Proceedings of the 3rd Annual
IEEE Conference on Automation Science and Engineering
Scottsdale, AZ, USA, Sept 22-25, 2007

On the Throughput of Clustered Photolithography Tools: Wafer Advancement and Intrinsic Equipment Loss

James R. Morrison, *Member, IEEE*, and Maruthi Kumar Mutnuri

Abstract — **For clustered photolithography tools we develop two analytic models characterizing the throughput. The models are essentially distinguished by the manner in which wafers advance through the tool and allow for many important features present in practical manufacturing systems. Such features include a diversity of lot populations (with different process times at each module) and disturbances to the ideal processing behavior such as delays to begin processing and delays incurred at specific modules. Our models thus allow us to quantify important classes of intrinsic equipment loss.**

I. Introduction

Semiconductor wafer fabrication facilities require extensive capital commitments to construct (recently rising to on the order of US$5 billion for a modern 300 mm wafer facility) and thus must be designed for efficiency and operated intelligently to ensure a timely return on the construction investment. One key target for efficient design and operation within such a facility is the photolithography scanner toolset which is typically the production bottleneck and the most expensive toolset per unit (on the order of US$20 million per scanner). Such tools are often clustered together with pre-expose and post-expose tracks, whose purpose is to coat the wafers prior to the photolithography scan via the application of photo resist chemicals and to develop the image imbedded in the resist subsequent to the scan. The collection of the pre-expose track, the scanner and the post-expose track is often referred to as a *clustered photolithography tool*. Though it is common for fabricators to develop simulation and static capacity models, there are few analytic models that have been developed to characterize their throughput performance which incorporate practical issues faced during production. The need for such models is emphasized by the International Technology Roadmap for Semiconductors (ITRS) which has increasingly incorporated fabricator performance objectives that address small lot sizes and a diversity of product types – both of which can have significant implications for the throughput of our toolset of interest.

Initial and final manuscripts received April 30, 2007 and July 31, 2007, respectively.

This work was supported in part by the Intel Corporation.

Please address all correspondence to the first author.

James R. Morrison is with the Department of Engineering and Technology at Central Michigan University (E-mail: morri1j@cmich.edu).

Maruthi Kumar Mutnuri (E-mail: mmaruthik@yahoo.com) is with the College of Business Administration at Central Michigan University.

Photolithography cluster tools consist of a collection of process stages through which each wafer must pass in sequential order. Each process stage is conducted by distinct modules within the tool that are dedicated to the process, though multiple modules may be devoted to a process stage to increase throughput potential. Once entering the tool, wafers from a lot follow one another from one stage to the next until they have received service from all the stages (there may be two or more wafers at a single stage to which multiple modules are devoted). The first wafer of a lot may proceed immediately behind the last wafer of the preceding lot, so that wafers from more than one lot can be in process (in different modules) at a time. We call this phenomenon *production parallelism*. In fact, for clusters with a large number of modules in relation to the lot size (e.g., twenty seven modules and twelve wafers per lot) it is possible for wafers from three or more lots to be in service.

Existing queueing models are limited in their applicability as they generally fail to address the production parallelism, interactions between different classes of product lots and operating conditions. Typical queueing models are based on the G/G/m-queue subject to tool failures. As analysis of these models is difficult, approximations such as those of [1] and [2] have been employed to obtain analytic expressions for the mean cycle time. To incorporate the production parallelism inherent in clustered photolithography tools, [3] and [4] essentially considered the first module as the server and treated the remaining process time as an independent post production delay before the lot was allowed to exit the system. As such, though the production parallelism was incorporated, the queueing models in [3] and [4] did not allow for possible interactions caused by diverse production time requirements among the modules and for different classes of product lots.

Petri nets have been employed to model, design and optimize semiconductor manufacturing cluster tools, see for example [5] or [6]. Though existing approaches could be used to model clustered photolithography tools, (under the typically deterministic timing assumption) they are less tractable for analyzing the throughput with random disturbances to their nominally periodic operation. In addition, the intent of such models is more to optimize the robot task sequence, a problem that we do not consider here. The work of [3] does begin to develop models along these lines; however, they essentially assume that the process times in the modules are a fixed constant (which is

1-4244-1154-8/07/$25.00 ©2007 IEEE.

not true of the actual tools). Flow shop models (or their Petri-net equivalents) could be used to model such tools, though the existing results are primarily focused on algorithms for optimizing the order in which the lots are to be processed (see, for example, the texts [7, 8, 9] and recent work such as that in [10]).

