A SIMULATION-BASED ANALYSIS OF THE CYCLE TIME OF CLUSTER TOOLS IN SEMICONDUCTOR MANUFACTURING

Heiko Niedermayer
Institute of Computer Science
University of Tübingen
D-72076 Tübingen, Germany
E-mail: niederma@informatik.uni-tuebingen.de

Oliver Rose
Institute of Computer Science
University of Würzburg
D-97074 Würzburg, Germany
E-mail: rose@informatik.uni-wuerzburg.de

KEYWORDS
simulation, manufacturing, semiconductor, cluster tools

ABSTRACT

Cluster tools are widely used in modern semiconductor manufacturing facilities. In parallel mode they offer high throughput at the cost of a complex behaviour with regard to lot cycle times. The reason is that cluster tools are behave like small factories themselves. We analyze the slow-down of the processing of a lot that is caused by other lots in the tool and examine how the slow-down factor can be used for scheduling and for predicting lot cycle times. This cycle time analysis is mandatory for production planning and can only be done by simulation so far.

INTRODUCTION

Since the middle of the 1990s cluster tools are becoming more and more important in semiconductor manufacturing. The most recent manufacturing facilities consist almost exclusively of cluster tools. Cluster tools are machines that combine several processing steps in one machine. They can be regarded as small factories inside a factory. They consist of loadlocks, processing chambers, and handlers. Figure 1 shows the structure of a simple cluster tool with 2 loadlocks, 5 chambers, and 1 handler.

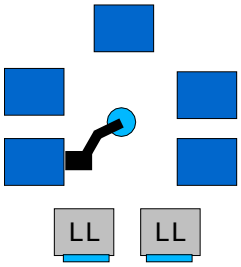

Figure 1: A Simple Cluster Tool Model

Each loadlock can be loaded with one lot. A lot is a box with wafers, e.g., 25 wafers. Then, the tool processes the lot. Most modern cluster tools have 2 loadlocks. Each wafer of the lot is scheduled inside the cluster tool by the scheduler of the tool. The handlers are used for moving the wafers between the chambers and loadlocks. The chambers are machines to process wafers.

One advantage of clustering processing steps is that the processing of the wafers is pipelined. This reduces the cycle time of the lot as only the processing time at the bottleneck of the steps limits the cycle time and not the sum of the raw processing times of all steps.

An additional advantage is that cluster tools save clean-room space. Inside the cluster tool there is vacuum, hence, a low number of particles. As a consequence, the clean-room quality outside the tool can be lower than in traditional fabs.

A disadvantage of cluster tools is that their behavior is more complex than the behavior of simpler machines. The cycle time of a lot is not constant but depends on the situation inside the cluster tool during the processing of the lot. This is due to the fact that in parallel mode cluster tools are able to process lots in parallel that share the same resources.
When a machine processes only one lot the cycle time is simply determined by a constant or by a single random variable. When a cluster tool processes lots in parallel this is more complex. Each lot overlaps with other lots. During the overlaps the lots share the same resources and the lot cycle time depends considerably on the lot combinations inside the cluster tool. The shared use of resources slows down the processing of the lots. As Figure 2 illustrates comparing case A (single mode) and case B (overlap) the gain in makespan is less than the change in start time: $\Delta c \leq \Delta d$.

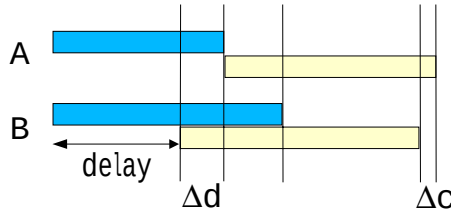

Figure 2: Overlap Scenario

Despite of the fact that the processing of each lot is slowed down the overall throughput and the utilization of the expensive machines inside the cluster tool are

Proceedings 15th European Simulation Symposium
Alexander Verbraeck, Vlatka Hlupic (Eds.)
(c) SCS European Council / SCS Europe BVBA, 2003
ISBN 3-936150-28-1 (book) / 3-936150-29-X (CD)

higher than in single mode where only one lot is processed at a time.

Considering the different overlaps and the different lot types, the number of possible situations is huge. The overlap size can range from almost 0 seconds to the complete cycle time of a lot, say, 4000 seconds. Figure 3 illustrates that the cycle time of a particular lot depends considerably on its amount of overlap with the lot that is produced in parallel on a cluster tool.

