Distributed Performance Evaluation of a Controlled IC Fab

Chung-Hsien Kuo and Han-Pang Huang

***Abstract*—Integrated circuit (IC) foundry fabs are difficult to model due to the increasing product mixes and flexible routes. An imprecise fab model cannot be used to evaluate fab performance or estimate product cycle time. The distributed colored timed Petri net (DCTPN) is a type of high-level Petri net with embedded entity attribute, time, communication, and remote object invocation properties that contribute to realistic descriptions, distributed modeling, and manufacturing execution system integration. Furthermore, the resource-oriented DCTPN modeling extends product mix and flexible route modeling capabilities. The DCTPN conflict resolution and token competition rules are used to construct the dispatching system for a controlled IC fab. A simplified 200 mm IC fab with six functional areas is discussed and demonstrated based on different control policies.**

***Index Terms*—Dispatching system, distributed colored timed Petri net (DCTPN), integrated circuit (IC) foundry fab, manufacturing execution system (MES), performance evaluation, semiconductor manufacturing.**

I. INTRODUCTION

An integrated circuit (IC) foundry fab is one of the most important manufacturing systems in current industries. The increasing product mixes and flexible routes make IC foundry fabs complex. A fab may have more than hundreds of tools and tool groups. It may be hierarchically configured in terms of the plant, areas, tool groups, and tools. In a fab, tools are grouped into six functional areas: photo (PHOTO); sputter (SPUT); chemical vapor deposition (CVD); implantation (IMP); etching (ETCH); and diffusion (DIF) areas. To increase production efficiency, different types of products must use the same tool. Hence, the fab production can be characterized as concurrent operations, resource sharing, mutual exclusion, conflict, decision making, product mixes, parallel operations, and sequential operations. In general, queuing theory, neural networks, finite-state machine, and Petri nets (PNs) [1], [2], [5], [6], [8]–[14], [16], [18]–[21] have been used to model and analyze fab manufacturing systems. Since PNs are powerful for modeling manufacturing systems with concurrent and asynchronous properties [5], [18], it was selected in this study to model the production behaviors of an IC foundry fab.

To model products with various model mixes, the product and tool attributes must be considered. Typically, the standard PN is not suitable for modeling a system with high product mixes. This study, therefore, used a distributed colored timed Petri net (DCTPN) [11] to model a complex IC fab. Based on DCTPN, the entity attribute, time, communication, and remote object invocation can all be described. In addition, the modular description [17] and the standardized model interface extend the modeling capabilities for new systems and the remodeling of reconfigured systems.

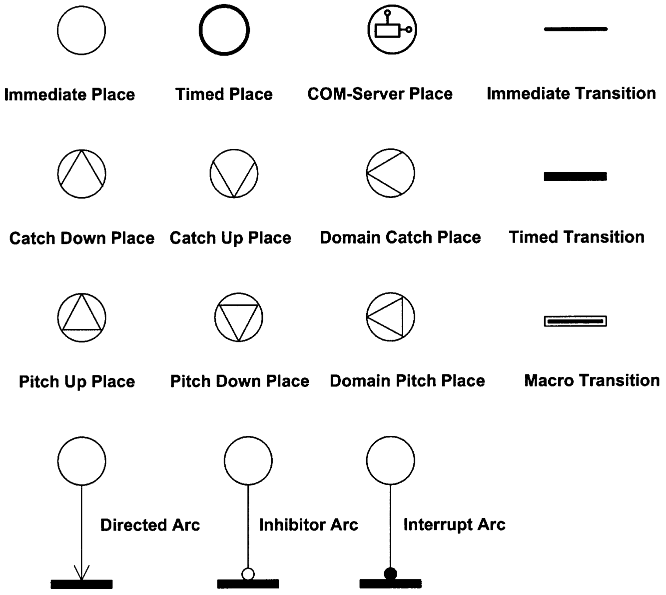


Fig. 1. DCTPN icon definitions.

A resource-oriented modeling approach was used in this study. The aim was to use the same DCTPN model to evaluate different types of products. The basic DCTPN tool model library is constructed based on different types of tools. Such a tool model library can be used to construct the DCTPN model of an entire fab. The DCTPN conflict resolution and token competition rules are used to construct the dispatching system. The DCTPN conflict resolution rules select the available resources (tools) for the wafer lots. The DCTPN token competition rules act as the wafer lot dispatching queue for resources (tools). The DCTPN simulation results can be used to estimate product cycle time and evaluate fab performance. A simplified 200 mm IC fab with six functional areas is discussed and demonstrated based on various control policies.

II. DCTPNs

DCTPNs [11] are a class of high-level PNs, based on the notion of colored timed Petri nets (CTPN) [10], [12], [13], [16] and of distributed systems [3]. DCTPN contains new additional elements, such as COM server, domain pitch and domain-catch places, and provides an object-based and distributed computing environment (DCE) [3]. A DCTPN is defined as an eleven-tuple structure, $\mathrm{DCTPN} = (P_t, T_t, P_0, T_0, P_c, P_d, T_m, A, B, F, C)$, where P_t is a set of timed places; T_t is a set of timed transitions; P_0 is a set of immediate places; T_0 is a set of immediate transitions; P_c is a set of communication places; P_d is a set of COM server places; T_m is a set of macro transitions; A is a set of directed arcs; B is a set of inhibitor arcs; F is a set of interrupt arcs, and C is the color set for the transitions and places. The DCTPN icon definitions are shown in Fig. 1. These elements are further described as follows.

