



# Lab Manual

[Marvell NanoLab](#)[Member login](#)[Lab Manual Contents](#)[MercuryWeb](#)[Berkeley Microlab](#)

## Chapter 3.2

# **GCAWS2 Wafer Alignment Key Design Guide**

(gcaws2 - 384)

### 1.0 **Title**

GCAWS2 Wafer Alignment Key Design Guide

### 2.0 **Purpose**

GCA Steppers employs a unique **off-axis** wafer alignment system that separately aligns a mask to the optical column and a wafer to the optical column. Alignment of the wafer to the optical column is referred to as wafer alignment. This is done manually by utilizing wafer alignment keys printed and etched into the wafers at previous layers. Alignment key specification is outlined in this manual.

### 3.0 **Scope**

This document describes the design guidelines for wafer alignment keys for the GCA 4800/6200 DSW Wafer Stepper System. It details requirements for dimensions and placement of manual wafer alignment keys.

### 4.0 **Applicable Documents**

#### [Revision History](#)

The following documents are published by GCA and are kept in the NanoLab office.

**4800 DSW Wafer Stepper System Instrument Instruction Manual** (Document Part No. 011819, Rev. 1 Printed 12/15/82). This manual details the hardware.

**DSW WAFER STEPPER SYSTEM - System Control** (Document Part No. 011817, Rev. 4C Printed 10/25/82). This manual is the software manual that details how to specify, edit, and run a job.

See also [Chapter 3.1](#) - Mask Generation Using CAD Software.

### 5.0 **Definitions & Process Terminology**

N/A

### 6.0 **Safety**

N/A

### 7.0 **Statistical & Process Data**

N/A

### 8.0 **Process Notes**

#### 8.1 **Readily Available Wafer Alignment Keys (gds Format)**

You can find a .gds copy of both **dark field** and **clear field** version of the alignment marks in the .gds format on silicon.

At silicon prompt, simply type: `cd /mercury4/cad/gcaws2`

Following files are available for lab members to download:

clear\_field\_gcaws2\_key.gds

dark\_field\_gcaws2\_key.gds

These marks can be incorporated into your chip layout design by importing them first into Cadence or L-Edit or similar CAD software that you are using for your design, as long as .gds format is accepted. One can also generate these keys in KIC layout design tool, as per instruction provided in the Appendix section of this manual.

## 9.0 Procedure for Generating the Alignment Key and Proper Stepping Distance for GCAWS2

The standard wafer alignment keys are used by an operator to align the wafer manually.

[Figure1](#) shows schematics of a clear field (light field) alignment key at the wafer level.

[Figure2](#) shows schematics of a dark field alignment key at the wafer level.

### 9.1 Alignment Key Placement Guideline

The GCA Stepper at Berkeley is installed with a 100-millimeter wafer chuck and objective lenses spaced 76.2 millimeter (3.0 inches) apart. Wafer alignment keys must be forward of the wafer center in the Y direction by 3.2 millimeters (0.125 inch). You can consult page 6-3 of the **4800 DSW Wafer Stepper System Instrument Instruction Manual**. The shaded region of the illustration shows the allowable region for placing wafer alignment keys.

Wafer alignment requires two alignment keys: a right-hand wafer alignment key (X and Y, or primary, alignment key), and a left-hand wafer alignment key (theta, or secondary, alignment key). There are two ways to place the keys so that they are physically separated by 76.2 millimeters (3 inches), the objective separation, on the wafer.

#### 9.1.1 Placement Method 1

This method is preferred when you don't want to worry about the step size, and it is the simpler of the two. This first method involves using a reticle that has only one wafer alignment key and specifying a step size in X, so that there will be two keys spaced exactly 76.2 millimeters in your wafer layout. A typical example of a wafer layout is shown below. The die size is defined as the size of dies on a wafer, and step size in X is defined, as the horizontal distance between centers of successive exposures in the array. Step size in X determines the stepping distance between the adjacent columns.

When specifying a job on the GCA Stepper, you should indicate that standard keys are used and enter an appropriate step size in X, as per table1 below. Your step size should be greater than the die size in order to generate scribe lanes or streets between your dies. This means you need to size your dies appropriately. This extra area between your printed dies will help you see the die border, and later can be used for wafer dicing at the end of your process (diamond saw will run through these streets not impacting your product die). A typical scribe lane size is in the order of 100  $\mu\text{m}$  – 200  $\mu\text{m}$  at wafer level.