In the industry, simulation models and static capacity models have been used to assess toolset performance and predict throughput. Such models are eminently applicable; however they suffer from disadvantages. Simulation models do not provide formula from which an intuitive understanding of the system performance may be gleaned. Static capacity models do not typically include important features of the system performance, favoring ease of implementation over the capability to model system components and interactions.

Our objective is to analytically characterize the tool performance from the perspective of the external operation as opposed to attempting to optimize the internal logistics of the robot movement. The analyses will allow us to characterize the throughput and evaluate the so-called *intrinsic equipment loss* (throughput reduction) which may derive from numerous sources including multiple classes of lots, delay to enter the tool and reticle removal from or insertion to the scanner. To achieve our objectives, we develop two models to analytically characterize the throughput behavior of clustered photolithography tools. Each model supposes a different underlying design for wafer advancement (though we suppress the explicit dependence upon the robot task sequence and include wafer transport time in the process time, so that our models provide an upper bound on the throughput performance). The models readily account for the typical aperiodic, transient and failure prone environment in which such tools operate.

The paper is organized as follows. In Section II, we describe the general system and its operation. The features of and throughput results for two models, distinguished by the manner in which wafers advance, are detailed separately in Sections III and IV. Section III studies synchronous wafer advancement (which is related to the popular backward sequence of robot task scheduling) and Section IV studies asynchronous wafer advancement (which may serve as an upper bound on system performance for all robot task sequences). Examples of the results for Model I and Model II are presented in Sections V and VI, respectively. Concluding remarks are presented in Section VII.

II. GENERAL SYSTEM DESCRIPTION

Our models of a clustered photolithography tool may be considered a serial processing cluster tool, or a variant of a flow shop. There are M modules, labeled $m_1, m_2, \ldots, m_M$ at which wafers must receive service. Each module conducts a unique operation on the wafer processed in that module. The wafers require service in sequential order from the modules, so that, after receiving processing at module m_i, a wafer proceeds next to module m_{i+1}, unless $i = M$ after which the wafer exits the tool. As the wafers undergo different operations in different modules, the process time required by a wafer from module m_i may be different from module m_j. Further, each production lot (a collection of W wafers with the exact same processing requirements) may be of a different class than other lots. That is, the production time required to process a wafer of lot class ℓ in module m_j is denoted as Δ^{ℓ}_j. Figure 1 depicts a clustered photolithography tool with fifteen modules ($M = 15$). Two wafer handling robots are depicted, though we include handling time in the module process times and otherwise ignore the robots.

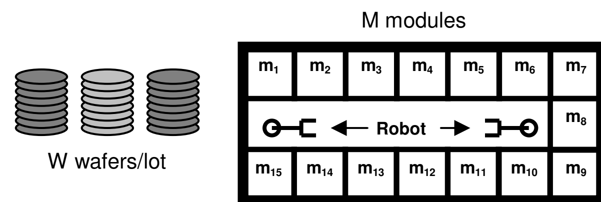

Figure 1. A clustered photolithography tool.

Though we stated in this Section that each module provides a distinct operation, it is quite common for redundant modules to be employed. *Redundant modules* provide an alternate path for wafers requiring a particular operation and allow for a potential increase in the throughput. This can be readily included in our model via the approach discussed in [3]. In addition, modules which are present solely for the purpose of holding a wafer between stages of production, termed *buffer modules* (e.g., a wafer cassette module prior to the scanner), are easily incorporated by introducing one module for each such buffer module with a production time $\Delta^{\ell}_j = 0$. Any of the modules in Figure 1 may thus be a buffer module (there are typically 12 to 24 prior to the scanner in practical tools).

Lot production proceeds as follows. When a lot of W wafers arrives at the tool, it waits until all wafers from lots of higher priority have received service from the first module m_1. At that instant, the first wafer of that lot may enter the tool to begin production. If the first module is vacant when the lot arrives, it may enter production immediately subject to the restrictions stated below. A lot is complete when its last wafer completes processing from the last module m_M and exits the tool. We do not specify the priority of the lots (i.e., the order in which they are to receive attention from the tool), and assume that this has already been determined.

