DeviS: Motorola’s Near-Real-Time Device Information and Visualization System Using SAS/AF® and SAS/SCL.

Larry Worley, Motorola Inc., Austin, TX
Jim Nelson, Motorola Inc., Austin, TX

ABSTRACT

This paper provides an overview of DeviS, a near-real-time information retrieval and visualization system developed for device engineers at Motorola’s MOS-11 metallic oxide semi-conductor fabrication facility in Austin, Texas. The system was developed to provide device engineers and their managers with an easy-to-use point-and-click environment for navigating an hierarchical data structure with a wide variety of reports. The system was written using SAS/AF®, SAS/SCL, SAS/FSP® and SAS/ACCESS®. In addition, the reports provided by DeviS also use the SAS® Base and Macro code, SAS/GRAPH®, SAS/QC®, and SAS/STAT®.

INTRODUCTION

In a semiconductor fabrication facility, round wafers of silicon, with diameters from 3” to 8” and an initial thickness of about 300 microns (mm), are subjected to many processing steps in the creation of an integrated circuit. At the end of the processing step, the wafer is divided into rectangular "die." These die, when attached to metallic frames (which provide the visible insertion pins) and packaged in plastic, become integrated circuit chips with which all computer users are familiar. A single silicon wafer may have from less than 100 to more than 10,000 die, depending upon the size of the die and the size of the wafer. Typically, 25 wafers are processed as a lot.

modified semiconductor structure may be measured. Also, as the wafer nears the end of the fabrication process, it may be subjected to up to four tests including electrical diagnostic tests as well as functional tests. Additional functional tests are performed after the die are packaged.

The device engineer who is responsible for a particular part or device, monitors the progress of lots of wafers of her device through the entire flow of manufacturing steps. Upon completion of processing, this engineer is responsible for determining the final disposition of wafers in the lot. In other words, do wafers get moved forward to the final packaging facility, do they need to be re-worked, or do they need to be scrapped?

Consequently, these engineers are faced with the task of analyzing large amounts of data from disparate sources in a timely fashion. In addition, they must determine whether to request process changes for future and in-line lots. Both of these decisions are highly critical decisions with high degrees of risk.

HISTORY OF DEVI S

Prior to the development and introduction of DeviS, the data analytical needs of the device engineers at MOS-11 had been served by a variety of very disjointed products. These products suffered from a variety of malaise, including unreliability, non-portability and general volatility. to fully utilize these tools, an engineer had to possess substantial amount of programming expertise and database knowledge, in addition to engineering expertise.

In early 1994, a task force was organized at MOS-11 to replace these products with a unified dependable system. This system was to be easy to use and was to require little knowledge of the data storage structures. In addition, the system was to be readily extensible - both in terms of new devices being entered in the system and in terms of adding new reports and analyses to the system. This was the genesis of DeviS, which entered production in November 1994. The system is still being refined and extended as this paper is written.

STRUCTURE of DEVI S

The structure of DeviS may be broken into the following six areas:

1. A graphical user interface for choosing data to be analyzed or modified, as well as for choosing the analyses to be run.

2. A collection of production databases to which DeviS requires access.

3. Non-interactive reports which must be locally extensible.

Figure 1 - Picture of Silicon Wafer (Die)

At many process stages, critical distances on the newly
4. Interactive reports and analyses which must also be extensible.
5. A variety of editors and table maintenance utilities. These also must be extensible and are directed at users, as well as system support personnel.
6. An application output manager for managing both text and graphical outputs which gives the users an easy way of directing output to local print queues.

The Graphical User Interface

The graphical interface was written to provide two basic functions:
1. An intuitive method for choosing the data to be analyzed.
2. A quick and convenient way for choosing the analyses to be conducted on the chosen data.

The Data Selection Picklists

The interface provides up to four hierarchical levels of data selection, with a fifth non-hierarchical level of time. The data values available in each of the four hierarchical levels are displayed in SAS/AF Frame List Box Objects. These tables are dynamically filled or emptied as the user moves up and down the hierarchical structure.

The data values, which are included in these lists, are derived by using SAS/ACCESS to dynamically read a relational database. This database is constantly being updated as the testers generate new data. These list values also could be obtained from a local database or view. The structure of this dataset may be modified to fit local requirements.

This portion of the code was written to be easily site configurable. The list box titles, as well as the variable or column names from which the values are obtained, are read from an SCL list which is locally configurable.

Analyses and Tool Picklists

In addition to the data pick lists, there are also two other picklists on the front end. One of these provides a list of analyses and reports which may be selected. The other provides a list of tools which may be selected. The contents of both of these lists as well as the list box titles are modified to reflect the current state of the user's selection of data.

The data for these list boxes are obtained from a data set maintained locally. This data set contains such items as the report name, the catalog entry and type to execute for the report, and allowable data categories for the report, as well as other items.

