Simulating the Semiconductor Manufacturing Process through AutoSched Advanced Processing

An Exploratory Study to Enhance NXP's ASAP Model Forecast Accuracy

By
Nick van Lambalgen

UT Supervisors:
Dr. Berend Roorda
Dr. Reinoud Joosten

NXP Supervisors:
Ir. Eduardo Bueno López

Department of Industrial Engineering and Management
Track: Financial Engineering and Management
University of Twente
Conducted at NXP Semiconductors, Nijmegen

21 November 2024

Abstract

In this thesis, we explore modifications to enhance the accuracy of NXP's AutoSched Advanced Processing (ASAP) model, aiming to address stakeholders' challenges in using ASAP effectively to improve operation efficiency of ICN8, NXP's semiconductor plant in Nijmegen. Our analysis of operations at ICN8 revealed that inefficiencies in the operation rate represent the largest source of efficiency loss, while tool availability rate ranks as the third largest source of efficiency loss. These inefficiencies are areas where the ASAP model could drive improvement.

As of 2023, NXP's 72-hour ASAP model forecasts daily moves at ICN8 with an average absolute move deviation of 15.8 percent per cap group (groups of machines with similar capabilities) for the 35 largest cap groups. This large deviation limits the effectiveness of ASAP for improving ICN8's operation efficiency. Especially ASAP's potential for pinpointing future bottlenecks cannot be utilized as the stakeholders need an ASAP model that is able to forecast absolute moves per cap groups with a deviation of no more than 5 percent to trust the model.

To enhance the forecasting accuracy of NXP's ASAP model, we implemented the process times per tool and tool idle times into the ASAP model. Validation of these modifications demonstrated an average reduction of 39.2 percent in forecasted move deviation across cap groups, reducing the average daily forecasted absolute deviation for the 35 largest cap groups from 19.1 to 11.7 percent in 2024.

While this improvement does not meet the 5% deviation target necessary for accurately pinpointing bottlenecks and optimizing the overall efficiency of ICN8's production process, it represents a significant step toward this goal. Further refinements could yield even better results. Additionally, these improvements could assist NXP in making better decisions regarding operator staffing, tool maintenance planning, and reducing the weekly maintenance time required for the ASAP model itself.

Preface

I hereby present my master thesis “Simulating the Semiconductor Manufacturing Process through AutoSched Advanced Processing: An exploratory study to enhance NXP’s ASAP Forecast Accuracy”, which marks the completion of my master’s degree in Industrial Engineering and Management, with a specialisation in Financial Engineering and Management, at the University of Twente.

I would like to express my gratitude to Dr. Berend Roorda as my supervisor at the University of Twente. His continuous guidance and valuable feedback greatly supported the development of this thesis, providing me with the direction and clarity needed throughout the research process. I am equally grateful to my supervisor at NXP, Ir. Eduardo Bueno López, who encouraged me to critically assess my decisions and provided valuable feedback that greatly enhanced my work.

Additionally, I wish to express my appreciation to the staff at NXP for their support throughout the project. A special thanks to Herwin van Hoof for helping me master the ASAP model quickly and providing the necessary data, Yuri van Heertum for tracking my progress and giving additional feedback on my choices, and Deen Colenbrander for assisting with the mathematical foundation of my thesis.

I also want to thank Reinoud Joosten for helping me as my second supervisor. He provided me with valuable feedback to further improve my thesis.

Finally, I am deeply grateful to my family and friends for supporting me throughout my studies. Their encouragement has motivated me to always strive for my best.

I hope you enjoy reading this thesis!

Nick van Lambalgen

21 November 2024

Contents

List of figures

List of tables

List of abbreviations

Abbreviation	Definition
AE	Availability Efficiency
AMAT	Applied Material
APF	Advanced Productivity Family
ASAP	AutoSched Advanced Processes
CRITO	Critical Tools
IC	Integrated Circuit
ICN8	Name of semiconductor plant NXP Nijmegen
fab	Synonym for a semiconductor plant
KPI	Key Performance Indicator
MAPE	Mean Absolute Percentage Error
MTBPM	Mean Time Between Preventive Maintenance
MTTR	Mean Time to Repair
OEE	Overall Equipment Effectiveness
OFE	Overall Factory Effectiveness
PTIME	Process time
STN	Station (or machine)
VBA	Visual Basic for Applications
WIP	Work in Progress
ROI	Return on Investment

1. Introduction

1.1. Company description

The thesis project is conducted at the Industrial Engineering Department of NXP Semiconductors in Nijmegen. NXP Semiconductors is a global leader in secure connectivity solutions for embedded applications (*NXP Overview*, 2023). After separating from Philips in 2006, the company has expanded into a global organisation. Currently, NXP is active in more than thirty countries from which two sites are located in the Netherlands: its head office located in Eindhoven and a manufacturing plant in Nijmegen, where integrated circuits (ICs) are produced.

Currently, NXP Nijmegen operates approximately seven hundred machines, each contributing to the production of various types of 200-millimeter IC wafers. These machines are distributed across two floors of the manufacturing plant ICN8, where they are operated in coordination to produce the IC wafers. Operators and robots transport lots (vacuum-sealed boxes that can hold up to 25 IC wafers) from machine to machine until the IC wafers are finished. Producing a single wafer involves a complex process and takes several days to complete.

NXP Nijmegen strives to maximise profit by meeting quantity and quality targets at the lowest possible cost. The company continuously seeks to improve its production process by optimising resource distribution to achieve these goals effectively. Key factors that NXP can assign to maximise efficiency include the types and number of machines that must operate each day and the number of operators needed for those machines. However, assigning the right resources in advance is challenging due to the lengthy and complex production process, the variety of wafer types, and the movement of lots through approximately seven hundred machines, where numerous events can occur that disrupt the process. Nevertheless, NXP needs to distribute the resources, which is why NXP acquired the simulation program AutoSched Advanced Processing (ASAP) in 2016.

1.2. The core problem

NXP values the ASAP program for its ability to generate simulations that forecast the future behaviour of the production process at ICN8. By using accurate simulations, NXP can identify bottlenecks and solve them through resource allocation. This approach streamlines the production process, which increases overall efficiency and reduces costs.

Despite ASAP's potential, the current model falls short of accurately simulating the production process of ICN8, resulting in significant deviation from the actual output. These deviations lead stakeholders to make incorrect decisions regarding resource allocation. To improve decision-making, stakeholders require a 72-hour model that simulates cap groups moves with an average absolute deviation of no more than 5% from the actual results. This level of accuracy is essential to identify and resolve bottlenecks before they cause problems in the production process.

Thus, the core problem is: The ASAP model of NXP cannot forecast bottlenecks because it has an average absolute daily deviation of more than 5% from the actual moves.

1.3. Research problem and objective

Since 2016, several engineers have attempted to improve NXP's ASAP model. Despite efforts, the model remains inaccurate, resulting in low stakeholder confidence. For shareholders to use the model effectively and pinpoint future bottlenecks, it must forecast cap group moves with an average absolute daily deviation of no more than 5% in 72-hour simulations. Prior to the start of this thesis, the accuracy of the model was unclear, though it was known to be insufficient. To achieve the 5% target, NXP needs to identify the details that the existing model currently lacks and determine the necessary adjustments for more accurate simulations. This research aims to support NXP in this effort by identifying ways to validate and improve NXP's ASAP model.

1.4. Research questions

To improve NXP's ASAP model, potential changes that can benefit the model should be found and validated. The primary research question is:

- How should the ASAP model be modified to improve 72-hour simulations based on daily moves per cap group?

The main focus of the thesis is to understand and enhance the ASAP model for NXP's entire production process and validate the impact of these changes on the its accuracy. To achieve this goal, the following sub-questions are stated:

1) How can the ASAP model impact the production process of ICN8?
   a) What is ASAP?
   b) How does the production process of ICN8 work?
   c) How does NXP utilise ASAP?
2) How does the ASAP model of NXP function?
   a) What data does the ASAP model of NXP utilise to simulate?
   b) How accurate is the current 72-hour ASAP model?
3) What adjustments could improve the ASAP model?
4) How do the results of the modified model compare to those of the older model?
5) What benefits arise from the modified ASAP model?
   a) What impact does the modified model have on cost reduction?
   b) What impact does the modified model have on process optimisation?

2. Literature review

Given the complexity of simulating the semiconductor manufacturing process, this section reviews key concepts necessary for understanding the context and scope of this thesis. The main areas covered in this section include the manufacturing process of semiconductors, modelling a production process, and the program AutoSched Advanced Processing. These concepts provide a foundational understanding of the practical and theoretical aspects necessary for modifying the ASAP model.

2.1. Manufacturing process of a semiconductor

Producing semiconductors is a complex process involving multiple manufacturing steps. The process begins with the creation of wafers, which are typically composed of nearly pure silicon(*From sand to silicon "Making of a chip" illustrations*, 2011). Silicon is extracted from sand, which contains about 25% silicon and is the second most abundant chemical element in the Earth's crust. Since sand consists of both oxygen and silicon, pure silicon can be obtained by melting the sand until the molten substance is nearly pure silicon. This molten substance is then formed into monocrystalline silicon ingots. These ingots are eventually cut into very thin plates and polished, resulting in the wafers.

Once a wafer is prepared, an integrated circuit (IC) is fabricated on the wafer. NXP Nijmegen is one of the companies that is capable of producing IC wafers. IC wafers consist of multiple layers that are applied to the wafer one at a time. For each layer added on the wafer, the wafer passes through multiple tools, involving hundreds of production steps per wafer(*From sand to silicon "Making of a chip" illustrations*, 2011).

The process begins by applying an oxidation layer and a photoresist layer to the wafer (*From sand to silicon "Making of a chip" illustrations*, 2011). A mask with the designed circuits is then used to precisely remove the photoresist layer through photolithography. In addition to removing the photoresist layer of the circuit, the oxidation layer of the circuit design is also removed through dry or wet etching. Although both techniques achieve similar results, dry etching is more precise but is also more expensive(Geng & Zhou, 2004). Once the etching step is completed, the remaining photoresist layer is removed, leaving behind a circuit pattern.

After the etching process, the wafer undergoes the deposition and ion implantation processes to enhance the conductivity of the wafer (Geng & Zhou, 2004). The wafer then moves to the metal department, where metal wiring is added to enable electrical connectivity through the circuit on the wafer. After the wiring is adjusted, ions are placed on the layer and the layer is re-polished to ensure it is smooth enough. This process is repeated until all layers are added.

Finally, the wafer is tested using electrical die sorting to verify if the wafer is fabricated correctly. This step is performed at NXP Nijmegen before the IC wafer is shipped to other facilities in Asia (Geng & Zhou, 2004).

Outside of NXP Nijmegen, semiconductors are produced from IC wafers. This stage involves retesting the wafers, cutting them into pieces, and packaging them (Seon, 2017). The packaged semiconductors are then subjected to final testing before they are ready for use.

2.2. Modelling an IC wafer production process

Modelling the IC wafer production process in a semiconductor plant can offer significant benefits to the company. Accurate modelling enables stakeholders to monitor the plant's status and optimise resource utilisation, increasing production efficiency and reducing operational costs (Ramirez-Hernandez, Li, et al., 2005). Given the substantial revenue generated by a semiconductor plant, even minor improvements in efficiency can lead to considerable profits (Hunter, Delp, et al., 2002). Additionally, since wafers are expensive products that can easily turn into scrap due to disruptions and delays, minimising bottlenecks through effective modelling reduces waste and increases profitability further (Bettayeb, Bassetto, et al., 2014).

Accurately modelling the production process of IC wafers is a complex challenge(Hunter, Delp, et al., 2002). Manufacturing a wafer involves hundreds of production steps, each performed by numerous diverse machines, making it difficult to get an overview of the entire production process. At ICN8, NXP's Dutch semiconductor plant, this challenge is even more complicated as only parts of the ICN8 fab are designed to optimise transport, due to the complex extensions of the factory over the years. This layout of machines causes lots to travel long distances across the plant, complicating the task of gaining an overview of the entire production process.

ICN8 is especially interested in modelling both the plant as a whole and the individual cap groups. The advantage of modelling the entire plant is that such a model can provide relatively accurate results of the plant's inputs and outputs. These data can be used to make informed decisions about ramping up or down production to meet demand (Klein & Kalir, 2006). On the other hand, simulating individual cap groups allows for identification of potential bottlenecks before they become problematic (Kusters, 2021). This information is beneficial because it gives the stakeholders the ability to take countermeasures to prevent disruptions.

Both simulation models are difficult to develop, yet simulating at the cap group level requires a higher level of precision to be effective. Unlike plant-level simulations, where deviations can often offset one another, cap group level simulation demands highly accurate data, as even minor inaccuracies can obscure bottlenecks in the model. Achieving this level of precision is especially difficult because certain variables are almost unpredictable. As a result, assumptions play a significant role in determining the accuracy of the simulation results, which are crucial for identifying bottlenecks (Kiba, Lamiable, et al., 2009).

2.3. AutoSched Advanced Processing

As described in the previous subsection, large amounts of data are required to accurately simulate the production process of IC wafers. The vast amount of data is necessary due to the complexity of the semiconductor plant, where numerous variables influence the overall behaviour of the production process. This complexity, in combination with unpredictable data, poses significant challenges for an effective simulation (Taylor, 2005).

While many simulation programs exist for general manufacturing processes, most cannot effectively simulate semiconductor plant operations due to the large amount of data involved (Taylor, 2005). This, causes the simulation models to produce outcomes that does not reflect reality and cannot be used.

Applied Materials has developed AutoSched Advanced Processing (ASAP) specifically to simulate the complex production process of a semiconductor plant, providing more accurate and detailed results compared to general programs(Ping, Liow, et al., 2007). Multiple semiconductor factories worldwide rely on ASAP because it can handle the large volumes of data that are required to effectively simulate a semiconductor plant. Since 2016, NXP Nijmegen has utilised ASAP. While NXP's ASAP model can sufficiently simulate the production process of ICN8 at the overall plant level, it does not provide the individual cap group results needed for the stakeholder to proactively manage future bottlenecks (Kusters, 2021)

The data used for AutoSched AP can be extracted from Advanced Productivity Family (APF) (Tien, Teck, et al., 2006), a program that is also used by NXP for continuous data collection and updates. Each ASAP simulation run uses the latest information, enhancing the accuracy of the results.

3. Methodology

This section outlines the research design, data collection methods, and analysis techniques used to achieve the study's objectives. A methodology ensures that the research is systematically structured and reliable. A robust methodology not only enhances clarity and reliability of the paper but also ensures that the study is effectively managed. Given the large quantities of data involved, structuring these dataset are crucial as it minimises errors, thereby enhancing the credibility of the study.

3.1. Quantitative research approach

The research problem focuses on improving the existing ASAP model to identify future bottlenecks in the production process at ICN8. To improve the model effectively, research must be conducted on multiple areas, including ICN8's production process, simulating a semiconductor plant, and the workings of ASAP.

A quantitative research approach is beneficial for this type of research due to the vast amount of data used, the need for replicable results, and the necessity for statistical comparison (Creswell, 2014). The most suitable research method for this thesis is the experimental research method, as it evaluates how changes in variables within the existing model affect performance and can be used to determine how the model can be improved.

3.2. The methods and data collection techniques

To develop and validate the new model, several data collection methods must be applied. Important data collection techniques include literature research, observational studies, and experiments(Creswell, 2014). These methods are needed to gather information about NXP and ASAP.

3.2.1. Literature research

By conducting a literature study, knowledge can be obtained from other researchers who have conducted research in similar fields. This includes information about the production process of semiconductors, the workings of ASAP, and findings of other model engineers that work with ASAP. The literature research also aids in saving time by utilising prior findings, avoiding redundant efforts.

3.2.2. Observational studies

Observational studies will be conducted to analyse the performance of the existing ASAP model by comparing previous simulation outputs with real-world data from ICN8. Comparing the model and real data is necessary to identify the underlying causes of its poor performance. This step is critical for understanding why the existing model's forecast deviates significantly from the actual moves.

3.2.3. Experiments

To analyse the new changes made to the older model, samples must be generated by running the new model under the same conditions as the existing model. The samples are necessary to compare the results of both models to reality. By comparing how the results of both models align with reality, the impact of changes to the new model can be observed. This experimental approach will provide the necessary data to determine the effectiveness of the model improvements.

4. ICN8 efficiency gaps and ASAP's role at NXP

The main objective of this thesis is to improve the forecasting performance of NXP's ASAP model. To emphasize the importance of this project, the current inefficiencies in ICN8's its operations are highlighted, along with an explanations of how ASAP is used to minimize these inefficiencies. These insights justify the need for carrying out this thesis and illustrate the potential impact on NXP's production efficiency.

4.1. The inefficiencies of ICN8

Like many manufacturing facilities, ICN8 faces inefficiencies that prevent it from operating at its maximum capacity. These inefficiencies arise from factors such as tool breakdowns, human errors, and machine performance. By correctly allocating resources, which can be done with the help of ASAP, many of these inefficiencies could be mitigated. To determine the potential areas of improvement that can be impacted by ASAP, we examine the specific inefficiencies at ICN8.

One effective method to measure process inefficiency for each machine is through the Overall Equipment Effectiveness (OEE) (Ferko & Znidarsic, 2007), which is calculated in equation (1). The OEE formula normally consists of three metrics:

1. Operation rate (Or): The percentage of time the machine is actually processing.
2. Performance rate (Pr): The percentage of the actual processing speed compared to the optimal processing speed.
3. Quality rate (Qr): the percentage of successful executed process steps.

$$OEE(\Delta T) = Or(\Delta T) * Pr(\Delta T) * Qr(\Delta T) \tag{1}$$

NXP currently collects these data for each cap group (c) and tool (t) on both a daily and yearly basis. Because machine inefficiencies fluctuate daily, the annual inefficiencies per tool (ΔT) is used. By calculating the OEE for each tool, we can gather the necessary information to determine the Overall Factory Efficiency (OFE) of ICN8 (Ferko & Znidarsic, 2007). This provides a clear and reliable overview of the facilities inefficiencies.

The operation rate and the availability rate (Ar) are typically combined as part of the operation rate. However, NXP separates these metrics and uses the following formula to calculate the Overall Equipment Effectiveness per tool (OEE_t) (Heukelom, 2020), as is stated in equation (2). Here, NXP refers to the availability rate as the percentage of time the machine is available for processing and the operation rate as the actual processing time of the machines against the available processing time.

$$OEE_t(\Delta T) = Ar_t(\Delta T) * Or_t(\Delta T) * \mathrm{Pr_t}(\Delta T) * Qr_t(\Delta T) \tag{2}$$

The following formulas are used by NXP to calculate all necessary metrics in equation (2) (Heukelom, 2020).

