2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Scheduling Analysis of Cluster Tools with Buffer/Process Modules

Jingang Yi
Dept. of Mech. Eng.
San Diego State University
San Diego, CA 92182
jgyi@mail.sdsu.edu

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

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

Mike Tao Zhang
AzFSM Industrial Eng.
Intel Corporation
Chandler, AZ 85248
mike.zhang@intel.com

***Abstract*— Modeling and scheduling of cluster tools are critical to improving the productivity and to enhancing the design of wafer processing flows and equipment for semiconductor manufacturing. In this paper, we extend the decomposition methods in [1] for multi-cluster tools with buffer/process modules (BPMs). The computation of the lower-bound cycle time (fundamental period) is presented. Optimality conditions and robot schedules that realize such lower-bound values are then provided using "pull" and "swap" strategies for single-blade and double-blade robots, respectively. The impact of BPMs on throughput and robot schedules is studied. It is found that such an impact depends on the BPM processing time and the cycle times of the decomposed clusters on both sides of BPMs. A chemical vapor deposition (CVD) tool is used as an example of multi-cluster tools to illustrate the proposed method, analysis, and algorithms. The numerical and experimental results demonstrate the effectiveness and efficiency of the algorithms.**

I. INTRODUCTION

Cluster tools are widely used as semiconductor manufacturing equipment. In general, a cluster tool consists of three types of modules (Fig. 1): cassette modules (CM) store the unprocessed and processed wafers, process modules (PM) execute semiconductor manufacturing processes, such as chemical vapor deposition (CVD), and transfer modules (TM), which are robot manipulators, move wafers among process modules and between process and cassette modules. During a semiconductor manufacturing process, wafers are transported by robots from the cassette module, sequentially go through various process modules, and then return to the cassette module.

We consider an inter-connected M-cluster tool shown in Fig. 2. A buffer module B_i between $\mathbb{C}_i$ and $\mathbb{C}_{i+1}$ (Fig. 2) is called a *buffer/process module (BPM)* if it also functions as a process module with processing time t_i^{BP}. The use of BPMs can make a cluster tool more compact and save the tool footprint and cost. The existence of BPMs could, however, affect the throughput and the robot schedule because its dual role as a process module could introduce a significant complexity in the analysis. We discuss robot scheduling of a multi-cluster tool with BPMs. Our goal is to find an optimal schedule that minimizes cycle time and, therefore, maximizes throughput.

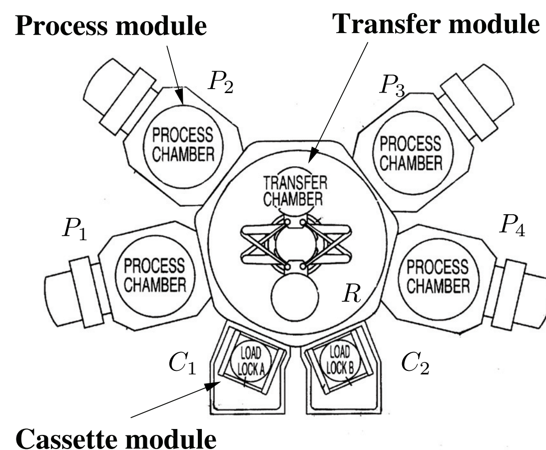


Fig. 1. A schematic of a cluster tool.

For cluster tools, robot movement and wafer processing sequences repeat cyclically at steady state. Like most of the literatures, we consider the cycle time for a *one-wafer action sequence* as the optimization objective. A *one-wafer action sequence* is defined as a sequence of robot actions which pick and place each module exactly once [2].

In [3], [4], analytical models of steady-state throughput are discussed for a cluster tool equipped with single-blade and double-blade robots. A single-blade robot can usually hold only one wafer at a time while a double-blade robot has two independent arms and, therefore, can hold two wafers at a time. For a single-blade robot, Perkinson *et al.* [3] propose a "pull" (or so-called downhill) optimal schedule strategy for the robot moving sequence. For a double-blade robot, Venkatesh *et al*. [4] propose one optimal schedule by a "swap" action. Dawande *et al.* [2] summarize the sequencing and scheduling in robotic cells, which is similar to cluster tools. Geismar *et al.* [5] extend the result in [6] and discuss the throughput and scheduling analysis of a robotic cell with a single-gripper robot and parallel stations. Ding *et al.* [7] extend the network model in [8] for a general study of a multi-cluster tool.

Recently, Geismar *et al.* [9] discussed a robotic cell with three single-gripper robots for semiconductor manufacturing.

1-4244-0602-1/07/$20.00 ©2007 IEEE.

