

Dear CATS User:

On behalf of the CATS team I am pleased to present to you the latest release of CATS News. The mask manufacturing community is under stress to deliver high quality masks for sub-10nm technology nodes while, at the same time, maintaining very high mask throughput for masks for IoT, automotive and other consumer products on legacy nodes. Mask data preparation is an important step in meeting these end customer demands and we continue to innovate to deliver high performance for MDP and capabilities to ensure high mask quality. In this edition of CATS News we provide articles on improving turn-around time, mask data verification and additional capabilities to improve mask quality.

I welcome your feedback on the articles and ideas for what you would like to see in future editions of CATS News.

Anjaneya Thakar

CATS, Product Marketing Manager

CATS news

CATS® and SmartMRC®—A Happy Marriage of Gold-Standard Tools

Dr. Kokoro Kato, CATS R&D

Synopsys acquired the SmartMRC business unit from Hitachi High Technologies Corporation in August 2015. The intent of this acquisition was to achieve a synergy between CATS® and SmartMRC®, because CATS® is a fracturing tool and SmartMRC® is a verification tool. So far, this combination has exceeded expectations.

What is SmartMRC®?

SmartMRC® was first developed more than 12 years ago to provide verification functionalities required for Mask Rule Check (MRC). It comprehensively covers the verification requirements from mask shops around the world, and (similar to CATS® for fracturing) is the de facto standard in mask data verification. The following illustration represents the fundamental MRC functions in SmartMRC®.

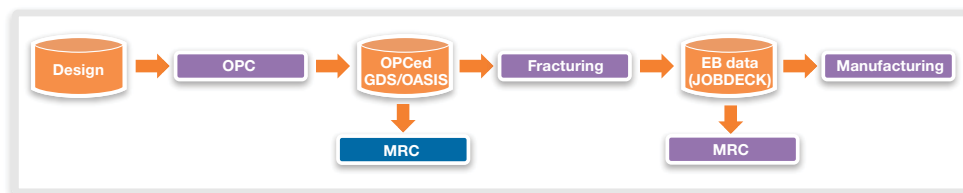


Figure 1. Fundamental SmartMRC Functions

SmartMRC® is used not only by mask shops but also in the OPC divisions of semiconductor companies and foundries. As shown in the following diagram, OPC divisions use SmartMRC® for verification before signing off on OASIS files. Mask shops use it as the final QA tool before mask fabrication.

Continued on page 2

In This Issue

CATS® and SmartMRC®—A Happy Marriage of Gold-Standard Tools	1
TCUT for Reducing Shot Count and Sliver on VSB Tools	3
Improve Analysis with CATS Shot Count Estimator	5
Asymmetric SCALE Operations for Advanced Processes	6
Mask Metrology Support for KLA-Tencor LMS IPRO6 Tool	8

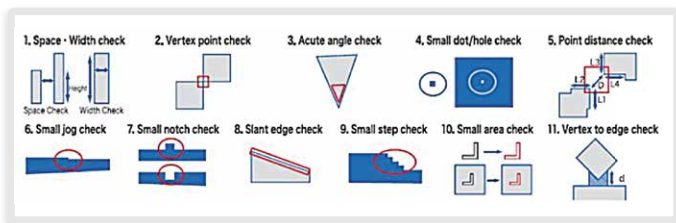


Figure 2. SmartMRC in Manufacturing Flow.

Expected Synergies

After the acquisition, our respective teams convened to discuss how to maximize the synergy between these two gold-standard tools, and as a result, numerous new and creative solutions are being developed, some of which have already been implemented and made available. These new solutions include:

1. Error Browsing Functions in the CATS Graphical Interface

Because CATS already possesses a stable and familiar GUI, it was a natural extension to enable browsing of MRC errors coming out of SmartMRC®. An optimized error browser was something already on SmartMRC users' wish list, and the development team moved quickly to make this a reality. Moreover, in addition to MRC errors, the error browser can also display the results of density calculations.

