ABSTRACT

Title of Thesis:	MODELING AND SIMULATION OF A SEMICONDUCTOR MANUFACTURING FAB FOR CYCLE TIME ANALYSIS Aditya Shinde, Master of Science in Systems Engineering, 2018
Thesis Directed By:	Dr. Michael Fu, Decision, Operations & Information Technologies Robert H. Smith School of Business

The goal of the thesis is to conduct a study of the effects of scheduling policies and machine failures on the manufacturing cycle time of the Integrated Circuit (IC) manufacturing process for two processor chips, namely Skylake and Kabylake, manufactured by Intel. The fab simulation model was developed as First in First Out (FIFO), Shortest Processing Time (SPT), Priority based (PB), and Failure FIFO (machine failures) model, and the average cycle times and queue waiting times under the four scheduling policy models were compared for both the Skylake and Kabylake wafers. The study revealed that scheduling policies SPT and PB increased the average cycle time for Skylake wafers while decreasing the average cycle time for the Kabylake wafers, when compared to the base FIFO model. Machine failures increased the average cycle time for both types of wafers.

MODELING AND SIMULATION OF A SEMICONDUCTOR MANUFCATURING FAB FOR CYCLE TIME ANALYSIS

by

Aditya Ramaji Shinde

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Master of Science
2018

Advisory Committee:
Professor Michael Fu, Chair
Professor Jeffrey W. Herrmann
Professor Steve I. Marcus

Preface

The goal of this thesis is to present a comprehensive study of the manufacturing processes involved in the fabrication of Integrated Chips (IC's). The motivation for this research stemmed from my passion in semiconductors and their wide applications in the electronic industry.

With the advent of technology and the need to maintain a strong competitive position in the market, semiconductor manufacturing companies employ various techniques to enhance systems in the manufacturing process while producing chips with better performance at similar or lower costs.

This extensive scope for improvement in various aspects of the IC manufacturing process, fueled my passion to pursue a master's thesis research in this industry. This thesis is a result of the best understanding of the IC fabrication process by the author.

Dedication

I would like to dedicate this thesis to my beloved parents. I am deeply indebted to them for their continuous support and love through all my endeavors in life.

Acknowledgements

I am using this opportunity to express my gratitude to everyone who supported me throughout the course of this thesis. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the thesis work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the thesis.

I express my warm thanks to my advisor and mentor, Dr. Michael Fu, for his continuous support and guidance throughout the course of this thesis.

I would also like to thank my thesis committee members, Dr. Herrmann and Dr. Marcus, and all the other people who provided me with the great feedback and their proactive contributions and involvement in each phase of my thesis.

I am also very grateful to Dr. John MacCarthy, the Director of the Systems Engineering Program at The University of Maryland, who gifted me the opportunity to pursue a master's degree at an esteemed University and for assisting me throughout my journey as a master's student. I would also like to thank all the Professors in the Systems Engineering program, for imparting rich knowledge. Lastly, I would like to thank all my fellow graduate students for their moral and knowledge support.

Thank you,

Aditya Ramaji Shinde

Table of Contents

List of Tables

List of Figures

List of Abbreviations

IC	Integrated Circuit
FIFO	First in First Out
SPT	Shortest Processing Time
PB	Priority Based
SysML	Systems Modeling Language
FOUP	Front Opening Unified Pod
WIP	Work-in Process
MBSE	Model Based System's Engineering
BDD	Block Definition Diagram
DES	Discrete Event Simulation
SIMAN	Simulation Language
RNG	Random Number Generator
MTBF	Mean Time Between Failures
MTTR	Mean Time to Repair

Chapter 1: Introduction

General

An integrated circuit (IC) is a device made of interconnected electronic components that are imprinted onto a tiny slice of a semiconducting material, such as silicon or germanium. An integrated circuit is smaller than a fingernail and can hold millions of circuits that are capable of performing a wide range of computing operations at high speeds. Monocrystalline silicon was identified as the main-substrate that can be used to manufacture IC's. This material is abundantly available in nature and has very special properties making it extremely affordable and appealing. It acts as a semiconductor, wherein it conducts electricity under some conditions and alternatively acts as an insulator in others. These properties have enabled IC's to be extensively used in electronic devices like computers and mobile phones.

Semiconductor manufacturing takes place under constant change of manufacturing conditions. With the advent of process technology, the size of the area per function on the wafer has reduced to almost half. Also, many new chips with complex architectures are introduced, which need to be accommodated in the existing process technology. This calls for continuous process improvement in the semiconductor chip manufacturing process to cater to the fast-changing market demands.

IC's undergo manufacturing in production units called fabs. Big giants like Intel, Texas Instruments and Apple manufacture IC's in their own fabs, whereas other companies like Advanced Micro Devices and Qualcomm outsource the manufacturing process to other global chip manufacturers around the world. Fabs require expensive devices to function, and estimates suggest the cost of establishing a new fab plant to values as high as $3-$4 billion. The central part of the fab, referred to as the clean room, houses the machines required for the manufacturing process. This room is designed as a dust-free environment, since even a small speck of dust can ruin the micro-circuit. The room also maintains a controlled temperature and humidity and is also damped against vibration.

Moore's Law

Moore's Law was an observation made by Gordon E. Moore, the co-founder of Intel. This law states that the number of transistors per area doubles approximately every two years (Moore, 1975). Because of the accuracy with which Moore's Law has predicted past growth in IC complexity, it is viewed as a reliable method of calculating future trends as well, setting the pace of innovation, and defining the rules and the very nature of competition. And since the semiconductor portion of electronic consumer products keeps growing by leaps and bounds, the Law has aroused in users and consumers an expectation of a continuous stream of faster, better, and cheaper high-technology products (Schaller, 1997). Further, the simple idea that transistor density is continually increasing means computing power goes up just as costs and energy consumption go down. As of today, the number of transistors on an integrated chip have substantially increased from a mere 10^3 in 1970 to 10^9 in 2017.

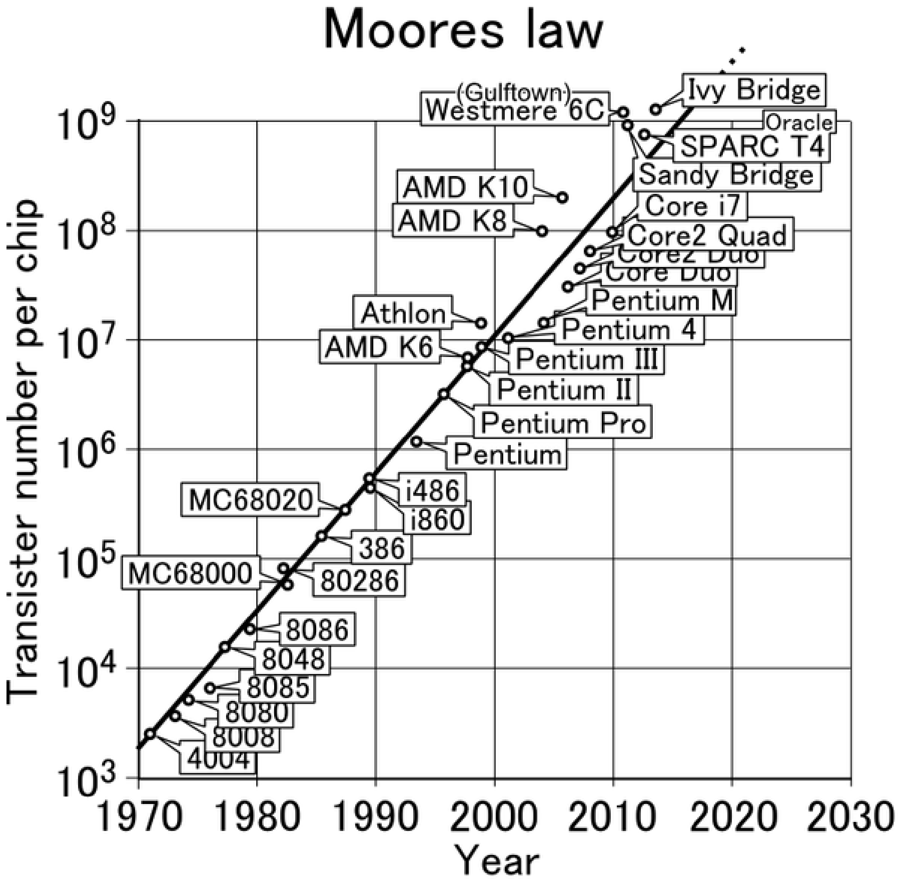

Figure 1.1 Moore’s Law Validity (Cringely, 2013)

Chip Design and Architecture

The scope of the study was confined to manufacturing two different types of processor chips manufactured by Intel, namely Skylake (i7-7800X) and Kabylake (i5-7400T).

Skylake Architecture

Skylake is a chip micro-architecture that was launched by Intel in August 2015 as a successor to Broadwell to overcome processing delays. Skylake was branded as the 6th generation of Intel’s processors (Intel Developer Forum, 2015). The thesis deals with the process analysis of the i7 family of the Skylake processor. This processor has a high-end performance and uses the 14 nanometer (14 nm) lithography process of semiconductor manufacturing.

The following table describes the specifications of a Skylake (i7-7800X) processor,

Specification Type	**Specification Description**
Processor Name	i7-7800X
Lithography Process	14nm
Average Customer Price	$386
Number of Cores	6
Number of Thread	12
Memory Channels	4

Table 1.1 Intel Skylake (i7-7800X) Specifications (Intel, 2016)

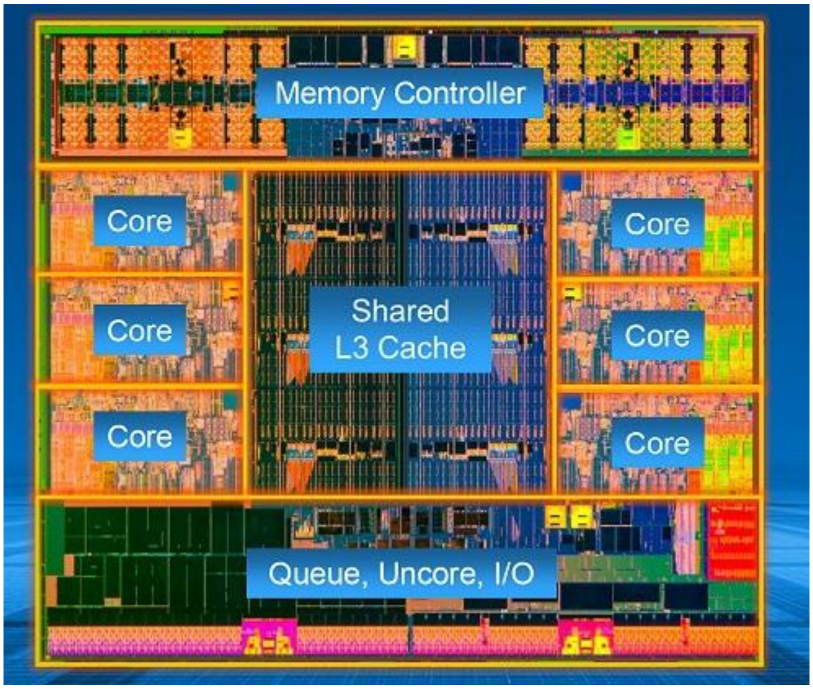

Figure 1.2 Intel i7 Processor Die Shot (Intel, 2017)

Kabylake Architecture

Kabylake is a chip micro-architecture that was launched by Intel in August 2016. Kabylake is branded as the 7th generation Intel processor (Intel Developer Forum, 2016). The thesis deals with the process analysis of the i5 family of the Kabylake processor. This processor has a mid-range performance and uses the 14 nanometer (14 nm) lithography process of semiconductor manufacturing.

The following table describes the specifications of a Kabylake (i5-7400T) processor,

Specification Type	**Specification Description**
Processor Name	i5-7400T
Lithography Process	14nm
Average Customer Price	$185
Number of Cores	4
Number of Thread	4
Memory Channels	2

Table 1.2 Intel Kabylake (i5-7400T) Specifications (Intel, 2016)

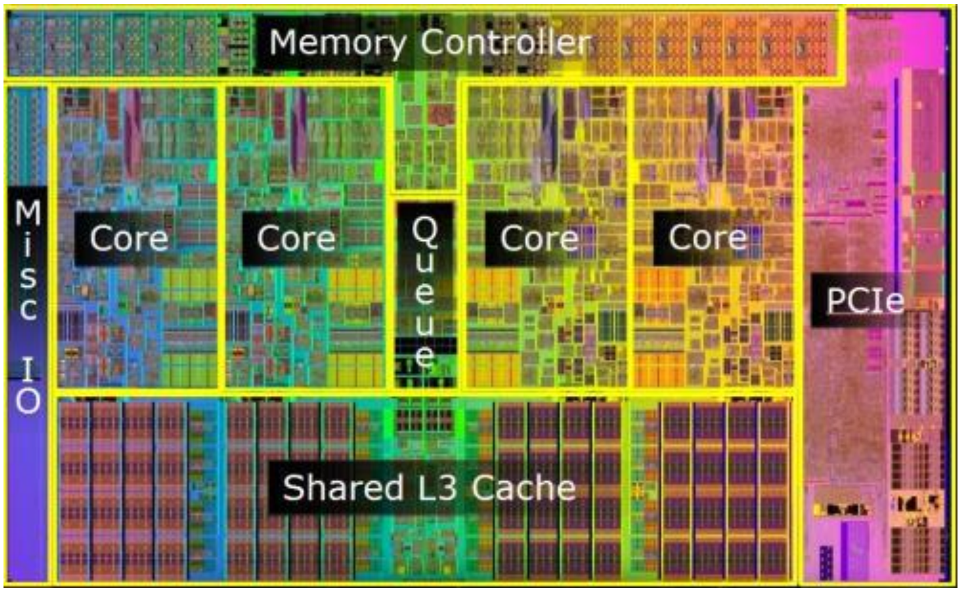

Figure 1.3 Intel i5 Processor Die Shot (Intel, 2017)

A semiconductor chip’s core defines its performance. It comprises of a logical execution unit containing cache and functional units. More number of cores can be associated with higher performance.

A semiconductor chip’s thread defines its ability to run multiple applications on a single core. It increases processor throughput, improving overall performance on threaded software. Higher number of threads can be associated with higher application loading capacity.

A semiconductor chip’s memory channel is its memory bandwidth availability for the CPU. It facilitates the random-access memory (RAM) capability of a CPU. Higher the number of memory channels, better the memory bandwidth availability.

Skylake, as can be observed from the analysis, has higher values for the chip specifications compared to Kabylake. This suggests that Skylake is a much complex processor compared to Kabylake and hence requires advance processing at some stages of the IC manufacturing process.

Thesis Organization

The goal of the thesis is to conduct a study on the different factors that affect the production cycle time of the IC manufacturing process. The research encompasses 2 major factors that have a significant effect on the cycle time, namely: production sequences (FIFO, SPT, and Priority Based) and machine failures (mean time between failure and mean time to repair).

The study is conducted on an ARENA simulated representative IC fab for two different processor chips manufactured by Intel. Skylake being an i7 processor is assumed to have a complex structure and hence requires more number of iterative processes and higher processing times at some processing stations.

The first part of the study involves developing a process architecture using SysML modeling language. These architectures were used to construct the FIFO base model on Arena. Further, queueing analysis and sensitivity study were performed to analyze the number of resources required at each processing station to attain maximum resource utilization for a stable system operation.

The second part of the study involves evaluating the performance of the fab for different queue sequences. The manufacturing process is first simulated as a first-in first-out (FIFO) queue model, and the average queue waiting times at each processing station are recorded. These processes are then subjected to SPT (Shortest Processing Time), priority-based queueing (higher priority for Kabylake over Skylake) and their effect on cycle times are observed.

The third part of the study involves incorporating machine failure in terms of mean time between failure (MTBF) and mean time to repair (MTTR) in the simulation model, to develop the FIFO failure model. This cycle time generated in this model is also compared to the base FIFO model.

The final part of the thesis presents a hypothesis study for statistical significance of the cycle time using the ANOVA test of the four simulation models developed. The individual cycle time differences of the models are compared using the post-hoc Tukey Test.

Chapter 2: Semiconductor Manufacturing Process Challenges

Semiconductor Manufacturing Process

The technology behind engineering an IC goes far beyond the simple assembling of individual components. In fact, microscopic circuit patterns are built on multiple layers of various materials, and only after these steps have been repeated a few hundred times is the chip finally complete (Samsung, 2015). It involves multiple photolithographic and chemical processing steps during which the electronic circuit layers are gradually developed over the silicon wafer. The entire process from sand to packed silicon chips takes almost 3-4 weeks.

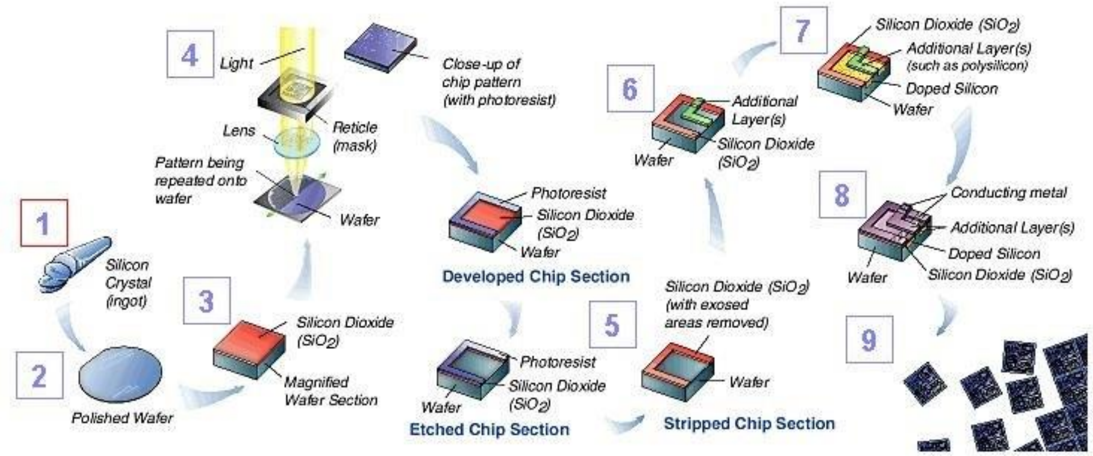
1
Silicon Crystal (ingot)
2
Polished Wafer
3
Silicon Dioxide (SiO_2)
Magnified Wafer Section
4
Light
Lens
Reticle (mask)
Pattern being repeated onto wafer
Wafer
Close-up of chip pattern (with photoresist)
Photoresist
Silicon Dioxide (SiO_2)
Wafer
Developed Chip Section
Photoresist
Silicon Dioxide (SiO_2)
Wafer
Etched Chip Section
5
Silicon Dioxide (SiO_2) (with exosed areas removed)
Wafer
Stripped Chip Section
6
Additional Layer(s)
Silicon Dioxide (SiO_2)
Wafer
7
Silicon Dioxide (SiO_2)
Additional Layer(s) (such as polysilicon)
Doped Silicon
Wafer
8
Conducting metal
Additional Layer(s)
Doped Silicon
Silicon Dioxide (SiO_2)
Wafer
9

Figure 2.1 IC Manufacturing Process (Fishman, 2017)

Semiconductor Manufacturing Process Components

Fab: Semiconductor fab is the facility that houses the IC fabrication process. The air inside a fab cleanroom is filtered and recirculated continuously, and employees wear special clothing such as dust-free gowns, caps, and masks to help keep the air particle-free.

