Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

HIGH SPEED SEMICONDUCTOR FAB SIMULATION FOR LARGE, MEDIUM AND SMALL LOT SIZES

Peter C. Bosch

Highpoint Software Systems, LLC

S42 W27451 Oak Grove Lane

Waukesha, WI, USA

Robert L. Wright

International Sematech Manufacturing Initiative

2706 Montopolis Drive

Austin, TX, USA

ABSTRACT

This paper describes an analysis performed to assess the fidelity, scalability, and performance of the Sage® Fab Advisor™ semiconductor fab simulation engine executing two years of fab operations across a range of lot sizes. We describe the demand and fab operations models used, as well as the tools and methodology used in conducting the analysis. Our results were validated against a well-known model running on a well-known toolset, showing performance to be very competitive with that model. Further, we show that our engine's performance, running this model, scales almost linearly from 25 wafer lot sizes down to single wafer lot sizes. That is, simulation time increases roughly linearly with respect to the number of lots being processed.

1 INTRODUCTION

SEMATECH, ISMI (a subsidiary of International SEMATECH), SRC (Semiconductor Research Corporation), and the National Science Foundation have periodically combined their efforts since the mid 1990's to collaborate with universities to seek out potential improvements for semiconductor manufacturing. The Factory Operations Research Center (FORCe) enlisted the help of industry experts, professors, and students from around the world to solve real-world problems. One area of concern that was explored is the lengthy run-time of complex simulations in the semiconductor industry. There is interest and need in the semiconductor industry to develop simulation software that significantly reduces the time to answer specific wafer fab questions. Simulations are used to evaluate the interactions of complex real-world systems, which are difficult to solve mathematically (Law and Kelton 2000). As a simulation progresses and results are obtained, the results and behavior of the model should mimic the real-world situation.

Run-times can range from hours for less complex simulations, expand to a few days for moderately complex simulations utilizing 25-wafer carriers, and then up to a few weeks as carrier capacity decreases. The influence of doubling, tripling, and quadrupling the number of lots or carriers in the simulation greatly impacts simulation run-time. This is true for models that do not include the automated material handling system (AMHS) component in the simulation and is even more of an issue for simulations that do include the AMHS. ISMI and many of their member companies use modeling tools for complex simulations that require increased model run-time as complexity increases. Therefore, a solution to multi-day run-times for simulations needed to be explored. Software needs to provide answers and data of interest in a timely fashion for tactical decision making.

Next generation 300 mm factories will become even more complex compared to today's existing factories. In 2008, projects at ISMI within the Next Generation Factory Program are focused on cycle time improvement and cost reduction strategies. Each project has the potential to make positive productivity gains for semiconductor integrated circuit manufacturers.

Business models for some semiconductor individual device makers may require them to reduce cycle time. One approach to improving cycle time is to reduce wafer lot size. With this approach, improvements in cycle time have been demonstrated within device makers' factories. From a simulation view, by decreasing lot size (or carrier capacity) from 25 wafers to 12 wafers, run-times of simulation software can increase. ISMI modeling experience has shown that reducing lot sizes even further to one-fourth of the original front opening unified pod (FOUP) capacity from 25-wafer to 6-wafer lot sizes dramatically increases simulation run-time. Some ISMI

member companies have shown an interest in operating their factories with smaller lot sizes. Therefore, it is of great interest to simulation users that simulation software run-times be reduced to accommodate higher complexity simulations with equal wafer factory capacity while using smaller lot/carrier sizes. Progress has been made in the development of simulation software to reduce run-time regardless of the number of entities in the simulation.

This study conducted with Highpoint Software Systems attempts to build a prototype simulation tool with the intent of significantly reducing the run-time of the software under all conditions. In order to do this, a new and more advanced software platform needs to be employed.

2 APPLICATION DESCRIPTION

2.1 Technology Stack

This study was performed using the Sage® Fab Advisor™ (SFA), an application which Highpoint Software Systems, LLC (Highpoint) built for SEMATECH and ISMI.

