

## **CYCLIC SCHEDULING OF CLUSTER TOOLS WITH NON-IDENTICAL CHAMBER ACCESS TIMES**

Dae-Kyu Kim  
Chihyun Jung  
Tae-Eog Lee

Yu-Ju Jung

Korea Advanced Institute of Science & Technology  
Department of Industrial & Systems Engineering  
Yuseong-gu, Daejeon, KOREA

POSCO  
Goedong-dong, Nam-gu, Pohang  
Gyeongbuk, KOREA

### **ABSTRACT**

Most cluster tool scheduling studies assume identical access times between chambers, or do not discuss impact of the access times although the optimal scheduling rule and the cycle time can depend on the access times or physical configuration of parallel chambers. We examine cyclic scheduling problems for cluster tools that have non-identical access times. We first develop Petri net models of tool behaviors and analyze the cycle time by identifying the workloads of the process steps. We prove that the conventional backward and swap sequencing strategies are still optimal for a single-armed and a dual-armed cluster tools, respectively, when a process step is the bottleneck and the tool repeats a minimal cyclic work cycle. We also present a closed form formula for the cycle time and identify a co-prime condition on the number of parallel chambers for which the cycle time is independent of the order of using parallel chambers.

### **1 INTRODUCTION**

Cluster tools have been widely used for semiconductor industry. A cluster tool combines several single-wafer processing modules (PMs) or chambers and loadlocks (LLs) for wafer loading and unloading within a closed environment together with a wafer handling robot or transport module (TM). The robot may have a single arm or dual arms. The angle between the dual arms is generally fixed. A dual-armed cluster tool is more expensive than a single-armed cluster tool but is known to have higher throughput (Venkatesh et al. 1997).

Each wafer undergoes a number of process steps. One or more PMs are assigned to each process step. A long process step or the bottleneck process step is assigned to multiple PMs in order to reduce the tool cycle time (Perkinson, Gyurcsik, and McLarty 1996). Recent fabs tend to reduce the number of process steps, down to two or three, that are processed in a cluster tool and instead increase the number of parallel PMs for each process step. Figure 1 illustrates a radial-type dual-armed cluster tool with six PMs for a wafer flow pattern with four and two parallel chambers for process steps 1 and 2, respectively. Such parallelization can also be found in linear-type tools or track systems for coating and developing processes for photolithography. In a track system, a process step may have five or more vertically stacked PMs (Oh 2000). Therefore, the robot's access times to parallel chambers are significantly different due to vertical or linear robot movements. The differences can be more than a few seconds or even more than 10 seconds. Radial movements of the robot arms in a radial-type tool are usually short, in the range of a few seconds, and hence are often assumed to be identical (Kim et al. 2003, Geismar, Dawande, and Sriskandarajah 2004, Lee 2008, Paek and Lee 2008). However, even a radial-type tool may have significantly longer radial movement times when robot movements are controlled to be very slow in order to limit particle generation for extreme cleanness. Nonetheless, most cluster tool scheduling studies assume identical access

times for parallel chambers or do not examine the impacts of non-identical access times in depth although the optimal scheduling rule and the cycle time can depend on the access times or physical configuration of parallel chambers. When the access times to parallel chambers are significantly different, they should be considered in scheduling. Conventional scheduling rules such as backward sequence for radial-type single-armed tools and swap sequence for radial-type dual-armed tools, which disregard differences in access times to parallel chambers and deal with parallel chambers identically, are well known to be optimal and popularly used in the industry. They may be no longer optimal when the access times are significantly different. The access time differences may cause a large deviation in the total flow times of wafers and cause excessive quality variation (Oh 2000).

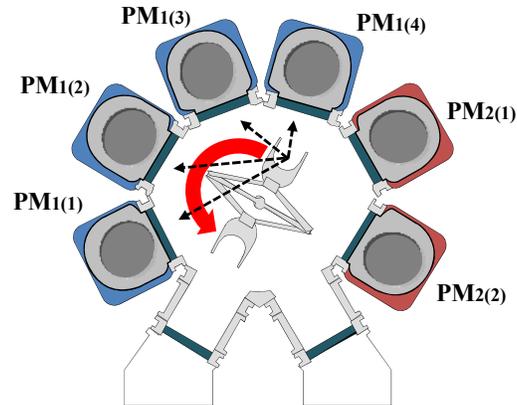


Figure 1: A dual-armed cluster tool with wafer flow pattern (4,2)

In this paper, we examine a cyclic scheduling problem for cluster tools that have non-identical chamber access times. We first develop a Petri net model of the tool behavior and compute the cycle time by analyzing the workloads of the process steps. We prove that the backward and swap sequence are still optimal for a single-armed and a dual-armed cluster tool, respectively, when a process step is the bottleneck and the tool repeats a minimal cyclic work cycle. We then present a closed form formula for the tool cycle time. We also identify a co-prime condition on the number of parallel chambers for each process steps for which the cycle time is independent of the order of using the parallel chambers.