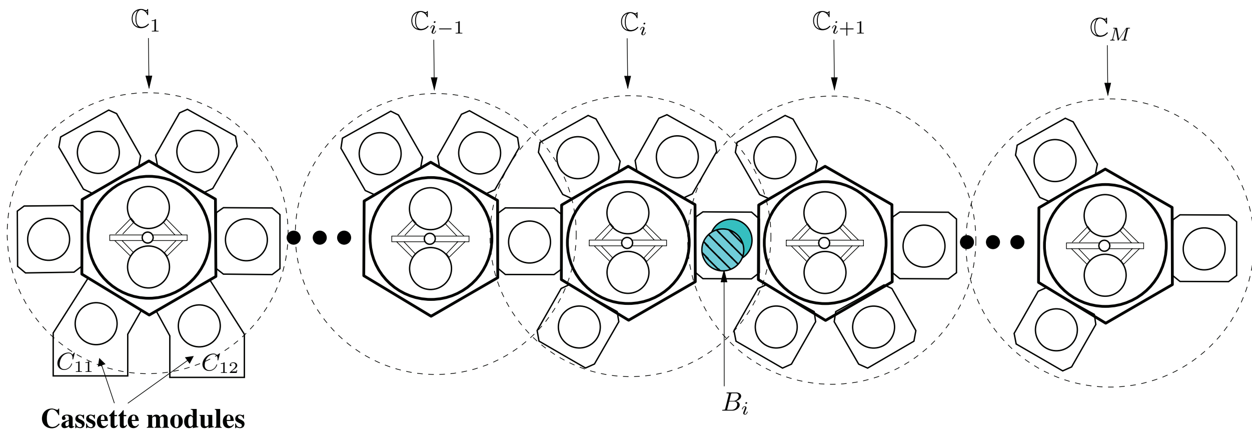


Fig. 2. A schematic of an inter-connected M-cluster tool.

In [7], an integrated event graph and network model is used to find all optimal schedules for a multi-cluster tool. In [10] and [11], several rule- or priority-based heuristic scheduling methods of robot actions of multi-cluster tools have been discussed. However, there are few analyses and comparison studies of those heuristic methods in terms of optimality. This paper extends the robot scheduling results in Yi *et al.* [1] for multi-cluster tools. The main goal of this study is to analytically investigate the throughput and robot scheduling of multi-cluster tools with BPMs.

The remainder of this paper is organized as follows. We present and extend the robot scheduling results for a single-cluster tool in section II. In section III, we present algorithms to compute the lower-bound of the minimal cycle time and discuss the optimality conditions and optimal robot schedules. In section IV, we analyze the buffer/process modules (BPMs). An example of throughput analysis and robot scheduling is investigated for a CVD tool in section V. Finally, we summarize with concluding remarks.

II. Cluster Tool Configurations and Single-Cluster Scheduling

A. Cluster tool configuration

For the M-cluster tool shown in Fig. 2, we assume that robot R_i, $i = 1, \cdots, M$, takes the same amount of time T_i to pick and place a wafer and that robot transfer time T_i and PM processing time t_{ij} are deterministic. We also consider the assumptions that (1) all wafers follow the identical visit flow $\mathbf{V}$, (2) cassette modules always have wafers/spaces, (3) each robot R_i is either single-bladed or double-bladed, (4) buffer module B_i (or BPM) has either one- or two-wafer capacity, and (5) each cluster must connect to at least one, but at most two other clusters, and these clusters cannot form a loop inter-connection.

We also call cluster $\mathbb{C}_i$ a *transfer cluster* if (1) R_i is a single-blade robot, (2) there is no process module in $\mathbb{C}_i$, and (3) both side buffer modules (or BPM) have one-wafer capacities. Due to the fact that there is not enough wafer storage space to flexibly move wafers, we will handle transfer clusters slightly different.

For the *cyclic production* pattern in which wafers are driven by a fixed sequence of robot actions, we can only consider the sequencing and timing of robot actions. We can denote the robot schedule π as a doublet of its actions ACT_i and their relative starting times ST_i in one cycle: π=$\{ACT_i, ST_i\}$, $i = 1, 2, \cdots, m$, and m is the total number of robot actions. We define the *optimal schedules* as the set of any repeated one-wafer cycle under which the throughput of the cluster tool is maximized. It is observed that if an optimal schedule π_o maximizes the throughput μ, it must minimize the cycle time $T(\pi)$. In this paper, we adopt the terminology "fundamental period," denoted as $\mathbf{FP}$ in the literature, for the minimal one-wafer cycle time [1], [3], [4], namely, $\mathbf{FP} = \min_{\pi} T(\pi)$.

B. Single-cluster tool optimal schedule

1) Maximal cassette waiting time strategy: A single-cluster tool could be running in two possible regions: *process-bound* and *transfer-bound* regions. For a single-blade robot, the robot "pull" strategy is optimal and for double-blade robot, the "swap" strategy is optimal [3], [4]. Considering a N-PM cluster tool, denote the maximum and minimum processing time as $t^{\max}$ and $t^{\min}$, respectively, $t^{\max} = \max_{1\leq j\leq N}\{t_j\}$, $t^{\min} = \min_{1\leq j\leq N}\{t_j\}$. If we consider the cassette module functions as a process module P_0 with zero processing time, i.e. $t_0 = 0$, then we can calculate the fundamental period $\mathbf{FP}$ as [1]

$$\mathbf{FP} = 2rT + \max_{1\leq j\leq N}\{t_j, t_0, 2(N+1-r)T\}, \qquad (1)$$

where $r = 1$ if R is double-blade and $r = 2$ if single-blade.

