A Due-Date-Based Algorithm for Lot-Order Assignment in a Semiconductor Wafer Fabrication Facility

Yeong-Dae Kim, *Member, IEEE*, June-Young Bang, Kwee-Yeon An, and Seung-Kil Lim

***Abstract*—This paper focuses on a lot-order assignment problem, called the pegging problem, in a semiconductor wafer fabrication facility. Pegging is a process of assigning wafer lots to orders for wafers. We consider two types of pegging strategies: hard pegging strategy, under which the lot-order assignment is not changed once lots are assigned to orders; and soft pegging strategy, under which the lot-order assignment can be changed during the production period. For the soft pegging strategy, we develop three operational policies and three algorithms for the pegging problem of assigning lots to orders with the objective of minimizing total tardiness of the orders. To evaluate performance of the suggested policies and algorithms, we perform simulation experiments using real factory data as well as randomly generated data sets. Results of the simulation tests show that the repegging policies and the algorithms operated under the soft pegging strategy give better results than the hard pegging strategy.**

***Index Terms*—Lot-order assignment, scheduling, semiconductor wafer fab.**

I. Introduction

IN THIS PAPER, we consider lot-order assignment problems in a semiconductor wafer fabrication facility. To survive in competitive business environments, semiconductor manufacturing companies must meet customers' demands in terms of quality, quantity, and due dates. These days, even small companies can design their own semiconductor chips without a wafer fabrication facility (wafer fab) and want (and place orders to) a semiconductor manufacturer to produce their products according to their designs. If product deliveries are frequently late, the manufacturer may have to pay a significant amount of penalty charges or may lose the customers. However, it is very difficult to optimize the material flow in the fab due to the complexity of the manufacturing process. Therefore, the manufacturer needs to develop an effective and efficient scheduling and control policy in order to meet due dates of orders.

Semiconductor products are manufactured through the process of wafer fabrication, probe or electrical die sorting (EDS), assembly, and final test. Among the four, wafer fabrication is the most complex and time-consuming process. It involves a complex sequence of processing steps with a large number of operations. In this paper, we consider a production scheduling and control problem in a wafer fab in which wafers for orders with different due dates are produced. The due date of an order for wafers in the wafer fab is set by considering the due date of the order for final products associated with the wafers, i.e., semiconductor chips, and production lead time required for EDS, assembly and final test. If an order is completed later than the due date in the fab, it is not easy to meet the due date of the final products. Therefore, meeting due dates in the fab is also very important.

An order for wafers is specified by the due date, product (wafer) type, and the number of wafers to be produced. Usually, in semiconductor wafer fabs, wafers are processed in a lot of 25 wafers (or less). Here, a wafer lot is the basic processing and transfer unit, that is, it denotes a set of wafers that are processed and moved together. One or more wafer lots need to be processed for an order, and hence it is assumed in this study that an order is composed of one or more wafer lots. Since wafer types of different orders may be the same in many cases, a lot can be assigned to multiple orders. However, for an efficient order management or managerial convenience, a wafer lot is usually assigned to one order in the fabs. Note that if a lot is assigned to one order, one can monitor and control progresses of orders relatively easily by checking the progresses of the lots. Such assignment of lots to orders is called *pegging*. This paper focuses on a pegging problem in a semiconductor wafer fabrication facility. In the fab considered in this study, a wafer lot is usually composed of 25 wafers, but there may be lots that are composed of less than 25 wafers, since sizes of orders, i.e., the numbers of wafers to be produced for orders, are not necessarily multiples of 25.

To complete or satisfy an order of a customer, who is an outside customer or the EDS line, all lots for the order should be completed (in the fab) and delivered to the customer. Therefore, pegging is as important as operations scheduling for meeting due dates of orders (and other managerial objectives). Also, lot-order assignments may have to be changed when unexpected events happen. For example, if an urgent order arrives or if machines fail and some lots cannot be processed, lot-order reassignment, to be called repegging in this study, may help to reduce tardiness of related orders.

Manuscript received April 16, 2006; revised November 1, 2007.

Y.-D. Kim and J.-Y. Bang are with the Department of Industrial Engineering, Korea Advanced Institute of Science and Technology, Daejon 305-701, Korea (e-mail: ydkim@kaist.ac.kr; juneyb@kaist.ac.kr).

K.-Y. An is with the Semiconductor Memory Division, Samsung Electronics Co., Gyeonggi-Do 449-711, Korea (e-mail: kweeyeon.an@samsung.com).

S.-K. Lim is with the Division of e-Business Information Technology, Sungkyul University, Gyeonggi-Do, 430-742, Korea (e-mail: *seungkil*@sungkyul.edu).

Digital Object Identifier 10.1109/TSM.2008.2000261

There are two pegging strategies: hard pegging strategy and soft pegging strategy. Under the hard pegging strategy, once an assignment of lots to orders is made, the assignment is fixed and never changed. On the other hand, under the soft pegging strategy, the assignment of lots to orders can be changed if such a change results in better performance. The hard pegging strategy is adopted in most production systems currently, whereas the soft pegging strategy is adopted in only a few systems. In small and simple manufacturing systems, soft pegging may be executed manually by the manager of the system or supervisor of the manufacturing line. However, the soft pegging strategy has not been implemented in many complex systems such as semiconductor wafer fabs. In this paper, we suggest pegging policies that can be operated under the soft pegging strategy and develop algorithms that can be used for the policies.

