Steady-State Throughput and Scheduling Analysis of Multi-Cluster Tools for Semiconductor Manufacturing

Jingang Yi
New Product Development Division
Lam Research Corporation
Fremont, CA 94538, USA
jingang.yi@cal.berkeley.edu

Shengwei Ding
Dept. of IEOR
University of California at Berkeley
Berkeley, CA 94720, USA
dingsw@ieor.berkeley.edu

Dezhen Song
Dept. of Computer Science
Texas A&M University
College Station, TX 77843, USA
dzsong@cs.tamu.edu

***Abstract*— Cluster tools are widely used as semiconductor manufacturing equipment. While throughput analysis and scheduling of single-cluster tools have been well-studied, the corresponding research on multi-cluster tools is still at early stage. This paper analyzes steady-state throughput and scheduling of multi-cluster tools. Based on the analysis, we propose a decomposition method to reduce a multi-cluster tool problem to multiple single-cluster tool problems. We then apply the throughput and scheduling results from existing research for each single-cluster tool. For a M-cluster tool, we present an $O(M)$ throughput calculation and robot scheduling algorithm. A chemical-mechanical planarization (CMP) polisher is used as an example of the multi-cluster cluster tools to illustrate the proposed decomposition method and algorithms.**

I. Introduction

Cluster tools are widely used as semiconductor manufacturing equipment. In general, a cluster tool is defined as an integrated, environmentally isolated manufacturing system consisting of process, transport, buffer, and cassette modules mechanically linked together (Fig. 1). Cassette modules (CM) store the unprocessed and processed wafers. Process modules (PM) execute the semiconductor manufacturing processes, such as deposition, etching and chemical-mechanical planarization. Transfer modules (TM), which are robot manipulators, move the wafers among process modules and between process and cassette modules. Within a single-cluster tool, only one robot is serving multiple process and cassette modules (Fig. 1(a)). A multi-cluster tool consists of several single clusters that are inter-connected through buffer modules (BM) (Fig. 1(b)). Since wafers are processed to produce the integrated circuits using a multiple sequential process steps, modeling analysis and scheduling of multi-cluster tools is critical to improve the productivity and enhance the design of wafer processing equipments.

In this paper, we will discuss modeling, analysis and scheduling for a multi-cluster tool. We consider a general topological connection among the multi-cluster tools. We propose a method to decompose the multi-cluster tools into multiple independent single-cluster tools and then apply the known throughput and scheduling analysis of the single cluster tools. The method also accommodates different inter-connection types between two connected clusters due to different buffer and transfer modules. For a M-cluster tool, we present an $O(M)$ throughput calculation and robot scheduling algorithm. A CMP polisher is used as an example of the multi-cluster tools to illustrate the proposed decomposition method and algorithms.

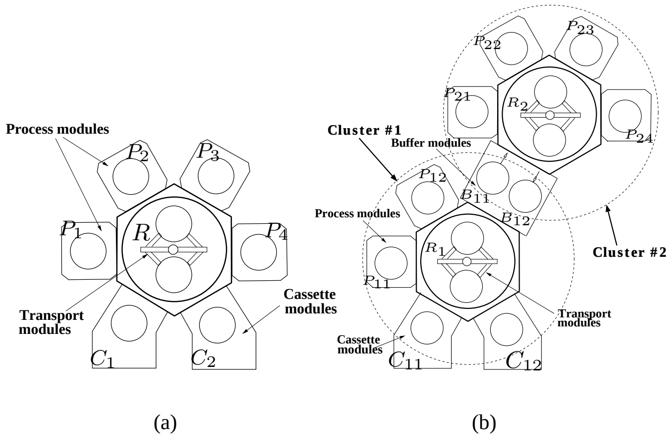

Fig. 1. A schematic of cluster tools, (a) single-cluster tool, (b) two-cluster tool.

The rest of paper is organized as follows. We begin with related work in section II and discuss the structure of the multi-cluster tools in section III. Then we present the decomposition method for multi-cluster tools and the algorithms for calculating the minimal fundamental period (FP) and scheduling analysis in section IV. An example of the modeling analysis and process scheduling is investigated for a CMP polisher in section V. Finally, we summarize the concluding remarks and future work in section VI.

II. Related work

In [1], [2], analytical models of steady-state throughput were discussed for a single-cluster tool. To model and simulate the single-cluster tools, [3] and [4] used Petri nets to study the performance of semiconductor manufacturing processes. Several researchers have discussed the process scheduling for a single-cluster tool, for example, [5] and [6] discussed the optimization of the double-blade robot schedule to maximize the throughput of a cluster tool with residency constraints on process and transfer modules. Recently, [7] discussed and compared the use of a three-blade robot and a buffer module for scheduling a single-cluster tool with residency constrains. In [8], [9], scheduling analysis of one robot flowshop was discussed for the single- and double-gripper robots that were not used for semiconductor manufacturing industry.

All of work above discussed the single-cluster tool configuration. The single-cluster tool scheduling for a single wafer process flow is relatively straightforward. With the increasing complexity of the semiconductor manufacturing processes, a multi-cluster tool is needed to accommodate the industry needs. Fig. 1(b) shows an example of a two-cluster tool. For such a multi-cluster tool, wafer flow modeling and scheduling is much more complicated compared with the single-cluster tool because the multiple robots within a multi-cluster tool can move and transfer wafers simultaneously and coordinately. In [10], [11], several rule or priority based heuristic scheduling methods of transfer modules (robots) within the multi-cluster tools have been discussed. However, there is few analysis and comparison study of those heuristic methods in terms of optimality. The main goal of this study fills such a gap and investigates the throughput and scheduling of a general configuration of multi-cluster tools.

III. Single- and Multi-cluster Tools

Nomenclature[1]

$\mathbb{C}_i$	Cluster i of an M-cluster tool.
N_i	Number of process modules (PMs) in $\mathbb{C}_i$.
$C_{ij}(C_j)$	Cassette module j in $\mathbb{C}_i$.
C^*_{ij}	Fictitious cassette module j in a decoupled $\mathbb{C}_i$.
$P_{ij}(P_j)$	Process module j in $\mathbb{C}_i$.
B_{ij}	The j^{th} buffer module between $\mathbb{C}_i$ and $\mathbb{C}_{i+1}$.
B_i	$B_i = \bigcup_j B_{ij}$: Collection of buffer modules between $\mathbb{C}_i$ and $\mathbb{C}_{i+1}$.
$\mathbf{S}_i$	Wafer capacity of B_i.
R_i	The robot manipulator for $\mathbb{C}_i$.
$\mathbf{FP}$	The minimal fundamental period (FP) of the multi-cluster tool.
$\mathbf{FP}^*_i$	The minimal fundamental period (FP) of the decoupled $\mathbb{C}_i$.
$\mathbf{V}$	Wafer process (visit) of the multi-cluster tool.
$\mathbf{V}^*_i$	Wafer process (visit) of a decoupled $\mathbb{C}_i$.
$T_i(T)$	The time interval R_i picks/places a wafer.
$t_{ij}(t_j)$	Process time at module P_{ij} (P_j).