When the single-cluster robot scheduling is applied to a multi-cluster tool, it is important to analyze the robot idle time at the inter-connected buffer modules. We need to indeed consider how to allocate the robot idle time (or waiting time) at the cassette modules. We define the *robot cassette waiting time*, t^R, as the time lag of robot R between the moments when finishing the action "pick an unprocessed wafer from input cassette" and starting the subsequent action "place a processed wafer into output cassette." It is noted that the "pull" and "swap" robot strategies for a single-cluster tool

are maximal cassette waiting time strategies. We can also find alternative "pull" and "swap" strategies to minimize the robot cassette waiting time.

2) Minimal cassette waiting time strategy: We can consider scheduling robot R to wait as long as possible at the process module with the minimal processing time. We denote the robot waiting time at the PM with the minimum processing time as t_{idle}^{PM}. Let $\delta t = t^{\max} - t^{\min}$. Then the maximal robot waiting time at the module with the minimum processing time is $t_{idle}^{PM} = \min\{\delta t, t_{idle}\}$.

We propose the following minimal cassette waiting time "pull" and "swap" strategies that minimize t^R by allocating most of t_{idle} to the process module with minimal processing time: (1) Robot R's action sequence follows the maximal cassette waiting time "pull" and "swap" strategies, respectively. (2) Single-blade robot R waits t_{idle}^{PM} before moving an unprocessed wafer into the PM with processing time $t^{\min}$. (3) Double-blade robot R places a processed wafer into the cassette right after it picks an unprocessed wafer from the cassette.

Using either the minimal cassette waiting time "pull" strategy (for single-blade robot) or the "swap" strategy (for double-blade robot), we can show that the fundamental period $\mathbf{FP}$ (1) can be maintained unchanged while the robot cassette waiting time $t_{\min}^R$ is minimized as

$$t_{\min}^R = \begin{cases} 0 & \text{if } r = 1 \\ \max\{t^{\min} - 2(N-1)T, 0\} + 2T & \text{if } r = 2. \end{cases} \tag{2}$$

Without confusion, we will abuse the notation t^R to denote $t_{\min}^R$ in the rest of the paper unless explicitly indicated.

It is common that there may exist several identically parallel PMs that perform exactly the same functionality. We consider a single cluster tool $\mathbb{C}$ with N process steps. We denote l_i as the number of parallel modules for process P_i, $i = 1, \cdots, N$. Define the least common multiple (LCM) of l_i as $\lambda = \text{LCM}(l_1, \cdots, l_N)$. We can extend the results in [5] and re-write Eq. (1) as

$$\mathbf{FP} = 2rT + \max_{1 \le i \le N} \left\{ \frac{t_i + 2r(1-l_i)T}{l_i}, t_0, 2(N+1-r)T \right\}. \tag{3}$$

III. Optimal Scheduling of Multi-Cluster Tools

A. Computation of the lower-bound fundamental period

In [1], a decomposition method to decouple the interconnection among clusters is presented to analyze the steady-state performance and robot scheduling results. The key of the decomposition approach is to decouple the link between clusters. As shown in Fig. 2, for $\mathbb{C}_i$, we know that wafers flow in or out of the cluster through either buffer modules or cassette modules. $\mathbb{C}_i$ exchanges wafers with $\mathbb{C}_{i-1}$ through B_{i-1}, $i > 1$. B_{i-1} plays dual roles: for $\mathbb{C}_i$, B_{i-1} acts like a fictitious cassette module; for $\mathbb{C}_{i-1}$, on the other hand, it acts like a fictitious process module.

We can consider decoupling a multi-cluster tool into a set of independently running single-cluster tools by treating buffer modules as either fictitious cassette modules or fictitious process modules. We can then find the minimal fundamental period $\mathbf{FP}_i^{d*}$ [1] for each $\mathbb{C}_i$. After we obtain the set $\{\mathbf{FP}_i^{d*}\}, i = 1, \cdots, M$, we can identify the lower-bound value as the largest $\mathbf{FP}_i^{d*}$, which will determine $\mathbf{FP}$ for the entire system.

Assume that a decoupled $\mathbb{C}_i$ has $N_i \ge 0$ PMs and denote the fictitious process module as the (N_i+1)th PM, denoted by $P_{i(N_i+1)}^*$. We assume that $P_{i(N_i+1)}^*$ has a fictitious processing time $t_{i(N_i+1)}^{d*}$ and the fictitious cassette module C_i^* has a wafer supply time t_{i0}^{d*}. From Eq. (1), we can obtain $\mathbf{FP}_i^{d*}$ as follows.

$$\mathbf{FP}_i^{d*} = \begin{cases} 4T_i + t_{i(N_i+1)}^{d*} + t_{i0}^{d*} & \text{if } \mathbb{C}_i \text{ is a transfer cluster} \\ 2r_iT_i + t_i^{md} & \text{otherwise,} \end{cases} \tag{4}$$

where $t_i^{md} = \max_{1 \le j \le N_i} \{t_{ij}, t_{i(N_i+1)}^{d*}, t_{i0}^{d*}, 2(N_i+2-r_i)T_i\}$.