2.1.1 Microsoft .NET

At the lowest level of the technology stack is Microsoft's .NET technology base, which provides languages, libraries, and tools designed for implementing a broad swath of general purpose programming challenges—from operations support and business applications to high performance gaming applications—in this study it was applied to simulations.

2.1.2 Sage® Simulation Libraries

On the .NET base is built Highpoint's foundation Sage® simulation libraries (originally written as the product HighMAST), which primarily uses the C# programming language. Details of the core libraries can be found in Bosch (2003), but in general, this level of the stack provides fundamental elements such as resource management, time management, and process and material flow.

2.1.3 Sage® Advisor Framework

On top of the Sage libraries, there is a general purpose engine that loads and runs models. This general purpose engine supports an XML form for models, databases of pre-built components, and ancillary elements such as report writers and a transport system. It has been used for educational, semiconductor, and pharmaceutical manufacturing models.

2.1.4 Sage® Fab Advisor™

Sage® Fab Advisor™ is the customer-facing and semiconductor specific layer of the technology stack and is comprised of two main parts. The first is a set of classes custom built for the semiconductor manufacturing domain. These classes are designed and built with the specific requirements of the domain in mind, as they pertain to the behavioral, integration, performance, and other specific needs of that domain. We believe this is a key reason for the high performance and easy integration of the solutions we have built, including SFA. During the project, our requirements underwent some rewriting such that reading and running one of the ISMI standard models and others similar to it became a central point of evaluation. The results of the validation study involving this model was to be seen as one of the more important litmus tests of success, So, the second main part of the SFA application, the importer module, was created to read data from these models.

2.2 Model Features

Model components include a series of input files providing information and data on wafer starts per unit time, equipment information, and the routes in which the wafers follow.

2.2.1 Equipment

Equipment data includes quantities of tools by tool group and performance characteristics such as wafer throughput and reliability of equipment. In addition, equipment is defined for external and internal lot capacity. Processing times can be defined in terms of the time to process a single wafer, as well as the time a batch tool requires for processing multiple lots. Tools have local storage, can process multiple wafers in parallel, multiple lots in parallel (batching), or multiple wafers or batches in series (cascading). Priority rules allow for varying the time-to-service lots after their arrival at a tool or tool family, and probabilistic distributions can be assigned to components to represent the stochastic nature of manufacturing. Equipment can be dedicated to specific process flows and process steps.

2.2.2 Process Flows/Routes

Routes consist of a defined sequence of process steps with appropriate tools assigned to each step. Multiple routes can be simulated simultaneously to represent the flows of different products. The time required to perform setups is characterized when lots of differing part numbers arrive. After a current product is processed at a tool and no other like-lots are available, a setup will occur before process-

ing a new product lot. Although not captured in this study, alternate routes representing re-work can be simulated.

2.2.3 Wafer Starts

Wafers can be started in the simulation software in variable lot sizes. Priority rules are set based on lot type; standard product priority, or lots with higher priority levels. Lot release rates can be simulated with probability distributions depending on the demand of products. The simulations of this study included lot sizes of 25-, 13-, 5-, and even single-wafer lot sizes to test the most extreme scenarios where the number of entities are 25-times more abundant. Lots are stored in infinite capacity stockers to represent storage locations.

2.2.4 Batching

The SFA tool includes a model of batching that, while not yet perfect, seems still to produce good numbers in validation (see section 3.1). The batching mechanism seems to be slightly more efficient than the one in our gold standard model, but can tie up a tool for a little longer than necessary—two effects that we believe will offset each other. Altering this mechanism is a relatively simple matter.

The batching mechanism is employed when a "batching" tool is paired with a "same setup" dispatcher.