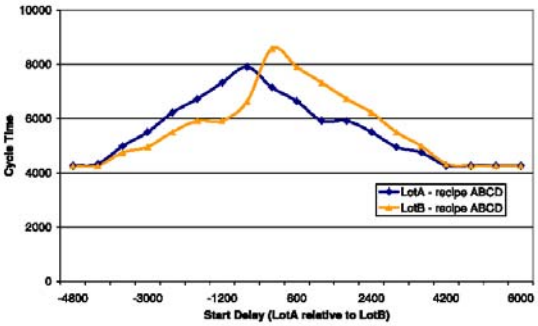

Figure 3: Cycle Time and Overlap

Additional factors are the number of recipe combinations and the different lot sizes. Thus, the cycle time cannot be computed in advance. Instead the complete scenario has to be simulated to determine the lot cycle times. At the moment simulation is the only approach to determine the performance of cluster tools in a detailed manner.

If we need to evaluate a schedule for a cluster tool we have to simulate the schedule to determine the lot cycle times, lot completion times, and the makespan.
For simulation, we use the cluster tool simulator CluSim that was developed at the Department of Computer Science at the University of Würzburg by Mathias Dümmler and a number of students (Dümmler 1999; Bohr 1999; Schmid 1999). Dümmler proposed a genetic algorithm for optimization computing the fitness for each schedule with the simulator. CluSim is also used at Infineon Technologies for cluster tool optimization.

When we use simulation for computing the lot cycle time this takes much more CPU time than simply taking a fixed cycle time from a data set (maybe with a setup taken from a setup matrix). Therefore, search approaches that test various schedules are more expensive for cluster tools than for other machines.

In this paper, we analyze how lots that are processed in parallel slow down each other. In the next Section we introduce the slow-down factor and some of its properties. Then we show that to a large degree the slow-down factor can be explained by a change of bottlenecks. Finally, we use the slow-down factor for approximation and scheduling.

RELATED WORK

A lot of cluster tool research focused on the scheduling inside the cluster tool and on simulation. Analytic performance analysis was done by (Perkinson et al. 1994). They analyzed cluster tools with one loadlock and no parallel chambers, identical deterministic transport and process times. Developers of simulation software still use these Perkinson models for evaluating the correctness of their simulator. Later Perkinson et al. extended their model allowing for, e.g., redundant chambers (Perkinson et al. 1996). There are also approaches using petri nets, for instance, for single mode cluster tools (Srinivasa 1998).

Simulation for analyzing cluster tool performance was used in (Atherton et al. 1990) and (Koehler et al 1999). Both papers show that simulation is mandatory for accurate prediction of performance estimates like cycle times or chamber utilitzations.

A detailed introduction to cluster tools can be found in (Atherton and Atherton 1995).

Considering large fab scheduling problems efficient methods are needed to schedule facilities with cluster tools. Our approach of simulating or measuring slow-down factors and use them for scheduling or for cycle time prediction can save a lot of time during optimization.

SLOW-DOWN FACTORS FOR LOTS

To study the effects of overlaps we introduce the slow-down factor. Lot B has an influence on lot A and usually the processing of lot A will take more time than without lot B (Figure 4).

Figure 4: Lot A Is Slowed Down By Lot B

Definition 1: Slow-down Factor
The slow-down factor of lot A while processed in parallel with lot B is defined as

$$SDF(A, A+B) = \frac{Cycletime(A, A+B)}{Cycletime(A)} \quad (1)$$

where *Cycletime(A,A+B)* is the cycle time of lot A when it is processed together with lot B and *Cycletime(A)* is the cycle time of lot A when it is processed alone (single mode).

The slow-down factor is a measure for how much lot B disturbs lot A. We can use this information both for scheduling and for approximating the lot cycle times.
For the rest of the paper, we only consider cluster tools with 2 loadlocks because of their importance in manufacturing. We created 4 test sets with 12 recipes each. Most recipes of the sets "Dresden" and "Dresden fast handler" were inspired by descriptions of etch centers at Infineon Technologies' Dresden factory. The set "Villach" is based on descriptions of Endura cluster tools at Infineon Technologies' Villach factory (Seidel 2001). The last set "Simple" contains recipes where a single chamber is the bottleneck.

