The Use of Digital Image Processing for IC Reverse Engineering

Raúl Quijada, Arnau Raventós, Francesc Tarrés
Signal Theory and Communications Department
UPC – Barcelona Tech
e-mail: francesc.tarres@ upc.edu

Roger Durà, Salvador Hidalgo
Reverse Engineering Group
IMB-CNM CSIC
e-mail: salvador.hidalgo@ csic.es

***Abstract*— IC Reverse engineering is the process to analyze an integrated circuit to obtain information about its design, materials, logic circuitry, functionality, performance and other relevant features. The increasingly complexity of microchips using a greater number of layers and logic gates makes this process unaffordable when using traditional methods that rely on human inspection and analysis. Therefore, digital image processing is presented as a fruitful field for automation. In this paper a system for the circuitry extraction, analysis and presentation is described. It is divided in three blocks: 2D Image Tiling, Logic Gates Localization & Recognition and Microchip Navigator. The paper presents an overview of the complete system and is mainly based on the description of the image processing algorithms that are applied to the different blocks such as image stitching, customized Scale Invariant Feature Transform (SIFT) and logic gate localization & recognition.**

***Index Terms*—Image stitching, mosaicing, object recognition, IC reverse engineering.**

I. INTRODUCTION

In the recent years Reverse Engineering (RE) has gained popularity in activities such as performance and security benchmarking, control quality certifications and support patent licensing [1]. The main idea of RE is to analyze a final product to obtain information concerning its design, its constituent materials and components, functionality, performance and other relevant features. According to the state-of-the-art in Integrated Circuit Reverse Engineering[1], this practice can be grouped in several specialization areas: product teardowns, system level analysis, process analysis and circuit extraction. Only the latter is treated in this paper.

The main idea of RE is to analyze a final product to obtain information concerning its design, the materials and components that constitutes it, functionality, performance and other relevant features. According to the state-of-the-art in Integrated Circuit Reverse Engineering [1], this practice can be grouped in several specialization areas: product teardowns, system level analysis, process analysis and circuit extraction. Only the latter is treated in this paper.

Circuit extraction is the IC reverse engineering part focused on the microchip schematic and netlist retrieval. Essentially, the analysis process involves a decapsulation of the microchip to apply a delayering method that consists in stripping off the microchip regarding its metal layers. For each layer an imaging process is performed that requires large magnifications to distinguish and identify the components, e.g., logic gates, memory cells, etc. The next step corresponds to the annotation of the components and their interconnections. Finally, the net list and schematic can be obtained. There are other implicit operations such as verification and circuit simulation that are necessary to validate the process, since imaging and annotation processes are prone to errors.

Digital image processing is presented as a fruitful field for automation tasks related with the imaging and annotation stages. The output of the imaging process contains thousands of single images that have to be stitched into a single image per layer. Accurated stitching requires automatic and reliable finding of saliency points in the individual images. Later, object recognition is used to locate and identify the logic gates.

The contribution of this paper is the presentation of a system whose purpose is the automation of some stages of the circuit extraction process, in particular, imaging and annotation. Hence, the system starts from the pictures obtained from the imaging process and outputs a formatted netlist (in Verilog) to be managed by commercial software in order to generate the schematic. This system has been divided in three blocks: 2D Image Tiling, Logic Gates Localization & Recognition and Microchip Navigator. The first block deals with the mosaicing reconstruction obtained from the imaging stage. A customized version of the Scale Invariant Feature Transform (SIFT) [2] is used for finding correspondences between images, whereas the reconstruction is performed by a region-growing algorithm. The next block faces component localization & identification using morphology for image enhancement and Pearson correlation as similarity measure for the recognition. The last block is addressed to annotate the interconnections between the components and to align images between consecutive layers.

The organization of the paper is divided in the following sections. The description of the implemented system as well as the explanation of the used algorithm are defined in Section II. The results are presented in Section III, and finally, Section IV contains the conclusions and future work.

978-1-4799-3866-7/14/$31.00 ©2014 IEEE

II. SYSTEM DESCRIPTION

The following subsections are focused on the problem statement and the technologies used in the different blocks that compose the complete system.

A. 2D Image Tiling

This block deals with the stitching of the images obtained from the imaging process. Its purpose is to reconstruct a mosaic for each layer of the microchip.

