Proceedings of the 2009 Winter Simulation Conference
M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, eds.

EQUIPMENT MODELS FOR FAB LEVEL PROUDCTION SIMULATION: PRACTICAL FEATURES AND COMPUTATIONAL TRACTABILITY

James R. Morrison

Dept. of Industrial & Systems Engineering
373-1 Gwahangno, Yuseong-gu
KAIST
Daejeon 305-701, Republic of Korea

ABSTRACT

Equipment models used in fab-level simulation do not typically include features such as internal wafer buffers and setups that depend on wafer locations inside the tool. Such features are especially important to system performance in the presence of smaller lot sizes and greater product diversity. In this paper, we provide an introduction to flow line models that allow one to incorporate these key practical behaviors. We then develop flow line models for clustered photolithography tools and conduct simulations to assess the quality of the models. Despite the fact that the models only incorporate wafer transport robots via a constant addition to the process time, they can be quite accurate. When tested against data from a clustered photolithography tool in production, the model predictions for throughput and cycle time were within 1% and 4%, respectively. The computational requirements are about one order of magnitude less than is otherwise possible.

1 INTRODUCTION

The anticipated proliferation of small lot sizes and increased product diversity brought about by a transition to 450 mm wafer sizes will present challenges for existing simulation models of semiconductor wafer fabrication (Pillai 2006). Fab-level simulation models of production typically consist of a thousand tools, or more. Each tool model allows for numerous practical features such as first wafer delay, throughput rate, batch sizes, setups between dissimilar lots, tool failure and sampling, to mention a few. The simulation is used to answer questions about cycle time, throughput, production control policies, tool purchases, etc. that cannot be well addressed via less detailed queueing or spreadsheet models.

For many tools, current tool models accurately express the essentials of tool behavior. However, as lot sizes decrease and product diversity increases, behaviors not currently included in typical models will play a greater role in system performance. In particular, features such as internal wafer buffers and setups with start times that depend upon the location of wafers within a tool require closer examination. A key tool where both of these elements exist is the clustered photolithography tool. With the goal of modeling such tools, we provide an introduction to deterministic flow line models that allow for the inclusion of additional model elements with less computation than would be required of a model incorporating both wafer and transport robot movement. While deterministic flow line models are limited by the assumptions that make them tractable, nevertheless, one can extend them sufficiently to include diverse classes of lots and state dependent setups.

Flow line models have been studied for many years and there is a rich body of work addressing their analysis, control and design. The first key results on deterministic flow lines were developed in Avi-Itzhak (1965) and Friedman (1965). Results for general classes of flow lines may be found in texts such as Buzacott and Shanthikumar (1993) and Altiok (1996) or survey papers such as Dallery and Gershwin (1992) or Papadopoulos and Heavey (1996). In the classic work of Avi-Itzhak (1965), a recursion for the exit times of customers from a deterministic flow line with a single class of lots was developed. This recursion could be used to simply model such tools in simulation, however, it does not allow visibility to the interior of the tools without fully tracking the evolution of each wafer through each processing module. Thus, setups that depend on the wafer locations in the tool cannot be included. Also, though Avi-Itzhak (1995) later allowed for product dependent process times, the neat recursive structure of the initial results was lost.

To allow visibility into the nature of wafer movement internal to a deterministic flow line, Morrison (2008) and Morrison (2009a) developed an exact decomposition of a flow line into segments called channels. A recursive relationship for the movement within each channel was demonstrated that allowed for a significant reduction in computational complexity over a module by module simulation. This model can be used to incorporate setups that can only begin once an initial portion of the

978-1-4244-5771-7/09/$26.00 ©2009 IEEE

tool empties. Subsequently, Morrison (2009b) addressed the issue of different classes of lots with their own process times. We first provide an introduction these results.

We then turn our attention to converting a generic flow line model for a clustered photolithography tool into a tractable deterministic flow line model. The conversion results in some small loss of fidelity, but as we shall see, the loss is quite reasonable for the purpose of fab-level simulation. We conduct simulations to assess the quality of the resulting flow line model. We observe that, for the representative system studied, the tractable model provides throughput and cycle time predictions within about 1% and 3% of the generic flow line, respectively. Also, the models have been tested on clustered photolithography tools in production and gave predictions for throughput and cycle time within 1% and 4% of actual values, respectively.

The paper is organized as follows. In Section 2, we provide a description of clustered photolithography tools and deterministic flow lines. Theoretical results for the models are reviewed in Section 3. In Section 4, we discuss computational complexity. We turn to the modeling of a fictitious but representative clustered photolithography tool in Section 5. Concluding remarks are presented in Section 6.

2 SYSTEM DESCRIPTION

We begin with an overview of a clustered photolithography tool, hereafter referred to as a CPT. We then turn our attention to deterministic flow lines. We mention similarities and differences between the two. Later, we will show how to construct a flow line model for a representative CPT. We use the shorthand DFL to refer to a deterministic flow line.

2.1 Clustered Photolithography Tools

A CPT consists of a track tool and a photolithography scanner clustered into a single piece of equipment. The track tool consists of numerous processing modules that conduct pre-scan and post-scan operations. Conceptually, the process is quite similar to the process of making film, taking a picture and developing the picture. The pre-scan processing modules prepare wafers for the scanner by depositing light sensitive coatings on the wafers, baking and cooling the wafer. Once the wafer has received all pre-scan operations, it is ready to be exposed to the desired pattern of light in the scanner. The scanner tool conducts an alignment operation and then scans a pattern of light over the wafer. The post-scan operations include image development, baking and chilling. There is often a buffer for storing wafers just before the scanner. A post-scan buffer may also be used. Typically, the pre-scan and post-scan modules are segregated, that is, no module used for a pre-scan operation is used for a post-scan operation, and vice-versa. To ensure that the scanner, which can cost on the order of US $20 million, is the bottleneck of the CPT, redundant modules that provide identical processing are devoted to pre-scan and post-scan operations with long durations.

Wafers enter the tool and advance from one operation to the next via wafer handling robots. There are numerous such robots in the track tool, between the track and scanner tools and internal to the scanner. A robot control algorithm is used to direct the actions of the robots and advance the wafers to their next stage of processing.

