Exit Recursion Models of Clustered Photolithography Tools for Fab Level Simulation

Jung Yeon Park, Kyungsu Park, and James R. Morrison, *Member, IEEE*

***Abstract*—In semiconductor wafer fabricators (fabs), clustered photolithography tools (CPTs) are often the bottleneck. With a focus on fab-level simulation, we propose a new class of equipment models for CPTs called exit recursion models (ERMs). These models are inspired by concepts from flow line theory. We describe the intuition behind ERMs and provide the parameterization and simulation equations. These ERMs are data-driven empirical models and we develop three types based on different data perspectives: 1) tool log; 2) wafer log; and 3) lot log. To assess the quality of the proposed models, we conduct three classes of simulation experiments. A detailed CPT model, an affine model, and an empirical flow line model are used as the baselines. We consider mean cycle time, lot residency time, throughput time, and computation time as our primary performance metrics. The results suggest that ERMs are more accurate and robust than the affine models for all metrics and sometimes rival the performance of the empirical flow line models considered. ERMs require about 1.9 times as much computation as an affine model and about 250 times less computation than an empirical flow line model. ERMs may be helpful to increase the accuracy of fab-level simulation results without significant additional computation.**

***Index Terms*—Fab-level simulation, clustered photolithography tools, cycle time, throughput.**

I. Introduction

SIMULATION of semiconductor wafer fabricators (fabs) can be used to help assess the implications of changes to capacity, operating practices and wafer mix. As clustered photolithography tools (CPTs) are very expensive and typically serve as the fab bottleneck, they play a large role in dictating the fab throughput and cycle time. It is especially important to model bottlenecks in system simulation because they often determine the throughput [1]. We develop a new class of CPT equipment models for use in this context called exit recursion models (ERMs). We use simulation to investigate their fidelity and computational complexity relative to benchmark models.

A. Fab-Level Simulation

Fab-level simulation is often employed, see [2]–[8], to support wafer fab efficiency efforts. As discussed in [2] and [9], selecting the appropriate level of detail for a model is vital for any fab-level simulation. While simple models are easier to understand and construct, they may omit important elements of the system behavior. Models that are more complex can be more accurate, at the cost of computation time [10]. A good model should be computationally tractable while maintaining an acceptable level of fidelity.

One alternative to simulation is to use queuing network models to analyze and optimize system behavior. Inside of such a model, the appropriate workstation model must be determined [11]–[14]. Queuing models rely on numerous simplifying assumptions. This makes analysis more tractable but renders the models less precise and unable to answer certain key questions of interest (e.g., which production schedule is superior?). Our focus here is on simulation models which are quite popular with practitioners.

B. CPT Equipment Models

Equipment models are an essential component of fab-level simulators. They are often constructed based on historical production log data (empirical data) and transform inputs, such as lot arrival time and the number of wafers in a lot, into outputs, such as lot start time and lot completion time. We focus on empirical models for CPTs.

There are several models of CPTs that have been considered in the literature for fab-level simulation. Affine models are popular for use in fab-level simulation and industry planning engines [9], [15], [16]. Though simulation software, such as *Autosched AP*, allows one to construct any type of equipment model, affine models are often the default choice to model tools in fab-level simulation due to computation constraints [17]. Flow lines have been studied for many years as models of manufacturing systems [18], [19] and have recently have been used to model CPTs [20]–[25]. Detailed models of CPTs include wafer transport robots [26]–[34].

For CPTs, [35] showed that affine models are accurate only for throughput and not for other important measures such as cycle time and lot residency time. Furthermore, they are inaccurate when used in conditions deviating from the training conditions; they are not robust. Flow lines, while accurate for the same metrics, demand about 470 times more computation time than an affine model.

C. Contribution and Organization

In this paper, we propose a new class of equipment models called exit recursion models (ERMs) of CPTs for use in

Manuscript received July 10, 2016; revised September 26, 2016; accepted October 5, 2016. Date of publication October 21, 2016; date of current version February 1, 2017. This work was supported by KAIST High Risk High Return Project under Grant N10150064. *(Corresponding authors: Kyungsu Park; James R. Morrison.)*

The authors are with the Department of Industrial and Systems Engineering, KAIST, Daejeon 34141, South Korea (e-mail: james.morrison@kaist.edu; jpark0@kaist.ac.kr; kyungsu@kaist.ac.kr)

Digital Object Identifier 10.1109/TSM.2016.2618894

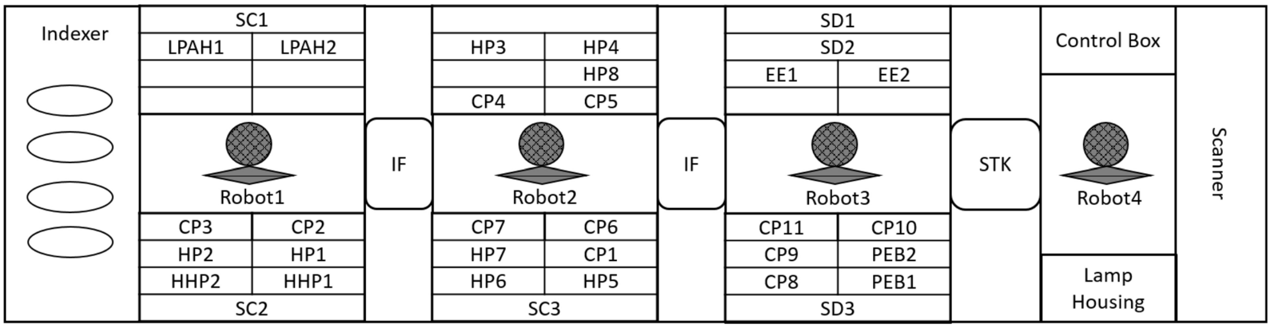

Fig. 1. Layout of a CPT.

fab-level simulation. These are based on intuition from flow line behavior and theoretical exit recursion results [36], [37]. We construct ERMs based on empirical data at three levels of detail: tool log (detailed wafer advancement from process to process), wafer log (wafer start and completion times at the entrance and exit of the tool), and lot log (lot start and completion times at the entrance and exit of the tool). We conduct simulations to assess their performance and computational complexity relative to the existing affine and empirical flow line models. We investigate the robustness of ERMs to changing fab conditions. The numerical experiments suggest that ERMs outperform affine models and are often comparable to flow line models, while only requiring about 1.9 times as much computation as an affine model.

The paper is organized as follows. In Section II, we describe the CPT system, detailed simulation model, affine model, and flow line model. The new ERMs, with varying levels of detail, are introduced in Section III. We describe the structure of our numerical experiments in Section IV and the detailed results in Section V. Concluding remarks are provided in Section VI.

Note that while we do not conduct fab-level simulation studies here, we discuss qualitatively how the use of ERMs is anticipated to improve their results in Section V-E.

II. Preliminaries

We now describe the detailed CPT, affine, and flow line models that serve as our benchmarks for the ERMs. Our focus is on empirical models.

A. Detailed CPT Model

A clustered photolithography tool (CPT) uses light to create patterns on the surface of wafers using a patterned mask (reticle). It consists of pre-scan processes, the scanner, and post-scan processes. There can be several redundant modules for each process. Wafers are transported to each step in their process flow via wafer transport robots under the guidance of a task allocation policy. Wafers arrive in batches, called lots, at the front of the CPT. There may be numerous kinds of setups; we consider pre-scan track and reticle alignment setups between lots. The average run length of lots of the same recipe is called the train size.

We use the CPT configuration from [26] based on data from the semiconductor industry. See Fig. 1. This CPT contains four clusters, which consist of process modules and robots. The labeled boxes are abbreviations of processes: spin coaters (SC), low pressure adhesions (LPAH), cold plates (CP), hot plates (HP/HHP), edge exposures (EE), post exposure bake hot plates (PEB), and spin developers (SD). There is a 16 wafer buffer that serves the scanner called the stacker (STK) and two one wafer interface buffers (IF) between clusters. There are three process flows with 16, 16, and 18 process steps, respectively. All of the process times are constant. For each recipe, the scanner is the bottleneck with a process time of 100 seconds per wafer. For our CPT, each process step (excluding the STK buffer) can only hold wafers of the same class, regardless of the number of available modules. The CPT does not have to be empty in order for a lot to start processing; several lots may be in process at a given time, which we term parallelism.

We use a detailed discrete-event simulation model (DS) of this CPT from [34] and [35]. We use this same detailed simulation model for all of our numerical experiments. We assume it is exact and use it as the baseline throughout this paper. Hereafter, we refer to the data obtained from this detailed model as our “true data” and call this the “true system”. All the other models are first parameterized using this true data, and then used for simulation.

B. Affine Model

Affine models (AF), also called $Ax+B$ models, are widely used equipment models for fab simulations. These models calculate the lot start and exit times from a tool using two parameters, A and B. The parameter A is the wafer throughput time (mean time between wafer exits from the tool) within a lot. The parameter B, the so called first wafer delay, is the time for the first wafer of a lot to complete processing. These constant values may depend on the previous and current lot classes. Parameterization and simulation equations are provided in [35].

Linear models have been used as well, see [17], [35], but we do not consider them as affine models are more accurate.

