Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

INTEGRATED RTS-RTD SIMULATION FRAMEWORK FOR SEMICONDUCTOR MANUFACTURING SYSTEM

Seongho Cho[1], Donguk Kim[2], Sangchul Park[1]

[1]Dept. of Industrial Engineering, Ajou University, Suwon, REPUBLIC OF KOREA
[2]LG Production and Research Institute, LG-ro 222 Pyeongtaek, 17709, REPUBLIC OF KOREA

ABSTRACT

The complexity of modern semiconductor fabrication (FAB) systems makes it difficult to implement integrated simulation systems that combine production and logistics simulators. As a result, these simulators have traditionally been developed independently. However, in actual FAB operations, information exchange between Real-Time Schedulers (RTS) and Real-Time Dispatchers (RTD) coordinates production activities. To address this issue, we propose a coupled RTS–RTD simulation framework that integrates production and logistics simulators into a unified environment. In addition, we introduce a dynamic decision-making rule that enables flexible responses when logistical constraints prevent execution of the original production schedule. Simulation experiments were conducted using the SMT2020 and SMAT2022 datasets. The results show that selectively following RTD decisions, instead of strictly adhering to RTS-generated schedules, can significantly improve production efficiency in FAB operations.

1 INTRODUCTION

The advancement of the Fourth Industrial Revolution has led to a surge in semiconductor demand. In particular, with the development of AI and IoT industries, the demand for semiconductors has further increased to support large-scale computing and data processing (Gartner. 2025). In response, global companies are prioritizing R&D expansion and production capabilities, while governments are positioning the semiconductor industry as a key national sector. With competition among companies, FABs are becoming larger and more complex. The complexity of modern FABs now precludes effective problem-solving through conventional intuitive or simplistic approaches (Sun and Rose 2015; Sun et al. 2016).

The semiconductor chip manufacturing process consists of hundreds of processing steps and is highly complex due to factors such as re-entrant flow, lot prioritization, and queue time constraints (Kopp et al. 2020a). As a result, it takes approximately two to three months for a single wafer to pass through a FAB and be completed as a semiconductor chip. This prolonged production cycle presents significant challenges in production management, leading to financial losses for manufacturers due to factors such as overproduction and delivery delays. Consequently, numerous studies have been conducted on operational optimization for semiconductor production systems to enhance efficiency and mitigate these issues.

For practical optimization of production systems, it is necessary to adopt an approach that incorporates both production scheduling and logistics. El Khayat et al. (2006) proposed both a mathematical programming model and a constraint programming approach for integrating material handling and scheduling in a job shop environment. While their work provides valuable insights into the integration of these two domains, the proposed models were validated only in small-scale scenarios involving fewer than ten machines and five vehicles, which limits their scalability and generalizability to large, real-world systems. Liang et al. (2022) proposed a Mixed-Integer Linear Programming (MILP) model that integrates material handling and production scheduling. Although their approach extends the mathematical rigor of prior studies, it relies on simplifying assumptions such as constant transportation times between workstations and a relatively static setting with fewer than ten Automated Guided Vehicles (AGVs). These assumptions limit the

