



Conversion of non-calendar to calendar-time based preventive maintenance schedules for semiconductor manufacturing systems

Semiconductor
manufacturing
systems

259

José A. Ramírez-Hernández and Emmanuel Fernandez
*Department of Electrical and Computer Engineering, University of Cincinnati,
Cincinnati, Ohio, USA, and*

Matilda O'Connor and Nipa Patel
Advanced Micro Devices Inc., Austin, Texas, USA

Abstract

Purpose – The aim of this paper is to present the rationale, a numerical example and a case study of the application of an algorithm to convert non-calendar based preventive maintenance (PM) schedules into calendar-time format for semiconductor manufacturing systems (SMS). The resulting calendar-time PM schedules can be utilized as a baseline within a PM scheduling optimization process.

Design/methodology/approach – The algorithm utilizes estimations of work-in-process (WIP) and system parameters to estimate an equivalent calendar-time schedule for PM schedules based on different units. A numerical example based on fictitious data illustrates the utilization of the conversion algorithm within a mixed PM scheduling scenario, including wafer, processing-time and energy-based PM tasks for multi-chamber tools. In addition, a case study illustrates the accuracy of the algorithm by comparing estimated PM targets (i.e. due, warning and late dates) with historical data from a real semiconductor fabrication facility.

Findings – Results from the case study validated the conversion algorithm by showing accurate estimations of PM targets (i.e. due, warning and late dates). The accuracy of the algorithm depends, however, on good estimates for WIP levels within the planning horizon.

Originality/value – The conversion algorithm may be utilized not only in SMS but also in other industries that require the conversion of non-calendar based PM schedules into calendar-time format for PM optimization and operational purposes.

Keywords Preventive maintenance, Semiconductors, Programming, Real time scheduling

Paper type Research paper

Introduction

Preventive maintenance (PM) is an important operation in semiconductor manufacturing systems. PM is performed by intentionally taking off-line tools to perform a prescribed maintenance task. A good PM schedule improves tools availability and production performance, while reducing the costs derived from PM operations, Work-In-Process (WIP) inventory, and tool breakdowns (Yao *et al.*, 2001;



2002, 2004). When PM tasks are properly performed, these will produce a trade-off between planned down-time due to PM operations and costly unplanned tool failures.

The scheduling of PM tasks in a semiconductor fab is a non-trivial operation. On the one hand, the importance of PM in semiconductor manufacturing is clearly justified by the large capital invested in the equipment utilized in the fabrication process and the profits derived from this enterprise. For instance, the revenues of the current semiconductor industry reached more than \$213 billion during 2004 (SIA, 2006) and a new fab using technology of 300 mm wafers can cost between \$2.5 and \$3.5 billion (Blau, 2003), with about 80 percent of this cost going to equipment costs. On the other hand, due to the complexity of scheduling PM tasks, heuristic methods (e.g. using cumulative experience from the fab operation) are regularly used to address this problem. Utilizing optimization or mathematical-based methods to optimally schedule PM tasks in semiconductor fabs have received significant attention recently, see, e.g. Yao *et al.* (2001, 2002, 2004), where models and algorithms are presented to optimally schedule PM tasks in semiconductor manufacturing systems, by making use of operational information (e.g. estimated WIP, tool parameters, number of technicians required per PM operation).

This paper continues the line of research presented in (Ramírez-Hernández and Fernández-Gaucherand, 2003), where an algorithm for converting non-calendar to calendar-time based PM schedules for semiconductor manufacturing systems was proposed. A primary motivation for the conversion algorithm was to utilize the resulting calendar-time PM schedules as a base-line within the PM scheduling optimization algorithms given in Yao *et al.* (2001, 2002, 2004). While in (Ramírez-Hernández and Fernández-Gaucherand, 2003) the objectives were to present the mathematical model and details of the conversion algorithm, along with a simple numerical example for one-chamber tools and wafer-based PM schedules, in this paper we significantly expand the objectives as follows. First, we discuss in detail the reasons that justify the conversion of non-calendar based PM schedules. Second, this paper presents an improved version of the conversion algorithm proposed in Ramírez-Hernández and Fernández-Gaucherand (2003) by allowing the estimation of calendar-time targets that may occur at the beginning of the sampling periods. The previous version of the algorithm assumes that estimated calendar-time targets are reached before the current sampling period. This, however, yielded additional computations. Thus, in the current version of the algorithm such computations are avoided by checking if calendar-time targets were obtained at the beginning of the current sampling period. In addition, the mathematical model of the algorithm has been refined by including a dependence on the type of PM task for the cumulative amount of units completed (e.g. wafers) by each tool/chamber since the last PM task. Third, here we present a more general numerical example with fictitious data that considers multi-chamber tools within a mixed scenario of PM tasks including wafer, processing-time, and energy-based PM schedules. Fourth, this paper includes a case study on the accuracy of the conversion algorithm in which actual historical data from a real semiconductor fab were used as a comparison for estimated calendar-time targets for a wafer-based PM schedule. The results of this case study demonstrate that a good estimation of these targets is generated by the algorithm; therefore, the corresponding calendar-time based schedules can be efficiently incorporated into a PM scheduling optimization process. Finally, we also discuss how estimated calendar-time

PM schedules obtained through our conversion algorithm can be incorporated into a PM optimization algorithm, e.g. Yao *et al.* (2001, 2002, 2004).

The remainder of the paper is organized as follows: section II provides some background on both maintenance models and the PM scheduling optimization approach proposed in Yao *et al.* (2001, 2002, 2004) for semiconductor manufacturing systems. The same section also presents the rationale for the conversion algorithm. In section III, a refined version of the conversion algorithm given in Ramírez-Hernández and Fernández-Gaucherand (2003) is presented. A numerical example for the conversion of wafer, processing-time, and energy-based PM windows into calendar-time based PM windows with multi-chamber tools is presented in section IV, along with a case study of the accuracy of the algorithm considering industrial data from a real semiconductor fab. A brief discussion about how the estimated calendar-time PM schedules generated by the conversion algorithm are incorporated into the optimization algorithm of Yao *et al.* (2001, 2002, 2004) is presented in section V, and conclusions are given in section VI.

