IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC'2000); Nashville, Tennessee, 8–11 Oct. 2000, pp.3063–3068.

Timed Petri Net Models of Cluster Tools

W.M. Zuberek
Department of Computer Science
Memorial University of Newfoundland
St.John's, Canada A1B 3X5

Abstract

Timed Petri nets are used as models of cluster tools, representing not only the concurrency activities of different chambers, but also the durations of these activities. Structural analysis, based on net invariants, provides basic performance measures, such as throughput and cycle time. The results are obtained in symbolic form, so many specific variants can easily be compared without repeated analyses of the models.

1. Introduction

A cluster tool is an integrated, environmentally isolated manufacturing system consisting of process, transport, and cassette modules, mechanically linked together [2]. Process modules, also called chambers, perform specific technological operations on consecutive wafers which are moved by a robotic transporter from one chamber to another. The factors which stimulate an increased use of clustered tools in recent years include improved yield and throughput, reduced contamination, better utilization of the floor space, and reduced human intervention [7].

Because of high throughput requirements, cluster tools perform a number of activities concurrently, for example, different wafers are processed in different chambers at the same time, and also the robotic transporter can be moving to a position required by the next step. Petri nets [5, 3] are formal models developed specifically for representation of concurrent activities and for coordination of such activities, i.e., for ordering specific actions or for performing actions simultaneously by more than one component of a system. Petri nets are often called 'condition–event systems' because their two type of basic elements, called places and transitions, represent the (satisfied or unsatisfied) conditions of some events, and the events which can occur only when all conditions associated with them are satisfied. Formally, Petri nets are represented by bipartite graphs (i.e., graphs with two types of vertices, one representing places, and the other transitions), and directed arcs connecting these two types of vertices. The dynamic behavior of nets is represented by the so called tokens associated with places. The distribution of these tokens can change (as an occurrence of some events), representing the sequences of events occurring in the modeled systems.

In order to analyze the performance of modeled systems, the durations of all activities must also be taken ito account. Several types of nets "with time" have been proposed by associating "time delays" with places [6], or occurrence durations with transitions [1, 11] of net models. Also, the introduced temporal properties can be deterministic [6, 11], or can be random variables described by probability distribution functions (the negative exponential distribution being probably the most popular choice) [1, 11].

Analysis of timed net models based on their behavior (represented by the set of states and transitions between states) is known as reachability analysis. For complex models, the exhaustive reachability analysis can easily become difficult because of very large number of states (for some models the number of states increases exponentially with the size of the model, which is known as the "state explosion problem"). Several approaches can be used to deal with the excessive numbers of states. One approach reduces the number of states by using state aggregation (i.e., by combining groups of states into single 'superstates'); another uses symmetries of the state space. For some classes of net models, the performance properties can be derived from the structure of the net models; this approach is known as structural analysis. The most popular example of this approach is analysis based on place–invariants (or P–invariants) for models covered by families of simple cyclic subnets (which are implied by P–invariants).

The paper develops timed Petri net models of cluster tools, including their steady–state as well as transient behaviors. For the steady–state, the performance of the model is obtained by structural analysis, so it is an alternative approach to the one presented in [8]. Transient behaviors are analyzed by tracing the sequences of reachable states. The approach presented in this paper is derived from earlier work on modeling and analysis of schedules for manufacturing cells [12].

2. Cluster Tools

The cluster tools analyzed in this paper are m–chamber cluster tools with one robotic transporter. Each of the chambers performs a unique process, and there is a single chamber for each process. The only explicit storage facility is the loadlock. For single–blade tools, the robotic transporter can carry only one wafer at a time. The model assumes that all wafers have the same process sequence, and that no chambers are revisited, as in [4].

A sketch of a 4–chamber cluster tool is shown in Fig.1, where LL denotes the loadlock to store cassettes of wafers; C1, C2, C3 and C4 are process chambers which modify the properties of the wafers, and R is a robotic transporter (or simply a robot) which moves the wafers between the loadlock and the chambers as well as from one chamber to another.