C. Empirical Flow Line

Flexible flow lines consist of a series of processes $P_1, P_2, \ldots, P_M$ from which wafers receive service sequentially. See Fig. 2 for an example of a flexible flow line.

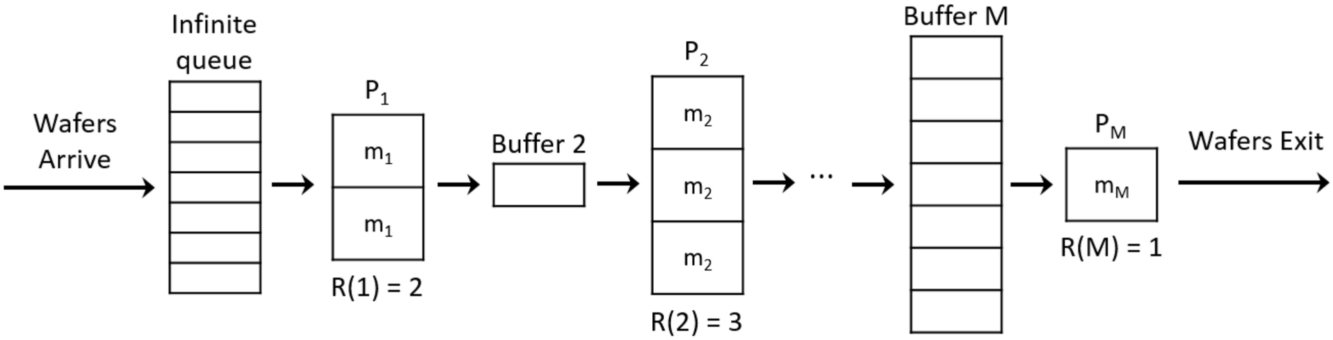

Fig. 2. Flexible flow line.

There may be redundant modules devoted to each process. In Fig. 2, P_1 has a redundancy *R(1)* = *2*. Finite buffers may be provided between each process.

We use an *empirical* flow line (EFL) as one of our benchmark models. EFLs are not given the deterministic module process times nor the wafer handling robot times. Rather they estimate the process times at each module using the true system's detailed tool log data. Once parameterized, the progression of wafers is described using the elementary evolution equations (EEEs); see [34] or [36]. EFLs have been shown to have good accuracy on all metrics considered; however, they require approximately 470 times more computation than AF. EFL parameterization and simulation equations are given in [35].

D. Metrics

Let a_i, S_i, and C_i be defined as the arrival time, processing start time and processing completion time of lot i at the tool, respectively. We consider cycle time, residency time, and throughput time for each lot i, respectively defined as

$$\begin{aligned} CT_i &= C_i - a_i \\ LRT_i &= C_i - S_i \\ TT_i &= min(C_i - S_i,\ C_i - C_{i-1}) \end{aligned}$$

with the initial condition $C_0 = -\infty$. The throughput time TT_i is the time from the exit of lot *i-1* to the exit of lot i, excluding any idle time on the tool during that interval.

Computation time is also of great importance. We measure the average CPU time required to calculate the start and completion times of all lots for a single replication of our simulation experiments after the parameters have been found.

III. Exit Recursion Models

Equipment models internal to a fab-level simulation should be accurate and fast. We next develop a new class of models called exit recursion models (ERMs) with improved fidelity and with similar computational requirements relative to AF. These models are inspired by concepts from flow line theory.

Early flow line research, see [36] and [37], demonstrated that the exit times of wafers from a deterministic flow line can be expressed in the form of an exit recursion:

$$E(w) = max\left(a_w + \sum_{m=1}^{M} \tau_m, E(w-1) + \tau_B\right) \tag{1}$$

for $w \geq 1$, where, a_w is the arrival time of wafer w to the flow line, *E(w)* is the exit time of wafer w, τ_m is the process time of process m, and B is the bottleneck process index. The initial condition is $E(0) = -\infty$. The throughput of wafers from the tool is dictated by the bottleneck process time τ_B.

The first term in the max of (1) characterizes the no bottleneck contention (NBC) case for an incoming wafer; it simply arrives and proceeds freely through the tool. The second term in the max of (1) characterizes the bottleneck contention (BC) case for a wafer. In that case, it departs one bottleneck process time after its predecessor.

Motivated by (1), ERMs contain two terms, an NBC term and a BC term, that are used to estimate the lot start and completion times. The true data is divided into NBC and BC cases, which are then used separately to extract the model parameters.

ERMs are empirical in that they use previous log data to determine their parameters (parameter extraction). These model parameters are then used in the model simulation equations. The model simulation equations predict the lot start and completion times, which can be used for important performance metrics such as cycle time or throughput.

When parameterizing a model with previous log data, there is a cost associated with accessing more detailed log data. Often, more effort and time is needed to analyze and interpret detailed log data for use in parameter extraction. Therefore, less detailed data is frequently used due to ease of access. Considering this possibility, we introduce three types of ERMs that differ according to the level of detail of the available log data: tool log, wafer log and lot log, which consequently have different parameter extraction equations.

In this section, we first provide some notation, then introduce the model simulation equations, and finally the parameter extraction equations.

A. Notation

Key notation for the true data is provided in Table I. Most are self-explanatory. For $\Omega(i, w)$, an example can help. Suppose *W(1)*=*W(2)*=*W(3)*=*25* wafers per lot, then

TABLE I
LIST OF NOTATION

Notation	Definition
L	Total number of lots
$W(i)$	Number of wafers in lot *i* (lot size)
k, k'	Indices for lot classes, $k, k' \in \{1, \dots, K\}$
k_i	Class of lot *i*
$L(k)$	Set of lot indices for all lots in class k, $L(k) = \{i \mid k_i = k\}$
$L(k, k')$	Set of lot indices for lots of class k preceded by a lot of class k', $L(k, k') = \{i \mid k_i = k, k_{i-1} = k'\}$
$\Omega(i, w)$	Overall wafer index of *w*-th wafer in lot *i*
B	Index of bottleneck process (scanner)
$R(k, m)$	Number of redundant modules for process *m* for class *k*
a_i	True arrival time instant of lot *i*
$X_{w,m}$	True entry time of wafer *w* into process *m*
L_i	True load time instant of lot *i*
S_i	True start time instant of lot *i*
C_i	True completion time instant of lot *i*
$B_{\Omega(i,w)}$	True start time instant of *w*-th wafer in lot *i* (instant that wafer begins processing at the first process)
$F_{\Omega(i,w)}$	True completion time instant of *w*-th wafer in lot *i* (instant that wafer finishes processing at the last process)

TABLE II
MODEL SIMULATION EQUATIONS FOR EXIT RECURSION MODELS

Vacation time of previous lot	$\tilde{V}_{i-1} = \begin{cases} \widetilde{V_m}(i-1), & k_i = k_{i-1} \\ \widetilde{V_p}(i-1), & k_i \neq k_{i-1} \end{cases}$
Load time	$\tilde{L}_i = \max(a_i, \tilde{V}_{i-1})$
Start time	$\tilde{S}_i = \tilde{L}_i + E^{k_i, k_{i-1}}$
Completion time	$\tilde{C}_i = \max\begin{pmatrix} \tilde{S}_i + FWD^{k_i} + A_1^{k_i}(W(i)-1), \\ \tilde{C}_{i-1} + B(k_i, k_{i-1}) + A_2^{k_i}(W(i)-1) \end{pmatrix}$
Vacation times of current lot	$\widetilde{V_m}(i) = \tilde{C}_i - D_m^{k_i}$ $\widetilde{V_p}(i) = \tilde{C}_i - D_p^{k_i}$

$\Omega(3, 7) = 57$. Load time L_i may be different from the start time S_i, if the tool is experiencing a setup. (We do not model load ports explicitly.)

The concepts of load time, start time, first wafer delay, vacation time and completion time are illustrated in Fig. 3. We use a single server flow line for simplicity; the concepts are easily transferable to a CPT. The load time and the start time are the time instants when a lot has been loaded into the tool and when the lot starts processing inside the tool, respectively. The first wafer delay is the time it takes for the first wafer of a lot to exit the tool once it starts processing. The vacation time is the time instant that a lot vacates the first process sufficiently to allow processing of the next lot. The completion time of a lot is the time instant when the last wafer finishes processing and exits the tool. These are important concepts that will be utilized in the formulation of ERMs.

B. Model Simulation Equations

We provide the model simulation equations for all ERMs in Table II. Given parameters such as the first wafer delay (FWD^{k_i}) and the throughput time for each wafer ($A_2^{k_i}$) and other inputs such as lot arrival times, wafers per lot, and lot class, the ERM model simulation equations estimate the lot start $\tilde{S}_i$ and completion times $\tilde{C}_i$ for each lot *i*, with the initial conditions $\tilde{V}_0 = -\infty, \tilde{C}_0 = -\infty$. From the estimated start and completion times, we calculate $\widetilde{CT}_i$, $\widetilde{LRT}_i$, and $\widetilde{TT}_i$ for each lot *i*.