Availability rate

The availability rate, displayed in equation (3), is the average percentage of available time across all shifts (s) for a tool. Here, the equipment time is the total time of a shift, whereas the available equipment rate

(ae) refers to the percentage of time the machine is available for operation, which is calculated in equation (4).

$$Ar_t = \frac{\sum_s (ae_{t,s} * eqptime_s)}{\sum_s (eqptime_s)}$$

(3)

The total equipment uptime can be divided into unscheduled downtime ($udtime$) and scheduled downtime ($sdtime$). Unscheduled downtime refers to the tool downtime caused by unplanned events, such as machine breakdowns and maintenance delays. Scheduled downtime, on the other hand, refers to the time a tool is unavailable due to planned maintenance for inspection and repairs.

$$ae_{t,s} = \frac{eqptime_{t,s} - udtime_{t,s} - sdtime_{t,s}}{eqptime_{t,s}} = \frac{eqpuptime_{t,s}}{eqptime_{t,s}}$$

(4)

Operation rate

The operation rate is the average percentage of uptime during which the equipment is actually processing (5). The rate of the actual processing time is calculated for each tool based on its assigned shift. Equipment uptime refers to the time a machine is able to process, and lost time refers to the time a machine is not processing, while it could process.

$$Or_t = \frac{\sum_s \left(eqpuptime_{t,s} - losttime_{t,s}\right)}{\sum_s (eqpuptime_{t,s})}$$

(5)

Performance rate

The performance of a machine is influenced by the processing speed and the average size of the batch. When batches are incomplete or machines process below the optimal (gold) standard, the machine is performing below its maximum potential. NXP calculates the performance rate for each tool using the performance rate formula (6).

$$\mathrm{Pr_t} = \frac{\sum_s (Batchsize_{t,s})}{\sum_s (GoldenBatchsize_{t,s})} * \frac{\sum_s (Goldenprocesstime_{t,s})}{\sum_s (Processtime_{t,s})}$$

(6)

Quality ratio

In ICN8, processes can be incorrectly executed, leading to rework, where wafers must undergo the same process. These reworks cause delays which can be calculated using the quality ratio in equation (7). Effective equipment time refers to the total time the machine is effectively processing, while rework time refers to the time a machine has to do rework caused by quality delays.

$$Q_t = \frac{\sum_s (eqpefftime_{t,s} - reworktime_{t,s})}{\sum_s (eqpefftime_{t,s})}$$

(7)

In summary, NXP calculates the OEE for each tool and cap group by multiplying all loss factors together. By analysing efficiency metrics at each step, it becomes evident where the inefficiencies in ICN8 lie. For the cap groups, the OEE is determined by summing the fractional inefficiencies of each tool within the cap group and multiplying these summations. Using the same mathematical approach for all machines of ICN8 (N), the Overall Factory Efficiency (OFE) can be calculated. This results in the following equation (8):

$$OFE = \sum_{t=1}^{N}(Ar_t * W1_t) * \sum_{t=1}^{N}(Or_t * W2_t) * \sum_{t=1}^{N}(Pr_t * W3_t\ ) * \sum_{t=1}^{N}(Qr_t * W4_t)$$

(8)

Equation (8) is complex but essential for calculating efficiencies based on the fraction of remaining possible moves per tool. The calculation is crucial for two main reasons:

1) Tools vary in their ability to execute moves. As a result, the number of possible moves per tool differs. For example, if a fab consists of two tools where machine A could potentially execute 1000 moves per day while machine B can only handle 10 moves per day, an inefficiency in machine A would have a larger impact on the total number of executed moves in the fab. To address this, a weight is assigned to each machine based on the maximal number of moves it can perform.
2) The inefficiencies are based on the remaining efficiency of the machines. For example, in a fab where machine A is capable of making 1000 moves per day and machine B of 10 moves per day, machine A would account for approximately 99 percent of the total amount of moves, while machine B would account for just 1 percent. If the availability rate of machine A is 1 percent and machine B is 100 percent, then machine A would only be available to make 10 moves, matching the 10 moves that machine B can execute. As the remaining number of possible moves is equal for both machines in this case, the operator efficiency of both tools would have an equal effect on the overall fab efficiency. This example illustrates the importance of continually updating the fractions for calculating each efficiency level.

The fraction of tools in the fab, denoted as (W), is calculated using the formulas in equations (9), (10), (11), and (12) and must be updated as efficiency changes:

$$W1_t = \frac{moves_t}{total\ moves}$$

(9)

$$W2_t = W1_t * AR_t$$

(10)

$$W3_t = W2_t * Or_t$$

(11)

$$W4_t = W3_t * \mathrm{Pr_t}$$

(12)

By using these formulas, the OFE of ICN8 can be calculated. The results obtained illustrate the financial benefit and importance of ASAP, though specific values remain confidential. Instead, inefficiencies are listed in order from largest to smallest, as shown in Table 1.

Table 1 Inefficiencies ICN8.

Efficiency factor	Efficiency ranking
Quality rate (most efficient)	1
Availability rate	2
Performance rate	3
Operation rate (least efficient)	4

Table 1 shows that the largest contributor to inefficiency at ICN8 is the operation rate, mainly due to operator inefficiency and underutilised machines caused by the product mix (different types of semiconductors produced) (Yon-Chun & L. Hsuan, 2000). The second-largest contributor to overall factory inefficiency is the performance rate, which is impacted by incomplete batches and delays from aging machines. Availability ranks third, which is impacted by equipment breakdowns and maintenance delays. Lastly, there are some inefficiencies in the fab caused by rework.

4.2. NXP's use of ASAP

A simulation program such as ASAP that forecasts future process behaviour is especially useful for improving planning and resource allocation. Enhancing these areas can significantly boost the availability rate and the operation rate, leading to a positive impact on the efficiency of ICN8. The operation and availability rates represent the largest and third-largest inefficiencies of ICN8, as shown in Table 1.

The availability rate is influenced by both unplanned and planned maintenance. Although ASAP cannot directly address inefficiencies caused by machine breakdowns, it can assist in optimising maintenance planning. By scheduling machine maintenance at optimal times, additional bottlenecks can be prevented, resulting in a more efficient production process. Currently, ASAP helps determine the best timing for machine maintenance to minimise disruptions to ICN8's production process.

Moreover, ASAP also positively influences the operation rate. During the daily CRITO (Critical Tools) discussions, stakeholders allocate resources, such as operators and machines, to optimise production and meet targets. The 72-hour model of ASAP is one of the programs that provide insight into ICN8's upcoming performance. The model currently forecasts the daily future number of moves per cap group, the daily completed number of lots, the placement of the lots, and the WIP (Work-in-progress) per cap group, which indicates the number of lots that are waiting to be processed by a cap group. The more accurate the ASAP model, the better-informed stakeholders' decisions become, resulting in better resource distribution and a higher operation rate.

5. The working and performance of NXP's ASAP model

Since 2016, NXP has focused on developing an ASAP model to simulate the production process of ICN8. The program ASAP was chosen, as it is specially designed to model the production processes of a semiconductor plant using real-time data. ASAP is particularly valuable due to its ability to identify potential bottlenecks before they occur, providing NXP with crucial insights to address these issues proactively, which increases overall efficiency.

5.1. Working of ASAP

The goal of ASAP is to simulate the maximum possible number of moves within a production system while meeting targets and respecting constraints. Using the provided input data, ASAP conducts a single simulation over a defined period to determine the daily number of moves made per machine.

ASAP operates through multiple interconnected text files that contain information about the initial status of the plant and anticipated future events, such as breakdowns, maintenance, and production targets. The core file in this setup is the "option.rdf" file, which directs ASAP on which files to use and how to interpret their data.

Essential files that are required for a simple simulation include the order.txt, part.txt, route.txt and stn.txt files. These files, like all files within ASAP, are connected to each other, for example:

- The order.txt file outlines the production targets by specifying the types of products that need to be produced.
- The part.txt file determines the route of processes required to produce each type of product.
- The route.txt specifies the individual processes that are included within each route. Here the recipe, process time and cap group are assigned to each step.
- The stn.txt file contains information about which machines are capable of executing the processes.

The connections between these files are illustrated in Figure 1.

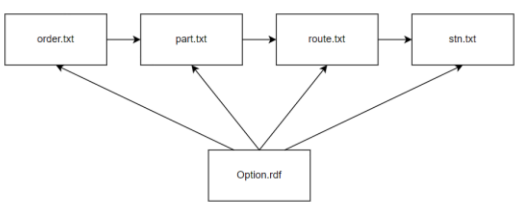

Figure 1 Connection between ASAP files.

Other important files contain information about tool status, number of tools, process capacity, tool failure probability, tool repair time, tool types, chambers per tool, rework levels (child lots), wafer priorities, operator efficiency, number of available operators, storage size, lot size, lot status, plant layout, product mix, process capability, planned maintenance, wafer arrival rate, production step time, wafer type recipe, travel times and holds (Ramirez-Hernandez, Li, et al., 2005) (Miller, 1990). The files in which these functions are assigned can be found in Appendix A.1.

After running the ASAP model of NXP, the output data will include the key performance indicators (KPIs) necessary to make effective decisions. These KPIs include work in progress (WIP), cycle time, machine utilisation and moves per cap group(Ramirez-Hernandez, Li, et al., 2005).

5.2. Generated input data

At NXP's ICN8 plant, real-time data are collected from all machines, lots, and orders. Initially, this raw data is unstructured and lacks explanations. Over time, employees convert this raw data into a structured, readable format suitable for modelling purposes. Eventually, the data can be automatically converted, updated and stored in APF. The existing ASAP model used by NXP relies on the information stored in APF, which provides the organised and up-to-date data necessary to execute simulations. Figure 2 shows the data files, totalling approximately 55 MB, required to run a single ASAP simulation.

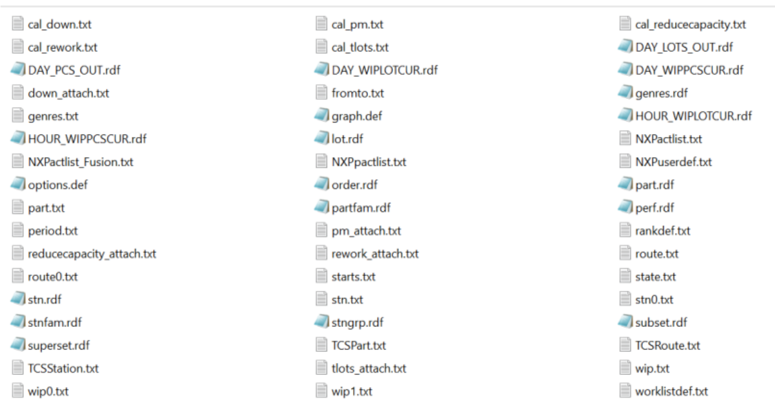

Figure 2 Input data existing ASAP model.

5.3. Output data

By running a single simulation on NXP's ASAP model, multiple sets of output data are generated that can be used by stakeholders. This includes information on the predicted behaviour of machines, lots, work-in-progress (WIP), and on-time deliveries, as shown in Figure 3. The simulated output is then converted into a format usable by stakeholders to make decisions during the CRITO discussion. The capgroup.rep file is particularly important for this thesis, as this file shows the number of forecasted moves by ASAP per cap group.

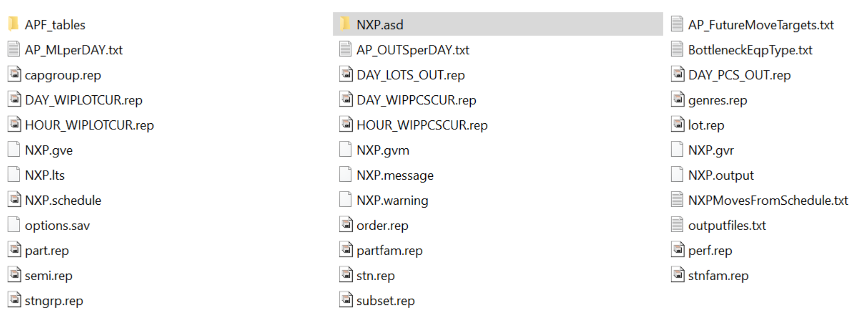

Figure 3 Overview output files.

In addition to the report files, data can also be observed through a Gantt viewer, as can be seen for machines in Figure 4. The Gantt viewer is an additional function that continuously displays the status, process and placement of lots or machines over a specified period. The legend is described in Appendix A.2.

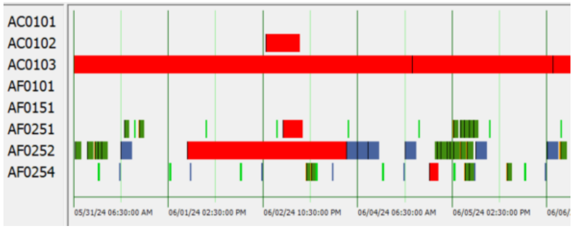

Figure 4 Gantt viewer example ASAP.

5.4. The accuracy of the existing model

Before improving the ASAP model of NXP, it is essential to evaluate its current effectiveness in simulation. Here the simulated moves for both ICN8 as a whole and the individual cap groups of ICN8 are compared to the actual moves, as these simulation results are most valuable for NXP. Stakeholders who use ASAP have reported that, while ASAP provides sufficient results for ICN8 as a whole, its simulations for individual cap groups are insufficient for identifying bottlenecks.

To assess the accuracy of the ASAP model, the daily number of simulated moves by ASAP over a 72-hour simulation period during 2023 will be compared to the actual results of 2023 for both ICN8 as a whole and individual cap groups. Due to IT issues encountered during the generation of the 72-hour model, the first three days of the historical 14-day model are used as an alternative to the 72-hour model. According to NXP, the models are the same but are configured slightly differently. Therefore, the results are comparable and valid for use in the validation process.

To calculate the model's accuracy, we exclude the worst 5% of samples. This decision is based on the fact that some samples are influenced by events such as fab stops and voltage dips, which cause extreme outliers that the ASAP model does not take into account.

We evaluate the model's accuracy based on its ability to forecast the daily number of moves per cap group. Because the scales of moves vary greatly between cap groups, the mean absolute percentage error (MAPE) (Khair, 2017)method will be used, as can be seen in equation (13), to calculate the accuracy of the model for each cap group (c) on each day (d).

$$MAPE_{d,c} = \frac{|real\ moves_{d,c} - ASAP\ moves_{d,c}|}{real\ moves_{d,c}} * 100\%$$

(13)

The downside of using MAPE is that it cannot handle zeros and can exaggerate the impact of outliers. Although the most extreme 5% of events, such as fab stops or voltage dips, are excluded, smaller cap groups may still encounter situations where the number of moves is very small or zero. For example, if only a single actual move is made and 51 moves are forecasted, MAPE would yield a 5000% error, making the impact of the result on the average out of proportion.

To counteract this issue, the total deviation of ICN8 is based on the average cap group, with a weight added to all cap groups based on the number of moves made in 2023. The weight ensures that the average deviation is based on the number of moves made rather than on the deviation per cap group. Furthermore, the focus is placed on the results of the 35 cap groups that make the most moves, which together accounted for 80% of the moves made in 2023. The remaining cap groups make too few moves, leading to excessive extreme deviations in the MAPE calculations, causing the results of the smaller cap groups to be disproportionate (Goodwin, 1999).

Since no validation model is perfect, the average daily number of deviated moves for the fab as a whole, along with the average sum of deviations per individual cap group, is also calculated to provide further insight. While this result does not directly indicate the model's accuracy, it offers additional insight when comparing this model to future modified versions.

The accuracy model with the calculations and results are detailed in Appendix B. The most valuable results are stated in Table 2.

Table 2 ASAP result 2023.

ASAP performance over 2023	
Average absolute total moves deviation per day (all data)	843%
Average absolute total moves deviation per day (best 95% data)	15.1%
Average total moves deviation per day (best 95% data)	14.7%
Average number of over forecasted moves per day (best 95% data)	1983
Average absolute moves deviation per cap group for 35 largest cap groups (best 95% data)	15.8%
Average absolute number of moves deviated per cap group for 35 largest cap groups (best 95% data)	3183
Median absolute moves deviation per cap group for 35 largest cap groups (best 95% data)	14.9%
Median absolute moves deviation per cap group for all cap groups (best 95% data)	17.3%

As anticipated by NXP stakeholders, the ASAP model failed to meet the target of a 5% average absolute move deviation per cap group. For 2023, the model demonstrated an absolute deviation of 15.8% per cap group for the 35 largest cap groups, which is three times higher than NXP's target.

Additionally, while the daily move deviation of ICN8 as a whole is slightly better than the cap group deviation, it still has a deviation of 15.1%. This is concerning because engineers had previously assumed that these deviations would be much smaller due to the fact that the overshoot and undershoot of cap groups would balance each other out. However, this assumption proved to be inaccurate.

Furthermore, using the median values of the results, it is evident that the remaining 20% of moves increase the overall deviation from 14.9 to 17.3 percent. Although this significant difference is influenced by the exaggeration of some cap groups, it indicates that smaller cap groups are more difficult to forecast accurately compared to the larger cap groups.

Lastly, the necessity of excluding the worst 5% of data becomes apparent. When including all data, the extreme deviations cause the model to have an average accuracy that is out of proportion (843% for whole fab simulations). By excluding the 5% worst data, the average deviation of the largest cap groups is 15.1%, which is close to the median cap group deviation of 14.9%. This indicates that it is unnecessary to exclude more data as the most extreme deviations are removed for the largest cap groups.

In figures 5 and 6 below, the deviations per sample per cap group of the 35 largest cap groups in the 2023 model:

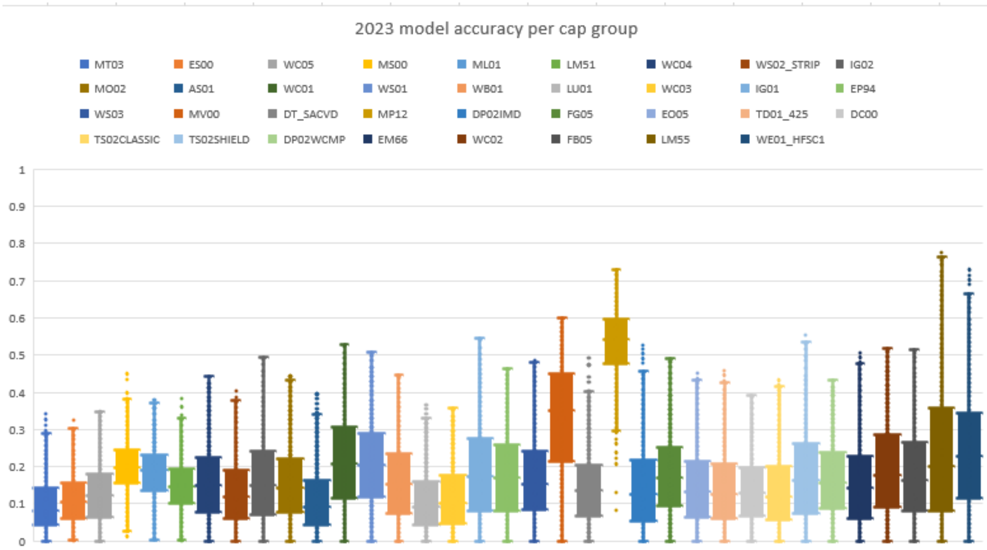

Figure 5 Absolute deviation biggest cap groups.

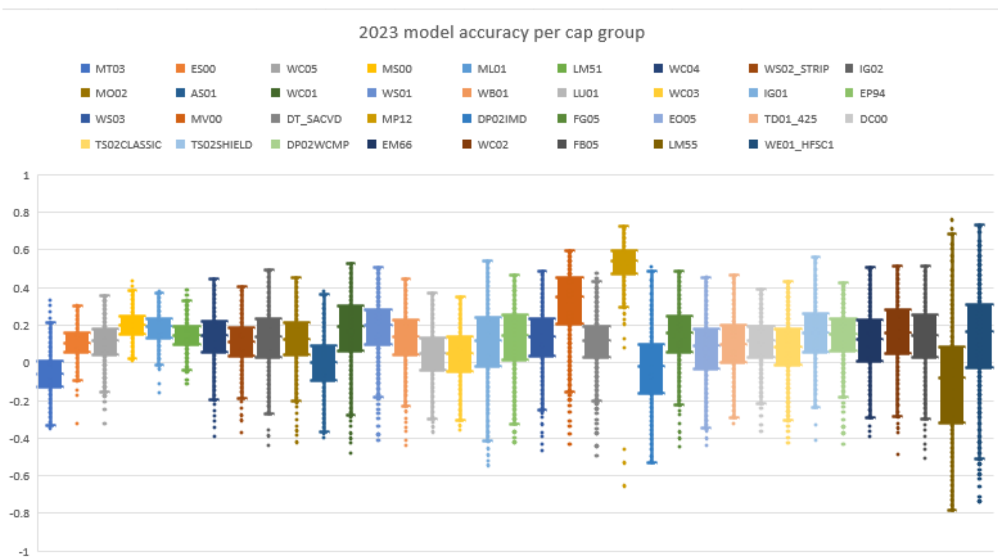

Figure 6 Deviation biggest cap groups.

Figures 5 and 6 illustrate the deviation results for each cap group, with the largest cap groups on the left to the smallest cap groups on the right. As can be observed, the model's performance varies significantly across different cap groups. Especially the cap groups MV00 and MP12 exhibit extreme deviations. The data indicates a clear trend: The smaller the cap group, the worse the simulation results produced by ASAP.

Additionally, as shown in Figure 6, most cap groups tend to overpredict, which further explains why the simulation results of ICN8 as a whole have a large move deviation.

