Proceedings of the 2013 Winter Simulation Conference
R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, eds

PREDICTION OF PRODUCT LAYER CYCLE TIME USING DATA MINING

Michael Hassoun

Industrial Engineering and Management
Ariel University
Ariel, ISRAEL

ABSTRACT

Based on a simulated non volatile memory (NVM) fab, we show that forecasting the steady state cycle time of process segments is possible using certain segment characteristics. We also show that the cycle time predictability is highly dependent on the choice of the segmentation, with the more efficient segmentation corresponding to the product layers.

1 INTRODUCTION

Cycle time (CT) is certainly one of the main performance measures in manufacturing and maybe the most important one for the ever changing memory market.

The rapid decrease in market value and the relative absence of brand fidelity set the rules of the game: the first manufacturer to hit the market with a device higher in capacity, smaller in size (thus cheaper) and with the ability to respond quickly to the sharp changes in demand will win the round.

Numerous researchers have studied CT, focusing on its causes and methods for reducing it. To that end, they have proposed optimal or heuristic methods, tackled the CT reduction of the whole fab, or aimed at local toolset optimization. In an effort to simplify the models in use, Rose (1998) suggested that the most important characteristic of the semiconductor fabrication plant remains the reentrance flow of the wafers among the work stations. He studied fab behavior on an extremely simplified structure comprising a fully modeled bottleneck station and a time delay. In other research, Rose (1999), and later Johnson et al. (2005), showed that although CT prediction needed to be improved, this type of simplification is highly relevant to both the researcher and the practitioner.

Aggregating operations to simplify analysis of the extremely complex semiconductor manufacturing environment is common practice (Rose 1998). Nonetheless, to our knowledge, no formal study has been conducted on the best way to aggregate operations. In this paper, we not only show that the CT of a carefully chosen segment of sequential processing steps is predictable based on the segment's characteristics, we also bring to the reader's attention the fact that the best way of segmenting the process seems to be by following the re-entrant loops corresponding to the product layer. We address the question of the CT predictability by using the NVM SEMATECH fab simulation benchmark, which is described in the next section. The data produced by the simulation is analyzed through data mining techniques in Section III. The simulation model and some parts of the experimentation framework have already been described in a former publication (Hassoun et al. 2010), but for the sake of clarity, we review both in their integrality in the next section.

1.1 Sematech Dataset 1 NVM Fab Characteristics

The sematech dataset 1 is one of six standard models aimed at mimicking real fab behavior that have been broadly used as research benchmarks (Palmeri et al. 1997; Hunter et al. 2002; Iwata et al. 2003; Dai et al. 2002). This model describes plant structure and operation in great detail, and numerous characteristics of the true fab level of complexity are expressed.

The NVM fab model is characterized by two high volume products (defined by a process route) produced on 68 toolsets (groups of identical tools). The total number of tools in the plant is 211. The number of processing steps needed to complete Product 1 and Product 2 are 210 and 245, respectively. The operations are characterized by processing batch definitions (wafer, lot, lot batch), post process cooling time, sequence dependent setups for two Implant tools, etc. Rework and in-line scrap are also modeled, at both the wafer and the lot levels (some lots are fully reworked/scrapped, others are partially reworked/scrapped). The lot population, together with the average lot size, is therefore slightly decreasing along the process. In addition, lots containing a very small number of wafers (most of the time a single wafer) appear and disappear following short pre-defined rework loops. The release rate of the 48-wafer lots is constant and stands at about one lot every three hours for Product 1 and one lot every six hours for Product 2, thus leading to a total of 4,000 wafers per week. Both the mean time between failures (MTBF) and the mean time to repair (MTTR) have exponential distributions, while processing, cooling, setup and moving times are constant.

Human operators are modeled, and their ability to run specific operations is differentiated. We count a total of 83 operators grouped in 28 operator types. Depending on the operation, the operator is needed for lot loading, unloading, all or part of the process, and the lot transport.

Based on all these characteristics, we discovered that this model is unstable (overstressed) and that the CT and WIP continue to grow with time. We therefore adjusted it in two ways: We set the release rate at 90% of the SEMATECH definition (3,600 wafers per week) and increased the head count of operator number 7 from 1 to 2. Under these new conditions the model showed converging steady state behavior while it was still close to being fully utilized (such that any increase in the release rate would destabilize it).

1.2 Data Structure

In our experiment, a single observation is obtained for each segment of operations (process steps) under a certain scenario. The structure of the observation data is described here. Each segment's characteristics were constructed from its operations' variables. Some of the operation variables describe the operation in itself; others are related to the toolset on which the operation is processed. The operation variables consist of parameters set prior to the simulation run and of the performance measures that result from the simulation run. Most of the variables directly related to the operation describe its position in the line (distance from last bottleneck, next operation of the same product on the station, etc.). Most performance measures (availability, utilization, mean and standard deviation of the number of down times, etc.) are variables related to the station that runs the operation. One remarkable exception is the lot Inter-Arrival Time to the operation, which is related to the WIP flow and not directly to the station performance. We also defined any station having a load above 90% as a bottleneck.