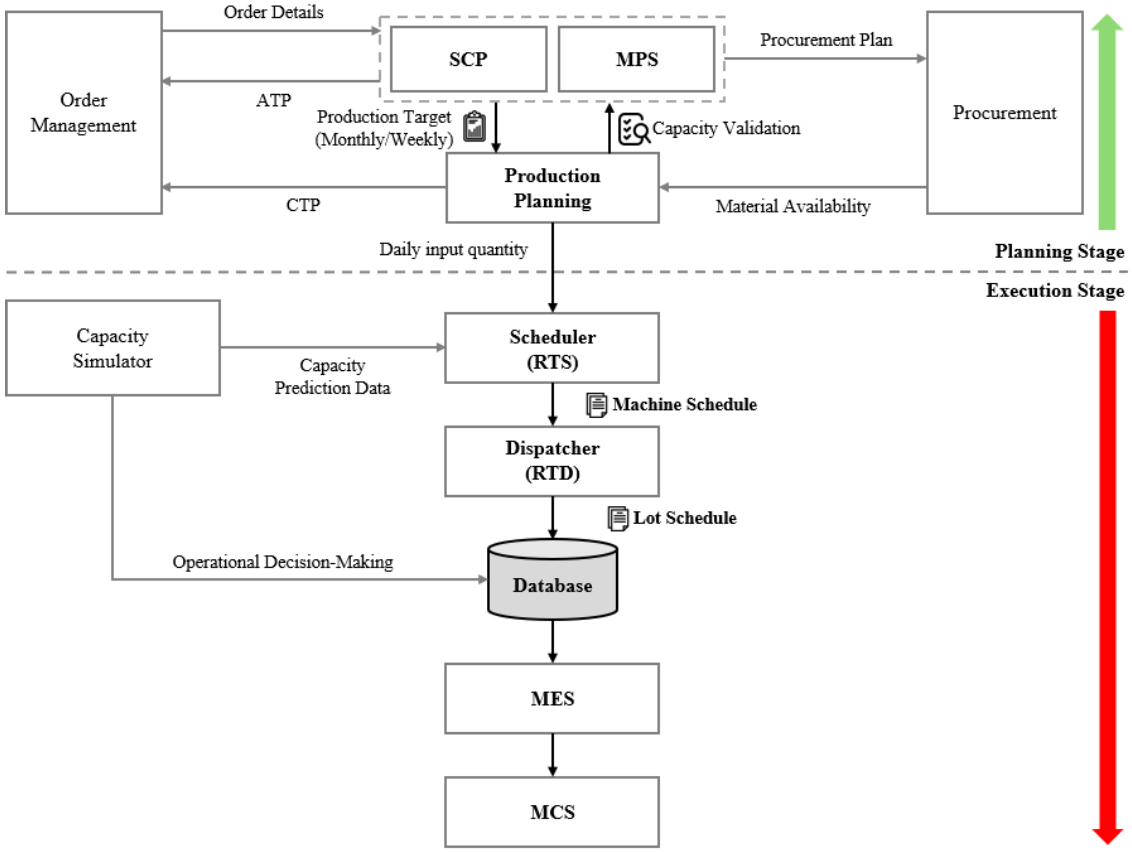

Figure 1: Framework of semiconductor manufacturing operation system.

model's ability to represent the dynamic and uncertain nature of large-scale manufacturing systems. These mathematical optimization-based approaches offer theoretical rigor and precise solutions, but they fall short in fully capturing the scale, dynamics, and complexity of real-world FABs. As a result, simulation-based approaches have been widely explored for their flexibility in modeling complex systems with diverse variables and resource interactions.

Simulations for analyzing production systems incorporate both production scheduling and logistics. However, despite advances in computing power, the complexity of large-scale semiconductor manufacturing hinders the development of integrated simulations. As a result, simulations for FABs have been studied independently, with Automated Material Handling System (AMHS) and production scheduling each adopting distinct approaches to achieve their respective objectives (Sakr et al. 2023). Several studies have focused on optimizing Real-Time Schedulers (RTS) for production systems and Real-Time Dispatchers (RTD) for AMHS. (Gupta and Sivakumar 2002; Ghasemi et al. 2024). These studies have been tested and validated using independent simulations and have contributed significantly to enhancing the performance of both systems. As shown in Figure 1, RTS and RTD in large-scale FABs interact closely during the execution stage. RTS establishes equipment-level work plans, while RTD formulates lot-level work schedules. This indicates that relying solely on independent systems fails to capture the full complexity of FABs. This highlights the importance of a simulation environment that simultaneously integrates RTS and RTD.

In this paper, we propose a coupled simulation framework that integrates RTS and RTD simulators. We develop a unified simulation system by enabling data exchange between RTD and RTS and conduct simulations that reflect the real-world conditions of FABs. Additionally, we introduce a dynamic decision-making rule that enables the system to adopt RTD when RTS results are infeasible due to logistical constraints. This framework enables a representation of the modern FAB complexity and provides practical insights for manufacturers. The contributions of this study are summarized as follows:

- We develop a coupled simulation framework for RTS and RTD.
- We establish a dynamic decision-making rule considering logistical constraints.
- We provide a realistic representation of semiconductor FAB operations and support their optimization.

The rest of this paper is structured as follows: Section 2 presents the dataset for the FAB simulator and provides an overview of the two FAB simulators used for RTS and RTD. Section 3 details the method for coupling RTS and RTD simulators. Section 4 introduces the dynamic decision-making rule for RTD. Section 5 provides the experimental setup and outcomes. Finally, Section 6 concludes the paper and discusses the directions for future research.

2 FAB SIMULATOR

Two simulators are utilized to replicate the FAB environment. One focuses on scheduling, while the other handles dispatching and AMHS. To perform a realistic FAB simulation, it is necessary to utilize a dataset that reflects the actual FAB. For this purpose, we use the Semiconductor Manufacturing Testbed 2020 dataset (SMT2020) (Kopp et al. 2020b). SMT2020 is a dataset that reflects the high complexity and large scale required in modern FABs and consists of four distinct datasets. Table 1 presents a summary of each dataset within SMT2020. In this study, the fourth dataset is utilized, which represents the highest level of complexity.