Fictitious cassette C_i^*'s supply time t_{i0}^{d*} can be considered as the minimum loading delay time of R_{i-1} [1]

$$t_{i0}^{d*} = \begin{cases} 2r_{i-1}T_{i-1} & \mathbf{S}_{i-1} = 1 \\ \max\{T_{i-1} - (2r_i - 1)T_i, 0\} & \mathbf{S}_{i-1} = 2. \end{cases} \tag{5}$$

The value of $t_{i(N_i+1)}^{d*}$ depends on the minimal loading time delay at $P_{i(N_i+1)}^*$. Similarly, we can calculate $t_{i(N_i+1)}^{d*}$ as

$$t_{i(N_i+1)}^{d*} = \begin{cases} 2r_{i+1}T_{i+1} + t_{i+1}^R & \mathbf{S}_i = 1 \\ \max\{T_{i+1} - (2r_i - 1)T_i, 0\} & \mathbf{S}_i = 2. \end{cases} \tag{6}$$

B. Optimality conditions of the lower-bound fundamental period

The lower-bound fundamental period $\mathbf{FP}$ computed in the previous section might not be realized due to the fact that we use a minimal time interaction between adjacent clusters in the computations. Therefore, it is natural to ask what are the optimality conditions under which the computed $\mathbf{FP}$ is feasible, and how can an optimal robot schedule under these conditions be found. We have the following results [2].

Proposition 1: For an M-cluster tool, the computed fundamental period $\mathbf{FP}$ in the previous section is feasible if for each cluster $\mathbb{C}_k$, $k > 1$, the minimal robot cassette waiting time t_k^R satisfies the following condition

$$t_k^R \le \begin{cases} \mathbf{FP} - 2r_{k-1}T_{k-1} - 2T_k & \text{if } \mathbf{S}_{k-1} = 1 \\ 2\mathbf{FP} - 2r_{k-1}T_{k-1} - 2T_k & \text{if } \mathbf{S}_{k-1} = 2. \end{cases} \tag{7}$$

C. Robot scheduling

The following is a "no waiting" schedule that has been implemented to achieve $\mathbf{FP}$: once the wafer has been placed into the process module, the process can start right away. For such a schedule, each process starting time is completely dependent on the robot action starting time. For robot R_i and decoupled $\mathbb{C}_i$, we denote its schedule as π_i. After a proper timing shift of π_i's starting time by the inter-connection relationships, they can be fitted into a multi-cluster schedule π with fundamental period $\mathbf{FP}$. Algorithm 1 describes an

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

[2] Due to the page limit, we neglect all propositions' proofs in this paper.

algorithm to find optimal robot schedule. In Algorithm 1, m_i denotes the the number of robot R_i actions of decoupled $\mathbb{C}_i$.

Algorithm 1: A "no-wait" optimal robot scheduling.

Input : Cluster tool configuration, wafer flow $\mathbf{V}$, and fundamental period $\mathbf{FP}_m$
Output: Scheduling $\boldsymbol{\pi}$ for $\mathbf{V}$
Obtain the decomposed schedule $\pi_i = \{ACT_i^j, ST_i^j\}$, $j = 1, \cdots, m_i$, for cluster $\mathbb{C}_i$, $i = 1, \cdots, M$, using (a) "swap" strategy ($r_i = 1$) or (b) the minimal cassette waiting time "pull" strategy ($r_i = 2$).
Initialize system schedule as $\boldsymbol{\pi} \leftarrow \pi_1$.
for $i = 2$ *to* M **do**
  Search for $ACT_{i-1}^s \in \pi_{i-1}$ that picks wafers from B_{i-1}/BP_{i-1}. Mark ACT_{i-1}^s starting time as ST_{i-1}^s.
  $T_i^{\text{shift}} \leftarrow ST_{i-1}^s - 2T_i$.
  for $j = 1$ *to* m_i **do**
    Update $ST_i^j \leftarrow ST_i^j + T_i^{\text{shift}}$.
  end
  $\boldsymbol{\pi} \leftarrow \boldsymbol{\pi} + \pi_i$.
end

IV. Buffer/Process Modules (BPMs)

We consider that there is a BPM BP_k between $\mathbb{C}_k$ and $\mathbb{C}_{k+1}$,$1 \leq k \leq M-2$, (Fig. 3). BP_k has either a one- or two-wafer capacity. We denote the incoming BPM (wafer flow from $\mathbb{C}_k$ to $\mathbb{C}_{k+1}$) as BP_{k1} and the outgoing BPM (wafer flow from $\mathbb{C}_{k+1}$ to $\mathbb{C}_k$) as BP_{k2}, respectively. If BP_k has a one-wafer capacity, $\mathbf{S}_k = 1$, then BP_{k1} and BP_{k2} share the same physical buffer device. If BP_k has a two-wafer capacity, $\mathbf{S}_k = 2$, BP_{k1} and BP_{k2} are independent buffer devices. For presentation simplicity, we use Fig. 3 to represent both cases. Let t_{k1}^{BP} and t_{k2}^{BP} be the processing time of BP_{k1} and BP_{k2}, respectively.

