Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

A REVIEW OF SCHEDULING THEORY AND METHODS FOR SEMICONDUCTOR MANUFACTURING CLUSTER TOOLS

Tae-Eog Lee

Dept. of Industrial and Systems Engineering
335 Gwahang-no KAIST
Yusung-gu, Deajeon, KOREA

Abstract

Cluster tools, which combine several single-wafer processing modules with wafer handling robots in a closed environment, have been increasingly used for most wafer fabrication processes. We review tool architectures, operational issues, and scheduling requirements. We then explain recent progress in tool science and engineering for scheduling and control of cluster tools.

1 INTRODUCTION

The semiconductor manufacturing industry has made continual innovations in wafer processing tools. One of the most important recent innovations is clustering several processing modules for single-wafer processing(SWP). The basic material handling unit in a fab is a wafer cassette that consists of 25 wafers. Most wafer fabrication processes are chemical processes and hence traditionally most tools processed wafers of a cassette in batch to maximize the throughput. However, as the wafer size increases and quality requirements become stricter due to circuit shrinkage, the batch processing technology becomes difficult to assure wafer quality. It is because it is hard to control uniformity of gas or chemical diffusion on all large wafer surfaces. Therefore, most fabrication processes had extensively adopted SWP technology that processes wafers one by one in a chamber. When SWP technology is used, wafer transfer tasks between SWP chambers can be excessive. Therefore, in order to reduce unloading, storing, moving, and loading tasks for individual wafers, several SWP chambers for a number of subsequent process steps are combined into a single tool so that wafers flow through the chambers one by one without going out from the tool. *Cluster tools* combines several SWP chambers within a closed environment together with a wafer handling robot. Cluster tools or track equipment have been increasingly used for most processes, including photolithography, etching, deposition, and even testing.

In this paper, we first review tool architectures, operational issues, and scheduling requirements. We then explain recent progress in tool science and engineering for scheduling and control of cluster tools. For cyclic scheduling, we explain notions of schedule quality, minimizing the tool cycle time, controlling wafer delays within a processing chamber, and concepts of workload balancing for reducing wafer delays. For non-cyclic scheduling, we review dispatching rules for cluster tool scheduling. Finally, we explain control software architecture for scheduling and control.

2 CLUSTER TOOL ARCHITECTURES, OPERATION, AND SCHEDULING REQUIREMENTS

2.1 Basic Cluster Tool Architecture and Operation

Cluster tools have different architectures as illustrated in Fig. 1. Most tools have radial configurations of chambers so that the robot move times between chambers are minimized and uniform. Linear configurations can flexibly add or remove chambers, but have longer non-uniform move times. Due to space restriction, a usual radial-type tool can afford no more than six chambers. Chambers are often called *process modules(PMs)*. A tool mostly has a single wafer-handling robot. The robot may have a single arm or dual arms. The angle between the dual arms is fixed to keep opposite positions. Dual-armed tools are known to have higher throughput than single-armed tools (Venkatesh et al. 1997), but have higher cost. A tool usually has two loadlocks. A new wafer cassette is loaded into the cassette at a loadlock. For convenience, we call a cassette at a loadlock just a loadlock.

Wafers in a cassette mostly require identical recipes. Wafers in a loaded cassette are unloaded and loaded into a chamber one by one. Wafers undergo a sequence of process steps. A process step that takes long may be performed by more than one chamber, which are regarded as parallel chambers. Therefore, common wafer flow patterns through

978-1-4244-2708-6/08/$25.00 ©2008 IEEE

chambers are series-parallel. Once a wafer completes all process steps, it is returned to the loadlock. Once the last wafer in the cassette at a loadlock is loaded into a chamber, wafers are next loaded from another loadlock. Therefore, a tool can keep processing identical wafers as long as identical wafer cassettes are supplied to the tool.

A tool often has an aligner, just after a loadlock. After a wafer is unloaded by a robot from a cassette at a loadlock, its position on a robot arm may be misaligned due to mechanical accuracy problems of the cassette, the loadlock, or the robot arms. When the robot try to load a misaligned wafer into a chamber, the wafer may collide with the chamber's slot and be damaged or dropped down. Therefore, a wafer unloaded from a loadlock is aligned at an aligner with help of a laser beam. The aligning time is just five to ten seconds. A tool may have a cooler module, where a hot wafer unloaded from the last chamber is cooled down before it is returned to the loadlock. It is because a hot wafer may affect other wafers in the loadlock.

In a cluster tool, there is no intermediate buffer between the chambers due to space restriction. In some case, a chamber that is not used for wafer processing can be used as a buffer, where a wafer can stay temporarily until a robot becomes available (Kim et al. 2003, Rostami and Hamidzadeh 2004). However, this increases the robot workload and cools down a wafer excessively before it is loaded into the next chamber.

2.2 Advanced Cluster Tool Architectures and Operation

Chambers repeat vacuuming and venting cycles to process loaded wafers. Some tools have intermediate vacuuming buffers between chambers and loadlocks while keeping the whole area for chambers vacuum so that chambers immediately start processing without vacuuming and venting cycles (Paek and Lee 2002).

As seen in Fig. 1(b), some new cluster tools use multiple wafer slots in a chamber in order to improve the throughput higher than SWP tools by processing several wafers together (Jung 2006). However, those new tool architectures tend to increase scheduling complexity significantly.

