Modelling, control and simulation of an IC wafer fabrication system: a generalized stochastic coloured timed Petri Net approach

MING-HUNG LIN† and LI-CHEN FU*

This study presents a generalized stochastic coloured timed Petri net(GSCTPN) to model an IC wafer fabrication system. According to the GSCTPN, it models the dynamic behaviours of the IC fabrication system, such as loading, reentrant processing, unloading and machine failure. Furthermore, modular and synthesis techniques are used to construct a large and complex system model. The two major sub-models are the Process-Flow Model and the Transportation Model. The Transportation Model incorporates a simple motion-planning rule and a collision avoidance strategy to solve the variable speed and traffic jam problems of vehicles. This work also describes a simulation based performance analysis and schedule adjustment. To demonstrate the promise of the proposed work, this study makes actual Taiwanese IC wafer fabrication systems the target plant layout for implementation.

1. Introduction

Wafer fabrication is the most expensive phase of semiconductor manufacturing (Sze 1983, Wein *et al.* 1988, Uzsoy *et al.* 1992). Significant risk is involved in wafer fabrication because of enormous investment costs. To survive such a competitive and risky environment, the company must not only improve quality and throughput but also satisfy the demands of customers. If product delivery is frequently late, the company loses customer and market goodwill, ultimately influencing long-term sales opportunities. Additionally, wafer fabrication involves complex processing with operations typically numbering in the hundreds, extending production cycle time. Long production cycle time increases the expense of wafer fabrication because it adversely influences product yield, and the ability to predict the market trends. Additionally, product life cycles are generally short in the semiconductor industry, and hence, finished goods inventories constantly risk obsolescence.

There are many comparative studies and reports (Leachman and Hodges 1996, Robertson and Gargini 1998), which have developed a systematic account of the practices that explain best manufacturing performance in semiconductor production on a world-wide basis. The advent of production 300mm wafer fabs is poised to occur at a time when wafer-processing technology is undergoing rapid changes. The International 300 mm Initiative (I300I) approach for transport of wafers specifies the front opening unified pod (FOUP) with a capacity of either 13 or 25 wafers (it is a user option as to which is implemented) (Plata 1997, Csatary *et al.* 1998). The size and weight of the 25 wafer lot/FOUP combination is almost 18 pounds (8.2kg),

Revision received January 2000

† Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, Republic of China.

* To whom correspondence should be addressed. e-mail: lichen@ccms.ntu.edu.tw

International Journal of Production Research ISSN 0020–7543 print/ISSN 1366–588X online © 2000 Taylor & Francis Ltd
http://www.tandf.co.uk/journals

which is beyond the ergonomic limit for repeated manual handling. When the cubic volume of the FOUP is considered, the combined size and weight become important factors for 300mm factory planning.

Automated material handling systems (AMHS) are utilized in almost all 200 mm fabs for transport of wafer lots between the stockers at each process bay ('Interbay transport'). Human operators use transport from the stockers to the process equipment ('Intrabay transport'). The Interbay system used most widely is of the overhead track (OHT) type. The Intrabay systems are such as AGVs (automated guided vehicles), RGVs (rail guided vehicles) and so on. It would be expected that the major change in AMHS will occur within the bay. The 300 mm fabs will make use of automated intrabay delivery to each process equipment. This results in minimized interference with operators and tool service/maintenance work on the intrabay material flow as well as narrow operator aisles. Thus, the material handing system's basic function is to remove the task of pod handling from the human operators. Reduced mishandling and misprocessing, operator and tool productivity gains, and enhanced material accountability are only a few of the benefits (Plata 1997, Csatary *et al.* 1998, Missale 1998).

The intrabay transport system still requires further development about positioning accuracy, reliability and operator safety issues. However, even with the use of sufficient safety measures at the vehicles, this can cause negative effects on operator efficiency due to psychological concern (Plata 1997). Service and maintenance work on the equipment front end can block the movement path and seriously affect wafer lot delivery to equipment ready to process material. The intrabay transport system can be more easily maintained compared to OHT systems since they operate on floor level. Clearly, a part transportation system is flexible and economical. However, it also produces numerous routing possibilities, complicating the decision problem. Since these IC wafer fabrication systems are dynamic, solutions should maximize the timeliness of their control schedule. It is desirable to respond fast to changes in their internal and external environments.

Machine failures are common in IC fabrication. Machine idleness due to failure hinders throughput and, ultimately, production costs. Therefore, most machines need regular maintenance to ensure high availability (Uzsoy *et al.* 1992, Johri 1993, Uzsoy *et al.* 1994, Duenyas *et al.* 1994, Leachman and Hodges 1996). However, the duration and frequency of periodic maintenance also influence performance. Thus, periodical maintenance is an important consideration in the system model.

Recent works by Uzsoy *et al.* (1992, 1994) Johri (1993), and Duenyas *et al.* (1994) highlight the difficulties in planning and scheduling wafer fabrication facilities and wafer queues. These works also survey the literature on related topics. Effective shop floor scheduling can significantly reduce cycle time. The benefits of effective scheduling include higher machine utilization, shorter cycle time, higher throughput rate, and greater customer satisfaction. This is particularly true in semiconductor manufacturing, with its rapidly changing markets and complex manufacturing process (Li *et al.* 1996, Narahari and Khan 1997, Kim *et al.* 1998a). Yet, in many wafer fabrications the product spends more time waiting than actually being processed, so there is great potential for reducing waiting time. Other issues related to wafer fabrication systems have also been considered, such as batch processing systems (Glassey and Weng 1991) and maintenance scheduling as well as staff policy (Mosle *et al.* 1998). However, since wafer fabrication is a complex discrete event system, schedules cannot easily be realized, and thus modelling a complex wafer fabrication manufacturing system becomes imperative. A good model not only helps schedule employment, but also helps monitor the status of lots and machines efficiently, which helps fine tune the scheduling policy.

Petri Net (PN) (Desrochers and Al-Jaar 1994) is important in the modelling. Petri-Net (Murata 1989) is a graphical and mathematical modelling tool applicable to many systems. Petri-Net can be used as a visual-communication aid similarly to flow charts, block diagrams, and networks. Moreover, tokens simulate the dynamic and concurrent activities of systems. Petri-Net is widely common in traditional computer science research, such as communication protocols (Diaz 1982), distributed systems (Ayache *et al.* 1982), compilers and operating systems (Baer and Ellis 1977). Finally, Petri-Net is highly effective for modelling (Narahari and Viswanadham 1985, Valavanis 1990), simulating (Valavanis 1990), and scheduling (Bispo *et al.* 1992) discrete event systems, such as flexible manufacturing systems. It is also useful for designing and implementing controllers for manufacturing (Zhou *et al.* 1990). In addition, some investigations have concentrated on performance analysis (Ramamoorthy and Ho 1980, Molly 1982, 1985, Hillion and Proth 1989). Nevertheless, the massive complexity of a real-world system tends to engender a large Petri-Net with numerous places and transitions. This kind of large net can be constructed by merging sub-nets. However, performing analysis through the reachability graph or invariance methods is nearly impossible. Hence, many researchers have developed a reduction technique (Lee andd Favrel 1985, Lee *et al.* 1987) capable of functioning as a synthesis method if appropriately applied. Moreover, high-level Petri-Net, such as coloured Petri-Net (Kasturia *et al.* 1988), or some extended Petri-Net (Valavanis 1990) can be used to overcome graphical complexity.

Using Petri-Net for modelling and performance analysis of IC wafer fabrication system is not new. Previous researches are discussed below. Zhou and Jeng (1998) proposed modelling and analysis of semiconductor manufacturing systems via Petri nets. Zhou reviewed applications of Petri-Net in semiconductor manufacturing automation. It proceeds to discuss the use of modules and a general method for constructing a system model. Timed Petri-Net are introduced for simulation, performance evaluation, and scheduling purposes. This can be served as a simple tutorial paper. Meanwhile, Srinivasan (1998) modelled the cluster tools in semiconductor manufacturing and developed an analytical technique that aims to find the state cycle and state period. Srinivasan has considered only deterministic firing delays. Nevertheless, modelling with stochastic firing delays is necessary for a practical plant.

Janneck and Naedele (1998) presented a tool called CodeSign and allowed object-oriented design and simulation of high-level timed Petri-Net models. The developed model is compared with a previous spreadsheet model in four configurations with actual measurements on the machine. Although the results show that a prediction error obtained from CodeSign is smaller than that from the spreadsheet model, the proposed modelling approach can not provide constructions for dynamic model structure. However, it is necessary to adopt abstraction concepts from modern programming languages, such as parametric components. Xiong and Zhou (1998) proposed and evaluated the Petri-Net based hybrid heuristic search strategies and their applications in semiconductor test facility scheduling. Xiong and Zhou described two Petri-Net based heuristic search strategies, called hybrid

BF-BT and hybrid BT-BF, for scheduling semiconductor test facility. However, in practical plant, it is necessary that a schedule can be fine tuned based on simulation to meet rapid changes in system parameters without taking long-run off-line scheduling again.

Allam and Alla (1998) proposed hybrid Petri-Net as an approximation of discrete Petri-Net to tackle the state explosion problem in analyzing large-scale semiconductor manufacturing systems. However, the hybrid Petri-Net is difficult to analyze. Additionally, Lin and Whung (1998) used the coloured timed Petri-Net (CTPN) to model the furnace in IC wafer fabrication. The dynamic behaviours of a furnace are emulated via the CTPN model. However, the CTPN model is not hierarchical and not easy enough to use. Jeng *et al.* (1998) reported a project of applying Petri-Net to detailed modelling, qualitative analysis, and performance evaluation of the etching area in an IC wafer fabrication system. This model deals with only one type of wafer. It is necessary to consider multiple types of wafers. Furthermore, the proposed model must extend the modular modelling approach and formally define the class of systems.

This work differs from others in many details. First, it proposes a systematic means of constructing a generalized stochastic coloured timed Petri-Net (GSCTPN) model for modelling an IC wafer fabrication system. However, designing a Petri-Net model for an IC wafer fabrication system is extremely tedious and has debugging difficulties. It is necessary to adopt abstraction concepts from modern programming languages, such as parametric components. This systematic modelling method can be implemented as an automatic Petri-Net generator (APNG). Using such generators, building a Petri-Net model of a FMS is no longer difficult and time consuming. The user merely needs to input the system information of a practical wafer fabrication system, such as the number of automated guided vehicles (AGVs), the number of machines and their geometric relationships, and the process flows of parts requiring processing. Furthermore, the proposed GSCTPN model considers multiple types of wafers. Second, the proposed GSCTPN model extends the modular modelling approach and formally defines the class of systems that can be dealt with using the approach. For multiple-load and variable-speed automated guided vehicle (AGV) systems, the GSCTPN model has a control rule already embedded into it to prevent vehicle collision problems. The transporting system again aims to introduce a control method into the Petri-Net model to ensure a jam-free condition among carriers. Third, this study describes a simulation based performance analysis and schedule adjustment. A schedule can be fine tuned based on simulation to meet rapid changes in system parameters without taking long run off-line scheduling again. The resulting net model is checked for important qualitative properties in IC fabrication systems. Simulation is used to obtain performance measures. The validated model can be used to answer many ‘what-if’ questions, such as predicting the throughput. To demonstrate the promise of the proposed work, a real-word IC wafer fabrication system located in Taiwan is used as a target plant layout for implementation.

The rest of this paper is organized as follows. Section 2 describes an IC wafer fabrication system. Section 3 introduces a GSCTPN systematic system modelling method. Section 4 presents simple control policies for AGV, lot release and lot scheduling. Section 5 describes simulation based performance analysis and schedule adjustment. Conclusions are finally made in Section 6.

2. System description

2.1. *IC wafer fabrication process*

IC wafer fabrication is a multi-stage process with reentrant flows. A typical wafer undergoes hundreds of processing steps using different machines over several weeks. These steps include oxidization, photo resistor coating, developing, etching, ion doping, chemical deposition, and diffusion. Generally, the machines in an IC wafer fabrication system are grouped into four functional sections: the *photolithography area*, the *diffusion area*, the *etching area*, and the *thin film area*.

The wafer processing unit in IC wafer fabrication is called a *lot*. When *lots* arrive at the IC factory, they are processed in the above mentioned four areas. The photo area arranges patterns of photo resistors on wafer. After processing in the photo area, wafers usually go to the etching area, where the part of wafers not covered by photo resistors is etched. The diffusion area and the thin film area deposit some material or implant ions on the wafers.

Figure 1 indicates that an IC plant is a base of *tunnels*. Tunnels in the same area are located in close proximity to each other to reduce wafer transportation time. Machines are sited beside the tunnels, and grouped into a *bay*. Generally, automated material handling system in IC wafer fabrication comprises two subsystems: *Interbay*

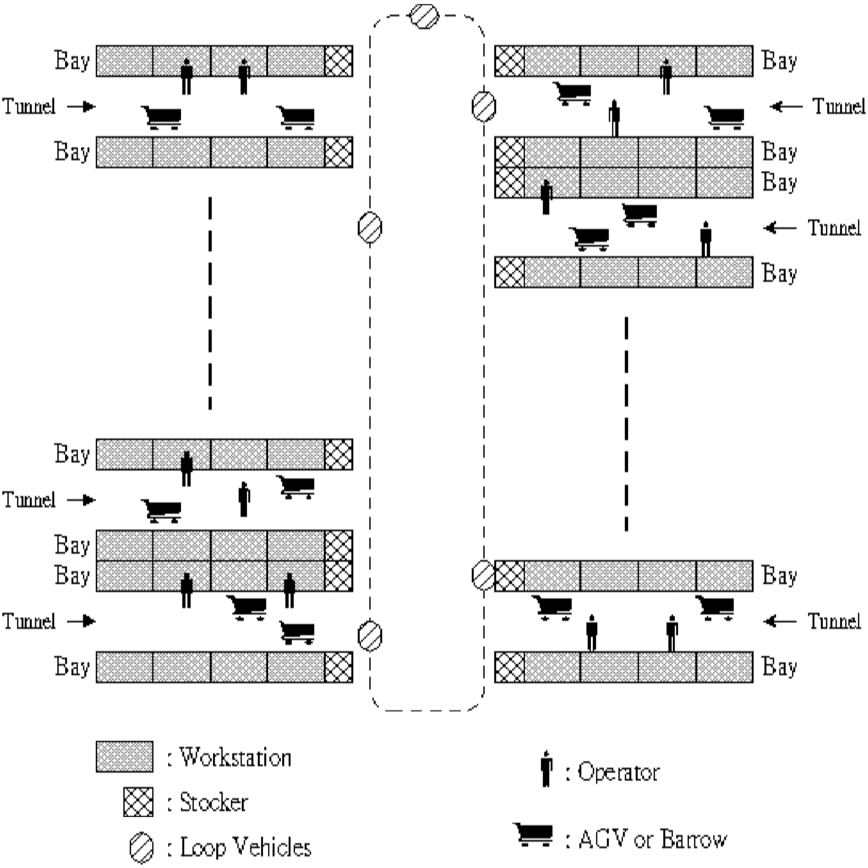


Figure 1. Physical layout.

system and *Intrabay system*. Multiple automated guided vehicles (AGVs) are used for wafer transportation. Each AGV carries multiple loads, and has variable speed. AGVs are allowed to accelerate in movement. In the Interbay system, AGV are called *loop vehicles* which can move to any stop unidirectionally. Each stop in Interbay serves a transit station transporting lots between different Intrabays. The transit stations are denoted as *Interbay stations*. However, the Intrabay system is composed of several independent tracks. Tracks in different area are not overlapping. Meanwhile, tracks in the Intrabay system are more comprehensive. When an AGV wants to move between stops within the same area, it has many routing choices. Meanwhile, three phases must be completed if the system wants to transport lots between stops located in different bays. First, an AGV transports lots to the Interbay station. Next, a loop vehicle takes the lots to the Interbay station of the destination bay. Finally, an AGV transports the lots to the destination stop. The performance of the AGV greatly affects the overall performance. Thus, the AGV dispatching, AGV routing and traffic jam problems of vehicles must be considered to maximize performance.

