

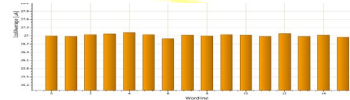
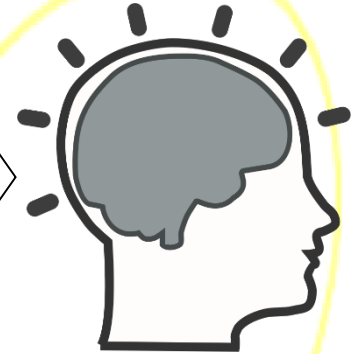
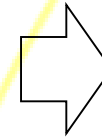
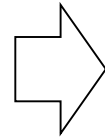
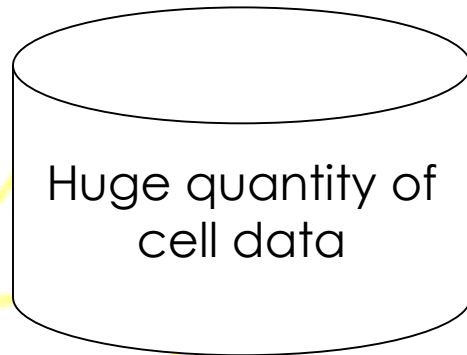
BarnieMAT

Bitmap Analysis Software
Product Presentation

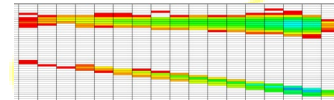
Fast Track to Your NVM

A long-exposure photograph of a highway at night, showing light trails from cars. The left side of the road has blue and white light trails, while the right side has orange and red light trails. The road curves into the distance under a dark sky.

Fast Learning of Array Effects



data reduction



visual analysis

- Unique industrial-level data processing tool, since 2007
- Used by 20+ memory makers and research institutes
- Covers all available test platforms and memory technologies
 - SRAM, DRAM, NAND, NOR, RRAM, PCM, ...
 - SLC, MLC, 3D, embedded
 - Up to 64Gbit arrays, up to 256 levels per cell



BarnieMAT Applications

Technology Development	Design Validation, Characterization Qualification	Yield Learning Failure Analysis
Programmability	Endurance effects	Engineering analysis of high volume production
Disturb sensitivity	Retention effects	Pattern recognition
Distribution tail bits	Statistical fail analysis	BIST signature
Topologic signatures	Algorithm evaluation	