5.5. Findings from the original model

Since 2016, NXP engineers have made significant efforts to improve NXP's ASAP model, implementing multiple changes over the years. These enhancements have resulted in a model that even the program developer, AMAT (Applied Materials), finds impressive. However, the results indicate that the current ASAP model falls significantly short of meeting NXP's 5% target for simulating absolute moves at the cap group level. The following conclusions can be drawn from the findings:

1) Simulation ICN8 as a whole: Excluding the worst 5% of outliers caused by circumstances such as fab stops or sudden breakdowns during 2023, the 72-hour simulation model of ASAP was able to simulate the total number of daily moves with an average absolute deviation of 15.1% per day.

2) Simulating individual cap groups: Excluding the worst 5% of outliers caused by circumstances such as fab stops or sudden breakdowns during 2023, the 72-hour simulation model of ASAP was able to simulate the moves per cap group with an average absolute deviation of 19.7% per day.

3) Overprediction tendency: The model tends to overpredict. During 2023, the model forecasted, on average, 14.7% more moves than the actual moves.

4) The smaller the cap groups, the larger the deviation: There is a clear trend that shows the fewer moves a cap group makes, the larger the deviation between the simulated moves of ASAP and the actual moves.

6. Assessing modifications

The primary goal is to improve the forecasting accuracy of NXP's ASAP model. By analysing and comparing potential areas for improvement, the most promising components are identified and validated for further development. These components are expected to have a positive impact on the model within the available timeframe.

6.1. Possible modifications

We identified several components of the existing ASAP model that, if modified, could significantly enhance the model's accuracy and performance. The areas for improvement were determined through research, discussions, and observations of the input data. Enhancing these components is expected to improve the model's forecasting accuracy. The identified components include operator behaviour, tool process time, lot order, tool chambers, and tool placement (Kusters, 2021). Each of these potential modifications is briefly discussed in this subsection.

6.1.1. Operator behaviour

The ASAP model of NXP does not account for operators and instead moves lots instantaneously to the next machine once the process is completed. This causes the model to overestimate the total number of moves made. Previously, engineers tried to forecast operator behaviour through the implementation of idle times for each machine. This approach ensures that the machine cannot process while it is idle. However, incorporating idle time files has not yet led to an improvement in the model.

6.1.2. Tool process time

Currently, the ASAP model assigns process times to production steps based on the average process time per cap group. However, there are significant variations between the process times required by different machines to perform specific process steps. By incorporating tool process times, the model will more accurate simulate tool behaviour.

6.1.3. Lot order

The model chooses the lot order to maximise the daily number of moves executed (Rahim, Ahmad, et al., 2013). This approach results in a relatively high number of forecasted moves on the first day, as the faster processes are executed. However, the faster and slower processes are more evenly distributed over the days. By implementing changes in how the lots are chosen, the model would better reflect actual conditions.

6.1.4. Tool chambers

A large number of tools at NXP consist of multiple chambers in which a process can be executed. Most of the time, these chambers have distinct characteristics and cannot execute identical steps (Kusters, 2021). Currently, the model does not take different chambers into account and instead assigns all functions of the tool to all its chambers. Additionally, the model does not account for the possibility that a separate chamber can break down, instead assuming that the entire machine is either operational or broken.

6.1.5. Tool placement

Producing an IC wafer involves multiple processes, which require the wafer to travel between various tools. The time needed to move a wafer depends on the distance between these machines. Currently, the ASAP model uses travel times between large areas to estimate the actual travel times between individual machines. Because these areas are relatively large, using historical travel times between specific tools rather than entire areas would more accurately reflect real conditions.

6.1.6. Most promising modifications

Due to the complexity of the ASAP model and the limited time available, the focus has been placed on developing and implementing tool process times and tool idle times. These adjustments are expected to improve the model's forecast accuracy while being feasible within the available timeframe. Despite previous unsuccessful attempts to integrate the idle time file, the new effort aims to explore the combined impact of idle and process times more effectively. The foundation of the idle time methodology is developed in combination with Herwin van Hoof.

6.2. Process time per tool

The process time per tool currently used in NXP's ASAP model is based on the average process time across all tools within a cap group. Previously, the decision to assign the average cap group process time to tools was based under the assumption that this generalisation would minimally affect the model while contributing to a more simplified version of ASAP.

6.2.1. Potential benefits of incorporating tool process times

Each machine within ICN8 possesses a unique set of properties, including its location, the number of runnable types of processes it can execute, the number of chambers it contains, and process speeds for different types of processes. These properties directly influence the machine's capacity to handle a certain volume of moves, making the performance of each machine distinct.

Currently, ASAP assigns the average process times per cap group to all machines within that particular cap group. The effect of generalising process time is neglectable in cases where the difference in process times between tools is minimal. However, upon closer examination of historical data, significant differences in process times between tools can be observed. This variation is primarily caused by newer machines, which often execute certain processes much faster than older machines. Figure 7 illustrates the process times of different machines performing the same process step.

II0301	IA130E11040AA	1771	IG02
II0302	IA130E11040AA	1440	IG02
II0305	IA130E11040AA	1425	IG02

Figure 7 Different process times between machines (measured in seconds) as recorded in the SFC file.

The impact of varying process times becomes especially significant during tool disruptions, such as when machines break down or undergo planned maintenance. A disruption of a newer machine with faster processing capabilities and more chambers significantly impacts overall process flow compared to a disruption of an older, slower machine with fewer chambers within the same cap group. This generalisation in process times can lead to an averaging effect within the simulation, causing the impacts of individual machine disruptions to be either too small or too large.

The potential benefit of including tool process times into the ASAP model lies in the ability to better capture the effects of the individual tools and allow the model to forecast moves more effectively.

6.2.2. Input data tool process times

NXP continuously collects data related to its manufacturing process in ICN8. Among these data are the tool process time per process for each machine, which is necessary for the new model. The data is currently stored in a single Excel file named “SFC_date”, which includes detailed information such as the process time per machine, the type of process, the number of chambers, and the specific cap group.

The process times in the SFC files are based on the average duration it takes for a lot or batch (group of lots) to enter a particular machine successfully, complete its processing, and exit. These data are then used to update a single Excel file, which contains the most current process times. As new data are added, older data are overwritten, so only the most recent data are stored.

The older ASAP model, which relies on average cap group process times, consistently uses the latest data. To ensure the new model remains accurate and consistent, it must also incorporate the most up-to-date tool process times. This approach improves forecast accuracy and ensures the model’s replicability.

Although there is no historical data for these process files due to ongoing data overwriting, since July 5th 2024, NXP has implemented a new feature within APF to save the older process file versions. Consequently, the new model’s sample size is limited to data collected after July 5th 2024.

6.2.3. process.txt

According to the beginner’s class notes from Applied Materials (*AutoSched AP Beginning Class Notes*, 2007), ASAP allows for the implementation of a so-called process time matrix. This matrix enables the specification of process times for each step based on predefined criteria, making it possible to assign different process times to individual machines. In addition, the process time matrix is also beneficial as it overrides the cap group process time implemented in the route file. As a result, there is no need to modify the route file of the older model, as ASAP will automatically prioritise the process time of the process file. Furthermore, if a particular step is not incorperated into the matrix, ASAP will use the cap group process time from the route file, preventing significant errors.

The process file is constructed according to the following details, as illustrated in Figure 8:

- STEP: Specifies the product and the corresponding process step, separated by a colon (“:”). This is necessary as different products may have varying times for the same step.
- STN: The machine (or station) capable of executing the specified step.
- PTIME: The process time required for the machine to complete the step, using historical process data as described in 6.2.2.
- PTUNITS: The units in which the process time is measured.

The functionality of all columns is explained in Appendix C.

	A	B	C	D
1	STEP	STN	PTIME	PTUNITS
2	rpart1:1	Stn1	30	min
3	rpart1:1	Stn2	25	min
4				

process.txt

Figure 8 Process time matrix example.

Additionally, to enable the use of the process.txt file, the option.def file must be modified accordingly, as shown in Figure 9.

17		file	starts.txt
18	GENRES_FILES	none	genres.txt
19	OPERATOR_FILE	none	
20	BOM_FILES	none	
21	SETUP_FILES	none	
22	PROCESS_FILES	file	process.txt
23	ATTACH_FILES	file	down_attach.txt
24		file	pm_attach.txt
25		file	tlots_attach.txt
26		file	rework_attach.txt
27		file	reducecapacity_attach.txt

Figure 9 Turn on process.txt in option.def.

6.2.4. Preliminary testing of the process.txt file

To verify the functionality of the process.txt file, we conducted initial testing using a simplified version of the ASAP model. This version includes two identical machines, each responsible for processing a large number of IC wafers, with each wafer requiring a single processing step for completion. We assigned identical properties to both machines, including equal processing times for this step.

In order to assess the model's impact, we first ran it without a process.txt file. As shown in Figure 10, both machines completed 120 moves in 24 hours, consistent with the route file's specification that each move takes 12 minutes per machine. Additional details of the simplified ASAP model are provided in Appendix C.

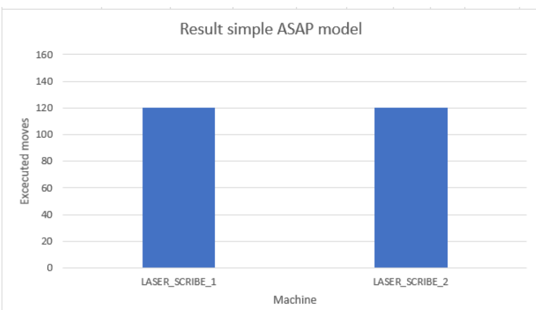

Figure 10 Simple ASAP result.

Next, we integrated the process.txt file with updated process times into the simplified ASAP model. In this configuration, we set the process times for "LASER_SCRIBE_1" and "LASER_SCRIBE_2" to 10 and 15 minutes. As shown in Figure 11, the simulation results indicate that the number of moves aligns with the specified settings: "LASER_SCRIBE_1" completing 144 moves and "LASER_SCRIBE_2" completing 96 moves over a 24-hour period in ASAP. This outcome confirms that the process.txt file correctly overrides the route file timings, ensuring machine-specific process times are accurately applied.

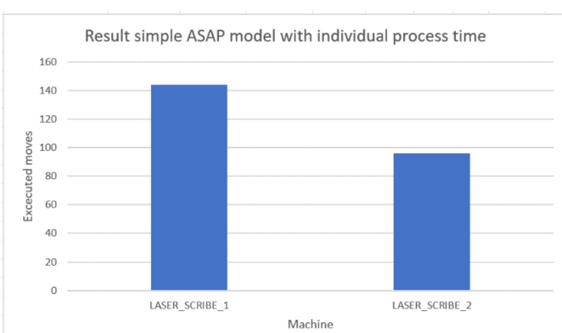

Figure 11 Simple ASAP result with process.txt file.

6.3. Idle time

The current ASAP model used by NXP does not incorporate operators. In the model, lots are moved instantaneously from one machine to another as soon as processing is complete. However, in reality, delays occur in the movement of lots between machines due to human errors and other delays.

Simulating operator behaviour is challenging because human interactions are difficult to predict. Over the past several years, model engineers have made multiple attempts to incorporate operator behaviour into the simulation. Despite these efforts, these attempts did not lead to robust improvements in the model's forecast accuracy. Therefore, this subsection focusses on incorporating machine idle times of machines as a more effective and robust option for incorporating human influence into ASAP.

6.3.1. The potential benefit of implementing idle time

As illustrated in the 2023 results in Figure 6, the model tends to overestimate the number of executed moves. This overprediction is primarily caused by the model's failure to account for operator behaviour, leading to the unrealistic assumption that lots move instantaneously between machines.

In reality, lots often experience delays in movement, contributing to increased idle time for machines. Idle time is defined as the percentage of a day when a machine is capable of processing but remains inactive. This idle time can fluctuate significantly from day to day and between different machines.

At ICN8, machines that process quickly and have a relatively high number of chambers tend to experience lower idle times. NXP prioritises these machines to maximise their utilisation. In contrast, machines that process more slowly and have fewer chambers are given lower loading priority, resulting in a higher idle time.

To address this issue in ASAP and better capture the impact of operator efficiency, we propose incorporating idle time into the model. By doing so, we expect the model to generate more accurate results, as it now incorporates tool process times, leading to a more realistic representation of machine utilisation and cap group performance.

6.3.2. Input data idle time

Since 2020, NXP has been continuously gathering machine performance data to create an idle time data file. These data are stored in APF, from which the necessary data can be extracted. Figure 12 provides an overview of the data from APF that is needed to generate the idle time files. These data includes machine Availability Efficiency (AE) and historical idle times. The machine efficiency represents the percentage of time the machine is able to operate on a daily basis, whereas the idle time reflects the percentage of time the machine is not processing even though it could have been.

Although the idle time data has been collected since 2020, data on tool process times have only been gathered since July 5th 2024, making the process time the limiting factor for generating samples.

Datum	Eqp	OE	AE	Idle
6/1/2024	AF0251	7.16	100	0.9284
6/1/2024	AF0252	19.84	100	0.8016
6/1/2024	AF0254	14.33	100	0.8567
6/1/2024	AF0255	27.63	100	0.7237
6/1/2024	AM0101	49.78	100	0.5022
6/1/2024	AM0102	50.25	100	0.4975
6/1/2024	AS0101	0.01	100	0.9999
6/1/2024	AS0102	0	100	1
6/1/2024	AS0103	0	100	1
6/1/2024	AS0104	0	100	1
6/1/2024	AS0106	0.84	100	0.9916
6/1/2024	AS0109	0.33	82.09	0.9967
6/1/2024	AS0110	0	100	1

Figure 12 Historical idle time per machine.

The idle time data contain outliers due to daily fluctuations and interruptions, such as machine breakdown or fab stops. To ensure that the idle times used for the model are reliable and representative, the average idle time is calculated over multiple historical days, excluding the outliers. The steps taken to exclude the outliers are further highlighted in subsection 7.1.2, which is developed in combination with Herwin van Hoof.

6.3.3. cal_reducecapacity.txt & reducecapacity_attach.txt

The method for implementing idle time into the ASAP model involves incorporating two new files into ASAP: “cal_reducecapacity.txt” and “reducecapacity_attach.txt”.

In the “cal_reducecapacity.txt” file, the idle time for each machine is specified. These data are entered in the MTTR (Mean Time to Repair) column, which represents the amount of time, relative to MTBPM (Mean Time Between Preventive Maintenance), during which the machine cannot be used in the model. By setting the MTBPM value to 1, the model turns the machine off for an amount of time corresponding to the value in the MTTR column. The value in MTTR must be positive and less than 1 (relative to MTBPM):

- 0 means the machine is never idle and is continuously used for processing.
- 1 means the machine is always idle and cannot be used for processing.

The value 1 is chosen for MTBPM as it serves as a convenient reference point, representing the machines entire available time or 100 percent.

Figure 13 illustrates how the file is structured for two machines, where “LASER_SCRIBE_1” is never idle and “LASER_SCRIBE_2” is idle for 50 percent of the time.

PMCALNAME	PMCALTYPE	TRACE	MTBPMDIST	MTBPM	MTBP	MTTR	MTTRUNITS	PM_ACTLIST
ReduceCapacity_LASER_SCRIBE_1	MTTPM_by_cal	gantt	constant	1	hr	0	hr	ReduceCapacityPMActList
ReduceCapacity_LASER_SCRIBE_2	MTTPM_by_cal	gantt	constant	1	hr	0.5	hr	ReduceCapacityPMActList

Figure 13 cal_reducecapacity.txt example.

The “reducecapacity_attach.txt” file functions as the file where the correct idle time is assigned to the appropriate machine. This file is necessary for ASAP, as ASAP needs to know which delays are associated with which machines. If the machines within NXP remain the same, the data in this file do not need to be changed. Figure 14 shows the structure of the file as it must be written. Here, the names in the CALNAME field must match the names in the PMCALNAME field from Figure 13. The explanation of the remaining functions of the columns can be found in Appendix C.

RESTYPE	RESNAME	CALTYPE	CALNAME	FOADIST	FOA	FOA2	FOAUNITS
stn	LASER_SCRIBE_1	pm	ReduceCapacity_LASER_SCRIBE_1	uniform	0	0	hr
stn	LASER_SCRIBE_2	pm	ReduceCapacity_LASER_SCRIBE_2	uniform	0	0	hr

Figure 14 reducecapacity_attach.txt example.

6.3.4. Preliminary testing idle time

To verify the functionality of the idle files, preliminary testing is conducted using the same simplified version of the ASAP model without including the process file, as described in subsection 6.2.4. For the verification of the idle time effect, an idle time of 0.5 hours per hour is implemented into the "LASER_SCRIBE_2" machine, while the "LASER_SCRIBE_1" machine has none, as shown in Figure 13.

By running the model with the idle time files, where a 50% idle time is assigned to the "LASER_SCRIBE_2" machine, the "LASER_SCRIBE_2" machine executed significantly fewer moves. The results, shown in Figure 15, confirm that the "LASER_SCRIBE_2" machine executed 60 moves instead of the previous 120 moves, as expected with an idle time of 50%. Upon closer inspection in the Gantt viewer, in Figure 16, it can be seen that "LASER_SCRIBE_2" machine is idle for 50% of the time (indicated in yellow), proving the correct functioning of the idle files.

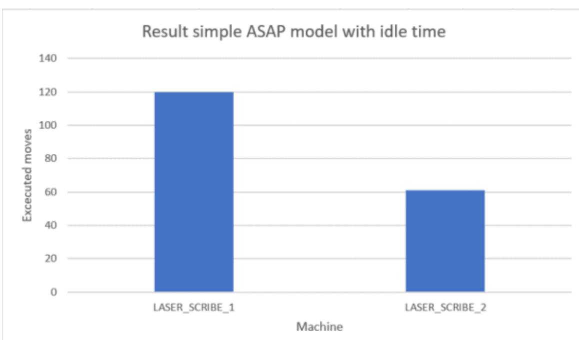

Figure 15 Simple ASAP result with cal_reducecapacity.txt & reducecapacity_attach.txt files.

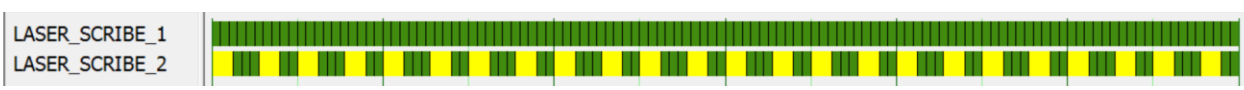

Figure 16 Gantt viewer idle time.

6.4. Additional implementations

In the ASAP model of NXP, step names which are used in the process.txt file include an extra colon for readability, separating the step name from the step flow. However, the process.txt file only supports a single colon, causing errors when the additional colon is used. To resolve this, after consulting with the model engineers, we replaced the extra colon in the step names within the process.txt, wip.txt and route.txt file with the "@" symbol. This adjustment allows the ASAP model to function correctly while preserving the readability of the step names.

7. Implementation modifications into NXP's ASAP model

After developing and validating potential modifications in a simplified version of the ASAP model, as outlined in section 6, the changes are implemented in NXP's operational model. This section details the integration of tool process times and idle times into the model, describing the steps taken throughout the modification process.

7.1. File generation for the process and idle times

Generating the process file and idle time files for NXP's model presents a complex challenge. While the simple model of section 6 worked with fabricated input data in a specific scenario, real historical data is now used, processed, and applied to all possible cases.

7.1.1. Implementing tool process times

To generate an up-to-date process.txt file, data of the process step names, types of products, tools, and process times must be integrated. The required input data are stored in two daily-updated files, namely the route.txt file and the newly generated SFC file.

The route.txt file, containing 34713 lines of data, specifies the required cap groups and recipes to execute each process step of a product. The SFC file, which contains 25778 lines of data, consist of historical process time per machine per recipe, including details about the cap group and number of chambers. To determine the time required for machines to execute a particular process, the appropriate machines must be matched with the corresponding process. Both the route.txt and the SFC files contain information about the recipes and cap groups, enabling ASAP to link the cap group and recipe for each process to the machines of that cap group that are capable of performing it. For each combination, all the necessary data is available to generate a single line of data for the process.txt file. Once all matches are found and correctly placed into the process.txt file, it is complete and ready for implementation.

Given the large quantity of data involved, we developed a Visual Basic for Applications (VBA) script to generate the process.txt files. The code compares each line in the route.txt file with the SFC file to match cap groups and recipes. Comparing all possible data points would require the script to go through (34713 * 25778 = 895 million checks) per process.txt file, which would be time-consuming. Therefore, the SFC file is first sorted alphabetically by cap group. The code then narrows down the search within the cap groups.