The main problem arises from the delayering and imaging procedures. In our case a mechanical polishing for delayering and optical imaging using manual shooting has been used. Common inconveniences of such methodology are: irregular polishing, some regions are more polished than others, and non-uniform overlapping among images including gaps due camera positioning errors. Other factors may be optical distortion that magnifies the outer regions of the image.

Gate-array parts may also introduce serious difficulties for recomposing since images present high symmetry, which are mainly formed by nodes, contact points and homogeneous regions (see Fig. 1 and Fig.2).

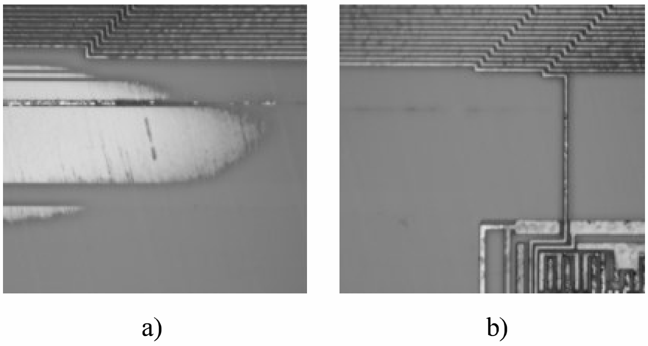

Figure 1. Images a) and b) represent a matching example that contains homogeneity areas. It is important to note the non-uniform polishing represented by the white spot from the left region of the image a).

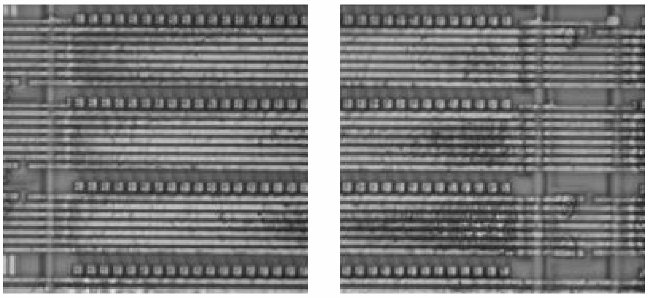

Figure 2. Images c) and d) shows a matching example with a significant similarity.

1) Algorithm Description

The stages of 2D Image Tiling system are listed below:

- Compute the translation vectors between images using an algorithm for automatic feature detection and matching between neighboring images. The algorithm is a customized version of SIFT [3].
- Cluster the translation vectors for each pair of images using a hierarchical tree based on the Euclidean distance.
- Score the matching reliability based on the percentage of translation vectors from the largest cluster.
- Apply a region growing algorithm to reconstruct the mosaic, where the more reliable matchings are merged first.

SIFT is a method that finds features in the objects of a set of images and searches for correspondences to identify these objects towards changes of scale, rotation, translation, etc. An adaptation of the algorithm may be used to find features in neighboring images and find the translation vectors for stitching the mosaic [2]. Its robustness in scale and rotation makes it an attractive solution to solve the optical distortions. Besides, its feature descriptor presents a reliable performance against variations in illumination and chromacity. However, the characteristics of our scenario (see Fig. 1) decrease the performance of the generic SIFT version; hence the methodology has to be adapted to these problems.

Our SIFT version adopts the idea proposed in [3] when presenting the Upright-Speed Up Robust Feature algorithm (U-SURF). SURF is considered an evolution of SIFT, where its computational cost is significantly decreased, yet its robustness is almost similar or slightly inferior in some tests. U-SURF is then the SURF version when rotation is not considered. In our case the rotation can be neglected; thereby the efficiency can be improved by disabling SIFT rotation invariant method. This adaptation can be understood as the Upright-SIFT version. Another main difference between our version and the generic SIFT is the feature descriptor, where our descriptor is focused on the distinctiveness getting advantage that no rotation is appreciable. The descriptor is obtained through orientation histogram with a double precision where 16 bins are used instead of 8. On the other hand, the generic version proposes a spatial interpolation to subpixel accuracy to increase descriptor robustness against rotation but reducing distinctiveness; consequently, this interpolation is dismissed since rotation is negligible.

It is important to note that the repeatability in such scenario must be increased due to the high similarity and homogeneity of the images.