1) **Place**: $\mathbf{P} = \{\mathrm{p1}, \mathrm{p2}, \mathrm{p3}, \mathrm{p}n\}$ is a finite set of places, $n \geq 1$ includes the immediate, timed, COM server, and communication (catch-up, catch-down, pitch-up, pitch-down, domain-catch, and domain-pitch) places. An immediate place is used to describe the status, command, condition, or resource. A timed place [10] is designed to model the type of resource and the tokens in it be-

---

Manuscript received February 9, 2001; revised July 15, 2001; April 30, 2002; and January 30, 2003. This paper was recommended for publication by Associate Editor X. Xie and Editor N. Viswanadham upon evaluation of the reviewers' comments. This work was supported in part by the National Science Council under Grant NSC 88-2218-E-002-003, and in part by the Taiwan Semiconductor Manufacturing Corporation (TSMC). This paper was presented in part at the IEEE International Conference on Robotics and Automation, San Francisco, CA, April 24-28, 2000.

C.-H. Kuo was with the Robotics Laboratory, Department of Mechanical Engineering, National Taiwan University, Taipei 10660, Taiwan, R.O.C. He is now with the Department of Mechanical Engineering, Chang Gung University, Tao-Yuan 333, Taiwan, R.O.C.

H.-P. Huang is with the Robotics Laboratory, Department of Mechanical Engineering, National Taiwan University, Taipei 10660, Taiwan, R.O.C. (e-mail: hanpang@ntu.edu.tw).

Digital Object Identifier 10.1109/TRA.2003.819720

come available after an elapsed time. A pitch-up (pitch-down) place sends tokens to a corresponding catch-down (catch-up) place at the higher (lower) level net among the DCTPN nets. The new additional elements with respect to CTPN [10], [12], [13], [16] are COM server, domain pitch, and domain-catch places. A COM-server place is designed for implementation and practical issues. The simulation usually needs the actual data as the initial condition and inputs to generate results. The DCTPN model uses the COM server place to execute remotely the distributed component object model (DCOM)-based manufacturing execution system (MES). When a token enters a COM-server place, the remote DCOM server can be executed once. The attributes of the new token (color sets) act as the input for the remote DCOM server. The results of DCOM server execution will be the new token attributes (color sets). For example, the order release server can be constructed as a COM-server place for generating new orders in terms of various order-releasing rules. The order-releasing rule is defined as the entering token attribute. Once such a token enters the COM-server place, this token becomes unavailable. Consequently, the DCOM-based order release server is executed. The returned value (i.e., a releasing order with lot identification (ID), product ID, lot priority, commit due date, and so on) will update the token attributes. This token then becomes available.

2) **Transition** [10], [12], [13], [16]: $\mathrm{T} = \{\mathrm{t}1, \mathrm{t}2, \mathrm{t}3, \ldots, \mathrm{t}m\}$ is a finite set of transitions, $m \geq 1$, including immediate, timed, and macro transitions. Their definitions are the same as the CTPN transitions [10], [12], [13], [16]. An immediate transition is used to model an event such as a control decision, and the timed transition is used to model a step of the process that requires some time to be completed. It handles only one token at each firing time. A macro transition is designed for the net refinement that is composed of transitions, places, and arcs. The interconnection between the macro transition and the refined net are established by communication places.
3) **Attribute-based color sets** [10], [12], [13]: $C(p)$ and $C(t)$ represent the color sets for places in P and transitions in T, respectively. The attribute-based color sets are used to identify tokens in a system. In this manner, various lots and tools can be properly described and distinguished. Note that the token ID and the time stamp are basic attributes of the color sets. The time stamp will be updated when the state is changed.
4) **Input, output, inhibitor, and interrupt functions** [10], [12], [13]:

   $\mathrm{I}(p,t)(a,b) : C(p) \times C(t) \rightarrow V(N)$ is an input function. It describes the enabling condition of transition t of color b with respect to the place p of color a, where $V(N)$ is a nonnegative integer function. It indicates the arc cardinality in DCTPN. The $V(N)$ value can be either a fixed value or variable value that refers to the token color sets.

   Similarly, $\mathrm{O}(p,t)(a,b) : C(p) \times C(t) \rightarrow V(N)$ is an output function. It defines the token release rules when a transition t completes firing. In DCTPN, five types of output functions are defined to assign the attribute value of the released tokens. They are: 1) the same attribute value with the current transition operation; 2) the same attribute value with the next operation according to the product route (it is also an attribute value) of the transition; 3) the other attribute value with current transition operation; 4) a user-defined value; and 5) the transition name. These releasing functions fit the fab operations.

   $\mathrm{Inh}(p,t)(a,b) : C(p) \times C(t) \rightarrow V(N)$ is an inhibitor function, and the transition t must be an immediate or timed transition. The place connected with an inhibitor arc is called an inhibitor place. When an inhibitor place contains a number of tokens of the same token color sets of an output transition, and is greater than or equal to the number of arcs going to the transition, the output transition is disabled and cannot be fired.