Wafer: Wafers can be defined as a thin slice of silicon crystal that is used in the fabrication of semiconductor chips. Usually the wafer serves as a substrate for the microcircuit to be built in and on it. Wafers undergo processing at each stage of the semiconductor manufacturing process and are finally cut at the end of the fabrication. A single silicon wafer may consist of anywhere between 450-700 chips. Chips fabricated on the same wafer generally possess the same architecture.

Product: Fabricated chip/IC serves as the final product of a semiconductor manufacturing process. These undergo different levels and routes of processing at each stage of the manufacturing process based on their architecture.

FOUP: FOUP (Front Opening Unified Pods) acts a material handling system for transportation of wafers. It is specialized plastic enclosure designed to hold silicon wafers securely and safely in a controlled environment. Wafers are transported in batches of 25 around the fab facility.

Wafer Batch: A batch of wafers refers to a lot or collection of wafers. Generally, batches are formed, where more than one wafer can be processed at a time.

Route: Process route can be described as the path taken by the wafer inside the fab facility. The route for each wafer differs based on complexity and architecture. Complex wafers experience reentrant flows through the same equipment/route fabricate gradual layers. Alternatively, some lesser complex wafers skip some workstations and routes because of their simple architectures. All wafers start at the initial point and process through the route to reach the packaging station.

Process operation: A process operation is a step within the route of the where different equipment's work on the wafer to customize its architecture. Most of the processing operations are performed on a single wafer.

Machine: The tools that perform operations on these wafers are the fabricating machines. The machines performing similar operations are grouped at product workstations.

Recipe: Each wafer undergoes specific process steps at workstations referred to as the recipe for the chip. The recipe for the chip is determined by the architecture that needs to be fabricated.

Scheduling Rule: A scheduling rule dictates which job among those waiting for service is to be scheduled in preference to others. Scheduling a job means scheduling the next operation of the job.

Work-In Process: Work in process can be defined as those entities that are being either just fabricated or waiting in a queue or buffer to be processed. Production management's aim to minimize and control the work in process in order to manage capacity and reduce slack in the production system (LeanKit, 2018).

Cycle Time: Manufacturing cycle time refers to the time required or spent to convert raw materials into finished goods. Technically, it is the length of time from the start of production to the delivery of the final products. It comprises of process time, move time, inspection time, and queue time (AccountingVerse, 2017).

Unit Manufacturing Processes

The following section briefly describes the high-level unit manufacturing stages in manufacturing integrated circuits.

Silicon Wafer Fabrication

Before a semiconductor can be built, sand needs to be converted to silicon. Sand, especially Quartz, has high percentages of Silicon in the form of Silicon dioxide (SiO_2), and is the base ingredient for semiconductor manufacturing (Intel, 2011). Sand is first melted to a temperature of 1420^0C, above the melting point of silicon. Dopants from the Group III and Group IV elements (Boron, Phosphorous, Arsenic, etc.) of the periodic table are added to give desired electrical properties to the melted silicon.

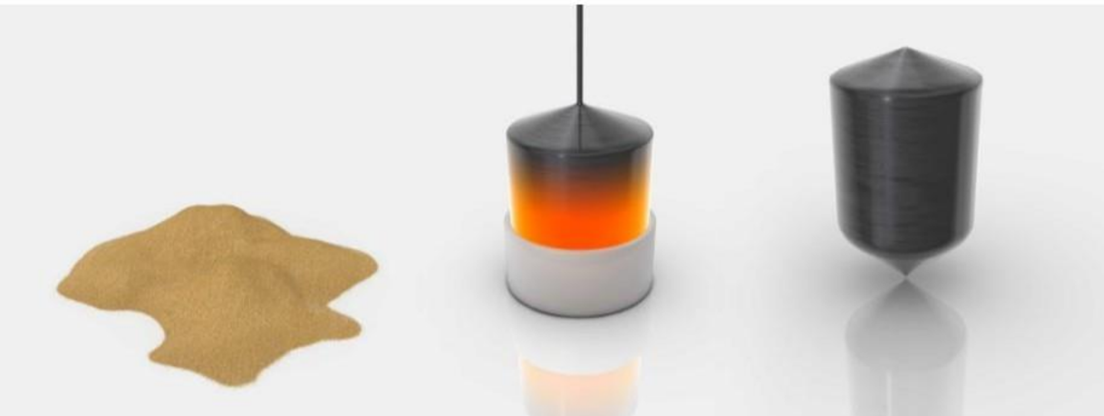

Figure 2.2 Sand to Silicon Ingot (Intel, 2011)

The molten silicon is then purified in multiple steps until the manufacturing quality of Electronic Grade Silicon is reached. The molten silicon is then molded into large cylindrical ingot rods that weigh around 100 kg and have a silicon purity of 99.9% (Intel, 2011). The ingot growth process begins from a single purified silicon seed that grows the crystal in opposite direction to mold the melt. The initial growth is rapid and decreases subsequently to allow the diameter of the ingot to increase to the required dimension, generally 300mm. The ingot growth process can take anywhere from 1 week to 2 weeks. Once pure silicon ingots are formed, they are sliced into very thin wafers which have a thickness of 1mm (Silicon Valley Micro-electronics, 2015).

Thermal Oxidation

Semiconductor manufacturing companies like Intel and Global Foundries receive sliced wafers on which they fabricate the processing circuits. Thermal oxidation is the process of exposing silicon wafers to a temperature of 800-1200^{0}C to grow a layer of silicon-dioxide. A single furnace accepts many wafers at the same time in a specially designed quartz rack (Appels et al. (1970)). These quartz racks can enter the furnace vertically or horizontally, depending upon the uniformity, thickness and time constraints of the oxide-layer deposition.

Oxidation can be of two types, depending on the quality and the thickness of the oxide layer desired. There is a trade-off in using wet and dry oxidation for thermal oxidation of silicon wafer, as wet oxidation has higher growth rate compared to dry oxidation, while on the other hand, the higher oxidation date of wet oxidation leaves behind dangling bonds at the oxide surface which leads to leakages in the current flow inside chips

Hence, depending on the quality and thickness of the oxide layer required, the time for oxidation is decided. The time for oxidation is hence governed by the Deal-Grove model which gives the relation between the oxidation time (τ) and oxide-layer thickness (X_0).

$$\tau = \frac{\mathrm{X}_0^2}{B} + \frac{X_0}{\frac{B}{A}}$$

where A and B are process-related constants (Liua, et al., 2016).

Photolithography

This process involves the use of light-induced polymerization to transform liquid resin into a solid polymer in the lithographic illuminated areas. The photopolymers are then exposed to real-time infrared spectroscopy using UV light or laser to cure the solid polymer (Decker, 2002).

A photo-resist material is applied which changes its characteristics upon exposure to light, i.e. either softening or hardening depending on the type of photoresist. Chromium masks are generally used for photolithography. Depending on the circuit architecture and the areas to be illuminated, positive or negative photoresist material is used. Although positive photoresist materials require a higher exposure time and are more expensive than the negative photoresists, they are more widely used than the later because of their higher step coverage (Printed Circuit Imaging, 2014).

The high-level processes modelled under Photolithography as a part of the thesis are follows:

- Applying Photo-resist Material
- Stepper Exposure
- Photoresist Development

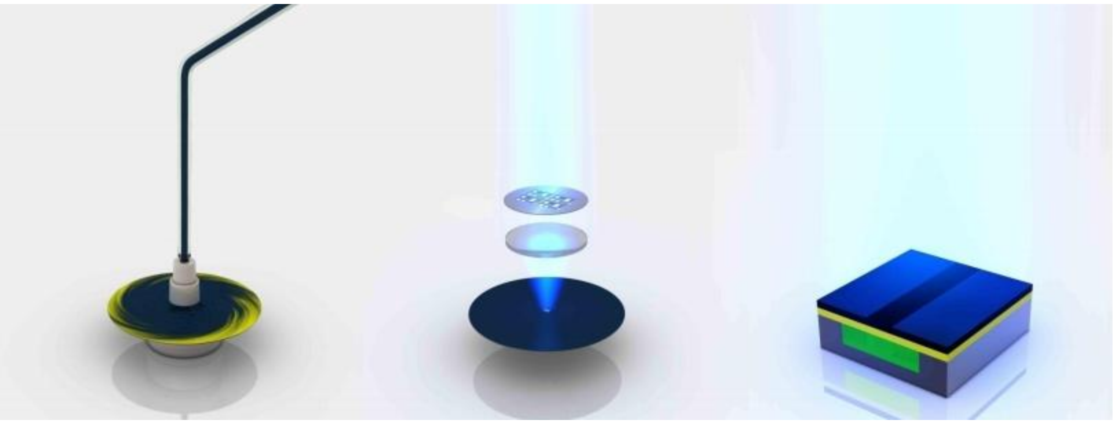

Figure 2.3 Three Stages of Photolithography (Intel, 2011)

Ion Implantation

Ion Implantation is the process of transferring ions from one element into a solid target by fast accelerations at a low temperature. This leads to changes in the physical, chemical and electrical properties of the solid target. Ion ranges are between 10 nanometers and 1 micrometer. Ion implantation is highly effective at the surface of the solid target. The concentration and energy of these ions gradually decreases as they travel through the solid, due to collisions with the atoms of the solid target and drag from the electron orbitals (Hamm, Robert, & Marianne, 2012).

Doping is a common technique used in ion implantation to introduce dopant impurities into the crystalline silicon. Dopants, when injected into the semiconductor act as charge carriers and depending upon the dopant element used, a hole is formed for the p-type dopant and an electron for the n-type dopant (Philip Laube, 2010).

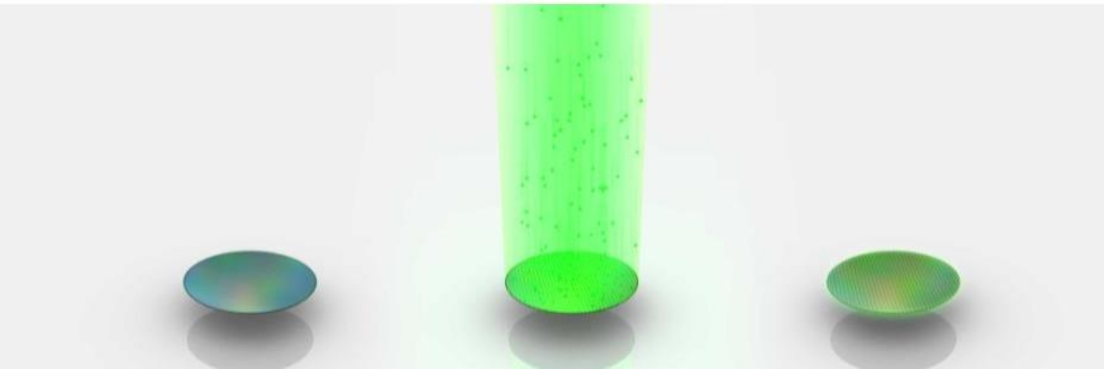

Figure 2.4 Ion Implantation Stages (Intel, 2011)

Hence, the energy of the ions, the ion species, ion dosage and the target scanning areas greatly affect the time required for the implantation process. The ion dosage time varies amongst processor chips, depending on the ion dosage and level of complexity of the chip (Cheung, 2010).

The following equation is used by chemical engineers at semiconductor manufacturing industries to calculate the implant time and dosage required for a processor chip (Cheung, 2010):

$$Dose = \frac{(Ion\ Beam\ Current) * (Implant\ Time)}{Ion\ Beam\ Scanning\ Area}$$

Etching

Etching is process that is used either to remove material from the surface of the wafer or to create a pattern on it. Etching in semiconductor manufacturing can be of two type, Dry etching and Wet Etching. Dry or plasma etching used for circuit-defining steps, while Wet etching is used mainly to clean wafers. Although both methods are used extensively, dry etching is more frequently used over wet etching. Etching removes material away from the patterns developed in the photolithography step. The sequence of patterning and etching is repeated multiple times during the chip-making process (Applied Materials, 2017).

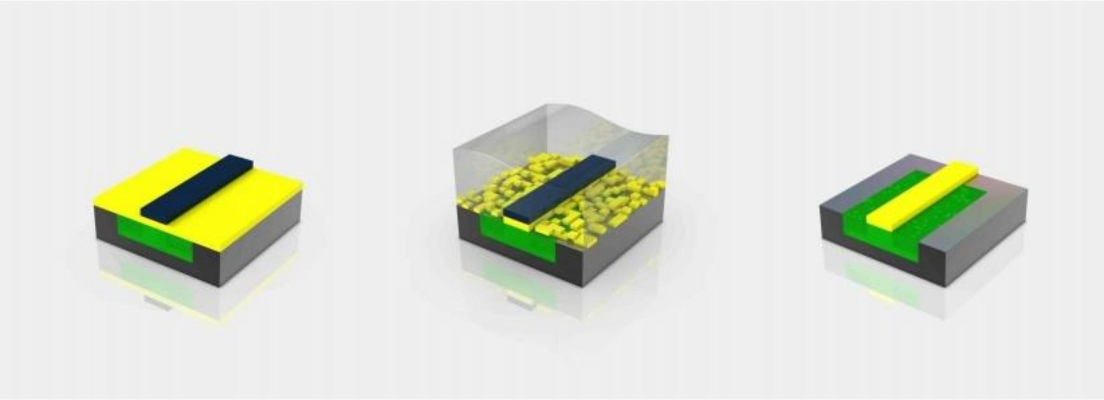

Figure 2.5 Stages in the Etching Process (Intel, 2011)

Etchants crystalline and remove material from the surface of the wafer depending upon several factors like:

- Wafer side facing the etchant
- Required etch uniformity
- Etch agent
- Etch agent type

Taking these factors into consideration, semiconductor manufacturers deduce the time to etch wafers to avoid over-etching or under-etching of the wafers (National Taipei University of Technology, 2006).

Gate Formation and Metal Deposition

Gate Formation and metal deposition is the process of fabricating high-k dielectric gates and metal layers onto the wafer. Gate formation starts with developing a temporary gate electrode and dielectric, which is then etched away. This is followed by adding molecular layers of high-k dielectric to the wafer surface, which is again etched at areas where it is undesired. The combination of this and the high-k material gives the transistor much better performance and reduced leakage than would be possible with a traditional silicon dioxide gate (Intel , 2012).

Metal deposition is the process of making connections to link circuits together. Conductive pathways are formed on the surface of the wafer by metal deposition. Materials like copper, aluminum, nickel and other alloys are used to form connections. A single layer recipe is fabricated during one visit of the wafer at the processing station. It is followed by polishing the polished layer. Layers of deposited metal are fabricated for a single wafer based on its defined circuit architecture (Intel , 2012).

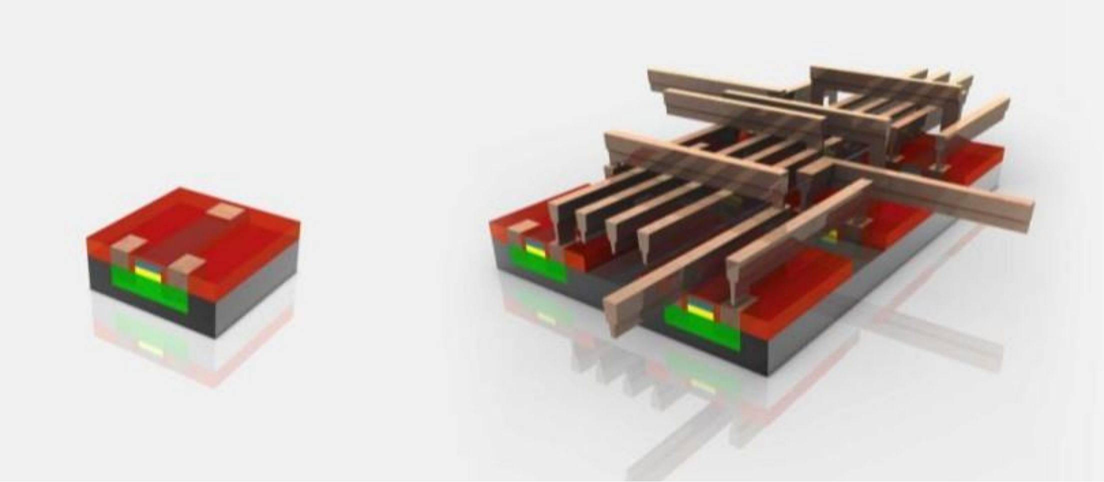

Figure 2.6 Stages in the Gate Formation and Metal Deposition Process (Intel, 2011)

Board Assembly and Packaging

This is the last step in the semiconductor manufacturing process. Once the wafer visits all the stations in its predefined routes, it is cut into individual chips. This is followed by assembling cut chips on a printed circuit board (PCB and making the necessary connections. The assembled PCB's are then packed and shipped to the customer.

Process Challenges

- IC Manufacturing is known for its multiple or large number of process steps. Even with the most modern technology, a wafer visits on an average 25 processing stations during fabrication.

- The fabrication process is complex, with most of the tools being single wafer processing tools to fabricate wafers with varying architectures. Also, the recipe for each wafer varies; hence different routes and reentrant flows to the same workstation are common.

