3-1-2005

A Simulation Study of Automated Material Handling Systems in Semiconductor Fabs

Julie Christopher

Rochester Institute of Technology

A SIMULATION STUDY OF AUTOMATED MATERIAL HANDLING SYSTEMS IN SEMICONDUCTOR FABS

A Thesis

Submitted in partial fulfillment of the

requirements for the degree of

Master of Science in Industrial Engineering

in the

Department of Industrial & Systems Engineering

Kate Gleason College of Engineering

by

Julie C. Christopher

B.S., Industrial Engineering, Rochester Institute of Technology, 2005

March, 2005

Thesis/Dissertation Author Permission Statement

Title of thesis or dissertation: A Simulation Study of Automated Material Handling Systems in Semiconductor Fabs

Name of author: Julie Christopher
Degree: BS/MS
Program: Industrial and Systems Engineering
College: Engineering

I understand that I must submit a print copy of my thesis or dissertation to the RIT Archives, per current RIT guidelines for the completion of my degree. I hereby grant to the Rochester Institute of Technology and its agents the non-exclusive license to archive and make accessible my thesis or dissertation in whole or in part in all forms of media in perpetuity. I retain all other ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Print Reproduction Permission Granted:

I, _____, hereby **grant permission** to the Rochester Institute Technology to reproduce my print thesis or dissertation in whole or in part. Any reproduction will not be for commercial use or profit.

Signature of Author: _____ Date: 3/11/05

Print Reproduction Permission Denied:

I, _____, hereby **deny permission** to the RIT Library of the Rochester Institute of Technology to reproduce my print thesis or dissertation in whole or in part.

Signature of Author: _____ Date: _____

Inclusion in the RIT Digital Media Library Electronic Thesis & Dissertation (ETD) Archive

I, _____, additionally grant to the Rochester Institute of Technology Digital Media Library (RIT DML) the non-exclusive license to archive and provide electronic access to my thesis or dissertation in whole or in part in all forms of media in perpetuity.

I understand that my work, in addition to its bibliographic record and abstract, will be available to the world-wide community of scholars and researchers through the RIT DML. I retain all other ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation. I am aware that the Rochester Institute of Technology does not require registration of copyright for ETDs.

I hereby certify that, if appropriate, I have obtained and attached written permission statements from the owners of each third party copyrighted matter to be included in my thesis or dissertation. I certify that the version I submitted is the same as that approved by my committee.

Signature of Author: _____ Date: _____

DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING

KATE GLEASON COLLEGE OF ENGINEERING

ROCHESTER INSTITUTE OF TECHNOLOGY

ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

M.S. DEGREE THESIS

The M.S. Degree Thesis of Julie C. Christopher

has been examined and approved by the

thesis committee as satisfactory for the

thesis requirement for the

Master of Science degree

Approved by:

_____

Dr. Michael E. Kuhl, Thesis Advisor

_____

Dr. Karl D. Hirschman

_____

Dr. Moises Sudit

ABSTRACT

A critical aspect of semiconductor manufacturing is the design and analysis of material handling and production control polices to optimize fab performance. As wafer sizes have increased, semiconductor fabs have moved toward the use of automated material handling systems (AMHS). However, the behavior of AMHS and the effects of AMHS on fab productivity are not well understood. The first aspect of the research involves the development of a design and analysis methodology for evaluating the throughput capacity of AMHS. A set of simulation experiments is used to evaluate the throughput capacity of an AMHS and the effects on fab performance measures. This research utilizes two simulations of SEMATECH fab data of actual production fabs. The AMHS vehicle utilization point at which fab performance is degraded is studied. Results show a large increase in lot cycle time at a vehicle utilization of 75%, far below the maximum 100% utilization. These results stress the importance of using a performance indicator that takes into account the performance of the entire fab and not only the AMHS. The second aspect of this research involves the study of AMHS and tool dispatching rules. The hypothesis of this study is that fab performance is affected by both the choice of AMHS and tool dispatching rules as well as their interaction. A full factorial design experiment is conducted to test this hypothesis. Results show that for each fab tested the vehicle rules, machine rules, and their interactions are significant using an ANOVA test on average lot cycle time and other fab performance measures. Additional analyses are conducted to identify robust combinations of AMHS and tool dispatching rules among those tested. The overall results of this study indicate that AMHS and tool dispatching rules effect fab performance and must be considered together when trying to optimize fab performance.

TABLE OF CONTENTS

1 INTRODUCTION

Semiconductor manufacturing is a unique industry in many respects. Long and complicated process routes, small features and tolerances, the presence of reentrant flow, and a large amount of rework result in complex manufacturing systems that present many challenges for factory optimization.

In semiconductor fabrication facilities (fabs), wafers travel through a clean room factory in lots containing 20 to 50 wafers. These lots can have process routes of up to 600 steps with cycles times measured in months. These are among the longest in manufacturing. The lengthy process routes result in the presence of a large amount of work in process (WIP). One of the most unique aspects of semiconductor manufacturing is the presence of reentrant flow. Devices such as transistors are formed on a wafer through a series of processes that deposit material onto the wafer in a series of layers. Each feature that is created on a wafer requires processing at multiple tools. The wafer visits many of the same tools again for subsequent layers. A single tool may be visited over 20 times by a lot before processing is complete.

With months of cycle time dedicated to each lot and hundreds of chips on a single wafer, the monetary value of a single lot can be extremely high. When completed, a lot can be valued at anywhere between $10,000 and $1,000,000 depending on the type and size of product. The processes also work with very small tolerances. Critical dimensions can be as small as 60 nm. If any process is misaligned by even a small amount, the resulting product may not function. With their high value, wafers are not be scrapped if at all possible. As a result, a large amount of rework can exist.

Since the semiconductor manufacturing industry is extremely competitive and the life cycle of products is relatively short, efficient manufacturing methods are critical to the success of

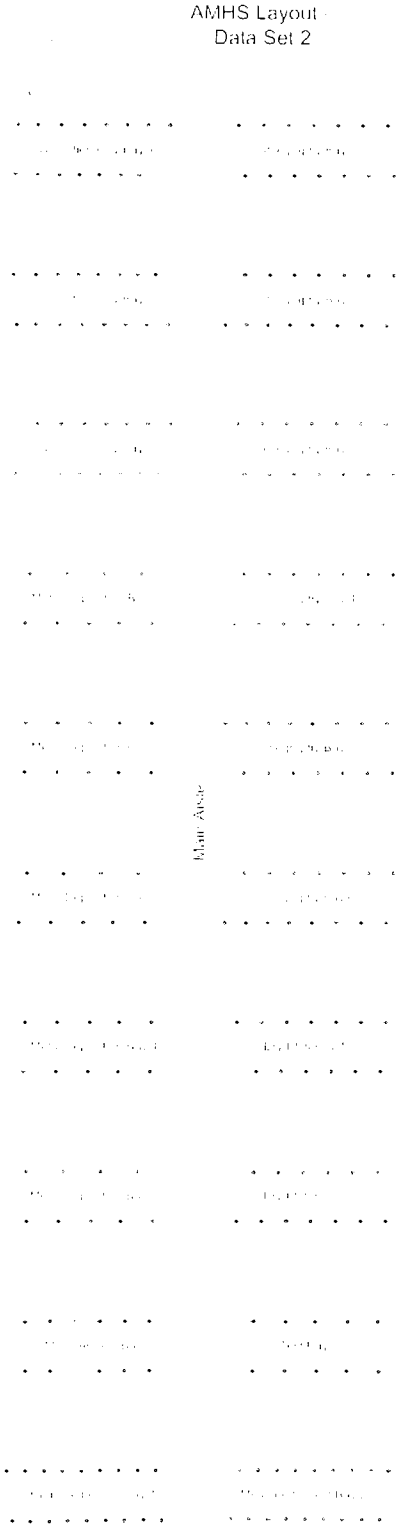


Figure 1.1: Fab and AMHS Layout for SEMATECH Data Set 2

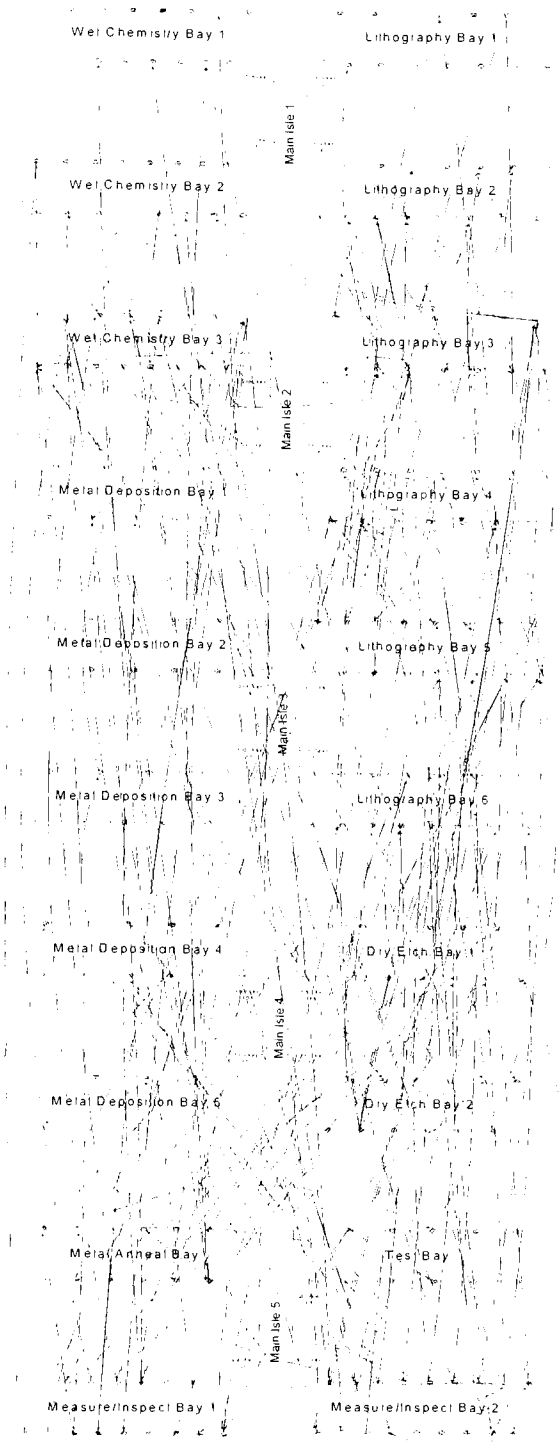


Figure 1.2: Process Route for a Single Product from SEMATECH Data Set 2

a company. One increasingly important aspect of semiconductor manufacturing is material handling. As the size (currently up to 300 mm in diameter) of silicon wafers used in semiconductor manufacturing increases, fully automated material handling systems (AMHS) are essential. There are multiple reasons why this is true. Chrisos & Nestel-Patt (1998) list four. First, an automated material handling system can help improve yield. Geometries on the wafers are being built smaller and smaller. As they become smaller, the same amount of particle contamination can cause more damage. Without contact with workers, contamination is decreased which increases yield. There is also less of a chance that the wafers will be physically damaged. One pod full of 300 mm wafers can be worth up to $1 million. Just one dropped by an operator can cause significant loss. Second, tools are becoming more expensive and fabs want to make sure they are fully utilized. Automation can help make manage lots and make sure they are at tools when they are needed. Third, a lot can weigh over 20 lbs. At this weight, it is not safe for operators to be carrying throughout a shift. Fourth, as time goes on, the process of manufacturing semiconductors becomes more and more complex. To optimize fab operations and remain competitive and profitable, new software will need to be used to schedule each material handling step. For this to become a reality, the material handling system in the fab must be fully automated.

In a fab, the automated material handling system (AMHS) is a major contributor to overall performance. To illustrate this point, Figure 1.1 shows a representative layout of a semiconductor fab where processes are arranged in bays along a main aisle. In this example, a typical interbay/intrabay material handling systems is depicted in which automated guided vehicles travel along the main aisles and within the bays to move lots from one process to the next. In Figure 1.2, the routing of a single product (wafer lot) is superimposed on the fab layout.

(To give an idea of scale, a fab may encompass as much as 200,000-300,000+ square feet). As a result of the large amount of material handling required, efficient material handling systems and control policies are required.

Preliminary studies show that the AMHS can account for up to 40% of the lot cycle times. Tool idle time due to material handling delays can also decrease tool availability and utilization significantly. The research presented here attempts to determine the relationship between fab performance and the utilization of the AMHS.

The first area of research focuses on the relationship between the capacity of the AMHS and fab performance as measures by key fab metrics including cycle time and throughput. A methodology is created to determine the capacity of AMHS and resulting effects on fab performance relative to AMHS utilization.

The second are of research focuses on fab control policies for AMHS and fab tools and their effect on fab performance. Although pervious research has been conducted on these two production control policies independently, this research investigates the interrelationship of AMHS and tool dispatching rules and their effect on fab performance.

2 PROBLEM STATEMENT

The semiconductor industry is known to have very high operation costs. One contributor to these high costs is the automated material handling system (AMHS). As discussed in the introduction, yield, tool utilization, ergonomics, and increased fab complexity are all factors that contribute to the need for automated material handling systems in semiconductor fabs. Furthermore, Papronty (2000) reports that projections for AMHS in 300mm fabs range from $50-100 million. Consequently, effective utilization and control of AMHS is essential. If the AMHS is used ineffectively, tool utilization can be decreased significantly due to delays caused by the material handling system causing a loss of potential return on a high cost of investment. The goal of this study is to improve the efficiency of fabs through effective use and understanding of automated material handling systems, the rules that control them, and their interactions with fab processes.

While simulation is a very good tool to model semiconductor factories, it does not have the capability to easily assign vehicle dispatching rules to the AMHS. The first step is to develop these rules through a link with AutoMod and customization. Next, a good method to evaluate the capacity of an AMHS through simulation will be developed. Currently, suppliers list the capacity of their systems in the moves it is capable of in a set period of time. However, this does not give the fab much information about the point at which the AMHS will begin to have a negative effect on operations. Different dispatching rules for selecting which lot should be moved next by a vehicle will then be considered. It appears reasonable that the time at which a lot is needed at a tool would be closely related to the assigned machine dispatching rule. To take this into account another simulation study will be performed to study the various combinations of vehicle dispatching rules with machine dispatching rules to attempt to optimize the overall

system. The study will test if an interaction can be found between vehicle and machine dispatching rules in the semiconductor industry. The best combinations of rules will also be identified.

The ability to effectively evaluate AMHS capacity and to identify the best combination of control rules would arm semiconductor manufacturers with a new understanding of how to improve the operations of their AMHS. This in turn will begin to reduce the high cost of these systems since more capacity will be achieved from a smaller, less expensive system. More importantly, shorter transportation times can make shorter lead times and an increased throughput possible. These results will enable semiconductor manufacturers to more efficiently utilize their equipment and become more competitive.

3 LITERATURE REVIEW

There are five important areas of background and research that are relevant to the problem outlined. First, a background of semiconductor fabrication is reviewed to obtain an understanding of the unique processing environment involved. Next, a review of AMHS research shows a general overview of the work that has been performed in the area. Specifically focused on dispatching rules, both machine and vehicle dispatching rules have been shown to have a significant impact on fab performance in separate lines of research. Finally, studies involving an interaction between machine and vehicle dispatching rules in a flexible manufacturing (FMS) environment are discussed.

3.1 Semiconductor Fabrication Background

The goal of semiconductor manufacturing is to process silicon wafers to build diodes and transistors that create computer chips. The process begins with bare silicon wafers that are grouped into lots that typically consist of about 25 wafers. These lots travel through the fab and are processed together. An important aspect of semiconductor manufacturing is the reentrant flow that wafers follow throughout the fab. Each feature created in the wafer requires a round of processing. Although the number of processes included and the order may change, these processes must be repeated again and again for each additional feature. Some of the main processes in the semiconductor industry are: lithography, ion implantation, chemical vapor deposition, oxidation, diffusion, etch, chemical mechanical polish, metallization, and clean. Each of these processes is described below and is referenced from Wolf and Tauber (2000).

3.1.1 Processing Steps

Lithography is the process of drawing a pattern on the wafers. First, the wafer is coated with a photoresist. Next, certain areas of the wafer are exposed to light. A mask or reticle

blocks the light from areas that are not exposed. The photoresist is then developed and the unwanted portions of the photoresist are removed. The wafer is then etched. In this step, the pattern on the photoresist is transferred to the silicon dioxide. When the wafer is processed in subsequent steps, only areas that are not patterned with the silicon dioxide layer will be exposed for processing.

Ion implantation is the process where charged atoms are shot into the wafer. This is done to create impurities and dope the wafer changing its properties. After the impurities are introduced the wafer must go through an anneal to make the impurities electrically active.

Chemical vapor deposition (CVD) is a process where gas is deposited onto the wafer to create a film. Thin films are commonly created by CVD. Since the material is deposited on the wafer, this process does not consume any of the original wafer.

Oxidation is the process of creating a layer of silicon dioxide on the wafer. This is done by heating the wafer in presence of an oxidizing ambient. These silicon dioxide layers have many uses in a semiconductor device including insulation, gate oxides, and as a blocking layer for etch and implant.

Diffusion is the movement of impurities in the wafer. At a high temperature, impurities in or around the wafer will move in a direction to attempt to create an even percentage of impurities throughout the wafer. Diffusion can be used to drive in dopants introduced during implantation or to introduce new dopants that are supplied by a source.

Etch is the process of removing material from the wafer with a chemical reaction. Wet etch is performed by submersing the wafer into a liquid etchant. This method etches isotropically. Because etching happens in all directions at the same rate, a large undercut will occur. If a pattern is being etched into the wafer, some of the area underneath the mask layer

will be removed along with the exposed area. Dry etching on the other hand is anisotropic and will not remove much material that is underneath the mask layer.

A chemical mechanical polish is used for planarization. As the wafer is processed, the surface develops bumps and becomes uneven. This causes a problem during the lithography step, since it may cause some of the wafer surface to be outside of the depth of field. Chemical mechanical polish is performed by rotating a wafer on a polishing pad that contains slurry.

Metallization is the step where the metal that is deposited onto the wafer. This metal is used for layers of contacts to doped areas of the wafer and interconnects.

Cleaning is important during wafer processing because any small particles on the wafer can seriously damage the product and contaminate equipment. There are many methods that can be used to clean a wafer including chemically assisted removal, electrical repulsion of the particle and the surface, and mechanical dislodgement.

3.1.2 Process Routes

A lot follows a predefined process route throughout the fab. A typical process route may require a lot to be processed at up to 600 steps. A sample route is shown in Figure 1.2. Each arrow in the figure shows the lot being moved from one tool to the next. This route is process route 1 taken from data set number 2 from the SEMATECH data sets, which can be found at http://www.eas.asu.edu/~masmlab/home.htm. These data sets are supplied anonymously from actual factories for research purposes. This specific dataset is for a factory that performs only backend operations for seven ASIC and memory products. While the route itself is from an actual factory, factory layout data is not provided. The layout shown is created to mimic a reasonable layout for the factory; however the actual layout may be different. Since the route is only for backend operations, it is relatively small at 198 process steps. Many of the

tool sets are visited by a single lot 8 or 9 times. Since fabs are typically arranged with similar tools placed together, the next stop in a lot's route is rarely close by. According to Chrisos & Nestel-Patt (1998), a 300-mm wafer can travel approximately 8-10 miles though its process route. From the figure alone, it is easy to see the large amount of traveling that a lot must perform and the importance of an efficient material handling system.

3.1.3 Fab Layout

Figure 1.1 displays the fab layout created for SEMATECH data set 2. This is the same fab shown in Figure 1.2 without the process route. The solid dots represent tool locations, the solid triangles represent turntables on the main aisle, the squares with sharp edges represent stockers, and the remaining lines represent tracks of the AMHS. This is a typical fab layout. Tools are placed into bays according to tool type. Each of these bays can be accessed from a main aisle. Tools of the same type are placed near each other. The first reason for this type of layout is due to the nature of the processes. All of lithography must be performed under yellow lights. If lithography tools were spread over the fab, either the entire fab would be required to have yellow light or the processes must be fully enclosed. There are also contamination issues with metals used and other processes. A second reason is the complicated process routes that the lots must follow. It would be extremely difficult to arrange tools to minimize material handling with such long process routes and reentrant flow for one part type let alone multiple that could be produced in a single fab. Extra tools and clean room space can not be added frivolously because of the high cost of each. While the vast majority of fabs use the bay layout, a few, like AMD Fab 30 in Dresden, use a work cell layout.

3.2 AMHS Research

The most commonly used material handling system is made for the bay fab layout. A separate system is used for movement between bays, interbay movement, and for movement within a bay, intrabay movement. In this setup, a lot is moved by the intrabay system from the tool it finished processing on to a stocker in the bay near the center aisle. Stockers are the main storage areas for lots, but there also may be small buffer storage near each tool. Stockers also act as a connection point between the interbay and intrabay systems. From the stocker, the lot is moved by the interbay system to the stocker of the bay where it will be processed next. From this second stocker, it is moved by the intrabay system for the new bay to the tool it will be processed on. The interbay and intrabay systems do not communicate with each other. For movements that need to be made from one tool to another within the same bay, the lot may go directly from tool to tool or through the stocker depending on the system setup and the queue length.

According to Lin et al. (2001), there are four types of intrabay systems available to fabs: rail-guided vehicles (RGV), automated-guided vehicles (AGV), personnel-guided vehicles (PGV), and overhead hoist transporter (OHT). Some fabs may not even use an intrabay material handling system. These fabs require the use of operators to carry lots directly from the stocker to the tool. There may be one or more vehicles per bay that can move bi-directionally or uni-directionally. Some fabs may use different types of intrabay systems depending on the volume requirements for specific bays.

Interbay transporters are usually overhead monorail or hoist systems. However, studies have compared overhead monorail systems with vehicles to continuous transportation systems that use conveyor belt type equipment. The most common configuration for an overhead system

consists of loops of track that have turntables where lots can be transferred between loops. These turntables provide an improvement over a single track loop because they allow the lot to turn around or cross the aisle without traveling all the way to the end of the aisle. This type of setup is shown in Figure 3.2. As with an intrabay system, the tracks may be uni-directional or bi-directional. Other layouts have also been used. One type consists of a loop with two uni-directional tracks, each which carries wafers in opposite directions. This method provides the ability to place multiple vehicles on each track without worrying about deadlock.

Much research has been done on automated material handling systems in the semiconductor industry using simulation. Simulation is a good tool for this environment because of the complex interactions between the processes that would be difficult to account for in an analytical method. However, most of the research to date is focused on the design of these systems.

Several researchers deal with comparing two or more different types of material handling systems. Kurosaki et al. (1997) compare two material handling systems. The first uses a monorail for interbay movement and Automated Guided Vehicles (AGVs) for intrabay movement. In the second system, both interbay and intrabay movement is completed by one overhead transportation system that can move lots directly from tool to tool. They conclude that separate interbay and intrabay systems were recommended for fabs in which there is a large amount of movement within the same bay, and a single system is recommended for fabs in which there is a large amount of movement between bays.

Paprotny et al. (2000) perform a simulation study to compare continuous flow systems to overhead monorail material handling systems. They found that the average delivery time of the overhead monorail system is half that of the continuous flow system and the standard deviation

of the delivery time is greater for the overhead monorail system. They conclude that in practice the advantages and disadvantages of each system should be considered by the specific fab before a system is implemented.

Once the type of material handling system is selected the layout must be designed to fit the fab in question. Nadoli and Pillai (1994) discuss the use of a simulation model of a material handling system during design to obtain information about performance indicators, such as stocker utilizations, waiting time for vehicles, delivery time, vehicle utilization, and vehicle capacity lost due to congestion. They also test several scheduling algorithms for an intrabay rail guided vehicle. Similarly, Jefferson et al. (1996) use a simulation model to determine stocker and vehicle utilization at different wafer start rates along with the time needed to deliver batches of lots to the furnaces in diffusion.

Mackulak and Savory (2001) compare intrabay material handling layouts with one stocker per bay and one stocker for every two bays each using overhead hoist vehicles. They use a simulation model built in AutoMod to compare these systems. Using the average delivery time as a performance indicator they found that the system with one stocker for each bay performs better than the second system. However, they point out that there can be a great cost savings by decreasing the number of stockers, so both systems may still be considered.

While the above studies compare specific material handling systems, other research attempts to generate general guidelines to design. Pillai et al. (1999) stress the importance of designing the layout of the fab and the material handling system together. This way trade offs can be made between the two. Otherwise, the material handling system would need to adapt to whatever fab layout it is given.

Sturm et al. (2003) propose a method to evaluate alternative AMHS designs. They show that AMHS performance is normally evaluated with global performance metrics, but that local problems such as vehicle congestion or stocker overflow could occur if they are not specifically looked into. To avoid this problem, they use a three level approach: first calculate global performance indicators, then calculate local move performance statistics, and lastly calculate node to node congestion. Nazzal and Bodner (2003) propose another method to evaluate AMHS designs. They list seven stages for AMHS design: manufacturing model construction, profile analysis, architectural AMHS design, elaborative physical AMHS design, elaborative behavioral AMHS design, AMHS model construction, and model evaluation and finalized design.

All of the above papers focus on creating the most efficient design of a new material handling system. However, most engineers will have to work with a material handling system that is already in existence. A material handling system that worked well when the fab was built may not be optimal as new tools are added and the fab evolves. Even if the system meets the current requirements, there can be areas of improvement. Although extremely valuable in the design of material handling systems, these papers do not provide much help in increasing the efficiency of an existing material handling system. On the other hand, Glüer (2002) states that among other areas of the fab, material handling systems need continuous monitoring and improvement in order to stay competitive. Starting with a simulation model, he developed other tools, like Petri Net models and heuristics, to provide quick guidelines that can be used for daily operation decisions. Glüer (2002) goes into more detail of one of these tools, MaxFlow theory, which uses a simple calculation to estimate the remaining availability of the material handling system providing information that can help prioritize jobs.

Mori, et al. (2001) look into the material handling system of a fab that has added a second line to increase capacity. In this case the existing material handling system must work with the tools and layout that is added in the new area. They propose a control method to manage movements between the lines.

3.3 Vehicle Dispatching Rules Research

Vehicle dispatching rules are used to assign a vehicle and lot to each other to be transported. According to Egbelu and Tanchoco (1984) vehicle dispatching rules can be separated into two categories: vehicle-initiated and work center-initiated. A vehicle-initiated rule is followed if a vehicle is looking for work and has more than one task to choose from. A work center-initiated rule is used if a task arrives to a queue and has multiple empty vehicles to choose from. Only one of these categories of rules can be active at any given time.

Because of their high cost, most material handling systems in the semiconductor industry are not designed with a large amount of excess capacity. Therefore, rarely will a lot be required to choose between two or more idle vehicles and work center-initiated rules are used far less than vehicle-initiated rules. For those occasions, the rule most widely used chooses the nearest vehicle to minimize unloaded travel time for the vehicles.

