Proceedings of the 2003 Winter Simulation Conference
S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds.

RESIDENT-ENTITY BASED SIMULATION OF BATCH CHAMBER TOOLS IN 300MM SEMICONDUCTOR MANUFACTURING

Nirmal Govind

Department of Industrial and Manufacturing Engineering
The Pennsylvania State University
University Park, PA 16802, U.S.A.

David Fronckowiak

300mm Industrial Engineering
IBM Microelectronics
Hopewell Junction, NY 12533, U.S.A.

ABSTRACT

This paper describes a resident-entity based pilot simulation study of a class of tools used in 300mm semiconductor manufacturing known as the *wets* tools or the *wet benches*. These are *batch chamber tools* - they have several chambers or tanks, each of which can accommodate a batch of wafers, usually more than one lot size. We develop a simulation model for the wets processing area that is based on the resident-entity paradigm, but makes use of transient-entity-type modeling when more information needs to be tracked. Resident-entity models tend to be much faster than transient-entity simulation models that are common in semiconductor manufacturing. The model developed captures most of the internal workings of a wets tool and at the same time, models different types of tools. We used the model to evaluate the effects of scheduling policies and batching parameters on the performance of the wets process area.

1 INTRODUCTION

IBM's Microelectronics Division is in the process of ramping up its 300mm semiconductor fabrication facility (fab) in Fishkill, NY. This fab has some of the most advanced equipment used in a semiconductor fabrication facility and features a fully automated material handling system. Given that most of the 300mm tooling is considerably different from its predecessors, it is critical to understand the functioning of these tools in a short time frame in order to have them configured to run in the best possible manner at the earliest. Due to the complexity of these tools, simulation has a great deal to offer since the modeler can adequately model most of the inner workings of the tools, and more importantly, simulation provides the capability to analyze the effects of changes in system settings before they are put in place.

The class of tools modeled in this paper is what we will call the *batch chamber* tools. They are found in the wets processing areas in a fab and are used for cleaning wafers and etching. We describe these tools in more detail in a later section, but for now, it suffices to note that these tools are complex enough that an analytical model would be difficult, if not impossible, to construct. We also note that the batch chamber tools are similar to cluster tools, which have received quite a bit of attention recently, however a significant difference is that a cluster tool processes one wafer at a time in a chamber whereas a batch chamber tool processes a batch of wafers in a chamber. This adds more complexity to the model in terms of the batching decisions that need to be made for these tools. A simulation model was built to analyze the effects of various batching parameters and scheduling policies on the performance of the tools in the wets area. This model is based on the resident-entity modeling paradigm that we describe below.

A discrete event simulation model typically consists of two types of entities - those that are always present in the system (or constant, as far as the system is concerned) are called *resident-entities*, and those that are temporarily in the system are called *transient-entities*. For example, in a simulation of a queueing system, the servers in the system are the resident entities and jobs, that enter the system at some point and leave after being processed, are the transient entities. This classification of entities gives rise to two simulation paradigms. The resident-entity simulation paradigm focuses on the *resident-entity cycles* whereas the transient entity paradigm tracks the transient entities as they move through the system. Thus, in a queueing system, the server cycle would comprise of, for instance, three states - busy, free and down. There are advantages and disadvantages to both the paradigms but the noticeable advantage of resident-entity simulations is that they tend to be much faster than the transient-entity simulations. This is primarily due to the fact that transient-entity models become very slow as the number of transient entities (jobs, for instance) increases and the memory footprint grows. Resident-entity models do not suffer from this problem since the number of resident entities remains a constant and hence the execution time does not depend on the number of transient entities since they are not explicitly modeled.

Typical models of semiconductor wafer fabs that use transient-entity modeling are able to provide a considerable amount of detail on each lot as it moves through the fab. Roeder, Fischbein, Janakiram, and Schruben (2002) have demonstrated, in the case of a wafer fab, that a pure resident-entity model runs at least one order of magnitude faster than a transient-entity model of the same system. They note that a pure resident-entity simulation may not provide adequate amount of information on the transient entities in the system. Schruben and Roeder (2003) conclude, based on an enhanced simulation model (of the same system) that provided the missing information on the transient entities, that the faster run times are due to the inherent differences in the event scheduling paradigm as opposed to the process flow paradigm that is used in most commercially available simulation packages.