- There is an intermixture of batches/lots of wafers containing different recipes, hence assigning priority to lots in order to meet schedule constraints and reduce manufacturing cycle time poses a big challenge. This might also lead to long queues at some stations.

- At some stations wafers having the same recipe, which can be processed in lots, are batched together. The challenge here is to calculate the optimum batch size at each station for faster and quality processing.

- The biggest challenge in this industry is to identify the type and quantity of each wafer to be manufactured, as the optimum goal of any fab is to achieve shorter cycle times, increase throughput and acquire more flexibility to products while maintaining the same production costs.

Some of the high-level advantages of short cycle times are (Stubbe, 2010):

- lean inventory
- faster time to market
- fast yield learning
- fast excursion finding
- less reliance on demand forecast
- flexibility in product output enabling fast reactions to customer demand.

Chapter 3: Literature Review and Approaches in the Past

IC Manufacturing Modeling and Simulation

Pillai, Bass, Dempsey, & Yellig (2004) developed and implemented a simulation model of Intel's 300-mm fab that uses 90-nm high volume manufacturing and 65-nm technology. Their objective of developing this simulation was to understand the key attributes associated with the simulators that were being utilized for capacity planning, automation systems designs and tactical manufacturing execution and decision support for continuous improvement. Their simulations presented a dynamic behavior of several production tools and engineering lots along with automated material handling system (AMHS) behavior in the production simulation. They identified 5 critical data elements in the model; automated input data integration (DI) system with the automatic model builder and simulation runs configurator; production equipment and work-in process (WIP) management rules simulator; intrabay AMHS simulator; data analysis systems for reviewing model outputs and model validation and calibration. They observed that when lots are prioritized, priority lots start to dominate equipment capacity, and regular production lots may be in danger of getting increasingly deprioritized, resulting in an inordinate increase in its cycle time. It was also observed that the cycle time (fab velocity) increases at lower priority lots with increasing number of higher priority lots.

Domaschke, Brown, Robinson, & Leibl (1998) developed a discrete-event simulation model to evaluate the production practices of a high-volume semiconductor back-end operations of the Siemens fab. The goal of their simulation was to identify the potential areas for productivity improvement that would yield a 60% reduction in the manufacturing cycle time. The scope of the paper was confined to the analysis of the Assembly, Burn-In and the test operations in the fab. The higher work in process and cycle times problem faced by Siemens was related to an exceed in the original loading plan due to a high production demand. The software used for modelling and simulating the fab was Factory Explorer. The simulation model used MTBF and MTTR aspects to express and record the machine down times. Real-time observations and conversations with operators and engineers on the shop floor were made to collect valid input data for the simulation. In order to obtain the data from the same data warehouse, a team of IT and Computer Integrated Manufacturing engineers was formed to collect, organize and store shop floor data. For the purpose of model validation, historical records of factory cycle time, cycle time by tool group, equipment utilization, and average inventory were compared against model outputs and were found to be within 10% of the value range. The results of the simulation revealed that improvement in areas like cross-functional teams at the shop floor, using smaller transport lot sizes, lower variability in lots and operating non-constraint equipment have a great impact on the cycle time. Implementing these factors in the simulation model, the fab achieved a 41% reduction in average cycle time.

Arisha & Young (2004) discuss the importance of simulation in the semiconductor fab to meet the complexities of market and process steps. They identify the various phases a semiconductor fab production planner goes through: factory layout design, factory construction, process selection and design, start-up and full production. They describe the qualities of a good simulation model as good correlation with the existing system performance, good integrity in the model and timeliness. They also present a comparative study between a complex and simple model and explain how a process modeler should confine model scope based on the current production bottleneck areas and the required level of detail. The paper discusses how additional experiment features like length of simulation run, warm-up period, number of replications and design of simulation experiments using DOE techniques, saves time, provides high quality outputs (avoid misinterpretation) and better statistical control. Lastly, the paper also provides a comparative analysis on the advantage and disadvantages of using modeling and simulation in semiconductor manufacturing.

Tullis, Mehrotra, & Zuanich (1990) developed a discrete-event simulation model of their R&D fabrication facility at Hewlett Packard using the ManSim software to analyze capacity limitations and capacity changes that impact the manufacturing cycle time. Since their model featured more than one type of product, the lot start rate and the maximum work-in process values were set for each product. They also modelled random delays for equipment breakdown and repair times, material transfer etc between some recipe steps. Equipment reliability parameters like MTBF and MTTR were specified for each piece of equipment and preventive maintenance schedules were generated for workstations. A time-consuming but educational data collection technique was adopted, by collecting fab related data from the shop floor technicians and supervisors through interviews and written requests. The analysis was extended to observing the effects of operator skills (skilled vs non-skilled), staffing and shift schedules on cycle times. They also incorporated different queueing sequences like First-In-First-Out, Shortest-Processing-Time, and Least-Work-In-Next-Queue at bottleneck work-stations to observe their effects on cycle times. The results of the analysis identified the effects of adding staff or resources that had the greatest effects on cycle time. The results also provided a comparative cycle time study between base model and models developed through improved staffing, removal of unscheduled maintenance and the combination of both. The results were presented using Pareto charts and chicken charts (to display lot re-visits at stations).

Becker (2003) introduces the concept of using Petri nets in a semiconductor manufacturing model for the complete production process, to observe its effects on the total simulation time. The modelling is performed on a MASM Lab developed tool, PSim, which is based on combined queueing and Petri net formalism. They used a modular approach by first assigning the structure of a machine as a Petri net and then instantiating as many machines as needed. The process considered for the study involves a two-product system of making non-volatile chips, where the first product needs 210 production steps and the second product needs 245 production steps. Both the product routes need 28 different machines with 87 and 102 different configurations for product 1 and 2, respectively. The machines used in the Petri net model are also subjected to failure and maintenance. First the lots are tested by machines (based on complexity), to test whether the lots need to be reworked or need to be scrapped. This is followed by wafer level machine testing at certain stations, which then assigns an input buffer token for wafers waiting for processing. Queueing logic was used wherever appropriate and features like queueing disciple and priority lots was tested on the model. The simulation was run for 50,000 hours on an 800 MHz Linux PC. The results observed as an outcome of the simulation was that the system took between 19-48 hours for processing and validating the model to reach the defined confidence intervals.

Cycle Time Analysis

Chen (2013) presents a systematic procedure to plan and evaluate cycle time reduction actions by evaluating the factors that influence the job cycle time. Additionally, the relationship between the controllable factors and job cycle time are fitted with back propagation network. He identified the following reasons as pinnacle to shortening the cycle time; each job represents an opportunity cost for the factor, long cycle times lead to accumulation of WIP and the risk of wafer contamination in larger cycle times. He identified the factors associated with job cycle time as follows; utilizations of the bottleneck machines, queue length and product waiting time at a bottleneck machine, job type (size, priority and processing time) and the worker productivity. He used the method of stepwise backward elimination which involved the deletion of each factor variable, to optimize a fitness indicator (t-test). This method helped identify the factor variable that could improve fitness the most, and this process was repeated until no further improvement could be achieved. Further, a BPN (Back-propagation neural network) was established to fit the relationship between the controllable variables and the job cycle time. Based on the fitted BPN, actions for controlling the factors that affect the cycle time were planned. He proposed an estimated 7% decrease in the cycle time using this approach.

Janakiram (1996) discusses the use of Theory of Constraints (TOC) and simulation to achieve fab cycle time reduction at Motorola's advanced custom technologies R&D Fab. The cross-functional team associated with this project was trained on the principle of theory of constraints to develop custom cycle time reports, device techniques to measure theoretical cycle time and use multiples of theoretical cycle times to make wafer fab comparisons. Benchmarking was performed within Motorola to determine how other labs and fabs were measuring their performance to study their cycle time reduction techniques and the use of cycle time as a fab metric. A five-step rule identifying the elements of TOC was used as follows:

- Identify the system's constraints
- Decide how to exploit the system's constraints
- Subordinate everything else to the above decision
- Elevate the system's constraints
- Test and reiterate

The process of using TOC and simulation resulted in identifying bottleneck stations, one of them being the inspection station where the lots were piling up. Critical analysis revealed that these problems were caused due to inadequate sampling plans and previous processes having lesser load.

Stubbe (2010) discuss how a Next Generation Production System (NGPS) concept can address the cycle time needs. The high-level elements of the NGPS were identified as small lot manufacturing, transition to mini-batch and single wafer processing, changes in cluster tool design, and rapid, high volume material handling systems. Her model incorporates reentrancy in flow lines where the same workstation is visited several times during at different steps. Also, since there are more than one products being modeled, the recipe for each product varies, which is an important feature in her model. She also introduces the aspect of x-factor, which is the ratio of actual cycle time to raw process time. The purpose of using the x-factor is associated with performance comparison of routes of different lengths and fabs having different routes or technologies. For performing the simulation experiments, she used Factory Explorer 2.8 along with MS Excel as output interface. The analysis involved replacing large batch tools with mini-batch or single wafer tools at some stations. The original fab consisted of 15% batch tools. All batch tools were replaced with single wafer tools to achieve a cycle time reduction benefit of 40%. Batches comprised of four classes, namely original batch, mixed-product batch, mini-batch, hybrid batch and single wafer tool batch. Comparative analysis revealed that single wafer tools had the greatest reductions in the cycle time.

Production Sequencing

Silva et al. (2012) study the effects of production sequencing rules in the performance of Job Shop and Flow Shop manufacturing environments. The rules they considered for their study were SIPT (Shortest Imminent Processing Time), EDD (Earliest Due Date), DLS (Dynamic Least Slack), LWQ (Least Work in

next Queue), FIFO (First In First Out), LIFO (Last In Last Out), CR (Critical Ratio) and LS (Least Slack). They modeled 8 machines and 10 different products using ARENA software, accounting for randomness in product arrival and service times. Excel was used to evaluate the effect of the work in process in relation to the total tardiness and the total number of tardy orders. The simulation was run for 20 replications and 1000 minutes each for a 95% confidence for each run. The results for the Job Shop environment suggested that the best performance was presented by the EDD and LIFO sequencing rules. On the Flow Shop environment, the results suggested that the SIPT and the CR rules demonstrated the best performance. Further, since their prominent metric of interest was number of tardy orders, they concluded that LIFO and SIPT had the best overall performance for Job Shop and Flow Shop, respectively.

Wein (1998) analyzed the impact of scheduling on the performance measure, mean throughput time, on a lot of wafers. He developed a representative but fictitious model of fab that was developed using the SIMAN simulation language. For the purpose of the analysis, four types of input mechanisms were evaluated, namely, Poisson, deterministic, closed loop and workload regulating (releases a lot of wafers into the system whenever the total amount of remaining work in the system for any bottleneck station falls below a prescribed level). Certain sequencing rules were developed by identifying the stations that are heavily utilized and using a Brownian network model to approximate a multiclass queueing network. Simulation results suggested that scheduling had a great impact (35-45 percent reduction in average total queueing time) on the fab performance. In particular workload regulating, closed loop and deterministic inputs had better performance over Poisson inputs as it substantially reduced the mean and variability of throughput times. Further, queueing analysis also suggested that reducing variability at the input also enhanced the performance of the fab.

Queueing Theory and Con-WIP

Shanthikumar, Ding, & Zhang (2007) survey the applications of queueing theory in semiconductor manufacturing systems (SMS). The paper discusses methods to reduce cycle time using queueing theory in addition to simulation. They begin with analyzing the queueing models (M/M/1 and M/G/1) that are formed at the single machine stations. Further they study how these models closely represent the actual queue behavior though being oversimplified assumptions. The next study is performed on the multimachine stations using a G/G/m system where they analyze the effects on queue waiting times due to higher variances in the interarrival or service times. Further, they analyze the methods to obtain numerical solutions to complex multi-server systems that experience machine breakdowns and other interruptions that occur within the facility. The numerical solutions calculate how a smart scheduler will try to push out maintenances when the WIP level is high and resume them after the system load becomes relatively low. They discuss the queueing network models developed by Jackson as a dynamic job shop Poisson external arrival process, exponentially distributed processing times, and Markovian job transfers between tools. Further, since this model does not instantiate the realistic complications of a manufacturing system, they propose 3 important approaches namely: Decomposition approach, Fluid Networks and Diffusion Approximation. Their analysis concludes that the accuracy of classical queueing models is less satisfactory than that of simulation, partly because the complex operational behaviors of semiconductor fabs cannot be represented by one single model.

Pierreval et al. (2013) address the application of ConWIP (constant work-in process) in semiconductor manufacturing fabs to reduce work-in process and maintain good customer satisfaction. They discuss rules that are based on thresholds, which are known to influence system performance, and hence propose a model that sets these cards in the best way and avoids changing the number of cards too often. These cards control the manufacturing process by authorizing production only after receiving an order, called the pull mechanism. Hence, optimization involves determining, for each stage of the manufacturing process, the best number of cards so as to reduce the long-term production costs, while taking into account the cost caused by backward demands, WIP and inventory. They perform the simulation optimization on an ARENA simulation model using OPTQUEST. An upstream module was used to track demands entering the manufacturing process and accordingly reducing the changes in the number of cards. Additionally, order arrival trends were observed and were compared to a fixed number of demand arrivals. Hence, if an increase in demand was detected, then authorization to adaptation module was given to add an extra card; else, a new card was not added. Their experiments concluded with proving that decreasing the frequency of the number of card changes in the ConWIP system greatly affects the system performance and that this decrease can be achieved by using an adaptation module to detect trends and control the card generation.

Chapter 4: ARENA IC Fab Model

SysMl Architecture

The first step in developing a process/system model involves defining a baseline architecture. Systems Engineers use an approach referred to as Model Based Systems Engineering (MBSE) to develop system and process flow architectures to support system requirements traceability, design, analysis and verification and validation. This thesis uses the MBSE approach to identify the high-level system elements and process sequences to support system design and analysis of the IC Fab model and simulation.

MBSE is implemented on architecture modeling languages like Systems Modeling Language (SysML) and Unified Modeling Language (UML). This thesis uses SysML to capture the IC Fabs architecture by developing SysML structure and behavior diagrams. The MBSE architecture also helps convey the scope of the model to the stakeholders. For the purpose of this thesis, the scope of the wafer fabrication process is confined to the stage where the sliced and polished wafers reach the Intel Fab to the stage where the chips are tested and are ready for packaging.

System Block Definition Diagram

A block definition diagram (BDD) is used to display various kinds of system model elements and relationships between those elements to express information about the systems structure (Delligatti, 2013). The IC Fab BDD model decomposes the system architecture into the system of interest (SOI) domain, the system users and the SOI environment. The primary users of the IC Fab system are identified as the Fab owner and the maintainer, who interact with the system to trigger the IC fabrication processes. The IC Fab system domain is comprised of the IC Fab processing sub-elements. The sub-elements are modeled based on their functionality and resource groups. The wafers to be processed and the FOUP's used to transport the wafers around the Fab constitute the system environment, as they are external systems that interact with the SOI.

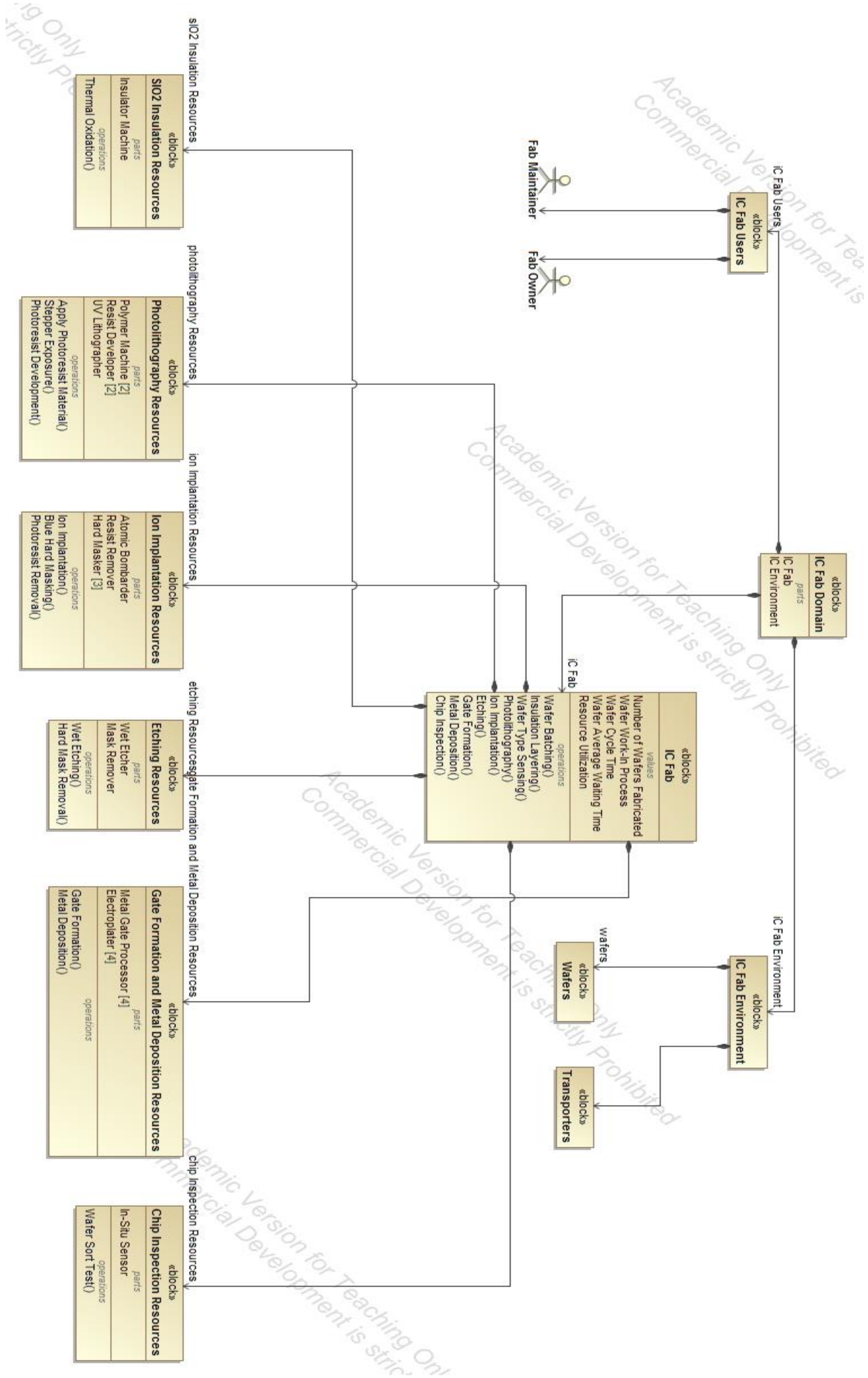

Figure 4.1 IC Fab System Block Definition Diagram

System Activity Diagram

An activity diagram is a kind of behavior diagram that provides a dynamic view of the system that expresses sequences of behaviors and event occurrences over time (Delligatti, 2013). The IC Fab activity diagram models a set of sequential actions that occur during the IC fabrication process. The Fab activity diagram comprises of swim lanes that organizes the model based on the high-level processes that are performed during the fabrication process. The model illustrates how the flow of activities occurs between different stations and what actions trigger this flow. Further, decision gates were used at different stages of the model to depict the wafer flow in different directions of the model based on some defined condition. This architecture was essentially useful to identify, design and analyze process routes and condition-based decisions to be taken at each and very processing stage of the IC Fab model.