Table 1: Comparison of SMT2020 datasets.

Dataset	Plan Type	# of Products	Due Date	Engineering Lot
1	MTS	2 products	Not Required	Not Contained
2	MTO	10 products	Required	Not Contained
3	MTS	2 products	Not Required	Contained
4	MTO	10 products	Required	Contained

However, SMT2020 does not include AMHS details. To simulate RTD, a dataset that includes AMHS within the same environment is required. In this regard, we utilize the Semiconductor Manufacturing with AMHS Testbed dataset (SMAT2022), developed by Lee et al. (2022). It extends the SMT2020 model by incorporating an AMHS system. The dataset is designed to simulate Overhead Hoist Transport (OHT), which transports lots along a dedicated rail. Additionally, it includes track buffers and stockers for lot storage. Table 2 presents the configuration of SMAT2022, while Figure 2 illustrates its OHT rail layout.

Table 2: Configuration of SMAT 2022.

Group	Feature
Layout	Spine Configuration 3 Interbays 40 Intrabays
AMHS	500 OHTs 734 ZCUs
Buffers	18,000 STB / UTBs 40 Stockers

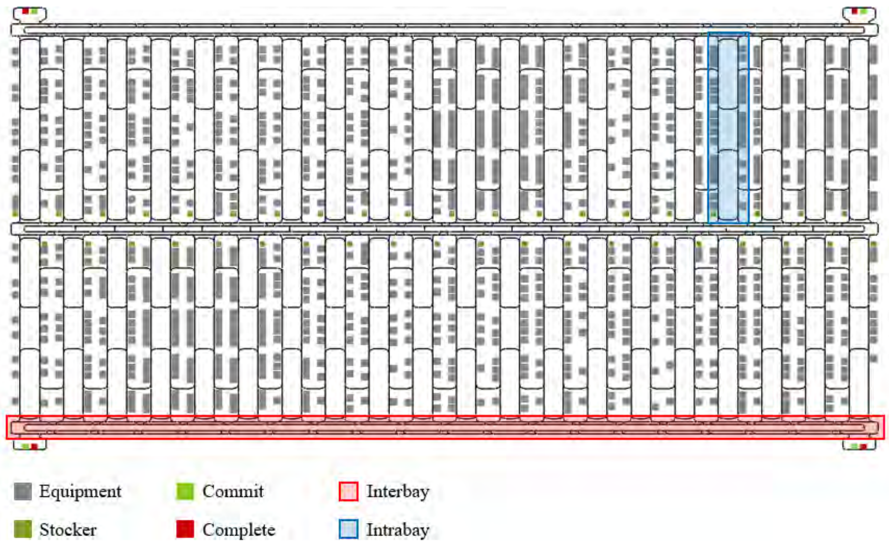

Figure 2: OHT rail layout of SMAT2022.

2.1 Simulator for Real-Time Scheduler

In this paper, we adopt MOZART (VMS Solutions 2025), simulation software for machine scheduling. MOZART is a comprehensive solution that includes methodologies, development tools, and libraries that aim to model real-world manufacturing environments. By utilizing virtual models, MOZART supports the effective construction and operation of production planning and scheduling systems. MOZART consists of various software modules designed for specific systems. In this study, we utilize MOZART LSE Studio, which provides a Discrete Event System Specification (DEVS) based what-if analysis environment (Concepcion and Zeigler 1988).

2.2 Simulator for Real-Time Dispatcher

In addition to MOZART, we employ the PINOKIO Simulator (CARLO 2025), for the RTD simulator. PINOKIO is a general-purpose solution based on DEVS for what-if simulations, similar to LSE Studio. It optimizes production system operations by verifying layouts, enhancing logistics, and refining planning through simulation. PINOKIO provides a high degree of modeling flexibility a flexible environment that developers can easily customize. Additionally, it supports custom modeling for lower-level control systems such as Manufacturing Execution System (MES), Material Control System (MCS), and OHT Control System (OCS). It also enables custom modeling of vehicles and equipment that operate under these control systems. Figure 3 presents the UI of PINOKIO, which supports the customization of both logical and geometric models. This capability to model production systems with distinct characteristics, such as those found in semiconductor manufacturing, led to the decision to adopt this software for our study.