In our wets simulation model, we use a *hybrid modeling* approach where some parts of the model are transient-entity like, which provides us with the information we need to employ fairly complex scheduling policies, and the rest of the model uses a resident-entity approach that allows us to take advantage of the faster run times. The wets model was developed in SIGMA (Schruben and Schruben 2001, Schruben 1991, Schruben 1990), which uses the event scheduling paradigm. SIGMA allows the user to build models graphically, the model in SIGMA is a graph with vertices representing events and edges that represent conditions being checked as well as time delays between events. The software places no limits on the complexity of the model and this makes SIGMA a very attractive option for modeling systems that require complex logic to be built into the model.

Models for process areas in the fab have been built in the past. However, there has been little work that has looked at modeling batch chamber tools. Mauer and Schelasin (1993) consider the problem of simulating integrated tools in the wets area. It is important to note that due to advances in technology, the wets tools used in 300mm manufacturing are considerably different from their counterparts in earlier generation fabs and some of these differences make the modeling of 300mm wets tools more complicated. For example, the 300mm wets tools have an internal buffer or stocker that can store lots before they are processed in the chambers. Kunesh, Mauer, Gow, Liehr, and Cohen (1994) evaluate the trade-offs between footprint advantages and throughput loss associated with such multifunction wets tools. They note that improved particle performance on the wafer surface (as a result of fewer air/liquid interfaces that the wafers need to pass through) is a strong justification for the use of such processing equipment.

As noted earlier, cluster tools are similar to the wets tools. Though there are significant differences between cluster tools and wets tools, the simulation models built for cluster tools have similar objectives to the one considered in this paper and hence we look at selected works that deal with cluster tool simulation. Dümmler (1999) uses a combination of simulation and a genetic algorithm to study the problem of sequencing lots on an available set of cluster tools to reduce cycle time. Koehler, Wulf, Bruska, and Seppanen (1999) develop a cluster tool simulation for thin film head production. They analyze the effect of various tool configurations on the cycle time of the cluster tool. LeBaron and Pool (1994) present a simulation model for cluster tools built in AutoMod, a simulation tool from Brooks Automation. They stress the importance of using a simulation model for cluster tools to perform what-if analysis. LeBaron and Hendrickson (2000) present a model for cluster tool simulation and use it to emulate a real cluster tool scheduler. Both these works stress on the importance of simulation in evaluating throughput and utilization of the tools and determining optimal configurations.

2 BACKGROUND

The wets process in semiconductor manufacturing is a set of operations that either clean the wafers or perform etching. Etching is a process by which some parts of a photoresist coating on a wafer are removed. The wets process is performed in specialized tools that contain multiple processing *tanks* or *chambers* (we will use these two terms interchangeably). The process involves the shooting of multiple streams of chemicals at the wafers in the tanks. Below, we look at the wets tool in more detail and try to bring out the level of complexity involved in the modeling of the tool. We then look at some of the reasons why a simulation model of the wets processing area is useful. The following definitions will be helpful in understanding the remainder of this paper.

An *operation* is defined as a sequence of chambers that a lot needs to visit in order to finish processing at a tool. An operation also specifies the processing time in each of the chambers. Usually, lots need to visit more than one process chamber in a tool to complete the operation at the tool. Hence, an operation typically consists of visits to multiple chambers in a particular order.

A *tool set* refers to a particular type of tool and may consist of several copies of the same tool type.

An *operation backup plan* specifies, for each operation, the set of tools that can be used to perform the operation. An operation backup plan for a particular operation may specify tools from different tool sets. The *primary tool set* for the operation is the tool set to which the first tool specified belongs. The tool sets to which the other tools belong are called *alternate tool sets*.

A *batch* refers to a set of wafers that is processed together in a tool. Batching is performed based on the operations. There exists a set of relationships between the operations that specifies the operations that can be batched together. The physical characteristics of the tool define the maximum number of wafers that can comprise a batch. The maximum batch size is usually more than one lot size (25 wafers). Once a batch is formed at a tool set, the wafers in the batch remain together until the end of processing on the batch at that tool set.

The *cycle time* of a lot at a tool set is defined as the time elapsed from the moment the lot arrives at the tool set until it leaves one of the tools of the tool set after processing.