A lot is handed off to a same setup dispatcher by the tool used in a preceding process step. If the lot is next for service, then the dispatcher looks at the lot's required setup, and based on that setup, checks to see if a tool is currently accumulating lots for a batch under that setup. If there is such a tool, the lot is sent to that tool. If there is not such a tool, then the dispatcher searches for another tool that is already equipped for that setup. If it finds such a tool, then it marks that tool as the "currently accumulating tool" and sends that lot and all following lots with that setup to that tool. If it does not find a tool already configured for that setup, and there is an idle tool, it reconfigures that tool, and makes it the currently accumulating tool. If it does not find a tool already configured for that setup, and there is not an idle tool, the lot is placed into tool family storage. Note that with sufficient backlog at the tool family, the tool will probably be fed a batch-full of lots as soon as it completes the previous batch.

2.2.5 Tool And Maintenance Downtime

Unscheduled downtimes are specifically represented by simulating distributions for mean time to fail and mean time to repair. Scheduled downtime is depicted in terms of the amount of time used for downtime pertaining to preventive maintenance.

2.2.6 Transport System

A tool-to-tool delivery system is assumed. While not modeling the AMHS dynamically, lots are delivered based on time simulated with a distribution captured in a from-to matrix with infinite delivery capacity. Future studies should incorporate the dynamic simulation of AMHS with a finite number of vehicles to see where congestion occurs as carrier capacity decreases and carrier count increases.

2.2.7 Reticles

Reticles or masks are required as new products enter the queue at lithography equipment. Wafer processing does not begin unless reticles are available.

2.3 Model Architecture

The domain objects in the SFA are designed to play certain roles and work together in specific ways.

At the highest level within the model there is a Market (comprised of a Demand Source and a Production Sink) and a Fab, shown in Figure 1. The Market expresses demand and consumes production, and the Fab consumes demand and generates production.

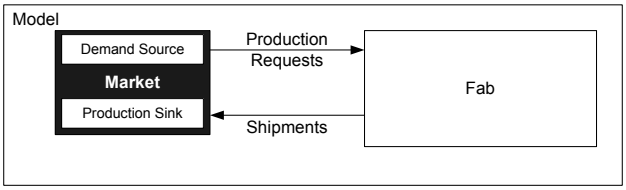

Figure 1: Overall Model Architecture

Figure 2 shows the Market Architecture. The Demand Source consists of a reader to load raw data and create a Demand Curve object for each product. The Demand Curve is read by a Production Request Generator which follows the time base of the simulation and generates orders for those products and quantities at appropriate times. It is also responsible for sinking the shipments generated from fulfilling those orders. In fulfilling this function, the Production Sink simply records and deletes the shipments.

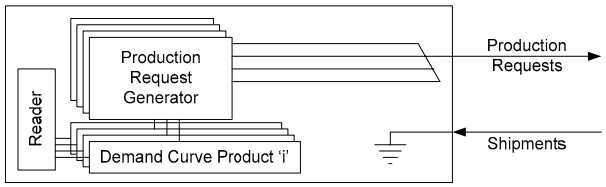

Figure 2: Market Architecture

The Fab consists of a Production Request Dispatcher whose job is to generate Lot Streams. The Lot Streams are responsible for generating, when requested by the Fab Product Dispatcher, new lots, and for correlating the returning, fulfilled lots with their originating Lot Stream and therefore, Order. The Fab Product Dispatcher creates lots, loads and assigns recipes and sends the lots to their first Tool Family, via the Transport System. At each Tool Family, a Tool Family Dispatcher is responsible for implementing family-wide selection rules, assigning lots to tools, and forwarding them through the Transport System to the next Tool Family specified in the recipe, or back to the Fab Product Dispatcher when the recipe has been completed.