A CPT is depicted in Figure 1. Three wafer handling are shown. The operations are denoted as P1, …, P11 (P for process) and there are redundant modules dedicated to some of them. For example, P2 has three modules devoted to it. There are both pre-scan and post-scan buffers shown. The scan process is denoted as P6 and shows a wafer, a reticle that contains the desired pattern and a light. Wafers enter the CPT at process P1 and are complete when they exit process P11.

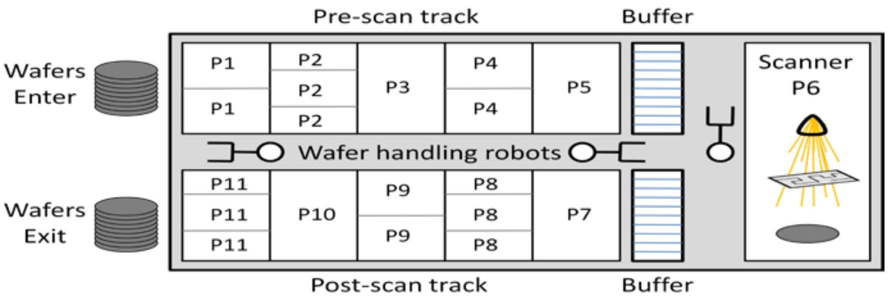

Figure 1: Conceptual layout of a clustered photolithography tool (CPT)

Different lots may require different processing. The modules are designed to provide a variety of operations. For example, a baking module can use different temperatures and durations; a module applying a photo resistive chemical is plumbed for multiple types of chemicals. The process times in a module are thus a function of the type of wafer. Some wafers may not require the service of all processes and skip a collection of process modules. When a module changes from one setting to another, a setup is required. Often, the setup is conducted on all modules in the pre-scan or post-scan track simultaneously and can only begin once the pre-scan track is empty of wafers. The scanner often conducts a setup when starting wafers from a new lot. The scanner setup is often longer in duration than the time required for post-scan track setup so that the post-scan setup can be ignored for modeling purposes.

2.2 Deterministic Flow Lines

A flow line consists of E process modules, denoted m_1, …, m_E, from which wafers require service in order. There is an infinite buffer before the first module and a buffer of finite capacity b_i before module m_i, i = 2, …, E. Let F denote the sum of the internal buffer spaces, that is, $F = b_2 + … + b_E$. Wafers arrive to the system singly or in a batch as an arbitrary process. Wafers are indexed in the order in which they arrive; the arrival time of wafer w is denoted as a_w. The index of wafers in a batch is assigned arbitrarily within the batch. For convenience, we assume that wafers in the buffer preceding m_1 are served in a FIFO fashion. This assumption can be easily generalized. Wafers in each internal buffer are served in a FIFO manner. After receiving service from a module m_i, a wafer advances to module m_{i+1}, if it is unoccupied. If module m_{i+1} is occupied and a slot is available in module m_{i+1}'s buffer, the wafer enters the buffer. If no slot is available in the buffer, then the wafer languishes in module m_i – preventing upstream wafers from accessing that module. This behavior is referred to as manufacturing blocking, see Dallery and Gershwin (1992). A DFL is depicted in Figure 2. There, the buffer serving module m_2 has a capacity of three wafers and the buffer serving module m_E has a capacity of five wafers.

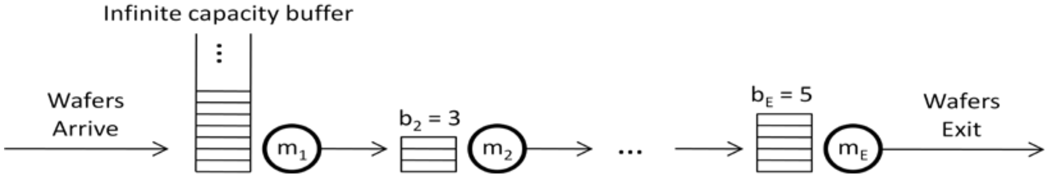

Figure 2: An example of a deterministic flow line (DFL)

There are K classes of wafers, 1, …, K. Use c(w) to denote the class of wafer w. All wafers in class k require the same deterministic service time from server m_j, call it τ_j^k. It is this feature that earns the flow line the deterministic moniker. When the number of classes is more than one, this fact may be highlighted by referring to the system as a multiclass DFL. Unless otherwise mentioned, we will be studying multiclass DFLs, so there is no need for the distinction.

As discussed in Avi-Itzhak (1965) and Morrison (2009a), buffers in a DFL can be modeled as a server with zero process time. The distinction is that buffer modules may be skipped if they are not needed, while a server with zero process time must be passed through. However, since the total time a wafer spends in a buffer module is identical to the time spent in a server with zero process time, the models are equivalent. We can thus equivalently replace a DFL model containing F buffer spaces and E process modules, with a DFL containing M = E + F process modules, where we set the service time to zero if the process module was formerly considered a buffer. Hereafter, we consider that all buffer modules have been replaced by a process module with zero service time.

Example 1 *Consider a DFL consisting of four process modules m_1, …, m_4. There is buffer space for one wafer between each server so that $b_2 = b_3 = b_4 = 1$. Here E = 4 and F = 3, so that the total number of locations for wafers internal to the DFL is M = E+F = 7. In the equivalent DFL modeling buffers as servers with zero process times, modules m_2, m_4 and m_6 replace the buffers and have $\tau_2^k = \tau_4^k = \tau_6^k = 0$.*

Let $X_{w,j}$ denote the time that wafer w starts service with module m_j. The *elementary evolution equations* dictate the progress of wafers through the DFL. They are

$$\begin{aligned} X_{w,1} &= \max\{a_w, X_{w-1,2}\}, \\ X_{w,j} &= \max\{X_{w,j-1} + \tau_{j-1}^{c(w)}, X_{w-1,j+1}\}, \ j = 2, …, M-1, \text{ and} \\ X_{w,M} &= \max\{ X_{w,M-1} + \tau_{M-1}^{c(w)}, X_{w-1,M} + \tau_M^{c(w-1)}\}. \end{aligned} \quad (1)$$

Define $\gamma_{w,j}$ to be the time that module m_j is finished conducting its service of wafer w, not including time spent queueing for the subsequent server. Define the delay in a module as $d_{w,j} := X_{w,j+1} – X_{w,j} – \tau_j^{c(w)}$, for j = 1, …, M. Note that $d_{w,M} = 0$, for all wafers w since there is no contention after the final server. Define $d_{w,0} := X_{w,1} – a_w$; it is the queueing to enter module m_1.