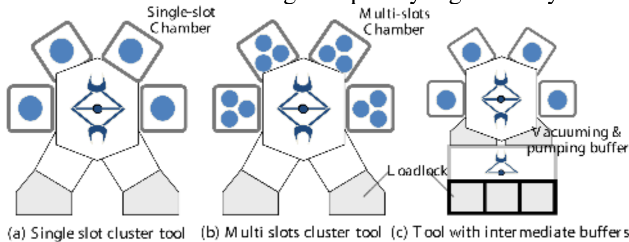

Figure 1: Tool architectures.

An integrated system of SWP chambers with multiple handling robots is often called a *track tool* or *track system*. Photolithography processes use track systems that supply steppers with wafers coated by photo-sensitive chemicals and develop the circuit patterns on the wafers that are formed by exposures to circuit pattern picture images at the steppers. Process modules for coating and developing, and accompanying baking and cooling modules are combined into a track tool with several robots as illustrated in Fig. 2. A process step can have five to ten number of parallel modules (Oh 2000, Yoon and Lee 1999), which are vertically stacked up.

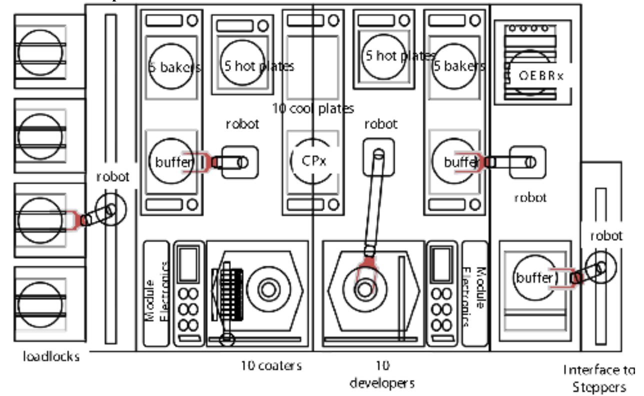

Figure 2: A track system.

An automated wet station also has a series of chemical and rinsing baths for cleaning wafer surfaces, which are combined by several robots moving on a rail (Lee, Lee, and Lee 2007). Recently, EDS processes for testing devices on wafers are automated to form a kind of track system. A number of testing tools for WBI(Wafer Burn-In) test, hot pre-test, cold pre-test, laser repair, and post-test are configured in series-parallel. Loading and unloading tasks are served by several robots moving on a rail. Such a robot is also called *rail-guided vehicle(RGV)*. RGVs are faster and has less vibration than AGVs(Automatic Guided Vehicles). A robot serves several baths within a zone of the rail. The zones are overlapped at a bath, where a cassette is transferred from a zone to the next zone. There is no intermediate buffers between the baths. Therefore, there can be robot collisions or deadlocks.

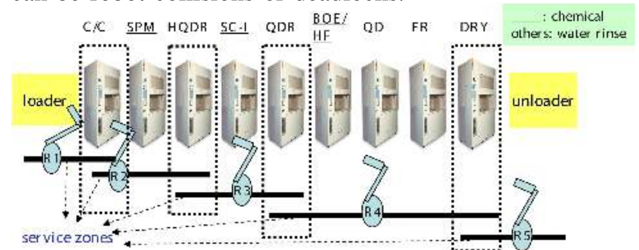

Figure 3: A wet station.

Several cluster tools are often connected through an intermediate buffer, through which wafers are directly transferred from a tool to another tool (Lopez and Wood 1998, Zuberek 2001a, Lopez and Wood 2003, Yi, Ding, and Song 2005, Ding, Yi, and Zhang 2006). Such multi-cluster tools are intended for reducing delays between tools.

The logical configurations of track equipment, wet stations, and multi-cluster tools are similar. They all can be viewed as a set of cluster tools that are serially connected. In track equipment, the chambers that are served by a robot can be viewed as a cluster tool. In a wet station, the baths that are served by a robot in a zone also can be regarded as a cluster tool. In most radial-type cluster tools, the robot moves only rotationally while a robot in a wet station should make horizontal moves. Horizontal moves can be slower than rotational moves and have significantly different move times depending on the source and destination of the move task. In track equipment, a robot should make both rotational and vertical moves due to vertically stacked parallel chambers. Those different robot moving patterns have different scheduling implications.

2.3 Tool Scheduling Requirements

Wafers mostly go through a sequence of process steps in series. When a process step has multiple chambers, one of them is used for a specific wafer. There can be different ways of using parallel chambers. They can be used randomly or a chamber that becomes available first may be taken by a wafer. Alternatively, the parallel chambers may be cyclically used. Since the parallel chambers for a process step perform an identical process step, they can be considered identical for scheduling as long as the move times to the chambers are identical. In a radial cluster tool, the moves time between chambers are small, not significantly different, and hence regarded as identical for most scheduling models. However, tools with horizontally or vertically configured chambers, such as wet stations and track equipment, should consider different move times between baths or chambers (Oh 2000).

Wafer flows in a cluster tool may look similar to conventional flow lines or assembly lines. However, there is no intermediate buffers between chambers and the moves are restricted by the availability of the robot arms. A wafer cannot go back to the cassette in a loadlock because a hot wafer with chemicals on the surface may affect other wafers in the cassette and it should not be excessively cooled down before starting the next process step. A wafer loaded into a chamber immediately starts processing since the chamber already has gases and heat.

For some processes, wafers visit some process steps again. For instance, unlike conventional chemical vapor deposition, atomic layer deposition process controls the deposition thickness by repeating extremely thin deposition multiple times. Therefore, a wafer reenters the chambers many times. For a reentrant chamber, the processing order between the reentered wafers and the first entering wafer should be determined. When a reentrant process step has parallel chambers, there can be different strategies of using the chambers (Lee and Lee 2006).