During a manufacturing process, wafers are transported by the robot from the cassette, sequentially going through various process modules, and then return to the cassette. A single-blade robot usually can only hold one wafer at a time. A double-blade robot has two independent arms and therefore can hold two wafers at the same time with one on each arm. We only consider the case that all wafers follow the same cyclic flow pattern. Our assumptions are:

Assumption 1 *Cassette and transfer modules assumptions.*

1. *Cassette modules always have wafers/spaces for transfer module (robot) to pick or place at any time.*
2. *Each cluster has only one transfer module (robot) and this robot has at most two blades* [2].
3. *The robot R_i of cluster $\mathbb{C}_i$ takes the same amount of time T_i to pick and place a wafer, spends zero time to travel to next module, and therefore needs a constant time $2T_i$ to transfer a wafer from one process module to the next module. Moreover, T_i are deterministic.*
4. *Process time t_{ij} are deterministic.*
5. *Wafer process flow $\mathbf{V}$ visits each PM only once.*

Remark 1 Assumption 1.1 of cassette module above implies that the cassette modules will not cause process flow starving/blocking of the cluster tools.

Remark 2 Although in this paper we mainly consider the cases where the robot transferring time is constrained by Assumption 1.3, the similar results however can be obtained and extended to the cases where the robot R_i spends different time intervals to pick and place a wafer.

A. Single-cluster tool

For a single-cluster tool, only one transfer module (robot) moves wafers between various modules. Fundamental period (FP) is defined as the elapsed time between the completion of processing of two consecutive wafers [1]. The concept of FP is equivalent to 1-unit cycle time defined for robotic flowshop [9]. Denote $\mathbf{FP}$ as the minimal fundamental period for a process on a cluster tool and we can then calculate the cluster tool maximum steady-state throughput as $1/\mathbf{FP}$.

According to [12], for a single-cluster tool that has N PMs ($N = 4$ for the example shown in (Fig. 1(a)) and is equipped with a single-blade robot, one optimal 1-unit cyclic scheduling is given by following robot moves: robot first picks up the wafer in P_N (assuming there is one wafer in each PM in a steady state) and places into C_2, then keep moving wafers from P_j to P_{j+1}, $j = 1, 2, \cdots, N-1$, and finally picks up a wafer from cassette C_1 and places it into P_1. With a double-blade robot, one optimal 1-unit cyclic scheduling is different due to the "swap" actions that two-blade robot can carry [8]: robot first picks up a wafer (on blade 1) from cassette C_1, moves to P_1, picks up the existing wafer in P_1 (with blade 2), and places the wafer (on blade 1) into P_1 ("swap" action). Then robot points to P_2, waits for process finishing and then "swaps" the wafer in P_2. The robot keeps swapping wafers through P_j, $j = 1, 2, \cdots, N$, and finally places the wafer to C_2.

[1]Notations in the parenthesis are for single cluster case.

[2]Most robots used in the semiconductor manufacturing industry have either one blade or two blades.

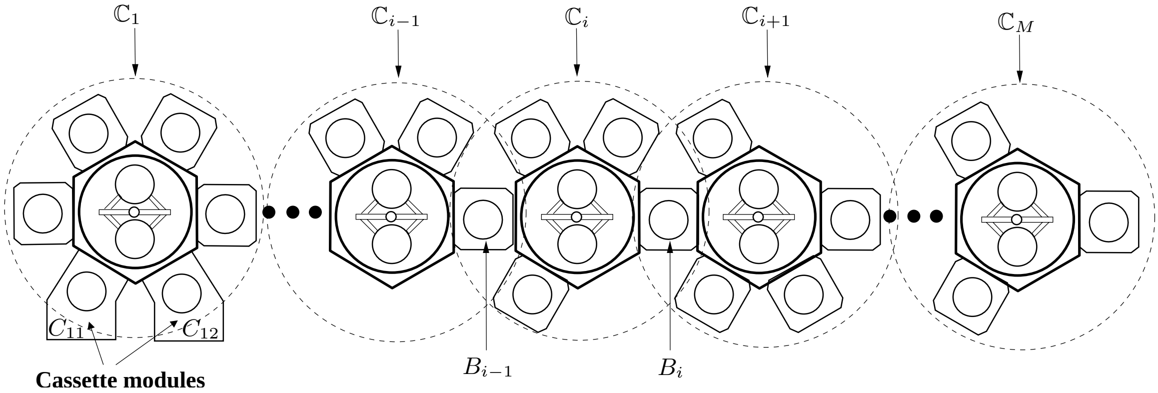

Fig. 2. A schematic of an inter-connected M-cluster tool.

Recall the notation for a single cluster case in *Nomenclature*. Depending on the process time t_j and transfer time $2T$, the scheduling of a single cluster is running under two possible regions: *process-bound* and *transfer-bound* regions [1], [2]. When the cluster is under a process-bound region, the largest processing time dominates the $\mathbf{FP}$ and the robot has some idle time. While in the transfer-bound region, the robot is always busy in transferring wafers and processing times are relatively small. Based on the optimal schedule described previously, we can calculate $\mathbf{FP}_s$ (for a single-blade robot) and $\mathbf{FP}_d$ (for a double-blade robot) in a simple form.

$$\mathbf{FP}_s = \begin{cases} 2(N+1)T & \text{if } t^{\max} < 2(N-1)T \\ t^{\max} + 4T & \text{if } t^{\max} \geq 2(N-1)T \end{cases} \tag{1}$$

$$\mathbf{FP}_d = \begin{cases} 2(N+1)T & \text{if } t^{\max} < 2NT \\ t^{\max} + 2T & \text{if } t^{\max} \geq 2NT \end{cases}, \tag{2}$$

where $t^{\max} = \max_{1 \leq j \leq N}\{t_j\}$ is the maximum process time of all N process modules. The first case in Eqs. (1) and (2) represents the transfer-bound region and the second case for the process-bound region.

B. Multi-cluster tool configurations and assumptions

A multi-cluster tool is defined as combination of several single clusters that are inter-connected through buffer modules. Fig. 1(b)) shows an example of a two-cluster tool. For a more general case, we consider an inter-connected M-cluster tool as shown in Fig. 2.