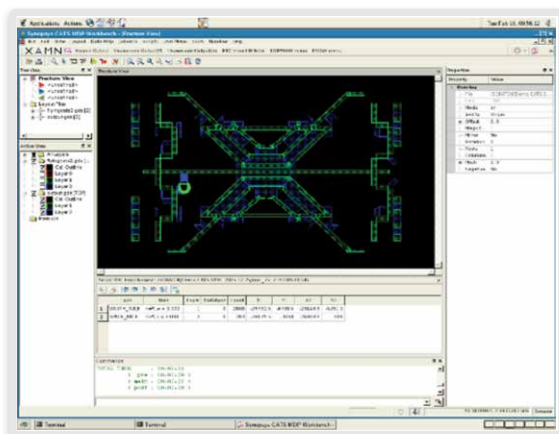


Figure 3. SmartMRC® for Verification

2. Applications of Pattern Matching Functionalities

Several types of Fuzzy Pattern Matching (FPM) are available in SmartMRC®. The application already provides a marking function for mask CDSEM in the CATS GUI utilizing this FPM. Additionally, SmartMRC® provides a unique algorithm that compares the Manhattan shapes of the mask layout against the curvilinear shapes in SEM images. FPM is a flexible tool that can be creatively applied to a wide variety of applications.

3. Pre-Processing of Fracture Operations

SmartMRC® performs global checks for POLO (Post OPC Layer Operation). POLO has been a critical process for many customers, and the slow TAT by hierarchical DRC tools has been a problem. The implementation of global checks, such as TOUCH or INTERACT, was historically problematic in a DP environment due to the interaction of data. SmartMRC® addressed this obstacle by utilizing an optimized stitching process between DP workers, achieving fast performance. The global check functions will bring great value to CATS customers with such POLO operations.

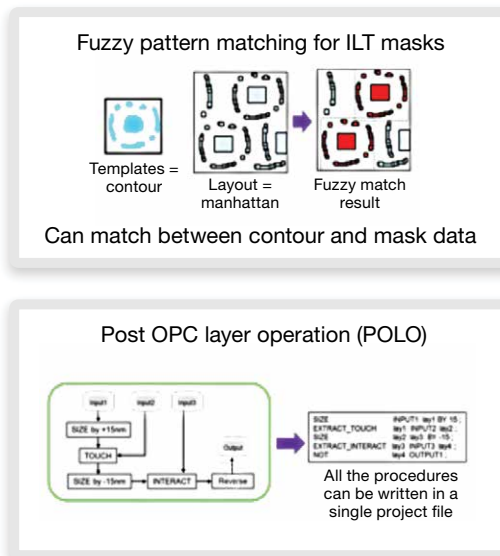


Figure 4. SmartMRC and Pre-Processing

CATS® and SmartMRC® make for a powerful combination, aimed at improving customer satisfaction and utility. We look forward to a future in which these tools can be leveraged in novel ways to provide a myriad of new solutions.



TCUT for Reducing Shot Count and Sliver on VSB Tools

John Burns, CATS CAE

The CATS ability to generate VSB12 data with an alternative algorithm goes back over six years, but in the past year, some specific enhancements were made to revitalize this “TCUT” method. These enhancements can significantly reduce your shot count, and thus machine write time. That being said, there are two key aspects of your data that can greatly affect the impact TCUT will have at your site.

Size Matters

TCUT processing will not even be triggered if your input polygons are smaller than a single shot size for your intended VSB12 machine. In the CATS CINC file, this value is controlled by

```
VSB_T MAX_SHOT_SIZE 0.50
```

as that corresponds to what can be exposed in a single shot on your tool.

Thus, if you have polygons that resemble this:

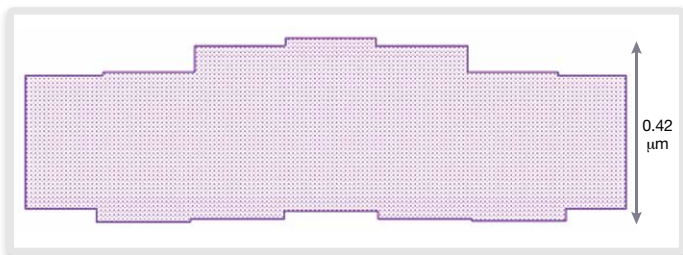


Figure 1. Polygon with Dimensions Smaller Than MAX_SHOT_SIZE

with a maximum height (0.42 μm) smaller than the MAX_SHOT_SIZE (in this case 0.5 μm), there is no point in enabling TCUT.

Are Your Vertices Aligned?

Suppose you have polygons such as those shown in Figure 1, but your tool supports a MAX_SHOT_SIZE of 0.25 μm . Such data might benefit by using TCUT. The next aspect that is critical to understand for qualifying the impact of TCUT is how your polygons are drawn. If your vertices from one side to another line up, such as those shown in Figure 2, then the output of TCUT will not have an advantage over the default BASIC shot cutting in CATS®.