2.1 Wets Processing Area

Below, we describe the wets tool and look at the mechanism by which lots that arrive at the wets area for processing are scheduled on the tools.

A 300mm wets tool has the following components - two load ports through which lots are loaded into the tool and exit the tool after processing, an internal buffer which can store lots until they are ready to go into the tanks, a series of tanks (or chambers) in which batches of wafers are processed, a main handler or robot that moves batches from the internal buffer to the tanks and also from one tank to another, a mini handler or robot within each tank that receives the batch from the main handler and holds the batch until the end of processing in the tank. Each tank is fitted with multiple inlets for chemical and water rinses. The combination of chemicals in a tank defines the process that is performed in the tank.

A lot arrival to the wets area corresponds to the end of processing for this lot at its previous step. Though in reality, the lot may be physically transported to a stocker in the fab, it can be thought of as joining the *virtual queues* at the tool sets defined for the current operation in the operation backup plan. Batches are formed at the tool set queues based on the operations needed for the lots in queue. Formation of a full batch from the lots in queue triggers the scheduling policy. The scheduling policy evaluates the tools in the primary tool set and any other tools specified by alternate tool sets in the operation backup plan, and decides which particular tool, from the primary or alternate tool sets, will be used to process the batch. If a full batch cannot be formed immediately, the lot waits in the queue until a full batch can be formed or a timer runs out. The timer is specified by the Wait No Longer-than Time (WNLT). Once the timer runs out, the existing lots in the queue are batched and sent to the tool that is selected by the scheduling policy. We note that the batching of lots at the tool set is a *virtual batching* and the lots do not get physically batched until they are inside the tool's internal buffer.

When a lot arrives at a tool, it is moved from the load port of the tool to the internal buffer. The internal buffer of the tool has a limited capacity in terms of the number of lots it can store. The tool has an internal scheduler that sequences the start of each batch after considering the current state of the tanks in the tool. The role of the internal scheduler is something worth looking at in more detail. The following are the main characteristics of this process that are important in this respect - (1) Due to the nature of the process, there are pre-determined constraints on the time that can be spent by a batch within certain types of tanks; (2) The tanks are scheduled for periodic *tank dumps* during which the entire tank is emptied and refilled with the necessary chemicals; (3) Some tanks need to be heated up in advance for a specified amount of time before the process starts; (4) A batch of wafers can be moved into the next tank in its chamber route only when that tank is empty and the main handler is free to pick up the batch from its current tank; (5) An operation may specify several alternate chambers for a particular process in a tool, thus if the primary chamber designated for processing is unavailable, the alternate chambers are evaluated to determine which alternate chamber can be used. The internal scheduler looks at the various batches (in the internal buffer) that are ready to be processed and evaluates the above mentioned constraints to determine which batch should be run first. Unfortunately, the algorithm that is used in the internal scheduler is not revealed by the vendor and this has implications on the validation of any simulation model developed for these tools.

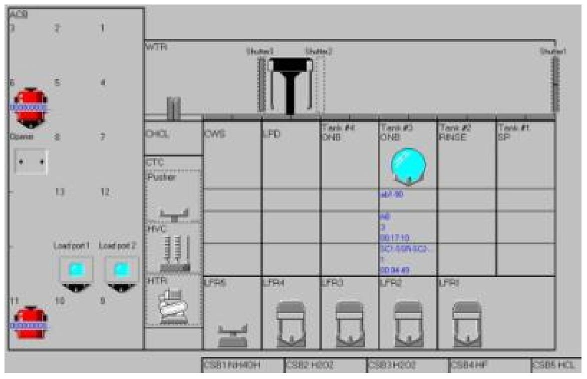

Figure 1: Wets Tool

Figure 1 is a schematic of the internal configuration of one of the wets tools. At the time this schematic was generated (using the tool's software), two lots (extreme left side of the picture) are in the internal buffer of the tool. The two load ports are also shown, adjacent to each other. A set of mechanical devices (aligned vertically in the picture) physically batch the wafers from the internal buffer and move the wafers to the area from where the main handler or robot can pick up the batch. The tool shown has six tanks, the first tank is a waiting area for new batches, the next three are processing tanks, each of which has a mini handler, and the last two tanks are linked processing tanks, sharing one mini handler between them. The main handler is shown waiting on top of the second tank and there is a batch being processed in the fourth tank.