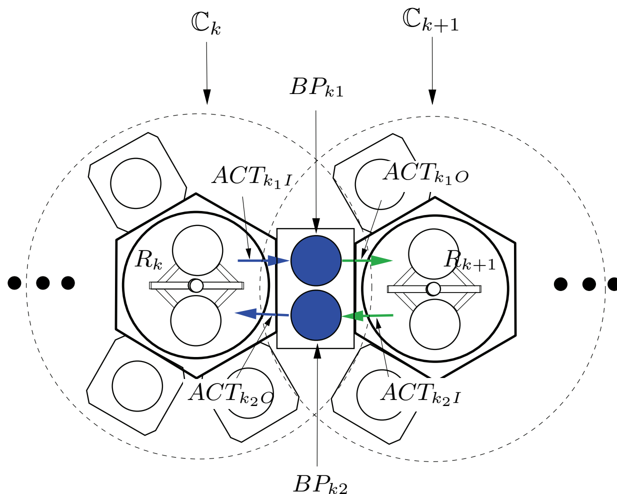


Fig. 3. A combined buffer/process module (BPM) with a two-wafer capacity.

We define the following notations $T_k^s = T_k + T_{k+1}$, $T_k^B = t_{k+1}^R + t_k^T$, and $t_k^T = 2(r_k - 1)T_k$, where t_k^R is the minimal robot cassette waiting time for the decoupled cluster $\mathbb{C}_k$. We can first compute the fundamental period by the decomposition algorithm in section III assuming that there were no BPM within the cluster tool, namely $t_{k1}^{BP} = t_{k2}^{BP} = 0$. We denote such a calculation as $\mathbf{FP}_0$. Depending on the BPM wafer capacity and processing times t_{k1}^{BP} and t_{k2}^{BP}, we can obtain the following results.

Proposition 2: For an M-cluster tool with a BPM BP_k between clusters $\mathbb{C}_k$ and $\mathbb{C}_{k+1}$, the fundamental period $\mathbf{FP}$ of the cluster tool can be calculated as

- If $\mathbf{S}_k = 1$

$$\mathbf{FP} = \begin{cases} \mathbf{FP}_0 & \text{if } t_{k1}^{BP} + t_{k2}^{BP} \leq \mathbf{FP}_0 - 2T_k^s - T_k^B \\ t_{k1}^{BP} + t_{k2}^{BP} + T_k^B + 2T_k^s, & \text{otherwise.} \end{cases} \quad (8)$$

- If $\mathbf{S}_k = 2$

$$\mathbf{FP} = \begin{cases} \mathbf{FP}_0 & \text{if } (t_{k1}^{BP}, t_{k2}^{BP}) \in \bigcup_{i=1}^{4} \mathcal{T}_i \\ \max\{t_{k1}^{BP}, t_{k2}^{BP}\} + T_k^s & \text{if } (t_{k1}^{BP}, t_{k2}^{BP}) \in \bigcup_{i=5}^{6} \mathcal{T}_i \\ \frac{t_{k1}^{BP} + t_{k2}^{BP} + T_k^B}{2} + T_k^s, & \text{if } (t_{k1}^{BP}, t_{k2}^{BP}) \in \mathcal{T}_7, \end{cases} \quad (9)$$

where $\mathcal{T}_1$-$\mathcal{T}_7$ are defined as Eqs. (17a)-(17g) on the next page (and graphically shown in the t_{k1}^{BP}-t_{k2}^{BP} plane in Fig. 4). Moreover, the "pull" strategy for single-blade robots and the "swap" strategy for double-blade robots can be used to achieve $\mathbf{FP}$ calculated above.

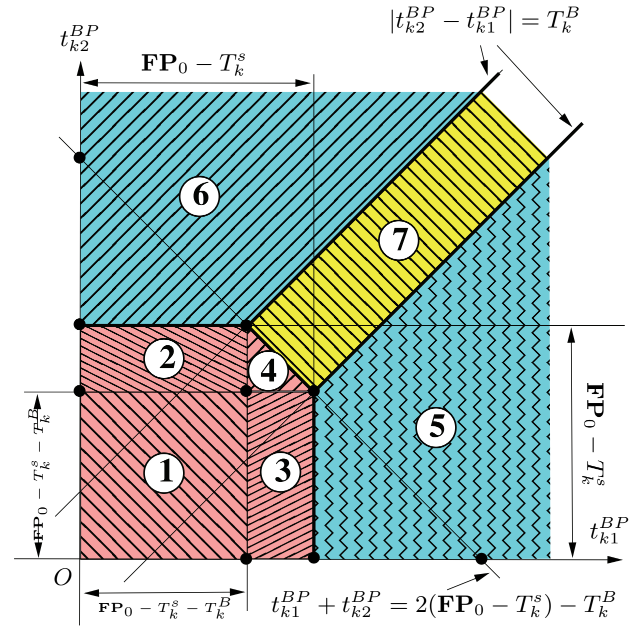


Fig. 4. $\mathbf{FP}$ calculation for different BPM process times t_{k1}^{BP} and t_{k2}^{BP} if $\mathbf{S}_k = 2$.