In some processes, a chamber should be cleaned after a specified number of wafers are processed or when the sensors within a chamber detect significant contamination (Jung 2006). Most fabs clean chambers periodically. Chamber cleaning is also a job to be scheduled.

A wafer can remain in a chamber after processing until it is unloaded by a robot. This wafer waiting time is called *wafer delay* or *wafer residency time*. A wafer waiting within a chamber is subject to surface quality problems due to residual gases and heat within the chamber. Some processes, for instance, low pressure chemical vapor deposition (LPCVD) which uses high temperature instead of high pressure, have strict wafer delay constraints. When the wafer delay exceeds an upper limit, the wafer is scrapped. In a wet station, a wafer cleaned at a chemical bath should be immediately unloaded and rinsed at a water bath (Lee, Lee, and Lee 2007). Coating and developing processes for photolithography also can have similar strict time constraints (Oh 2000). Even for most other processes, wafer delays are not desirable. They should be reduced, eliminated, or regulated to be constant for better or uniform wafer surface quality.

Process times or tasks times are rather constant and can be regarded as deterministic in most scheduling models. However, they can be subject to random variation, mostly within a few percent. There can be exceptional delay, even rare, due to abnormal process conditions. A wafer alignment task sometimes fails and are retried. There should be discretion whether the scheduling model assumes deterministic process and task times without random variation or considers exceptional delays or stochastic times. Complexity and difficulty in modeling and analysis and the degree of randomness should be considered. Deterministic models can give simpler analysis and better insights, from which we may develop a method of handling random variation for practical implementation. Stochastic models may be more realistic, but analysis and scheduling optimization may be limited and often require model simplifications such as Markovian or exponential distribution assumptions by sacrificing the reality. Simulation might provide a realistic model, but has limitations in identifying causal or parametric relationships.

Integrated tools mostly limit intermediate buffers. Therefore, blocking and waiting are common and even deadlocks can occur. Reentrance, wafer delays, cleaning cycles, and uncertainty all increase scheduling complexity significantly. Tool productivity by intelligent scheduling and control is critical for maximizing the fab productivity.

3 SCHEDULING STRATEGIES FOR CLUSTER TOOLS

There can be alternative scheduling strategies for cluster tools. First, a dispatching rule determines the next robot task depending on the tool state. It can be considered dynamic and real-time. However, it is hard to optimize the rule. We are only able to compare performances of heuristically designed dispatching rules by computer simulation. Second, a schedule can be determined in advance. This method can optimize the performance if a proper scheduling model can be defined. When there is a significant change in the tool situation, rescheduling is made. However, scheduling complexity due to the number of jobs, the number of process steps, and the number of distinct robot tasks may limit the computational time and optimality of a scheduling algorithm. *Cyclic scheduling* makes each robot and each chamber repeat identical work cycles (Lee and Posner 1998, Lee, Lee, and Lee 2007). Once the robot task sequence is determined, all work cycles are determined. Most academic works on cluster tool scheduling consider cyclic scheduling. Cyclic scheduling has merits such as reduced scheduling complexity, predictable behavior, improved throughput, steady or periodical timing patterns, regulated or bounded task delays or wafer delays and work-in-progress, and reduced variation of wafer flow times (Lee and Posner 1998, Oh 2000, Lee 2000b, Lee, Lee, and Lee 2007). In cyclic scheduling, the timings of tasks can be controlled in real-time depending on the associated event occurrences while the sequence or work cycle is predetermined.

4 CYCLIC SCHEDULING

4.1 Modeling Cluster Tool Behavior

Operational behaviors of cluster tools can be well modeled by Petri nets. A Petri net is a graphical and mathematical modeling framework for discrete event systems (Murata 1989). Mathematical analysis for cycle time computation and logical properties such as existences of potential deadlocks can be performed depending on the class of Petri nets. Transitions, places, arcs, and tokens usually represent activities or events, conditions or activities, precedence relations between transitions and places, and entities or conditions, respectively. They are graphically represented by rectangles, circles, arrows, and dots, respectively.

A cluster tool that repeats identical work cycles by cyclic scheduling can be modeled as a timed event graph(TEG), a class of Petri nets (Murata 1989), where each place has only one input and output transitions. That is, once a robot task sequence is given, a TEG is defined. An example of TEG model for dual-armed cluster tools is given in Fig.4. Once a TEG model is made, the tool cycle time, the optimal robot task sequence, the wafer delays, and the optimal timing schedules can be systematically identified (Lee 2000b, Lee, Lee, and Lee 2007). The tool cycle time is the maximum of the circuit ratios in the TEG model, where the *circuit ratio* of a circuit is the sum of the total times in the circuit to the number of the tokens in the circuit.

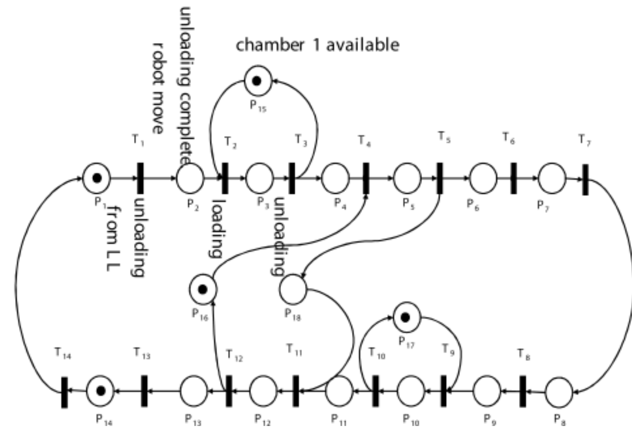

Figure 4: A timed event graph model for a dual-armed cluster tool.