In practical manufacturing systems, activities such as tool qualifications, setup times, resist dispenser dumps and routine tool maintenance impinge upon the attainment of throughput. These generally unavoidable disruptions reduce the throughput potential of the tool as do complete tool failures or module failures and contribute to intrinsic equipment loss. To study the effect of such activities as well as to characterize the influence of diverse lot populations, we deduce the time between lot completions.

Let the lots be indexed by the order in which they enter the tool, so that lot ℓ_i is preceded by lot ℓ_{i-1} and succeeded by lot ℓ_{i+1}. Denote the class of lot ℓ_i as $\mathcal{C}(i)$. Let b_i and c_i denote the time instants that lot ℓ_i begins and completes service, respectively, so that the total production time for lot ℓ_i is $P_i = c_i - b_i$. Let

$$T_i := \min\{ c_i - c_{i-1}, P_i \}$$

denote the time between the completion of the previous lot and lot ℓ_i, unless lot ℓ_i enters production on an idle tool, in which case use the production time. For throughput calculations, T_i may be used as the amount of time associated with the production of the W wafers in lot ℓ_i.

III. Model I: Synchronous Wafer Advancement

In the first model, we assume that the robot wafer transfer sequence has been designed such that wafers are required to advance in synchronization. That is, we impose the following two assumptions.

Assumption A1: Wafer advancement. If any module in the cluster contains a wafer, all wafers must wait until all modules have completed their tasks. At that time epoch, all wafers advance simultaneously.

Assumption A2: The first module. If any module in the cluster is providing service, a wafer may only begin production in module m_1 at a time epoch corresponding to the wafer advancement of Assumption A1.

Assumptions A1 and A2 may be essentially implemented by using the popular backward sequence of wafer movement (proven to be optimal in some contexts, see [11]). The throughput in Model I is an upper bound on the throughput for the backward sequence robot protocol.

To simplify the exposition, we impose one additional assumption. It can be removed at the expense of increased notation.

Assumption A3. Limit on wafers per lot. At most two lots may receive service from the cluster at a given time, that is, $W \geq M - 1$.

We introduce further notation. In the event that the lot queue is empty, the tool is returning from failure, lot production is delayed pending a recipe setup or myriad other reasons, there may be empty modules between wafers receiving production. Let k(i) denote the number of empty modules between the first wafer of lot ℓ_i and the last wafer of the preceding lot, if there is one, otherwise let k(i) = M-1. Thus, when lot ℓ_i enters production on a completely empty tool k(i) = M-1. If lot ℓ_i begins production immediately subsequent to its predecessor then k(i) = 0. Noting that k(i) may be used to model the duration of events such as tool failures and setup events, Theorem 1, stated next, allows us to assess their consequences.

Theorem 1: Time between lot completions with diverse lot populations. Consider a serial processing cluster tool satisfying assumptions A1, A2 and A3. Let lot l_i of class $\mathcal{C}(i)$ wafers be succeeded by lot l_{i+1} of class $\mathcal{C}(i+1)$. Then

$$T_i = \sum_{j=M-k(i)}^{M-1} \Lambda^{C(i)}_{j,M+1} + (W-M+1)\Lambda^{C(i)} + \sum_{j=1}^{k(i+1)} \Lambda^{C(i)}_{0,j} + \sum_{j=2+k(i+1)}^{M} \max\left(\Lambda^{C(i)}_{0,j}, \Lambda^{C(i+1)}_{j-[1+k(i+1)],M+1}\right),$$

where

$$\Lambda^{C(i)}_{p,q} = \max_{\{r:1\leq r\leq p \text{ or } q\leq r\leq M\}} \left\{\Delta^{C(i)}_r\right\} \text{ and}$$

$$\Lambda^{C(i)} = \Lambda^{C(i)}_{p,p}, \forall p \in \{1,2,3...M\}.$$

Proof: With assumptions A1, A2 and A3, wafers proceed simultaneously once all wafers in the cluster have completed their process times. That is, only when the maximum over all occupied modules of the terms $\Delta^{\mathcal{C}}_j$ have elapsed do wafers advance. Thus, we have that the first summation is the time required for the first wafer of lot l_i to reach the last module m_M after the preceding lot l_{i-1} exits the tool. The next term is the time required for the last wafer of lot l_i to exit the first module. The second summation is the additional time until the first wafer of the succeeding lot l_{i+1} enters the first module. The third and final summation is the time required for last wafer of the lot l_i to exit the tool. □

For the case where there is only one class of lots, we can drop the dependence upon $\mathcal{C}(i)$ and obtain the following corollary.