## 2 SCHEDULING PROBLEMS FOR PARALLEL PROCESSING MODULES

PMs are assigned to the process steps, one or more to each step, according to the wafer processing recipe and the process times. In a cluster tool, a wafer unloaded from a LL takes a sequence of process steps by going through processing at the PMs according to the recipe, and returns to the LL. A LL is regarded as a PM with zero processing time in view of scheduling. A long process step can be assigned multiple identical PMs to reduce the cycle time for the process step. Therefore, wafer flow patterns are series-parallel. Cluster tools mostly process identical wafers continuously, at least 25 identical wafers in a wafer cassette. Then, a tool, more specifically the robot and each chamber, mostly repeats identical work cycles. Such cyclic scheduling has advantages such as steady operation, simplified scheduling problems, and controlled wafer delays and is used by most tools (Lee 2008). The tool operation cycle, including each PM's work cycle and the robot moves between the PMs, is determined in accordance with a cyclic sequence of the robot tasks such as loading and unloading a wafer at a PM or LL. Single-armed tools mostly have used a *backward sequence* that after an operation of transferring a wafer from a PM of process step  $n$  to a PM of process step  $n+1$  is completed, a similar operation is made from a PM of process step  $n-1$  to a PM of process step  $n$ . A *swap operation* for a dual-armed tool unloads a wafer completed in a PM into an empty robot arm and loads a wafer on another robot arm into the PM. A *swap sequence* repeats the swap operation for a PM of each process step in the process step sequence. The two sequences do not care

about the order in which the parallel PMs are used. Figures 2 and 3 illustrate the two sequences. The backward sequence for single-armed tools and the swap sequence for dual-armed tools have been known to minimize the tool cycle time. The optimality is proven by assuming identical access times to parallel chambers or no parallel chamber (Lee, Lee, and Shin 2004; Paek and Lee 2008). However, when the access times are significantly different, the optimality should be verified. In fact, some tools or processes require high-level cleanliness within the tool environment. To do this, the tool environment is kept as vacuum and the robot movement speed is controlled low to avoid risk of particle generation. Therefore, robot movement between PMs takes more than a few seconds. In the case, tool scheduling should consider non-identical robot movement times and count different access times to parallel PMs that have been regarded to be identical.

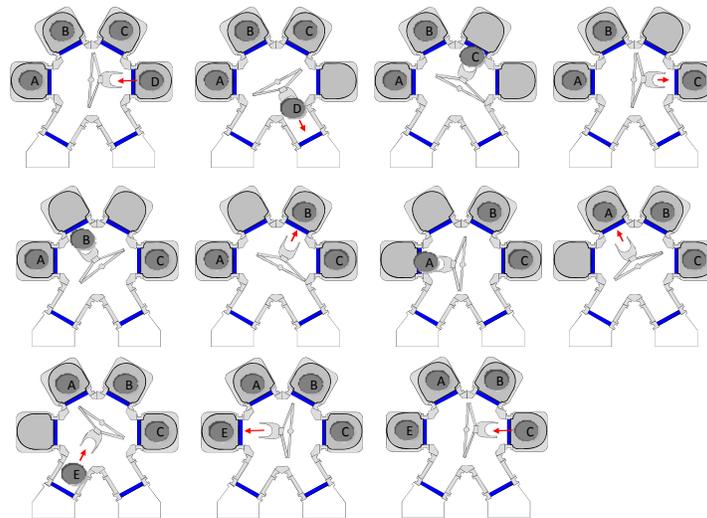


Figure 2: Backward sequence for a single-armed cluster tool

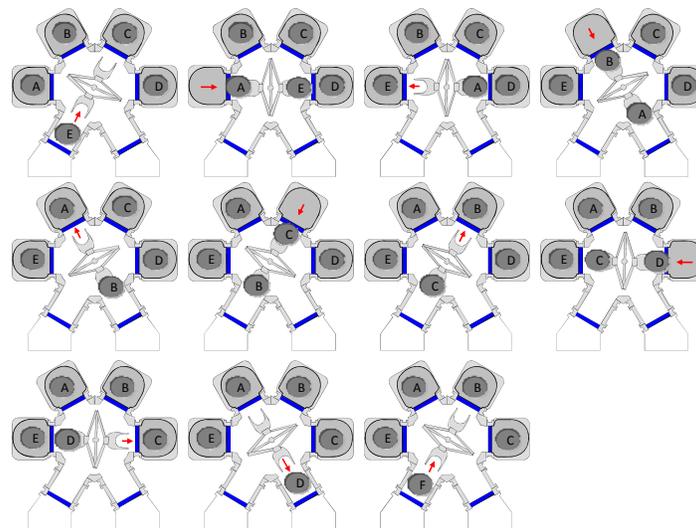


Figure 3: Swap sequence for a dual-armed cluster tool

There are tools that have significantly different access times to PMs and many parallel chambers. A track system for photolithography performs photo-sensitive chemical coating on a wafer surface, supplies to an attached stepper for exposing a coated wafer surface to the circuit image, and then develops the ex-

posed image. It has several robots and many PMs. Since the PMs are clustered into each robot, a track system can be viewed as a group of several interconnected cluster tools. Each process step tends to have many parallel PMs in order to match its throughput rate with that of an extremely expensive stepper. Figure 4 illustrates a track system that has five or ten vertically stacked PMs for each process step. The handling robot should make vertical movements as well as angular movements for such vertically stacked PMs. Vertical movement times tend to be significantly long and different depending on the stack positions, a few seconds to more than ten seconds. Such variance of access times to parallel PMs creates variability in the throughput times of wafers or wafer delays within PMs. Such time variability causes variability in wafer surface quality because coating and developing processes are sensitive to wafer delays and times between processes. The order for assigning wafers to parallel PMs affects the variability and even the throughput rate or the tool cycle time. Oh (2000) shows that cyclic ordering can reduce the variability significantly. We also need to examine whether the backward and swap sequences between the process steps are still optimal and how the order of using parallel chambers affects the tool cycle time.