For the multi-cluster tools that we study in this paper (as shown in Fig. 2), we have the following assumptions.

Assumption 2 *Topological constraints of the multi-cluster tools.*

1. *Each cluster within a tool must connect to at least one but no more than two other clusters.*
2. *The multiple clusters within a tool cannot form a loop inter-connection.*

The problem we consider here is to find the minimal fundamental period ($\mathbf{FP}$) and a corresponding robot schedule of a given wafer process $\mathbf{V}$ for the multi-cluster tools as shown in Fig. 2.

IV. Multi-cluster Analysis Using a Decomposition Method

A. Cluster decomposition

To analyze the multi-cluster systems, we propose an approach to decouple the interconnection among clusters and then apply the steady-state performance and scheduling results for each decoupled single-cluster tool.

The key of the approach is how to decouple the link between clusters. As shown in Fig. 2, for cluster $\mathbb{C}_i$ in a multi-cluster system, we know that wafers flow into or out of the cluster through either buffer modules or cassette modules. Let's assume $i > 1$. Cluster $\mathbb{C}_i$ exchanges wafers with $\mathbb{C}_{i-1}$ through buffer modules B_{i-1}. Therefore, for $\mathbb{C}_i$, B_{i-1} acts like a fictitious cassette module. On the other hand, for $\mathbb{C}_{i-1}$, B_{i-1} acts like a fictitious process module.

Therefore, we can decouple a multi-cluster tool into a set of single clusters by treating buffer modules as either fictitious cassette modules or fictitious process modules. Assuming those single cluster runs independently, we can then find the shortest feasible fundamental period $\mathbf{FP}_i^*$ for each decoupled cluster $\mathbb{C}_i$. After we obtain the set of $\{\mathbf{FP}_i^*\}, i = 1, \cdots, M$, we can identify the cluster with the largest $\mathbf{FP}_i^*$ to be the bottleneck of the systems, which will determine $\mathbf{FP}$ for the entire system.

Fig. 3 shows an example of how to decouple a two-cluster tool (as shown in Fig. 1(b)) with a two-wafer capacity buffer module into two single clusters. For such a two-cluster tool, we can apply the decomposition method and consider two single clusters as shown in Fig. 3.

We consider to construct the two decoupled clusters. For decoupled $\mathbb{C}_1$, wafers leave through buffer module B_{11} and back through buffer module B_{12}. So B_1 are considered as one fictitious process module P_{13}^*. [3] Moreover, the process time of P_{13}^* depends on the the following cluster and we will calculate this value in the next section.

For decoupled $\mathbb{C}_2$, the buffer modules B_{11} and B_{12} become the fictitious cassette modules C_{21}^* and C_{22}^*. So we have the second single cluster with 4 processing modules and two fictitious cassette modules as shown in Fig. 3.

[3] We use the superscript "*" to denote the variables associated with fictitious modules.

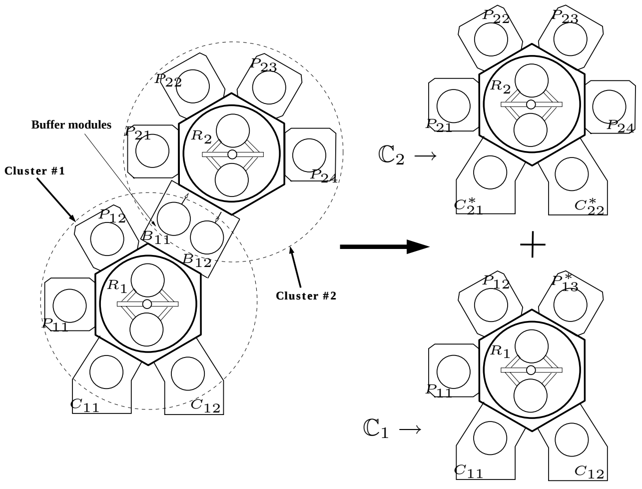

Fig. 3. An example of the decoupling method for an inter-connected 2-cluster tool shown in Fig. 1(b).

Suppose that the wafer process flow $\mathbf{V}$ for the cluster tool shown in Fig. 3 is as follows.

$$\begin{aligned}\mathbf{V}: \ & C_{11} \xrightarrow{R_1} P_{11} \xrightarrow{R_1} P_{12} \xrightarrow{R_1} B_{11} \xrightarrow{R_2} P_{21} \xrightarrow{R_2} P_{22} \\ & \xrightarrow{R_2} P_{23} \xrightarrow{R_2} P_{24} \xrightarrow{R_2} B_{12} \xrightarrow{R_1} C_{12}\,. \end{aligned} \tag{3}$$

Then after decomposition, the wafer flows for the two single-cluster tools are

$$\begin{aligned}\mathbf{V}_1^*: \ & C_{11} \xrightarrow{R_1} P_{11} \xrightarrow{R_1} P_{12} \xrightarrow{R_1} P_{13}^* \xrightarrow{R_1} C_{12} \\ \mathbf{V}_2^*: \ & C_{21}^* \xrightarrow{R_2} P_{21} \xrightarrow{R_2} P_{22} \xrightarrow{R_2} P_{23} \xrightarrow{R_2} P_{24} \xrightarrow{R_2} C_{22}^*\,.\end{aligned}$$

Now, there are two remaining problems needed to be addressed before we can compute the shortest feasible fundamental period $\mathbf{FP}_i^*$ for the decoupled single clusters. The first problem is caused by the difference between real cassette modules and the fictitious cassette modules. Based on our assumption, there is always a wafer available, if the robot wants to pick a wafer from the real cassette module. The waiting time $t_{10} = 0$ for the first cluster $\mathbb{C}_1$. However, this is not true for cluster $\mathbb{C}_i$, $i > 1$, with the fictitious cassette module. Robot R_i can not pickup a wafer from the fictitious cassette module before robot R_{i-1} finishes loading the wafer. This incurs a loading delay t_{i0}^*. Note that we are computing the shortest feasible fundamental period for each decoupled cluster, we do not care about the possible delay caused by different cyclic periods of clusters. The second problem is the processing time $t_{i(N_i+1)}^*$ of the fictitious processing module for decoupled $\mathbb{C}_i$. The decoupled $\mathbb{C}_i$ has N_i real processing modules and we can always denote the fictitious process module as the (N_i+1)th PM by Assumption 2. We will discuss how to compute t_{i0}^* and $t_{i(N_i+1)}^*$ later in the section. Now let's focus on how to compute the $\mathbf{FP}_i^*$ for known t_{i0}^* and $t_{i(N_i+1)}^*$.