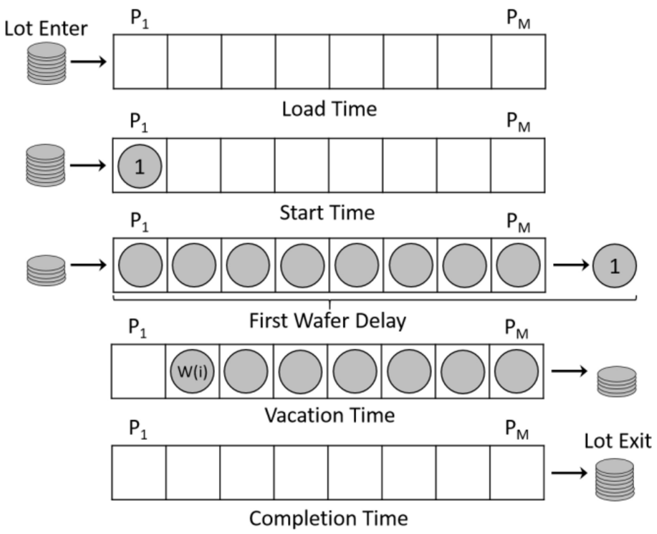

Fig. 3. Example of a flow line with a single server for each process.

Note that Table II is ordered in a logical sequence; i.e., the vacation times of the previous lot must be calculated before the load times, which must then be calculated before the start time, and so forth. We will provide intuition about each of the variables and parameters in the following subsections.

1) Vacation Times: To model the parallelism of the CPT, we use the concept of vacation time from the entrance of the tool. Let $\tilde{V}_{i-1}$ be the vacation time of lot *i-1*, which is the time instant that lot *i-1* vacates the modules which must be empty before lot *i* can begin production. To calculate $\tilde{V}_{i-1}$, we only need to consider the first process. As the first process can only serve wafers of the same class at a time, the vacation time can differ depending on whether lot *i-1* and lot *i* have the same class. Let $\widetilde{V_m}(i-1)$ be the estimated time at which *any* module of the first process becomes completely vacant of wafers from lot *i-1*. (There are no other wafers of lot *i-1* available to enter the first process.) Let $\widetilde{V_p}(i-1)$ be the estimated time at which *all* modules of the first process become completely vacant of wafers from lot *i-1*. (This is the instant at which the last wafer of lot *i-1* exits the first process.) Once lot *i* arrives at the tool and its class is known, the estimated vacation time of the previous lot $\tilde{V}_{i-1}$ is chosen as either $\widetilde{V_m}(i-1)$ or $\widetilde{V_p}(i-1)$. Note that, even though $\tilde{V}_{i-1}$ can only be determined once lot *i* has arrived (since we need to know its class), $\widetilde{V_m}(i-1)$ and $\widetilde{V_p}(i-1)$ do not need to know lot *i*'s class. The vacation times are estimated based on the lot completion time minus a constant ($D_m^{k_i}$ for $\widetilde{V_m}(i)$ and $D_p^{k_i}$ for $\widetilde{V_p}(i)$). The parameter $D_m^{k_i}$ (or $D_p^{k_i}$) can be interpreted as the average time difference between the lot completion time and the time instant when one module of (or the entire) first process is emptied.

2) Load and Start Times: $\tilde{L}_i$, the estimated load time of lot *i*, is the maximum of either the arrival time or the vacation time of the previous lot. The class of lot *i* is denoted as k_i. The start time $\tilde{S}_i$ is interpreted as the time at which the first wafer starts production in the first process. Any delay associated with a setup prior to entry experienced by the first wafer of a lot is contained in $E^{k_i,k_{i-1}}$.

3) Completion Times: The completion time is defined as the time instant that the last wafer of a lot exits the last process. Our completion time equation has a similar structure to the exit recursion equation (1). In the NBC term

$$\tilde{S}_i + FWD^{k_i} + A_1^{k_i}(W(i) - 1)$$

the completion time of lot i is the sum of the start time of the first wafer ($\tilde{S}_i$), the first wafer delay (FWD^{k_i}), and the time required for all of the remaining $W(i) - 1$ wafers to exit the tool (at a rate of one every $A_1^{k_i}$ units of time). This is a batch version of the NBC term in (1). Reticle setup time experienced by the first wafer of a lot is contained in FWD^{k_i}.

In the BC term

$$\tilde{C}_{i-1} + B(k_i, k_{i-1}) + A_2^{k_i}(W(i) - 1)$$

the completion time of lot i is the sum of the completion time of lot *i-1*, the time required for the first wafer of lot i to exit the tool after $\tilde{C}_{i-1}$, and the time required for all of the remaining $W(i) - 1$ wafers to exit the tool (at a rate of one every $A_2^{k_i}$ units of time). This is a batch version of the BC term in (1). Intuitively, the term $B(k_i, k_{i-1})$ includes the bottleneck processing time for that wafer, any reticle setup time at the bottleneck, and correction terms for differences in process times after the bottleneck; see [23, Lemma 2].

C. Description and Parameterization

We develop three types of ERMs which can be used according to the level of detail of available log data: tool log, wafer log, and lot log. Table III provides the model parameter extraction equations for each type of ERM. There are columns for each of the three classes of ERMs. For each ERM class, the rows provide the details of how the model parameters are calculated. Recall that lots from the true data are classified into NBC or BC, which is used to calculate the model parameters. The three classes of ERMS have different definitions of NBC and BC and may also calculate parameters differently. The intuition behind the model parameters were detailed previously and will be omitted in this subsection.

The rows NBC, BC, and "Lot indices" describe how to identify sets of lot indices with certain properties. The "NBC" row defines a set of lot indices ϕ_1 whose lots are guaranteed to experience no bottleneck contention with the previous lot. The "BC" row defines a set ϕ_2 containing the indices of lots that experience bottleneck contention (they are delayed behind their predecessor). The "Lot Indices" row identifies lots that have some property relative to their successor lot.

The model parameter calculations are provided in the remaining rows. For the BC cases, let

$$B(k, k') = \begin{cases} B^{k,k'}, & \text{for tool log} \\ B^{k}, & \text{for wafer log and lot log} \end{cases}$$

when using the simulation equations in Table II.

1) Tool Log: The tool log ERM can be used when data on detailed wafer advancement ($X_{w,m}$) from process to process within the tool is available. With the detailed wafer advancement data, it is possible to classify every lot in the tool log as NBC or BC. For tool log, NBC occurs for lot i if and only if its first wafer's entry time into the bottleneck is after the earliest possible entry time. For our CPT model, the earliest possible entry time is the time the last wafer of lot *i-1* enters the *B+1*th process plus the minimum time needed for the scanner's robot to pick a wafer from the stacker, move to the scanner, and place the wafer into the scanner. This minimum robot activity time we denote as the bottleneck contention workload (BCW)

$$BCW = \delta + 2\varepsilon$$

where δ and ε are the robot's move time and pick/place time, respectively. For our CPT, $\delta = 3$ seconds and $\varepsilon = 1$ second.

Note that $BCW = \delta + 2\varepsilon$ for our particular CPT configuration. BCW may differ for other CPTs. Furthermore, in practice one should consider instead $(1 + \alpha) \times BCW$ to allow for random event durations, where $\alpha \geq 0$. As α increases, the number of lot indices included in ϕ_2 will increase, possibly leading to lower fidelity.

The index i of a lot of class k is included in $L_{=}(k)$ if the next lot *i+1* has the same class k. Similarly, we define $L_{\neq}(k)$ as the set of lot indices of class k lots where the next lot is not class k. These are used to calculate the vacation time related parameters D_m^k and D_p^k, defined previously.

Rather than resorting to linear least squares estimation (LSE), A_1^k and FWD^k are calculated separately as empirical averages over all NBC cases. This choice is intentional. LSE would preserve the average lot residency time over the population and minimize the sum of the squared errors. However, the empirical average approach preserves the average first wafer delay and average per wafer throughput time. Preserving the averages is practically important as one wants the tool model to provide high fidelity throughput estimates as a first priority for lot level metrics as well as for wafer level metrics.

We similarly treat A_2^k and $B^{k,k'}$ for the BC cases. D_m^k, D_p^k, and $E^{k,k'}$ are also calculated as averages (and are in fact LSE since we are estimating a single constant parameter).

2) Wafer Log: The wafer log ERM can be constructed with data on only wafer start ($B_{\Omega(i,w)}$) and exit times from the tool ($F_{\Omega(i,w)}$). Without the complete tool log data, it is not possible to exactly characterize BC for every lot. Rather, we construct the sets ϕ_1 and ϕ_2 using conditions that guarantee NBC and BC, respectively. We ignore lots that cannot be definitely categorized into NBC or BC, meaning we use a partial sample of the wafer log data. We can guarantee NBC if $S_i > C_{i-1}$ for lot i. We can guarantee BC for lot i if $a_i \leq B_{\Omega(i-1,W(i-1))}$ and $k_i = k_{i-1}$. We require $k_i = k_{i-1}$, otherwise, different process flows, process times, and setups occlude our ability to determine BC with limited data. Because we require $k_i = k_{i-1}$ when defining ϕ_2, the first wafer delay parameter $B^{k,k'}$ can be simplified to B^k.

The set $\phi_{2+} = \{i | i + 1 \in \phi_2\}$ contains the indices of lots whose successor is guaranteed to experience BC.