By narrowing the search area to the 134 different cap groups, the code now needs to make approximately 6.3 million checks to find all matches. (895 million / 134 = 6.7 million). With these adjustments, along with some other optimisations, the laptop used in this research generates the process.txt file in about 10 minutes, which is a acceptable timeframe, considering that multiple samples need to be generated for validation.

Whenever there is a match between the two files, the following steps are executed in the code:

1) The code verifies that the data are error-free. Specifically, the runcard of the SFC file may not contain the value "na". If the runcard entry is "na", the corresponding data is excluded from the process.txt file.
2) The code ensures that the SFC file contains at least one chamber. Although these data points should have already been filtered out during step 1 (removing "na"), this additional check is necessary as a safeguard to eliminate any remaining errors.

3) The historical processing time in the SFC file must be batches-based (average historical process time per groups of lots). This batched-based data accounts for almost all data. Exceptions are not taken into account due to limited research time.
4) The process name (or step name) and product are added to the STEP column in the format: (stepname:product), which can be extracted from the route file. As noted in in subsection 6.4, the step name must not contain a colon.
5) The corresponding machine is added to the STN column.
6) The historical processing time from the SFC file is multiplied by the number of units per batch (column UNITS in the SFC file) and then added to the PTIME column. The historical processing time reflects the average processing time per lot (multiple lots can be processed at once), but the actual processing time per lot should be used instead. For example, if a batch requires 4000 seconds to process four lots, the process file indicates that each lot takes 1000 seconds on average. However, in reality, each individual lot spends 4000 seconds in the machine.
7) Lastly, because the processing time is measured in seconds, the term "SEC" is added to the PTUNITS column.

Eventually, whenever the script is executed, it generates a process file containing approximately 120,000 lines of data. The file must then be saved in the input data list, see Figure 2. Additionally, the option.rdf file must be adjusted accordingly, see Figure 9, after which the ASAP model is ready to forecast using the tool processing times. A complete description of the code can be found in Appendix D.

7.1.2. Implementing idle times

To effectively include idle time in NXP's model, the idle time file must be developed and updated daily. At first, both the cal_reducecapacity.txt and the reduce_capacityattach.txt files need to be developed. Once the files are developed, the MTTR column of the cal_reducecapacity.txt must be updated daily to ensure the latest data is used to calculate the idle times.

Although the framework for these files and the option.rdf file has already been incorporated into NXP's model, both files currently lack data and, therefore, have no impact on the model's outcome. To populate both files correctly and apply idle times that reflect real-world conditions, we will use the idle time calculations provided by our colleague, Herwin van Hoof.

The input data for the new model will be similar to that used in the simple ASAP model, as illustrated in figures 13 and 14. However, the new model will use stations from the stn.txt rather than the LASER_SCRIBE machines. Besides, the MTTR column, which reflects idle time per machine, must be updated daily.

The forecasted idle time implemented in the MTTR columns will be based on 28 days of historical data extracted from APF, as illustrated in Figure 12. The process for updating the idle time for each machine is executed by a VBA script and requires the following steps:

1) Gather historical idle time data from 28 days prior to the simulation for a particular machine with an efficiency availability of at least 50%. Historical data samples where the machine availability is lower than 50% are excluded as they may represent abnormalities such as breakdown.
2) Calculate the 80^{th} percentile of the remaining data. This threshold is used to exclude abnormally large idle time samples caused by factors such as insufficient operators or variations in IC wafer types processed in the fab. By using the 80^{th} percentile, we establish a more realistic and representative idle time for the model.

3) The maximum idle time per machine that can be implemented into the idle files is capped at 20%. Machines with high idle time samples are typically used less frequently and exhibit greater day-to-day variability. Including these high idle times can negatively affect the model's performance, potentially forecasting unnecessary bottlenecks if the machine usage increases.

By developing the cal_reducecapacity.txt and reduce_capacityattach.txt files and updating the MTTR column on a daily basis, the idle time files will represent accurate data that can be used to increase ASAP's forecast accuracy. The whole code is written in Appendix D.

7.2. Running the ASAP model

The next step is to transform NXP's ASAP model and running the modified version. For this, we use the model transformer to modify the 14-day models. Due to IT errors in generating the 72-hour model, we used the first 3 days of the 14-day model for testing the samples, as is described in subsection 5.4. This approach maintains consistency with the project scope and provides results comparable to those of the intended 72-hour model.

After running both the original and the modified models, no additional errors were encountered. The capgroup.rep file, which contains the data used to test moves, as can be seen in section 5.3, was successfully updated. Furthermore, a difference in running time between the two models can be observed in figures 17 and 18:

```
Factory Read   =     0:00:03
Initialization =     0:00:00
Simulation     =     0:00:51
Final Reports  =     0:00:01
Total          =     0:00:55
```

Figure 17 Running time old 14 day model.

```
Factory Read   =     0:00:05
Initialization =     0:00:00
Simulation     =     0:00:54
Final Reports  =     0:00:00
Total          =     0:00:59
```

Figure 18 Running time new 14 day model.

The new model takes approximately 7% longer to run simulations. However, this is not an issue for NXP as the total daily simulation time is still neglectable. Currently, NXP runs 9 required simulations per day, totalling a simulation time of around 480 seconds. A 7% increase would add only 34 seconds to the daily runtime, which has no impact on operations.

7.3. Verification of model integrity

To prevent the introduction of unintended effects on the model from the new files, such as disregarding essential data from other files, it is crucial to validate the results. In the case the interactions between the files in the new model are unplanned, unwanted results can appear which can lead to misinterpretations, and therefore to inaccurate conclusions. Verifying the model's integrity increases reliability while minimising the risk of misinterpretations.

To confirm that the modified files interact as intended, we configured them to use the same input data as the older model. If the simulation results of the new model are identical to those of the older model, it confirms that the new files have the intended effect on the model.

In the process.txt file, we adjusted the process times for each machine to the average time per cap group as stated in the route.txt. Additionally, the idle times in the MTTR column of the cal_reducecapacity.txt file are set to zero, indicating that there is no idle time. These modifications ensure that the new model's files contain the same input data as the older model.

STEP	STN	PTIME	PTUNITS		cap group
PA100TM	AM0101	2034	sec		2047.998
PA100TM	AM0102	2070	sec		2047.998
PA100TM	MT0901	378	sec		388.002
PA100TM	MT0902	397	sec		388.002
PA100TM	WC0551	2565	sec		2349
PA100TM	WC0552	2455	sec		2349
PA100TM	WC0555	2530	sec		2349
PA100TM	WC0557	2490	sec		2349
PA100TM	WC0558	2448	sec		2349
PA100TM	WC0559	2489	sec		2349

Figure 19 Calculated cap group times used in the tool process time file (PTIME).

PMCALNAME	PMCALTYF	TRACE	MTBPMDI	MTBPM	MTBPMUI	MTTR	MTTRUNIT	PM_ACTLIST
ReduceCapacity_AF0251	MTTPM_bgantt		constant	1	hr	0	hr	ReduceCapacityPMActList
ReduceCapacity_AF0252	MTTPM_bgantt		constant	1	hr	0	hr	ReduceCapacityPMActList
ReduceCapacity_AF0254	MTTPM_bgantt		constant	1	hr	0	hr	ReduceCapacityPMActList
ReduceCapacity_AF0255	MTTPM_bgantt		constant	1	hr	0	hr	ReduceCapacityPMActList
ReduceCapacity_AM0101	MTTPM_bgantt		constant	1	hr	0	hr	ReduceCapacityPMActList
ReduceCapacity_AM0102	MTTPM_bgantt		constant	1	hr	0	hr	ReduceCapacityPMActList
ReduceCapacity_AS0101	MTTPM_bgantt		constant	1	hr	0	hr	ReduceCapacityPMActList
ReduceCapacity_AS0102	MTTPM_bgantt		constant	1	hr	0	hr	ReduceCapacityPMActList
ReduceCapacity_AS0103	MTTPM_bgantt		constant	1	hr	0	hr	ReduceCapacityPMActList
ReduceCapacity_AS0104	MTTPM_bgantt		constant	1	hr	0	hr	ReduceCapacityPMActList
ReduceCapacity_AS0106	MTTPM_bgantt		constant	1	hr	0	hr	ReduceCapacityPMActList
ReduceCapacity_AS0108	MTTPM_bgantt		constant	1	hr	0	hr	ReduceCapacityPMActList
ReduceCapacity_AS0109	MTTPM_bgantt		constant	1	hr	0	hr	ReduceCapacityPMActList
ReduceCapacity_AS0110	MTTPM_bgantt		constant	1	hr	0	hr	ReduceCapacityPMActList
ReduceCapacity_AS0111	MTTPM_bgantt		constant	1	hr	0	hr	ReduceCapacityPMActList

Figure 20 Set MTTR values to zero.

After running the new model with adjusted input data, we compared the results in moves per cap group with those of the older model. As shown in Figure 21, both models produce identical output results for the moves per cap group.

PERIOD	RELATIVE	SUPERSET	LOTSTARTS	LOTCOMPS	CYCLECUR	CYCLEAVG	PRC
~Report time: 07/26/2024 06:30:00 :							
25/07/2024 06:30:00	Y	AM01	55	55	0:25:11	1:29:28	33
25/07/2024 06:30:00	Y	MV00	342	340	0:29:40	0:17:10	9
25/07/2024 06:30:00	Y	MT09	41	39	0:49:36	0:41:25	34
25/07/2024 06:30:00	Y	WS01	360	337	0:42:14	2:21:41	29
25/07/2024 06:30:00	Y	WC05	1045	981	0:57:58	1:37:20	33
25/07/2024 06:30:00	Y	FG05	247	180	6:21:24	8:13:41	41
25/07/2024 06:30:00	Y	MT03	1577	1552	0:15:56	0:26:30	20
25/07/2024 06:30:00	Y	MP06	9	9	0:27:31	0:19:23	3
25/07/2024 06:30:00	Y	LM40	138	119	4:31:39	2:48:48	25
25/07/2024 06:30:00	Y	LM00	46	43	1:44:43	1:54:31	35
25/07/2024 06:30:00	Y	LM51	1124	904	4:53:59	4:15:03	16
25/07/2024 06:30:00	Y	LM55	170	144	3:22:18	5:15:24	14
25/07/2024 06:30:00	Y	ML01	1066	1058	0:08:22	0:14:39	1
25/07/2024 06:30:00	Y	EP94	325	183	14:36:54	7:21:18	13

1	PERIOD	RELATIVE	SUPERSET	LOTSTARTS	LOTCOMPS	CYCLECUR	CYCLEAVG	PRC
2	~Report time: 07/26/2024 06:30:00 :							
3	25/07/2024 06:30:00	Y	AM01	55	55	0:25:11	1:29:28	33
4	25/07/2024 06:30:00	Y	MV00	342	340	0:29:40	0:17:10	9
5	25/07/2024 06:30:00	Y	MT09	41	39	0:49:36	0:41:25	34
6	25/07/2024 06:30:00	Y	WS01	360	337	0:42:14	2:21:41	29
7	25/07/2024 06:30:00	Y	WC05	1045	981	0:57:58	1:37:20	33
8	25/07/2024 06:30:00	Y	FG05	247	180	6:21:24	8:13:41	41
9	25/07/2024 06:30:00	Y	MT03	1577	1552	0:15:56	0:26:30	20
10	25/07/2024 06:30:00	Y	MP06	9	9	0:27:31	0:19:23	3
11	25/07/2024 06:30:00	Y	LM40	138	119	4:31:39	2:48:48	25
12	25/07/2024 06:30:00	Y	LM00	46	43	1:44:43	1:54:31	35
13	25/07/2024 06:30:00	Y	LM51	1124	904	4:53:59	4:15:03	16
14	25/07/2024 06:30:00	Y	LM55	170	144	3:22:18	5:15:24	14
15	25/07/2024 06:30:00	Y	ML01	1066	1058	0:08:22	0:14:39	1
16	25/07/2024 06:30:00	Y	EP94	325	183	14:36:54	7:21:18	13

Figure 21 ASAP results old model left and new model with process and idle files right.

The identical results confirm that the use of the modified files in the model does not introduce unintended effects. This verification ensures that the new files function as expected, thereby enhancing the reliability of the results.

8. Validating the modified model's results

We validate the new model's performance by comparing it against both the old model and real-world results. To do this, we generate 30 daily samples from 25 July 2024 to 24 August 2024 using the modified 14-day model and analyse each sample. Here, we compare the daily deviations in moves for the first three days of each sample to ensure the results align with the scope of the 72-hour simulation performance.

Our assessment focuses on the daily deviation in moves at both the individual cap group level and the overall fab level. By examining these deviations, we aim to determine whether the adjustments have led to improvements, quantify the impact of the improvements, and identify which cap groups are benefitting or adversely affected by the changes. This analysis will offer deeper insights into the new model, supporting its implementation and justifying the replacement of the older NXP model.

8.1. Comparison of the new and old models

To compare the performance of the new model with the old model, the accuracy of the simulated number of moves per cap group in both the new and old models must be compared with the actual executed moves per day. We have required data for both the old model, the modified model and actual performed moves in identical environments. We have developed a script to automate the process of extracting, filtering, and organising the data from multiple sources into a single Excel file for efficient analysis.

By systematically storing the data from the selected samples, the deviation in moves can be easily compared at both the cap group level and the fab level. We present the deviated number of moves from the first three days of the validation period at both levels in the following table:

Table 3 Results 30 samples of 2024 for old and new model.

Comparing new and older model (best 95% data)	Old model	New model
Sum of daily average number of absolute moves deviation per cap group for all cap groups	3811	2307
Average daily number of total absolute number of deviated moves per day	3099	644
Number of large cap groups better	7	28
Number of cap groups better	30	104

When comparing the average performance for both the individual cap groups and the fab as a whole, the new model has, on average, 2307 deviated moves, while the older model has 3811. The new model shows significantly less deviation compared to the older model. The average deviation per sample shows a 39.2% reduction at cap group level and a 79.4% reduction at fab level. The formulas are stated in equations (14) and (15), where N is the total number of machines.

$$cap\ move\ reduction = \frac{\sum_{t=1}^{N}(1 - \frac{total\ cap\ dev\ new\ model_t}{total\ cap\ dev\ old\ model_t})}{N} = 39.2\% \quad (14)$$

$$fab\ move\ reduction = \frac{\sum_{t=1}^{N}(1 - \frac{total\ fab\ dev\ new\ model_t}{total\ fab\ dev\ old\ model_t})}{N} = 79.4\%$$

(15)

To further demonstrate the effect of the modifications, the cap group improvement per sample is displayed in Figure 22. This indicates that the modifications have a positive impact, with even the least successful achieving a 28 percent improvement.

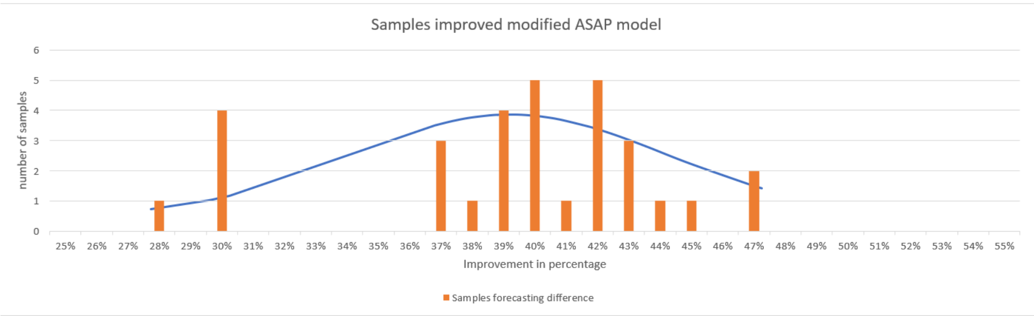

Figure 22 Comparison of deviation in moves between old and new model across samples.

8.2. Analyse new model performance

In addition to comparing the new and old models, it is important to analyse why the modified model performs better than the older model. The focus will be on visualising the forecast's accuracy for each cap group and evaluating why the modified model achieved better results by comparing the results to the older model. For this, the 35 largest cap groups will be compared to each other just as is done in the accuracy model in subsection 5.4.

When comparing the results from the 2023 model, 30 samples of the 2024 model and the 30 samples of the modified model, new data are obtained, as can be seen in Table 4.

Table 4 Improvement of modified model.

Comparing 72-hour model results using best 95% of samples	Model 2023	Model 30 samples 2024	Modified model 30 samples 2024
Average absolute total moves deviation per day (all data)	843%	23.2%	5%
Average absolute total moves deviation per day (best 95% data)	15.1%	22.3%	4.1%
Average total moves deviation per day (best 95% data)	14.7%	22.3%	2.2%
Average number of over forecasted moves per day (best 95% data)	1983	2945	277
Average absolute moves deviation per cap group for 35 largest cap groups (best 95% data)	15.8%	19.1%	11.7%
Average absolute number of moves deviated per cap group for 35 largest cap groups (best 95% data)	3183	3811	2307
Median absolute moves deviation per cap group for 35 largest cap groups (best 95% data)	14.9%	19%	10.6%
Median absolute moves deviation per cap group for all cap groups (best 95% data)	17.3%	21.6%	15.2%

The following conclusions can be drawn from the table:

1) There were no voltage dips or fab stoppages during the sampling period. As a result, the model did minimally benefit from removing the worst 5% of data since no disruptions occurred during the 30 samples.
2) The model NXP made less accurate forecasts in 2024 compared to 2023. This is caused by frequent changes in the fab, leading to periods that were relatively easier or more difficult to forecast. In 2024, the plant underwent other changes compared to 2023, which caused increased idle time. Nevertheless, the modified model performed better than the older model did with the 2023 data.
3) The modified model showed significant improvement in both the plant level and the cap group level. These improvements are also better than the results of the older 2023 model.
4) The median of the 35 largest cap groups with 95% data is relatively close to the 95% median data. This assures that extreme deviations do not exaggerate the MAPE results.

By visualising the results, we can clearly see how each cap group is forecasted and whether the forecasted samples are too high or too low. The comparison of absolute accuracy between the models, illustrated in figures 23 and 24, reveals that the modified model forecasted most cap groups significantly better.

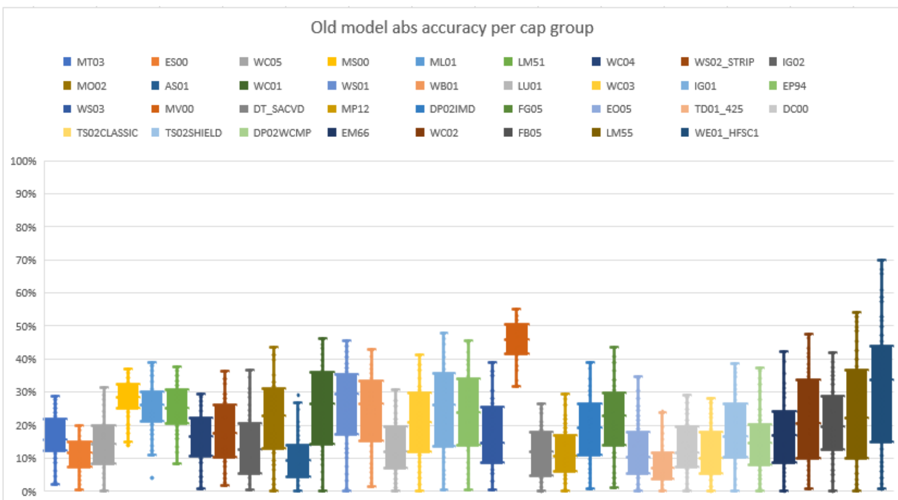

Figure 23 Old model absolute deviation for biggest cap group.