2.2. *Hierarchical view of flexible wafer fabrication system*

Figure 2 represents a hierarchical view of a flexible wafer fabrication system. This work develops a Petri-Net model for a wafer fabrication system and gives a simula-

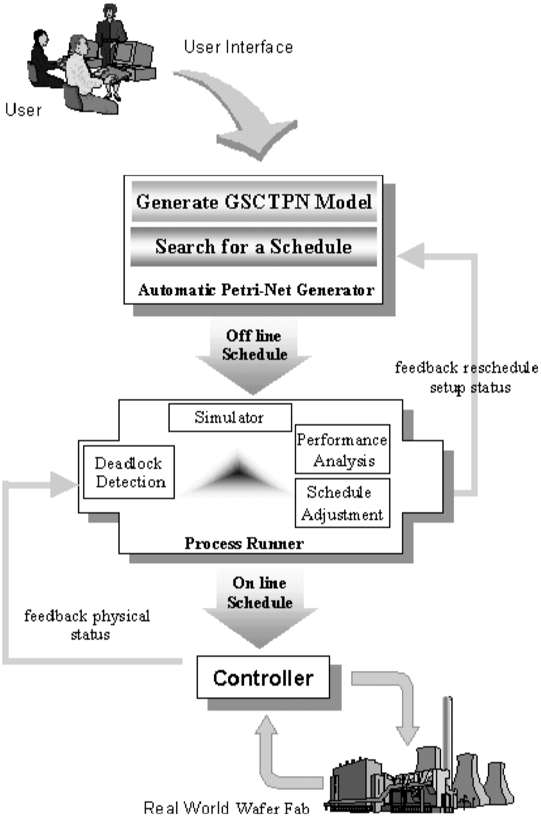


Figure 2. Hierarchical view for flexible wafer fabrication system.

tion-based performance analysis and schedule adjustment. The automatic Petri-Net generator (APNG) developed herein has a graphic user interface that permits users to input the specifications of a wafer fabrication system, based on which, the operator can create a GSCTPN model of the wafer fabrication system. Considering the Petri-Net model features, a feasible firing sequence can be found by following a path from the initial marking to the final marking. However, each marking of a Petri-Net represents a system state. Thus, the system time is continuously monitored in a timed place Petri-Net. Consequently, the transitions of firing sequence can represent part processing routes, shared resource use, and the timing of events in these events.

In the Petri-Net model, an off-line schedule for the system can be obtained using the queuing theory approach (Rogers and White 1991), mathematical programming (France 1982, Sarin *et al.* 1988, Agnetis *et al.* 1990, Hoitomt 1990, Luh 1990, Sawik 1990, Rogers and White 1991) heuristic algorithms (France 1982, Anderson and Nyirenda 1990, Mahmoodi *et al.*1990, Rao and Wang 1991), or the artificial intelligence approach (Chen *et al.* 1987, Shaw 1988, Lee and DiCesare 1992). In fact, this off-line schedule is not yet ready for running. Therefore, it is inputted to a so called *process runner* to eventually produce an on-line schedule which will drive the controller to control the real world wafer fabrication system. Therefore, the entire structure Figure 2 depicts constitutes a dynamic scheduler. Notably, the *process runner* contains three major components, (1) a schedule adjuster, used to adjust the off-line schedule to cope with events such as the delay of an AGV arrival or a machine break down; (2) simulator, used to simulate the environment of the real world wafer fabrication system and provide the basis for issuing a search; and (3) deadlock detection mechanism, which detects and avoids potential deadlocks. Furthermore, the controller not only controls each real physical object but also adjusts the physical status to the *process runner* so the simulator can use feedback data to update the simulation database.

Most scheduling, planning, or layout problems depend on resolving the optimization methods algorithm. However, the criteria represented in analytic form are NP-hard problems. Simulation with discrete events has shown efficient in analyzing production systems. Several stochastic methods exist for avoiding local optimum caused by simulation. The joint use of simulation models and optimization methods has been widely studied (Dolgui and Ofitserov 1997). This investigation describes a schedule adjustment method that uses data from the previous optimization method to find a new starting point that approach the optimum. When the environmental parameters slightly change, the optimal schedule can be fine tuned based on simulation to produce a new on-line schedule, and to keep pace with rapidly changing system parameters without requiring lengthy off-line scheduling.

A systematic modelling method requires several sub-models to simulate the job shop problem in a wafer fabrication system. These sub-models can model the material handling systems and part processing under precedence constraints. Before the modelling, a model relation graph is depicted in Figure 3. The model proposed here is a general model, which does not focus on special cases. Restated, once the equipment information is provided, the model generator automatically generates the wafer-processing model. This model can support numerous process flows of different products by changing the token colour, provided these process flows are performed in the fab using the proposed generalized stochastic coloured timed Petri-Net (GSCTPN) model. The following section describes the details of the proposed Process Flow Model and Transportation Model.

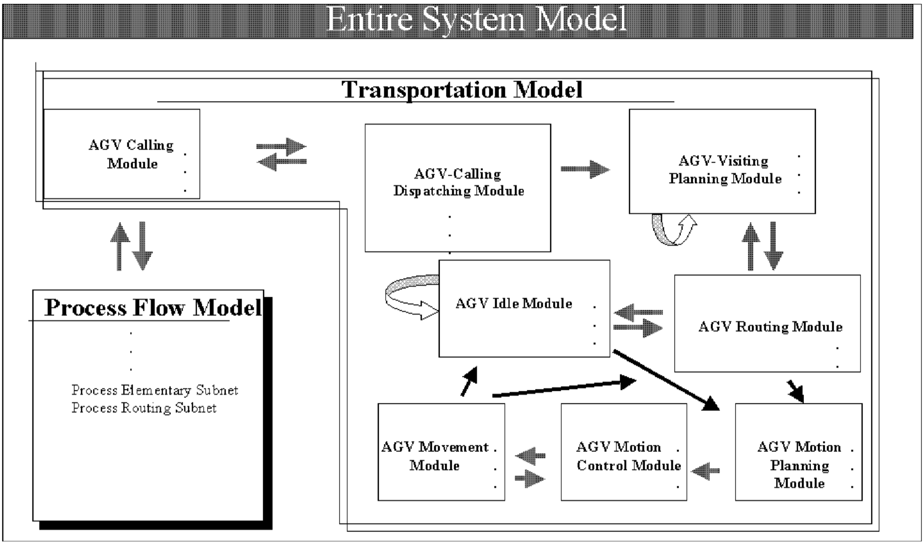


Figure 3. Model relaitonship graph.

3. Generalized stochastic coloured timed Petri-Net modelling

3.1. *Preliminaries*

The proposed generalized stochastic coloured timed Petri-Net (GSCTPN) extends the framework of the original PN with colour, time and modular attributes. The GSCTPN also permits the use of inhibitor arcs and random switches. Inhibitor arcs prevent transitions from firing when certain conditions are true. Meanwhile, the random switch resolves conflicts between two or more immediate transitions. The GSCTPN contains immediate transition and transitions with exponentially distributed firing times. Hereon, for simplicity, the latter are referred to as exponential transitions. Markings within which at least one immediate transition is enabled are called vanishing markings. Meanwhile, markings within which only exponential transitions are enabled are called tangible markings. With a tangible marking, any enabled transition can fire next. The actual transition that fires depends on the firing rates of the enabled exponential transitions. For a vanishing marking, only the enabled immediate transitions are allowed to fire.

The GSCTPN model in this work contains two major sub-models. One is called the Transportation Model, while the other is called the Process Flow Model. Both models are flexible. The Transportation Model attempts to model the lot transportation, AGV motion, and AGV travelling. Meanwhile, the Process Flow Model attempts to model lot routing, reentrant processing, machine failure, and periodic maintenance. The two sub-models, of course, interconnect to respond to the triggers from each other.

The GSCTPN model presented herein classifies the places and transitions into the five categories listed below:

Resource Places model production resources like transportation carriers, machines, and the control right of AGV stops in the transportation system. If a resource place is marked, the resources corresponding to marked resource places are free and available.

Intermediate places model the process flow of a lot or the movement of an AGV. Marked intermediate places, indicate that the last operation of a lot or the last movement of an AGV has been completed and that the intermediate place is ready for the next action.

Communication places represent the signals or conditions that are sent to or received from the other sub-models to signal upcoming events. Communication places are also used to link different modules. Based on communication places, a hierarchical and modular model of wafer fabrication can be developed.

Operation transitions represent that a specific operation is in progress. Of course, each transition is associated with an exponentially distributed firing time. The association is according to the colour set of input places.

Intermediate transitions model the satisfaction of conditions. For conflicting enabled immediate transitions, only one transition is allowed to fire at a time according to a predefined probability distribution. In this model, a constant or a marking-dependent function that depends on the processing steps of a lot is sufficient to specify the firing of the transitions. To model the technological precedence constraints for processing lots, the planned transitions according to the predefined processing flow are assigned high probability.

3.2. *Modelling for process flow*

The process flow model can be separated into two micro-models, each with different characteristics. One is the process routing module and the other is process elementary module. The following describes each micro-model in detail.

3.2.1. *Process routing module*

The process routing module models the logical process flow of the manufacturing systems. The basic concept of this module is as follows. First, the machines in the fab are divided into *n* workstations (machine groups or processing unit family), each containing one or more identical machines (or processing units). Initially, a token (lot) with the colour xy000 enters the model, where xy is the route ID, and 000 is current operation step. Lots then check-in and their colour is changed to xy001, in preparation for the first operation. Each operation has an associated workstation, thus resulting in much travel to the proper workstation in the fab and operations are performed according to the predefined process flow. After the lot completes the current operation, its colour number is increased by one, and it is ready to undergo the next operation. Systematically, after the lot completes all the operations, it reaches the finish.

A lot is released into the IC plant according to certain rules, which can be called *lot release* rules. Furthermore, before a lot is fed into workstations for operation, it must make a sequence of choices and acquire resource usage rights in terms of a token. A lot will undergo service in three phases. First, an operation type is selected for the lot according to predefined processing flow. Second, check that it is the lot that does the choosing an available workstation. Third, check that it is the lot that does the selecting of an available machine. After finishing operations, the lot releases each resource. The notations of transitions and places used in Figure 4 are explained as follows:

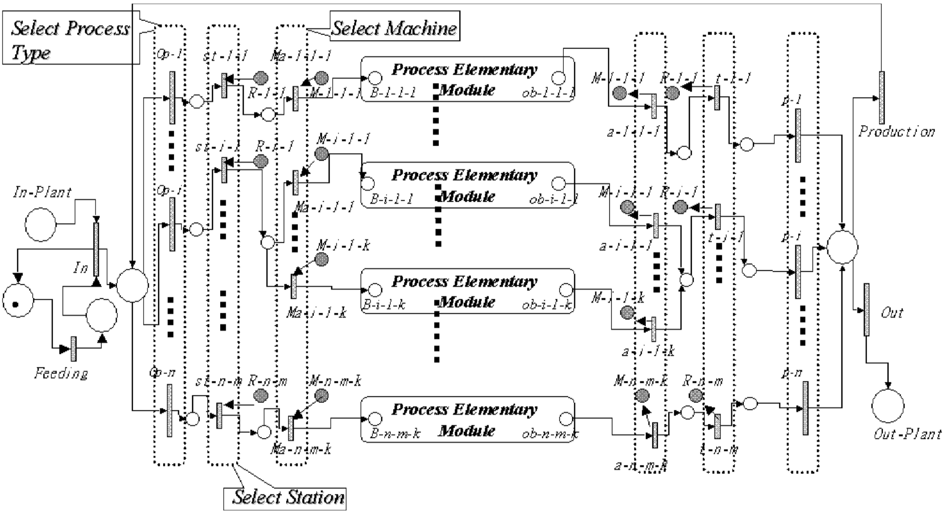


Figure 4. Process routing module.

R-i-j The place *R-i-j* is a resource place. It can be conceived of as the control right of using the *j*-th workstation of the *i*-th type of operation.

M-i-j-l The place *M-i-j-l* is a resource place. It represents the control right of the *l*-th machine of the *j*-th workstation for the *i*-th type of operation. *k* is total number of machines at the *j*-th workstation.

In-plant The place *in-plant* is an intermediate place. When a lot is released to the fab, a token with the colour (i.e., xy000) is placed here.

Out-plant The place *out-plant* is an intermediate place. A token at *out-plant* represents that the corresponding lot has completed all the operations and is ready to move.

B-i-j-l The place *B-i-j-l* is a communication place. A token at *B-i-j-l* indicates the lot is ready for processing by the machine *Mac-i-j-l.*

ob-i-j-l The place *ob-i-j-l* is a communication place. A token at *ob-i-j-l* indicates the lot processing has finished.

Feeding It is an operation transition. When *Feeding* is fired, the colour number of the token that fired the *Feeding* increases by one. This event signifies that a lot is released into the fab and prepared for the first operation.

In It is an intermediate transition.

Out When the transition *Out* is fired, the lot has completed all operations and entered *Out-Plant*.

Production The transition *Productionis* an intermediate transition. When *Production* is fired, the colour number of the token that fired the *Production* increases by one. This increase represents that the corresponding lot has finished the current operation and is ready for the next operation.

Op-i The transition *Op-i* is an intermediate transition. It indicates that lots select an operation type according to predefined processing flows.

st-i-j similarly, the transition *st-i-j* is also an intermediate transition. It indicates that lots choose one workstation from among those available according to a predefined processing flow.

Ma-i-j-l The transition *Ma-i-j-l* is also an intermediate transition. It represents that a lot has selected one of the machines available at the *j*-th workstation according to predefined processing flow or random choice.

Op-i Finally, the transition *Op-i* is another intermediate transition. It has a colour set that denotes the coloured tokens that can be fired through this transition.

3.2.2. *Process elementary module*

The process elementary module comprises four subnets as Figure 5 displays, each with different characteristics. The four subnets are, the processing subnet, inspection subnet, machine subnet, and inspection machine subnet. However, the inspection subnet and inspection machine subnet have to be added if wafers require inspection. The following are detailed descriptions of each of these subnets.

3.2.3. *Processing subnet*

According to Figure 5, the essential idea of modelling processing subnet is that, before a lot can monopolize any resources, it must acquire its control right via a token.

man-i-j-l The place *man-i-j-l* is a resource place. It represents that if a lot wants to move to the machine *Mac-i-j-l* located at the stop *x*, it must acquire the token of the resource place *man-i-j-l*.

To-x The place *To-x* is a communication place. If the current location of the wafer differs from that of machine *Mac-i-j-l*, a signal must be sent to the communication place *To-x*. This signal calls an AGV to come to the place where the wafer resides.

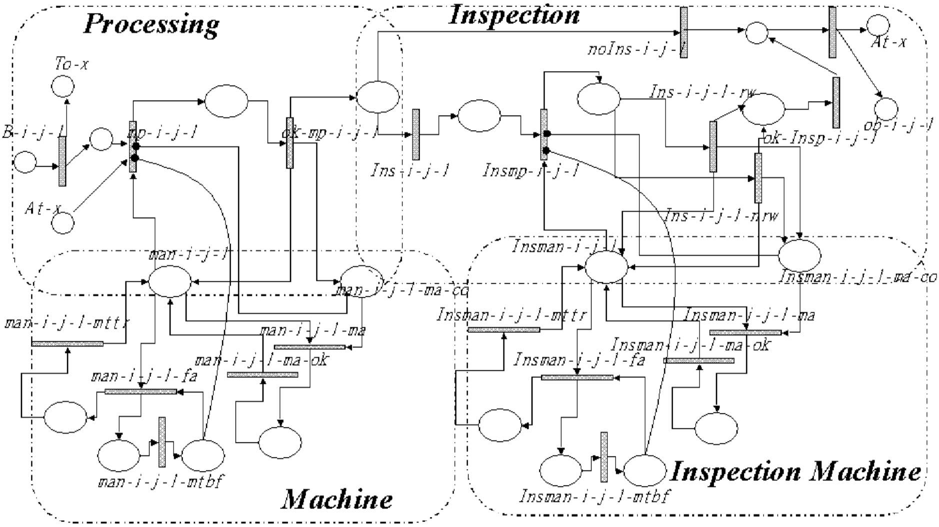


Figure 5. Process elementary module.