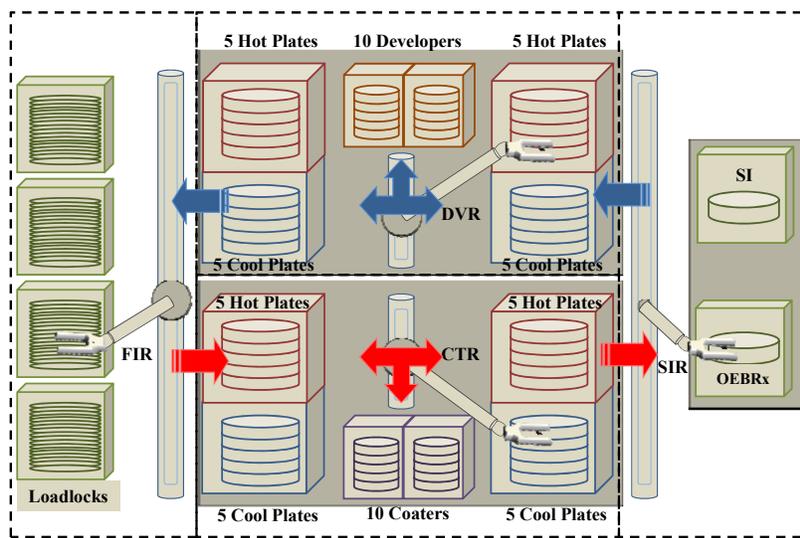


Figure 4: A track system with parallel PMs

Another example is provided by linearly configured cluster tools as seen in Figure 5. A linear cluster tool can flexibly extend the number of PMs (Yi et al. 2007), but have significantly different access times for the PMs due to linearly-connected chambers. Therefore, different access times to parallel PMs should be considered for scheduling.

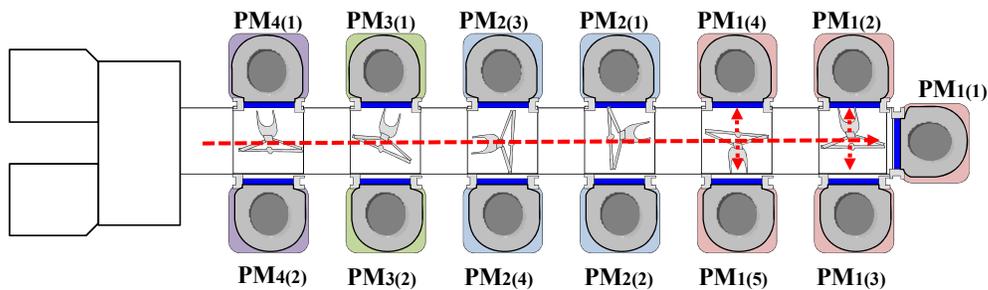


Figure 5: A linear cluster tool with parallel PMs

We examine cyclic scheduling of tools with distinctive parallel PMs that repeats identical work cycles and hence the robot and each PM performs identical sequence of tasks. Consider a *wafer flow pattern*

with parallel PMs, indicated by  $(m_1, m_2, \dots, m_n)$ , where  $m_i$  is the number of parallel PMs for process step  $i$ . When parallel PMs for each process step are treated identically by assuming identical access times, a cycle can be defined by a *pass* of each wafer from the first process step to the last process step. It is because we do not care which one of parallel PMs is used. Based on the above discussion, we should schedule parallel PMs distinctively. Therefore, a cycle should be defined by counting all parallel PMs distinctively. A minimum tool cycle is defined by  $k$  passes of wafers, where  $k$  is the same as the LCM(*Least Common Multiple*) of the  $m_i$ 's, indicated by  $\text{LCM}[m_1, m_2, \dots, m_n]$ . It is because the parallel PMs of each process step are used cyclically. Figure 6 illustrates such an LCM cycle. For each LCM cycle, the tool completes  $k$  wafers. Since only a single type of jobs or wafers are processed, the sequence of the robot tasks for loading and unloading at each PM determines a tool cycle. Therefore, we wish to determine an optimal robot task sequence to make an LCM cycle that has the minimum cycle time.

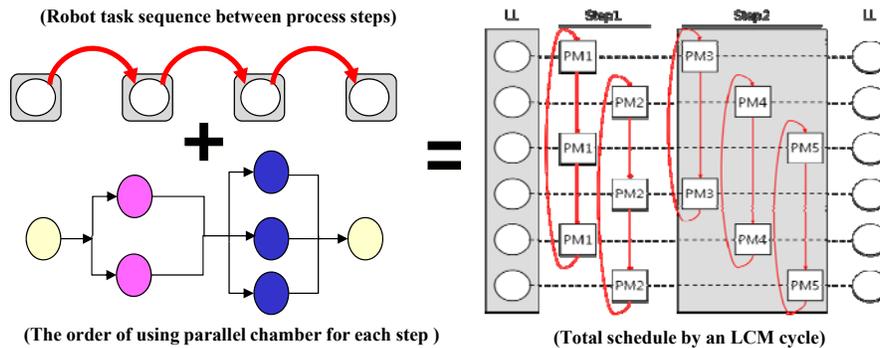


Figure 6: An LCM cycle

### 3 SINGLE-ARMED CLUSTER TOOL

#### 3.1 Petri Net Modeling

For scheduling analysis of cluster tools, we use Petri net models. A Petri net is a mathematical and graphical modeling framework for discrete event dynamic systems (Murata 1989). It consists of transitions, places, arcs, and tokens. They are denoted by bars, circles, directed arcs, and dots, respectively. They represent events or activities, condition or activities, precedence relation, and condition of places, respectively. If a transition has tokens at each input place and the sojourn time of a token at each input place is larger than the token holding time of the place, it becomes enabled. An enabled transition fires after a firing delay of the transition, if any. When a transition fires, each input place loses a token and each output place gets a new token.

