Automated Probe-Mark Analysis for Advanced Probe Technology Characterization

Yu-Rong Jian  
Realtek Semiconductor Corporation, Hsinchu, Taiwan

Ferenc Fodor  
IMEC, Leuven, Belgium

Cheng-Wen Wu  
National Tsing Hua University, Hsinchu, Taiwan

Erik Jan Marinissen  
IMEC, Leuven, Belgium and Eindhoven University of Technology, Eindhoven, The Netherlands

Editor’s notes:  
This article describes a software tool for automated probe-mark analysis that minimizes the chance of human subjectivity and error.  
—Paul Bogdan, University of Southern California

Advanced packaging technologies in which multiple dies are integrated as a system in a single package have attractive benefits [1]. They enable heterogeneous integration, such that different functions can be implemented in the most suitable and cost-effective semiconductor technology. Large dies can be split up into high-yielding smaller dies. Innovative interconnects such as arrays of micro-bump and through-silicon vias provide high-density, high-bandwidth, and low-energy interfaces between adjacent dies and reduce the footprint of the system chip. In the past decade, we have seen increasingly more of these 3D-stacked IC (3D-SIC) products on the market, especially for memories, high-performance, and mobile applications [2], [3].

Before stacking, the dies for a 3D-SIC need to be subjected to a prebond test to obtain an acceptable compound stack yield. Prebond tests are typically performed at wafer level and always involve some form of probing to gain electrical access to the design-for-test infrastructure built into the die [4]. While the bottom die(s) of the stack offer regular probe access via their interface to the package substrate, that is not the case for the nonbottom dies in the stack which only have micro-bumps as functional interfaces. These micro-bumps typically come in large fine-pitch arrays, which cannot be probed with conventional probe technologies. To gain test access, we have two options. We can extend the die design with dedicated prebond probe pads, but this is expensive, as it requires extra design efforts, silicon area, and test time, and still leaves the actual functional interface through the micro-bumps untested [4]. Alternatively, we can employ advanced probe technology to probe these large-array fine-pitch micro-bumps [5]–[7].

In semiconductor wafer probing, the probe tips normally leave marks on the probe targets. Probe targets can be wire-bond pads, dedicated probe pads, Cu pillars, or micro-bumps. In this work, we characterize and evaluate advanced probe cards meant for probing micro-bumps. We use the probe marks to determine if indeed the probe tips landed correctly on the probe target. Also, we want to ensure that the probe marks do not negatively
affect the primary function of the probe targets if that happens to be different from merely probe access. Therefore, we would like to know the relative position and size of the probe mark.

We have developed a Python-based software tool for automated probe-mark analysis for micro-bumps to characterize the quality of probing of large-array fine-pitch micro-bumps in the context of prebond testing of 3D-SICs. Automatic analysis of probe marks was previously reported by Wang et al. [8]. They used image-processing techniques for the analysis of probe marks on bond pads. They extracted the pads from the microscope images, and then found the probe mark on each pad. In another publication [9], the authors proposed an algorithm for the inspection of IC pads and bond pads by means of image-processing. They extracted the pads and identified the probe marks by doing a XOR operation on two images, one with the marks and the other with the marks filled up. Then, they identified the probe mark, and computed the probe mark area.

In our application, Ø25-µm micro-bump arrays serve as probe targets, which are significantly smaller than regular probe pads. Furthermore, the advanced probe cards used leave a diagonally shaped probe mark [5]. The probe marks are very light, and the probe mark area is a very small percentage of the probe target. Moreover, the related prior work covers only a single probe mark per probe target, while our algorithm also handles multiple probe marks per probe target.

For JEDEC wide-I/O2 interfaces consisting of 1752 micro-bumps at 40-µm pitch, scanning electron microscope (SEM) images of probe marks on the four corner micro-bumps of the array serve as input for our analysis tool. Using image-processing routines, the tool identifies the contours, centers, and surface areas of both probe target and probe mark. Subsequently, it calculates the probe-mark’s offset from the center of the probe target. These offsets are used to separate the contributions of both probe station and probe card to the overall probe-to-pad alignment (PTPA) accuracy of the test system [7]. The area ratio of probe mark and probe target is a generally accepted measure of the probe’s “damage” to the probe target. In addition, our tool can distinguish multiple probe marks on the surface of the micro-bumps.