In a second computation phase we processed the operation vectors to obtain the segment's characteristic vector (Table 1). Some of the variables are obvious metrics and directly describe the segment (availability, number of bottlenecks, etc.). Others are based on more evolved parameters, like the utilization over availability ratio (U/A), a measure of performance commonly used in fabs. At the end of the process, each segment, the length of which depends on the segmentation chosen, is characterized by a vector of 31 features.

Table1: Segment descriptors.

Descriptor	Variables	Remarks
Length		Number of steps in segment
Availability	Avg, minimum, average and max of Coef. of Variation over the segment	
Level of tool sharing	Avg of number of operations on a tool, min and max of the number of steps on the same tool.	
Special regime operations	Number of batch operations and maximum batch size in the segment. Number of operations being the start, middle or end of a rework loop	
Location	Distance from the beginning of the process, from the end of the process	Measured in number of steps, at the first stop of the segment
Load and BN operations	Average and maximum loads in the segment, Number of BN's, minimum number of tools in a BN station	BN defined as tools loaded above 90%
Utilization	Avg and stdv among operations in segment, avg and max of U/A.	
Number of tools in the segment		
Number of tool breakages	Average and max of tool breakage in the segment	
Stdv of inter-arrival time	Average, maximum	
Process time	Average	
Scheduling method		FCFS or LBA
Cycle time	Avg	

1.3 Experimental Design

After setting the simulation framework and its data processing infrastructure, we contemplated the type of experiment that would be the most suitable to achieve our goal, i.e., to correlate CT with various segment descriptors under a range of situations. We clearly needed to generate a large number of observations that were sufficiently different from one another. On the other hand, we decided, a priori, to discard any data from a simulation that would not stabilize. We therefore wanted to maintain the highest possible level of control and reduce the chances of creating an exploding WIP situation. With these two contradictory ideas in mind, we created a total of 400 scenarios, presented in Table 2, under two scheduling regimes: 200 under FCFS (First Come First Served) and 200 under LBA (Lowest Buffer Ahead). We randomly altered the basic settings supplied by SEMATECH (after our changes as described earlier) at two levels: First, we changed the product mix for each experiment. The original figures were 2400 and 1200 wafers per week for Products 1 and 2, respectively. For each experiment, the release rate of Product 1 was randomly chosen from a uniform distribution between 2000 and 2600. A total release rate of 3600 wafers per week was maintained, and the release rate for Product 2 was set accordingly.

Table 2: 400 simulation scenarios based on modulation of MIMAC values.

	Total: 400 Scenarios			
Mix	Product 1 from U[2000; 2600] Prod.2 to complete to 3600 total wafer starts			
Scheduling	FCFS - 200 scenarios		LBA - 200 scenarios	
Availability	Avail kept cst. Both MTTR and MTBF are multiplied by a factor from U[0.5; 2]	MTTR set at original MIMAC value, MTBF multiplied by a factor from U[0.7; 1.05] for BN and from U[0.8; 2] for non BN stations	Avail kept cst. Both MTTR and MTBF are multiplied by a factor from U[0.5; 2]	MTTR set at original MIMAC value, MTBF multiplied by a factor from U[0.7; 1.05] for BN and from U[0.8; 2] for non BN stations

In parallel, to change the operational stress on each station individually, we altered station availability definitions in two ways: For the first half of the scenarios, we multiplied both the MTBF and the MTTR of each station by the same random factor taken from a uniform distribution in a range of [0.5; 2]. In these scenarios, the resulting availability was not changed, and we acted only on the variance of the availability. In the second half of the scenarios, the overall availability was changed. The MTTR was set at its original value, and the MTBF was multiplied by a factor chosen randomly from a uniform distribution. The distribution ranges were [0.7; 1.05] for bottleneck and [0.8; 1.2] for non-bottleneck stations under the SEMATECH basic scenario. This differentiation was made to minimize the chances of creating an unstable scenario. At this stage, we conducted a single simulation run for each of the 400 scenarios and examined their stabilizations. We found that 34 of the FCFS scenarios and 39 of the LBA scenarios diverged, and hence, we discarded their data. The remaining usable raw data formed a list of 148,785 observation vectors (455 operations over two products × 327 valid scenarios), each of which comprised 32 variables (scheduling method was added to the basic vector).