   $\mathrm{Int}(p,t)(a,b) : C(p) \times C(t) \rightarrow V(N)$ is an interrupt function, and the transition t must be a timed transition. The place connected with an interrupt arc is called an interrupt place. When an interrupt place contains a number of tokens of the same token color sets of an output transition, and is greater than or equal to the number of arcs going to the transition, the firing of the output (timed) transition is interrupted immediately.
5) **DCTPN Color Markings**: $\mu(p) : C(p) \rightarrow N; \forall p \in P$ is a $(n \times 1)$ vector. Each element is denoted as $\mu(p_i)$, and it is defined as follows:

   $\mu(p_i) = \sum_{h=1}^{ui} n_{ih} a_{ih}$, where n_{ih} equals $\mu(p_i)(a_{ih})$ the number of tokens with colors a_{ih} in place p_i at this instant; u_i is the total number of colors in place p_i; and $\mu_0(p)$ is the initial marking.
6) **Time function** [10], [12], [13]: It is simply the time attribute for the timed places and timed transitions. For the timed transition, $f(t(b_{jk}))$ is defined as the elapsed time for the timed transition t associated with color b_{jk} to complete the firing. For the timed place, $f(p(a_{ih}))$ is defined as the elapsed time for the tokens staying in the timed place p associated with color a_{ih}. Therefore, the elapsed times for the timed transition firing and the tokens staying in the timed place depend on the attribute-based color sets. For example, the processing time may depend on the product type and its current operation stage. In this case, such a process can be modeled as a timed transition, and the product type and operation stage form the attribute-based color sets. Therefore, the elapsed time for the timed transition firing is determined by their attribute-based color sets. Note that the stochastic time can also be applied to the attribute-based time function to overcome more model uncertainties.
7) **DCTPN enabling and firing, reachability, safeness and boundedness, liveness and deadlock**: These properties are the same as the definitions of CTPN (see [10], [12], [13], and [16] for details).

Hence, DCTPN is suitable to model the IC foundry fab. The comparisons between DCTPN and the standard PN models (including CTPNs) are illustrated as follows.

1) The color sets in DCTPN are different from color PNs. In DCTPN, the color set is a set of manufacturing attributes. In a fab model, they may be the lot ID, lot priority, product type, and so on. Hence, elements in DCTPN are actually related to the physical entities. Based on the attribute-based color sets, production control can be implemented in a precise way. Consequently, the time function is defined according to the color sets. This is a useful feature of our resource-oriented modeling approach that cannot be easily obtained using standard colored PN models.
2) DCTPN is designed for resource-oriented modeling, and it is independent of product routes. Therefore, the same DCTPN model can be used to evaluate different types of products. This cannot be easily done using standard PN models.
3) The DCTPN resolution rules are developed to solve the competition and conflict problems when enabling and firing. In this manner, DCTPN can be used to construct the production control system in terms of applying these resolution rules. This cannot be easily done using standard PN models.
4) Using standard PN models, it is hard to link the initial marking with the physical manufacturing data (e.g., fab MES database).

