Petri Net Models for Analysis and Control of Re-Entrant Flow Semiconductor Wafer Fabrication

Jonathan D. Green
Nicholas G. Odrey
Lehigh University

Report No. 98W-006

PETRI NET MODELS FOR ANALYSIS AND CONTROL OF RE-ENTRANT FLOW SEMICONDUCTOR WAFER FABRICATION

Jonathan D. Green[1] and Nicholas G. Odrey[2]
Department of Industrial and Manufacturing Systems Engineering, Lehigh University, Bethlehem, PA 18015

Abstract

Re-entrant flow manufacturing lines, such as occur in semiconductor wafer fabrication, are characterized by a product routing that consists of multiple visits to a workstation or group of workstations during the manufacturing process. In such re-entrant lines, work at different stages of the process compete for the same piece(s) of equipment, resulting in a more complex scheduling problem than for that of a flow line.

Here, a re-entrant flow manufacturing line is modeled using generalized Petri nets. Petri nets provide a graphical and mathematical representation of a discrete event system that allows for analysis of the system's behavioral and structural properties. Two advantages of a Petri net modeling approach are the ability to model both conflict among resources and the multiple decisions that must be made in scheduling. Three Petri net models representing a re-entrant flow line with three work centers and six machines are presented. Discussed is how these models may be used to represent a variety of queueing disciplines and work release policies. These models also may be used to analyze performance measures such as cycle time and work in process, and as part of a real-time shop floor control system.

Keywords: Problem Type - Production Planning & Control, Re-Entrant Flow, Real Time Flow Control, Scheduling. Measurement Metric - No Specific. Modeling Technique - Simulation. Organizational Decision Level - Factory

(1) Graduate student, (2) Professor

1. INTRODUCTION

A re-entrant flow manufacturing line, typified by semiconductor wafer fabrication, can be modeled by generalized stochastic Petri nets (GSPN). Various authors have considered scheduling policies in re-entrant manufacturing systems, but typically have used queueing theory and simulation. Petri nets give some advantages over such methods. As opposed to queueing network models, Petri nets have the advantage of being able to model nonproduct-form features such as blocking, synchronization, and most importantly here, priority queueing disciplines. In addition, Petri nets can be exercised to simulate the real system with the advantage of providing a mathematical representation of the system so that structural and behavioral properties may be analyzed. Finally, Petri nets may be used as real-time controllers.

Petri nets have been suggested by others as a means to model the semiconductor manufacturing process. For example, Cavalieri, Mirabella, and Marroccia [2] described a colored Petri net approach to scheduling a Flexible Semiconductor Manufacturing System. They employed a heuristic algorithm to search the reachability tree of the Petri net to find a pre-specified low cost scheduling solution. Kim and Desrochers [8] introduced a method for automatically structuring a Petri net model for semiconductor manufacturing and discussed how such measures as capacity, on-line turn around time, and work in process can be measured. A primary purpose here is to provide greater detail on how to model scheduling and work release policies in re-entrant flow systems. Also provided is a review of literature on performance analysis of semiconductor manufacturing systems, and various models depicting different sequencing and work release policies are presented and described.

The paper is organized as follows: Section 2 provides a short description of Petri nets, while section 3 provides a description of re-entrant lines and the semiconductor wafer fabrication process. Relevant literature regarding re-entrant lines and scheduling and performance analysis in semiconductor wafer fabrication is contained in section 4. In section 5, Petri nets representing different dispatching policies for a re-entrant line considered here are developed. Also discussed is how these models may be used as part of a shop floor control system. Conclusions and recommendations for further work are discussed in section 6.

2. PETRI NETS

Petri nets [5] are a graphical mathematical and modeling tool that can be used to represent many types of dynamic systems. A *Petri net* is a bipartite directed graph represented by a five-tuple (P,T,I,O,H), where P is a finite set of *places*, T is a finite set of *transitions*, I is an input mapping corresponding to the set of *directed arcs* from P to T (*input arcs*), O is an output mapping corresponding to the set of directed arcs from T to P (*output arcs*), and H is a set of *inhibitor arcs*. Graphically, places are represented by circles and transitions by lines or bars.

A *marked Petri net* contains *tokens* (represented by dots) which reside in places and travel along arcs, while the *marking* of a Petri net is a mapping that assigns a nonnegative finite integer number of tokens to each place. The initial marking of a Petri net is designated marking m_0. The movement of tokens through the net is regulated by *firings* of transitions. A transition is said to be *enabled* when each one of its input places contains at least one token. When a transition fires, a token is removed from each input place, and a token is placed in each output place. Arcs can be given a *weight* of more than one, in which

case more than one token (the weight of the arc) must reside in an input place to enable the corresponding transition. If a weight is associated with an output arc, more than one token will be placed in the corresponding output place. Inhibitor arcs are attached from a place to a transition, and can be recognized by a small circle on the end of the arc adjacent to the transition. The existence of a token in a place with an inhibitor arc restricts the firing of the associated transition. A weight can be associated with an inhibitor arc, in which case more than one token must reside in the place to inhibit the transition firing. A typical example for the use of an inhibitor arc would be in a finite buffer. No tokens flow along inhibitor arcs.

Places represent *resources*. The existence of one or more tokens in a place represents the availability of the resource. A transition firing represents an *activity*. The time of the activity can be zero in the case of an *immediate transition* or greater than zero, in which case the ordinary Petri net becomes a *timed Petri net*. Deterministic *timed transitions* are represented by a thick solid bar, while *immediate transitions* are represented by a thin bar. If the time associated with a transition is random, the net is called a *stochastic timed Petri net* and the transition is represented by a thick unfilled rectangle. In this paper, *generalized stochastic Petri nets*, for which the transition times may be immediate, constant, or stochastic, are used. A timed transition may be thought of as having a start and end time to its firing. The transition will start firing as soon as it is enabled, with the token remaining in the associated place until the end time. This distinction is important when modeling machine breakdowns, which is discussed in section 5.