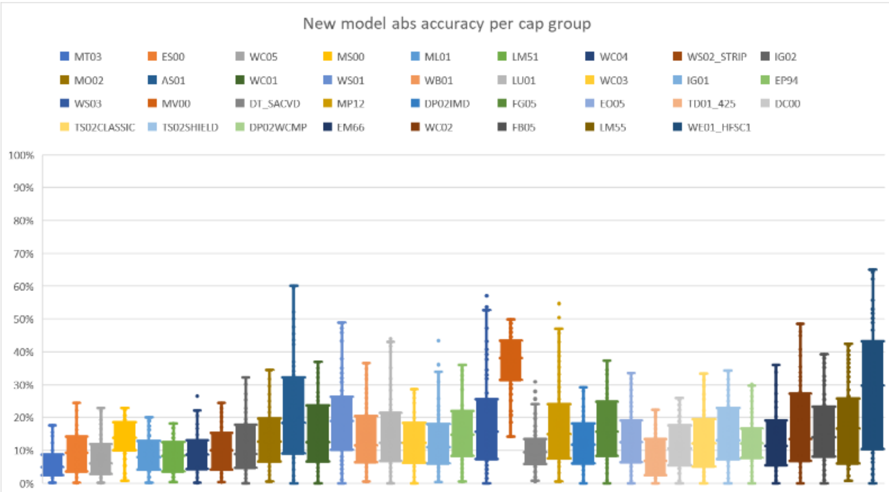

Figure 24 New model absolute deviation for biggest cap group.

A comparison of the average MAPE values per cap group, shown in Table 5, demonstrates that 28 of the 35 cap groups show improved results. While most cap groups are now better forecasted, the model still cannot forecast cap groups with an average absolute move deviation of less than 5%.

Table 5 Comparing absolute MAPE moves per cap group.

average abs MAPE deviation	old model	new model	average abs MAPE deviation	old model	new model	average abs MAPE deviation	old model	new model
MT03	16%	6%	WS01	26%	20%	EO05	12%	14%
ES00	11%	9%	WB01	24%	14%	TD01_425	8%	8%
WC05	14%	8%	LU01	13%	15%	DC00	13%	11%
MS00	28%	14%	WC03	21%	12%	TS02CLASSIC	11%	13%
ML01	26%	9%	IG01	25%	13%	TS02SHIELD	18%	15%
LM51	25%	8%	EP94	23%	15%	DP02WCMP	15%	13%
WC04	16%	9%	WS03	17%	19%	EM66	17%	13%
WS02_STRIP	18%	10%	MV00	45%	37%	WC02	22%	17%
IG02	14%	11%	DT_SACVD	12%	10%	FB05	21%	16%
MO02	22%	14%	MP12	12%	18%	LM55	24%	18%
AS01	10%	21%	DP02IMD	19%	12%	WE01_HFSC1	32%	28%
WC01	25%	15%	FG05	23%	17%			

To further analyse the deviations across cap groups, we examine both the undershoots and overshoots of the models. As illustrated in figures 25 and 26, the older model, similar to what is shown in Figure 6, over-forecasts the moves of almost all cap groups, leading to an overprediction of the entire fab targets. In contrast, the modified model does not consistently over-forecast, aligning much closer to the real moves. This leads to a significant improvement in the accuracy of the entire fab simulation results.

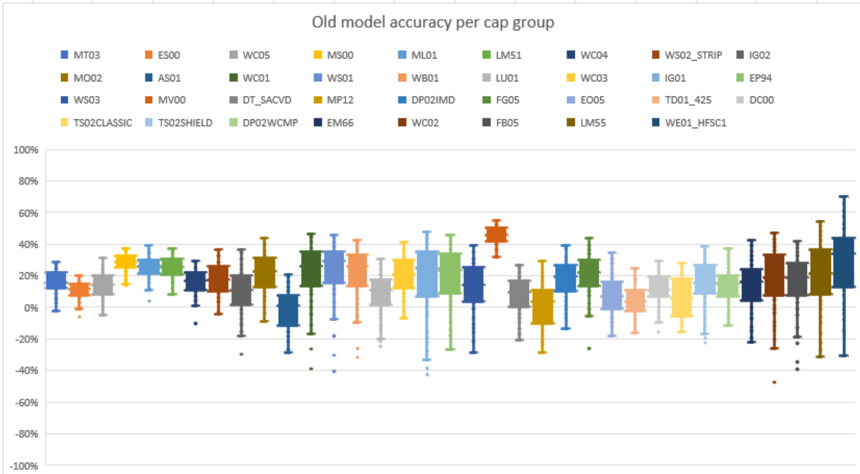

Figure 25 Old model deviation for biggest cap group.

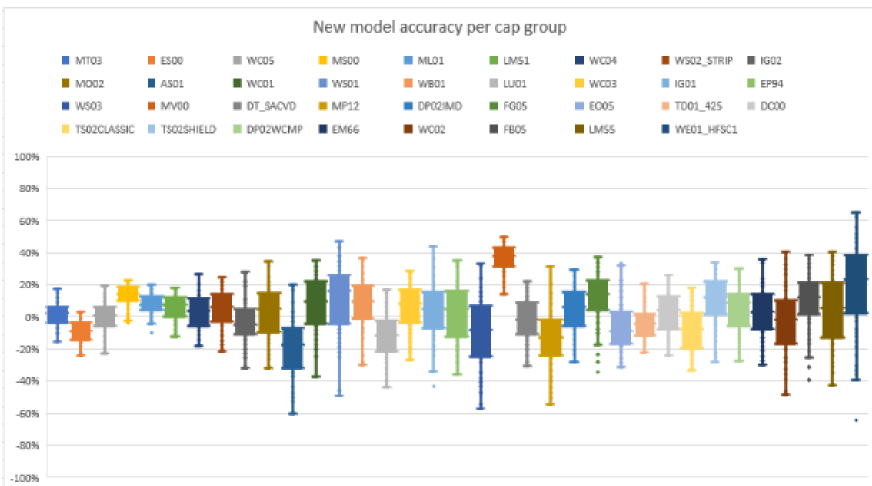

Figure 26 New model deviation for biggest cap group.

When comparing the average deviated MAPE per cap group, it becomes apparent that most cap groups still tend to over-forecast. In addition, all cap groups that performed worse due to the modifications now under-forecast on average.

Table 6 Comparing MAPE moves per cap group.

average MAPE deviation	old model	new model	average MAPE deviation	old model	new model	average MAPE deviation	old model	new model
MT03	16%	2%	WB01	22%	9%	EO05	8%	-6%
ES00	11%	-9%	WB01	22%	9%	TD01_425	4%	-4%
WC05	14%	0%	LU01	9%	-12%	DC00	11%	3%
MS00	28%	14%	WC03	20%	6%	TS02CLASSIC	6%	-8%
ML01	26%	8%	IG01	19%	3%	TS02SHIELD	17%	10%
LM51	25%	6%	EP94	21%	3%	DP02WCMP	14%	5%
WC04	15%	3%	WS03	13%	-10%	EM66	14%	2%
WS02_STRIP	18%	5%	MV00	45%	37%	WC02	18%	-3%
IG02	10%	-3%	DT_SACVD	8%	-2%	FB05	16%	9%
MO02	21%	3%	MP12	1%	-14%	LM55	21%	5%
AS01	-1%	-19%	DP02IMD	18%	4%	WE01_HFSC1	30%	18%
WC01	22%	9%	FG05	22%	12%			

9. The value of ASAP

To further highlight the need for implementing the modifications into ASAP, we analyse the financial impact of both the short-term and long-term effects of modifying NXP's ASAP model. The short-term analysis evaluates the implementation of the modified ASAP model, examining costs, risks, and expected benefits. In contrast, the long-term analysis focuses on the potential benefits and costs associated with further improving the model until it can effectively pinpoint bottlenecks. By evaluating these two perspectives, we can estimate the return on investment (ROI) for the modified model and the potential future version of ASAP, thereby justifying this thesis and supporting future research on ASAP.

We calculate the ROI based on the values and costs associated with ASAP. Since the benefits and some costs can only be determined once the model is implemented, which is not done during the writing of this thesis, various estimates are necessary. These estimates are supported by input from stakeholders and the 2016 business case, which provides data that justified the decision of purchasing ASAP in 2016. Due to the sensitivity of the data, all monetary values in this section are expressed in units, where one unit represents the engineering costs per hour.

9.1. Short-term costs and benefits

In the short-term, modifying ASAP is expected to enhance its performance. Although the modified model will not fully realise its potential for identifying bottlenecks that are used during CRITO discussions, these modifications represent a significant step toward that goal, and are expected to yield other short-term benefits.

9.1.1. Short-term costs

The costs of modifying the model primarily stem from the human resources needed for developing, testing and implementing. The costs for modifying ASAP include:

Developing and validating costs

Approximately 70 hours of engineering time was spent in meetings, validation, data preparation, and feedback to modify NXP's ASAP model.

Implementation costs

The modified ASAP model is implemented into APF, the program used to run NXP's existing ASAP model. This implementation is necessary as APF automatically generates new samples alongside NXP's ASAP model. Once the decision to replace the old model is made, switching between the two models can be performed easily. Implementing this modified model required approximately 24 hours of model engineers' time.

Testing and validation

The modified model's accuracy in forecasting moves is validated using 30 samples. Although these samples already indicate that the modified model outperforms the older version, additional samples and WIP data are needed before full implementation. Over time, more samples will be automatically generated through APF, which will resolve the issue of sample scarcity over time. However, validating the effect on WIP is more challenging, as there was no model developed to validate the WIP results of new ASAP models. We estimate that developing and validating such a model requires approximately 40 hours, though it can be reused for future adjustments.

Implementation risk

In the event that the modifications are improperly implemented, or if stakeholders misinterpret the results due to the prior model's over-forecasting, this could lead to misallocation of resources and project delays. Although unlikely, this risk must be considered. To mitigate this risk, approximately 25 hours will be needed, divided among multiple stakeholders, to ensure everyone is informed of the changes.

The total estimated short-term costs for developing, testing and validating the model are approximately 165 units.

9.1.2. Short-term benefits

While the modifications do not unlock the full potential of ASAP, they offer several other immediate benefits. These are:

Improved staffing decisions

Firstly, the modified model is much more accurate in showing whether targets are achieved. A better understanding of targets allows NXP to reduce the likelihood of overstaffing and understaffing. Hiring a full-time flexible operator from Randstad currently costs NXP around 1500 units per year. While the exact financial benefit is uncertain without full implementation, the improved accuracy of the model is projected to save NXP the costs of hiring a full-time flexible worker for six months, resulting in annual savings of approximately 750 units.

Reduction in maintenance workload

Secondly, up to 9 hours of maintenance are required weekly for ASAP, with most of this time spent investigating unexpected results from the model. A more reliable model is expected to reduce these issues, potentially saving about 1 hour of maintenance time per week, or 52 units annually.

Improvement in tool maintenance planning

ASAP is currently used to plan preventive maintenance of tools. As mentioned in subsection 4.1, availability efficiency is the third-largest inefficiency at ICN8, suggesting that better planning could help reduce this loss. While machines can always break down and require preventive maintenance, the loss cannot be reduced to 0%. Although the exact impact ASAP will have on the availability efficiency through better maintenance planning is unknown, it is expected to lead to improvements.

The total short-term value is estimated at approximately 802 units per year, derived from improved resource allocation and reduced maintenance time.

9.2. Long-term costs and benefits

While the expected short term benefits are significant, the real value of ASAP lies in its ability to pinpoint future bottlenecks. To achieve this, the model must forecast absolute moves with an average deviation of less than 5%. This thesis showed that the modifications reduced the average deviation from 19.1% to 11.7%.

To estimate the costs necessary to develop a model capable of achieving the 5% deviation, we assume that the costs to improve the model in terms of percentage reduction in move deviation will be similar to those outlined in this thesis. Our analysis focuses on the 2024 results, as the data necessary to reproduce the 2023 samples using a modified model have been deleted and cannot be recovered. Additionally, the 2024 results are worse than those of the 2023 model, providing a more conservative estimate.

9.2.1. Long-term costs

The long term costs will be incurred as NXP continues to refine the model to improve its accuracy from 11.7% to the target of 5% to forecast bottlenecks. If students will take over the project in the future, we assume that the costs are approximately 165 units to achieve a similar percentage improvement.

$$\frac{Modified\ model\ deviation - Wanted\ deviation}{Modified\ model\ deviation} = \frac{0.117 - 0.05}{0.117} = 57.26\%$$

(16)

To achieve a 5% forecast target, a new better model must limit its move deviation by another 57.26 ,as can be seen in equation (16). Based on past performance, the modifications have improved the model's ability to forecast moves in the 35 largest cap groups from 19.1% to 11.7%, which is an improvement of 38.72%, see equation (17).

$$\frac{Old\ model\ deviation - New\ model\ deviation}{Old\ model\ deviation} = \frac{0.191 - 0.117}{0.191} = 38.72\%$$

(17)

Assuming 165 units spent on research leads to a 38.72% reduction in forecasted move deviation, the costs to achieve a model that forecasts with a 5% deviation would require an additional 286.2 units, see equation (18).

$$Costs\ developing\ model\ with\ 5\%\ deviation = 165\ units * \frac{\log(0.4274)}{\log(0.6126)} = 286.2\ units$$

(18)

9.2.2. Long-term benefits

The primary long-time value, as was stated in subsection 4.2, lies in improving the operation rate and availability rate. Although the exact efficiency improvement requires validation through testing, a 1% increase in overall fab efficiency of ICN8 is anticipated based on the assumption in the 2016 business case. This anticipated efficiency improvement could either increase production capacity or reduce operating costs.

Pinpointing bottlenecks

The most significant potential benefit of ASAP is its ability to pinpoint future bottlenecks. For the stakeholders to trust the ASAP, the model must forecast moves within a 72-hour window with 5% daily deviation per cap group. If this is achieved, a 1% increase in factory efficiency is expected to yield large financial benefits, as outlined in the 2016 business case. Specifically, when demand exceeds production, this efficiency would lead to an annual 46667 units increase in revenue in 2016. Adjusting for the 29.7%

inflation rate from 2016 to 2024, this figure rises to 60505 units as can be seen in equation (19), leading to a profit increase of 45000 units (*Inflation Since 2016*, 2024).

$$Inflation\ adjusted\ revenue = Revenue * (1 + inflation\ rate) = 46667 \text{ units} * (1 + 0.297) = 60505 \text{ units} \tag{19}$$

In the period when production capacity exceeds demand, a 1% increase in efficiency is expected to reduce the need for operators by 1%. With around 200 operators working at NXP Nijmegen, this reduction could lead to annual cost savings of around 3,000 units.

Forecasting lot

Furthermore, ASAP has the potential to forecast when the lots are ready. Currently, when a lot's timing is off, it must be communicated with the business lines. Stakeholders that configure the lots, used ASAP in the past, but due to its bad performance, the stakeholders started to rely on their own models. Stakeholders report they spent between 2 to 4 hours per week reconfirming lots with business lines. An accurate model could decrease this time and improve customer satisfaction.

Setting targets

Lastly, ASAP as well as other simulation programs can provide the CFO with information about the fabs. These forecasts help set production targets across all fabs, ensuring overall demand is met efficiently. Better target-setting improves the likelihood that goals are met efficiently. Additionally, the CFO communicates the data with shareholders. transparent data can improve shareholder confidence, potentially boosting NXP's stock price.

In summary, the value of ASAP varies significantly depending on whether demand exceeds production. Given the unpredictability of semiconductor demand, we assume that demand exceeds production 20% of the time. Excluding other potential benefits of ASAP, pinpointing bottlenecks generates an income of 11400 units, see figure (20).

$$Value\ ASAP = 0.2 * 45000\ units + 0.8 * 3000\ units = 11400\ units \tag{20}$$

To illustrate how ASAP is used by all stakeholders, Figure 27 has been created to clearly show all the connections:

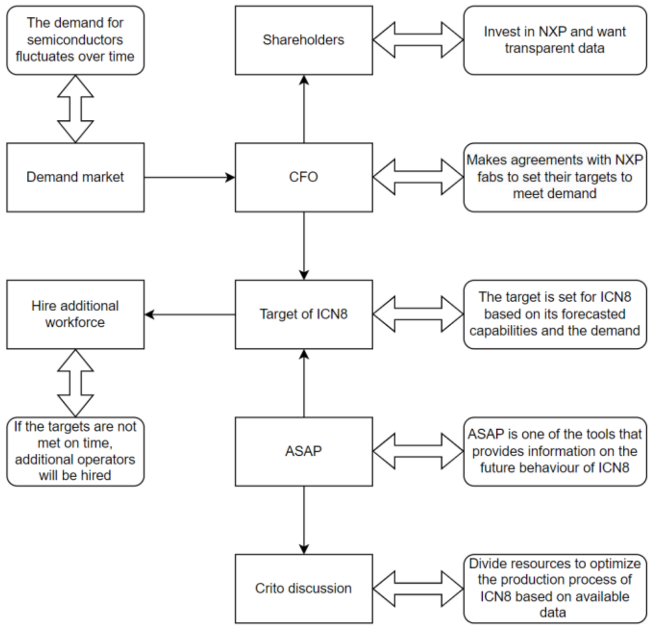

Figure 27 Utilisation of ASAP.

9.3. Return on investment of ASAP

To justify the investments made in ASAP during this thesis and to support further research, the incremental return on investment (ROI) for both short-term and long-term will be analysed. The incremental ROI calculations are based on the additional investments made to improve the model compared to the incremental benefits gained.

9.3.1. ROI framework

The return on investment represents the financial benefit that will be obtained within a year of investment. Equation (21) is used to calculate the ROI:

$$ROI = \frac{(Total\ benefits - Total\ costs)}{Total\ costs} \tag{21}$$

Since the costs for maintaining ASAP and the licensing fees will be incurred regardless of the outcome of the thesis, the focus is set on the additional incurred costs and gained benefits for modifying ASAP. Therefore, equation (22) calculates the incremental ROI:

$$Incremental\ ROI = \frac{(Total\ incremental\ benefits - Total\ incremental\ costs)}{Total\ incremental\ costs} \tag{22}$$

9.3.2. Short-term incremental ROI

The short-term incremental ROI is based on the costs and expected benefits that will be obtained by the new modification of the model developed during this thesis. According to subsection 9.1.1, the costs made to develop, implement and test modifications are 165 units. The expected annual benefits are estimated at 802 units.

$$Incremental\ ROI_{short-term} = \frac{802 - 165}{165} * 100\% = 386.06\%$$

(23)

The short-term incremental ROI of 386.06%, calculated in equation (23), demonstrates that the modifications to ASAP deliver a significant return on investment. The costs incurred during the thesis is justified by the benefits, which far exceed the costs.

9.3.3. Long-term incremental ROI

The long-term incremental ROI is based on the expected costs and expected benefits that will be obtained by further modifying the mode until it is accurate enough to pinpoint bottlenecks. Based on the figures in subsection 9.2, the expected costs for further improvement is 286.2 units, while the expected annual benefit is 11400 units.

$$\text{Incremental } ROI_{long-term} = \frac{11400 - 286.2}{286.2} * 100\% = 3883.3\%$$

(24)

The long-term incremental ROI is forecasted at 3864%, calculated in equation (24), highlighting the significant advantage of further enhancing ASAP. Therefore, future investment in improving ASAP is strongly recommended.

While both incremental ROI results are abnormally large, two factors must be considered. First, the costs associated with improving the model are minimal, with human resource being the only significant expenses. Second, ASAP has been under development since 2016, during which NXP has invested substantial funds in licensing, maintenance, and developing ASAP. Despite these investments, the model has shown a negative ROI over the years. When accounting for these costs, it may take over a year for ASAP to deliver a positive ROI once it is able to pinpoint bottlenecks effectively.

10. Discussion

We made multiple assumptions and choices to achieve the goal of modifying the entire ASAP model of NXP. Although the results are reliable, additional research into other KPIs and modifications could further enhance the model. In this section, we describe the limitations and potential future improvements.

10.1. Limitation

Modifying and validating a complex model like ASAP, which has been developed over the past eight years, presents significant challenges. Although the modifications resulted in improvements, certain assumptions and constraints may have influenced our findings. The following limitations are important to consider when interpreting the results.

10.1.1. Using moves per cap groups as a key KPI

We have validated the effectiveness of the ASAP model by comparing actual and simulated moves to calculate the daily average deviation in moves per cap group. While the move metric is the most important KPI at NXP, moves are not directly used to pinpoint bottlenecks. Instead, work-in-progress (WIP) is used to identify bottlenecks (Kusters, 2021). However, both metrics are strongly correlated. Therefore, we assume that improvement in forecasting moves lead to improvements in forecasting WIP.