The following assumptions and definitions are useful.

Assumption A0 *There is a distinguished module m_B for which $\tau_B^k \geq \tau_j^k$, for all j, all k, and $\tau_B^k > \tau_i^k$, for all $i < B$, all k. This module is called the* bottleneck.

Assumption A1 *The service times are such that $\tau_j^{k+1} = \eta_k\ \tau_j^k$, for k = 1, ..., K-1, where $0 < \eta_k < 1$. We use η to denote $\eta_1 * \eta_2 * ... * \eta_{K-1}$. Thus, wafers of class c_1 are the slowest in every module and wafers of class c_K are the fastest.*

Definition 1 *A server is called a* dominating module for wafers of class k *if $\tau_j^k > \tau_i^k$, for all $i < j$.*

Assumption A2 For class 1 wafers, the service times between the dominating modules satisfy $\tau_j^1 \leq \eta^{j-\beta(\alpha)}\ \tau_{\beta(\alpha)}^1$, for $\beta(\alpha) < j < \beta(\alpha+1)$, $\alpha = 1, \ldots, \sigma-1$. For $j > \beta(\sigma)$, $\tau_j^1 \leq \eta^{j-B}\ \tau_B^1$.

Under assumption A1, the dominating modules are the same for all classes. In this case, use σ to denote the number of dominating modules and employ α as their index. Let $\beta(\alpha)$ be the module index of dominating module alpha. Thus, if module m_7 is the third dominating module, $\beta(3) = 7$; also, $\beta(\sigma) = B$.

Definition 2 *Under assumption A1, the modules $m_{\beta(\alpha)}, \ldots, m_{\beta(\alpha+1)}$ are termed* channel-α.

The next example serves to demonstrate the above definitions and assumptions.

Example 1 *Consider a DFL with seven modules. There are three customer classes. The service times are depicted in Table 1. The τ_j^k satisfy Assumption A0; module m_6 is the bottleneck. The τ_j^k satisfy Assumption A1 with $\eta_1 = ⅔$ and $\eta_2 = ¾$. That is, $\tau_j^2 = ⅔\ \tau_j^1$ and $\tau_j^3 = ¾\ \tau_j^2$, for all j. Note that $\eta = \eta_1 * \eta_2 = ½$. The dominating modules, whose columns are shaded in Table 2, are m_1, m_4 and m_6. Also, Assumption A2 is satisfied since, for example, $\tau_2^2 = 20 \leq \eta\ \tau_1^2 = 0.5*(40) = 20$ and $\tau_3^2 = 0 \leq \eta\ \tau_2^2 = 0.5*(20) = 10$. There are two channels; channel-1 is the set of modules $\{m_1, \ldots, m_4\}$ and channel-2 is $\{m_4, \ldots, m_6\}$.*

Table 1: Service times for the DFL of Example 1

τ_j^k	m1	m2	m3	m4	m5	m6	m7	m8
Class 1	60	30	0	75	37.5	90	45	22.5
Class 2	40	20	0	50	25	60	30	15
Class 3	30	15	0	37.5	18.75	45	22.5	11.25

2.3 Comparison Between a CPT and a DFL model

We first mention differences between the CPT and general DFL. Unlike the CPT, there is but a single path from entrance to exit in a DFL. Nor are there wafer transport robots in the DFL, wafers immediately transfer from one location to the next when available. In the nominal DFL, there are no setups. We will discuss how to incorporate setups later. Also, we will show an easy way to convert a multipath CPT into a similar single path system.

We next turn to Assumptions A0, A1 and A2. Generally in a CPT, the photolithography scanner is the bottleneck for each class of wafer. However, this is not always true, though it is certainly desirable due to the substantially higher cost of the scanner. Assumption A0 restricts to the case of a single bottleneck module. Assumptions A1 and A2 are more restrictive, but they are required to allow the mathematical structure we will discuss in the sequel.

Despite the restrictive nature of Assumption A1 and A2 and the differences between a real CPT and a DFL, a DFL can serve as a good model for cycle time and throughput. Further, it is possible to analyze their behavior with about one order of magnitude less computation than is required for a DFL without A1 and A2. Beyond this, because we ignore the robot in the DFL, there is substantial additional computational reduction. These observations will be discussed further in a later section.

3 WAFER ADVANCEMENT IN DETERMINISTIC FLOW LINES

It was shown in Morrison (2009b) that there is no contention possible after the bottleneck in a multiclass DFL under Assumptions A1 and A2. Further, the channels in such a system behave in a regular way and serve as a decomposition of the DFL. Using these facts, one can develop recursions for the manner in which wafers advance within the flow line without resorting to a full simulation of the entire DFL. Further, one can incorporate state dependent setups and setups at the bottleneck. Here we provide an introduction to these results.

Note that there may be *setups at the bottleneck* corresponding to reticle alignment delays experienced between lots in the photolithography scanner. Thus, hereafter, we allow the bottleneck process times to depend upon the wafer. That is, while A1 and A2 hold with the given nominal process times, let $\tau_{w,B} = \tau_B^{c(w)} + s_{w,B}$, where $s_{w,B} \geq 0$.

3.1 No contention after the bottleneck

Under assumptions A1 and A2 for the bottleneck and post-bottleneck modules, it can be shown that there is no contention after the bottleneck. The implications of this are stated in Theorem 1. For convenience we let $T_{i,j}^{k} \coloneqq \sum_{l=i}^{j} \tau_l^k$.