The mathematical representation of a Petri net allows one to analyze important behavioral and structural properties of the net. Properties of Petri nets that are dependent on the initial marking are referred to as *behavioral properties*. Those that are independent

of the initial marking are called *structural properties*. Some important behavioral properties are reachability, boundedness, liveness, and reversibility. A marking m_r is said to be *reachable* from marking m_0 if there exists a firing sequence that will yield m_r. The reachability set is the set of all markings reachable from m_0 and is designated by $R(m_0)$. A Petri net is said to be *k-bounded* with respect to an initial marking m_0 if each place in the net gets at most k tokens for all markings in the reachability set $R(m_0)$. A Petri net is *live* with respect to a marking m_0 if, for any marking in the reachability set $R(m_0)$, it is possible to ultimately fire any transition in the net. Liveness, a property that is often difficult to verify, guarantees the absence of *deadlocks*, which occur when a marking is reached where the firing of one or more transitions is no longer possible. A Petri net is *reversible* if the initial marking m_0 is reachable from every marking in the reachability set $R(m_0)$. Reversibility implies that the model can reinitialize itself. This is important for the automatic recovery from errors and failures, since it guarantees that the system will, in a finite number of steps, return to an admissible state. Properties of liveness, boundedness, and reversibility are independent.

The definitions of three structural properties, those properties of a Petri net independent of the initial marking m_0, will be mentioned here. A Petri net is *structurally bounded* if it is bounded for any finite initial marking. A Petri net is *structurally live* if it is live for any finite initial marking. Finally, a Petri net is *completely controllable* if any marking is reachable from any initial marking. In Petri net applications, complete controllability is not normally satisfied. For a detailed yet succinct tutorial on Petri nets, their properties, and applications, see Murata [12]. For a comprehensive text on Petri nets, see Desrochers and Al-Jaar [5].

3. RE-ENTRANT FLOW LINES AND SEMICONDUCTOR MANUFACTURING

A re-entrant line is one in which a product visits the same pieces of equipment multiple times throughout the manufacturing process. The example here is in semiconductor manufacturing, in which multiple layers of miniature circuitry are built up on a wafer of silicon or another semiconductor material. Wafers are grouped into lots, typically of 20 to 50 wafers each, and processed through the manufacturing line. For each layer, a sequence of similar processes must be undertaken repeatedly, often on the same pieces of equipment. This results in competition by wafers at different points in the manufacturing process for the same piece of equipment, resulting in decisions on which lot of wafers to process next. The machines performing these processes are very expensive, and therefore it is imperative to use scheduling and work release policies that maximize system performance. The most common performance measure in semiconductor manufacturing is throughput time (also called cycle time), or the time from when a lot enters the line to the time it leaves. The most expensive pieces of equipment in the wafer fab (the common name for the room where the processes are undertaken) are the photolithography machines, which define photoresist patterns for electronic circuitry on the wafer. Because of the expense of this and other wafer fabrication equipment, high utilization of equipment, especially of photolithography machines, is another typical performance goal.

Figure 1 represents a full-scale re-entrant line considered by Lu, Ramaswamy, and Kumar [11]. Due to the size of their model, we will consider a smaller model (Figure 2) of a re-entrant line, previously studied by Narahari and Khan [13]. We assume that only one product is produced in the line. The assumption of one product is very real for many wafer

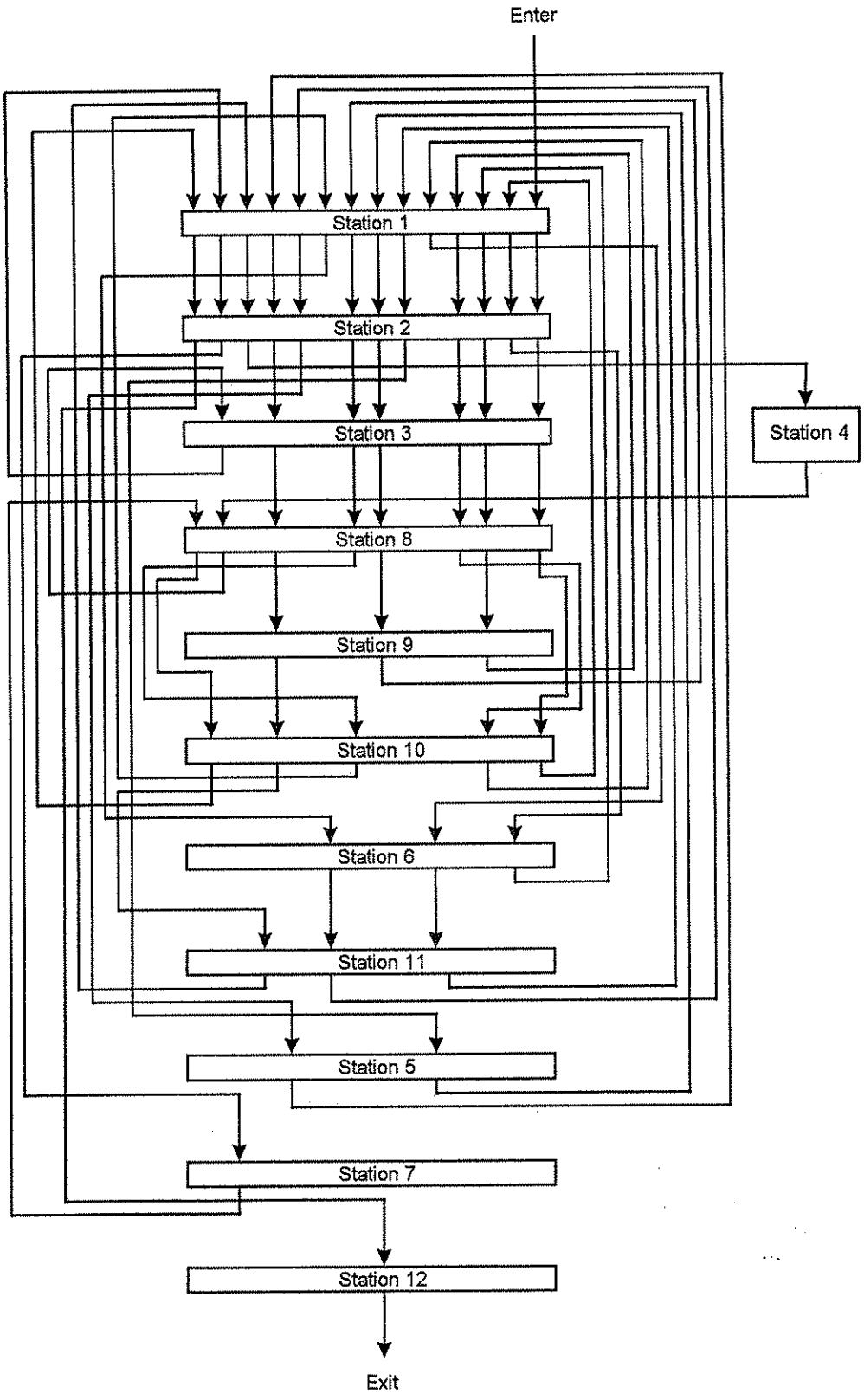

Figure 1 - An aggregated model of a production line, from Lu, Ramaswamy, and Kumar [11]