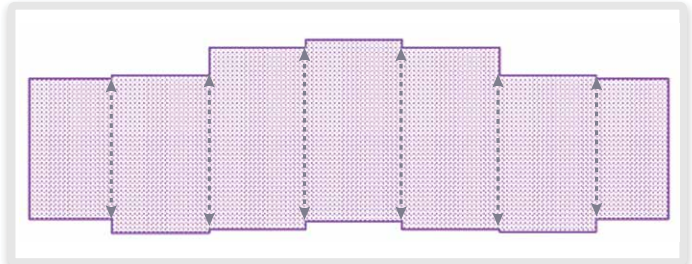


Figure 2. Polygon with Aligned Vertices

Frequently, OPC correction does not ensure aligned vertices. Subsequently, even simple process biasing to aligned data could result in vertices becoming misaligned. Also, if you are running a fracture with REVERSE YES, that too can create misaligned vertices in the fractured data, coming from the space between the input polygons

An Example of an Excellent TCUT Candidate

Figure 3 shows how a polygon with misaligned vertices would fracture with the default VSB_T SHOT_OPT BASIC, method in CATS® for a VSB12 machine with a MAX_SHOT_SIZE of 0.25.

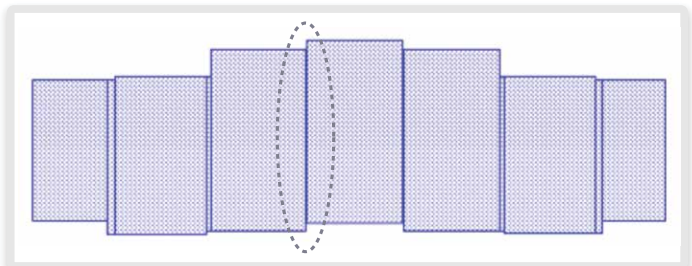


Figure 3. Polygon with misaligned vertices fracture without TCUT

In Figure 3, the cuts are all vertical, and none of the vertices are aligned. Indeed, in the area circled in red, there is actually a narrow rectangle of only two nanometers. The total number of shots to expose this polygon is 26 (2x13 figures) when considering the extra slicing that will automatically occur at the exposure tool if the dimensions are greater than MAX_SHOT_SIZE for that particular machine.



However, because this polygon has features that are taller than the MAX_SHOT_SIZE with misaligned vertices, if you select the VSB_T SHOT_OPT TCUT option, the results would resemble Figure 4:

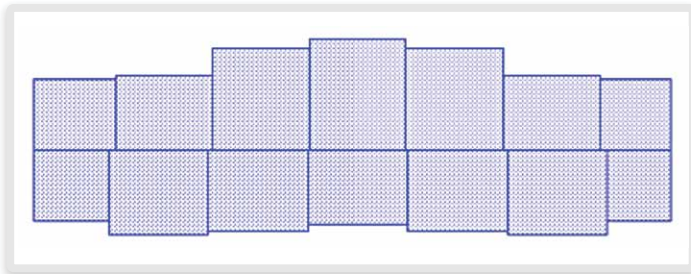


Figure 4: Polygon with misaligned vertices fracture with TCUT

Note that there are still vertical cuts, but they stop at the middle, forming a T-intersection with the one long horizontal cut now created, hence the name TCUT for this approach.

More importantly, the total number of shots to expose the exact same polygon is now 14 (no additional exposure tool slicing is required). For more information on shot count estimates, refer to the *“Improve Analysis with CATS Shot Count Estimator”* article in this edition.

Something else to note here is that due to the nature of TCUT, and the input misalignment of vertices in this specific example, there are now no sliver figures with this approach.

Of course, the results will be data dependent, and you should test TCUT on your flow and data. There are no additional licenses required.

To utilize TCUT fracturing, simply use the following command:

```
VSB_T SHOT_OPT TCUT
```

This functionality is available only in CATS Version K-2015.09-SP6 or later. Additionally, the enhancements for TCUT were made as part of an architectural upgrade to the CATS engine. Please contact your local CAE to ensure that your existing hardware and CATS installation will support that upgrade.



Improve Analysis with CATS Shot Count Estimator

William Moore, CATS R&D

The CATS shot count estimator is used to estimate the number of shots required to print a pattern for variable-shape beam (VSB) tools fractured using the PRODUCE VSB_T command. The shot count information is reported to both the screen and the CATS log file. This information can help users analyze the most efficient slicing strategies (such as BASIC or TCUT). The shot count in turn can be used to judge the potential length of a given exposure and therefore also the tool cost.