2.2 Need for Simulation

We briefly describe some of the reasons why a simulation study of the wets processing area was undertaken. As described above, the combination of chemicals in a tank determines the process that is performed in the tank. Different types of tanks are defined based on the process that is performed in the tank. The assignment of tank types in a tool is done by the process engineer. Assigning tanks in a tool is equivalent to specifying the various operations that the tool can perform. There may be several operations that need to be performed in the wets area depending on the number of products in the fab. Since the number of tools available is limited, this assignment needs to be done in such a manner that the average cycle time in the wets processing area is minimized. Hence a simple constraint that may be put in place could ensure that there exists at least one backup tool for each operation. Thus if one tool is down for some reason (maintenance or failure), the other tool could be used to run the same operation. The problem of determining optimal tanks assignments is a complex problem in itself and will be addressed in a later paper.

In this paper, we evaluate the performance of different scheduling policies that decide which of the available tools that can perform an operation is the best candidate, given the current status of the tools. The scheduling policy impacts the cycle time of the tools in the wets area since a policy that balances the work-in-process (WIP) at the tools well, based on the operation backup plan, will achieve lower cycle times than other policies that do not route the lots efficiently to the tools. We also evaluate batching parameters that have a significant impact on the cycle time.

3 SIMULATION MODEL

3.1 Overview

The simulation was created in SIGMA which uses the event scheduling paradigm (Schruben 1983). The simulation proceeds by scheduling events that schedule future events based on conditions that are checked when an event occurs. SIGMA converts the model, built as an *event graph*, to C source code which is run using Visual C++. We discuss the event graph of the SIGMA model in the following subsection.

The model captures the inner workings of the wets tools by modeling the resident entities *within a tool* - individual tanks, the internal buffer, the main handler. Lots are batched before they go into the tanks for processing and the model also tracks the batches that are formed. The batches in this model represent the transient entities. We take a closer look at the hybrid nature of this model below. In addition to modeling the working of individual tools, it was also necessary to model the set of tools in the wets processing area in order to evaluate the effect of different scheduling policies. Thus each tool itself is also a resident entity in the model.

The wets tools may be subject to two types of down times, one at the chamber or tank level where each tank can be individually unavailable, and the other at the tool level where all tanks in the tool are unavailable. The former may be done in the case of tank dumps that was described earlier, the latter may be the result of an alarm that necessitates the shutting down of the tool. Our model does not take into account the chamber level down times but tool level down times were modeled. However, tool down time experiments will not be addressed here.

3.2 Verbal Event Graph

A verbal event graph is a representation of the simulation model (the event graph) where the state changes at the vertices, conditions that are checked along the edges and time delays along the edges are described using regular English phrases. A verbal event graph helps the modeler validate the model with the concerned parties by walking them through the model, describing the state changes, edge conditions and time delays. The basic building block of an event graph is shown in Figure 2. This graph is read as follows: When event *A* occurs, after the state changes at *A* have taken place, if condition *i* is true, schedule event *B* to occur after a time delay of *t*.

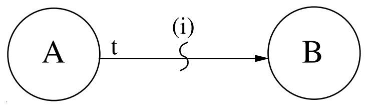

Figure 2: Building Block of an Event Graph

The SIGMA model developed has approximately 60 vertices and more than 100 edges. Due to the size of this model, we chose to construct a simplified verbal event graph of the model by logically combining the vertices and edges and came up with a verbal event graph that preserves the logical structure of the model. This is shown in Figure 3. We used this simplified verbal event graph to conduct a simulation walk-through with the process engineers.

3.3 Hybrid Modeling

A pure resident-entity based event scheduling simulation model models the resources in the system explicitly; state changes for these resources are made when an event occurs. For example, in the case of the wets tool, the tool is a resident entity and the queue at the tool (number of jobs in queue) is updated when a lot arrives at the tool or leaves the tool after processing. The queue is tracked by means of a variable that is stored in memory when the program runs. Since