Corollary 1: Time between lot completions with one class of lots. If all the lots are of the same class $\mathcal{C}$ then the time between the lot completions is given by,

$$T_i = \sum_{j=M-k(i)}^{M-1} \Lambda_{j,M+1} + (W-M+1)\Lambda + \sum_{j=1}^{k(i+1)} \Lambda_{0,j} + \sum_{j=2+k(i+1)}^{M} \Lambda_{j-[1+k(i+1)],j},$$

where

$$\Lambda_{p,q} = \max_{\{r:1\le r\le p \text{ or } q\le r\le M\}} \{\Delta_r\}, \text{ and}$$
$$\Lambda = \Lambda_{p,p}, \forall p \in \{1,2,3...M\}.$$

The theorem and corollary detail the interactions between lots, the consequences of time delays between lots and the diversity of module process times for Model I.

IV. Model II: Asynchronous Wafer Advancement

In the second model, we assume that the wafer transport system has been designed such that wafers may advance independently (until becoming delayed by an occupied module ahead of them). That is, rather than the Assumptions A1, A2 and A3 of Model I, we impose the following assumptions where $x_j(w)$ is the entry time of wafer w to module m_j. (Of course, lot ℓ_i contains wafers w = (i-1)M+1, …, iM.) Note that we now restrict attention to a single class of lots. This allows us to obtain analytic results beyond the basic iterative cycle time and/or throughput relations of Assumption 5.

Assumption A4: A single class of lots. The process time for a wafer in module m_j is Δ_j, independent of the lot.

Assumption A5: Asynchronous wafer advancement. A wafer in module m_i proceeds to the next module m_{i+1} (or exits the tool if i = M) once it completes processing and the next module is vacant. That is, letting $\ell(w)$ denote the index of the lot containing wafer w and a_i denote the arrival time of lot ℓ_i to the system,

$$x_1(w) = \max\{a_{l(w)}, x_2(w-1)\},$$
$$x_j(w) = \max\{x_{j-1}(w) + \Delta_{j-1}, x_{j+1}(w-1)\}, \text{ for } 2 \le j \le M-1,$$
$$x_M(w) = \max\{x_{M-1}(w) + \Delta_{M-1}, x_M(w-1) + \Delta_M\}.$$

Further notation is presented next. Let $\Delta = (\Delta_1, ..., \Delta_M)^T$ be the column vector of the module process times and let $e = (1, …, 1)^T$ be the vector of ones with the same dimension. Thus, $e^T\Delta$ is the sum of the process times for a wafer. Let $\Lambda = \max_j \{\Delta_j\}$ be the bottleneck process time. We will use B to denote the index of the bottleneck module, that is, the module m_B (with the smallest index in case two modules qualify as the bottleneck) such that $\Delta_B \ge \Delta_j$ for all modules j. Recall that the number of modules in the track is M.

For lots, recall that the completion time of a lot ℓ_i is given as c_i. Finally, recall that we are using a_i to denote the arrival time of lot ℓ_i to the system.

The following lemmas may be proved via induction.

Lemma 1: No module contention occurs after the bottleneck. In Model II, for $B \le j < M$, the start time of wafer w on module m_{j+1} is given as

$$x_{j+1}(w) = x_j(w) + \Delta_j.$$

For the last module, the time wafer w exits the tool equals $x_M(w) + \Delta_M$.

Lemma 2: Wafers enter each post-bottleneck module no earlier than Λ time units after their predecessor. In Model II, for j ≥ B, the start time of wafer w+1 on module j is constrained as

$$x_j(w+1) \ge x_j(w) + \Lambda.$$

For the last module, the time wafer w + 1 exits the tool is greater or equal to the completion time of wafer w plus Λ.

Lemmas 1 and 2 may be employed to prove Theorem 2 which is stated next. It characterizes the completion times of lots, thereby enabling the calculation of the time between lot completions. Recall that a_i is the arrival time of lot ℓ_i.