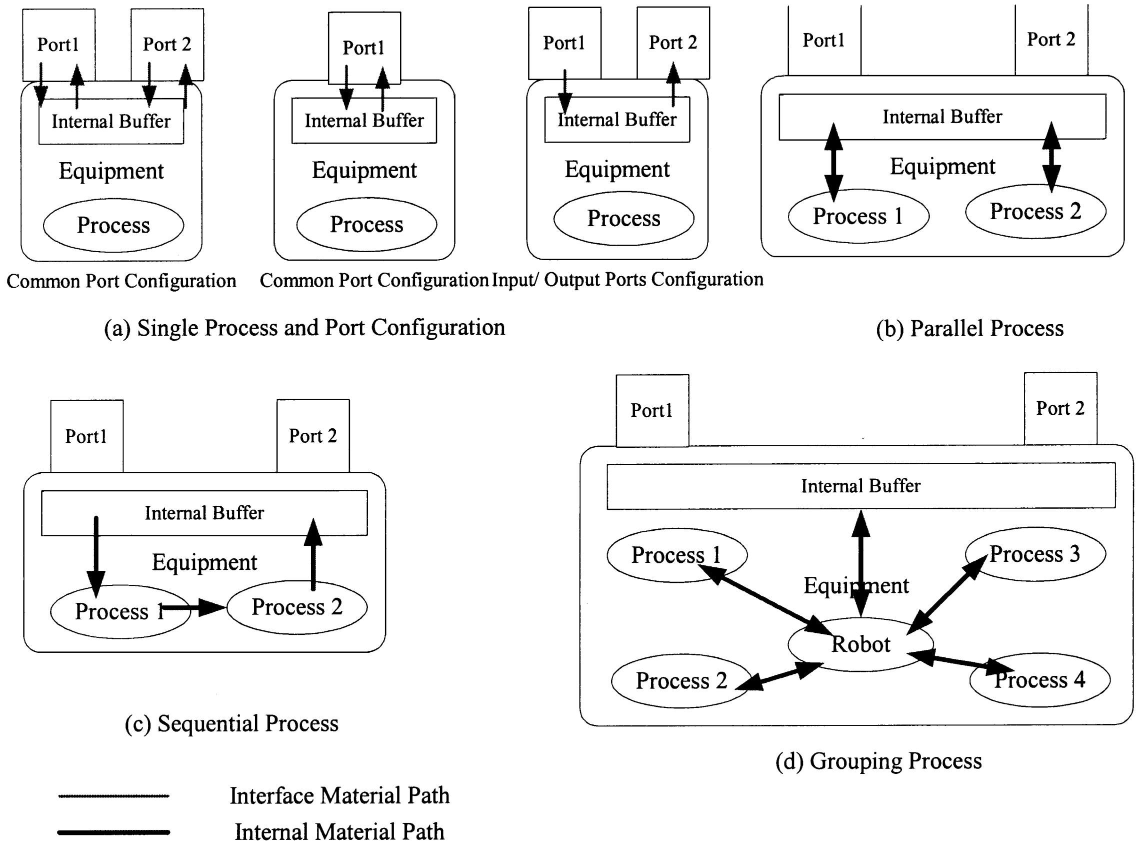


Fig. 2. Different types of tools in the fab.

In DCTPN, the COM-server place defines a set of interfaces to connect with the DCOM-based MES server. Hence, the DCTPN model can use the actual manufacturing data to simulate the system.

III. MODELING A CONTROLLED IC FAB USING DCTPN

When modeling a controlled IC fab, the physical manufacturing components must be mapped to the DCTPN elements. They are summarized as follows. An immediate place is used to model the manufacturing condition, status, command, or shopfloor buffer. A timed place is used to model the conveyor, and the boundedness of the timed place is the maximum number of workpieces on a conveyor simultaneously. A COM server place is used to model the MES server. The communication places provide the messages passed among the models; an immediate transition is used to model an event or decision. A timed transition is used to model the resource operation that requires an elapsed time, and a macro transition is used to model a tool, a tool group, or an area. The relationships between the places and transitions are established via three types of arcs.

The production control in the IC fab model is achieved by applying the conflict transitions and competitive token resolution rules [11]. Since DCTPN is designed for modeling a modern manufacturing system, the resolution rules are developed based on manufacturing characteristics. The rules for resolving the conflict transitions are developed based on the following properties: 1) firing count for conflict transitions; 2) firing time for conflict transitions; 3) queue length (note that the queue length of a transition is calculated from the number of tokens in a "queue length referred" input place for this transition) for conflict transitions; 4) priority of conflict transitions; and 5) matching the transition name and the token color set.

In addition, the token competition rules [11] are used to assign the releasing priorities for the tokens in a place. The rules for resolving token competition problems are developed based on the following token properties: 1) first-come, first-served (FCFS); 2) priority; 3) last-come first-served (LCFS); 4) earliest commit due date (EDD) attributes; 5) next processing time attributes; 6) weighted total processing time attributes; 7) remaining processing time attributes; 8) remaining operation count attributes; and 9) releasing time attributes.

To hierarchically construct a fab model, the commonly used basic tool models are constructed to form a tool model library. The behaviors of the tools can be characterized as the port and the process configurations. A port is a mechanical interface that transports wafers into a tool. Fig. 2 shows the operations of six types of tools. Fig. 2(a) shows three single-wafer process tools with different port configurations. The "common port configuration" indicates that the lot enters and exits the tool via the same port. The "input/output port configuration" indicates that the lot enters the tool via an input port and exits the tool via an output port. In addition, the operation for the "single lot/wafer process" indicates that the tool processes only a single lot or wafer. The material flows for these types of port configurations are assigned by the arrows indicated in Fig. 2(a). Fig. 2(b) shows a parallel process tool configuration, and the processes in it operate in parallel. Fig. 2(c) shows a sequential process tool configuration, and the processes in it operate in sequence. Fig. 2(d) shows a grouping process tool configuration, and the processes in it operate in terms of a specified route. The vacuum robot is used to transport wafers among chambers. Note that the port configurations in Figs. 2(b)-(d) are similar to the port configuration in Fig. 2(a).

A DCTPN tool model of the "route determines the single-wafer process tool with two common ports" is illustrated in Fig. 3. This tool has three process chambers. In this model, the physical meanings