B. Logic Gates Localization & Recognition

The current complexity of the integrated circuits entails tens of thousands of logic gates to be detected, located and identified. Identifying gate structures in silicon is not difficult for an expert in reverse engineering but the workload is so huge that automation may play a key role in drastically reducing the analysis time.

This stage consists in the detection, localization & identification of the logic gates. The main problem encountered stems from the images obtained from the polishing step. Its non-uniformity behavior alters locally the image intensities; thus it complicates the recognition process (see Fig.1).

1) Algorithm Description

The strategy applied to solve this problem is divided in localization & recognition.

Localization is targeted to detect and locate logic gates within the cell layer. The algorithm is based on the energy level of the image gradient, where a threshold is empirically defined to discern regions that likely contain logic gates. As the cell layer background is quite homogeneous its energy level represents low values, thereby logic gates belongs to those regions that surpasses the threshold (see Fig.3).

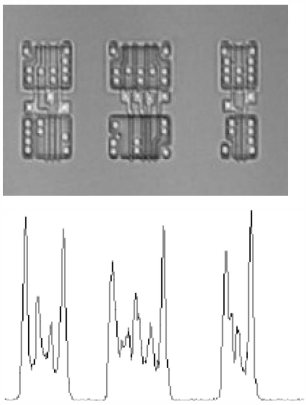

Figure 3. Cell layer region with logic gates and its energy.

The recognition stage is focused to find instances of several logic gate patterns through the previous defined regions. As there are no changes in scale and rotation in this scenario, the Pearson correlation is a suitable measure to compute the similarity between a pattern and their instances. However, the complexity results from the irregular polishing that alters the image intensities, and consequently the first and second statistical moments computed by the Pearson correlation. To reduce this inconvenience a binary mask is applied to each correlation with the aim to highlight the more reliable logic gate parts. Those parts usually belong to the silicon region of the logic gates as can be observed in Fig.4 represented by the white gaps.

Pearson correlation is a reliable similarity measure but it is computationally expensive. However, it can be significantly reduced if the correlation is only computed in a few points for each logic gate. Those points must be robust against polishing imperfections, as for instance the contact points of the logic gates (see Fig.4). Therefore morphology is applied to detect the contact points. It is important to highlight that the logic gate pattern and its instance must be aligned according to the detected contact point before applying the correlation. Therefore, some contact points must be previously identified for the logic gates patterns.

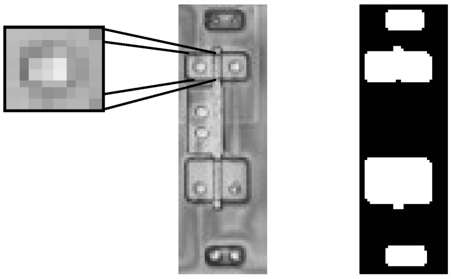

Figure 4. From left to right, a logic gate pattern image highlighting a contact point and its mask.

C. Microchip Navigator

The large magnification needed in the imaging process to identify the microchip components entails a remarkable number of images to be stitched. Due to their dimensions it results in huge images, which are difficult to be handled by photo-viewer programs. Moreover, the information (meta-information) of the identified components as well as their interconnection must be also represented, which supposes an increase in computational cost. All these inconveniences require the development of a specific graphic user interface adapted to this problem.

The Microchip Navigator is presented as a program able to manage large images, represent the meta-information of the microchip components and it is a complete graphical user interface to define the nodes to interconnect those components. Besides, it implements an image alignment function to correct local misalignments that may have been produced in the image capturing.

1) Algorithm Description

The Microchip Navigator can be divided in two distinguishable modules: the navigation and the node definition.

Navigation deals with the management of large images, the representation of the meta-information of the previously identified components (see Fig.4) and the image alignment among the current and the adjacent microchip layer images. In order to solve the computational limitations to handle larges images, the proposed solution is similar to Google Maps, the mosaics are divided in tiles of 256x256 pixels and only those tiles requested by the user concerning his/her location within the layer are loaded in real time. The number of tiles to be loaded depends on the computer screen resolution and size. The navigation is performed in 3D since the layers are stacked; thus the user can navigate down or up through the microchip layers. It is important to highlight that the navigator computational cost does not depend on the total number of tiles of the microchip; the required tiles are loaded in real time avoiding hardware constraints.