In order to apply and extend the $\mathbf{FP}$ calculation for a single-cluster tool to a decoupled $\mathbb{C}_i$, we consider the loading delay time t_{i0}^* as an extension time of robot R_i, and $t_{i(N_i+1)}^*$ as the processing time of fictitious module $P_{i(N_i+1)}^*$. With this observation, we can extend Eqs. (1) and (2) and calculate the shortest feasible fundamental period $\mathbf{FP}_i^*$ for decoupled single-cluster tool $\mathbb{C}_i$ as follows.

$$\mathbf{FP}_i^* = \begin{cases} t_i^{\max *} + 4T_i + t_{i0}^*, & \text{if } R_i \text{ is single-blade and } N_i + N_i^* = 1 \\ 2(N_i + N_i^* + 1)T_i + t_{i0}^*, & \text{if } R_i \text{ is single-blade, } N_i + N_i^* > 1, \text{ and } t_i^{\max *} < 2(N_i + N_i^* - 1)T_i + t_{i0}^* \\ t_i^{\max *} + 4T_i, & \text{if } R_i \text{ is single-blade, } N_i + N_i^* > 1, \text{ and } t_i^{\max *} \geq 2(N_i + N_i^* - 1)T_i + t_{i0}^* \\ 2(N_i + N_i^* + 1)T_i + t_{i0}^*, & \text{if } R_i \text{ is double-blade and } t_i^{\max *} < 2(N_i + N_i^*)T_i + t_{i0}^* \\ t_i^{\max *} + 2T_i, & \text{if } R_i \text{ is double-blade and } t_i^{\max *} \geq 2(N_i + N_i^*)T_i + t_{i0}^*, \end{cases} \tag{4}$$

where N_i^* ($= 0$ or 1) is the number of fictitious PM in $\mathbb{C}_i$ and $t_i^{\max *} = \max_{1 \leq j \leq N_i}\{t_{ij}, t_{i(N_i+1)}^*\}$ is the maximum process time of all N_i PMs and the fictitious PM. In the following subsection, we will discuss the how to compute loading delay t_{i0}^* of fictitious cassette module and fictitious processing time $t_{i(N_i+1)}^*$ of the decoupled $\mathbb{C}_i$.

B. Analysis of loading delay of fictitious cassette modules

Consider decoupled clusters $\mathbb{C}_i$, the loading delay t_{i0}^* associated with fictitious cassette module C_{ij} can be considered as wafer inter-arrival time with no starvation. With no starvation means if whenever R_i wants to pick a wafer from C_{ij}, there should be a wafer available unless R_{i-1} is still unloading the wafer. With this observation, we have the following results of loading delay t_{i0}^* for fictitious cassette module C_{ij} .

Proposition 1 *The loading delay t_{i0}^* of the fictitious cassette module of decoupled cluster $\mathbb{C}_i$ can be considered as an extra transferring time for R_i and is calculated as,*

$$t_{i0}^* = \max\{m_{i-1}T_{i-1} - 2n_iT_i, 0\} \tag{5}$$

where

$$m_{i-1} = \begin{cases} 1 & \mathbf{S}_i \geq 2 \textit{ and } R_{i-1} \textit{ is double-blade} \\ 2 & \mathbf{S}_i \geq 2 \textit{ and } R_{i-1} \textit{ is single-blade, or} \\ & \mathbf{S}_i = 1 \textit{ and } R_{i-1} \textit{ is double-blade} \\ 4 & \mathbf{S}_i = 1 \textit{ and } R_{i-1} \textit{ is single-blade,} \end{cases}$$

and

$$n_i = \begin{cases} N_i + N_i^* + \frac{1}{2}, & \mathbf{S}_i \geq 2 \textit{ and } R_i \textit{ is double-blade} \\ N_i + N_i^* & \mathbf{S}_i \geq 2 \textit{ and } R_i \textit{ is single-blade, or} \\ & \mathbf{S}_i = 1 \textit{ and } R_i \textit{ is double-blade} \\ N_i + N_i^* - 1 & \mathbf{S}_i = 1 \textit{ and } R_i \textit{ is single-blade}. \end{cases}$$

Proof: We sketch the proof briefly. The t_{i0}^* depends on two factors: (1) how fast the robot R_{i-1} to create a vacancy (by removing the processed wafer from C_{ij}^*) and refill the demand (by placing an unprocessed wafer into C_{ij}^*), and (2) how fast the robot R_i will come back to pick (place) a wafer after it places (picks up) a wafer into (from) C^*_{ij}. The first term $m_{i-1}T_{i-1}$ in Eq. (5) represents the minimal time that the robot R_{i-1} needs to access to C^*_{ij} when the robot R_i cannot pick or place a wafer from and into C^*_{ij}. The second term $2n_iT_i$ implies the time gap between the robot R_i picks up and places a wafer from and into C^*_{ij}. Depending on the robots configurations (either single- or double-blade), capacity of fictitious cassette module $\mathbf{S}_i$ and the pick/place time T_{i-1} and T_i for robots R_{i-1} and R_i respectively, the final delay time t^*_{i0} will be varying. Parameters m_{i-1} and n_i are used to describe different combinations of system and interface configurations of $\mathbb{C}_{i-1}$ and $\mathbb{C}_i$ respectively. ■

Remark 3 If the pick/place times of all robots are equal, $T_i = T$, the loading delay time t^*_{i0} of $\mathbb{C}_i$ by Eq. (5) can be further simplified by the fact that $N_i + N^*_i \geq 1$ as follows.

$$t^*_{i0} = \begin{cases} 0 & \mathbf{S}_i \geq 2 \textit{ or } \mathbf{S}_i = 1 \\ & \textit{and } R_{i-1} \textit{ and } R_i \textit{ are double-blades} \\ \max\{2T(2 - N_i - N^*_i), 0\} & \mathbf{S}_i = 1, \\ & R_{i-1} \textit{ is double-blade, and} \\ & R_i \textit{ is single-blade or} \\ & R_{i-1} \textit{ is single-blade, and} \\ & R_i \textit{ is double-blade} \\ \max\{2T(3 - N_i - N^*_i), 0\} & \mathbf{S}_i = 1, \\ & R_{i-1} \textit{ is single-blade, and} \\ & R_i \textit{ is single-blade}. \end{cases}$$

Normally if one cluster has more than three modules, i.e. $N_i + N^*_i \geq 3$, then $t^*_{i0} = 0$, which implies that no additional loading delay is needed if we can increase the number of PMs within a cluster and thus improve the tool efficiency.