Background and rationale for the conversion algorithm

Background on maintenance models and PM scheduling optimization for semiconductor manufacturing systems

In general, different types of models have been proposed to address the problem of maintenance. A detailed review of the literature of maintenance models for multi-component (e.g. multiple tools) systems is presented in Dekker *et al.* (1997) and Wildeman (1996), where models are classified according to the interactions of the components in the system. This classification includes models with economic, structural, and stochastic dependence. Models with economic dependence are utilized with the objective of reducing costs derived from performing maintenance tasks in a joint form over several components. The modeling approach with structural dependence focuses on situations in which the maintenance of a failed component also implies the maintenance of other components in the system. The last classification, stochastic dependent models, considers the case where the state of a component influences the lifetime distribution of other components in the system. Similarly, models with economic dependence can be grouped into stationary or dynamic models. In the former, the rules for maintenance are static and are mainly utilized for long-term planning. The latter is utilized for short-term planning and the maintenance decisions can change over the planning horizon. Stationary models also include those of the corrective, preventive, and opportunistic type. For instance, dynamic models can be either of finite or rolling-horizon type. A detailed example of these models is provided in Wildeman *et al.* (1997). Finally, another modeling approach is that based on the condition of the components. Such approach, denominated condition-based modeling (Wang and Christer, 2000), deals with the problem of deciding when to perform maintenance based on the condition of a system subject to random deterioration.

Regarding the classification of the maintenance models given above, the approach in Yao *et al.* (2001, 2002, 2004) can be considered of the economic dependence type (e.g. aims consolidation of PM tasks to save costs), dynamic, and with a rolling-horizon. The optimization objectives are focused either on the maximization of availability (i.e. uptime) or throughput over a set of selected tools while minimizing inventory and maintenance costs. In Yao *et al.* (2001, 2002, 2004), the optimization problem is

represented by a two-hierarchical model with a Markov Decision Process (MDP) (Puterman, 1994; Bertsekas, 2000) at the higher level, and a Mixed Integer Programming (MIP) problem at the lower level. While the MDP defines the PM policies, e.g. baseline frequencies or schedules of PM tasks, the MIP utilizes such policies, along with operative information, such as estimated WIP levels and technicians availability per period, to optimize the schedules where PM tasks are suggested to be performed and over a finite horizon. The optimization process is then repeated for both a new planning horizon and operative information, so decisions change dynamically for each new planning horizon. The optimization objectives for the MIP models given in Yao *et al.* (2001, 2002, 2004) are as follows:

$$\text{MIP Objective 1 : } \max \sum_{t=1}^{T_p} \sum_{i=1}^M \left(b_i \cdot V_i(t) - c_i^I \cdot I_i(t) - \sum_{l=1}^{\rho_i} c_i^l \cdot a_i^l(t) \right), \quad (1)$$

$$\text{MIP Objective 2 : } \max \sum_{t=1}^{T_p} \sum_{i=1}^M \left(b'_i \cdot X_i(t) - \sum_{l=1}^{\rho_i} c_i^l \cdot a_i^l(t) \right), \quad (2)$$

where T_p is the number of time units or periods in the planning horizon, M is the number of tools (or tool chambers), $V_i(t)$ is availability of tool i in period t , b_i is the profit coefficient for availability of tool i , $I_i(t)$ is the workload level (i.e. WIP) for tool i in period t , c_i^I is the cost coefficient for inventory in tool i , ρ_i is the i number of PM tasks on tool i , $a_i^l(t)$ is a binary decision variable (1): do PM, 0: do not do PM) for PM task l on tool i in period t , and c_i^l is the cost of performing PM task l on tool i . In addition, in (2) $X_i(t)$ is the wafer throughput of tool i in period t and b'_i is the profit coefficient for throughput on tool i . The MIP objectives in (1) and (2) are also subject to different constraints on inventory levels, availability of resources (e.g. maintenance technicians per period), tools availability and throughput. On the one hand, the MIP Objective 1 aims to maximize the availability of tools while minimizing inventory and PM task costs. On the other hand, MIP Objective 2 deals with the maximization throughput in the tools while minimizing PM costs. The performance of the algorithms proposed in Yao *et al.* (2001, 2002, 2004) have been verified in simulation case studies by using industrial data in Yao *et al.* (2001, 2002, 2004); Crabtree *et al.* (2006). Results from these studies have shown that the algorithm can outperform the best PM scheduling methods used in practice, or at least consistently match the performance of human experts in charge of these operations.

Rationale for the conversion of non-calendar to calendar-time based PM schedules

In semiconductor manufacturing systems, the duration between PM tasks are usually described by “PM windows”, which are intervals of time (e.g. hours, shifts), wafers produced, or energy spent at each tool/chamber during the fabrication process. Each PM window is completely specified by, e.g. a warning, due, and late amount of units (e.g. hours, wafers produced, kilowatt-hours consumed). Thus, a PM schedule corresponds to a set of PM windows associated to tools and their applicable PM tasks. PM schedules are often implemented in a calendar time basis, and models/algorithms, as those in Yao *et al.* (2001, 2002, 2004), have been formulated to use input data in this same format, which is easy to understand and implement by operators and managers.

However, the semiconductor industry also makes use of other types of PM schedules, such as:

- *Wafer-based*. PM windows are specified in terms of the wafers produced at each tool.
- *Processing-time based*. The amount of time spent by each tool during the processing of material is utilized to the PM windows (e.g. processing time spent in the tool).
- *Energy-based*. The energy consumed by each tool during the processing of material is used to specify the PM windows (e.g. kilowatt-hours consumed).

An interesting idea is for non-calendar PM schedules, as those listed above, to be used to generate an equivalent calendar schedule by using a suitable conversion algorithm. Given a non-calendar based PM schedule, the conversion algorithm should take as input operational information (e.g. tool parameters, estimated WIP) to generate estimates of the PM windows in a calendar-time based format.

Motivated by the work in Yao *et al.* (2001, 2002, 2004), three main reasons justify the need to convert non-calendar based PM schedules into calendar-time format:

- (1) The dimensionality of the mixed integer programming (MIP) formulation of the optimal PM scheduling algorithm in Yao *et al.* (2001, 2002, 2004) is proportional to the main index in the decision variables, i.e. t in (1) and (2). Thus, if time units in the form of shifts or hours per day are considered, as opposed to, e.g. thousands of wafers per day, then a lower dimensional MIP is obtained.
- (2) The optimal PM scheduling algorithm (Yao *et al.*, 2001) needs an absolute scale of reference for all the PM windows to provide consolidation of PM tasks during the optimization process. This reference can only be provided if a calendar-time planning horizon is considered.
- (3) Calendar-time based PM schedules are preferred over non-calendar schedules for ease of use and implementation by technicians and tool managers in semiconductor fabs.