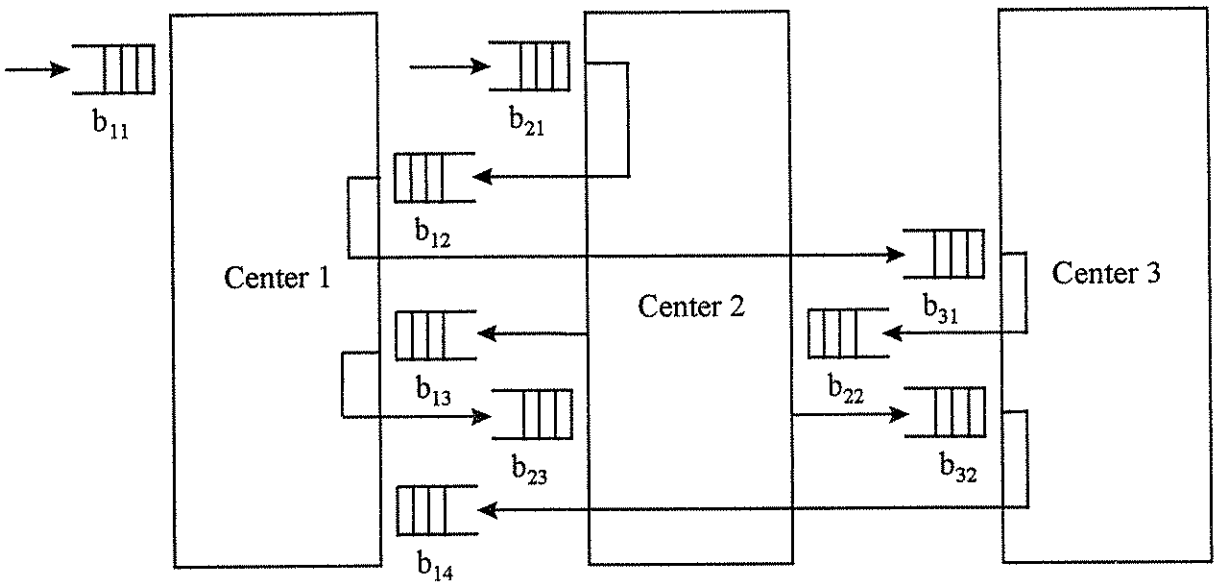

Figure 2 - A re-entrant line with three work centers, from Narahari and Khan [13]

fabs, especially those which produce computer memory chips at a high volume. Other wafer fabs, for example, those that produce application specific integrated circuits (ASICs), may produce thousands of different types of wafers. For wafer fabs producing multiple products, the addition of color to the Petri net would be required. Although the complexity of the models presented in this paper will be less than those of a full-scale wafer fab, the concepts can be extended to larger models.

4. ASSOCIATED LITERATURE

The literature on semiconductor wafer fabrication can be categorized in a number of ways. Some papers deal with production planning, while others go into more detail on determining the best scheduling and input control policies. For a comprehensive review of production planning and scheduling models in the semiconductor industry, see Uzsoy, Lee, and Martin-Vega [15, 16]. Some researchers have also developed queueing models, which are used to predict average cycle times and throughput rates, and compare the results to simulation

models, while some have used simulation models exclusively. The simulation models are typically more useful than queueing models for analyzing various scheduling and work release policies, since the queueing models generally have restrictive assumptions.

A seminal paper on re-entrant lines was written by P.R. Kumar in 1993 [9]. Kumar recognized that re-entrant lines were a new class of manufacturing systems that could not be treated as either a flow shop, with its fixed an acyclic routing, or as a job shop, which is characterized by random routings, each having a low volume. In addition to developing a queueing model for a re-entrant flow line with first come first served (FCFS) queueing discipline, Kumar reiterated results of an earlier simulation study that he had conducted with S.H. Lu [10]. In this study, a re-entrant line with 5 machines and five sequential visits to each machine was simulated using four different scheduling policies, FCFS, first buffer first served (FBFS), last buffer first served (LBFS), and a least slack (LS) policy designed to reduce the variance of lateness. They determined that for this type of re-entrant line, the LBFS policy resulted in the lowest mean delay time in the system, while the LS policy resulted in the lowest variance in cycle times.

Queueing network models

Queueing networks have been used extensively to model semiconductor manufacturing facilities. Burman, et. al. [1], reported on a queueing network model of the photolithography area and compared its outputs a validated simulation model's predictions for the same area.

Chen, et al [3], developed a queueing network model of a semiconductor R&D facility that processed a variety of different types of products with between 2 and 200 operations per product routing.

Dai, Yeh, and Zhou [4] developed a queueing network using the QNET method following Harrison and Nguyen [6, 7], modeling the FBFS and LBFS queueing disciplines. They used *d*-dimensional reflected Brownian motion under a heavy traffic assumption to model the workload process, where *d* represents the number of workstations.

In two papers, Narahari and Khan [13, 14] use a queueing network and Mean Value Analysis to provide performance measures for re-entrant flow lines using fixed buffer priority policies such as FCFS, FBFS, and LBFS. In the first paper they analyzed both of the models shown in Figures 1 and 2, and found very good agreement in mean throughput times between the queueing model and a simulation model. The second paper extended the analysis done in the first paper to include inspections and probabilistic routings, and also showed close agreement in mean throughput time between the queueing network and simulation models.