Previously, the measurement of the offsets of the probe marks on the micro-bumps was done manually; we printed images of probed micro-bumps on paper and used a ruler to measure the probe-mark offsets. However, the manual procedure was time consuming. Therefore, our objective was to reduce the time spent on measurements, by automatic analysis of the images of probed micro-bumps.

In this article, we compare manual and automatic measurements. The results show that the automatic approach yields comparable results with the measured offsets we have gathered by manual measurements, but at a much faster speed. Moreover, the algorithms in our software tool are deterministic, such that different runs on the same data will give identical results, while the results of the manual approach were often influenced by human subjectivity. In addition, we also show our automated approach to be robust in the sense that the tool handles top-view as well as tilted-view images, images with varying SEM contrast settings, and multiple probe marks (due to multiple touch-downs) per probe target.

Data preparation

The identification of the probe marks on the micro-bumps is a multistep process, following the flow shown in Figure 1. Due to its very strong image processing library, we have chosen to implement the algorithm in Python. Starting from SEM images, several standard image processing steps are used on the raw data in conjunction with self-developed algorithms. For an image of size $M \times N$ pixels, most of the algorithm steps have a complexity of $O(MN)$; the only exception is the edge detection step, which has a complexity of $O(MN \log MN)$ (see Figure 1) [10], [11].

Each time a micro-bump image is taken using one of our SEM tools, the resulting file is a combination of three pictures of the same location, in different, predefined conditions. These conditions provide distinct micro-bump images under different “lighting” conditions, as shown in Figure 2a. We refer to these images as Images 1, 2, and 3. Even though all three images are of the same micro-bump, once edge detection algorithms are applied, the resulting contours have a large variation, as shown in Figure 2b–d. Edge detection is a basic operation in image processing literature on feature detection and extraction. We have implemented three of the most-used edge detection algorithms, viz. Prewitt, Sobel, and Canny [10], [11]. We experimented...
with these three algorithms and noted that in our specific application the Canny algorithm had the most-complete edges of the micro-bump and probe mark and generates the least number of unwanted edges. Hence, we decided to focus on the Canny algorithm in the rest of this article.

We have observed that Image 1 results in the cleanest micro-bump contour. Therefore, we decided to use Image 1 solely to identify micro-bump contours. Due to occasionally unclear edges, some of the contours could not be detected by Canny edge detection only. The main cause turned out to be blurry input images. To address this challenge, we added an image sharpening operation before Canny edge detection. Image sharpening is a commonly used technique to increase the apparent sharpness of an image [11]. The combination of image sharpening and Canny edge detection is applied on all our input files as a preprocessing step in our analysis.

As the micro-bumps are cylindrical, the contour of their probe surface is in fact a circle. However, tilted-view images of micro-bumps are more valuable, as they also contain processing-related information about the height uniformity of the arrays. For this reason, we prefer to take tilted-view images of our probe marks. In these tilted images, the micro-bump contour becomes an ellipse.

To identify the contour, the algorithm first finds the top ($P_{top}$), left-most ($P_{left}$), and right-most ($P_{right}$) edge pixels in the image. Subsequently, it calculates the mid-point ($O_{LR}$) between $P_{left}$ and $P_{right}$ as shown in Figure 3a. Theoretically, $O_{LR}$ should be the center of the micro-bump contour. However, when we used $O_{LR}$ as an origin of the ellipse, we ended up identifying an incorrect contour of the micro-bump, as...
shown in Figure 3b. To prevent this from happening, we decided to also calculate the mid-point of the micro-bump in the y-direction. Hence, first we need to find the bottom edge pixel \( P_{\text{bottom}} \) to get the correct y position of the micro-bump center.

Taking advantage of the symmetrical properties of an ellipse, we can use \( P_{\text{top}} \) and \( O_{LR} \) to find the missing point. The difference in the x-direction between \( P_{\text{top}} \) and \( O_{LR} \) is equal to the difference in the x-direction between \( O_{LR} \) and \( P_{\text{bottom}} \). At first, we find the point \( P \), as shown in Figure 4. \( P \) has an x-coordinate identical to \( P_{\text{bottom}} \) and a y-coordinate identical to \( O_{LR} \). Next, we find the first edge pixel that we meet from \( P \) while traveling downwards in the y-direction, as indicated by the arrow in Figure 4. This first encountered edge pixel will be considered as \( P_{\text{bottom}} \).