The BPM analysis can be integrated into the fundamental period computation algorithms discussed in the previous

$$\mathcal{T}_1 = \Big\{(t_{k1}^{BP}, t_{k2}^{BP}) \in \mathbb{R}_+^2 \big| t_{ki}^{BP} \le \mathbf{FP}_0 - T_k^s - T_k^B, i = 1, 2\Big\}, \tag{17a}$$

$$\mathcal{T}_2 = \Big\{(t_{k1}^{BP}, t_{k2}^{BP}) \in \mathbb{R}_+^2 \big| t_{k1}^{BP} \le \mathbf{FP}_0 - T_k^s - T_k^B, \text{ and } \mathbf{FP}_0 - T_k^s - T_k^B \le t_{k2}^{BP} \le \mathbf{FP}_0 - T_k^s\Big\}, \tag{17b}$$

$$\mathcal{T}_3 = \Big\{(t_{k1}^{BP}, t_{k2}^{BP}) \in \mathbb{R}_+^2 \big| t_{k2}^{BP} \le \mathbf{FP}_0 - T_k^s - T_k^B, \text{ and } \mathbf{FP}_0 - T_k^s - T_k^B \le t_{k1}^{BP} \le \mathbf{FP}_0 - T_k^s\Big\}, \tag{17c}$$

$$\mathcal{T}_4 = \Big\{(t_{k1}^{BP}, t_{k2}^{BP}) \in \mathbb{R}_+^2 \big| \mathbf{FP}_0 - T_k^s - T_k^B \le t_{ki}^{BP} \le \mathbf{FP}_0 - T_k^s, i = 1, 2, \text{ and } t_{k1}^{BP} + t_{k2}^{BP} \le 2(\mathbf{FP}_0 - T_k^s) - T_k^B\Big\}, \tag{17d}$$

$$\mathcal{T}_5 = \Big\{(t_{k1}^{BP}, t_{k2}^{BP}) \in \mathbb{R}_+^2 \big| t_{k1}^{BP} > \mathbf{FP}_0 - T_k^s, \text{ and } t_{k1}^{BP} - t_{k2}^{BP} \ge T_k^B\Big\}, \tag{17e}$$

$$\mathcal{T}_6 = \Big\{(t_{k1}^{BP}, t_{k2}^{BP}) \in \mathbb{R}_+^2 \big| t_{k2}^{BP} > \mathbf{FP}_0 - T_k^s, \text{ and } t_{k2}^{BP} - t_{k1}^{BP} \ge T_k^B\Big\}, \tag{17f}$$

$$\mathcal{T}_7 = \Big\{(t_{k1}^{BP}, t_{k2}^{BP}) \in \mathbb{R}_+^2 \big| t_{k1}^{BP} + t_{k2}^{BP} > 2(\mathbf{FP}_0 - T_k^s) - T_k^B, \text{ and } \big|t_{k2}^{BP} - t_{k1}^{BP}\big| < T_k^B\Big\}. \tag{17g}$$

section. Suppose there exist Q BPMs within the M-cluster tool, where $Q \le M - 1$, and we denote the BPM indexing set as $\mathcal{Q}$. We can calculate the fundamental period of the cluster tool with BPMs based on Proposition 2. Algorithm 2 describes such a modified fundamental period calculation.

Algorithm 2: **FP** calculation of a cluster tool with BPMs.

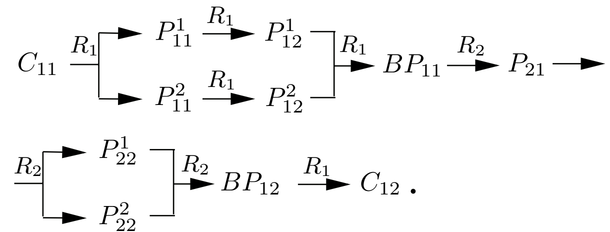
Input : Cluster tool configuration and wafer flow $\mathbf{V}$
Output: Fundamental period $\mathbf{FP}$ for $\mathbf{V}$
Calculate $\mathbf{FP}_0$ assuming $t_{ij}^{BP} = 0,\ i \in \mathcal{Q},\ j = 1, 2$
for $i \in \mathcal{Q}$ **do**
  Calculate $\mathbf{FP}_i$ for each BPM BP_i using Eqs. (8) or (9).
end
$\mathbf{FP} \leftarrow \max_{i \in \mathcal{Q}} \{\mathbf{FP}_i\}$.

For robot scheduling with BPMs, it is proper to schedule in the way such that BP_{k2} process ends right before action "R_k picking wafer from BP_{k2}" starts, and BP_{k1} process starts right right after action "R_k placing wafer into BP_{k1}" ends. Then in Algorithm 1, we can modify the calculation $T_i^{\text{shift}} \leftarrow ST_{i-1}^s - 2T_i - t_{(i-1)2}^{BP}$.

V. Experimental Examples

In this section, we demonstrate how to apply the proposed methodology to semiconductor manufacturing practice and show one example that has been used at Intel Corporation.