There have been a number of research articles on production scheduling and control problems in semiconductor manufacturing systems, such as problems of lot release control, lot scheduling in serial processing workstations, and batch scheduling in batch processing workstations. Several lot release rules are developed and used for release of wafer lots into the fab in many studies including those of Wein [1], Glassey and Resende [2], and Kim *et al.* [3]. In these rules, information on the workload at a bottleneck workstation is used for the lot release. In most previous studies on lot scheduling problems in wafer fabs, dispatching rules have been used for sequencing. Also, researchers focus on lot scheduling problems on bottleneck workstations of the fabs, such as the photolithography workstation, in most studies (Graves *et al.* [4], Lou and Kager [5], Lee *et al.* [6], Min and Yih [7], Yoon and Lee [8], and Lin *et al.* [9]). Batch scheduling problems have been dealt with in a few studies as well. For example, Glassey and Weng [10] give a method for scheduling jobs of a single job family on a single batch processing machine, and Fowler *et al.* [11], Robinson *et al.* [12] and Fowler *et al.* [13] deal with multiproduct and multiserver cases.

Most of previous research on production scheduling in semiconductor manufacturing systems have focused on objectives related to throughput, cycle time or equipment utilization, but due-date related performance measures have not been considered very often although meeting due dates of customers' orders is very important in the current competitive market environments. To deal with due-date related performance measures, researchers have developed several scheduling methods. For instance, Kim *et al.* [14], [15] suggest dispatching rule-based algorithms for lot release control and lot scheduling, and Kim *et al.* [16] develop a real-time scheduling method in a wafer fab, for the objective of minimizing tardiness of orders for wafers. Also, Jain *et al.* [17] develop a generalized stochastic Petri net model for wafer fabrication and develop a simulated annealing-based scheduling strategy, and Mason *et al.* [18] propose strategies for rescheduling jobs in complex job shops such as wafer fabs, for objectives related with due dates.

Although there are many practical applications of the soft pegging strategy, application to semiconductor manufacturing systems is very rare. Also, there are not many research articles on pegging problems, and these problems are considered more often for assembly lines or manufacturing shops that produce final products. Steiner and Yeomans [19] consider a pegging problem in a just-in-time assembly system and propose an integer programming approach. For semiconductor assembly and test facilities, Knutson *et al.* [20] and Fowler *et al.* [21] formulate pegging problems of lot-order matching as integer programs and suggest heuristic algorithms after transforming the problems into bin packing problems. Carlyle *et al.* [22] extend the research and suggest refined algorithms for the pegging problems dealt with in Knutson *et al.* [20] and Fowler *et al.* [21]. Without giving a detailed procedure for lot-to-order (re)assignment, Wu [23] presents a systematic pegging method for wafer fabs, in which assignments for lots of certain orders are fixed while assignments for lots of other orders may be changed. Recently, Bang *et al.* [24] propose soft pegging algorithms that can be used for reassignments of lots to orders in wafer fabs.

In this paper, we suggest pegging policies that can be used under the soft pegging strategy and develop pegging algorithms for the policies for the objective of minimizing total tardiness of orders. Here, the tardiness of an order, order i, is defined as $\max(C_i - d_i, 0)$, where C_i and d_i are the completion time and due date of order i, respectively. The completion time of an order is the time when all wafers for the order have been completed in the fab. This paper is organized as follows. In the next section, we present several pegging policies for the soft pegging strategy in a semiconductor wafer fab. In Section III, we suggest algorithms for lot-order (re-)assignment for the objective of minimizing tardiness of orders. The performance or effect of the pegging policies and the pegging algorithms is investigated by a simulation study in Section IV. Section V concludes the paper with a short summary and suggestions for further research.

II. Pegging Policies

For the hard pegging strategy, one only needs to decide how to assign wafer lots to orders, since assignments are fixed until lots are completed in the fab, that is, lots are not reassigned to other orders in the fab once the assignments are done. For the soft pegging strategy, however, it is necessary to determine the operational policy. For example, one needs to determine when and how lot-order assignments are to be changed in addition to how the initial lot-order assignment is to be made. Such an operational policy, called the pegging policy here, for the soft pegging strategy should be determined carefully since tardiness of orders as well as ease of implementation in the fab may be affected by the policy. Although there may be various decisions to be made for the pegging policy, we focus on when lots are to be reassigned to orders and which lots and orders are to be included for reassignments.

In general, the system performance can be improved if repegging is performed more frequently and/or if more orders and more lots are considered for repegging. However, it may not be desirable to perform repegging too often or to take all orders and lots into consideration, since it takes time to collect and retrieve information on the states of the fab as well as wafer lots and orders, such as the location of each wafer lot in the fab. Note that there are thousands of wafer lots and hundreds of orders at any point of time in a typical wafer fab. In addition, we need to consider the stability of production schedules at the downstream process, i.e., the EDS line. Frequent repegging, which changes the release times of wafers into the EDS line, makes it difficult for the EDS line to prepare for production operations since setup operations are performed according to the expected arrival time of wafer lots, i.e., the expected completion time at the wafer fab.