Column Number	Step Size in X (mm)
<b>n</b>	<b>76.2 / (n - 1)</b>
14	5.86154
13	6.35000
12	6.92727
11	7.62000
10	8.46667
9	9.52500
8	10.88571
7	12.70000
6	15.24000
5	19.05000

**Table1 - Die Column and Step Size Chart**

9.1.1.1 Example

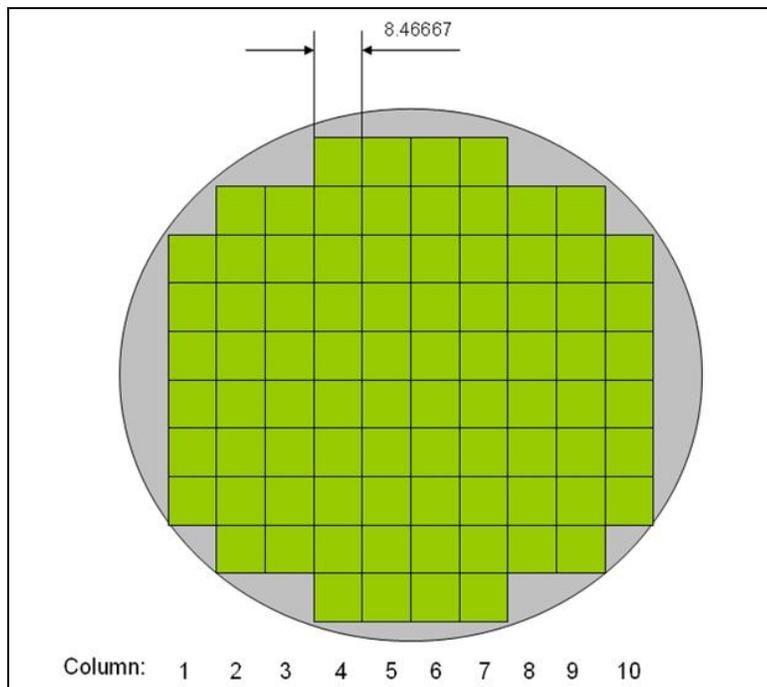
This example has 10 columns and step size of 8.46667 millimeters.

Each die will repeat the alignment mark, once on the wafer. Spacing between the first alignment key to n<sup>th</sup> alignment key (n - 1 die over) for this particular die size/stepper job, and at the wafer level can be calculated by the following formula:

$$\text{Spacing} = (n - 1) \times 8.4667 \text{ mm}$$

$$\text{Spacing between the first alignment key and the 9<sup>th</sup> die} = 9 \times 8.46667 = 76.2 \text{ mm.}$$

This will provide proper mark spacing for the two microscopes objectives at a distance of exactly 76.2 mm apart for GCAWS2 stepper.



9.1.2 Placement Method 2

The second method involves using a reticle that has two wafer alignment keys, and specifying a step size in X, so that two keys will become the left- and right-hand wafer alignment keys. This method essentially provides an extra key in the die to correct for proper alignment marks spacing between the first and  $N_{th}$  die, needed to be spaced specifically 76.2 mm apart at the wafer layout. This method is used when the step size in X and the die size are predetermined. The offset used for the placement of the extra key in the dies will make the secondary key show up at correct spacing under the objective lenses of the stepper Microscopes.

## 9.1.2.1 Algorithm

Follow the algorithm below to calculate the separation of the two wafer alignment keys on the reticle, and the number of columns.

Variables Definitions:

Input: OS = objective spacing = 76.2 (mm)  
 DS = die size (mm) = user's input  
 SSX = step size in X (mm) = user's input

Output: WAKRS = The separation of the wafer alignment keys on the reticle, to be used when designing the reticle.  
 NCOL = The number of columns of dies, to be entered when specifying the job on the GCA Stepper.

```

Algorithm: # this is a line of comments.
           # I assume the following functions:
           # intdiv(a, b) = integer divide
           #           = integer(a / b)
           # mod(a, b) = the remaining function
           #           = a - intdiv(a, b) * b
if mod(OS, SSX) < DS then
           # The right-hand key on the reticle is the left-hand wafer.
           # alignment key, and the left-hand key on the reticle is
           # the right-hand wafer alignment key.
           # Key offset is associated with the left-hand key on the
           # reticle.
           NCOL = intdiv(OS, SSX) + 1
           WAKRS = mod(OS, SSX)
else
           # The right-hand key on the reticle is the right-hand wafer
           # alignment key, and the left-hand key on the reticle is
           # the left-hand wafer alignment key.
           # Key offset is associated with the right-hand key on the
           # reticle.
           NCOL = intdiv(OS, SSX) + 2
           WAKRS = SSX - mod(OS, SSX)