We define the center of our micro-bump contour \( O_{\text{MB}} \). The x-coordinate of \( O_{\text{MB}} \) remains the same as \( O_{LR} \), but the y-coordinate is defined by the mid-point between \( P_{\text{top}} \) and \( P_{\text{bottom}} \). Figure 3c shows the two newly identified points, and Figure 3d illustrates the contour drawn with \( O_{\text{MB}} \) as its origin. We can see that we have correctly identified the micro-bump contour, i.e., \( O_{\text{MB}} \) is indeed the correct center of the micro-bump. Using this technique, we can always ensure to find the correct center and contour of the micro-bump.

Images 2 and 3 are best for the identification of the probe mark itself. Each image contains distinct information on the probe mark, thus merging the two images results in the most complete probe mark contour. After edge detection, the pixels of the image become an array of “0s” (background) and “1s” (edge). In this article, merging refers to performing a pixel-by-pixel OR operation on two images. The resulting array is then used for probe mark identification.

Next, we identify the probe mark, using the result of merging Images 2 and 3. However, the image also contains a significant number of edge pixels of the micro-bump contour as well as some noise. This can distract from identifying the correct probe-mark location as the number of edge pixels in the proximity of the probe mark may be lower than in some parts of the contour. Therefore, we filter out the edges at the outer boundary of the micro-bump contour; we call this contour removal.

**Automated probe mark recognition**

For the software to be able to detect the probe mark, we need to provide information about the expected size of the probe mark. The probe cards used for probing the micro-bumps have arrays of probes with diagonally placed, square probe tips. The actual contact between probe tip and probe target happens only with the heel of the probe tip, hence probe marks typically have a diagonally oriented rectangular shape [6].

In our software tool, we use a tilted rectangle to represent the bounding box of the probe mark and scan the image to find the position where the rectangle encloses the largest number of edge pixels.

The dimensions of the bounding box that is used during probe-mark recognition depend on 1) the diagonal angle (\( \theta \)) of probe tips and probe mark, the lengths of the 2) horizontal and 3) vertical axes of the micro-bump contour, expressed both in millimeters and in pixel count. For top-view images, \( \theta \) is 45°, which corresponds to the angle of the probe needles on the probe card. Knowing this, we calculated the angle for the tilted-view images and found \( \theta \) to be approximately 30°. The size of the bounding box is also determined by the probe tips of the card [5]. For tilted-view images, we use a bounding box of 6 \( \mu m \times 1.5 \mu m \) to generate the rectangle. For top-view images, we use a bounding box of 6 \( \mu m \times 2.5 \mu m \).

Next, we use the rectangle to scan the image. The number of edge pixels inside the rectangle is assigned as weight to its center pixel. Eventually, we find the pixel with the maximum weight \( (W_{\text{max}}) \). We conclude that the pixel with the largest weight represents the center point of the probe mark.
In case there is no probe mark on the micro-bump, \( W_{\text{max}} \) is significantly smaller than in the presence of an actual probe mark. This allows us to set a threshold value \( T_{\text{pm}} \) to decide whether there is a probe mark on the micro-bump. If \( W_{\text{max}} \) is less than \( T_{\text{pm}} \) during the first iteration of the analysis loop, we conclude that there is no probe mark. After identifying the centers of micro-bump contour and probe mark, we can calculate the offset between them.

In general, a micro-bump is probed only once, as every time we probe, we are damaging the surface of the micro-bump. However, there are situations when a micro-bump array must be touched on multiple times, resulting in multiple, sometimes overlapping probe marks. This in return makes probe mark identification more complex. Recognition of multiple probe marks only considers the cases where the overlapping area of the marks is small enough, such that they can be distinguished, as shown in Figure 5.