When repegging is performed under the soft pegging strategy, all the current lot-order assignments are ignored and lots are reassigned to orders that are best for the lots according to the current information of the lots, orders and the fab. In this study, three policies are proposed under the soft pegging strategy. These policies can be implemented in the real fab without much difficulty. The differences between these policies are the points in time when repegging is performed and the sets of orders and lots that are considered for repegging.

Periodic Repegging Policy (PRP): In this policy, repegging is performed periodically. Orders and lots of all product families are considered for repegging whenever repegging is performed. It is assumed that the interval of repegging is 8 h, but we test 24 h and 48 h as the interval as well in this study. It is assumed that repegging is performed just before every shift begins, when the interval is 8 h.

Event-Based Repegging Policy 1 (EBRP 1): In this policy, repegging is performed only when urgent orders arrive. As a result of repegging, lots with more progresses, i.e., those that are in the later stages of processing, are reassigned to more urgent orders, and we have more chance to meet the due date with such repegging. In this policy, all lots and all orders are considered for repegging whenever repegging is performed.

Event-Based Repegging Policy 2 (EBRP 2): This policy is identical to the above policy except for the sets of orders and lots considered for repegging. In this policy, repegging is performed only for the product families associated with the urgent order(s) and/or orders affected by unusual events such as machine breakdowns.

III. Pegging Algorithms

In this section, we present pegging algorithms for the lot-order reassignment problems that are to be solved when repegging is performed. Note that scheduling problems are imbedded in a lot-order reassignment problem, since schedules of lots are needed for evaluation (and actual implementation) of alternatives for lot-order assignments. Since optimal solutions for the lot-order reassignment problems cannot be obtained within a reasonable amount of computation time, we develop heuristic algorithms for the problems. Note that the scheduling problem embedded in the lot-order assignment problem is a hybrid job shop scheduling problem with reentrant flows, and that the job shop scheduling problem, a special case of the hybrid job shop problem, is shown to be NP-complete [25]. (In hybrid job shops, there may be multiple parallel machines in each processing stage.)

It is assumed in this study that the list scheduling method is used for scheduling in the fab as it is used practically in many fabs. In the list scheduling method, when a machine becomes available for processing a lot, a lot with the highest priority is selected among those that are available at the time and scheduled on the machine. Note that this method can be used for dynamic scheduling as well as static scheduling. One can reduce total tardiness by using good scheduling rules. When an unforeseen event occurs such as arrival of an urgent order and machine breakdowns, schedules already made and being implemented may have to be changed. However, it may be better to reassign lots to orders (and obtain a new schedule) than only to reschedule the lots. In other words, one may get a better result by rescheduling the lots after repegging rather than just rescheduling the lots.

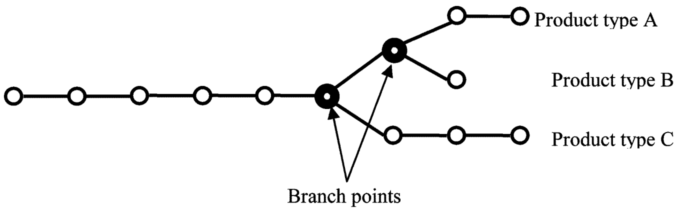


Fig. 1. Illustration of branch points.

Basically, repegging can be performed for wafer lots and orders of the same product type. However, even though wafer lots are not of the same type, assignment of those lots to orders can be changed if the product types of associated lots are of the same product family and processing of those lots have not progressed beyond a certain point, called the *branch point*. Here, the branch point denotes the point in the processing steps from which operations of two or more product types (of the product family) become different. Note that the processing steps for wafers of different product types are identical up to the branch point. Therefore, lots of different product types of the same family are exchangeable up to the branch point. Branch points are illustrated in Fig. 1, in which operations for wafer products are denoted as nodes. In the figure, there are two branch points, one for product types A and B, and one that divides the processing steps of type C and those of types A and B.

Under the soft pegging strategy, assignments of lots to orders may be changed. Therefore, to reduce the total tardiness of orders, we can reassign lots that have been processed more to more urgent orders with less slack time (and lots that have been processed less to less urgent orders with more slack time). However, because of restrictions (due to the manufacturing processes) on the reassignment of lots to orders, repegging can be performed for lots and orders of the same product family only. This means repegging for one family is independent of that for other families. Therefore, we develop algorithms that can be used to repeg lots and orders of a product family. These algorithms can be applied to each product family when repegging is performed.

In the suggested algorithms, when repegging is performed, the current lot-order assignments are ignored, and lots are reassigned to orders based on the urgency of the orders. The basic idea is to assign the most progressed lots, i.e., lots with the most progress, to the most urgent order at the moment. With this method, we may be able to satisfy due dates of the orders more effectively. In the algorithms, the urgency of an order is determined with a priority rule under the assumption that the most progressed lots among those that can be assigned to the order at the moment are assigned to the order. Because reassignments can be done among lots (of orders for the same product family) with the same manufacturing process, it should be checked whether processing for the lots has been progressed beyond the branch point.