3 COUPLING METHODOLOGY FOR RTD AND RTS SIMULATORS

Generally, different programs are developed for different purposes, and differences in data formats, platforms, and framework compatibility make integration challenging. However, when programs are designed for the same target system, they inherently share common elements, even if their data formats and definitions differ. In particular, LSE Studio and PINOKIO are both simulators for production systems. Their respective datasets, SMT2020 and SMAT2022, represent the same system, suggesting the presence of shared components. This section describes the development of an interface for communication between the two simulators, data exchange, and the operational mechanism of the final framework.

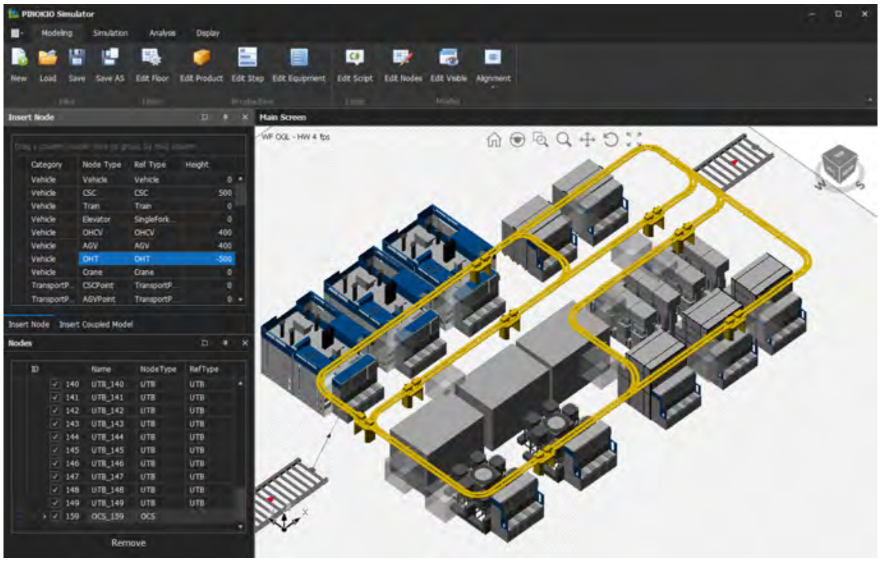

Figure 3: PINOKIO Simulator.

3.1 Communication System for Independent Simulators

One of the major challenges in integrating two simulators is the potential discrepancy in their simulation time. Even when simulating the same system, differences in how each simulator manages time can lead to inconsistencies in object states. This, in turn, affects the overall reliability of the simulation results. To address this issue, this study implements an interface between the simulators using Transmission Control Protocol (TCP). TCP is a protocol that ensures the sequential transmission of data packets and provides reliable data transfer, thereby maintaining consistency in simulation states and enabling synchronized communication between the simulators (Postel 1981). Figure 4 illustrates the implemented communication system, designed based on the data flow of a semiconductor manufacturing system, where RTS transmits the scheduling results to RTD for execution. In this system, RTS recurrently transmits scheduling results to RTD for execution until the simulation ends. A server socket is developed in LSE Studio, while a client socket is implemented in PINOKIO, ensuring consistency with the data flow of the FABs. We use the ZeroMQ (ZMQ) library for communication and Google Protocol Buffers for data serialization (Hintjens 2013; Currier 2022).

In addition to communication, maintaining the same level of abstraction in the simulation models is crucial for successful integration. Differences in the level of abstraction can lead to discrepancies between models due to variations in data interpretation. To prevent this issue, the models of each simulator have to be designed to operate at the same level of abstraction. For example, if the production scheduler generates a schedule considering the preventive maintenance (PM) plan of specific equipment, the logistics simulator must also reflect this PM schedule in its operations. In other words, when the production scheduler incorporates equipment maintenance into process scheduling, the logistics simulator should adhere to the same maintenance schedule to adjust material flow and transportation accordingly. To achieve this, we designed simulation models for both the production scheduler and logistics simulator to operate at the same level of abstraction, minimizing inconsistencies. This setup enables a more accurate representation of a real semiconductor FAB.

3.2 Data Exchange Between RTS and RTD

Through the communication system, two simulators periodically exchange data to achieve their respective objectives. This enables dynamic integration between the production schedule and logistics information within the FAB. The RTD simulator captures a snapshot of the system at a specific simulation time and