Slow-down factors and start delays

Considering the overlap of two lots we have to deal with the influence of different delays between the start times of the two lots. Processing the wafers is a cyclic process. The initial delay between the first and the second lot determines when the wafers of the second lot start to disturb the wafers of the first lot in this cyclic process. This may lead to different solutions for the internal schedule and therefore to different slow-down factors.
The figures show that depending on the lot combination the slow-down factor can be almost independent of the delay as well as highly variable for different delays. In Figure 5 the peaks are roughly 20 % higher than the average slow-down factor for this combination.

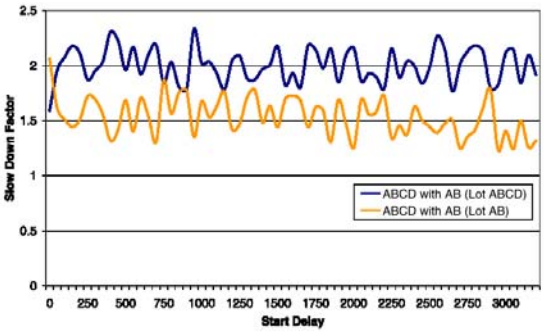

Figure 5: Slow-down Factor and Start Delay

Figure 6 shows that the first lot may be preferred by the cluster tool. This depends on the internal scheduler of the tool. Our simulator definitely tends to prefer the first lot. When the lot is the first (delay = 0) then the figure shows a slow-down factor of 1.5. When it is the second lot in the tool then the slow-down factor is almost 3. The slow-down factor decreases for higher delays, as the lot becomes the first lot when the other lot is replaced with the next lot.

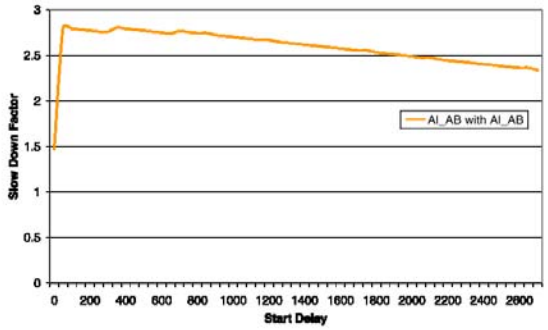

Figure 6: First Lot Preferred

The value range of slow-down factors

The optimal slow-down factor is 1. This means that the lot was not slowed down and it was as fast as in single mode. One might expect that 2 is a maximum and isreached as slow-down factor for identical lots, but this is not true. The scheduler may prefer some lots and penalize others, as fairness and throughput may be conflicting targets in some occasions. Precedence constraints can cause a wafer to block wafers of the other lot that could be processed with a higher throughput rate.
It is obvious that on average the slow-down factor should be less than 2. Otherwise, parallel-mode processing would be worse than single-mode processing and the lots should be processed one after the other.

Slow-down factors and pump and vent times

When a lot enters the cluster tool its loadlock needs to be pumped to the vacuum level of the cluster tool. The time for this operation is called pump time. During this time this lot does not influence other lots. So, the slow-down factor during this interval is 1 for both lots.
When all wafers of a lot are completed then normal air pressure has to be restored in its loadlock. The time for this operation is called vent time and during this time the slow-down factor is 1. However, pump and vent times are small compared to the overall cycle time of lots with standard lot sizes of, say, 25 wafers. So, we only consider the effects of pump and vent times when we have to deal with small lot sizes like in the next section.

Slow-down factors and lot size

As Figure 7 shows the variation of the slow-down factor and therefore of the cycle time is larger for small lot sizes and decreases when the lot size is increased.

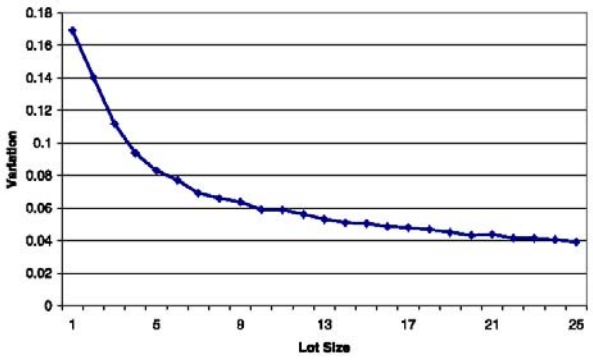

Figure 7: Variation Due to Lot Size

The slow-down factor itself did not significantly change for different lot sizes in general. However, when there is variation the variation tends to increase the slow-down factor. Thus, averages over all possible combinations will result in larger slow-down factors for small lot sizes.
Let us consider a lot with just one wafer. The time the wafer has to wait for a critical chamber to become available strongly effects the lot cycle time. Hence, obviously the transient is more important for lots with small lot sizes. Most wafers of lots with large lot sizes are processed during the steady state.