In this study, the pegging algorithms are specified by the rules for determining urgencies of the orders. We devise three algorithms, i.e., three different priority rules for determining the urgencies. These rules are based on slack times of the orders. Since slack time of an order cannot be calculated exactly because of the complexity of the manufacturing process for wafers and the material flow in the fab, it is estimated with a certain method. The priority rules are presented in the following. First, we give notation used in the description of the rules. Note that i denotes the index of the order for which the priority is computed currently and j denotes the index of a lot that is considered for being assigned to order i.

Λ_i Set of indices of lots that can be assigned to order i, i.e., those that are progressed more than others but not beyond the branch point.

d_i Due date of order i.

P_i Sum of processing times of lots to be assigned to order i.

n_i Number of lots that are to be assigned to order i.

h_i Difference in remaining work of the first lot, the most progressed lot, and that of the last lot, the least progressed lot, among the lots that are to be assigned to order i.

t Current time when a repegging decision is made.

R_j Remaining work of lot j, i.e., the sum of processing times of remaining operations that should be performed for lot j.

W_j Estimated waiting time of lot j at the bottleneck workstation (photolithography workstation in the fab considered in this study) that will be incurred when the lot visits the bottleneck workstation.

α, δ Parameters used in a priority rule.

Now, we present priority functions of the three priority rules, named estimated slack (ES), modified slack over estimated completion time for pegging (MSEC-P), and modified estimated slack over remaining work 2 (MES/RW2), which are modified from scheduling rules named slack, MSEC-S, and ES/RW2 of Kim *et al.* [15], respectively. Although there are various scheduling rules that can be modified for pegging algorithms, these three scheduling rules are used as base rules to be modified in this study. Note that these rules worked well for the minimization of tardiness of orders in the study of Kim *et al.*, possibly because urgencies of orders can be more exactly estimated by these rules as discussed in Kim *et al.* We modify these three rules in this study since good estimation of urgencies of orders may result in good performance of pegging algorithms. In the three modified rules, an order with the smallest value of the priority function has the highest priority, or considered most urgent (see bottom of page).

As mentioned above, slack time of an order is set to the slack time of a lot with the least progress among lots that are to be assigned to the order. Also, the estimated waiting time of lot j, W_j, represents the sum of estimated waiting times that will be incurred when the lot visits a bottleneck workstation, the photolithography workstation in the fab considered in this study, and it is computed as the product of the average work-in-process inventory (WIP) level of the bottleneck workstation, the average processing time of the lot at the workstation, and the number of times the lot still has to visit the workstation until it is completed in the fab.

In ES, only remaining work and waiting time in the bottleneck stations are considered. The remaining work is computed as the sum of processing times of remaining operations that are to be performed until the lot is completed. The second and third methods are slightly modified from those suggested by Kim *et al.* [15]. The priority function of MSEC-P denotes the ratio of a modified slack time to an estimated completion time, while that of MES/RW2 is a modified version of the ratio of estimated slack to remaining work. In MES/RW2, the term $a \cdot h_i/\{P_i \cdot (n_i+1)\}$ can be regarded as a penalty term, which becomes smaller if lots with similar progresses are assigned to the order. Note that if some lots assigned to an order are progressed much while others are progressed very little, the order is given a large penalty value in the rule. In this rule, α and δ are parameters whose values should be determined after tests on several candidate values. (In the simulation experiments performed in this study, α and δ were set to 15 and 0.0001, respectively, after tests on candidate values.) Wafer lots that are progressed most but not beyond the branch point are reassigned to the order with the highest priority.

The overall procedure for the lot-order (re-)pegging algorithms suggested in this study can be summarized as follows. This procedure can be applied to lots and orders of product family f.

Procedure 1 (Pegging Algorithm for Family f):

Step 0. Let L_f and O_f be the sets of wafer lots and orders of product family f, respectively, that have not been completed yet.

$$\mathrm{ES}: \min_{j\in\Lambda_i}\{d_i - R_j - W_j - t\}$$

$$\mathrm{MSEC-P}: \min_{j\in\Lambda_i}\left\{\frac{(d_i - R_j - W_j - t)^+ + 1}{t + R_j + W_j}\right\}$$

$$\mathrm{MES/RW2}: \min_{j\in\Lambda_i}\left[\max\left\{\frac{\{d_i - R_j - W_j - t + \alpha \cdot h_i/P_i(n_i+1)\}^+}{(R_j + W_j)}, \delta \cdot R_j\right\}\right]$$

Step 1. For each order in O_f, compute the priority function values assuming lots which are progressed most among those in L_f but not beyond the branch point are assigned to the order. Select an order with the highest priority, i.e., an order with the minimum priority function value. Let the index of the selected order be $i*$.