The choice of vehicle-initiated dispatching rules has a greater impact on the efficiency of the system. Examples of common vehicle-initiated rules are: First Encountered First Served (FEFS), Longest Waiting Time (LWT), Shortest Travel Time (STT), and Largest Queue Size. FEFS chooses the lot that is closest to the vehicle minimizing the unloaded travel time. LWT chooses the lot that has been in queue the longest. This rule attempts to avoid a large variation in the amount of time that a lot must wait to be transported, which overall should produce less variation in the lots cycle time and easier prediction of completion dates. Largest queue size

chooses the lot from the stocker or tool that has the most lots waiting at it. This rule attempts to minimize the occurrence of storage areas becoming filled to capacity so that no other lots could be brought in. The machine dispatching rules listed above can also be used as vehicle dispatching rules. For example, if SPT were used as a vehicle-initiated dispatching rule, the lot with the shortest processing time at the next tool would be transported.

Most of the research on vehicle dispatching rules is for flexible manufacturing systems (FMS) or AGV systems that are not specific to a semiconductor fab. However, this research provides valuable background information and many of the concepts discusses may be applicable to the semiconductor industry.

Yamashita (2001) performs a study to compare three work center-initiated dispatching rules for an AGV system with multiple vehicles. The first rule is "first encountered first served" (FEFS), the second rule is "the nearest vehicle in time" which minimizes the waiting time of the first part that arrives, and the third rule is "the policy of the nearest vehicle in distance" which "dispatches the vehicle in the nearest position to the earliest arriving item even if this vehicle is not available before any other vehicle in the system as long as the increase of the waiting time of the item is less than a certain value." Using a Markov chain-based state-space reduction technique to calculate waiting time distributions, he finds that the mean waiting for FEFS is smaller than the waiting time for the other two rules.

Some recent research has been done on vehicle dispatching rules in the semiconductor industry. Lin et al. (2001) perform a study on the effect of different dispatching rules on average transport time, waiting time, throughput, and vehicle utilization in a double-loop interbay AMHS. Only one work center-initiated rule is used, nearest idle vehicle. Two vehicle-initiated rules are used: first encounter first served (FEFS) where the vehicle moves to the nearest waiting

lot, and longest waiting time (LWT) where the vehicle moves to the lot that has been waiting the longest. Different rules for selecting the loop on which the vehicle should travel are also considered. To compare these rules, a simulation model was run at different levels of transportation flow rate and vehicle amount. The SD-NV/FEFS, with SD as the loop selection rule, is the top performer out of the tested rules. It is shown that the selection of the dispatching rule can have a significant effect on performance indicators.

Sigireddy, et al. (2003) perform a study comparing five scheduling rules to prioritize lots in queues waiting for vehicles in a semiconductor fab. The rules are: MaxWIP defined as the "destination bay with maximum WIP", Most-Choices defined as "destination bay with most 'number of choices' where 'number of choices' for a bay is defined as the total number of unique destinations for all the lots at its stocker", Longest Processing Time (LPT) defined as the "farthest destination bay amongst all the lots at its current location", and Shortest Processing Time (SPT) defined as the "nearest destination bay amongst all the lots at its current location." FIFO is used as a baseline since it is commonly used in practice by default. These rules are compared using two simulation models, one of a small fab and one of a large fab. In the small fab, all of the rules perform almost identically except for SPT which performs poorly. In the large fab, which is more representative of a real fab, the MaxWIP and Most-Choices rules perform better than the others based on cycle time and queue build-up. The authors conclude that "there could be considerable performance gain with intelligent and well-tested scheduling rules over FIFO in situations with high variability and high queue formations." However, this study only looks at scheduling rules to prioritize lots at each stocker, and does not consider dispatching rules to match a lot with a vehicle.

Hasenbein et al. (2004) perform a simulation study with a simplified fab model to compare five vehicle-initiated dispatching rules. The fab does not include rework, scrap, finite buffers, or maintenance. Processing times are deterministic. The only randomness present in the fab is created by the arrival rate. The dispatching rules considered are FIFO, Shortest Travel Time, Longest Travel Time, Most Choices, and MaxWIP. It is found that MaxWIP produces the shortest wafer cycle time through its entire route. According to the authors, "if further simulation bears out the trends presented here, it would indicate that at least a 5 percent to 10 percent reduction in cycle time could be achieved by making simple alterations in vehicle scheduling algorithms." They also note the need to test additional rules on more complex fab models.

3.4 Machine Dispatching Rules Research

Dispatching rules are used at each process step to determine which lot out of those waiting should be processed next. Some common dispatching rules used in the semiconductor industry are: Shortest Processing Time (SPT), Earliest Due Date (EDD), Critical Ratio (CR), First In First Out (FIFO), Shortest Number in Next Queue, and Random. The SPT rule selects the lot with the shortest processing time at the current step. This way, the lots that can be processed more quickly do not have to wait for a long job to be completed. A shortcoming with this rule is that long jobs tend to get stuck in the system when there are long queues and shorter jobs continuously are processed before them. To avoid this problem, there is often a time limit placed on how long a job can wait in the queue before it is assigned the next to be processed. The EDD rule selects the lot with the earliest due date to process attempting to minimize the number of late jobs. An issue for this rule is that lots toward the end of their process routes are generally given more priority since they are usually closer to their due dates. The CR rule takes

this issue into account. This rule chooses the lot with the minimum ratio of time until its due date over its remaining processing time. This ratio will be negative if the lot is already late, between zero and one if there is more remaining processing time left than there is time until its due date, and greater than one if the time until due date is greater than the remaining processing time. FIFO simply processes jobs in the order they arrive. This rule is often used by default or as a secondary rule to break ties created by another rule being used. The Shortest Number in Next Queue rule selects the job that has the least number of lots waiting in queue at the next processing step in its route. This rule attempts to minimize the tools' idle time by trying to fill queues that are or may soon become empty first. In a fab that does not have a set dispatching rule the operator selects the lot to be processed next. This is often done at random. Many more dispatching rules are used in the semiconductor industry; there are too many to list. AutoSched AP has 27 built in rules to choose from for use in a simulation. Researchers continue to create rules that are variations or combinations of existing rules or new rules altogether.

Machine dispatching rules have been more thoroughly researched in the semiconductor industry than vehicle dispatching rules. However, background research can still be found in other areas. An early paper by Blackstone et al. (1982) reviews and compiles past studies on dispatching rules. Thirty-four dispatching rules are defined including remaining processing time, shortest imminent operation, earliest due date (EDD), minimum slack time, random selection, first in first out (FIFO), shortest number in next queue, critical ratio, seven that are combinations of rules listed in the paper, etc. They stress that cost effectiveness is the "only relevant measure of performance for a dispatching rule", but flowtime, lateness, and tardiness are commonly used since they are easier to compute. Compiling past studies, they find that in general the rule SI

worked the best overall. However, this rule may create some very late jobs. The SI rule selects the job with the shortest processing time.

O'Neil (1991) performs a study to investigate lot start distributions, equipment loading rules, and nine different machine scheduling rules. The scheduling rules considered are: FIFO, shortest next process time, shortest total remaining processing, least lots in the next queue, random, EDD, critical ratio, and slack. He finds through a simulation experiment that the combination of rules used causes a significant impact on WIP and cycle time. The critical ratio rule shows the best results for the fab studied. However, this study only looks at a single fab and is difficult to generalize.

Mittler and Schoemig (1999) use simulation models of two large fabs to compare five machine dispatching rules based on the start rate weighted sum of the mean and variance of cycle time. The rules studied are Minimum Inventory Variability Scheduling (MIVS) taken from Li, Tang and Collins (1996), fluctuation policies for the mean of the cycle time (FSMCT) taken from Lu et al. (1994), fluctuation policies for the variance of cycle time (FSCVT) also taken from Lu et al. (1994), FIFO, and EDD. For the first fab, both the lowest mean and standard deviation of cycle time are given by FSVCT. For the second fab FSVCT still results in the lowest standard deviation of the cycle time. However, the lowest mean cycle time is achieved with MIVS. Because of the different results on the two models, this study shows the importance of testing dispatching rules on more than one type of fab. No suggestion is given to attempt to explain the characteristics of the fabs that caused the different results.

Recent research has shifted into the area of dynamic machine dispatching rules. The idea is that as the WIP mix and the fab state changes, the optimal dispatching rule may also change and a method should be in place to periodically monitor a determine the best rule. Hsieh et al.

(2001) perform a study to evaluate four dispatching rules combined with a wafer release policy that could be reviewed and changed each week. The dispatching rules considered are: FSVCT, LDF, OSA taken from Li et al. (1996), and FIFO. Throughput rate and layer mean cycle time are used as performance indicators. The fab studied normally operates under the FSVCT dispatching rule. However, it is found that during a circumstance of unusual machine failure it can be beneficial to change to the LDF dispatching rule.b

3.5 Interaction between Machine and Vehicle Dispatching Rules

Researchers have shown in FMS and AGV systems that machine and vehicle dispatching rules are closely related. A study by Sawik (1995) researches the benefits of simultaneously scheduling vehicles and machines in a FMS environment. In this study, the same rule is always used for both machine and vehicle-initiated dispatching rules, and combinations of these rules with the machine selection rules are considered. The machine and vehicle dispatching rules are: Shortest Total Processing & Transportation Time (STP&TT), Shortest Uncompleted Processing & Transportation Time (SUP&TT), Least Work Remaining (LWKR), Least Uncompleted Work Remaining (LUWKR), Most Work Remaining (MWKR), and Most Uncompleted Work Remaining (MUWKR). The machine selection rules are: Shortest Transportation Time (STT), Smallest Queue Size (SQS), Largest Queue Size (LQS), Largest Workload & Transfer Time (LW&TT), and Largest Uncompleted Workload & Transfer Time (LUW&TT). This study does not specifically calculate an interaction, but found that the combination of rules SUP&TT and SQS or LUW&TT works best when looking at the total schedule length. Although machine and vehicle dispatching rules are always the same in this study, this provides indication that the methods used to schedule machines and vehicles are related.

Sabuncuoglu and Hommertzheim (1992) perform an experiment to compare twelve machine scheduling rules and six vehicle scheduling rules in an FMS environment. The machine scheduling rules are: SPT, LWKR, MWKR, FCFS, smallest value of operation time multiplied by total operation time (SPT.TOT), smallest value of operation time divided by total operation time (SPT/TOT), largest value of operation time multiplied by total operation time (LPT.TOT), largest value of operation time divided by total operation time (LPT/TOT), fewest number of operations remaining (FOPNR), most number of operations remaining (MOPNR), first arrived first served (FAFS), and random. The AGV scheduling rules used, which are vehicle-initiated, are: FCFS, MWKR, largest output queue size (LOQS), shortest traveling distance (STD), largest queue size including incoming and outgoing parts (LQS), and fewest number of operations remaining (FOPNR). These rules were tested on one FMS layout varying machine and AGV load levels, queue capacities and AVG speeds. It is found that SPT and SPT.TOT are the best machine scheduling rules in regard to mean flowtime regardless of the vehicle scheduling rule, and STD and LQS are the best AGV rules regardless of the machine scheduling rules. However, once again, interaction effects are not calculated.

As part of their study on the design and evaluation of AGV systems, Raju and Chetty (1993) investigate the effect of machine and vehicle scheduling rules using a Petri net. Five vehicle-initiated dispatching rules are used: random, SPT, shortest travel time (STT), total processing time (TPT), and remaining processing time (RPT). Four work center-initiated dispatching rules are used: random, longest idle vehicle (LIV), nearest idle vehicle (NIV), and least utilized vehicle (LUV). Four machine scheduling rules are used: SPT, TPT, RPT, and random. When varying each of these rules by themselves, SPT and TPT are shown to be the best machine scheduling rules based on buffer sizes and machine utilizations, STT is found to be the

best vehicle-initiated dispatching rule based on machine and AGV utilization, and NIV is shown to be the best work center-initiated dispatching rule based on machine and AGV utilization. When orthogonal experimentation is performed to look at the combination of these rules for batches of products, different results are obtained for batches with different volume requirements. The first batch type shows rules RPT SPT and NIV to work the best, the second TPT SPT and NIV and the third TPT STT and NIV Since the optimal combination of rules changes due to the batch type it is not clear what effect is provided by the interaction of these rules.

Lee and Maneesavet (1999) performs a simulation experiment comparing five rail-guided vehicle dispatching rules and three machine scheduling rules in a FMS environment. Throughput, flow time, and WIP are used as performance indicators. The RGV dispatching rules are concerned with pushing and pulling empty and full pallets which does not apply to the semiconductor industry. However, the important result from this experiment is that the vehicle dispatching rules, the machine scheduling rules, and their interactions are found significant by an ANOVA test.

These papers show that the possibility of an interaction between machine and vehicle dispatching rules has been studied by researchers. In at least one case an interaction is found to be significant, and in others good combinations of rules are recommended. On the other hand, no papers could be found that study these effects in the semiconductor industry. It seems reasonable that the interactions would also occur in a fab, but this is not proven and with its unique environment, different rules or combinations of rules may prove optimal.

4 SIMULATION METHODOLOGY FOR FABS HAVING AMHS

Research in the area of semiconductor manufacturing must take into account the large amount of interactions and uncertainty in the environment. This research aims to optimize fab performance through the effective use of an AMHS. First, by developing a method to determine AMHS capacity needs for a fab. Second, by using an enhanced understanding of tool and vehicle dispatching rules. With the complex environment of semiconductor fabs, mathematical and traditional methods cannot accurately represent the system. To obtain meaningful results, simulation is used for research purposes.

Simulation is widely used for research in the semiconductor industry. Mackulak and Savory (2001), Mittler and Schoemig (1999), and Lee and Maneesavet (1999), each used simulation in their semiconductor related research. More information about these studies, and others, can be found in the literature review section.

A simulation model can create a virtual representation of a fab. The model can then be run to mimic all of the activities that take place in the fab in a specified time frame. Different scenarios can be compared and statistics can be collected to compare them. During runs of the model, wafers flow through the fab and encounter all of the steps and transportation that would occur in the actual fab. The virtual representation of a running fab allows for complexity and uncertainty to easily be taken into account.

To reduce the time required to build a model and also reduce run time, model simplifications and assumptions must be made. One common simplification is to model the AMHS by providing distributions of travel times that the lots normally encounter. This simplification can work well for many studies, but is insufficient for studies specifically relating

to the AMHS. It would be impossible to optimize the performance of the AMHS without specifically modeling the movement of vehicles in the system.

It is important to use realistic data when building a model of a semiconductor fab with an AMHS for research purposes. As outlined in the previous sections, a semiconductor manufacturing fab is unlike any other manufacturing environment. Guidelines and rules created for other manufacturing environments in general may not apply in the semiconductor industry. Similarly, research performed for the semiconductor industry must use data that is as closely related to the actual operating data of a fab as possible.

The best data to use when creating a simulation model is actual data collected from real fabs. In this research, data of actual fabs from the SEMATECH data sets was used. The SEMATECH data sets are supplied anonymously by real fabs to be used for research purposes. These data sets can be found on the Arizona State University website at http://www.eas.asu.edu/~masmlab/home.htm. Along with each data set, fab descriptions and sample simulation results are provided.

A major problem encountered in modeling the SEMATCH data sets for use with AMHS studies is that data for the layout and material handling system is not provided. Some of this data may not even exist for the fabs. The data sets are up to 10 years old which is before the time that fully automated material handling systems even existed. In this research the layout and AMHS information was created to mimic a typical fab. The fabs are laid out with similar tools grouped in bays that can each be reached from a central aisle. The material handling systems are comprised of an interbay system that carries lots between bays along the central aisle and an intrabay system for each bay to carry lots within the bay. Both the interbay and intrabay systems are fully automated overhead systems set up as a loop of track. The interbay system includes

turntables to shorten travel distance across the aisle. At the end of each bay is a stocker used to store lots and transfer them between the interbay and intrabay systems. The layout and specifications for the AMHS were built with the help of Dr. Karl Hirschman in the Microelectronics department at RIT and Texas Instruments. Similar layouts are widely used in the literature. Examples of studies that use similar fab layouts are Pillai et. al. (1999) and Kurosaki et. al. (1997). Since these are by far the most commonly used fab and AMHS layouts, others were not studied.

Models of two separate SEMATECH data set fabs are used in this research. The first is data set 2, a smaller fab that produces a small number of ASIC products. The second is data set 2, a larger make-to-order fab that produces a larger range of products. The two fabs make good modeling choices because they contain data such as rework, preventative maintenance, and down schedules that is missing from other datasets. Also, the fact that the two fabs are different sizes and have product mixes make it possible to compare the results of two different types of fabs.

The simulation models for this research are built in Brooks Automation software AutoSched AP, which is specialized simulation software for semiconductor fabs. However, AutoSched AP does not have the desired capabilities to model the layout and AMHS of the fabs in detail. AutoMod, also by Brooks, was used to model the material handling system. The two models are connected by the amap extension provided by Brooks.

4.1 ASIC Fab

The data set for the ASIC fab is a fab with seven ASIC and memory products, 97 different types of tools, approximately 10,000 wafer starts per month, and an average of 26 process steps per mask layer. Rework, scrap, and batching were modeled. A sample product routing for this model can be found in appendix A.

Figure 1.1 shows the layout of the fab and AMHS for data set 2. The layout consists of a main aisle and twenty tool bays. Each bay is 20 ft. wide and 75 ft. long. There is one stocker present for each bay shown in the figure by the square blocks. Each dot represents a tool in the fab and the triangles represent turntables on the interbay AMHS. Twenty vehicles operate on the main aisle and can move anywhere along the main aisle track. Each bay also has one or two vehicles dedicated to the bay depending on its specific needs.

4.1.1 Model Assumptions

Assumptions the model of data set 2 include:

- Operators were not modeled. The fab was assumed to have a fully automated AMHS.
- Reticles were not modeled
- The need for certain lots to be processed on specific tools in a family was not modeled.
- Setups and maintenance were modeled as given in the data set on a mean time to failure and mean time to repair basis.
- Rework and scrap probability were listed both by lot and by wafer. Since AutoSched will only allow for one of them, only rework and scrap probability by wafer was modeled.
- A wafer travel time within the tool was listed in the dataset. In the model, this time was added into the processing time.
- While load and unload times were specified in the route for each product in the dataset, AutoSched required the load and unload times to be specified per tool. The average load and unload times per tool across all of the process routes were used.

- Travel time between tools was also listed in the process routes in the dataset. However, this time was not used in the model. Travel time was determined in AutoMod by the actual time a vehicle took to retrieve and move the lot.
- In the dataset, the maximum batch size at a step in the process route was often less wafers than the lot size. This caused the lots to be stop at that tool without being processed. When the simulation was run for 300 days, the result was that zero lots completed processing. To resolve this problem, the minimum and maximum batch size units were changed from wafers to lots. If the maximum batch size was less than the lot size, the processing time was multiplied by the number of batches that fit in the lot. The new maximum batch size was always rounded up. If the maximum batch size was greater than the lot size the maximum batch size was divided by the lot size. The new maximum batch size was always rounded down. The processing time was left alone.
- Buffer areas near each tool were not present.
- Stockers were set to an infinite capacity.
- Vehicles could pass other vehicles that were stopped for loading or unloading at a tool or stocker.
- Other complexities involved in semiconductor manufacturing that were not provided in the data set were not included.

4.1.2 Verification and Validation

The SEMATECH dataset came with sample simulation results of the same factory. The results of the AutoSched model were compared with the results SEMATECH results. The simulation was run for four years with a one year warm up period. The simulation was only run

in AutoSched and did not include the material handling system built in AutoMod in order to more closely represent the conditions of the SEMATECH sample simulation. Because the fab no longer had an AMHS, operators were modeled. It was assumed that operators did not take breaks during their shifts, or if they did, another operator was there to replace them and only one operator was required to run each tool. They were only required during load and unload times. The sample simulation was run at 95% capacity, but the AutoSched model could only be run at 70% before the WIP at bottleneck stations increased beyond control. This is most likely due to the fact that operators and rework were not modeled in the sample simulation, but they were included in the AutoSched model.

The product processing times for the sample and AutoSched models are shown in Table 4.1 below. The SEMATECH results were stated as the raw processing times. It is not clear if load and unload times were included. The AutoSched results include load, unload, and travel times along with raw processing times. Given these differences, the results are reasonably close.

Table 4.1: Processing Time for Each Product in ASIC Fab

Product Number	SEMATECH Results (hrs)	AutoSched Results (hrs)
1	164.2	182.4
2	195.8	216.0
3	153.1	172.8
4	178.8	196.8
5	237.3	259.2
6	50.2	57.6
7	187.5	206.4

The average cycle times for the sample and AutoSched models are shown in Table 4.2. These differences can most likely be attributed to the fact that rework and operators were not modeled in the sample simulation.

Table 4.2: Average Cycle Time for Each Product in ASIC Fab

Product Number	SEMATECH Results (hrs)	AutoSched Results (hrs)
1	293	344
2	389	470
3	328	406
4	342	363
5	499	568
6	63	91
7	423	431

4.2 Make-to-Order Fab

The make-to-order model was built by Laubisch (2003) in AutoSched AP. The fab processes 11 different memory products. This fab contains 73 different types of tools, approximately 21,400 wafer starts per month, and an average of 35 process steps per mask layer. Rework, scrap, and batching were modeled. However, a material handling system was not modeled and no fab layout data was provided. Typical fab and AMHS layouts, similar to those used for the ASIC fab, were created in AutoMod. Figure 4.2 shows the layout of the fab and AMHS for the make-to-order. This fab is larger than the first fab. Each bay is 100 ft. long. Twenty-seven vehicles operate along the main aisle, and each bay contains between one and six vehicles.

4.2.1 Model Assumptions

Assumptions for the model of the make-to-order fab include:

- Operators were not modeled. The fab was assumed to have a fully automated AMHS.
- Reticles were not modeled
- Travel time between tools was also listed in the process routes in the dataset. However, this time was not used in the model. Travel time was determined in AutoMod by the actual time a vehicle took to retrieve and move the lot.

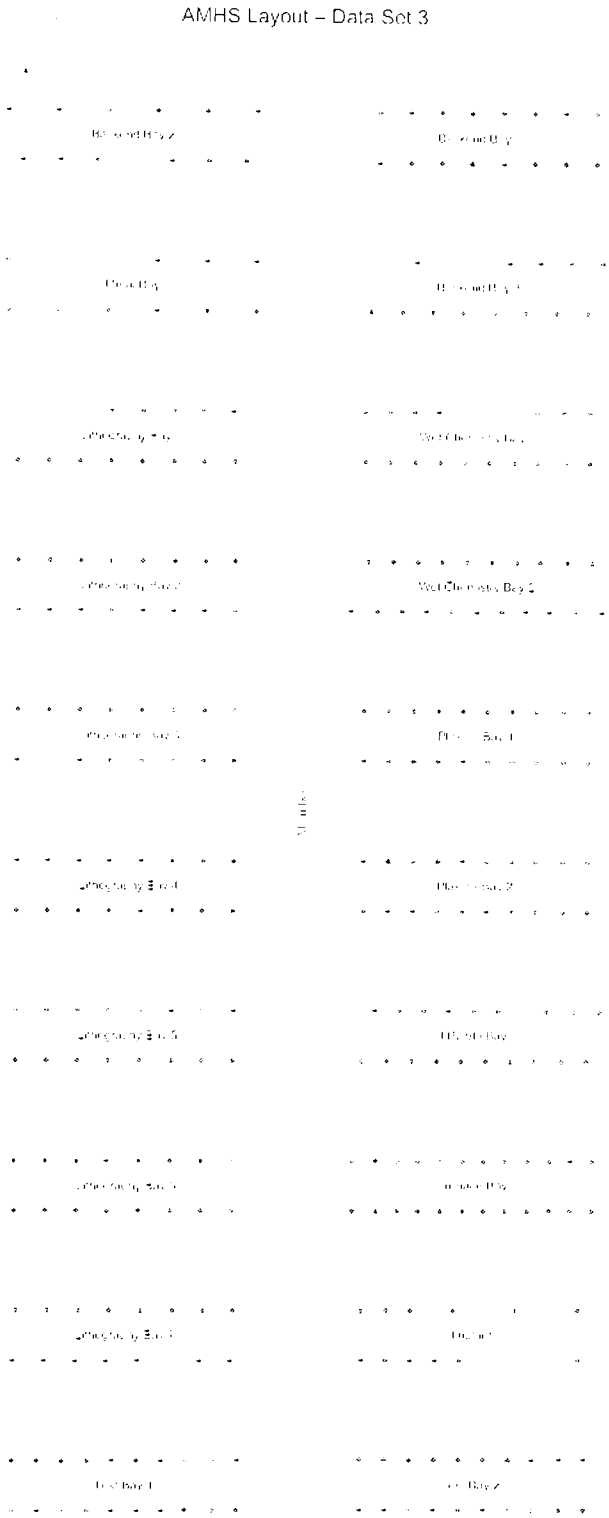


Figure 4.2: Fab and AMHS Layout for Make-to-Order Fab

- Buffer areas near each tool were not present.
- Stockers were set to an infinite capacity.
- Vehicles could pass other vehicles that were stopped for loading or unloading at a tool or stocker.
- Other complexities involved in semiconductor manufacturing that were not provided in the data set were not included.

4.2.2 Verification and Validation

The verification and validation of the make-to-order fab was performed by Greg Laubisch and is taken from Laubisch (2003). One replication of the model was run with a warm-up period of 10,000 hours and a run length of 50,000. This replication was compared with a sample run provided with the data set. Throughput and cycle time were collected. Table 4.3 shows the throughput values for the results provided by SEMATECH and the results obtained from the sample run. The throughput results, along with the cycle times which differed from the sample results by and average of 7.3%, are each very close. The results suggest that the model is adequate.

Table 4.3: Throughput for Each Product for Make-to-Order Fab

Product	SEMATECH Throughput (lots)	AutoSched Throughput (lots)
A	2445	2449
AT	687	686
C	686	687
D	586	587
F	489	490
H	5974	5976
R	5195	5192
U	1906	1909
X	3234	3233
Y	2056	2056
Z	540	540

4.3 Summary

The use of simulation models provide a good method of capturing the complexities of semiconductor manufacturing this research encounters. The two simulation models that are used are representative of realistic fabs through the use of actual fab data and widely used layouts. The models are used in two experiments in this research. The first, a methodology for determining the capacity of an AMHS, is described in Chapter 5. The second, the study of AMHS and tool dispatching rules, is described in Chapter 6.

5 CAPACITY ANALYSIS

5.1 Scope

The goal of the capacity analysis experiment is to develop a good understanding of the relationship between the size and capacity of a material handling system and the requirements for operations of a fab. Currently, suppliers may list the capacity of their systems in the moves it is capable of in a set period of time. However, this does not give the fab much information about the point at which the AMHS will begin to have a negative effect on operations. Also, an AMHS that is operating to optimize its own performance may unknowingly be causing adverse affects on other operations. This experiment attempts to show the importance of using a performance indicator that takes entire fab performance into account. It also provides a guideline to determine the required capacity of an AMHS for a specific fab based on the maximum utilization of AMHS vehicles. This experiment was presented at the Winter Simulation Conference and is published in the proceedings, Kuhl and Christopher (2004).