Currently supported formats include VSB12 and VSB12i for NuFlare tools.

The shot count estimator is enabled using the VSB_T SHOT_ESTIMATE YES command. Only flows with DIRECT_OUTPUT YES (not using WRITEFILE) are supported for shot count estimation.

The following example shows a typical CATS include file (CINC) used for shot count estimation for FORMAT VSB12 (commands required to use the shot count estimator highlighted):

```
CLEAR
FORMAT VSB12
PRODUCE VSB_T
RESOLUTION 0.001
DIRECT_OUTPUT YES
INPUT flyingcats2.gds
STRUCT ARRAY
VSB_T SLIVER_WIDTH 0.1
VSB_T MAX_SHOT_SIZE 1
VSB_T HALO_SIZE 12
VSB_T SHOT_OPT TCUT
VSB_T SHOT_ESTIMATE YES
OUTPUT output_00/chip.cnf
```

Figure 1. Shot Count Estimation for VSB12

For this CINC file, the shot count information is reported as:

Shot count estimate: 2248 shots, 375 figures
(20 rects + 295 xtraps + 60 ytraps)

In contrast, for FORMAT VSB12i, a shot count estimate is reported per attribute. A typical CINC file resembles:

```
CLEAR
FORMAT VSB12i
PRODUCE VSB_T
RESOLUTION 0.001
DIRECT_OUTPUT YES
INPUT flyingcats2.gds
STRUCT ARRAY
VSB_T SLIVER_WIDTH 0.1
VSB_T MAX_SHOT_SIZE 1
VSB_T HALO_SIZE 12
VSB_T SHOT_OPT TCUT
VSB_T SHOT_ESTIMATE YES
ATTRIBUTES ./layers.attr
OUTPUT output_00/chip.cnf
```

Figure 2. Shot Count Estimation for VSB12i

For this CINC file (the test case is different due to the use of attributes), the shot count information is reported as:

Shot count estimate for attribute 0: 1925 shots, 353 figures
(20 rects + 277 xtraps + 56 ytraps)

Shot count estimate for attribute 1: 92 shots, 4 figures
(4 rects + 0 xtraps + 0 ytraps)

Shot count estimate for attribute 2: 300 shots, 18 figures
(0 rects + 18 xtraps + 0 ytraps)

The shot count estimator has been designed to match the shot count estimate produced by the NuFlare shot count estimator tool for VSB12 and VSB12i, and testing confirms that it does. With the CATS shot count estimator, the estimate comes for free with a fracture with negligible additional performance overhead.

The shot count estimator is available in CATS Version K-2015.09-SP6 and later.



Asymmetric SCALE Operations for Advanced Processes

Peter Moberts, CATS R&D

During fracturing, SCALE operations have long been employed to define a scale factor for either the primary and/or the secondary file. Leading-edge processes, such as EUV masks, have necessitated that CATS® introduce new functionality to handle asymmetric scaling. With the development of our improved fracture engine (from CATS Version K-2015.09), CATS® can flexibly handle such functionality while minimizing associated risk.

First, CATS® added the EXPRESSION command, which can be used to define one or multiple Booleans between buffers and/or files. Besides Booleans, the EXPRESSION command also can contain a SIZING or SCALE operator. The EXPRESSION command works in combination with the SLOT and FUNCTION commands. Consider the following simple example of these commands using an expression:

```
Command: Slot S1          ! Definitions for slot S1
Command: Input myfile.gds ...
Command: Layer 1          ...
Command: Slot S2          ! Definitions for slot S2
Command: Input myfile.gds ...
Command: Layer 2          ...
Command: Output xor_01.cft
Command: Expression final = XOR (SIZING (S1, 0.1),
SIZING (S2, -0.1))
Command: Function EXPRESSION final
```

In this example, two slots are defined, with each slot having its own sizing, and the results of those sizings undergoing an XOR.

Moreover, it is possible to define *multiple* expressions. The following example shows the previous example rewritten using multiple expressions, including SCALE as a final step:

```
.....
.....
Command: Expression T1 = SIZING (S1, 0.1)
Command: Expression T2 = SIZING (S2, -0.1)
Command: Expression T3 = XOR (T1, T2)
Command: Expression final = SCALE (T3, 4)
Command: Function EXPRESSION final
```

This new functionality will become available for flows that use the improved fracture engine, e.g. the conversion of MODE5 to VSB12. Not all flows are currently supported, so please consult the CATS manuals and release notes for the details pertaining to your flow. Additionally, please refer to the CATS *Command Manual* for detailed descriptions of the SLOT, EXPRESSION, and FUNCTION commands.