C. Analysis of fictitious PM processing time

For the fictitious PM processing time $t^*_{i(N_i+1)}$ of $\mathbb{C}_i$, the calculation is different. We have to consider the fictitious processing time $t^*_{i(N_i+1)}$ as the processing time at module $P_{i(N_i+1)}$ to represent the time delay that results from the robot R_{i+1} and the buffer module B_i. Similar as loading delay t^*_{i0} of the fictitious cassette module, we can obtain $t^*_{i(N_i+1)}$ calculation results in the following proposition.

Proposition 2 *The fictitious PM processing time $t^*_{i(N_i+1)}$ of decoupled cluster $\mathbb{C}_i$ can be considered as an extra transferring time for R_i and is calculated as,*

$$t^*_{i(N_i+1)} = \begin{cases} 0, & \mathbf{S}_{i+1} \geq 2. \\ 2T_{i+1}, & \mathbf{S}_{i+1} = 1, R_{i+1} \textit{ is double-} \\ & \textit{-blade} \\ 4T_{i+1}, & \mathbf{S}_{i+1} = 1, R_{i+1} \textit{ is single-} \\ & \textit{blade and } N_{i+1} + N^*_{i+1} \geq 2, \\ \mathbf{FP}^*_{i+1} - t^*_{(i+1)0}, & \mathbf{S}_{i+1} = 1, R_{i+1} \\ & \textit{is single-blade and} \\ & N_{i+1} + N^*_{i+1} = 1 \end{cases} \quad (6)$$

Proof: See the Appendix. ■

D. Throughput calculations and robot scheduling

With the analysis above, the calculation algorithm of $\mathbf{FP}$ for the multi-cluster tools is described as Algorithm 1. First, we decompose the M-cluster tool into M single clusters with methods discussed previously. The calculation starts from the last decoupled cluster $\mathbb{C}_M$ which does not have any fictitious PM and then propagates backward to the first decoupled cluster $\mathbb{C}_1$. Then $\mathbf{FP}^*_i$ of each decoupled single cluster can be calculated with the fictitious processing and cassette modules. The calculation of $\mathbf{FP}^*_i$ is based on Eq. (4). The whole tool $\mathbf{FP}$ is calculated as the maximum of the M decoupled single clusters.

Algorithm 1: FP calculation for a multi-cluster tool.

Input : A cluster tool configuration and wafer flow $\mathbf{V}$
Output: Fundamental period $\mathbf{FP}$ for $\mathbf{V}$
Decompose the M-cluster tool into M single-cluster tools
Construct the wafer flows $\mathbf{V}^*_i$, $i = 1, 2, \cdots, M$, for each single cluster $\mathbb{C}_i$
for $i = M$ *to* 1 **do**
  Construct t^*_{i0} by Eq. (5)
  Construct $t^*_{i(N_i+1)}$ by Eq. (6)
  Calculate $\mathbf{FP}^*_i$ for cluster $\mathbb{C}_i$ using Eq. (4)
end
$\mathbf{FP} = \max_{1\leq i\leq M}\{\mathbf{FP}^*_i\}$

For the robot scheduling, we denote the schedule π_i of robots R_i, $i = 1, 2, \cdots, M$, as a doublet of its actions ACT_i and their relative starting times ST_i in one cycle: π_i=$\{ACT^j_i, ST^j_i\}$, $j = 1, 2, \cdots, L_i$, where L_i is number of robot actions in decoupled $\mathbb{C}_i$. It can be found that after we have calculated $\mathbf{FP}^*_i$ for each decoupled cluster $\mathbb{C}_i$, we can first schedule the robot R_1 according to the single cluster configuration (section III-A). Then we extend the schedule period to the calculated minimal system fundamental period $\mathbf{FP}$. Next, we can schedule R_2 according to the single cluster configuration and again expand to system $\mathbf{FP}$. After a proper timing shift of R_2's schedule according to the inter-connection, these two schedules can be fitted into the same fundamental period $\mathbf{FP}$. Repeating this procedure from R_3 to R_M, we can find the optimal schedule of the entire system.

However, the optimal schedule of the multi-cluster tool $\boldsymbol{\pi} = (\pi_i), i = 1, 2, \cdots, M$, is not necessarily unique. Without loss of generality, we assume that the decoupled cluster $\mathbb{C}_k$, $1 \leq k \leq M$, has the largest calculated fundamental period $\mathbf{FP}^*_k$. Since for each robot R_i, $i \neq k$, there could be some scheduling flexibility due to the fact that robot R_i always has some idle time. We propose a "no-wait" schedule method as in Algorithm 2. This algorithm leads to a unique scheduling solution by forcing all robot movements started as early as possible.

Algorithm 2: A "no-wait" optimal robot scheduling.

Input : A cluster tool configuration, wafer flow $\mathbf{V}$, and fundamental period $\mathbf{FP}$
Output: Scheduling $\boldsymbol{\pi}$ for $\mathbf{V}$
for $i = 1$ *to* M **do**
    From the single cluster configuration of cluster $\mathbb{C}_i$, obtain schedule π_i={ACT_i^j, ST_i^j}
end
Initialize system schedule as $\boldsymbol{\pi} = \pi_1$
for $i = 2$ *to* M **do**
    Find the action ACT_{i-1}^s that places wafers to $\mathbb{C}_i$
    Find the action ACT_i^t that takes wafers into $\mathbf{B_i}$, and its end time $ET_i^t = ST_i^t + T_i$
    Update all ST_i^j, $ST_i^j \leftarrow ST_i^j + (ST_j^s - ET_i^t)$
    Modulus all ST_i^j by $\mathbf{FP}$, $ST_i^j \leftarrow \mathrm{mod}(ST_i^j, \mathbf{FP})$
    $\boldsymbol{\pi} \leftarrow \boldsymbol{\pi} + \pi_i$
end

V. EXPERIMENT EXAMPLES

Fig. 4 shows a simplified CMP cluster polisher. The CMP process is used to planarize the wafer surface for a better control of photolithography performance in semiconductor manufacturing. The CMP polisher can be modeled as a four-cluster tool. There are two two-blade robots R_1 and R_2, an one-blade robot R_3 and a special indexer R_4. R_4 moves wafers simultaneously from processing modules P_{41} to P_{42}, P_{42} to P_{43}, P_{43} to P_{44} and P_{44} to P_{41}, respectively. The wafers go through the cluster tool as the following flow chart:

$$C_{11} \xrightarrow{R_1} P_{11} \xrightarrow{R_2} B_2 \xrightarrow{R_3} P_{41} \xrightarrow{R_4} P_{42} \xrightarrow{R_4} P_{43} \xrightarrow{R_4} P_{44}$$
$$\xrightarrow{R_4} P_{41} \xrightarrow{R_3} B_2 \xrightarrow{R_2} P_{21} \xrightarrow{R_2} P_{22} \xrightarrow{R_2} P_{12} \xrightarrow{R_1} C_{12}\,.$$

The robot transferring and each process module processing time are listed in Table I for a standard CMP process.