A subset of Petri nets in which each place has exactly one input transition and output transition is called *event graphs* (Murata 1989). An event graph is also called a *decision-free* Petri net in the sense that there is no place where a token should be routed to after one of its multiple output transitions. An event graph is used for modeling cyclic behavior of a discrete event system such as a cluster tool that repeats an identical work cycle. An event graph that has nonzero token holding times at the places or nonzero firing delays at the transitions is called a *timed event graph* (TEG). The cycle time of a TEG is the same as the critical circuit ratio, the maximum of the sum of the token holding times and firing delays to the number of tokens in each circuit. It can be computed by a linear programming model or a polynomial-time algorithm. When the robot task sequence is given to a cluster tool, the behavior can be modeled by a TEG and the tool cycle time is the same as the cycle time of the TEG. We therefore model a tool with non-identical parallel PMs by a TEG. Figure 7 shows a TEG model for a single-armed cluster tool with two process steps and two parallel PMs for the second process step. The tool is operated by the backward sequence.

Transitions  $U_i$  and  $L_i$  represent an unloading task and a loading task at each process step  $i=1, \dots, n$ , respectively.  $U_0$  and  $L_3$  describe the robot task at a loadlock. The  $p$ -th PM of process step  $i$  is indicated by  $PM_{i(p)}$ . Places  $T_{i(p)(q)}$  and  $P_{i(p)}$  indicate a wafer transfer task from  $PM_{i(p)}$  to  $PM_{i+1(q)}$ ,  $q$ -th PM of process step  $i+1$ , and a processing task at a  $PM_{i(p)}$ , respectively. For a place  $T_{i(p)(q)}$  for a wafer transfer of two sequential process steps, we do not specify process step  $i+1$  for notational simplicity. For example, a place  $T_{1(1)(2)}$  means a transfer task from first PM of process step 1 to second PM of process step 2. When a process step  $i$  does not have parallel PMs, we can omit the second index for one of the parallel PM. Place  $M_{i(p)(q)}$  represents a movement of the robot with an empty arm from  $PM_{i(p)}$  of process step  $i$  to  $PM_{j(k)}$  of process step  $j$ . The transitions and places connected by the dotted arcs indicate the robot work cycle. The availability of the robot arm is indicated by a token at place  $A_{i(p)}$ . The token holding times at the places or the firing delays for transitions are the process times or the robot task times.

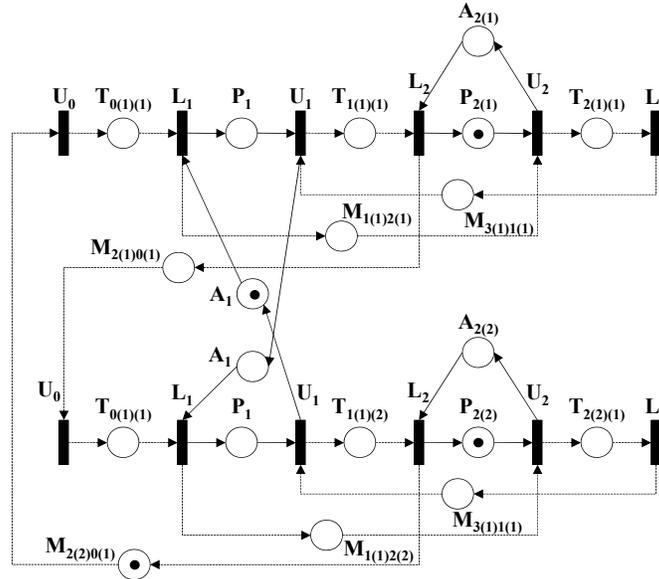


Figure 7: A TEG model for a single-armed cluster tool with the backward sequence

### 3.2 Workload Analysis

The workload measure of a resource such as a PM or a robot is defined to be the same as the minimal time to complete a wafer at the resource, that is, the sum of all task times for a work cycle of the resource (Kim et al. 2003; Lee, Lee, and Shin 2004). We extend the definition to an LCM cycle.

**Definition 1.** Suppose an LCM cycle with  $k$  wafer passes. The workload of a PM of process step  $i$  is the sum of the unloading, loading, transporting and processing task times, to produce  $k_i$  wafers, where  $k_i = k/m_i$  and  $m_i$  is the number of parallel PMs of process step  $i$ . The workload of the robot is the sum of all robot task times to produce  $k$  wafers.

We examine the necessary tasks for completing a work cycle at a PM,  $PM_{i(p)}$ . A wafer should be unloaded from a PM of process step  $i-1$ , transported to  $PM_{i(p)}$ , loaded into the PM, processed there, unloaded from there, transported to a PM of process step  $i+1$ , and loaded into the PM, and the freed robot should move to a PM of process step  $i-1$ . The times for the tasks are  $u$ ,  $\delta_{i-1,i(p)}$ ,  $l$ ,  $p_i$ ,  $u$ ,  $\delta_{i(p),i+1}$ ,  $l$ , and  $\delta_{i+1,i-1}$ , respectively. For times  $\delta_{i-1,i(p)}$ ,  $\delta_{i(p),i+1}$ , and  $\delta_{i+1,i-1}$ , the second index for the parallel PMs is omitted when we do not refer to a specific PM. Figure 8 illustrates the tasks for a work cycle for a PM,  $PM_{i(p)}$ . The tasks are required for a PM work cycle regardless of the robot task sequence or the tool work cycle. Of course, during processing a wafer at a PM, the robot can do other tasks. Since the actual work cycle can have delays

between the tasks, the sum of the task times is the minimum time of the work cycle, and hence defines the workload of the PM. By examining the robot movements between parallel PMs of consecutive process steps, we obtain the following result.