Both the SCALE command and the SCALE operator apply symmetrical scaling in X and Y. Now, the SCALE operator in an EXPRESSION command has been extended to accept a *ScaleX* and a *ScaleY* value. With this extension, a design can be scaled differently in X and Y. A symmetrical scale can be applied with a single value. When two values are specified, the first one is used as value for *ScaleX* and the second one for *ScaleY*. When the *ScaleX* value is larger than *ScaleY*, the design is stretched in the X-direction. When *ScaleY* is larger than *ScaleX*, the design is stretched in the Y-direction.

In the following example, the x-coordinates are scaled by 2 while the y-coordinates are kept the same:

```
Command: Expression T1 = .....
Command: Expression T2 = SCALE (T1, 2.0, 1.0)
```

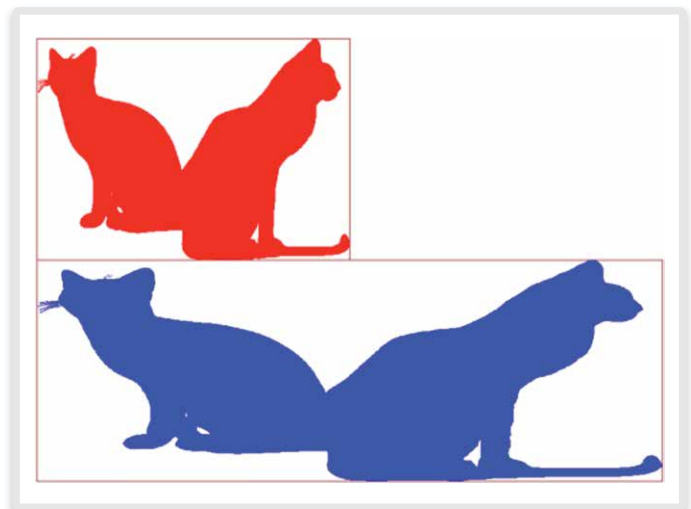


Figure 1. X-coordinate Scaling Showing Original Shape (Red) and Scaled Shape (Blue)

In contrast, in the following example, the y-coordinates are scaled by 2 while the x-coordinates are kept the same:

```
Command: Expression T1 = .....
Command: Expression T2 = SCALE (T1, 1.0, 2.0)
```

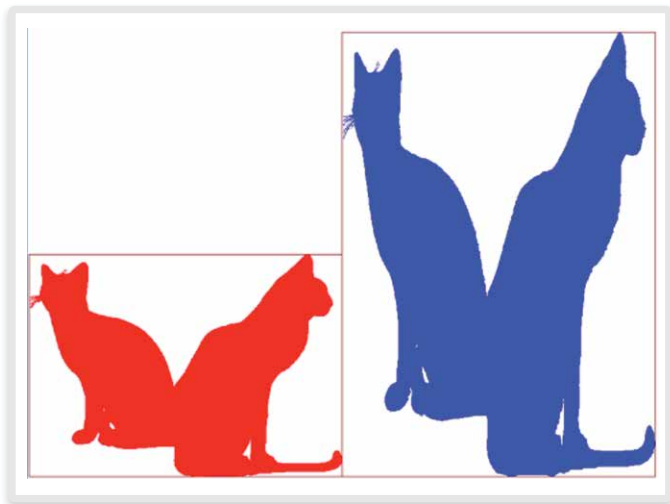


Figure 2. Y-coordinate Scaling Showing Original Shape (Red) and Scaled Shape (Blue)

In the following example, the x-coordinates are scaled by 3 and the y-coordinates are scaled by 2:

Command: **Expression** T1 =

Command: **Expression** T2 = SCALE (T1, 3.0, 2.0)

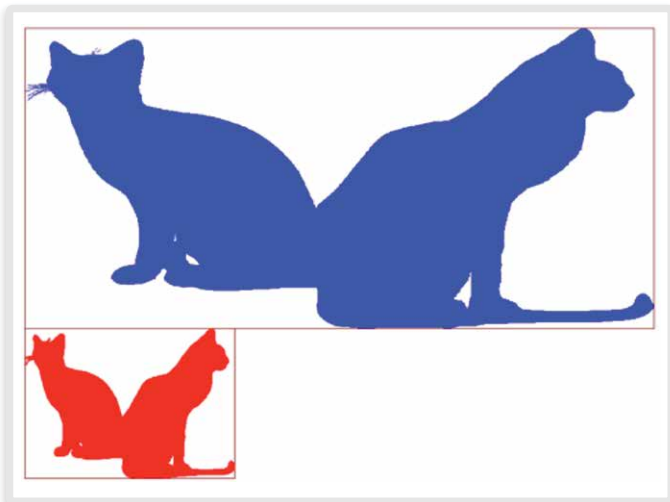


Figure 3. X- and Y-coordinate Scaling Showing Original Shape (Red) and Scaled Shape (Blue)