Underlying Data Sources

DevIS directly accesses several production databases, which have been created using a variety of third-party and Motorola-developed tools. The decision was made to give the users real-time access to these databases so that data values were as current as possible. This ruled out the use of a Data Warehouse solution.

The end-of-line test data consists of electrical measurements and categorical data created from tests conducted on completed wafers. These data have been the primary focus of device engineers. These data are stored in ORACLE databases, which is created and modified by a Motorola developed product.

In-line data, created during the fabrication process, exist in several locations. A large amount of measurement and categorical data is maintained by the factory's Work-In-Process(WIP) tool, PROMIS®. These data are stored in operating system specific VSAM files. We are currently investigating a third party product which would present these tables as ORACLE tables to SAS/ACCESS. Additional in-line data are created and stored by several equipment automation tools which have been developed internally. These data are primarily stored in a SyBase® databases.

The final test data consists of measurements from functional tests conducted after die have been packaged in plastic. This testing is done at different locations for the different product groups. These data are also stored in ORACLE databases, created and modified by Motorola developed products.

SAS/ACCESS software is the vehicle which has made it possible to drive to the destination of a tool which provides users with near real-time access to data sources.

USER NAVIGATION OF GUI INTERFACE

The following series of four displays indicate a typical sequence of choices as the user navigates through the front-end GUI of DevIS. This display indicates the state of the system as the user initially enters it, or after the user has de-selected all data selection criteria.

![Figure 2 - Initial DevIS User Interface](image)

Based on performance considerations, the design requires the user to make a selection in the top level
picklist, which in this implementation is termed the product group. After this choice is made the next display appears.

![DevIS Standard Analysis](image)

**Figure 3 - DevIS User Interface After Product Selection**

When the top level group has been selected, there are five important changes to the frame:

1. The second level picklist is filled in with the data levels which exist for the top level selection. In this implementation, the term used for the second level is "maskset" that indicates a group of semiconductor devices or parts which use common photolithographic masks in their processing.
2. The title of the analyses picklist is changed to indicate that the analyses are assigned to the highest level of data—in this case, product.
3. The list of analyses in the analyses picklist has been created, containing only those analyses which may be run using product level data for the chosen product.
4. The title of the tools picklist has changed to indicate that the tools are assigned to the highest level of data—in this case, product.
5. The list of tools in the tools picklist has been created, containing only those tools which may be run using product level data for the chosen product.

At this point, the user can proceed by selecting either a single second level data criteria, a single tool, or one or more analyses. If a single second level data criteria is chosen, then the third level data picklist is filled and the titles and data in the analyses and tools picklists are changed accordingly.

![DevIS User Interface After Maskset Selection](image)

**Figure 4 - DevIS User Interface After Maskset Selection**

If the user then chooses a third-level data criteria, the fourth-level data picklist is filled, and the bottom lists updated accordingly.

![DevIS User Interface After Device Selection](image)

**Figure 5 - DevIS User Interface After Device Selection**

Finally, if the users selects a fourth level data criteria, only the bottom lists are updated.
The Analyses

The analyses list box contains three types of entries:

1. Non-interactive reports
2. Interactive reports
3. Interactive data extractions.

Generally, these three types are pre-defined in terms of format, type of analysis, type of output, etc. However, all of them allow the user to select the appropriate data entity to be analyzed or extracted, and also to select a destination and file or dataset names for extractions.

Non-Interactive Reports

The non-interactive reports generate either textual and/or graphical output in pre-determined formats. The text reports may be saved automatically to an export directory.

As DevI$S$ matures, it is anticipated that many of these reports will be available through some type of batch processing. These reports have been developed in several different ways. SAS/SCL is used for all SAS/BASE, SAS/Macro, SAS/QC, SAS/GRAPH, SAS/ACCESS and SAS/STAT are also used to varying degrees on these reports.

The following chart is an example of a graphical report which is available through DevI$S$.

Interactive Reports

Currently, the interactive reports consist of several flavors of a graphical wafermap tool, and of a tool which allows the engineers to test for a variety of spatial patterns on individual wafers and across lots. Both of these tools have been developed at MOS-11. The wafermap tools consist of a complex frame entry which allows the user to select the wafers to analyze, the colors to use for representing various data values, and the location or position of each individual wafer map, as well as other functions. This GUI then generates data step code which uses SAS/GRAPH to represent each wafer of a lot graphically. In addition, a variety of composite maps may be created which summarize across wafers of a lot.

Below is an example of a single viewport wafermapper:
Figure 8 - Example of a Single Wafer Map

The following display is an example of the wafermapper with wafers for an entire lot displayed with experimental split information.

Figure 9 - Example of a Wafer Map for an Entire Lot

Interactive Data Extractions

These analyses provide an easy way for users to extract the data used by DevIS into either SAS datasets and/or tab delimited text files. This provides a way of exporting data to other tools for ad hoc analysis.