There are Petri net models for tools with reentrant wafer flows (Zuberek 2004, Lee and Lee 2006), tools with intermediate buffers (Paek and Lee 2002), wet stations (Lee, Lee, and Lee 2007), tools with multi-slot chambers (Jung 2006), and tools with cleaning cycles (Kim 2006, Lee 2008). There are different Petri net modeling practices, which may use different class of Petri nets such as asymmetric Petri nets or colored Petri nets (Zuberek 2001b, Zuberek 2004, Wu and Zhou 2007b, Wu and Zhou 2007a).

4.2 Schedule Quality

For a cluster tool with a given cyclic sequence, there can be different classes of schedules, each of which corresponds to a firing schedule of the TEG model. A periodic schedule repeats an identical timing pattern for each d work cycles. When $d = 1$, the schedule is called *steady*. In a steady schedule, the task delays such as wafer delays are all constant. In a d-periodic schedule, the wafer delays have d different values, while the average is the same as that of a steady schedule. The period d is determined from the TEG model. A schedule that starts each task as soon as the preceding ones complete is called *earliest*. An earliest schedule can be generated by the earliest firing rule of the TEG model that fires each transition as soon as it is enabled. In other words, an earliest starting schedule need not be generated and stored in advance. The TEG model with the earliest firing rule can be used as a real-time scheduler or controller for the tool. Therefore, an earliest schedule can be implemented by an event-based control, which initiates a task when an appropriate event, for instance, a task comple-

tion, occurs. Therefore, an earliest starting schedule based on such event-based control has merits (Lee 2000b, Shin, Lee, Kim, and Lee 2001, Lee, Lee, and Lee 2007). First, potential logical errors due to message sequence changes can be prevented. When a tool is controlled by a pre-determined timing schedule, communication or computing delays may cause a change in a message sequence and hence a critical logical error. For instance, a robot may try to unload a wafer at a chamber before processing at the chamber is not completed and hence when the wafer slot is still closed. Second, the earliest schedule minimizes the average tool cycle time. Therefore, the most desirable schedule is a *steady and earliest starting schedule(SESS)*. For a cluster tool with cyclic operation, there always exists a SESS. Fig.5 illustrates an example of SESS for the TEG model. A SESS can be computed in advance using the max-plus algebra or a kind of longest path algorithms (Lee 2000b) and implemented by an event-based controller based on the TEG model (Shin, Lee, Kim, and Lee 2001, Lee, Lee, and Lee 2007).

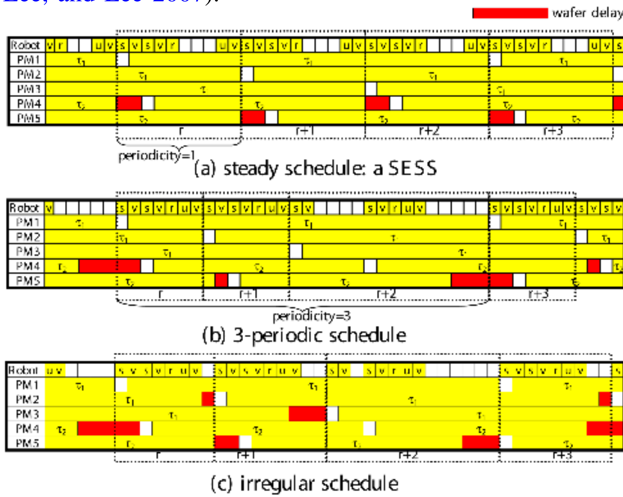

Figure 5: Examples of schedules.

4.3 Cycle Time Optimization

For basic cluster tool models, there are robot task sequences that are popularly used in the industry and known to have good performance. They are the *backward* sequence for single-armed tools and the *swap* sequence for dual-armed tools. For a single-armed tool, the backward sequence moves a wafer from a chamber of the last process step n to the loadlock, then moves a wafer from a chamber of process step $n-1$ to a chamber of the last process step, and continues similar wafer transfer for all the preceding process steps. To process step 1, a wafer is moved from the loadlock. After this, the robot repeats the wafer move from the last process step to the loadlock. Therefore, the robot performs the backward sequence of wafer moves cyclically. For a dual-armed tool, the robot exchanges the wafer on a robot arm with the wafer processed at a chamber of a process step i by using the two arms. Then, this swap operation is performed for a chamber of the process step $i+1$ that the wafer on a robot arm should visit next. Although the backward sequence and the swap sequence are generally believed to be optimal for single-armed tools and dual-armed tools, respectively, the optimality was not completely proved. For 3-chamber dual-armed tools, a full enumeration of all possible robot task sequences indicates that the swap sequence has the minimum cycle time (Sethi, Sidney, and Sriskandarajah 2001). However, it is not known for dual-armed tools with more than three chambers. However, it is recently proved that the swap sequence is optimal for dual-armed tools (Paek and Lee 2008). The key idea is to show that the tool cycle time for the swap sequence is the same as the workload of a chamber work cycle or a robot work cycle. The tool cycle time cannot be shorten to less than the workload. The same idea can be easily applied to show that the backward sequence is optimal for a single-armed tool, where the tool cycle is also identified as the maximum of the workloads of the work cycles of the chambers and the robot (Shin, Lee, Kim, and Lee 2001).