Simulation models

Simulation models also have been proven useful in analyzing scheduling policies in re-entrant flow systems. Two of the most extensive studies were performed by Wein [17] and Lu, Ramaswamy, and Kumar [11]. Wein used a simulation model to analyze a variety of input control and sequencing rules and their effect on mean throughput time. Wein found that although scheduling has a significant impact on wafer fab operation, larger improvements were gotten from discretionary input control. Specifically, deterministic,

closed loop, and workload regulating input provided improved performance over Poisson input by reducing both the mean and variability of throughput times.

Lu, Ramaswamy, and Kumar [11] also investigated input control and scheduling policies in semiconductor manufacturing, using both Wein's model and a aggregate model intended to approximate a full-scale production line. They defined a new subclass of least slack policies named *Fluctuation Smoothing Policy for Variance of Lateness* (FSVL), *Fluctuation Smoothing Policy for Variance of Cycle Time* (FSVCL), and *Fluctuation Smoothing Policy for Mean Cycle Time* (FSMCT). The conclusion was that the suggested FLMCT/FSVCL policies were best for reducing the mean queueing time and the standard deviation of cycle time, except for when a closed-loop release policy was used. The closed loop policy makes the mean cycle time relatively insensitive with respect to the scheduling policy.

5. PETRI NET MODELS OF RE-ENTRANT FLOW SYSTEMS

A re-entrant flow system with 3 work centers each having a maximum of 4 processing stages in considered here (Figure 2). The interpretation of places and transitions for the Petri net model of the system considered is given by Table 1. As mentioned in section 4, scheduling and work release policies can have a dramatic effect on wafer fabrication performance measures such as average cycle time and standard deviation of cycle time. Here, we show how Petri net models can be used to model some of the scheduling (i.e. queueing) and input regulation policies examined by other researchers.

Closed loop work release policy, multiple queueing disciplines

Figure 3 shows a Petri net model of the system with a closed loop work release

Table 1: Places and Transitions for Figures 3, 4, and 5

Name	Interpretation
P1	Machine 1 processing a part at stage 1
P2	Machine 2 processing a part at stage 1
P3	Machine 1 processing a part at stage 2
P4	Machine 3 processing a part at stage 1
P5	Machine 2 processing a part at stage 2
P6	Machine 1 processing a part at stage 3
P8	Machine 3 processing a part at stage 2
P9	Machine 1 processing a part at stage 4
P10	Machine 1 available
P11	Machine 2 available
P12	Machine 3 available
P13	Shared buffer for machine 1 (Figures 3 and 5) Buffer for machine 1 at stage 1 (Figure 4)
P14	Shared buffer for machine 2 (Figures 3 and 5) Buffer for machine 2 at stage 1 (Figure 4)
P15	Shared buffer for machine 3 (Figures 3 and 5) Buffer for machine 1 at stage 2 (Figure 4)
P16	Buffer for machine 3 at stage 1 (Figure 4)
P17	Buffer for machine 2 at stage 2 (Figure 4)
P18	Buffer for machine 1 at stage 3 (Figure 4)
P19	Buffer for machine 2 at stage 3 (Figure 4)
P20	Buffer for machine 3 at stage 2 (Figure 4)
P21	Buffer for machine 1 at stage 4 (Figure 4)
T1	Lot supplied to machine 1 at stage 1
T2	Lot supplied to machine 2 at stage 1
T3	Lot supplied to machine 1 at stage 2
T4	Lot supplied to machine 3 at stage 1
T5	Lot supplied to machine 2 at stage 2
T6	Lot supplied to machine 1 at stage 3
T7	Lot supplied to machine 2 at stage 3
T8	Lot supplied to machine 3 at stage 2
T9	Lot supplied to machine 1 at stage 4
T10	Processing time of machine 1 at stage 1
T11	Processing time of machine 2 at stage 1
T12	Processing time of machine 1 at stage 2
T13	Processing time of machine 3 at stage 1
T14	Processing time of machine 2 at stage 2
T15	Processing time of machine 1 at stage 3
T16	Processing time of machine 2 at stage 3
T17	Processing time of machine 3 at stage 2
T18	Processing time of machine 1 at stage 4
T19	New lot supplied to system (Figure 5)

policy, also known as the constant work in process (WIP) policy, in which many queuing disciplines may be modeled. Tokens in places p_1, p_3, p_6, and p_9 would represent lots being processed at work center 1, but at four different stages of the process. A token in place p_{10} would indicate that a machine at center 1 is available for processing. Therefore, the total number of tokens residing in these five places at any given time will always equal the number of machines at work center 1. A similar argument could be made for work centers 2 and 3, where places p_{11} and p_{12} respectively represent machine availability, while places p_2, p_5, and p_7 for work center 2 and places p_4 and p_8 for work center 3 represent different processing stages at their respective work centers.

The marking shown here indicates that there are three machines at work center 1, two at work center 2, and one, which is currently available, at work center 3. Work center 1 has two lots in process at stage 1 and one lot in process at stage 4, while work center 2 has one lot in process at each of stages 2 and 3. Transitions t_{10} through t_{18}, which represent the processing times for the various work centers and processing stages, are depicted by unfilled rectangles, indicating stochastic timed transitions. Since this Petri net has both immediate and stochastic transitions, it is a called a generalized stochastic Petri net (GSPN). The closed loop work release property of this net is modeled by the arc connecting transition t_{18} to place p_{13}, which represents the buffer for work center 1. The token in place p_9 indicates a lot at its final processing stage. When transition t_{18} is finished firing, one token will be placed in both p_{10}, indicating that a machine is available, and p_{13}. The token placed in p_{13} represents a new lot entering the system concurrent to the lot from place p_9 leaving the system. This keeps the number of lots in the system constant.