5.2 General Methodology

To determine the capacity of an AMHS a base simulation model must first be created that contains fab operations in the desired amount of detail in addition to the AMHS whose capacity is to be evaluated. This base model is then modified to make the AMHS the bottleneck. Making the AMHS the bottleneck is an important step to insure that changes to the performance indicators are actually caused by the AMHS. To do this, the bottleneck tools are first identified by running the model and finding the tools with the largest queue times and utilizations. Duplicate tools are added to these bottleneck areas, and the process is repeated until the introduction of additional tools has no effect on the lot cycle time. The AMHS model is not modified in this process.

To collect capacity data, the model is run at varying release rates. The data is studied to determine the point at which cycle time began to sharply increase. The main performance indicator used in this experiment is the average cycle time of the lots through their entire route. Other statistics to collect include WIP and AMHS vehicle utilization. Since it is made the bottleneck of the system, any increase in cycle time can be attributed to the AMHS. The point at which cycle time begins to sharply increase can then be mapped to the maximum vehicle utilization that appears during that release rate. If an increase occurs below a maximum utilization of 100%, the experiment will show that degradation to the entire fab can occur before the AMHS is fully utilized, and the entire fab performance should be taken into account when studying an AMHS. The required number of vehicles for the fab can then be calculated to not exceed the maximum utilization. Table 5.1 shows a summary of these steps.

Table 5.1: Summary of Methodology for determining required AMHS capacity

Step	Action
1	Build a base simulation model of the fab with an AMHS
2	Make the AMHS the bottleneck of the fab
3	Run model at varying release rates
4	Determine release rate at which cycle time sharply increases and match to maximum vehicle utilization
5	Calculate vehicles required in the system based on the maximum utilization found

5.3 Experiment Performed Using ASIC Fab

The capacity analysis experiment is performed on the simulation for the ASIC fab. This model contains an AMHS, built in AutoMod with all other fab operations in AutoSched AP. Fab operations are based on real fab data obtained from the SEMATECH data sets. AMHS and layout data are not available for the real fab, but a material handling system and layout are created to mimic those typically found in fabs. The tools of the fab are arranged in bays according to tool type. Each bay can be accessed from a main aisle running down the length of

the fab. The AMHS contains an interbay system, running down the main aisle of the fab used to transport lots between bays. Separate intrabay systems are present for each bay to handle the movement of lots within bays. One stocker is located in each bay to store lots and to act as a transfer point between the interbay and intrabay systems. Each system contains vehicles running unidirectionally. Greater model and fab detail can be found in chapter 4.

The AMHS is made the bottleneck of the fab using the method outlined in the general methodology section. The original model contains 277 tools. The number of tools increases to 469 to make the AMHS the bottleneck.

To collect capacity data, the model is run at varying release rates varying from 5,000 to 15,000 wafers/month. Cycle time is graphed against release rate to determine the release rate at which cycle time began to sharply increase. Maximum vehicle utilization is also graphed against release rate to determine the maximum vehicle utilization that corresponds to the release rate at which the cycle time begins to sharply increase. Since only one fab model is used in this analysis, the results for the specific utilization at which degradation occurs may not be valid for other fabs in general. The method however, can be applied to any other fab.

5.4 Results

The raw data results for this experiment are included in Appendix B. The 95% confidence interval of the average cycle time of lots for all part types in the factory over the various release rates is shown in Figure 5.1. The average cycle time begins at approximately 9.44 days per lot at a release rate of 5,000 wafers per month. It then rises slowly until a release rate of approximately 9,000 wafers per month is reached, where the slope becomes much larger. This increase is due to the vehicle capacity since stockers are assumed to have an infinite capacity. The plot only shows release rates up to 10,000 wafers/month. When the simulation is

run with release rates up to 15,000 wafers/month, all runs with release rates above 10,000 wafers/month do not reach steady state. The fab is overloaded at these rates.

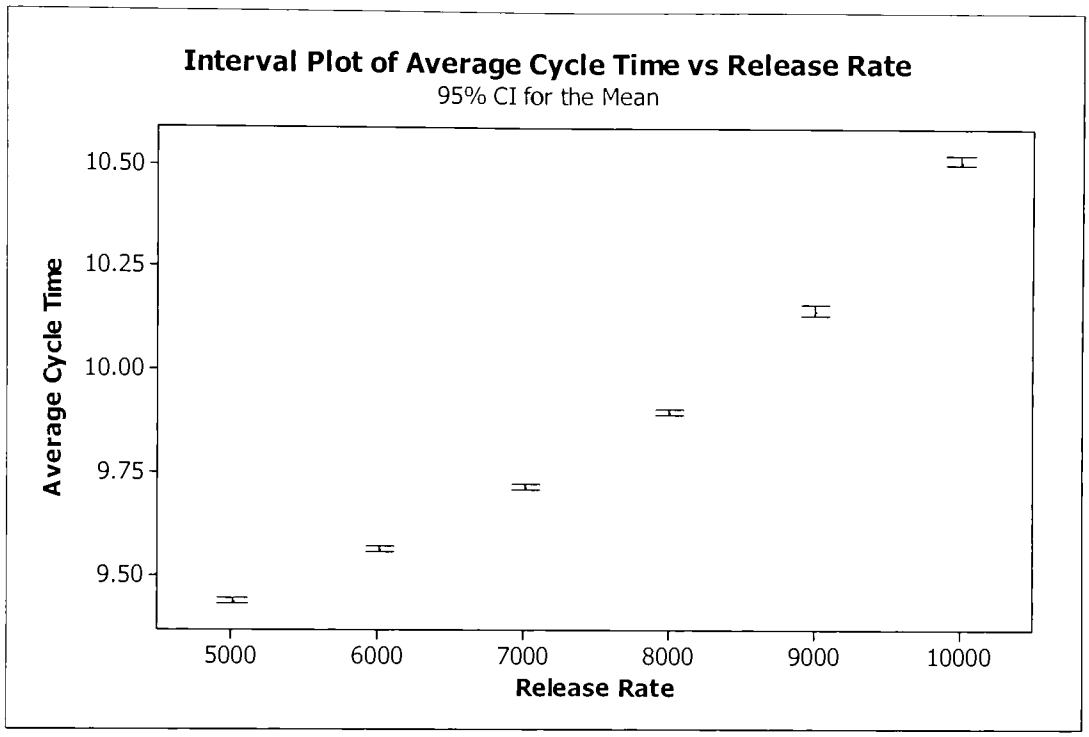


Figure 5.1: Average Cycle Time vs. Release Rate

The maximum vehicle utilization versus the release rates is shown in Figure 5.2. Comparing Figure 5.1 and Figure 5.2, it can be seen that the increase in slope in average cycle time from Figure 5.1 corresponds to a vehicle utilization of about 75% for the highest utilized vehicle.

5.3 Conclusions

Using average cycle time as the performance indicator, this experiment shows that the performance of a factory can be diminished by the material handling system before the point at which any of the vehicles are 100% utilized. When designing a material handling system for a factory or considering increasing wafer release rates, the capacity of the material handling system should be considered in conjunction with performance factors from production and not

only by itself. These results show that optimization solely focused on the AMHS can have unwanted affects on other areas and performance of the fab. While this material handling system would have been capable of running with vehicles utilized at 90% or higher, the negative effects to production may not have been acceptable.

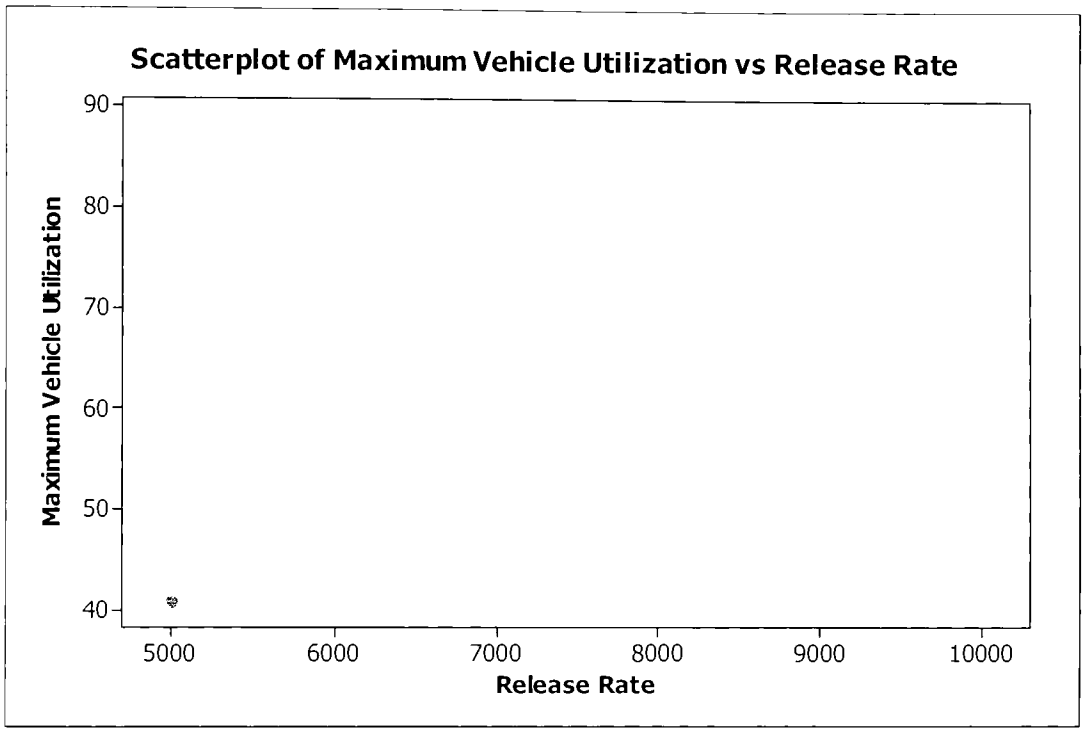


Figure 5.2: Maximum Vehicle Utilization vs. Release Rate

The methodology itself is an important outcome of this analysis. While specific results obtained from this experiment may not relate to all fabs, the methodology can be used for any fab in general. This method provides a good solution to determine the size and capacity of an AMHS required for a specific fab.

6 MACHINE AND VEHICLE DISPATCHING RULES

Machine and vehicle dispatching rules have been shown to have a significant impact on fab performance. An interaction between the two has also been found in other manufacturing environments. Additional detail on past research can be found in chapter 3. The hypothesis of this study is that as well as the separate contributions from machine and vehicle dispatching rules, the interaction between the two of them will have significant impact on fab performance. To test this hypothesis, a full factorial design experiment is performed.

6.1 Scope and Methodology

This experiment considers five machine dispatching rules and seven vehicle-initiated dispatching rules were considered. Machine dispatching rules are used at a tool to select the next lot in the queue to be processed. The machine dispatching rules evaluated are:

- **Shortest Processing Time (SPT)**: The lot with the shortest processing time at the current step will be chosen.
- **Earliest Due Date (EDD)**: The lot that is due first will be chosen.
- **First in First Out (FIFO)**: The first lot that arrived in the queue will be chosen.
- **Critical Ratio (CR)**: The lot with the lowest critical ratio will be chosen.
    - CR = (time until due)/(remaining processing time)
- **Critical Ratio including Queue time (CRQ)**: Similar to CR, the lot with the lowest CRQ will be chosen. This calculation, however, includes average queue time and average transit time along with remaining processing time.
    - CRQ = (time until due)/(remaining processing time + remaining average queue time + remaining average transit time)

The selection of machine dispatching rules is chosen to reflect commonly used rules in research for the semiconductor industry and also those that have been shown to work well. FIFO is chosen as a baseline rule and appears in most studies performed on dispatching rules both within the semiconductor industry and in manufacturing in general. It is listed in Blackstone et al (1999), along with CR and EDD, in a summary of early dispatching rule research. It is also used by O'Neil (1991) and Mittler and Schoemig (1999) in studies to compare machine dispatching rules in the semiconductor industry. CR, SPT, and EDD are all also used in O'Neil's study. EDD is used by Mittler and Schoemig (1999). At least two of the rules chosen to be used in this study have been shown as the top rule out of those selected in past research. O'Neil (1991) finds CR to work the best. Sabuncuoglu and Hommertzheim (1992) find SPT to work the best in a FMS environment. However, due to the complex environment, this rule may not be found to work as well in the semiconductor industry. CRQ is not found in any of the background research reviewed. It is chosen as an alteration to CR to take more of the time delays a lot is required wait for during its route. All of the rules chosen are standard rules in AutoSched AP, the simulation software used for this analysis. The rules chosen also encompass three main types of dispatching rules. FIFO is a rule based on the queue properties. SPT is a rule based on properties of the process that the lot is waiting for. EDD, CR, and CRQ are rules based on overall route properties.

Vehicle-initiated dispatching rules are used by a vehicle to determine the waiting lot that should be picked up next. Vehicle-initiated dispatching rules are only used when there are two or more lots waiting for transport. The vehicle-initiated dispatching rules that are considered are:

- **First Encountered First Served (FEFS)**: The lot which is closest to the vehicle will be chosen.

- **First in First Out (FIFO)**: The first lot that arrived in the queue will be chosen.
- **Shortest Travel Time (STT)**: The lot that had the shortest travel time will be chosen.
- **Earliest Due Date (EDD)**: The lot that is due first will be chosen.
- **Critical Ratio (CR)**: The lot with the lowest critical ratio will be chosen.
    - CR = (time until due)/(remaining processing time)
- **Critical Ratio including Queue time (CRQ)**: Similar to CR, the lot with the lowest CRQ will be chosen. This calculation, however, includes average queue time and average transit time along with remaining processing time.
    - CRQ = (time until due)/(remaining processing time + remaining average queue time + remaining average transit time)
- **Most Utilized Station (MUS)**: The lot traveling to the station family that has the highest utilization will be chosen. The goal is to keep bottleneck stations from starving. A simplification made in programming this rule is that the ranking for the utilization of the stations is determined by a run of the model with FIFO used for both the machine and vehicle dispatching rule.

FIFO, EDD, CR, and CRQ are chosen as vehicle dispatching rules to duplicate the machine dispatching rules to determine if using the same dispatching rules for both the machine and vehicle rules can be advantageous. Instead of SPT, STT is used for the vehicle dispatching rule to reflect the travel time the lot will encounter instead of the processing time. Raju and Chetty (1993) find STT to work the best in their study comparing vehicle dispatching rules for AGV systems. FEFS is chosen as a common vehicle dispatching rule. Yamashita (2001) finds it as the rule that works the best for an AGV system, and Lin et al (2001) find it to work the best

for a study specifically within the semiconductor industry. MUS is not found in the literature reviewed. It is created to attempt to improve utilization of the bottleneck stations.

Work center-initiated dispatching rules are used by a lot to determine which idle vehicle should transport it. These rules are only used when there are two or more idle vehicles for a lot to choose from. The work center-initiated vehicle dispatching rule is set to nearest idle vehicle (NIV) for all runs. This rule simply chooses the closest vehicle if a lot requests a transfer and there are two or more idle vehicles that could fill the request. Different work center-initiated rules will not be considered since it is rare that there would be two or more idle vehicles. NIV is chosen because it is the dispatching rule most commonly used.

Both the ASIC fab and Make-to-Order fab simulation models are used to run the experiment to determine if the results were applicable to more than just one specific fab. A full 2 factor factorial design is performed on each model. This requires 35 runs of each model, 70 runs total. Each run includes three 365 day replications and one 365 day warm up period. Performance indicators of average cycle time, standard deviation of cycle time, average WIP, X Theoretical value, and percentage on time are recorded. All of the performance indicators are chosen to take into account the performance of the entire fab as shown to be favorable in the capacity analysis in Chapter 5. These performance indicators have also been shown to be commonly used in the semiconductor industry, used in studies such as O'Neil (1991), Mittler and Schoenmig (1999), and Laubisch (2003). More information on these studies can be found in Chapter 3.

To set up the models for this experiment, the base models described in section 4 need to be altered for each dispatching rule. All of the altered models can be found in Appendix M on the CDs included with this thesis. The machine dispatching rules are changed in AutoSched AP.

All of the machine dispatching rules used are preprogrammed into AutoSched AP. They are applied by selecting the required rule under the FWLRANK and RWLRANK columns in the stn.txt file. The vehicle dispatching rules are changed in AutoMod. The work center initiated vehicle dispatching rule does not need to be changed since it is set to NIV by default. The vehicle dispatching rules are programmed manually except for FIFO, which is the default. These rules are programmed by altering the programming of the source files in the AutoMod model, and in some cases adding load attributes and variables. The EDD, CR, CRQ, and MUS vehicle dispatching rules all require lot attributes that are stored in AutoSched and are not available in AutoMod. To transfer this information from AutoSched to AutoMod changes have to be made to the amap extension written in C++. A detailed documentation of the AutoMod and amap programming required along with code can be found in Appendix C.

The models are run using the batch means method. Before the experiments can be run, the warm up period and batch length have to be set. Also, a due date for lots needs to be set to be used in the EDD, CR, and CRQ dispatching rules. The analysis used to calculate these parameters is included in Section 6.2, Model Preparation.

The results are analyzed with ANOVA to determine if there are interactions between machine and vehicle dispatching rules. If there are interactions present, it shows that an extra benefit can be obtained by good combinations of these rules, an performing studies that include both would create a greater benefit than studying them separately. Tukey tests are also performed to identify which combinations of rules outperform the others. Results between the two fabs are compared to determine if the same combinations of rules prove to work well for more than just one fab.

6.2 Model Preparation

The experiments are run using the batch means method. In this method, only one warm up period is run. The remainder of the run is split up into batches of equal size. The statistics from the batches are treated as though they are from separate replications for analysis. When using this method, the batch lengths must be long enough to ensure there is no autocorrelation between the batches. This method decreases the total amount of simulation run time since it requires only one warm up period. The total simulation time is an important consideration for these experiments since one replication can take up to 15 hours. Before the experiments can be run, the warm up period and the batch length must be calculated.

In addition, the due dates for the lots are not given in the data sets. These dates are needed in the EDD, CR, and CRQ dispatching rules. To obtain a useful output in the percentage of on time lots performance indicator, a reasonable due time is assigned. The due time for each part is assigned assuming that 85% of the lots would be on time under the baseline model with FIFO used for both the vehicle and machine dispatching rules.

6.2.1 Warm Up Period Determination

The length of the warm up period for each model needs to be long enough for the model to reach steady state before statistics are collected. Two statistics are recorded, cycle time and X theoretical value, and plotted in a time series graph to determine the point at which the models reach steady state. This analysis is performed in the baseline model with FIFO used for both the vehicle and machine dispatching rules for both data sets.

The plot of cycle time vs. time is shown in Figure 6.1 and the plot of x theoretical value vs. time is shown in Figure 6.2 for the ASIC fab. The data is recorded for 1000 lots exiting the system. The cycle time rises sharply for the first 50 lots, then rises slowly until lot 500 and

evens out. The X theoretical value shows a slight increase for the first 200 lots and evens out. The plots suggest that the model reaches steady state after 500 lots have been processed in the system. The 500th lot is completed 63 days after the model start date, making 63 days the minimum warm up period length. With the addition of a safety factor to account for variability and differences between dispatching rule scenarios, the warm up period is set to one year. This may seem like a very long period, but with lot cycle times of up to 20 days, the warm up period for this model will be much longer than models for other industries that may have cycle times in the range of hours.

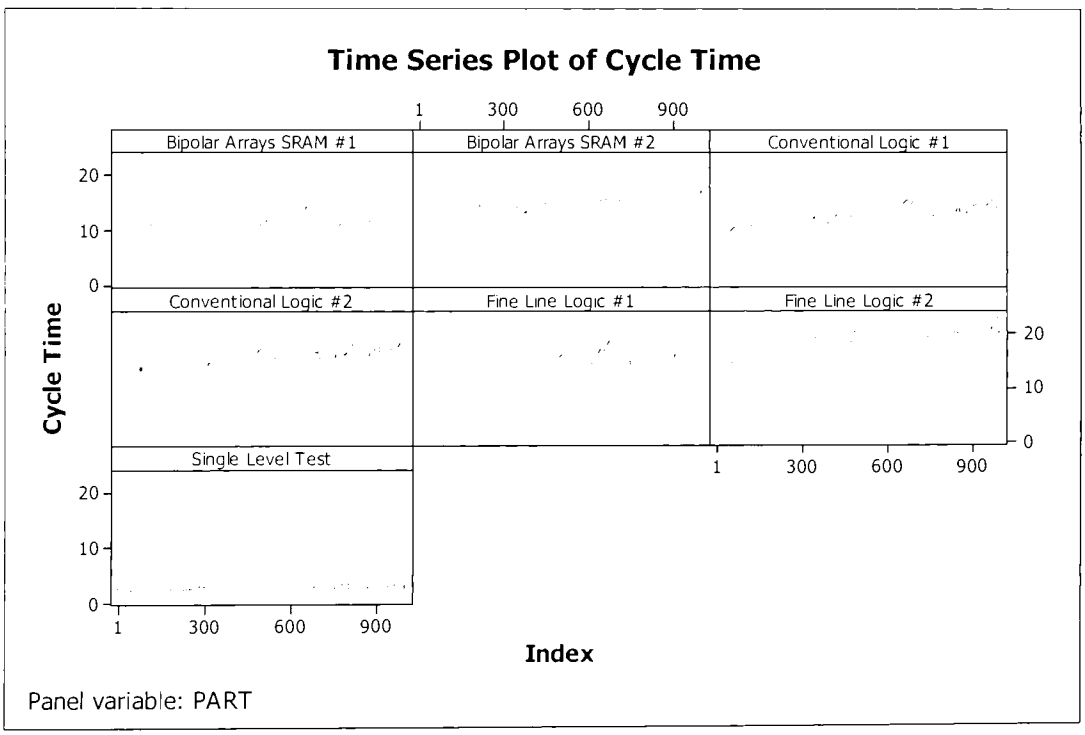


Figure 6.1: Time Series Plot of Cycle Time by Part for the ASIC Fab

The plot of cycle time vs. time is shown in Figure 6.3 and the plot of x factor vs. time is shown in figure 6.4 for the make-to-order fab. The data is recorded for 1000 lots exiting the system. The cycle time behaves similar to the cycle time in the ASIC fab, rising sharply for the first 50 lots, then slowly until approximately lot 500. The X theoretical value, on the other hand,

appears to be steady from the beginning. The plot suggests that the model for the make-to-order fab is also in steady state after 500 lots exit the system. In this model, the 500th lot is completed 56 days after the model start date. As in the ASIC fab, the warm up period is set to one year with the addition of a safety factor.

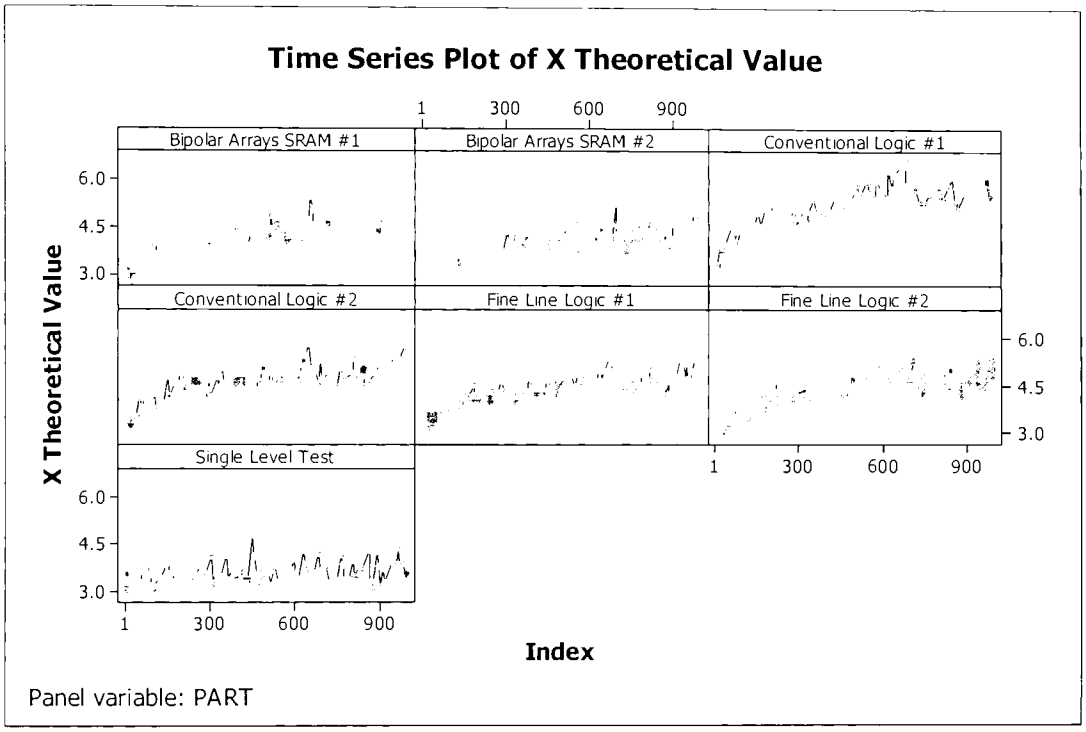


Figure 6.2: Time Series Plot of X Factor by Part for the ASIC Fab

The plot of cycle time vs. time is shown in Figure 6.3 and the plot of x factor vs. time is shown in figure 6.4 for the make-to-order fab. The data is recorded for 1000 lots exiting the system. The cycle time behaves similar to the cycle time in the ASIC fab, rising sharply for the first 50 lots, then slowly until approximately lot 500. The X theoretical value, on the other hand, appears to be steady from the beginning. The plot suggests that the model for the make-to-order fab is also in steady state after 500 lots exit the system. In this model, the 500th lot is completed 56 days after the model start date. As in the ASIC fab, the warm up period is set to one year with the addition of a safety factor.