First, we use connected-component labeling to remove the noise outside of the bounding rectangle. For a given image, connected-component labeling assigns a unique label to each adjacent set of pixels with similar features [11], [12]. We consider only those connected components that are totally or partially overlapping with the rectangle and remove all other components. Next, we use the ratio between the width of scanning rectangle \( (d_r) \) and the probe-mark contour \( (d_m) \) to check if the probe marks are overlapping. If the ratio is greater than a user-defined threshold, the algorithm first separates the probe marks. To do so, we find the mid-line \( L_m \) of the probe mark’s contour, as shown in Figure 5c. If the center of the rectangle with the largest edge-pixel weight is to the right of \( L_m \), we separate the edge pixels on the right of \( L_m \). If the center is between \( L_m \) and \( L_e \), we remove the edge pixels on the right of \( L_m \). The separated probe mark is analyzed following the procedure for single probe marks. Finally, we loop through the algorithm as long as \( W_{\text{max}} \) is larger than \( T_{\text{pm}} \), i.e., there are still probe marks on the image.

In the penultimate step of the algorithm we calculate the probe-mark area. It starts with determining the outer shape of the probe mark. For this, we scan the image along the horizontal and vertical axes. The horizontal scanning is performed from both sides along the x-axis, i.e., left-to-right (east-bound) and right-to-left (west-bound). We connect the first edge pixel we meet during the east-bound scan with the first edge pixel we meet during the west-bound scan. The vertical scanning is done in a similar manner, this time connecting the edge pixels from top to bottom. Next, we take the union of the two images because there are some holes around the probe-mark contour if we choose the intersection operation.

However, we found that there are still some unfilled pixels inside the probe mark. The algorithm then repeats the scanning process, but this time takes the intersection of the images resulting from the two respective scans. The reason is that there may be extra pixels extended beyond the original probe-mark contour if we perform the union operation. Finally, we get the shape of the probe mark. To determine the probe-mark area, we count the number of pixels inside the shape of the probe mark. As we know the sizes of micro-bump and probe mark in terms of pixels, as well as the actual size of the micro-bump, we can calculate the actual probe-mark area and its relative size with respect to the micro-bump. As the final step of the algorithm, the data gathered during the analysis is saved in a text file.

Experimental results

We have validated the correct operation and accuracy of our new algorithm on synthetic images of micro-bumps with probe marks. Subsequently, we have used our algorithm on sets of SEM images of real micro-bumps to assess the functionality, accuracy, and flexibility of our automated approach.

To validate our new software tool, we were in need for micro-bump images for which the probe mark location was known \textit{a priori}, such that we could check if the results of the algorithmic approach were indeed correct. We generated two data sets of 1,000 synthetic micro-bump images each, in which we placed a single probe mark at
random, yet known, coordinates on the surface of each micro-bump.

The first data set contained images with a fixed probe mark size while in the second data set we used probe marks with varying sizes, shapes, and rotations. Figure 6 summarizes the differences between the known and detected coordinates for both the detection of the micro-bump contour as well as the probe mark itself.

The first conclusion from Figure 6 is that our software tool correctly finds the contour of the micro-bumps for all 2,000 images. When it comes to the detection of the probe mark, we can see that fixed-size probe marks result in more accurate detections—which is to be expected, given that the scanning rectangle we use for the detection of the probe mark has a fixed size and angle, as discussed in the previous section. Nevertheless, when we put the results in context, i.e., translate the pixel values back to micrometers, we can see that at its worst, the algorithmic approach is within ± 0.5 µm accuracy with a mean error of 0.01 and – 0.02 µm (stdev: 0.07 and 0.04 µm) for the x-axis and y-axis, respectively.

Next, we have used our new software tool on three data sets with actual SEM images of actual probed micro-bumps coming from various Ø300-mm wafers with wide-I/O1 and wide-I/O2 micro-bumps [5]–[7]. The first data set contained 70 probed micro-bump images. Typical examples of the outputs of the new software tool are shown in Figure 7. The tool generates images of the analyzed micro-bumps, placing a Cartesian coordinate system through the mid-point of the micro-bump, as well as highlighting the identified probe mark. The example in Figure 7a shows an

![Figure 6. Validation results on synthetic images with (a) fixed and (b) varying probe mark sizes.](image)

![Figure 7. Outputs of the analysis: (a) single probe mark, (b) no probe mark, (c) and (d) overlapping probe marks, and (e) uses a top-view image as input.](image)
interesting case where the probe needle landed right on the edge of the micro-bump. Even for such an extreme case, the software was able to find the probe mark. In Figure 7b, we show an example of an analyzed micro-bump that was not probed. The algorithm correctly returns a “No Probe Mark” message.

