

## Department of Industrial and Systems Engineering and Management(ISEM) **IE3100R System Design Project | AY2024/25**

**Group 19: Skyworks Global Pte Ltd** 

Kian Zhang Yu, Lee Minseo, Ren Yilun, Wong Jun Xiang

Department Supervisor: A/P Cheung Wang Chi / Industry Supervisor: Rachel Ng



# Wafer Start Modelling Engine to Optimize Output

## 1. Company Background

Skyworks Global Pte Ltd is a wholly owned entity of Skyworks Solutions Inc., a global company headquartered in the United States that produces semiconductors.

## 3. Objective & Deliverables

#### **Final Deliverables:**

- Simulation Model of Production Process
- Compares model-predicted wafer production targets against actual output, while monitoring real-time Work In Progress(WIP) levels to identify optimization gaps
- Dashboard with shipment projections of WIP, the past 6 weeks' demand, estimated equipment use and fab cycle time

## 2. Problem Statement

#### Ship to other Ship from Japan **Wafer Process** Testing countries

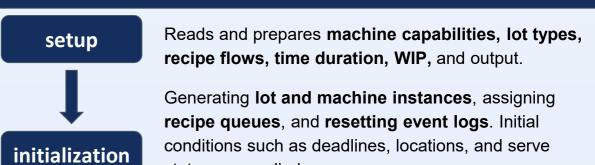
Problem Background: Maximize the output of a wafer fabrication facility while maintaining quality, minimizing costs, and adhering to operational constraints.

## **Constraints**

- **Equipment utilization:** Maximize machine uptime and reduce downtime.
- **Cycle time:** Minimize production time per wafer.
- Cost efficiency: Optimize material usage, energy, and labor costs.
- Supply chain: Ensure timely availability of raw materials and components.

Develop a simulation engine to optimize lot-to-machine assignments in a wafer fabrication environment.

## 4. Simulation Framework



status are applied.

Executes a **discrete-event simulation**, assigning lots to machines using a combination of heuristics and ILP. Events such as machine availability, recipe simulation completion, and ACR updates drive the timeline forward.

> Produces a complete schedule and event logs, performance metrics for lots and machines.

Show visualized suggested schedule for users.

## 5. Parameters Setup and Data Processing

## Lot & Machine Database with Availability and Mapping (based on Oracle data)

- Lot Initialization: Each lot is created with ID, part number, size, deadline, deadline, state, priority, location, size, current recipe, and current recipe time. Recipes are derived by recursively expanding structured process definitions from SQL databases.
- Machine Initialization: Machines are set up with id, location, recipes, serve end time, part number, current recipe, lot id
- Only available machines are included.

## Time Parameters Estimation throughout the Manufacturing Process (based on Oracle data)

- **Recipe Time Estimation**: Processing times are computed using production throughput data (UPH), converted to seconds per unit. Fallback logic ensures missing data is handled.
- Travel Time Estimation: Estimated using historical lot movements between equipment locations, producing average, min, and max travel durations for each location pair.
- Switching Time Estimation: Based on part grouping logic (length and width), switching times are sampled from categorized time ranges using a conversion matrix.

## Input Integration (data from ACTL)

- ACTL Data: Real-time WIP lot data is pulled from ACTL table (filtered by technology, material type, and due dates).
- User Input: Additional lots are read from user-input CSV read wafer quantity, and split based on size limits.

All inputs are unified into: lot\_dict(All lot instances) / recp\_dict(Mapping from recipe to lots) / machine\_instances\_dict(All machine instances)

output

 Loads part–recipe mapping from a preprocessed JSON file, then mapped to a recipe sequence and stored as PartNum objects.

## Lots

Recipe

- Loads real-time WIP lots from production data.
- Creates a Lot object with part number, size, deadline, deadline, state,

6. Initialization

- priority, location, size, current recipe, and current recipe time
- Categorizes it under its current recipe.
- Stores lots in lot\_dict and recipes in recp\_dict.

## **Machines**

 Available machines initialized with capabilities, location, and serve status.

## **User Input Lots**

Custom lots split and initialized, then merged with ACTL lots.

# 7. Simulation Model for Wafer Manufacturing Process

**Start Simulation** 

## **Event-Driven Execution**

The simulation follows a priority-based event queue.

## **Key events**

MACHINE_AVAILABL E	Machine ready for processing.
LOT_ASSIGNMENT	Assigning lots to machines.
RECP_FINISHED	Recipe completion updates.
UPDATE_ACR	Updating Aging Critical Ratio (ACR) values.