```

## 9.1.2.2 Example

Assuming die size (DS) equals to 10.0 millimeters and objective spacing (OS) equals to 76.2 millimeters. Following table was compiled:

SSX (mm)	9	10	11	12
WAKRS (mm)	4.2	6.2	0.8	4.2
NCOL	9	8	8	7

Which clause was followed:

then then else then

Which key on the reticle is chosen as the right-hand wafer alignment key (the X and Y, or primary, alignment key):

L L R L

An example of the left-hand key on the reticle is chosen as the right-hand wafer alignment key.

SSX = 12

Column	1	2	3	4	5	6	7
Reticle	L R	L R	L R	L R	L R	L R	L R
Wafer	R L	R L	R L	R L	R L	R L	R L
Alignment Key spacing	0	12	24	36	48	60	72 76.2

An example of the right-hand key on the reticle is chosen as the right-hand wafer alignment key.

SSX = 11

Column	1	2	3	4	5	6	7
Reticle	L R	L R	L R	L R	L R	L R	L R
Wafer	R L	R L	R L	R L	R L	R L	R L
Alignment Key spacing	0 11	22	33	44	55	66	76.2

Where the first row of numbers represents the columns, R represents the right-hand key on the reticle and L represents the left-hand key on the reticle, and the last row of numbers represents the spacing between wafer alignment keys. Note that the image on the wafer is an inverted image of the reticle, and this inversion applies to both X and Y.

9.1.3 Alignment Key Requirement

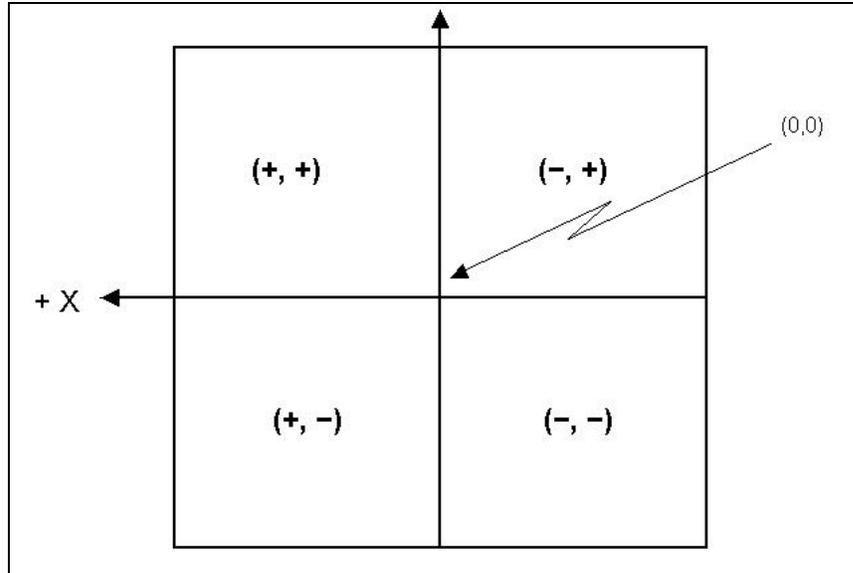
Under ideal conditions the alignment keys on the wafer will appear as having two-micron wide, solid black lines on a white background. Alignment keys having a black outline with a white center and a grey background, as viewed on the TV monitor, should be avoided. However, black-outlined keys with a grey center and grey background are generally acceptable as long as the line width of the black edges are equal.

Reticles should contain as many wafer alignment keys, in both polarities, as possible to allow maximum flexibility. However, an area 500 μm by 500 μm surrounding a wafer alignment key must be clear of any 45° or 135° lines. This applies to device geometry or other wafer alignment keys.

Examples of kic files of standard GCA alignment keys of both polarities can be found in argon. At argon node simply type: cd /mercury4/cad/examples/Krueger2.nwell. The kic instance is called GCA, and this and associated kic files can be copied to your directory for inclusion in your layout. Note that the layout is done in 0.1 microns/lambda.