10.1.2. Validating methods for the model

We validated the modified model by comparing the performance difference between the modified and older models, as well as comparing the modified model with the actual moves. We validated the accuracy of the model by using the number of deviated moves and the median absolute percentage error (MAPE) per cap group for validation. While no validation method is flawless, combining both MAPE and the number of deviated moves provided a more reliable understanding of the model's accuracy.

The number of deviated moves provides limited insight into the individual cap groups or the exact accuracy of the model. Nonetheless, it is useful for comparing the forecasting accuracy between the older and newer models. Therefore, we assume that a lower total daily average of deviated moves per cap group indicates a better-performing model.

The MAPE calculation can exaggerate large deviations, which skew the average. Since the model does not account for fab stops and voltage dips, some samples are significantly off, causing the average MAPE to be inaccurate. To mitigate the negative impact of the MAPE method, we excluded the worst 5% of samples and only used the 35 largest cap groups, representing 80% of moves in 2023, to calculate the accuracy at the cap group level. Excluding these extremes is essential for calculating the accuracy of the entire fab using MAPE calculations, as including them would inflate the fab deviation to an average of 843% in 2023, which is not representative. By excluding these data, the average fab deviation decreased to 15.1%, which is close to the median of 14.9%, making the results more reliable. Thus, we assumed that excluding these data still yield valid findings.

10.1.3. Sample period

Data needed to generate the process files were not saved previously. This prevented the research from simulating the impact of modifications over different time spans and only allowed the generation of samples from a specific period. As the fab continuously changes, ASAP can simulate the fab more effectively during some periods than others. Although the 2024 samples are harder for ASAP to simulate compared to 2023 samples, we assume that the modifications equally benefit the model in all periods.

10.1.4. Assumption tool process times file

The process file does not account for all historical process times. While we generate most process times through the same method, the real data often contain exceptions that require further investigation before being included into the process file. These exceptions include variation in how we calculate process time (which must be converted to time per lot), forecasted process times for processes that have not run in a long time, and incomplete data, especially when the product recipe is missing. Around 10 percent of all process times contain exceptions. Therefore, we assume that excluding these exceptions and using 90 percent of all historical process times is sufficient to create a reliable process file.

10.1.5. Idle time file

Implementing idle time is challenging because it depends on human behaviour. While idle time improves the model, we implemented it as a temporary solution to account for human behaviour. We calculated idle time using the upper 20th percentile of the previous 28 days of idle times, where the machine's availability exceeds 50 percent. If this result exceeds 20%, we set the forecasted idle time to 20%. While this approach is not perfect, it prevents idle time from being exaggerated, especially since the behaviour of a tool can change due to breakdowns/repairs, product mix and operator availability, which fluctuate daily. These changes reduce the effectiveness of using tool idle time over longer periods, as the implemented idle time is a fixed value. This could cause additional bottlenecks if the importance of a tool varies over time. However, Since this thesis focuses on the first 72 hours, this issue does not pose concern. Therefore, we assume that these methods for calculating the idle time reflect reality.

10.1.6. Financial benefit of ASAP

Lastly, all calculations we made to forecast the financial benefit of ASAP are based on assumptions made by several stakeholders of NXP. The real financial benefit of ASAP lies in the stakeholders' ability to better allocate resources. Therefore, the true value of ASAP can only be measured once the modified ASAP model is implemented and actively used.

10.2. Future research

The modified models demonstrate substantial improvements compared to the older model. While these improvements are substantial, they remain insufficient for pinpointing bottlenecks. For future improvement, we have identified several areas that require additional research.

10.2.1. Refining the modifications

Both the process file and the idle time files show improvement despite not being perfect. The process file does not include all data, as exceptions are left out and instead the cap group time is used. The idle time file, on the other hand, is based on an up-to-date approximation of historical idle times, which could be improved per cap group in the short time and replaced with operator files in the long term.

10.2.2. Implementing operator file

The idle time file we have currently implemented in ASAP is a relatively simple file for incorporating operator behaviour. We assume that the idle time file is effective only in the short term and serves as an approximation for human behaviour. To properly include operators, we need to integrate a so called operator file, which need detailed information about operators such as the number of available operators, the cap group placement of each operator, the speed of each operator, the types of processes each operator can execute and the probability and length of illness. Collecting these data presents significant challenges because it involves gathering large amounts of information about human behaviour, which are difficult to obtain, and not everyone is willing to be monitored. Given the extent of improvement due to including idle times, it seems likely that incorporating the operator file would significantly improve the model, especially for longer runs. Therefore, further research on the operator file is strongly recommended.

10.2.3. Modify tool placement file

The model does not currently account for transport time between tools and instead uses the historical average transport time between areas. This approach lead to small over- or under-estimations of transport times for each process. While we expect this adjustment to have a limited impact on the model's accuracy, implementing more precise transport times is feasible within a relatively short time frame and is therefore worth considering.

10.2.4. Implementing chambers into ASAP

The model of NXP does not include chamber breakdowns. Many machines used in ICN8 consist of multiple process steps, often with multiple chambers per process step. Although some chambers have different functions within a process step, the machine can continue to produce wafers at a reduced capacity if a chamber fails in a multi-chamber process step. ASAP, does not account for this and excludes the entire tool if any chamber goes down. The assumption of NXP does not reflect reality and negatively impacts the simulation results.

10.2.5. Simulation over longer time periods

Our research was focused on improving the ASAP results for the first 72 hours, as this period is crucial for the CRITO discussion. However, it is also valuable for NXP to know how the factory would behave over a longer period. Therefore, future research should aim to validate the impact of the modifications over a longer simulation period.

10.2.6. Simulation impact on WIP

We primarily validated the model based on how accurately ASAP can forecast moves per cap group. However, the eventual WIP per cap group will be used for pinpointing bottlenecks. While move forecasting is strongly correlated with WIP forecasting, additional research is needed to assess the extent of the improvement in WIP to further validate the improvements in ASAP.

11. Conclusion

We investigated how the 72-hour ASAP model could be modified to improve its forecasting accuracy for individual cap groups. Since NXP acquired ASAP in 2016, stakeholders have reported that the model has been unable to identify bottlenecks, as it fails to meet the required average absolute deviation in moves of no more than 5 percent per cap group. Consequently, we focused on the following research question:

How should the ASAP model be modified to improve 72-hour simulations based on daily moves per cap group?

Our initial research evaluated both ICN8's and ASAP's status. We identified that the largest inefficiency in ICN8's production lies in operation rate, with availability rate being the third-largest inefficiency. Both of these areas could be improved by modifying ASAP. We found that ASAP's historical forecasts failed to meet the 5 percent daily move deviation per cap group, with 2023 samples showing an average deviation of 15.8 percent per cap group.

After further research, we discovered that the model lacked detailed process times and did not account for operator influences. It relied on the average process time per cap group and assumed that lots did not experience any transport delays, instead moving instantaneously once each process was completed. By incorporating these inefficiencies into the older model through the implementation of an additional tool process time file and idle time files, the ASAP model was able to capture the actual performance of ICN8. Our validation of the modified model showed a 39.2 percent reduction in daily move deviation per cap group, significantly improving the model's ability to forecast bottlenecks. The forecasted daily deviation of moves for 2024 improved from 19.1 to 11.7 percent for individual cap groups and from 23.2 to 5 percent for the entire fab.

Despite this significant improvement in move forecasting, the modified model is still unable to simulate at the 5 percent deviation target required to identify bottlenecks. Additional samples and further research on WIP are necessary to validate the model, as the current findings are based on a single month of 2024, which proved more challenging for the model to forecast compared to 2023. Nevertheless, these improvements are expected to assist in optimising staffing and reducing maintenance time, potentially gaining an incremental ROI of 386 percent. If the model continues to improve at this rate and incurs similar costs as outlined in this thesis, the projected incremental ROI for improving the model until it can pinpoint bottlenecks could reach 3883 percent, excluding maintenance costs and losses from the previous years.

In conclusion, although further research is required to achieve NXP's goal of accurately pinpointing bottlenecks, this thesis represents a significant step towards developing an ASAP model capable of identifying future bottlenecks in ICN8's production process.

Reference

Applied Materials. (2007). *AutoSched AP Beginning Class Notes.* Santa Clara, California: Internal document, Applied Materials. Retrieved June 10, 2024

Bettayeb, B., Bassetto, S., & Sahnoun, M. (2014). Quality control planning to prevent excessive scrap production. *Journal of Manufacturing Systems*, 400-411.

Creswell, J. W. (2014). *Research Design Qualitative, Quantitative, and Mixed Methods Approaches.* London: Sage.

Ferko, R., & Znidarsic, A. (2007). Using OEE approach for improving manufacturing performance. *Informacije*, 105-111.

Geng, H., & Zhou, L. (2004). *How semiconductor chips are made.* Oregon & Palo Alto: Intel Corporation & Hewlett-Packard Company. Retrieved from https://studylib.net/doc/18073905/how-semiconductor-chips-are-made

Goodwin, P. (1999). On the asymmetry of the symmetric MAPE. *International Journal of Forecasting*, 405-408.

Heukelom, F. v. (2020). *OEE definitions ICN8.* Nijmegen: Internal report, NXP. Retrieved August 28, 2024

Hunter, J., Delp, D., Collins, D., & Si, J. (2002). Understanding a semiconductor process using a full-scale model. *IEEE Transactions on Semiconductor Manufacturing*, 285-289.

Intel Corperation. (2011). *From sand to silicon "Making of a chip" illustrations.* Intel corperation. Retrieved from https://download.intel.com/newsroom/kits/chipmaking/pdfs/Sand-to-Silicon_32nm-Version.pdf

Khair, U. (2017). Forecasting error calculation with mean absolute deviation and mean absolute percentage error. *Journal of Physics: Conference Series*, 1-6.

Kiba, J.-E., Lamiable, G., Dauzére-Pérès, S., & Yugma, C. (2009). Simulation of a full 300mm semiconductor manufacturing plant with material handling constraints. *STMicroelectronics*, 1601-1609.

Klein, M., & Kalir, A. (2006). A full factory transient simulation model for the analysis of expected performance in a transition period. *Proceedings of the Winter Simulation Conference* (pp. 1836-1839). Monterey: IEEE.

Kusters, J. (2021). *ASAP ten behoeve van CriTo processen optimaliseren.* Nijmegen: Internal document, NXP. Retrieved May 16, 2024

Miller, D. J. (1990). Simulation of a semiconductor manufacturing line. *Communications of the ACM*, 98-108. Retrieved from Simulation of a semiconductor manufacturing line (acm.org)

NXP. (2023). *NXP corporate overview*. Retrieved from https://www.nxp.com/docs/en/supporting-information/NXP-CORPORATE-OVERVIEW

Ping, B. G., Liow, L. F., Gupta, A. K., Lendermann, P., Turner, S. J., & Wang, X. (2007). Analysis of a borderless fab using interoperating. *International Journal of Production Research*, 675-698.

Rahim, S., Ahmad, I. C., & Hashim, a. (2013). *Development of Manufacturing Simulation Model of Semiconductor Fabrication.* World Academy of Science.

Ramirez-Hernandez, J. A., Li, H., Fernandez, E., McLean, C. R., & Leong, S. K. (2005). A framework for standard modular simulation in semiconductor wafer fabrication systems. *Proceedings of the Winter Simulation Conference* (pp. 1-17). Orlando: NIST. Retrieved from https://www.nist.gov/publications/framework-standard-modular-simulation-application-semiconductor-wafer-fabrication

Savings.org. (2024, September 20). *Inflation since 2016*. (Saving) Retrieved from https://www.saving.org/inflation/by-year/2016#:~:text=Inflation%20Since%202016%20%241%20in,2016%20equals%20%241.30%20in%202024.

Seon, K. (2017). *Introduction to Semiconductor Manufacturing and FA Process.* Infineon Technologies. Retrieved from https://www.nexty-ele.com/nat/wp-content/uploads/sites/3/2017/10/6.-Manufacturing-process-of-seminconductor-and-course-of-failure-analysis.pdf

Taylor, S. J. (2005). Interoperating AutoSched AP using the high level architecture. *Proceedings of the winter simulation conference* (pp. 394-401). Orlando: IEEE.

Tien, V. S., Teck, Y. E., Govindasamy, G. D., & Kamaruddin, S. (2006). Application of Autosched AP Simulation Model in Wafer Fab. *International Conference on Semiconductor Electronics* (pp. 846-850). Kuala Lumpur: IEEE.

Yon-Chun, C., & L. Hsuan, H. (2000). A Methodology for Product Mix Planning in Semiconductor Foundry Manufacturing. *IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING*, 278-285.

Appendix A- Working ASAP

The ASAP model is a complex program with numerous functions and know-how. Here, the workings of ASAP will be described in more detail. Writing the whole ASAP model itself is unnecessary, the detail relevant to replicating the research will be highlighted.

A.1. File description ASAP

ASAP operates with a large number of input and output files. While the most important files are described in this thesis, not all are covered in detail. The other important input and output files are stated in Table 7 and Table 8.

Table 7 ASAP input file descriptions.

Input files	Description
Option.rdf	The "option.rdf" file is the main file of ASAP. Here, input data can be assigned to particular functions. Furthermore, it can be specified what output data must be written in which text files.
Period.txt	The period.txt file contains information on the period during which the simulation must run.
Order.txt	The order.txt file is needed to let the model know what types of semiconductors must be produced. Here, the number of semiconductor types, the start date, and the due date can be set.
Part.txt	The part.txt file shows what types of processes are required to produce a type of process. Here multiple process routes can be assigned for producing a type of semiconductor, including the reworking process of an IC wafer, in case any mistakes occur during production.
Route.txt	The route.txt file shows what processes are required to produce a type of IC wafer. Here, cap groups that can execute each process are assigned to every process step in combination with the processing time that is needed to execute the process. Furthermore, depending on the process, certifications are stated which indicate whether a machine in a cap group can execute the particular process step.
Stn.txt	The stn.txt file describes what tools are assigned to each cap group. Although the cap groups consist of machines with similar functionalities, most machines still differ from each other and can execute alternative processes. Therefore, the number of chambers per machine and the exact processes a machine can execute are assigned to every machine in the station file.
Cal_reducecapacity.txt	The reducecapacity.txt file contain information about the length of idle time per call.
Reducecapacity_attach.txt	The reducecapacity_attach.txt file connects the reduce capacity calls to the right tool
Wip.txt	The WIP.txt file shows the initial WIP per cap group is stated
Down_attach.txt	The down_attach,txt file contains information about when machines are or are expected to be down.
Fromto.txt	The fromto.txt file shows the distance in walking time between two points. Currently, the time per area is implemented.
Process.txt	The process.txt file contains information about tool process times

Table 8 ASAP output file description.

Output files	Description

Capgroup.rep	Shows the number of executed moves per day per cap group
Stn.rep	Shows the number of moves made per tool
Day_lOTS_OUT.rep	Shows the number of finished lots per day
DAY_PCS_OUT.rep	Shows the number of finished wafers per day
DAY_WIPLOTCUR.rep	Shows the total work in progress lots per day
Lot.rep	Shows the status of all lots in ICN8 per day

A.2. Gantt viewer Legenda

Below are the legend and the description displayed of the Gantt viewer. The legend description is directly sourced from the Applied materials helpdesk (*AutoSched AP Beginning Class Notes*, 2007).

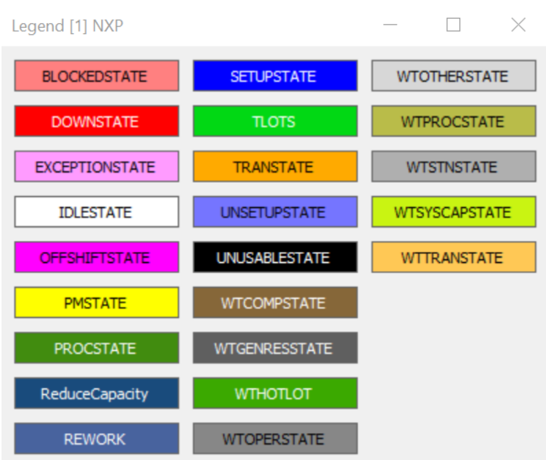

Figure 28 Gantt viewer legend.

Table 9 Legenda Gantt viewer.

Event	Description
BLOCKEDSTATE	Blocked from processing due to either a certification change or down event.
DOWNSTATE	Unavailable due to a down unavailable definition.
EXCEPTIONSTATE	Unavailable due to an exception unavailable definition.
IDLESTATE	Idle.
NOSTATE	In no state.
OFFSHIFTSTATE	Unavailable due to being off-shift.
PMSTATE	Performing a PM (Preventive Maintenance) work order.
PROCSTATE	Processing.
SETUPSTATE	Setup.
TRANSTATE	Traveling from one location to another.
UNSETUPSTATE	Unsetup.
UNUSABLESTATE	Unusable due to a resource 'QTY' change in time-phased-parameters.
WTCOMPSTATE	Waiting for components (sub-parts).
WTOPERSTATE	Waiting for an operator.
WTOTHERSTATE	Unavailable because another claimed resource is unavailable.
WTPROCSTATE	Waiting to process either in cascade mode or no internal capacity available.
WTSTNSTATE	Waiting for a station.
WTSYSCAPSTATE	Waiting for system travel capacity.
WTTOOLSTATE	Waiting for a tool.
WTTRANSTATE	Waiting for transportation.

Appendix B – Calculation accuracy ASAP

Previously, there was no way to calculate how accurately ASAP can forecast moves. Therefore, a model has been developed that compares the daily simulated moves with the real executed moves per cap group. For this, the correct real moves and ASAP moves are extracted, sorted and then calculations are performed to determine the accuracy.

B.1. Real move counter

The first step is to extract and count the daily real moves executed per tool. Each day, NXP generates a file that lists all process steps, monitor steps, and rework steps. A custom developed code is used to count all process moves per machine, excluding the monitor and rework steps. In ASAP, these steps are represented as additional idle time rather than actual steps. The code generates data for three days, worth of daily executed moves starting from the desired start date.

	CapGroup	Moves
day 1	[redacted]	0
day 1		3
day 1		44
day 1		260
day 1		63
day 1		33
day 1		85
day 1		57
day 1		157
day 1		13
day 1		4
day 1		3
day 1		21
day 1		1
day 1		132
day 1		12
day 1		48

Figure 29 Counted real moves per cap group per day.

B.2. ASAP and real move sorter

When ASAP is run, it generates an output file named "capgroup.rep", which the daily moves per cap group for the next 14 days are stated. First these moves and the moves of the real moves are clearly ordered (Figure 30).

	CapGroup	[redacted]						
20240725	day 1	1	1	43	285	75	38	65
	day 2	3	8	38	276	83	41	56
	day 3	0	2	54	291	80	35	65
	day 4	4	4	46	271	73	44	63
	day 5	2	5	46	258	77	47	60
	day 6	0	5	64	231	76	40	78
	day 7	1	7	62	261	61	38	65
	day 8	3	7	49	251	88	50	65
	day 9	1	1	36	281	67	33	81
	day 10	2	3	44	303	79	46	65
	day 11	1	2	45	259	69	46	80
	day 12	0	2	44	251	59	28	78
	day 13	2	2	77	211	83	25	61
	day 14	4	0	62	249	58	20	68
20240726	day 1	3	8	38	276	83	41	56
	day 2	0	2	54	291	80	35	65
	day 3	4	4	46	271	73	44	63

Figure 30 Ordered ASAP moves.

Next, because only the first three first days are necessary, the additional days (day 4 to 14) are removed.

	CapGroup	[redacted]					
20240725	day 1	0	7	55	276	93	12
	day 2	0	3	51	221	80	17
	day 3	6	0	50	247	64	36
20240726	day 1	3	10	64	213	118	26
	day 2	0	2	51	283	59	11
	day 3	2	7	50	238	60	46
20240727	day 1	0	7	55	252	125	23
	day 2	0	3	51	227	62	7
	day 3	5	4	51	198	33	47
20240728	day 1	1	11	54	238	100	36
	day 2	4	4	51	228	66	19
	day 3	1	0	51	219	28	23
20240729	day 1	1	13	56	200	109	38
	day 2	2	4	52	257	53	10
	day 3	0	0	52	234	36	43
20240730	day 1	1	13	39	233	113	28
	day 2	0	2	46	200	52	9

Figure 31 Ordered first 3 days ASAP.