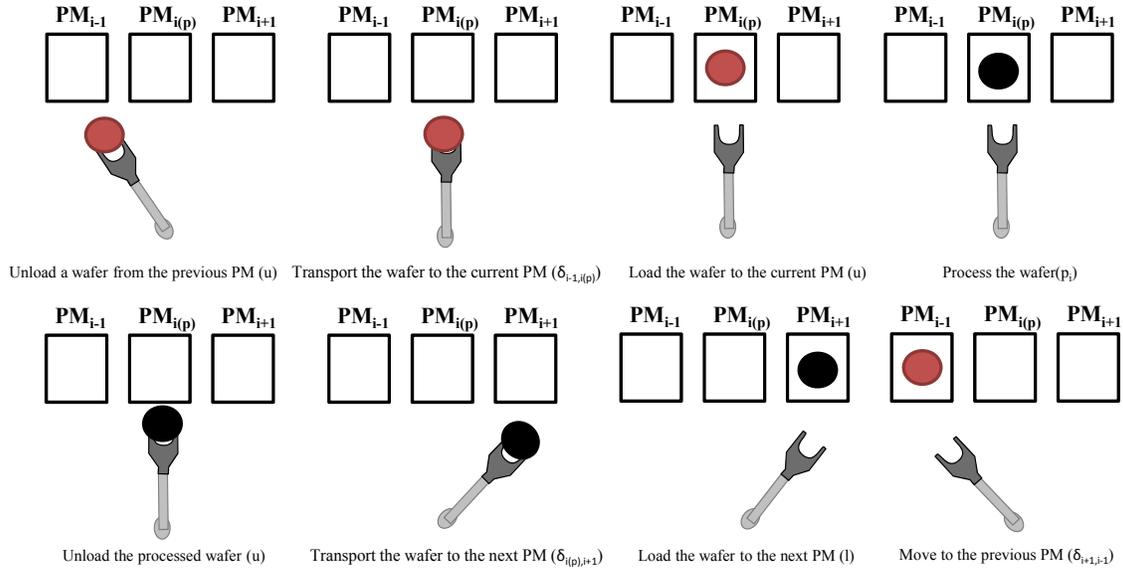


Figure 8: The work cycle of a PM of a single-armed tool

**Lemma 1.** *The workload of  $PM_{i(p)}$  in a single-armed tool with a series-parallel wafer flow pattern is*

$$\frac{k}{m_i}(p_i + 2u + 2l) + \sum_{c=1}^{k/m_i} (\delta_{i(p),i+1_{f(c)}} + \delta_{i+1_{f(c)},i-1_{f(c+m_i)}} + \delta_{i-1_{f(c+m_i)},i(p)}) \quad \forall i,$$

where  $\delta_{i+1_{f(c)},i-1_{f(c+m_i)}} \leq \delta_{i+1_{f(c)},j} + u + \delta_{j,j+1} + l + \delta_{j+1,i-1_{f(c+m_i)}} \quad \forall j, j \neq i-1.$

**Corollary 1.** *If the numbers of parallel PMs for three consecutive process steps,  $m_{i-1}$ ,  $m_i$ , and  $m_{i+1}$ , are co-prime, then the workload of  $PM_{i(p)}$  is*

$$\frac{k}{m_i}(p_i + 2u + 2l) + \frac{k}{m_i}(\delta_{i-1,i(p)}^{ave} + \delta_{i(p),i+1}^{ave} + \delta_{i+1,i-1}^{ave}),$$

where  $\delta_{i-1,i(p)}^{ave} = \frac{1}{m_{i-1}} \sum_{k=1}^{m_{i-1}} \delta_{i-1(k),i(p)}$ ,  $\delta_{i(p),i+1}^{ave} = \frac{1}{m_{i+1}} \sum_{q=1}^{m_{i+1}} \delta_{i(p),i+1(q)}$ , and  $\delta_{i+1,i-1}^{ave} = \frac{1}{m_{i-1}m_{i+1}} \sum_{k=1}^{m_{i-1}} \sum_{q=1}^{m_{i+1}} \delta_{i-1(k),i+1(q)}$ ,

and  $\delta_{i+1_{f(c)},i-1_{f(c+m_i)}} \leq \delta_{i+1_{f(c)},j} + u + \delta_{j,j+1} + l + \delta_{j+1,i-1_{f(c+m_i)}} \quad \forall j, j \neq i-1.$

An LCM cycle makes  $k$  wafer passes. It consists of  $k(n+1)$  unloading and loading tasks and  $2k(n+1)$  wafer transfer or robot move tasks. The sequence between the latter tasks determines the work cycle since the former tasks follow the latter tasks. We wish to determine the optimal robot task sequence between the latter tasks. A robot task sequence is defined as  $\sigma=(\sigma(1), \dots, \sigma(2k(n+1)))$ , where  $\sigma(h)$  indicates the  $h$ -th robot task. Index  $f(c)$  represents  $c$ -th pass of a wafer. The robot's workload simply follows.

**Theorem 1.** *The workload of a transfer robot is*

$$k(n+1)(u+l) + \sum_{c=1}^k \sum_{h=1}^{2(n+1)} \delta_{\sigma(h)_{f(c)}, \sigma(h+1)_{f(c)}}.$$

### 3.3 Optimal Sequence

By examining the circuit ratios of the TEG model, we can establish the following theorem.