“Extremely fast analysis of qualification data”
Senior Specialist, Infineon

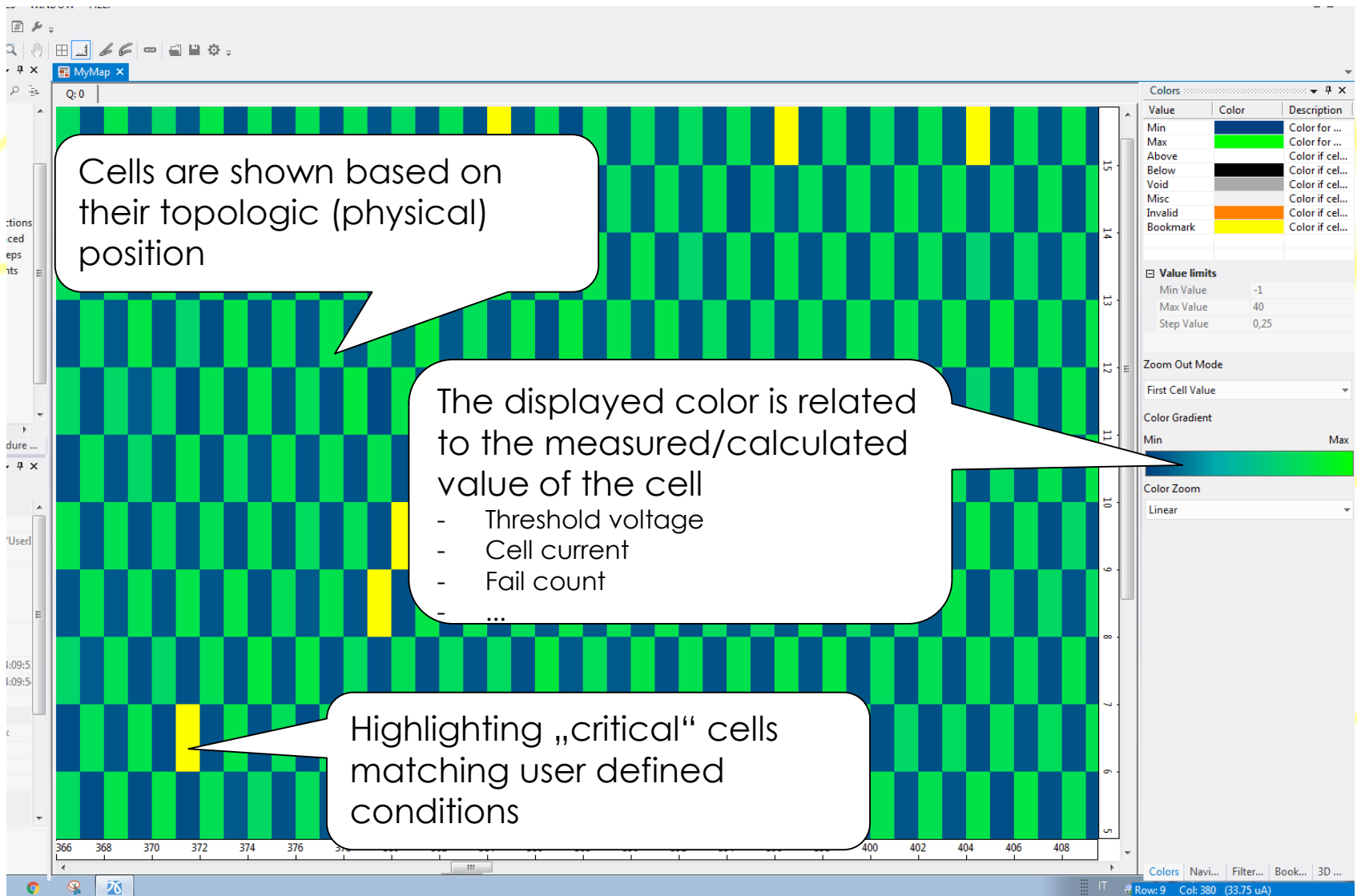


BarnieMAT

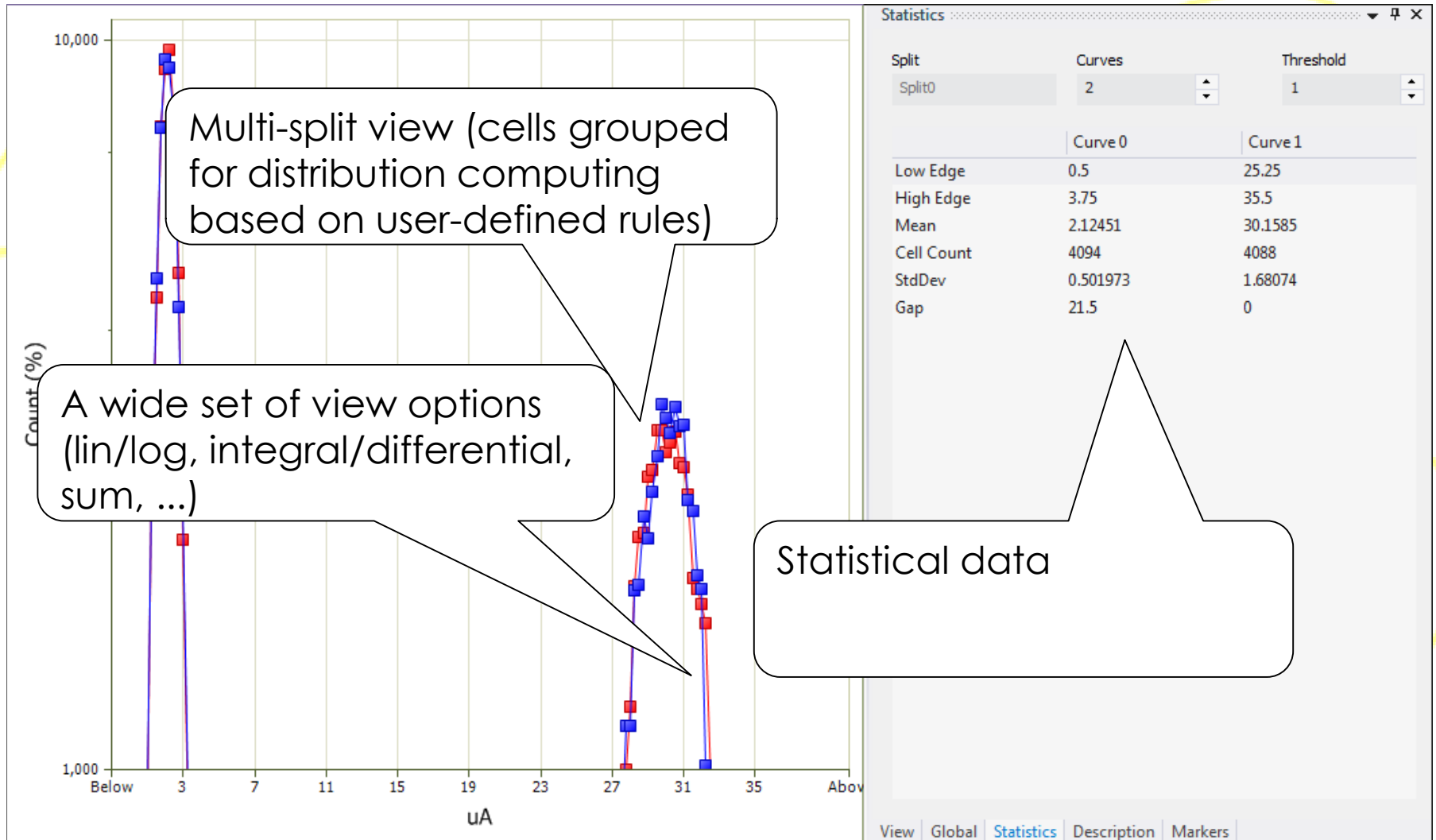
Best-in-class Presentation Tools



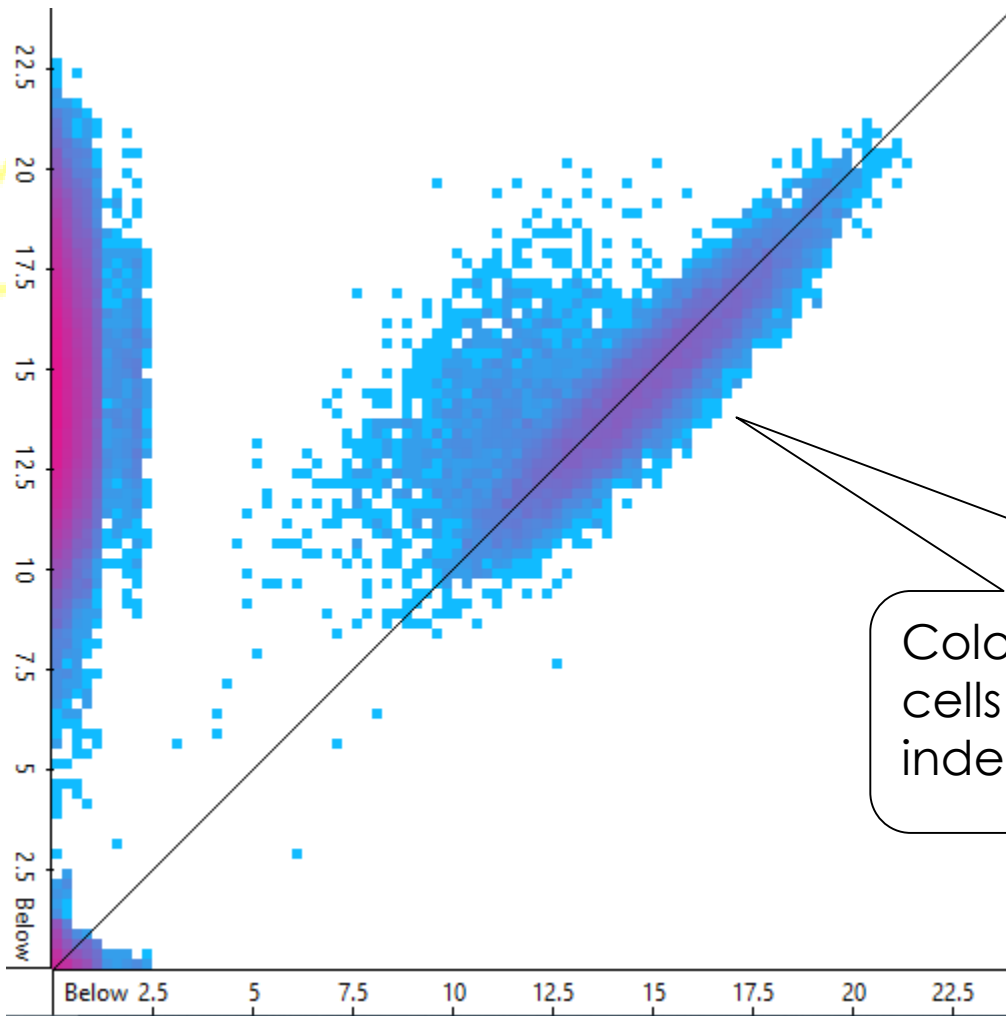
Bitmap Viewer



Distribution Analyzer