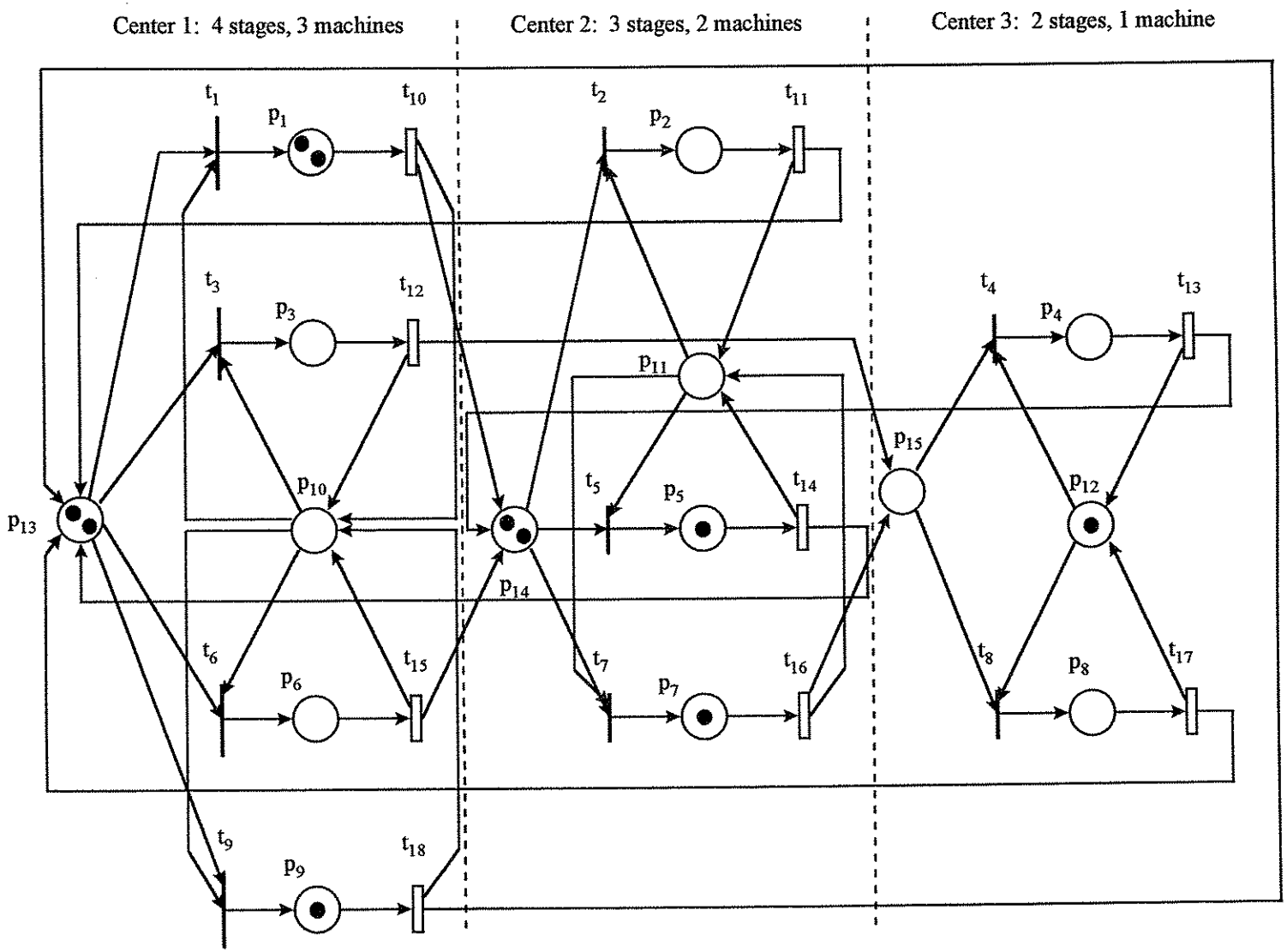

Figure 3: Re-entrant flow manufacturing system with closed loop work release policy

Common buffer places, represented by p_{13}, p_{14}, and p_{15}, are provided for each machine, allowing for a wide range of queueing disciplines to be modeled. In order to model the queueing disciplines of first in first out (FIFO) or last in first out (LIFO), a firing list may be kept for each work center. As each token arrives in a buffer, an entry to the list would indicate the transition to be activated by that token. In this example, when a token arrives to place p_{15} due to the firing of transition t_{16}, transition t_8 will be added to the list for work center 3. For FIFO queuing, the entry would be put at the end of the list, and for LIFO the front of the list. When a machine becomes available, the transition at the top of the list is fired and that entry removed from the list.

Another way to examine different queuing disciplines would be to use a colored Petri net (CPN). In a CPN, each type of token is unique, although there may be more than one token of any given type. Attributes associated with each type of token may be used to examine different queueing disciplines from a list of tokens in a queue. For example, the least slack policies developed by Lu, et al [11] were based on calculations derived from product attributes, and could easily be modeled using a colored Petri net. Other common queueing disciplines that can be modeled in this manner are shortest process time (SPT), in which lots are sequenced at a work center according to the shortest expected process time at that work center, and earliest due date, in which the lot with the earliest due date is processed first at a given work station. Many other queueing disciplines, such as those examined by Lu, et al [11] and by Wein [17] can also be modeled using colored Petri nets. The queueing policy selected for a given system depends on the goal(s) in mind. For example, a policy that reduces mean cycle time may not address the problem of late orders. Petri nets such as those discussed here can be used to determine system performance

measures for different queueing policies so that the best policy for a specific situation may be chosen.

Closed-loop work release policy, last buffer first served queueing discipline

Figure 4 represents a system with a closed-loop work release policy and a last-buffer-first-served (i.e. shortest remaining processing time) queueing discipline, previously studied by Wein [17] and Lu, et al [11]. To model this type of queueing discipline, buffers at each work station are segregated according to the processing stage. Then, inhibitor arcs of capacity 1 are connected between all later buffer places to earlier transitions of the same work center. This inhibits earlier transitions from firing if there are any parts in later buffers. Figure 4 has been marked with the same total amount of tokens in buffers and machines as in Figure 3, but the buffer tokens reside in the place associated with the next processing stage for that token. At work center 1, the token in place p_{18} inhibits the firing of transitions t_1 and t_3. Since there is no token in place p_{21}, transition t_6 will fire once a token is placed in p_{10} unless a token arrives at place p_{21} first. In that case, transition t_9 will fire instead of t_6.