TABLE III
PARAMETER EXTRACTION EQUATIONS FOR EXIT RECURSION MODELS

<table>
<tr><th></th><th>Tool Log</th><th>Wafer Log</th><th>Lot Log</th></tr>
<tr><td>NBC</td><td>$\phi_1 = \left\{i \middle| X_{\Omega(i,1),B} > X_{\Omega(i-1,W(i-1)),B+1} + BCW\right\}$</td><td colspan="2">$\phi_1 = \{i|S_i > C_{i-1}\}$</td></tr>
<tr><td>BC</td><td>$\phi_2 = \left\{i \middle| X_{\Omega(i,1),B} \leq X_{\Omega(i-1,W(i-1)),B+1} + BCW\right\}$</td><td>$\phi_2 = \left\{i \middle| a_i \leq B_{\Omega(i-1,W(i-1))}, k_i = k_{i-1}\right\}$</td><td>$\phi_2 = \{i|a_i \leq S_{i-1}, k_i = k_{i-1}\}$</td></tr>
<tr><td>Lot Indices</td><td>$L_=(k) = \{i|k_i = k_{i+1} = k\}$
$L_\neq(k) = \{i|k_i = k, k_i \neq k_{i+1}\}$</td><td colspan="2">$\phi_{2+} = \{i|i+1 \in \phi_2\}$</td></tr>
<tr><td rowspan="4">Parameters</td><td colspan="2">$A_1^k = \frac{\sum_{i\in L(k)\cap\phi_1}\left(C_i - F_{\Omega(i,1)}\right)}{\sum_{i\in L(k)\cap\phi_1}(W(i)-1)}$
$FWD^k = \frac{\sum_{i\in L(k)\cap\phi_1}\left(F_{\Omega(i,1)} - S_i\right)}{|L(k)\cap\phi_1|}$</td><td>$A_1^k = \hat{\beta}_1$
$FWD^k = \widehat{TT1^k}(1) = \hat{\beta}_0 + \hat{\beta}_1$
(See Subsection III.C.3. Lot Log)</td></tr>
<tr><td>$A_2^k = \frac{\sum_{i\in L(k)\cap\phi_2}\left(C_i - F_{\Omega(i,1)}\right)}{\sum_{i\in L(k)\cap\phi_2}(W(i)-1)}$
$B^{k,k\prime} = \frac{\sum_{i\in L(k,k\prime)\cap\phi_2}\left(F_{\Omega(i,1)} - C_{i-1}\right)}{|L(k,k')\cap\phi_2|}$</td><td>$A_2^k = \frac{\sum_{i\in L(k)\cap\phi_2}\left(C_i - F_{\Omega(i,1)}\right)}{\sum_{i\in L(k)\cap\phi_2}(W(i)-1)}$
$B^k = \frac{\sum_{i\in L(k)\cap\phi_2}\left(F_{\Omega(i,1)} - C_{i-1}\right)}{|L(k)\cap\phi_2|}$</td><td>$A_2^k = \hat{\beta}_3$
$B^k = \widehat{TT2^k}(1) = \hat{\beta}_2 + \hat{\beta}_3$
(See Subsection III.C.3. Lot Log)</td></tr>
<tr><td>$D_m^k = \frac{\sum_{i\in L_=(k)}\left(C_i - X_{\Omega(i,W(i))-R(k,1)+1,2}\right)}{|L_=(k)|}$
$D_p^k = \frac{\sum_{i\in L_\neq(k)}\left(C_i - X_{\Omega(i,W(i)),2}\right)}{|L_\neq(k)|}$</td><td colspan="2">$D_m^k = \frac{\sum_{i\in L(k)\cap\phi_{2+}}(C_i - L_{i+1})}{|L(k)\cap\phi_{2+}|}$
$D_p^k = D_m^k - (R(k,1)-1)\times A_2^k$</td></tr>
<tr><td colspan="3">$E^{k,k\prime} = \frac{\sum_{i\in L(k,k\prime)}(S_i - L_i)}{|L(k,k')|}$</td></tr>
</table>

The parameters A_1^k, A_2^k, FWD^k, B^k and $E^{k,k'}$ are calculated similarly to the tool log case.

For the vacation time related parameters D_m^k and D_p^k, we do not have access to the entry times to the second process. We use BC lots and consider the load time of the next lot. If the next lot experiences BC, then the difference between the lot completion time of the current lot and the load time of the next lot is used as D_m^k. Note that the BC cases experience no class change and thus D_m^k is used. D_p^k is calculated using the number of redundant modules of the first process and the wafer throughput time.

3) Lot Log: The lot log ERM requires the least amount of data: lot start (entrance of first wafer, S_i) and completion (exit of last wafer, C_i) times. It is not possible to exactly characterize when bottleneck contention occurs. We construct the sets ϕ_1 and ϕ_2 similarly to the wafer log case. Note that the condition for ϕ_2 is stricter.

Without wafer-level data we resort to linear regression to estimate the first wafer delay and wafer throughput time parameters. We treat the NBC and BC cases separately. For the NBC cases for a given lot class k, we construct an average throughput time for each value of n (the wafers per lot) as

$$TT1^k(n) = \frac{\sum_{i\in LS^k(n)\cap\phi_1}(C_i - S_i)}{\left|LS^k(n)\cap\phi_1\right|}$$

where $LS^k(n) = \{i|k_i = k, W(i) = n\}$ is the set of lot indices with class k and lot size of n. For the regression to successfully estimate the two parameters for each of the NBC and BC cases, the true data must contain several lot sizes for each lot class for both cases.

We thus obtain one data point for each n and consider the relationship between n and throughput time as

$$TT1^k(n) = \beta_0 + \beta_1 n$$

As the number of lots may not be equal for each n, we use weighted linear least squares estimation (WLS), with $|LS^k(n)\cap\phi_1|$ as the weights for each n value, to obtain the estimates $\widehat{\beta_0}$ and $\widehat{\beta_1}$. Note that this approach is well known as equivalent to conducting ordinary LSE on the entire data set but is much faster; see [38]. Set the per wafer throughput time as $A_1^k = \widehat{\beta_1}$. Set the first wafer delay, which is the predicted throughput time with $n = 1$, as $FWD^k = \widehat{TT1^k}(1) = \widehat{\beta_0} + \widehat{\beta_1}$. Since we resort to WLS, the wafer level averages need not be preserved.

The BC case is treated similarly. The parameters D_m^k, D_p^k and $E^{k,k'}$ are calculated as in the wafer log model.

D. Summary

Using intuition from flow lines, we develop ERMs that consider the concept of bottleneck contention. Depending on the level of detail of the available true data, three types of ERMs are proposed, with different parameter extraction equations given in Table III. Once the parameters are calculated, simulations using the models may be conducted based on the equations in Table II.

IV. SIMULATION DESCRIPTION

A. Simulation Overview

We consider the detailed CPT model (DS) as our true system and use an affine model (AF) and empirical flow line

TABLE IV
SIMULATION CASES FOR EACH TYPE OF SIMULATION

Parameter	Type I				Type II	Type III				
	Simulation Values			No. Cases	Simulation Values	Training Values	Simulation Values			No. Cases
	Min	Max	Increment				Min	Max	Increment	
Train Size	1.5	6	0.5	10	Same as Type I	1.5 3 6	1.5	6	0.5	30

Lot Size	{1, 2, 3}	{23, 24, 25}	2	12		{3, 4, 5}
						{13, 14, 15}
						{23, 24, 25}
							{1, 2, 3}	{23, 24, 25}	2	36

Loading	0.90	0.99	0.01	12*		0.85 0.95 0.99	0.90	0.99	0.01	36*
Setup Duration	40%	180%	20%	8						0
STK Size	2	20	1	19						0
Penultimate Bottleneck Process Time	40%	180%	20%	8						0
Pre-scan Track Process Time	40%	180%	20%	8						0
Process Time Profiles	–			3						0
Train Size and Lot Size	–			0		Train = 6 and Lot size = {23, 24, 25}	T = 3 and L = {1, 2, 3}	T = 3 and L = {23, 24, 25}	Lot size: 2	12
Totals	80				80	114				

* Loading levels of 0.80 and 0.85 were simulated in addition to 0.90 ~ 0.99.

model (EFL) as benchmarks. We assess the performance of the ERMs with respect to mean $\widetilde{CT}$, $\widetilde{LRT}$, $\widetilde{TT}$ and computation times. We use JAVA on an i5-3570 CPU computer with 16 GB of RAM.

For each case, we simulate for 15,000 lots (about one year) and conduct 30 independent replications. The tool is initially empty. There are no tool failures. Lots arrive according to a Poisson process. (This is not required by the ERMs.) The average number of lots of the same class to arrive in a row (train size) is set and random lot classes are assigned by a simple Markov chain model. The reticle alignment setup for every lot and pre-scan track setup for class changes are uniformly distributed in [210, 260] and [240, 420], respectively. The loading levels are a fraction of the JIT throughput of the DS model. Our baseline is set to average train size of 3, lot sizes of 23, 24 and 25 (with probabilities 0.25, 0.5, 0.25, respectively), and loading of 0.95.

For each replication, DS is run first to generate data that is used to parameterize the AF, EFL, and ERMs. For the ERM lot log case, we use ordinary LSE instead of WLSE with the $TT1^k(n)$ values as it is simpler. The results are nearly identical. Model simulations are then conducted. Computation times do not include the time used for parameter extraction.