EDITORS AND TABLE MAINTENANCE UTILITIES

Split Lot Editor for Experiments

A split lot editor has been created to enable users to analyze a lot which has been split for experimental purposes. The editor allows the user to assign different wafers to different split level, and to assign a title and label to each split level. This split definition can then be used by most of the lot-level reports and data extractions.

The user is able to save this definition and then toggle its activity on and off in the future. In addition, a user may publish the split, i.e., save the split definition in a public catalog where other users can read the split into their DevIS application and then easily use it.

Limits Editor

A limits editor has been created to enables users to modify boundary, specification, and control limits for a variety of parameters. These limits are used by several reports, and are also used in chron jobs which perform data summarization. This editor also has a field which users may modify to determine whether certain parameters will be included in reports.

The limits editor copies data from a global table into a data set which is local to an individual user. This user may then modify this data as she sees fit for use with her own reports. If the user has the appropriate privileges, he may load the modified limits back into the global table. Batch reports use the global data, whereas individual users have the option of selecting whether to use the local or global limits table.

Figure 10 - Class Probe Limits Editor User Interface

The following display is an example of a SPC report produced in batch mode and which utilizes the limits modified by the limits editor:

Figure 11 - Example SPC Report

Table Maintenance Utilities

There are several utility tools planned which will allow the device engineers who own particular parts to modify the underlying control data needed for generating reports for their respective parts.

APPLICATION OUTPUT MANAGER

When analyzing the capabilities of SAS's output manager and graph manager, it was determined that these tools were not adequate for providing a user-friendly environment for managing both graphic and text output. As a result, a new tool, the Application Output Manager,
was developed at MOS-11. This tool provides the users with a structured way of performing the following tasks:
1. locating the output of a variety of different analyses,
2. saving this output,
3. combining the output with output from other tools,
4. printing the output to a variety of local print queues on a variety of printers and operating systems.

In accomplishing these tasks, the Application Output Manager frees the user from knowing the various operating system and printer-specific options for printing both text and graphic output.

The following display is an example of the Application Output Manager's main window:

![Figure 12 - Application Output Manager](image)

**FUTURE DIRECTIONS**

Five areas have been identified as future areas for development in DeviS.

A metadatabase is currently being developed which will have a global knowledge of where the various databases are and of their schema. This metadatabase will be able to dynamically generate views to the databases which the reports in DeviS will then use. This will make the individual report and analysis programs much more data independent.

Additional reports and analytical tools are constantly being planned. There is a steady stream of requests for new items from the user base.

Cooperative processing on PowerMacs is being explored. MacIntoshes are the traditional Motorola solution for desktop computing. By porting DeviS and our other SAS applications to the MACs, we will be able to remove the X-terminals we have on most users desks. In addition, we will be able to distribute the computing load more evenly.

We are looking at using the Image Class in SAS/AF 6.11 to overlay photographic images on to the graphical wafermaps. This will prove to be a powerful analytical tool for our users when we are able to incorporate it.

The final area for future development at this point in time is to begin using the object-oriented programming techniques SAS/AF for re-designing the internal workings of the front end GUI.

**CONCLUSION**

DeviS has moved very quickly from a product with low user acceptance to one with extremely high user acceptance. The initial low acceptance can be attributed primarily to two factors: 1) an existing product, with which the users were comfortable, was being replaced, and 2) the risk of slow performance which is inherent using a prototypical development methodology.

The high level of current acceptance can be partially attributed to ease and speed with which we can build additional reports and extend the capabilities of the system. In addition, the intuitive interface has allowed new users to quickly learn to use DeviS.

The following SAS capabilities have contributed significantly to the success of DeviS:

1. SAS/ACCESS has enabled the near-real-time access of data from multiple data sources.
2. SAS's Multiple Vendor Architecture has allowed DeviS to ported to several platforms.
3. The frame builder from SAS/AF allowed a very rapid development of the front-end GUI, as well as other graphical interfaces employed in DeviS.

The following limitations of SAS 6.08 were also noted:

1. The output manager does not provide a very functional user interface for managing output.
2. Discrepancies between in the handling of textual output and graphical output presented design problems.
3. Slow performance of graphic procedures became very apparent in an interactive use.

Hopefully, these drawbacks will be corrected with version 7 of SAS.

**ACKNOWLEDGMENTS**

Several staff members of JJT Consulting, Inc. provided valuable assistance in the preparation of this paper. They are Elizabeth Crockett, Carlton Andrews, Brian Sharon, John T. Stokes, and Erich Stokes. George Campbell and Kevin Cox of Motorola also contributed to this paper.

SAS, SAS/AF, SAS/FSP, SAS/ACCESS, SAS/GRAPH, SAS/QC, and SAS/STAT are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

For more information about this article, contact the author:

Larry Worley  
6501 William Cannon Drive West, MD OE214  
Austin, TX 78735-8598  
512-891-6436  
larry_worley@oakqm3.sps.mot.com