Bottleneck workload regulating policy

Figure 5 represents a system with an open-loop work release policy, specifically a bottleneck workload regulating policy, and the same queueing scenario as in subsection A. A similar type of work regulating policy was studied by Wein [17] and Lu, et al [11]. In this figure, it is assumed that work center one is the only bottleneck. A new lot will be released into the system by the firing of transition t_{19} only when the number of lots in buffer place p_{13} drops below 2. This is modeled by connecting an inhibitor arc of capacity 2 between p_{13} and t_{19}. Other work release policies could also be modeled with minor

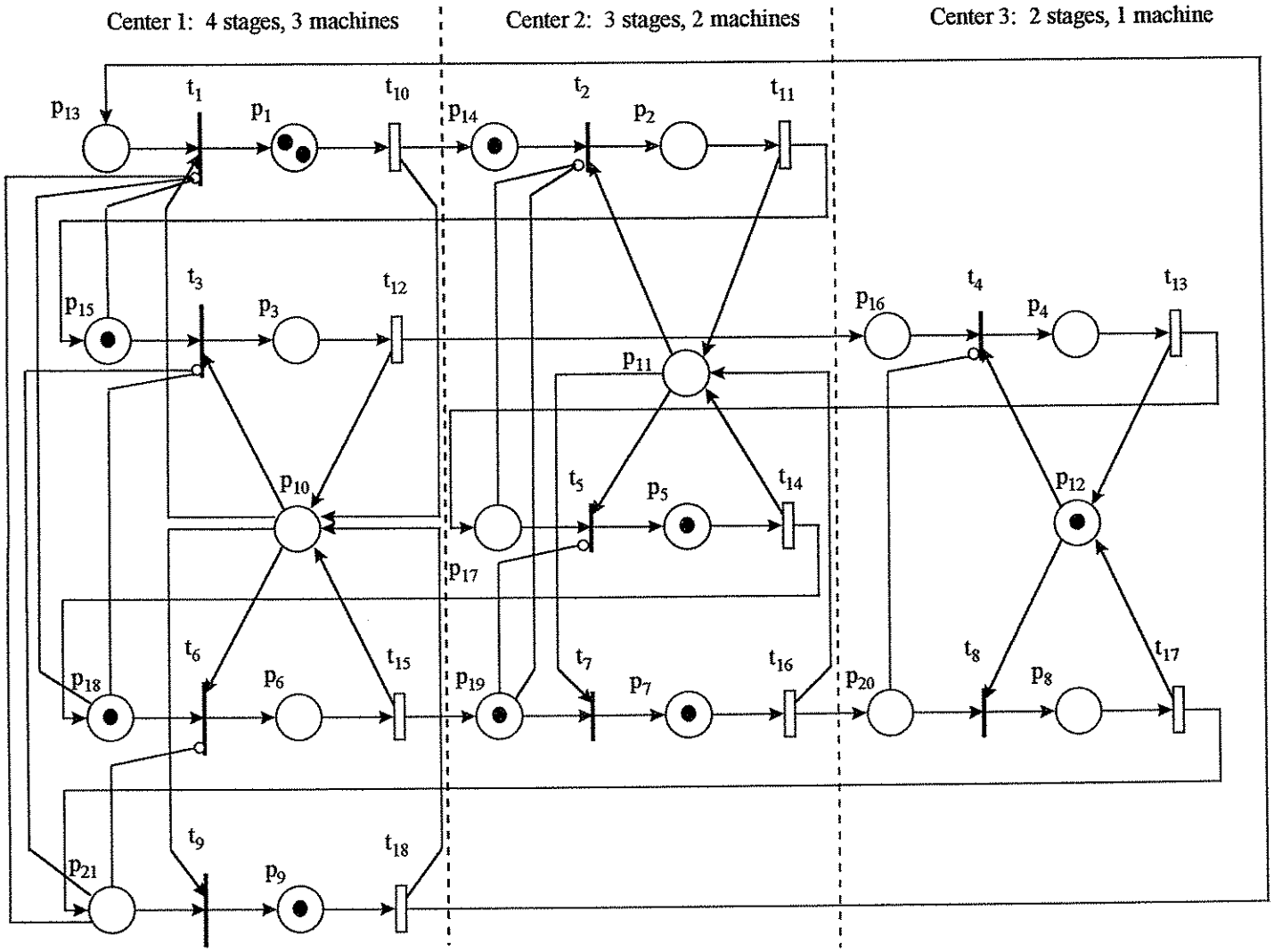

Figure 4: Re-entrant flow manufacturing system with closed-loop work release policy and last buffer first served queueing discipline