B. Simulation Types

We follow the approach proposed in [35] and conduct 274 different simulation studies (consisting of 30 replications each) with all models. See Table IV. There, % values indicate % of the nominal value. There are three types of simulations.

Type I simulations compare the model predictions with the DS values using the exact same input data. For example, the lot arrival times to the DS system are exactly the same as those to the ERMs for each replication. We vary 8 parameters for a total of 80 Type I studies.

Type I simulations assess model fidelity under ideal conditions. Computation times are determined from Type I simulations under baseline settings.

Type II simulations compare the model predictions with the DS values using different random input values that were generated from the same distributions. For example, the arrival times to the DS system are generated from a different realization of the Poisson process (with same arrival rate) as those to the ERMs for each replication. We conduct the same 80 cases as in Type I. Type II simulations assess model performance with fixed operating conditions for a Monte Carlo simulation approach.

In Type III simulations, the internal settings (e.g., STK capacity, module process time) of the DS are held constant. One set of arrival process parameters (consisting of lot size, mean interarrival time and train size settings) generates the data used to parameterize the AF, EFL and ERMs. Then, one of the arrival process parameter settings is changed (e.g., mean interarrival time). All models, including DS, are simulated using the new settings. Type III simulations consider changes in operating conditions and address robustness via Monte Carlo simulation.

V. RESULTS AND DISCUSSION

We label the tool log ERM as ERM1, wafer log ERM as ERM2, and lot log ERM as ERM3.

A. Comparison of Computation Times

To assess the computational requirement for each model, we measure the CPU computation time required for all Type I simulations that are conducted at the baseline settings. There is one baseline setting case in each of the 8 Type I studies and each consists of 30 replications. The average CPU computation time required over these 240 replications and averaged over

TABLE V
MODEL COMPUTATION TIMES

Model	Computation Time (ms)	Scaled Computation Time
DS	17,372,368	97,941.47
AF	177	1.00
ERMs	334	1.88
EFL	83,596	471.30

Computation times have been scaled so that AF has a value of one in the third column.

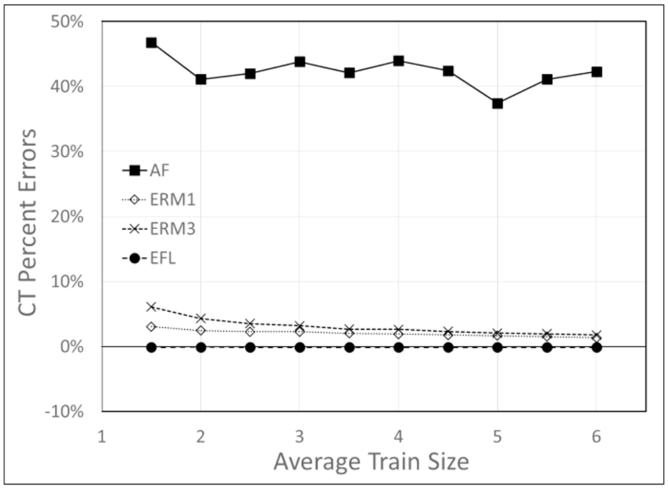

Fig. 4. Percent errors in cycle time for different train sizes.

all ERM models is provided in Table V. ERMs require about 1.9 times as much computation as AF. Compared to EFL, ERMs are approximately 250 times more computationally friendly.

B. Type I Simulations

We provide some details on train size, lot size and STK capacity studies. The other cases are largely similar to these so we only provide an overview.

For all Type I simulations, the mean LRT prediction performance is largely similar. AF exhibits 50% error or more for mean LRT because it does not well address parallelism. All ERM types exhibit less than 5% errors. ERM1 has about 4-5% mean LRT errors, while ERM2 and ERM3 are around 2-4%. EFL exhibits 4-5% mean LRT errors.

Throughout, we drop the word "mean" and say LRT, CT and TT errors when referring to the error in the mean values for those metrics over all data in the case.

1) Train Size Cases: The average train size was varied from 1.5 to 6 in increments of 0.5. The percent errors in average $\widetilde{CT}$, $\widetilde{LRT}$, and $\widetilde{TT}$ relative to the true data averages are provided in Figs. 4 to 6, respectively. ERM2 has been omitted for visibility; it has slightly lower errors than ERM3 for all metrics.

AF is inferior to the ERMs for CT. AF has errors up to 47% for CT. ERM1 and ERM3 have 3.5% and 6.1% errors for CT, respectively. EFL is superior with CT errors less than 0.11%.

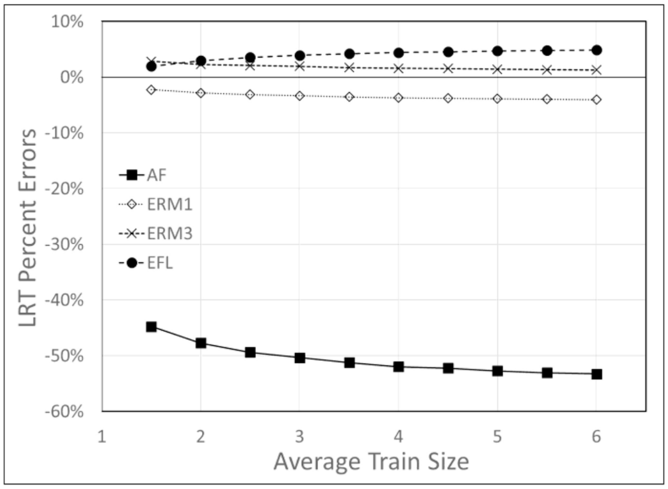

Fig. 5. Percent errors in lot residency time for different train sizes.

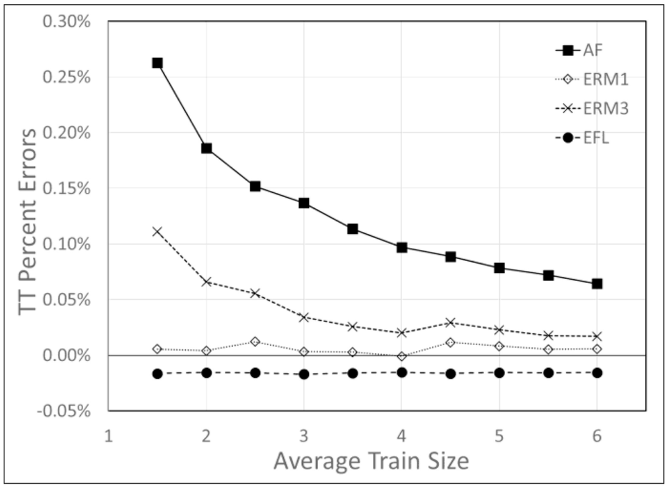

Fig. 6. Percent errors in throughput time for different train sizes.

All models exhibit less than 0.3% error for TT. For this study and metric, ERM1 is more accurate than EFL.

2) Lot Size Cases: The lot size set is varied from {1, 2, 3} to {23, 24, 25} in steps of 2 with the probabilities unchanged.

At lot sizes of {1, 2, 3}, ERM has extremely high errors as high as 5.5×10^7% for CT, 9000% for LRT, and 150% for TT. However, at lot sizes of {3, 4, 5} and above, ERMs are relatively accurate. The perspective taken when developing the ERMs did not consider a lot size of 1. While this is a limitation of the model, ERMs should do well for lot sizes of 3 and above.

Figs. 7 and 8 provide the percent errors in CT and TT, respectively, not including the lot size set {1, 2, 3}. ERMs predict CT well, with less than 9% errors. AF has errors up to 225%. EFL exhibits less than 7% error for CT. All models have less than 0.5% error for TT. ERMs exhibit slight increases in error as the lot sizes become smaller.

3) STK Capacity Cases: We varied STK capacity from 2 to 20. Figs. 9 and 10 provide the percent errors in CT and TT, respectively.

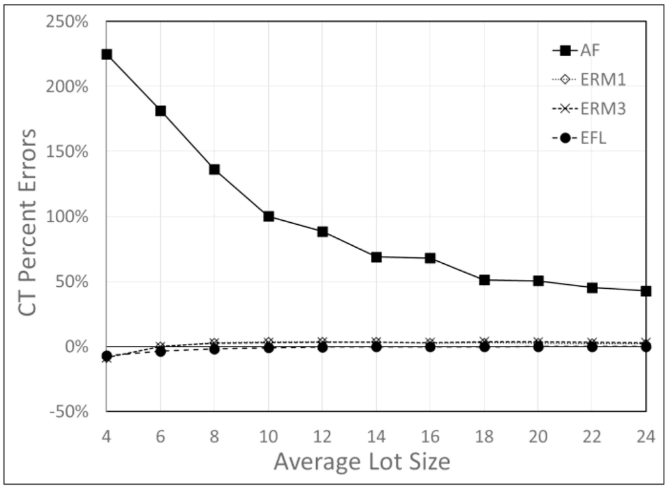

Fig. 7. Percent errors of cycle time for different lot sizes.