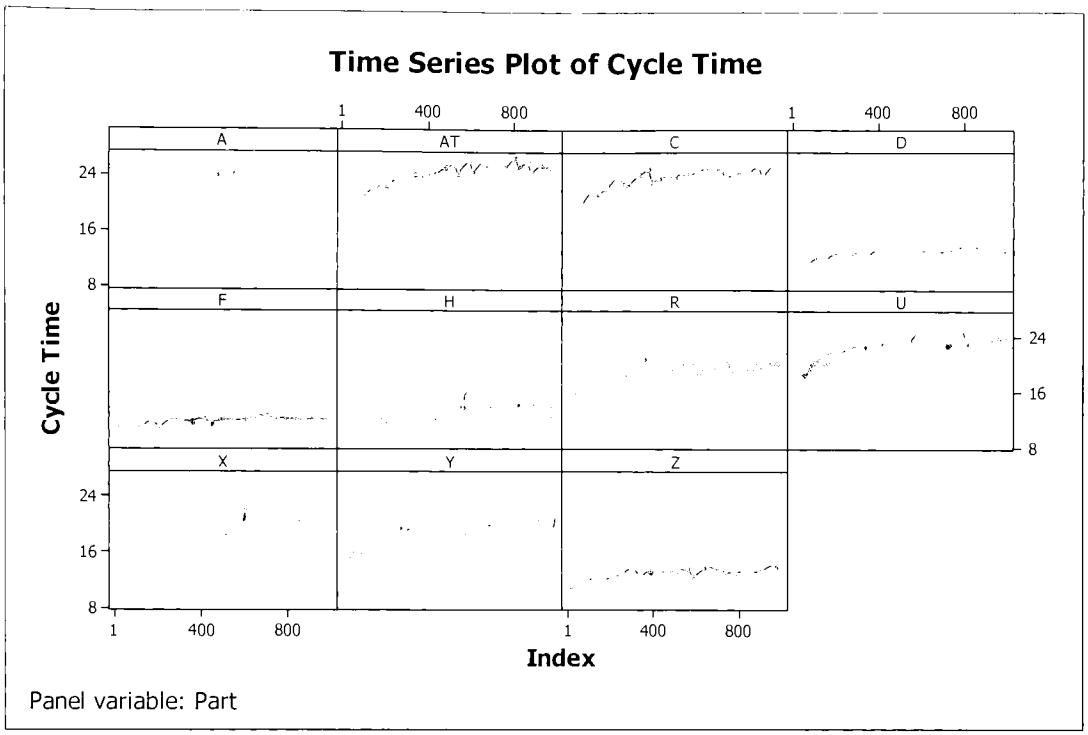


Figure 6.3: Time Series Plot of Cycle Time by Part for Make-to-Order Fab

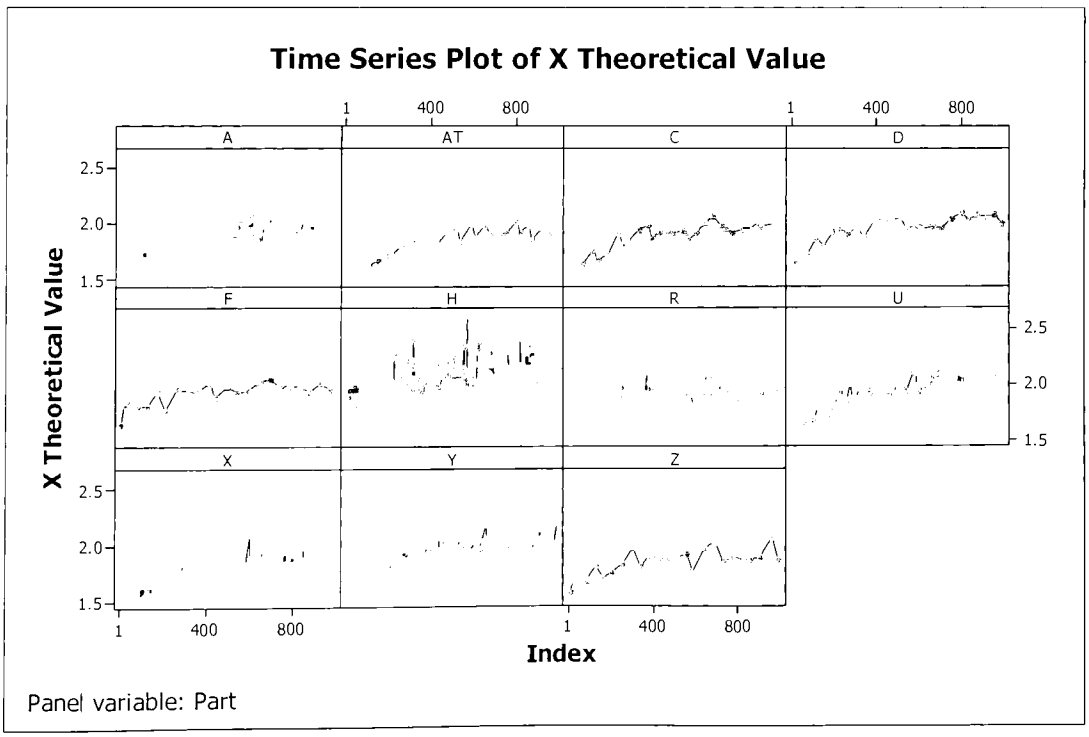


Figure 6.4: Time Series Plot of X Theoretical Value by Part for Make-to-Order Fab

6.2.2 Batch Length Determination

The length of each batch must be set long enough so that the statistics appear as though they are taken from separate replications. Otherwise, statistical analysis performed may not be valid. To determine the minimum batch length, the autocorrelation of the cycle time of lots is graphed for each data set. Figure 6.5 shows the plot of autocorrelation vs. the number of lags for the FIFO baseline model of the ASIC fab. Although difficult to see in the graph, the first lag at which the autocorrelation reaches zero is lag 4265. Therefore, with a minimum of 4265 lot completions in a batch, the statistics between batches do not show a correlation. At this point they act like separate replications. A batch length of 4265 lot completions corresponds to approximately 412 days. Normally, this number would be multiplied by a safety factor of about 10 to obtain the batch length. However, with the long run time that would be required for a 4120 day batch a smaller safety factor is chosen. The batch lengths are set to 2 years for this model

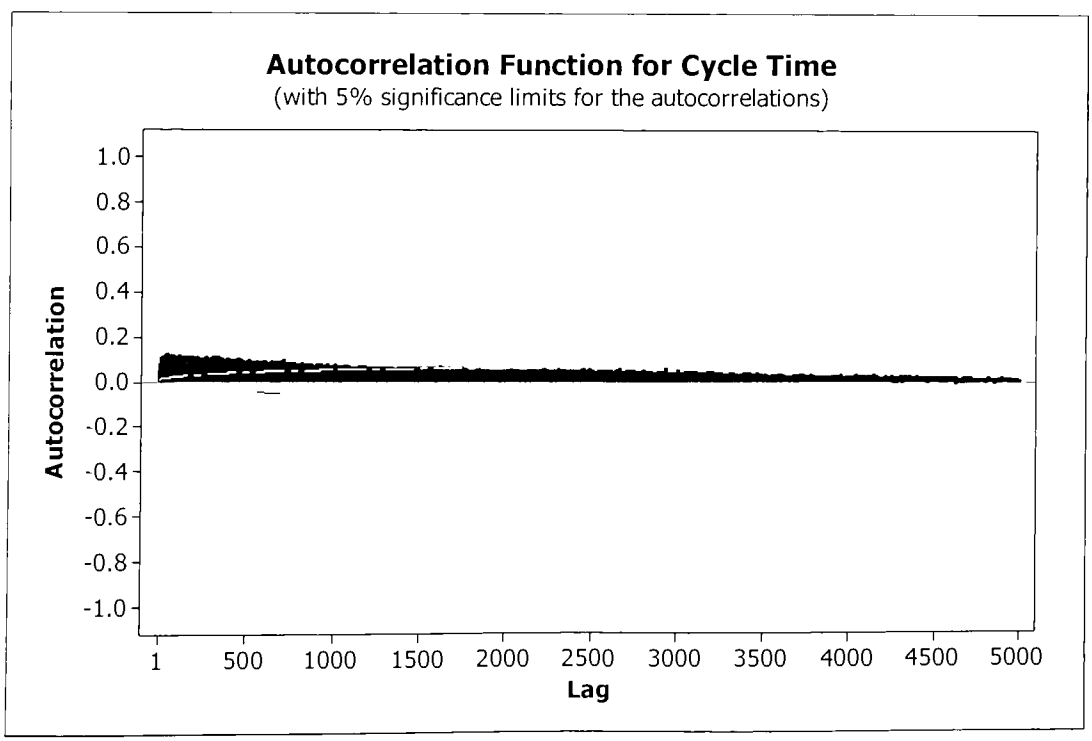


Figure 6.5: Autocorrelation Plot for ASIC Fab

Figure 6.6 shows the plot of autocorrelation vs. the number of lags for the FIFO baseline model of the make-to-order fab. This plot looks much different than that of data set 2. The autocorrelation is almost nonexistent from the beginning, and it reaches zero after only 16 lags. However, a batch length of only 16 lags would create batches that are shorter than the cycle times of the lots. Although this batch length does not show autocorrelation, it would not intuitively be a good choice. The batch length should be several times the cycle time. Otherwise, statistics may be severely influenced by small occurrences or bubbles of the same product type leaving the fab. Taking all of this into account along with simulation run time, the batch length for this model is set to 1 year.

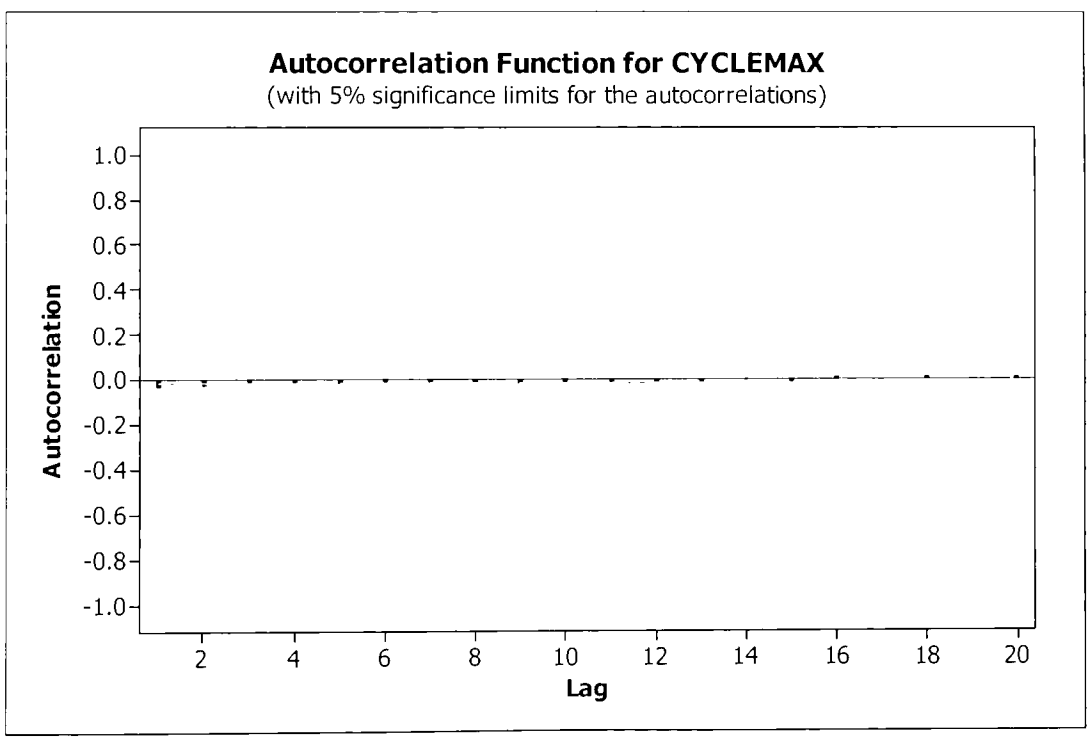


Figure 6.6: Autocorrelation Plot for Make-to-Order Fab

6.2.3 Lot Due Date Determination

In industry, the due date of a lot or part may be calculated based on when it is needed for the customer. However, no customer orders or requirements are present in the models. To set a

due date, it is assumed that 85% of the lots would finish on time. To calculate the due dates, a sample run with five replications of the FIFO baseline model is performed. The average cycle time and standard deviation of cycle time from the sample run of the ASIC fab are shown in Table 6.1 for each part type. Assuming a normal distribution for cycle time, the time from the start of the lot until its due date is calculated as the time for which 85% of the lots would complete processing on time. Looking at a normal distribution table, 85% of all observations fall below 1.04 standard deviations from the mean value. This makes the formula to calculate the time until a lot is due as follows:

Time to Due Date = 1.04(Standard Deviation of Cycle Time) + Average Cycle Time

The calculated time until the due date for each part is listed in Table 6.1 for the ASIC fab. That time is then added to the start date of the first lot of each part released in the simulation to obtain the due date for the first lot of each part, which is listed in the Due Date column. These due dates are placed in the DUE field in the order.txt file of the model. AutoSched automatically assigns the subsequent lots of each part a due date that allowed the same period of time from lot start date to due date. The model is then run a second time to determine if the normality assumption holds true and the percentage of on time lots is close to the calculated value of 85%. The percentage on time varies from 78.822% to 86.094% for the different part types. The percentage for each part type is listed in Table 6.1. For the purposes of this experiment, these values are sufficient.

The analysis for due date calculation is repeated for the make-to-order fab. Table 6.2 lists the average cycle time and standard deviation for each part. As in the ASIC fab, the cycle time is assumed to have a normal distribution, and a target of 85% on time delivery is set. A second run shows that using the due dates listed in 6.2, the percentage of on time lots range from

83.41% to 87.36% for each part. The percentages of on time lots for all of the parts are listed in 6.2. For the purposes of this experiment, these values are sufficient.

Table 6.1: Due Date Calculation Data for ASIC Fab

PART	Average Cycle Time	Standard Deviation	Time to Due Date	Recorded % On Time	Due Date
Bipolar Arrays S 1	15.76	2.20	18.05	78.82	02/02/04 09:13:37
Fine Line Logic 1	21.24	3.24	24.62	79.15	02/09/04 00:49:34
Conventional Log 1	17.89	2.81	20.81	79.80	02/05/04 07:28:15
Bipolar Arrays S 2	16.84	1.89	18.80	81.86	02/03/04 09:09:52
Fine Line Logic 2	27.04	4.08	31.28	79.57	02/15/04 22:46:05
Single Level Tes	3.22	0.37	3.60	86.09	01/19/04 08:30:31
Conventional Log 2	19.37	1.97	21.42	82.38	02/06/04 06:01:28

Table 6.2: Due Date Calculation Data for Make-to-Order Fab

PART	Average Cycle Time	Standard Deviation	Time to Due Date	Recorded % on time	Due Date
Y	19.67	0.67	20.36	87.31	01/21/00 16:41:00
R	20.03	0.65	20.70	85.31	01/22/00 00:53:28
X	19.52	0.60	20.14	85.84	01/21/00 11:16:25
U	23.62	0.88	24.53	85.83	01/25/00 20:47:31
A	24.36	0.82	25.21	84.90	01/26/00 12:59:48
H	12.84	0.50	13.36	86.25	01/14/00 16:34:39
C	24.34	0.83	25.20	86.46	01/26/00 12:48:52
AT	25.23	0.81	26.07	83.41	01/27/00 09:40:31
F	12.06	0.47	12.54	84.67	01/13/00 20:59:54
Z	13.26	0.46	13.74	87.36	01/15/00 01:43:18
D	13.56	0.51	14.09	86.91	01/15/00 10:03:33

6.3 Results

Raw data collected in this experiment along with full statistical analyses are included in the appendices. This section includes a summary of the results for each data set.

6.3.1 ASIC Fab Results

The raw data collected in this experiment for data set 2 is shown in Appendix D. As can be seen from this data, many of the runs that include SPT for the machine dispatching rule do not reach steady state. This indicates that the total WIP in the fab for these dispatching rule combinations will continually increase causing an overload. Because of this, SPT is considered a poor choice for the machine dispatching rule for this fab and is not included in the analysis. ANOVA tests at 95% confidence levels show that the machine dispatching rule, vehicle dispatching rule, and their interactions are significant for all of the performance indicators tested. The performance indicators are average cycle time, standard deviation of cycle time, average WIP, average X theoretical value, and percentage of on time lots. The ANOVA test for average cycle time is shown in Table 6.3. The remainder of the ANOVA tests are included in Appendix F.

Table 6.3: Two-way ANOVA of Cycle Time versus Vehicle, Machine for ASIC Fab

DF	SS	MS	F	P
6.00	309.18	51.53	86.90	0.00
3.00	162.03	54.01	91.08	0.00
18.00	103.02	5.72	9.65	0.00
56.00	33.21	0.59		
83.00	607.43			

Table 6.4 shows the results of the Tukey test performed for average cycle time on the ASIC fab. The bars under the significance column are drawn across rule combinations which cannot be shown to be statistically significantly different from each other. The remainder of the Tukey tests performed on all of the dispatching rules are included in Appendix H. Table 6.5 shows the top rule combinations from Tukey tests of all the rule combinations used for the ANOVAs. Under each performance indicator is the top group of rule combinations for that

performance indicator. The rule combinations in each group were not shown to be significantly different from each other.

Table 6.4: ASIC Fab Tukey Test on Average Cycle Time for all Rule Combinations

Dispatching Rule	Average Cycle Time (days)	Significance
VFEFS MFIFO	15.647	
VMUS MFIFO	15.78	
VFEFS MCR	15.907	
VEDD MFIFO	16.547	
VSTT MFIFO	16.627	
VFEFS MEDD	17.383	
VMUS MEDD	18.233	
VMUS MCR	19.127	
VMUS MCRQ	19.127	
VFEFS MCRQ	19.347	
VCRQ MFIFO	19.753	
VFIFO MFIFO	19.773	
VSTT MEDD	19.88	
VSTT MCR	19.953	
VSTT MCRQ	19.953	
VEDD MCR	20.7	
VEDD MCRQ	20.7	
VEDD MEDD	20.747	
VCRQ MCR	20.857	
VCRQ MCRQ	20.857	
VCR MCR	20.967	
VCR MCRQ	20.967	
VCR MFIFO	21.26	
VFIFO MCR	21.69	
VFIFO MCRQ	21.69	
VCRQ MEDD	24.5	
VCR MEDD	25.303	
VFIFO MEDD	26.48	

Dispatching rule combinations that appear in the top group in at least 4 out of the 5 performance indicators are highlighted in Table 6.5. These rule combinations are studied further to attempt to determine which rule combinations were the best among the top group. Tukey tests are performed a second time with only these top rule combinations. The top rule combinations from these second tests are shown in Table 6.6. Again, there is no statistical significant

Table 6.5: Rule Combinations in Top Significance Groups of Tukey Tests for ASIC Fab

Average Cycle Time	Standard Deviation of Cycle Time	Average WIP	Average X Theoretical Value	On Time %
VFEFS MFIFO	VFEFS MCR	VFEFS MFIFO	VFEFS MFIFO	VEDD MFIFO
VMUS MFIFO	VFEFS MFIFO	VMUS MFIFO	VMUS MFIFO	VFEFS MCRQ
VFEFS MCR	VMUS MFIFO	VFEFS MCR	VFEFS MCR	VMUS MFIFO
VEDD MFIFO	VSTT MFIFO	VEDD MFIFO	VEDD MFIFO	VMUS MCR
VSTT MFIFO	VEDD MFIFO	VSTT MFIFO	VSTT MFIFO	VMUS MCRQ
VFEFS MEDD		VFEFS MEDD	VFEFS MEDD	VSTT MFIFO
			VMUS MEDD	VSTT MCR
				VSTT MCRQ
				VFEFS MEDD
				VEDD MCR
				VEDD MCRQ
				VCR MCR
				VCR MCRQ
				VCRQ MCR
				VCRQ MCRQ
				VMUS MEDD
				VCRQ MFIFO
				VFIFO MFIFO
				VSTT MEDD

Table 6.6: Rule Combinations in Top Significance Groups of Tukey Tests on Top 6 Rule Combinations for ASIC Fab

Average Cycle Time	Standard Deviation of Cycle Time	Average WIP	Average X Theoretical Value	On Time %
VFEFS MFIFO	VFEFS MCR	VFEFS MFIFO	VFEFS MFIFO	VEDD MFIFO
VMUS MFIFO	VFEFS MFIFO	VMUS MFIFO	VMUS MFIFO	VMUS MFIFO
VFEFS MCR	VMUS MFIFO	VFEFS MCR	VFEFS MCR	VSTT MFIFO
VEDD MFIFO	VSTT MFIFO	VEDD MFIFO	VEDD MFIFO	VFEFS MEDD
VSTT MFIFO	VEDD MFIFO	VSTT MFIFO	VSTT MFIFO	

difference between the rule combinations in each group. The full Tukey Tests are included in Appendix I. Three rule combinations appear in the top significance group for all 5 of the performance indicators. These combinations are highlighted in the table. They are: MUS for the

vehicle rule and FIFO for the machine rule, EDD for the vehicle rule and FIFO for the machine rule, and STT for the vehicle rule and FIFO for the machine rule.

6.3.2 Make-to-Order Fab Results

The raw data for the make-to-order fab is included in Appendix E. As can be seen from this data, many of the runs that include STT for the vehicle dispatching rule do not reach steady state. As SPT is considered a poor choice for the machine dispatching rule for the ASIC fab, STT is considered a poor vehicle dispatching rule choice for the make-to-order fab and is not included in the analysis. However, when SPT is used for a machine dispatching rule in this data set, it does not show the drastically poor results and overload the fab as it does in the ASIC fab. For the make-to-order, SPT is included in the analysis. ANOVA tests at 95% confidence levels show that the machine dispatching rule, vehicle dispatching rule, and their interactions are significant for all of the performance indicators tested. These ANOVA tests are included in Appendix G.

When Tukey tests are performed on this data, the results are inconclusive. Almost all of the rule combinations are in one group where the combinations are not statistically different from each other. These Tukey tests are included in Appendix J. To try to reduce the variability present in the analysis, all of the rule combinations that do not reach steady state or are in the bottom significance group of the percentage of on time lots Tukey test are removed from the analysis. All of the rule combinations that are removed have a percentage of on time lots below 2%. With such a small amount of lots produced on time, these rule combinations are considered poor choices. Tukey tests are performed a second time using the remaining 17 rule combinations. Table 6.7 provides a summary of the rule combinations that appear in the top significance group for each performance indicator. The same six rule combinations appear in the

top significance group for all of the performance indicators. The full Tukey tests can be found in Appendix K.

Table 6.7: Rule Combinations in Top Significance Groups of Tukey Tests on Top 17 Rule Combinations for Make-to-Order Fab

Average Cycle Time	Standard Deviation of Cycle Time	Average WIP	Average X Theoretical Value	On Time %
VFEFS MFIFO	VCR MFIFO	VFEFS MFIFO	VFEFS MFIFO	VFEFS MFIFO
VMUS MFIFO	VCRQ MFIFO	VMUS MFIFO	VCRQ MFIFO	VCRQ MFIFO
VCRQ MFIFO	VFEFS MFIFO	VCRQ MFIFO	VMUS MFIFO	VCR MFIFO
VCR MFIFO	VMUS MFIFO	VCR MFIFO	VCR MFIFO	VMUS MFIFO
VFIFO MFIFO	VFIFO MFIFO	VFIFO MFIFO	VEDD MFIFO	VFIFO MFIFO
VEDD MFIFO	VEDD MFIFO	VEDD MFIFO	VFIFO MFIFO	VEDD MFIFO

The Tukey tests are then performed a third time with only the top 6 rule combinations shown in Table 6.5 that are found from the second set of Tukey tests. The top rule combinations from this third set of Tukey tests is summarized in Table 6.8. The full Tukey tests are included in Appendix L. The rule combination with the vehicle dispatching rule set to FEFS and the machine dispatching rule set to FIFO is the only rule combination that appears in the top significance group for all of the performance indicators.

Table 6.8: Rule Combinations in Top Significance Groups of Tukey Tests on Top 6 Rule Combinations for Make-to-Order Fab

Average Cycle Time	Standard Deviation of Cycle Time	Average WIP	Average X Theoretical Value	On Time %
VFEFS MFIFO	VCR MFIFO	VFEFS MFIFO	VFEFS MFIFO	VFEFS MFIFO
	VCRQ MFIFO			VCRQ MFIFO
	VFEFS MFIFO			VCR MFIFO
				VMUS MFIFO

6.4 Conclusions

The ANOVA results of this experiment show that the machine dispatching rule, the vehicle dispatching rule and their interaction are all significant for all performance indicators in

both fab data sets. The combination of vehicle and machine dispatching rules can have a significant impact on the overall performance of the fab. There is a reduction in the average cycle time of 20.87% in the ASIC fab, and 1.77% in the make-to-order fab in the rule combination with the lowest cycle time as opposed to a FIFO baseline. The presence of an interaction shows that there can be an extra benefit obtained by using rules that work well together. Also, rules that do not work well together can show a worse performance than expected. When deciding which vehicle dispatching rules are appropriate for a particular fab, engineers should study these rules in conjunction with the machine dispatching rules. Otherwise, the machine dispatching rule may no longer be the optimal choice for the new environment and fab productivity could be lost. These results reinforce the capacity analysis results that the entire fab performance should be taken into account when studying an AMHS and not just the performance of the AMHS.

The interaction between machine and vehicle dispatching rules appears in both of the fabs studied. This shows that it is possible for the interaction to exist, but does not guarantee that it will always be present. A case where the interaction may not be present is if the AMHS is extremely over or under utilized. If the AMHS is under utilized, there will always be a vehicle available to pick up a lot, and no vehicle dispatching rule will be needed for vehicle selection. Because of this, the vehicle dispatching rule, and hence the interaction, will not be significant. If the AMHS is over utilized, very long queues will form for vehicles. The tools will be starved of work waiting for the vehicles to transport the lots. Since there will be no queues in front the each tool, a machine dispatching rule will not be needed. The machine dispatching rule, and the interaction, will not be significant. Even though the interaction will not hold true in each and

every case, each fab has the possibility of added benefits or harm, and the interaction effect should be studied closely to optimize fab performance.

The Tukey tests are performed to determine which combinations of rules give the best performance. These tests are performed on both fabs to determine if the results can be generalized. Three optimal rule combinations are found for the ASIC fab that can not be differentiated statistically, and one optimal rule combination is found for the make-to-order fab. The optimal rule combination for the make-to-order fab, vehicle FEFS and machine FIFO, does not appear in the group of optimal rules for the ASIC fab. However, it is very close to the top group in the ASIC fab, appearing in the top significance level for four out of the five dispatching rules. One of the top rules for the ASIC fab, vehicle STT and machine FIFO, does not even make it to the last Tukey test for the top six rules in the make-to-order fab. This shows that while vehicle FEFS and machine FIFO is shown to be robust for the two fabs, the optimal combination of rules is highly dependent on the specific fab. A combination of rules that work well for all fabs in general can not be determined from these experiments.

Although most of the machine dispatching rules tested have an identical counterpart in the vehicle dispatching rules that are tested, none of the top dispatching rule combinations uses the same rule for both the machine rule and the vehicle rule. Is should not be assumed that a dispatching rule that works well in a fab as the machine dispatching rule should also be used for the vehicle dispatching rule of a new AMHS.

7 CONCLUSIONS RECOMMENDATIONS FOR FUTURE RESEARCH

7.1 Conclusions

The high operational costs of the semiconductor industry drive a need for a continuous effort to improve efficiency and reduce costs. Simulation has been shown as a good tool for use in analysis for the industry. Two experiments are presented in this thesis using simulation.