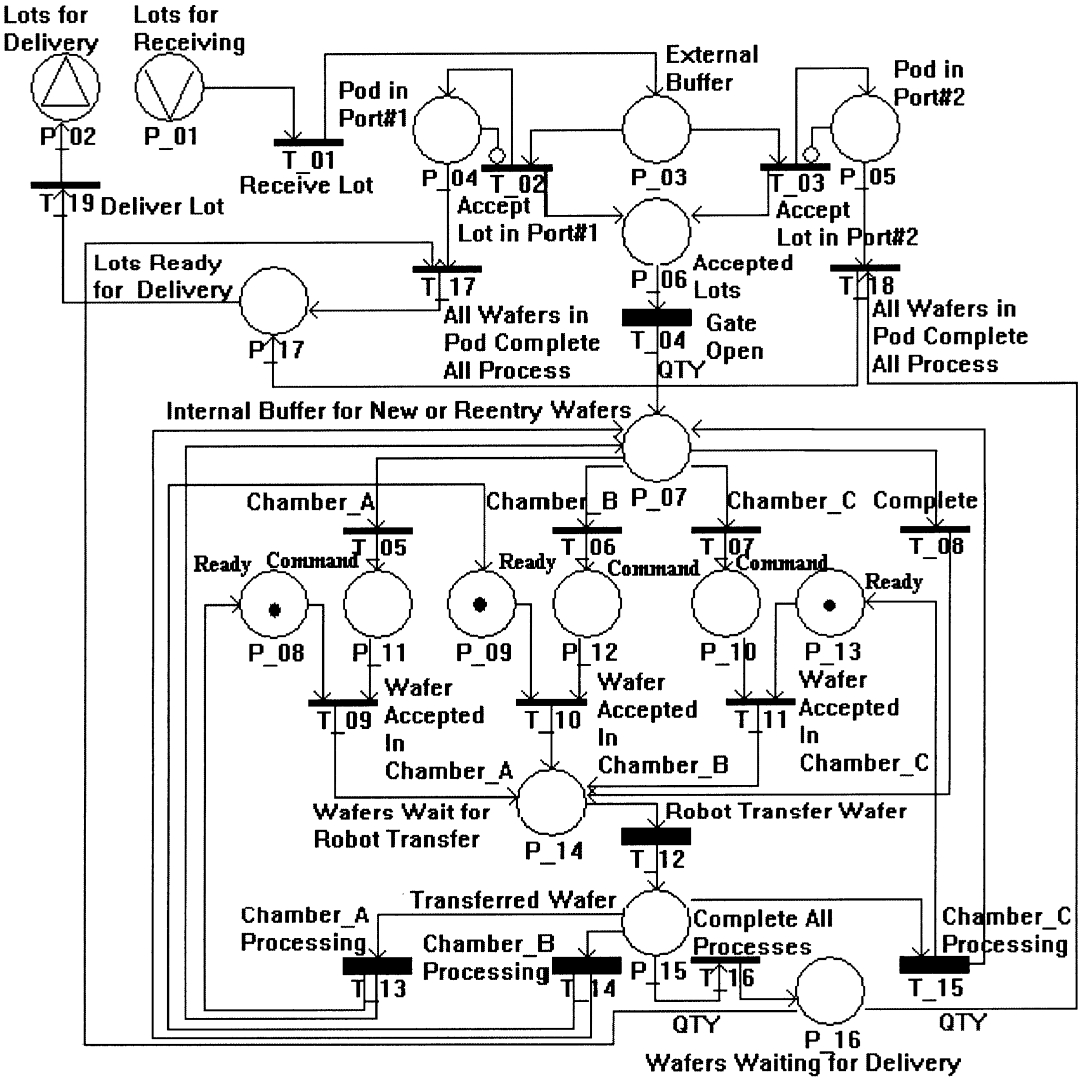


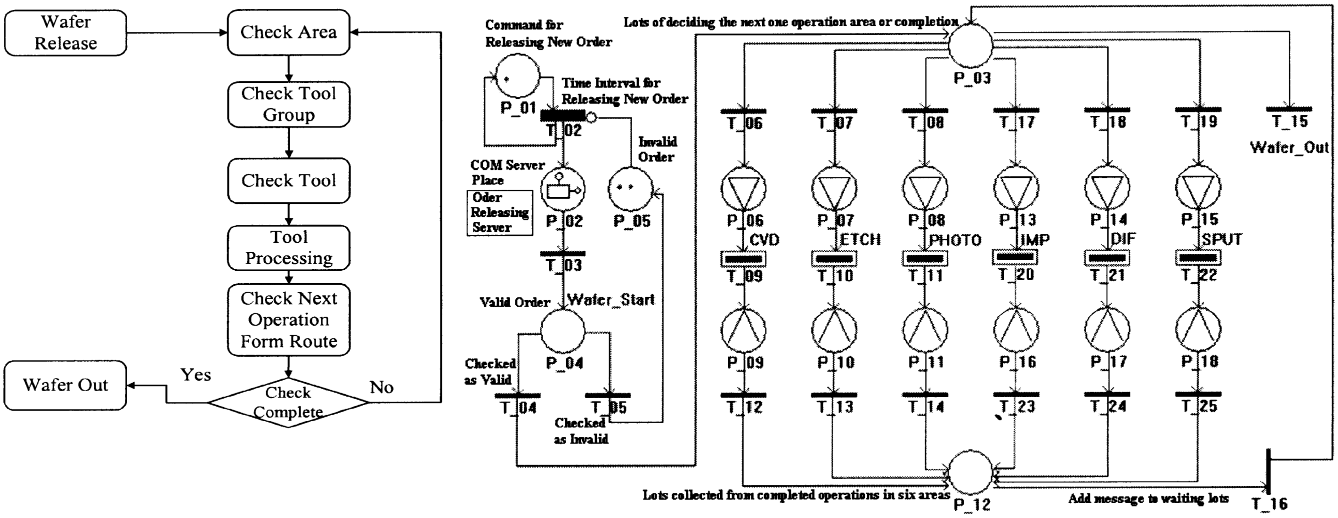
Fig. 3. DCTPN model of a route-determined single-wafer process tool with two common ports.