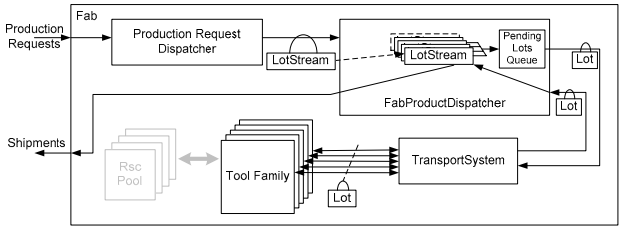

Figure 3: Fab Architecture

The Tool Family consists of a set of tools and a common dispatcher. Tools execute lot processing and are subject to failure and maintenance. The Transport System moves lots from tool to tool, with possible stops at interim storage locations (stockers) in between, and a time cost for each route.

Tool Selection is accomplished based on criteria expressed in the recipe, and is either same-setup or shortest queue/FIFO, and can include an override rule, which declares that a lot marked as "Hot" takes precedence in the tool over others waiting for processing.

Batching is done per wafer, or per lot, and supports a minimum and maximum size, a patience parameter which determines how long a tool is willing to wait to form a batch.

Calendaring is the mechanism whereby the simulation engine models failures, preventive maintenance, and the time required to return a tool to service. Tool down times are imposed non-preemptively, that is only once the current lot has exited the tool.

3 TEST DESIGN

We designed and built for performance, frequently using a validation process during development to gauge how close we were to our validation and performance goals.

The Gold Standard against which we validated the study was the ISMTap.asd model, available at the start of this effort on the ISMI website (see References for the web address), and run on Applied Materials then-current version of Autosched AP (ASAP). This decision was made primarily because that result set had already undergone a reasonable amount of scrutiny. We aimed for results from our execution of the gold standard recipe that matched within 10% (acceptable) or 5% (excellent) of those obtained by the AMAT engine.

Once we had obtained what we felt was a sufficiently valid result set, we ran a test that held market demand constant, but varied lot sizes, resulting in a significant increase in lot counts.

3.1 Validation

We built a spreadsheet that held each tool family's recorded idle and processing times, in percent of total simulation time, for the ISMI reference model running on AMAT's tools, and the SFA model. We continued iterating with changes in the SFA mechanisms, until we had reached acceptable values, with two exceptions, described below.

Table 1 shows the results obtained and their meanings. The detailed table of results is available at Bosch (2007b).

Table 1: Validation in Brief

Validity	# Families	Notes (idle and processing time deviation)
Excellent	52	< 5% deviation on both
Split	6	<5% deviation on one, <10% other
Acceptable	3	Both < 10%
Unacceptable	2	One or more >10%

We decided to stop with two tool families still not meeting the acceptance criteria, since we believed we understood the reasons for the variance to a degree that we were confident it would not have a significant effect upon the performance results.

Following are the two tool families for which we saw unacceptable deviation from the standard values.

The Furn_OxAn(I) tool family spent 15% more time in idle with SFA than it did with AMAT. We believe this to be due to a more efficient batching algorithm implemented in SFA than in AMAT—SFA achieved an

average of 3.5 lots per batch and AMAT achieved an average of 2.8 lots per batch.

The Wet_Bench_F tool family in SFA spent 9% less time in idle, and 9% more time in setup than in AMAT, but within 2% of each other in processing time.

Based on the above, we decided that while there may still be questions we would like to answer, we had a good enough basis upon which to conduct a batch-scaling performance test.

3.2 Performance Test Characteristics

Our performance test models each used one of four lot sizes, 1, 5, 13, and 25 wafers-per-lot, and approximately the same wafer release rate (i.e., when lots were 1/N smaller, they were released N times faster). Each model had 30 process flows with equal proportions of 596, 584, and 505 steps. The exception was the single-wafer lot model, where we used 587 tools, divided among 63 tool families. In the single wafer lot model, we increased the number of CVD_LowK tools from 60 to 90 to achieve stable production. Raw data shows that this is because it became a bottleneck tool family, and its quantity at the single-wafer lot model was simply insufficient to keep up with the production quotas, in wafers per month, that the other models managed.

We ran each of the models for 1018 production days.