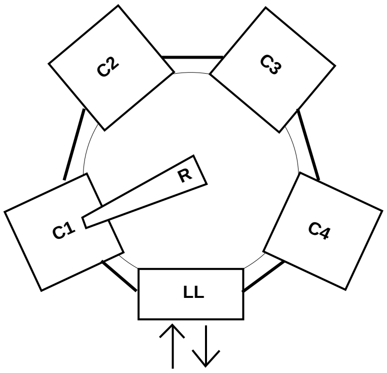

Fig.1. An outline of a 4–chamber cluster tool.

When a batch of wafers arrives at an empty cluster tool, it is placed in the loadlock which is then typically pumped down to vacuum. All the time required to get a batch into the cluster and ready for processing is denoted as τ_{load}. The robot, assumed to be idle at the loadlock, moves the first wafer to the first chamber. For simplicity, it is assumed that the chambers are numbered as they appear in the process sequence. When the process in the first chamber is finished, the wafer is moved to the second chamber, after which the second wafer can be moved into the first chamber. After a number of such wafer transports, the first wafer arrives back at the loadlock. When all wafers have been processed and returned to the loadlock, the loadlock is raised to atmospheric pressure and the batch is removed. The time interval between when the last wafer arrives at the loadlock and when the batch is removed is denoted as τ_{unload}.

In general, the time to process a batch consists of the following [4]: τ_{load}, the time τ_{init} to reach steady state, the time spent in steady state τ_{steady}, the time τ_{end} to process final wafers, and τ_{unload}, as sketched in Fig.2.

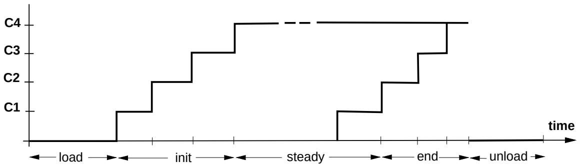

Fig.2. A sketch of batch processing.

The initial transient period τ_{init} is due to the fact that the tool is empty at the beginning of each batch, and the final transient period τ_{end} processes the final wafers until the tool becomes empty. The steady part of the batch processing typically includes $n-m+1$ identical cycles, where n is the number of wafers in the batch, and m – the number of chambers.

3. Steady–state behavior

The model for the steady–state behavior is the simplest one, so it is discussed first. Several elements of the cluster tool can be ignored for steady state considerations (e.g., the loadlock). Moreover, it is assumed that all chambers are used concurrently, i.e., when the i-th wafer is moved to chamber 1, the $(i-1)$-th wafer is processed in chamber 2, and $(i-2)$-th wafer is processed in chamber 3, and so on. The sequence of the operations in each cycle is as follows (it is assumed that the cycle begins when a new wafer is moved to chamber 1):

- pick next wafer from loadlock, transport it to chamber 1 and load it; chamber 1 can start its process;
- move to chamber 4, unload the wafer (when ready), transport it to loadlock and drop it there;
- move to chamber 3, unload the wafer (when ready), transport it to chamber 4 and load it; chamber 4 can start its process;
- move to chamber 2, unload the wafer (when ready), transport it to chamber 3 and load it; chamber 3 can start its process;
- move to chamber 1, unload the wafer (when ready), transport it to chamber 2 and load it; chamber 2 can start its process;
- return to loadlock to begin another cycle.

A Petri net modeling this sequence of operations is shown in Fig.3.

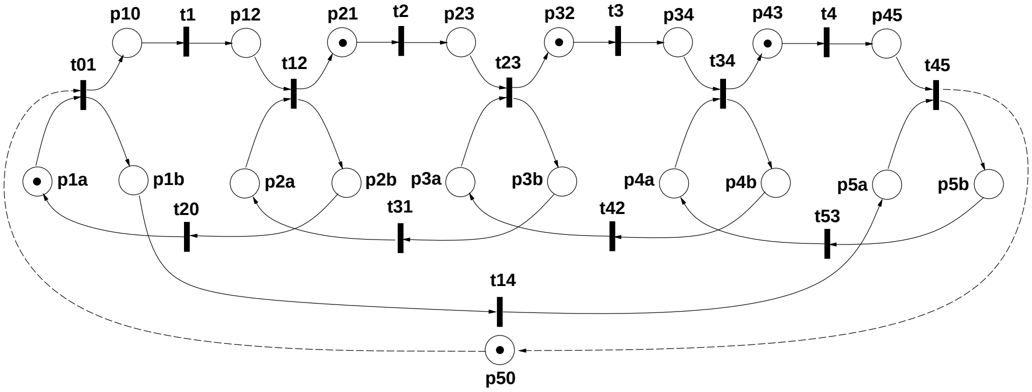

Fig.3. Petri net model for the steady-state behavior.