Theorem 1 *Under Assumptions A1 and A2, the following hold for systems with M > B*

$$X_{w,j+1} = X_{w,j} + \tau_j^{c(w)} + s(w) \cdot I_{\{j=B\}},\ \text{for all } B \leq j < M,$$

$$X_{w+1,j} \geq X_{w,j} + \tau_B(w) + \left[T_{B,j-1}^{c(w+1)} - T_{B,j-1}^{c(w)}\right],\ \text{for all } B \leq j \leq M.$$

The departure times from the last modules obey

$$\gamma_{w,M} = X_{w,j} + \tau_M^{c(w)},$$

$$\gamma_{w+1,M} \geq X_{w,M} + \tau_B(w) + \left[T_{B,M}^{c(w+1)} - T_{B,M}^{c(w)}\right].$$

This result allows us to essentially ignore dynamics after the bottleneck module and simply replace all post-bottleneck modules with a post-processing delay equal to $T_{B+1,M}^{c(w)}$.

3.2 Delays inside the flow line

To model setups that depend upon the location of wafers, we must have visibility to wafer advancement inside the DFL. As an alternate option to the elementary evolution equations (1), a structural property of the channels allows us this visibility. This result is stated after a few helpful definitions.

Definition 3 *Let $Y^{\alpha}(w)$ denote the total delay wafer w experiences in the modules of channel-α, excluding delay in the final module of that channel. That is $Y^{\alpha}(w) = \sum_{j=\beta(\alpha)}^{\beta(\alpha+1)-1} d_{w,j}$.*

Definition 4 *Let $S_p^{\beta(\alpha)}(w)$, for $p > \beta(\alpha)$, denote the maximum possible delay wafer w may experience in the modules m_p, ..., $m_{\beta(\alpha)-1}$. That is $S_p^{\beta(\alpha)}(w) = \left\{\sum_{j=w+p-\beta(\alpha)}^{w-1}\left[\tau_{\beta(\alpha)}^{c(j)} + d_{j,\beta(\alpha)}\right]\right\} - T_{p,\beta(\alpha)-1}^{c(w)}$.*

Theorem 2 *Consider a DFT under assumptions A1 and A2 with $\sigma > 2$. For each channel-α, $1 \leq \alpha < \sigma$, the following recursions hold for the channel delays for w = 1, 2, ...,*

$$Y^{\alpha}(w) = min\left\{\begin{matrix} S_{\beta(\alpha)}^{\beta(\alpha+1)}(w), Y^{\alpha}(w-1) + \tau_{\beta(\alpha+1)}^{c(w-1)} + d_{w-1,\beta(\alpha+1)} - max\left\{\tau_{\beta(\alpha)}^{c(w-1)}, a_w^{\alpha} - X_{w-1,\beta(\alpha)}\right\}^{+} \\ + \left[T_{\beta(\alpha),\beta(\alpha+1)-1}^{c(w-1)} - T_{\beta(\alpha),\beta(\alpha+1)-1}^{c(w)}\right] \end{matrix}\right\}^{+} ,\text{(3)}$$

where $\{.\}^{+} = max\{0,.\}$, $d_{w,B} = s_{w,B}$, $a_w^{\ 1} = a_w$, $a_w^{\ \alpha} = X_{w,\beta(\alpha)}$. To calculate $S^{\alpha}(1)$, let $\tau_{w,B} = \tau_{1,B}$, w = 0, -1, -2, ... The start times at channel-α are given as

$$X_{w,1} = max\left\{a_w, X_{w-1,1} + \tau_1^{c(w-1)} + d_{w-1,1}\right\},$$

$$X_{w,\beta(\alpha)} = a_w + d_{w,0} + \sum_{\varphi=1}^{\alpha-1} Y^{\varphi}(w) + T_{1,\beta(\alpha)-1}^{c(w)},\ for\ \alpha = 1, \ldots, \sigma - 1.$$

In each channel-α, the delays at module j = β(α), ..., β(α+1)–1 are given b

$$d_{w,j} = min\left\{\tau_{\beta(\alpha+1)}^{c(w+j-\beta(\alpha+1))} + d_{w+j-\beta(\alpha+1),\beta(\alpha+1)} - \tau_{\beta(\alpha)}^{c(w)}, Y^{\alpha}(w) - S_{j+1}^{\beta(\alpha+1)}(w)\right\}^{+}. \tag{4}$$

The initial conditions are $Y^{\alpha}(0)=0$, $a_0 = -\infty$, $X_{0,1} = -\infty$, and $d_{w,j} = 0$, $w \leq 0$.

Theorem 2 has several implications. For a wafer in channel-α, Equation (4) states that if the wafer experiences delay in any module of that channel except the last one, then there is a first module at which delay occurs, the delay is zero for preceding modules in the channel, and the delay at subsequent modules in the channel is the maximum possible. Equation (3) states that the total delay a wafer experiences in a channel can be recursively calculated.

3.3 Practical features

Computationally, the approach of Theorem 2 is roughly equivalent to that of the elementary evolution equations (1). We next give a recursion for the evolution of delay in a single channel DFL. The computation is smaller since there is only one $Y^{\sigma-1}(w)$ to assess. As we shall see, using a single channel model – even for a more complicated system – will allow us to appropriately incorporate setups as well as accurately model practical CPTs.

Corollary 1 *The delay in a single channel DFT satisfying assumptions A1 and A2 obeys the recursion*

$$Y^{\sigma-1}(w) = min\left\{S_1^B(w), Y^{\sigma-1}(w-1) + \tau_{w-1,B} - max\left\{\tau_1^{c(w-1)}, a_w - X_{w-1,1}\right\} \; + \left[T_{1,B-1}^{c(w-1)} - T_{1,B-1}^{c(w)}\right]\right\}^+, \qquad (5)$$

with initial conditions $Y^{\sigma-1}(0)=0$, $a_0 = -\infty$, $X_{0,1} = -\infty$, $\tau_{w,j} := \tau_{1,j}$, for all $w \leq 0$. The delay in each module of channel-(σ–1) is given as

$$d_{w,j} = min\left\{\tau_{w+j-B,B} - \tau_j^{c(w)}, Y^{\alpha}(w) - S_{j+1}^B(w)\right\}^+.$$

The entry times to the channel obey the recursion

$$X_{w,1} = max\left\{a_w, X_{w-1,1} + \tau_1^{c(w-1)} + d_{w-1,1}\right\},$$

with the initial conditions above ($Y^{\sigma-1}(0)=0$ implies that $d_{0,1} = 0$).