There were 8 failure calendars and 8 maintenance calendars, providing 1779 attaches.

The transport system included 17063 separate routes with transport costs of between 5 seconds and 5-½ minutes.

4 TEST EXECUTION

4.1 Test Machine

The machine used to perform the test was a Lenovo T60 laptop with T2600, 2.16 GHz CPU and 3 GB of RAM. This chip has a 667 MHz front side bus, and a 2 MB L2 cache running at 2.16 GHz. The simulation process was constrained to a single CPU, and only a few hundred incremental megabytes of RAM were actually used by the simulation process for each run. The operating system in use on this machine was Windows XP, service pack 2.

4.2 Results

We collected six sets of data for each simulation. These data were WIP level over time, tool, and tool family reports, each describing the percent of time each tool or tool family spent in each state, queue levels for tools and WIP levels for stockers, and a lot report that shows cycle time and several other data points for every tenth lot.

We present run time, both raw and normalized, WIP level and abridged tool family utilization data in this paper. More complete data are available at Bosch (2007).

Table 2 describes the run times achieved for each lot size. Note the lack of dramatic increase in run-time, as we decrease lot size.

Table 2: Execution in Brief

Wafers per Lot	Total Number of Lots	Total Number of Wafers	Run Time (sec)
1	1150166	1150166	7222.3
5	230047	1150235	1238.2
13	88491	1150383	449.8
25	46025	1150625	309.1

Figure 4 shows run time for the four tests plotted against the number of lots processed in the simulation. Note that SFA maintained a very nearly constant time per thousand lots processed.

In Figure 5, we plot run times per thousand lots. If run time were blowing up, we would expect to see this rise dramatically, the fewer the wafers per lot, but instead, we see a relatively constant set of values, implying that we have not encountered a rise in the rate of increasing run time that could signify diminishing returns.

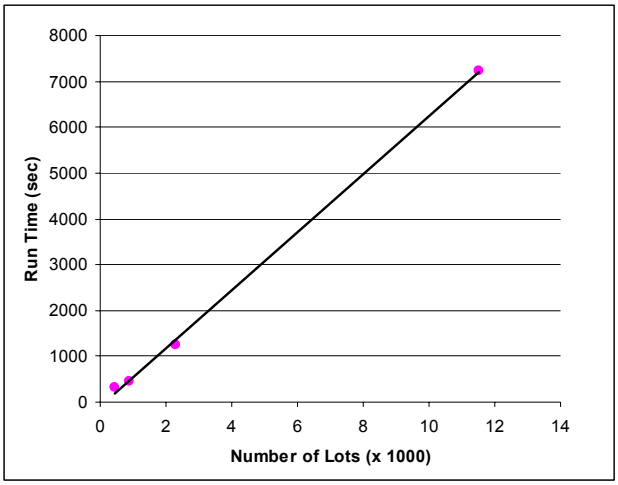

Figure 4: Simulation Run Times

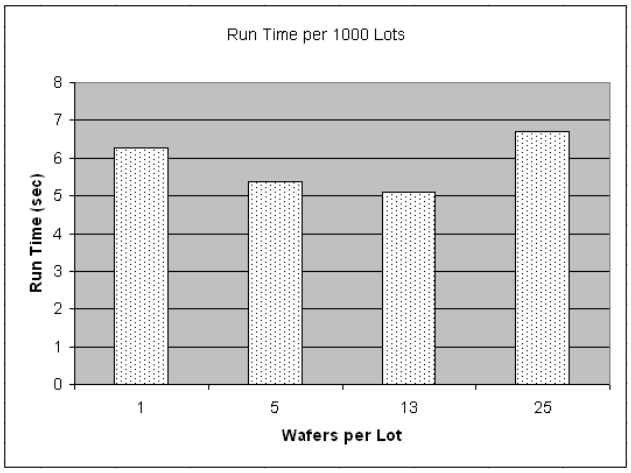

Figure 5: Simulation Run Times per 1000 Lots

Figure 6 shows the WIP levels, in lots, present in the Fab over time in each simulation run, and with the exception of the 25-wafer experiment, reflect similar WIP level, in wafers, across each experiment. We plan to investigate the 25-wafer experiment exception.