The real moves are sorted in the same manner as the ASAP samples.

B.3. ASAP and real move sorter

Eventually, the accuracy of the model in daily moves for both the entire fab and individual cap groups can be calculated. First, the number of daily deviated moves per cap group is calculated using the formula:

$$number\ of\ moves\ deviated = ASAP\ moves - real\ moves \tag{25}$$

	CapGroup	[redacted]									
20240725	day 1	-1	6	12	-9	18	-26	48	6	17	0
	day 2	-3	-5	13	-55	-3	-24	15	8	32	1
	day 3	6	-2	-4	-44	-16	1	9	1	45	9
20240726	day 1	0	2	26	-63	35	-15	49	12	24	2
	day 2	0	0	-3	-8	-21	-24	4	-6	46	7
	day 3	-2	3	4	-33	-13	2	16	-3	32	-4
20240727	day 1	0	5	1	-39	45	-12	57	12	33	8
	day 2	-4	-1	5	-44	-11	-37	25	-4	34	2
	day 3	3	-1	5	-60	-44	0	21	-2	7	3
20240728	day 1	-3	7	8	-33	27	-8	52	-5	27	5
	day 2	2	-1	5	-30	-11	-28	19	9	25	6
	day 3	1	-5	-13	-12	-48	-17	6	2	11	7
20240729	day 1	-1	8	10	-58	32	-9	42	9	24	9
	day 2	2	-1	-12	26	-23	-30	13	0	27	2
	day 3	-1	-7	-10	-27	-25	5	15	17	40	-7

Figure 32 deviated moves per cap group per day.

A positive value indicates that ASAP overpredicted the number of moves made in a cap group whereas a negative value indicates that ASAP underpredicted the number of moves made per cap group. In addition, the absolute deviation is calculated using the following formula:

$$abs\ number\ of\ moves\ deviated = |ASAP\ moves - real\ moves| \tag{26}$$

	CapGroup	[redacted]										
20240725	day 1	1	6	12	9	18	26	48	6	17	0	2
	day 2	3	5	13	55	3	24	15	8	32	1	2
	day 3	6	2	4	44	16	1	9	1	45	9	2
20240726	day 1	0	2	26	63	35	15	49	12	24	2	5
	day 2	0	0	3	8	21	24	4	6	46	7	2
	day 3	2	3	4	33	13	2	16	3	32	4	2
20240727	day 1	0	5	1	39	45	12	57	12	33	8	2
	day 2	4	1	5	44	11	37	25	4	34	2	0
	day 3	3	1	5	60	44	0	21	2	7	3	0
20240728	day 1	3	7	8	33	27	8	52	5	27	5	4
	day 2	2	1	5	30	11	28	19	9	25	6	0
	day 3	1	5	13	12	48	17	6	2	11	7	1
20240729	day 1	1	8	10	58	32	9	42	9	24	9	4
	day 2	2	1	12	26	23	30	13	0	27	2	1
	day 3	1	7	10	27	25	5	15	17	40	7	0
20240730	day 1	1	8	25	2	37	12	53	9	19	4	6
	day 2	1	5	16	61	9	29	17	20	49	8	0
	day 3	3	7	3	35	52	13	11	14	26	2	0
20240731	day 1	1	4	8	7	32	15	53	0	36	7	7

Figure 33 Absolute deviated moves per cap group per day.

Now, all moves and absolute moves are stated per day per cap group. By counting the total number of deviated moves for the 2023 result, the 2024 result, and the modified 2024 result, the accuracy of the entire plant per day can be calculated.

Furthermore, it is desired to identify which cap groups deviate and if the deviation is significant. Some cap groups make a large number of moves whereas other cap groups do not. The larger the number of moves made by the cap groups, the less significant a single deviation becomes. To easily calculate this deviation, the MAPE formula is used to calculate the mean absolute percentage error for each cap group each day, as shown in Figure 34.

$$abs\ MAPE_{d,c} = \frac{|real\ moves_{d,c} - ASAP\ moves_{d,c}|}{real\ moves_{d,c}} * 100\% \tag{27}$$

$$MAPE_{d,c} = \frac{real\ moves_{d,c} - ASAP\ moves_{d,c}}{real\ moves_{d,c}} * 100\% \tag{28}$$

	CapGroup	[redacted]							
20240725	day 1	infinite	86%	22%	3%	19%	217%	42%	13%
	day 2	infinite	167%	25%	25%	4%	141%	21%	24%
	day 3	100%	infinite	8%	18%	25%	3%	12%	6%
20240726	day 1	0%	20%	41%	30%	30%	58%	47%	32%
	day 2	0%	0%	6%	3%	36%	218%	6%	60%
	day 3	100%	43%	8%	14%	22%	4%	20%	10%
20240727	day 1	0%	71%	2%	15%	36%	52%	47%	43%
	day 2	infinite	33%	10%	19%	18%	529%	28%	13%
	day 3	60%	25%	10%	30%	133%	0%	26%	10%
20240728	day 1	300%	64%	15%	14%	27%	22%	45%	17%
	day 2	50%	25%	10%	13%	17%	147%	24%	28%
	day 3	100%	infinite	25%	5%	171%	74%	7%	8%
20240729	day 1	100%	62%	18%	29%	29%	24%	41%	28%
	day 2	100%	25%	23%	10%	43%	300%	14%	0%
	day 3	infinite	infinite	19%	12%	69%	12%	19%	71%
20240730	day 1	100%	62%	64%	1%	33%	43%	40%	29%
	day 2	infinite	250%	35%	31%	17%	322%	21%	74%
	day 3	infinite	infinite	7%	12%	144%	35%	14%	58%
20240731	day 1	50%	36%	15%	3%	34%	65%	45%	0%

Figure 34 MAPE per cap group per day.

Although the MAPE formula effectively calculates the deviation per cap group, some values are extreme or even infinite. This occurs because MAPE cannot handle zeros and can exaggerate the impact of outliers. By excluding the most extreme 5% of events, which are caused by fab stops or voltage dips, and only using the 35 largest cap groups that together account for 80% of the moves made in 2023, an average MAPE based on a weight in moves can be calculated which is close to the median MAPE, demonstrating that the most extreme values are not considered. (Goodwin, 1999)

In addition, a weight is added based on the number of moves made by the cap group in 2023 divided by the total number of moves made by the largest 35 cap groups in 2023. This weight ensures the average deviation is reflects the number of moves made rather than just the deviation per cap group.

Biggest caps		[redacted]											
moves													
Weights		0.13606	0.091894	0.083327	0.071842	0.073294	0.066308	0.044071	0.03174	0.028087	0.02254	0.024557	0.020783
		Total moves biggest caps											
		[redacted]											
all caps		1	2	3	4	5	6	7	8	9	10	11	12
moves		[redacted]											
Weights													
		0.108817	0.073494	0.066643	0.057457	0.058618	0.053032	0.035247	0.025385	0.022463	0.018027	0.01964	0.016622
		total moves											
		[redacted]											

Figure 35 Weights based on moves for both all cap groups and 35 largest cap groups.

By applying these steps for the 2023 results of the old model, the 2024 results of the old model, and the 2024 results of the modified model, the following values can be obtained which can be found in Table 4.

Appendix C – Simple ASAP model

To prove the functionality of both the idle time files and the process times file, a simple ASAP model has been developed of two identical machines in which modifications can be added to clearly highlight the effect. To replicate the simple model made for identifying properties, the development of the simple ASAP model is stated including the validated workings of the process times file and the idle time file, as detailed below.

C.1. Properties of the Simple model

The simple model uses the minimal number of necessary files to be able to successfully run without errors. It is implemented that 2 tools must continuously run to work through an order that is larger than the machines are able to process (10000 lots). To produce a single product, a single step must be executed that can be done on both machines in 12 minutes. The model simulates a duration of a single day and parameters such as walking distance, lot distribution, breakdown, and operators are not taken into account. The following parameters are implemented into the simple model:

Table 10 Parameters simple model.

Number of tools	2
Tool names	LASER_SCRIBE_1, LASER_SCRIBE_2
Number of process types	1
Process type name	Part1
Number of process steps required (route)	1
Process route name	r_1
Process step name	1_scribe1
Order quantity (for Part1)	10000
Number of simulated days	1
Process time per step	12 min for both machine

By using the parameters in Table 10, The following files are developed to generate results of a simple ASAP model:

SIM_START	06/01/16 08:00:00			
EVENT_TRACE_ON(Y/N)	Y			
EVENT_TRACE_ALL(Y/N)	Y			
GANTT_TRACE_ON(Y/N)	Y			
GANTT_TRACE_ALL(Y/N)	Y			
USE_CALENDARS(Y/N)	Y			
USE_PREV_MAIN_FILE(Y/N)	Y			
USE_CMRG_RANDOM_NUMBERS(	Y			
USE_THREADED_READER(Y/N)	Y			
COMMENT_CHARACTER	*			
PRODUCT_FILES	file	part.txt	outfile	part.rep
STATION_FILES	file	stn.txt		
ORDER_FILES	file	order.txt		
GENRES_FILES	none			
PMORDER_FILES	none			
OPERATOR_FILES	none			
BOM_FILES	none			
SETUP_FILES	none			
PROCESS_FILES	none			
ATTACH_FILES	none			
ACTIONLIST_FILES	file	$ASI/lib/stdactl.txt		
CLASSDEF_FILES	file	$ASI/lib/stdclass.txt		
PREEMPTACTIONLIST_FILES	file	$ASI/lib/stdpactl.txt		
EXTERNALDEF_FILES	file	$ASI/lib/stdfunc.txt		
STNREP_FILES	file	stn.rdf	outfile	stn.rep
OPERREP_FILES	none			
GENRESREP_FILES	none			
PARTREP_FILES	none			
LOTREP_FILES	none			
SETREP_FILES	none			
PERFREP_FILES	none			
FAMREP_FILES	none			
STNGRPREP_FILES	none			
SUPERSETREP_FILES	none			
SUBSETREP_FILES	none			
PARTFAMREP_FILES	none			
PARTGRPREP_FILES	none			
OPERCLASSREP_FILES	none			
ORDERREP_FILES	none			
GENRESFAMREP_FILES	none			
ROUTE_FILES	file	route.txt		
CONTREP_FILES	file			
	none			
PERIOD_FILE	file	period.txt		
FIELD_SUB_DELIMIT_CHARACTER	|			

Figure 36 Option.def file standard simple ASAP model.

STNFAM	STN	RULE	TRACE	BATCHMN	BATCHMX	WNLTTM	WNLTU	STNCERT
LASER_SCRIBE	LASER_SCRIBE_1	rule_FIRST_A		1	1			r_1:-1:laser1
LASER_SCRIBE	LASER_SCRIBE_2	rule_FIRST_A		1	1			r_1:-1:laser1

Figure 37 stn.txt file standard simple ASAP model.

PART	ROUTE	ROUTEFILE
part1	r_1	route.txt

Figure 38 part.txt file standard simple ASAP model.

PERIOD	PERIODSTART	REPORT	RESET
06/01/16 08:00:00	06/01/16 08:00:00	yes	yes

Figure 39 Period.txt file standard simple ASAP model.

ROUTE	STEP	DESC	STNFAM	PTIME	PTUNIT:	PTPER	STNCERT	SETUP	REJOIN
r_1	1_scibe1	WAS_SCRIBE	LASER_SCRIBE	12	min	per_batch	r_1:-1:laser1		

Figure 40 Route.txt file standard simple ASAP model.

ORDER	LOT	PART	PIECES	REPEAT	RUNITS	RPT#	START	DUE	TRACE
Order1	L1	part1	48	1	min	10000	06/01/16 08:00:00	06/08/16 17:00:00	gantt

Figure 41 Order.txt file standard simple ASAP model.

Result simple ASAP model

After running the ASAP model, each machine executes 120 moves which is equal to the number of minutes per day 1440 divided by the process time of 12 minutes. Therefore, the simple model behaves exactly as anticipated.

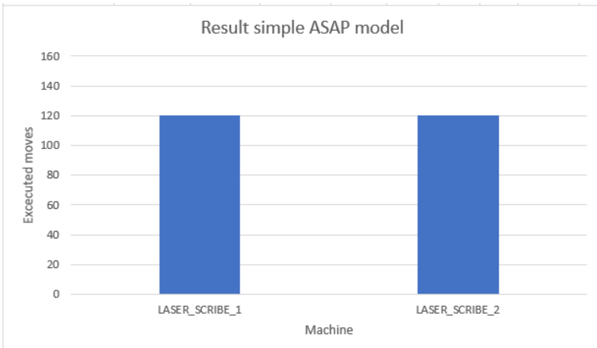

Figure 42 Result standard simple ASAP model.

C.2. Effect process file

Firstly, the effect of implementing tool process times is observed. Here, a file named process.txt is added with new process times per machine, 10 min and 15 min. The data of the process.txt files overrides the data in the route file, which makes that the route file does not have to be adjusted. The following files are adjusted and implemented.

SIM_START	06/01/16 08:00:00			
EVENT_TRACE_ON(Y/N)	Y			
EVENT_TRACE_ALL(Y/N)	Y			
GANTT_TRACE_ON(Y/N)	Y			
GANTT_TRACE_ALL(Y/N)	Y			
USE_CALENDARS(Y/N)	Y			
USE_PREV_MAIN_FILE(Y/N)	Y			
USE_CMRG_RANDOM_NUMBERS(	Y			
USE_THREADED_READER(Y/N)	Y			
COMMENT_CHARACTER	'			
PRODUCT_FILES	file	part.txt	outfile	part.rep
STATION_FILES	file	stn.txt		
ORDER_FILES	file	order.txt		
GENRES_FILES	none			
PMORDER_FILES	none			
OPERATOR_FILES	none			
BOM_FILES	none			
SETUP_FILES	none			
PROCESS_FILES	file	process.txt		
ATTACH_FILES	none			
ACTIONLIST_FILES	file	$ASI/lib/stdactl.txt		
CLASSDEF_FILES	file	$ASI/lib/stdclass.txt		
PREEMPTACTIONLIST_FILES	file	$ASI/lib/stdpactl.txt		
EXTERNALDEF_FILES	file	$ASI/lib/stdfunc.txt		
STNREP_FILES	file	stn.rdf	outfile	stn.rep
OPERREP_FILES	none			
GENRESREP_FILES	none			
PARTREP_FILES	none			
LOTREP_FILES	none			
SETREP_FILES	none			
PERFREP_FILES	none			
FAMREP_FILES	none			
STNGRPREP_FILES	none			
SUPERSETREP_FILES	none			
SUBSETREP_FILES	none			
PARTFAMREP_FILES	none			
PARTGRPREP_FILES	none			
OPERCLASSREP_FILES	none			
ORDERREP_FILES	none			
GENRESFAMREP_FILES	none			
ROUTE_FILES	file	route.txt		
CONTREP_FILES	file			
	none			
PERIOD_FILE	file	period.txt		

Figure 43 Option.def file standard simple ASAP model including process.txt.

STEP	STN	PTIME	PTUNITS
r_1:1_scibe1	LASER_SCRIBE_1	10	min
r_1:1_scibe1	LASER_SCRIBE_2	15	min

Figure 44 process.txt file simple ASAP model.

Result simple ASAP model with process.txt

By adjusting the option.rdf file and including the process.txt file, the new process times per tool override the previous cap group process time of 12 minutes. Now it can be observed in Figure 45, that the LASER_SCRIBE_1 machine executed 144 moves or the number of minutes per day divided by 10 and the LASER_SCRIBE_2 machine executed 96 moves or the number of minutes per day divided by 15. This simple model proves the anticipated workings of the process.txt file.

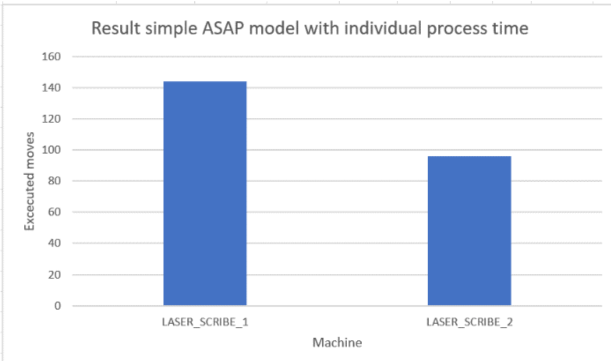

Figure 45 Result simple ASAP model including process.txt.

C.3. Effect idle time

Secondly, the effect of implementing idle times is observed. Implementing idle time must be done by adding two files namely the cal_reducecapacity.txt file and the reducecapacity_attach.txt file. The data of the length of idle time per hour is implemented into the cal_reducecapacity.txt file, where the called Reducecapacity_LASER_SCRIBE_1 is set to have no idle time and an idle time of 50% is assigned to Reducecapacity_LASER_SCRIBE_2. The reducecapacity_attach.txt on the other hand connects the reduced capacity to the right machines. The following files are adjusted and implemented.

SIM_START	06/01/16 08:00:00			
EVENT_TRACE_ON(Y/N)	Y			
EVENT_TRACE_ALL(Y/N)	Y			
GANTT_TRACE_ON(Y/N)	Y			
GANTT_TRACE_ALL(Y/N)	Y			
USE_CALENDARS(Y/N)	Y			
USE_PREV_MAIN_FILE(Y/N)	Y			
USE_CMRG_RANDOM_NUMBERS(	Y			
USE_THREADED_READER(Y/N)	Y			
COMMENT_CHARACTER	'			
PRODUCT_FILES	file	part.txt	outfile	part.rep
STATION_FILES	file	stn.txt		
ORDER_FILES	file	order.txt		
GENRES_FILES	none			
PMORDER_FILES	none			
OPERATOR_FILES	none			
BOM_FILES	none			
SETUP_FILES	none			
PROCESS_FILES	none			
ATTACH_FILES	file	reducecapacity_attach.txt		
ACTIONLIST_FILES	file	$ASI/lib/stdactl.txt		
CLASSDEF_FILES	file	$ASI/lib/stdclass.txt		
PREEMPTACTIONLIST_FILES	file	$ASI/lib/stdpactl.txt		
EXTERNALDEF_FILES	file	$ASI/lib/stdfunc.txt		
STNREP_FILES	file	stn.rdf	outfile	stn.rep
OPERREP_FILES	none			
GENRESREP_FILES	none			
PARTREP_FILES	none			
LOTREP_FILES	none			
SETREP_FILES	none			
PERFREP_FILES	none			
FAMREP_FILES	none			
STNGRPREP_FILES	none			
SUPERSETREP_FILES	none			
SUBSETREP_FILES	none			
PARTFAMREP_FILES	none			
PARTGRPREP_FILES	none			
OPERCLASSREP_FILES	none			
ORDERREP_FILES	none			
GENRESFAMREP_FILES	none			
ROUTE_FILES	file	route.txt		
CONTREP_FILES	file			
	none			
PERIOD_FILE	file	period.txt		
FIELD_SUB_DELIMIT_CHARACTER	|			
PMCAL_FILES	file	cal_reducecapacity.txt		

Figure 46 Option.def file standard simple ASAP model including idle time files.

PMCALNAME	PMCALTYPE	TRACE	MTBPMDIST	MTBPM	MTBP	MTTR	MTTRUNITS	PM_ACTLIST
ReduceCapacity_LASER_SCRIBE_1	MTTPM_by_cal	gantt	constant	1	hr	0	hr	ReduceCapacityPMActList
ReduceCapacity_LASER_SCRIBE_2	MTTPM_by_cal	gantt	constant	1	hr	0.5	hr	ReduceCapacityPMActList

Figure 47 cal_reducecapacity.txt.

RESTYPE	RESNAME	CALTYPE	CALNAME	FOADIST	FOA	FOA2	FOAUNITS
stn	LASER_SCRIBE_1	pm	ReduceCapacity_LASER_SCRIBE_1	uniform	0	0	hr
stn	LASER_SCRIBE_2	pm	ReduceCapacity_LASER_SCRIBE_2	uniform	0	0	hr

Figure 48 Reducecapacity_attach.txt.

Result simple ASAP model with idle time files

By modifying the option.rdf file and including the idle time files, an idle time of 0.5 is implemented into the LASER_SCRIBE_2 machine. This idle time ensures that the machine operates for only half of the time. If the total process time extends half of the time in a hour, the idle time will be compensated in the next hour. As can be seen in figures 49 and 50, the LASER_SCRIBE_2 machine completes exactly half the numbers of moves made by the LASER_SCRIBE_1 machine, which has no idle time.