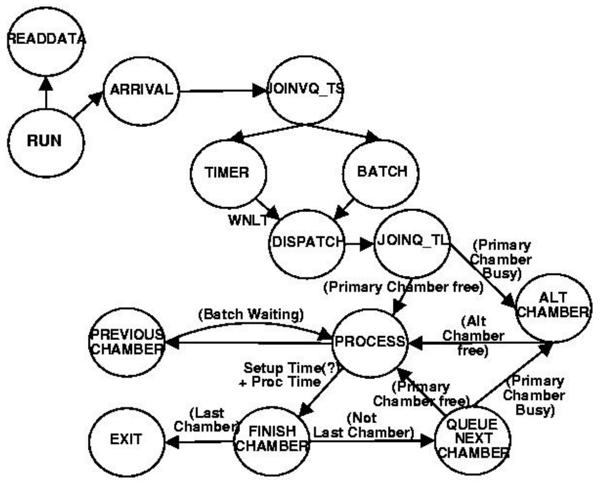

Figure 3: Simplified Verbal Event Graph

there are a limited number of tools and hence queues, the amount of space required to store this information is not very significant. However, by simply looking at the *number* of jobs in queue, it is not possible to determine what the characteristics of these jobs (lots) are. For example, lots are batched in the wets area before they are processed. It becomes necessary to know which lots make up each batch in order to track cycle time for each lot separately. Pure resident-entity simulation does not attempt to capture this information.

We adopt an approach that models resident entities in the same fashion as a pure resident-entity simulation, however, we capture the necessary information on the transient entities. This approach is illustrated here using an example of batching of lots. Lots are batched together before they start processing in the tanks. A batching identifier determines if two lots can be batched together. A batch (identified by a *batch number*) moves through various tanks in the tool, as specified by its operation, before exiting the tool. To keep track of the cycle time for each lot in the batch, an array for each batch identifier (at a tool) is created that stores the arrival time for each lot as it joins a *virtual queue* of its corresponding identifier. The unique "number" or position that a particular lot holds in its identifier queue is then stored. To keep track of which lots are in a batch, each batch number stores these unique numbers. Thus, at any given point in time, the cycle time of a lot in a batch can be determined by looking up the entry time corresponding to this unique number for this lot in its identifier queue.

The storage space required in this hybrid approach is more than that of a pure resident-entity approach and less than a transient-entity approach. However, preliminary results indicate that the additional storage space required is a small price to pay for the speed-up obtained by using the resident-entity approach combined with information tracking when necessary.

4 MODEL INPUTS AND VALIDATION

4.1 Model Inputs

The *deterministic inputs* to the model comprised of tool configuration data - number of tanks, qualification of each tank; operation data - specifies the order of tanks to be visited (chamber route) and the processing time in each tank; Tool Set parameters - e.g. Wait No Longer-than Time (WNLT) for batching and other operational data such as movement times of the main handler between tanks.

The *stochastic inputs* to the model comprised mainly of the arrival stream of lots to the wets area. Since the fab was in a ramp mode and not up to full production at the time of this study, it was not possible to collect real data from the fab and employ one of the standard methods for input modeling (Law and Kelton 1991). However, IBM Microelectronics has developed a full fab model for the 300mm fab using a customized version of AutoSched AP (ASAP) from Brooks Automation. The inputs to this model are the projected wafer starts for the fab once it enters production mode. The arrival streams to the various tool sets in the wets processing area from this model were used to derive input streams for the wets model in SIGMA.

We expatiate on the process of generating input arrival streams from the ASAP model as the results of a simulation are only as good as the inputs that drive the simulation. From the ASAP full fab model, data was collected on the arrivals to the wets area - the time that an arrival occurred, the operation that was to be performed on the arrival and the tool set that it needed. Using this information, the probability that a new arrival to the wets area needs a particular operation was computed for each of the operations performed in the wets area. This probability was used to randomly assign operations to lots as they arrived.

To generate inter-arrival times for the model, the time between arrivals to the wets area from the ASAP model was used to create an empirical distribution function (EDF) for the inter-arrival time. This EDF was randomly sampled to generate inter-arrival times in the simulation model. We note that before settling on an EDF based input stream, we investigated the performance of the model for some other input modeling techniques discussed in (Barton, Cheng, Chick, Henderson, Law, Leemis, Schmeiser, Schruben, and Wilson 2002).

4.2 Validation

Due to the lack of real representative data from the fab (since the fab was not in steady state production mode), it was not possible to validate the simulation model using

Table 1: Factors and Factor Levels

Factor/Level	-1	1
SCHED	Q and Server at 1^{st} Chamber	Total in Tool
WNLT	10 mins	20 mins
MaxB	3	Max Possible

real data. Hence we chose to validate the SIGMA model with output obtained from the ASAP simulation model.