Consider once more a DFL consisting of one or more channels. We now turn our attention to the modeling of setups that can only begin once all modules prior to a distinguished one are vacant. Such setups may occur in CPTs when changing from one class of lot to another. In this case, before the new type of lot can enter the first module, all modules in the pre-scan track must be empty. Typically, this means that some wafers of the previous lot are located in the pre-scan buffer and will continue to advance into the scanner as it requires them. Once the pre-scan track is empty, the setup of those modules is initiated. When the setup is complete, wafers from the new lot may enter production.

Two possible methods to determine when the setup commences are discussed next. First, one could wait for both the next lot to arrive as well as for the pre-scan track to empty. However, if it is known what lot will next be processed on the tool, one can simply begin the setup as soon as the required modules are empty. Let P(w) denote the index of the last module that must be vacant before the setup of wafer w. That is, m_1, …, $m_{P(w)}$ must be vacant. Let $V_{w-1,P(w)}$ denote the time instant at which wafer w-1 vacates module P(w). Let $\tau_S(w)$ denote the duration of the setup required. If no setup is required for a wafer, we set P(w) = 1 and $\tau_S(w) = 0$.

With this notation, for the case where the setup commences only when both wafer w has arrived and the modules are vacant, let $X^*_{w,1}$:= max{ a_w, $V_{w-1,P(w)}$ } + $\tau_S(w)$. For the case where the setup does not wait for the wafer to arrive, let $X^*_{w,1}$:= max{ a_w, $V_{w-1,P(w)} + \tau_S(w)$ }. Let $\alpha_{P(w)} \in \{1, \dots, \sigma\}$ denote the index of the dominating module prior to P(w), so that $\beta(\alpha_{P(w)}) \leq P(k) < \beta(\alpha_{P(w)}+1)$. In general, the vacation time of wafer w-1 from a module P(w) may be calculated as

$$V_{w-1,P(w)} = X_{w-1,1} + T_{1,P(w)}^{c(w-1)} + \sum_{\alpha=1}^{\alpha_{P(w)}-1} Y^{\alpha}(w-1) + \sum_{n=\beta(\alpha_{P(w)})}^{P(w)} d_{w-1,n}.$$

The next result follows.

Corollary 2 *For a DFL under Assumptions A1 and A2 with state dependent setups as described above, Theorem 2 holds with a_w replaced with $X^*_{w,1}$. If the DFL consists of a single channel, Corollary 1 similarly holds.*

In addition to setups, it is common for wafers to arrive in batches termed wafer lots. Such lots may consist of up to twenty five wafers. Because all wafers in a lot arrive to the tool simultaneously and a setup is not required between the wafers within lot, the recursions for wafer delay may be simplified. Let g denote the index of wafer lots. Use Ω(g,w) to denote the wafer index of the w-th wafer of lot g. For example, if 300 wafers have already arrived to the tool and the next arriving lot is lot 15, the 7-th wafer of lot 15 is wafer number Ω(15,7) = 307. Use W(g) to denote the number of wafers in lot g. For convenience, also use c(g) to denote the class of the wafers in lot g. Thus, c(Ω(g,w)) = c(g).

Corollary 3 *Consider a single channel DFT under Assumptions A1 and A2 with batch arrivals of wafer lots. Assume that the first wafer of a lot may have $s_{w,B} \neq 0$, but all others have $s_{w,B} = 0$. The following recursion holds for the delay the last channel, for w = 1, …, W(g) – 1,*

$$Y^{\sigma-1}\big(\Omega(g,w+1)\big) = min\left\{S_1^B\big(\Omega(g,w+1)\big), Y^{\sigma-1}\big(\Omega(g,w)\big) + \tau_{\Omega(g,w),B} - \tau_1^{c(g)}\right\} \; .$$

4 COMPUTATIONAL COMPLEXITY

The recursions of Theorem 2 require just about as much computation as do the elementary evolution equations (1). However, restricting attention to a single channel DFL and employing simplifications for wafer lots allows one to reduce the complexity. The following theorem from Morrison (2009b) compares the operations required for the various approaches. Use FS to denote "full simulation" as in the elementary evolution equations (1). Let TH2 denote the approach of Theorem 2. Let C3 denote the approach of Corollary 3. Note that addition and subtraction are considered equivalent, as are maximization and minimization. Recall that B is the index of the bottleneck module, σ is the number of dominating modules and K is the number of classes of wafers.

Note that although we assume a fixed W wafers per lot, the recursions developed work for W(g) a function of the lot.

Theorem 3 *The computations required to simulate G lots of W wafers each are shown in Tables 2 and 3. Note that C2 assumes there is a single channel and hence $\sigma = 2$.*

Table 2: Initialization computations for simulation of a DFL

Method	# of Add	# of Mult
FS	0	0
TH2	$K^2(2B-\sigma-1)+K(2\sigma-B-K-1)+2B-2$	1
C2	K(K-1)(B-1)	0

Table 3: Recursion computations for simulation of a DFL

Method	# of Add	# of Max
FS	GWB-1	GWB-B
TH2	$(17\sigma-20)(WG-1)$	$(5\sigma-6)(WG-1)$
C3	9G(W+1)	2G(W+2)

Note that the computation required for the FS method depends greatly upon the index of the bottleneck module. If this index is less than around 10, the FS approach can be better. Often, however, the bottleneck is module twenty or more, depending on the number of buffer modules. The next example demonstrates the practical application of the results and reviews the computational complexity of each approach.

Example 2 *Consider a multiclass DFT consisting of 45 modules as is common in real tools. Suppose there are K=3 classes of wafers. Let B = 35, $\sigma = 3$, $\beta(\sigma-1) = 8$. We will determine the calculations required to simulate for 100 lots, each consisting of W = 25 wafers. Table 4 gives the results. The C3 recursion requires a total of about 1/6 the computations of FS.*

Table 4: Computation required to simulate 100 lots in Example 2

Initialization		
Method	# of Add	# of Mult
FS	0	0
TH2	563	1
C2	204	0

Recursion		
Method	# of Add	# of Max
FS	87499	87465
TH2	77469	22491
C2	23400	5400

5 MODELING OF A CPT

In this section we first develop a generic flow line models for clustered photolithography tools. We then show how to convert this model into a flow line obeying Assumptions A1 and A2. Finally, we demonstrate via simulation that, for a fictitious yet representative CPT model, the simplified one channel DFL model satisfying assumptions A1 and A2 predicts the cycle time and throughput quite well. Subsection 5.1 discusses the development of a generic DFL from a CPT. Subsection 5.2 compares the DFL models via simulation. In the last subsection, we mention the results obtained when applying these models to a clustered photolithography tool in production.