The model shown in Fig.3 contains four sections modeling the four chambers, each represented by one transition (t_1, t_2, t_3 and t_4, respectively). Each of these transitions has one input and one output place (representing the conditions "wafer loaded in chamber" and "wafer ready for unloading"). The remaining part represents the sequence of steps of one complete cycle of the robot. This sequence begins (as indicated by the initial marking) by picking a wafer from the loadlock (transition t_{01}). The initial marking of places p_{21}, p_{34} and p_{43} indicates that the chambers (except of chamber 1), in the moment of picking a new wafer from the loadlock, are loaded with (previous) wafers.

The operations represented by transitions are as follows:

transition	*operation*
t_{01}	pick next wafer from the loadlock, move it to chamber 1 and load;
t_{14}	move the robot to chamber 4;
t_{45}	unload the wafer from chamber 4, move it to loadlock and drop;
t_{53}	move the robot to chamber 3;
t_{34}	unload the wafer from chamber 3, move it to chamber 4 and load;
t_{42}	move the robot to chamber 2;
t_{23}	unload the wafer from chamber 2, move it to chamber 3 and load;
t_{31}	move the robot to chamber 1;
t_{12}	unload the wafer from chamber 1, move it to chamber 2 and load;
t_{20}	move the robot to loadlock.

In order to obtain the effect of steady–state, place p_{50} is used as "input" and "output" of the cluster tool. When a wafer is finished, a token is deposited in p_{50}, and the same token is used as the next wafer a moment later. The initial marking of p_{50} is irrelevant (as long as it is nonzero), and the behavior is exactly the same if more than one token is assigned initially to p_{50}. Moreover, it can be shown that this place has no effect on the performance of the model, so it can be removed (with the two arcs shown in broken lines).

All transitions are timed transitions, and the occurrence times associated with them represent the (average) times of the corresponding operations.

The net shown in Fig.3, after removal of place p_{50}, has 5 P–invariants; the sets of transitions of subnets implied by these P–invariants are:

invariant	*set of transitions*
1	$t_1, t_{01}, t_{12}, t_{20}$
2	$t_2, t_{12}, t_{23}, t_{31}$
3	$t_3, t_{23}, t_{34}, t_{42}$
4	$t_4, t_{34}, t_{45}, t_{53}$
5	$t_{01}, t_{14}, t_{45}, t_{53}, t_{34}, t_{42}, t_{23}, t_{31}, t_{12}, t_{20}$

It is known [6] that if a net is covered by a family of conflict–free cyclic subnets, the cycle time of the net is equal to the maximum cycle time of the covering subnets:

$$\tau_0 \ = \ \max(\tau_1, \tau_2, ..., \tau_k)$$

where k is the number of subnets covering the original net, and each τ_i, $i = 1, ..., k$ is the cycle time of the subnet i, which is equal to the sum of occurrence times associated with the transitions, divided by the total number of tokens assigned to the subnet:

$$\tau_i \ = \ \frac{\sum_{t \in T_i} f(t)}{\sum_{p \in P_i} m(p)}.$$

Since the P–invariant implied subnets cover the model shown in Fig.3, its cycle time τ_0 is:

$$\tau_0 \ = \ \max(\tau_1, \tau_2, \tau_3, \tau_4, \tau_5)$$

where τ_i denotes the cycle time of the subnet i:

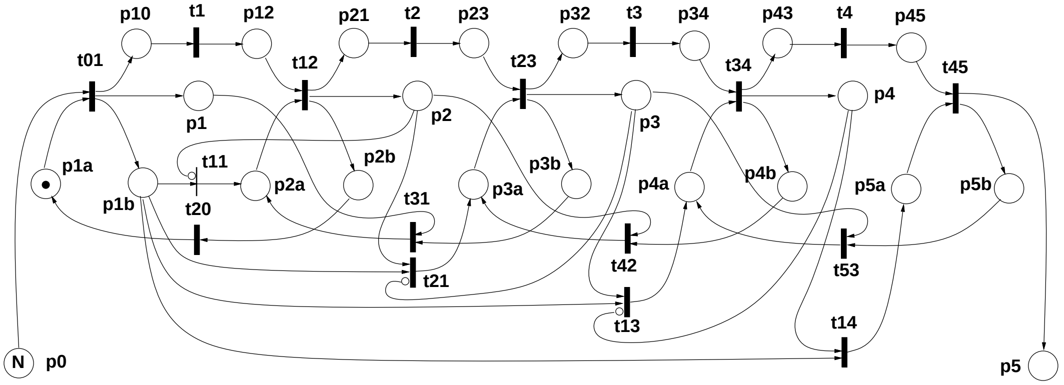

Fig.4. Petri net model for the initial transient behavior.

$$\begin{aligned}
\tau_1 &= f(t_1) + f(t_{01}) + f(t_{12}) + f(t_{20}),\\
\tau_2 &= f(t_2) + f(t_{12}) + f(t_{23}) + f(t_{31}),\\
\tau_3 &= f(t_3) + f(t_{23}) + f(t_{34}) + f(t_{42}),\\
\tau_4 &= f(t_4) + f(t_{34}) + f(t_{45}) + f(t_{53}),\\
\tau_5 &= f(t_{01}) + f(t_{14}) + f(t_{45}) + f(t_{53}) + f(t_{34}) +\\
&\quad f(t_{42}) + f(t_{23}) + f(t_{31}) + f(t_{12}) + f(t_{20}).
\end{aligned}$$

If τ_0 is equal to any one of the first four terms, the model is called "process bound" because the duration of the process performed by the corresponding chamber determines the cycle time (and the throughput) of the tool; if the cycle time is equal to the last term, the model is called "transport bound" [9].

4. Initial Transient Behavior

The initial transient behavior is due to the fact that the chambers, at the beginning of each batch, are empty. Consequently, the sequence of operations is slightly different at the beginning of the batch than in the steady state, and this difference must be captured by the model. Fig.4 shows the model representing the initial transient behavior (as an extension of the steady–state model).

The extension of the model shown in Fig.4 includes additional places p_1, ..., p_4, which indicate (if marked) that the chamber is loaded. These additional places are tested (by inhibitor arcs connected to transitions t_{11}, t_{21} and t_{13}) to "shorten" the robot's cycle an eliminate its visits to chambers which are empty.

The initial sequence of operations is described by the following steps (with the corresponding transitions):

transition	*operation*
t_{01}	pick a wafer from the loadlock,
	move it to chamber 1 and load;
t_{11}	wait for the end of operation;
t_{12}	unload the wafer from chamber 1,
	move it to chamber 2 and load;
t_{20}	move the robot to loadlock;
t_{01}	pick a wafer from the loadlock,
	move it to chamber 1 and load;
t_{21}	move the robot to chamber 2;
t_{23}	unload the wafer from chamber 2,
	move it to chamber 3 and load;
t_{31}	move the robot to chamber 1;
t_{12}	unload the wafer from chamber 1,
	move it to chamber 2 and load;
t_{20}	move the robot to loadlock;
t_{01}	pick a wafer from the loadlock,
	move it to chamber 1 and load;
t_{13}	move the robot to chamber 3;
t_{34}	unload the wafer from chamber 3,
	move it to chamber 4 and load;
t_{42}	move the robot to chamber 2;
t_{23}	unload the wafer from chamber 2,
	move it to chamber 3 and load;
t_{31}	move the robot to chamber 1;
t_{12}	unload the wafer from chamber 1,
	move it to chamber 2 and load;
t_{20}	move the robot to loadlock.

The duration of this transient behavior can be evaluated from a slightly modified Petri net model. It can be shown [11] that the graph of reachable states for any conflict–free bounded net can only be a straight path (if the behavior is finite) or a path with a cycle (if the behavior is infinite); the cycle in this case represents the steady state behavior of the model. So, the model shown in Fig.4 needs to be slightly modified to create its cyclic, infinite behavior representing the