Justified by the previous rationale, the next section presents a refined and improved version of the conversion algorithm given in Ramírez-Hernández and Fernández-Gaucherand (2003).

Review of the conversion algorithm

In this section, the general conditions, variables, parameters, and the mathematical model of a refined version of the conversion algorithm in Ramírez-Hernández and Fernández-Gaucherand (2003) are presented. The version of the algorithm presented here has been improved by allowing the estimation of calendar-time targets that may occur at the beginning of sampling periods, thus avoiding additional computations for the calendar-time targets. This improvement is included in the third step of the algorithm. In addition, the mathematical model has been refined by including dependence on the PM task for the so-called cumulative amount of units completed for a given tool/chamber.

t_k Accumulated time after k time periods in the planning horizon, where:

$$t_{k+1} = t_k + T, \quad (3)$$

with $t_k \geq 0, t_k \in \mathbb{R}; k = 0, 1, 2, \dots, N; N \geq 0, N \in \mathbb{Z}, T > 0, T \in \mathbb{R}$. As mentioned earlier, the planning horizon is divided in N time slots, and T is the time period for the estimated WIP samples (see Figure 1).

$C_{t_k,l}^{ij}$ Cumulative amount of wafers completed, processing-time units spent, or energy units consumed by chamber j , tool i , at time t_k , since the last PM task l (see Figure 1). In practice, every time that PM task l is completed for tool i and chamber j , any count of units processed for the given tool, chamber, and PM task is reset to zero. In the conversion algorithm it is assumed that the time for the last completion of PM task l for tools i and chamber j is previous to t_0 . Thus, for any i, j, l we have that $C_{t_0,l}^{ij} \geq 0$ and $C_{t_k,l}^{ij} > 0$ for $k > 0$, with $C_{t_k,l}^{ij} \in \mathbb{R}$.

$W_l^{ij}, D_l^{ij}, L_l^{ij}$ PM window limits defining the warning, due, and late amount of units completed (targets, e.g. see Figure 1) in the tool-chamber j , tool i and PM task l , with $0 \leq W_l^{ij} \leq D_l^{ij} \leq L_l^{ij}$, and $W_l^{ij}, D_l^{ij}, L_l^{ij} \in \mathbb{R}$. The values of W_l^{ij}, D_l^{ij} , and L_l^{ij} are given in the same units as $C_{t_k,l}^{ij}$.

r_i, r_{ij} Average throughput rate parameters for tool i , and chamber j at tool i , respectively; with $r_i, r_{ij} > 0$, and $r_i, r_{ij} \in \mathbb{R}$. The value of these parameters depends on the type of units utilized by the estimated WIP.

α_r^{ij} Average throughput proportion corresponding to the chamber j , with respect to the throughput rate for tool i , with $\alpha_r^{ij} > 0, \alpha_r^{ij} \in \mathbb{R}$. For instance, when the WIP is given in time units, then:

$$r_{ij} = r_i + \alpha_r^{ij}. \quad (4)$$

See also Tables I, II, and III for additional parameterizations of r_{ij} according to the type of WIP units utilized.

$\theta_{t_k}^i$ Processing capacity for tool i in the time interval $[t_k, t_{k+1}]$ (e.g. hours, wafers), with $\theta_{t_k}^i \geq 0, \theta_{t_k}^i \in \mathbb{R}$.

$\theta_{t_k}^{ij}$ Processing capacity for chamber j at tool i in the time interval $[t_k, t_{k+1}]$; with $\theta_{t_k}^{ij} \geq 0, \theta_{t_k}^{ij} \in \mathbb{R}$.

Parameter	WIP: Time units	WIP: Wafer units
r_{ij}	$r_i \cdot \alpha_r^{ij}$	1
$\theta_{t_k}^{ij}$	$\theta_{t_k}^i \cdot \alpha_e^{ij}$	$\theta_{t_k}^i \cdot \alpha_e^{ij} \cdot r_i \cdot \alpha_r^{ij}$
γ_l^{ij}	$\frac{1}{r_{ij}}$	$\frac{1}{r_i \cdot \alpha_r^{ij}}$

Table I.
Parameterization
according to WIP units

- $\varepsilon_{t_k}^i$ Estimated WIP at tool i at the beginning of time t_k , with $\varepsilon_{t_k}^i \geq 0, \varepsilon_{t_k}^i \in \mathbb{R}$.
- $\varepsilon_{t_k}^{ij}$ Estimated WIP in chamber j , tool i , at the beginning of time t_k , with $\varepsilon_{t_k}^{ij} \geq 0, \varepsilon_{t_k}^{ij} \in \mathbb{R}$.
- $\Delta\varepsilon_{t_k}^{ij}$ Remaining WIP in queue to be processed in chamber j , tool i , by t_k , with $\Delta\varepsilon_{t_k}^{ij} \geq 0, \Delta\varepsilon_{t_k}^{ij} \in \mathbb{R}$, where

$$\Delta\varepsilon_{t_k}^{ij} = \begin{cases} 0 & \text{if } \varepsilon_{t_k}^{ij} + \Delta\varepsilon_{t_{k-1}}^{ij} \leq \theta_{t_k}^{ij}, \\ (\varepsilon_{t_k}^{ij} + \Delta\varepsilon_{t_{k-1}}^{ij}) - \theta_{t_k}^{ij} & \text{if } \varepsilon_{t_k}^{ij} + \Delta\varepsilon_{t_{k-1}}^{ij} > \theta_{t_k}^{ij}. \end{cases} \quad (5)$$

From (5), the remaining workload to be processed by t_k is zero if the current WIP, along with the remaining work from the previous period, does not exceed the uptime for the tool/chamber, i.e. the tool/chamber is able to process all the accumulated work during the period t_k . Otherwise, the excess of workload is retained to be processed during the next time periods.