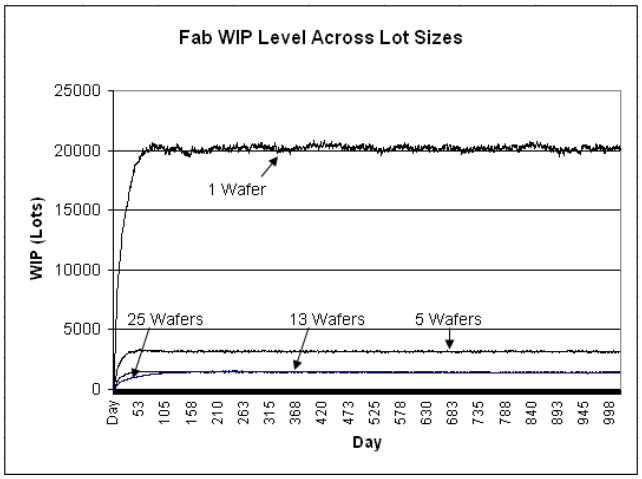

Figure 6: WIP Level In Fab for Each Lot Size

Figures 7, 8, 9, and 10 show the utilization for a selected 30 tool families. We do not show the full 63 tool families' utilization simply for reasons of space and resolution in this report. The full charts are available at Bosch, P.C. (2007a).