Modeling capabilities and methodologies developed during these experiments give contributions to industry in themselves. The first experiment is a capacity analysis aimed at giving industry the ability to effectively evaluate AMHS capacity. The step by step method followed for this experiment can be followed by any fab to determine the required AMHS capacity. A summary of these steps can be found in Table 5.1. The second experiment studies tool dispatching rules and vehicle dispatching rules and determines if there is an interaction between them. To perform this experiment various AMHS vehicle dispatching rules are required to be modeled. However, common simulation software used for the semiconductor industry, AutoSched AP, does not have the capability to choose different vehicle dispatching rules. The modeling of the rules requires not only custom functions within the model, but changes to code in an extension to the software itself. The code for the rules created here can be copied into other AutoSched models and used for future research.

Results obtained from the experiments provide additional contributions. The capacity analysis experiment shows total fab performance degradation, attributed to the AMHS, occurring well below the point at which the AMHS is fully utilized. This experiment shows the importance of using a performance indicator that takes the entire fab performance into account, instead of just the performance of the AMHS, when performing AMHS research. If the AMHS

is optimized independently, negative side affects may appear elsewhere in the fab that are not taken into account.

The dispatching rules experiment results show that for both the ASIC and Make-to-Order fabs tool dispatching rules, AMHS vehicle dispatching rules and their interactions are statistically significant for all the performance indicators tested. The significance of the interaction displays that an extra benefit may be obtained by choosing good combinations of rules, and conversely worse results that would be expected may be obtained by using rules that do not work well together. Tool and AMHS vehicle dispatching rules should be studied together to take advantage of the extra benefit.

The conclusions and contributions from this research provide a fab with greater insight on how to improve fab operations. Methods, modeling code, guidelines, and lessons learned from the experiments presented can be applied to industry to efficiently utilize a fab. This in turn can reduce costs and make manufacturers become more competitive.

7.2 Recommendations for Future Research

Fully automated material handling systems are new to the semiconductor industry, and there is vast amount of new research that is needed.

The capacity analysis presented in Chapter 5 is performed on only one fab data set. If experiments were done on multiple fab sizes and types, a range for the optimal vehicle utilization may be found to be used as a guideline in industry. With this guide, a warning could be flagged when the utilization of a vehicle exceeds a certain number. Product mixes and volumes are constantly changing in fabs, and this warning could prevent capacity loss in such a fast paced environment. The work presented here shows the importance of using performance indicators

related to the entire fab productivity when studying capacity. Future research could take this a step further to determine the importance of specific performance indicators in relation to others.

The dispatching rules experiment presented in Chapter 6 studies a combination of 5 machine and 7 vehicle dispatching rules. This only represents a small portion of the dispatching rules that exist. The rules are chosen for this experiment based on popular rules from past research. Future research could look into different rules, particularly rules from other industries that have not been tested in the semiconductor industry. Much research has gone into the behavior of automated guided vehicles. This research could be of great help, but much of it is not being utilized in the semiconductor industry. Also, new rules could be created and tested.

Vehicle dispatching rules determine which lot the vehicle should pick up. This is only one aspect of a vehicle's behavior. A vehicle must also decide the path it must take to pick up the lot and where it will park when it is idle. For the experiments run for this thesis, the vehicle always travels the shortest path and parks in the spot where it becomes idle. Although simplified here, these decisions are not as trivial as they may seem. A vehicle may want to travel a path with a longer distance if it can avoid traffic and delays. Also, it may want to keep a path clear that will be needed in the immediate future for a high priority lot. Instructing vehicles to park at the location where they become idle can cause a delay if the vehicle is blocking the path of another vehicle. There may also be a gain in productivity by sending the vehicle to the location where it will most likely be needed next.

Finally, the dispatching rules experiment finds an interaction present between two rules used in a fab. Other interactions may be present. An interaction between machine dispatching rules and rework strategies was found by Greg Laubisch (2003). The areas where these

interactions exist should be identified so that engineers will know which rules may be affected when one is changed elsewhere in the fab.

References

Blackstone, J. H., Phillips, D. T., & Hogg, G. L. (1982, January-February). A state-of-the-art survey of dispatching rules for manufacturing job shop operations. *International Journal of Production Research, 20(1)*, 27-45.

Chrisos, J., & Nestel-Patt, J. (1998, July). Integration risks in 300-mm wafer fab automation. *Solid State Technology, 47(7),* 193-201. Retrieved July 19, 2004, from http://vnweb.hwwilsonweb.com.

Egbelu, P J., & Tanchoco, J. M. A. (1984). Characterization of automatic guided vehicle dispatching rules. *International Journal of Production Research, 22(3),* 359-374.

Glüer, D. (2002). Continuous improvement techniques for automated material handling systems. *12th International Conference on Flexible Automation and Intelligent Manufacturing, July 15-17* (pp. 1214-1223).

Hasenbein, J., Sigireddy, S., & Wright, R. (2004, August). Taking a queue from simulation: The many rules of scheduling semiconductor material handling. *IIE Transactions, 36(8)*, 39-43. Retrieved August 5, 2004, from www.iienet.org.

Hsieh, B., Chang, S., & Chen, C. (2001). Dynamic scheduling rule selection for semiconductor wafer fabrication. *Proceedings of the 2001 IEEE International Conference on Robotics & Automation, May 21-26* (pp. 535-540). Retrieved July 2, 2004, from http://ieeexplore.ieee.org.

Jefferson, T., Rangaswami, M., & Stoner, G. (1996). Simulation in the design of ground-based intrabay automation systems. *Winter Simulation Conference Precedings, December 8-11* (pp. 1008-1013). Retrieved March 30, 2004, from http://ieeexplore.ieee.org.

Kuhl, M., & Christopher, J. (2004). Capacity analysis of automated material handling systems in semiconductor fabs. *Winter Simulation Conference Precedings, December 5-8* (pp. 1962 – 1966).

Kurosaki, R., Nagao, N., Komada, H., Watanable, Y., & Yano, H. (1997). AMHS for 300 mm wafer. *IEEE International Symposium on Semiconductor Manufacturing Conference Precedings, October 6-8* (pp. D13-D16). Retrieved March 30, 2004, from http://ieeexplore.ieee.org.

Laubisch, G. (2003). *A simulation study of dispatching rules and rework strategies in semiconductor manufacturing.* Masters thesis, Rochester Institute of Technology, Rochester, New York.

Lee, J., & Maneesavet, R. (1999, January). Dispatching rail-guided vehicles and scheduling jobs in a flexible manufacturing system. *International Journal of Production Research, 37(1),* 111-123. Retrieved July 2, 2004, from http://www.engineeringvillage2.org.

Li, S., Tang, T., & Collins, D. W. (1996). Minimum inventory variability schedule with applications in semiconductor fabrication. *IEEE Transactions on Semiconductor Manufacturing, 9(1),* 145-149.

Lin, J. T., Wang, F., & Yen, P. (2001, April 15). Simulation analysis of dispatching rules for an automated interbay material handling system in wafer fab. *International Journal of Production Research, 39(6),* 1221-1238. Retrieved July 2, 2004, from http://www.engineeringvillage2.org.

Lu, S. C. H., Ramaswamy, D., & Kumar, P. R. (1994). Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants. *IEEE Transactions on Semiconductor Manufacturing, 7(3),* 374-388.

Mackulak, G.T., & Savory, P. (2001, August). A simulation-based experiment for comparing AMHS performance in a semiconductor fabrication facility. *IEEE Transactions on Semiconductor Manufacturing,* 14(3), 273-280. Retrieved March 30, 2004 from www.engineeringvillage2.org.

Mittler, M., & Schoemig, A. K., (1999). Comparison of dispatching rules for semiconductor manufacturing using large facility models. *Winter Simulation Conference Proceedings, December 5-8* (pp. 709-713). Retrieved July 2, 2004, from http://ieeexplore.ieee.org.

Mori, K., Ono, A. & Iwasaki, J. (2001). Dynamical control method of AMHS for multi-production lines. *IEEE International Sympsium on Simiconductor Manufacturing Conference Precedings, October 8-10* (pp. 83-86). Retrieved March 30, 2004, from http://ieeexplore.ieee.org.

Nadoli, G., & Pillai, D. (1994). Simulation in automated material handling systems design for semiconductor manufacturing. *Winter Simulation Conference Precedings, December 11-14* (pp. 892-899). Retrieved March 30, 2004, from http://ieeexplore.ieee.org.

Nazzal, D., & Bodner, D.A. (2003). A simulation-based design framework for automated material handling systems in 300mm fabrication facilities. *Winter Simulation Conference Precedings, December 7-10* (pp. 1351-1359). Retrieved March 30, 2004, from http://ieeexplore.ieee.org.

O'Neil (1991). Performance evaluation of lot dispatching and scheduling algorithms through discrete event simulation. *3rd Annual International Semiconductor Manufactoring Science Symposium held in conjunction with SEMICON/West. May 20-22* (pp. 21-24). Retrieved July 2, 2004, from http://engineeringvillage2.org.

Papronty, I., Juin-Yan Shiau, Yo Huh, & Mackulat, G.T. (2000). Simulation based comparison of semiconductor AMHS alternatives: continuous flow vs. overhead monorail. Winter *Simulation Conference Proceedings, December 10-13* (pp.1333-1338). Retrieved April 15, 2004, from http://ieeexplore.ieee.org.

Pillai, D., Quinn, T., Kryder, K., & Charlson, D. (1999). Integration of 300 mm fab layouts and material handling automation. *IEEE International Sympsium on Simiconductor Manufacturing Conference Precedings, October 11-13* (pp. 23-26). Retrieved March 30, 2004, from http://ieeexplore.ieee.org.

Raju, K .R., & Chetty, O. V. K. (1993, May). Design and evaluation of automated guided vehicle systems for flexible manufacturing systems: an extended timed Petri net-based approach. *International Journal of Production Research, 31(5),* 1069-1096. Retrieved July 7, 2004, from http://weblinks1.epnet.com.

Sabuncuoglu, I., & Hommertzheim, D. L. (1992, July). Experimental investigation of FMS machine and AGV scheduling rules against the mean flow-time criterion. *International Journal of Production Research, 30(7),* 1617-1635. Retrieved July 7, 2004, from http://weblinks1.epnet.com.

Sawik, T. (1995). Dispatching scheduling of machines and vehicles in a flexible manufacturing system. *IEEE Symposium on Emerging Technologies & Factory Automation, October 10-13* (pp. 3-13). Retrieved July 2, 2004, from http://www.engineeringvillage2.org.

Sigireddy, S., Hasenbein, J., & McCormack, D. W. (2003, April 30). Implementation of automated material handling system (AMHS) vehicle scheduling rules. *International SEMATECH Technology Transfer #03044392A-ENG.*

Sturm, R., Seidelmann, J., Dorner, J. & Reddig, K. (2003, February). An approach to robust layout planning of AMHS. *Winter Simulation Conference Precedings, December 7-10* (pp. 1366-1372). Retrieved March 30, 2004, from http://ieeexplore.ieee.org.

Wolf, S., & Tauber, R.N. (2000). *Silicon Processing for the VLSI Era. Volume 1 – Process Technology.* Sunset Beach, CA: Lattice Press.

Yamashita, H. (2001, May). Analysis of dispatching rules of AGV systems with multiple vehicles. *IIE Transactions, 33(10),* 889-895. Retrieved July 2, 2004 from http://www.engineeringvillage2.org

Appendices

These appendices contain raw data, analysis results, and additional information on the models and experiments used in this thesis. Appendix A contains a sample route of the model of data set 2. Appendix B provides raw data from the capacity analysis experiment. Appendix C documents and explains the programming required to implement the vehicle dispatching rules in AutoMod. Appendices D-L provide additional information on the dispatching rules experiment. Appendices M-N are CDs that contain all of the simulation files that are used for the experiments in this thesis.

Appendix A. Sample Part Route of ASIC Fab

The full model of the ASIC fab contains 25 files each consisting of multiple pages. The full model would consume too much space to be included in print. Instead, Table A.1 shows the routing for the Bipolar Arrays SRAM #1 part as an example of a part routing. Table A.1 does not include all of the columns that exist in the route.txt file of the model. Although not included in print, the routes for the make-to-order fab follow a similar format. The full models, along with the original data sets used to create the models, are included in the CD attached to the thesis.

Table A.1: Bipolar Arrays SRAM #1 Part Routing from route.txt File of ASIC Fab

STEP	STNAM	PTIME	PTUNITS	PTYPE	DESC	REWORK	RWKTYPE
Route1_Step1	Lift Off: Batch_stn2	25	min	lots/batch	Lift Off23	0	piece
Route1_Step2	Clean: Multi-Sequence_stn3	62	min	lots/batch	Clean24	0	piece
Route1_Step3	Ash: Single Wafer_stn1	1.42	min	piece	Ash 1	0	piece
Route1_Step4	Clean: Multi-Sequence_stn1	31	min	lots/batch	Clean 2	0	piece
Route1_Step5	Apply Resist: Conveyor_stn1	4.83	min	piece	Apply Resist 4	0	piece
Route1_Step6	Bake: Batch_stn2	38	min	lots/batch	Bake27	0	piece
Route1_Step7	Apply Resist: Conveyor_stn4	5.62	min	piece	Apply Resist 7	0	piece
Route1_Step8	Apply Resist: Conveyor_stn2	2.48	min	piece	Apply Resist 5	0	piece
Route1_Step9	Expose UV: Single Wafer_stn1	1.18	min	piece	Expose UV 8	0	piece
Route1_Step10	Develop: Multi-Sequence_stn1	20	min	lots/batch	Develop11	0	piece
Route1_Step11	Inspect: Inspect_stn1	29	min	lot	Inspect13	0.05	piece
Route1_Step12	Silylation: Multi-Sequenc_stn1	21	min	lots/batch	Silylation14	0	piece
Route1_Step13	Bake: Batch_stn3	33	min	lots/batch	Bake39	0	piece
Route1_Step14	Lift Off: Batch_stn1	114	min	lots/batch	Lift Off15	0	piece
Route1_Step15	Inspect: Inspect_stn3	37	min	lot	Inspect60	0.05	piece

Route1_Step16	Clean: Multi-Sequence_stn5	16	min	lots/batch	Clean51	0	piece
Route1_Step17	Evaporation: Batch_stn1	217	min	lots/batch	Evaporation18	0	piece
Route1_Step18	Measure: Batch_stn3	6	min	lots/batch	Measure71	0	piece
Route1_Step19	Measure: Batch_stn1	3	min	lots/batch	Measure58	0	piece
Route1_Step20	Measure: Batch_stn4	8	min	lots/batch	Measure72	0	piece
Route1_Step21	Lift Off: Batch_stn3	200	min	lots/batch	Lift Off79	0	piece
Route1_Step22	Clean: Multi-Sequence_stn6	35	min	lots/batch	Clean80	0	piece
Route1_Step23	Inspect: Single Wafer_stn1	0.492	min	piece	Inspect42	0	piece
Route1_Step24	Ash: Single Wafer_stn4	6.14	min	piece	Ash78	0	piece
Route1_Step25	Measure: Inspect_stn2	26	min	lot	Measure69	0	piece
Route1_Step26	Clean: Single Wafer_stn1	3.7	min	piece	Clean82	0	piece
Route1_Step27	Quench: Multi-Sequence_stn1	16	min	lots/batch	Quench94	0	piece
Route1_Step28	Bake: Batch_stn1	33	min	lots/batch	Bake26	0	piece
Route1_Step29	Measure: Inspect_stn2	26	min	lot	Measure69	0	piece
Route1_Step30	Preclean: Multi-Sequence_stn1	26	min	lots/batch	Preclean29	0	piece
Route1_Step31	Sputter: Batch_stn2	326	min	lots/batch	Sputter30	0	piece
Route1_Step32	Measure: Batch_stn5	9	min	lots/batch	Measure83	0	piece
Route1_Step33	Measure: Batch_stn2	23	min	lots/batch	Measure66	0	piece
Route1_Step34	Inspect: Single Wafer_stn2	0.1	min	piece	Inspect68	0	piece
Route1_Step35	Measure: Single Wafer_stn1	0.1	min	piece	Measure67	0	piece
Route1_Step36	Chem/Mech Polish: Batch_stn1	42.5	min	lots/batch	Chem/Mech Polis33	0	piece
Route1_Step37	Clean: Conveyor_stn1	1.456	min	piece	Clean52	0	piece
Route1_Step38	Inspect: Single Wafer_stn2	0.4	min	piece	Inspect68	0	piece
Route1_Step39	Measure: Single Wafer_stn1	0.5	min	piece	Measure67	0	piece
Route1_Step40	Clean: Conveyor_stn1	1.456	min	piece	Clean52	0	piece
Route1_Step41	Bake: Batch_stn8	63	min	lots/batch	Bake53	0	piece
Route1_Step42	Clean: Multi-Sequence_stn4	26	min	lots/batch	Clean34	0	piece
Route1_Step43	Nitride Deposition: Batch_stn1	42	min	lots/batch	Nitride Deposition35	0	piece
Route1_Step44	Measure: Inspect_stn3	5	min	lot	Measure70	0	piece
Route1_Step45	Clean: Multi-Sequence_stn1	31	min	lots/batch	Clean 2	0	piece
Route1_Step46	Apply Resist: Conveyor_stn1	4.89	min	piece	Apply Resist 4	0	piece
Route1_Step47	Apply Resist: Conveyor_stn2	2.48	min	piece	Apply Resist 5	0	piece
Route1_Step48	Expose UV: Single Wafer_stn1	1.18	min	piece	Expose UV 8	0	piece
Route1_Step49	Develop: Multi-Sequence_stn1	20	min	lots/batch	Develop11	0	piece
Route1_Step50	Inspect: Inspect_stn1	29	min	lot	Inspect13	0.018	piece
Route1_Step51	Silylation: Multi-Sequenc_stn1	21	min	lots/batch	Silylation14	0	piece
Route1_Step52	Bake: Single Wafer_stn1	2.44	min	piece	Bake75	0	piece
Route1_Step53	Reactive Ion Etch: Single_stn3	2.3	min	piece	Reactive Ion Etch44	0	piece
Route1_Step54	Inspect: Inspect_stn3	52	min	lot	Inspect60	0.018	piece
Route1_Step55	Evaporation: Batch_stn2	93	min	lots/batch	Evaporation19	0	piece
Route1_Step56	Measure: Batch_stn3	6	min	lots/batch	Measure71	0	piece
Route1_Step57	Measure: Batch_stn1	7	min	lots/batch	Measure58	0	piece
Route1_Step58	Lift Off: Batch_stn2	145	min	lots/batch	Lift Off23	0	piece
Route1_Step59	Clean: Multi-Sequence_stn3	62	min	lots/batch	Clean24	0	piece
Route1_Step60	Inspect: Single Wafer_stn1	0.492	min	piece	Inspect42	0	piece
Route1_Step61	Ash: Single Wafer_stn2	1.34	min	piece	Ash25	0	piece
Route1_Step62	Bake: Batch_stn1	33	min	lots/batch	Bake26	0	piece
Route1_Step63	Measure: Inspect_stn2	34	min	lot	Measure69	0	piece
Route1_Step64	Sinter: Batch_stn1	59	min	lots/batch	Sinter37	0	piece
Route1_Step65	In Line Test: Single_stn1	10.1	min	piece	In Line Test28	0	piece
Route1_Step66	Clean: Multi-Sequence_stn1	31	min	lots/batch	Clean 2	0	piece
Route1_Step67	Apply Resist: Conveyor_stn1	4.89	min	piece	Apply Resist 4	0	piece
Route1_Step68	Apply Resist: Conveyor_stn2	2.48	min	piece	Apply Resist 5	0	piece
Route1_Step69	Expose UV: Single Wafer_stn1	1.18	min	piece	Expose UV 8	0	piece
Route1_Step70	Develop: Multi-Sequence_stn1	20	min	lots/batch	Develop11	0	piece
Route1_Step71	Inspect: Inspect_stn1	29	min	lot	Inspect13	0.006	piece
Route1_Step72	Silylation: Multi-Sequenc_stn1	21	min	lots/batch	Silylation14	0	piece
Route1_Step73	Bake: Batch_stn3	33	min	lots/batch	Bake39	0	piece
Route1_Step74	Reactive Ion Etch: Single_stn3	2.3	min	piece	Reactive Ion Etch44	0	piece
Route1_Step75	Inspect: Inspect_stn3	37	min	lot	Inspect60	0.006	piece
Route1_Step76	Clean: Multi-Sequence_stn2	16	min	lots/batch	Clean17	0	piece

Route1_Step77	Inspect: Single Wafer_stn3	0.2	min	piece	Inspect76	0	piece
Route1_Step78	Evaporation: Batch_stn2	110	min	lots/batch	Evaporation19	0	piece
Route1_Step79	Measure: Batch_stn3	6	min	lots/batch	Measure71	0	piece
Route1_Step80	Measure: Batch_stn1	3	min	lots/batch	Measure58	0	piece
Route1_Step81	Lift Off: Batch_stn2	145	min	lots/batch	Lift Off23	0	piece
Route1_Step82	Clean: Multi-Sequence_stn3	62	min	lots/batch	Clean24	0	piece
Route1_Step83	Inspect: Single Wafer_stn1	0.492	min	piece	Inspect42	0	piece
Route1_Step84	Inspect: Inspect_stn6	36	min	lot	Inspect85	0	piece
Route1_Step85	Ash: Single Wafer_stn2	1.34	min	piece	Ash25	0	piece
Route1_Step86	Bake: Batch_stn1	33	min	lots/batch	Bake26	0	piece
Route1_Step87	Measure: Inspect_stn2	34	min	lot	Measure69	0	piece
Route1_Step88	Preclean: Multi-Sequence_stn1	26	min	lots/batch	Preclean29	0	piece
Route1_Step89	Sputter: Batch_stn2	438	min	lots/batch	Sputter30	0	piece
Route1_Step90	Measure: Batch_stn5	9	min	lots/batch	Measure83	0	piece
Route1_Step91	Measure: Batch_stn2	23	min	lots/batch	Measure66	0	piece
Route1_Step92	Inspect: Single Wafer_stn2	0.1	min	piece	Inspect68	0	piece
Route1_Step93	Chem/Mech Polish: Batch_stn1	42.5	min	lots/batch	Chem/Mech Polis33	0	piece
Route1_Step94	Clean: Conveyor_stn1	1.456	min	piece	Clean52	0	piece
Route1_Step95	Inspect: Single Wafer_stn2	0.4	min	piece	Inspect68	0	piece
Route1_Step96	Clean: Conveyor_stn1	1.456	min	piece	Clean52	0	piece
Route1_Step97	Bake: Batch_stn8	63	min	lots/batch	Bake53	0	piece
Route1_Step98	Clean: Multi-Sequence_stn4	26	min	lots/batch	Clean34	0	piece
Route1_Step99	Nitride Deposition: Batch_stn1	75	min	lots/batch	Nitride Deposition35	0	piece
Route1_Step100	Measure: Inspect_stn3	5	min	lot	Measure70	0	piece
Route1_Step101	Clean: Multi-Sequence_stn1	27	min	lots/batch	Clean 2	0	piece
Route1_Step102	Ash: Single Wafer_stn1	1.42	min	piece	Ash 1	0	piece
Route1_Step103	Apply Resist: Batch_stn1	33	min	lots/batch	Apply Resist 3	0	piece
Route1_Step104	Apply Resist: Conveyor_stn2	2.48	min	piece	Apply Resist 5	0	piece
Route1_Step105	Expose UV: Single Wafer_stn2	1.18	min	piece	Expose UV 9	0	piece
Route1_Step106	Develop: Multi-Sequence_stn1	20	min	lots/batch	Develop11	0	piece
Route1_Step107	Inspect: Inspect_stn1	47	min	lot	Inspect13	0.021	piece
Route1_Step108	Bake: Batch_stn7	33	min	lots/batch	Bake50	0	piece
Route1_Step109	Reactive Ion Etch: Single_stn2	1.1	min	piece	Reactive Ion Etch22	0	piece
Route1_Step110	Inspect: Inspect_stn5	23	min	lot	Inspect64	0	piece
Route1_Step111	Ash: Single Wafer_stn2	2.84	min	piece	Ash25	0	piece
Route1_Step112	Clean: Multi-Sequence_stn3	62	min	lots/batch	Clean24	0	piece
Route1_Step113	Ash: Single Wafer_stn2	2.84	min	piece	Ash25	0	piece
Route1_Step114	Measure: Inspect_stn2	16	min	lot	Measure69	0	piece
Route1_Step115	Clean: Multi-Sequence_stn1	31	min	lots/batch	Clean 2	0	piece
Route1_Step116	Apply Resist: Conveyor_stn1	4.89	min	piece	Apply Resist 4	0	piece
Route1_Step117	Apply Resist: Conveyor_stn2	2.48	min	piece	Apply Resist 5	0	piece
Route1_Step118	Expose UV: Single Wafer_stn1	1.18	min	piece	Expose UV 8	0	piece
Route1_Step119	Develop: Multi-Sequence_stn1	20	min	lots/batch	Develop11	0	piece
Route1_Step120	Inspect: Inspect_stn1	29	min	lot	Inspect13	0.022	piece
Route1_Step121	Silylation: Multi-Sequenc_stn1	21	min	lots/batch	Silylation14	0	piece
Route1_Step122	Bake: Batch_stn3	33	min	lots/batch	Bake39	0	piece
Route1_Step123	Reactive Ion Etch: Single_stn3	2.8	min	piece	Reactive Ion Etch44	0	piece
Route1_Step124	Inspect: Inspect_stn3	52	min	lot	Inspect60	0.022	piece
Route1_Step125	Clean: Multi-Sequence_stn2	16	min	lots/batch	Clean17	0	piece
Route1_Step126	Inspect: Batch_stn2	11	min	lots/batch	Inspect73	0	piece
Route1_Step127	Inspect: Single Wafer_stn3	0.2	min	piece	Inspect76	0	piece
Route1_Step128	Evaporation: Batch_stn2	97	min	lots/batch	Evaporation19	0	piece
Route1_Step129	Measure: Batch_stn3	6	min	lots/batch	Measure71	0	piece
Route1_Step130	Measure: Batch_stn1	7	min	lots/batch	Measure58	0	piece
Route1_Step131	Lift Off: Batch_stn2	145	min	lots/batch	Lift Off23	0	piece
Route1_Step132	Clean: Multi-Sequence_stn3	62	min	lots/batch	Clean24	0	piece
Route1_Step133	Inspect: Single Wafer_stn1	0.492	min	piece	Inspect42	0	piece
Route1_Step134	Ash: Single Wafer_stn2	1.34	min	piece	Ash25	0	piece
Route1_Step135	Bake: Batch_stn1	33	min	lots/batch	Bake26	0	piece
Route1_Step136	Measure: Inspect_stn2	34	min	lot	Measure69	0	piece
Route1_Step137	Sinter: Batch_stn1	59	min	lots/batch	Sinter37	0	piece