5.1 Converting a CPT to a DFL

In this subsection we consider a representative CPT and discuss several modeling approaches to convert it to a DFL. We conduct these investigations via several examples. The following notation is helpful. Let N denote the number of processes in a CPT. Process time for a wafer of class k in process p is denoted as W_p^k. Let R_p denote the number of modules devoted to process p. Let B_p denote the number of buffer spaces allocated to process p. Assume there is a transfer time for unloading or loading wafers that cannot be conducted in parallel with the process. Denote this time by τ_T. It represents overhead in entering or exiting a module, but does not include queueing for a wafer transport robot or module. For example, if there is a single module for a process p, then the minimum time between wafer departures from that process is $W_p^k + \tau_T$. In general, the maximum rate of wafer departures from a process p is one wafer every $(W_p^k + \tau_T)/R_p$ units of time. We thus effectively add τ_T seconds to each process time. Hereafter, we will assume that the CPT is not robot limited and account for the robot only via this additional process time for each process. The next example demonstrates the conversion of a CPT into a DFL.

Example 3 *Consider the CPT of Figure 1. There are three classes of wafer lots, K = 3. There are 24 pre-scan buffer slots and none after process P6. Process time for a wafer in process p of class k is denoted as W_p^k. Assume that process P6 is the bottleneck. Table 5 gives the process times W_p^k. Let $\tau_T = 5$ seconds.*

The flow line model consists of a series of $\sum_{p=1}^{N} R_p + \sum_{p=2}^{N} B_p$ modules. For each process p, the DFL consists of R_p serial modules with class dependent process times $\tau_j^k = (W_p^k + \tau_T)/R_p$. Prior to the modules for process p are B_p modules with zero process time. Thus, since $R_1 = 2$, $\tau_1^k = \tau_2^k = (W_1^k + 5)/2$. In particular, $\tau_1^1 = \tau_2^1 = 50$ s, $\tau_1^2 = \tau_2^2 = 45$ s and $\tau_1^3 = \tau_2^3 = 40$ s. For our example, $R = (R_1, \ldots, R_N) = (2, 3, 1, 2, 1, 1, 1, 3, 2, 1, 3)$. The resulting DFL process times are given Table 6. The bottleneck module for each class is module m_{34}, so that B = 34. Note that Assumptions A1 and A2 are not satisfied.

Table 5: Process times by class for the system of Example 3

Class	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
1	95	160	30	85	40	55	38	140	95	41	143
2	85	130	40	75	37	45	24	115	65	39	112
3	75	115	32	65	37	40	19	85	67	35	82

Table 6: Process times by class for the DFL model of Example 3

Class	t_1^k	t_2^k	t_3^k	t_4^k	t_5^k	t_6^k	t_7^k	t_8^k	t_9^k	t_{10}^k	…	t_{33}^k	t_{34}^k	t_{35}^k	t_{36}^k	t_{37}^k	t_{38}^k	t_{39}^k	t_{40}^k	t_{41}^k	t_{42}^k	t_{43}^k	t_{44}^k
1	50	50	55	55	55	35	45	45	45	0	…	0	60	43	48⅓	48⅓	48⅓	50	50	46	49⅓	49⅓	49⅓
2	45	45	45	45	45	45	40	40	42	0	…	0	50	29	40	40	40	35	35	44	39	39	39
3	40	40	40	40	40	37	35	35	42	0	…	0	45	24	30	30	30	36	36	40	29	29	29

The DFL model at this stage will not typically satisfy Assumptions A1 and A2. However, one can easily generate DFL models that do, while still retaining properties such as total process time and bottleneck process rate. While there is some loss of fidelity, the modeling error is quite small for the models under consideration. The approach is to create a multiclass DFL that retains the bottleneck process times per class by defining $\eta_1 = \tau_B^2/\tau_B^1$, …, $\eta_{K-1} = \tau_B^K/\tau_B^{K-1}$ and $\eta = \eta_1 * \eta_2 * \ldots * \eta_{K-1}$. For class 1, recall that the process time of the penultimate dominating module $m_{\beta(\sigma-1)}$ is denoted $\tau_{\beta(\sigma-1)}^1$. To obtain a DFL model satisfying A1 and A2, we define the following process time parameters υ_j^k for module j and class k. Let $\upsilon_1^1 = \tau_{\beta(\sigma-1)}^1$, $\upsilon_2^1 = \eta\ \tau_{\beta(\sigma-1)}^1$, $\upsilon_3^1 = (\eta)^2\ \tau_{\beta(\sigma-1)}^1$, …, $\upsilon_{B-1}^1 = (\eta)^{B-1}\ \tau_{\beta(\sigma-1)}^1$ and $\upsilon_B^1 = \tau_B^1$. For the other classes, define $\upsilon_j^{k+1} = \eta_k\ \upsilon_j^k$, for classes k = 2, …, K and all modules. The rationale for placing the process time $\upsilon_1^1 = \tau_{\beta(\sigma-1)}^1$ of the penultimate dominating module at the first module of the new DFL model is to obtain a flow line with a single channel! Thus, we can have the computational reduction properties associated with a focus on single channel DFLs (as in Corollary 1).

To ensure that the total process time for a wafer of class k remains the same, we add modules after the bottleneck. Assuming there is no contention after the bottleneck, the post-bottleneck modules merely have the effect of adding a fixed additional process time. For each class, set this post-bottleneck process time to $\sum_{j=1}^{M} \tau_j^k - \sum_{j=1}^{B} \upsilon_j^k$, whenever this is greater than or equal to 0. If this difference is negative the approach discussed will not work. One could rectify this by setting some of the intermediate process times before the bottleneck to 0.

We illustrate the details via example.