The metrics used for validating the model were the average cycle time at a tool (averaged over all the lots that were processed at the tool) and the utilization of the tool. Though it would have been useful to breakdown the average cycle time by process or product at each wets tool set, this was not done as this information was not readily available from the ASAP simulation model. The ASAP model reports the average cycle times and utilizations for the various tool sets. This output was compared with the output from the wets simulation model in SIGMA. It is important to note that the ASAP model used a particular scheduling policy for the wets area, which was one of the policies that we experimented with. Using this scheduling policy for the wets tools in the SIGMA model, the average cycle times at the tool sets and the utilizations matched the output from the ASAP model very closely.

As part of the process of validating the model, simulation walk-throughs were done with the wets process engineers. Feedback from these walk-throughs was used to refine the model.

It is also worth mentioning at this point that due to the advanced capabilities of these tools, the software on the tool can generate a *tool log* that has a record of every action that the tool performs on a lot, from the time the lot is loaded onto the load port until it leaves the tool. The tool has an operating system and the tool software can dump the tool log onto a file that can be exported to a computer.

5 EXPERIMENTS

5.1 Factors Varied

The primary aim of the experiments conducted was to evaluate the effect of scheduling policies and tool parameters on the average cycle time at each tool set and the utilization of the tools. The factors that were varied and the levels of each factor are as shown in Table 1.

SCHED refers to the scheduling policy that was used. Two levels were investigated - Level -1 corresponds to a policy that checks each of the possible tools that the lot can be processed at, to ensure that the first chamber that a new batch (created from a lot arrival that just occurred) needs to visit is free; if the chamber is not free, then it also checks the number of batches in the virtual queue for that chamber. Based on the checks, the lot is sent either to the tool which has the first chamber free, or if the first chamber in all tools is busy, then to the tool which has the lowest number of queued lots in the virtual queue for the chamber. Level 1 corresponds to a policy that simply checks the total number of lots at each of the possible tools and sends the lot to the tool with the minimum number of lots.

WNLT stands for Wait No Longer-than Time, which is the maximum time that a lot is allowed to wait at a tool set for a batch to be formed. Once the lot has waited for a time equal to WNLT, this lot (and any other lot that arrived during this interval) is scheduled, as an incomplete batch, on a tool that belongs to the tool set and starts processing. If enough lots arrive at the tool set to form a batch before the WNLT timer expires, the full batch is scheduled on a tool for processing. Level -1 for this factor corresponds to a value of 10 minutes and Level 1 corresponds to a value of 20 minutes.

MaxB is a parameter that specifies the maximum number of batches that are allowed in a tool at any given time. Due to concerns that a quality problem could potentially result in the loss of several batches of wafers, a rule is put in place that sets this parameter. Level -1 corresponds to a maximum of three batches in a tool and Level 1 allows the tool to hold as many as is physically possible.

5.2 Performance Measure

The performance measure used as a response in the experiments was the average cycle time of a lot at each of the tool sets. Since a tool set may have multiple copies of the same tool, the average across the tools in a tool set was computed by weighting the average cycle time at each tool by the number of lots that passed through the tool.

We would like to point out that an average cycle time measure for a tool set effectively combines several dependent responses (average cycle time at each tool in a tool set) into a single response. The individual cycle times are correlated since the average cycle time at each of the tools within a tool set depends on the scheduling policy that routes lots to the tools within a tool set. In the case of tool sets that are backups for other tool sets for some operations, the average cycle times at these tool sets will be correlated and hence the usual design of experiments will not apply.

5.3 Experiment Design

A 2-level full factorial design in three factors was chosen since the short execution time of the simulation did not necessitate a fractional design. Thus, there were a total of eight design points. Each design point was replicated five times resulting in a total of $8 \times 5 = 40$ simulation runs. 95% confidence intervals were computed for each of the responses. The analysis of the designed experiment was done in SAS.

We would like to briefly mention some of the characteristics and advantages of model development in SIGMA. The model is developed graphically and then converted to C code by SIGMA. The model was run for a simulated time of 160 days and a warmup analysis was done to ensure that any initialization bias present was eliminated. The execution time for a single run of the model varied between 30 and 60 seconds, depending on the the scheduling policy and parameter settings. SIGMA allows the specification of an experiment file that contains the settings for each of the factors for the runs in the experiment. Due to the fast execution and automated experiment setup, running a design of experiments (DOE) analysis is quick and easy.