Step 2. Let the set of lots that can be assigned to order $i*$, i.e., those that are progressed more than others but not beyond the branch point be Λ_{i*}. Reassign lots in Λ_{i*} to order $i*$. Let $O_f \leftarrow O_f - \{i*\}$ and $L_f \leftarrow L_f - \Lambda_{i*}$.

Step 3. If $O_f = \phi$, stop. Otherwise, go to *Step* 1.

IV. Simulation Experiments

The three repegging policies with the three lot-order (re-)assignment algorithms suggested in this study are evaluated through simulation experiments. For the experiments, we generated problem instances based on data of a real fab in a semiconductor manufacturing company in Korea. The following summarize information of the real fab as well as wafers and orders used in the simulation model.

1) Eight major processing workstations were included in the model: chemical/mechanical polishing, chemical vapor decomposition, diffusion, dry etching, implantation, photolithography, sputter, and wet chemical etching, each with multiple parallel machines. There are 501 machines, and 196 of them are batch processing machines.
2) There are 1098 different product types and these products are aggregated into 90 different product families.
3) The size of (the number of wafers for) an order ranges from 25 to 300 wafers. Order sizes of approximately 70% of the orders are integer multiples of 25, while those of the others are not. Other than this, there are no specific order sizes that appear more frequently than others, that is, the order sizes may be considered to be uniformly distributed. All lots except at most one lot for each order are composed of 25 wafers.
4) The processing time for a product (wafer) on a machine ranges from 5 to 240 min.
5) The number of operations required for a product ranges from 121 to 266.
6) Each product is composed of 10 to 15 layers of circuits, and hence, each wafer lot should visit workstations up to 10 to 15 times.

In the simulation model, it is assumed that orders for approximately 3000 wafers arrive in each day, as in the real fab. The due date of order i, d_i, is given as

$$d_i = a_i + P_i \cdot \mathrm{TN}\ (2.184, 0.74^2; 1.1, \infty)$$

where a_i is the time when order i arrives, P_i is the sum of processing times of all operations for the order and $\mathrm{TN}(m, v; l, u)$ is a random number generated from a truncated normal distribution with mean m, variance v, lower limit l, and upper limit u. We consider three scenarios by varying the percentage of urgent orders and due dates of urgent orders. We assume that there is no urgent order in scenario 1, while the percentages of urgent orders are 5% and 10% in scenarios 2 and 3, respectively. The due date of an urgent order, i, was set as $d_i = a_i + 1.0 \cdot P_i$ in both scenarios. Note that it was found from simulation results that most of the urgent orders in scenarios 2 and 3 became tardy under hard pegging policy, although lots of the urgent orders were given higher (scheduling) priorities than the other lots. We did not consider machine breakdowns or yield losses in the simulation model.

As a method for releasing lots into the fab, a priority rule suggested in Kim *et al.* [15], named the order slack rule (OS), was used for selection of a lot to be released, and the uniform release rule (UNIF), as given in Glassey and Resende [2], was used for determining the time the selected lot is to be released. In the method, a lot with the highest priority is released into the fab first. The priority of lot j is computed as

$$d'_j - Q_j - t - W_j - \beta \cdot N_j$$

where d'_j is the due date of lot j, Q_j is the sum of processing times of all operations for lot j, W_j is an estimated total waiting time of lot j at the bottleneck station, N_j is the number of lots required for the order associated with lot j, and β is a parameter used in the release rule. In the simulation, β was set to 10 (after tests on several candidate values). Also, in UNIF, a selected lot is released into the fab in a constant rate (up to 3000 wafers a day) regardless of the current systems states. (Note that a rule with the same basic concept as that of UNIF is used in the fab considered in this study.)

In the simulation experiments, we used scheduling rules given in Kim *et al.* [15], named ES/RW2, for lot scheduling at serial processing workstations, and PUCH for scheduling at batch-processing workstations, since they showed good performance in terms of due-date related performance measures. In ES/RW2, priorities of the lots are determined by estimated slack time per remaining work, and the operation due date is given to each operation considering its remaining work, not to each lot or order that includes the operation. On the other hand, in PUCH, information on processing urgencies of lots and the number of waiting lots in the queue is used for selecting a batch to be processed first, grouping wafer lots into batches, and determining processing sequences of these batches. See Kim *et al.* [15] for more details of these rules.

The simulation model was coded with Factor/AIM, a simulation software developed by Pritsker Corporation, with additional user codes written in the C language. The simulation experiments were performed on a personal computer with a Pentium 4 processor running at 2.8 GHz clock speed. The actual inventory level (at the time this research was conducted) of each workstation of the fab considered in this study was used for initial states of simulation runs. For each combination of the repegging policies and pegging algorithms, ten simulation runs of the length of six months were made, and results of the last five months were used for comparison.