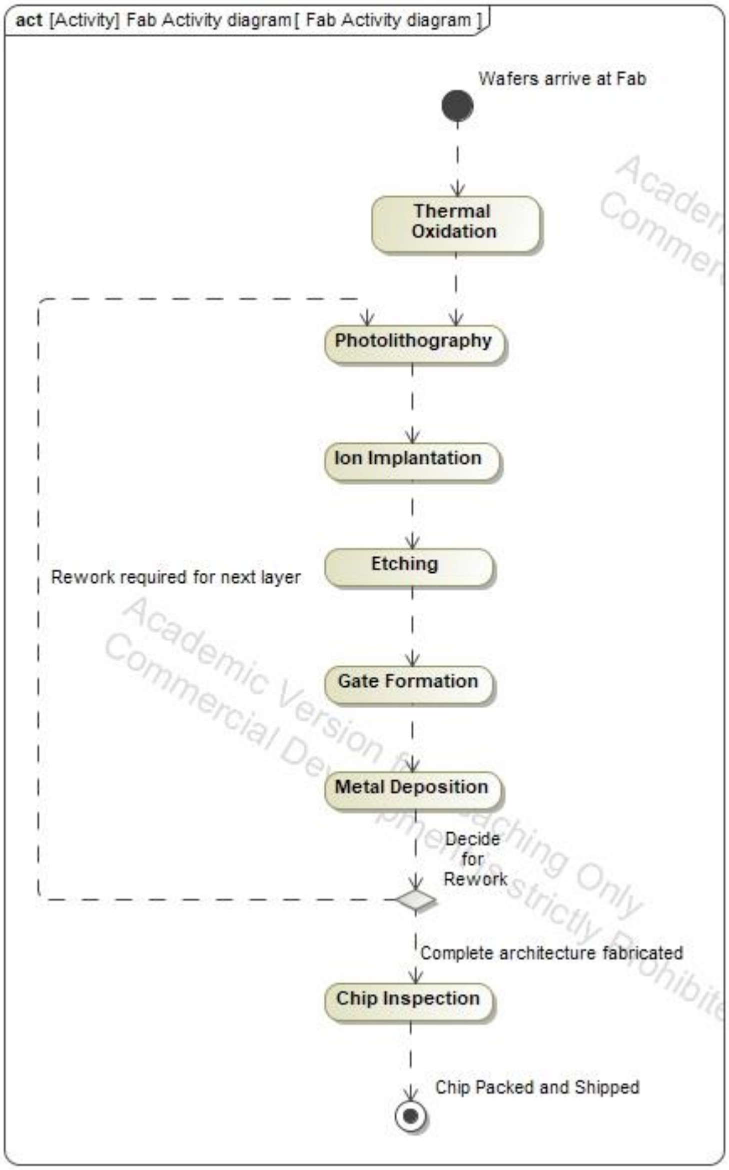

Figure 4.2 IC Fab System High-Level Activity Diagram

Discrete-Event Simulation (DES)

DES Overview and Simulation Software

Discrete-event simulation (DES) models the operation of a system as a discrete sequence of events in time. Each event occurs at a particular instant in time and marks a change of state in the system (Robinson, 2004). Both the nature of the state change and the time at which the change occurs mandate precise description.

Arena is a discrete-event simulator that makes modeling easy and also provides flexibility at the lowest level simulation modules using the SIMAN simulation language. These functionalities and ease of use makes Arena the primary simulation tool used of this study, with MATLAB and Excel being used as secondary tools for specific computations and data analysis.

Arena Random Number Generator (RNG) and Statistical Analyzer

One of the important aspects to be considered while developing simulation models is the randomness in the input to ensure that the model is a representation of reality and can be subject to several uncertain events. Hence, it is important as a simulator to have the knowledge of the probability distributions from which the observations would be generated, and the random number generator (RNG) used to calculate the values of each of the observation.

The old version of Arena used an Linear Congruential Generator (LCG) with $m = 2^{31} - 1, a = 7^5 and\ c = 0$; the cycle length of it being 2.1 billion, whereas the updated versions of Arena (14 and newer) use a new RNG, which uses the same ideas as LCG, but involve two separate component generators that are later combined, and the recursion to get to the next value looks back beyond just the single preceding value.

Arena new RNG (Kelton, Sadowski, & Swets, 2010):

- The two separate recursions

  $A_n = (1403580A_{n-2} - 810728A_{n-3})\ mod\ 4294967087$

  $B_n = (527612B_{n-1} - 1370589B_{n-3})\ mod\ 4294944443$

- The program then combines these two values at the nth step

  $Z_n = (A_n - B_n)\ mod\ 4294967087$

- Finally, the RNG delivers the nth random number:

$$U_n = \frac{Z_n}{4294967088}, for\ Z_n > 0$$

Statistical Analyzers

Input Analyzer

The Input Analyzer is a standard tool built-in Arena that is designed to fit distributions to the observed data, provide estimates of their parameters, and measure how they fir the data. The thesis uses the Arena Input Analyzer to decide the best theoretical probability distributions to be used to generate sample data based on mathematical formulation.

Output Analyzer

The Output Analyzer is a separate application that is a part of Arena that is used to analyze the results obtained from the Arena simulation. The thesis uses this application to plot curves, compare means and calculate the confidence intervals for the data.

Arrival Process Distribution

Arrival process for a system characterizes the input sources to the simulation model. Arrival process for the fab system was calculated based on the approximate data values obtained from online research followed by identifying the best theoretical distribution to represent the arrival process.

The parameter for the exponential distribution is the mean(β) and the probability density function is calculated as (Kelton, Sadowski, & Swets, 2010):

$$f(x) = \begin{cases} \frac{1}{\beta} e^{-\frac{x}{\beta}} & for\ x > 0 \\ 0 & otherwise \end{cases}$$

Arrival process for the fab is characterized by two important elements, namely the interarrival times and the arrival process probability distribution. The exponential distribution was selected to generate data observations for the entity arrival process and hence drive the simulation. Exponential distribution was selected, because it being a continuous theoretical distribution, models time between independent events or the interarrival times for machine part arrivals in manufacturing systems (Kelton, Sadowski, & Swets, 2010). The mean time between arrivals for the Skylake wafer is approximated to 0.38 hours and Kabylake wafer to 0.3 hours based on the data obtained (Lapedus, 2017).

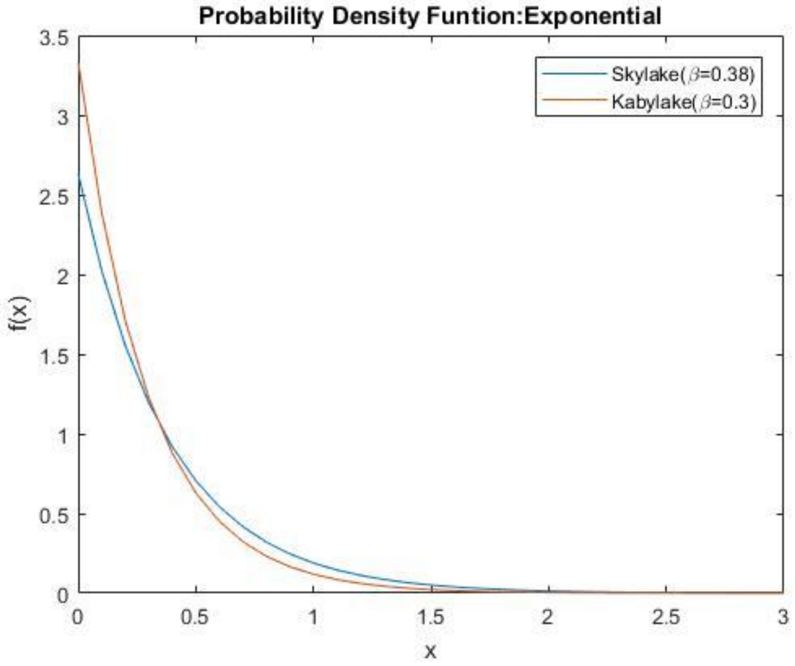

Figure 4.3 Exponential Interarrival Times

Service Process Distribution

Service process for a system characterizes the entity processing time distribution at each station. The service process at a processing station is featured by the number of servers and the service time characteristic or the probability distribution. The service process distribution was generated by obtaining data from various online sources and was fitted into a triangular distribution using the data obtained from sources referred in Chapter 2.

The parameters for the triangular distribution are the minimum (a), mode(m), and the maximum (b). The probability density function is calculated as follows (Kelton, Sadowski, & Swets, 2010):

$$f(x) = \begin{cases} \dfrac{2(x-a)}{(m-a)(b-a)} & for\ a \le x \le m \\ \dfrac{2(b-x)}{(b-m)(b-a)} & for\ m \le x \le b \\ 0 & otherwise \end{cases}$$

Most of the service processing distributions in the fab model were generated using a triangular distribution because the data sets obtained were in the form of minimum, maximum and the most likely or modal values. Further, it has the advantage of allowing a non-symmetric distribution of values around the most likely, which is commonly encountered in real manufacturing processes. On the other hand, it is also a bounded distribution, and hence erroneous machine activities, outside the minimum and maximum values, which are very rare in semiconductor manufacturing fabs, are not encountered in the model.

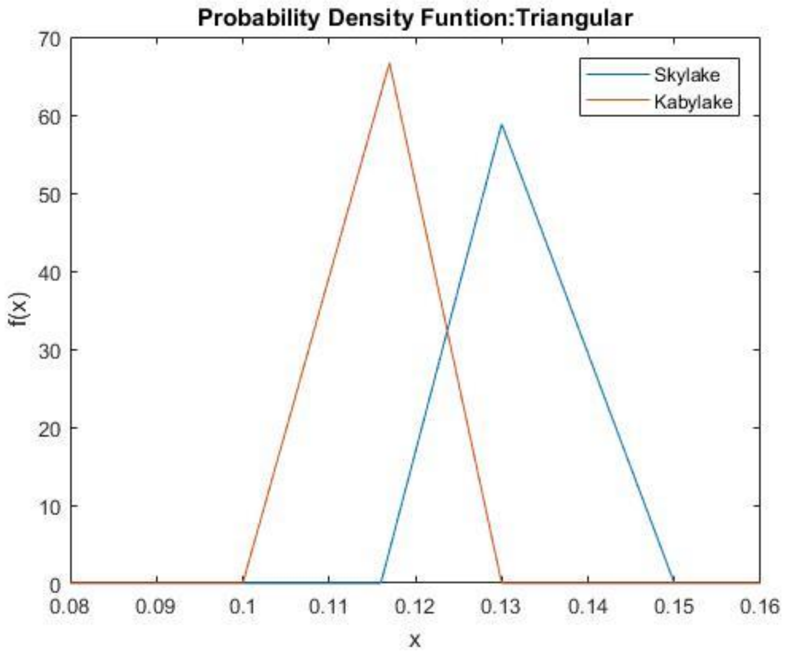

Figure 4.4 Polymer Machine Service Time Distribution

Little’s Law

Little’s law is an important concept in queueing theory for its simplicity and generality. Little’s law states that, under steady state conditions, the average number of items in a queueing system equals the average rate at which items arrive multiplied by the average time an item spends in the system (Little & Graves, 2008). It is mathematically expressed as

$$L = \lambda * W$$

where,

L =average number of items in the queueing system,

W = average waiting time in the system for an item,

λ =average number of items arriving per unit time

This law relates to the fundamental quantities in manufacturing process. The Work-In Process (WIP) in a stable system can be expressed as a product of throughput (T) multiplied by the average cycle time (CT). This relation is used to study the effects of varying WIP’s (pull system) at different simulation times on the measured cycle times.

$$WIP = T * CT$$

Dispatching rules (FIFO, SPT, Priority Based)

In production systems, the key task of implementing Lean Principles lies in understanding how parts flow between each workstation and what dispatching rule decides which part to be processed first. This helps production planners determine which parts flow easily, which parts need direction and which sections need the highest level of control. The thesis models the fab manufacturing process using 3 different dispatching rules namely: First In First Out (FIFO), Shortest Processing Times (SPT) and Priority Based (PB).

FIFO Model

The first-in first-out (FIFO)queue dispatching rule is the most basic dispatching rule where entities are processed on a first-come first-served orderly basis. The goal of FIFO is to prevent earlier orders from being delayed in favor of newer orders, which would result in an increased lead time and delay. The base model of the thesis uses FIFO dispatching rule at the various processing stations. Since there are two different entities (Skylake and Kabylake) being fabricated in the fab, and the same machines processing them, the first one to arrive at a processing station gets served first followed by the next in queue, when the first one exits the processing workstation.

SPT Model

The shortest processing time dispatching rule processes the entities in the order of increasing processing times. The dispatching rule uses the algorithm of identifying the shortest assigned processing time among the two entities being fabricated and places the entity with the shortest assigned processing time to the start of the queue at a particular instance in the simulation time. The major goal of applying SPT to a fabrication model is to minimize the average cycle time.

Priority-Based Model

The priority-based dispatching rule processes entities in the order of the predefined priority for an entity. The model has two different entities (Skylake and Kabylake wafers) being fabricated at the same time; hence, several factors go into deciding which product to prioritize over the other. The dispatching rule uses the algorithm of assign a numerical priority value to each part waiting in the queue of a machine group and to select for processing next, the one with the minimum or maximum value of the priority index. Priority-based dispatching rule being a heuristic algorithm has an aim to achieve an acceptable solution but not the accurate solution.

To assign the higher priority amongst the two wafers, the method weighting was used. In this procedure, weights are assigned on a scale of 1-10 for chip production factors, namely; chip revenue, chip rework, chip processing time, wafer production waste generated, customer popularity and defects. Based on the weights assigned to these wafers, the total weighted score is calculated. The scoring analysis reveals that Kabylake has higher priority over Skylake. (Appendix B, Page 88)

Arena IC Fab Model

The purpose of this section is to walk the reader through the procedure used for developing the 4 models that were used for analyzing the performance of the IC Fab. The FIFO model forms the basic model, upon which the SPT, Priority Based and Failure FIFO models are developed.

Model Performance Measures

The performance of a model depends upon the performance measures or the model measures of effectiveness. The two-important metrics that were measured as a part of the thesis were the cycle time and the average waiting time. These metrics were evaluated over the 4 IC Fab models that were developed using Arena. The four factor models whose impacts on the two-metric's studied are:

a. Base Model (FIFO)

b. SPT Model

c. PB Model (Priority Based)

d. Failure FIFO (Machine Failures in FIFO)

Model Algorithm

The following sections discuss the components and characteristics of the model.

Batching

The entities are carried around in the fab using FOUPS in batches. Batching of entities was done for the stations where the processing takes place in batches and not on individual wafers. The entities keep arriving at the FOUP batching station until the batch size is reached, after which they proceed as a batch to the processing stations. The stations where the entities were batched for processing are shown in the table below:

Processing Station	**Number of Entities/Batch**
Insulator Machine	25
Atomic Bombarder	5
Wet Etcher	25
In-Situ Sensor	5

Table 4.1 Batching Stations and Number of Entities per Batch

A separate batch of predefined batch size is formed differently for both the Skylake wafer and the Kabylake wafer. Arena provides two types of batching options, namely the temporary batch and the permanent batch. The thesis uses the temporary type batching, because the IC fabrication process is a combination of batching and single wafer processing stations and hence involves batching and separating entities depending upon the processing strategy at each station.

Entity Attribute

This thesis models the fabrication process of two entities, namely the Skylake wafer and the Kabylake wafer. The chip design and architecture, as discussed in Chapter 1, are clear indications of the fact that both the entities have different processing times at each workstation, which is defined by their level of complexity and architecture. Hence, processing times are generated from a triangular distribution and assigned for each of the entities before entering a workstation.

Processing Station

A processing station is the workstation where the entities are worked upon for an assigned period of time. The processing stations first seize the singular entity or batch to be processed and delays the simulation clock for the time assigned previously using the triangular distribution. Once the simulation clock advances to the allocated value, the entity is released from the resource, so that other entities can seize it. Each of the processing work-stations are characterized by a set of resources to process entities, and the number of resources is anywhere between 1-4 resources per workstation.

Entity Type Decide Logic

The entities that exit the processing station enter a decide module. The decide module is an abstraction of the sensors that are used by the fab to bifurcate the entity type before it enters the next processing station. The decide module is programmed to allocate routes to the individual entities to the next processing station.

Repeat Logic

The repeat logic is modeled at the end of the gate formation and metal deposition processing module. The purpose of the repeat logic is to count the number of passes and automatically route the entities to the inspection station or back to the photolithography station and through all the stations following it. As discussed in Chapter 2, several layers are built over one another, and one layer is fabricated per cycle. Due to the complexity and extensive functionality of the Skylake wafer, the repeat logic is programmed to route Skylake wafer entities thrice through the stations starting from photolithography. On the other hand, the Kabylake wafer entities are sent back only twice through these stations. Once the entities complete their required level of fabrication, a decide module routes them to the inspection station.

Push System

The modeling approach considered is of a push system, wherein the work-in process at the shop floor is not controlled. The reason behind modeling the fab as a push system was to avoid excessive back locks on orders that wait to start processing. A pull system can also be used to model the fab but requires advance production techniques knowledge and hence is beyond the scope of this study.