## **Key Functions**

1. Processes Events: Handles events such as machine availability, lot assignments, and recipe completions using an event queue.

handle_event	Routes the event to the appropriate handler based on its type.
handle_machine_available	Marks a machine as available and adds it to the idle pool.
handle_lot_assignment	Attempts lot-to-machine assignments using ILP.
handle_recp_finished	Updates lot status and machine status after recipe completion.

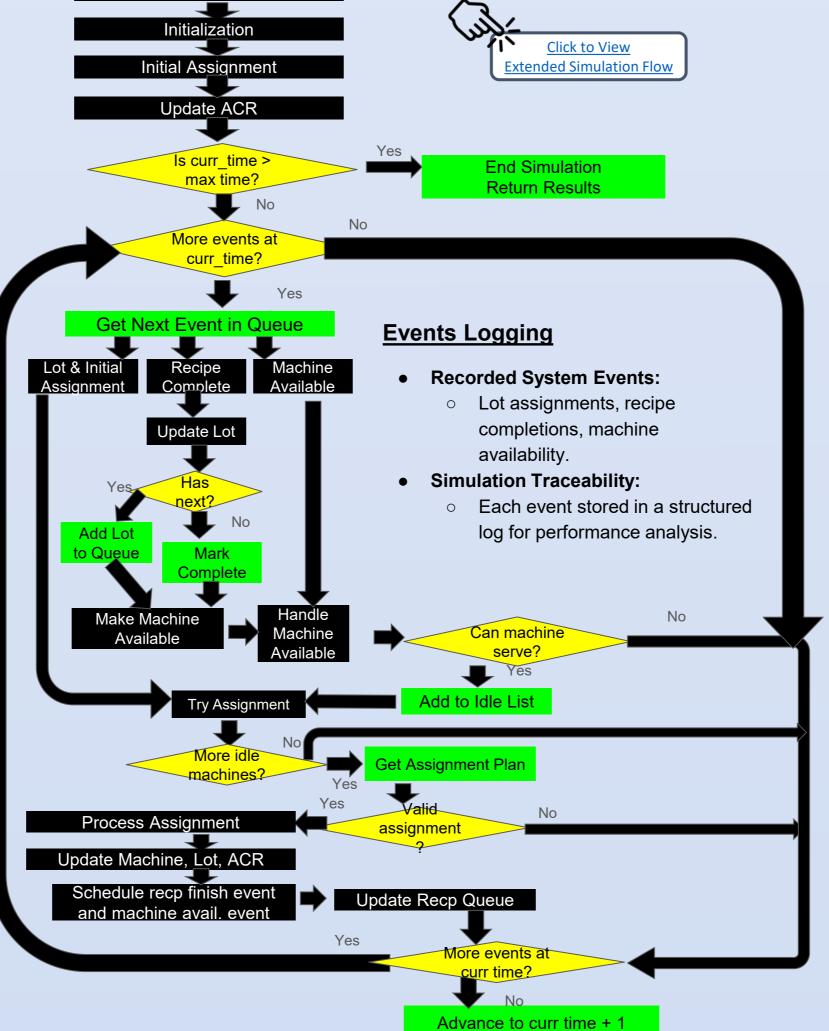
2. Updates System State: Updates lot and machine status, location, and ACR values, add the timestamp when updated

	update_machine	Updates the machine with new lot info and schedules end time.
	update_lot	Updates lot metadata upon assignment and schedules recipe completion.
	update_ACR	Updates the ACR values for all lots based on simulation time.
	add_to_timestamp	Adds future events to the priority queue.

3. Assigns Lots via ILP: Uses Integer Linear Programming to optimally assign lots to machines based on cost (processing, traveling, switching).

try_assignments	Checks if assignments can be made; triggers assignment logic.	
get_assignment_plan	Builds and solves the ILP for optimal lot-to-machine allocation.	
assign_lots_to_machines	Writes assignment decisions to the lot and machine instances.	