For evaluation of the performance of repegging policies and the repegging algorithms developed for the soft pegging strategy, they were compared with the hard pegging strategy, which had been adopted in the real fab considered in this study. Results of the simulation experiments are given in Table I, which shows the percentage reduction in total tardiness obtained from the repegging policies/algorithms from that obtained from the hard pegging strategy operated with ES/RW2

TABLE I
PERFORMANCE OF PEGGING STRATEGIES AND PEGGING ALGORITHMS

Pegging policy	Pegging algorithm	Scenario 1	Scenario 2	Scenario 3
PRP	ES	34.5 (9.2)†	46.6 (12.7)	56.3 (13.4)
	MSEC-P	40.7 (11.6)	46.4 (12.5)	61.4 (11.5)
	MES/RW2	48.7 (11.2)	56.5 (13.7)	64.9 (12.3)
EBRP1	ES	-	45.1 (14.7)	57.5 (11.4)
	MSEC-P	-	46.8 (15.3)	57.8 (13.8)
	MES/RW2	-	54.4 (14.7)	64.4 (11.5)
EBRP2	ES	-	34.0 (12.7)	47.1 (14.0)
	MSEC-P	-	33.4 (10.5)	47.1 (13.6)
	MES/RW2	-	34.6 (12.2)	52.5 (10.8)

† Average and standard deviation (in parenthesis) of the percentage reduction from the total tardiness obtained from the hard pegging strategy with ES/RW2 and PUCH used for scheduling at serial processing workstations and batch processing workstation, respectively

and PUCH as scheduling rules. Note that repegging is not performed in scenario 1 under the event-based repegging policies because there are no urgent orders that trigger the repegging process.

All the policies under the soft pegging strategy worked better than the hard pegging strategy. By allowing flexible reassignments of lots to orders, the soft pegging strategy provided environment for better order management. There was no significant difference between the performance of the periodic repegging policy (PRP) and the event-based repegging policy 1 (EBRP1), but they worked better than the event-based repegging policy 2 (EBRP2). This shows that it is better to include more lots and orders for repegging. It was found that repegging was performed 1.3 and 2.6 times a day in scenarios 2 and 3, respectively, under the event-based repegging policies (the frequency was 3 times a day in the periodic repegging policy).

Among the repegging algorithms, MES/RW2 consistently worked better than the other two algorithms. Note that in MES/RW2, lots with similar progresses tend to be assigned to the same order because of the penalty term, $\alpha \cdot h_i / \{P_i \cdot (n_i + 1)\}$. In addition, results of different scenarios show that the advantage of using the soft pegging strategy becomes more apparent as there are more urgent orders. In all scenarios, repegging policies PRP and EBRP1 with pegging algorithm MES/RW2 outperformed other policies and pegging algorithms.

To see the effect of scheduling rules on the performance, we performed another series of tests. In this series of tests, we compare results of using the repegging policies with ES/RW2 and PUCH and those with scheduling rules used in the real fab considered in this study. In the real fab, the earliest due date (EDD) rule is used at serial workstations and the EDD rule and the minimum batch size (MBS) rule are used at batch-processing workstations. Under the EDD rule, lots with earlier due dates have higher priorities. Under the MBS rule, lots are processed on a batch-processing machine only if the number of lots available for being processed is no less than the minimum batch size (*MBS*), which is prespecified for each product family and machine. If the number of available lots is less than *MBS*, processing for the lots is deferred until the number of available lots becomes equal to *MBS*. The *MBS* was set to 3 (after tests on a few candidate values) in the tests. For all repegging policies, MES/RW2 was used as the pegging algorithm. Results of the tests are given in Table II.

TABLE II
PERFORMANCE OF SCHEDULING RULES

	EDD	ES/RW2
	EDD+MBS	PUCH
Hard Pegging	-	54.4% (7.6)†
PRP	26.3 (7.4)	75.5 (12.6)
EBRP1	26.2 (9.1)	73.7 (11.3)
EBRP2	19.4 (8.1)	68.5 (10.4)

† Average and standard deviation (in parenthesis) of the percentage reduction from the total tardiness obtained from the hard pegging strategy with EDD and EDD+MBS used for scheduling.

TABLE III
PERFORMANCE DIFFERENCE ACCORDING TO REPEGGING DURATION

Interval between re-pegging	Scenario 1	Scenario 2	Scenario 3
8 hours	48.7 (11.2) †	56.5 (13.7)	64.9 (12.3)
24 hours	44.6 (10.4)	53.1 (11.7)	61.4 (12.1)
48 hours	36.1 (12.5)	44.5 (12.2)	54.3 (12.7)

† Average and standard deviation (in parenthesis) of the percentage reduction from the total tardiness obtained from the hard pegging strategy with ES/RW2 and PUCH used for scheduling

As shown in the table, the scheduling rules that were shown to work better than other research, i.e., ES/RW2 and PUCH, gave significantly better results than the scheduling rules that are currently used in the wafer fab under consideration, EDD and EDD+MBS, for each pegging policy. This result shows that scheduling methods are also very important since lots (assigned to the orders) must be processed with the scheduling methods until the next repegging decision is made. In this table as well, we can see PRP and EBRP1 consistently worked better than EBRP2 and the hard pegging strategy.