Example 4 *For the DFL of Example 3, B = 34 and we have* $\eta_1 = \tau_B^{2}/\tau_B^{1} = 50/60 = 5/6$, $\eta_2 = \tau_B^{3}/\tau_B^{2} = 45/50 = 0.9$ *and* $\eta = \eta_1\ \eta_2 = 0.75$. *For class 1 wafers, the penultimate dominating module is* m_3 *and has* $\tau_{\beta(\sigma-1)}^{1} = 55$. *We thus set* $\upsilon_1^{1} = 55$, $\upsilon_2^{1} = \eta\ 55 = 41.25$, $\upsilon_3^{1} = (\eta)^2\ 55 = 30.9375$, *...,* $\upsilon_{B-1}^{1} = (\eta)^{B-2}\ 55 = (¾)^{32*}55$ *and* $\upsilon_B^{1} = 60$. *Obeying Assumption A1 gives* $\upsilon_j^{2} = \eta_1\ \upsilon_j^{1}$ *and* $\upsilon_j^{3} = \eta_2\ \upsilon_j^{2}$, *for* $j \leq B$. *To determine the post-bottleneck process time it is convenient to use the fact that* $\sum_{j=1}^{B} \upsilon_j^k = \tau_B^1 + \tau_{\beta(\sigma-1)}^1(1-\eta^{B-1})/(1-\eta)$ = *60 + 220*(1-0.75*^{33}*) ≈ 297.9834. Since* $\sum_{j=1}^{M} \tau_j^1 = 977$, *we set the post-bottleneck processing times for class 1 to* $\sum_{j=1}^{M} \tau_j^k - \sum_{j=1}^{B} \upsilon_j^k = 977 - 297.98 \approx 697.02$. *Similarly, for classes 2 and 3. The module process times for a single channel DFL model satisfying assumptions A1 and A2 is shown in Table 7.*

Table 7. Process times of a DFL model satisfying Assumptions A1 and A2

Class	ν_1^{k}	ν_2^{k}	ν_3^{k}	...	ν_{33}^{k}	ν_{34}^{k}	Post-B Sum
1	55.00	$55(\eta)^1$	$55(\eta)^2$	...	$55(\eta)^{32}$	60	697.0
2	η_1*55	η_1*$55(\eta)^1$	η_1*$55(\eta)^2$	...	$\eta_1 55(\eta)^{32}$	50	588.7
3	η_1*η_2*55	η_1*η_2*$55(\eta)^1$	η_1*η_2*$55(\eta)^2$	...	η_1*η_2*$55(\eta)^{32}$	45	497.0

5.2 Simulations to assess loss of fidelity

The multiclass DFL model of Example 4 was developed from the original CPT model and obeys assumptions A1 and A2. It will thus be amenable to analysis as in Corollaries 1, 2 and 3. The question that remains, however, is this: How accurate is the new model in comparison to the original? In the remainder of this subsection we use the notation DFL-1-A to denote a DFL possessing a single channel (the "1") and satisfying both A1 and A2 (the "A").

First note that a DFL-1-A system will typically allow setups to start earlier than a generic one channel DFL (or a DFL model of a CPT). The reason for this is that wafers in the DFL-1-A system will advance faster through modules m_1, …, m_P due to the geometric decay of the process times. This of course assumes that the sum of the process times for modules m_1, …, m_P in the DFL-1-A system is less than in the generic one channel DFL. Note that the elementary evolution equation for $X_{w,1}$ can in general be written as $X_{w,1} = max\{a_w, V_{w-1,P(w)}\} + \tau_S(w)$, including the possibility of setups. Recall that if no setup is required for a wafer, we set P(w) = 1 and τ_S(w) = 0.

To account for the tendency of the DFL-1-A system to allow wafers into the system too early, we use the following adjusted start time for wafers in module m_1. Set $X_{w,1} = max\{a_w, V_{w-1,P(w)} + A_{w-1,P(w)}\} + \tau_S(w)$, where the adjustment $A_{w-1,P(w)}$ is defined next.

$$A_{w,P(w+1)} := \begin{cases} min\left\{\sum_{j=2}^{B}\left[\tau_j^{c(w)} - \upsilon_j^{c(w)}\right], max\left\{0, Y^1(w) + \sum_{j=2}^{B-1}\left[\tau_j^{c(w)}\right] - \sum_{l=w-B+2}^{w-1}\left[\upsilon_j^{c(w)} + s_{w,B}\right]\right\}\right\}, & for\ P(w+1) = 1, \\ \sum_{j=1}^{P}\left[\tau_j^{c(w)} - \upsilon_j^{c(w)}\right], & for\ P(w+1) > 1. \end{cases}$$

Since most of the terms in this expression can be calculated during the initialization, the overall complexity of the resulting simulation is not harmed too much.

In order to address the question of how much is lost when converting from a DFL model of a CPT to a DFL-1-A, we conduct simulation experiments for the models described in Examples 3 and 4 above.

Example 5 *We compare the systems of Examples 3 and 4. Suppose that all lots consist of 12 wafers, that is, W = 12. Lots arrive to the system as a Poisson process; the rates will be given later. Lots are processed in a first come first served manner. Each arriving lot is of class 1, 2 or 3 with equal probability independent of all other random events. For each simulation case, we conduct 20 replications each consisting of 1250 lots. We use only the last 1000 lots for our data to ensure we use steady state values. The mean time a lot is in the tool (from the entry of wafer 1 at module* m_1 *to the exit of wafer W from the last module) is calculated for each replication. This gives 20 values for each simulation study (one value that is the average over the last 1000 lots in the replication). The mean and standard deviation of these 20 values are then calculated and reported in Table 8. Similarly for the TBLO, which is shorthand for the time between lots out of the tool. It is defined as* $T_g := min\{c_g - s_g,\ c_g - c_{g-1}\}$, *where* c_g *is the completion time of lot g from the tool and* s_g *is the start time of the first wafer of lot g on the tool. The throughput is calculated as the total number of wafers completed divided by the simulation time.*

In all simulations we assume that the setup module $P = 9$, $\tau_S(w)$ is uniformly distributed from 2 to 4 minutes. Setups are required when changing from on class of wafer to another. The delay $s_{w,B}$ for the first wafer of every *lot at the bottleneck is uniformly distributed from 4 to 6 minutes. Four cases are studied:*

- *EX3: JIT – This case studies the system of Example 3. Here there is an infinite supply of lots at the entrance of the system; JIT stands for just in time. The result is that the simulation gives the maximum system throughput.*
- *EX4: JIT – This case studies the system of Example 4. Here there is an infinite supply of lots at the entrance of the system; JIT stands for just in time. The result is that the simulation gives the maximum system throughput.*
- *EX3: 90% – This case studies the system of Example 3. The arrival rate of wafers is set to 90% of the JIT throughput from case EX3: JIT.*
- *EX4: 90% – This case studies the system of Example 4. The arrival rate of wafers is set to 90% of the JIT throughput from case EX4: JIT.*

The results of the simulations are given in Table 8.