The second data set, with 20 micro-bump images, was used for developing the algorithms for multiple probe mark recognition. An interesting case is where the probe marks are slightly overlapping yet are still distinguishable. Such an example is shown in Figure 7c and d, confirming that our proposed solution of separating the probe marks manages to do the job. Using these two data sets, we configured our algorithm, such that it can analyze all 90 samples correctly and without any operator intervention.

Finally, we also tried top-view images with different contrast conditions. In the previous section, we described the required modifications in our software for top-view images to be working, which mainly concern the size of the scanning rectangle. For top-view images, all samples were correctly analyzed. One example is shown in Figure 7e.

The third data set contained images in different contrast settings of probed micro-bumps. While taking the micro-bump images on the SEM tool, we have tried different combinations of the most used voltage and contrast settings of the SEM tool, to test out the capabilities of our algorithm, without touching the code. The result is summarized in Table 1.

We used five different contrasts and three different voltage levels. In the table, “C” denotes the case when the tool managed to identify the contour of the micro-bump, and “P” indicates successful detection of the probe mark. In most cases the tool managed to identify both “CP.” However, for voltage level B, our software seemed to struggle, independent of the contrast settings. When we investigated the input files, we observed that the failing cases were out-of-focus images. This means that even though we apply image sharpening at the beginning of the analysis, there is still a minimum level of focus that is required for a successful analysis. Therefore, it is important to have input files of high-quality images. Fortunately, this can be taken care of when the images are taken on the SEM tool. Our experiments on both synthetic as well as real images of probed micro-bumps confirm the benefits of automated versus manual probe mark detection and analysis.

The main benefit is repeatability: the automated approach is more accurate and, unlike the manual approach, always produces the same result for the same input. The second benefit is a drastic reduction in analysis time. It took us about 5 h to analyze the 115 micro-bumps from data sets 1–3 by hand, which comes down to about 154 s per micro-bump; our new software, on the other hand, did the same job in approximately 4 min, averaging to about 2.1 s per micro-bump. Moreover, the automatic analysis offers extra features such as automated double probe mark detection and analysis and probe mark area calculation; these options are not available, or very inefficient to achieve by hand.

In this Article, we have presented an in-house developed software tool for automated probe mark analysis. The software has replaced the rather time-consuming manual analysis, which was previously used for probe-mark characterization. Using the tool, we have excluded the subjectivity of the person doing the measurements and consequently, the accuracy has improved. Furthermore, we have drastically reduced the time spent on the analysis. In addition, automation has enabled new features not available in the manual approach, such as double probe mark detection and analysis and probe mark area calculation.

Furthermore, we have worked on and tested the flexibility of the tool with the help of SEM images of probed micro-bumps under various “lighting” and contrast settings.
References


Yu-Rong Jian is a Digital IC Designer at Realtek Semiconductor Corporation, Hsinchu, Taiwan. Jian has an MSc in electrical engineering from the National Tsing-Hua University, Hsinchu (2018).

Ferenc Fodor works at IMEC, Leuven, Belgium, as a Test Development Engineer, focusing on advanced probe card technologies and process characterization in the context of 3D-stacked ICs from 2016. Fodor has a bachelor’s degree in industrial automation and applied informatics from the Technical University of Cluj-Napoca, Cluj-Napoca, Romania (2015).

Cheng-Wen Wu is with the Department of Electrical Engineering, National Tsing-Hua University, Hsinchu, Taiwan. Wu has a PhD degree in electrical and computer engineering from the University of California at Santa Barbara, Santa Barbara, CA. He is a Fellow of IEEE. He is also a member of the Advanced Research Advisory Committee and the Industrial Technology Research Institute (ITRI).

Erik Jan Marinissen is the Scientific Director at IMEC, Leuven, Belgium, and a Visiting Researcher at the Eindhoven University of Technology (TU/e), Eindhoven, The Netherlands. Marinissen has PDEng and MSc degrees in computing science from TU/e. He is a Fellow of IEEE.

Direct questions and comments about this article to Yu-Rong Jian, Realtek Semiconductor Corp., Hsinchu City 300, Taiwan; yrjian_larc@gapp.nthu.edu.tw.