2 PREDICTION OF CT AND PROCESS SEGMENTATION

2.1 Prediction Models and their Evaluation

We used commercial, off-the-shelf data mining software to explore the database. We addressed two main questions about CT predictability:

- Can one predict the steady state CT of a segment of operations corresponding to the product layer?
- What is the line segmentation method that allows for the best CT prediction?

We avoided "black box" type models, and used the regression tree technique and the classification and regression tree (CART) algorithm to generate the trees (Han et al. 2012). First, the data were randomly divided into a training set (70% of the tuples – each representing one operation under one scenario), on which the regression tree was built with the segment CT as the predicted variable.
CART builds a binary tree, and since the target (CT) is continuous, it uses the least squared deviation (LSD) impurity measure for branching. With $CT(i)$ and $\overline{CT}(t)$ denoting, respectively, the CT of tuple *i* and the mean CT of node *t* population, the LSD at *t* is simply the within-node variance and is given by:

$$R(t) = \frac{\sum_{i \in t} \left(CT(i) - \overline{CT}(t) \right)^2}{N(t)}$$

The algorithm recursively chooses the next split as the one that maximizes the value of $R(t) - p_L R(t_L) - p_R R(t_R)$ where t_L and t_R are the left and right nodes generated at *t* by the split, and p_L and p_R the proportion of tuples from *t* in these nodes, respectively.

Once built, the prediction model was applied to the test set (the 30% remaining vectors), and its performance was analyzed. We repeated this process several times to limit the risk of a biased separation between the training and test sets, which could potentially affect model performance. The performance criterion of the model is its explained variance, which was computed as follows:
The prediction error for one instance was defined as the gap between the CT predicted by the model and the actual CT of the segment:

$$err(i) = \overline{CT}(i) - CT(i)$$

where *i* is tuple index in the test set. Model performance can be evaluated by the mean square of its prediction errors. We can compare this value to the variance of the test set to get the proportion of unexplained variance.
The proportion of variance explained by the model is thus:

$$Explained\,Variance = 1 - \frac{\sum_{i=1}^{n} err(i)^2 \Big/ n-1}{Variance(test\,set)}$$

where *n* is the number of observations in the test set.

2.2 Layer CT Predictability

We begin by presenting the performance of models obtained by the CART algorithm with the layer CT as a target.

The first step in this analysis was to segment the process following the re-entrance loops. We defined the "Develop" step, which ends every Litho sequence in the process (Vapor prime; Coat; Expose; Develop), as the end of the layer. Any definitions based on the Litho operations could have been acceptable.

After this aggregation phase, we had 10,464 tuples, each of which represented a specific layer in a certain scenario (32 layers × 327 valid scenarios). This array of 10,464 tuples by 32 descriptors (Table 1) represents the raw data for the CART modeling. At this stage we repeated the random split 10 times to obtain 1) a training set on which the prediction model was built, and 2) a test set on which model performance was measured. The resulting explained variance ranged from 84% to 88%, and its average was

86.1%. We thus conclude that the variables describing the segment contain enough information to predict the layer's CT with reasonable precision.

2.3 Segment CT Predictability

Next, we wanted to compare the results obtained thus far for layers to the CT predictability for operation segments of different lengths. We repeatedly segmented the process in fixed length groups of operations beginning with the first operation. We then discarded the last segments, whose lengths typically did not fit the length definition for the same experiment, and conducted five replications of the model building procedure as described in section 1. We tested the ability to predict the CT of individual operations. Then incrementally larger segmentations were tested, from 2 to 40 operations in increments of two, and from 45 to 100 operations in increments of five (setting the segment length at 100 operations yielded only two usable segments). Finally, we tried our prediction method on the whole process.

The performances of the models obtained (Figure 1) prompt a few observations. First, the best performance in this experiment is still lower than the average performance in the layer experiment presented in the previous section. Second, CT value prediction for short and long segments is more difficult than for segments of intermediate length. Prediction models for the CT of a single operation score about 40% explained variance, and those for the whole process score about 22% explained variance. In the framework of this study, we can only postulate on the reasons for such behavior: The shorter segment's CT, which exhibited greater change under different scenarios, is more difficult to predict. On the other hand, as the aggregation progresses and adds more operations to a segment, part of the difference between segments is lost. To illustrate this phenomenon, we consider the important variable "maximum load in the segment." As the segments become longer, more of them include the same high load tool, and therefore, they are characterized by the same value for this variable. This progressive loss of differentiation as the segments get longer makes it harder for the CART algorithm to build a prediction model based on this variable.