4. Logs Events: Records all actions in a global event log for traceability and analysis.



## **Key Transitions from Key Events**

**Trigger** 

MACHINE_AVAILABLE event	<ul> <li>→ handle_machine_available()</li> <li>→ machine added to idle_machines</li> </ul>
RECP_FINISHED event	<ul> <li>→ handle_recp_finished()</li> <li>→ lot marked as finished, machine rescheduled</li> </ul>
LOT_ASSIGNMENT event	<ul> <li>→ handle_lot_assignment()</li> <li>→ attempts assignment via ILP</li> </ul>
UPDATE_ACR event	→ update_ACR() → schedules next ACR update
Successful ILP assignment	→ assign_lots_to_machines() → machine and lot status updated
Assignment completion	→ add_to_timestamp() → schedules next events

**Resulting Action / Transition** 

#### **Assignment Logics: Integer Linear Programming Variables** $l_i$ : Lot i, where i = 1, ..., n, each with a recipe $r_i$ and urgency score ACR<sub>i</sub>.

 $m_i$ : Machine j, where j = 1, ..., m, capable of recipes in set  $C_i$ .  $x_{ij} \in \{0, 1\}$ : 1 if lot  $l_i$  is assigned to machine  $m_i$ , 0 otherwise.  $P_i \in \{0, 1\}$ : 1 if machine  $m_i$  is idle (not assigned), 0 otherwise. M: Penalty per idle machine.

## **Assignment Cost Function**

 $S(l_i, m_i) = (Recipe_{ii} + Max(Switch_{ii} + Travel_{ii})) \times (ACR_i)^2$ , if  $r_i \in C_i$ Recipe<sub>ii</sub>: Processing time of recipe  $r_i$  on machine  $m_i$ 

- Switch<sub>ij</sub>: Time to switch machine  $m_i$  to recipe  $r_i$ .
- Travel<sub>ii</sub>: Travel time from lot  $l_i$ 's location to  $m_i$ .
- ACR<sub>i</sub>: Adjusted Critical Ratio, prioritizing urgent lots.
  - Switch & travel can be done concurrently, only the larger value matters

 $\min \sum_{i=1}^{m} \sum_{j=1}^{m} S\!\left(l_{i}, m_{j}\right) \!\!\times\! x_{ij} \!+\! M \!\!\times\! \sum\limits_{j=1}^{m} P_{j}$ **Constraints: Each lot** is assigned to at most **one machine**:

**Each machine** processes at most **one lot**: Infeasible assignments are prohibited:

 $x_{ij} = 0 if r_i \neq C_{jfor} all i, j$ 

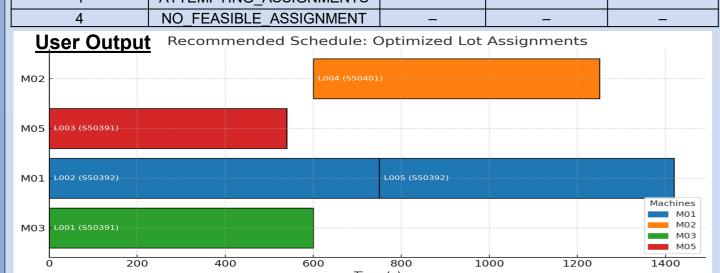
Each machine is either assigned a lot or marked as idle:

 $\sum_{i=1}^{n} x_{ij} + P_j = 1$  for all j

## 8. Result and Validation

**Extended User Output** 

#### **General Event Log (Future Event List) End Time Timestamp Event Type** Lot ID Machine ID ASSIGNMENTS MADE 0 MACHINE AVAILABLE M01 RECP FINISHED L001 4 ATTEMPTING\_ASSIGNMENTS 4



# 9. Improvements

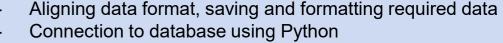
- Performance Optimization: Use heuristic pre-filtering or hybrid solvers to reduce ILP computation time, especially under high-load scenarios.
- 2. Anticipatory Queueing for Better Utilization: Include soon-to-be-ready machines in the ILP pool instead of only IDLE machines to reduce the IDLE time during max(Switch, *Travel*) to further improve utilization rate.
- 3. Enhanced Metrics Collection: Extend event logs to calculate metrics such as: Average turnaround time per lot, Machine idle rate, Number of overdue lots.
- 4. Robust Input Validation: Include more extensive errorchecking for user-input data and file formats.

## 10. Skills & Takeaways



- Python:
  - Developed a **discrete-event simulation engine** for lotto-machine assignment in manufacturing
  - hybrid assignment strategy combining heuristics, Hungarian algorithm, and global Integer Linear **Programming (ILP)**
  - Enabled detailed **event logging** for system traceability and post-simulation analysis. Classes of Lots, Machines and PartNum (Part Numbers)





Input Template