Finally, to see the effect of intervals between repegging on the performance of the periodic repegging policy, we performed another series of experiments, in which ES/RW2 and PUCH were used for scheduling and MES/RW2 was used for repegging. In this test, we compare three cases, in which 8 h, 24 h, and 48 h are used as the intervals. Table III shows results of this test. As expected, better results were obtained from more frequent repegging. This tells that we need to reassign lots to orders according to changed states of the lots, orders and the fab more frequently, if the shop floor management technology of the system allows such frequent repegging. However, the rate of improvement diminishes as the interval decreases. Note that there was a greater change in the performance when the repegging interval was decreased from 48 to 24 h (to a half) than when it was decreased from 24 to 8 h (to a third).

As can be seen from the results given above, the soft pegging strategy with the suggested pegging policies outperforms the hard pegging strategy. Especially, the periodic repegging policy with frequent repegging works very well regardless of repegging algorithms (used for lot-order reassignments) and scheduling rules (used for scheduling wafer lots in the fab). Also, ES/RW2 and PUCH work very well in terms of total tardiness in this system for all pegging policies used for the system. We can argue that by using good pegging policies and good scheduling rules we can significantly improve scheduling performance of the wafer fab.

V. CONCLUDING REMARKS

In this paper, we evaluated performance of the soft pegging strategy, under which the assignment of lots to orders can be changed during the production period. We identified three pegging policies for the soft pegging strategy, by which we can reassign lots to orders considering progresses and due dates of wafer lots and orders, and developed three (re-)pegging algorithms for the policies. These policies and algorithms were tested for order management in a semiconductor fabrication facility. Results of simulation experiments showed the soft pegging strategy gave better results than the hard pegging strategy in terms of tardiness of the orders. From the experiments, it is also found that better results can be obtained if lots are reassigned to orders more often and if more lots and orders are considered for repegging.

The repegging policies suggested in this study can be easily adopted in many other semiconductor wafer fabs since only the management policy needs to be changed and a relatively simple software is needed to implement the policy. The soft pegging strategy operated with repegging policies and algorithms may be considered as an effective and viable tool for handling disturbances of the system, such as machine breakdowns and arrival of urgent orders, in addition to the strategy of rescheduling wafer lots, which is commonly used in most wafer fabs. An extensive study may be needed on the relationships between scheduling methods and repegging methods, since the performance of the system depends not only on the repegging methods but also on the scheduling methods. Also, a further study may be needed on triggering conditions for event-based repegging policies, such as machine breakdowns and sudden drops in yield rates.

REFERENCES

[1] L. M. Wein, "Scheduling semiconductor wafer fabrication," *IEEE Trans. Semiconduct. Manuf.*, vol. 1, pp. 115–130, Aug. 1988.

[2] C. R. Glassey and M. G. C. Resende, "A scheduling rule for job release in semiconductor fabrication," *Oper. Res. Lett.*, vol. 7, pp. 213–217, Oct. 1988.

[3] Y.-D. Kim, D.-H. Lee, J.-U. Kim, and H.-K. Roh, "A simulation study on lot release control, mask scheduling, and batch scheduling in semiconductor wafer fabrication facilities," *J. Manufact. Syst.*, vol. 17, pp. 107–117, Mar. 1998.

[4] S. C. Graves, H. C. Meal, D. Stefek, and A. H. Zeghmi, "Scheduling of re-entrant flow shops," *J. Oper. Manage.*, vol. 3, pp. 197–207, Aug. 1983.

[5] S. X. C. Lou and P. W. Karger, "A robust production control policy for VLSI wafer fabrication," *IEEE Trans. Semiconduct. Manuf.*, vol. 2, pp. 159–164, Nov. 1989.

[6] K.-H. Lee, Y. Hong, and S.-Y. Kim, "Production scheduling in semiconductor wafer fabrication process," *J. Korean Inst. Ind. Eng.*, vol. 21, pp. 357–369, Sep. 1995.

[7] H. S. Min and Y. Yih, "Selection of dispatching rules on multiple dispatching decision points in real-time scheduling of a semiconductor wafer fabrication system," *Int. J. Prod. Res.*, vol. 41, pp. 3921–3941, Nov. 2003.

[8] H. J. Yoon and D. Y. Lee, "Deadlock-free scheduling of photolithography equipment in semiconductor fabrication," *IEEE Trans. Semiconduct. Manuf.*, vol. 17, pp. 42–54, Feb. 2004.

[9] J. T. Lin, F. K. Wang, and P. C. Kuo, "A parameterized-dispatching rule for a logic IC sort in a wafer fabrication," *Prod. Plan. Contr.*, vol. 16, pp. 426–436, Jul. 2005.

[10] C. R. Glassey and W. W. Weng, "Dynamic batching heuristic for simultaneous processing," *IEEE Trans. Semiconduct. Manuf.*, vol. 4, pp. 77–82, May 1991.

[11] J. Fowler, G. L. Hogg, and D. T. Phillips, "Control of multiporduct bulk service diffusion/oxidation processes," *IIE Trans.*, vol. 24, pp. 84–96, Sep. 1992.

[12] J. K. Robinson, J. Fowler, and J. F. Bard, "Use of upstream and downstream information in scheduling semiconductor batch operations," *Int. J. Prod. Res.*, vol. 33, pp. 1849–1869, Jul. 1995.