#### 9.1.4 Alignment Key Requirement

When specifying a job on the GCAWS2, you have to enter the location of the right-hand (primary or X and Y) wafer alignment key with respect to the reticle center. Alignment key offset, also known as key offset, is defined as the distance in X and Y of the wafer alignment key from the center of the right-hand alignment die. If you are using method one to place wafer alignment keys, give the key offset of your single wafer alignment key. If you are using method two, you must give the key offset of the alignment key that is located on the right hand side of a wafer. The alignment key offset is in millimeters and it is measured on the wafer, that is, it is measured after the 10 to 1 reduction from the reticle to the wafer. The sign convention is illustrated below.



If standard keys are used, that is, the manual keys on the wafers are spaced exactly 76.2 millimeters apart, then manual alignment can be easily performed. However, if the manual alignment keys are not standard, you can specify that nonstandard keys are used and enter the right- and left-hand alignment die and alignment key offsets.

## 10.0 Troubleshooting Guidelines

N/A

11.0 Figures & Schematics

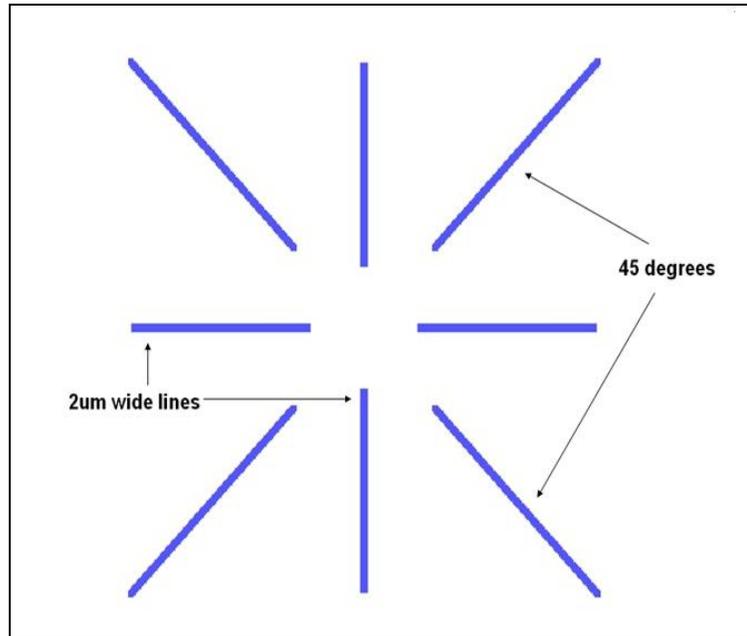


Figure 1

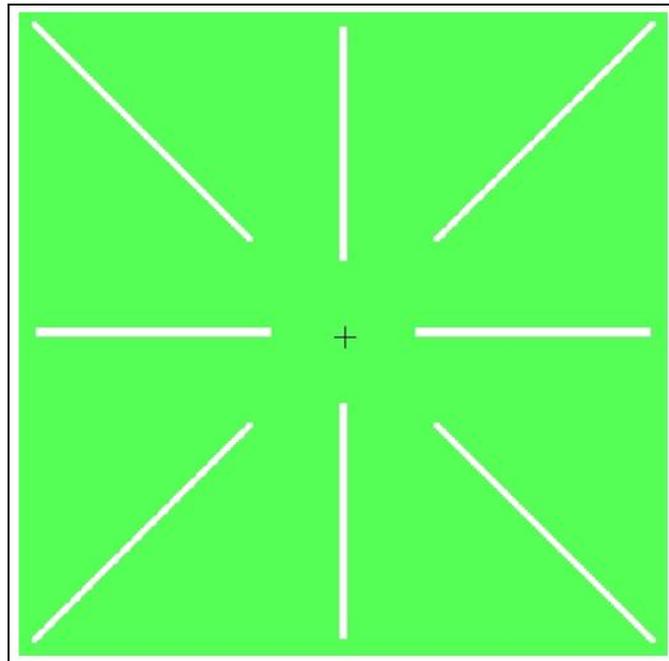


Figure 2

## 12.0 Appendix

### ***KIC CAD Layout Tool Information***

#### **12.1 How to Generate Alignment key using the KIC cad layout tool**

The following information may or may not be relevant, depending on whether or not you can find the source code of the program referred to in this section.

You can use `wakey` to automatically generate a wafer alignment key layout in KIC data format. The executable file is `~cad/bin/wakey` on argon.

The `wakey` program executes in a self-prompting mode and produces three files: `name.w.kic`, `name.qw.kic`, and `name.w.doc`, where `name` is the file name that you specify. `name.w.kic` has the whole wafer alignment, `name.qw.kic` has the quarter wafer alignment key layout, and `name.w.doc` documents the dimensions and parameters.