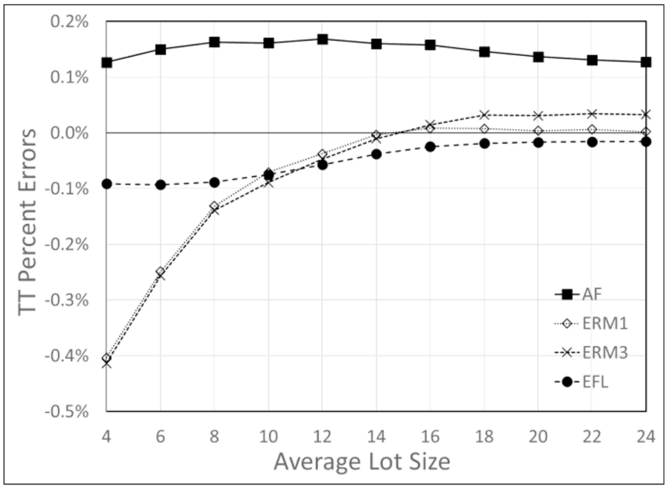

Fig. 8. Percent errors of throughput time for different lot sizes.

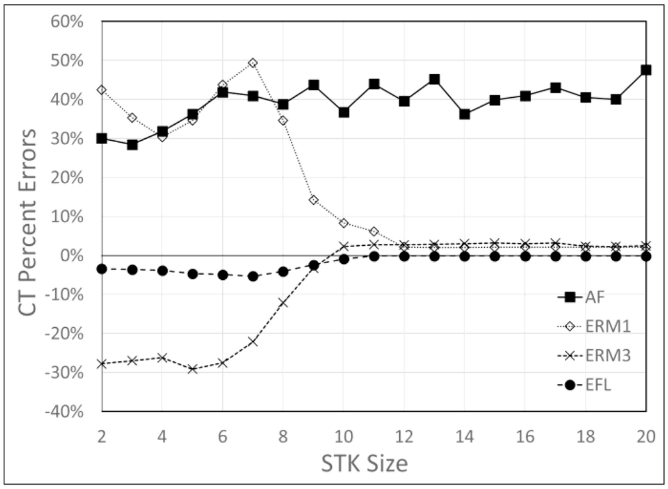

Fig. 9. Percent errors of cycle time for different STK sizes.

For CT, AF has up to 50% error. ERMs have less than 10% CT error for STK capacity of 10 wafers or more. For STK capacity less than 10 wafers, ERMs are comparable to AF with up to 50% errors. EFL errors were less than about 6% for CT.

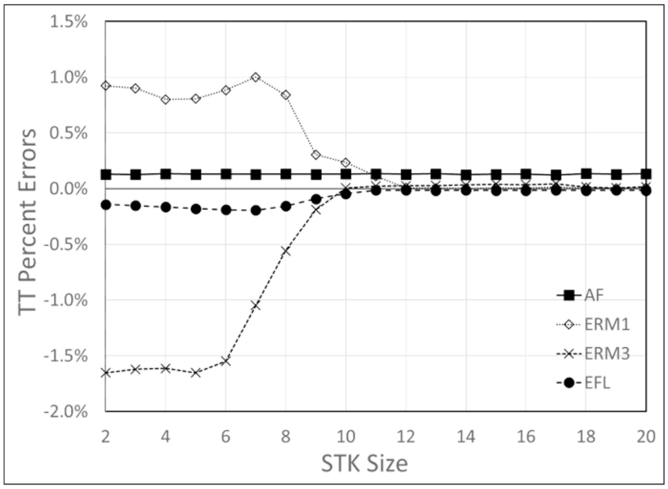

Fig. 10. Percent errors of throughput time for different STK sizes.

AF has less than 0.15% TT error. ERMs have less than about 0.15% errors for STK capacity of 10 wafers or more. For STK capacity less than 10 wafers, the ERMs exhibit up to 1.7% error. EFL has less than 0.15% error for TT.

4) Type I Simulations for Other Parameters: Similar results are obtained for the other parameters. In most cases, ERMs are much more accurate than AF, but less accurate than EFL for CT and LRT. ERMs are more accurate than AF and sometimes even more accurate than EFL for TT.

5) Secondary Metrics: With baseline settings, we next consider the standard deviation, mean absolute error, and the error's standard deviation for the model predictions; see Table VI. Every entry there is given in seconds.

For standard deviation, we consider the model as good if it has similar value as DS for CT, LRT or TT. AF is quite far from DS in for standard deviation of all metrics. ERMs and EFL are both similar to DS.

For mean absolute error and error standard deviation, smaller is better. AF is by far the worst. ERMs are quite good relative to it. EFL is outstanding.

This behavior continues for all Type I cases.

C. Type II Simulations

Type II simulations are Monte Carlo simulations that assess the model performance at current operating conditions. Different realizations of the random variables with unchanged distributions are used. The results are similar to those of the Type I with slightly higher errors and variability. We omit them for brevity.

D. Type III Simulations

Type III simulations help assess model robustness to changes in operating conditions. We next provide some detail on the simulation procedure since Type III cases are not as simple as Type I or II.

- For train size cases, the models were separately parameterized with average train sizes of 1.5, 3, and 6 and

TABLE VI
SECONDARY METRICS FOR TYPE I – BASELINE LOT CONDITIONS

Model	CT				LRT				TT			
	Average	σ	|Error|	Error σ	Average	σ	|Error|	Error σ	Average	σ	|Error|	Error σ
DS	30,535.1	24,903.4			5,904.7	980.2			2,927.4	251.2		
AF	43,922.9	36,254.3	13,777.8	17,442.8	2,931.4	78.7	2,973.2	983.5	2,931.4	78.7	104.2	238.9
ERM1	31,234.4	24,851.5	752.0	615.9	5,708.8	827.6	293.6	325.7	2,927.5	250.0	32.3	93.9
ERM2	31,516.5	25,087.7	983.5	677.3	5,997.5	873.5	252.3	309.7	2,928.4	248.4	32.3	94.2
ERM3	31,520.8	25,090.3	988.1	681.8	6,019.6	877.5	255.6	309.4	2,928.4	248.3	32.3	94.2
EFL	30,502.8	24,902.3	32.3	20.6	6,136.6	1,184.3	246.6	221.3	2,926.9	245.7	18.5	24.5

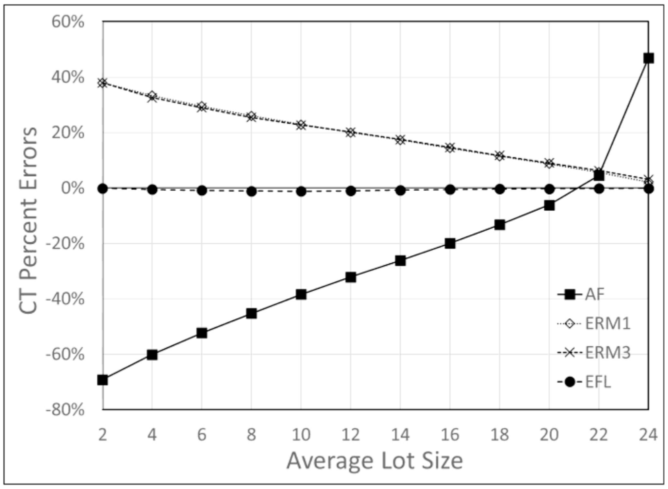

Fig. 11. Percent errors of cycle time for different lot sizes. Models are parameterized with lot sizes {23, 24, 25}.

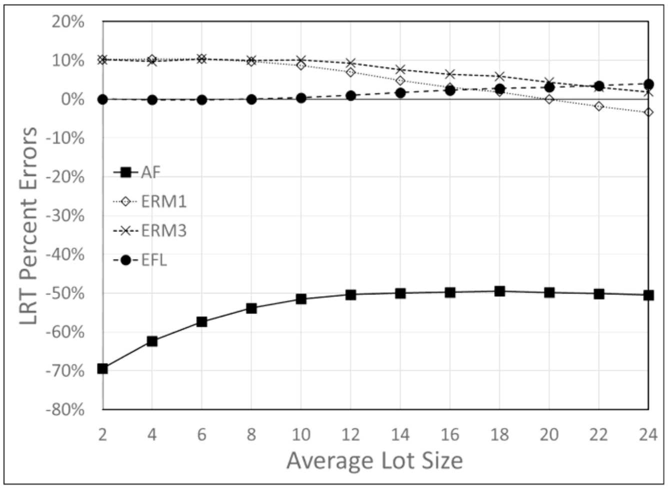

Fig. 12. Percent errors of lot residency time for different lot sizes. Models are parameterized with lot sizes {23, 24, 25}.

simulated with varying train sizes from 1.5 to 6 in increments of 0.5.

- For lot size cases, the models were separately parameterized with lot sizes {3, 4, 5}, {13, 14, 15}, {23, 24, 25} and simulated with varying lot sizes from {1, 2, 3} to {23, 24, 25} in increments of 2.
- For the loading cases, the models were separately parameterized with loading levels of 0.85, 0.95, and 0.99 and simulated at 12 different loading levels.