A. Cluster simplification

The decoupled clusters of the CMP polisher do not always have the exact configuration of multi-cluster tools shown in Fig. 2. For example, $\mathbb{C}_1$ have two buffer modules B_{11} and B_{12} which also serve as two PMs. The indexer R_4 of $\mathbb{C}_4$ moves wafers simultaneously among PMs and R_4 is different with single- or double-blade robots that we discussed before. In order to apply the results discussed in previous sections, we have to re-arrange these clusters and convert them into a standard multi-cluster configurations as shown in Fig. 2. Note that this re-arrangement only perserves the system equivalence in terms of the throughput and scheduling.

For decoupled $\mathbb{C}_1$, we can separate PMs with buffer modules. Fig. 5(a) shows a decoupled single cluster $\mathbb{C}_1'$ that represents the same system configuration with robot R_1'. Two PMs, P_{11}' and P_{12}', are separated with fictitious P_{13}^*. P_{13}^* also serves as buffer module B_1 for $\mathbb{C}_2$ with $\mathbf{S}_1 = 2$,

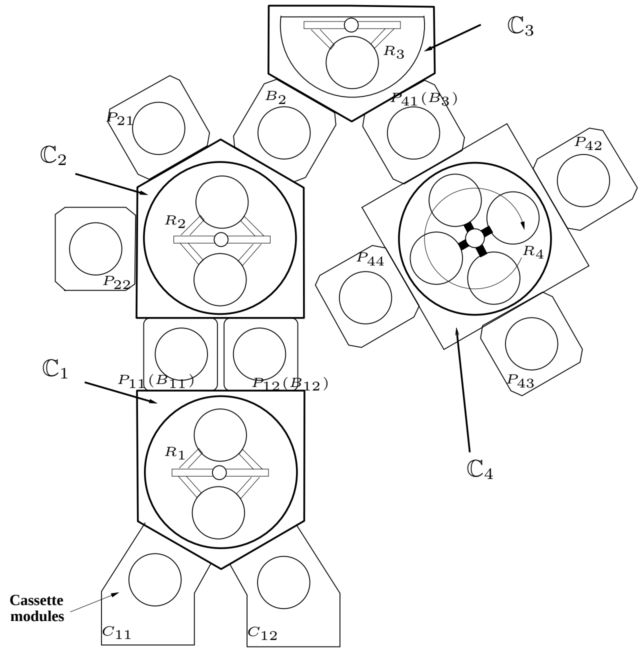

Fig. 4. A simplified CMP multi-cluster polisher.

which represents two-wafer capacity buffer modules B_{11} and B_{12}. The wafer flow of $\mathbb{C}_1'$ is then $\mathbf{V}_1^* : C_{11} \to P_{11}' \to P_{13}^* \to P_{12}' \to C_{12}$. In order to keep $\mathbb{C}_1'$ equivalently as $\mathbb{C}_1$, we have to consider the following modifications:

1. Robot R_1' needs T_1 to pick/place wafers from C_{11} into P_{11}' and P_{12}' into C_{12}.
2. The transfer time that R_1' swaps wafers from P_{11}' to P_{13}^*, and P_{13}^* to P_{12}' are zero.
3. The new processing time t_{11}' and t_{12}' of P_{11}' and P_{12}', respectively, must include the pick/place time T_2 of R_2 because of the dual-role of processing and buffer modules for P_{11} and P_{12}.

$$t_{11}' = t_{11} + T_2\,, \quad t_{12}' = t_{12} + T_2\,.$$

For $\mathbb{C}_4$, the indexer R_4 is a special type of robot and we can use an equivalent double-blade robot R_4' (as shown in Fig. 5(b)) with different transferring time to emulate the indexer activities. The wafer flow for $\mathbb{C}_4'$ is defined as $\mathbf{V}_4^*$: $P_{41}' \to P_{42}' \to P_{43}' \to P_{44}' \to P_{41}'$. We can consider that the transfer times of R_4' between P_{4i}' are different for each $i = 1, 2, 3, 4$ in order to represent the same indexer activities:

1. The total time R_4' picks up a wafer from P_{41}' and swaps with an existing wafer in P_{42}' is T_4;
2. The transferring time that R_4' swaps wafers among P_{42}', P_{43}', and P_{44}' and places a wafer back into P_{41}' are zero.
3. The processing time for each PM of $\mathbb{C}_4'$ is assigned equally as

$$t_{42}' = t_{43}' = t_{44}' = \max\{t_{42}, t_{43}, t_{44}\} \qquad (7)$$

B. Numerical results

With the cluster re-arrangement, we can directly apply the decomposition technique discussed in section IV to the 4-cluster CMP polisher. Table II illustrates the throughput

TABLE I

PROCESS AND TRANSFER TIME OF THE CMP POLISHER

Activities	**Time variables**	**Values** (s)
R_1 pick/place	T_1	10
R_2 pick/place	T_2	15
R_3 pick/place	T_3	10
R_4 indexing	T_4	5.5
P_{11} processing	t_{11}	10
P_{12} processing	t_{12}	20
P_{21} processing	t_{21}	30
P_{22} processing	t_{22}	30
P_{42} processing	t_{42}	60
P_{43} processing	t_{43}	60
P_{44} processing	t_{44}	60

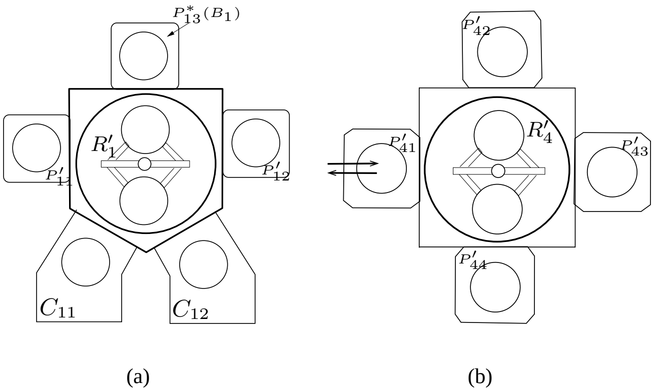

Fig. 5. Equivalent clusters, (a) $\mathbb{C}'_1$ for $\mathbb{C}_1$, (b) $\mathbb{C}'_4$ for $\mathbb{C}_4$ with a double-blade robot R'_4.

calculation for each decoupled cluster using Algorithm 1. In Table II we also list the fictitious PM number N^*_i, loading delay t^*_{i0}, fictitious processing time $t^*_{i(N_i+1)}$, and buffer size $\mathbf{S}_i$ for each decoupled cluster $\mathbb{C}_i$. Although the robot transfer time of $\mathbb{C}'_1$ and $\mathbb{C}'_4$ are varying among various modules and the analytical results in section IV are for symmetric robot transferring, we can still apply the results with minor adjustments. For example, when we calculate t^*_{31} for $\mathbb{C}_3$, we need to consider the time $2T'_4$ (Eq. (6)) that R'_4 can refill the buffer B_3 $(= P'_{41})$, which is 5.5 sec.