Thin film tools are widely used in semiconductor manufacturing to deposit metals onto a silicon wafer surface using either chemical vapor deposition (CVD), physical vapor deposition (PVD), sputter, etc. Fig. 5 shows a layout of an ALD/CVD cluster tool. This is a two-cluster tool. The service cluster $\mathbb{C}_1$ includes a double-blade robot R_1, cassettes C_{11} and C_{12}, and four process modules (chambers): parallel process modules P_{11}^i and P_{12}^i, $i = 1, 2$. The processing cluster $\mathbb{C}_2$ includes a double-blade robot R_2 and three process modules (chambers) with P_{22}^1 and P_{22}^2 are parallel modules. BP_{11} and BP_{12} are BPMs. All processing wafers follow the visit route (the split arrows indicate the flow at parallel PMs)



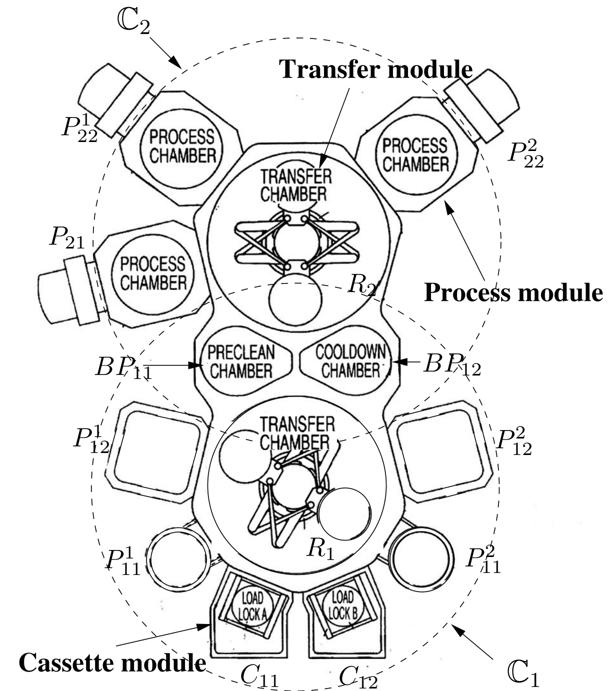


Fig. 5. A schematic layout of a CVD cluster tool.

TABLE I
Process and transfer time (in sec.) of the CVD cluster tool

T_1	T_2	t_{11}	t_{12}	t_{11}^{BP}	t_{12}^{BP}	t_{21}	t_{22}
10.9	10.5	78.7	144.3	0	68.6	64.4	230.6

The CVD cluster tool can be decomposed into two single clusters, $\mathbb{C}_1$ and $\mathbb{C}_2$, as shown in Fig. 5. We can directly

apply the decomposition technique discussed in section III to this two-cluster tool. We have to use Eq. (3) for parallel process modules in both clusters $\mathbb{C}_1$ and $\mathbb{C}_2$. For $\mathbb{C}_1$ and $\mathbb{C}_2$, the redundant level is $l_{11} = l_{12} = 2$, and from Eqs. (3)-(4), we have $\mathbf{FP}_1^{d*} = 83.05$ sec. $\mathbf{FP}_2^{d*} = 125.8$ sec. We can calculate the $\mathbf{FP}_0$ of the two-cluster CVD cluster tool as $\mathbf{FP}_0 = \max\{\mathbf{FP}_1^{d*}, \mathbf{FP}_2^{d*}\} = 125.8$ sec.

For BPMs BP_{11} and BP_{12}, we find that

$$\max\{t_{11}^{BP}, t_{12}^{BP}\} + T_1 + T_2 = 90 < \mathbf{FP}_0 = 125.8 \text{ sec.} \quad (18)$$

Therefore, by Eq. (9), BPMs BP_{11} and BP_{12} do not have any impact on the entire cluster tool throughput. Therefore, the fundamental period for the cluster tool is

$$\mathbf{FP} = \mathbf{FP}_0 = 125.8 \text{ sec.}. \quad (19)$$

From Proposition 1, we know that the computed $\mathbf{FP}$ for the CVD cluster tool is achievable. To illustrate the optimal schedule, we label all robot actions as in Table II. Following Algorithm 1, we compute the robot schedules[3] as shown in Table II, which complies with the calculated $\mathbf{FP}$. We further use an event-graph/network based method [7] to verify the optimal scheduling for the CVD cluster tool. The simulation gives the same results. The production at one Intel Corporation fab achieved a throughput of 28.6 wafers per hour (125.8 sec. cycle time per wafer) at the steady state. The production results further validate the analytical and simulation studies.