3D Distribution



Color codes the number of cells in function of two independent measurements



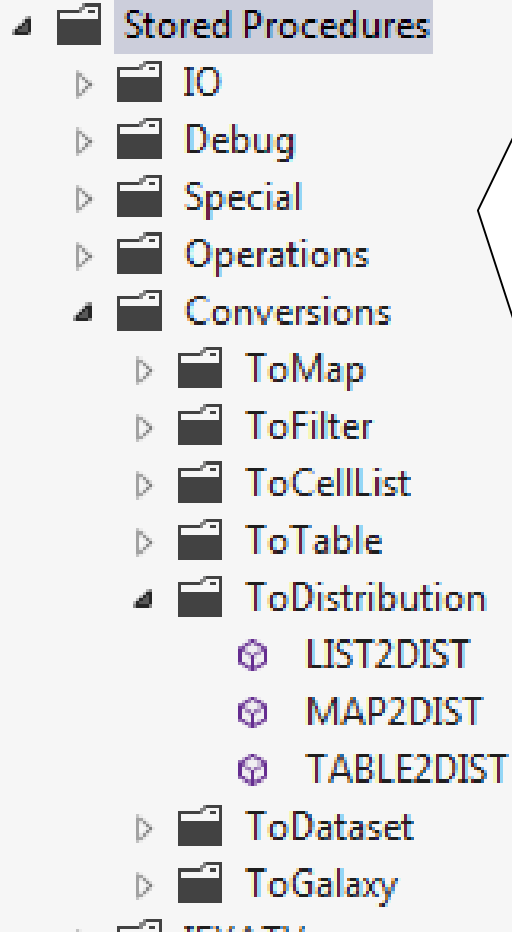
BarnieMAT

Advanced Processing Engine



Speed Optimized Processing Tools

Search stored procedure



- Compressions
 - ...extraction of a distribution from a map
 - ...vt average per wordline
- Operations
 - ...cell-per-cell difference of two maps
- Overlays
 - ...block level fail density map
- Pattern recognition
 - ...identification of neighbour bit fails
- Trace-back
 - ...topologic position of the distribution tail bits
- ...



