Agilent Technologies

Considerations for Maximizing Throughput on the Agilent 5DX

This information pertains to:

Software Revision: 5.x
Hardware: 5DX
ID #: 99-12-021-04

Throughput Considerations

Factors that affect throughput for a panel on the 5DX (listed in approximate order of importance).

1. Setup of the FOV file (FOV_DATA.NDF). Generally the use of the largest FOV and the smallest snap margins possible will result in the fastest test time.
2. Setup of the algslice file (ALGSLICE.NDF). Using slices other than pad level will increase test time. Certain algorithms require a lot of computation and slow down test times.
3. Issues related to multi-board panels.
4. Setup of the camera files (CAMERA.NDF, one per board side for a double-sided circuit board). To maximize throughput, reduce the number of different camera settings in the same area of the board. In areas of significant shading, use the same camera index for all shaded components. Do not use the noise reduction files.
5. Stage speed.
6. Alignment.
7. Surface map setup.
8. Image Window. Minimizing the Image Window can increase throughput.

1. Setup of the FOV File

The most important factors in determining program test time are:

- the FOV and image format (512 or 1024), that will be used to inspect each component, and
- the snap margin, how much area around the edge of the image will not contain tested joints.

Test time is controlled almost exclusively by image acquisition time plus the time necessary to move the board to the next view (move time) or the camera readout time, whichever is greater. For any given application, move time will be the limiting factor for some views, and camera readout will limit the speed for other views. Camera readout is faster for 512 images than 1024 images, so if the same number of views is used in each mode, the 512 based test will be faster than a 1024 based test. A panel test using 1024 mode will generally be considerably faster than a test using 512 mode because there will be fewer views (larger FOVs can be used).

Note: Minimizing the number of images used to test the board will maximize throughput.

FOVs

There are no definite rules about what FOV is needed to test what component type and pitch successfully. What works well for one application may not work on another. Joints with more solder are easier to test than those with very little. Other factors that need to be considered include the trade-off between false alarms and false escapes, as well as throughput. Recommended minimum FOVs for testing various components as a function of pitch in 1024 and 512 modes are shown below.

Pitch	Minimum FOV (1024 image)	Minimum FOV (512 image)
10	260	----
15	400	260
20	650	320
25	800	400
30	800	650
50	800	650
50	800	800

The default FOV file supplied with the system contains settings that correspond to these recommendations. Some joint types do not conform to these rules well. BGAs, for example, would be tested at 800 in 1024 mode and either 650 or 800 in 512 mode. It is recommended, however, that BGAs be tested at either 400 or 650 in 1024 mode, and either 320 or 400 in 512 mode.

NOTE: The use of the recommended minimum FOV for each component will not necessarily result in the shortest test time. This is primarily dependent on the mix of components on the board.

If many components tested at a larger FOV actually appear in the images taken at a smaller FOV, it may be faster to remove the larger FOV and test all at the smaller FOV. The effect of this can be estimated by compiling the program using different settings in the FOV file, and comparing the number of views in the compilation report.

Snap Margin

The snap margin specifies the area around the edges of the image where there can be no tested joints. There are two ways the snap margin can be applied. The larger of the listed snap margin or inter-pad distance (IPD) can be used to establish how close to the edge of the image a tested pad may be placed. This is the default technique. An attribute line in the FOV file would like this:

	Pitch	FOV	Snap Margin	Pixel Resolution
FOV	12	400	20	1024

Alternately, the snap margin can be added to the IPD to determine how close a tested pad may be to the edge of the image. To do this, an additional field (ADD) denoting this is added to the end of the attribute line. In the latter case, a smaller snap margin can be used. This is the recommended choice. An attribute line in the FOV file would look like this:

	Pitch	FOV	Snap Margin	Pixel Resolution

FOV	12	400	10	1024	ADD

NOTE: If the ADD choice is used, an image resolution must be specified. If it is not, a warning will appear in CADTRAN.LOG, and the ADD will be ignored.