Table 8. Results of the simulations of Example 5

	EX3: JIT	EX4: JIT
Mean Time in Tool (hours)	0.682	0.682
Std of 20 Values for Mean Time in Tool (hours)	0.020	0.021
Mean TBLO (hours)	0.261	0.261
Std of 20 Values for Mean TBLO (hours)	0.001	0.002
Throughput (wafers/hour)	46.0	46.0

	EX3: 90%	EX4: 90%
Mean Time in Tool (hours)	0.608	0.606
Std of 20 Values for Mean Time in Tool (hours)	0.020	0.019
Mean TBLO (hours)	0.280	0.280
Std of 20 Values for Mean TBLO (hours)	0.005	0.005
Throughput (wafers/hour)	41.4	41.4

The mean time in tool and mean TBLO for the systems of Examples 3 and 4 are virtually identical under the same arrival process conditions. Thus the relaxation from the DFL model of a CPT to a DFL-1-A causes little loss of fidelity. This is due to the adjustments $A_{w,P(w+1)}$ and the fact that the CPT system studied has somewhat similar throughput rates for each process. It is quite common in practice for the throughput rates of each process to be similar due to the use of redundant chambers to ensure that the scanner is the bottleneck.

5.3 Comparison of a DFL model with actual data from a production CPT

We have tested the models against data from a CPT in production. For this case, the use of a DFL model created from the approach of Example 3 provided throughput and cycle time predictions within 0.8% and 3% of the real tool data, respectively. Part of this small loss of fidelity may be due to the fact that we are largely ignoring the wafer transport robots. We also tested a DFL model satisfying Assumptions A1 and A2, as in Example 4, and obtained results within 1% and 4% of the cycle time and throughput, respectively. Thus, the conversion from a basic DFL model to one with a single channel satisfying Assumptions A1 and A2 resulted in very little loss of fidelity.

6 CONCLUDING REMARKS

We have introduced multiclass deterministic flow line models and described how practical features such as state dependent setups and reticle alignment delays for the first wafer in a lot can be incorporated. It was discussed how the models can be more computationally tractable than a normal flow line simulation by about one order of magnitude.

Following the introduction to the models, we next discussed the application of the models to the simulation of clustered photolithography tools (CPT). A deterministic flow line model (DFL) was developed for a CPT that consisted of a single channel with geometrically decaying service rates from module to module. As a result, the theory discussed previously in the paper readily applies and can be used for tractable simulation of the system. To address the key question of how well the models describe the behavior of a CPT, we conducted simulation studies. It was demonstrated that with an explicitly defined adjustment to wafer start times, the DFL model was extremely accurate. In fact, in tests with a CPT in production, the models have provided cycle time and throughput values within 1% and 4% of actual, respectively.

An important caveat is that the models ignore wafer transport robots except to add a constant travel time to the module process times. However, based on the study of data from a real CPT as just mentioned, this seems sufficient for the purposes of fab-level simulation. Thus, the models proposed appear to be promising candidates to replace existing less expressive tool models in fab-level simulation for select key groups of tools.

There are numerous avenues for future research. First, it would be interesting to quantify the loss of fidelity brought about by ignoring the wafer transport resource. Also, a theoretical comparison of various DFLs to quantify the exact loss of fidelity caused by moving the penultimate dominating module to the first position and the geometric decay of process times would be of interest. A practical avenue of study would be to develop simplifications for cases where the module setups are conducted without requiring the entire pre-scan track to be vacant.

REFERENCES

Altiok, T. 1996. *Performance evaluation of manufacturing systems*. New York, New York: Springer-Verlag.

Avi-Itzhak, A. 1965. A sequence of service stations with arbitrary input and regular service times. *Management Science* 11(5): 565-571.

Avi-Itzhak, A., and M. Yadin. 1995. A sequence of servers with arbitrary input and regular service times revisited. *Management Science* 41(6): 1039-1047.

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

Dallery, Y., and S. B. Gershwin. 1992. Manufacturing flow line systems: A review of models and analytical results. *Queueing Systems* 12(1): 3-94.

Friedman, H. D. 1965. Reduction methods for tandem queueing systems. *Operations Research* 13(1): 121-131.

Morrison, J. R. 2008. Flow lines with regular service times: Evolution of delay, state dependent failures and semiconductor wafer fabrication. *Proceedings of the 4th IEEE Conference on Automation Science and Engineering*, 247-252. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Morrison, J.R. 2009a. Deterministic flow lines with applications. To appear in the *IEEE Transactions on Automation Science and Engineering*.

Morrison, J.R. 2009b. Regular flow line models for semiconductor cluster tools: A case of lot dependent process times. To appear in the *Proceedings of the 5th IEEE Conference on Automation Science and Engineering*. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Papadopoulos, H. T., and C. Heavey. 1996. Queueing theory in manufacturing systems analysis and design: A classification of models for production and transfer lines. *European Journal of Operational Research* 92(1): 1–27.

Pillai, D. 2006. The future of semiconductor manufacturing: Factory integration breakthrough opportunities. *IEEE Robotics & Automation Magazine* 13(4): 16-24.

AUTHOR BIOGRAPHY

JAMES R. MORRISON is an Assistant Professor in the Department of Industrial and Systems Engineering at KAIST, Republic of Korea. He received his Ph.D. in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign. Beginning in September 2008, he served as a co-chair for the IEEE Robotics and Automation Society Technical Committee on Semiconductor Manufacturing Automation. He is active in IEEE. From 2000 to 2005, he was with the IBM Fab Operations Engineering department at IBM's Burlington Vermont semiconductor wafer fabricator. His email is <james.morrison@kaist.edu>.