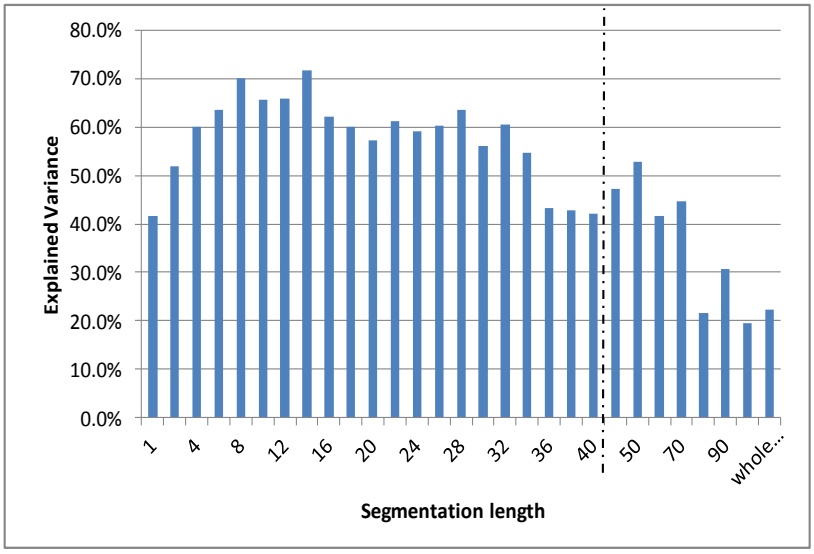

Figure 1: Prediction performances for fix length segments.

Finally, we remark that there is a plateau of higher values at intermediate segment lengths, peaking with models based on fourteen operations, an observation that may or may not be related to this length value being the closest to the average layer length.

As we see from this analysis, the mere agglomerations of operations in segments do not fully explain the high levels of layer CT predictability.

3 CONCLUSION

The results of the experimentation conducted here are twofold: first we assess the ability, in theory, to predict CT in fabs through the use of off-the-shelf data mining software. Second, but no less important, we show that the operation segments corresponding to the product layers seem to behave following more predictable dynamics than under other segmentation scenarios. These ideas can be easily implemented in real fabs, where more accurate CT modeling and control will ultimately improve CT, which is among the most important goals conducted by Industrial Engineers.

REFERENCES

Dai, J. G. and S. Neuroth. 2002. "DPPS Scheduling Policies in Semiconductor Wafer Fabs." In *Proceedings of the 2002 International Conference on Modeling and Analysis of Semiconductor Manufacturing*, edited by G. T. Mackulak, J.W. Fowler and A. Schomig, 194-199.

Han, J., M. Kamber, and J. Pei. 2012. *Data Mining: Concepts and Techniques*. 3rd ed. Waltham, MA: Morgan Kaufmann Publishers.

Hassoun, M. and G. Rabinowitz. 2010. "Hunting Down the Bubble Makers in Fabs." *IEEE Transaction on Semiconductor Manufacturing* 23:13-20.

Hunter, J., D. Delp, D. Collins, and J. Si. 2002. "Understanding a Semiconductor Process Using a Full-Scale Model." *IEEE Transactions on Semiconductor Manufacturing* 15: 285-289.

Iwata, Y., K. Taji, and H. Tamura. 2003. "Multi-Objective Capacity Planning for Agile Semiconductor Manufacturing." *Production Planning and Control* 14: 244-254.

Johnson R. T., J. W. Fowler and G. T. Mackulak. 2005. "A Discrete Event Simulation Model Simplification Technique." In *Proceedings of the 2005 Winter Simulation Conference*, edited by N. Steiger and M. E. Kuhl, 2172-2176. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Palmeri, V. and D. W. Collins. 1997. "An Analysis of the "K-Step Ahead" Minimum Inventory Variability Policy® Using SEMATECH Semiconductor Manufacturing Data in a Discrete-Event Simulation Model." In *Proceedings of the 6th International Conference on Emerging Technologies and Factory Automation*, 520-527. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Rose, O. 1998. "WIP Evolution of a Semiconductor Factory After a Bottleneck Workcenter Breakdown." In *Proceedings of the 1998 Winter Simulation Conference*, edited by D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, 97-103. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Rose O. 1999. "Estimation of the Cycle Time Distribution of a Wafer Fab by a Simple Simulation Model." In *Proceedings of the International Conference on Semiconductor Manufacturing Operational Modeling and Simulation*. 133-138.Saltelli, A. 2002. "Sensivity Analysis for Importance Assessment." *Risk Analysis* 22: 579-590.

AUTHOR BIOGRAPHY

MICHAEL HASSOUN is a lecturer in the Industrial Engineering Department at the Ariel University, Israel. He earned his PhD and MSc in Industrial Engineering from Ben-Gurion University of the Negev, Israel, and his BSc in Mechanical Engineering from the Technion, Israel. He worked at Intel fab 8 as a functional area industrial engineer. His email address is michaelh@ariel.ac.il.