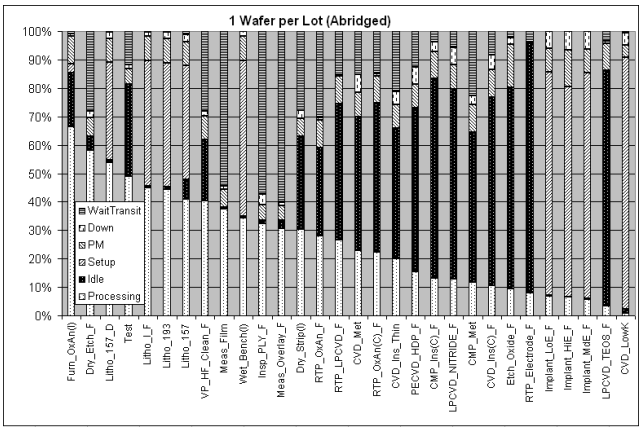

Figure 7: Tool Utilization (abridged) 1 Wafer per Lot

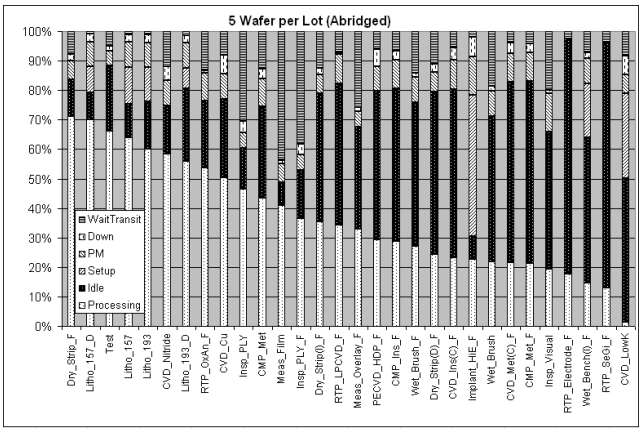

Figure 8: Tool Utilization (abridged) 5 Wafer per Lot

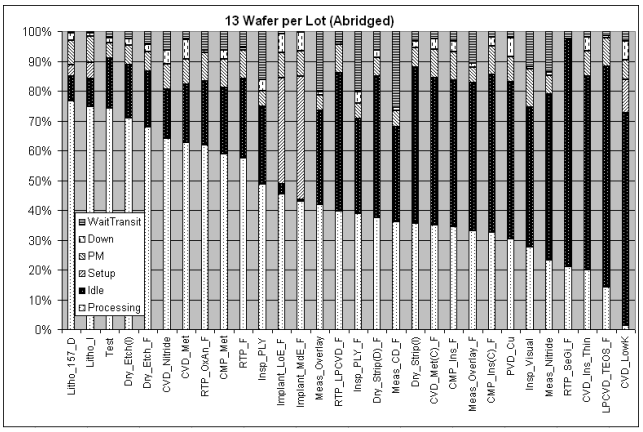

Figure 9: Tool Utilization (abridged) 13 Wafer per Lot

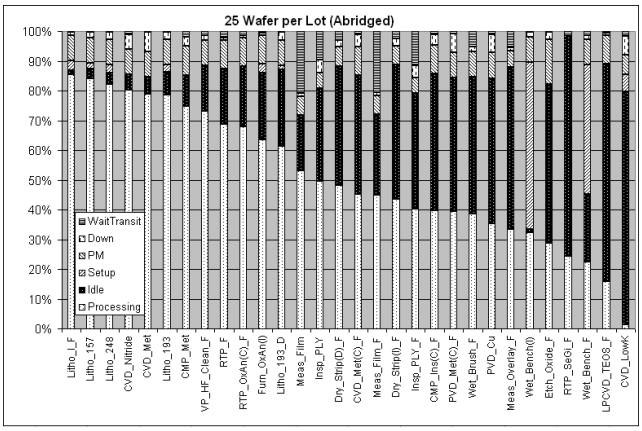

Figure 10: Tool Utilization (abridged) 25 Wafer per Lot

Note that in Figures 7–10, different tools became the bottleneck as lot size changed. Dry_Etch(I) was the bottleneck in the 25-wafer baseline model. In the 13 and 5 wafer models, Litho_I_F was the bottleneck, And the bottleneck in the 1-wafer model is not shown, but from above, we believe it was the CVD_LowK family. We composed another chart showing the top ten tool families in the 25 wafer scenario, and their ranks in the other scenarios. This information is shown in Figure 11. Recall, also, that we had to increase the number of CVD_LowK from 60 in the 5-, 13-, and 25- wafer scenarios, to 90 in the 1 wafer scenario so that it would not be the (destabilizing) bottleneck.

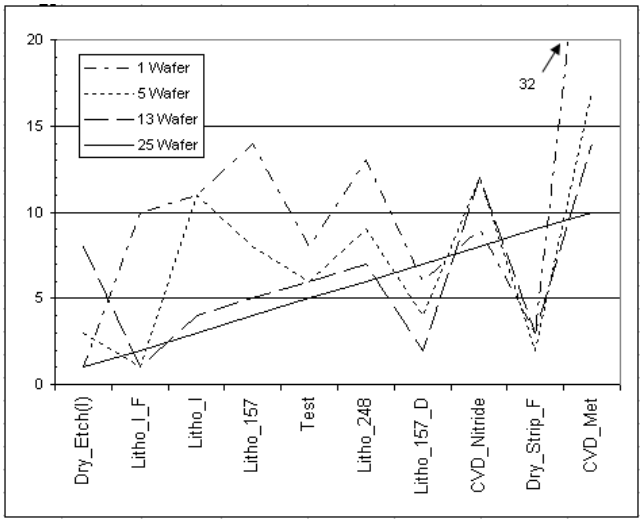

Figure 11: Changes in "Most Utilized" Rank per Scenario

In Figure 11, Dry_Etch(I) is the most utilized tool in the 25- and 1-wafer-per-lot scenarios, but the third most utilized in the 5-wafer scenario, and the eighth most utilized in the 13-wafer scenario. Where Dry_Etch(I) is not the most utilized tool family, in the 5- and 13-wafer scenarios, the Litho_I_F tool family is the most utilized.