Slow-down factor variation

Finally, we examine the variation of the slow-down factors within the same recipe combination. Table 1 lists the average slow-down factors for all combinations, their average variation within each lot combination and the average minimum and maximum slow-down factor for each lot combination.

Table 1: Average Slow-down Factors (SDF)

Test set	*SDF*	*Var.*	*Min.*	*Max.*
Dresden fast handler	2.0	0.05	1.7	2.1
Dresden	2.2	0.07	1.8	2.3
Villach	2.0	0.08	1.6	2.1
Simple	1.9	0.05	1.7	2.1

Table 1 also shows that not all recipe combinations make sense, since in average the slow-down factor is roughly 2. A slow-down factor above 2 is not better than processing the lots one after another in single mode.

SIMULATION AND APPROXIMATION OF SLOW-DOWN FACTORS

Simulation

For the computation of simulated slow-down factors we created specific simulation studies. For any lot combination we simulated one lot of type A (lot A) being processed parallel to lots of type B. To ensure that lot A is always parallel to a lot of type B we used more than one of these lots (lot B, lot C.). As illustrated in Figure 8, for each lot combination we also simulated different delays between the start of the first lot of type B and lot A.

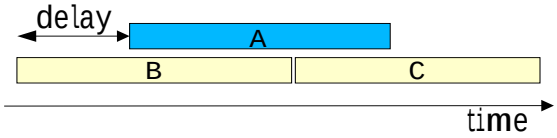

Figure 8: Simulating Slow-down Factors

Approximation

The slow-down factor indicates how recipes disturb each other. For different combinations there is usually a change in bottlenecks and we assumed that this change determines the slow-down factor to a considerable extend.

Definition 2: MBRPT
MBRPT (maximum flow bottleneck raw processing time) is our approximation approach for the slow-down factor. The approximation is computed as follows:

$$SDF(A, A+B) \approx \frac{MRPT(Bottleneck, A, A+B)}{RPT(Bottleneck, A)} \qquad (2)$$

where *RPT(Bottleneck,A)* is the bottleneck work load of a work load distribution for lot A alone and *MRPT(Bottleneck,A,A+B)* is the bottleneck work load of a work load distribution for lot A and lot B where the work load of all chambers that are not used by lot A is set to 0.

To compute a workload distribution we ignore the precedence constraints of the recipes. We recommend a heuristic algorithm using the LFJ rule (least flexible job first). The general problem of optimally distributing work is NP-hard and very similar to scheduling parallel machines with machine dedication. The LFJ rule is optimal for this problem when the sets of the machine dedication are nested and the processing times for parallel machines are equal (Pinedo 2001).

MBRPT approximates the simulated slow-down factors with an average error of 25 to 35 % depending on the scenario. This error is rather high, but this is no surprise as MBRPT assumes completely fair scheduling while the scheduler of the cluster tool may prefer lots.
The bias of MBRPT varies from – 2 % to 2 %, i.e., MBRPT is practically unbiased.

SCHEDULING APPROACH BASED ON SIMULATION RESULTS

Approximating lot cycle times

Given the cycle time of the lots in single mode (processed alone in the cluster tool) and given a simulated slow-down factor for each lot combination we can use these values to compute approximate lot cycle times. For each overlap we determine the length of the overlap and how much of the work of each lot has been completed during this overlap. With this idea we can predict the lot cycle times for all lots in a scenario with only few floating-point operations.

We simulated 20 scenarios with 20 lots each and then compared the predictions based on the simulated slow-down factors with the simulation results. Table 2 shows the average prediction error for the end time and cycle time of a lot. As the slow-down factor varies up to 20 % for different delays between the overlapping lots the prediction quality is limited by this variation. Additionally, prediction errors add errors to the predictions for the next lots. Analysis showed that the error is high for lot combinations with poor performance and when a lot is treated unfair by the cluster tool simulator (slow-down factor > 3). The error is smaller for lot combinations with high throughput.

