Improvement of Cluster Tool Performance Using Simulation

Khalid Ahmed Ibrahim[1]

[1]*Karary University, Associate Professor, Faculty of Computer Science and Information Technology*

Abstract - *This paper describes the concept of clustering which plays a significant role in semiconductor manufacturing plants. In this project, cluster tools have been used for wafer fabrication. Cluster tool is like a machine with number of processing chambers, a handler and number of Load locks. A lot is a box filled with wafers and it is to be loaded in the load lock, ready for processing. Cluster tools can be operated in parallel mode. Load lock consists of lots. When lots are undergoing a process in processing chambers different kinds of lot combinations may be possible. There is a dramatic reduction in speed when lots are processed in parallel. The fact that the lots are utilizing the same resources does not help the cause either. In order to overcome this problem, lots are to be scheduled in such a way that scheduled lot combinations do not affect each other. The cycle time of the individual lots and lot combinations would be computed. Since cycle time cannot be computed during the time of process, simulation is adopted. Usage of Genetic algorithms plays a key role in determining the optimal sequence of lots*

Key Words: Clustering, Semiconductor, Wafer Fabrication, Load Lock, Cycle Time, Simulation

1.INTRODUCTION

Cluster tools are machines that combine various processing sequence in one machine. They consist of load locks, handlers and other machines to process the wafers of the lots in the load locks.

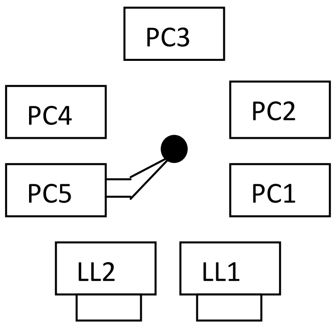


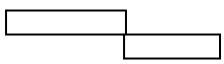
Figure 1. Cluster tool model

Most modern cluster tools have 2 load locks. Each load lock can be loaded with the lot. The container that has wafers is a lot. Then, the cluster tool processes the lot. The wafer in the lot is scheduled inside the cluster tool. The scheduler of the tool carries out this work. The handlers are like transfer robot that are used for moving the wafers between the chambers and load locks. The chambers are machines to process the wafers. Since the processing of the wafers is pipelined the cycle time of the lot in cluster tool is reduced when compared with the cycle time of a lot using the consecutive machines.

The inconvenience with cluster tools is that their behavior is more complex than the behavior of simpler machines. Since the cluster tool is operated in parallel mode, it is able to process lots in parallel that share the same resources. The cycle time of a lot is not constant but depends on the situation inside the cluster tool during the processing of the lot. 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.

No overlap Lot A

Lot B



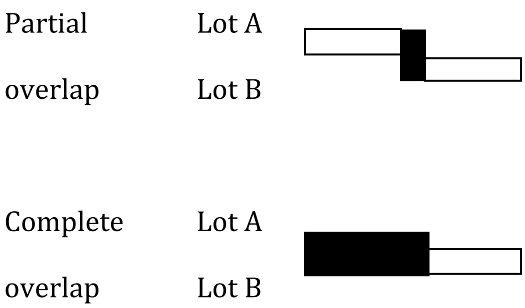


Figure 2. Various overlapping scenarios

1.1. Single Mode

Most cluster tools can operate in single mode as it offers lowest tool cycle times. According to this mode, it does not support the processing of more than one lot at a time.

1.2. Parallel Mode

In parallel mode, a cluster tool is like a set of parallel machines. Each load lock is identical to one machine. The difference with real parallel machines is that these "pseudo-parallel" machines share the same resources. Thus, there is an influence of one machine on the other and vice versa. The fact is that the tool cycle times of lots are correlated. As a consequence, all overlaps of lots inside the cluster tool contribute to the tool cycle time.

So, it is not possible to compute tool cycle times in parallel mode directly without computing the tool cycle times for the complete scenario. This can be done via simulation. The simulation approach is used to speed up the computation of the tool cycle time with an approximation.

Definition 1. **Slow-down Factor**

The slow-down factor of lot A while processed in parallel with lot B is defined as

$$SDF(A,A+B) = \frac{CT(A,A+B)}{CT(A)}$$

where *CT(A,A+B)* is the cycle time of lot A when it is processed together with lot B and *CT(A)* is the cycle time of lot A when it is processed alone (single mode). For simplification of the definition, an assumption can be made that lot B is large enough so that lot A is processed in parallel with lot B for its complete tool cycle time. When the slow-down factors for all lot combinations is computed. Then those values and the tool cycle times in single mode is used to approximate the tool cycle times of the lots. To compute the tool cycle times, start with the first overlap and use the slow-down factors of the lots to determine the length of the overlap and the work that is done for each lot. This process is repeated until all lots are completed.

Definition 2.**Cycletime of the lot**
The time required by the lot to complete the process.

3. Algorithm

```
INPUT: Queue with lots in the order given by the schedule, Time
WHILE Queue.notEmpty( ) and Clustertool.notEmpty( )
DO
  FOR all empty loadlocks DO
     IF Queue.nextLotReady(Time)
     THEN
       Add Q.next() to loadlock and set
       its start time.
    ENDIF
  ENDFOR
```

```
LengthOfOverlap=
    TimeToNextEvent (all  lots in
    cluster tool);
Time = Time + LengthOfOverlap
FOR all lots in cluster tool DO
        Determine the amount of work
       done for lot during the overlap.
   IF lot is completed
   THEN
        Remove lot from load lock and set
        its completion time.
   ENDFOR
ENDWHILE
```

4. Cluster Tool Simulation

Simulation is used for two purposes. The first one is to determine slow-down factors for all combinations of lots. The slow-down factor may vary depending on various factors like the manner in which lots overlap and the order of processing of lots

The second purpose of simulation is to evaluate the schedules computed by the different optimization approaches

5. Simulation Engine

A simulation engine can be developed for cluster tools. The simulation model can consists of an arbitrary number of cluster tools, each of which can have an arbitrary number of process chambers, load locks and handlers. The most important parameters that can be specified in the simulation model are listed in Table 1.