Resource Levels and Allocation

Resource levels at each station through the fabrication process have a major impact on the fab performance measures. Developing the basic FIFO model involved making decisions of the required level of resources at each station to attain maximum utilization and minimal waiting of entities for processing. Once the required number of resources were assigned to each station, a resource allocation rule was developed. The random (equiprobable) resource allocation rule was selected, so that entities select available resources randomly. The resource levels at each station are described in the following table:

Workstation	Number of Machines
Insulator Machine	1
Polymer Machine	2
UV Lithographer	1
Resist Developer	2
Atomic Bombarder	1
Resist Remover	1

Hard Masker	3
Wet Etcher	1
Mask Remover	1
Metal Gate Processor	4
Electroplater	4
InSitu Sensor	1

Table 4.2 Resource Levels at Processing Stations

It can be observed from the resource allocation table that some of the workstations have more than one resource. This is because these are single wafer workstations, which are preceded by batch workstations, where wafer processing is done in batches. Hence, a load of 5 to 25 wafers arrives at these workstations, which requires more than one resource for processing to avoid excessive work-in process and back locks.

Resource Failures

Failures are primarily intended to model events that cause the resource to become unavailable for a period of time. Every manufacturing facility experiences machine failures, either scheduled or unscheduled. Typically, when a machine failure occurs, the machine is down/non-operational, followed by the machine undergoing repair before becoming operational again. Failures in Arena can be modeled as either count based, or time based. The model uses the time-based algorithm, as data was obtained in the form of mean time between failures (MTBF) and mean time to repair (MTTR), which were modeled as exponential distributions. The machine MTBF and MTTR data is provided in the table below:

Resource	**MTBF (days)**	**MTTR(hours)**
Mask Remover Failure	EXPO (70)	EXPO (3.5)
Bombarder Failure	EXPO (47)	EXPO (5)
Hard Masker Failure	EXPO (150)	EXPO (8)
Resist Developer Failure	EXPO (120)	EXPO (7)
UV Lithographer Failure	EXPO (73)	EXPO (3)

Table 4.3 Resource Failures in terms of MTBF and MTTR

Further, Arena provides three rules to model failures, namely, Ignore, Wait and Preempt. The thesis uses the preempt rule wherein the resource stops the processing of the entities and resumes operation after the down time duration. The ongoing processes for the station are interrupted immediately when the simulation clock enters the failure mode and resume when the resource becomes available and the clock exits the failure.

Warm-up Period

The IC Fab model starts out empty of entities and all resources are idle. So, if a model starts empty and idle, where entities eventually become congested, the output data for some period of time after initialization understates eventual congestion (Kelton, Sadowski, & Swets, 2010). In other words, the queues at resources get congested eventually, and hence the results might experience a low bias. To avoid this initialization bias, and to make the simulation more realistic and steady state, the IC Fab model was given a warm-up period. The warm-up period runs the simulation for the entire defined run but starts collecting statistics after the end of the warm-up period.

The challenge with modeling the initialization bias for the model was to decide the how to throw some entities around in the model and at what time. This problem was approached by using the half-width smallness criterion and removing all other replication and simulation stopping criterion. The Arena time-persistent output statistic DSTAT was used and a simulation stopping condition of [DHALF(Total_WIP_FIFO) <1] was used with a confidence interval of 95%. Th results of the simulation were obtained as follows.

Time-Persistent Statistic	Run Length (days)	Average	Half-Width	Minimum Value	Maximum Value
Total WIP	43	88.16	0.94	0	145

Table 4.4 Warm-Up Period Simulation Result

Hence, the FIFO and all the models that were subsequently built on FIFO were modeled with a warm-up period of 43 days. Further remedy to initialization bias were running the model for a long period of time (43+365 =408 days) .

Chapter 5: Simulation Results and Analysis

Purpose

The goal of a simulation study is to implement a model in a specific environment that allows the model's execution over time. As discussed in Chapter 4, the base model for the fab is developed as FIFO, upon which SPT, Priority Based and Failure FIFO models are developed. The results and statistics are presented and analyzed in this section. The simulation output analysis was performed on the Arena Output Analyzer and Excel (data exported from Arena).

Cycle Time Analysis

The average cycle time for 10 replications of the 3 models (SPT, PB and Failure FIFO) are compared to the base model (FIFO).

FIFO vs SPT

The average cycle time analysis result between FIFO and SPT is shown below. It can be observed from the clustered column graph that the average cycle time for the Skylake FIFO is much less than that for the Skylake SPT, while on the other hand, Kabylake SPT has a smaller average cycle time than Kabylake FIFO model. Technically, the SPT is supposed to perform better than the FIFO, but since we have two wafers being processed at the same time, and Kabylake having lower processing time compared to Skylake, Kabylake entities tend to be pushed ahead in the queue near the processing station and the Skylake entities tend to be pushed behind in the queue. Hence, on an average the cycle time increases for the Skylake wafer and decreases for the Kabylake wafer when the FIFO model is converted to an SPT model.

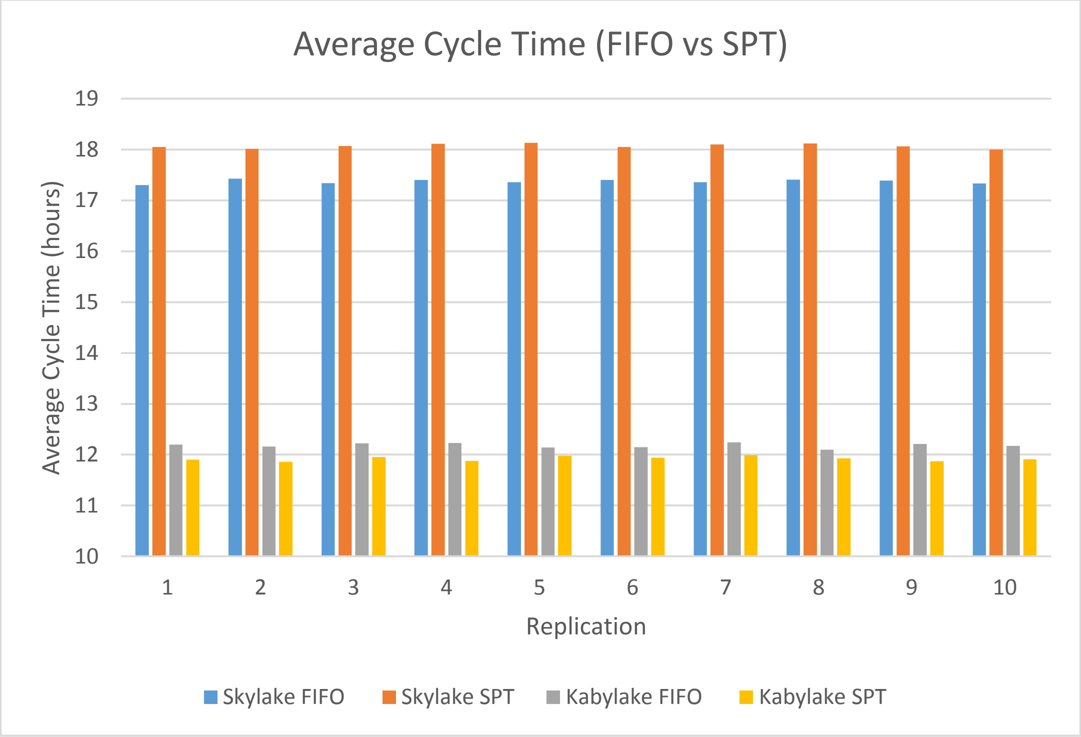

Figure 5.1 FIFO vs SPT Average Cycle Time

FIFO vs PB

The clustered graph below provides a comparative study between the average cycle time for FIFO and PB. It can be observed from the graph that Skylake FIFO has a much lower average cycle time compared to the Skylake PB model, while Kabylake PB has a smaller average cycle time than Kabylake FIFO model. This can be explained by the fact that Kabylake is the prioritized wafer, and hence Kabylake entities are pushed to the start of the processing station queue, hence reducing the overall average cycle time. On the other hand, Skylake entities are pushed back in the queue, which results in an increased time in the fab, and hence an increased average cycle time.

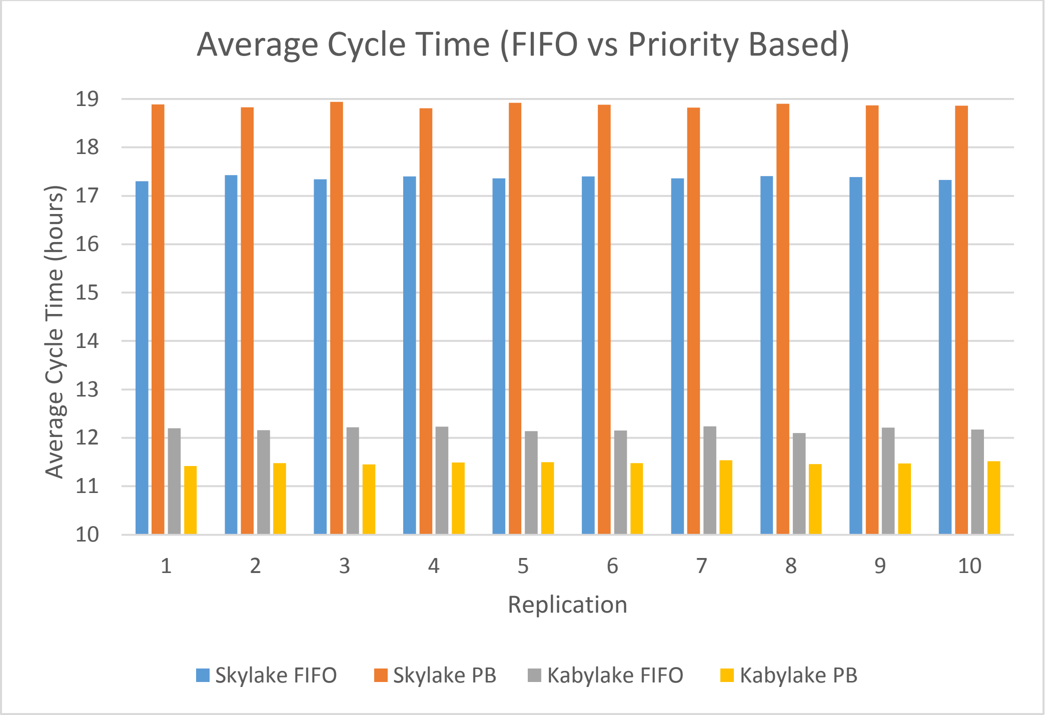

Figure 5.2 FIFO vs PB Average Cycle Time

FIFO vs Failure FIFO

The clustered graph below presents a comparative study between the average cycle time for the FIFO and Failure FIFO. It can be observed from the curve that the average cycle time for both the Skylake and Kabylake FIFO is much less than that for the Skylake and Kabylake Failure FIFO. This result can be associated to the fact that failures are modeled as preemptive; hence, when a machine fails, processing is stopped, and long queues are formed at these stations, leading to an increase in the overall cycle time for the entities.

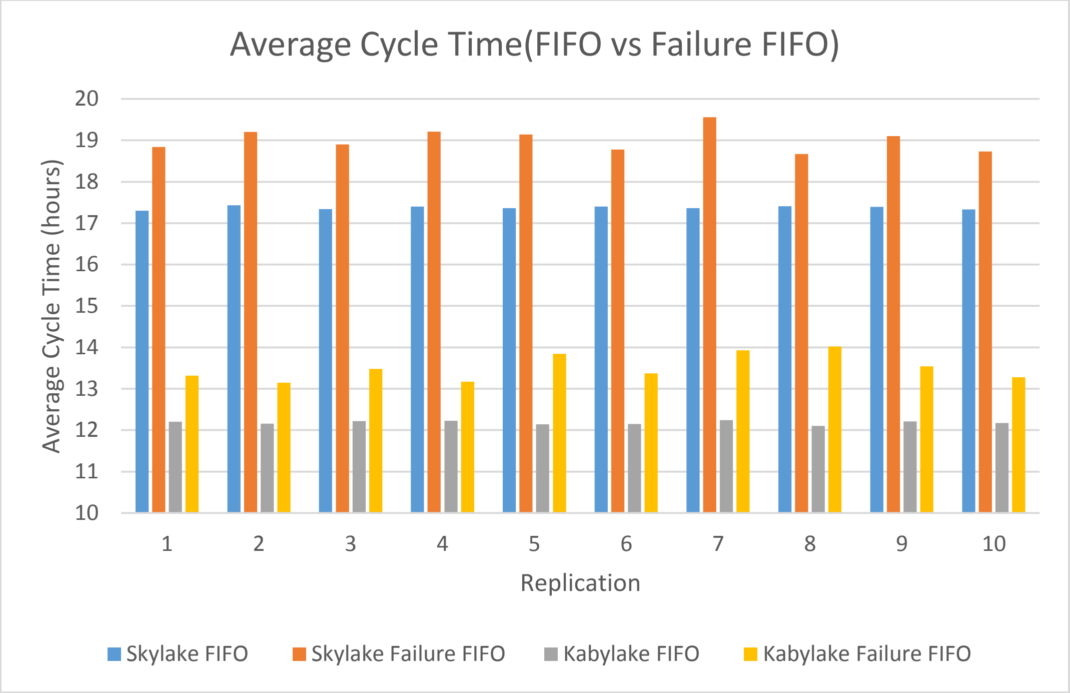

Figure 5.3 FIFO vs FIFO Failure Average Cycle Time

Average Waiting Time Analysis

Average waiting time

One of the important output performance measures for a fabrication process is the mean time spent by an entity at a processing station. This performance measure provides inference about the efficiency of the queue model formed at processing stations as a factor of the queueing sequences and machine failures at these stations. This metric is plotted against the arrival times of the entities, and this curve helps identify the simulation time at which the simulation becomes stable. The curves following this section provide an analytical study of the effects of queueing sequences and machine failures on the average waiting time for each entity and its comparison against the base model (FIFO).

FIFO vs SPT

The plot shown below provides a comparative study between the average waiting time and the arrival time for the Skylake and Kabylake FIFO vs SPT. It can be observed from the curve that average waiting time for the Skylake FIFO (red-line) is lesser as compared to that of the Skylake SPT (black-line), while average waiting time for the Kabylake FIFO (green-line) is higher as compared to that of the Kabylake SPT (blue-line). It can also be observed that the process simulation stabilizes after 80 days from the start of the simulation time and 37 days from the warm-up period. This model graphical output can be verified with the expected model behavior, as the Kabylake wafer entities have a shorter processing time, and hence Skylake entities tend to be pushed behind in the queue by the SPT logic programmed at the stations, which leads to higher waiting times for Skylake entities compared to the Kabylake entities.

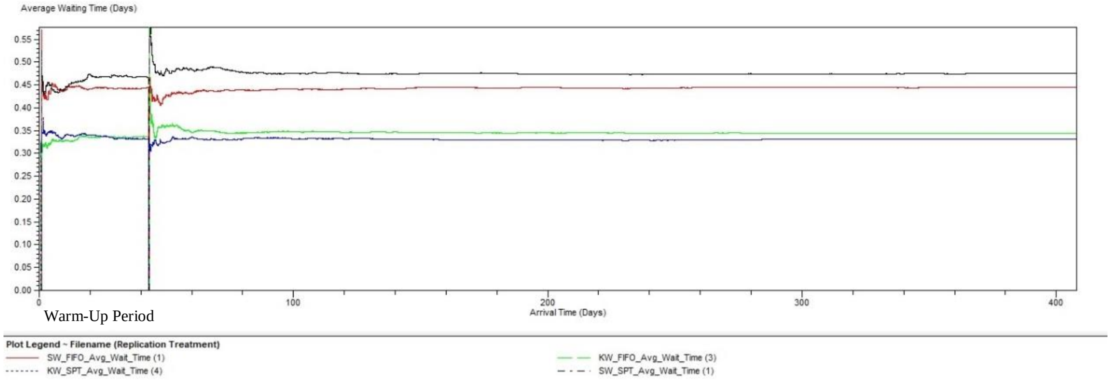

Figure 5.4 FIFO vs SPT Average Waiting Time

FIFO vs PB

The plot below provides a comparative study between the average waiting time vs arrival time for the Skylake and Kabylake FIFO vs PB. It can be observed from the curve that the average waiting time for the Skylake PB (blue-line) is higher as compared to the Skylake FIFO (red-line), while average waiting time for the Kabylake FIFO (green-line) is higher as compared to that of the Kabylake SPT (black-line). It can also be observed that the process simulation stabilizes after almost 60 days from the start of the simulation and 17 days from the end of the warm-up period. This model output can be verified with the expected model behavior, as the Kabylake wafer entities are assigned a higher priority over Skylake entities and hence Skylake entities, are pushed behind in the queue by the PB logic programmed at the stations.

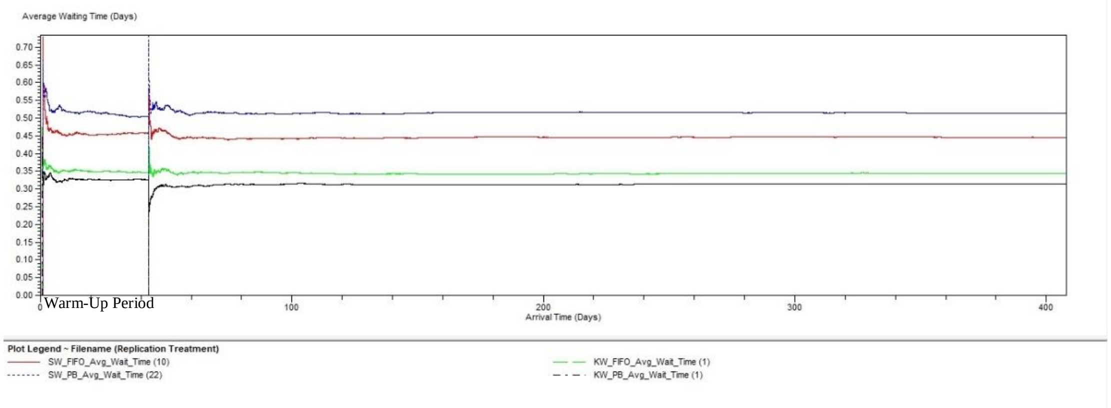

Figure 5.5 FIFO vs PB Average Waiting Time

FIFO vs FIFO Failure

The plot below provides a comparative study between the average waiting time vs arrival time for the Skylake and Kabylake FIFO vs FIFO Failure. It can be observed from the curve that the average waiting time for the Skylake FIFO Failure (blue-line) and Kabylake FIFO Failure (black-line) are higher as compared to the Skylake FIFO (red-line) and Kabylake FIFO (green-line), respectively. It can also be observed that the process simulation stabilizes after almost 55 days from the start of the simulation and 12 days from the end of the warm-up period. This model output can be verified with the expected model behavior, as the machine failures at various stations during the simulation run, being modeled as preemptive, ceases the resource operation which leads to large queue formations at these stations and hence larger waiting times.