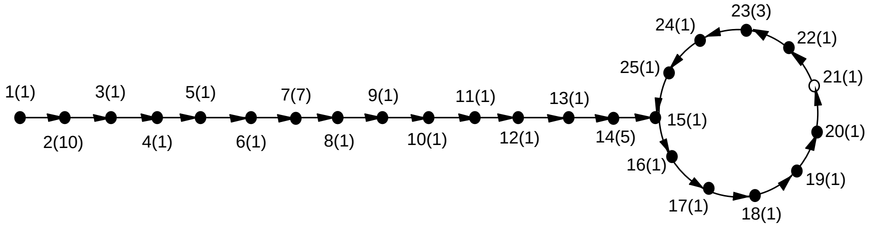

Fig.5. State graph for the modified net shown in Fig.4.

steady–state. This can be done by 'merging' places p_0 and p_5. Then, however, it can be observed that, similarly as for the net shown in Fig.3, the merged place can be deleted (together with its arcs) as it has no effect on the performance of the model. Such modified model can then be analyzed for the initial transient behavior that determines the time τ_{init}.

For example, assuming (just for the sake of this example) that $f(t_1) = f(t_2) = f(t_3) = f(t_4) = 10$, and all other timed transitions have the associated time equal to 1, the state graph corresponding to the (modified) net of Fig.4 is shown in Fig.5, where the time spent in each state is given in parentheses. State 21 is the state in the cyclic part in which the next wafer is fetched from the loadlock, so $\tau_{init} = 26$ (the sum of times associated with states 1 to 20 minus the cycle time), while the cycle time $\tau_0 = 13$ (the sum of times associated with the cyclic states).

The steady–state solution (discussed earlier), for the same timing values, results in $\tau_1 = \tau_2 = \tau_3 = \tau_4 = 13$ and $\tau_5 = 8$, so $\tau_0 = 13$, and the model is clearly process bound.

5. Final Transient Behavior

The final part of processing a batch of wafers is also different from the steady–state because there are no more wafers in the loadlock, but the wafers remaining in the chambers must complete their operations and be transported back to the loadlock. The robot's operations in this case are restricted to fewer machines in each remaining "cycle", first M1, M2, M3 and M4, then M2, M3 and M4, then only M3 and M4, and finally just M4. The Petri net model for this part of processing is shown in Fig.6.

The initial marking function shown in Fig.6 corresponds to a transport–bound case of cluster tool; for a process-bound case, the tokens should be assigned to p_{21}, p_{32} and p_{43} rather than to p_{23}, p_{34} and p_{45}.

The sequence of required operations begins with the last wafer picked from the loadlock:

transition	*operation*
t_{01}	pick a wafer from the loadlock, move it to chamber 1 and load;
t_{14}	move the robot to chamber 4;
t_{45}	unload the wafer from chamber 4, move it to loadlock and drop;
t_{53}	move the robot to chamber 3;
t_{34}	unload the wafer from chamber 3, move it to chamber 4 and load;
t_{42}	move the robot to chamber 2;
t_{23}	unload the wafer from chamber 2, move it to chamber 3 and load;
t_{31}	move the robot to chamber 1;
t_{12}	unload the wafer from chamber 1; move it to chamber 2 and load,
t_{2e}	move the robot to chamber 4;
t_{45}	unload the wafer from chamber 4, move it to loadlock and drop;
t_{53}	move the robot to chamber 3;
t_{34}	unload the wafer from chamber 3, move it to chamber 4 and load;
t_{42}	move the robot to chamber 2;
t_{23}	unload the wafer from chamber 2, move it to chamber 3 and load;
t_{3e}	move the robot to chamber 4;
t_{45}	unload the wafer from chamber 4, move it to loadlock and drop;
t_{53}	move the robot to chamber 3;
t_{34}	unload the wafer from chamber 3, move it to chamber 4 and load;
t_{44}	wait for the end of operation
t_{45}	unload the wafer from chamber 4; move it to loadlock and drop;
t_{55}	end of batch processing

The duration of the final transient behavior can be estimated on the basis of required operations, or it can be obtained from analysis of the net model (Fig.6). In this case no modification is needed; the behavior of the model in Fig.6 is finite, so it directly determines the time τ_{end}. For the timing information used in Section 4, the final transient behavior is $\tau_{end} = 32$.