At-x The place *At-x* is a communication place. When the AGV arrives, it loads the wafer. The AGV then starts to move to the destination. Finally, when the AGV reaches the destination, starts the unloading and releases the loading space of the AGV, a token is moved into the communication place *At-x*.

mp-i-j-l The transition *mp-i-j-l* is an operation transition. The *mp-i-j-l* is associated with exponentially distributed firing times according to the colour set of input places. As mentioned before, the colour set represents the current step of the process flow, implying that operating time can depend on the current step number of a lot and lot type.

3.2.5. *Inspection subnet*

According to Figure 5, wafers may or may not be inspected and reprocessed for some operations. For example, if photo resistors are not completely removed in etching, the step is repeated.

noIns-i-j-l The transition *noIns-i-j-l* is an intermediate transition. If wafer inspection is not required, the intermediate transition *noIns-i-j-l* is fired.

Insmp-i-j-l The transition *Insmp-i-j-l* is an operation transition. The operation transition *Insmp-i-j-l* represents the inspection operation. The inspection time also depends on the current step number of the lot and lot type.

Ins-i-j-l-rw The transition *Ins-i-j-l-rw* is an intermediate transition. If a test wafer of a lot fails an inspection, the intermediate transition *Ins-i-j-l-rw* will be fired.

Ins-i-j-l-nrw The transition *Ins-i-j-l-nrw* is an intermediate transition. If the intermediate transition *Ins-i-j-l-rw* fires, the intermediate transition *Ins-i-j-l-nrw* also fires. In practice, only one is allowed to fire at a time according to the fabrication statistics. Finally, tokens are moved into both communication places *At-x* and *ob-i-j-l*. This process signals the completion of the current processing step.

3.2.5. *Machine subnet and inspection machine subnet*

According to Figure 5, machine failure is common in IC fabrication.

man-i-j-l-mtbf The transition *man-i-j-l-mtbf* is an operation transition. The operation transition *man-i-j-l-mtbf* must be fired periodically, according to the fabrication statistics. This phenomena is called *mean time between failure* (MTBF).

man-i-j-l-fa The transition *man-i-j-l-fa* is an intermediate transition. If transition *man-i-j-l-mtbf* is fired, intermediate transition *man-i-j-l-fa* is enabled and will be fired immediately. When *man-i-j-l-fa* is fired, a token leaves the resource place *man-i-j-l*.

man-i-j-l-mttr Transition *man-i-j-l-mttr* is an operation transition. The repair time is called *mean time to repair* (MTTR). After the operation transition *man-i-j-l-mttr*, the machine repair is complete and the token returns to place *man-i-j-l*.

i-j-l-ma-co The place *i-j-l-ma-co* is a communication place. Machines need regular maintenance. Initially, communication place *)i-j-l-ma-co* is empty. The output arc weight for place *i-j-l-ma-co* is *man-i-j-l-ma-th*.

man-i-j-l-ma The transition *man-i-j-l-ma* is an operation transition. Once transition *mp-i-j-l* has fired *man-i-j-l-ma-th* times, the operation transition *man-i-j-l-ma* fires. After the firing, a token departs from the resources place *man-i-j-l}*.

man-i-j-l-ma-ok The transition *man-i-j-l-ma-ok* is an intermediate transition. Once the associated time of *man-i-j-l-ma* has passed, the intermediate transition *man-i-j-l-ma-ok* fires, and the token returns to place it *man-i-j-l*. Transitions and places in the Inspection Machine subnet, resemble those in the machine subnet.

3.3. *Modelling for transportation system*

The transportation model can be broken into several micro-models, each with different characteristics. They are, the AGV-calling module, AGV-calling dispatching module, AGV-visiting planning module, AGV idle module, AGV routing module, AGV motion planning module, AGV motion control module, and AGV movement module. Each micro model is described in detail as follows.

3.3.1. *AGV-calling dispatching module*

Tour_L_x_S_y: The place *Tour_L_x_S_y* is a communication place. A token in this place implies that wafer will have a tour loaded from the stop *x* and stored at the stop *y*.

Tour_L_x_S_y_ok: The place *Tour_L_x_S_y_ok* is a communication place. A token in the communication place *Tour_L_x_S_y_ok* indicates that the transporting tour from stop successfully.

According to Figure 6, a lot is currently located at stop 5 and the next serving workstation is located at stop 19. The place *Tour_L_5_S_27* is a communication place. Before the next operation, the communication place *To-19* is marked to signal AGV and invokes an AGV to come to the location of the wafer. Stop 19 is located in a different bay, and it must be completed with the following steps. First, the communication place *Tour_L_5_S_27* is marked. This marking means that wafer will have a tour loading from stop 5 and storing at stop 27. When the AGV arrives, it loads the wafer. The AGV then moves to the interbay station 27. The place *Tour_L_5_S_27_ok* is a communication place. A token in communication place

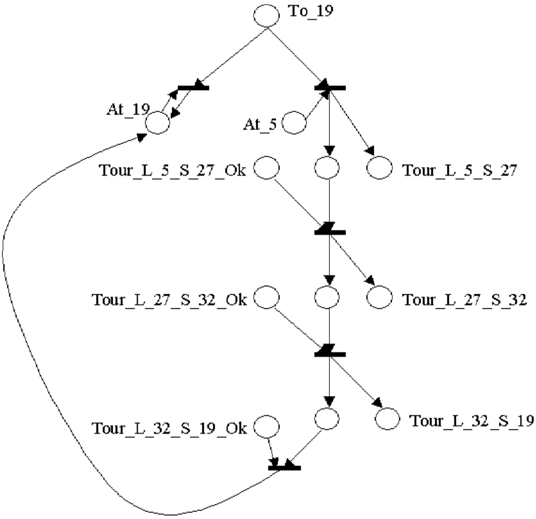


Figure 6. AGV-calling module.

Tour_L_5_S_27_ok signifies that the transporting tour from stop 5 to stop 27 has successfully finished. Next, the loop vehicle takes lots from interbay station 27 to 32. Finally, an AGV at the destination-bay transports lots from interbay station 32 to the destination stop 19. When the AGV arrives at stop 19, it proceeds to store the wafer. A token at communication place *Tour_L_32_S_19_ok* reveals that the transporting tour from 32 to 19 is completed. Finally, a token enters communication place *At-19*. This event indicates that the next operation will start.

3.3.2. *AGV − calling dispatching module*

Two issues in the AGV-calling dispatching module are AGV-calling dispatching and precedence constraints for transporting.

V_x_WC: The place *V_x_WC* is an intermediate place. When a token enters the intermediate place *V_x_WC* it means that a lot is waiting for AGV *x* to transport it.

Disp_x: The transition *Disp_x* is also an intermediate transition. If AGV *x* is selected, the intermediate transition *Disp_x* fires.

V_x_B: Meanwhile, place *V_x_B* is a resource place. If the buffer of selected AGV *x* has free space, the resource place *V_x_B* is marked.

As_OK_x: The transition *As_OK_x* is an intermediate transition.

V_x_L_y: Meanwhile, the place *V_x_L_y* is a communication place. That a token in this place represents that a signal is generated to invoke an AGV *x* to come to the place *y* of the wafer.

V_x_L_y_OK: The place *V_x_L_y_OK* is another communication place. If the communication place *V_x_L_y_OK* is marked, it indicates that AGV *x* has arrived at the stop *y* and finished loading the wafer.

V_x_S_y: The place *V_x_S_y* is a communication place. That a token in this place represents that a signal is generated to invoke an AGV *x* to come to the stop *y* for storage at the location of the wafer.

V_x_S_y_OK: The place *V_x_S_y_OK* is also a communication place. If the communication place *V_x_S_y_OK* is marked, it indicates that AGV *x* has arrived at stop *y* and has finished storing the wafer.

Figure 7 illustrates, that, for example, the presence of a token at the communication place *Tour_L_5_S_27*, signals that a lot transporting is to be served. The transportation is completed in two stages. First, an AGV is chosen according to the dispatching policy. A good dispatching policy can increase AGV utilization. Initially, three AGVs are assumed to be in the intrabay 1. If AGV 1 is selected, the intermediate transition *Disp_1* is fired. When a token enters the intermediate place *V_1_WC*, it means that a lot is waiting for AGV 1 to transport it. If the buffer of selected AGV contains free space, the resource place *V_1_B* is marked, and firing of the intermediate transition *As_OK_1* is enabled.

Next, precedence constraints for transporting lots are considered. The time that the AGV takes to load one lot must happen before storing it. When the intermediate transition *As_OK_1* fires, the communication place *V_1_L_5* is marked. The marking represents that a signal is generated to invoke an AGV 1 to come to location of the wafer. If the communication place *V_1_L_5_OK* is marked, it indicates that AGV 1 has arrived and finished loading the wafer. When the communication

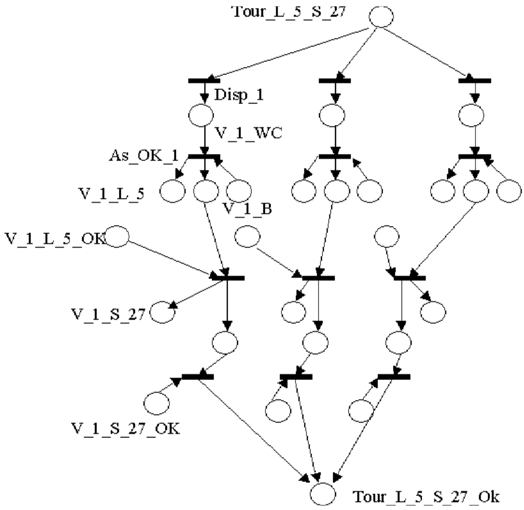


Figure 7. AGV-calling dispatching module.

place *V_1_S_27* is marked, it represents that the system invokes an AGV 1 transporting wafer from stop 5 to stop 27 for storage. Meanwhile, a situation in which the communication place *V_1_S_27_OK* is marked implies that AGV 1 has arrived at stop 27 and has finished storing the wafer. Finally, when the lot tour is finished successfully, the communication place *Tour_L_5_S_27* is marked.

3.3.3. *AGV − visiting planning module*

Each multiple-load AGV may have to visit several stop points for loading or storing wafers. Thus, good route planning is needed to solve vehicle traffic jam problems.

V_x_To_y: The place *V_x_To_y* is a communication place, and a token there represents that AGV *x* must visit stop *y*.

V_x_L_C_y: The place *V_x_L_C_y* is also a communication place, and a token there indicates that the goal of AGV *x* is to load a lot at stop *y*.

V_x_S_C_y: Also, the place *V_x_S_C_y* is a communication place, and a token at place *V_x_S_C_y* indicates that the goal of AGV *x* is to store a lot at stop *y*.

Vi_x_OK: Meanwhile, the place *Vi_x_OK* is another communication place, and when it has a token, it indicates that the current visit is finished and the AGV proceeds to the next visit.

V_x_Idle: Finally, the place *V_x_Idle* is also a communication place. If no visiting queues exist, the communication place *V_x_Idle* will be marked, indicating that AGV *x* is idle.

For example, there are eight stops(2,3,4,5,6,7,26,27) in intrabay 1. Figure 8 illustrates that if the communication place *V_1_L_5* is marked, AGV 1 must go to stop 5 to load a lot. After this, the communication places *V_1_To_5* and *V_1_L_C_5* are both marked. If the communication place *V_1_S_27* is marked, it means that AGV 1 must arrive at stop 27 for storing.

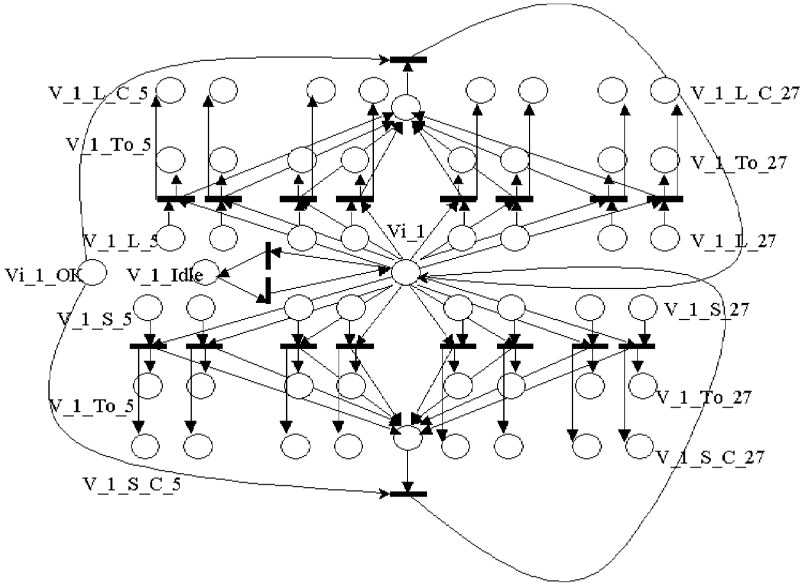


Figure 8. AGV-Visiting planning module.

3.3.4. *AGV routing module*

The AGV routing module corresponds to the decision-making model for AGV routing. Each time an AGV wants to change stops, it must first decide its route. Once the routing decision is be determined the AGV can move along that route. Therefore, for each stop, a corresponding Petri-Net model is necessary to take care of the routing decision. Generally, stops and tracks are assumed to have a capacity of more than one. The basic concept of this model is as follows. When an AGV moves from its current stop to the adjacent stop, two steps are necessary. Firstly, the AGV needs to receive the 'right' to move on the relevant track. If the 'right' is not granted, the next stop on the route may be reselected. If this condition is satisfied, the AGV can begin travelling on that track.

Wa_x_y_z: The place *Wa_x_y_z* is an intermediate place. If AGV *x* at current stop *y* selects adjacent stop *z* as its route, the intermediate place *Wa_x_y_z* is marked.

B_y_z: The place *B_y_z* is a resource place. If the resource place *B_y_z* has tokens, it represents that the 'right' is ready.

Ha_x_y_z: The place *Ha_x_y_z* is an intermediate place.

Wi_x_S_z: The place *Wi_x_S_z* is also an intermediate place.

Want_S_z: Meanwhile, the place *Want_S_z* is a communication place, representing that AGV *x* is travelling to the stop *z*. Simultaneously, the control right of the stop *y* will be released to allow another AGV to use it as a destination or pass-by stop. A token in the communication place *Want_S_z* also represents that AGV *x* broadcasts a 'right wanted' message to each of the idle AGVs.

B_S_z: The place *B_S_z* is a resource place. The AGV must acquire the control right of the selected stop to ensure that the buffer of the stop has free space. If the resource place *B_S_z* contains tokens, it represents the buffer has free space or some idle AGVs had left from stop *z*.

HV_x_S_z: The place *HV_x_S_z* is an intermediate place.

Tim_x_S_z: The transition *Tim_x_S_z* is an operation transition. If the intermediate place HV_x_S_z is marked, the operation transition *Tim_x_S_z* is enabled and AGV *x* can enter the adjacent stop *z*. Therefore, if no tokens remain in the resource place *B_S_z*, a *back moving* will occur. The ownership right of the track will be released to allow another AGV to use it, and the resource place *B_y_z* to contain a token again.

Speed_x: The place *Speed_x* is a communication place. A token in colour set of communication place *Speed_x* represents the speed of AGV *x* at that time. The presence of the token implies that variable-speed AGV may consume a different amount of time for each journey to the adjacent stop.

St_z_x: The place *St_z_x* is an intermediate place that indicates that the current location of AGV *x* is the stop *z*. However, the stop *z* is not the goal stop *w*, a token is returned to the communication place V_xTo_w.