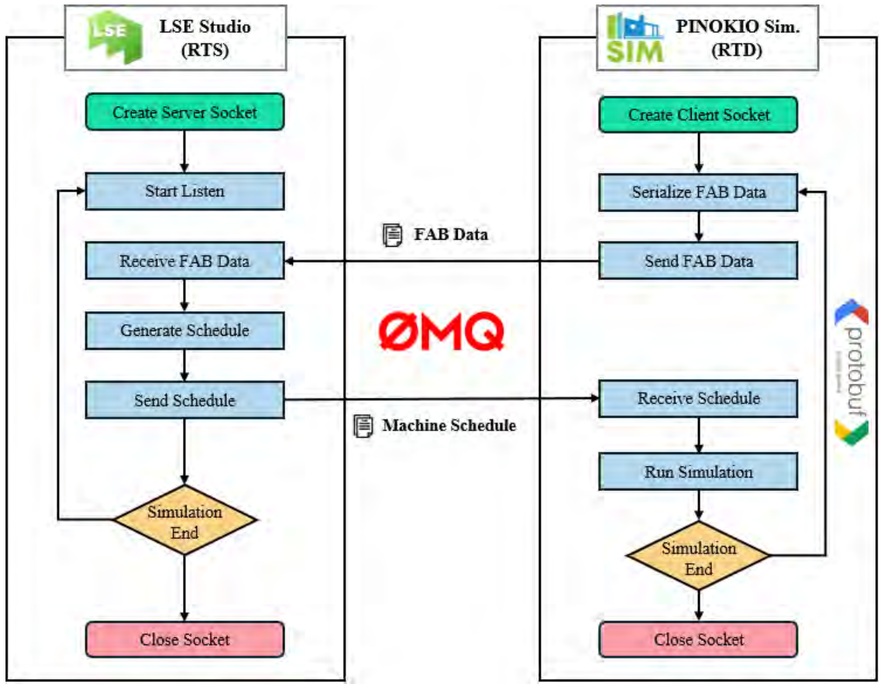

Figure 4: Communication system for independent simulators.

transmits it to the RTS simulator. As shown in Table 3, this data includes information such as Work In Process (WIP), PM schedules for equipment, transport time, and simulation configuration parameters.

The WIP information is essential for scheduling, as it provides details on all lots currently in process. This includes the lot identifier (Lot ID), the identifier of the equipment currently processing each lot (Equipment ID), and the timestamp of the most recent state update. The PM information of equipment is utilized to determine its availability. PM is generally categorized into two types: time-based and wafer-count-based maintenance. Time-based maintenance refers to periodic inspections that are performed after a predefined time interval since the last maintenance. In contrast, wafer-count-based maintenance is initiated when the number of wafers processed by the equipment exceeds a certain threshold. The data includes the maintenance type, status data depending on the type, the equipment identifier, and the Mean Time to Repair (MTTR) required for equipment recovery. The Transport time reflects the material flow within the FAB and provides OHT transfer times between all pairs of bays. As shown in Figure 5, the transfer distance is defined as 1 for intra-bay transfer and 2 for inter-bay transfer. A typical production scheduler treats transport time as a constant or as values derived from a probability distribution. In contrast, this study utilizes data collected from an RTD simulator to enable more realistic scheduling. The simulation configuration includes the start time and the duration of the snapshot.

The RTD simulator transfers FAB snapshot data to the RTS simulator via the communication system, and the RTS simulator performs scheduling based on this information. Table 4 presents the scheduling results generated by the RTS simulator, which are subsequently returned to the RTD simulator and applied to simulation execution. By iteratively performing this process between the RTS and RTD simulators, scheduling mismatches caused by FAB variability can be minimized. Finally, we developed a coupled simulation framework that integrates RTS and RTD, which enables high-fidelity modeling of actual FAB operations.

Table 3: Description of snapshot data components.

Data Category	Details
WIP	Equipment ID Lot ID Lot State Lot Priority Lot Due Date Last State Change Time Product ID Current Step ID Wafer Quantity
EQP PM	Equipment ID Wafer Count for PM Last PM Started Time PM Type MTTR
Transport Time	From Bay ID Destination Bay ID Transfer Time
Simulation Configuration	Simulation Period Simulation Start Time

Figure 5: Definition of bay distance.

4 DYNAMIC DECISION MAKING RULE FOR RTD

In this study, we propose a dynamic decision-making method to improve production efficiency in FABs by utilizing the previously introduced coupled simulation framework. The method dynamically balances reliance between RTS and RTD, according to their respective roles. The RTS generates schedules to optimize overall production operations, but it has limitations in accounting for logistical congestion. As a result, two types of production losses may occur.

Table 4: Description of schedule data components.