6 RESULTS

To understand the effects of the factors on cycle time, plots of the main effects and interactions were analyzed. These plots were generated using SAS.

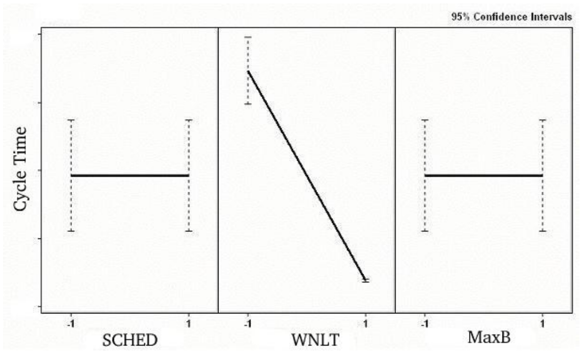

Figure 4: Effect of Factors on Cycle Time: Tool Set 1

Figure 4 shows the effects of the factors on the cycle time of Tool Set 1. To maintain confidentiality, the values of the ordinates have been removed from this plot. As seen, there is a significant impact (seen by the difference in cycle time values between the two levels) of WNLT and no impact of SCHED or MaxB on cycle time. This tool set has certain characteristics that made it different from the rest - it has only one tool and hence SCHED has no effect on cycle time; it also has only one tank within the tool and hence MaxB has no effect on the cycle time.

Figure 5 shows the effect of the factors on Tool Set 2. This is a more typical tool set. There is a significant effect of SCHED and a moderate effect of WNLT and MaxB on the cycle time. This tool set has multiple copies of tools and has several tanks in each tool. Hence the scheduling policy and the maximum number of batches allowed in the tool also affect the cycle time.

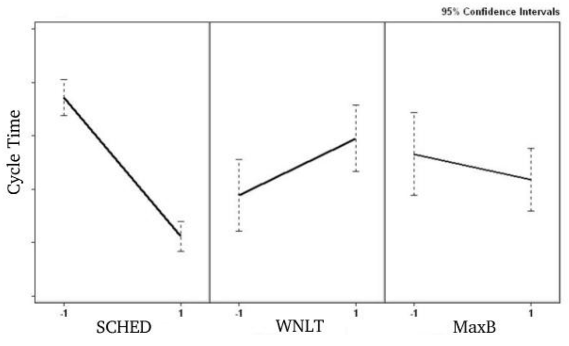

Figure 5: Effect of Factors on Cycle Time: Tool Set 2

Similar plots were analyzed for the other tool sets that were modeled in the wets area. These plots provide a clear picture of the policy and parameter settings that should be used at the various tool sets.

7 CONCLUSIONS

In this pilot simulation study, a resident-entity based event scheduling simulation model was used to analyze the effects of some of the factors that affect cycle time in the wets processing area. Using designed experiments, meaningful results were obtained that helped in understanding the impact of these factors. The wets processing area is a part of the fab that has received little attention in terms of models that analyze the performance of the area. The wets tool is a batch chamber tool and very few published works exist that look at modeling these tools using simulation. We have attempted to provide a concise description of the process area and the working of the tools in this area. In addition, a resident-entity based hybrid modeling approach has been introduced and some advantages of this approach have been highlighted. In this study, we have analyzed the effects of a small fraction of factors that affect the performance of the wets area; there are several other factors that warrant analysis similar to this, using simulation, and we hope there will be more work in this area in the future.

ACKNOWLEDGMENTS

We would like to thank Chaunese Turner at IBM Microelectronics for helping with the simulation inputs and whose valuable expertise in the wets area helped us gain a better understanding of the process. We are grateful to Sameer Shikalgar at IBM Microelectronics and Ed MacNair at IBM Research for help with the modeling and data collection efforts, especially with providing inputs from the full fab model. We also thank the wets process engineers, Russ Arndt and Hong Lin, and wets equipment engineer, Richard Hyatt, for their full co-operation and support. We thank Professor Russell Barton and Professor Deborah Medeiros at the Pennsylvania State University and Aseem Joshi at IBM Microelectronics for their helpful comments during several discussions related to this project. Finally, we would like to express our deepest gratitude to Professor Lee Schruben at the University of California at Berkeley, who has been a constant source of encouragement and guidance, and provided us with invaluable help on various modeling issues.