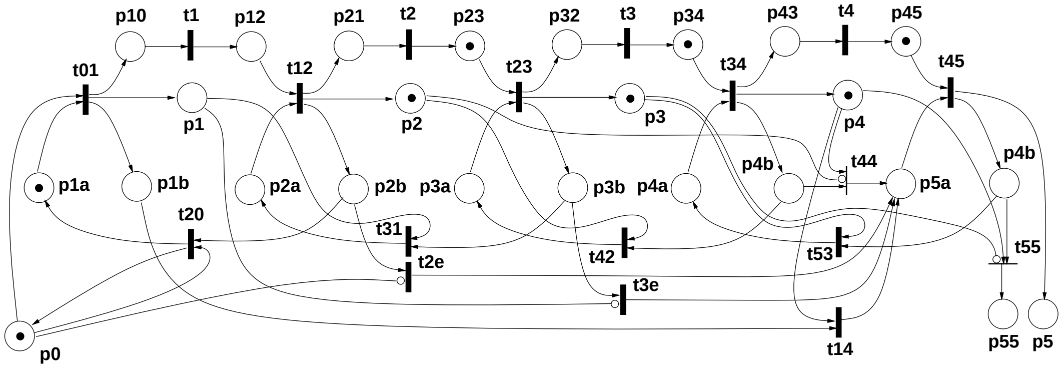

Fig.6. Petri net model for the final transient behavior.

6. Concluding Remarks

A modeling approach is proposed which can be applied to a variety of cluster tools, with different configurations and different characteristics. A complete performance analysis can easily be performed, including steady-state as well as transient behaviors. The performance characteristics are derived in symbolic form, which provides very efficient analysis of specific configurations, described by sets of numerical parameters.

Symbolic results derived from analysis of net models correspond directly to the fixed and incremental cycle time proposed in [10], where the (average) time required for processing a batch of n wafers is characterized by two parameters, τ_{fixed}, the 'fixed cycle time', and τ_0, the incremental time per one wafer during the steady–state behavior:

$$\tau_{batch} \;=\; \tau_{fixed} + n\tau_0$$

The only significant difference with respect to formulas derived earlier in this paper is that the simple formula of [10] does not take the transient behaviors into account, so it underestimates the batch processing time.

References

[1] Ajmone Marsan, M., Conte, G., Balbo, G., "A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems"; ACM Trans. on Computer Systems, vol.2, no.2, pp.93–122, 1984.

[2] Burggraaf, P., "Coping with the high cost of wafer fabs"; Semiconductor International, vol.18, no.3, pp.45-50, 1995.

[3] Murata, T., "Petri nets: properties, analysis and applications"; Proceedings of IEEE, vol.77, no.4, pp.541–580, 1989.

[4] Perkinson, T.L., MacLarty, P.K., Gyurcsik. R.S., Cavin III, R.K., "Single-wafer cluster tool performance: an analysis of throughput"; IEEE Trans. on Semiconductor Manufacturing, vol.7, no.3, pp.369-373, 1994.

[5] Reisig, W., "Petri nets – an introduction"; Springer–Verlag 1985.

[6] Sifakis, J., "Use of Petri nets for performance evaluation"; in: "Measuring, modeling and evaluating computer systems", pp.75–93, North–Holland 1977.

[7] Singer, P., "The driving forces in cluster tool development"; Semiconductor International, vol.18, no.8, pp.113-118, 1995.

[8] Srinivasan, R.S., "Modeling and performance analysis of cluster tools using Petri nets"; IEEE Trans. on Semiconductor Manufacturing, vol.11, no.3, pp.394-403, 1998.

[9] Venkatesh, S., Davenport, R., Foxhoven, P., Nulman, J., "A steady–state throughput analysis of cluster tools: dual–blade versus single-blade robots"; IEEE Trans. on Semiconductor Manufacturing, vol.10, no.4, pp.418–423, 1997.

[10] Wood, R., "Simple performance models for integrated processing tools"; IEEE Trans. on Semiconductor Manufacturing, vol.9, no.3, pp.320–328, 1996.

[11] Zuberek, W.M., "Timed Petri nets – definitions, properties and applications"; Microelectronics and Reliability, vol.31, no.4, pp.627–644, 1991.

[12] Zuberek, W.M., Kubiak, W., "Timed Petri nets in modeling and analysis of simple schedules for manufacturing cells"; Journal of Computers and Mathematics with Applications, vol.37, no.11/12, pp.191–206, 1999.