Data Category	Details
Production Schedule	Equipment Group ID Equipment ID Lot ID Lot Due Date Lot Priority Wafer Quantity Current Step ID Arrival Time to Next Equipment Job Start Time for Next Equipment Job End Time for Next Equipment

First, if a lot arrives at a machine that is still processing the previous job, the lot must wait. This may violate the queue-time constraint, which ultimately leads to a production loss (Klemmt and Mönch 2012). Second, when the arrival of the next lot is delayed, the machine may become idle, thereby decreasing equipment utilization. In such cases, the RTD should reevaluate the schedule rather than adhering to it without consideration of the current context. If the expected production loss exceeds the potential benefit of following the original schedule, it is more reasonable for the RTD to make independent dispatch decisions.

The proposed method is based on the expected arrival time of the lot T_a and the expected idle time of the equipment T_e. As shown in Equation 1, T_a can be calculated by adding the bay-to-bay transfer time T_{move} to the dispatch time T_d.

$$T_a = T_d + T_{\text{move}} \tag{1}$$

As shown in Equation 2, T_e is obtained by summing the remaining processing time T_{remain} and the processing times of the reserved lots $T_{reserved}$, starting from T_d.

$$T_e = T_d + T_{\text{remain}} + \sum T_{\text{reserved}} \tag{2}$$

Finally, Equation 3 is used to calculate the loss value L, which accounts for both possible cases of production inefficiency. Figure 6 illustrates how the loss is predicted at the dispatch time. If the resulting L exceeds a predefined threshold r, it is considered more efficient to follow the RTD decision rather than the schedule generated by the RTS. In this study, simulations are conducted by gradually varying the value of r in order to determine an appropriate threshold.

$$L = \max(0, T_e - T_a) + \max(0, T_a - T_e) \tag{3}$$

5 EXPERIMENTAL SETUP AND RESULTS

For the experiments, SMT2020 is modeled using LSE Studio, while SMAT2022 is implemented in the PINOKIO simulator. As summarized in Table 5, we conducted a one-year simulation warm-up period to reflect realistic FAB operations, ensuring that FAB reached a stable state with sufficient ongoing production activity.

During simulation, the RTD simulator was set to make decisions every simulated hour. The RTS simulator generated new schedules for a two-day period at a time. As shown in Table 6, the operating ratios of each system during the simulation were similar. The execution times of each system during the simulation were similar. The measured times include not only computation but also data serialization and transmission. Specifically, it took an average of 17.4 seconds for the MOZART RTS and an average of 17.6

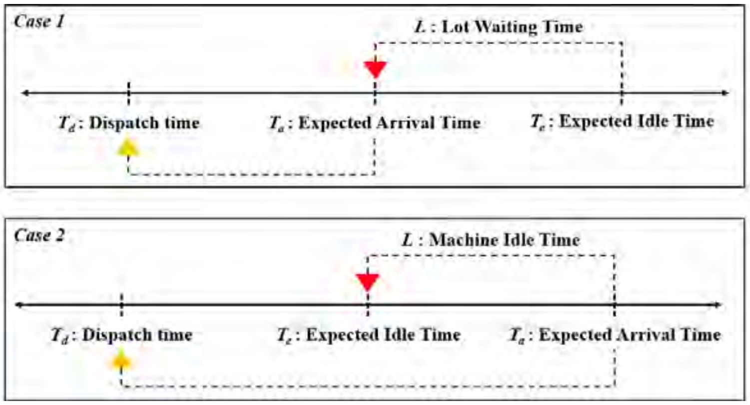

Figure 6: Loss prediction process at dispatch time..

seconds for the PINOKIO RTD simulator. These simulations were performed on a system equipped with an running Windows 10, equipped with an Intel Core i7-14700K CPU, 32 GB of RAM, and an NVIDIA GeForce RTX 4060 Ti GPU.

Table 5: Experimantal setup.

Parameter	Value
Schedule Generation Interval	1 Hour
Schedule Horizon	2 Days
Warm-up Period	1 Year
Total Simulation Time	90 Days
Threshold r	0, 5, 10, 15, 20, 25, 30, ∞

Table 6: System execution time during simulation.

Simulation Type	Elapsed Time
PINOKIO Sim.	17619ms
MOZART LSE Studio	17470ms

The total simulation period for each experiment was set to three months. To evaluate production efficiency under different thresholds, three performance indices were measured; 1) Throughput, 2) On-time delivery rate, and 3) Equipment utilization. The threshold r was varied across eight levels, starting from 0 and increasing in increments of 5 minutes, up to a theoretical maximum ($r = \infty$) that corresponds to fully following the RTS schedule.