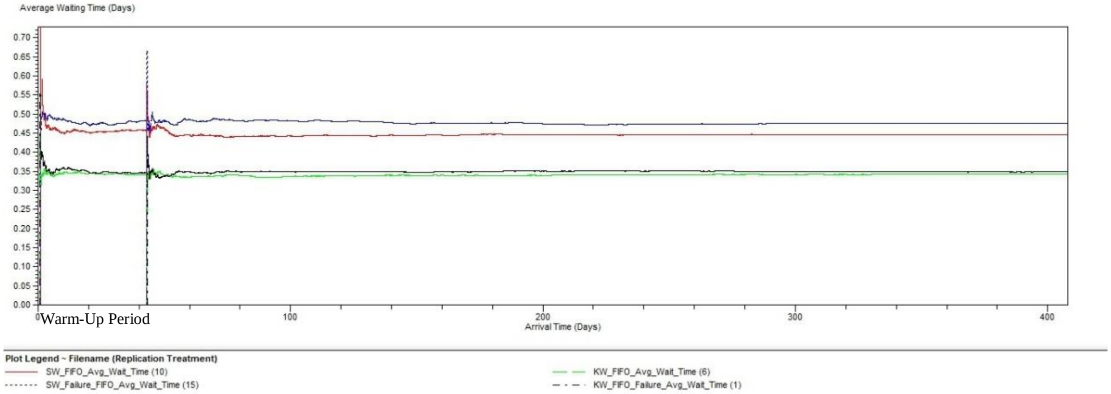

Figure 5.6 FIFO vs FIFO Failure Average Waiting Time

Statistical Output Analysis

Every simulation model is built with the intent to provide the best instantiation the real process/system. Since random samples from probability distributions are typically used to drive a simulation model through time, these estimates are just particular realizations of random variables that may have large variances. As a result, these estimates could, in a particular simulation run, differ greatly from the corresponding true characteristics for the model (Law, 2010). This thesis uses

ANOVA and Tukey post-hoc analysis test to study the statistical significance of the four models built using Arena.

ANOVA Test

Analysis of variance (ANOVA) is a statistical method that is used to analyze the difference between among group means. ANOVA was used to compare the 4 models for statistical significance which is measured as a factor of the p-value and the F-values compared to the threshold and critical values, respectively. The ANOVA test uses hypothesis-based testing to the analyze the statistical results. The null (H_0) and the alternative (H_A) hypothesis for the 4 models are defined below:

$$H_0: Model\ Type\ has\ no\ significant\ effect\ on\ the\ cycle\ time$$

$$H_A: Model\ Type\ has\ a\ significant\ effect\ on\ the\ cycle\ time$$

One-way ANOVA is performed at 95% confidence level for the 4 models to observe effects on cycle time for individual wafers (Skylake and Kabylake). 500 samples of cycle time from each model group were collected for analysis.

Skylake ANOVA Analysis

The ANOVA analysis was performed for the cycle time values obtained for the Skylake wafer from the 4 models developed on Arena. The results of the ANOVA analysis are shown below.

ANOVA: Single Factor			
Summary			
Groups	*Count*	*Sum*	*Variance*
Cycle Time FIFO	500	10050.42979	16.15543946
Cycle Time SPT	500	11906.71653	43.89846558
Cycle Time PB	500	10680.66323	16.23457269
Cycle Time Failure FIFO	500	10700.61588	16.50301961

ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	3612.02	3	1204.03	**51.92**	$\mathbf{2.68 * E^{-32}}$	**2.61**
Within Groups	46302.95	1996	23.24			
Total	49914.96	1999				

Table 5.1 Skylake ANOVA Analysis

It can be observed from the ANOVA analysis that the p-value is much lower than the threshold (0.05) value for 95% confidence. Also, the F-critical value is smaller than the calculated F value, which implies that there is a significant effect on the cycle time based on the model type. Hence, we reject the null hypothesis and conclude that “Model type has a significant effect on the cycle time” for Skylake wafer.

Kabylake ANOVA Analysis

The ANOVA analysis was performed for the cycle time values obtained for the Kabylake wafer from the 4 models developed on Arena. The results of the ANOVA analysis are shown below.

ANOVA: Single Factor			
Summary			
Groups	*Count*	*Sum*	*Variance*
Cycle Time FIFO	500	7105.928255	11.1911937
Cycle Time SPT	500	6815.738208	11.4076661
Cycle Time PB	500	6652.210187	7.77764276
Cycle Time Failure FIFO	500	7452.622217	10.2646881

ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	741.64	3	247.21	$\mathbf{24.33}$	$\mathbf{1.84 * E^{-15}}$	$\mathbf{2.61}$
Within Groups	20279.95	1996	10.16			
Total	21021.59	1999				

Table 5.2 Kabylake ANOVA Analysis

It can be observed from the ANOVA analysis that the p-value is much lower than the threshold (0.05) value for 95% confidence. Also, the F-critical value is smaller than the calculated F value, which implies that there is a significant effect on the cycle time based on the model type. Hence, we reject the null hypothesis and conclude that “Model type has a significant effect on the cycle time” Kabylake wafer.

Tukey Test

Tukey’s honest significance difference test is a post hoc test that is based on studentized range distribution. It is an extension to a ANOVA test, as the ANOVA test tells if the results are significant overall but does not tell exactly where the differences lie. The Tukey test is used to compare the significant difference between the 4 models as a factor of the q-value (studentized range statistic) compared to the critical q-value. The equation for the q-value is shown below:

$$q = \frac{(M_i - M_j)}{\sqrt{(\frac{MS_w}{n})}}$$

Where,

$M_i: Mean\ for\ Group\ 1$

$M_j: Mean\ for\ Group\ 2$

$MS_w: Total\ mean\ square\ within\ groups\ obtained\ from\ ANOVA$

$n: Number\ of\ Samples$

Skylake Tukey Test

The Tukey test was performed simultaneously to the set of all pairwise comparisons of the 4 models from the Skylake ANOVA test results that were obtained previously. The pairwise results from the Tukey test helped determine which groups among the 4 group differ significantly. The results of the Tukey test for Skylake wafer cycle time is shown in the table below:

Post Hoc Analysis (Tukey)								
Possibilities		**Mean Diff**	**n(Grp1)**	**n(Grp2)**	**SE**	**q**	**q-critical**	**Decision**
FIFO	SPT	3.71	500	500	0.22	17.24	3.63	**Reject Null**
FIFO	PB	1.26	500	500	0.22	5.85	3.63	**Reject Null**
FIFO	Failure FIFO	1.31	500	500	0.22	6.04	3.63	**Reject Null**

SPT	PB	2.45	500	500	0.22	11.38	3.63	**Reject Null**
SPT	Failure FIFO	2.41	500	500	0.22	11.19	3.63	**Reject Null**
PB	Failure FIFO	0.039	500	500	0.22	0.18	3.63	**Fail to Reject Null**

Table 5.3 Skylake Tukey Test

It can be observed from the Tukey table that the q value is greater than the q-critical value for all the model comparisons except the PB vs Failure FIFO comparison model. This pairwise comparison infers that the cycle times for the Skylake wafer in the models differ significantly except for the PB and Failure FIFO models. In other words, PB and Failure FIFO models are closely related and do not have a significant difference in the cycle times.

Kabylake Tukey Test

The Tukey test was performed simultaneously to set of all pairwise comparisons of the 4 models from the Kabylake ANOVA test results that were obtained previously. The pairwise results from the Tukey test helped determine which groups among the 4 group differ significantly. The results of the Tukey test for Kabylake wafer cycle time is shown in the table below:

Post Hoc Analysis (Tukey)								
Possibilities		**Mean Diff**	**n(Grp1)**	**n(Grp2)**	**SE**	**q**	**q-critical**	**Decision**
FIFO	SPT	0.58	500	500	0.14	4.07	3.63	**Reject Null**
FIFO	PB	0.91	500	500	0.14	6.36	3.63	**Reject Null**

FIFO	Failure FIFO	0.69	500	500	0.14	4.86	3.63	**Reject Null**
SPT	PB	0.33	500	500	0.14	2.29	3.63	**Fail to Reject Null**
SPT	Failure FIFO	1.27	500	500	0.14	8.93	3.63	**Reject Null**
PB	Failure FIFO	1.61	500	500	0.14	11.23	3.63	**Reject Null**

Table 5.4 Kabylake Tukey Test

It can be observed from the Tukey table that the q value is greater than the q-critical value for all the model comparisons except the SPT vs PB comparison model. This pairwise comparison infers that the cycle times for the Kabylake wafer in the models differ significantly except for the SPT and PB models. This result can be related to the similar model behavior of the SPT and the PB for a Kabylake wafer. Since Kabylake wafer is the prioritized wafer and has shorter processing times compared to the Skylake, Kabylake wafers tend to be pushed ahead in the queue at the processing stations for both the models.

Chapter 6: Conclusion and Future Scope

In this dissertation the effects of different scheduling policies on the cycle time for manufacturing two types of wafers was studied. It was observed that scheduling policies have a drastic effect on the cycle time, both negative and positive depending on the wafer type.

The cycle time analysis results from Section 5 reveal that sequencing rules SPT and PB have a negative impact on the cycle times for the Skylake wafer, while on the other hand have a positive impact on the cycle times of the Kabylake wafer. Skylake wafers experienced an increase in the average cycle time for the SPT, PB and the Failure FIFO models when compared to the FIFO model. Kabylake wafers experienced a decrease in the cycle time for the SPT and PB scheduling policies, and an increase in the cycle time for the Failure FIFO model.

The average waiting time curves also demonstrate that the scheduling policies influence the average waiting time for each wafer. It was found that the average waiting times for the Skylake entities increased drastically for the SPT and the PB scheduling policies, but moderately for Failure FIFO. For the Kabylake wafer the average waiting time decreases drastically for the SPT and the PB scheduling policies and increases moderately for the Failure FIFO model.

The model comparison using the ANOVA and the Tukey Test confirms that the means of the four models are statistically different from one another at a 95% confidence level and hence model type has a statistically significant effect on the cycle time. Post-hoc model comparison also reveals that the cycle time samples of Skylake entities for the PB model are not statistically significantly different from that of the Failure FIFO model. As far as the Kabylake entities are concerned, post hoc analysis suggests that the SPT and the PB models do not have statistically significant different cycle times.

Based on the results that were obtained from the study, it can be concluded that the models did verify their expected behavior. Considering the practical manufacturing process scenario of the semiconductor industry, the dissertation models and the simulations at their best capability encapsulate the real IC fabrication process, given the accuracy of the data that was available. The data used to model the fabrication process was obtained from random distributions; hence the models could have been more accurate if access to real data were available, or if real data were fitted to a probability distribution.

Since these models instantiate the fabrication process of two entities at one time in a representative fab, these models could be used in the industry to simulate the fabrication process by modifying the model components to match the architecture of the fab. Several inferences like effects of scheduling and machine failures on cycle times can be made using these models. Also, model type effects on machine utilization, throughput, lot sizing, work-in process could be studied in detail using these models.

Future Research

Constant WIP

The scope of the thesis does not include the concept of a Con-WIP model or a pull system to control the work in process. The literature review in Chapter 3 discusses the benefits of controlling the WIP and a make-to-order system. This concept can be implemented in the current fab representative model, and the effects on cycle time due to various factors could be analyzed. Modeling the fab as a Con-WIP system would involve assigning cards to each wafer to track and control the WIP at each time step during the simulation run. Further research could also be extended to modeling the fab as a complex Kanban system where the cards are related to the certain wafer type and the quantity in contrast to the Con-WIP system where cards are only related to the quantity. These concepts are known to promote a lean production and prevent overloading of the system and overproduction.

Profit/Cost Analysis

The scope of this thesis does not analyze the cost variables associated with the IC manufacturing process. Cost can be associated to the two wafers being manufactured, which can help the management decide the number of each wafer to be produced. While introducing new resources, a cost study can be performed to calculate the duration for the return on investment. Also, other costs that affect the production targets such as material holding cost, revenue loss due to delay could also enhance the models use for process optimization.

Other Queueing Sequences

The scope of the thesis only encompasses FIFO, SPT and Priority-Based queueing sequences for cycle time comparison. Other queueing sequences that should be considered include earliest due dates (EDD), manufacturing slack time remaining (STR = time remaining before due date-remaining processing time), longest expected processing time (LTNV), etc.