The wafer alignment key is centered about the origin, and you can use the KIC **instance** command to place a wafer alignment key anywhere on your mask layouts.

The alignment key generated is on layer NI, you can change the layer by either using the KIC **change-layer** command to change the layer of the `name.qw.kic` file or using `vi` to edit you `kic` files.

Example: Your `name.qw.kic` file should have a line like this:

```
L NI;
```

The above line tells KIC that the layout is on layer NI.

You can change NI to the layer of your choice, such as:

```
L ND;
```

The above line tells KIC that the layout is on layer ND.

Alternatively, there are examples of layout keys that can be used as KIC instances. These `kic` files are located on argon in `/cad/examples/krueger2.nwell`. File names include: CW, CD, CP, CC, CM, CV, CM2, GCA.

#### **12.2 How to Generate Negative-Polarity Keys**

The above procedure will produce a positive-polarity alignment key, that is, a dark key on light background. This section describes how to make a negative-polarity alignment key, that is, a light key on dark background.

12.2.1 Generate a positive-polarity key using the above procedure.

12.2.2 Use `kictocif` to transform the **xx.kic** file to **xx.cif** file. Use `ciftomann` to transform the **xx.cif** file to **yy.kic** file, with commands to invert and scale up.

The **yy.kic** file has the KIC layout of a negative-polarity key that has the exact dimensions of the original positive-polarity key.

**xx** and **yy** are file names.

Example: Generate a positive-polarity key in file **test.w.kic**, using 1 lambda equaling 0.2 microns.

Type **kictocif -L 0.20 -I test.w.kic -a**

where

-L 0.20 = one lambda equals to 0.20 microns

-I test.w.kic = the name of the input **.kic** file

-a = translate all layers

The above command produces file called **test.cif**.

Type **ciftomann -y -l test.log -p P -f invert.com \*\* continued from previous line \*\* -o ltest.kic -i test.cif**

where

- y = answer yes to continue with execution
- l test.log = the name of the log file that contains possible error messages
- p PG3600 = the pattern generator is GCA PG3600
- f invert.com = the name of the command file that has the following two lines:  
scale 50 NI invert

where

- scale 50 = scale the whole mask by 50 = all coordinates \* 50
- NI invert = invert the layer NI
- o ltest.kic = the name of the output **.kic** file -i test.cif = the name of the input **.cif** file

The file **ltest.kic** has the negative-polarity key layout.

Alternatively, you can use the negative polarity key kic files located on argon in /cad/examples. File names include: CWi, CDi, CPi, CCi, CMi, CVi, CM2i, GCA.

### 12.3 Low Flash Count Alignment Keys for GCAWS2 Wafers Stepper (KIC files)

Two new kic layouts of a wafer alignment key, one positive and one negative, have been generated. These new layouts feature extremely low flash counts -- 12 for the positive version and 21 for the negative (or inverse) version. Some currently utilized keys have over one thousand flashes. A script to label, offset and resize these keys is also available. By using these keys in your new layouts you will:

- 12.3.1 Get much faster display redraws of expanded keys.
- 12.3.2 Realize a significant time savings on mask generation jobs.
- 12.3.3 Conform to the GCA Specification provided in drawing number A-006023 rev 2.

#### Warnings

- 12.3.4 These layouts were done with lambda = 0.1 micron.
- 12.3.5 These have larger area surrounding the keys so you may have trouble dropping them into old designs. This is to conform to note 3 which states Area within  $\pm 250$  microns of mark must be clear of 45 degree marks. These boxes allow that much space when placed next to each other on the wafer.
- 12.3.6 The layer name must be edited using a text editor before you can instantiate these into your kic layout. The fourth line of the file: L NI; must be changed to reflect some layer name that is in your .KICLayers file.
- 12.3.7 There are no text labels on these two layouts.

#### Details

The files can be found in /cad/examples/Wafer-Alignment- Keys-for-gcaws.

- gca\_wa\_key.kic (positive version)
- gca\_wa\_key\_inv.kic (negative version)
- gca\_wa\_keys.kic (used by make\_gca\_key, contains four layers)
- make\_gca\_key (a script to make a labeled, offset, key on a specific layer. This is a modification of a Richard Moroney script. This can resize the layout using setenv KIC\_MPL. This will add 100+ flashes for labels.)