of the places and transitions are indicated as the caption beside the corresponding elements. In addition, the model for the control system is also illustrated. The wafer lot dispatching can be implemented by applying the "competitive tokens resolution rules" to T_04, T_09, T_10, T_11, and T_12. For example, if T_12 uses the FCFS rule, then the tokens in P_14 (denoted as the wafers waiting for robot transfer) are released (i.e., dispatched physically) in terms of their entrance sequence. Similarly, if T_12 uses the "shortest processing time" rule, then the tokens in P_14 are released in terms of the shortest next operation times for the waiting wafers. Therefore, the rules for T_04, T_09, T_10, T_11, and T_12 transitions can be adjusted to fit the practical control policies.

The conflict condition may occur at P_03, P_07, P_15, and P_16 places. The production mechanism for this tool can be designed by applying the "conflict transition resolution rules" to these places. For example, if P_07 uses the "reference transition name" rule, then the only transition that can be fired is selected from the enabled transitions for T_05, T_06, T_07, and T_08 (denoted as process chamber A, B, C, and complete operation in this tool, respectively). In this case, each transition has a specified name. If the "next operation chamber ID" attribute for the releasing token matches any transition name, then this transition can be fired. This is important for the route-determined process chambers. In addition, if P_03 uses the "highest conflict transition priority" rule, then the transition of the highest priority for T_02 and T_03 is fired when they are both enabled. Note that P_07 is important because it is an internal buffer. P_07 receives tokens from T_04 with the arc cardinality (QTY, it is the number of wafer pieces for a pod/lot that is attached in the color sets of the releasing token) as new entry parts, and it also receives tokens from T_13, T_14, and T_15 as the tool reentry parts.

The input function used in T_17 and T_18 enhance the product model mix capability. T_17 and T_18 transitions are the output transitions for P_16, and they are essentially in conflict. In this model, the enabling and firing conditions for T_17 are: 1) P_16 must contain at least QTY tokens; 2) P_04 must contain one token; and 3) the "*lot ID*" attribute for the token in P_04 and the QTY tokens in P_16 must all be identical. Therefore, even when two different lots are mixed in the same tool, the lots can be separated when all processes are complete according to their original lots. In addition, the initial marking for this tool model is also indicated as the tokens (marked as ready for each process chamber) in P_08, P_09, and P_13.

When the basic model library is constructed, the DCTPN system models can be established from the tool model to the tool group model, from the tool group model to the area model, and from the area model to the fab plant model. Fig. 4(a) shows their relationships. They constitute a plant-level DCTPN fab model, as shown in Fig. 4(b). The places



(a) (b)

Fig. 4. Fab operations. (a) Operation flows of wafer lots. (b) Plant-level DCTPN model.

TABLE I
DESCRIPTIONS OF PLACES AND TRANSITIONS IN FIG. 4(b)

Icon ID	Type	Description
P_01	P_0	Command for releasing new order
P_02	P_d	COM Server place for releasing new order; it invokes remote MES server
P_03	P_0	Lots deciding the next one operation area or completion
P_04/ P_05	P_0	Valid/ Invalid order (lot)
P_06/ P_07/ P_08 P_03/ P_04/ P_15	P_c	Lots delivering to CVD/ ETCH/ PHOTO/ IMP/ DIF/ SPUT area level model
P_09/ P_10/ P_11 P_16/ P_17/ P_18	P_c	Lots receiving from CVD/ ETCH/ PHOTO/ IMP/ DIF/ SPUT area level model
P_12	P_0	Lots collected from completed operations in six areas
T_02	T_t	Time interval for releasing new orders (inhibited by invalid orders in P_05)
T_03	T_0	Transfer lot order from order releasing server
T_04/ T_05	T_0	Check new order as valid/ invalid
T_06/ T_07/ T_08 T_17/ T_18/ T_19	T_0	Decide the next operation is CVD/ ETCH/ PHOTO/ IMP/ DIF/ SPUT in terms of lot route
T_12/ T_13/ T_14 T_23/ T_24/ T_25	T_0	Collect completed operations from CVD/ ETCH/ PHOTO/ IMP/ DIF/ SPUT areas
T_15	T_0	All processes are finished in terms of lot route
T_16	T_0	Add message to waiting lots
T_09/ T_10/ T_11 T_20/ T_21/ T_22	T_m	Macro transition indicates a net refinement of area CVD/ ETCH/ PHOTO/ IMP/ DIF/ SPUT

and transitions in Fig. 4(b) are further described in Table I. Such a resource-oriented modeling approach is clear and reconfigurable [14]. However, such a DCTPN fab model with detailed production dynamics description is impossible to do using the standard PN.