- α_ε^{ij} WIP proportion corresponding to chamber j with respect to the WIP for tool i , with $\alpha_\varepsilon^{ij} \geq 0, \alpha_\varepsilon^{ij} \in \mathbb{R}$. For example, when time units are used for the WIP, we have that

$$\varepsilon_{t_k}^{ij} = \varepsilon_{t_k}^i \cdot \alpha_\varepsilon^{ij}. \quad (6)$$

See Tables I, II, and III for other parameterizations of $\varepsilon_{t_k}^{ij}$.

- t_W, t_D, t_L Times representing the estimated times for the warning, due, and late amount of units completed are reached respectively (see Figure 1), with $t_W \geq 0, t_D \geq 0, t_L \geq 0$, and $t_W, t_D, t_L \in \mathbb{R}$.

Table II.
Parameterization for conversion from processing-time to calendar-time based PMs

Parameter	Parameterization
r_{ij}	1
$\theta_{t_k}^{ij}$	$\theta_{t_k}^i \cdot \alpha_\varepsilon^{ij} \cdot \phi_l^{ij}$
γ_l^{ij}	$\frac{T}{C_{t_{(\cdot)+1}^j}^{ij} - C_{t_{(\cdot)}^j}^{ij}}$
α_ε^{ij}	$\varepsilon_{t_k}^i \cdot \alpha_\varepsilon^{ij} \cdot \phi_l^{ij}$

Table III.
Parameterization according to WIP units, conversion from energy-based to calendar-time based PMs

Parameter	Time units	Wafer units
r_{ij}	Chamber power	Chamber power
$\theta_{t_k}^{ij}$	$\theta_{t_k}^i \cdot \alpha_\varepsilon^{ij}$	$\theta_{t_k}^i \cdot \alpha_\varepsilon^{ij}$
γ_l^{ij}	$\frac{1}{r_{ij}}$	$\frac{1}{r_{ij}}$
$\varepsilon_{t_k}^{ij}$	$\varepsilon_{t_k}^i \cdot \alpha_\varepsilon^{ij}$	$\varepsilon_{t_k}^i \cdot \alpha_\varepsilon^{ij} \cdot \phi_l^{ij}$

$t_{\bar{W}}, t_{\bar{D}}, t_{\bar{L}}$	Largest time t_k where the cumulative amount of units completed at that time, $C_{t_k,l}^{ij}$, is less or equal than the target amount of units for the warning, due, and late amount of units completed, respectively (without going over these targets, e.g. see Figure 1); where $t_{\bar{W}} \geq 0, t_{\bar{D}} \geq 0, t_{\bar{L}} \geq 0$, and $t_{\bar{W}}, t_{\bar{D}}, t_{\bar{L}} \in \mathbb{R}$.
$\Delta W_l^{ij}, \Delta D_l^{ij}, \Delta L_l^{ij}$	Differences between the targets $W_l^{ij}, D_l^{ij}, L_l^{ij}$ and the units counts $C_{t_{\bar{W}},l}^{ij}, C_{t_{\bar{D}},l}^{ij}$, and $C_{t_{\bar{L}},l}^{ij}$, respectively (see Figure 1), with $\Delta W_l^{ij} \geq 0, \Delta D_l^{ij} \geq 0, \Delta L_l^{ij} \geq 0$, and $\Delta W_l^{ij}, \Delta D_l^{ij}, \Delta L_l^{ij} \in \mathbb{R}$.
γ_l^{ij}	Conversion factor utilized to estimate the amount of time required to complete the processing of the amount of units indicated by $\Delta W_l^{ij}, \Delta D_l^{ij}$, and ΔL_l^{ij} ; with $\gamma_l^{ij} > 0$ and $\gamma_l^{ij} \in \mathbb{R}$.
ϕ_l^{ij}	Conversion factor utilized to maintain consistency in the WIP units when processing-time and energy-based PM tasks are considered.

The times t_W, t_D , and t_L represent the main outcomes of the conversion algorithm that later can be easily presented in a calendar-time format. For instance, if the resulting estimated time t_W is given in hours elapsed from the initial time t_0 , and assuming that the date and time at that point is known, then a corresponding representation in date and time can be directly obtained for t_W .

From the list of parameters and variables listed above, the values of $r_{ij}, r_i, \theta_{t_k}^{ij}, \varepsilon_{t_k}^{ij}$, and γ_l^{ij} need to be properly parameterized according to type of units utilized by the estimated WIP (i.e. time or wafers units) and the type of PM task (i.e. wafer, processing-time, or energy-based). This is necessary to maintain consistency in the units being utilized in the parameters and variables of the algorithm. From Ramírez-Hernández and Fernández-Gaucherand, 2003, the specific parameter values are given in Tables I, II, and III:

In addition, the corresponding wafers differences between the targets $W_l^{ij}, D_l^{ij}, L_l^{ij}$ and the actual units completed count $C_{t_k,l}^{ij}$ (e.g. see Figure 1), are given as follows:

$$\Delta(\cdot)_l^{ij} = (\cdot)_l^{ij} - C_{t_{(\cdot)},l}^{ij}, \quad (7)$$

where the term (\cdot) corresponds to W, D or L .

Conversion algorithm

The following is an algorithm that can be utilized to estimate the times t_W, t_D , and t_L that define the PM window for PM task l applied to tool i and chamber j :

Algorithm. Let the cluster tool i , chamber j , PM task l , $\Delta\varepsilon_{t_{k-1}}^{ij}$ the remaining incoming work accumulated in a finite capacity queue at time t_{k-1} , and let $C_{t_0,l}^{ij} \leq W_l^{ij}$, $C_{t_0,l}^{ij} \leq D_l^{ij}$, and $C_{t_0,l}^{ij} \leq L_l^{ij}$ for any i, j, l . Thus, the conversion algorithm follows the next steps:

(i) if $(\varepsilon_{t_k}^{ij} + \Delta\varepsilon_{t_{k-1}}^{ij}) \leq \theta_{t_k}^{ij}$ then:

$$C_{t_{k+1},l}^{ij} = C_{t_k,l}^{ij} + r_{ij} \cdot (\varepsilon_{t_k}^{ij} + \Delta\varepsilon_{t_{k-1}}^{ij}); \quad (8)$$

$$\Delta \varepsilon_{t_k}^{ij} = 0; \text{ else :} \tag{9}$$

(ii) if $(\varepsilon_{t_k}^{ij} + \Delta \varepsilon_{t_{k-1}}^{ij}) > \theta_{t_k}^{ij}$, then :

$$C_{t_{k+1},l}^{ij} = C_{t_k,l}^{ij} + r_{ij} \cdot \theta_{t_k}^{ij}; \tag{10}$$

$$\Delta \varepsilon_{t_k}^{ij} = (\varepsilon_{t_k}^{ij} + \Delta \varepsilon_{t_{k-1}}^{ij}) - \theta_{t_k}^{ij}; \tag{11}$$