Using Algorithm 1, we can calculate the $\mathbf{FP}$ of the 4-cluster CMP polisher as

$$\mathbf{FP} = \max_{1 \le i \le 4} \{\mathbf{FP}^*_i\} = 120 \text{ sec.}$$

After the fundamental period is found, we can verify it with a feasible schedule. First, we label all the robot actions as in Table III and find an optimal schedule for each decoupled single cluster. According to Algorithm 2, we merge these schedules and make a final system schedule as shown in Table IV, which complies with the calculated $\mathbf{FP}$.

We further use an alternative simulation based method [13] to verify the optimal scheduling for the CMP polisher. The simulation gives the same results. However, the throughput analysis and scheduling using the decomposition method proposed in this paper is more straightforward and compliments the simulation methods.

TABLE II

CALCULATION RESULTS FOR THE CMP POLISHER BY CLUSTER DECOMPOSITION ("D" FOR DOUBLE-BLADE; "S" FOR SINGLE-BLADE.)

$\mathbb{C}_i$	$\mathbf{S}_i$	R_i	N_i	N^*_i	t^*_{i0} (s)	$t^*_{i(N_i+1)}$ (s)	$\mathbf{FP}^*_i$ (s)
$\mathbb{C}'_1$	2	D	2	1	0	0	40
$\mathbb{C}_2$	2	D	2	1	0	45.5	120
$\mathbb{C}_3$	1	S	0	1	0	5.5	45.5
$\mathbb{C}'_4$	1	D	3	0	0	0	65.5

TABLE III

ACTION LABELS FOR THE CMP POLISHER

ACT#	Actions	Robot (-blade)	Time (s)
1	$C_{11} \to P_{11}$ pick	R_1-1	10
2	$C_{11} \to P_{11}$ place	R_1-1	10
3	$P_{11} \to B_2$ pick	R_2-1	15
4	$P_{11} \to B_2$ place	R_2-1	15
5	$B_2 \to P_{41}$	R_3	20
6	Index $P_{41} \to P_{42}, \cdots, P_{44} \to P_{41}$	R_4	10
7	$P_{41} \to B_2$	R_3	15
8	$B_2 \to P_{21}$ pick	R_2-2	15
9	$B_2 \to P_{21}$ place	R_2-2	15
10	$P_{21} \to P_{22}$ pick	R_2-1	15
11	$P_{21} \to P_{22}$ place	R_2-1	15
12	$P_{22} \to P_{12}$ pick	R_2-2	15
13	$P_{22} \to P_{12}$ place	R_2-2	15
14	$P_{12} \to C_{12}$ pick	R_1-2	10
15	$P_{12} \to C_{12}$ place	R_1-2	10

TABLE IV

AN OPTIMAL SCHEDULE FOR THE CMP POLISHER

ACT#	14	11	5	15	3	1	6	7	2	13	15	4	10	15	12
Start time (s)	0	0	0	10	15	20	20	25.5	30	30	45	60	75	90	105

VI. CONCLUSION

In this paper, we presented a decomposition method to study the steady-state throughput and robot scheduling analysis of a multi-cluster tool for semiconductor manufacturing. The proposed method utilized the cyclic scheduling and analysis results obtained previously for single-cluster tools. Algorithms to calculate the maximum throughput and one optimal schedule of multi-cluster tools were given and analyzed. The results provided an efficiently systematic method to study the throughput and schedules of any multi-cluster tools. An application example of chemical-mechanical planarization (CMP) polisher illustrated the efficiency and complexity of the proposed methods. Cyclic scheduling and analysis of a multi-cluster tool with random processing time is a natural extension of work presented in this paper.

REFERENCES

[1] T. Perkinson, P. McLarty, R. Gyurcsik, and R. Cavin, "Single-Wafer Cluster Tool Performance: An Analysis of Throughput," *IEEE Trans. Semiconduct. Manufact.*, vol. 7, no. 3, pp. 369–373, 1994.

[2] S. Venkatesh, R. Davenport, P. Foxhoven, and J. Nulman, "A Steady-State Throughput Analysis of Cluster Tools: Dual-Blade Versus Single-Blade Robots," *IEEE Trans. Semiconduct. Manufact.*, vol. 10, no. 4, pp. 418–424, 1997.

[3] R. Srinivasan, "Modeling and Performance Analysis of Cluster Tools Using Petri Nets," *IEEE Trans. Semiconduct. Manufact.*, vol. 11, no. 3, pp. 394–403, 1998.

[4] W. Zuberek, “Timed Petri Nets in Modeling and Analysis of Cluster Tools,” *IEEE Trans. Robot. Automat.*, vol. 17, no. 5, pp. 562–575, 2001.

[5] S. Rostami, B. Hamidzadeh, and D. Camporese, “An Optimal Periodic Scheduler for Dual-Arm Robots in Cluster Tools with Residency Constraints,” *IEEE Trans. Robot. Automat.*, vol. 17, no. 5, pp. 609–618, 2001.

[6] S. Rostami and B. Hamidzadeh, “Optimal Scheduling Techniques for Cluster Tools With Process-Module and Transport-Module Residency Constraints,” *IEEE Trans. Semiconduct. Manufact.*, vol. 15, no. 3, pp. 341–349, 2002.

[7] ——, “An Optimal Residency-Aware Scheduling Technique for Cluster Tools With Buffer Module,” *IEEE Trans. Semiconduct. Manufact.*, vol. 17, no. 1, pp. 68–73, 2004.

[8] Q. Su and F. Chen, “Optimal Sequencing of Double-Gripper Granty Robot Moves in Tightly-Coupled Serial Production Systems,” *IEEE Trans. Robot. Automat.*, vol. 12, no. 1, pp. 22–30, 1996.

[9] Y. Crama, V. Kats, J. van de Klundert, and E. Levner, “Cyclic Scheduling of Robotic Flowshops,” *Annals of Operations Research*, vol. 96, no. 1, pp. 97–124, 2000.