Verifying the “correctness” of output can be challenging with this type of operation, because the output file is, by definition, stretched compared to the input file. Therefore, to validate an asymmetrically scaled file, an inverse scale can be applied to the stretched result,

resulting in a symmetrical result once again.

(1) Input -> Scale (3, 2) -> Output_1

(2) Result -> Scale (2, 3) -> Output_2

(3) Input -> Scale (6) -> Output_3

The net scale of Output_2 is 6 and should be the same as Output_3. Because of the order of operations, slight differences are expected between Output_2 and Output_3 for non-Manhattan data.

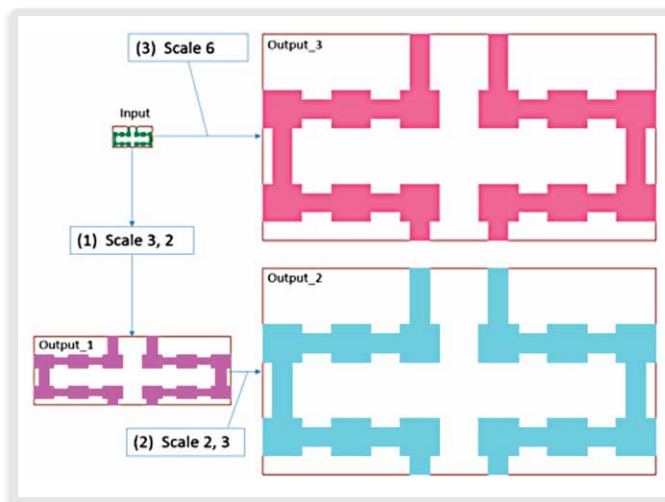


Figure 4. Verification

EUV lithography is expected to be used in production for the 5nm node, if not sooner. Wafer throughput has been a particularly challenging item for this novel technology with a move toward higher NA (above 0.5) being regarded as a way to work around this issue. Because the EUV mask is a reflective mask, the pupil IN and pupil OUT cannot overlap. In order to raise the NA, an “anamorphic” optics strategy has been proposed where the magnification is kept at 4x in one direction, but is increased to 8x in the perpendicular direction. The enhancement to SCALE outlined in this article will allow the application of the asymmetric scaling required by EUV lithography.



Mask Metrology Support for KLA-Tencor LMS IPRO6 Tool

Hari Konnanur, CATS CAE, and Herman Boerland, CATS R&D

Background

The KLA-Tencor LMS IPRO6 mask registration tool is designed to accurately verify pattern placement performance of reticles at the 1X nm node. As the technology nodes shrink, OPC becomes increasingly complex and requires densely sampled registration data to qualify mask results. With this complexity, the previous limited number of measurement points are not sufficient to characterize masks. Moreover, under-sampling also results in unrecognized pattern exclusions that reduce mask coverage.

Motivation

The LMS IPRO6 tool provides a methodology to support fast and accurate model-based image processing algorithms for on-device pattern-based metrology to detect and characterize pattern-dependent placement errors.

As described in the following sections, this model-based algorithm delivers the accuracy needed to validate registration errors. This article discusses the off-tool process and use model for creating all necessary data for the tool using CATS®.

CATS Metrology Use Model

CATS® can be used to create recipe files for several Metrology tools. Recently, support for the KLA-Tencor LMS IPRO6 tool was added as well. The process for creating those recipe files is similar for all tools (including LMS IPRO6):

- ▶ Input a JOBDECK.
- ▶ Setup the global settings for the metrology tool.
- ▶ Add Alignment marks.
- ▶ Add all marks.
- ▶ Write a CATS Jobdeck, which contains all the information required to create the files for the tool.
- ▶ Use WRITEMARK to create the recipe/measurement file.

This process can be done either manually (GUI-flow) or automated (using scripts).

GUI Flow

All metrology functionality can be found in the Tree View, Property Browser, and Fracture View (via the right-mouse button).