For example, Figure 9 reveals the AGV routing module which guides an AGV to move from the current stop 2 to stop 5. If AGV 1 selects adjacent stop 7 as its route, the intermediate place *Wa_1_2_7* is marked if the resource place *B_2_7* contains tokens. Meanwhile, when the intermediate places *Ha_1_2_7* are marked, tokens immediately enter both of the intermediate places *Wi_1_S_7* and the communication place *Want_S_7*. This behaviour represents that AGV 1 is travelling to stop 7. If the resource place *B_S_7* contains tokens, it represents the buffer has free space or some idle AGVs had left stop. In this situation, the intermediate place *HV_1_S_7* is marked. The operation transition *Tim_1_S_7* is enabled and AGV 1 can enter the adjacent stop 7. The operation transition *Tim_1_S_7* is associated with a firing time

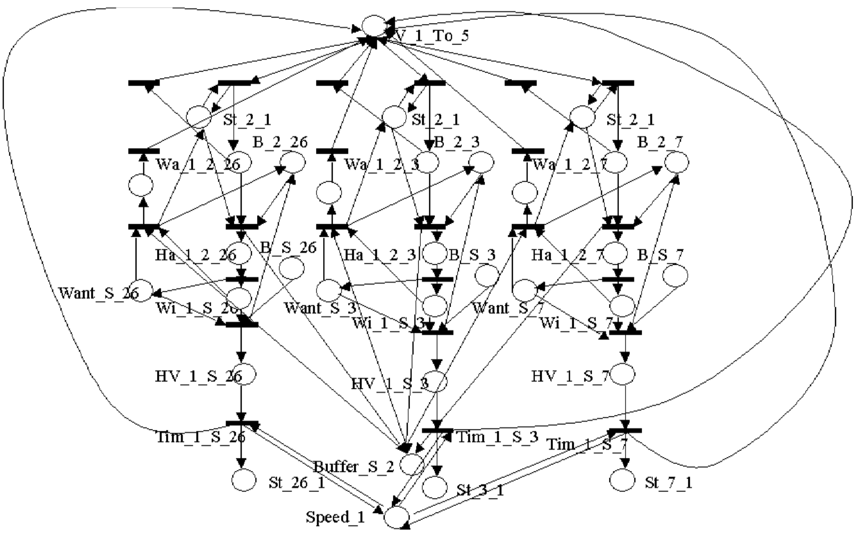


Figure 9. AGV routing module.

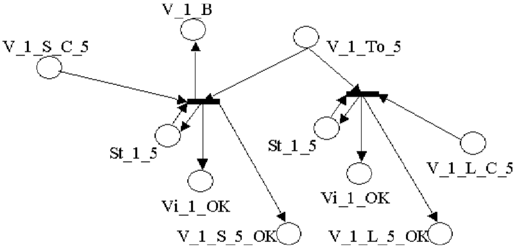


Figure 10. AGV routing module.

according to the colour set of input place *Speed_1*. After the time associated with operation transition *Tim_1_S_7* has passed, the intermediate place *St_7_1* is marked.

When AGV 1 arrives at stop 5 as Figure 10, one of two transitions fire according to the status of the communication place *V_1_L_C_5* and the communication place *V_1_S_C_5*. A token enters the communication place *Vi_1_OK*, signals the AGV-visiting planning module that current visiting is completed. After this, the AGV proceeds to the next visiting. Either of the communication places *V_1_S_5_OK* and *V_1_L_5_OK* may be marked. The marking represents that a signal has been sent to the AGV-calling dispatching module to confirm that loading or storing at stop 5 is completed.

3.3.5. *AGV idle module*

Notably, until now the AGV routing module has already embedded a control rule into it to prevent vehicle collision problems. In an IC fabrication system, the transporting system may have more than one carrier to transport materials among different workstations. However, collision problems no longer exist and, hence, traffic jam problems due to multiple carriers must be solved. In the AGV routing module, the AGV can select the available adjacent stop to achieve partially jam-free conditions among carriers.

The AGV idle module aims to introduce a control method into the Petri-Net model to guarantee jam-free conditions. The method provided is called *recursively leaving* strategy, whose basic idea is described below. When a missioned AGV wants to proceed to a stop, it sends a message to the AGVs that occupy the stop. An idle AGV is selected and it leaves that stop.

places and transitions: The notations of places and transitions are the same with those in the AGV routing module.

For example as Figure 11 indicates, if the communication place *V_1_Idle* is marked, it represents that AGV 1 is idle. If AGV 1 is currently at stop 2, a token in communication place *Want_S_2* means that a missioned AGV wants to move to stop 2 but that stop 2 is occupied. Only one idle AGV will be selected to leave stop 2. In this example, the AGV 1 is selected. After a message ‘Want_S_2’ is sent to the idle AGV 1, it starts to move to the next adjacent stop and leaves its previous location unoccupied. After this, the communication place *B_S_2* is marked. Then, if all subsequent adjacent stops are occupied, another idle AGV on the adjacent stops will leave. This recursive processing continues until the jam-free conditions are satisfied. However, the layout geometry and the number of AGVs limit the effectiveness of this recursive departure strategy. When the number of AGVs increases, the frequency of

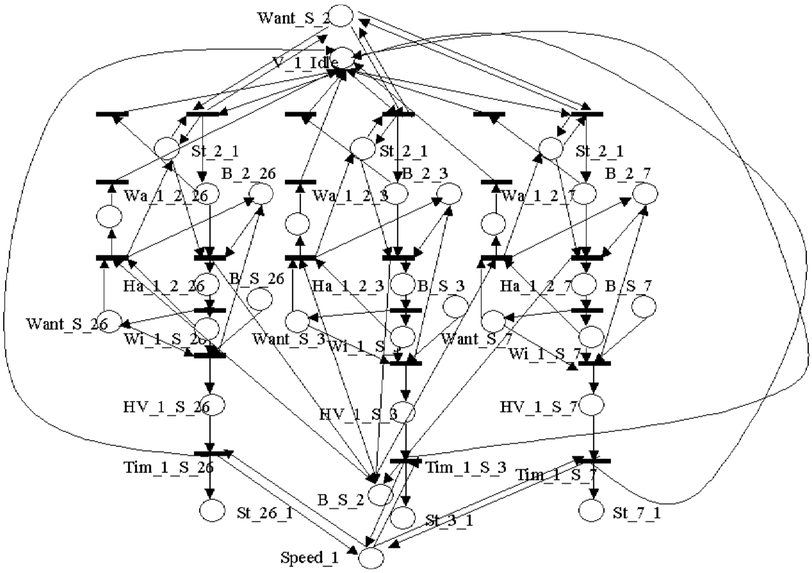


Figure 11. AGV idle module.

issuing the 'Want' message is quite high. This implies that the system performance will deteriorate.

3.3.6. *AGV movement module*

Multiple automated guided vehicles (AGVs) are used for wafer transportation. Each AGV carries a multiple-load, and can move at variable speeds. The AGV movement module attempts to model the characteristics of AGV movement. The AGV movement module models the complete behaviour of AGVs which are in transit except for the actions of loading and unloading.

V_x_A: The place *V_x_A* is a communication place which represents that a 'start accelerating' command is generated for AGV *x*.

V_x_ST: The place *V_x_ST* is an intermediate place which represents that the status of movement of AGV *x* has reached the *stop* stage.

V_x_AC: The place *V_x_AC* is an intermediate place, indicating that the status of movement of AGV *x* has reached the *accelerating* stage. If AGV *x* is at the 'stop' stage and the communication place *V_x_A* is marked, AGV *x* will proceed to the 'accelerating' stage, and mark the intermediate place *V_x_AC*.

V_x_CV: The place *V_x_CV* is an intermediate place which indicates that the status of movement of AGV *x* is at the *constant-speed* stage. However, if both the intermediate place *V_x_CV* and the communication place *V_x_A* are marked, it indicates that there is a 'start accelerating' command. Then, the AGV *x* will enter the 'accelerating' stage and the intermediate place *V_x_AC* will be marked.

V_x_DA: The place *V_x_DA* is a communication place, which signifies that a 'start decelerating' command is generated for AGV *x*.

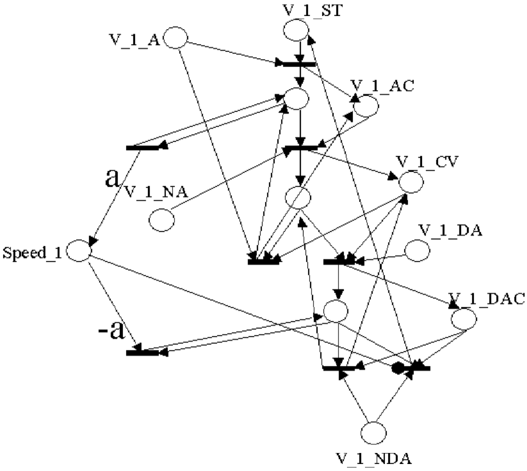


Figure 12. AGV movement module.

V_x_DAC: The place *V_x_DAC* is an intermediate place, representing that the status of movement of AGV *x* is at the *decelerating* stage. If tokens are both at the intermediate place *V_x_CV* and the communication place *V_x_DA*, it represents that a 'start decelerating' command is to be generated. The intermediate place *V_x_DAC* will be marked.

V_x_NDA: The place *V_x_NDA* is a communication place, representing that a 'stop decelerating' command is generated for AGV *x*.

For example, as Figure 12 indicates, the status of AGV 1's movement can be classified into four stages, namely, V_1_ST, V_1_AC, V_1_CV and V_1_DAC. When the AGV moves into the 'constant-speed' stage when the speed is not equal to zero. Otherwise, AGV moves into the 'stop' stage and the intermediate places *V_1_ST* is marked.

3.3.7. *AGV motion planning module*

Each time an AGV approaches the next adjacent stop, it must be in one of two statuses. One is the *non-braking* status, while the other is the *braking* status.

V_x_Going: The place *V_x_Going* is a communication place. A token at the communication place *V_x_Going* signifies that AGV *x* is in non-braking status.

V_x_Braking: Meanwhile, the place *V_x_Braking* is also a communication place, and when this communication place *V_x_Braking* is marked it indicates that it is braking status.

other places: The notations of other places and transitions are the same with those in AGV routing module.

For example, Figure 13 illustrates the AGV motion planning module, which guides an AGV while moving from the current stop 2. This work assumes that the next selected adjacent stop is 7. Initially, an AGV is not braking, and an inhibitor arc *d* leaves the input place *B_2_7*. If the resource place *B_2_7* has tokens, then the control right of the track-arc is ready and AGV 1 remains not braking. Otherwise, the communication place *V_1_Braking* is marked, meaning that AGV 1 enters the

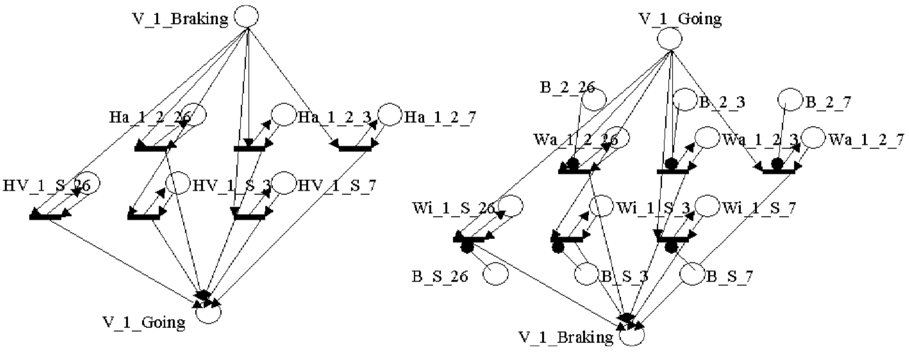


Figure 13. AGV motion planning module.

braking status and waits for the control right. If the resource place *B_2_7* has tokens again, the intermediate place *Ha_1_2_7* is marked immediately. A token leaves the communication place *V_1_braking* and enters the communication place *V_1_ going* , AGV 1 enters the non-braking status again. The same method is used as for the resource place *B_S_7*.

3.3.8. *AGV motion control module*

places and transitions: The notations of places and transitions are the same as those in AGV movement module.

The AGV motion control module embeds a control rule into the model so that the AGV can move properly. The basic concept of this model is as follows. When an AGV is not braking and is at the 'stop' or 'constant-speed' stage mentioned in the AGV movementmModule, a 'start accelerating' command will be generated. This command asks the AGV to accelerate. If the AGV is at the 'decelerating' stage, a 'stop decelerating' command will be generated. When an AGV is braking and is at the 'constant-speed' stage, a 'start decelerating' command will be generated. Meanwhile, if an AGV is braking and at the 'accelerating' stage, a 'stop accelerating' command will be generated. Furthermore, if an AGV is braking and is at the 'decelerating' stage, the AGV will have a speed of zero. Then, a 'stop decelerating' command will be generated. For example, Figure 14 represents the AGV motion control

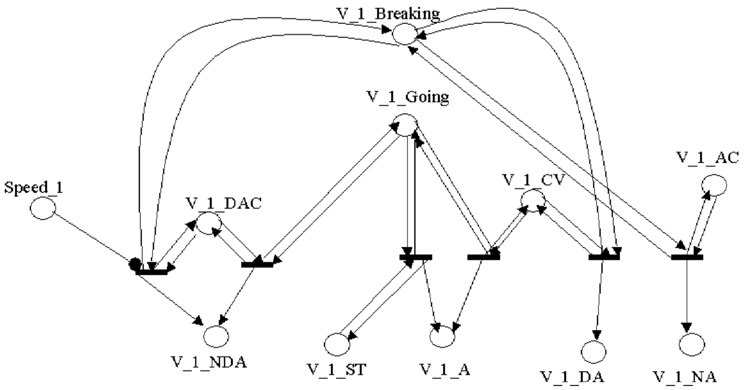


Figure 14. AGV motion control module.

module, which generates proper motion control commands for the AGV 1. A token in the communication place *V_1_going* represents that AGV 1 is not braking. If the communication place *V_1_braking* is marked it means that it is not braking.

Notably, the AGV motion control module and AGV motion planning module have already embedded a motion control strategy into the multiple-load and variable-speed AGV systems.

4. Simple rules in process runner

4.1. *Lot scheduling*

In the IC fabrication systems, numerous lots queue up for the machine. Lot scheduling attempts to determine what type of lot will be operated first. Many heuristic rules exist have been proposed in previous studies (Li *et al.* 1996, Nahahaari and Khan 1997, Kim *et al.* 1998a, Kim *et al.* 1998b, Glassey and Weng 1991), which can be described as listed below:

FIFO: First-in-first-out, where lots are scheduled in the order they arrive.

LTF: Longest time first. where lots with the longest processing times are prioritized. This policy represents a worst-case scenario.

STF: Shortest time first, where lots with the shortest processing times are prioritized.

LRF: Least remaining step first, where lots with least remaining step first are prioritized.

SSF: Shortest setup time first, where lots with shortest setup time are prioritized.

4.1.1. *Lot release*

Furthermore, many heuristic rules of lot release exist, which can be described as listed below:

Idle avoidance: Start a new lot to avoid bottlenecks and idle workstation due to a lack of work. Thus, a new lot is released when the virtual inventory at the bottleneck workstation falls below a predetermined value.

CONWIP: Constant work-in-process (WIP), which means starting a new lot whenever a lot is completed. With this rule, the WIP level controls the throughput.

POISS: Lots enter the facility according to a Poisson process.

DETERMIN: Inter-arrival times of lots are constant. This policy is also called UNIF(uniform).

WR(C): Workload regulating input is used. When the expected amount of work in the plant for the bottleneck station drops to C hours, then a new lot is released into the plant.

PWR(C): Parametric workload regulating input is used. This policy resembles WR, but only a portion of lots in the plant are considered when calculating the workload of the bottleneck workstation in PWR.

4.2. *AGV-calling dispatching*

The control policies of AGV can be classified into four types, namely, AGV-Calling dispatching, AGV-Visiting planning, AGV routing, and making an Idle-AGV depart. As mentioned in the Process Elementary Module, if the current loca-

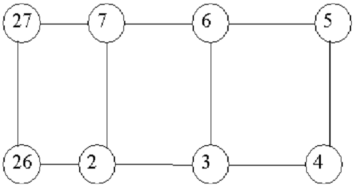


Figure 15. Layout directed graph.

tion where the wafer resides is differs from that of the workstation of the next operation, a signal must be sent to AGV. This signal invokes an AGV to come to the place where the location of the wafer. An AGV serves a lot transporting, called a *transporting task*. A heuristic rule *workload regulating*(WR) is used, Namely, when the expected amount of workload for the AGV drops to k transporting jobs, then dispatch a transporting job to the AGV.