Theorem 2: Completion time of lot ℓ_i. For Model II, the completion time of lot ℓ_i is given as

$$c_i = \max\{a_i + e^T\Delta, c_{i-1} + \Lambda\} + (W-1)\Lambda,$$

with initial condition for an empty tool

$$c_i = a_i + e^T\Delta + (W-1)\Lambda.$$

Theorem 2 enables the iterative calculation of the completion time of a lot based on the preceding lot. The first term in the maximization of Theorem 2 may be interpreted as the time taken for the first wafer in the lot to exit the tool when that first wafer *does not* experience module contention with the preceding lot. The second term in the maximization is the time taken for the first wafer in the lot to exit the tool when the first wafer *does* experience module contention with the preceding lot. The last term is the time to complete the remaining wafers after the first wafer of the lot exits the tool.

One can readily deduce the time that should be attributed to lot ℓ_i for throughput calculations $T_i := \min\{ c_i - c_{i-1}, P_i \}$.

Corollary 2: A recursion for T_i. In Model II,

$$T_i = W\Lambda + \max\{0, e^T\Delta - \Lambda + \min\{0, a_i - c_{i-1}\}\}.$$

The theorem and corollary allow us to characterize the implications of late lot arrivals and can be used to model the effect of failures to load the lots promptly. Other events may reduce throughput as well, such as a failure or pause of a specific module. As an example, when a reticle (also called a mask, which holds the pattern that is to be scanned onto the wafer) is removed from the tool for maintenance or replacement, the scanner ceases all operation for the duration of the reticle removal. Thus, if a wafer is in the scanner during such an event, that wafer incurs a delay. In

fact, there are typically up to four wafers which may be in the scanner at a given moment. We model this as a delay experienced at the bottleneck module (it is easy to generalize when additional modules after the bottleneck are included in the pause, but not for modules prior to the bottleneck).

Let $\tau_R(r)$ and $d_R(r)$ denote the start time of the r-th reticle change (pause in the bottleneck) and its duration, respectively. Assume for simplicity that only the bottleneck module is paused by the reticle removal. The following corollary of Theorem 2 assesses the implications of a pause in the processing at the bottleneck module.

Corollary 3: Pause in the bottleneck module. Let i be the smallest index such that

$$\max\{a_i + e^T\Delta, c_{i-1} + \Lambda\} + (W-1)\Lambda > \tau_R(r) + \sum_{j=B+1}^{M} \Delta_j .$$

The completion time of all lots preceding lot ℓ_i are not changed by the r-th pause (employ the recursion of Theorem 2). The completion time of lot ℓ_i is

$$c_i = f_i + \max\{0, g_i\},$$

where

$$f_i = \max\{a_i + e^T\Delta, c_{i-1} + \Lambda\} + (W-1)\Lambda ,$$

is the original completion time of the lot (if there had been no pause) and

$$g_i = \min\left\{ d_R(r), \tau_R(r) + d_R(r) + W\Lambda + \sum_{j=B+1}^{M} \Delta_j - f_i \right\}.$$

The maximization term in the completion time in Corollary 3 is the delay introduced by the pause in the bottleneck module, if any.

V. Examples for Model I

We consider three examples to illustrate the application of the results for Model I.

Example 1: Consider a serial processing cluster tool with synchronous wafer advancement as in Model I that consists of eleven modules (M = 11). For wafers from lots of class 1, the module process times are $\Delta^1_1 = 20$, $\Delta^1_2 = 25$, $\Delta^1_3 = 40$, $\Delta^1_4 = 35$, $\Delta^1_5 = 30$, $\Delta^1_6 = 50$, $\Delta^1_7 = 15$, $\Delta^1_8 = 35$, $\Delta^1_9 = 45$, $\Delta^1_{10} = 20$ and $\Delta^1_{11} = 30$ seconds. For wafers from lots of class 2, the module process times are $\Delta^2_1 = 30$, $\Delta^2_2 = 35$, $\Delta^2_3 = 50$, $\Delta^2_4 = 45$, $\Delta^2_5 = 40$, $\Delta^2_6 = 60$, $\Delta^2_7 = 25$, $\Delta^2_8 = 45$, $\Delta^2_9 = 55$, $\Delta^2_{10} = 30$ and $\Delta^2_{11} = 40$ seconds. Suppose that a lot ℓ_0 has just exited the tool and the gap between it and lot ℓ_1 is one module (k(1) = 1). Let lot ℓ_1 of class 1 run before lot ℓ_2 of class 2 with a one module gap in front of lot ℓ_2 (k(2) = 1). We employ Theorem 1 to calculate the value of T_1 for different lot sizes. Table 1 depicts T_1 (in this case, the time between the departure of lot ℓ_0 and the departure of lot ℓ_1) as the lot size for all lots changes over the values W = 10, 11, 12, 13 and 14. The calculation details are omitted.