**Theorem 2.** *The backward sequence minimizes the cycle time for a single-armed cluster tool with a series-parallel wafer flow pattern, if the following condition is satisfied:*

$$\begin{cases} p_i \geq \{m_i(n+1) - 2\}(u+l) + \delta_{i(p),n} + \dots + \delta_{i+2,i(p)} & i = 1 \\ p_i \geq \{m_i(n+1) - 2\}(u+l) + \delta_{i(p),i-2} + \dots + \delta_{i+2,i(p)} & \forall i, i \neq 1, i \neq n-2 \\ p_i \geq \{m_i(n+1) - 2\}(u+l) + \delta_{i(p),i-2} + \dots + \delta_{1,i(p)} & i = n \end{cases}$$

Then the cycle time of a single-armed cluster tool with a series-parallel wafer flow pattern under the backward sequence is

$$\lambda = \max_{p \in \{1, 2, \dots, m_i\}} \left\{ \frac{k}{m_i} (p_i + 2u + 2l) + \sum_{c=1}^{k/m_i} (\delta_{i(p),i+1_{f(c)}} + \delta_{i+1_{f(c)},i-1_{f(c+m_i)}} + \delta_{i-1_{f(c+m_i)},i(p)}) \right\},$$

where  $\delta_{i+1_{f(c)},i-1_{f(c+m_i)}} \leq \delta_{i+1_{f(c)},j} + u + \delta_{j,j+1} + l + \delta_{j+1,i-1_{f(c+m_i)}} \quad \forall j, j \neq i-1.$

The tool cycle time cannot be less than the maximum of the workloads of the PMs and the robot. We observe that the cycle time of Theorem 2 is the same as the maximum workload of the PMs. Therefore, if the tool is *process-bound* as in most practical tools, in other words, the workloads of the PMs are more than the robot workload, the tool cycle time by the backward sequence is optimal.

**Corollary 2.** *If the numbers of parallel PMs for three consecutive process steps,  $m_{i-1}$ ,  $m_i$ , and  $m_{i+1}$ , are co-prime, the cycle time of a single-armed cluster tool with a series-parallel wafer flow pattern under the backward sequence is*

$$\lambda = \max_{p \in \{1, 2, \dots, m_i\}} \left\{ \frac{k}{m_i} (p_i + 2u + 2l) + \frac{k}{m_i} (\delta_{i-1,i(p)}^{ave} + \delta_{i(p),i+1}^{ave} + \delta_{i+1,i-1}^{ave}) \right\},$$

where  $\delta_{i-1,i(p)}^{ave} = \frac{1}{m_{i-1}} \sum_{k=1}^{m_{i-1}} \delta_{i-1(k),i(p)}$ ,  $\delta_{i(p),i+1}^{ave} = \frac{1}{m_{i+1}} \sum_{q=1}^{m_{i+1}} \delta_{i(p),i+1(q)}$ , and  $\delta_{i+1,i-1}^{ave} = \frac{1}{m_{i-1}m_{i+1}} \sum_{k=1}^{m_{i-1}} \sum_{q=1}^{m_{i+1}} \delta_{i-1(k),i+1(q)}$ ,

and  $\delta_{i+1_{f(c)},i-1_{f(c+m_i)}} \leq \delta_{i+1_{f(c)},j} + u + \delta_{j,j+1} + l + \delta_{j+1,i-1_{f(c+m_i)}} \quad \forall j, j \neq i-1.$

In Corollaries 1 and 2, we can see that the cycle time of a single-armed cluster tool under the backward sequence is invariant to the order in which the parallel PMs are used.

**Corollary 3.** *For a single-armed cluster tool with a backward sequence, the minimum cycle time is irrelevant to the order of using parallel PMs if the numbers of parallel PMs of the bottleneck process step and the preceding and succeeding process steps are mutually co-prime.*

We also compute the cycle time of a special case where the transfer times are all constant. We confirm the known result (Lee, Lee, and Shin 2004).

**Corollary 4.** *If the transfer times are all identical, the cycle time of a single-armed cluster tool with a series-parallel wafer flow pattern under the backward sequence is*

$$\max \left\{ \frac{k}{m_i} (p_i + 2u + 2l + 3\delta), k(n+1)(u+l+2\delta) \right\}.$$

## 4 DUAL-ARMED CLUSTER TOOL

### 4.1 Petri Net Modeling

We can develop a TEG model for a dual-armed tool with the swap sequence as illustrated in Figure 9. It represents the behavior of a dual-armed cluster tool under the swap sequence. The tool has wafer flow pattern (2,1). Transitions  $U_i$  and  $L_i$  and place  $P_{i(p)}$  indicate unloading, loading, and processing tasks. Place  $T_{i(p)j(q)}$  indicates a robot transfer task from  $PM_{i(p)}$  to  $PM_{j(q)}$ . Thus, Places  $T_{i(p)i(p)}$  represents 180° rotation of the dual-arm at  $PM_{i(p)}$ . All places  $T_{i(p)j(q)}$  are for wafer transfer tasks, except place  $T_{3(1)0(1)}$  ( $=T_{n+1(1)0(1)}$ ) which is an empty moving task from and to the loadlock. The dotted arcs indicate the robot work cycle. In this model, the availability place for the robot arms is not necessary because it can be indicated by tokens at place  $T_{i(p)i(p)}$ . The token holding times for each place  $T_{i(p)j(q)}$  are different depending on how the parallel PMs are used.