Formula-Driven Computing

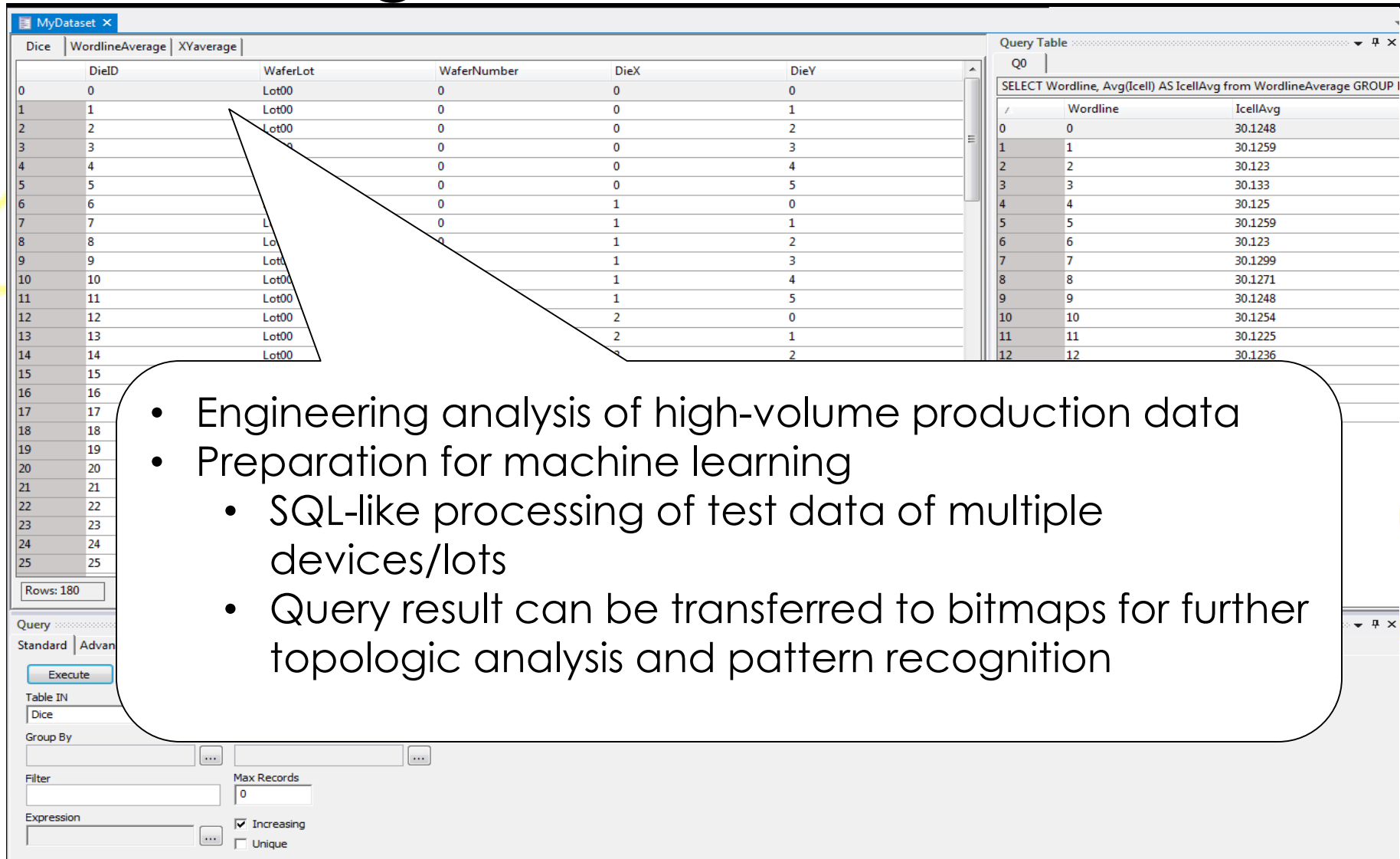
The image shows a software interface for data processing. On the left, there is a grid of cells with a checkerboard pattern of green and blue. The grid is labeled with row numbers 7, 8, 9, and 10. On the right, there is a configuration panel with the following sections:

- Filter**: Object
- FDL**: Value: `DATA > 33 AND SECTOR = 2 AND BL%2 = 0`
- Output**: Name, Object Type: Bookmark
- Process**: Run, Abc

A yellow callout bubble points to the FDL section, and another yellow callout bubble points to the grid.