If small snap margins are used, such as those supplied with the default FOV file, warning messages may be generated during test. These need to be looked at and addressed, as they may indicate problems that will adversely affect algorithm performance. These can almost always be dealt with without increasing snap margin.

Warnings from the Locator algorithm, "Error in Template Locator call of region Lock, CAD position used.", mean that the locator search region has extended outside of the area of the image. The data is unreliable, therefore the CAD position for the joint is being used. Essentially, the Locator algorithm is not being run on these joints. These warnings can and should be eliminated by limiting the search region, using the Locator thresholds SEARCH_DIST or SEARCH_DIST_ACROSS. The values of these thresholds can be reduced from the default of 1.0 to 0.5 or less without fear of failing to search far enough to locate the joint. The pitch of a component also affects search distances, so another way to deal with this problem may be to reduce the pitch.

Warnings from the SPC algorithm will vary from joint type to joint type, depending on what measurements are made outside of the pad area. A way to eliminate the warnings can generally be found by displaying the SPC graphics for the joint type and testing the view in question. Look for the measurement regions that extend to the edge of the image.

Either the Short or Solderball algorithms may also generate warnings. Reducing the IPD_REDUC threshold value can eliminate these. This should be done before doing joint based learning for shorts.

Board Warpage

Another factor that needs to be considered is board warpage. If a particular panel type typically displays significant warpage, some or all of the views may contain joints that are far enough from the plane of focus that they cannot be tested reliably. Joints in the center of the image may be at the correct Z-height to give a good image, but the warpage may be severe enough that joints near the edges are not properly focused. Some joint types (FPGullwing and BGA, for example) have smaller tolerances for focal plane variations than other joint types. Adding more surface map points will not help in this case, since the correct Z-height was found for the center of the image.

There are only a few things that can be done to alleviate this situation.

- The snap margin can be increased, but normally it would need to be increased very dramatically and this is not an attractive solution. With a larger snap margin, only a smaller area of the image is used to test joints. The panel program will generally have many more views if this is done, dramatically decreasing throughput.
- A smaller FOV can be used to inspect components in those areas where warpage is typically very severe, however FOV is set on a package basis, so other components of the same package type will be tested at the smaller FOV as well.

Examples

The total number of views used to test the board, as reported in the compilation report, is a good indication of the test time. Reducing the number of views will generally decrease the test time. The division of views/FOV is also provided, and can be useful information. If there are only a few views at a given FOV, switching the components tested at that FOV to an even smaller FOV that is already being used to test other components, may result in a net decrease in the number of views and the test time. Even if the number of views increases slightly, the test time may be reduced, since the average move time may be reduced significantly. (The system will make a complete pass of the board for each FOV used in a panel program, testing all views at the smallest FOV, then moving to next, and finally testing at the largest FOV.)

Here is an example using a panel program in 512 mode. The original compilation report listed:

FOV	# Views	# Slices	# Joints
320	228	264	756
400	11	11	26
650	3	3	6
TOTAL	242	278	788

Since there are only a few joints tested at 650, it may be faster to test these at 400 instead. Other components tested at 400 may fit in the same views. By commenting out the 650 FOV line in the FOV file and recompiling, components originally scheduled to be tested at 650 will be tested at 400. The new compilation report shows:

FOV	# Views	# Slices	# Joints
320	228	264	756
400	14	14	32
TOTAL	242	278	788

The total number of views did not decrease, but the average move times almost certainly did, since the system did not have to go back to the same areas tested at 400 and test again at 650. The test in this form will run slightly faster than with the 6 joints tested at 650. This can be taken a step farther by inspecting everything at the 320 FOV. The new compilation report shows:

FOV	# Views	# Slices	# Joints
320	242	278	788

Again, the panel program requires the same number of views, but this test should be somewhat faster due to decrease in the average move time.