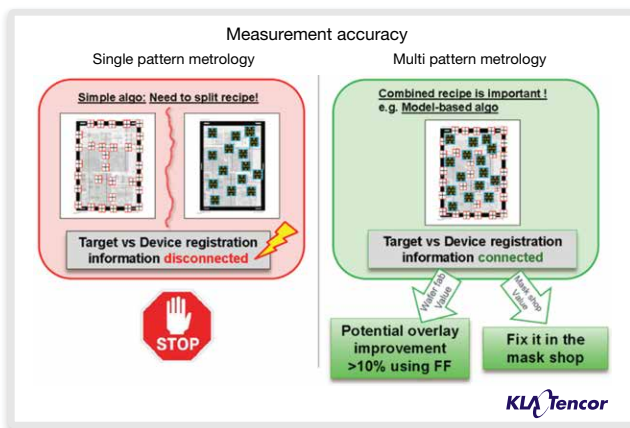


Figure 1. Measurement Accuracy in Pattern Metrology

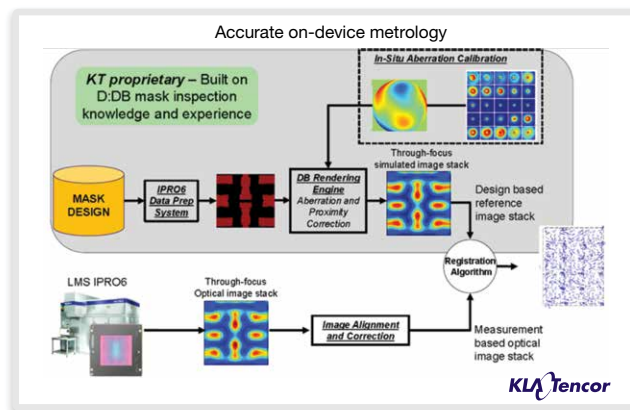


Figure 2. On-Device Metrology

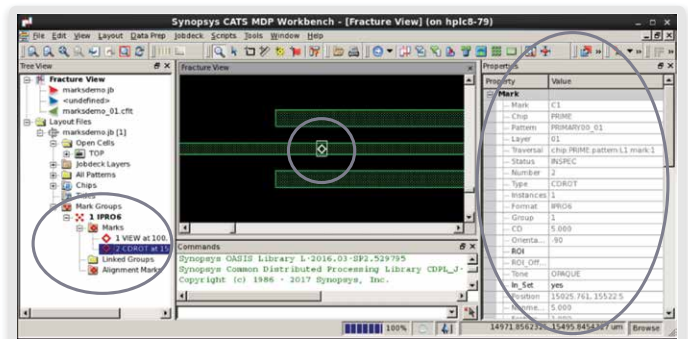


Figure 3. LMS IPRO6 and the CATS Graphical Interface



Automated Flow

When all measurement locations are known upfront, a point-list can be created. Using the CATS scripting language (CSL), this list can be read and marks can be automatically generated.

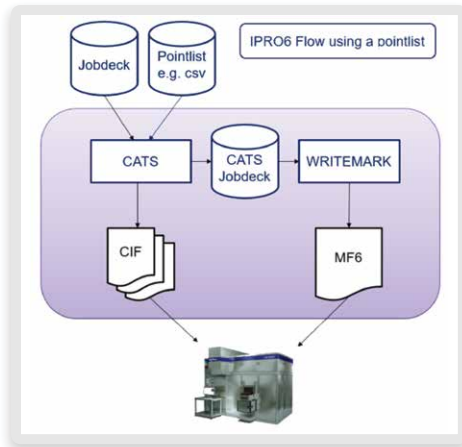


Figure 4. LMS IPRO6 Flow in CATS®

LMS IPRO6 and Marks for the Model-Based Algorithm

As described in the first section, a new algorithm has been implemented for the LMS IPRO6 tool. This model-based algorithm uses a polygon description of the design data instead of a description of the region-of-interest (ROI) that needs to be measured.