Table 7 and Figure 7 present the performance indices for the experimental results. The throughput peaked when the threshold r was set to 5 minutes and showed a decreasing trend as r increased. In particular, when r approached ∞, meaning that RTD fully adhere to the RTS schedule, throughput dropped by over 50%, indicating a significant performance degradation. Similarly, the on-time delivery rate peaked when r was set to 5 minutes. Equipment utilization remained relatively stable across the entire range of thresholds, but showed a noticeable decline when r was set to ∞. These results indicate that applying RTD decisions selectively may improve FAB production efficiency.

6 CONCLUSION

This study proposes a methodology for constructing a coupled simulator that integrates RTS and RTD in large-scale semiconductor FABs. The proposed approach enables a simulation environment that more

Table 7: Experimantal result.

r	Schedule Execution Rate	Throughput	On Time Delivery Rate	Utilization
0 min	0%	3566	93.49%	75.58%
5 min	16.85%	3691	95.82%	75.52%
10 min	30.98%	3607	94.76%	75.49%
15 min	39.68%	3570	92.68%	75.53%
20 min	46.96%	3567	91.82%	75.39%
25 min	52.72%	3588	94.37%	75.55%
30 min	56.30%	3569	93.31%	75.42%
∞	100%	1161	0%	41.27%

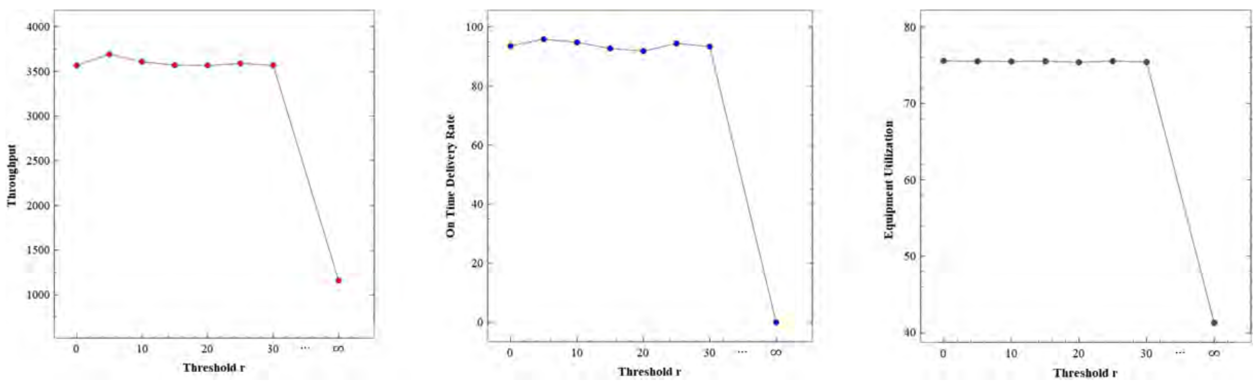

Figure 7: Graph of experimental results.

accurately reflects real-world operations. In particular, it addresses the absence of an integrated production and logistics simulation system that can capture the complexity of modern FABs, which are defined by large scale and dynamic behavior, even under limited computational resources. To achieve this, a communication interface was developed to enable synchronization between the RTS and RTD simulators, resulting in a unified simulation framework. Furthermore, a dynamic operational rule was proposed to coordinate RTS and RTD activities in a way that maximizes production efficiency.

The experiments were conducted using the SMT2020 and SMAT2022 dataset, and performance was evaluated based on key indicators including throughput, on-time delivery rate, and equipment utilization. The experimental results indicate that selectively applying RTD decisions can enhance FAB production efficiency. Moreover, the proposed method offers a practical solution for improving FAB operations without requiring substantial modifications to existing RTS–RTD systems.

However, since the threshold values were explored empirically and not optimized, future research will focus on developing adaptive threshold adjustment techniques. In addition, the proposed framework will be validated in real FAB environments beyond the SMAT testbed to further demonstrate its practical effectiveness.

ACKNOWLEDGMENTS

This work is supported by the Ministry of Trade, Industry and Energy (MOTIE), Republic of Korea (RS-2022-00155650).

REFERENCES

CARLO 2025. PINOKIO SIMULATOR, https://carlo.co.kr/simulator.html, accessed 13th March.

Concepcion, A. I., and B. P. Zeigler. 1988. “DEVS Formalism: A Framework for Hierarchical Model Development”. *IEEE Transactions on Software Engineering* 14(2):228–241.

Currier, C. 2022. “Protocol buffers”. In *Mobile Forensics–The File Format Handbook: Common File Formats and File Systems Used in Mobile Devices*, 223–260. Springer.

El Khayat, G., A. Langevin, and D. Riopel. 2006. “Integrated Production and Material Handling Scheduling using Mathematical Programming and Constraint Programming”. *European journal of operational research* 175(3):1818–1832.