4.3. *AGV-visiting planning*

Each multiple-load AGV may have to visit several stops for loading or storing the wafers that it is carrying. A good visiting plan is necessary to solve vehicle traffic jam problems by deciding the visiting sequence of an AGV. A heuristic rule *shortest distance first* (SDF) is used. The stops closest to the current location are visited first. To achieve this, for the example of the intrabay 1, the system layout must first be transformed to a *layout directed graph* which is shown in Figure 15. Each arc has the same distance.

4.4. *AGV routing*

Each time an AGV wants to move from the current stop to another stop, it must first determine its route. The routing policy can be made and the AGV can move along that route. The heuristic rule *shortest path first* (SPF) is used. Every two nodes in Figure 15 have a planned static shortest path. However, at some point, a limit will be reached due to AGV *traffic congestion.* A simple heuristic rule *congestion solving* is used. A flow control redirects an AGV through a non-congested area or slows an AGV that wants to enter a congested area.

4.5. *Selection of idle AGV*

When a missioned AGV wants to enter a stop, it informs AGVs that occupy the stop. An idle AGV is selected to leave that stop. A heuristic rule *shortest idle time first* (SIF) is used, in which the AGVs with the shortest IDLE time depart first. The reason is as follows. when AGVs have a balanced load, there is a higher probability that the AGV with a long idle time will be dispatched a transporting job.

5. Performance analysis and scheduling adjustment by simulation

5.1. *Plant layout and process flow: a case study*

To illustrate the promise of the work, a real-word IC wafer fabrication system is made the target plant layout for implementation. Producing ICs is very complex and involves hundreds or thousands of machines and processing steps. For easy illustration and confidentiality, this study only presents a portion of these manufacturing machines. However, the process flow modelled here is based on an actual process

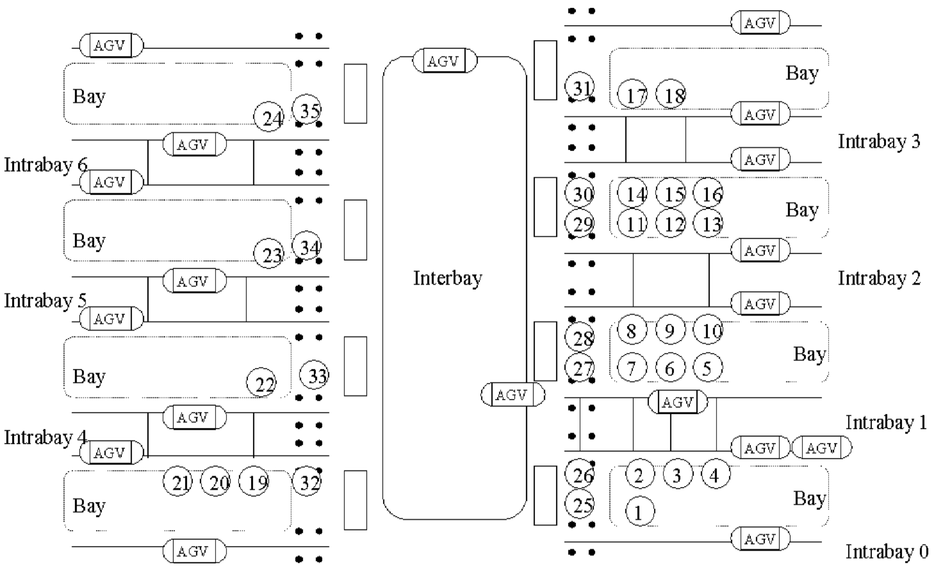
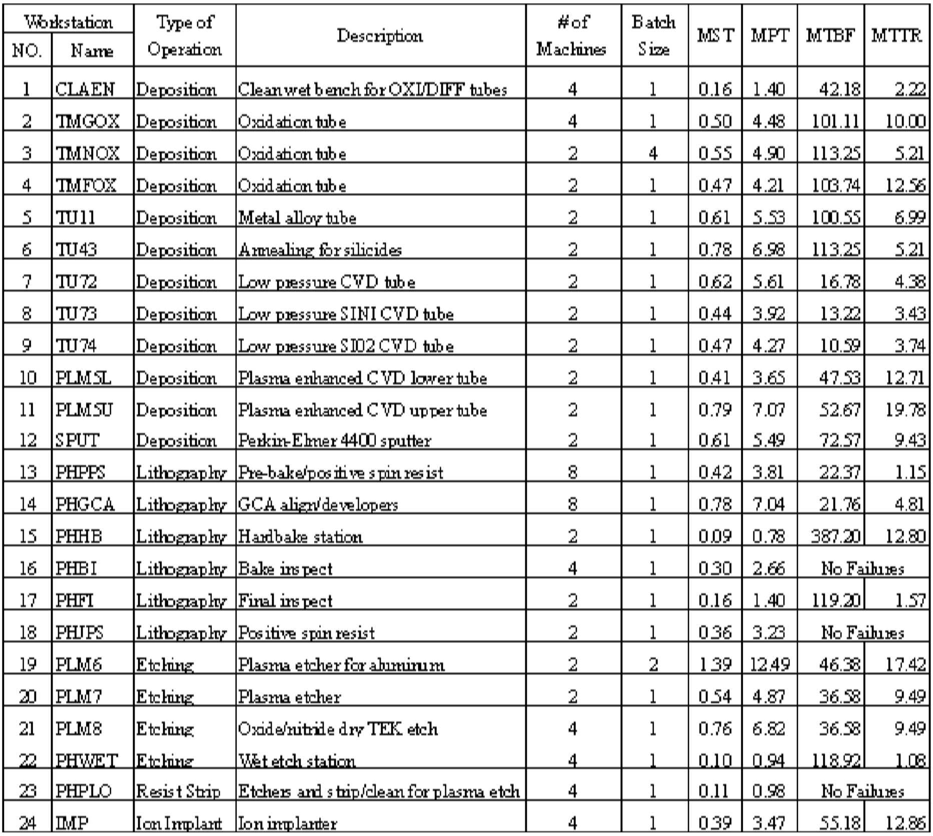


Figure 16. Plant layout.

under development in a Taiwanese wafer fabrication system. Thus, it stills gives an overall picture of IC fabrication.

As shown in Figure 16, there are 24 workstations arranged in locations 1-24. The corresponding AGV stops are also arranged in locations 1–24. Multiple automated guided vehicles (AGV) are used for wafer transportation. In the interbay system, the corresponding AGV stops are arranged in locations 25–35. Meanwhile, this system has only a single-loop track. Tracks in different areas do not overlap and are denoted as Intrabay 0–7.

As Figure 17 reveals, for the resist strip process, reentry occurs 23 times for each lot. Twenty-four workstations are divided into six types, and each of the multi-server workstations comprises several identical pieces of equipment. Figure 17 classifies the resources according to function. In the simulation model, the machine downtime, consisting of unscheduled breakdowns as well as scheduled maintenance is included. Time between failures and repair time for each workstation is randomly generated from uniform distributions with given mean values. In the Fab model, the mean processing time (MPT) contains the mean processing time and mean setup time. The simulation model presented here, separates these values. The MPT in the current model equals 0.9XMPT in the Fab model, and the mean setup time (MST) equals 0.1XMPT in the Fab model. In the simulation model presented here, each lot entering the fab is based on a specific process flow. The model contains two different process flows, and the sequence of stations to be visited in three process flows are listed in Figure 18 and Figure 19, respectively. The Re-entry operations on the same machine is assumed to possibly have different processing time, since different recipes must be processed. Additionally, if the machine deals with processes with different recipes, the machine must be set-up. Here, the processing time (PT) for a lot is randomly generated from a uniform distribution between 0.9XMPT and 1.1XMPT, where MPT for each workstation is provided in Figure 17. The setup



Workstation		Type of Operation	Description	#of Machines	Batch Size	MST	MPT	MTBF	MTTR
NO.	Name								
1	CLAEN	Deposition	Clean wet bench for OXI/DIFF tubes	4	1	0.16	1.40	42.18	2.22
2	TMGOX	Deposition	Oxidation tube	4	1	0.50	4.48	101.11	10.00
3	TMNOX	Deposition	Oxidation tube	2	4	0.55	4.90	113.25	5.21
4	TMFOX	Deposition	Oxidation tube	2	1	0.47	4.21	103.74	12.56
5	TU11	Deposition	Metal alloy tube	2	1	0.61	5.53	100.55	6.99
6	TU43	Deposition	Annealing for silicides	2	1	0.78	6.98	113.25	5.21
7	TU72	Deposition	Low pressure CVD tube	2	1	0.62	5.61	16.78	4.38
8	TU73	Deposition	Low pressure SINI CVD tube	2	1	0.44	3.92	13.22	3.43
9	TU74	Deposition	Low pressure SI02 CVD tube	2	1	0.47	4.27	10.59	3.74
10	PLMSL	Deposition	Plasma enhanced CVD lower tube	2	1	0.41	3.65	47.53	12.71
11	PLMSU	Deposition	Plasma enhanced CVD upper tube	2	1	0.79	7.07	52.67	19.78
12	SPUT	Deposition	Perkin-Elmer 4400 sputter	2	1	0.61	5.49	72.57	9.43
13	PHPPS	Lithography	Pre-bake/positive spin resist	8	1	0.42	3.81	22.37	1.15
14	PHGCA	Lithography	GCA align/developers	8	1	0.78	7.04	21.76	4.81
15	PHHB	Lithography	Hardbake station	2	1	0.09	0.78	387.20	12.80
16	PHBI	Lithography	Bake inspect	4	1	0.30	2.66	No Failures	
17	PHFI	Lithography	Final inspect	2	1	0.16	1.40	119.20	1.57
18	PHJPS	Lithography	Positive spin resist	2	1	0.36	3.23	No Failures	
19	PLM6	Etching	Plasma etcher for aluminum	2	2	1.39	12.49	46.38	17.42
20	PLM7	Etching	Plasma etcher	2	1	0.54	4.87	36.58	9.49
21	PLM8	Etching	Oxide/nitride dry TEK etch	4	1	0.76	6.82	36.58	9.49
22	PHWET	Etching	Wet etch station	4	1	0.10	0.94	118.92	1.08
23	PHPLO	Resist Strip	Etchers and strip/clean for plasma etch	4	1	0.11	0.98	No Failures	
24	IMP	Ion Implant	Ion implanter	4	1	0.39	3.47	55.18	12.86

Figure 17. Equipment description.

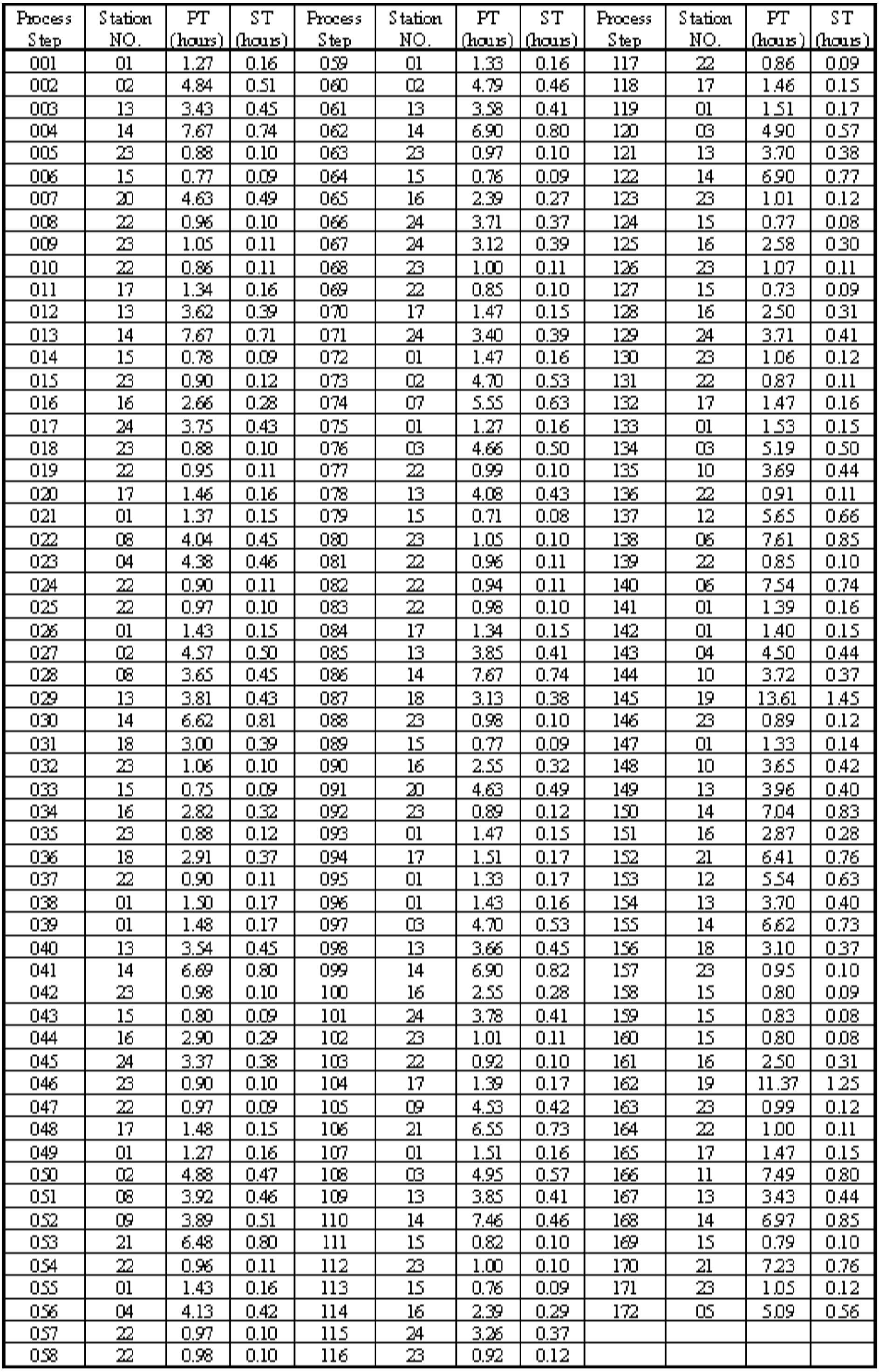
time (ST) is treated similarly, also being randomly generated from a uniform distribution between 0.9XMST and 1.1XMST.

5.2. *Simulation result*

Since the system model is large, simulation is used to obtain performance measures. In this simulation experiment, due to the nature of IC wafer processing, operation time varies little between lots. For ease of illustration, this work only focuses on one intrabay in the material handling system. Meanwhile, the other interbay and intrabays are treated with the same simulations. Considering the *warm-up* period of the system, each simulation run is implemented for an extended period. A simulation experiment is conducted as follows.