The image alignment becomes an important function to define the interconnection between components in the annotation stage. A misalignment between layers may lead to errors in the node tracking process. The image alignment function implemented in the navigator uses anchor points previously defined by the user to shift layer regions locally according to the resulting translation vector among layers.

The node definition module uses hierarchical trees to define the interconnection among microchip components. These trees are formed by four different types of node points: beginning, corner, divergent and terminal. Beginning points use to be an output contact point of a logic gate, corners are used to join two tracks, divergent are like corners but it relates more than two tracks and terminal is where the node finished, which usually belongs to the input contact point of another logic gate. This module is currently assisted by the user who has to define the node points of the tree.

Once all the interconnections have been defined, the netlist can be automatically generated in a professional format, as for instance Verilog.

III. RESULTS

The entire system has been tested successfully over a microchip formed by several layers with a total number of pixels per mosaic superior to 40000x40000, whereas the number of logic gates reaches the order of tens of thousands. The results are presented according to the different blocks, where the computational cost is treated with special attention since it is an interesting parameter to quantify the performance of the automation. No computational cost is presented for the navigator, since it is assisted by the user.

A. 2D Image Tiling

A first implementation was tested using Pearson correlation, but its computational cost and efficiency lead us to develop the customized SIFT implementation. Regarding the computational cost, this implementation outperforms the Pearson correlation version by a factor of 20, whereas the descriptor robustness penalizes the computational cost by a factor 2.5 concerning the generic SIFT version. For instance, the average time required for the customized SIFT implementation for images of 800x640 pixels is around 2 seconds, whereas the generic version is more or less 0.8 seconds.

The matching performance is calculated for a sub-region of one of the metal layers formed by 1200 images. The correct matching percentage in the horizontal direction is about 96.94%, whereas in the vertical direction is 77.64%. However, the best combination of both directions defined by the region growing algorithm obtains a 99.17%.

B. Logic Gates Localization & Recognition

The results of this module contains the performance of the contact points detection algorithm and the logic gates localization & recognition.

A recall of 99.94% and a precision 77.77% is obtained for the contact points detection algorithm using a ground truth of 3500 contact points. In this case the recall is preferable rather than precision. A low recall affects the logic gate location & recognition performance, whereas precision only penalizes the computational cost.

The logic gate localization & recognition performance is more difficult to evaluate, since a help module was implemented to assist the user for the correct detection validations due to the high complexity of recognition stage. Therefore, the recall is out of interest and the attention is focused to the precision obtaining a value of 57.14% for the evaluated microchip.

The computational cost depends of the number of different logic gates patterns of the evaluated microchip and the number of instances to be located. In the evaluated microchip there are around 150 different logic gate patterns and more or less 15000 logic instances. An average time of 30 minutes for each set of 120 logic gates is estimated for the recognition process in a laptop. This time includes the processing time as well as the user interaction.

IV. CONCLUSIONS

In this paper a system to automate the mosaic reconstruction, the logic gates localization & recognition and the definition of the microchip components interconnection is presented. The system is divided in three blocks: 2D Image Tiling, Logic Gates Localization & Recognition and Microchip Navigator. First block uses our SIFT version to stitch mosaic images. The identification of the logic gates is achieved by means of the Pearson correlation as similarity measure function in the second block. To reduce the computational complexity the correlation is only applied to a very small set of candidate points. Finally, the Microchip Navigator is a rich user interface able to represent and align microchip mosaics as well as to define its components interconnection.

Future research lines are focused on the automatic tracking of the nodes that connect the microchip components. Moreover, the imaging process is also being improved using SEM technology aided by an electronically guided camera positioning.

ACKNOWLEDGMENT

The authors of this papers want to thank the contribution of graduate student Albert Martín.

REFERENCES

[1] Torrance, R.; James, D., "The state-of-the-art in semiconductor reverse engineering," Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE , vol., no., pp.333,338, 5-9 June 2011

[2] Lowe, D.; "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[3] Bay H., Tuytelaars T., and Van Gool L., "Surf: Speeded up robust features," European Conference on Computer Vision, 2006

[4] IMB-CNM CSIC Reverse Engineering Group. imb-cnm.csic.es/