Route1_Step138	In Line Test: Single_stn1	12.9	min	piece	In Line Test28	0	piece
Route1_Step139	Preclean: Multi-Sequence_stn1	26	min	lots/batch	Preclean29	0	piece
Route1_Step140	Sputter: Batch_stn3	346	min	lots/batch	Sputter31	0	piece
Route1_Step141	Measure: Batch_stn5	19	min	lots/batch	Measure83	0	piece
Route1_Step142	Measure: Batch_stn2	23	min	lots/batch	Measure66	0	piece
Route1_Step143	Inspect: Batch_stn1	10	min	lots/batch	Inspect65	0	piece
Route1_Step144	Clean: Multi-Sequence_stn4	26	min	lots/batch	Clean34	0	piece
Route1_Step145	Nitride Deposition: Batch_stn1	75	min	lots/batch	Nitride Deposition35	0	piece
Route1_Step146	Measure: Inspect_stn3	5	min	lot	Measure70	0	piece
Route1_Step147	Clean: Multi-Sequence_stn1	27	min	lots/batch	Clean 2	0	piece
Route1_Step148	Bake: Batch_stn5	33	min	lots/batch	Bake46	0	piece
Route1_Step149	Apply Resist: Batch_stn1	33	min	lots/batch	Apply Resist 3	0	piece
Route1_Step150	Apply Resist: Conveyor_stn3	2.48	min	piece	Apply Resist 6	0	piece
Route1_Step151	Expose UV: Single Wafer_stn2	1.18	min	piece	Expose UV 9	0	piece
Route1_Step152	Develop: Multi-Sequence_stn2	17	min	lots/batch	Develop12	0	piece
Route1_Step153	Inspect: Inspect_stn1	47	min	lot	Inspect13	0.021	piece
Route1_Step154	Bake: Batch_stn7	38	min	lots/batch	Bake50	0	piece
Route1_Step155	Reactive Ion Etch: Single_stn1	6.6	min	piece	Reactive Ion Etch16	0	piece
Route1_Step156	Inspect: Inspect_stn5	27	min	lot	Inspect64	0	piece
Route1_Step157	Ash: Single Wafer_stn2	2.84	min	piece	Ash25	0	piece
Route1_Step158	Bake: Batch_stn1	33	min	lots/batch	Bake26	0	piece
Route1_Step159	Clean: Multi-Sequence_stn3	62	min	lots/batch	Clean24	0	piece
Route1_Step160	Ash: Single Wafer_stn2	1.34	min	piece	Ash25	0	piece
Route1_Step161	Measure: Inspect_stn2	28	min	lot	Measure69	0	piece
Route1_Step162	Clean: Multi-Sequence_stn2	16	min	lots/batch	Clean17	0	piece
Route1_Step163	Evaporation: Batch_stn2	93	min	lots/batch	Evaporation19	0	piece
Route1_Step164	Measure: Batch_stn3	6	min	lots/batch	Measure71	0	piece
Route1_Step165	Measure: Batch_stn1	3	min	lots/batch	Measure58	0	piece
Route1_Step166	Clean: Multi-Sequence_stn1	27	min	lots/batch	Clean 2	0	piece
Route1_Step167	Bake: Batch_stn5	33	min	lots/batch	Bake46	0	piece
Route1_Step168	Apply Resist: Conveyor_stn3	2.48	min	piece	Apply Resist 6	0	piece
Route1_Step169	Expose UV: Single Wafer_stn2	1.18	min	piece	Expose UV 9	0	piece
Route1_Step170	Develop: Multi-Sequence_stn2	17	min	lots/batch	Develop12	0	piece
Route1_Step171	Inspect: Inspect_stn1	47	min	lot	Inspect13	0.011	piece
Route1_Step172	Bake: Batch_stn4	33	min	lots/batch	Bake43	0	piece
Route1_Step173	Ash: Single Wafer_stn3	1.42	min	piece	Ash49	0	piece
Route1_Step174	Etch: Multi-Sequence_stn1	21	min	lots/batch	Etch36	0	piece
Route1_Step175	Inspect: Inspect_stn4	12	min	lot	Inspect63	0	piece
Route1_Step176	Clean: Multi-Sequence_stn3	62	min	lots/batch	Clean24	0	piece
Route1_Step177	Measure: Inspect_stn2	28	min	lot	Measure69	0	piece
Route1_Step178	Preclean: Multi-Sequence_stn1	26	min	lots/batch	Preclean29	0	piece
Route1_Step179	Sputter: Batch_stn3	452	min	lots/batch	Sputter31	0	piece
Route1_Step180	Measure: Batch_stn5	19	min	lots/batch	Measure83	0	piece
Route1_Step181	Measure: Batch_stn2	23	min	lots/batch	Measure66	0	piece
Route1_Step182	Inspect: Batch_stn1	10	min	lots/batch	Inspect65	0	piece
Route1_Step183	Sinter: Batch_stn1	74	min	lots/batch	Sinter37	0	piece
Route1_Step184	Clean: Multi-Sequence_stn1	27	min	lots/batch	Clean 2	0	piece
Route1_Step185	Bake: Batch_stn5	33	min	lots/batch	Bake46	0	piece
Route1_Step186	Apply Resist: Batch_stn1	33	min	lots/batch	Apply Resist 3	0	piece
Route1_Step187	Apply Resist: Conveyor_stn3	2.48	min	piece	Apply Resist 6	0	piece
Route1_Step188	Expose UV: Single Wafer_stn2	1.18	min	piece	Expose UV 9	0	piece
Route1_Step189	Develop: Multi-Sequence_stn2	17	min	lots/batch	Develop12	0	piece
Route1_Step190	Inspect: Inspect_stn1	47	min	lot	Inspect13	0.011	piece
Route1_Step191	Bake: Batch_stn6	33	min	lots/batch	Bake47	0	piece
Route1_Step192	Ash: Single Wafer_stn3	1.42	min	piece	Ash49	0	piece
Route1_Step193	Etch: Multi-Sequence_stn1	26	min	lots/batch	Etch36	0	piece
Route1_Step194	Inspect: Inspect_stn4	12	min	lot	Inspect63	0	piece
Route1_Step195	Clean: Multi-Sequence_stn3	62	min	lots/batch	Clean24	0	piece
Route1_Step196	Measure: Inspect_stn2	28	min	lot	Measure69	0	piece
Route1_Step197	Bake: Batch_stn1	33	min	lots/batch	Bake26	0	piece

Appendix B. Capacity Analysis Raw Data

This Appendix contains the raw data collected in the capacity analysis experiment for the ASIC fab. The maximum vehicle utilization, shown in Table B.1, is taken as an average over those five replications for each release rate. Table B.2 contains the average cycle time, average WIP, and average X theoretical value, which are recorded for five replications at each release rate. All replications marked with "*" after the machine rule do not reach steady state. In some cases where steady state is not reached, the simulation is not able to run through all three replications and no data is recorded.

Table B.1: Maximum Vehicle Utilizations Recorded for Capacity Analysis of ASIC Fab

Release Rate (wafers/month)	Maximum Vehicle Utilization
5000	40.8
6000	50.8
7000	59
8000	67.3
9000	75.5
10000	83.9
11000	*
12000	*
13000	*
14000	*
15000	*

Table B.2: Raw Data Recorded for Capacity Analysis of ASIC Fab

Release Rate (wafers/month)	Average Cycle Time (days)	Average WIP (lots)	Average X Theoretical Value
5000	9.43	70.87	2.91
5000	9.43	70.88	2.92
5000	9.44	70.95	2.92
5000	9.44	70.88	2.92
5000	9.44	70.92	2.92
6000	9.57	86.25	2.95
6000	9.56	86.19	2.96
6000	9.56	86.21	2.96
6000	9.57	86.24	2.96
6000	9.56	86.21	2.95
7000	9.71	102.1	3
7000	9.72	102.22	3
7000	9.71	102.16	3
7000	9.72	102.26	3
7000	9.71	102.14	3
8000	9.89	118.94	3.05
8000	9.90	118.94	3.06
8000	9.90	119.06	3.06
8000	9.89	118.92	3.05
8000	9.90	119	3.05
9000	10.14	137.12	3.13
9000	10.13	137.04	3.13
9000	10.15	137.28	3.13
9000	10.14	137.16	3.13
9000	10.16	137.34	3.13
10000	10.51	157.97	3.24
10000	10.50	157.9	3.24
10000	10.52	158.09	3.24
10000	10.52	158.12	3.24
10000	10.51	158.03	3.24
11000	*		
12000	*		
13000	*		
14000	*		
15000	*		

Appendix C. Programming Required for Vehicle Dispatching Rules

The vehicle dispatching rules are programmed in AutoMod. The work center initiated vehicle dispatching rule does not need to be changed since it is set to NIV by default. The vehicle initiated dispatching rules are programmed manually except for FIFO, which is the default. These rules are programmed by altering the programming of the source files in the AutoMod model, and in some cases adding load attributes and variables. The EDD, CR, CRQ, and MUS vehicle dispatching rules all require lot attributes that are stored in AutoSched and are not available in AutoMod. To transfer this information from AutoSched to AutoMod changes must be made to the amap extension written in C++. The sections below display the code that is written for each dispatching rule along with an explanation.

FIFO Vehicle Dispatching Rule

FIFO is the default vehicle dispatching rule used by AutoMod for the amap extension. Although no extra programming needs to be performed to implement this rule, it is important to describe this code because this is the code that is altered to program the other rules. FIFO ranks the lots by the amount of time they have spent waiting in the queue. This attribute is stored in AutoMod and does not require additional attributes to be sent from AutoSched. However, some attributes, like the lot name and the destination storage, are sent from AutoSched to AutoMod by default to be used for other purposes. The amap extension code that sends the attributes to AutoMod is found in the *travelToStorage* function of the amap file in the extension. The code is not associated with specific models, but with the extension software itself. This file can be found at C:\AutoMod\asiext\amap which may vary depending on installation settings. A section of code from this function which sends the lot attributes is shown below. The lines of code shown here are not all consecutive.

```
FIString fromLocation;
FIString toLocation;

        if (aProj->runWithAutoMod() && theSE->prevSystem()) {
                fromLocation = theSE->prevSystem()->name() + ":" +
theSE->curStorage()->name();
                toLocation = theSE->prevSystem()->name() + ":" +
theSE->nextStorage()->name();
        }

MMSyncMsg Message;
Message.SetType(MOVE_REQUEST);
Message.SetLong(1, 2);                                  //Lot ptr
Message.SetString(2, theSE->name());                    //Lot name string
Message.SetLong(2,(FIInt)theSE);                        //Lot ptr
Message.SetString(3,fromLocation);                      //From name string
Message.SetString(4,toLocation);                        //To name string
Message.SetLong(5, 0);                                  //Message Number
Message.SetString(5,theSE->destStorage()->name());
SyncSendMessage(Message);
```

In the amap C++ code, theSE is an object that represents the lot that is being sent to AutoMod for transport. The first section of the code assigns values to the fromLocation and toLocation variables. When arrows are shown after the theSE object, the values of the attributes that appear after the arrow are used. For example, the variable fromLocation is set to the name of the system of the lots previous step followed by a colon followed by the name of the storage of the lots current location. Next, the object Message is filled with all of the parameters that need to be sent to AutoMod. Some of the parameters are filled by referring to the object and attribute values as in the previous example, some are filled through variables, such as fromLocation, and others with constants, for example "2" Lastly, the *SyncSendMessage* fuction is called to send the Message object to AutoMod.

The AutoMod code receiving these attributes is found in the *model mmsyncmessage* function of the *messages.m* source file. The section of this function that deals with lot move requests and attributes along with AutoMod commenting is listed below.

```
/*      This function receives and handles all messages passed from AP.

        DO NOT MODIFY THIS FUNCTION. */
begin model mmsyncmessage function

        set VAmapTempInt = MMSyncReadMessageType()

        if (VAmapTempInt = VAmapMOVE_REQUEST) then begin
                /*      Move message - moves a lot from one storage to another.
                        String Parameters
                        1   None
                        2 - Lot Name
                        3 - From Location <system:location>
                        4 - To Location <system:location>
                        5 - Final Destination Storage

                        Integer Parameters
                        1   Lot Type (1=FOUP, 2=LOT, 3=RETICLE)
                        2 - Lot Pointer */

                call FAmapMove(MMSyncReadMessageInteger(1),
MMSyncReadMessageString(2), MMSyncReadMessageInteger(2),
                        MMSyncReadMessageString(3),  MMSyncReadMessageString(4),
MMSyncReadMessageInteger(5) ,
                                        MMSyncReadMessageString(5))
        End
```

This code receives a message from AutoSched. It is comprised of a set of if statements to determine the type of message that is being sent. This is only the code for the move request messages. This function reads in all of the parameters sent from AutoSched and calls the *FAmapMove* function, also in the *messages.m* source file, to create a load that contains the values of these parameters as attribute values. The code for the *FAmapMove* function along with AutoMod commenting is shown below.

```
/*      FAmapMove - Creates and moves load as requested by ASAP.
        Returns a non-zero integer on success, 0 on failure.

        DO NOT MODIFY THIS FUNCTION.

        Parameters
        ----------
```

```
        Integer     theMaterial - The type of the lot
        String      theName - The name of the lot
        Integer     thePtr - Pointer to the lot (a unique identifier).
        String      FromLoc - System and storage the lot is coming from.
                                        Format is "<system>:<storage>"
        String      ToLoc - System and storage the lot is traveling to.
                                        Format is "<system>:<storage>"
        Integer     MsgNum - The message number.
        String      destination - The ultimate destination of the lot.
*/
begin FAmapMove function

        read VAmapMasterLoad AAmapSystemName, VAmapMasterLoad
        AAmapFromName from FromLoc with delimiter ":"
        read VAmapTempString, VAmapMasterLoad AAmapToName from ToLoc with
        delimiter ":"

        /* The lot can only move between storages in the same system. */
        if (VAmapTempString <> VAmapMasterLoad AAmapSystemName) then begin
                print "ERROR: from system-",VAmapMasterLoad AAmapSystemName,
        "different than to system-",VAmapTempString2,"killing load." to
        VAmapTempString
                print VAmapTempString to message
                call FAmapSendAbort(VAmapTempString)
                return false
        end

        set VAmapMasterLoad AAmapFromLocation to FromLoc
        set VAmapMasterLoad AAmapToLocation to ToLoc

        if theMaterial = 1 then set VAmapMasterLoad type = LAmapFoup
        else if theMaterial = 2 then set VAmapMasterLoad type = LAmapLot
        else if theMaterial = 3 then set VAmapMasterLoad type = LAmapReticle

        set VAmapMasterLoad AAmapLotName = theName
        set VAmapMasterLoad AAmapLotPtr = thePtr
        set VAmapMasterLoad AAmapMsgNum = MsgNum
        set VAmapMasterLoad AAmapDestinationStorage = destination

        clone VAmapMasterLoad to PAmapMakeMove
        return true
end
```

The parameters listed in the commenting show all of the lot attributes that are transferred from AutoSched to AutoMod. This code receives and processes the information in AutoMod.

There is also C++ code in the amap extension that sends the attributes from AutoSched. When AutoSched needs to move a lot from one tool or stocker to the next, it sends a message to AutoMod that calls this function. This function then creates a load or entity in AutoMod to be moved by the vehicles. Loads only exist in AutoMod while they are waiting to be moved or in the process of transport. Once a load has been transported, it is deleted from the model and transport times are sent back to AutoSched. To assist in creating a load, AutoMod uses a "Master Load" which is a load that is created at the beginning of the simulation with all of the attributes and properties desired for a load that needs to be created at the request of AutoSched. The master load, *VAmapMasterLoad*, is never processed or transported during the simulation.

The first few lines of the code check to make sure that the lot is only traveling within one system. In the model, the interbay AMHS is one system, and the AMHS for each bay is a separate system. If a lot needs to be moved from tools in one bay to another, the move would be accomplished through three different requests from AutoSched to AutoMod. First, a message is sent to AutoMod to move the lot from the first tool to the stocker in the same bay. When that is complete, AutoMod sends the transport time back to AutoSched. AutoSched then sends the message to move from the stocker of the first bay to the stocker of the second bay and again receives the travel time for that portion of the move. Last, AutoSched sends a message to move from the stocker to the tool in the second bay and receives the travel time for the last portion of the move. If the message tries to move a lot from one system to another in only one message to AutoMod, the code prints an error message and exits the function.

The next few lines of code are set statements. These lines take the lot attributes sent from AutoSched and insert their values into the master load attributes. The load that needs to be changed is stated as VAmapMasterLoad. Directly after VAmapMasterLoad is the attribute of

the master load that needs to be changed. These usually begin with AAmap. Finally, after the equals sign or "to" is the parameter that holds the value of the attribute sent from AutoSched.

The last step in the function makes a copy of the master load with the new attribute values. This copy will be the load that is transported and processed in the model.

The behavior of the vehicles is controlled in the *veh.m* source file. Specifically, the *FJobFinished* function is where the vehicle initiated dispatching rule can be set. This function is called automatically by a vehicle as soon as it has set down a load and is looking for new work. This occurrence, when a vehicle is changing from a busy to an idle state, is the only period when a vehicle dispatching rule would be needed. The only reason a vehicle would be idle to begin with is if there were no lots waiting in queue. In that case, the vehicle would be assigned to pick up the first lot that appears in the queue and a work center initiated vehicle dispatching rule may be required to choose between two or more idle vehicles. The code and AutoMod comments for the *FJobFinished* function are shown below.

```
/*      This function executes after the load has been delivered.
        If there are any loads waiting to be moved on the system
        the vehicle claims the first load on the waiting list.

        This function may be modified if different behavior is
        desired.

        Paramters
        VehiclePtr theVehicle - The vehicle which called this function.
*/
begin FJobFinished function
        if theVehicle AAmapSystemPtr loads waiting size > 0 then begin
                if theVehicle AAmapLot = null then begin
                        claim theVehicle AAmapSystemPtr loads waiting first for
theVehicle
                end
        end
        return true
end
```

This function first checks to see if there are any loads waiting in queue to be picked up by a vehicle. This is done through the if statement. If there are no loads in queue the vehicle parks at its current location. If there are lots waiting another if statement is entered to see if the vehicle has any other lots on board. In the models for this thesis, each vehicle is only capable of carrying one lot at a time so there will never be other lots on board. Once it confirms that there are no other lots on board the claim statement searches through all of the waiting lots that appear in the same system as the vehicle, referred to here as AAmapSystemPtr. The function tells the vehicle to claim the "loads waiting first" or the first load out of the list of loads that are waiting. The list of waiting loads is a work list that is ordered by the time a load has been waiting in the queue. Choosing the first load on the work list will choose the lot that has been waiting the longest. The order of loads on the work list can also be changed elsewhere in the model. Many of the dispatching rules could alternatively be programmed through altering the work list, but this does not provide the same amount of flexibility and capability as in the source files. For all dispatching rules programmed for these experiments, the work list is left to its default settings.

FEFS Vehicle Dispatching Rule

The FEFS dispatching rule chooses the lot that is closest to the vehicle to be picked up next. This distance is dependent on the layout of the fab and locations of the loads and vehicle which are all part of the AutoMod model. Since no additional information is needed from AutoSched, the amap extension and *messages.m* source file do not need to be altered. A variable of load pointer type, ChosenLot is added to the model under Variables in the Process System to be used in the *FJobFinished* function. This variable is also used in the remainder of the dispatching rules. The FEFS dispatching rule is programmed into the model in the *FJobFinished* function shown below.

```
begin FJobFinished function
        if theVehicle AAmapSystemPtr loads waiting size > 0 then begin
                if theVehicle AAmapLot = null then begin
                        choose the first load from theVehicle AAmapSystemPtr loads
waiting whose AAmapFromLocation distance to theVehicle current location is minimum
                                save choice as ChosenLot
                        claim ChosenLot for theVehicle
                end
        end
        return true
end
```

The claim line in the original code is replaced here by three lines. The first line is a choose statement that selects the load from the work list whose distance to the location of the vehicle is minimum. That load is then saved to the ChosenLot variable in the second line. The claim line then assigns that load to the vehicle.

STT Vehicle Dispatching Rule

The STT dispatching rule selects the load that requires the shortest amount of travel time from its current location to its destination. It does not include the time a vehicle must travel empty to pick up the load. As an estimate, the distance between the source and destination locations of the load is used in place of the travel time and no possible delays are taken into account. The first step in programming the STT dispatching rule is to add an attribute, TravelTime, under Loads in the process system. The value of the attribute is programmed to be filled in the *FAmapMove* function in the *messages.m* source. Since each of the locations used and the distance are already a part of the AutoMod model, the amap extension and the *model mmsyncmessage* function do not need to be altered. An extra set statement is added to the *FAmapMove* function to record the distance into the attribute shown below.

```
set VAmapMasterLoad TravelTime to VAmapMasterLoad AAmapFromLocation
distance to VAmapMasterLoad AAmapToLocation
```

Similar to the FEFS rule, the claim statement in the *FJobFinished* function is replaced with choose, save, and claim statements. The code for the function is shown below.

```
begin FJobFinished function
        if theVehicle AAmapSystemPtr loads waiting size > 0 then begin
                if theVehicle AAmapLot = null then begin
                        choose the first load from theVehicle AAmapSystemPtr loads
waiting whose TravelTime is minimum
                                save choice as ChosenLot
                        claim ChosenLot for theVehicle
                end
        end
        return true
end
```

The choose statement selects the load from the work list with the minimum travel time, the save statement assigns that load to the ChosenLot variable, and the claim statement assigns the load to the vehicle.

EDD Vehicle Dispatching Rule

The EDD dispatching rule chooses the waiting lot which has the earliest due date to be completely finished processing in an attempt to minimize late lots. The due date of the lots is not an attribute that was transferred from AutoSched to AutoMod. This attribute needs extra programming in the amap extension to be transferred between models. The rules CR and CRQ also need an attribute transferred from AutoSched to AutoMod which require similar programming. Therefore, the code to transfer the attributes for all three rules is explained in this section. The amap extension code for the *travelToStorage* function, which is shown in the FIFO dispatching rule section, altered to transfer all needed attributes is shown below.

```
FIString fromLocation;
FIString toLocation;

if (aProj->runWithAutoMod() && theSE->prevSystem()) {
        fromLocation = theSE->prevSystem()->name() + ":" + theSE-
>curStorage()->name();
```

```
        toLocation = theSE->prevSystem()->name() + ":" + theSE-
>nextStorage()->name();
}

// Calculate CR and CRQ
float CR;
float CRQ;

FIReal aDelta(((FILot*)theSE)->dueTime().secondsFrom(modelTime()));

if (aDelta > 0.0) {
        CR = (aDelta / theSE->remainingEstimatedProcessingSeconds());
}
else
{ // already late.
        CR = (aDelta * theSE->remainingEstimatedProcessingSeconds());
}

int counterSteps = 0;
FIDLinkListIterator p_it(theSE->route()->steps());
FIStep*         aPrimaryStep;
FIString currentStepName = theSE->step()->name();
bool flag = 0;
while ((aPrimaryStep = (FIStep*)p_it()) != 0) {

        if (!strcmp(aPrimaryStep->name(),currentStepName))
                // if they are equal
        {
                flag = 1;
        }

        if (flag)
        {
                counterSteps++;
        }
}
int remainingSteps = counterSteps;
        // counterSteps now has the number of steps for the remainder of the route

float CRQDen = theSE->remainingEstimatedProcessingSeconds() +
(remainingSteps * AVG_QUEUE_DELAY);

if (aDelta > 0.0) {
        CRQ = (aDelta / CRQDen);
}
else
```

```
{ // already late.
        CRQ = (aDelta * CRQDen);
}

MMSyncMsg Message;
Message.SetType(MOVE_REQUEST);
Message.SetLong(1, 2);                          //Lot ptr
Message.SetString(2, theSE->name());            //Lot name string
Message.SetLong(2,(FIInt)theSE);                //Lot ptr
Message.SetString(3,fromLocation);              //From name string
Message.SetString(4,toLocation);                //To name string
Message.SetLong(5, 0);                          //Message Number

// the destination
FIResFam* aStnFam = theSE->step()->firstRequiredStationFam();
if (aStnFam)
{
        Message.SetString(5,aStnFam->name());
}

long dueTime = aDelta;
Message.SetLong(6, dueTime);
Message.SetString(7,(FIString)CR);
Message.SetString(8,(FIString)CRQ);

SyncSendMessage(Message);
```

After the fromLocation and toLocation variables are assigned, new code is entered to calculate the CR and CRQ variables. First, the numerator of the CR is calculated, aDelta, as the number of seconds left until the lot is due. That number is then multiplied or divided by the remaining processing time of the lot depending on whether or not the lot is already late. This is the code that is shown as an example for how to calculate CR. The next section of code is a while loop to determine the number of steps that are left in the route. This loop counts the number of remaining steps by looping through the steps starting at the current step and stores the number in the variable remainingSteps to be used in the calculation of CRQ. The CRQ denominator is then calculated by taking the remaining processing time and adding to it the

number of steps remaining in the route multiplied by the average queue delay per step. The CRQ variable is then filled by multiplying or dividing the numerator from the CR, aDelta, by the denominator for CRQ, CRQden. The next section of code inputs the lot's attribute values into the Message object to be sent to AutoMod. The first 8 lines of code to this section are not changed. Although included in the original code provided by Brooks, the destination is not sent correctly. When checked during the model runtime, the destination value is always empty. It is altered in this code to send the station family required for the next step. Next, dueTime was saved into the array. This is the time until due used as the numerator for CR and CRQ. This time is used for the ranking of the EDD dispatching rule. Finally, CR and CRQ are entered into the object, and the *SyncSendMessage* function is called to send the object to AutoMod. When this code is compiled, a backup .dll file is created named amapg.dll. The original amap.dll file used to run the amap extension is left in tact. To use the new code in the amapg.dll file during a simulation run, the model must be run in the debug controller.