The swap or backward sequences may not be optimal when there are additional scheduling constraints such as wafer delay constraints or reentrant wafer flows or the tool has an advanced architecture other than a basic radial architecture with single-slot chambers. In the cases, we should have a way of determining an optimal sequence. The scheduling problem for a cluster tool with cyclic operation can be modeled by a mixed integer programming (MIP) model, for which there have numerous solution methods. The key idea is to develop a linear programming (LP) model to determine a steady schedule, that is, the timings of each tasks, for a TEG for a tool with a given robot task sequence, and then extend the LP model to an MIP model for determining the robot task sequence (Lee 2000b, Seo and Lee 2002). The MIP model can be extended for tools with reentrant wafer flows (Lee and Lee 2006), tools with wafer delay constraints (Kim, Lee, Lee, and Park 2003, Lee and Park 2005), and wet stations or track equipment (Lee, Lee, and Lee 2007). There is an alternative way of modeling MIP models for cluster tool scheduling, which reduces the search space of the robot task sequences, is based on an assignment problem, and hence is significantly more efficient than previous MIP modeling methods based on a Hamiltonian circuit search problem (Jung and Lee 2008).

In a dual-armed tool with advanced scheduling requirements such as time constraints or reentrant wafer flows, the swap-based sequence may not be optimal or feasible. The swap operation itself restricts operation of the two arms. Therefore, the two arms need to be more flexibly used although they cannot concurrently perform a task. For in-

stance, the robot may unload a wafer from a chamber to a robot arm and then load the wafer on another arm to other chamber instead of the chamber. We therefore should consider scheduling robot tasks without swap-restriction. Recently, a Petri net model and an MIP model to determine an optimal robot task sequence for dual-armed tools without swap-restriction are developed (Paek and Lee 2008).

4.4 Controlling Wafer Delays

4.4.1 Schedulability Analysis

There have been works on schedulability of a cluster tool or a robot cell, that is, existence of a feasible schedule (Calvez et al. 1997, Chen et al. 1998, Lei and Liu 2001, Rostami et al. 2001, Rostami and Hamidzadeh 2002, Rostami and Hamidzadeh 2004, Lee 2000a, Kim et al. 2003, Shin 2002, Lee and Park 2005, Kim and Lee 2002, Kim and Lee 2008). There are a necessary and sufficient condition for schedulability, that is, existence of a feasible schedule, based on circuits in an extended version of TEG, NEG, and a procedure of computing a feasible SESS (Lee and Park 2005). In fact, schedulability also can be verified by existence of a feasible solution in an associated LP. However, the necessary and sufficient condition identifies why the time constraints are violated, and often gives a closed form schedulability condition based on the scheduling parameters such as the process times, the robot task times, and the number of parallel chambers of each process step. For instance, the feasible range of the process times for a dual-armed cluster tool is $u+(n+1)v+ns+r \leq \min_{1\leq i\leq n} \frac{t_i+d_i+s}{m_i}$ and $\max_{1\leq i\leq n}\{\frac{t_i+s}{m_i}\} \leq \min_{1\leq i\leq n}\{\frac{t_i+d_i+s}{m_i}\}$. The scheduling method based a NEG can be used for other cluster tools or automated systems with cyclic operation that have time constraints. Even though there is time variation in process times and robot task times, once their varying ranges can be identified as a finite interval, the schedulability can be efficiently verified (Kim and Lee 2002, Kim and Lee 2008). This is useful for ensuring that the tool schedule is safe against time variation.

4.4.2 Wafer Delay Identification

When the initial timings are not appropriately controlled or a SESS is disrupted, the earliest schedule converges to a periodic schedule of which period is determined from the TEG (Kim and Lee 2003). Therefore, the wafer delays can be much larger than the constant value for a SESS. For a given wafer delay constraint, even if the schedulability condition is satisfied, that is, a feasible SESS exists, a periodic schedule may have wafer delays exceeding the limit. Therefore, we are concerned with whether such a periodic schedule with fluctuating wafer delays can satisfy the wafer delay constraint. There is a systematic method of identifying exact values of task delays or token delays at places of a TEG or wafer delays of a cluster tool for each type of schedule, steady or periodic, earliest or not (Lee, Sreenivas, and Lee 2006). Therefore, the worst-case wafer delay against time disruptions can be computed by identifying token delays for periodic schedules (Lee, Sreenivas, and Lee 2006).

4.5 Regulating Wafer Delays: Schedule Stability

Most schedulability analyses assume deterministic process and task times. When a cluster tool is operated by a SESS, the wafer delays are kept constant. However, in reality, there can be sporadic random disruptions such as wafer alignment failures and retrials or exceptional process times. In the case, the schedule is disturbed to a non-SESS, where the wafer delays fluctuate and may exceed the specified limits. However, there are regulating methods that make a disrupted schedule restored quickly. We have a *schedule stability* condition for which a disrupted earliest firing schedule of a TEG or a cluster tool converges to the original SESS regardless of the disruption size and a simple way of enforcing such stability by adding an appropriate delay to some selected tasks (Kim and Lee 2003). Therefore, we can regulate the wafer delays to be constant. Such stability control method is proven to be effective even when there are realistic persistent time variation, a few percent (Kim and Lee 2003, Kim and Lee 2002, Kim and Lee 2008).