To create such measurement, a mark of marktype VIEW must be created within CATS®:

```
JD ADD MARK DOPICK VIEW <x>,<y>
```

CATS® will accept the location without investigating the data as is needed for other marktypes such as CD, BOX, or CROSS. Instead, a snapshot of the data is used to create a polygon description file (CIF format). The size of this snapshot will by default be 4x4 um plus an oversize area of 2um. If another size is required, this can be accomplished using **JD ADD MARK DOPICK VIEW <x1>,<y1> <x2>,<y2>**, or after the mark is created using **JD MODIFY MARK PICK ROI=<width>,<height>**

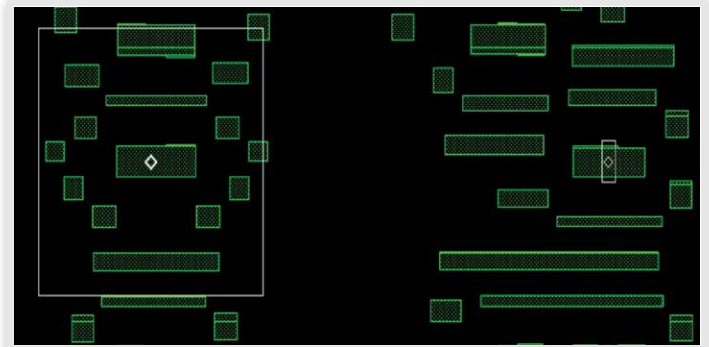


Figure 5. Model-Based Measurement Versus CD Measurement

Original LMS IPRO6 Algorithms

The original IPRO algorithms (measuring a region-of-interest) are still available for LMS IPRO6 and can still be created with CATS® using the generic marktypes: CD, BAR, L, CROSS, or BOX, or the rotated versions of these marks: CDRot, BarRot, or BoxRot.

LMS IPRO6 Alignment Marks

The setup of the alignment marks for the LMS IPRO6 is slightly different from the older IPRO tools (IPRO2 to IPRO4). For those tools, there were four special marktypes (AL, AH, BL, and BH), which needed to be created.

For LMS IPRO6, the alignment marks are built on top of the 2D marks: L, Cross, or Box. After two of those marks are created, the mark property ALIGNMENT can be set to either A or B.

The other alignment parameters must be set with the JD IPRO6 parameters PATTERN_AL, PATTERN_AM, PATTERN_BL, and PATTERN_BM.

Additional LMS IPRO6 Setup Parameters

To fully support the capabilities of the MF6 Measurement file for LMS IPRO6, several parameters have been added to the JD IPRO6 command, such as SUBSTRATE, PELLICLE, and PERFMODE.



LMS IPRO6 Output Files

When all steps are complete, several files are created, which must be sent to the LMS IPRO6 tool to start the measuring.

► MF6 Measurement file

```
MFVERSION 6.3
* WRITEMARK CATS K-2015.09-SP11-PRE-170116 529762
* CATS-produced IPRO6 Measure File, Jan 17 11:27:29 2017
* Input file ./Test_01.cjb
* Pattern Test_0101, Layer 1
OUTPUT 0
ORIGOFF 0.000,0.000,OFF
CDCAL CD CALIB OFF,OFF
TOOLINFO 1, Project information could go here

* Main Level
RETICLE
SITE TXY,0.0,0.0,CTR=0,0,DIM=4,4,FA=ON,FF=ON,FM=ON,FSA=OFF,FGA=OFF,FSIM=OFF,NI=80
$ TVI=50,FCE=OFF,ALG=1,FSL=OFF,TH=50,LI=20,HI=80,SLP=1,FLS=OFF,ZPC=mn,HPC=mn

SITE AXY,86950,47900.5
  FIELD 1X,DIM=14,0.98,CD=12,NE=2
  FIELD 2Y,DIM=3,1,CD=1,NE=2
SITE BXY,125542,48140.5
  FIELD 1X,DIM=14,0.98,CD=12,NE=2
  FIELD 2Y,DIM=3,1,CD=1,NE=2

SITE 1XY,86638.152,72818.131
  FIELD 1X,ALG=3,PTYP=Test_01_001
  FIELD 2Y,ALG=3,PTYP=Test_01_001
END
```

Figure 6. Output MF6 Measurement File



- ▶ For each model-based algorithm measurement, a CIF file is created, containing the polygon information.

```
(CIF file Test_01_001)
(written by CATS K-2015.09-SP11-PRE-170116 529762)
(written at Tue Jan 17 11:25:42 2017)
DS 1 1 100;
  9 001;
  L 0_0;
  B 160000 140000 866381520 728181312;
  L 1_0;
  P 866301522 728111320 866301534 728111320 866301534 728111319 866301522 728111319;
DF;
C 1;
E;
```

Figure 7. CIF File Example

Conclusion

For CATS users who have the LMS IPRO6 tool and want to take advantage of the model-based algorithm, a sample use model has been described and is now in production at several mask shops. For more detailed information, please contact your CATS customer support representative.