The test time for this panel could be greatly reduced by using the 1024 image mode and increasing the FOVs used to test each pitch. The compilation report shows:

FOV	# Views	# Slices	# Joints
650	105	141	756
800	12	12	32
TOTAL	117	153	788

Testing everything at the 650 FOV reduces the number of views and test time further:

FOV	# Views	# Slices	# Joints
650	107	143	788

2. Algslice File

Each additional slice used to test a component requires an added move and image acquisition. Using slices other than that used for the majority of the board for certain component types (for example SOTs or 2 pin diodes) also increases the total number of slices for the panel program, because other components in the same view will use the pad level. This will increase test time.

Include only those algorithms that are useful or required. If thresholds for these algorithms are always opened up so that no defects are called, eliminate that algorithm from the algslice file so that it is not even run. An example might be FPGullwing Shape. If subtype learning is not done for shape and the subview ringer threshold, MAX_RINGER_CHI_SQR, is always increased so that no defects are called, eliminate FPGullwing_Shape from the algslice file.

Some algorithms (notably Solderball and Voiding) are computationally intensive and can affect throughput. The higher the sensitivity these algorithms are run at, the slower they will be. For example, decreasing the IN_BALL_THICKNESS threshold in Solderball will increase the number of potential solderball candidates that are found. The algorithm will have to test each one to see if it meets the other criteria to be a solderball. This process can slow down the test time.

In order to determine if analysis is a significant factor in test time, turn all algorithms off and test the panel. Compare this time to the normal test time. If there is a large difference, you may want to turn algorithms off sequentially to determine which ones are primarily responsible for the increased test times.

If there is a desire to test some components at a slice other than that used for the majority of the board, forcing those components into separate views can speed up the test.

For instance, it has been reported that better measurement separation between open and good joints on BGAs can be obtained by taking slices somewhat above the center of the ball and somewhat below the pad level (instead of at the center of the ball and at pad level). The slice order line might read:

```
.SLICE_ORDER: BGA2 ALL TOP+9 (TOP-3)
```

If no changes are made to force the BGAs to separate views, a compilation report for one application reads:

FOV	# Views	# Slices	# Joints
650	153	185	6444

By forcing the BGAs to be in views by themselves (changing the Camera Index to one not used by surrounding components, see section on Camera Files below), the total number of images (slices) is reduced.

FOV	# Views	# Slices	# Joints

650	158	164	6444

The total number of views increases by 5, but number of slices decreases by 21, and the test runs about 10% faster.

3. Issues Related to Multi-Board Panels

The fastest way to test a multi-board panel is generally to define it as a single board. There are several drawbacks to this approach, however.

- It is necessary to create new board and component files with proper component coordinates and unique reference designators. While component coordinates can be generated in a spreadsheet program using the original single board file and the known offsets to the other boards, this can be a tedious operation.
- There is no option to skip an unpopulated board if the whole panel is defined as a single board. If an unpopulated board is tested, many defects are reported for a single condition, which complicates the repair process, defect a analysis, and so forth.
- If the panel is highly routed, there may be more difficulty with obtaining a good surface map, since surface map triangles will cross board boundaries, where there can be a Z-height discontinuity.

Another option is to use panel based alignment for multi-board panels, and the optimal selection of board order to minimize move times. This may still be considerably slower than defining the entire panel as one board.

4. Camera Files

Using different camera settings for different pins on a component or for nearby components can lead to a significant increase in the number of views and the test time. In general, joints can be tested reliably with a wide range of background levels. Using the Generate Camera Settings option under Panel Programming will usually produce many more different camera settings and views than necessary for a good test. It is recommended that this option not be used. If it is used, the camera files should be edited to eliminate extra views caused by a difference of only one in the shading level. Adjusting camera settings for shaded components can be as much for the benefit of the program developer as for the accuracy of the measurement.

Noise in the image can effect test reliability, particularly for joints that have relatively little solder or have significant shading. The signal-to-noise ratio in the image can be improved by making changes in the camera files. The default operation of the system is to take each image while the X-ray source and Rotary Scintillator move through one complete cycle. If it is desired, images can be taken during 2, 3 or 4 cycles. The image is integrated in the camera on the CCD chip. Since more light is collected, noise in the image due to the low light level is reduced, and the test results are more repeatable.