4.6 Reducing Wafer Delays: Workload Balancing for Tools

In a traditional flow line or shop, the workload of a process step is the sum of the process times of all jobs for the step. The bottleneck is the process step with the maximum workload. Imbalance in the workloads of the process steps cause waiting of the jobs or work-in-progress before the bottleneck. However, in automated manufacturing systems such as cluster tools, the workload is not easy to define because the material handling system interferes with the job processing cycle. To extend the workload definition, we can define the *generalized workload* for a resource as the circuit ratio for the circuit in the TEG that corresponds to the work cycle of the resource (Lee, Lee, and Shin 2004, Lee, Lee, and Lee 2007). For instance, the workload for a chamber at process step i with m_i parallel chambers in a single-armed tool is $\frac{p_i+2l+2u+3v}{m_i}$ because each work cycle of a chamber requires a wafer processing(p_i), two loading tasks($2l$), two unloading tasks($2u$), and three robot moves($3v$). A robot has workload $(n+1)(u+l+2v)$, the sum of all robot task times. Therefore, the overall tool cycle time is determined by the bottleneck resource as $\max\{\max_{k=1,2,\ldots,n} \frac{p_k+2l+2u+3v}{m_k}, (n+1)(u+l+2v)\}$. Imbalance between the workloads or circuit ratios causes task

delays such as wafer delays. In a single-armed tool, the workload imbalance between process step i's cycle and the whole tool cycle is $\max\{\max_{k=1,2,\ldots,n} \frac{p_k+2l+2u+3v}{m_k}, (n+1)(u+l+2v)\} - \frac{(p_i+2l+2u+3v)}{m_i}$. Notice that each chamber at process step i has cycle time $(p_i+2l+2u+3v)$ while the overall cycle time at the process step is $\frac{(p_i+2l+2u+3v)}{m_i}$. Therefore, the delay in each cycle of a chamber at process step i is m_i times as long as the workload imbalance at the process step. Consequently, the average wafer delay at a chamber at process step i is $m_i \max\{\max_{k=1,2,\ldots,n} \frac{p_k+2l+2u+3v}{m_k}, (n+1)(u+l+2v)\} - (p_i+2l+2u+3v)$ (Lee, Lee, and Lee 2007). We note that from the well-known queueing formula, Little's law, the average delay is proportional to the average work-in-progress. In a cluster tool, wafer delays are more important than the number of waiting wafers because of extreme limitation on the wafer waiting place. Wafer delays can be reduced or eliminated by balancing the circuit ratios. Such *generalized workload balancing* can be done by adding parallel chambers to a bottleneck process step, accommodating the process times within technologically feasible ranges, or delaying some robot tasks intentionally (Lee, Lee, and Lee 2007, Lee, Lee, and Shin 2004). We have a linear programming model that optimizes such workload balancing decisions under given restrictions (Lee, Lee, and Shin 2004, Lee, Lee, and Lee 2007). Workload balancing is essential for cluster tool engineering.

5 NON-CYCLIC SCHEDULING: DISPATCHING RULES

A dispatching rule dynamically determines the next robot task depending on the tool state, that is, the positions of the wafers, the expected residual times of the processes at the chambers, and the waiting times of wafers within chambers after processing. A simple dispatching rule may be to do the first requested robot task first. Many tool vendors are using dispatching rules, which may be developed or refined by experiments and experiences. Simulation models often can be used for developing and evaluating dispatching rules. Through simulation experiments or tool operation experiences, dispatching rules can be changed, improved, or tuned to have better performance. Dispatching rules should be used instead of cyclic scheduling rules or predetermined robot task sequences when the process times or robot robot task times are subject to excessive random variation. When the time variation is rather limited, cyclic scheduling that assumes deterministic times can give better performance and insights. It is also useful for controlling wafer delays. However, as the time variation increases, the performance of cyclic scheduling degrades and becomes worse than dispatching rules. Dispatching rules are also used when cyclic scheduling models are too large or too complicated to optimize the robot task sequence. The cases include tools with many repeated wafer flows, multi-slot chambers, or some chamber cleaning requirements (Perkinson, Gyurcsik, and McLarty 1996, Zuberek 2004, Lee and Lee 2006, Jung 2006, Kim 2006, Lee 2008). We note that cluster tools with cleaning cycles, multi-slots, and reentrance have more challenging scheduling problems. There are some works on using cyclic scheduling for the problems (Perkinson, Gyurcsik, and McLarty 1996, Jung 2006, Kim 2006, Lee and Lee 2006, Lee 2008).

Tool vendors often prefer dispatching rules because they are simpler than cyclic scheduling and easy to understand, and are expected to have better performance than predetermined static cyclic sequences due to their dynamic, real-time features. However, it is difficult to find an optimal dispatching rule. We only be able to design dispatching rules heuristically based on experiences and domain knowledge, and compare the rules by experiments or simulation and tune the parameters of the rules. Dispatching rules can be more robust than cyclic scheduling against unexpected event occurrences or exceptional time changes. However, when dispatching rules are used, it is difficult to predict the tool behavior without simulation, and wafer delays cannot be easily regulated or controlled. When time variations are relatively small and cyclic scheduling models can be easily defined and are not too complicated, cyclic scheduling can provide shorter cycle times based on optimization models and control wafer delays well by using the theoretical results on wafer delays explained above. Examples of dispatching rules can be found in (Yoon and Lee 2001, Lee 2008). Paek also provides a Petri net model for cluster tools and systematically defines diverse dispatching rules based on marking and token sojourn times at places in the net model (Paek and Lee 2008).

