*	*31*	*33*
	Found by BIST solution	34	36	33	35
Light	**Found by ATE**	69			
	Common images	*63*	*65*	*60*	*64*
	Found by BIST solution	92	65	67	68

Conversely, results regarding cluster detection obtained with a 1x5 mask size perfectly correlates the results found by the ATE-based solution for dark images (33/33 images in common). Moreover, this 1x5 mask size allows to detect three additional images containing a cluster with the BIST solution (36/33). After analysis, these images were confirmed to be bad images actually containing a cluster (defective row). This is an interesting feature of our BIST solution which was able to detect the cluster on dark *and* light images, whereas the ATE found the cluster *only* on light images.

Regarding light images, the use of a 1x5 mask prevented the BIST solution to identify four bad images containing a cluster (65 versus 69). However, a careful analysis revealed that these four images contained shaded rows that were found by the BIST solution on the dark images. So, finally the BIST do not miss any defect on any sensor, thus demonstrating the effectiveness of our solution.

Finally, for the sensor type considered in our experiments, we have chosen the 1x5 mask size, that offers the best tradeoff between lowest singlet and couplet error rate and good cluster detection.

Figure 11 depicts the singlet error distribution for dark and light images given in the form of a histogram. Each bar of the histogram refers to the number of images having the same singlet errors. Ideally, the two singlet error distributions should be placed around the "0" meaning that there is no error between ATE-based and BIST solutions. The standard deviation (σ in Fig. 11) is used to show that the dispersion of values around the mean is low, with 1.6 pixels in light images and 2.3 pixels in dark images. From the standard deviation definition, it means that 99.7% of the data is included inside [mean-3σ; mean+3σ] [23]. For dark and light images, we can see that the "mean±3σ" extrema values are +11.2 pixels (dark) and -6.8 pixels (light). These values represent respectively 0.0008% and 0.0005% of the total number of pixels in the CIS array.

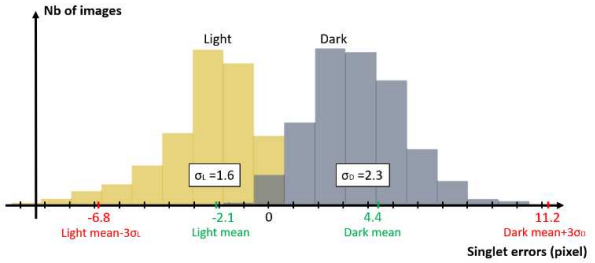

Figure 11: Distribution of singlet errors in dark and light images

We can also observe in Fig. 11 that the distribution of singlet error in dark images is shifted to the positive values and the distribution in light images is shifted to the negative values. This means that BIST testing has detected a bit more singlets than the based-ATE tests in dark images and a bit less in light images. A solution to even more increase the effectiveness of our BIST solution would consist in adjusting the constant values used in Equations (1) in order to shift the two distributions in Fig. 11 towards '0'.

With a mask size fixed at 1x5 thanks to information given in Tables I and II, we can now adjust the constant values used to define the thresholds. Note that by changing constant a_{max} or b_{max}, constants a_{min} and b_{min} will change too due to the dependency between the high and low thresholds of the range. Table III gives the mean error for singlet and couplet detection on dark images when the constant b_{max_DARK} varies. As can be observed, the choice of b_{max_DARK} = +120 is validated in the dark as it is the only one which has the lowest singlet error number. Note that, if the BIST engine detects more singlets with +110, it is because the higher threshold is set too low and the lower threshold is too high, resulting in a very restricted range of good pixel values. The same comment can be done with b_{max_DARK} = +140. The BIST engine misses a lot of defective pixels as the thresholds create a large range where all pixel values have been tagged as not defective. For the couplet mean error rate, we can see that the BIST classifies correctly double defective pixels even if the constant b_{max_DARK} varies.