Table 1 Model Parameters

Cluster tools	Number of Process chambers Number of Load locks Number of handlers
Handlers	Move time for every origin destination pair
Sequences	Number of process steps
Process steps	Process chambers qualified for a step Process time in these chambers
Lots	Number of wafers per lot
Wafers	Process sequence

The simulation engine computes a number of output statistics after a simulation run, such as cycle times of wafers and utilization of the components of a cluster tool.

6. Combination of the Lots

To identify combinations of two lots that lead to low lot cycle times when processed in parallel, simulation experiment can be done, using the cluster tool model.

Starting with an initially empty cluster tool, put a lot of process sequence i Σ {1,….,4} in load lock 1 and a lot of process sequence j Σ{1,….,4} in load lock 2 simultaneously. For all the possible combination of sequences, measure the cycle time for both lots. For every combination i, j , the ratios

Ti, j/Ti is computed , where Ti,j is the cycle time of a lot of sequence i when processed together with a lot of sequence j . Ti is the cycle time of a lot of sequence i when processed exclusively.

Sample test result

Obviously, there will be combinations that lead to more favorable cycle times for both lots than other combinations. For example, combining sequence 2 and 3 leads to an increase in cycle time of 29% for sequence 2 and of 38% for sequence 3. Hence, this is a more favorable combination than, e.g., sequence 1 and 3, which leads to an increase in cycle time of 85% for sequence 1 and of 50% for sequence 3. Combining process sequences 2 and 3 leads to shorter cycle times.

Table 2. Cycle Time ratios of lot combination

SEQUE -NCES	j =1	j =2	j =3	j =4
i =1	1.86	1.85	1.85	1.35
i =2	1.66	1.78	1.29	1.34
i =3	1.50	1.38	1.88	1.39
i =4	1.25	1.53	1.76	1.72

Table 2 can be used as a guideline for operators to choose combinations of process sequences that lead to short cycle times. However, when a large number of lots has to be sequenced on several cluster tools, this task can no longer be performed manually.

7. Optimization of Sequences of Lots using Genetic Algorithm

A heuristic method based on a genetic algorithm (GA) has been proposed to generate the lot sequences. The basic idea behind a GA is to imitate an evolutionary process: The best individuals or *genomes* of a generation survive and reproduce to pass on their genetic material to the next generation. Roughly spoken, the GA needs three pieces of input data: A data structure to represent the genomes, operators on this data structure that allow the genetic algorithm to create new solutions and an objective function to evaluate the *fitness* of a genome.

The default operators can be used on lists and arrays that are implemented and documented in the GAlib library to generate new genomes. As the objective function, the time required for processing all lots according to the sequence is used as the GA suggests. This time is derived using the simulation model of the cluster tools. The GA uses this objective function to evaluate a genome and to decide whether it is "fit" enough to survive and reproduce.

8. Conclusions

An approach is presented that uses a simulation for finding cycle time of the lots. It has been found that there is a vast amount of reduction in time as compared to conventional approaches. A genetic algorithm to find optimal sequences of lots at the cluster tool. Instead of testing all sequences, the GA need to test only selected sequences to find its best solution. Finally, the number of sequences generated to find the optimum and the run time of the GA can be displayed.

Sequences can be re–optimized in only a few minutes if the set of lots waiting for processing changes. Another advantage of the genetic algorithm is that it is an *anytime* algorithm. This means, that if a result is needed before the genetic algorithm has terminated, the computation can be stopped and the genetic algorithm will respond with the best current solution. When applied on the real manufacturing floor, the proposed algorithm can lead to a significant reduction in cycle times.

REFERENCES

[1] Goldberg, and Encore, Introduction to Genetic Algorithm, 1998 and 1997

[2] http://lancet.mit.edu/ga

[3] H. Niedermayer, and O. Rose, "A simulation-based analysis of the cycle time of cluster tools in semiconductor manufacturing", In proceedings of 15th European simulation symposium, Delft, Netherlands, 2003.

[4] H. Niedermayer, and O. Rose, "Approximation of cycle time of cluster tools in semiconductor manufacturing", In proceedings of annual IIE Engineering Research Conference, Texas, 2004.

[5] M. Dummler, "Modeling and optimization of cluster tools in semiconductor manufacturing", Ph.D thesis, Department of computer science, University of Wurzburg, Germany, 2004 .

[6] M. Dummler, "Using simulation and Genetic algorithm to improve cluster tool performance", In proceedings of the 1999 winter simulation Conference, 875-879

[7] S. Oechsner, and O. Rose, "A filtered beam search approach to scheduling cluster tools in semiconductor manufacturing", in proceedings of IERC'05, Atlanta, USA, May 14-18 2005.

[8] S. Oechsner, and O. Rose, "Scheduling cluster tools using filtered beam search and recipe comparison", In proceedings of 200 winter simulation conference,2005