Gartner. 2025. Gartner Forecasts Worldwide Semiconductor Revenue to Grow 14% in 2025, https://www.gartner.com/en/newsroom/press-releases/2024-10-28-gartner-forecasts-worldwide-semiconductor-revenue-to-grow-14-percent-in-2025, accessed 10th March.

Ghasemi, A., F. Farajzadeh, C. Heavey, J. Fowler, and C. T. Papadopoulos. 2024. “Simulation Optimization Applied to Production Scheduling in the Era of Industry 4.0: A Review and Future Roadmap”. *Journal of Industrial Information Integration*:100599.

Gupta, A. K., and A. I. Sivakumar. 2002. “Simulation Based Multiobjective Schedule Optimization in Semiconductor Manufacturing”. In *2002 Winter Simulation Conference (WSC)*, 1862–1870 https://doi.org/10.1109/WSC.2002.1166480.

Hintjens, P. 2013. *ZeroMQ: Messaging for Many Applications*. " O’Reilly Media, Inc.".

Klemmt, A., and L. Mönch. 2012. “Scheduling Jobs with Time Constraints between Consecutive Process Steps in Semiconductor Manufacturing”. In *2012 Winter Simulation Conference (WSC)*, 1–10 https://doi.org/10.1109/WSC.2012.6465235.

Kopp, D., M. Hassoun, A. Kalir, and L. Mönch. 2020a. “Integrating Critical Queue Time Constraints into SMT2020 Simulation Models”. In *2020 Winter Simulation Conference (WSC)*, 1813–1824 https://doi.org/10.1109/WSC48552.2020.9383889.

Kopp, D., M. Hassoun, A. Kalir, and L. Mönch. 2020b. “SMT2020—A Semiconductor Manufacturing Testbed”. *IEEE Transactions on Semiconductor Manufacturing* 33(4):522–531.

Lee, K., S. Song, D. Chang, and S. Park. 2022. “A New AMHS Testbed for Semiconductor Manufacturing”. In *2022 Winter Simulation Conference (WSC)*, 3318–3325 https://doi.org/10.1109/WSC57314.2022.10015476.

Liang, T., L. Zhou, and Z. Jiang. 2022. “Integrated Scheduling of Production and Material Delivery in a Make-to-order Flow Shop”. In *2022 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 1422–1426. IEEE.

Postel, Jon 1981. “Transmission control protocol”. Technical Report RFC 793 https://www.rfc-editor.org/rfc/rfc793, Information Sciences Institute, University of Southern California.

Sakr, A. H., A. Aboelhassan, S. Yacout, and S. Bassetto. 2023. “Simulation and Deep Reinforcement Learning for Adaptive Dispatching in Semiconductor Manufacturing Systems”. *Journal of Intelligent Manufacturing* 34(3):1311–1324.

Sun, C., H. Ehm, and T. Rose. 2016. “Evaluation of Small Volume Production Solutions in Semiconductor Manufacturing: Analysis from a Complexity Perspective”. In *2016 Winter Simulation Conference (WSC)*, 2535–2546 https://doi.org/10.1109/WSC.2016.7822292.

Sun, C., and T. Rose. 2015. “Supply Chain Complexity in the Semiconductor Industry: Assessment from System View and The Impact of Changes”. *IFAC-PapersOnLine* 48(3):1210–1215.

VMS Solutions 2025. MOZART, https://www.vms-solutions.com/en/product/mozart.php, accessed 13th March.

AUTHOR BIOGRAPHIES

SEONGHO CHO received a bachelor’s degree (2023) in industrial engineering and a master’s degree (2025) in industrial engineering from Ajou University, Republic of Korea. He is now a PH.D. student in industrial engineering, Ajou University. He is interested in simulation-based digital manufacturing systems, production scheduling, and automated material handling system. His email address is phillip4321@ajou.ac.kr.

DONGUK KIM is a Research Specialist at LG Production and Research Institute, Republic of Korea. He received his B.S. (2018), M.S. (2021), and Ph.D. (2025) degrees in Industrial Engineering from Ajou University, Republic of Korea. His research interests include simulation-based scheduling, planning, and digital manufacturing. His email address is ugiee.kim@lge.com.

SANGCHUL PARK was granted his B.S.(1994), M.S.(1996) and Ph.D.(2000) degrees in industrial engineering from Korea Advanced Institute of Science and Technology(KAIST). He is a professor in the Department of industrial engineering, Ajou University, Republic of Korea, since 2004. He is interested in modeling and simulation, combat simulation for defense, and digital manufacturing system. His email address is scpark@ajou.ac.kr.