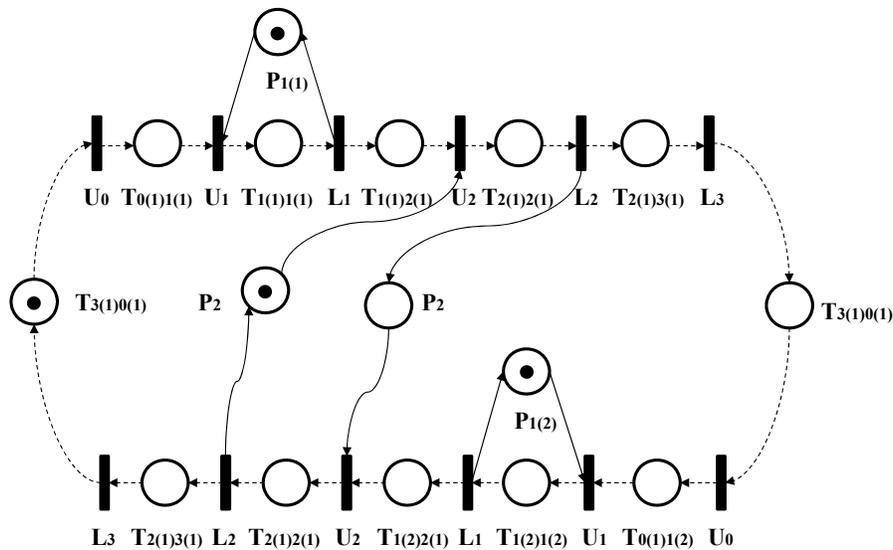


Figure 9: A TEG model for a dual-armed cluster tool with the swap sequence

### 4.2 Workload Analysis

Similarly as in a single-armed cluster tool case, we examine the workloads of each resource. We examine the work cycle of  $PM_{i(p)}$  to produce a wafer. It should have each one of processing, unloading, transporting, and loading tasks. In the following task descriptions, the symbol in a parenthesis indicates the task time. To achieve the minimum cycle time, a wafer which completes a processing tasks ( $p_i$ ) should be unloaded immediately by a robot arm ( $u$ ). Therefore, there should be another processing wafer on the other robot arm at the time, as illustrated in Figure 10. Otherwise, it takes additional time to pick up a newly processed wafer. Since the angle of the two robot arms remains 180°, the robot should make a 180° rotational movement ( $\delta_{i(p), i(p)}$ ) for transporting the wafer to the next PM. After loading the wafer to the next

PM ( $l$ ), the cycle of  $PM_{i(p)}$  is completed. To produce  $k_i$  wafers, similar cycles should be repeated  $k_i$  times. We have the following lemma for the workload of a PM.

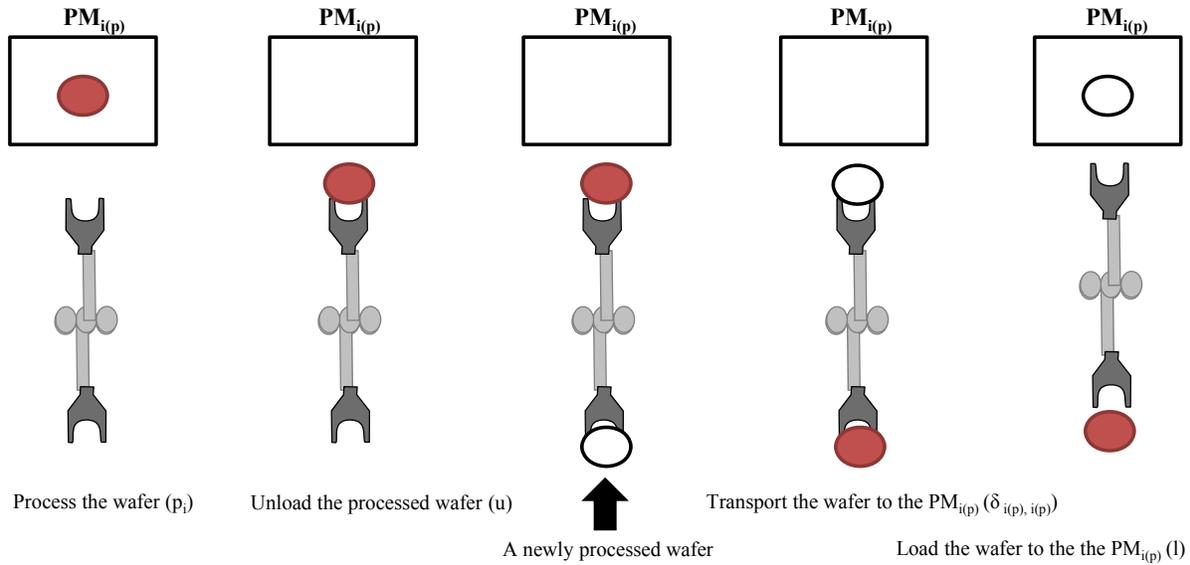


Figure 10: The work cycle of a PM in a dual-armed cluster tool

**Lemma 2.** *The workload of  $PM_{i(p)}$  in a dual-armed tool with a series-parallel wafer flow pattern is*

$$k_i(p_i + u + l) + \sum_{c=1}^{k_i} \delta_{i(p), i(p)_{f(c)}} .$$

### 4.3 Optimal Sequence

By analyzing the TEG model, we identify the cycle time of a dual-armed cluster tool under the swap sequence.

**Theorem 3.** *The swap sequence minimizes the cycle time for a dual-armed cluster tool with a series-parallel wafer flow pattern if the following condition is satisfied:*

$$p_i > \{m_i(n+1) - 1\}(u + l) + \delta_{i(p), i+1} + \dots + \delta_{i-1, i(p)} \quad \forall i .$$

*Then the cycle time of a dual-armed cluster tool with a series-parallel wafer flow pattern under the swap sequence is*

$$\lambda = \max_{p \in \{1, 2, \dots, m_i\}} \left\{ \frac{k}{m_i} (p_i + u + l) + \sum_{c=1}^{k/m_i} \delta_{i(p), i(p)_{f(c)}} \right\} .$$

The cycle time in Theorem 3 is the same as the maximum workload of a PM. Therefore, the swap sequence is optimal when the tool is process-bound.

**Theorem 4.** *If the robot work cycle is the bottleneck of a dual-armed cluster tool with a series-parallel wafer flow pattern under the swap sequence, the cycle time is*

$$\lambda = k(n+1)(u+l) + \sum_{c=1}^k \left( \sum_{i=0}^n \delta_{i,i+1,r(c)} + \sum_{i=1}^n \delta_{i(p),i(p),r(c)} + \delta_{n+1,0} \right).$$

As suggested by Theorem 3, the cycle time of a dual-armed cluster tool under the swap sequence is independent of the order of using parallel chambers. Theorem 4 suggests that if the numbers of parallel chambers of two sequential process steps are all co-prime, the cycle time is not affected by the order of using parallel PMs.