User-friendly formulas to define which cells to process and how to group them

“Quite Big Data”



The screenshot displays a software interface with two main data tables. The left table, titled 'Dice', contains columns for DieID, WaferLot, WaferNumber, DieX, and DieY. The right table, titled 'Query Table', shows the result of a SQL query: 'SELECT Wordline, Avg(Icell) AS IcellAvg from WordlineAverage GROUP BY Wordline'. The query result table has columns for Wordline and IcellAvg, with rows numbered 0 through 12.

- Engineering analysis of high-volume production data
- Preparation for machine learning
 - SQL-like processing of test data of multiple devices/lots
 - Query result can be transferred to bitmaps for further topologic analysis and pattern recognition



Advanced Flow Automation

Stored Procedure MAP2DIST Parameters

Creates a multi-split distribution from a map using filtering

<input checked="" type="checkbox"/> Name	Value	Type	Description
<input type="checkbox"/> IN	/Demo/diffmap	Map	Name of the map to be converted
<input checked="" type="checkbox"/> OUT	/Demo/DiffDist	Distribution	Name of the distribution to create
<input type="checkbox"/> Filter		Filter	Name of the filter object to be applied
<input type="checkbox"/> FDL	BL%2 = 0 AND Sector=1	String	Filter expression to be applied
<input type="checkbox"/> SPLITNAME		String	Filter expression to be applied

```
ot.py x Python Demo - Basics.py x
1 # Automatically generated - do not delete
2 import barnie
3
4 # this is a demo script just for showing how basic
5
6 # we can define functions
7 def MyFunction ( mapname, intro = "hello" ):
8     print(intro, ":", mapname)
9
10 # we can define constants - in fact they are python
11 MySourceDir = "/Demo/PythonDemo"
12 MyMap = MySourceDir + "/" + "MyMap"
13 MyDist = "/Temp/MyDist"
14
15 # let's prepare a random map but before doing it be sure that we have the w
16 # this shows also how to call stored procedures and how to pass parameters ;
17 barnie.execSp("DirectoryCreate", DIR = MySourceDir, FAILIFEXISTS = False)
18 OriginalDirectory = barnie.execSp("DirectoryGetCurrent", DIR = MySourceDir)
19 barnie.execSp("DirectoryChangeCurrent", DIR = MySourceDir)
20
21 # create a random map
22 barnie.execSp("RANDOMIZEEMAP", OUT = MyMap, MAPTYPE = "RANDOM")
23
24 # we can call stored procedures and pass the parameters
25 print("Converting", MyMap, "to", MyDist)
26 barnie.execSp("MAP2DIST", IN = MyMap, OUT = MyDist)
27
28 # built in commands (dot-led in the .b scripts like
```

- Support for interactive execution
 - On-line help
 - Hierarchic organization of the functions with search possibility
 - Proposal of the lastly used input

- Python scripting for automation
 - Built-in GUI script editor
 - On-line help and word extension
 - Syntax coloring



Customization and Integration

Task	What To Do
Dedicated importers and exporters for proprietary data formats	C++ code with library support (off-line conversion to BernieMAT format)
Description of the device characteristics	C++ code XML configuration of templates
Additional, user-specific processing tools	C++ code with library support
Integration in a data processing environment	Use of the command line interface
Real-time tester connection	TCP-IP based protocol, tester-side library available





BarnieMAT

**supported by a skilled team of
industry experts**

**thanks
for your time and consideration**