(iii) if $W_l^{ij} - C_{t_k,l}^{ij} \leq 0$, or $D_l^{ij} - C_{t_k,l}^{ij} \leq 0$, or $L_l^{ij} - C_{t_k,l}^{ij} \leq 0$, then the target W_l^{ij} , D_l^{ij} , or L_l^{ij} has been reached, respectively. For each true condition, if the corresponding target time has not been calculated from some previous time period, then it is computed as follows:

if $\Delta(\cdot)_l^{ij} = 0$, then $t_{(\cdot)} = t_k$, else:

$$t_{(\cdot)} = t_{k-1} \Rightarrow C_{t_{(\cdot)},l}^{ij} = C_{t_{k-1},l}^{ij}; \tag{12}$$

$$t_{(\cdot)} = t_{(\cdot)} + \Delta(\cdot)_l^{ij} \cdot \gamma_l^{ij}; \tag{13}$$

(iv) if $t_k = N \cdot T$ (i.e. the final time period in the planning horizon has been evaluated, then finish), if not, set $t_{k+1} = t_k + T$ and return to (i).

In (12) and (13) the term (\cdot) corresponds to either W , D , or L . The computation of the warning, due, or late time is given in (13). This time can be easily converted to a corresponding calendar-date that can finally be used within the algorithm for optimal PM scheduling.

The conversion algorithm can be applied sequentially over a list of cluster tools, their chambers, and the associated PM tasks. For each tool i , chamber j , and PM task l , the algorithm utilizes the corresponding WIP profile to compute the estimated number of units completed by time t_k , $C_{t_k,l}^{ij}$, until one or more of the targets W_l^{ij} , D_l^{ij} , or L_l^{ij} are reached. The time t_k is increased according to $t_{k+1} = t_k + T$. At each time t_k the amount $C_{t_k,l}^{ij}$ is compared against the targets W_l^{ij} , D_l^{ij} , and L_l^{ij} ; if the current count $C_{t_k,l}^{ij}$ is greater or equal to the target, then the counting process stops and the corresponding estimation of the target time is generated by counting the accumulated time from t_0 to the time where the target was reached. Once the planning horizon has been covered, i.e. $t_k = t_N$, the algorithm can be applied to a new tool i , chamber j , and PM task l .

An important observation is that precision in the estimation of the targets will be dependent on the accuracy of the projections for the incoming WIP at each tool. One of the key characteristics of the algorithm is that it will utilize the estimated WIP to in turn estimate the number of units completed by each tool and chamber (see (8)-(11)). Therefore, a good estimation of the WIP, together with correctly defined tool parameters, will yield more accurate estimates for the targets. The quality in the estimation of the WIP will depend on the software tools and methods (or practices) that each semiconductor fabrication facility utilizes for this purpose.

Numerical example and case study

The application and accuracy of the conversion algorithm is illustrated in this section by presenting both a numerical example with fictitious data and a case study with industrial data from a real semiconductor fab, respectively.

Numerical example: mixed PM types and multi-chamber tools

In this subsection, a numerical example is presented for the conversion of PM windows into calendar-time equivalents for three different types of PM tasks: wafer-based, processing-time based, and energy-based. A fictitious group of two tools and three PM tasks is considered, along with fabricated data chosen carefully for illustration purposes. The algorithm was implemented in C-language. The following are the conditions for the numerical example:

- (1) There are two tools: Tool 1, Tool 2 ($i = 1, 2$).
- (2) There were three PM tasks: PM 1 (processing-time based), PM 2 (wafer-based), and PM 3 (energy-based).
- (3) The PM windows baseline targets (warning, due, and late), and the units utilized are as follows:
 - PM 1: 450, 500, 550 (hours of processing time);
 - PM 2: 14400, 16000, 17600 (wafers produced); and
 - PM 3: 270, 300, 330 (kilowatt-hours).
- (4) Tool 1 has one chamber and $i = 1$ for this tool. Thus, $M_1 = 1, j = 1, \alpha_e^{ij} = 1$, and $\alpha_r^{ij} = 1$.
- (5) Tool 2 has two chambers in parallel: CH1 and CH2 (see Figure 2). For this tool $i = 2$, and $M_2 = 2, j = 1, 2$. In addition, $\alpha_e^{21} = \alpha_e^{22} = 0.5$ and $\alpha_r^{21} = \alpha_r^{22} = 0.5$; that is, each chamber receives a 50 percent of the total WIP and throughput rate of Tool 2.
- (6) A semiconductor fab with an operation of 24 hours, separated in 2 shifts, is considered.
- (7) The time unit utilized is hours.
- (8) The WIP data is given in hours of processing time (i.e. number of hours that the tool is expected to be utilized for processing). The sample period is $T = 12$ hours. Thus, $t_{k+1} = t_k + 12$ hours, with $k = 0, 1, \dots, 16$; and the planning horizon starts at t_0 .