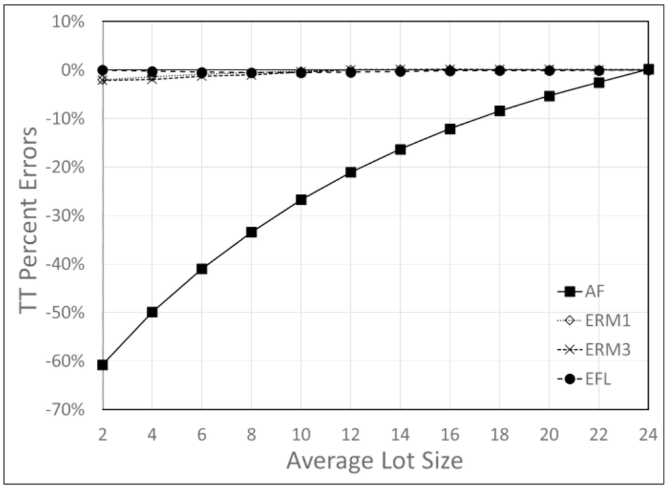

Fig. 13. Percent errors of throughput time for different lot sizes. Models are parameterized with lot sizes {23, 24, 25}.

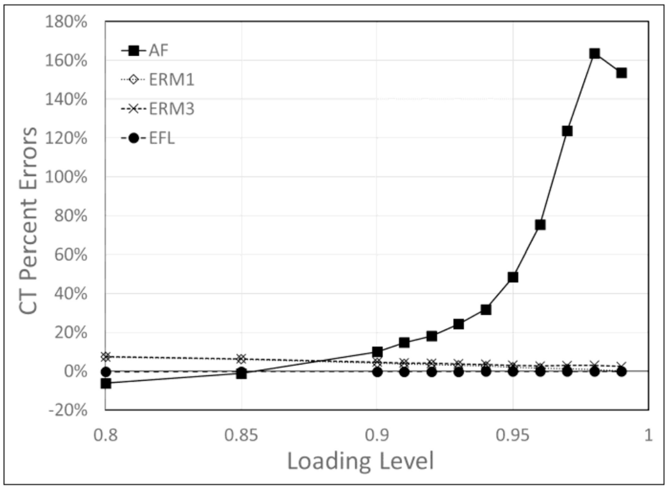

Fig. 14. Percent errors of cycle time for different loading levels. Models are parameterized with 0.95 loading.

- For the mixed cases, the models were parameterized with train size of 6 and lot sizes of {23, 24, 25} and then simulated with a train size of 3 and varying lot sizes from {1, 2, 3} to {23, 24, 25} in increments of 2.

While there are many cases, we show representative results.

Consider the cases when we parameterize at baseline settings and vary the lot sizes when simulating. Figs. 11 to 13 provide the percent errors in CT, LRT, and TT. ERM2 is similar to ERM3 and omitted. ERMs have errors up to 38% for

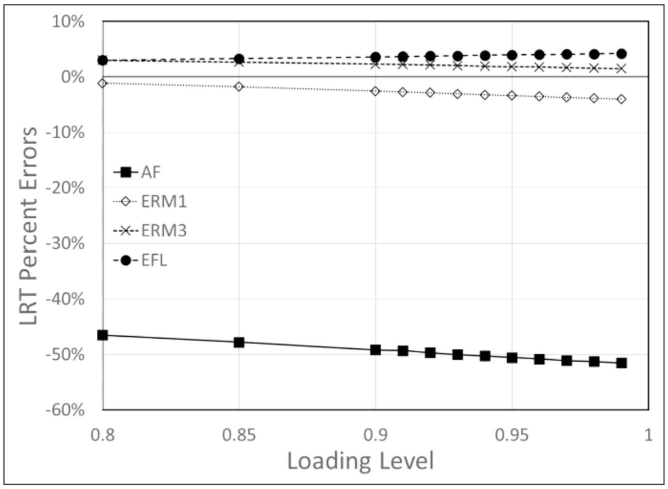

Fig. 15. Percent errors of lot residency time for different loading levels. Models are parameterized with 0.95 loading.

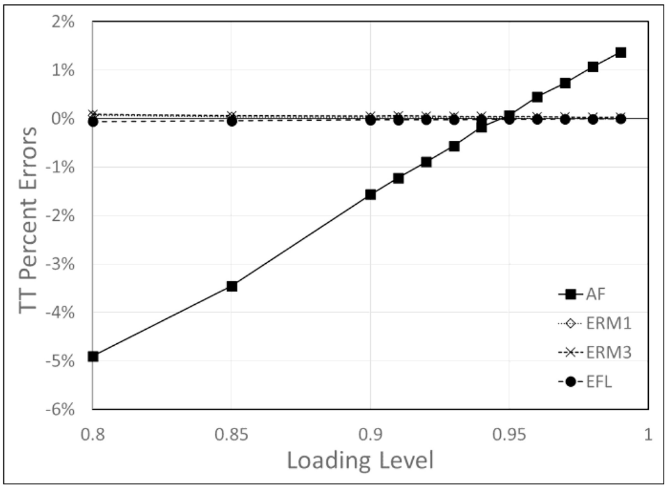

Fig. 16. Percent errors of throughput time for different loading levels. Models are parameterized with 0.95 loading.

CT, 10.3% for LRT, and 2.1% for TT. ERMs seem to have an intermediate level of robustness, between that of AF and EFL. They are inaccurate for CT, accurate for LRT, and very accurate for TT across all lot size cases considered.

Consider the cases when we parameterize at baseline settings and vary the loading when simulating. Figs. 14 to 16 provide the percent errors in CT, LRT, and TT. ERMs appear robust for all three metrics, with less than 7.5% errors for CT, 4% for LRT, and 0.09% for TT. They appear to be as robust as EFL for LRT and TT and superior to AF.

The other Type III simulations show similar results in that deviations from the training data can lead to increasing errors. Generally, AF are less robust than ERMs and ERMs are less robust than EFL. For deviations from the parameterization data in train size, all models retain their typical error performance.

E. Implications for Fab Level Simulation

Based on the results obtained from our numerical studies, it is possible to anticipate qualitatively the effect that ERMs (or detailed models or flow line models) will have when replacing affine models for CPTs in fab level simulation.

First, for a wafer fab in which a CPT toolset is the bottleneck (which is common due to the high cost of CPT tools), improved accuracy in CPT maximum throughput estimates (via JIT throughput time studies with affine models replaced by ERMs) will allow improved prediction of the fab's maximum throughput.

Second, improved cycle time predictions at a CPT toolset should improve fab overall cycle time estimations. The magnitude of the improvement in overall cycle time estimation might not match the improvement in CPT sector cycle time estimation because the CPT tool group is one of numerous others in a fab.

Third, improved lot residency time predictions (especially on a lot by lot basis beyond just the average) will enable improved scheduling decisions in simulation based schedulers and mathematical programming based schedulers. It is, however, not clear the extent to which a resulting production schedule would change and how much influence this would have on system-wide performance metrics.

Fourth, the computational requirements of using ERMs as opposed to affine models for CPTs can be roughly estimated. For example, if the affine models for CPTs account for about 20% of the fab simulation computation requirement, changing to ERMs (which require about 1.9 times the computation), would increase the overall fab simulation computations by about 18%.

We require additional numerical studies to more precisely assess the effect of using ERMs relative to affine models in a full fab simulation. Such a study depends on the development of one fab model with embedded detailed simulation models for CPTs, a second fab model with affine models and a third with ERMs. We plan to conduct such a study in the future.

VI. Conclusion

We proposed a new class of models of clustered photolithography tools for fab-level simulation. These exit recursion models (ERMs) were developed using intuition from flow lines, consider the concept of bottleneck contention and require seven parameters. They are intended to improve upon the popular affine models which require two parameters. We developed three classes of these empirical models based on the level of detail available in the parameterization data. We provided parameterization and simulation equations for ERMs.

To assess the performance of ERMs, we conducted numerous simulations across a range of parameters. Detailed CPT simulation, AF and EFL were used as benchmarks. While AF is not accurate for mean CT and mean LRT, the newly proposed ERM is shown to be quite accurate for all metrics, frequently with errors less than 6% for mean CT, 5% for mean LRT, and 0.1% for mean TT. For TT, ERMs can perform close to or even surpass the EFL. We explored the robustness of ERMs when simulation conditions deviated from the parametrization conditions. In general, ERMs were fairly robust for all three metrics, although mean CT can be inaccurate in some cases.

ERMs are about 250 times less computationally complex than EFL and only require 1.9 times as much computation as AF. While the models seem to strike a good balance between fidelity and computational complexity for fab-level simulation, ERMs have some limitations, such as high errors for lot sizes of 1 and 2 wafers and small STK sizes, due to the model structure. Mostly, ERMs perform well relative to AF and EFL.

There are several directions for future work. Can the ERM model structure be improved so that it can address the current limitations for lot sizes of 1 or 2 wafers and small STK capacity? Would an eight parameter model that includes information on internal workload at the STK improve performance or robustness? Is it possible to develop a parametric version of ERMs? Can ERMs be used instead of AF models for scheduling optimization and how much does that improve the results? It is also very important to study the quantitative effect of using such models in full fab simulations.

References

[1] W. J. Hopp and M. L. Spearman, *Factory Physics*. 3rd ed. Long Grove, IL, USA: Waveland Press, 2011.

[2] J. A. Jimenez, G. T. Mackulak, and J. W. Fowler, "Levels of capacity and material handling system modeling for factory integration decision making in semiconductor wafer fabs," *IEEE Trans. Semicond. Manuf.*, vol. 21, no. 4, pp. 600–613, Nov. 2008.