Further, we see that Litho_157_D and Dry_Strip_F are far more constrained in the 1-, 5-, and 13-wafer scenarios than in the 25-wafer scenario.

5 FUTURE RESEARCH

The Sage Fab Advisor is a prototype simulation software and provides the framework and opportunity to engage in further research. Further validation is in order to understand any correlation that might exist in the Sage Fab Advisor for simulation run times compared to other factors such as lot cycle time, equipment availability, and utilization. Further development prior to use in a factory setting will be required. Future work and analysis could include the impact on run-time of the Sage Fab Advisor by increasing product count to represent a high mix factory. Other simulation analyses should incorporate Next Generation Factory concepts; improving equipment availability, reducing variability of equipment failures and repair times, and implementing single wafer manufacturing.

6 CONCLUSION

The work entailed in this article develops an alternative and new simulation software. This effort has been the most significant effort in reducing simulation run-time while simulating complex semiconductor factories. In general, run times were reduced on average by approximately 15X. While we are not familiar with the internals of the tool that served as the control in this experiment, we believe that newer technology, and algorithms that were more closely-tailored to the problem at hand are likey explanations for a significant part of the difference in performance. The typical 25-wafer carrier simulation that simulated 33,000 wafer starts per month for 1,018 days only used about 5 minutes of execution time. As wafer count is reduced per lot size and the number of carriers increase in the simulation, we would typically see considerable increases in simulation run-time in currently-available simulation software that is capable of simulating the complexity of semiconductor manufacturing. However, in the SFA, reducing to 13-wafer lot size increased run-time to 7.5 minutes. And, increasing the number of lots to 5X by simulating 5-wafer carriers, run times with the same model characteristics only increased to a total run-time of 20.6 minutes. Furthermore, reducing lot size to single wafer lots, run-times increased to 2 hours. Simulation software has shown to take several weeks for this level of complexity.

This version of the SFA is v1.0 and therefore needs further development prior to use in a factory setting. Specifically, we are aware that we still have some questions to resolve surrounding equipment batching. It is hopeful that future collaborations will further develop this software for simulation users in the industry.

ACKNOWLEDGMENTS

The authors wish to express their gratitude to the member companies of ISMI for their support and direction. We would also like to thank Kranthi Adusumilli for his contributions.

REFERENCES

Bosch, P.C. 2003. Introduction to HighMAST. In *Proceedings of the 2003 Winter Simulation Conference*, ed. S. Chick, P.J. Sanchez, D. Ferrin and D.J. Morrice, 1852–1859. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Bosch, P.C. 2007a. Experimental data. Available via <http://www.highpointsoftware.com/SFA/SmallLots/RawData.pdf> [Accessed April 14, 2008].

Bosch, P.C. 2007b. Comparison available via <http://www.highpointsoftware.com/SFA/SmallLots/Comparison.pdf> [Accessed April 14, 2008].

Law, A. M., and W. D. Kelton. 2000. *Simulation modeling and analysis*, 3rd ed. New York: McGraw-Hill.

ISMI website: <http://www.sematech.org>.

AUTHOR BIOGRAPHY

PETER C. BOSCH is a founder of Highpoint Software Systems, a small and attentive decision-support technology firm in the upper Midwest. He holds a BSEE from the State University of New York, and is a Certified Java Developer and Microsoft Certified Solution Developer. Pete has published numerous technical articles on object-oriented development in these environments. Pete has been designing and building simulations since 1991, for Fortune 100 firms in aerospace, medical imaging, pharmaceutical manufacturing and investment banking. He has been leading large software projects since 1995.

ROBERT L. WRIGHT manages the discrete event simulation program and cost modeling efforts at ISMI. He has 12 years of experience in both dynamic simulation and static cost modeling. Robert received a BBA degree in Management and a Masters of Science in Technology from Texas State University.