W - Wafers Per Lot	10	11	12	13	14
T1	590	640	690	740	790

Table 1. T_1 for lot ℓ_1 varies with the number of wafers.

Table 1 demonstrates that T_1 has constant slope in relation to the lot size W for the values considered. □

Example 2: Consider the serial processing cluster tool of Example 1. To study the behavior of T_1 as the number of empty modules prior to the lots changes, consider the cases k(1) = k(2) = 0, 1, 2, 3 and 4 (i.e., both k(1) and k(2) are the same and range from 0 to 4). Let there be ten wafers in each lot (W = 10). Further, assume that a lot ℓ_0 has just exited the tool and the gap between it and lot ℓ_1 is k(1). Theorem 1 may be employed to calculate T_1.

T_1 has a constant slope (with respect to k(i)) for values of k(i) from 0 to 3. The slope changes when we allow a greater interlot gap. It can be observed that as the number of empty modules increases to four, there is one time epoch where the bottleneck module m_6 is empty. Also note that the increase in T_1 in all cases here is less than both of the bottleneck processing times Δ^1_6 and Δ^2_6. Table 2 depicts the result (the calculation details are omitted.). □

k(i) - Empty Modules Between Lots	0	1	2	3	4
T1	550	590	630	670	705

Table 2. T_1 for lot ℓ_1 varies with the empty modules.

Example 3: Consider a serial processing cluster tool with synchronous wafer advancement as in Model I, eleven modules (M = 11) and a single class of lots. The module process times are $\Delta_1 = 20$, $\Delta_2 = 25$, $\Delta_3 = 40$, $\Delta_4 = 35$, $\Delta_5 = 30$, $\Delta_6 = 50$, $\Delta_7 = 15$, $\Delta_8 = 35$, $\Delta_9 = 45$, $\Delta_{10} = 20$ and $\Delta_{11} = 30$ seconds. Suppose that a lot ℓ_0 has just exited the tool and the gap between it and lot ℓ_1 is one module (k(1) = 1). Let lot ℓ_1 run before lot ℓ_2 with a one module gap in front of lot ℓ_2 (k(2) = 1). We employ Corollary 1 to calculate the value of T_1 for different lot sizes. Table 3 depicts T_1 (in this case, the time between the departure of lot ℓ_0 and the departure of lot ℓ_1) as the lot size for all lots changes over the values W = 10, 11, 12, 13 and 14. The calculation details are omitted. □

W - Wafers Per Lot	10	11	12	13	14
T1	545	595	645	695	745

Table 3. T_1 for lot ℓ_1 varies with the number of wafers.

VI. Examples for Model II

We consider the following example to illustrate the

application of the results for Model II.

Example 4: Consider a serial processing cluster tool with asynchronous wafer advancement as in Model II consisting of sixteen modules. The module process times for wafers from the single class of lots are $\Delta_1 = 20$, $\Delta_2 = 25$, $\Delta_3 = 40$, $\Delta_4 = 35$, $\Delta_5 = 30$, $\Delta_6 = 0$, $\Delta_7 = 0$, $\Delta_8 = 0$, $\Delta_9 = 0$, $\Delta_{10} = 0$, $\Lambda := \Delta_{11} = 50$, $\Delta_{12} = 15$, $\Delta_{13} = 35$, $\Delta_{14} = 45$, $\Delta_{15} = 20$ and $\Delta_{16} = 30$. There are thus sixteen modules ($M = 16$) with five exclusively devoted to serve as buffers ($m_6, \ldots, m_{10}$) since they have zero processing time. Suppose that there are three lots of five wafers each arriving to the system at times $a_1 = 0$, $a_2 = 320$ and $a_3 = 560$ seconds. Further, assume that the first lot arrived to an initially empty tool. We can employ Theorem 2 to recursively evaluate the completion times:

$c_1 = a_1 + e^T\Delta + (W-1)\Lambda = 545$,
$c_2 = \max\{ a_2 + e^T\Delta, c_1 + \Lambda\} + (W-1)\Lambda = 865$, and
$c_3 = \max\{ a_3 + e^T\Delta, c_2 + \Lambda\} + (W-1)\Lambda = 1115$.