To test whether 2-cycle integration is needed, first set up the application using 2-cycle integration, and proceed until false alarms are at an acceptable level, then change to 1-cycle images, and see if the false alarm rate increases noticeably. Analysis of joints with heavy shading will also be improved with multiple cycle integration. Multiple cycle integration is specified in the camera file as a multiplier to the shading level. Data lines in the camera file might look like this:

```
@U6 1x2
```

where the 1 indicates the general shading level (in this case minimal) and 2 is number of cycles.

NOTE: Components tested at the same FOV and the same general shading conditions will be placed in separate views, if a different number of cycles is specified. It may actually increase throughput to use multiple cycles for additional components, if they would otherwise appear in a view that contains components that require multiple cycles.

To test this, first compile the panel program using only single cycle integration. Record the number of views at each FOV. Now add the appropriate lines to the camera files for multiple integration, and recompile. If the number of views increases significantly, comment out the multiple integration lines and recompile again. Use CheckCAD to determine the other components that are being tested in the same views as the components you wish to test with multiple integrations. Remove the comments from the camera files, add data lines for the added components identified in CheckCAD, and recompile.

For example, before adding multiple cycles, U21, a TSOP, was tested in 2 views along with several small resistors, R56, R101, R123, R60, and R63. The total number of views increased by 1 after the test is altered to use two cycles for U21. If the resistors above are also tested at 2 cycles, the number of views will be decreased, and the test will run faster.

NOTE: The option to reduce noise in the image using the Frame Average (NOISERED.NDF) files still exists for Series 2. It should not be used. Use the multiple integration feature of the camera file instead.

Frame averaging takes two or more discreet images and averages them, rather than integrating the image in the camera. It should not be used because it is much slower and produces results that are not as good.

5. Stage Speed

The 5DX has six motion profiles defined, referred to as stage speeds. The motion profiles control not only the top speed of the stage for each axis, but also accelerations, decelerations and other parameters that control how smoothly the stage moves. The motion profiles do not necessarily decrease the average move times in an ascending order. That is, for some programs speed 4 may be slower than speed 3, and so forth. To test this, increase the stage speed incrementally, note the test time and the results. When false calls start to increase, revert to the stage speed corresponding to the shortest test time that still gives minimal false calls due to board bounce.

Beginning at release 6.1, it is possible to configure the stage speed for surface mapping so that it runs at a speed different than the test. Since the fast, pipelined mode for surface does not involve moves in the Z-axis, it may be possible to surface map at a higher speed than used for the test, and hence decrease total beat rate. There is a new parameter in LASER.CFG:

Laser_MotionProfile

Valid values for this parameter are 0-6, with 1-6 representing the usual stage speeds. If the parameter is set to 0, the 5DX will use whatever stage speed is set elsewhere, either in the PANEL.NDF for that panel, or if nothing is set there, it will use the speed set in the User Interface. 0 is the default setting.

6. Alignment

The choice of alignment points that do not require surface mapping can save considerable time per panel.

If surface mapping of the alignment points is necessary, at least occasionally, there are two ways this can be accomplished. A data line can be placed in the panel file so that surface mapping of the alignment points is done before alignment is attempted. Alternately the system can be configured to surface map the alignment points if alignment fails. The choice of which option to use may depend on how frequently alignment fails without surface mapping. The first option applies only to those individual panel programs for which it is set, the second option will affect all panels tested.

Generally, alignment points located near the edge of the panel, held by the panel clamps, will not require surface mapping. On multi-board panels, where there are several boards across the 5DX in the X-axis, there must be alignment points located toward the center of the panel. To avoid this, use panel-based alignment or define the panel as a single board.

For loaderless systems (Series 2L) it may be possible to reduce load time using trailing edge detection. See chapter 3 of the Users' Guide for details.