The first step in preparing to receive the new attributes in AutoMod is to create a place for them to be stored. This is done by adding three load attributes under Loads in the Process System: AAmapDueTime, AAmapCR, and AAmapCRQ. The code below shows the AutoMod *mmsyncmessage* function in the *messages.m* file that was altered to transfer these three attributes.

```
begin model mmsyncmessage function

    set VAmapTempInt = MMSyncReadMessageType()

    if (VAmapTempInt = VAmapMOVE_REQUEST) then begin
        /*      Move message - moves a lot from one storage to another.
                String Parameters
                1   None
                2 - Lot Name
                3   From Location <system:location>
                4 - To Location <system:location>
                5 - Final Destination Storage
```

```
                    7 – Critical Ratio
                    8 – Critical Ratio plus Queue time

                    Integer Parameters
                    1   Lot Type (1=FOUP, 2=LOT, 3=RETICLE)
                    2 - Lot Pointer */
                    6 – Time until due

            call FAmapMove(MMSyncReadMessageInteger(1),
MMSyncReadMessageString(2), MMSyncReadMessageInteger(2),
                    MMSyncReadMessageString(3),  MMSyncReadMessageString(4),
MMSyncReadMessageInteger(5) ,
                                MMSyncReadMessageString(5),
MMSyncReadMessageInteger(6), MMSyncReadMessageString(7),
MMSyncReadMessageString(8))
        end
```

Comparing this code to the original *mmsyncmessage* function, three additional parameters are received from the amap extension and used to call the *FAmapMove* function. These parameters are the lot attributes of time until the due date of the lot, the critical ratio, and the critical ratio plus queue time. Three parameters are added to the *FAmapMove* function. Due to the syntax of AutoMod, they can not be added through the actual code. Instead, they are added in the *FAmapMove* function under Functions in the Process System. Three extra set statements are added to the *FAmapMove* function code in the *messages.m* source file to copy these parameters into the appropriate load attributes in AutoMod. This new code is shown below.

```
set VAmapMasterLoad AAmapCR = CR
set VAmapMasterLoad AAmapCRQ = CRQ
set VAmapMasterLoad AAmapDueTime = dueTime
```

Finally, the dispatching rule is programmed in the *FJobFinished* function of the *veh.m* source file. The code is displayed below.

```
begin FJobFinished function
        if theVehicle AAmapSystemPtr loads waiting size > 0 then begin
```

```
                if theVehicle AAmapLot = null then begin
                        choose the first load from theVehicle AAmapSystemPtr loads
waiting whose AAmapDueTime is minimum
                                save choice as ChosenLot
                        claim ChosenLot for theVehicle
                end
        end
        return true
end
```

The lot is chosen from the work list that has the minimum DueTime. This is the lot with the minimum time until the due date, or the lot that is closest to its due date. That lot is then saved into the ChosenLot variable and then claimed by the vehicle as the next lot it will transfer.

CR Vehicle Dispatching Rule

The CR dispatching rule chooses the lot with the lowest critical ratio to be transported. Like the EDD and CRQ dispatching rules, CR requires an additional attribute to be transferred from AutoSched to AutoMod. The programming required for this is included under the EDD dispatching rule. The code below shows the code for the dispatching rule in the *FJobFinished* function of the *veh.m* source file in AutoMod.

```
begin FJobFinished function
        if theVehicle AAmapSystemPtr loads waiting size > 0 then begin
                if theVehicle AAmapLot = null then begin
                        choose the first load from theVehicle AAmapSystemPtr loads
waiting whose AAmapCR is minimum
                                save choice as ChosenLot
                        claim ChosenLot for theVehicle
                end
        end
        return true
end
```

The choose statement selects the lot with the minimum critical ratio. The save statement saves that lot into the ChosenLot variable. The claim statement then assigns the chosen lot as the next lot for the vehicle to transport.

CRQ Vehicle Dispatching Rule

The CRQ dispatching rule chooses the lot with the lowest critical ratio plus queue time to be transported. Like the EDD and CR dispatching rules, CRQ requires an additional attribute to be transferred from AutoSched to AutoMod. The programming required for this is included under the EDD dispatching rule. The code below shows the code for the dispatching rule in the *FJobFinished* function of the *veh.m* source file in AutoMod.

```
begin FJobFinished function
        if theVehicle AAmapSystemPtr loads waiting size > 0 then begin
                if theVehicle AAmapLot = null then begin
                        choose the first load from theVehicle AAmapSystemPtr loads
waiting whose AAmapCRQ is minimum
                                save choice as ChosenLot
                        claim ChosenLot for theVehicle
                end
        end
        return true
end
```

The choose statement selects the lot with the minimum critical ratio plus queue time. The save statement saves that lot into the ChosenLot variable. The claim statement then assigns the chosen lot as the next lot for the vehicle to transport.

MUS Vehicle Dispatching Rule

The MUS vehicle dispatching rule selects the lot that is traveling to the station with the highest utilization. For this experiment the station utilizations used in the programming of this rule are taken from a baseline run of the model with FIFO used as both the machine and vehicle dispatching rules. Since one message from AutoSched to AutoMod can only move lots within one system, the immediate destination of many lots according to that message would be a stocker. However, the next tool that the lot will visit is used to determine the station utilization for the lot. An attribute is already present in AutoMod to receive the final destination station of

a lot. Upon further study, this attribute is not receiving the final destination station value and is always empty in the model. Since all of the programming is set up to receive the attribute in AutoMod, no additional programming needs to be performed. The amap extension is altered to send the station name to the destination attribute as documented in the EDD dispatching rule section above.

In addition to the destination attribute, a rank attribute, AAmapDestinationRank, is added to the load attributes under Load in the Process System. The value of the rank attribute is assigned in the *FAmapMove* function of the *messages.m* source file based on the utilization of the destination station of the lot. The highest utilized station family is given a rank of 1, the second highest utilized 2, and so on. The code below shows an example of the code for the *FAmapMove* function from the make-to-order fab. This code only includes the first and last two lines of code that assign a value to the rank attribute. The remainder of the code can be found in the simulation model files in the Appendix M and Appendix N CDs.

```
if (destination = "FURN3_Furnace_Tube") then set VAmapMasterLoad
AAmapDestinationRank = 1
        else if (destination = "FURN4_Furnace_Tube") then set VAmapMasterLoad
AAmapDestinationRank = 2
        else if (destination = "STI7_Dry_Rinse") then set VAmapMasterLoad
AAmapDestinationRank = 73
        else print "The destination rank has not been set, destination is " destination to
message
```

The raw data for the simulation results for the FIFO baseline model used to determine the destination rank values is included in Table C.1 for the ASIC fab and Table C.2 for the make-to-order fab. The code used to program the dispatching rule in the *FJobFinished* function of the *veh.m* source file is shown below.

```
begin FJobFinished function
        if theVehicle AAmapSystemPtr loads waiting size > 0 then begin
                if theVehicle AAmapLot = null then begin
```

```
                choose the first load from theVehicle AAmapSystemPtr loads
waiting whose AAmapDestinationRank is minimum
                        save choice as ChosenLot
                choose ChosenLot for theVehicle
            end
        end
        return true
end
```

The choose statement selects the lot with the lowest rank attribute, or highest utilization. The save statement then saves the selection into the ChosenLot variable. Finally, the claim statement assigns that lot to the vehicle.

Table C.1: Station Family Utilizations for FIFO Baseline Model of the ASIC Fab

Station Family	**Utilization**	**Rank**
Develop: Multi-Sequence_stn2	99.74%	1
Apply Resist: Conveyor_stn1	94.32%	2
Expose UV: Single Wafer_stn2	90.73%	3
Clean: Multi-Sequence_stn3	87.33%	4
Bake: Batch_stn1	86.72%	5
Nitride Deposition: Batch_stn1	86.03%	6
Develop: Multi-Sequence_stn1	82.66%	7
Clean: Multi-Sequence_stn6	78.60%	8
Clean: Conveyor_stn1	78.19%	9
Sputter: Batch_stn3	74.64%	10
Clean: Multi-Sequence_stn1	74.43%	11
Evaporation: Batch_stn2	74.19%	12
Apply Resist: Conveyor_stn3	72.98%	13
Lift Off: Batch_stn2	72.75%	14
Sputter: Batch_stn2	72.58%	15
Clean: Inspect_stn2	71.22%	16
Reflow: Single Wafer_stn1	70.81%	17
Preclean: Multi-Sequence_stn1	69.85%	18
Strip: Multi-Sequence_stn1	68.82%	19
Lift Off: Batch_stn3	67.84%	20
Reactive Ion Etch: Single_stn1	67.83%	21
Apply Resist: Batch_stn1	65.84%	22
Clean: Multi-Sequence_stn2	63.71%	23
Silylation: Multi-Sequenc_stn1	63.52%	24

Develop E-beam: Multi-Seq_stn1	63.00%	25
Evaporation: Batch_stn1	62.26%	26
Apply Resist: Conveyor_stn2	62.20%	27
Etch: Multi-Sequence_stn1	61.33%	28
Clean: Multi-Sequence_stn4	60.18%	29
Sinter: Batch_stn2	60.12%	30
Apply Resist: Conveyor_stn4	59.00%	31
Chem/Mech Polish: Batch_stn1	58.52%	32
In Line Test: Single_stn1	57.38%	33
Clean: Single Wafer_stn1	56.56%	34
Sinter: Batch_stn1	56.56%	35
Bake: Batch_stn5	56.40%	36
Chem/Mech Polish: Batch_stn2	54.69%	37
Align: Single Wafer_stn1	53.99%	38
Ash: Single Wafer_stn3	53.66%	39
Ash: Single Wafer_stn2	53.58%	40
Measure: Batch_stn5	52.89%	41
Measure: Batch_stn1	49.22%	42
Measure: Inspect_stn3	47.53%	43
Lift Off: Batch_stn1	47.00%	44
Measure: Batch_stn2	46.61%	45
Bake: Single Wafer_stn1	45.92%	46
Evaporation: Batch_stn3	45.27%	47
Expose E-beam: Single_stn1	44.98%	48
Test: Single Wafer_stn1	43.14%	49
Measure: Inspect_stn1	43.08%	50
Reactive Ion Etch: Single_stn2	42.37%	51
Reactive Ion Etch: Single_stn3	41.47%	52
Clean: Inspect_stn1	40.15%	53
Measure: Single Wafer_stn1	40.09%	54
Bake: Batch_stn4	38.92%	55
Bake: Batch_stn6	37.07%	56
Inspect: Inspect_stn3	36.28%	57
Quench: Multi-Sequence_stn1	35.49%	58
Measure: Inspect_stn2	35.17%	59
Inspect: Inspect_stn5	34.86%	60
Clean: Conveyor_stn2	34.73%	61
Measure: Batch_stn6	34.01%	62
Measure: Batch_stn3	33.37%	63

Inspect: Single Wafer_stn4	33.16%	64
Inspect: Single Wafer_stn1	32.61%	65
Bake: Batch_stn8	32.50%	66
Inspect: Inspect_stn1	32.32%	67
Clean: Multi-Sequence_stn5	31.88%	68
Bake: Batch_stn7	31.61%	69
Inspect: Inspect_stn4	30.56%	70
Measure: Inspect_stn4	30.00%	71
Ash: Single Wafer_stn4	28.15%	72
Ash: Single Wafer_stn1	27.81%	73
Expose UV: Single Wafer_stn1	27.37%	74
Bake: Batch_stn9	26.42%	75
ID Reader: Single Wafer_stn1	26.13%	76
Bake: Batch_stn3	25.78%	77
Inspect: Batch_stn1	24.49%	78
Bake: Batch_stn11	24.41%	79
Sputter: Batch_stn1	24.06%	80
Inspect: Single Wafer_stn3	23.69%	81
E-beam Inspect: Inspect_stn1	22.73%	82
Inspect: Inspect_stn6	19.55%	83
Inspect: Single Wafer_stn2	18.62%	84
Bake: Batch_stn2	17.94%	85
Inspect: Batch_stn2	16.29%	86
Measure: Batch_stn4	16.19%	87
Inspect: Inspect_stn2	14.71%	88
Clean: Batch_stn1	14.12%	89
Bake: Batch_stn10	13.16%	90

Table C.2: Station Family Utilizations for FIFO Baseline Model of the Make-to-Order Fab

Station Family	**Utilization**	**Rank**
FURN3_Furnace_Tube	99.96%	1
FURN4_Furnace_Tube	97.54%	2
FURN5_Furnace_Tube	97.46%	3
STEP_Photo_Stepper	96.76%	4
MED_Med_Current_Implant	95.69%	5
HIGH_High_Current_Implant	94.73%	6
FURN1_Furnace_Tube	93.00%	7
PR3_Probe	90.41%	8

PIRH2_Strip	87.57%	9
PHOS_Furnace_Tube	85.20%	10
COAT5_Coater	80.46%	11
PLAM_Dry_Etch	77.30%	12
LFE_Asher	75.95%	13
PRPH_Wet_Etch	75.78%	14
CVD5_CVD	71.89%	15
CVD1_CVD	70.46%	16
CVD4_CVD	69.72%	17
WATJ_Metal_Dep	68.17%	18
PR1_Probe	66.45%	19
ITP_Metrology	65.30%	20
COAT1_Coater	64.35%	21
FURN2_Furnace_Tube	64.20%	22
PR2_Probe	59.76%	23
TERM_Computer_terminal	56.47%	24
RINS2_Rinse	56.16%	25
CVD2_CVD	52.53%	26
EVAP_Metal_Dep	52.15%	27
NANO_Metrology	48.41%	28
MLAM_Dry_Etch	48.12%	29
TEG_Dry_Etch	47.51%	30
STRP_Strip	45.89%	31
METL_Metal_Dep	44.79%	32
VICK_Metrology	43.96%	33
PIRH1_Strip	43.81%	34
DEV_Develop	42.28%	35
HOOD_Rinse	40.77%	36
OLAM_Dry_Etch	38.37%	37
BAGR_Backgrind	38.23%	38
CVD3_CVD	37.99%	39
COAT4_Coater	37.91%	40
COAT2_Coater	37.87%	41
SCOP2_Metrology	37.67%	42
NITH_Wet_Etch	37.22%	43
LIS_Laser_Scribe	34.04%	44
COAT3_Coater	33.34%	45
TUBE_Alloy	33.21%	46
STI5_Dry_Rinse	32.68%	47

DEGH_Deglaze	32.45%	48
MTRX_Descum	30.02%	49
OXIH_Wet_Etch	28.61%	50
FSI_Clean	27.19%	51
STI6_Dry_Rinse	26.86%	52
STI3_Dry_Rinse	25.66%	53
RINS1_Rinse	24.99%	54
AERO_Metrology	23.98%	55
POLH_Wet_Etch	23.45%	56
STI4_Dry_Rinse	20.41%	57
ETCH_Clean	16.82%	58
RCA_Clean	15.19%	59
GENU_Sputter	13.57%	60
SCOP1_Metrology	12.48%	61
PAL_Photo_Aligner	12.28%	62
PRMH_Rinse	12.14%	63
ESTI_Dryer	11.77%	64
PERK_Photo_Aligner	11.59%	65
STI2_Dry_Rinse	11.46%	66
XLAM_Dry_Etch	11.29%	67
HB_Hard_Bake	6.82%	68
METH_Wet_Etch	5.54%	69
NOV_CVD	4.53%	70
STI1_Dry_Rinse	3.25%	71
BRAN_Asher	1.01%	72
STI7_Dry_Rinse	0.56%	73

Appendix D. ASIC Fab Dispatching Rule Raw Data

This appendix contains the raw data results for the ASIC fab obtained from the dispatching rule experiment. Table D.1 shows the average lot cycle time, standard deviation of cycle time, average WIP, average X Theoretical value, and the percentage of on time lots for all three replications of each rule combination. All replications marked with "*" after the machine rule do not reach steady state. In some cases where steady state is not reached, the simulation is not able to run through all three replications and no data is recorded.

Table D.1: Dispatching Rule Experiment Raw Data for the ASIC Fab

Vehicle Dispatching Rule	Machine Dispatching Rule	Average Cycle Time (days)	Standard Deviation of Cycle Time (days)	Average WIP (lots)	Average X Theoretical Value	Percentage of on time lots
FEFS	SPT	22.60	27.41	255.02	7.04	80.13
FEFS	SPT	25.35	35.01	249.95	8.01	79.95
FEFS	SPT	15.60	5.34	164.09	4.74	93.53
FEFS	EDD	17.80	7.98	187.29	5.21	96.19
FEFS	EDD	17.51	7.73	184.11	5.13	97.54
FEFS	EDD	16.84	7.34	177.22	4.94	99.45
FEFS	FIFO	15.97	5.34	168.07	4.81	62.26
FEFS	FIFO	15.69	5.17	164.93	4.73	66.46
FEFS	FIFO	15.28	4.86	160.74	4.62	79.13
FEFS	CR	16.00	5.15	168.25	4.81	52.31
FEFS	CR	15.92	5.14	167.49	4.79	61.11
FEFS	CR	15.80	5.07	166.15	4.75	74.89
FEFS	CRQ	20.70	7.54	217.8	6.13	97.86
FEFS	CRQ	18.41	7.19	193.5	5.45	99.46
FEFS	CRQ	18.93	7.31	199.2	5.62	99.69
FIFO	SPT*	29.44	53.01	368.49	9.4	75.65
FIFO	SPT*	77.33	124.90	2149.42	25.58	53.98
FIFO	SPT*					
FIFO	EDD	25.05	11.88	264.09	7.28	36.99
FIFO	EDD	29.10	12.90	306.01	8.45	8.85
FIFO	EDD	25.29	11.75	266.03	7.34	39.83
FIFO	FIFO	19.53	7.35	205.71	5.83	77.27
FIFO	FIFO	20.34	7.20	213.68	6.06	77.51
FIFO	FIFO	19.45	6.93	204.6	5.81	87.49
FIFO	CR	21.65	7.67	227.76	6.42	62.04

FIFO	CR	21.79	7.67	229.2	6.46	48.05
FIFO	CR	21.63	7.62	227.54	6.41	79.71
FIFO	CRQ	21.65	7.67	227.76	6.42	62.04
FIFO	CRQ	21.79	7.67	229.2	6.46	48.05
FIFO	CRQ	21.63	7.62	227.54	6.41	79.71
STT	SPT*	23.10	22.14	256.62	7.18	76.13
STT	SPT*	34.54	62.85	396.37	11.11	73.51
STT	SPT*	120.56	109.70	3887.4	40.68	4.27
STT	EDD	20.19	9.35	212.66	5.89	75.9
STT	EDD	19.96	8.94	209.64	5.82	78.7
STT	EDD	19.49	8.64	205.02	5.7	82.28
STT	FIFO	16.92	5.74	178.03	5.09	99.08
STT	FIFO	17.03	5.79	179.05	5.12	97.61
STT	FIFO	15.93	5.13	167.57	4.81	99.17
STT	CR	20.42	7.54	214.84	6.05	96.5
STT	CR	19.87	7.44	209	5.89	98.74
STT	CR	19.57	7.36	205.82	5.8	99.43
STT	CRQ	20.42	7.54	214.84	6.05	96.5
STT	CRQ	19.87	7.44	209	5.89	98.74
STT	CRQ	19.57	7.36	205.82	5.8	99.43
EDD	SPT	20.28	19.41	230	6.26	83.9
EDD	SPT	26.84	34.35	267.22	8.48	73.7
EDD	SPT	18.96	8.65	198.89	5.8	80.11
EDD	EDD	20.76	9.67	218.53	6.04	70.71
EDD	EDD	21.27	9.53	223.51	6.19	63
EDD	EDD	20.21	9.11	212.79	5.9	72.89
EDD	FIFO	16.68	5.70	175.47	5.01	99.47
EDD	FIFO	16.40	5.49	172.45	4.93	99.04
EDD	FIFO	16.56	5.56	174.25	4.98	99.06
EDD	CR	21.02	7.60	221.09	6.23	93.94
EDD	CR	20.83	7.56	219.05	6.17	97.28
EDD	CR	20.25	7.47	212.95	5.99	98.91
EDD	CRQ	21.02	7.60	221.09	6.23	93.94
EDD	CRQ	20.83	7.56	219.05	6.17	97.28
EDD	CRQ	20.25	7.47	212.95	5.99	98.91
CR	SPT*	26.54	33.32	372.51	8.33	74.91
CR	SPT*	124.48	149.17	3551.46	42.02	37.79
CR	SPT*					
CR	EDD	24.64	11.58	259.64	7.17	38.08
CR	EDD	26.04	11.79	273.7	7.56	26.3
CR	EDD	25.23	11.56	265.37	7.33	37.41
CR	FIFO	21.27	8.11	223.94	6.32	53.47
CR	FIFO	22.80	8.28	239.65	6.75	29.18
CR	FIFO	19.71	6.94	207.27	5.88	94.47
CR	CR	21.21	7.58	223.08	6.28	93.58

CR	CR	21.02	7.56	221.09	6.23	94.81
CR	CR	20.67	7.54	217.42	6.12	96.66
CR	CRQ	21.21	7.58	223.08	6.28	93.58
CR	CRQ	21.02	7.56	221.09	6.23	94.81
CR	CRQ	20.67	7.54	217.42	6.12	96.66
CRQ	SPT*	31.39	46.78	347.12	10.01	70.42
CRQ	SPT*	26.57	38.79	1803.79	8.4	75.91
CRQ	SPT*					
CRQ	EDD	24.31	11.47	256.19	7.07	42.41
CRQ	EDD	24.93	11.32	261.95	7.24	37.32
CRQ	EDD	24.26	11.07	255.18	7.05	43.96
CRQ	FIFO	19.39	7.26	204.09	5.79	82.27
CRQ	FIFO	20.72	7.41	218.15	6.17	77.15
CRQ	FIFO	19.15	6.73	201.15	5.72	95.66
CRQ	CR	21.40	7.60	225.04	6.34	91.01
CRQ	CR	21.10	7.55	221.93	6.25	93.99
CRQ	CR	20.07	7.47	211.12	5.95	96.8
CRQ	CRQ	21.40	7.60	225.04	6.34	91.01
CRQ	CRQ	21.10	7.55	221.93	6.25	93.99
CRQ	CRQ	20.07	7.47	211.12	5.95	96.8
MUS	SPT	18.90	12.99	201.86	5.8	84.99
MUS	SPT	19.36	11.78	200.49	5.96	81.1
MUS	SPT	15.82	5.26	166.43	4.81	93.47
MUS	EDD	19.12	8.56	201.07	5.59	86.85
MUS	EDD	18.02	7.79	189.48	5.29	96.72
MUS	EDD	17.56	7.56	184.68	5.16	97.74
MUS	FIFO	16.30	5.51	171.43	4.91	98.62
MUS	FIFO	15.63	5.00	164.33	4.72	98.84
MUS	FIFO	15.41	4.87	162.1	4.66	99.3
MUS	CR	20.17	7.51	212.27	5.98	97.93
MUS	CR	19.12	7.34	200.93	5.67	99
MUS	CR	18.09	6.96	190.42	5.37	99.74
MUS	CRQ	20.17	7.51	212.27	5.98	97.93
MUS	CRQ	19.12	7.34	200.93	5.67	99
MUS	CRQ	18.09	6.96	190.42	5.37	99.74

Appendix E. Make-to-Order Dispatching Rule Raw Data

This appendix contains the raw data results for the make-to-order fab obtained from the dispatching rule experiment. The same statistics that are collected for the ASIC fab are also collected for the make-to-order fab. These are shown in Table E.1. All replications marked with "*" after the machine rule do not reach steady state. In some cases where steady state is not reached, the simulation is not able to run through all three replications and no data is recorded.

Table E.1: Dispatching Rule Experiment Raw Data for Make-to-Order Fab

Vehicle Dispatching Rule	Machine Dispatching Rule	Average Cycle Time (days)	Standard Deviation of Cycle Time (days)	Average WIP (lots)	Average X Theoretical Value	Percentage of on time lots
FEFS	SPT	19.97	5.84	266.83	2.1	50.58
FEFS	SPT	19.94	5.94	266.45	2.09	50.99
FEFS	SPT	20.21	6.03	269.84	2.12	45.75
FEFS	EDD	19.68	6.14	262.62	2.05	46.08
FEFS	EDD	19.32	5.98	257.9	2.01	55.89
FEFS	EDD	19.32	5.95	257.92	2.01	55.76
FEFS	FIFO	18.28	4.30	244.01	1.94	94.8
FEFS	FIFO	18.37	4.35	245.3	1.95	92.34
FEFS	FIFO	18.33	4.31	244.74	1.95	93.74
FEFS	CR	20.78	5.02	277.44	2.21	1.19
FEFS	CR	20.95	5.06	279.66	2.23	0.99
FEFS	CR	21.03	5.07	280.69	2.24	0.37
FEFS	CRQ	20.78	5.02	277.44	2.21	1.19
FEFS	CRQ	20.95	5.06	279.66	2.23	0.99
FEFS	CRQ	21.03	5.07	280.69	2.24	0.37
FIFO	SPT	20.77	6.29	277.34	2.18	37.03
FIFO	SPT	21.01	6.41	280.47	2.2	32.61
FIFO	SPT	21.21	6.42	283.37	2.22	29.64
FIFO	EDD	20.34	6.27	271.5	2.12	32.16
FIFO	EDD	20.05	6.18	267.57	2.09	35.7
FIFO	EDD	20.03	6.14	267.45	2.09	36.04
FIFO	FIFO	18.68	4.38	249.37	1.99	86.44
FIFO	FIFO	18.64	4.39	248.9	1.98	84.86
FIFO	FIFO	18.65	4.38	249.08	1.99	86.23
FIFO	CR	26.52	6.44	354.26	2.82	0
FIFO	CR	27.11	6.46	361.88	2.89	0
FIFO	CR	27.46	6.50	366.52	2.93	0

FIFO	CRQ	26.52	6.44	354.26	2.82	0
FIFO	CRQ	27.11	6.46	361.88	2.89	0
FIFO	CRQ	27.46	6.50	366.52	2.93	0
STT	SPT*	52.13	19.22	1062.71	5.7	0
STT	SPT*	139.91	128.94	1684.64	14.34	0
STT	SPT*	142.28	120.07	1954.42	14.74	0
STT	EDD	20.41	6.30	272.52	2.12	31.69
STT	EDD	20.06	6.18	267.68	2.09	35.96
STT	EDD	20.12	6.20	268.61	2.1	34.59
STT	FIFO	18.58	4.38	248.04	1.98	88.75
STT	FIFO	18.69	4.41	249.4	1.99	85.91
STT	FIFO	18.67	4.41	249.37	1.99	86.43
STT	CR*	36.63	11.18	611.1	3.98	0
STT	CR*	58.40	17.86	1548.07	6.36	0
STT	CR*					
STT	CRQ*	36.63	11.18	611.1	3.98	0
STT	CRQ*	58.40	17.86	1548.07	6.36	0
STT	CRQ*					
EDD	SPT	20.99	6.73	280.52	2.19	38.97
EDD	SPT	20.89	6.63	278.77	2.18	38.08
EDD	SPT	21.53	7.09	288.1	2.24	32.67
EDD	EDD	20.48	6.40	273.49	2.13	31.49
EDD	EDD	20.59	6.42	274.87	2.14	31.06
EDD	EDD	20.25	6.31	270.44	2.11	32.24
EDD	FIFO	18.62	4.55	248.64	1.98	87.28
EDD	FIFO	18.60	4.55	248.38	1.97	87.18
EDD	FIFO	18.79	4.60	250.87	1.99	81.07
EDD	CR	25.88	6.32	345.45	2.75	0
EDD	CR	26.98	6.39	360.23	2.87	0
EDD	CR	27.53	6.44	367.54	2.94	0
EDD	CRQ	25.88	6.32	345.45	2.75	0
EDD	CRQ	26.98	6.39	360.23	2.87	0
EDD	CRQ	27.53	6.44	367.54	2.94	0
CR	SPT	20.47	5.94	273.59	2.15	41.57
CR	SPT	19.92	5.57	265.91	2.1	50.98
CR	SPT	20.58	6.10	274.83	2.16	40.67
CR	EDD	19.95	6.00	266.4	2.08	38.87
CR	EDD	19.67	5.86	262.45	2.05	47.12
CR	EDD	19.74	5.89	263.6	2.06	44.97
CR	FIFO	18.51	4.29	247.11	1.97	91.55
CR	FIFO	18.52	4.29	247.32	1.97	91.06
CR	FIFO	18.51	4.27	247.14	1.97	91.78
CR	CR	24.04	6.08	320.95	2.55	0
CR	CR	24.83	6.17	331.51	2.64	0
CR	CR	25.46	6.25	339.89	2.71	0

CR	CRQ	24.04	6.08	320.95	2.55	0
CR	CRQ	24.83	6.17	331.51	2.64	0
CR	CRQ	25.46	6.25	339.89	2.71	0
CRQ	SPT	20.77	6.29	277.34	2.18	37.03
CRQ	SPT	21.01	6.41	280.47	2.2	32.61
CRQ	SPT	21.21	6.42	283.37	2.22	29.64
CRQ	EDD	20.34	6.27	271.5	2.12	32.16
CRQ	EDD	20.05	6.18	267.57	2.09	35.7
CRQ	EDD	20.03	6.14	267.45	2.09	36.04
CRQ	FIFO	18.44	4.28	246.16	1.96	94.36
CRQ	FIFO	18.51	4.31	247.14	1.97	92.26
CRQ	FIFO	18.51	4.29	247.08	1.97	92.56
CRQ	CR	24.29	6.12	324.23	2.58	0
CRQ	CR	25.37	6.21	338.72	2.7	0
CRQ	CR	25.67	6.27	342.68	2.73	0
CRQ	CRQ	24.29	6.12	324.23	2.58	0
CRQ	CRQ	25.37	6.21	338.72	2.7	0
CRQ	CRQ	25.67	6.27	342.68	2.73	0
MUS	SPT*	25.83	10.03	356.73	2.71	4.1
MUS	SPT*	42.97	34.28	620.93	4.5	0.02
MUS	SPT*	65.97	61.04	914.81	6.87	0
MUS	EDD	20.66	6.32	275.9	2.15	30.03
MUS	EDD	20.43	6.24	272.8	2.13	31.33
MUS	EDD	20.74	6.33	276.83	2.16	28.94
MUS	FIFO	18.43	4.33	246.04	1.96	91.58
MUS	FIFO	18.50	4.35	246.91	1.97	89.07
MUS	FIFO	18.50	4.33	246.96	1.97	91.03
MUS	CR*	34.86	9.13	465.55	3.73	0
MUS	CR*	46.83	15.17	632.09	5.02	0
MUS	CR*	62.62	22.78	879.02	6.64	0
MUS	CRQ*	34.86	9.13	465.55	3.73	0
MUS	CRQ*	46.83	15.17	632.09	5.02	0
MUS	CRQ*	62.62	22.78	879.02	6.64	0

Appendix F. ASIC Fab Dispatching Rule ANOVA Results

This appendix contains the ANOVA tests performed in Minitab on the ASIC fab for the dispatching rules experiment. Tables F.1 – F.5 show the ANOVA tests for each of the performance indicators. The machine dispatching rule SPT is not included in this analysis since many of the runs with this rule do not reach steady state. It was considered to be a poor choice before the analysis began.