IV. SIMULATION RESULTS

A simplified fab model was constructed and analyzed using DCTPN. Six functional areas, 47 tools, and 19 tool groups are used. In order to examine the product-mix DCTPN model, six major product models with various routes are evaluated. Each product route contains more than 150 operations (tool groups). In this simulation model, 60 lots with six mixed product models are simulated. The production behaviors are generated using wafer dispatching rules (in terms of competitive tokens resolution rules) and tool dispatching rules (in terms of conflict transitions resolution rules). The resource-oriented fab DCTPN models are constructed as shown in Fig. 3 and Fig. 4(b). The remaining models are also constructed in the same manner. They are not illustrated in this paper. Note that all distributed models communicate with one another through the communication places. These models are constructed using the DCTPN real-time simulation environment [4], [7], [14]. In practice, the communication place is developed based on the NTU-NET communication protocol [15]. NTU-NET supports real-time, priority-control, and fault-tolerance communication libraries.

Several cases are discussed in this paper. The first case examines bottleneck elimination using additional new tools in a tool group. Fig. 5 shows a work-in-progress (WIP) chart for a tool group \$I_TG_2 (a tool group in the IMP area) based on two and four tool

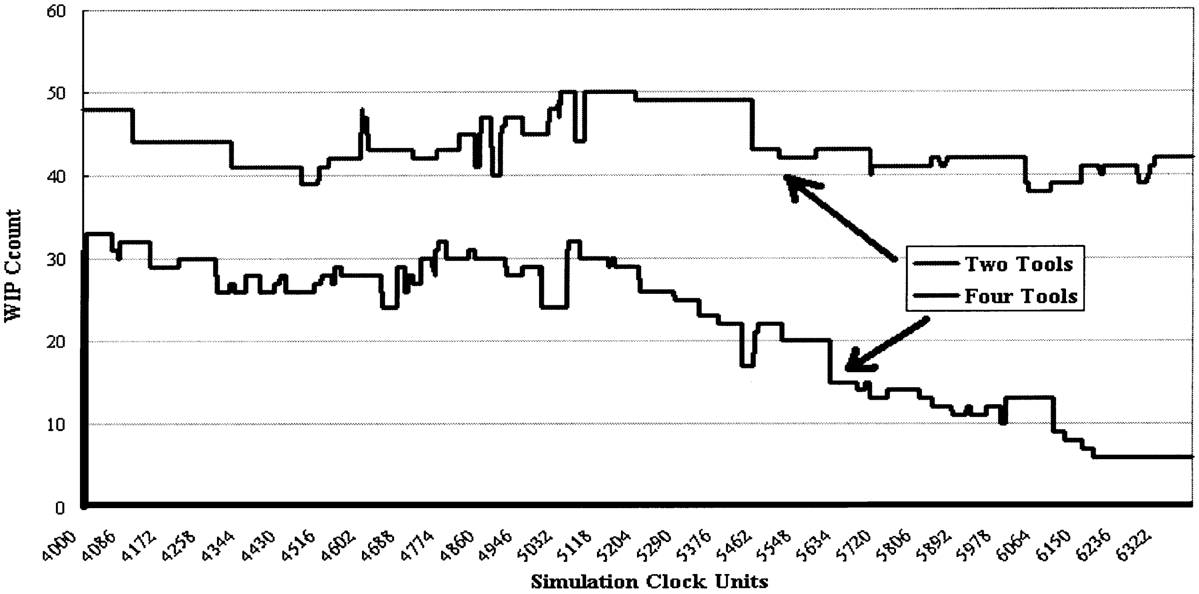


Fig. 5. Bottleneck elimination by adding new tools to a tool group.

TABLE II
PERFORMANCE EVALUATION USING DIFFERENT CONTROL POLICIES

Control Policy	Fab Operation Time (simulation units)	Operation Move (# of operations /1000 units)	Fab Throughput (# of lots /1000 units)	Average Cycle Time (simulation units)
Lot: Earliest Commit Due Date Tool: Sequential	20535	429.3	2.92	13904.65
Lot: Wafer Priority Tool: Sequential	20360	433.0	2.95	13941.50
Lot: First Come First Serve Tool: Minimum Operation Time	20829	423.2	2.88	17575.13

configurations. Note that the WIP count is measured in terms of the simulation clock units recorded from 4000 to 6400. At beginning, this tool group contains only two tools. Apparently, the value of waiting lots for this tool group is too high, as indicated in Fig. 5. If two additional tools are added to this tool group and the new model is resimulated again, the WIP is then reduced, as indicated in Fig. 5. In addition, the fab throughput is increased from 2.19 (lots/1000 simulation units) to 2.92 (lots/1000 simulation units).

The second case demonstrates production control using the “satisfying commit due date of lots” and the “sequential use of redundant tools” rules. Such a control aims to push the lots with earlier commit due dates. The third case uses the “selecting highest lot priority” and the “sequential use of redundant tools” control policies. Such a control aims to push the lots with higher manufacturing priority. The final case uses the “FCFS lot” and the “selecting the minimum redundant tool operation time” control policies. Such a control aims to balance the utilization of redundant tools. The performance evaluations, including throughput measures, operation move, average cycle time, total production time, and tool utilization for these cases are given in Table II. Due to high product mixes and flexible routes, the performance evaluation strongly depends on the product combinations. Therefore, DCTPN actually provides such information to determine the best control policies when various products are mixed.