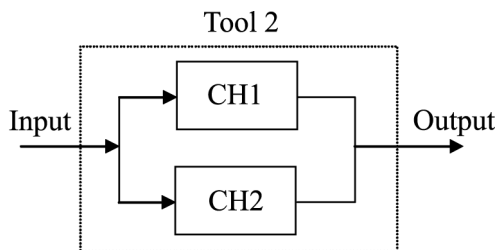


Figure 2.
Tool 2 with two chambers
in parallel

- (9) We consider a planning horizon of 8 days (192 hours divided in $N = 16$ time slots). This horizon begins at day 1 (t_0), 07:00:00 (hours: minutes: seconds), and finishes at day 8 (t_{16}), 07:00:00.
- (10) The throughput rates at Tools 1 and 2 are as follows: $r_1 = 22$ wafers/hour, and $r_2 = 40$ wafers/hour. In tool 1, given that there is only one chamber, then $r_{11} = r_1$. For Tool 2, since $\alpha_r^{21} = \alpha_r^{22} = 0.5$, we have that $r_{21} = r_{22} = \alpha_r^{21} \cdot r_2 = 20$ wafers/hour.
- (11) We consider that tools are available 87.5 percent of the time, then at each time period tk the tool availability is defined as: $\theta_{t_k}^i = 10.5$ hours, for all i, t_k . Thus, $\theta_{t_k}^1 = 10.5$ hours for Tool 1. For Tool 2, and by using the parameterizations indicated in Table I when the estimated WIP is given in time units, $\theta_{t_k}^{ij} = \theta_{t_k}^i \cdot \alpha_e^{ij}$, therefore $\theta_{t_k}^{21} = \theta_{t_k}^{22} = 5.25$ hours, for all t_k .
- (12) The power consumption in Tool 1 is 2 kilowatts, and each chamber at Tool 2 consumes 3 kilowatts.
- (13) The initial count of units processed at t_0 , $C_{t_0,d}^{ij}$, is given for each pair (Tool, PM task) in Table IV.

These (fictitiously) “estimated” incoming WIP in each tool is given at time period t_k in the planning horizon. Figure 3 shows the WIP profile for Tools 1 and 2.

Using the data previously presented, and applying the conversion algorithm, the resulting equivalent calendar-time based PM windows are shown in Figure 4. For each pair (Tool, PM task) there is a time window delimited by the corresponding targets: warning (W), due (D) and late (L) dates. The estimated target date is indicated in calendar form, including day (between 1 and 8) and time (in 24 hours format without seconds).

As depicted in Figure 4, some of the calendar-time based PM windows are presented as partial views of the complete PM window, because some of the targets are not included in the planning horizon (e.g. Tool 2-CH2 and PM 2 only presents a Due date estimation).

Case study: accuracy of an estimated calendar-time PM schedule in a real semiconductor fab

This subsection presents details about the accuracy of the conversion algorithm to estimate the wafer targets (e.g. due dates) for several wafer-based PM windows. The precision of the estimated calendar-time schedule was verified by using historical data that registered the amount of wafers completed per tool at the estimated target dates (i.e. obtained with the conversion algorithm) in a real semiconductor fab. The accuracy was measured in terms of the error percentage between the real wafer count by the

Table IV.
Initial units count $C_{t_0,d}^{ij}$
for each pair
tool-(chamber), PM task

Tool-chamber/PM	PM 1	PM 2	PM 3
Tool 1	470	15,500	150
Tool 2-CH1	460	15,900	220
Tool 2-CH2	450	14,500	220
PM 1: processing-time based (hours); PM 2: wafer-based (wafers); PM 3: energy-based (kilowatt-hours)			

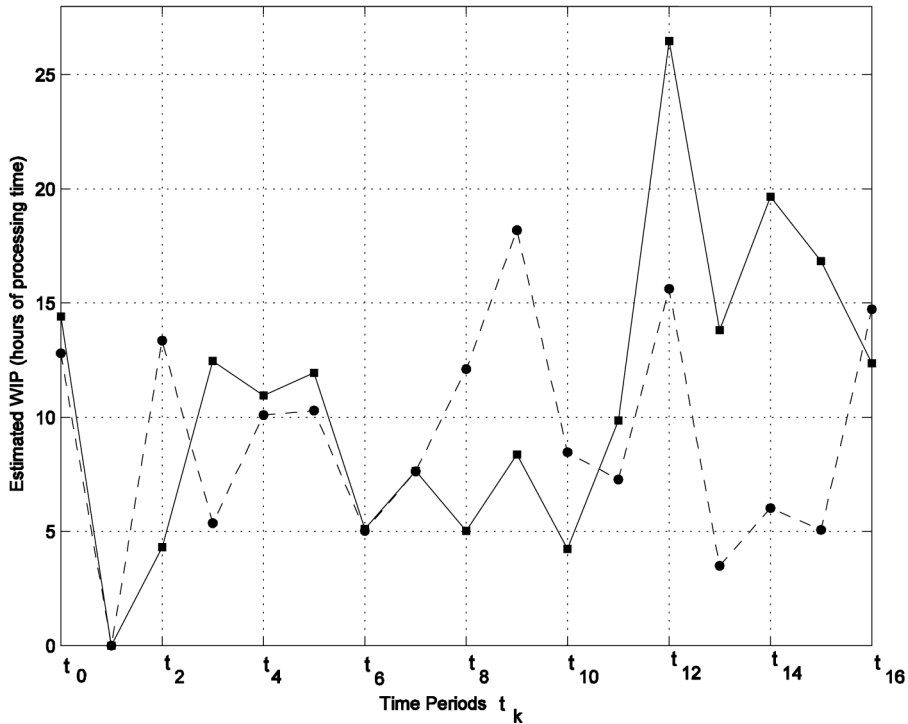


Figure 3.
WIP profile for Tool 1
(squares, solid line) and
Tool 2 (dots, dashed line)

estimated target date and the target amount of wafers (i.e. warning, due, and late date) used to obtain the estimated target dates. This measure is denominated Target Estimation Error (TEE) and it is defined as follows:

$$TEE = \frac{\text{Target} - \text{RealWaferCount}}{\text{Target}} \times 100$$

Table V lists the results obtained for a group of five tools MDep1, ..., MDep5; and 3 different PM tasks PMW1, PMW2, and PMW3. For instance, the estimation error of the due date for the PM task PMW1 at tool MDep1 is 0.28 percent.