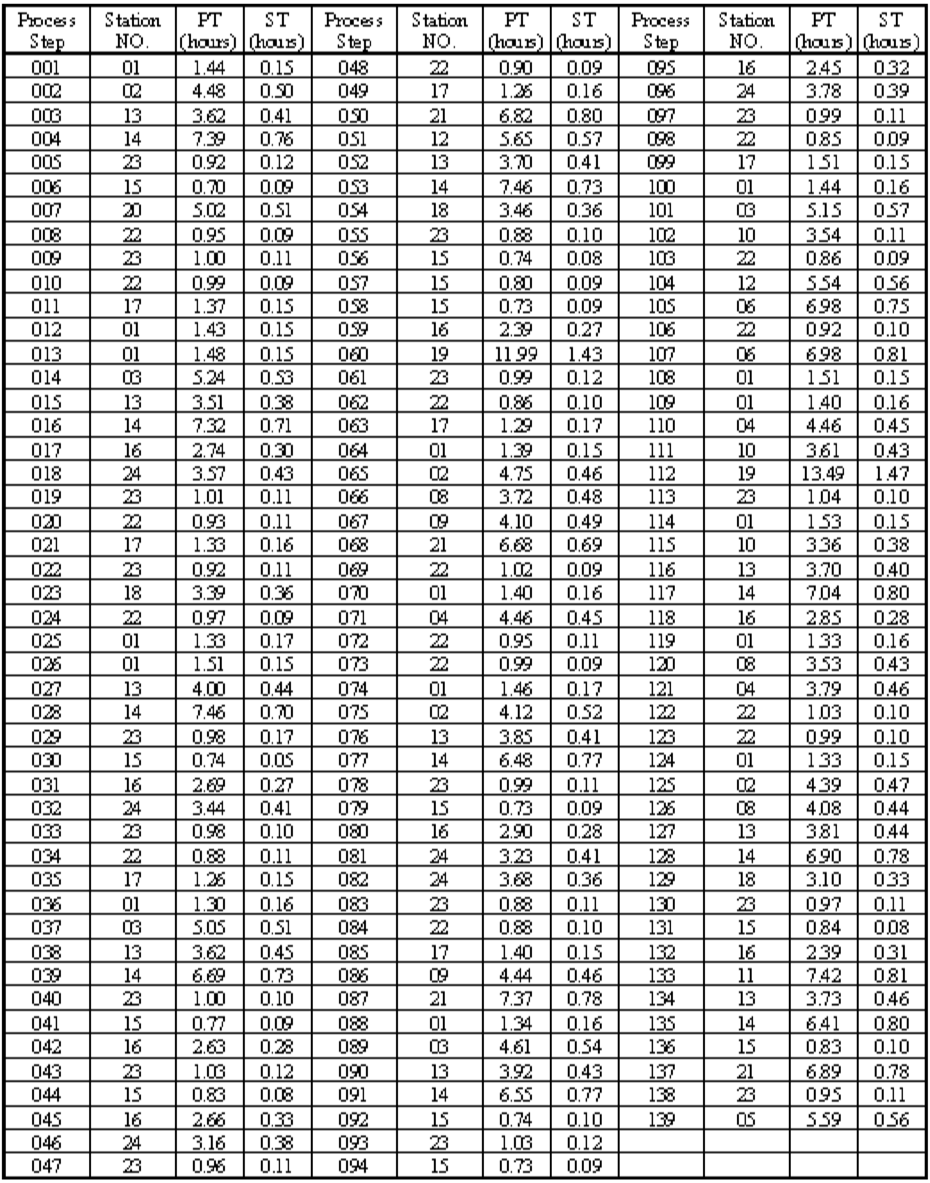
5.2.1. *Lot scheduling*

First, let the release rule be DETERMIN and the arrival interval of input lot be 36 hours. Thus, one lot is released into the wafer factory every 36 hours. Three scheduling rules must be implemented, namely FIFO, LRF, and SSF. Moreover, this simulation contains 10 runs. The first run is for 20,000 hours and the last one runs for 38,000 hours. The Figures 20–22 shows the simulation results, where SSF performs better than the alternatives. Obviously, the cycle time of the lot is much



Process Step	Station NO.	PT (hours)	ST (hours)	Process Step	Station NO.	PT (hours)	ST (hours)	Process Step	Station NO.	PT (hours)	ST (hours)
001	01	1.27	0.16	059	01	1.33	0.16	117	22	0.86	0.09
002	02	4.84	0.51	060	02	4.79	0.46	118	17	1.46	0.15
003	13	3.43	0.45	061	13	3.58	0.41	119	01	1.51	0.17
004	14	7.67	0.74	062	14	6.90	0.80	120	03	4.90	0.57
005	23	0.88	0.10	063	23	0.97	0.10	121	13	3.70	0.38
006	15	0.77	0.09	064	15	0.76	0.09	122	14	6.90	0.77
007	20	4.63	0.49	065	16	2.39	0.27	123	23	1.01	0.12
008	22	0.96	0.10	066	24	3.71	0.37	124	15	0.77	0.08
009	23	1.05	0.11	067	24	3.12	0.39	125	16	2.58	0.30
010	22	0.86	0.11	068	23	1.00	0.11	126	23	1.07	0.11
011	17	1.34	0.16	069	22	0.85	0.10	127	15	0.73	0.09
012	13	3.62	0.39	070	17	1.47	0.15	128	16	2.50	0.31
013	14	7.67	0.71	071	24	3.40	0.39	129	24	3.71	0.41
014	15	0.78	0.09	072	01	1.47	0.16	130	23	1.06	0.12
015	23	0.90	0.12	073	02	4.70	0.53	131	22	0.87	0.11
016	16	2.66	0.28	074	07	5.55	0.63	132	17	1.47	0.16
017	24	3.75	0.43	075	01	1.27	0.16	133	01	1.53	0.15
018	23	0.88	0.10	076	03	4.66	0.50	134	03	5.19	0.50
019	22	0.95	0.11	077	22	0.99	0.10	135	10	3.69	0.44
020	17	1.46	0.16	078	13	4.08	0.43	136	22	0.91	0.11
021	01	1.37	0.15	079	15	0.71	0.08	137	12	5.65	0.66
022	08	4.04	0.45	080	23	1.05	0.10	138	06	7.61	0.85
023	04	4.38	0.46	081	22	0.96	0.11	139	22	0.85	0.10
024	22	0.90	0.11	082	22	0.94	0.11	140	06	7.54	0.74
025	22	0.97	0.10	083	22	0.98	0.10	141	01	1.39	0.16
026	01	1.43	0.15	084	17	1.34	0.15	142	01	1.40	0.15
027	02	4.57	0.50	085	13	3.85	0.41	143	04	4.50	0.44
028	08	3.65	0.45	086	14	7.67	0.74	144	10	3.72	0.37
029	13	3.81	0.43	087	18	3.13	0.38	145	19	13.61	1.45
030	14	6.62	0.81	088	23	0.98	0.10	146	23	0.89	0.12
031	18	3.00	0.39	089	15	0.77	0.09	147	01	1.33	0.14
032	23	1.06	0.10	090	16	2.55	0.32	148	10	3.65	0.42
033	15	0.75	0.09	091	20	4.63	0.49	149	13	3.96	0.40
034	16	2.82	0.32	092	23	0.89	0.12	150	14	7.04	0.83
035	23	0.88	0.12	093	01	1.47	0.15	151	16	2.87	0.28
036	18	2.91	0.37	094	17	1.51	0.17	152	21	6.41	0.76
037	22	0.90	0.11	095	01	1.33	0.17	153	12	5.54	0.63
038	01	1.50	0.17	096	01	1.43	0.16	154	13	3.70	0.40
039	01	1.48	0.17	097	03	4.70	0.53	155	14	6.62	0.73
040	13	3.54	0.45	098	13	3.66	0.45	156	18	3.10	0.37
041	14	6.69	0.80	099	14	6.90	0.82	157	23	0.95	0.10
042	23	0.98	0.10	100	16	2.55	0.28	158	15	0.80	0.09
043	15	0.80	0.09	101	24	3.78	0.41	159	15	0.83	0.08
044	16	2.90	0.29	102	23	1.01	0.11	160	15	0.80	0.08
045	24	3.37	0.38	103	22	0.92	0.10	161	16	2.50	0.31
046	23	0.90	0.10	104	17	1.39	0.17	162	19	11.37	1.25
047	22	0.97	0.09	105	09	4.53	0.42	163	23	0.99	0.12
048	17	1.48	0.15	106	21	6.55	0.73	164	22	1.00	0.11
049	01	1.27	0.16	107	01	1.51	0.16	165	17	1.47	0.15
050	02	4.88	0.47	108	03	4.95	0.57	166	11	7.49	0.80
051	08	3.92	0.46	109	13	3.85	0.41	167	13	3.43	0.44
052	09	3.89	0.51	110	14	7.46	0.46	168	14	6.97	0.85
053	21	6.48	0.80	111	15	0.82	0.10	169	15	0.79	0.10
054	22	0.96	0.11	112	23	1.00	0.10	170	21	7.23	0.76
055	01	1.43	0.16	113	15	0.76	0.09	171	23	1.05	0.12
056	04	4.13	0.42	114	16	2.39	0.29	172	05	5.09	0.56
057	22	0.97	0.10	115	24	3.26	0.37				
058	22	0.98	0.10	116	23	0.92	0.12				

Figure 18. Process flow for route 1.



Process Step	Station NO.	PT (hours)	ST (hours)	Process Step	Station NO.	PT (hours)	ST (hours)	Process Step	Station NO.	PT (hours)	ST (hours)
001	01	1.44	0.15	048	22	0.90	0.09	095	16	2.45	0.32
002	02	4.48	0.50	049	17	1.26	0.16	096	24	3.78	0.39
003	13	3.62	0.41	050	21	6.82	0.80	097	23	0.99	0.11
004	14	7.39	0.76	051	12	5.65	0.57	098	22	0.85	0.09
005	23	0.92	0.12	052	13	3.70	0.41	099	17	1.51	0.15
006	15	0.70	0.09	053	14	7.46	0.73	100	01	1.44	0.16
007	20	5.02	0.51	054	18	3.46	0.36	101	03	5.15	0.57
008	22	0.95	0.09	055	23	0.88	0.10	102	10	3.54	0.11
009	23	1.00	0.11	056	15	0.74	0.08	103	22	0.86	0.09
010	22	0.99	0.09	057	15	0.80	0.09	104	12	5.54	0.56
011	17	1.37	0.15	058	15	0.73	0.09	105	06	6.98	0.75
012	01	1.43	0.15	059	16	2.39	0.27	106	22	0.92	0.10
013	01	1.48	0.15	060	19	11.99	1.43	107	06	6.98	0.81
014	03	5.24	0.53	061	23	0.99	0.12	108	01	1.51	0.15
015	13	3.51	0.38	062	22	0.86	0.10	109	01	1.40	0.16
016	14	7.32	0.71	063	17	1.29	0.17	110	04	4.46	0.45
017	16	2.74	0.30	064	01	1.39	0.15	111	10	3.61	0.43
018	24	3.57	0.43	065	02	4.75	0.46	112	19	13.49	1.47
019	23	1.01	0.11	066	08	3.72	0.48	113	23	1.04	0.10
020	22	0.93	0.11	067	09	4.10	0.49	114	01	1.53	0.15
021	17	1.33	0.16	068	21	6.68	0.69	115	10	3.36	0.38
022	23	0.92	0.11	069	22	1.02	0.09	116	13	3.70	0.40
023	18	3.39	0.36	070	01	1.40	0.16	117	14	7.04	0.80
024	22	0.97	0.09	071	04	4.46	0.45	118	16	2.85	0.28
025	01	1.33	0.17	072	22	0.95	0.11	119	01	1.33	0.16
026	01	1.51	0.15	073	22	0.99	0.09	120	08	3.53	0.43
027	13	4.00	0.44	074	01	1.46	0.17	121	04	3.79	0.46
028	14	7.46	0.70	075	02	4.12	0.52	122	22	1.03	0.10
029	23	0.98	0.17	076	13	3.85	0.41	123	22	0.99	0.10
030	15	0.74	0.05	077	14	6.48	0.77	124	01	1.33	0.15
031	16	2.69	0.27	078	23	0.99	0.11	125	02	4.39	0.47
032	24	3.44	0.41	079	15	0.73	0.09	126	08	4.08	0.44
033	23	0.98	0.10	080	16	2.90	0.28	127	13	3.81	0.44
034	22	0.88	0.11	081	24	3.23	0.41	128	14	6.90	0.78
035	17	1.26	0.15	082	24	3.68	0.36	129	18	3.10	0.33
036	01	1.30	0.16	083	23	0.88	0.11	130	23	0.97	0.11
037	03	5.05	0.51	084	22	0.88	0.10	131	15	0.84	0.08
038	13	3.62	0.45	085	17	1.40	0.15	132	16	2.39	0.31
039	14	6.69	0.73	086	09	4.44	0.46	133	11	7.42	0.81
040	23	1.00	0.10	087	21	7.37	0.78	134	13	3.73	0.46
041	15	0.77	0.09	088	01	1.34	0.16	135	14	6.41	0.80
042	16	2.63	0.28	089	03	4.61	0.54	136	15	0.83	0.10
043	23	1.03	0.12	090	13	3.92	0.43	137	21	6.89	0.78
044	15	0.83	0.08	091	14	6.55	0.77	138	23	0.95	0.11
045	16	2.66	0.33	092	15	0.74	0.10	139	05	5.59	0.56
046	24	3.16	0.38	093	23	1.03	0.12				
047	23	0.96	0.11	094	15	0.73	0.09				

Figure 19. Process flow for route 2.

longer with FIFO. Moreover, FIFO increases rate of cycle time and WIP(work-in-process).

5.2.2. *Lot release*

The scheduling rule is fixed as LRF. The arrival interval of input lot is 36 hours for the constant input of DETERMIN and the mean of POISS. The WIP level is maintained at 600 for CONWIP. This simulation contains 10 runs. The duration of the first run is 20,000 hours while the final run is 38,000 hours. Figure 23 shows that

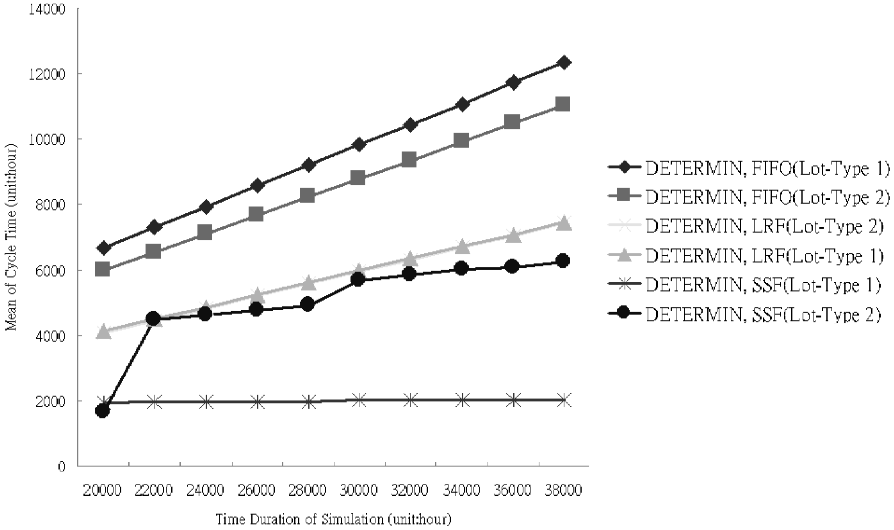


Figure 20. The relationship between lot scheduling rule and cycle time.

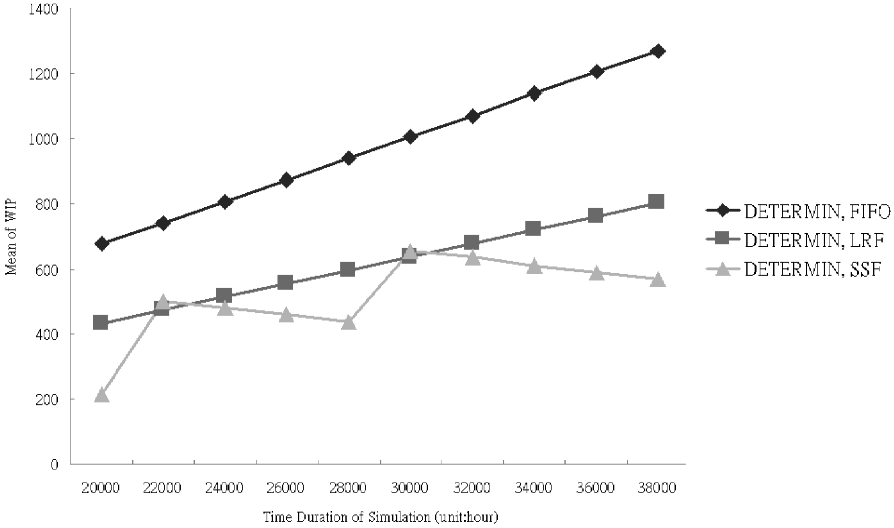


Figure 21. The relationship between lot scheduling rule and cycle time: 20,000 hours.

the cycle time will increase with the time duration of the run. Figure 25 indicates that POISS have more completed lots than DETERMIN and CONWIP during the initial runs with a shorter period. When the run is lengthy, CONWIP and DETERMIN perform better than POISS. Meanwhile, the number of completed lots initially increases quickly and reaches the upper bounds. From Figure 26, CONWIP will maintain a certain WIP level while the others will increase according to the duration of the run. From Figure 23, Figure 25 and Figure 26, CONWIP appears better than

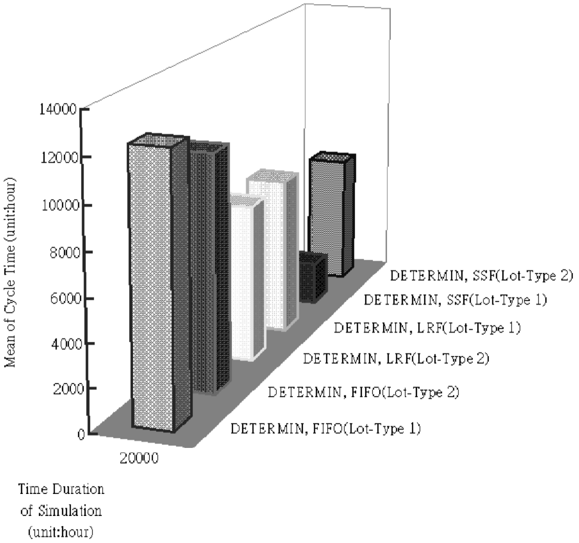


Figure 22. The relationship between lot scheduling rule and work-in-process.