TABLE III. MEAN ERROR RATE FOR SINGLET AND COUPLET DETECTION IN DARK IMAGES WITH VARIATION OF B_{MAX_DARK}

	b_{max_DARK}	110	120	130	140
Dark	**Singlet**	37.30	4.40	-21.90	-44.30
	Couplet	0.05	-0.08	-0.15	-0.22

TABLE IV. NUMBER OF DARK IMAGES WITH A CLUSTER DEPENDING ON B_{MAX_DARK}

	b_{max_DARK}	110	120	130	140
Dark	**Found by ATE-based test**	33			
	Common images	*33*	*33*	*32*	*31*
	Found by BIST solution	38	36	34	35

Table IV gives the number of detected dark images with a cluster when b_{max_DARK} varies. The same comments as done for Table II can be raised since, for example, the BIST solution with b_{max_DARK} = +110 allows to detect five more

images than the ATE-based solution, but nevertheless detects the same 33 bad images found than the ATE-based solution.

We have done the same comparison for light images by varying constant a_{max_LIGHT}. Table V shows the mean error rate for singlets and couplets. As for dark images settings, the choice of 15% has been found to be the most appropriate due to the lowest singlet error rate and considering that the couplet error rate is very good irrespective of the percent value.

TABLE V. MEAN ERROR RATE FOR SINGLET AND COUPLET DETECTION IN LIGHT IMAGES WITH VARIATION OF A_{MAX_LIGHT}

	a_{max_LIGHT}	10%	15%	20%
Light	**Singlet**	16.00	-2.10	-4.20
	Couplet	0.64	0.00	-0.01

As for dark images, the number of images with cluster was computed and results are given in Table VI. These results show that 15% is indeed the most appropriate constant value.

TABLE VI. NUMBER OF LIGHT IMAGES WITH A CLUSTER DEPENDING ON A_{MAX_LIGHT}

	a_{max_LIGHT}	10%	15%	20%
Light	**Found by ATE**	69		
	Common images	*67*	*65*	*59*
	Found by BIST solution	143	65	62

From all the above experiments and results, we can conclude that correctly setting values of constants a_{max}, b_{max} and a_{min}, b_{min} is crucial, as well as correctly selecting the mask size. A **precise** setting is mandatory to avoid image missclassification due to a false number of singlets or couplets. Regarding couplet detection, the proposed BIST performs always very well irrespective of the mask size or the constant values.

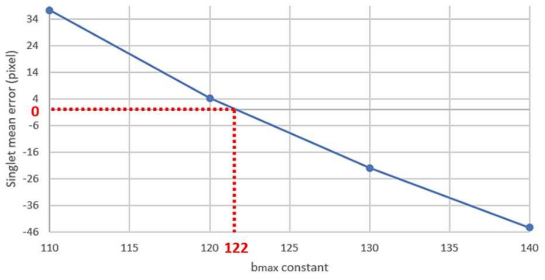

Figure 12: Singlet mean error for dark images with respect to b_{max_DARK} constant variation

To be even more precise in the analysis of the b_{max_DARK} parameter, linear approximations have been done to find the optimal constant value. Figure 12 reports the result of the linear approximations and shows that b_{max_DARK} should be precisely set to +122 to get a perfect correlation between number of singlets found by the BIST solution and those found by the ATE-based solution.

As for b_{max_DARK}, linear approximations have also been done to find the optimal value for a_{max_LIGHT}. The result led to 14.4%.

C. Discussion

With a correct setting of constant values, the proposed BIST solution is able to appropriately sort the dark or light images into good or bad images. Nevertheless, PASS and FAIL sensors classification from dark and light images require further analysis. More precisely, if a dark **or** a light image is labelled as bad by the BIST solution, the sensor will be declared as FAIL. A sensor will be declared as PASS if both dark **and** light images are identified as good. To be "good", an image must not contain any cluster and we need to verify that the number of singlets and couplets are under admitted values defined according to a given application.

Considering the whole dataset of 2,400 sensors, the BIST solution has a **99.95%** correlation with the PASS/FAIL classification provided by the ATE-based solution. Only one CIS has been misclassified as FAIL by the BIST engine.