Table 2: Prediction Errors

Test set	*Avg. Error [end time]*	*Avg. Error [cycle time]*
Dresden fast handler	14.3 %	26.3 %
Dresden	11.3 %	26.4 %
Villach	14.8 %	19.7 %
Simple	6.5 %	16.3 %

Scheduling with slow-down factors

The simulated slow-down factors can help to decide whether a lot combination is good or whether its performance is poor. When using a search algorithm to explore the search space of all possible schedules, the slow-down factor may be a good heuristic which paths to examine and which paths to ignore. As the simulated slow-down factor is an average taken from experiments with different delays it is a good measure for general lot compatibility while it may not be suitable for highly accurate predictions of particular lot cycle times for a scenario as in the last section. Pilot studies using dispatching rules on the basis of slow-down factors show promising results.

CONCLUSIONS

Among other advantages parallel mode cluster tools offer high throughput and high utilization. We demonstrated that the lot cycle times for cluster tools cannot be determined without simulating or predicting the complete scenario. This is caused by lots overlapping and sharing resources, hence, slowing down each other. Slow-down factors help to understand how lots disturb each other and can be used for a fast approximating of lot cycle times and as heuristic for scheduling.

Further studies have to be made for more representative results on the properties of slow-down factors and on the quality of the approximation approach. The predictions on the basis of slow-down factors have to be improved as they are promising for scheduling heuristics and can provide lot cycle time predictions with only few floating-point operations instead of long simulation runs.

Standard scheduling approaches do not take into account that the lot cycle times of parallel lots are correlated. This will be a field of further research in cluster tool optimization.

REFERENCES

Atherton, L.F. and R.W. Atherton. 1995. *Wafer Fabrication: Factory Performance and Analysis*. Kluwer.

Atherton, R.W; F.T. Turner; L.F. Atherton; and M.A. Pool. 1990. “Performance Analysis of Multi-Process Semiconductor Manufacturing Equipment.” In *Proceedings of theIEEE/SEMI Advanced Semiconductor Conference 1990*.

Bohr, M. 1999. “Schedulingverfahren für Cluster Tools in der Halbleiterfertigung.” Master thesis. Department of Computer Science. University of Würzburg, Germany.

Dümmler, M. 1999. “Using simulation and genetic algorithms to improve cluster tool performance.” In *Proceedings of the 1999 Winter Simulation Conference*. 875-879.

Koehler, E.J.; T.M. Wulf; and A.C. Bruska. “Evaluation of Cluster Tool Throughput For Thin Film Head Productions.” In *Proceedings of the 1999 Winter Simulation Conference*. 714-719.

Perkinson, T.L.; P.K. McLarty; R.S. Gyurcsik; and R.K. Cavin III. 1994. “Single-Wafer Cluster Tool Performance: An Analysis of Throughput.” *IEEE Transactions on Semiconductor Manufacturing*, 7 (3), 369-373.

Perkinson, T.L.; P.K. McLarty; R.S. Gyurcsik; and R.K. Cavin III. 1996. “Single-Wafer Cluster Tool Performance: An Analysis of the Effects of Redundant Chambers and Revisitation Sequences on Throughput.” *IEEE Transactions on Semiconductor Manufacturing*, 9 (3), 384-400.

Pinedo, M. 2001. *Scheduling. Theory, Algorithms, and Systems*. 2nd edition. Prentice-Hall.

Schmid, M. “Modellierung und Simulation von Cluster Tools in der Halbleiterfertigung.” Master thesis. Department of Computer Science. University of Würzburg, Germany.

Seidel, G. 2001. “Simulation und Optimierung von Cluster Tools in der Halbleiterfertigung”. Master thesis, Institute of Mathematics. Technical University of Graz, Austria.

Srinivasan, R.S. 1998. "Modelling and Performance Analysis of Cluster Tools Using Petri Nets." In IEEE Transactions on Semiconductor Manufacturing Vol. 11, 1998.

AUTHOR BIOGRAPHIES

HEIKO NIEDERMAYER is Ph.D. candidate at the Institute of Computer Science (Chair of Computer Networking and Internet) at the University of Tübingen. He received an M.S. degree in Computer Science from the University of Würzburg. His e-mail address is: niederma@informatik.uni-tuebingen.de.

OLIVER ROSE is assistant professor in the Department of Computer Science at the University of Würzburg, Germany. He received an M.S. degree in applied mathematics and a Ph.D. degree in computer science from the same university. He has a strong background in the modeling and performance evaluation of high-speed communication networks. Currently, his research focuses on the analysis of semiconductor and car manufacturing facilities. He is a member of IEEE, ASIM, INFORMS, and SCS. His web address is: www3.informatik.uni-wuerzburg.de/~rose.