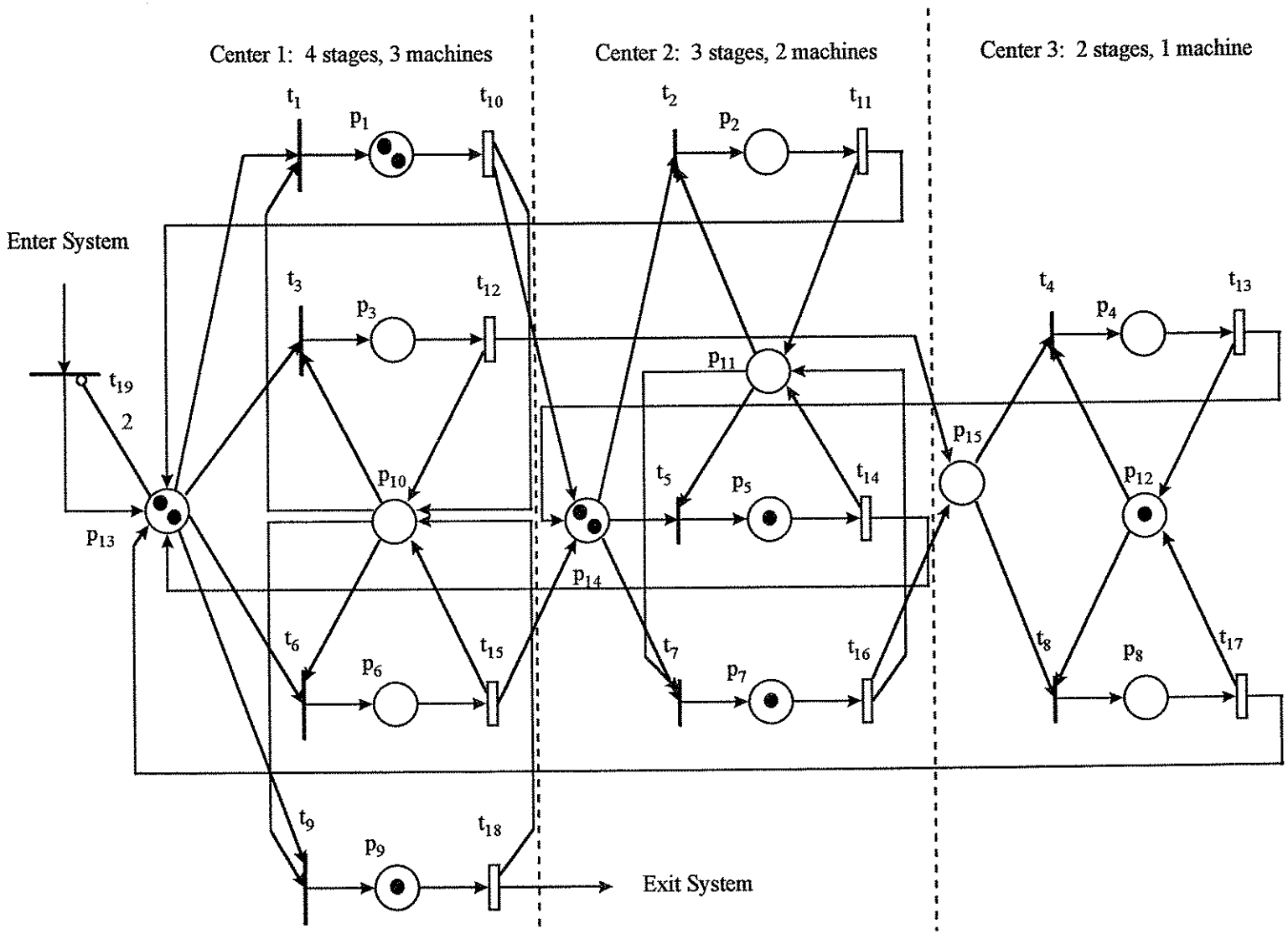

Figure 5: Re-entrant flow manufacturing system with bottleneck workload regulating work release policy

modifications to this net. A two-bottleneck work regulating policy can be modeled by attaching another inhibitor arc to transition t_{19} from the buffer place associated with the second bottleneck work station. A Poisson work release could be modeled by removing the inhibitor arc and making t_{19} an exponential transition. A deterministic work release policy could be modeled by making the firing time of transition t_{19} a constant.

Modeling machine breakdowns

Machine breakdowns have not been included in Figures 3-5, but could be modeled as shown in Figure 6 with the addition of two places, one representing a machine in a repair state and one representing a machine up and running at any stage, and two timed transitions per machine, representing time between failures and machine repair time. Here, the machine will break down only when it is running.

The operation of this net is such that the firing of immediate transition t_1 will remove the token from the "machine available" place and place one token each in the "machine stage 1" and "machine running" places, at which time the transitions marked "processing time for stage 1" and "time between failures" will both begin firing. Depending on which of the two transitions completes firing first, the token will be removed from the "machine running" place and a token will either go to the output buffer or to the "machine being repaired" place. The timer for the transition not yet finished firing must be paused. When a transition's timer is paused and the transition is later enabled, its timer may either be reset to zero or be resumed where it left off. The choice depends on the specific situation. For transitions representing time between failures, the timer should be paused and restarted. Otherwise, the transition may never fire if time between failures is much longer than the processing time. For transitions representing processing times, the choice depends on

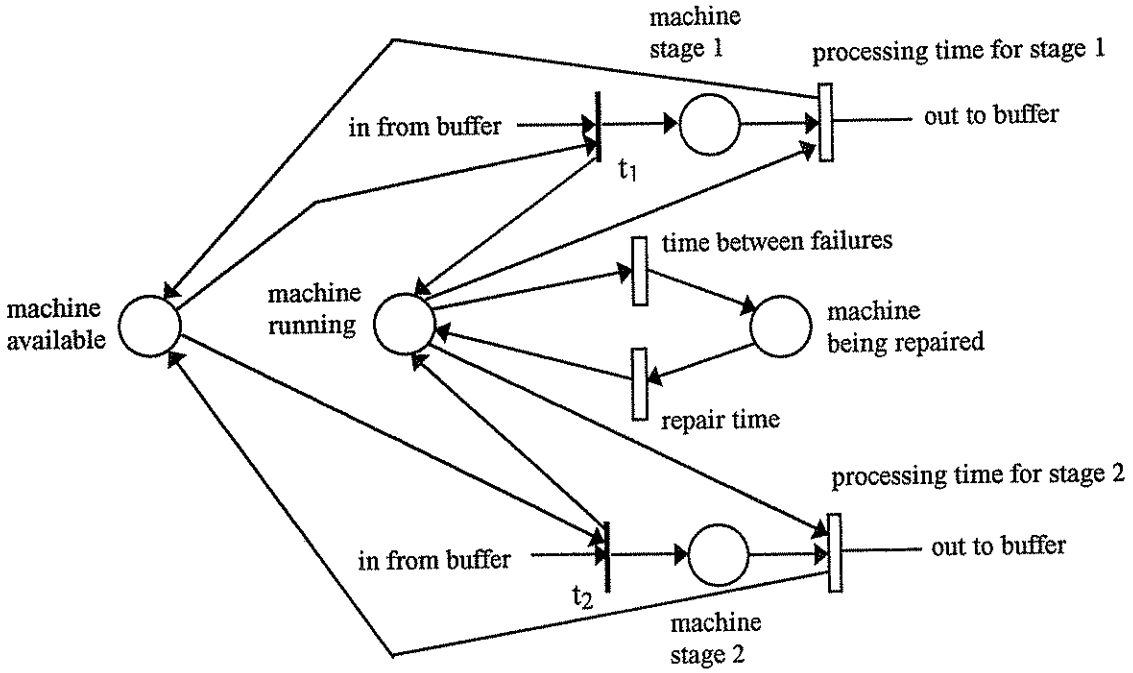

Figure 6: Two-stage machine with random machine breakdowns

whether the process must be started over or if it can be resumed at the point that it was interrupted.