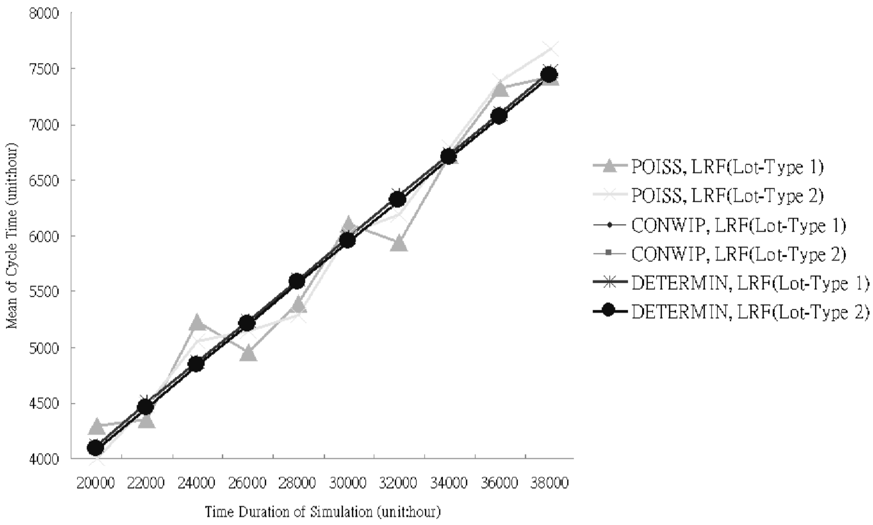


Figure 23. The relationship between lot release rule and cycle time.

DETERMIN. This result indicates that the number of bottleneck machines will increase, when a large number of lots is released into the Fab.

5.2.3. *Arrival interval*

Given that the scheduling rule is FIFO while the release rule is DETERMIN, the arrival interval of the lot was varied from six hours to seventy-two hours. Moreover, this simulation also contains 10 runs. The duration of the first run is 20,000 hours while the final run is 38,000 hours. Figure 27–28 reveals the simulation results. In

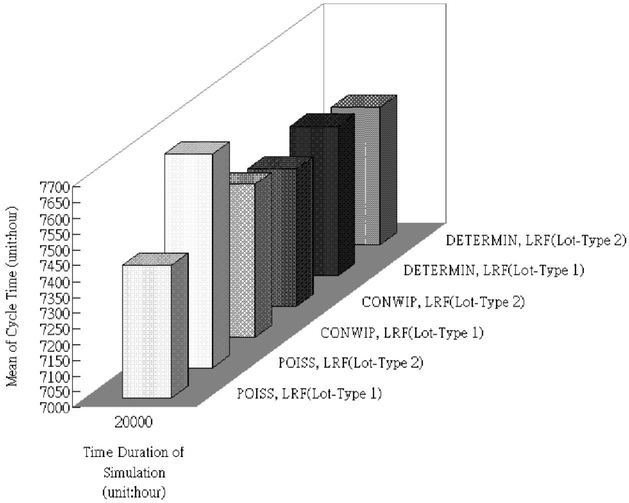


Figure 24. The relationship between lot release rule and cycle time: 20,000 hours.

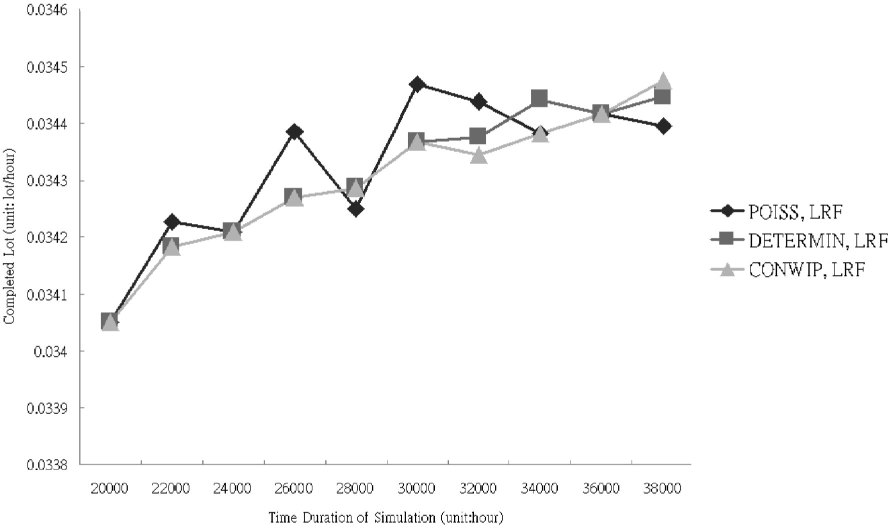


Figure 25. The relationship between lot release rule and number of completed lot.

Figure 27, the throughput is worst when the arrival interval is six hours. This finding implies that wafer flow is congested when many lots are in the factory. Notably, seventy-two hours is not the ideal in Figure 27. This observation indicates that the ideal arrival interval is between 54 hours and 72 hours. Meanwhile, the analysis reveals that the optimal arrival interval is 2.6 days. Figure 28 shows that the WIP increases according to the rate of lot arrival.

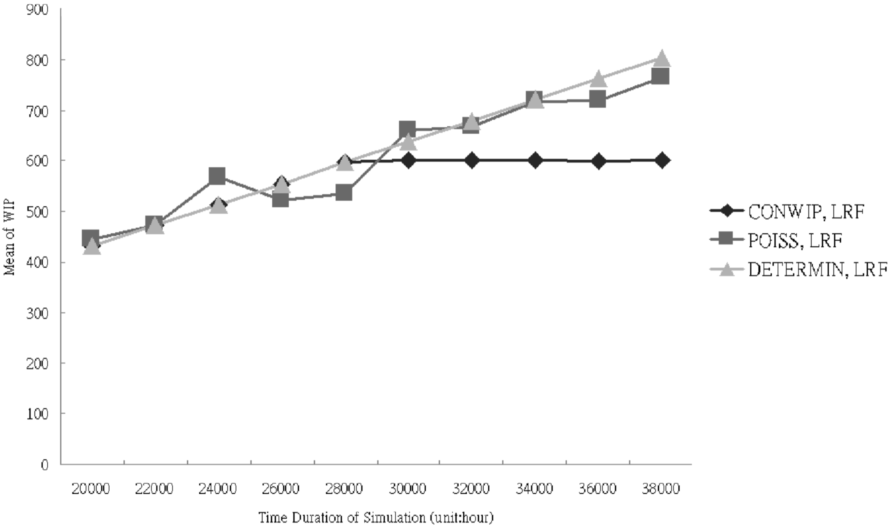


Figure 26. The relationship between lot release rule and work-in-process.

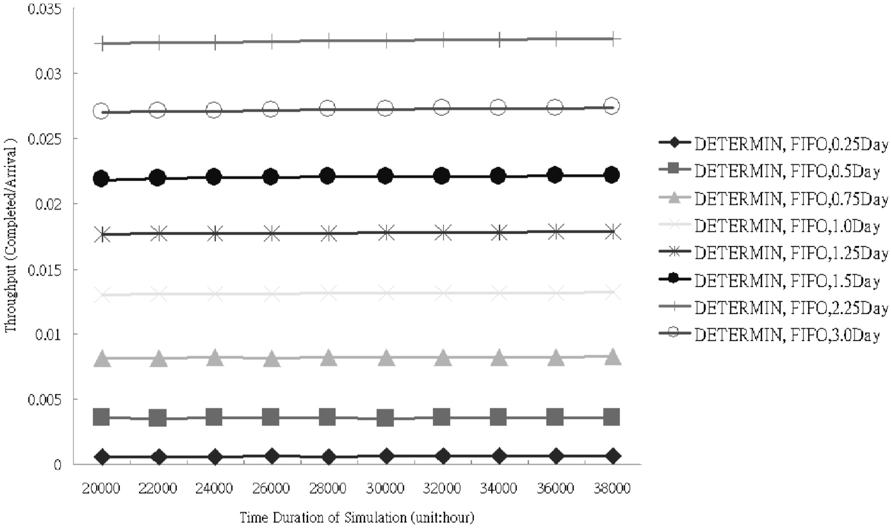


Figure 27. The relationship between lot arrival interval and throughput.

5.2.4. *AGV routing and AGV visiting*

The intrabay 1 is implemented, where the physical layout is a *layout directed graph*, which is shown in Figure 15. Intrabay 1 contains three multiple-load AGVs. There are three rule combinations. (WR, SDF, SPF) denoting that the AGV dispatching rule is WR, visiting rule is SDF, and routing rule is SPF. Meanwhile, (WR, FCFS, SPF) presents that the dispatching rule is WR, visiting rule is FCFS(fist come first serve), and routing rule is SPF. Finally, (WR, FCFS, RANDOM) presents that the dispatching rule is WR, visiting rule is FCFS, and routing rule is RANDOM. There are five runs in this simulation. The AGV speed of the first run is 1/0.25 m/

min. and it is 1/1.75 m/min. for the final run. Figure 29–30 indicates that the gap among the three different combinations increases according to 1/(speed of AGV). This phenomenon indicates that raising the speed of AGV increases performance. Meanwhile, regression analysis indicates that the optimum speed of AGV is 4 (m/min.) in this simulation. Figure 29–30 also shows that the routing rule influences performance than the visiting rule in this simulation experiment. However, there is a stage where the additional speed does not offer a significant increase in performance.

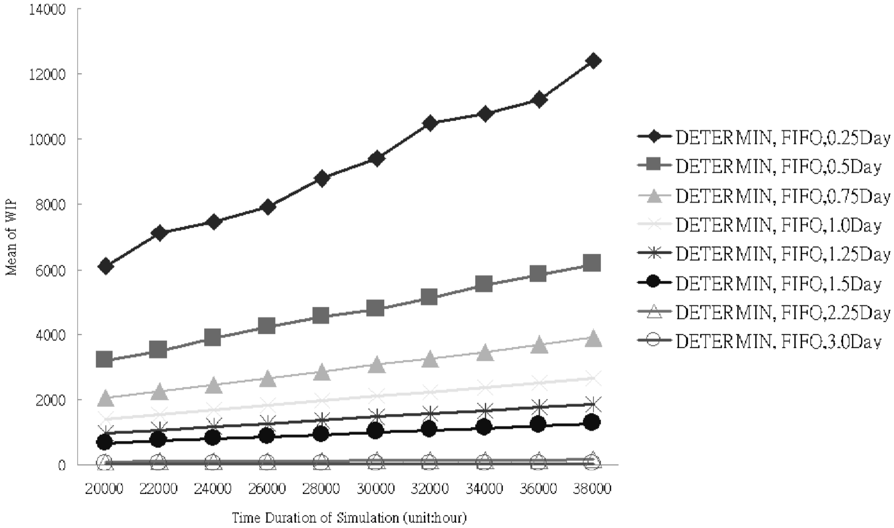


Figure 28. The relationship between lot arrival interval and work-in-process.

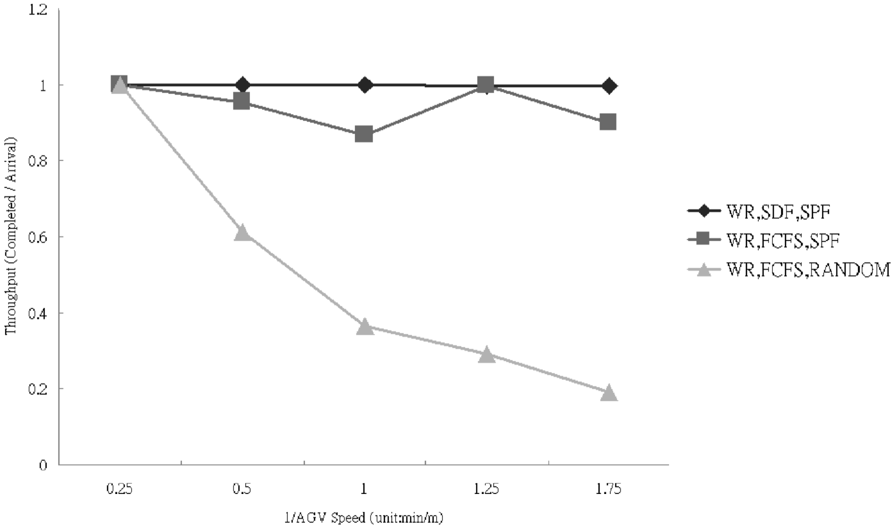


Figure 29. The relationship between speed of AGV and routing rule.

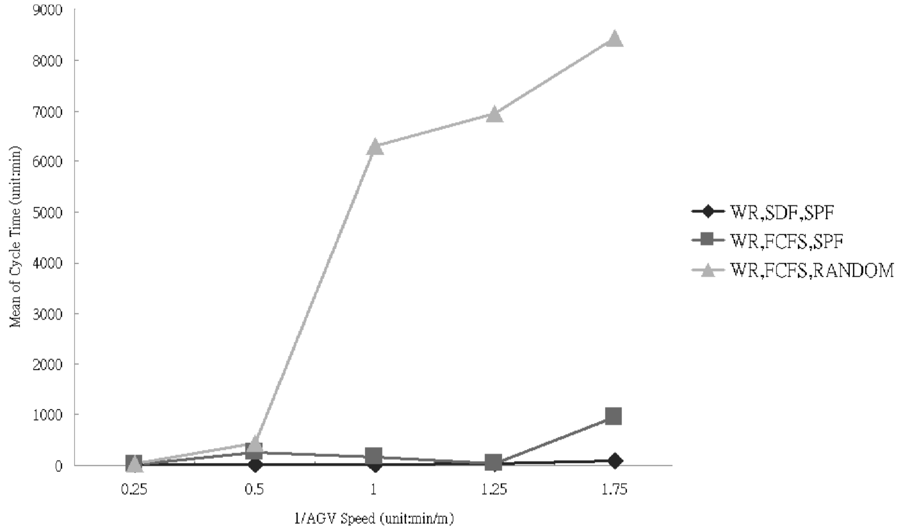


Figure 30. The relationship between speed of AGV and routing rule.

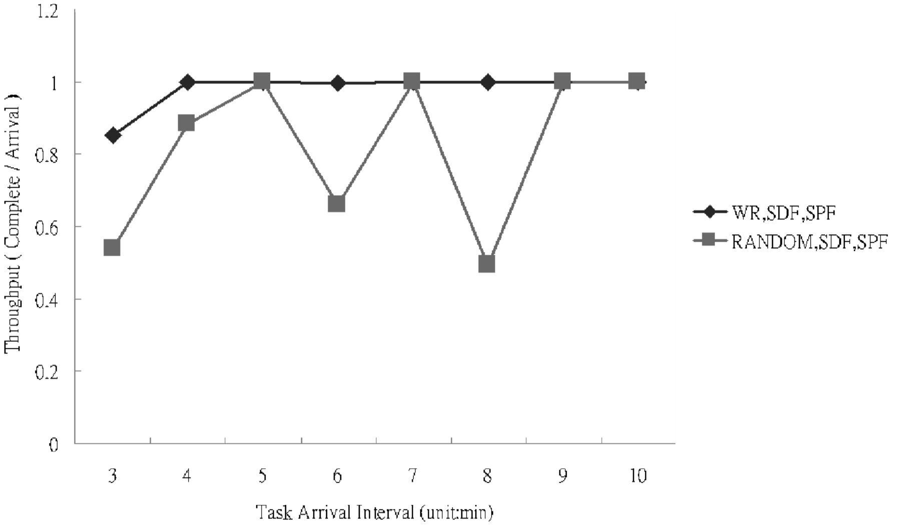


Figure 31. The relationship between visiting rule and task arrival interval.