6 SCHEDULER IMPLEMENTATION

A scheduler should monitor the key events from each PMC and the TMC through the module manager. The events include starts and completions of wafer processing or robot tasks. Then, the scheduler determines the states of the modules and scheduling decisions as specified by the scheduling logic or rules, and issues the scheduling commands to the module manager. Therefore, the timings of tasks are controlled based on events rather than are determined in advance. The scheduler should be able to change the scheduling logic or rules flexibly to cope with wafer flow pattern changes or even tool configuration changes. Those requirements can be met by implementing the scheduling logic or rules by an extended finite state machine(EFSM) (Shin et al. 2001). An EFSM models state change of each module and embeds a short programming code for the scheduling logic or procedure. The scheduling logic also includes procedures for handling exceptions such as wafer alignment failures, processing chamber failures, robot arm failures, etc.

7 CONCLUSION

Cluster tools have unique scheduling requirements and challenges. The tool behavior can be well modeled and analyzed by a popular formal modeling method for discrete event systems, Petri nets. For most practical cases where time variations are not so serious, cyclic scheduling is desirable for optimizing the cycle time and controlling the wafer delays. There have been significant theoretical works in cycle time optimization and wafer delay control. Such tool science needs to be further implemented to significantly improve the productivity and quality of tools in production. Fabs as well as tool vendors will focus on tool productivity in the future and tool science and engineering will be their core competence.

Future fabs tend to integrate process tools or modules with automated material handling systems. Tight coupling between wafer processes and material handling leads to simultaneous scheduling of wafer processing tools and material handling devices. Cluster tool scheduling models and theory, which handle wafer processing and material handling in an integrated model, can be extended for such future automated fab systems.

REFERENCES

Calvez, S., P. Aygalinc, and W. Khansa. 1997, May. p-time Petri nets for manufacturing systems with staying time constraints. In *Proceedings of IFAC CIS '97 Conference on Control of Industrial Systems*, 495–500.

Chen, H., C. Chu, and J.-M. Proth. 1998. Cyclic scheduling of a hoist with time window constraints. *IEEE Transactions on Robotics and Automation* 14 (1): 144–152.

Ding, S., J. Yi, and M. T. Zhang. 2006. Multicluster tools scheduling: An integrated eventgraph and network model approach. *IEEE Transactions on Semiconductor Manufacturing* 19 (3): 339–351.

Jung, C. 2006. Steady state scheduling and modeling of multi-slot cluster tools. Master's thesis, Department of Industrial & Systems Engineering, KAIST, Daejon, Korea.

Jung, C., and T.-E. Lee. 2008. An efficient scheduling method based on an assignment model for robotized cluster tools. In *Proceedings of 2008 IEEE International Conference on Automation Science and Engineering*, 1–6.

Kim, H. J. 2006. Scheduling and control of dual-armed cluster tools with post processes. Master's thesis, Department of Industrial & Systems Engineering, KAIST, Daejon, Korea.

Kim, J.-H., and T.-E. Lee. 2002. Discrete event systems with bounded random time variation. In *Proceedings of Spring Joint Conference of KIIE and KORMS*, 1–7. Taejon, Korea.

Kim, J.-H., and T.-E. Lee. 2003, September. Schedule stabilization and robust timing control for time-constrained cluster tools. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1039–1044.

Kim, J.-H., and T.-E. Lee. 2008. Schedulability analysis of time-constrained cluster tools with bounded time variation by an extended petri net. *IEEE Transactions on Automation Science and Engineering* 5 (3): 490–503.

Kim, J.-H., T.-E. Lee, H.-Y. Lee, and D.-B. Park. 2003. Scheduling of dual-armed cluster tools with time constraints. *IEEE Transactions on Semiconductor Manufacturing* 16 (3): 521–534.

Lee, H.-Y. 2000a. Scheduling and determination of feasible process times for CVD cluster tools with a dual end effector. Master's thesis, Department of Industrial & Systems Engineering, KAIST, Daejon, Korea.

Lee, H.-Y., and T.-E. Lee. 2006. Scheduling single-armed cluster tools with reentrant wafer flows. *IEEE Transactions on Semiconductor Manufacturing* 19 (2): 224–240.

Lee, J.-S. 2008. Scheduling rules for dual-armed cluster tools with cleaning processes. Master's thesis, Department of Industrial & System Engineering, KAIST, Daejon, Korea.

Lee, T.-E. 2000b. Stable earliest starting schedules for periodic job shops: a linear system approach. *International Journal of Flexible Manufacturing Systems* 12 (1): 59–80.

Lee, T.-E., H.-Y. Lee, and S.-J. Lee. 2007. Scheduling a wet station for wafer cleaning with multiple job flows and multiple wafer-handling robots. *International Journal of Production Research* 45 (3): 487–507.

Lee, T.-E., H.-Y. Lee, and Y.-H. Shin. 2004. Workload balancing and scheduling of a single-armed cluster tools. In *Proceedings of Asian-Pacific Industrial Engineering and Management Systems Conference*, 1–6.

Lee, T.-E., and S.-H. Park. 2005. An extended event graph with negative places and negative tokens for time window constraints. *IEEE Transactions on Automation Science and Engineering* 2 (4): 319–332.

Lee, T.-E., and M. E. Posner. 1998. Performance measures and schedules in periodic job shops. *Operations Research* 45 (1): 72–91.