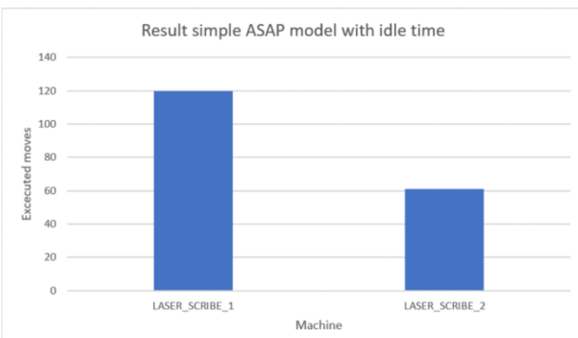

Figure 49 Result simple ASAP model including idle time model.

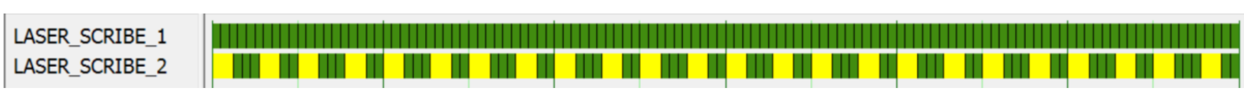

Figure 50 Result GANTT of simple ASAP model including idle time files.

Appendix D – Modification generator description

We generated 30 samples using up-to-date data from the corresponding day. These samples include the process.txt and idle time files. For each sample, the input files process.txt, cal_reducecapacity.txt, and the reducecapacity_attach.txt are included or modified. To generate a single modified ASAP model sample, all possible events must be implemented, requiring the processing of a large quantity of data. Although this process for future samples can eventually be automated in APF, we developed a custom made code to make historical data samples to quantify the validation of the improvement. Below, all steps of the code will be highlighted in more detail.

Extract data

To generate a modified model for a specific day, the right data available on the desired day is extracted. This includes paths for the route file, WIP file, option file, tool process time, stn.txt file, and idle files. These files are essential for modifying the model and must all be extracted on the same day to ensure consistency.

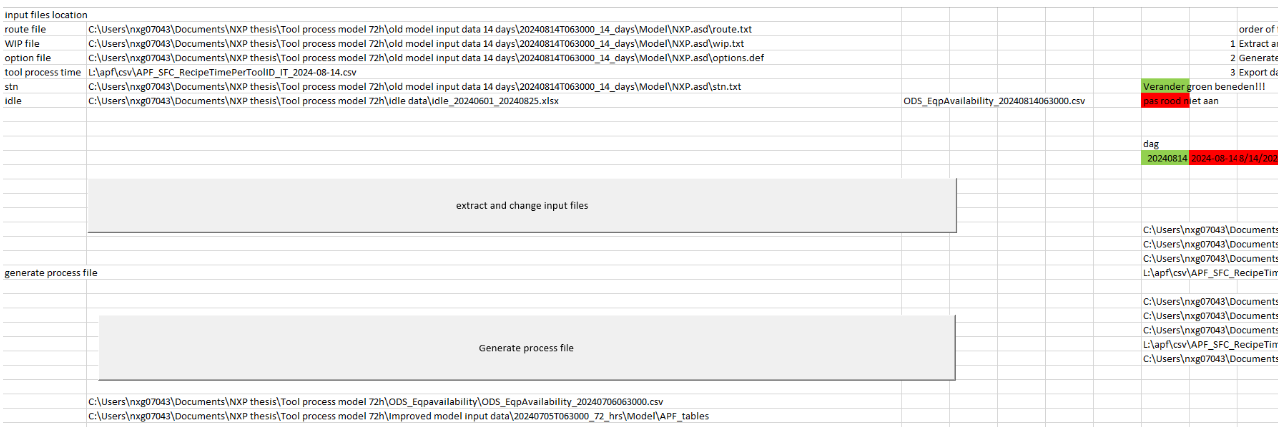

Figure 51 Main excel file for modifying model.

First, data is extracted from the Excel file using the following code:

```
Dim wbI As Workbook
Dim path_old_stn, path_old_route, path_old_wip, path_old_option, path_old_processtime, path_old_idle As String
Dim i As Double
Dim j  As Double
Set wbI = ThisWorkbook
Set oldstndata = wbI.Sheets("stn") 'set right names to right sheet
Set oldroutedata = wbI.Sheets("route;")
Set oldwipdata = wbI.Sheets("wip;")
Set oldoptiondata = wbI.Sheets("option no process")
Set newroutedata = wbI.Sheets("route") '
Set newwipdata = wbI.Sheets("wip")
Set newoptiondata = wbI.Sheets("options")
Set oldprocessdata = wbI.Sheets("process time per tool")
Set oldidledata = wbI.Sheets("idle_data")

Sheets("stn").Cells.Clear 'clears sheets
Sheets("route;").Cells.Clear
Sheets("route").Cells.Clear
Sheets("wip;").Cells.Clear
Sheets("wip").Cells.Clear
Sheets("option no process").Cells.Clear
Sheets("options").Cells.Clear
Sheets("process time per tool").Cells.Clear

path_old_route = Cells(2, 2) 'pinpoint where data files are located
path_old_wip = Cells(3, 2)
path_old_option = Cells(4, 2)
path_old_processtime = Cells(5, 2)
path_old_stn = Cells(6, 2)
path_old_idle = Cells(7, 2)

Set oldstnsheet = Workbooks.Open(path_old_stn)
oldstnsheet.Sheets(1).Cells.Copy oldstndata.Cells
oldstnsheet.Close SaveChanges:=False

Worksheets("stn").Columns("A").Delete
```

Figure 52 Variables names and paths are assigned.

Secondly, the file data is extracted and distributed across multiple excel sheets. Some sheets are used solely for data extraction purposes, while others are modified and later implemented in the model.

```
Set oldroutesheet = Workbooks.Open(path_old_route) 'copy files to sheets
oldroutesheet.Sheets(1).Cells.Copy oldroutedata.Cells
oldroutesheet.Sheets(1).Cells.Copy newroutedata.Cells
oldroutesheet.Close SaveChanges:=False

Set oldwipsheet = Workbooks.Open(path_old_wip)
oldwipsheet.Sheets(1).Cells.Copy oldwipdata.Cells
oldwipsheet.Sheets(1).Cells.Copy newwipdata.Cells
oldwipsheet.Close SaveChanges:=False

Set oldoptionsheet = Workbooks.Open(path_old_option)
oldoptionsheet.Sheets(1).Cells.Copy oldoptiondata.Cells
oldoptionsheet.Sheets(1).Cells.Copy newoptiondata.Cells
oldoptionsheet.Close SaveChanges:=False

Set oldprocesssheet = Workbooks.Open(path_old_processtime)
oldprocesssheet.Sheets(1).Cells.Copy oldprocessdata.Cells
oldprocesssheet.Close SaveChanges:=False

Set oldidlesheet = Workbooks.Open(path_old_idle)
oldidlesheet.Sheets(1).Cells.Copy oldidledata.Cells
oldidlesheet.Close SaveChanges:=False
```

Figure 53 Data is extracted from paths and copied into the sheets.

The idle time file contains a large amount of data. To generate the idle time file, we used the latest 28 days from before the desired simulation day. Data from after cannot be included, as it did not exist at the time, and using it would result in the simulation relying on future data.

```
i = 2
Sheets("Main sheet").Cells(11, 11) = Sheets("Main sheet").Cells(11, 10).Value 'draws date from cell
Do While Sheets("idle_data").Cells(i, 1) <> ""                                 '(the date of the wanted sample must be placed here)
    If Sheets("idle_data").Cells(i, 1) = Sheets("Main sheet").Cells(11, 11) Then
        Do While Sheets("idle_data").Cells(i, 1) <> ""
            If Sheets("idle_data").Cells(i, 1) <> Sheets("Main sheet").Cells(11, 11) Then
                Range(Sheets("idle_data").Cells(i, 1), Sheets("idle_data").Cells(100000, 5)).Clear 'removes data generated
            End If                                                                                  'after normal time the model is generated
            i = i + 1
        Loop
    End If
    i = i + 1
Loop
```

Figure 54 Future idle time data is removed.

```
j = 0

Do While i > 1
    If Sheets("idle_data").Cells(i, 1) <> Sheets("Main sheet").Cells(11, 11) Then
        Do While i > 1
            If Sheets("idle_data").Cells(i, 1) <> Sheets("idle_data").Cells(i + 1, 1) Then
                j = j + 1
            End If
            If j > 28 Then
                Range(Sheets("idle_data").Rows(2), Sheets("idle_data").Rows(i)).Delete 'use 28 days of historical idle time data
                i = 2
            End If
            i = i - 1
        Loop
    End If
    i = i - 1
Loop
```

Figure 55 historical idle time data older than 28 days before the sample data is removed.

The data needed to generate the process file on the other hand exists of incomplete data. Therefore, rows with incomplete data are removed as they are unusable for generating tool process times. Instead, ASAP will use the older CAP group process times in these separate cases.

```
i = 2

Do While Sheets("process time per tool").Cells(i, 1) <> ""  'Remove row if data is missing
    If Sheets("process time per tool").Cells(i, 3) = "" Then
        Sheets("process time per tool").Cells(i, 1).EntireRow.Delete
    End If
    If Sheets("process time per tool").Cells(i, 5) = "" Then
        Sheets("process time per tool").Cells(i, 1).EntireRow.Delete
    End If

    i = i + 1
Loop

i = 2


End Sub
```

Figure 56 Incomplete data in the process file is removed.

Data names modifier

In the process.txt file, the colon function is used to mark and split the step name and machine type in a single column. However, the current steps named by NXP already contain one or more colons, for reading purposes. The extra colon causes the ASAP model of NXP to not be able to properly run the process.txt file, which results in errors.

To address this issue, after consultation with the model engineers, it is decided to replace the additional colon in all the step names within both the wip.txt and route.txt files with the "@" character. This change allows the process.txt to function properly, while also maintaining the readability of the step names. The following code changes the names correctly.

```
i = 2

Do While Sheets("route;").Cells(i, 1) <> ""
    Sheets("route").Cells(i, 3) = WorksheetFunction.Substitute(Sheets("route;").Cells(i, 3), ":", "@") 'Change step names route file
    Sheets("route").Cells(i, 10) = WorksheetFunction.Substitute(Sheets("route;").Cells(i, 10), ":", "@")
    Sheets("route").Cells(i, 25) = WorksheetFunction.Substitute(Sheets("route;").Cells(i, 25), ":", "@")
    Sheets("route").Cells(i, 33) = WorksheetFunction.Substitute(Sheets("route;").Cells(i, 33), ":", "@")
    i = i + 1
Loop

i = 2

Do While Sheets("wip;").Cells(i, 1) <> ""
    Sheets("wip").Cells(i, 7) = WorksheetFunction.Substitute(Sheets("wip;").Cells(i, 7), ":", "@") 'change step names wip files
    Sheets("wip").Cells(i, 14) = WorksheetFunction.Substitute(Sheets("wip;").Cells(i, 14), ":", "@")
    i = i + 1
Loop
```

Figure 57 Step names are transformed.

Option.rdf modifier

To make the model account for the process time file, the following adjustments have to be made in the option.rdf file. These changes make sure the model takes the new files into account and uses the new files correctly. The idle time is already incorporated into the option.rdf file of the older model.

```
For i = 1 To 100
    If Sheets("options").Cells(i, 1) = "PROCESS_FILES" Then 'implements process.txt file into ASAP
        Sheets("options").Cells(i, 2) = "file"
        Sheets("options").Cells(i, 3) = "process.txt"
        Exit For
    End If
Next i
```

Figure 58 The process time files is assigned to the option.rdf file.

Generating Process.txt file

Whenever all data is correctly extracted and prepared, both the process and idle time files will be created. The following variables names are set:

```
Sub Generate_process_file()

Application.ScreenUpdating = False
Dim rows_route As Double
Dim rows_process As Double
Dim start_cap As Double
Dim end_cap As Double
Dim start_location(1 To 150) As Variant
Dim end_location(1 To 150) As Variant
Dim caps As Double
Dim process_length As Double
Dim process_name(1 To 150) As Variant
Dim step As String
Dim cap_name As String
Dim recept_name As String
Dim units As Double
Dim proceff_stn(1 To 40) As Variant
Dim proceff_start(1 To 40) As Variant
Dim proceff_end(1 To 40) As Variant
Dim proceff_factor As Double
Dim i As Double
Dim j As Double
Dim z As Double
Dim machine As Variant
Dim idletimevalue As Double
```

Figure 59 Assign variables for modification model.

the process time file is empty and needs the following functions to be set in the columns. The idle time file columns function are already implemented and do not need this modification.

```
Sheets("process").Cells.Clear
Sheets("process").Cells(1, 1) = "STEP" 'Set functions for process file
Sheets("process").Cells(1, 2) = "STN"
Sheets("process").Cells(1, 3) = "PTIME"
Sheets("process").Cells(1, 4) = "PTUNITS"
```

Figure 60 Insert functions to process time file.

To generate a complete process time file, two files must be compared to identify which tools can execute a specific process step and its associated time. These data can be found in the route.txt file, with 34713 lines of data, and the SFC file, with 25778 lines. To generate a complete process file, all data must be compared to each other. However, comparing all data would require the code to process approximately 0.89 billion lines of data. To optimize this process, the placement of the process times for all machines is organized per cap group. Additionally, any unavailable data is excluded from the following code:

```
Do While Sheets("process time per tool").Cells(process_place, 1) <> "" 'remove incomplete and some special events process file
    process_name(caps) = Sheets("process time per tool").Cells(process_place, 14)
    start_location(caps) = process_place
    Do While Left(Sheets("process time per tool").Cells(process_place, 14), 1) <> ""
        If Left(Sheets("process time per tool").Cells(process_place, 14), 4) <> Left(Sheets("process time per tool").Cells(process_place + 1, 14), 4) Then
            If Sheets("process time per tool").Cells(process_place, 11) = "NA" Then
                process_place = process_place + 1
                Exit Do
            Else
                end_location(caps) = process_place
                process_place = process_place + 1
                caps = caps + 1
                Exit Do
            End If
        End If
        process_place = process_place + 1
    Loop
Loop
```

Figure 61 Implement search area for generating process times.

When the model has found a match between the two files, the saves the process step, tool name, process time in seconds, and process unit “sec” in the process.txt sheet. A colon is added between the process step and product type to correctly add the step. Furthermore, the SFC file contains data of the historical average time a tool needs to process a complete batch with multiple lots, divided by the number of lots per batch. However, the actually average process time for a lot within the machine is needed, which corresponds to the process time per batch. To calculate this, the number of lots (or units) per batch is multiplied with the average time required to process a batch.

```
end_location(caps) = process_place - 1
process_place = 2
caps = 1

Do While Sheets("route").Cells(rows_route, 1) <> "" 'Generate process files
    proceff_factor = 1
    For caps = 1 To 150
        If Sheets("route").Cells(rows_route, 4) = process_name(caps) Then
            If Sheets("route").Cells(rows_route, 7) = "per_batch" Then
                recept_name = Sheets("route").Cells(rows_route, 29)
                If recept_name <> "" Then
                    start_cap = start_location(caps)
                    end_cap = end_location(caps)
                    step = Sheets("route").Cells(rows_route, 2) & ":" & Sheets("route").Cells(rows_route, 3)
                    For rows_process = start_cap To end_cap
                        If Sheets("process time per tool").Cells(rows_process, 2) = recept_name Then
                            units = Sheets("process time per tool").Cells(rows_process, 13)
                            If units > 0 Then
                                Sheets("process").Cells(Location, 1) = step 'insert process data into excel
                                Sheets("process").Cells(Location, 2) = Sheets("process time per tool").Cells(rows_process, 1)
                                Sheets("process").Cells(Location, 3) = Sheets("process time per tool").Cells(rows_process, 3) * units
                                Sheets("process").Cells(Location, 4) = "sec"
                                Location = Location + 1
                            End If
                            caps = 200
                        End If
                    Next rows_process
                End If
            End If
        End If
    Next caps
    rows_route = rows_route + 1
Loop
```

Figure 62 Implement process times per process for each machine in process.txt file.

Generating cal_reducecapacity.txt file and reducecapacity_attach files

The idle time file is contains data of the idle time for each station. The idle per station is calculated based on historical data. For each station, the lower 20 percentile of data in which the station has an availability of at least 50% is used. Additionally, a maximal idle time of 20% may be implemented. This following figures show the code in combination with the output of one machine:

```
rows_route = 2
Location = 2
process_place = 2
caps = 1
i = 2
j = 0
Do While Sheets("cal_reducecapacity").Cells(i, 1) <> "" 'calculate idle time for every tool
    Sheets("idle_data_proc").Range("A2:A30").ClearContents
    machine = Right(Sheets("cal_reducecapacity").Cells(i, 1), 6)
    j = 0
    idletimevalue = 0
    z = 2
    Do While Sheets("idle_data").Cells(z, 1) <> ""
        If Sheets("idle_data").Cells(z, 2) = machine Then
            If Sheets("idle_data").Cells(z, 4) > 50 Then
                Sheets("idle_data_proc").Cells(j + 2, 1) = Sheets("idle_data").Cells(z, 5)
                j = j + 1
            End If
        End If
        z = z + 1
    Loop
    If j < 4 Then
 idletimevalue = 0.2
    Else
        idletimevalue = Sheets("idle_data_proc").Cells(2, 3)
        If idletimevalue > 0.2 Then
            idletimevalue = 0.2
        End If
    End If
    Sheets("cal_reducecapacity").Cells(i, 7) = idletimevalue
    i = i + 1
Loop
```

Figure 63 Calculate idle time for each machine.

A	B	C
idle times		
0.1769		0
0		
0		
0		
0.0002		
0.0446		
0.087		
0		
0		
0.3958		
0.3644		
0.0805		
0.983		

Figure 64 Example idle time calculation.

Implementing modification into ASAP

Finally, the data must be integrated into the route.txt, wip.txt, option.rdf, process.txt, cal_reducecapacity.txt and reducecapacity_attach of ASAP. Afterwards, the ASAP model using the new data must do a single simulation and the data is ready for validation.

```
Application.ScreenUpdating = True 'update screen for calculation
Application.ScreenUpdating = False

Application.DisplayAlerts = False 'remove confirm alert
```

Figure 65 Update data of all sheets.

```
path_old_route = Cells(2, 2)
path_old_wip = Cells(3, 2)
path_old_option = Cells(4, 2)

path_new_route = WorksheetFunction.Substitute(path_old_route, "old model input data", "Improved model input data")
path_new_wip = WorksheetFunction.Substitute(path_old_wip, "old model input data", "Improved model input data")
path_new_option = WorksheetFunction.Substitute(path_old_option, "old model input data", "Improved model input data")
path_new_process = WorksheetFunction.Substitute(path_new_route, "route.txt", "process.txt")
path_new_idle_calreducecapacity = WorksheetFunction.Substitute(path_new_route, "route.txt", "cal_reducecapacity.txt")
path_new_idle_reducecapacity = WorksheetFunction.Substitute(path_new_route, "route.txt", "reducecapacity_attach.txt")
```

Figure 66 Assing paths of ASAP model in which modifications have to be implemented.

```
Sheets("route").Select
Application.ActiveWorkbook.SaveAs Filename:=path_new_route, FileFormat:=xlText
Sheets("wip").Select
Application.ActiveWorkbook.SaveAs Filename:=path_new_wip, FileFormat:=xlText
Sheets("options").Select
Application.ActiveWorkbook.SaveAs Filename:=path_new_option, FileFormat:=xlText
Sheets("process").Select
Application.ActiveWorkbook.SaveAs Filename:=path_new_process, FileFormat:=xlText
Sheets("cal_reducecapacity").Select
Application.ActiveWorkbook.SaveAs Filename:=path_new_idle_calreducecapacity, FileFormat:=xlText
Sheets("reducecapacity_attach").Select
Application.ActiveWorkbook.SaveAs Filename:=path_new_idle_reducecapacity, FileFormat:=xlText


Application.DisplayAlerts = True 'set confirm alert back
```

Figure 67 Implement modifications into older ASAP model.