Corollary 2 allows us to readily determine T_i as follows:

$T_2 = W\Lambda + \max\{0, e^T\Delta - \Lambda + \min\{0, a_2 - c_1\}\} = 320$, and
$T_3 = W\Lambda + \max\{0, e^T\Delta - \Lambda + \min\{0, a_3 - c_2\}\} = 250$.

If the first reticle change occurs at time $\tau_R(1) = 850$ seconds and has duration $d_R(1) = 100$ seconds, we can employ Corollary 3 to identify which lot is the first to have the potential to be delayed by the pause in the bottleneck module. Since $i = 3$ is the smallest index that satisfies the condition in Corollary 3, c_1 and c_2 are not changed and we adjust c_3. The subsequent lot ℓ_4 will use the adjusted c_3 to recursively determine c_4. We have $c_3 = f_3 + \max\{0, g_3\}$, where $f_3 = \max\{560+345, 865+50\} + (5-1)*50 = 1115$ and $g_3 = \min\{100, 850+100+250+145-1115\} = 100$. Thus $c_3 = 1215$ seconds. □

VII. Concluding Remarks

We have developed two models for the clustered photolithography tool which allow for features common in practical manufacturing. The models characterize key sources of intrinsic equipment loss.

Additional features could be incorporated into the models. In the case of synchronous wafer advancement (Model I), reticle change events and setup times prior to a lot should be considered. In the case of asynchronous wafer advancement (Model II), setup times prior to a lot should be included as well as a more general approach to reticle change events (allowing for modules upstream of the bottleneck to pause). For either model, it would also be of interest to study the effect of a failure in an arbitrary module. Another important task that should be undertaken is the development of recommendations for manufacturing operation based upon the model results.

Finally, the design of such a tool starting from its functional requirements, incorporating the requirement that it operate in a high mix and small lot size environment and account for the aperiodicity and transient behavior of a fabricator, should be undertaken. The design could then proceed to the development of robot wafer transfer mechanisms to implement the desired behavior.

References

[1] W. J. Hopp and M. L. Spearman, *Factory Physics: Foundations of Manufacturing Management, 2nd ed*. New York, NY: Irwin/McGraw-Hill, 2001.

[2] J. A. Buzacott and J. G. Shanthikumar, *Stochastic models of manufacturing systems*, Prentice Hall, Englewood Cliffs, 1993.

[3] J. R. Morrison and D. P. Martin, "Performance evaluation of photolithography cluster tools: Queueing and throughput models," to appear in *OR Spectrum*, 2007.

[4] J. R. Morrison and D. P. Martin, "Practical extensions to cycle time approximations for the G/G/m-queue with applications," to appear in *IEEE Transactions on Automation Science and Engineering*, 2007.

[5] H.-Y. Lee and T.-E. Lee, "Scheduling single-armed cluster tools with reentrant wafers flows," *IEEE Transactions on Semiconductor Manufacturing*, vol. 19, no. 2, pp. 226-240, 2006.

[6] W. M. Zuberek, "Cluster tools with chamber revisiting – modeling and analysis using timed Petri nets," *IEEE Transactions on Semiconductor Manufacturing*, Vol. 17, No, 3, pp. 333-344, 2004.

[7] R. W. Conway, W. L. Maxwell, and L. W. Miller, Theory of Scheduling. Reading, MA: Addison-Wesley, 1967.

[8] K. Baker, Introduction to Sequencing and Scheduling. New York, NY: Wiley, 1974.

[9] S. French, Sequencing and Scheduling. New York, NY: Wiley, 1982.

[10] I. Mourani, S. Hennequin and X. Xie, "Continuous Petri nets with delays for performance evaluation of transfer lines," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 3721-3726, April 2005.

[11] T.-E. Lee, H.-Y. Lee, and Y.-H. Shin, "Workload balancing and scheduling of a single-armed cluster tool," *Proceedings of the 5th APIEMS Conference*, Gold Coast, Australia, pp. 1–15, 2004.