It may be possible to reduce the alignment time by about 1 second on applications by shifting the alignment points from 1024 mode to 512 mode, and aligning at a single FOV. This FOV is preferably 800 (near the load height of the stage) or 400 (near the surface map Z height). To move the alignment points to the 512 mode it is necessary to redefine the components used for alignment as fiducials, and in the course of this, modify 4 or 5 of the NDF files. Below is an example showing the relevant parts of the files before and after modification.

Board.ndf:

OLD	NEW
.ALIGNMENT_PADS: 1 U2400 1 U2400 100	.ALIGNMENT_PADS: 1 fid2400 1 fid2400 100
.ALIGNMENT_PADS: 2 U90 132 U90 133	.ALIGNMENT_PADS: 2 fid90 132 fid90 133
.ALIGNMENT_PADS: 3 U1100 88 U1100 89	.ALIGNMENT_PADS: 3 fid1100 88 fid1100 89
@U2400 10335.0000 675.0000 90 TQFP100A	@U2400 10335.0000 675.0000 90 TQFP100A
@U90 7110.0000 3750.0000 270 TQFP176	@U90 7110.0000 3750.0000 270 TQFP176
@U1100 3510.497 3226.614 90 TQFP176	@U1100 3510.497 3226.614 90 TQFP176
	@fid2400 10335.0000 675.0000 90 TQFP100A
	@fid90 7110.0000 3750.0000 270 TQFP176
	@fid1100 3510.497 3226.614 90 TQFP176

Componen.ndf:

OLD	NEW
@U2400 true TQFP100A SM 10	@U2400 true TQFP100A SM 10
@U90 true TQFP176 SM 12	@U90 true TQFP176 SM 12
@U1100 true TQFP176 SM 12	@U1100 true TQFP176 SM 12
	@fid2400 true fid100A SM 10
	@fid90 true fid176 SM 12
	@fid1100 true fid176 SM 12

NEW

```
@fid176 19.6850 PLASTIC fiducial 1 -424.6060 -485.9060 R 7.0000 4.0000

@fid176 19.6850 PLASTIC fiducial 176 -485.9060 -424.6060 R 4.0000 7.0000

#
```

Fov_data.ndf

Package.ndf

OLD	NEW
.FOV: 12 400 10 1024 ADD	.FOV: 5 800 10 512 ADD
.FOV: 20 650 10 1024 ADD	.FOV: 12 400 10 1024 ADD
.FOV: 25 800 10 1024 ADD	.FOV: 20 650 10 1024 ADD
	.FOV: 25 800 10 1024 ADD
	@fid100a 800 512
@fid176 800 512	

NOTE: The added 512 FOV must be one of the three used for the regular test. Use of a larger FOV for alignment (650 or 800) will also decrease align time. It is easiest to set the pitch limit in the FOV file so that no components would naturally be tested at this FOV, then call out the fiducial packages to be tested there as done above. Only a few data lines for package shown here. Set the joint type to FIDUCIAL, but do not include any references to FIDUCIAL in the Algslice.ndf. This will result in added warning(s) (one for each subtype), but will allow for the faster alignment without adding extra views during inspection. The warnings read:

> No slices defined for FIDUCIAL <Subtype #>

A very small throughput increase can be realized by placing the last alignment point near the first surface map point and minimizing the X-axis moves between alignment points.

7. Surface Map Setup

The number of surface map points and the laser technique used can affect throughput. The optical pipelined mode is the fastest, the default, and the preferred technique. Since surface mapping in this mode is fairly fast (about 4 points/sec), it usually is not possible to make significant improvements by reducing the number of points. Eliminate any unused surface map points, which are listed in the compilation report.

8. Image Window

Minimizing the Image Window can decrease test times. The effect is more pronounced with slower processors and more computationally intensive programs. Applications that are governed almost solely by move time plus image acquire time may not be speeded up noticeably by minimizing the Image Window. To test this, note times with the Image Window displaying images and again with it minimized. This can vary from one panel program to another depending on the joint density, algorithms employed, and so forth, and from system to system depending on the processor speed.