Lee, T. E., R. Sreenivas, and H.-Y. Lee. 2006. Workload balancing for timed event graphs with application to cluster tool operation. In *Proceedings of IEEE International Conference on Automation Science and Engineering*, 1–6.

Lei, L., and Q. Liu. 2001, May. Optimal cyclic scheduling of a robotic processing line with two-product and time-window constraints. *INFOR* 39 (2): 185–199.

Lopez, M. J., and S. C. Wood. 1998. Systems of multiple cluster tools: Configuration and performance under

perfect reliability. *IEEE Transactions on Semiconductor Manufacturing* 11 (3): 465–474.

Lopez, M. J., and S. C. Wood. 2003. Systems of multiple cluster tools: Configuration, reliability, and performance. *IEEE Transactions on Semiconductor Manufacturing* 16 (2): 170–178.

Murata, T. 1989. Petri nets: properties, analysis and applications. *Proceedings of the IEEE* 77 (4): 541–580.

Oh, H. L. 2000. Conflict resolving algorithm to improve productivity in single-wafer processing. In *Proceedings of the International Conference on Modeling and Analysis of Semiconductor Manufacturing (MASM)*, 55–60.

Paek, J.-H., and T.-E. Lee. 2002. Operating strategies of cluster tools with intermediate buffers. In *Proceedings of The Seventh Annual International Conference on Industrial Engineering*, 1–5.

Paek, J.-H., and T.-E. Lee. 2008. Optimal scheduling of dual-armed cluster tools without swap restriction. In *Proceedings of 2008 IEEE International Conference on Automation Science and Engineering*, 1–6.

Perkinson, T. L., R. S. Gyurcsik, and P. K. McLarty. 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.

Rostami, S., and B. Hamidzadeh. 2002. Optimal scheduling techniques for cluster tools with process-module and transport-module residency constraints. *IEEE Transactions on Semiconductor Manufacturing* 15 (3): 341–349.

Rostami, S., and B. Hamidzadeh. 2004. An optimal residency-aware scheduling technique for cluster tools with buffer module. *IEEE Transactions on Semiconductor Manufacturing* 17 (1): 68–73.

Rostami, S., B. Hamidzadeh, and D. Camporese. 2001, October. An optimal periodic scheduler for dual-arm robots in cluster tools with residency constraints. *IEEE Transactions on Robotics and Automation* 17 (5): 609–618.

Seo, J.-W., and T.-E. Lee. 2002. Steady state analysis of cyclic job shops with overtaking. *International Journal of Flexible Manufacturing Systems* 14 (4): 291–318.

Sethi, S. P., J. B. Sidney, and C. Sriskandarajah. 2001, June. Scheduling in dual gripper robotic cells for productivity gains. *IEEE Transactions on Robotics and Automation* 17 (3): 324–341.

Shin, Y.-H. 2002. *Scheduling a single-armed cluster tool with time window constraints*. Ph. D. thesis, Department of Industrial & Systems Engineering, KAIST, Daejon, Korea.

Shin, Y.-H., T.-E. Lee, J.-H. Kim, and H.-Y. Lee. 2001. Modeling and implementating a real-time scheduler for dual-armed cluster tools. *Computers in Industry* 45 (1): 13–27.

Venkatesh, S., R. Davenport, P. Foxhoven, and J. Nulman. 1997. A steady-state throughput analysis of cluster tools: Dual-blade versus single-blade robots. *IEEE Transactions on Semiconductor Manufacturing* 10 (4): 418–424.

Wu, N., and M. Zhou. 2007a. Deadlock modeling and control of semiconductor track systems using resource-oriented petri nets. *International Journal of Production Research* 45 (15): 3439–3456.

Wu, N., and M. Zhou. 2007b. Real-time deadlock-free scheduling for semiconductor track systems based on colored timed petri nets. *OR Spektrum* 29:421–443.

Yi, J., S. Ding, and D. Song. 2005. Steady-state throughput and scheduling analysis of multi-cluster tools for semiconductor manufacturing: A decomposition approach. In *International Conference on Robotics and Automation*, 292–298.

Yoon, H. J., and D. Y. Lee. 1999. Real-time scheduling of wafer fabrication with multiple product types. In *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, 835–840.

Yoon, H. J., and D. Y. Lee. 2001. On-line scheduling mehod for track systems in semiconductor fabrication. 25 (3): 443–451.

Zuberek, W. M. 2001a. Timed Petri net models of multi-robot cluster tools. In *Proceedings of 2001 IEEE International Conference on Systems, Man, and Cybernetics*, 2729–2734.

Zuberek, W. M. 2001b. Timed Petri nets in modeling and analysis of cluster tools. *IEEE Transactions on Robotics and Automation* 17 (5): 562–575.

Zuberek, W. M. 2004. Cluster tools with chamber revisiting: Modeling and analysis using timed petri nets. *IEEE Transactions on Semiconductor Manufacturing* 17 (3): 333–344.

Author Bibliography

TAE-EOG LEE is a professor and head of Department of Industrial & Systems Engineering at KAIST, Korea. He received BS from Seoul National University, MS from KAIST, and PhD (1991) from The Ohio State University, Columbus, USA, all in industrial engineering. His research areas include semiconductor manufacturing automation, modeling, scheduling, and control, especially on cluster tools or automated systems. He is an associate editor of IEEE Transactions on Automation Science and Engineering and the director of Defense Modeling & Simulation Technology Research Center at KAIST.