**Corollary 5.** *For a dual-armed cluster tools under the swap sequence, the minimum cycle time is irrelevant to the order of using parallel PMs if the numbers of parallel chambers of two sequential process steps are all co-prime.*

We also compute a well-known special case where the access times are identical (Paek and Lee 2008).

**Corollary 6.** *If the access times are all identical, the cycle time of a dual-armed cluster tool with a series-parallel wafer flow pattern under the swap sequence is*

$$\max \left\{ \frac{k}{m_i} (p_i + u + l + \delta), k(n+1)(u + l + 2\delta) \right\}.$$

Most cluster tools in operation are process-bound. It is because PMs are more expensive and create higher values on wafers than the robot. Therefore, the throughput should not be lost due to waiting for the robot. Therefore, the backward and swap sequences are effective even when the access times for parallel PMs are significantly different, only if the tool is process-bound and some conditions on the number of parallel chambers are met. Nevertheless, there can be cases of robot-bound. For the cases, we can develop mixed integer programming models for determining an optimal robot task sequence in cluster tools with non-identical access times to parallel PMs (Jung 2009, Kim 2009).

## 5 CONCLUSION

We identified that the conventional backward and swap sequence are still optimal even when the access times for parallel PMs are significantly different only if the tool is process-bound and some conditions are satisfied. In the case, we also proved that the order of using parallel PMs is irrelevant. The co-prime conditions on the numbers of parallel chambers for process steps suggest how to configure parallel chambers for each process step. The results are most helpful for operating and designing cluster tools.

Further issues include scheduling of cluster tools with more complex and realistic requirement such as reentrant flows. It is worthwhile to examine dispatching rules for non-identical parallel PMs. In addition, since the dispatching of the lots from different products waiting in front of a group of cluster tools may cause a big impact on the overall cluster tool performance, it would be an interesting research issue to study how our internal cluster tool scheduling approaches interact with external scheduling approaches.

## REFERENCES

- Geismar, H. N., M. Dawande, and C. Sriskandarajah. 2004. "Robotic Cells with Parallel Machines: Throughput Maximization in Constant Travel-Time Cells." *Journal of Scheduling* 7 (5): 375-395.
- Jung, Y. J. 2009. "Cyclic Scheduling of Single-armed Cluster Tools with Different Wafer Transport Times between Parallel Chamber." Master's thesis, Department of Industrial & Systems Engineering, KAIST, Daejeon, Korea.
- Kim, J. H., T. E. Lee, H. Y. Lee, and D. B. Park. 2003. "Scheduling Analysis of Time-Constrained Dual-Armed Cluster Tools." *IEEE Transactions on Semiconductor Manufacturing* 16(3):521-534.

- Kim, D. K. 2009. "Cyclic Scheduling of Dual-armed Cluster Tools with Different Wafer Transport Times between Parallel Chamber." Master's thesis, Department of Industrial & Systems Engineering, KAIST, Daejeon, Korea.
- Lee, T. E. 2008. "A Review of Scheduling Theory and Methods for Semiconductor Manufacturing Cluster Tools." In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler, 2127-2135. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Lee, T. E., H. Y. Lee, and Y. H. Shin. 2004. "Workload Balancing and Scheduling of a Single-Armed Cluster Tool." In *Proceedings of Asia-Pacific Industrial Engineering and Management Systems Conference*, 1-15.
- Murata, T. 1989. "Petri Nets: Properties, Analysis and Application." *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*, 246-251.
- 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*, 103-108. IEEE.
- 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.
- 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.
- Yi, J., S. Ding, M. T. Zhang, and P. V. D. Meulen. 2007. "Throughput Analysis of Linear Cluster Tools." In *Proceedings of 2007 IEEE International Conference on Automation Science and Engineering*, 1063-1068. IEEE.

## AUTHOR BIOGRAPHIES

**DAE-KYU KIM** received B.S. and M.S. degrees from KAIST (Korea Advanced Institute of Science and Technology), Department of Industrial & System Engineering. He is now a doctoral student in the department. His research interests include scheduling and control of integrated manufacturing systems and modeling & simulation of defense systems. His email address is [kimdaekyu@kaist.ac.kr](mailto:kimdaekyu@kaist.ac.kr).

**CHIHYUN JUNG** is a post-doctoral researcher at KAIST (Korea Advanced Institute of Science and Technology), Department of Industrial & System Engineering, after finishing his Ph.D. at the same department in 2010. His research interests include modeling, simulation, and scheduling of discrete event systems and cluster tools. His e-mail is [chjung@kaist.ac.kr](mailto:chjung@kaist.ac.kr).

**YU-JU JUNG** received B.S. and M.S degrees from KAIST. Her master thesis was on single-armed cluster tool scheduling with non-identical parallel chambers. She is now working at POSCO Steel, Korea. Her e-mail is [yuju.jung@gmail.com](mailto:yuju.jung@gmail.com).

**TAE-EOG LEE** is a professor and head at Department of Industrial & Systems Engineering, KAIST (Korea Advanced Institute of Science and Technology). His research interests include cyclic scheduling theory, scheduling and control theory of timed discrete event dynamic systems such as timed Petri nets, and their application to scheduling and control of cluster tools for semiconductor manufacturing. He was an associate editor of *IEEE Transactions on Automation Science and Engineering*. He can be reached via email at [telee@kaist.ac.kr](mailto:telee@kaist.ac.kr).