Results obtained with our BIST engine shows that, from quite simple computations used to detect defective images, it allows to sort reliable PASS and FAIL information for a CIS under test. The optical part of a CIS is tested directly inside the sensor, without any interface that can add sources of errors (potential contact resistance, pad continuity, etc.). Regarding the coverage of optical algorithms usually performed on ATE, we estimate that more than 50% of these algorithms can be embedded inside the CIS under test, thus leading to a reduction of approximately 30% of the test time for each sensor. This estimation is based on a deep analysis of optical tests, demonstrating that 50% of them usually target local defects and hence can be embedded. The area taken by the BIST architecture has been estimated in terms of additionnal logic gates. It represents roughly only 1 % of the digital part of a sensor.

As BIST solutions presented in Section II.B, the proposed BIST engine does not embed all optical tests usually performed on an ATE. Compared to existing BIST approches, the advantages of the proposed BIST solution are that i) it represents less than a few percents of the total CIS area, far below what can be achieved with some of the existing solutions, and ii) all potential local defects in the pixel array can be detected, while only a part of them are covered by existing BIST solutions.

VI. CONCLUSION AND DISCUSSION

In this paper, we have presented a novel BIST solution for CIS testing. It is based on the detection of defects such as singlets, couplets and cluster in dark or light output images from a CIS under test. These detection information are used to define whether images are good or bad and, finally, if the CIS is PASS or FAIL. A software emulation of the BIST engine has been done to validate our solution. Experiments carried out on images coming from 2,400 CIS have shown that our proposed solution adequately classify CIS into PASS and FAIL categories in 99.95% of cases. In addition to be useful for testing CIS, this type of embedded test solution is also valuable for diagnosis purpose, since the coordinates of defective pixels stored inside the memory are easily reachable.

Despite its effectiveness, it is still possible to improve our solution by considering the following aspects:

- Our BIST solution currently focuses on local defects (singlets, couplets and clusters) and not yet on global defects such as shading (difference of values along the array, change of uniformity) or noises like temporal noise (non-desired variations of pixel values between different captures). Even if the probability to have a local defect is higher than the probability to have a global defect (except for some noises such as FPN and dark current, which are very common in CISs), complementing our solution to cover global defects is of real interest. To this end, simple modifications can be implemented to add functionalities to the BIST solution and hence cover these global defects.
- Bus latency to communicate between the external memory and the BIST module can be extensive and the size of the memory can be inadequate compared with what we need to store (too large or too small pixel data). A possible solution would consist in adding a small memory inside the CIS, together with the BIST hardware. This could be done at the cost of additional area overhead.
- Location information about defects were not used in our work to compute the correlation between optical tests performed on ATE and with the BIST solution. Indeed, the correlation relies only on the number of defects in each category (singlet, couplet and cluster). However, it may happen that two different defects are detected respectively by the ATE-based test and the BIST solution, and hence are considered in the correlation computation. Adding a comparison between coordinates of defective pixels detected by the BIST and those detected by the ATE could provide even better correlation information.

REFERENCES

[1] A. Dupret, M. Tchagaspanian, A. Verdant, L. Alacoque, and A. Peizerat, "Smart Imagers of the Future", *in Proc. IEEE Design Automation & Test in Europe*, 2011.

[2] M. Bigas, E. Cabruja, J. Forest, and J. Salvi, "Review of CMOS Image Sensors," *Microelectronics Journal*, vol. 37, no. 5, pp. 433-451, May 2006.

[3] B. Luo, F. Yang, and L. Yang, "Key Technologies and Research Development of CMOS Image Sensors", *in Proc. IITA International Conference on Geoscience and Remote Sensing,* 2010.

[4] Quality Testing Services, "Breakdown on Costs of Testing vs. Cost of Not Testing," May 1, 2019. https://qualitytest.net/product-testing-vs-not-testing-costs/

[5] A. Grochowski, D. Bhattacharya, T.R. Viswanathan, and K. Laker, "Integrated Circuit Testing for Quality Assurance in Manufacturing: History, Current Status, and Future Trends", *in Proc. IEEE Transactions on circuits and systems-II: analog and digital signal processing*, vol. 44, no.8, 1997.