TABLE II
ACTION LABELS FOR THE CVD CLUSTER TOOL

ACT_i	Actions	Robot blade	Time (s)	Start time (s)
1	$C_{11} \to P_{11}$ pick	R_{11}/R_{12}	10.9	0/125.8
2	$C_{11} \to P_{11}^1$ place	R_{11}	10.9	32.7
3	$C_{11} \to P_{11}^2$ place	R_{12}	10.9	158.5
4	$P_{11}^1 \to P_{12}^1$ pick	R_{11}	10.9	21.8
5	$P_{11}^2 \to P_{12}^2$ pick	R_{12}	10.9	147.6
6	$P_{11}^1 \to P_{12}^1$ place	R_{11}	10.9	54.5
7	$P_{11}^2 \to P_{12}^2$ place	R_{12}	10.9	180.3
8	$P_{12}^1 \to BP_{11}$ pick	R_{11}	10.9	43.6
9	$P_{12}^2 \to BP_{11}$ pick	R_{12}	10.9	169.4
10	$P_{11}^1 \to BP_{11}$ place	R_{11}	10.9	76.3
11	$P_{11}^2 \to BP_{11}$ place	R_{12}	10.9	202.1
12	$BP_{11} \to P_{21}$ pick	R_{21}/R_{22}	10.5	−24.2/101.6
13	$BP_{11} \to P_{21}$ place	R_{21}/R_{22}	10.5	7.3/133.1
14	$P_{21} \to P_{22}^1$ pick	R_{21}	10.5	−3.2
15	$P_{21} \to P_{22}^2$ pick	R_{22}	10.5	122.6
16	$P_{21} \to P_{22}^1$ place	R_{21}	10.5	28.3
17	$P_{21} \to P_{22}^2$ place	R_{22}	10.5	154.1
18	$P_{22}^1 \to BP_{12}$ pick	R_{21}	10.5	17.8
19	$P_{22}^2 \to BP_{12}$ pick	R_{22}	10.5	143.6
20	$P_{22}^1 \to BP_{12}$ place	R_{21}	10.5	−13.7
21	$P_{22}^2 \to BP_{12}$ place	R_{22}	10.5	112.1
22	$BP_{12} \to C_{12}$ pick	R_{11}	10.9	65.4/191.2
23	$BP_{12} \to C_{12}$ place	R_{11}	10.9	10.9/136.7

VI. CONCLUSION

In this paper, we extended a decomposition method to analyze the steady-state throughput and robot scheduling of a multi-cluster tool with buffer/process modules for semiconductor manufacturing. We first extended the existing single-cluster scheduling results with robot minimal cassette waiting time "pull" and "swap" strategies for single- and double-blade robots. Based on these extensions, we discussed the lower-bound of the fundamental period of multi-cluster tools and optimality conditions under which such a lower-bound cycle time is feasible. Algorithms to compute the maximum throughput and to schedule the robots were proposed and analyzed. The impact of the combined buffer/process modules (BPMs) on cluster tool throughput and scheduling depends on the BPM processing time and decomposed cluster cycle times on both sides of BPMs. The proposed analytical and computational approach provided an efficient systematic method to study the throughput and scheduling of multi-cluster tools. An example of a CVD cluster tool at Intel Corporation was used to illustrate the proposed decomposition methods.

REFERENCES

[1] J. Yi, S. Ding, and D. Song, "Steady-State Throughput and Scheduling Analysis of Multi-Cluster Tools for Semiconductor Manufacturing: An Decomposition Approach," in *Proc. IEEE Int. Conf. Robotics Automation*, Barcelora, Spain, 2005, pp. 293–299.

[3]The starting time for some robot actions (e.g. ACT_1) in Table II are listed twice because of the 2-wafer cycle due to the existence of several two-parallel PMs.

[2] M. Dawande, H. Geismar, S. Sethi, and C. Sriskandarajah, "Sequencing and Scheduling in Robotic Cells: Recent Developments," *J. Scheduling*, vol. 8, pp. 387–426, 2005.

[3] 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.

[4] 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.

[5] N. Geismar, M. Dawande, and C. Sriskandarajah, "Robotic Cells with Parallel Machines: Throughput Maximization in Constant Travel-Time Cells," *J. Scheduling*, vol. 7, pp. 375–395, 2004.

[6] T. Perkinson, R. Gyurcsik, and P. McLarty, "Single-Wafer Cluster Tool Performance: An Analysis of the Effects of Redundant Chambers and Revisitation Sequences on Throughput," *IEEE Trans. Semiconduct. Manufact.*, vol. 9, no. 3, pp. 384–400, 1996.

[7] S. Ding, J. Yi, and M. T. Zhang, "Multi-Cluster Tools Scheduling: an Integrated Event Graph and Network Model Approach," *IEEE Trans. Semiconduct. Manufact.*, vol. 19, no. 3, pp. 339–351, 2006.

[8] J. Herrmann, N. Chandrasekaran, B. Conaghan, M. Nguyen, G. Rubloff, and R. Zhi, "Evaluating the Impact of Process Changes on Cluster Tool Performance," *IEEE Trans. Semiconduct. Manufact.*, vol. 13, no. 2, pp. 181–192, 2000.

[9] N. Geismar, C. Sriskandarajah, and N. Ramanan, "Increasing Throughput for Robotic Cells with Parallel Machines and Multiple Robots," *IEEE Trans. Automat. Sci. Eng.*, vol. 1, no. 1, pp. 84–89, 2004.

[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] Y. Crama and J. van de Klundert, "Cyclic Scheduling of Identical Parts in a Robotic Cell," *Opers. Res.*, vol. 45, no. 6, pp. 952–965, 1997.