In general, the results from Table V demonstrate that the conversion algorithm provided satisfactory estimates of the target dates. Nonetheless, in the case of "MDep5, PMW3" the estimation errors were between 8.03 percent and 10.31 percent. This could be attributed to the quality in the projection of the incoming WIP for MDep5 by the end of the planning horizon. In this case, historical data from the fab indicates that at the beginning of the planning horizon tool MDep5 produced around 13 percent of the due amount of wafers. Consequently, the estimated warning and due date produced by the conversion algorithm were allocated near to the end of the planning horizon, for which WIP estimates may result more uncertain. As it was mentioned in section II, the precision of the conversion algorithm relies on the quality of the WIP estimations. Thus, the use of efficient and reliable methods to estimate WIP levels is then an important component in the accuracy of the proposed conversion algorithm. For

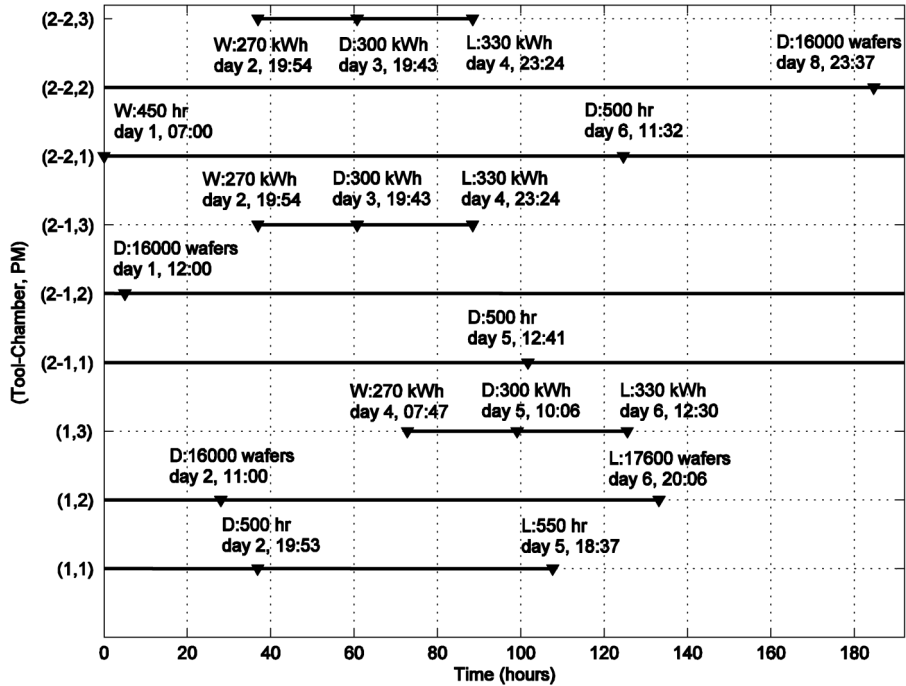


Figure 4. Estimated calendar-time based PM windows for the numerical example

Tool, PM task	Target estimation error (TEE)	
	Warning	Due
MDep1, PMW1	-	0.28
MDep1, PMW2	-	-0.06
MDep2, PMW3	-	-1.33
MDep2, PMW1	-	-3.70
MDep3, PMW3	1.06	≈ 0.00
MDep3, PMW1	-	1.48
MDep4, PMW3	3.14	-0.06
MDep5, PMW3	8.03	10.31

Table V. Estimation error of wafer targets using the conversion algorithm

instance, see Lin and Lee (2001) for details regarding research on efficient estimation of WIP levels in semiconductor manufacturing systems.

It can also be noted from Table V, that all the pairs (Tool, PM task) have a due date in the planning horizon, but not all have a warning, and none a late date. The error for these targets is not specified because these are beyond or before the considered planning horizon. The initial PM schedule for this case study was of special interest due to the number of overlapped PM windows which represented potential candidates for consolidation[1] of PM tasks (Yao *et al.*, 2001; 2002, 2004), and also because of the challenging task that is involved in the optimal scheduling process. Finally, the calendar-time based PM schedule derived from this example has been utilized in a simulation case study presented in Crabtree *et al.* (2006) where the improvement in

production performance was compared by utilizing either a baseline or an optimized PM schedule (see (Crabtree *et al.*, 2006) for details). In this study the converted PM scheduled (i.e. calendar-time based) was used as a baseline schedule that is optimized with the optimization PM scheduling algorithm discussed in Yao *et al.* (2001, 2002, 2004), and implemented in PMOST. The results of this study demonstrated improvement in production performance by applying the optimized PM schedule.

Incorporating estimated calendar-time pm schedules into the PM optimization algorithm

Estimated calendar-time PM schedules generated by the conversion algorithm can be utilized as initial or baseline schedules in the PM scheduling optimization algorithm given in Yao *et al.* (2001, 2002, 2004), which has been implemented in the *Preventive Maintenance Optimization Software Tool* (PMOST) (Crabtree *et al.*, 2006; Crabtree, 2003). Both the optimization algorithm and PMOST are able to handle PM tasks scheduled in either a day or shift basis; that is, PM tasks are assumed to start at the beginning of the corresponding period (e.g. shift). Shift-based PM scheduling, on the other hand, is generally preferred in semiconductor fabs because it provides technicians and tool managers with a clear idea of when PM tasks need to be performed (Crabtree, 2003). Furthermore, this approach allows for the collection of more accurate data for the PM optimization process. The calendar-time PM schedules obtained by the conversion algorithm provide information in terms of time (in hours, minutes, and seconds) elapsed since the beginning of the planning horizon, e.g. day and time. Currently, PMOST version G2.0 (Crabtree, 2003) takes this information and reassigns the schedule of events to the beginning of the time period in which such events fall, as per the output of the conversion algorithm.

As an example, Figure 5 shows how the current version of PMOST allocates the estimates of the due dates for two non-calendar based PM tasks, PM1 and PM2. In this example PMOST interprets that PM1 is due on day 1, first shift (7:00 am), and PM2 on day 2, second shift (7:00 pm).

Initially, it seems that the resolution of the estimates generated by the conversion algorithm, given in day, hours, and minutes, is more than what it is required by

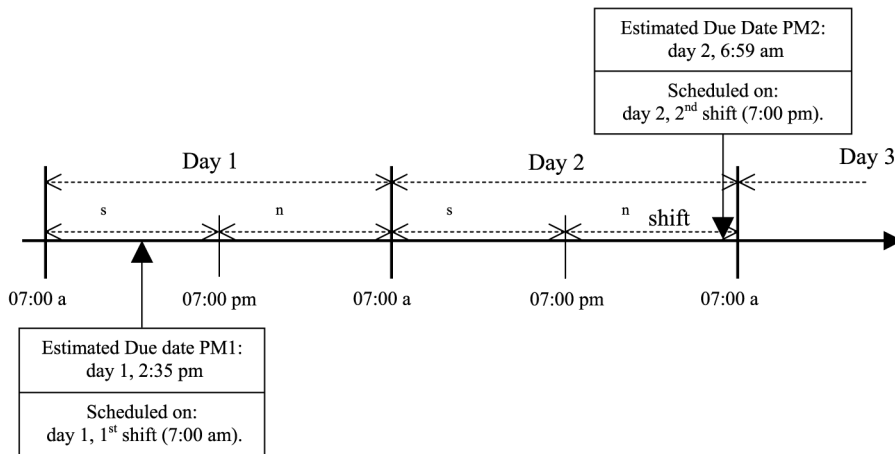


Figure 5.
Allocation of estimated
due dates for two PM
tasks in a shift basis
scheduling