[3] D. Fandel and R. Wright, "300 mm productivity detractors mitigation cost analysis," in *Proc. IEEE/SEMI Adv. Semicond. Manuf. Conf.*, Cambridge, MA, USA, 2008, pp. 222–227.

[4] J.-E. Kiba, G. Lamiable, S. Dauzère-Pérès, and C. Yugma, "Simulation of a full 300mm semiconductor manufacturing plant with material handling constraints," in *Proc. Winter Simulat. Conf.*, Austin, TX, USA, 2009, pp. 1601–1609.

[5] C.-N. Wang and C.-H. Wang. "A simulated model for cycle time reduction by acquiring optimal lot size in semiconductor manufacturing," *Int. J. Adv. Manuf.*, vol. 34, no. 9, pp. 1008–1015, Jan. 2007.

[6] E. Bass and R. Wright, "Modeling semiconductor factories for equipment and cycle time reduction opportunities," *Future Fab Int.*, vol. 24, pp. 50–55, 2008.

[7] H. Gao, F. Qiao, Y. Ma, and L. Kong, "A simulation based optimization approach for scheduling of a semiconductor manufacturing system," in *Proc. IEEE Int. Conf. Syst. Sci. Eng.*, Shanghai, China, 2014, pp. 251–254.

[8] L. Li and F. Qiao, "A modular simulation system for semiconductor manufacturing scheduling," *Przeglad Elektrotechniczny*, vol. 88, no. 1B, pp. 12–18, 2012.

[9] K. Schmidt, J. Weigang, and O. Rose, "Modeling semiconductor tools for small lotsize fab simulations," in *Proc. Winter Simulat. Conf.*, Monterey, CA, USA, 2006, pp. 1811–1816.

[10] R. Su, F. Qiao, Y. Ma, K. Lu, and L. Zhao, "Application of data-based parameter optimizing approaches in IC manufacturing system simulation model," in *Proc. Int. Conf. Inf. Technol. Softw. Eng.*, 2013, pp. 311–319.

[11] K. Wu, "Classification of queueing models for a workstation with interruptions: A review," *Int. J. Prod. Res.*, vol. 52, no. 3, pp. 902–917, 2014.

[12] K. Wu, "Taxonomy of batch queueing models in manufacturing systems," *Eur. J. Oper. Res.*, vol. 237, no. 1, pp. 129–135, Aug. 2014.

[13] K. Wu, N. Zhao, and C. K. M. Lee, "Queue time approximations for a cluster tool with job cascading," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 1200–1206, Apr. 2016.

[14] K. Wu, Y. Zhou, and N. Zhao, "Variability and the fundamental properties of production lines," *Comput. Ind. Eng.*, vol. 99, pp. 364–371, Sep. 2016. [Online]. Available: http://dx.doi.org/10.1016/j.cie.2016.04.014

[15] J. R. Morrison, "On the fidelity of the Ax+B equipment model for clustered photolithography scanners in fab-level simulation," in *Proc. Winter Simulat. Conf.*, Phoenix, AZ, USA, 2011, pp. 2029–2039.

[16] S. Radloff *et al.*, "First wafer delay and setup: How to measure, define and improve first wafer delays and setup times in semiconductor fabs," in *Proc. IEEE/SEMI Adv. Semicond. Manuf. Conf.*, Berlin, Germany, 2009, pp. 86–90.

[17] *AutoSched^{TM} AP 9.3.0 User's Guide*, Appl. Mater., Inc., Santa Clara, CA, USA, 2008.

[18] Y. Dallery and S. B. Gershwin, "Manufacturing flow line systems: A review of models and analytical results," *Queueing Syst.*, vol. 12, no. 1, pp. 3–94, 1992.

[19] T. Altiok, *Performance Evaluation of Manufacturing Systems*. New York, NY, USA: Springer-Verlag, 1996.

[20] K. Park and J. R. Morrison, "Controlled wafer release in clustered photolithography tools: Flexible flow line job release scheduling and an LMOLP heuristic," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 642–655, Apr. 2015.

[21] M.-C. Wu and C.-W. Chiou, "Scheduling semiconductor in-line steppers in new product/process introduction scenarios," *Int. J. Prod. Res.*, vol. 48, no. 6, pp. 1835–1852, Mar. 2010.

[22] C.-W. Chiou and M.-C. Wu, "Scheduling of multiple in-line steppers for semiconductor wafer fabs," *Int. J. Syst. Sci.*, vol. 45, no. 3, pp. 384–398, 2014.

[23] J. R. Morrison, "Multiclass flow line models of semiconductor manufacturing equipment for fab-level simulation," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 1, pp. 81–94, Jan. 2011.

[24] Y. J. Park and H. R. Hwang, "Minimization of total processing time in semiconductor photolithography process," *Appl. Mech. Mater.*, vol. 325, no. 1, pp. 88–93, Jul. 2013.

[25] W.-S. Kim and J. R. Morrison, "The throughput rate of serial production lines with deterministic process times and random setups: Markovian models and applications to semiconductor manufacturing," *Comput. Oper. Res.*, vol. 53, pp. 288–300, Jan. 2015.

[26] H. J. Yoon and D. Y. Lee, "Deadlock-free scheduling of photolithography equipment in semiconductor fabrication," *IEEE Trans. Semicond. Manuf.*, vol. 17, no. 1, pp. 42–54, 2004.

[27] H. N. Geismar, C. Sriskandarajah, and N. Ramanan, "Increasing throughput for robotic cells with parallel machines and multiple robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 1, no. 1, pp. 84–89, Jul. 2004.

[28] P. J. Byrne, "An analysis of semiconductor reticle management using discrete event simulation," in *Proc. Summer Comput. Simulat. Conf.*, San Diego, CA, USA, Jul. 2007, pp. 593–600.

[29] B.-H. Zhou, Q.-Z. Pan, S.-J. Wang, and B. Wu, "Modeling of photolithography process in semiconductor wafer fabrication systems using extended hybrid Petri nets." *J. Central South Univ. Technol.*, vol. 14, no. 3, pp. 393–398, 2007.

[30] F. Qiao, Y.-M. Ma, L. Li, and H.-X. Yu, "A Petri net and extended genetic algorithm combined scheduling method for wafer fabrication," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 1, pp. 197–204, Jan. 2013.

[31] F. Yang, N. Wu, Y. Qiao, and M. Zhou, "Petri net-based polynomially complex approach to optimal one-wafer cyclic scheduling of hybrid multi-cluster tools in semiconductor manufacturing," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 12, pp. 1598–1610, Dec. 2014.

[32] M. Zhou, "Modeling, analysis, simulation, scheduling, and control of semiconductor manufacturing systems: A Petri net approach," *IEEE Trans. Semicond. Manuf.*, vol. 11, no. 3, pp. 333–357, Aug. 1998.

[33] N. Wu and M. Zhou, "Real-time deadlock-free scheduling for semiconductor track systems based on colored timed Petri nets," *OR Spectr.*, vol. 29, no. 3, pp. 421–443, 2007.

[34] K. Park and J. R. Morrison, "Controlled wafer release in clustered photolithography tools: Flexible flow line job release scheduling and an LMOLP heuristic," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 642–655, Apr. 2015.

[35] J. Y. Park, K. Park, and J. R. Morrison, "Models of clustered photolithography tools for fab-level simulation: From affine to flow line," *IEEE Trans. Semicond. Manuf.*, to be published.

[36] B. Avi-Itzhak, "A sequence of service stations with arbitrary input and regular service times," *Manag. Sci.*, vol. 11, no. 5, pp. 565–571, 1965.

[37] H. D. Friedman, "Reduction methods for tandem queuing systems," *Oper. Res.*, vol. 13, no. 1, pp. 121–131, Feb. 1965.

[38] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*, 5th ed. Hoboken, NJ, USA: Wiley, 2012, pp. 188–191.

Jung Yeon Park received the B.S. and M.S. degrees in industrial and systems engineering from KAIST, South Korea, in 2014 and 2016, respectively. He is currently a Software Engineer with Samsung Electronics, DS Division, South Korea. His research interests are semiconductor equipment modeling and simulation and flow lines.

Kyungsu Park received the B.S. degrees in information and industrial engineering and chemical engineering from Yonsei University, South Korea, in 2008, and the Ph.D. degree from the Department of Industrial and Systems Engineering, KAIST, South Korea, in 2014.

He was a Senior Researcher with the Defense Agency of Technology and Quality and the Busan Institute of S&T Evaluation and Planning, South Korea. His research interest is on modeling and scheduling in semiconductor manufacturing.

James R. Morrison (S'97–M'00) received the B.S. degrees in electrical engineering and mathematics from the University of Maryland, College Park, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois at Urbana-Champaign. He was with the Fab Operations Engineering Department, IBM Corporation from 2000 to 2005. He is currently an Associate Professor with the Department of Industrial and Systems Engineering, KAIST, Daejeon, South Korea. His research interests include semiconductor wafer fabrication, systems of UAVs, and engineering of service systems. Since 2009, he has been the Co-Chair of the IEEE Robotics and Automation Society Technical Committee on Semiconductor Manufacturing Automation.