V. CONCLUSIONS

This paper proposed a performance evaluation and cycle time estimation for a controlled IC foundry fab using DCTPN. Since DCTPN provides attribute-based color sets, color set-based time function, and arc cardinality, it can model the fab with product mixes and flexible routes in a precise way. The remote MES invocations and the distributed model communications consequently extend the DCTPN capability. An entire fab model is constructed in terms of the DCTPN tool model library. Such a fab model can be controlled using the “competitive tokens” and the “conflict transitions” resolution rules, and further evaluated in terms of the DCTPN simulation results. Several cases involving a six-area fab were discussed. The cycle time estimation of products and performance evaluation of the fab were illustrated.

REFERENCES

[1] I. B. Adballah, H. E. I. Maraghy, and T. E. I. Mekkawy, “An efficient search algorithm for deadlock-free scheduling in FMS using Petri nets,” in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 2, Leuven, Belgium, May 1998, pp. 1793–1798.
[2] C. G. Cassandras, *Discrete Event Systems—Modeling and Performance Analysis*. Homewood, IL: Irwin/Aksen, 1993.
[3] G. Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems—Concepts and Design*. Reading, MA: Addison-Wesley, 2001.

[4] *DCTPN User Manual Version 1.0*. Taipei, Taiwan: Robotics Lab., Dept. Mech. Eng., Nat. Taiwan Univ., 1999.

[5] A. A. Desrochers and R. Y. AI-Jaar, *Applications of Petri Net in Manufacturing Systems—Modeling, Control, and Performance Analysis*. New York: IEEE Press, 1995.

[6] G. H. Hu, Y. S. Wong, and H. T. Loh, "An FMS scheduling and control decision support system based on generalized stochastic Petri net," *Int. J. Adv. Manufact. Technol.*, vol. 10, pp. 52–58, 1995.

[7] H. P. Huang and Y. H. Tseng, "Modeling and graphic simulator for integrated manufacturing systems," *Intell. Automat. Soft Comput.*, vol. 1, pp. 183–186, 1994.

[8] M. D. Jeng, X. Xie, and S. W. Chou, "Modeling, qualitative analysis, and performance evaluation of the etching area in an IC wafer fabrication system using Petri net," *IEEE Trans. Semiconduct. Manufact.*, vol. 11, pp. 358–373, Mar. 1998.

[9] M. D. Jeng, "A Petri net synthesis theory for modeling flexible manufacturing systems," *IEEE Trans. Systems, Man, Cybern. B*, vol. 27, pp. 169–183, Apr. 1997.

[10] C. H. Kuo, H. P. Huang, K. C. Wei, and S. S. H. Tang, "System modeling and real-time simulator for highly model-mixed assembly systems," *ASME J. Manufact. Sci. Eng.*, vol. 121, pp. 282–289, Feb. 1999.

[11] C. H. Kuo and H. P. Huang, "Modeling and performance evaluation of a controlled IC fab using distributed colored timed Petri net," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 3, San Francisco, CA, 2000, pp. 2191–2196.

[12] C. H. Kuo, H. P. Huang, and M. C. Yeh, "Object-oriented approach of MCTPN for modeling flexible manufacturing systems," *Int. J. Adv. Manufact. Technol.*, vol. 14, pp. 737–749, 1998.

[13] C. H. Kuo and H. P. Huang, "Failure modeling and process monitoring for flexible manufacturing systems by using colored timed Petri net," *IEEE Trans. Robot. Automat.*, vol. 16, pp. 301–312, June 2000.

[14] ——, "Reconfigurable cluster tool performance enhancement using distributed modular-based modeling technology," in *Proc. Int. Conf. Automation Technology*, vol. 1, Taipei, Taiwan, 2000, pp. 303–310.

[15] L. R. Lin and H. P. Huang, "Real-time networking for the implementation of CIM," in *Proc. Int. Conf. Automation Technology*, vol. 1, Taipei, Taiwan, 1996, pp. 21–28.

[16] S. Y. Lin and H. P. Huang, "Modeling and emulation of a furnace in IC fab based on colored timed Petri net," *IEEE Trans. Semiconduct. Manufact.*, vol. 11, pp. 410–420, Mar. 1998.

[17] S. S. Lu and H. P. Huang, "Modularization and properties of flexible manufacturing systems," in *Advances in Factories of the Future, CIM and Robotics*, M. Cotsaftis and F. Vernadat, Eds. Amsterdam, The Netherlands: Elsevier, 1993, pp. 289–298.

[18] T. Murata, "Petri nets: properties, analysis and applications," *Proc. IEEE*, vol. 77, pp. 541–580, Apr. 1989.

[19] N. Wu, "Necessary and sufficient conditions for deadlock-free operation in flexible manufacturing systems using a colored Petri net model," *IEEE Trans. Systems, Man, Cybern.*, vol. 29, pp. 192–204, Feb. 1999.

[20] M. C. Zhou and M. D. Jeng, "Modeling, analysis, simulation, scheduling, and control of semiconductor manufacturing systems: a Petri net approach," *IEEE Trans. Semiconduct. Manufact.*, vol. 11, pp. 333–357, Mar. 1998.

[21] M. C. Zhou and K. Venkatesh, *Modeling, Simulation, and Control of Flexible Manufacturing Systems—A Petri Net Approach*, Singapore: World Scientific, 1999.