Figure 6 represents a one machine work station. In order to model multiple-server work stations with machine breakdowns, separate "machine running" and "machine being repaired" places must be added for each machine with their respective failure and repair times.

Petri nets as a shop floor control system

In addition to using Petri net models to analyze performance measures such as cycle time, work in process, or on-time delivery performance for different scheduling and work release policies, they may be used as part of a real-time shop floor control system. This could be accomplished by programming the Petri net into a shop floor control software

package and tracking each lot as it proceeds through the wafer fab. The difference between using the Petri net for analysis versus control is in the transition firings. Instead of firing the transitions at pre-specified time intervals, they are fired by prompts to the system. As operators notify the system that a lot is started or completed on a given machine, the associated transition in the Petri net fires and the system state is updated. The system will notify the users when to release a new lot into the wafer fab, and which lot to process next on a given machine. Modifications to the models presented here may be required to add realism to the system. For example, modeling hot lots (lots that have priority over all others), monitor lots (non-product lots run periodically to monitor machine performance), and engineering holds.

6. CONCLUSIONS

This paper has presented Petri net models for re-entrant flow manufacturing systems. The example here is in semiconductor manufacturing. We have shown how Petri-nets can be used to model various scheduling and work release policies in these types of systems. We also discussed the advantages of Petri nets over both queueing and simulation models. With respect to queueing models, Petri nets are less restrictive in their assumptions and can model nonproduct-form features such as blocking, synchronization, and priority queueing disciplines. Advantages of Petri nets over simulation models are their compact graphical representation, allowing for visual analysis of complex systems, their mathematical foundation, allowing for analysis of behavioral and structural properties of the system, and their ability to be used as real-time controllers. Logical extensions of this work would be to model a full-scale wafer fabrication facility, add color to the Petri net to represent difference part types, and to model wafer yield, hot lots, monitor lots, and engineering holds. Also,

these nets could be programmed into a Petri-net software package and analyzed for different work release and scheduling policies.

REFERENCES

[1] D.Y. Burman, F.J. Gurrola-Gal, A. Nozari, S. Sathaye, and J.P. Sitarik, Performance Analysis Techniques for IC Manufacturing Lines, *AT&T Technical Journal*, vol. 65, no. 4, pp. 46-57, July/August 1986.

[2] S. Cavalieri, O. Mirabella, and S. Marroccia, Improving Flexible Semiconductor Manufacturing System Performance by a Coloured Petri Net-based Scheduling Algorithm, in: *1997 IEEE 6th International Conference on Emerging Technologies and Factory Automation proceedings*, Los Angeles, CA, pp. 369-374, September 1997.

[3] H. Chen, J.M. Harrison, A. Mandelbaum, A. Van Ackere, and L.M. Wein, Empirical Evaluation of a Queueing Network Model for Semiconductor Wafer Fabrication, *Operations Research*, vol. 36, no. 2, pp. 202-205, March-April, 1988.

[4] J.G. Dai, D.H. Yeh, and C. Zhou, The QNET Method for Re-entrant Queueing Networks with Priority Disciplines, *Operations Research*, vol. 45, no. 4, pp. 610-623, July-August, 1997.

[5] A.A. Desrochers, and R.Y. Al-Jaar, Applications of Petri Nets in Manufacturing Systems, IEEE Press, Piscataway, NJ, 1995.

[6] J.M. Harrison and V. Nguyen, The QNET Method for Two-Moment Analysis of Open Network Queueing Networks, *Queueing Systems: Theory and Applications*, vol. 6, no. 1, pp. 1-32, 1990.

[7] J.M. Harrison and V. Nguyen, Brownian Models of Multiclass Queueing Networks: Current Status and Open Problems, *Queueing Systems: Theory and Applications*, vol. 13, no. 1, pp. 5-40, 1993.

[8] J. Kim and A.A. Desrochers, Modeling and Analysis of Semiconductor Manufacturing Plants using Time Petri Net Models: COT Business Case Study, in: *IEEE International Conference on Systems, Man, and Cybernetics,* Orlando, FL, vol. 4, pp. 3227-3232, October 1997.

[9] P.R. Kumar, Re-entrant lines, *Queueing Systems: Theory and Applications: Special Issue on Queueing Networks*, vol. 13, no. 2, May 1993.

[10] S.H. Lu and P.R. Kumar, Distributed Scheduling Based on Due Dates and Buffer Priorities, *IEEE Transactions on Automation & Control*, vol. 36, pp. 1406-1416, 1991.

[11] S.C.H. Lu, D. Ramaswamy, and P.R. Kumar, Efficient Scheduling Policies to Reduce Mean and Variance of Cycle-Time in Semiconductor Manufacturing Plants, *IEEE Transactions on Semiconductor Manufacturing*, vol. 7, no. 3, pp. 374-388, August 1994.

[12] T. Murata, Petri Nets: Properties, Analysis and Applications, *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541-580, April 1989.

[13] Y. Narahari and L.M. Khan, Performance Analysis of Scheduling Policies in Re-entrant Manufacturing Systems, *Computers Operations Research*, vol. 23, no. 1, pp. 37-51, 1996.

[14] Y. Narahari and L.M. Khan, Modeling Reentrant Manufacturing Systems with Inspection Stations, *Journal of Manufacturing Systems*, vol. 15, no. 6, pp. 367-378, 1996.

[15] R. Uzsoy, C-Y Lee, L.A. Martin-Vega, A Review of Production Planning and Scheduling Models in the Semiconductor Industry, Part I: System Characteristics, Performance Evaluation and Production Planning, *IIE Transactions*, vol. 24, no. 4, pp. 47-60, September 1992.

[16] R. Uzsoy, C-Y Lee, L.A. Martin-Vega, A Review of Production Planning and Scheduling Models in the Semiconductor Industry, Part II: Shop-Floor Control, *IIE Transactions*, vol. 26, no. 5, pp. 44-55, September 1994.

[17] L.M. Wein, Scheduling Semiconductor Wafer Fabrication, *IEEE Transactions on Semiconductor Manufacturing*, vol. 1, no. 3, pp. 115-130, August 1988.