Appendices

Appendix A: SIMAN Code for FIFO model

```
;
;     Model statements for module:  BasicProcess.Create 3 (Skylake Wafer)
;

73$           CREATE,
1,HoursToBaseTime(0.0),SW:HoursToBaseTime(EXPO(0.38)):NEXT(74$);

74$           ASSIGN:        Skylake Wafer.NumberOut=Skylake Wafer.NumberOut
+ 1:NEXT(2$);


;
;
;     Model statements for module:  BasicProcess.Assign 57 (Cycle Time
Skylake)
;
2$            ASSIGN:        Cycle Time_SW=TNOW:NEXT(0$);


;
;
;     Model statements for module:  BasicProcess.Batch 3 (Skylake Wafer
Batch_IL)
;
0$            QUEUE,         Skylake Wafer Batch_IL.Queue;
77$           GROUP,         Skylake Batch,Temporary:25,Last,SW:NEXT(78$);

78$           ASSIGN:        Skylake Wafer Batch_IL.NumberOut=Skylake Wafer
Batch_IL.NumberOut + 1:NEXT(10$);


;
;
;     Model statements for module:  BasicProcess.Create 4 (Kabylake Wafer)
;

79$           CREATE,
1,HoursToBaseTime(0.0),KW:HoursToBaseTime(EXPO(0.3)):NEXT(80$);

80$           ASSIGN:        Kabylake Wafer.NumberOut=Kabylake
Wafer.NumberOut + 1:NEXT(3$);


;
;
;     Model statements for module:  BasicProcess.Assign 58 (Cycle Time
Kabylake)
;
3$            ASSIGN:        Cycle Time_KW=TNOW:NEXT(1$);


;
;
```

```
;     Model statements for module:  BasicProcess.Batch 4 (Kabylake Wafer
Batch_IL)
;
1$            QUEUE,         Kabylake Wafer Batch_IL.Queue;
83$           GROUP,         Kaylake Batch,Temporary:25,Last,KW:NEXT(84$);

84$           ASSIGN:        Kabylake Wafer Batch_IL.NumberOut=Kabylake
Wafer Batch_IL.NumberOut + 1:NEXT(11$);


;
;
;     Model statements for module:  AdvancedProcess.Delay 2 (SW IL Transfer)
;
10$           DELAY:         MinutesToBaseTime(UNIF(1,2)),,Other:NEXT(6$);


;
;
;     Model statements for module:  BasicProcess.Assign 5 (Assign SW
Insulator)
;
6$            ASSIGN:        Insulator Time=TRIA(1,1.1,1.2):NEXT(4$);


;
;
;     Model statements for module:  BasicProcess.Process 16 (SIO2 Insulation
Layering)
;
4$            ASSIGN:        SIO2 Insulation Layering.NumberIn=SIO2
Insulation Layering.NumberIn + 1:
                             SIO2 Insulation Layering.WIP=SIO2 Insulation
Layering.WIP+1;
88$           QUEUE,         SIO2 Insulation Layering.Queue;
87$           SEIZE,         2,VA:
                             SELECT(Insulator Machine,RAN, ),1:NEXT(86$);

86$           DELAY:         HoursToBaseTime(Insulator Time),,VA;
85$           RELEASE:       SELECT(Insulator Machine,LAST),1;
133$          ASSIGN:        SIO2 Insulation Layering.NumberOut=SIO2
Insulation Layering.NumberOut + 1:
                             SIO2 Insulation Layering.WIP=SIO2 Insulation
Layering.WIP-1:NEXT(8$);


;
;
;     Model statements for module:  BasicProcess.Decide 3 (Batch Type
Sensor_IL)
;
8$            BRANCH,        1:
                             If,Entity.Type==SW,136$,Yes:
                             Else,137$,Yes;
136$          ASSIGN:        Batch Type Sensor_IL.NumberOut True=Batch Type
Sensor_IL.NumberOut True + 1:NEXT(5$);

137$          ASSIGN:        Batch Type Sensor_IL.NumberOut False=Batch Type
Sensor_IL.NumberOut False + 1:NEXT(9$);


;
;
```

```
;     Model statements for module:  BasicProcess.Separate 7 (Seperate SW_IL)
;
5$             SPLIT::NEXT(138$);

138$           ASSIGN:        Seperate SW_IL.NumberOut Orig=Seperate
SW_IL.NumberOut Orig + 1:NEXT(13$);


;
;
;     Model statements for module:  BasicProcess.Separate 8 (Seperate KW_IL)
;
9$             SPLIT::NEXT(141$);

141$           ASSIGN:        Seperate KW_IL.NumberOut Orig=Seperate
KW_IL.NumberOut Orig + 1:NEXT(14$);


;
;
;     Model statements for module:  AdvancedProcess.Delay 3 (KW IL Transfer)
;
11$            DELAY:         MinutesToBaseTime(UNIF(1,2)),,Other:NEXT(7$);


;
;
;     Model statements for module:  BasicProcess.Assign 6 (Assign KW
Insulator)
;
7$             ASSIGN:        Insulator Time=TRIA(0.9,1,1.1):NEXT(4$);


;
;
;     Model statements for module:  BasicProcess.Assign 7 (Assign
Photopolymerization Time_SW)
;
13$            ASSIGN:        Photopolymerization
Time=TRIA(0.116,0.13,0.15):NEXT(12$);


;
;
;     Model statements for module:  BasicProcess.Process 17 (Applying
Photoresist Material)
;
12$            ASSIGN:        Applying Photoresist Material.NumberIn=Applying
Photoresist Material.NumberIn + 1:
                              Applying Photoresist Material.WIP=Applying
Photoresist Material.WIP+1;
147$           QUEUE,         Applying Photoresist Material.Queue;
146$           SEIZE,         2,VA:
                              SELECT(Polymer Machine,RAN, ),1:NEXT(145$);

145$           DELAY:         HoursToBaseTime(Photopolymerization Time),,VA;
144$           RELEASE:       SELECT(Polymer Machine,LAST),1;
192$           ASSIGN:        Applying Photoresist
Material.NumberOut=Applying Photoresist Material.NumberOut + 1:
                              Applying Photoresist Material.WIP=Applying
Photoresist Material.WIP-1:NEXT(18$);
```

```
;
;
;     Model statements for module:  BasicProcess.Decide 15 (Product Type
Sensor_SE)
;
18$           BRANCH,        1:
                             If,Entity.Type==SW,195$,Yes:
                             Else,196$,Yes;
195$          ASSIGN:        Product Type Sensor_SE.NumberOut True=Product
Type Sensor_SE.NumberOut True + 1:NEXT(16$);

196$          ASSIGN:        Product Type Sensor_SE.NumberOut False=Product
Type Sensor_SE.NumberOut False + 1:NEXT(17$);


;
;
;     Model statements for module:  BasicProcess.Assign 27 (Assign Exposure
Time_SW)
;
16$           ASSIGN:        Exposure
Time=TRIA(0.017,0.025,0.034):NEXT(15$);


;
;
;     Model statements for module:  BasicProcess.Process 27 (Steepper
Exposure)
;
15$           ASSIGN:        Steepper Exposure.NumberIn=Steepper
Exposure.NumberIn + 1:
                             Steepper Exposure.WIP=Steepper Exposure.WIP+1;
200$          QUEUE,         Steepper Exposure.Queue;
199$          SEIZE,         2,VA:
                             SELECT(UV Lithographer,RAN, ),1:NEXT(198$);

198$          DELAY:         HoursToBaseTime(Exposure Time),,VA;
197$          RELEASE:       SELECT(UV Lithographer,LAST),1;
245$          ASSIGN:        Steepper Exposure.NumberOut=Steepper
Exposure.NumberOut + 1:
                             Steepper Exposure.WIP=Steepper Exposure.WIP-
1:NEXT(22$);


;
;
;     Model statements for module:  BasicProcess.Decide 16 (Product Type
Sensor_PD)
;
22$           BRANCH,        1:
                             If,Entity.Type==SW,248$,Yes:
                             Else,249$,Yes;
248$          ASSIGN:        Product Type Sensor_PD.NumberOut True=Product
Type Sensor_PD.NumberOut True + 1:NEXT(20$);

249$          ASSIGN:        Product Type Sensor_PD.NumberOut False=Product
Type Sensor_PD.NumberOut False + 1:NEXT(21$);


;
;
;     Model statements for module:  BasicProcess.Assign 29 (Assign Resist
Development Time_SW)
```

```
;
20$           ASSIGN:        Resist Development
Time=TRIA(0.042,0.067,0.084):NEXT(19$);


;
;
;     Model statements for module:  BasicProcess.Process 28 (Photoresist
Development)
;
19$           ASSIGN:        Photoresist Development.NumberIn=Photoresist
Development.NumberIn + 1:
                             Photoresist Development.WIP=Photoresist
Development.WIP+1;
253$          QUEUE,         Photoresist Development.Queue;
252$          SEIZE,         2,VA:
                             SELECT(Resist Developer,RAN, ),1:NEXT(251$);

251$          DELAY:         HoursToBaseTime(Resist Development Time),,VA;
250$          RELEASE:       SELECT(Resist Developer,LAST),1;
298$          ASSIGN:        Photoresist Development.NumberOut=Photoresist
Development.NumberOut + 1:
                             Photoresist Development.WIP=Photoresist
Development.WIP-1:NEXT(26$);


;
;
;     Model statements for module:  BasicProcess.Assign 30 (Assign Resist
Development Time_KW)
;
21$           ASSIGN:        Resist Development
Time=TRIA(0.034,0.06,0.07):NEXT(19$);


;
;
;     Model statements for module:  BasicProcess.Assign 28 (Assign Exposure
Time_KW)
;
17$           ASSIGN:        Exposure
Time=TRIA(0.008,0.014,0.017):NEXT(15$);


;
;
;     Model statements for module:  BasicProcess.Assign 8 (Assign
Photopolymerization Time_KW)
;
14$           ASSIGN:        Photopolymerization
Time=TRIA(0.1,0.117,0.13):NEXT(12$);


;
;
;     Model statements for module:  BasicProcess.Decide 17 (Product Type
Sensor_II)
;
26$           BRANCH,        1:
                             If,Entity.Type==SW,301$,Yes:
                             Else,302$,Yes;
301$          ASSIGN:        Product Type Sensor_II.NumberOut True=Product
Type Sensor_II.NumberOut True + 1:NEXT(27$);
```

```
302$          ASSIGN:        Product Type Sensor_II.NumberOut False=Product
Type Sensor_II.NumberOut False + 1:NEXT(28$);


;
;
;     Model statements for module:  BasicProcess.Batch 11 (Skylake Wafer
Batch_II)
;
27$           QUEUE,         Skylake Wafer Batch_II.Queue;
303$          GROUP,         Skylake Batch,Temporary:5,Last,SW:NEXT(304$);

304$          ASSIGN:        Skylake Wafer Batch_II.NumberOut=Skylake Wafer
Batch_II.NumberOut + 1:NEXT(24$);


;
;
;     Model statements for module:  BasicProcess.Assign 31 (Assign Atomic
Bombardment Time_SW)
;
24$           ASSIGN:        Atomic Bombardment
Time=TRIA(0.167,0.2,0.23):NEXT(23$);


;
;
;     Model statements for module:  BasicProcess.Process 29 (Ion
Implantation)
;
23$           ASSIGN:        Ion Implantation.NumberIn=Ion
Implantation.NumberIn + 1:
                             Ion Implantation.WIP=Ion Implantation.WIP+1;
308$          QUEUE,         Ion Implantation.Queue;
307$          SEIZE,         2,VA:
                             SELECT(Atomic Bombarder,RAN, ),1:NEXT(306$);

306$          DELAY:         HoursToBaseTime(Atomic Bombardment Time),,VA;
305$          RELEASE:       SELECT(Atomic Bombarder,LAST),1;
353$          ASSIGN:        Ion Implantation.NumberOut=Ion
Implantation.NumberOut + 1:
                             Ion Implantation.WIP=Ion Implantation.WIP-
1:NEXT(30$);


;
;
;     Model statements for module:  BasicProcess.Decide 22 (Batch Type
Sensor_II)
;
30$           BRANCH,        1:
                             If,Entity.Type==SW,356$,Yes:
                             Else,357$,Yes;
356$          ASSIGN:        Batch Type Sensor_II.NumberOut True=Batch Type
Sensor_II.NumberOut True + 1:NEXT(29$);

357$          ASSIGN:        Batch Type Sensor_II.NumberOut False=Batch Type
Sensor_II.NumberOut False + 1:NEXT(31$);


;
;
```

```
;     Model statements for module:  BasicProcess.Separate 15 (Seperate
SW_II)
;
29$           SPLIT::NEXT(358$);

358$          ASSIGN:        Seperate SW_II.NumberOut Orig=Seperate
SW_II.NumberOut Orig + 1:NEXT(37$);


;
;
;     Model statements for module:  BasicProcess.Assign 60 (Assign Resist
Removal Time)
;
37$           ASSIGN:        Resist Removal
Time=TRIA(0.025,0.034,0.06):NEXT(36$);


;
;
;     Model statements for module:  BasicProcess.Process 47 (Photoresist
Removal)
;
36$           ASSIGN:        Photoresist Removal.NumberIn=Photoresist
Removal.NumberIn + 1:
                             Photoresist Removal.WIP=Photoresist
Removal.WIP+1;
364$          QUEUE,         Photoresist Removal.Queue;
363$          SEIZE,         2,VA:
                             SELECT(Resist Remover,RAN, ),1:NEXT(362$);

362$          DELAY:         HoursToBaseTime(Resist Removal Time),,VA;
361$          RELEASE:       SELECT(Resist Remover,LAST),1;
409$          ASSIGN:        Photoresist Removal.NumberOut=Photoresist
Removal.NumberOut + 1:
                             Photoresist Removal.WIP=Photoresist
Removal.WIP-1:NEXT(35$);


;
;
;     Model statements for module:  BasicProcess.Decide 23 (Product Type
Sensor_BHM)
;
35$           BRANCH,        1:
                             If,Entity.Type==SW,412$,Yes:
                             Else,413$,Yes;
412$          ASSIGN:        Product Type Sensor_BHM.NumberOut True=Product
Type Sensor_BHM.NumberOut True + 1:NEXT(33$);

413$          ASSIGN:        Product Type Sensor_BHM.NumberOut False=Product
Type Sensor_BHM.NumberOut False + 1:NEXT(34$);


;
;
;     Model statements for module:  BasicProcess.Assign 39 (Assign
Transistor Implantar Time_SW)
;
33$           ASSIGN:        Masking Time=TRIA(0.106,0.12,0.14):NEXT(32$);


;
```

```
;
;     Model statements for module:  BasicProcess.Process 33 (Blue Hard
Masking)
;
32$           ASSIGN:        Blue Hard Masking.NumberIn=Blue Hard
Masking.NumberIn + 1:
                             Blue Hard Masking.WIP=Blue Hard Masking.WIP+1;
417$          QUEUE,         Blue Hard Masking.Queue;
416$          SEIZE,         2,VA:
                             SELECT(Hard Masker,RAN, ),1:NEXT(415$);

415$          DELAY:         HoursToBaseTime(Masking Time),,VA;
414$          RELEASE:       SELECT(Hard Masker,LAST),1;
462$          ASSIGN:        Blue Hard Masking.NumberOut=Blue Hard
Masking.NumberOut + 1:
                             Blue Hard Masking.WIP=Blue Hard Masking.WIP-
1:NEXT(38$);


;
;
;     Model statements for module:  BasicProcess.Assign 40 (Assign
Transistor Implantar Time_KW)
;
34$           ASSIGN:        Masking Time=TRIA(0.08,0.106,0.12):NEXT(32$);


;
;
;     Model statements for module:  BasicProcess.Separate 16 (Seperate
KW_II)
;
31$           SPLIT::NEXT(465$);

465$          ASSIGN:        Seperate KW_II.NumberOut Orig=Seperate
KW_II.NumberOut Orig + 1:NEXT(37$);


;
;
;     Model statements for module:  BasicProcess.Batch 12 (Kabylake Wafer
Batch_II)
;
28$           QUEUE,         Kabylake Wafer Batch_II.Queue;
468$          GROUP,         Kaylake Batch,Temporary:5,Last,KW:NEXT(469$);

469$          ASSIGN:        Kabylake Wafer Batch_II.NumberOut=Kabylake
Wafer Batch_II.NumberOut + 1:NEXT(25$);


;
;
;     Model statements for module:  BasicProcess.Assign 32 (Assign Atomic
Bombardment Time_KW)
;
25$           ASSIGN:        Atomic Bombardment Time=TRIA(0.1, 0.13,
0.167):NEXT(23$);


;
;
;     Model statements for module:  BasicProcess.Decide 26 (Product Type
Sensor_WE)
```

```
;
38$           BRANCH,        1:
                             If,Entity.Type==SW,470$,Yes:
                             Else,471$,Yes;
470$          ASSIGN:        Product Type Sensor_WE.NumberOut True=Product
Type Sensor_WE.NumberOut True + 1:NEXT(39$);

471$          ASSIGN:        Product Type Sensor_WE.NumberOut False=Product
Type Sensor_WE.NumberOut False + 1:NEXT(40$);


;
;
;     Model statements for module:  BasicProcess.Batch 15 (Skylake Wafer
Batch_WE)
;
39$           QUEUE,         Skylake Wafer Batch_WE.Queue;
472$          GROUP,         Skylake Batch,Temporary:25,Last,SW:NEXT(473$);

473$          ASSIGN:        Skylake Wafer Batch_WE.NumberOut=Skylake Wafer
Batch_WE.NumberOut + 1:NEXT(43$);


;
;
;     Model statements for module:  BasicProcess.Assign 45 (Assign SW Wet
Etcher)
;
43$           ASSIGN:        Etching Time=TRIA(0.6,0.67,0.7):NEXT(41$);


;
;
;     Model statements for module:  BasicProcess.Process 36 (Wet Etching)
;
41$           ASSIGN:        Wet Etching.NumberIn=Wet Etching.NumberIn + 1:
                             Wet Etching.WIP=Wet Etching.WIP+1;
477$          QUEUE,         Wet Etching.Queue;
476$          SEIZE,         2,VA:
                             SELECT(Wet Etcher,RAN, ),1:NEXT(475$);

475$          DELAY:         HoursToBaseTime(Etching Time),,VA;
474$          RELEASE:       SELECT(Wet Etcher,LAST),1;
522$          ASSIGN:        Wet Etching.NumberOut=Wet Etching.NumberOut +
1:
                             Wet Etching.WIP=Wet Etching.WIP-1:NEXT(45$);


;
;
;     Model statements for module:  BasicProcess.Decide 27 (Batch Type
Sensor_WE)
;
45$           BRANCH,        1:
                             If,Entity.Type==SW,525$,Yes:
                             Else,526$,Yes;
525$          ASSIGN:        Batch Type Sensor_WE.NumberOut True=Batch Type
Sensor_WE.NumberOut True + 1:NEXT(42$);

526$          ASSIGN:        Batch Type Sensor_WE.NumberOut False=Batch Type
Sensor_WE.NumberOut False + 1:NEXT(46$);
```

```
;
;
;     Model statements for module:  BasicProcess.Separate 19 (Seperate
SW_WE)
;
42$           SPLIT::NEXT(527$);

527$          ASSIGN:        Seperate SW_WE.NumberOut Orig=Seperate
SW_WE.NumberOut Orig + 1:NEXT(48$);


;
;
;     Model statements for module:  BasicProcess.Assign 47 (Assign Mask
Removal Time)
;
48$           ASSIGN:        Mask Removal
Time=TRIA(0.017,0.025,0.034):NEXT(47$);


;
;
;     Model statements for module:  BasicProcess.Process 37 (Hard Mask
Removal)
;
47$           ASSIGN:        Hard Mask Removal.NumberIn=Hard Mask
Removal.NumberIn + 1:
                             Hard Mask Removal.WIP=Hard Mask Removal.WIP+1;
533$          QUEUE,         Hard Mask Removal.Queue;
532$          SEIZE,         2,VA:
                             SELECT(Mask Remover,RAN, ),1:NEXT(531$);

531$          DELAY:         HoursToBaseTime(Mask Removal Time),,VA;
530$          RELEASE:       SELECT(Mask Remover,LAST),1;
578$          ASSIGN:        Hard Mask Removal.NumberOut=Hard Mask
Removal.NumberOut + 1:
                             Hard Mask Removal.WIP=Hard Mask Removal.WIP-
1:NEXT(52$);


;
;
;     Model statements for module:  BasicProcess.Separate 20 (Seperate
KW_WE)
;
46$           SPLIT::NEXT(581$);

581$          ASSIGN:        Seperate KW_WE.NumberOut Orig=Seperate
KW_WE.NumberOut Orig + 1:NEXT(48$);


;
;
;     Model statements for module:  BasicProcess.Batch 16 (Kabylake Wafer
Batch_WE)
;
40$           QUEUE,         Kabylake Wafer Batch_WE.Queue;
584$          GROUP,         Kaylake Batch,Temporary:25,Last,KW:NEXT(585$);

585$          ASSIGN:        Kabylake Wafer Batch_WE.NumberOut=Kabylake
Wafer Batch_WE.NumberOut + 1:NEXT(44$);
```

```
;
;
;     Model statements for module:  BasicProcess.Assign 46 (Assign KW Wet
Etcher)
;
44$           ASSIGN:        Etching Time=TRIA(0.35,0.42,0.5):NEXT(41$);


;
;
;     Model statements for module:  BasicProcess.Decide 28 (Product Type
Sensor_GF)
;
52$           BRANCH,        1:
                             If,Entity.Type==SW,586$,Yes:
                             Else,587$,Yes;
586$          ASSIGN:        Product Type Sensor_GF.NumberOut True=Product
Type Sensor_GF.NumberOut True + 1:NEXT(50$);

587$          ASSIGN:        Product Type Sensor_GF.NumberOut False=Product
Type Sensor_GF.NumberOut False + 1:NEXT(51$);


;
;
;     Model statements for module:  BasicProcess.Assign 49 (Assign Gate
Formation Time_SW)
;
50$           ASSIGN:        Gate Formation
Time=TRIA(0.17,0.22,0.27):NEXT(49$);


;
;
;     Model statements for module:  BasicProcess.Process 38 (Gate Formation)
;
49$           ASSIGN:        Gate Formation.NumberIn=Gate Formation.NumberIn
+ 1:
                             Gate Formation.WIP=Gate Formation.WIP+1;
591$          QUEUE,         Gate Formation.Queue;
590$          SEIZE,         2,VA:
                             SELECT(Metal Gate Processor,RAN,
),1:NEXT(589$);

589$          DELAY:         HoursToBaseTime(Gate Formation Time),,VA;
588$          RELEASE:       SELECT(Metal Gate Processor,LAST),1;
636$          ASSIGN:        Gate Formation.NumberOut=Gate
Formation.NumberOut + 1:
                             Gate Formation.WIP=Gate Formation.WIP-
1:NEXT(56$);


;
;
;     Model statements for module:  BasicProcess.Decide 30 (Product Type
Sensor_MD)
;
56$           BRANCH,        1:
                             If,Entity.Type==SW,639$,Yes:
                             Else,640$,Yes;
639$          ASSIGN:        Product Type Sensor_MD.NumberOut True=Product
Type Sensor_MD.NumberOut True + 1:NEXT(54$);
```

```
640$          ASSIGN:        Product Type Sensor_MD.NumberOut False=Product
Type Sensor_MD.NumberOut False + 1:NEXT(55$);


;
;
;     Model statements for module:  BasicProcess.Assign 53 (Assign Metal
Deposition Time_SW)
;
54$           ASSIGN:        Metal Deposition
Time=TRIA(0.18,0.21,0.25):NEXT(53$);


;
;
;     Model statements for module:  BasicProcess.Process 40 (Metal
Deposition)
;
53$           ASSIGN:        Metal Deposition.NumberIn=Metal
Deposition.NumberIn + 1:
                             Metal Deposition.WIP=Metal Deposition.WIP+1;
644$          QUEUE,         Metal Deposition.Queue;
643$          SEIZE,         2,VA:
                             SELECT(Electroplater,RAN, ),1:NEXT(642$);

642$          DELAY:         HoursToBaseTime(Metal Deposition Time),,VA;
641$          RELEASE:       SELECT(Electroplater,LAST),1;
689$          ASSIGN:        Metal Deposition.NumberOut=Metal
Deposition.NumberOut + 1:
                             Metal Deposition.WIP=Metal Deposition.WIP-
1:NEXT(67$);


;
;
;     Model statements for module:  BasicProcess.Assign 54 (Assign Metal
Deposition Time_KW)
;
55$           ASSIGN:        Metal Deposition Time=TRIA(0.13, 0.18,
0.22):NEXT(53$);


;
;
;     Model statements for module:  BasicProcess.Assign 50 (Assign Gate
Formation Time_KW)
;
51$           ASSIGN:        Gate Formation Time=TRIA(0.13, 0.17,
0.2):NEXT(49$);


;
;
;     Model statements for module:  BasicProcess.Decide 31 (Product Type
Sensor_Count)
;
67$           BRANCH,        1:
                             If,Entity.Type==SW,692$,Yes:
                             Else,693$,Yes;
692$          ASSIGN:        Product Type Sensor_Count.NumberOut
True=Product Type Sensor_Count.NumberOut True + 1:NEXT(68$);
```

```
693$          ASSIGN:        Product Type Sensor_Count.NumberOut
False=Product Type Sensor_Count.NumberOut False + 1:NEXT(69$);


;
;
;     Model statements for module:  BasicProcess.Assign 55 (SW Pass)
;
68$           ASSIGN:        SW Parts=SW Parts +1:
                             SW Passes=SW Passes + 1:NEXT(70$);


;
;
;     Model statements for module:  BasicProcess.Decide 32 (Decide for
Rework SW)
;
70$           BRANCH,        1:
                             If,SW Passes==3,694$,Yes:
                             Else,695$,Yes;
694$          ASSIGN:        Decide for Rework SW.NumberOut True=Decide for
Rework SW.NumberOut True + 1:NEXT(57$);

695$          ASSIGN:        Decide for Rework SW.NumberOut False=Decide for
Rework SW.NumberOut False + 1:NEXT(13$);


;
;
;     Model statements for module:  BasicProcess.Assign 56 (KW Pass)
;
69$           ASSIGN:        KW Parts=KW Parts +1:
                             KW Passes=KW Passes + 1:NEXT(71$);


;
;
;     Model statements for module:  BasicProcess.Decide 33 (Decide for
Rework KW)
;
71$           BRANCH,        1:
                             If,KW Passes==2,696$,Yes:
                             Else,697$,Yes;
696$          ASSIGN:        Decide for Rework KW.NumberOut True=Decide for
Rework KW.NumberOut True + 1:NEXT(58$);

697$          ASSIGN:        Decide for Rework KW.NumberOut False=Decide for
Rework KW.NumberOut False + 1:NEXT(14$);


;
;
;     Model statements for module:  BasicProcess.Batch 17 (Skylake Wafer
Batch_WST)
;
57$           QUEUE,         Skylake Wafer Batch_WST.Queue;
698$          GROUP,         Skylake Batch,Temporary:5,Last,SW:NEXT(699$);

699$          ASSIGN:        Skylake Wafer Batch_WST.NumberOut=Skylake Wafer
Batch_WST.NumberOut + 1:NEXT(59$);


;
```

```
;
;     Model statements for module:  BasicProcess.Process 41 (Wafer Sort
Test)
;
59$           ASSIGN:        Wafer Sort Test.NumberIn=Wafer Sort
Test.NumberIn + 1:
                             Wafer Sort Test.WIP=Wafer Sort Test.WIP+1;
703$          QUEUE,         Wafer Sort Test.Queue;
702$          SEIZE,         2,VA:
                             SELECT(InSitu Sensor,RAN, ),1:NEXT(701$);

701$          DELAY:         HoursToBaseTime(TRIA(0.20,0.45,0.60)),,VA;
700$          RELEASE:       SELECT(InSitu Sensor,LAST),1;
748$          ASSIGN:        Wafer Sort Test.NumberOut=Wafer Sort
Test.NumberOut + 1:
                             Wafer Sort Test.WIP=Wafer Sort Test.WIP-
1:NEXT(61$);


;
;
;     Model statements for module:  BasicProcess.Decide 34 (Batch Type
Sensor_WST)
;
61$           BRANCH,        1:
                             If,Entity.Type==SW,751$,Yes:
                             Else,752$,Yes;
751$          ASSIGN:        Batch Type Sensor_WST.NumberOut True=Batch Type
Sensor_WST.NumberOut True + 1:NEXT(60$);

752$          ASSIGN:        Batch Type Sensor_WST.NumberOut False=Batch
Type Sensor_WST.NumberOut False + 1:NEXT(62$);


;
;
;     Model statements for module:  BasicProcess.Separate 21 (Seperate
SW_WST)
;
60$           SPLIT::NEXT(753$);

753$          ASSIGN:        Seperate SW_WST.NumberOut Orig=Seperate
SW_WST.NumberOut Orig + 1:NEXT(65$);


;
;
;     Model statements for module:  BasicProcess.Separate 22 (Seperate
KW_WST)
;
62$           SPLIT::NEXT(756$);

756$          ASSIGN:        Seperate KW_WST.NumberOut Orig=Seperate
KW_WST.NumberOut Orig + 1:NEXT(66$);


;
;
;     Model statements for module:  BasicProcess.Batch 18 (Kabylake Wafer
Batch_WST)
;
58$           QUEUE,         Kabylake Wafer Batch_WST.Queue;
759$          GROUP,         Kaylake Batch,Temporary:5,Last,KW:NEXT(760$);
```

```
760$          ASSIGN:        Kabylake Wafer Batch_WST.NumberOut=Kabylake
Wafer Batch_WST.NumberOut + 1:NEXT(59$);


;
;
;     Model statements for module:  BasicProcess.Record 3
;
65$           DELAY:         0.0,,Other:NEXT(762$);

762$          TALLY:         Cycle_Time_SW_Tally,INT(Cycle
Time_SW),1:NEXT(761$);

761$          DELAY:         0.0,,Other:NEXT(63$);


;
;
;     Model statements for module:  BasicProcess.Dispose 10 (Skylake Packed
and Shipped)
;
63$           ASSIGN:        Skylake Packed and Shipped.NumberOut=Skylake
Packed and Shipped.NumberOut + 1;
763$          DISPOSE:       Yes;


;
;
;     Model statements for module:  BasicProcess.Record 4
;
66$           DELAY:         0.0,,Other:NEXT(765$);

765$          TALLY:         Cycle Time_KW_Tally,INT(Cycle
Time_KW),1:NEXT(764$);

764$          DELAY:         0.0,,Other:NEXT(64$);


;
;
;     Model statements for module:  BasicProcess.Dispose 11 (Kabylake Packed
and Shipped)
;
64$           ASSIGN:        Kabylake Packed and Shipped.NumberOut=Kabylake
Packed and Shipped.NumberOut + 1;
766$          DISPOSE:       Yes;
```