This number depends on physical layout. In the model presented here, the optimal number from analysis is approximately 5.

5.2.5. *AGV dispatching*

Two combinations must be implemented, (WR, SDF, SPF) and (RANDOM, SDF, SPF). Where (WR, SDF, SPF) denotes that the AGV dispatching rule is WR, visiting rule is SDF, and routing rule is SPF. (RANDOM, SDF, SPF) presents that the dispatching rule is RANDOM, visiting rule is SDF, and routing rule is SPF. The

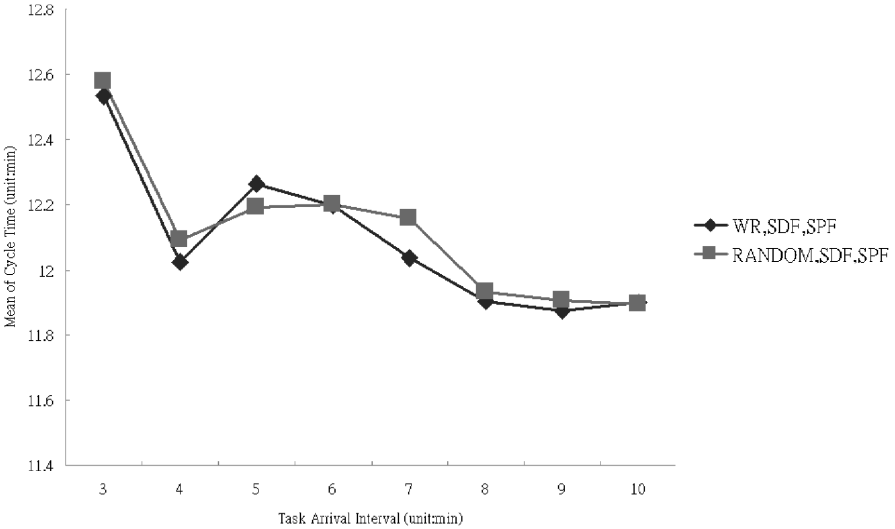


Figure 32. The relationship between visiting rule and task arrival interval.

loading and storing tasks are fed into the interbay 1. The task arrival is DETERMIN. Moreover, this simulation contains eight runs. The task arrival interval of the first run is 3 minutes and that of the last is 10 minutes. Figures 31–32 displays the results. The cycle time increases with task arrival rate. If the number of AGVs is large, we can predict that RANDOM and WR perform very differently. Figure 31 also shows that WR have stable performance than RANDOM.

6. Conclusions

The GSCTPN can model the complex process flows in wafer fabrication efficiently and the detailed manufacturing characteristics such as lot processing, machine setup, machine failure, batch processing, and reworking of defective wafers. Some control policies based on the GSCTPN model optimize certain *a priori* assigned aspects of performance. The plant layout and processing steps resemble those in Zhou and Jeng 1998 and Jeng *et al* 1998. Unlike the model proposed in Zhou and Jeng 1998 and Jeng *et al.* 1998, this work introduces a GSCTPN modelling of a general IC wafer fabrication system. The resulting net model overcomes the 'long' net modelling of the reentrant processing problem. It contains fewer places and transitions. Furthermore, the proposed GSCTPN model considers multiple types of wafers. The proposed GSCTPN model extends the modular modelling approach and formally defines the class of systems that can be dealt with using the approach. Notably, the GSCTPN model already has embedded a control rule embedded into it to prevent vehicle collision problems from occurring. The transporting system aims to introduce a control method into the Petri-Net model to guarantee a jam-free condition among carriers.

The other contribution of this research is in proposing a simulation based performance analysis and schedule adjustment. A schedule can be fine tuned, based on simulation to meet rapidly changing of system parameters without long-run rescheduling. Performance measures are obtained by simulation. The validated model can answer many 'what-if' questions, for example predicting the throughput. Various possible improvements exist. They will be examined in future studies and are described below.

Factor Combination: So far, the factors that are included in this simulation include lot scheduling rule, lot release rule, and lot arrival interval. However, this study does not address many factors that influence performance, including the frequency of machine breakdown and the number of bottleneck machines. Future studies will attempt to find out and combine these influences.

Numerical Analysis: Since lengthy simulations are often necessary to obtain results with sufficient accuracy. Numerical analysis techniques and simulation both have advantages and limitations. Thus, methods of combining analytical/numerical techniques and simulation will be examined in the future.

Acknowledgements

The authors would like to thank Worldwide Semiconductor Manufacturing Corporation (WSMC) for their valuable assistance and cooperation. In addition, the authors would like to thank the National Science Council of the Republic of China for financially supporting this research under Contract No. NSC 87-2212-E-002–068.

References

AGNETIS, A., ARBIB, C. and NICOLO, F., 1990, Part routing in flexible assembly systems. *IEEE Transactions on Robotics and Automation*, **6**, 697–705.

ALLAM, M. and ALLA, H., 1998, Modeling and simulation of an electronic component manufacturing system using hybrid Petri Nets. *IEEE Transactions on Semiconductor Manufacturing*, **11**, 374–383.

ANDERSON, E. J. and NYIRENDA, J. C., 1990, Two new rules to minimize tardiness in a job shop. *International Journal of Production Research*, **28**, 2277–2292.

AYACHE, J. M., COURRTIAT, J. P. and DIAZ, M., 1982, REBUS, a fault-tolerant distribution system for industrial real-time control. *IEEE Transactions on Computing*, **31**, 637–674.

BAER, J. L. and ELLIS, C. A., 1977, Model design and evaluation of a compiler for a parallel processing environment. *IEEE Transactions on Software Engineering*, **3**, 394-405.

BISPO, C. F. G., SENTIEIRO, J. J. S. and HIBBERD, R. D., 1992, Adaptive scheduling for high-volume shops. *IEEE Transactions on Robotics and Automation*, **8**, 696–706.

CHEN, H., HARRISON, J. M., MANDELBAUM, A., VAN ACKERE, A. and WEIN, L. M., 1988, Empirical evaluation of a queuing network model for semiconductor wafer fabrication. *Operation Research*, **36**, 202–215.

CHEN, C. L., LEE, C. S. G. and MCGILLEM, C.D., 1987, Task assignment and load balancing of autonomous vehicles in a flexible manufacturing system. *IEEE Transactions on Robotics and Automation*, **3**, 659–671.

CSATARY, P., NOLAN, D., WOLF, P. and HONOLD, A., 1998, 300mm Fab layout and automation concepts. *Future Fab*, **1**.

DESROCHERS, A. A. and AL-JAAR, R. Y., 1994, *Applications of Petri Nets in Manufacturing Systems: Modeling, Control, and Performance Analysis* (New York: IEEE Press).

DIAZ, M., 1982, Modeling and analysis of communication and cooperation protocols using Petri net based models. *Computer Networks*, **6**.

DOLGUI, A. and OFITSEROV, D., 1997, A stochastic method for discrete and continuous optimization in manufacturing systems. *Journal of Intelligent Manufacturing*, August, 405–413.

DUENYAS, L., FOWLER, J. W. and SCHRUBEN, L. W., 1994, Planning and scheduling in Japanese semiconductor manufacturing. *Journal of Manufacturing Systems*, **13**, 323–332.

France, S., 1982, *Sequencing and Scheduling: an Introduction to the Mathematics of the Job-shop* (New York: Wiley).

Glassey, R. C. and Weng, W. W., 1991, Dynamic batching heuristic for simultaneous processing. *IEEE Transactions on Semiconductor Manufacturing*, **4**, 77–82.

Hillion, H. P. and Proth, J.-M., 1989, Performance evaluation of job-shop systems using timed event-graphs. *IEEE Transactions on Automatic Control*, **34**, 3–9.

Hoitomt, D. J., Luh, P. B., Max, E. and Pattipati, K. R., 1991, Scheduling jobs with simple precedence constraints on parallel machines. *IEEE Control Systems Magazine,* **10**.

Janneck, J. W. and Naedele, M., 1998, Modeling a die bonder with Petri Nets: a case study. *IEEE Transactions on Semiconductor Manufacturing*, **11**, 404–409.

Jeng, M. D., Xie, X. and Chou, S. W., 1998, Modeling, qualitative analysis, simulation, and performance evaluation of the etching area in an IC wafer fabrication system using Petri Nets. *IEEE Transactions on Semiconductor Manufacturing*, **11**, 358–373.

Johri, P. K., 1993, Practical issues in scheduling and dispatching in semiconductor wafer fabrication. *Journal of Manufacturing Systems*, **12**, 474–485.

Kasturia, E., Docesare, F. and Desrochers, 1988, Real-time control of multilevel manufacturing systems using coloured Petri nets. *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 1114–1119.

Kim, Y.-D., Kim, J.-U., Lim, S.-K. and Jun, H.-B., 1998, Due-date based scheduling and control policies in a multi-product semiconductor wafer fabrication facility. *IEEE Transactions on Semiconductor Manufacturing*, **11**, 155–164.

Kim, Y.-D., Lee, D.-H. and Kim, J.-U., 1998, A simulation study on lot release control, mask scheduling, and batch scheduling in semiconductor wafer fabrication facilities. *Journal of Manufacturing Systems*, **17**, 107–117.

Leachman, R. C. and Hodges, D. A., 1996, Benchmarking semiconductor manufacturing. *IEEE on Semiconductor Manufacturing*, **9**, 158–169.

Lee, K. H. and Favrel, J., 1985, Hierarchical reduction method for analysis and decomposition of Petri nets. *IEEE Transactions on Systems, Man and Cybernetics*, **15**, 272–280.

Lee, K. H., Favrel, J. and Baptiste, P., 1987, Generalized Petri net reduction method. *IEEE Transactions on Systems, Man and Cybernetics*, **17**, 297–303.

Lee, L. Y. and Dicesare, F., 1992, Experimental study of a heuristic function for FMS scheduling. *JAPAN/USA Symposium on Flexible Automation*, **2**, 1171–1177.

Li, S., Tang, T. and Collins, D. W., 1996, Minimum inventory variability schedule with applications in semiconductor fabrication. *IEEE Transactions on Semiconductor Manufacturing*, **9**, 145–1496.

Lin, S. and Whung, H.-P., 1998, Modeling and emulation of a furnace in IC Fab based on colored-timed Petri Net. *IEEE Transactions on Semiconductor Manufacturing*, **11**, 410-420.

Luh, P. B., Hoitomt, D. J., Max, E. and Pattipati, K. R., 1990, Scheduling generation and reconfiguration for parallel machines. *IEEE Transactions on Robotics and Automation*, **6**, 687–696.

Mahmoodi, F., Dooley, K. J. and Starr, P. J., 1990, An investigation of dynamic group scheduling heuristics in a job shop manufacturing cell. *International Journal of Production Research*, **28**, 1695–1711.

Missale, R., 1998, Expectations on automation for 300mm semiconductor Fab operations. *Future Fab*, **1**.

Molloy, M. K., 1982, Performance analysis using stochastic Petri nets. *IEEE Transactions on Computers*, **31**, 913–917.

Molloy, M. K., 1985, Discrete time stochastic Petri nets. *IEEE Transactions on Software Engineering*, **11**, 417–423.

Mosley, S. A., Teyner, T. and Uzsoy, R. M., 1998, Maintenance scheduling and staffing policies in a wafer fabrication facility. *IEEE Transactions on Semiconductor Manufacturing*, **11**, 316–323.

Murata, T., 1989, Petri nets: properties, analysis and applications. *Proceedings of the Institute of Electrical and Electronics Engineers*, **77**, 541–580.

Narahari, Y. and Khan, L. M., 1997, Modeling the effect of hot lots in semiconductor manufacturing systems. *IEEE Transactions on Semiconductor Manufacturing*, **10**, 185–188.

NARAHARI, Y. and VISWANADHAM, N., 1985, A Petri net approach to modeling and analysis of flexible manufacturing system. *Annals of Operations Research*, **3**, 449–472.

PLATA, J. J., 1997, 300 mm Fab design – a total factory perspective. *Future Fab*, **1**, 21–7.

RAO, G. N. and WANG, H. P., 1991, Evaluation and dynamic allocation of AS/RS control rules: a simulation study. *The International Journal of Advanced Manufacturing Technology*, 62–82.

RAMAMOORTHY, C. V. and HO, G. S., 1980, Performance evaluation of asynchronous concurrent systems using Petri Nets. *IEEE Transactions on Software Engineering*, **6**, 440–449.

ROBERTSON, F. and GARGINI, P., 1998, 300mm conversion capability. *Future Fab*, **1**.

ROGERS, R. V. and PRESTON WHITE, K., JR, 1991, Algebraic, mathematical programming, and network models of the deterministic job-shop scheduling problem. *IEEE Transactions on Systems, Man, and Cybernetics*, **21**.

SAIT, S. M. and YOUSSEF, H., 1995, *VLSI Physical Design Automation: Theory and Practice* (New York: McGraw-Hill).

SARIN, S. C., AHN, S. and BISHOP, A. B., 1988, An improved branching scheme for the branch and bound procedure of scheduling *n* jobs on *m* parallel machines to minimize total weighted flowtime. *International Journal of Production Research* **26**, 1183–1191.

SAWIK, S., 1990, Modelling and scheduling of a flexible manufacturing system. *European Journal of Operational Research*, **45**, 177–190.

SHAW, M. J., 1988, Knowledge-based scheduling in flexible manufacturing systems: an integration of pattern-directed inference and heuristic search. *International Journal of Production Research*, **26**, 821–844.

SRINIVASAN, R. S., 1998, Modeling and performance analysis of cluster tools using Petri Nets. *IEEE Transactions on Semiconductor Manufacturing*, **11**, 394–403.

SZE, S. M., 1983, *VLSI Technology* (New York: McGraw-Hill).

UZSOY, R., LEE, C.-Y. and MARTIN-VEGA, L. A., 1992, A review of production planning and scheduling models in the semiconductor industry partFF1: system characteristics, performance evaluation and production planning. *IIE Transactions*, **24**, 47–60.

UZSOY, R., LEE, C.-Y. and MARTIN-VEGA, L. A., 1994, A review of production planning and scheduling models in the semiconductor industry partFFo: shop-floor control. *IIE Transactions*, **26**, 44–55.

UZSOY, R., MARTIN-VEGA, L. A., LEE, C.-Y. and LEONARD, P. A., 1991, Production scheduling algorithm for a semiconductor test facility. *IEEE Transactions on Semiconductor Manufacturing*, **4**, 270–279.

VALAVANIS, K. P., 1990, On the hierarchical modeling analysis and simulation of flexible manufacturing systems with extended Petri nets. *IEEE Transactions on Systems, Man and Cybernetics*, **20**, 94–109.

XIONG, H. H. and ZHOU, M., 1998, Scheduling of semiconductor test facility via Petri Nets and hybrid huristic sarch. *IEEE Transactions on Semiconductor Manufacturing*, **11**, 384–393.

ZHOU, M. and DER JENG, M., 1998, Modeling, nalysis, simulation, scheduling, and control of semiconductor manufacturing system: a Petri net approach. *IEEE Transactions on Semiconductor Manufacturing*, **11**, 333–357.

ZHOU, M. C., DICESARE, F. and RUDOLPH, D., 1990, Control of a flexible manufacturing system using Petri nets. *Proceedings of the IFAC Congress*, **9**, 43–48.

ZURAWSKI, R. and ZHOU, M., 1994, Petri Nets and industrial applications: a tutorial. *IEEE Transactions on Industrial Electronics*, **41**, 567–583.