REFERENCES

Barton, R. R., R. C. H. Cheng, S. E. Chick, S. G. Henderson, A. M. Law, L. M. Leemis, B. W. Schmeiser, L. W. Schruben, and J. R. Wilson. 2002. Panel on current issues in simulation input modeling. In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yucesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, 353–369. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Davison, A., and D. Hinkley. 1997. *Bootstrap methods and their application*. New York, NY: Cambridge University Press.

Dümmler, M. A. 1999. Using simulation and genetic algorithms to improve cluster tool performance. In *Proceedings of the 1999 Winter Simulation Conference*, ed. P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, 875–879. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Koehler, E. J., T. M. Wulf, A. C. Bruska, and M. S. Seppanen. 1999. Evaluation of cluster tool throughput for thin film head production. In *Proceedings of the 1999 Winter Simulation Conference*, ed. P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, 714–719. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Kunesh, R. F., J. Mauer, C. Gow, M. Liehr, and S. L. Cohen. 1994, Feb. Evaluating multifunction baths in semiconductor wet-processing equipment. *Microcontamination* 12 (2): 15–20.

Law, A. M., and D. W. Kelton. 1991. *Simulation modeling and analysis*. 2nd ed. New York: McGraw Hill.

LeBaron, T. H., and R. A. Hendrickson. 2000. Using emulation to validate a cluster tool simulation model. In *Proceedings of the 2000 Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 1417–1422. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

LeBaron, T. H., and M. Pool. 1994. The simulation of cluster tools: a new semiconductor manufacturing technology. In *Proceedings of the 1994 Winter Simulation Conference*, ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila, 907–912. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Mauer, J. L., and R. E. A. Schelasin. 1993. The simulation of integrated tool performance in semiconductor manufacturing. In *Proceedings of the 1993 Winter Simulation Conference*, ed. G. W. Evans, M. Mollaghasemi, E. Russell, and W. E. Biles, 814–818. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Montgomery, D. C. 2001. *Design of experiments*. 5th ed. New York: John Wiley.

Roeder, T. M., S. A. Fischbein, M. Janakiram, and L. W. Schruben. 2002. Resource-driven and job-driven simulations. In *Proceedings of the 2002 International Conference on Modeling and Analysis of Semiconductor Manufacturing*, 78–83.

SAS Institute. *SAS OnlineDoc, Version Eight*. SAS Institute.

Schruben, D., and L. W. Schruben. 2001. *Graphical simulation modeling using SIGMA*. 4th ed. Berkeley, California: Custom Simulations.

Schruben, L. W. 1983. Simulation modeling with event graphs. *Communications of the ACM* 26 (11): 957–963.

Schruben, L. W. 1990. Simulation graphical modeling and analysis (SIGMA) tutorial. In *Proceedings of the 1990 Winter Simulation Conference*, ed. O. Balci, R. P. Sadowski, and R. E. Nance, 158–161. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Schruben, L. W. 1991. SIGMA tutorial. In *Proceedings of the 1991 Winter Simulation Conference*, ed. B. L. Nelson, W. D. Kelton, and G. M. Clark, 95–100. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Schruben, L. W., and T. M. Roeder. 2003. Fast simulations of large-scale highly-congested systems. *Simulation Transactions* (in press).

AUTHOR BIOGRAPHIES

NIRMAL GOVIND is a Ph.D. candidate in Industrial Engineering and Operations Research at the Pennsylvania State University, University Park. He holds an M.S. in Industrial Engineering and Operations Research from the University of California at Berkeley and a B.Tech in Mechanical Engineering from the Indian Institute of Technology, Madras. He can be reached at `<nirmalg@psu.edu>` and his web address is `<http://www.personal.psu.edu/nug107>`.

DAVID FRONCKOWIAK is the manager of the 300mm Industrial Engineering department at IBM Microelectronics in East Fishkill, NY. He has worked in the area of semiconductor process modeling with emphasis on fab productivity improvement for 10 years. He holds an M.S. in Industrial Engineering from the University of Arizona and an MBA from Marist College. He can be reached at `<fronckow@us.ibm.com>`.