[10] D. Jevtic, “Method and aparatus for managing scheduling a multiple cluster tool,” European Patent 1,132,792 (A2), Dec., 2001.

[11] D. Jevtic and S. Venkatesh, “Method and aparatus for scheduling wafer processing within a multiple chamber semiconductor wafer processing tool having a multiple blade robot,” U.S. Patent 6,224,638, May, 2001.

[12] M. Dawande, C. Sriskandarajah, and S. Sethi, “On Throughtput Maximization in Constant Travel-Time Robotic Cells,” *Manufacturing & Service Operations Management*, vol. 4, no. 4, pp. 296–312, 2002.

[13] S. Ding and J. Yi, “An Event Graph Based Simulation and Scheduling Analysis of Multi-Cluster Tools,” 2004, To be presented at the 2004 Winter Simulation Conference.

APPENDIX
PROOF OF PROPOSITION 2

The fictitious processing time $t^*_{i(N_i+1)}$ is the minimal time delay due to the robot R_{i+1} and the buffer module B_i. We only sketch the proof for two scenarios: (1) $\mathbf{S}_i = 2$, and (2) $\mathbf{S}_i = 1$. In the proof sketch, we first calculate $\mathbf{FP}^*_i$ with $t^*_{i(N_i+1)} = 0$ and $t^*_{i0} = 0$ and then find out what the minimal value for $t^*_{i(N_i+1)}$ should be in order to let the decoupled single clusters $\mathbb{C}_i$ and $\mathbb{C}_{i+1}$ work exactly as the connected clusters.

(1) **Case 1**: For the first case, we try to prove that by setting $t^*_{i(N_{i+1})} = 0$ we can always coordinate robots R_i and R_{i+1}. If $\mathbf{FP}^*_i \geq \mathbf{FP}^*_{i+1}$, the calculated $\mathbf{FP}^*$ for $\mathbb{C}_i$ is greater than $\mathbb{C}_{i+1}$ as shown in Fig. 6 for the Gannt diagram. Consider a fixed scheduling π_i of robot R_i under which $\mathbf{FP}_i$ is optimal (minimum). Under π_i, we can always construct a schedule, π_{i+1}, of the transfer robot R_{i+1} of $\mathbb{C}_{i+1}$ such that no conflict happens to act on each buffer module B_{ij}, $j = 1, 2$: we can schedule R_{i+1} to “pick” and “place” a wafer to fictitious module $P^*_{i(N_i+1)}$ at any time within the time period ΔT_i (Fig. 6),

$$\Delta T_i = \left\{ t \in \mathbb{R}_+ \mid t_i^{\text{pick}} \leq t < t_i^{\text{pick}} + \mathbf{FP}^*_i - 2T_i \right\},$$

where t_i^{pick} is the time R_i finishes picking a wafer from $C^*_{(i)1}$. Since the length of ΔT_i is $\mathbf{FP}^*_i - 2T_i$ [4] and in order to schedule a pick/place pair for R_{i+1}, it must satisfy

$$2T_{i+1} \leq \mathbf{FP}^*_i - 2T_i, \tag{8}$$

where

$$\begin{aligned} \mathbf{FP}^*_i &= \max\{2(N_i + N^*_i + 1)T_i, t_i^{\max *} + 2T_i\} \\ &\geq 2(N_i + N^*_i + 1)T_i, \end{aligned} \tag{9}$$

$$\mathbf{FP}^*_{i+1} \geq 2(N_{i+1} + N^*_{i+1} + 1)T_{i+1}. \tag{10}$$

If $T_i \geq T_{i+1}$, then it is obvious that Ineq. (8) is always satisfied because of (9). If $T_i < T_{i+1}$, then from $\mathbf{FP}^*_i \geq \mathbf{FP}^*_{i+1}$ and Ineq. (10), we have

$$\mathbf{FP}^*_i - 2T_i \geq 2(N_{i+1} + N^*_{i+1})T_{i+1} \geq 2T_{i+1}\,.$$

Therefore, Ineq. (8) always holds. If $\mathbf{FP}^*_i < \mathbf{FP}^*_{i+1}$, in

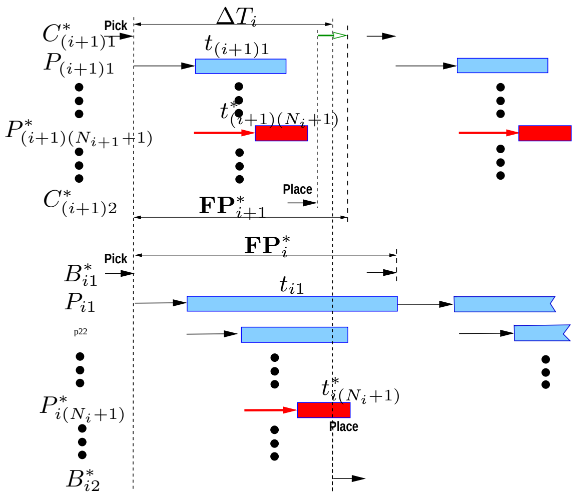

Fig. 6. Gannt diagram of two adjacent augmented single clusters with double-blade robots.

order to operate R_i and R_{i+1} without extra coordination time, similar to Ineq. (8), we need

$$2T_i \leq \mathbf{FP}^*_{i+1} - 2T_{i+1}, \tag{11}$$

we can prove that the inequality above always holds.

(2) **Case 2**: For the case when $\mathbf{S}_i = 1$, we can easily identify how long for robot R_{i+1} to refill B_i.

(i). If R_{i+1} is double-blade, then $2T_{i+1}$ is needed to swap wafer in B_i once wafer has been placed into B_i.

(ii). If R_{i+1} is single-blade and $N_{i+1} + N^*_{i+1} \geq 2$, then $4T_{i+1}$ is needed to pick and place another wafer into B_i since there are at least one PM in $\mathbb{C}_{i+1}$ can be used as a buffer to store a wafer.

(iii). If R_{i+1} is single-blade and $N_{i+1} + N^*_{i+1} = 1$, then the time needed to refill B_i is given by $\mathbf{FP}^*_{i+1} - t^*_{(i+1)0}$. When we calculate $\mathbf{FP}^*_{i+1}$ of $\mathbb{C}_{i+1}$, we consider the loading delay $t^*_{(i+1)0}$ of fictitious cassette module and we have to subtract it from $\mathbf{FP}^*_{i+1}$ in order to find how long a wafer can be refilled into B_i by R_{i+1}.

This concludes the Proposition.

[4]Here we consider the worst case when both robots are single-blade and they require $2T_i$ and $2T_{i+1}$ to access to B_{ij}, $j = 1, 2$, respectively.