[13] J. Fowler, G. L. Hogg, and D. T. Phillips, "Control of multiproduct bulk server diffusion/oxidation processes. Part 2: Multiple severs," *IIE Trans.*, vol. 32, pp. 167–176, Feb. 2000.

[14] Y.-D. Kim, J.-U. Kim, S.-K. Lim, and H.-B. Jun, "Due-date based scheduling and control policies in a multiproduct semiconductor wafer fabrication facility," *IEEE Trans. Semiconduct. Manuf.*, vol. 11, pp. 155–164, Feb. 1998.

[15] Y.-D. Kim, J.-G. Kim, B. Choi, and H.-U. Kim, "Production scheduling in a semiconductor wafer fabrication facility producing multiple product types with distinct due dates," *IEEE Trans. Robot. Automat.*, vol. 17, pp. 598–598, Oct. 2001.

[16] Y.-D. Kim, S.-O. Shim, B. Choi, and H. Hwang, "Simplification methods for accelerating simulation-based real-time scheduling in a semiconductor wafer fabrication facility," *IEEE Trans. Semiconduct. Manuf.*, vol. 16, pp. 290–298, May 2003.

[17] V. Jain, R. Swarnkar, and M. K. Tiwari, "Modelling and analysis of wafer fabrication scheduling via generalized stochastic Petri net and simulated annealing," *Int. J. Prod. Res.*, vol. 41, pp. 3501–3527, Jan. 2003.

[18] S. J. Mason, S. Jin, and C. M. Wessels, "Rescheduling strategies for minimizing total weighted tardiness in complex job shops," *Int. J. Prod. Res.*, vol. 42, pp. 613–628, Feb. 2004.

[19] G. Steiner and J. S. Yeomans, "Optimal level schedules in mixed-model, multi-level JIT assembly systems with pegging," *Eur. J. Oper. Res.*, vol. 95, pp. 38–52, Nov. 1996.

[20] K. Knutson, K. Kempf, and J. Fowler, "Lot-to-order matching for a semiconductor assembly and test facility," *IIE Trans.*, vol. 31, pp. 1103–1111, Nov. 1999.

[21] J. Fowler, K. Knutson, and M. Carlyle, "Comparison and evaluation of lot-to-order matching policies for a semiconductor assembly and test facility," *Int. J. Prod. Res.*, vol. 38, pp. 1841–1853, May 2000.

[22] M. Carlyle, K. Knutson, and J. Fowler, "Bin covering algorithms in the second stage of the lot to order matching problem," *J. Oper. Res. Soc.*, vol. 52, pp. 1232–1243, Nov. 2001.

[23] T. W. Wu, "Modular demand and supply pegging mechanism for semiconductor foundry," in *Proc. IEEE Int. Symposium on Semicond. Manuf.*, 2003, pp. 325–328.

[24] J.-Y. Bang, K.-Y. An, Y.-D. Kim, and S.-K. Lim, "A due-date based algorithm for order-lot pegging in a semiconductor wafer fabrication facility," in *Proc. 3rd Int. Conf. on Modeling and Analysis of Semicond. Manuf.*, 2005, pp. 175–180.

[25] M. R. Garey, D. S. Johnson, and R. Sethi, "The complexity of flowshop and jobshop scheduling," *Math. Oper. Res.*, vol. 1, pp. 117–129, May 1976.

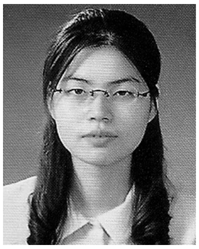
Yeong-Dae Kim (M'06) received the B.S. degree from Seoul National University, Seoul, Korea, the M.S. degree from the Korea Advanced Institute of Science and Technology (KAIST), Daejon, both in industrial engineering, and the Ph.D. degree from the University of Michigan, Ann Arbor, in industrial and operations engineering.

He is a Professor with the Department of Industrial Engineering, KAIST. His research areas include design and operation of manufacturing systems, operations scheduling and production planning, and inventory management.

Dr. Kim is a senior member of IIE and a member of INFORMS and the Operational Research Society.

June-Young Bang received the B.S. and M.S. degrees in mechanical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejon, where he pursuing the Ph.D. degree in the Department of Industrial Engineering.

His research areas include operations scheduling and production control.



Kwee-Yeon An received the B.S. and M.S. degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejon, both in industrial engineering.

Currently, she is with Samsung Electronics Co., Gyeonggi-Do, Korea, working as a Researcher on operational problems in semiconductor manufacturing systems. Her research interests include production planning and scheduling in semiconductor manufacturing systems and simulation.

Seung-Kil Lim received the B.S. degree from Pusan National University, Pusan, Korea, and the M.S. and Ph.D. degrees in industrial engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejon.

He is an Assistant Professor, Division of e-business Information Technology, Sungkyul University, Gyeonggi-Do, Korea. He worked for IBM Business Consulting Service as a Principal Consultant in the area of supply chain management for semiconductor manufacturing. His research interests include optimization of supply chain design and operation, production planning and control for semiconductor manufacturing, and business process reengineering with enterprise business applications.