Appendix B: Model Comparison Sheet

FIFO

Work-in Process			
Wafer	**Minimum**	**Average**	**Maximum**
Skylake	16	47	91
Kabylake	10	42	82
Total (Skylake+Kabylake)	44	89	146

Avg Waiting Time (hours)			
Wafer	**Minimum**	**Average**	**Maximum**
Skylake	0	10.74	16.7
Kabylake	0	8.3	13.9

Cycle Time (hours)			
Wafer	**Minimum**	**Average**	**Maximum**
Skylake	12.02	17.4	31.23
Kabylake	7.56	12.2	23.1

Wafer	Total Produced
Skylake	45,104
Kabylake	50,513
Total Wafers Produced	**95,617**

Priority Based

Work-in Process			
Wafer	**Minimum**	**Average**	**Maximum**
Skylake	19	51	100
Kabylake	9	40	81
Total (Skylake+Kabylake)	46	91	144

Avg Waiting Time (hours)			
Wafer	**Minimum**	**Average**	**Maximum**
Skylake	0	12.25	16.25
Kabylake	0	7.56	10.1

Cycle Time (hours)			
Wafer	**Minimum**	**Average**	**Maximum**
Skylake	12.31	18.86	31.93
Kabylake	7.46	11.47	21.21

Wafer	Total Produced
Skylake	44,967
Kabylake	50,636
Total Wafers Produced	**95,603**

SPT

Work-in Process			
Wafer	**Minimum**	**Average**	**Maximum**
Skylake	16	48.68	93
Kabylake	10	41	84
Total (Skylake+Kabylake)	40	90	146

Avg Waiting Time (hours)			
Wafer	**Minimum**	**Average**	**Maximum**
Skylake	0	11.43	14.43
Kabylake	0	8	11.05

Cycle Time (hours)			
Wafer	**Minimum**	**Average**	**Maximum**
Skylake	10.71	18.07	41.7
Kabylake	6.24	11.93	23.62

Wafer	Total Produced
Skylake	45377
Kabylake	50,704
Total Wafers Produced	**96,081**

Failure FIFO

Work-in Process			
Wafer	**Minimum**	**Average**	**Maximum**
Skylake	18	47	88
Kabylake	12	42	74
Total (Skylake+Kabylake)	49	89	142

Avg Waiting Time (hours)			
Wafer	**Minimum**	**Average**	**Maximum**
Skylake	0	10.71	13
Kabylake	0	8.37	10.58

Cycle Time (hours)			
Wafer	**Minimum**	**Average**	**Maximum**
Skylake	12.33	17.4	32.61
Kabylake	7.71	12.31	23.35

Wafer	Total Produced
Skylake	45405
Kabylake	50,098
Total Wafers Produced	**95,503**

Chip Name	Weighing Criterians (Scale 1-10)																		
	Chip Revenue			Chip Rework			Chip Processing Time			Material Waste Generated			Customer Popularity			Defects			Total Score
	Rating	Score	Weight	Rating	Score	Weight	Rating	Score	Weight	Rating	Score	Weight	Rating	Score	Weight	Rating	Score	Weight	
Skylake Wafer	2	0.67	35	1	0.33	8	1	0.33	15	1	0.33	6	1	0.33	25	2	0.67	11	48.67
KabyLake Wafer	1	0.33	35	2	0.67	8	2	0.67	15	2	0.67	6	2	0.67	25	1	0.33	11	51.33
Sum	3	More		3	Less		3	Less		3	Less		3	More		3	Less		100

Bibliography

References

1. AccountingVerse. (2017). *Managerial Manufacturing* . Retrieved from Manufacturing Cycle Time: http://www.accountingverse.com/managerial-accounting/responsibility-accounting/manufacturing-cycle-time.html

2. Ankenman, B., Nelson, B. L., Tongarlak, M., Fowler, J., Mackulak, G., Pabst, D., & Yang, F. (2007, December 21). Cycle Time Prediction for Semiconductor Manufacturing via Simulation on Demand.

3. Appels, J., Kooi, E., Paffen, M. M., Schatorje, J. J., & Verkuylen, W. H. (1970). *Local oxidation of silicon and its application in semiconductor-device technology,*. Philips.

4. Applied Materials . (2017). *Semiconductor Manufacturing* . Retrieved from Etching: http://www.appliedmaterials.com/semiconductor/products/etch/info

5. Arisha, A., & Young, P. (2004). Applications of Simulation in Semiconductor Manufacturing Facilities. *Nineteenth International Manufacturing Conference* . Ireland.

6. Becker, M. (2003). Modeling and simulation of a complete semiconductor manufacturing facility using Petri nets. Hannover, Germany.

7. Chen, T. (2013). A Systematic Cycle Time Reduction Procedure for Enhancing the Competitiveness and Sustainability of a Semiconductor Manufacturer. *Sustainibility* .

8. Cheung, N. (2010). Retrieved from UC Berkley: http://www-inst.eecs.berkeley.edu/~ee143/fa10/lectures/Lec_08.pdf

9. Cringely, R. X. (2013, October 15). *Cringely on Technology* . Retrieved from https://www.cringely.com/2013/10/15/breaking-moores-law/

10. Decker, C. (2002). Light-induced crosslinking polymerization. *Polymer International*, 1- 4.

11. Delligatti, L. (2013). SysML Distilled: A Brief Guide to Systems Modelling Language. In L. Delligatti, *SysML Distilled: A Brief Guide to Systems Modelling Language.* Addison-Welsey Professional .

12. Devadas D. Pillai, E. L. (2004). 300-mm Full-Factory Simulations for 90- and 65-nm IC Manufacturing. *IEEE*. IEEE.

13. Domaschke, J., Brown, S., Robinson, J., & Leibl, F. (1998). EFFECTIVE IMPLEMENTATION OF CYCLE TIME REDUCTION STRATEGIES FOR SEMICONDUCTOR BACK-END MANUFACTURING. *Winter Simulation Conference* . IEEE.

14. Eberts, D., Keil, S., Peipp, F., & Lasch, R. (2015). Shortening of cycle time in semiconductor manufacturing via meaningful lot sizes. *IEEE*. Dresden: IEEE.

15. Fishman, D. (2017, June 9). *Intel* . Retrieved from Intel Chip Manufacturing Process: https://www.intel.com/content/www/us/en/history/museum-making-silicon.html

16. Hamm, Robert, & Marianne. (2012). Industrial Accelerators and Their Applications. World Scientific. In Hamm, R. W., & M. E., *Industrial Accelerators and Their Applications. World Scientific* (pp. 243-249). World Scientific.

17. Intel . (2012, January). 22nm 3D/Trigate Transistors.

18. Intel. (2011). 32nm High-K/Metal Gate. *From Sand to Silicon, "Making of a Chip"*, 1-3.

19. Intel. (2016). *Intel Processor Chip Specifications*. Retrieved from https://ark.intel.com/compare/123589,97184

20. Intel. (2017). 5th Gen Intel Processor Families.

21. Intel Developer Forum. (2015). Semiconductor & Computer Engineering. San Francisco: Intel. Retrieved from WikiChip: https://en.wikichip.org/wiki/intel/microarchitectures/skylake_(client)

22. Intel Developer Forum. (2016). Semiconductor & Computer Engineering. San Francisco: Intel. Retrieved from WikiChip: https://en.wikichip.org/wiki/intel/microarchitectures/skylake_(client)

23. Janakiram, M. (1996). Cycle Time Reduction at Motorola’s ACT Fab. Mesa, Arizona, USA.

24. Kelton, W. D., Sadowski, R. P., & Swets, N. B. (2010). *Simulation with Arena*. New York: Mc Graw Hill.

25. Law, A. M. (2010). STATISTICAL ANALYSIS OF SIMULATION OUTPUT DATA:THE PRACTICAL STATE OF THE ART. *Winter Simulation Conference* , (pp. 65-74). Tucson, AZ.

26. LeanKit. (2018). *Leankit* . Retrieved from 4 Benefits of WIP Limits : https://leankit.com/learn/kanban/benefits-of-wip-limits/

27. Little, J. D., & Graves, S. C. (2008). Little's Law. In J. D. Little, & S. C. Graves, *Little's Law.*

28. Liua, M., Jin, P., Xu, Z., D.A.H.Hanaor, Gan, Y., & C.Q.Chen. (2016, September). *Elsevier.* Retrieved from Science Direct: https://www.sciencedirect.com/science/article/pii/S209503491630040X?via%3Dihub

29. Moore, G. E. (1975). Cramming more components onto integrated circuits. *Electronics*, 5.

30. National Taipei University of Technology. (2006, May 23). IC Fabrication . Taiwan.

31. Page, E. H., & Nance, R. E. (1994). *PARALLEL DISCRETE EVENT SIMULATION: A MODELING METHODOLOGICAL PERSPECTIVE.* Edinburg: ACM/IEEE.

32. Philip Laube. (2010). Semiconductor Technology . Hamburg.

33. Pierreval, H., Daures, A., Both, T., Szimczak, S., Gonzalez, P., & Framinan, J. (2013). A Simulation Optimization Approach for Reactive ConWIP Systems. *8th EUROSIM Congress on Modelling and Simulation* , (pp. 415-420). Wales, United Kingdom .

34. Pillai, D. D., Bass, E. L., Dempsey, J. C., & Yellig, E. J. (2004). 300-mm Full-Factory Simulations for 90- and 65-nm IC Manufacturing. *IEEE.* IEEE.

35. Printed Circuit Imaging. (2014). *Think and Thinker*. Retrieved from https://www.thinktink.com/stack/volumes/volvi/filmimag.htm

36. Robinson, S. (2004). The Practice of Model Develpment and Use . In S. Robinson, *The Practice of Model Develpment and Use* . Wiley.

37. Samsung. (2015, April 22). *Samsung Newsroom*. Retrieved from https://news.samsung.com/global/eight-major-steps-to-semiconductor-fabrication-part-1-creating-the-wafer

38. Schaller, R. R. (1997). Moore's law: past, present and future. *IEEE Spectrum*, 52 - 59.

39. Shanthikumar, J. G., Ding, S., & Zhang, M. T. (2007). Queueing Theory for Semiconductor Manufacturing Systems: A Survey and Open Problems. *IEEE* (pp. 513-521). California : IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING.

40. Silicon valley Micro-electronics. (2015). *Silicon valley Micro-electronics*. Retrieved from Silicon Wafer Manufacturing Process: https://www.svmi.com/silicon-wafer-manufacturing-semiconductor-process/

41. Silva, E. B., Costa, M. G., Silva, M. F., & Pereira, F. H. (2012). International Conference on Industrial Engineering and Operations Management . Sao Paolo.

42. Stubbe, K. (2010, January). DEVELOPMENT AND SIMULATION ASSESSMENT OF SEMICONDUCTOR PRODUCTION SYSTEM ENHANCEMENTS FOR FAST CYCLE TIMES.

43. Tullis, B., Mehrotra, V., & Zuanich, D. (1990). SUCCESSFUL MODELING OF A SEMICONDUCTOR R&D FACILITY. *Int'l Semiconductor Manufacturing Science Symposium.*

44. Wein, L. M. (1998, August). Scheduling Semiconductor Wafer Fabrication. *IEEE*. IEEE.