[6] R. Fei. "Alternative Solution to Improve the Production Test of Optical Sensors in CMOS Technology", PhD dissertation, Grenoble Alpes University, 2015.

[7] I.L. Fujimori, "CMOS Passive Pixel Imager Design Techniques", PhD dissertation, Massachusetts Institute of Technology , 2002, 50544970-MIT.pdf.

[8] M. Tusch, G. Cooper, and T. Atherton, "Combining an ISP and Vision Processor to Implement Computer Vision", Embedded Vision Alliance, 2019, https://www.edge-ai-vision.com/2019/01/combining-an-isp-and-vision-processor-to-implement-computer-vision/.

[9] G. Prenat, "Design of an Analog and Mixed-Signal BIST Architecture in Deep Submicron Technology," PhD dissertation, Institut National Polytechnique de Grenoble, 2005.

[10] A. Bounceur, "CAT Platform for Mixed-Signal Circuit Testing", PhD dissertation, Institut National Polytechnique de Grenoble, 2007.

[11] J. Chen, "CIS Testing Technology", Jetek Technology corporation, 2018. https://fr.slideserve.com/amos/cis-testing-technology

[12] M. Pedersen, J.Y. Hardeberg, "Full-Reference Image Quality Metrics: Classification and Evaluation", Foundations and Trends® in Computer Graphics and Vision, vol. 7, no. 1, pp. 1-80., March 2012.

[13] M.B. Kaplinski, G. Waligorski, V. Berezin, and E.R. Fossum, "Test Methodologies for Digital CMOS Camera-on-a-Chip Image Sensors", *in Proc. IEEE Workshop on CCDs and Advanced Image Sensors*, pp. 239-241, 1999.

[14] L. Lizarraga, S. Mir, and G. Sicard, "Experimental Validation of a BIST Technique for CMOS Active Pixel Sensors," *in Proc. IEEE VLSI Test Symposium*, pp. 189 – 194, 2009.

[15] L. Lizarraga, S. Mir, and G. Sicard, "Evaluation of a BIST Technique for CMOS Imagers," *in Proc. IEEE Asian Test Symposium*, 2007.

[16] L. Lizarraga, S.Mir, G. Sicard, and A. Bounceur, "Study of a BIST Technique for CMOS Active Pixel Sensors," *in Proc. IFIP International Conference on Very Large Scale Integration*, pp. 326-331, 2006.

[17] R. Fei, J. Moreau, and S. Mir, "BIST of Interconnection Lines in the Pixel Matrix of CMOS Imagers," *in Proc. IEEE International Workshop on Advances in Sensors and Interfaces*, pp. 174 –177, 2013.

[18] R. Fei, J. Moreau, S. Mir, A. Marcellin, C. Mandier, E. Huss, G. Palmigiani, P. Vitrou, and T. Droniou, "Horizontal-FPN Fault Coverage Improvement in Production Test of CMOS Imagers," *in Proc. IEEE VLSI Test Symposium*, pp. 1-6, 2015.

[19] G.H. Chapman, J. Leung, I. Koren, and Z. Koren, "Tradeoffs in Imager Design with Respect to Pixel Defect Rates", *in Proc. IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, 2010.

[20] C.-Y. Cho, T.-M. Chen, W.-S. Wang, and C.-N. Liu, "Real-Time Photo Sensor Dead Pixel Detection for Embedded Devices," *in Proc. International Conference on Digital Image Computing Techniques and Applications*, pp. 164 - 169, 2011.

[21] "Basic Principles of Image Sensors", Online courses "Optique Ingénieur", Maine University, http://www.optique-ingenieur.org/en/courses/OPI_ang_M05_C06/co/Contenu_18.html

[22] Advantest Corporation, "CMOS Image Sensor Test Solution", 2017. http://www.koreatest.or.kr/sub02/2017data/%EB%B0%95%EA%B8%B0%EC%88%98.pdf

[23] M. Perekupka, "Standard Normal Distribution: Definition & Example," Study.com, March 9, 2015. http://study.com/academy/lesson/standard-normal-distribution-definition-example.html