PMOST. That is, it is only necessary to determine to which day and shift an estimated target (e.g. due date) is allocated. However, in situations similar to those presented in Figure 5, where the due date for PM2 is estimated to occur at the end of day 2 but scheduled on the beginning of the second shift, the resolution provided by the conversion algorithm allows the possibility to decide, when convenient, to allocate the due date of PM2 at the beginning of day 3 (7.00 a.m.). These decisions could be easily implemented by considering that a target estimate will be allocated in the next period (e.g. shift) if it is beyond a threshold point in the current shift.

Conclusions

This paper presents the rationale, a numerical example, and a case study on the application of an algorithm for converting non-calendar based PM schedules into calendar format for semiconductor manufacturing systems. The numerical example illustrates how the proposed algorithm is utilized in the conversion of three different types of non-calendar time PM schedules where multi-chamber tools were considered. In addition, it is discussed how estimated calendar-time based PM schedules can be incorporated into the PM optimization process and tools developed in Yao *et al.* (2001, 2002, 2004), and (Crabtree, 2003; Crabtree *et al.*, 2006). Finally, a case study on the accuracy of the conversion algorithm is presented, where estimated PM targets are compared against operational data from a real semiconductor fab. The results show that, in general, the algorithm performs quite satisfactory. However, important consideration must be provided to the uncertainty of WIP level estimates in order to preserve accuracy in the estimations of the PM targets for the resulting calendar-time PM schedules.

The work presented here and in Yao *et al.* (2001, 2002, 2004) is open for further research. Two interesting alternatives to continue this line of research are provided by approaches such as opportunistic maintenance (Dekker *et al.*, 1997; Wildeman, 1996; Wildeman *et al.*, 1997) and condition-based maintenance (Wang and Christer, 2000). At difference of the proposed models and algorithms given here and in Yao *et al.* (2001, 2002, 2004), opportunistic maintenance models may be considered to take advantage of failure events to also schedule PM tasks and improve the overall PM scheduling optimization. Similarly, condition-based maintenance models can improve the PM optimization approach by allowing non-deterministic times for inspection, repair, or maintenance of equipment.

Note

1. Consolidation of PM tasks has proven (Yao *et al.*, 2001; 2002; 2004) to be a very useful characteristic in the optimization of PM schedules to reduce the costs associated with PM operations.

References

- Bertsekas, D.P. (2000), *Dynamic Programming and Optimal Control*, 2nd. ed., Vol. II, Athena Scientific, Belmont, MA.
- Blau, J. (2003), "News analysis: Europe's semiconductor makers are back in the game", *IEEE Spectrum Magazine*, pp. 18-19.
- Crabtree, J. (2003), "Optimal preventive maintenance scheduling in semiconductor fabs", Master's thesis (Electrical Engineering), University of Cincinnati, Cincinnati, OH.

-
- Crabtree, J., Ramírez-Hernández, J.A., Yao, X., Fernandez, E., Fu, M.C., Janakiram, M., Marcus, S.I., O'Connor, M. and Patel, N. (2006), "Optimal preventive maintenance scheduling in semiconductor manufacturing systems: software tool and simulation case studies" (submitted for publication), available at: www.ececs.uc.edu/~ramirejs/OptimalPM2006.pdf
- Dekker, R., Wildeman, R.E. and Schouten, F.A.V.D. (1997), "A review of multi-component maintenance models with economic dependence", *Mathematical Methods of Operations Research*, Vol. 45 No. 3, pp. 411-35.
- Lin, Y.-H. and Lee, Ch.-E. (2001), "A total standard WIP estimation method for wafer fabrication", *European Journal of Operations Research*, Vol. 131, pp. 78-94.
- Puterman, M.L. (1994), *Markov Decision Processes*, Wiley, New York, NY.
- Ramírez-Hernández, J.A. and Fernández-Gaucherand, E. (2003), "An algorithm to convert wafer to calendar-based preventive maintenance schedules for semiconductor manufacturing systems", *Proceedings of the 42nd IEEE Conference on Decision and Control, Maui, December 9-12*, pp. 5926-31.
- SIA (2006), *Semiconductor Industry Association Annual Report 2004*, available at: www.sia-online.org/pre_annual.cfm
- Wang, W. and Christer, A.H. (2000), "Towards a general condition based maintenance model for a stochastic dynamic system", *Journal of the Operational Research Society*, Vol. 51 No. 2, pp. 145-55.
- Wildeman, R.E. (1996), "The art of grouping maintenance", PhD thesis, Erasmus University, Rotterdam.
- Wildeman, R.E., Dekker, R. and Smit, A. (1997), "A dynamic policy for grouping maintenance activities", *European Journal of Operational Research*, Vol. 99 No. 3, pp. 530-51.
- Yao, X., Fernandez-Gaucherand, E., Fu, M. and Marcus, S.I. (2004), "Optimal preventive maintenance scheduling in semiconductor manufacturing", *IEEE Transactions on Semiconductor Manufacturing*, Vol. 17 No. 23, pp. 345-56.
- Yao, X., Fu, M., Marcus, S.I. and Fernandez-Gaucherand, E. (2001), "Optimization of preventive maintenance scheduling for semiconductor manufacturing systems: models and implementation", *Proceedings of the 2001 IEEE International Conference on Control Applications, Mexico City, September 5-7*, pp. 407-11.
- Yao, X., Fu, M., Marcus, S.I. and Fernandez-Gaucherand, E. (2002), "Incorporating production planning into preventive maintenance scheduling in semiconductor fabs", *Proceedings International Conference on Modeling and Analysis of Semiconductor Manufacturing (MASM) 2002, Tempe, AZ, April 10-12*, pp. 84-9.

Corresponding author

Emmanuel Fernandez can be contacted at: emmanuel@ececs.uc.edu