Table F.1: Two-way ANOVA of Cycle Time versus Vehicle, Machine for the ASIC Fab

Source	DF	SS	MS	F	P
Vehicle	6	309.182	51.5304	86.90	0.000
Machine	3	162.026	54.0088	91.08	0.000
Interaction	18	103.018	5.7232	9.65	0.000
Error	56	33.206	0.5930		
Total	83	607.433			

S = 0.7700 R-Sq = 94.53% R-Sq(adj) = 91.90%

Table F.2: Two-way ANOVA of Standard Deviation of Cycle Time versus Vehicle, Machine for the ASIC Fab

Source	DF	SS	MS	F	P
Vehicle	6	59.207	9.8679	120.25	0.000
Machine	3	153.789	51.2629	624.71	0.000
Interaction	18	37.675	2.0931	25.51	0.000
Error	56	4.595	0.0821		
Total	83	255.267			

S = 0.2865 R-Sq = 98.20% R-Sq(adj) = 97.33%

Table F.3: Two-way ANOVA of WIP versus Vehicle, Machine for the ASIC Fab

Source	DF	SS	MS	F	P
Vehicle	6	34257.7	5709.62	87.96	0.000
Machine	3	17967.3	5989.10	92.26	0.000
Interaction	18	11437.2	635.40	9.79	0.000
Error	56	3635.2	64.91		
Total	83	67297.4			

S = 8.057 R-Sq = 94.60% R-Sq(adj) = 91.99%

Table F.4: Two-way ANOVA of X Theoretical Value versus Vehicle, Machine for the ASIC Fab

Source	DF	SS	MS	F	P
Vehicle	6	25.3217	4.22028	85.66	0.000
Machine	3	10.5896	3.52988	71.65	0.000
Interaction	18	8.1725	0.45403	9.22	0.000
Error	56	2.7589	0.04927		
Total	83	46.8427			

S = 0.2220 R-Sq = 94.11% R-Sq(adj) = 91.27%

Table F.5: Two-way ANOVA of Percentage of On Time Lots versus Vehicle, Machine for the ASIC Fab

Source	DF	SS	MS	F	P
Vehicle	6	13381.5	2230.25	26.10	0.000
Machine	3	10146.8	3382.26	39.59	0.000
Interaction	18	13869.8	770.54	9.02	0.000
Error	56	4784.7	85.44		
Total	83	42182.7			

S = 9.243 R-Sq = 88.66% R-Sq(adj) = 83.19%

Appendix G. Make-to-Order Fab Dispatching Rule ANOVA Results

This appendix contains the ANOVA tests performed in Minitab on the make-to-order fab for the dispatching rules experiment. Tables G.1 – G.5 show the ANOVA tests for each of the performance indicators. The vehicle dispatching rule STT is not included in this analysis since many of the runs with this rule do not reach steady state. It was considered to be a poor choice before the analysis began.

Table G.1: Two-way ANOVA of Cycle Time versus Vehicle, Machine for the Make-to-Order Fab

Source	DF	SS	MS	F	P
Vehicle	5	2612.28	522.456	19.64	0.000
Machine	4	1647.45	411.863	15.48	0.000
Interaction	20	1690.59	84.530	3.18	0.000
Error	60	1595.86	26.598		
Total	89	7546.18			

S = 5.157 R-Sq = 78.85% R-Sq(adj) = 68.63%

Table G.2: Two-way ANOVA of Standard Deviation of Cycle Time versus Vehicle, Machine for the Make-to-Order Fab

Source	DF	SS	MS	F	P
Vehicle	5	1169.98	233.997	9.42	0.000
Machine	4	439.58	109.895	4.43	0.003
Interaction	20	1384.72	69.236	2.79	0.001
Error	60	1489.65	24.827		
Total	89	4483.93			

S = 4.983 R-Sq = 66.78% R-Sq(adj) = 50.72%

Table G.3: Two-way ANOVA of WIP versus Vehicle, Machine for the Make-to-Order Fab

Source	DF	SS	MS	F	P
Vehicle	5	528762	105752	19.19	0.000
Machine	4	307663	76916	13.96	0.000
Interaction	20	343640	17182	3.12	0.000
Error	60	330602	5510		
Total	89	1510667			

S = 74.23 R-Sq = 78.12% R-Sq(adj) = 67.54%

Table G.4: Two-way ANOVA of X Theoretical Value versus Vehicle, Machine for the Make-to-Order Fab

Source	DF	SS	MS	F	P
Vehicle	5	29.3408	5.86816	20.32	0.000
Machine	4	19.4133	4.85333	16.81	0.000
Interaction	20	18.9823	0.94911	3.29	0.000
Error	60	17.3248	0.28875		
Total	89	85.0612			

S = 0.5374 R-Sq = 79.63% R-Sq(adj) = 69.79%

Table G.5: Two-way ANOVA of Percentage of On Time Lots versus Vehicle, Machine for the Make-to-Order Fab

Source	DF	SS	MS	F	P
Vehicle	5	1963	392.6	71.22	0.000
Machine	4	97628	24406.9	4426.95	0.000
Interaction	20	3556	177.8	32.25	0.000
Error	60	331	5.5		
Total	89	103477			

S = 2.348 R-Sq = 99.68% R-Sq(adj) = 99.53%

Appendix H. ASIC Fab Tukey Tests for all Dispatching Rule Combinations

This appendix contains the Tukey Tests performed on the ASIC fab for the dispatching rule experiment. Tables H.1 – H.5 show the tests for each performance measure. These tests are performed with all of the rule combinations except the combinations that contain SPT as the machine dispatching rule.

Table H.1: ASIC Fab Tukey Test on Average Cycle Time for all Rule Combinations

Dispatching Rule	Average Cycle Time (days)	Significance
VFEFS MFIFO	15.647	
VMUS MFIFO	15.78	
VFEFS MCR	15.907	
VEDD MFIFO	16.547	
VSTT MFIFO	16.627	
VFEFS MEDD	17.383	
VMUS MEDD	18.233	
VMUS MCR	19.127	
VMUS MCRQ	19.127	
VFEFS MCRQ	19.347	
VCRQ MFIFO	19.753	
VFIFO MFIFO	19.773	
VSTT MEDD	19.88	
VSTT MCR	19.953	
VSTT MCRQ	19.953	
VEDD MCR	20.7	
VEDD MCRQ	20.7	
VEDD MEDD	20.747	
VCRQ MCR	20.857	
VCRQ MCRQ	20.857	
VCR MCR	20.967	
VCR MCRQ	20.967	
VCR MFIFO	21.26	
VFIFO MCR	21.69	
VFIFO MCRQ	21.69	
VCRQ MEDD	24.5	
VCR MEDD	25.303	
VFIFO MEDD	26.48	

Table H.2: ASIC Fab Tukey Test on Standard Deviation of Cycle Time for all Rule Combinations

Dispatching Rule	Standard Deviation of Cycle Time (days)	Significance
VFEFS MCR	5 12	
VFEFS MFIFO	5 123	
VMUS MFIFO	5 127	
VSTT MFIFO	5 553	
VEDD MFIFO	5 583	
VCRQ MFIFO	7 133	
VFIFO MFIFO	7 16	
VMUS MCR	7 27	
VMUS MCRQ	7 27	
VFEFS MCRQ	7 347	
VSTT MCR	7 447	
VSTT MCRQ	7 447	
VCRQ MCR	7 54	
VCRQ MCRQ	7 54	
VEDD MCR	7 543	
VEDD MCRQ	7 543	
VCR MCR	7 56	
VCR MCRQ	7 56	
VFIFO MCR	7 653	
VFIFO MCRQ	7 653	
VFEFS MEDD	7 683	
VCR MFIFO	7 777	
VMUS MEDD	7 97	
VSTT MEDD	8 977	
VEDD MEDD	9 437	
VCRQ MEDD	11 287	
VCR MEDD	11 643	
VFIFO MEDD	12 177	

Table H.3: ASIC Fab Tukey Test on Average WIP for all Rule Combinations

Dispatching Rule	Average WIP (lots)	Significance
VFEFS MFIFO	164.58	
VMUS MFIFO	165.95	
VFEFS MCR	167.3	
VEDD MFIFO	174.06	
VSTT MFIFO	174.88	
VFEFS MEDD	182.87	
VMUS MEDD	191.74	
VMUS MCR	201.21	
VMUS MCRQ	201.21	
VFEFS MCRQ	203.5	
VCRQ MFIFO	207.8	
VFIFO MFIFO	208	
VSTT MEDD	209.11	
VSTT MCR	209.89	
VSTT MCRQ	209.89	
VEDD MCR	217.7	
VEDD MCRQ	217.7	
VEDD MEDD	218.28	
VCRQ MCR	219.36	
VCRQ MCRQ	219.36	
VCR MCR	220.53	
VCR MCRQ	220.53	
VCR MFIFO	223.62	
VFIFO MCR	228.17	
VFIFO MCRQ	228.17	
VCRQ MEDD	257.77	
VCR MEDD	266.24	
VFIFO MEDD	278.71	

Table H.4: ASIC Fab Tukey Test on Average X Theoretical Value for all Rule Combinations

Dispatching Rule	Average X Theoretical Value	Significance
VFEFS MFIFO	4.72	
VMUS MFIFO	4.7633	
VFEFS MCR	4.7833	
VEDD MFIFO	4.9733	
VSTT MFIFO	5.0067	
VFEFS MEDD	5.0933	
VMUS MEDD	5.3467	
VMUS MCR	5.6733	
VMUS MCRQ	5.6733	
VFEFS MCRQ	5.7333	
VSTT MEDD	5.8033	
VCRQ MFIFO	5.8933	
VFIFO MFIFO	5.9	
VSTT MCR	5.9133	
VSTT MCRQ	5.9133	
VEDD MEDD	6.0433	
VEDD MCR	6.13	
VEDD MCRQ	6.13	
VCRQ MCR	6.18	
VCRQ MCRQ	6.18	
VCR MCR	6.21	
VCR MCRQ	6.21	
VCR MFIFO	6.3167	
VFIFO MCR	6.43	
VFIFO MCRQ	6.43	
VCRQ MEDD	7.12	
VCR MEDD	7.3533	
VFIFO MEDD	7.69	

Table H.5: ASIC Fab Tukey Test on the Percentage of On Time Lots for all Rule Combinations

Dispatching Rule	Percentage of on time lots	Significance
VEDD MFIFO	99.19	
VFEFS MCRQ	99	
VMUS MFIFO	98.92	
VMUS MCR	98.89	
VMUS MCRQ	98.89	
VSTT MFIFO	98.62	
VSTT MCR	98.22	
VSTT MCRQ	98.22	
VFEFS MEDD	97.73	
VEDD MCR	96.71	
VEDD MCRQ	96.71	
VCR MCR	95.02	
VCR MCRQ	95.02	
VCRQ MCR	93.93	
VCRQ MCRQ	93.93	
VMUS MEDD	93.77	
VCRQ MFIFO	85.03	
VFIFO MFIFO	80.76	
VSTT MEDD	78.96	
VFEFS MFIFO	69.28	
VEDD MEDD	68.87	
VFIFO MCR	63.27	
VFIFO MCRQ	63.27	
VFEFS MCR	62.77	
VCR MFIFO	59.04	
VCRQ MEDD	41.23	
VCR MEDD	33.93	
VFIFO MEDD	28.56	

Appendix I. ASIC Fab Tukey Tests for Top 6 Dispatching Rule Combinations

This appendix contains the Tukey Tests performed on the ASIC fab for the top six rule combinations in the dispatching rule experiment. Tables I.1 – I.5 show the tests for each performance measure. These tests are performed with only the rule combinations that are in the top significance group from the Tukey Tests performed on all of the rule combinations except the combinations that contain SPT as the machine dispatching rule.

Table I.1: ASIC Fab Tukey Test on Average Cycle Time for the Top 6 Rule Combinations

Dispatching Rule	Average Cycle Time (days)	Significance
VFEFS MFIFO	15.647	
VMUS MFIFO	15.76	
VFEFS MCR	15.907	
VEDD MFIFO	16.547	
VSTT MFIFO	16.627	
VFEFS MEDD	17.383	

Table I.2: ASIC Fab Tukey Test on Standard Deviation Cycle Time for the Top 6 Rule Combinations

Dispatching Rule	Standard Deviation of Cycle Time (days)	Significance
VFEFS MCR	5.12	
VFEFS MFIFO	5.1233	
VMUS MFIFO	5.1267	
VSTT MFIFO	5.5533	
VEDD MFIFO	5.5833	
VFEFS MEDD	7.6833	

Table I.3: ASIC Fab Tukey Test on Average WIP for the Top 6 Rule Combinations

Dispatching Rule	Average WIP (lots)	Significance
VFEFS MFIFO	164.58	
VMUS MFIFO	165.95	
VFEFS MCR	167.3	
VEDD MFIFO	174.06	
VSTT MFIFO	174.88	
VFEFS MEDD	182.87	

Table I.4: ASIC Fab Tukey Test on Average X Theoretical Value for the Top 6 Rule Combinations

Dispatching Rule	Average X Theoretical Value	Significance
VFEFS MFIFO	4 72	
VMUS MFIFO	4 7633	
VFEFS MCR	4 7833	
VEDD MFIFO	4 9733	
VSTT MFIFO	5 0067	
VFEFS MEDD	5 0933	

Table I.5: ASIC Fab Tukey Test on the Percentage of On Time Lots for the Top 6 Rule Combinations

Dispatching Rule	Percentage of lots on time	Significance
VEDD MFIFO	99 19	
VMUS MFIFO	98 92	
VSTT MFIFO	98 62	
VFEFS MEDD	97 73	
VFEFS MFIFO	69 28	
VFEFS MCR	62 77	

Appendix J. Make-to-Order Fab Tukey Tests for all Dispatching Rule Combinations

This appendix contains the Tukey Tests performed on the make-to-order fab for the dispatching rule experiment. Tables J.1 – J.5 show the tests for each performance measure. These tests are performed with all of the rule combinations except the combinations that contain STT as the vehicle dispatching rule.

Table J.1: Make-to-Order Fab Tukey Test on Average Cycle Time for all Rule Combinations

Dispatching Rules	Average Cycle Time (days)	Significance
VFEFS MFIFO	18.327	
VMUS MFIFO	18.477	
VCRQ MFIFO	18.487	
VCR MFIFO	18.513	
VFIFO MFIFO	18.657	
VEDD MFIFO	18.67	
VFEFS MEDD	19.44	
VCR MEDD	19.787	
VFEFS MSPT	20.04	
VCRQ MEDD	20.14	
VFIFO MEDD	20.14	
VCR MSPT	20.323	
VEDD MEDD	20.44	
VMUS MEDD	20.61	
VFEFS MCR	20.92	
VFEFS MCRQ	20.92	
VCRQ MSPT	20.997	
VFIFO MSPT	20.997	
VEDD MSPT	21.137	
VCR MCR	24.777	
VCR MCRQ	24.777	
VCRQ MCR	25.11	
VCRQ MCRQ	25.11	
VEDD MCR	26.797	
VEDD MCRQ	26.797	
VFIFO MCR	27.03	
VFIFO MCRQ	27.03	
VMUS MSPT	44.923	
VMUS MCR	48.103	
VMUS MCRQ	48.103	

Table J.2: Make-to-Order Fab Tukey Test on Standard Deviation of Cycle Time for all Rule Combinations

Dispatching Rules	Standard Deviation of Cycle Time (days)	Significance
VCR MFIFO	4.283	
VCRQ MFIFO	4.293	
VFEFS MFIFO	4.32	
VMUS MFIFO	4.337	
VFIFO MFIFO	4.383	
VEDD MFIFO	4.567	
VFEFS MCR	5.05	
VFEFS MCRQ	5.05	
VCR MSPT	5.87	
VCR MEDD	5.917	
VFEFS MSPT	5.937	
VFEFS MEDD	6.023	
VCR MCR	6.167	
VCR MCRQ	6.167	
VCRQ MEDD	6.197	
VFIFO MEDD	6.197	
VCRQ MCR	6.2	
VCRQ MCRQ	6.2	
VMUS MEDD	6.297	
VCRQ MSPT	6.373	
VFIFO MSPT	6.373	
VEDD MEDD	6.377	
VEDD MCR	6.383	
VEDD MCRQ	6.383	
VFIFO MCR	6.467	
VFIFO MCRQ	6.467	
VEDD MSPT	6.817	
VMUS MCR	15.693	
VMUS MCRQ	15.693	
VMUS MSPT	35.117	

Table J.3: Make-to-Order Fab Tukey Test on Average WIP for all Rule Combinations

Dispatching Rules	Average WIP (lots)	Significance
VFEFS MFIFO	244.68	
VMUS MFIFO	246.64	
VCRQ MFIFO	246.79	
VCR MFIFO	247.19	
VFIFO MFIFO	249.12	
VEDD MFIFO	249.3	
VFEFS MEDD	259.48	
VCR MEDD	264.15	
VFEFS MSPT	267.71	
VCRQ MEDD	268.84	
VFIFO MEDD	268.84	
VCR MSPT	271.44	
VEDD MEDD	272.93	
VMUS MEDD	275.18	
VFEFS MCR	279.26	
VFEFS MCRQ	279.26	
VCRQ MSPT	280.39	
VFIFO MSPT	280.39	
VEDD MSPT	282.46	
VCR MCR	330.78	
VCR MCRQ	330.78	
VCRQ MCR	335.21	
VCRQ MCRQ	335.21	
VEDD MCR	357.74	
VEDD MCRQ	357.74	
VFIFO MCR	360.89	
VFIFO MCRQ	360.89	
VMUS MSPT	630.82	
VMUS MCR	658.89	
VMUS MCRQ	658.89	

Table J.4: Make-to-Order Fab Tukey Test on Average X Theoretical Value for all Rule Combinations

Dispatching Rules	Average X Theoretical Value (days)	Significance
VFEFS MFIFO	1.9467	
VCRQ MFIFO	1.9667	
VMUS MFIFO	1.9667	
VCR MFIFO	1.97	
VEDD MFIFO	1.98	
VFIFO MFIFO	1.9867	
VFEFS MEDD	2.0233	
VCR MEDD	2.0633	
VCRQ MEDD	2.1	
VFIFO MEDD	2.1	
VFEFS MSPT	2.1033	
VEDD MEDD	2.1267	
VCR MSPT	2.1367	
VMUS MEDD	2.1467	
VCRQ MSPT	2.2	
VFIFO MSPT	2.2	
VEDD MSPT	2.2033	
VFEFS MCR	2.2267	
VFEFS MCRQ	2.2267	
VCR MCR	2.6333	
VCR MCRQ	2.6333	
VCRQ MCR	2.67	
VCRQ MCRQ	2.67	
VEDD MCR	2.8533	
VEDD MCRQ	2.8533	
VFIFO MCR	2.88	
VFIFO MCRQ	2.88	
VMUS MSPT	4.6933	
VMUS MCR	5.13	
VMUS MCRQ	5.13	

Table J.5: Make-to-Order Fab Tukey Test on the Percentage of On Time Lots for all Rule Combinations

Dispatching Rules	Percentage of on time lots	Significance
VFEFS MFIFO	93.627	
VCRQ MFIFO	93.06	
VCR MFIFO	91.463	
VMUS MFIFO	90.56	
VFIFO MFIFO	85.843	
VEDD MFIFO	85.177	
VFEFS MEDD	52.577	
VFEFS MSPT	49.107	
VCR MSPT	44.407	
VCR MEDD	43.653	
VEDD MSPT	36.573	
VCRQ MEDD	34.633	
VFIFO MEDD	34.633	
VCRQ MSPT	33.093	
VFIFO MSPT	33.093	
VEDD MEDD	31.597	
VMUS MEDD	30.1	
VMUS MSPT	1.373	
VFEFS MCR	0.85	
VFEFS MCRQ	0.85	
VCR MCR	0	
VCR MCRQ	0	
VCRQ MCR	0	
VCRQ MCRQ	0	
VEDD MCR	0	
VEDD MCRQ	0	
VFIFO MCR	0	
VFIFO MCRQ	0	
VMUS MCR	0	
VMUS MCRQ	0	

Appendix K. Make-to-Order Fab Tukey Tests for Top 17 Rule Combinations

This appendix contains the Tukey Tests performed on the make-to-order fab for the dispatching rule experiment. Tables K.1 – K.5 show the tests for each performance measure. These tests are performed with only the rule combinations that are not in the bottom significance group in the percentage of on time lots Tukey Test performed on all rule combinations except those with STT as the vehicle dispatching rule. All of the dispatching rule combinations excluded from these tests have an on time delivery percentage less than 2% and are considered to be poor choices.

Table K.1: Make-to-Order Fab Tukey Test on Average Cycle Time for Top 17 Rule Combinations

Dispatching Rule	Average Cycle Time (days)	Significance
VFEFS MFIFO	18.327	
VMUS MFIFO	18.477	
VCRQ MFIFO	18.487	
VCR MFIFO	18.513	
VFIFO MFIFO	18.657	
VEDD MFIFO	18.67	
VFEFS MEDD	19.44	
VCR MEDD	19.787	
VFEFS MSPT	20.04	
VCRQ MEDD	20.14	
VFIFO MEDD	20.14	
VCR MSPT	20.323	
VEDD MEDD	20.44	
VMUS MEDD	20.61	
VCRQ MSPT	20.997	
VFIFO MSPT	20.997	
VEDD MSPT	21.137	

Table K.2: Make-to-Order Fab Tukey Test on Standard Deviation of Cycle Time for Top 17 Rule Combinations

Dispatching Rule	Standard Deviation of Cycle Time (days)	Significance
VCR MFIFO	4 2833	
VCRQ MFIFO	4 2933	
VFEFS MFIFO	4 32	
VMUS MFIFO	4 3367	
VFIFO MFIFO	4 3833	
VEDD MFIFO	4 5667	
VCR MSPT	5 87	
VCR MEDD	5 9167	
VFEFS MSPT	5 9367	
VFEFS MEDD	6 0233	
VCRQ MEDD	6 1967	
VFIFO MEDD	6 1967	
VMUS MEDD	6 2967	
VCRQ MSPT	6 3733	
VFIFO MSPT	6 3733	
VEDD MEDD	6 3767	
VEDD MSPT	6 8167	

Table K.3: Make-to-Order Fab Tukey Test on Average WIP for Top 17 Rule Combinations

Dispatching Rule	Average WIP (lots)	Significance
VFEFS MFIFO	244 68	
VMUS MFIFO	246 64	
VCRQ MFIFO	246 79	
VCR MFIFO	247 19	
VFIFO MFIFO	249 12	
VEDD MFIFO	249 3	
VFEFS MEDD	259 48	
VCR MEDD	264 15	
VFEFS MSPT	267 71	
VCRQ MEDD	268 84	
VFIFO MEDD	268 84	
VCR MSPT	271 44	
VEDD MEDD	272 93	
VMUS MEDD	275 18	
VCRQ MSPT	280 39	
VFIFO MSPT	280 39	
VEDD MSPT	282 46	

Table K.4: Make-to-Order Fab Tukey Test on Average X Theoretical Value for Top 17 Rule Combinations

Dispatching Rule	Average X Theoretical Value	Significance
VFEFS MFIFO	1 9467	
VCRQ MFIFO	1 9667	
VMUS MFIFO	1 9667	
VCR MFIFO	1 97	
VEDD MFIFO	1 98	
VFIFO MFIFO	1 9867	
VFEFS MEDD	2 0233	
VCR MEDD	2 0633	
VCRQ MEDD	2 1	
VFIFO MEDD	2 1	
VFEFS MSPT	2 1033	
VEDD MEDD	2 1267	
VCR MSPT	2 1367	
VMUS MEDD	2 1467	
VCRQ MSPT	2 2	
VFIFO MSPT	2 2	
VEDD MSPT	2 2033	

Table K.5: Make-to-Order Fab Tukey Test on the Percentage of On Time Lots for Top 17 Rule Combinations

Dispatching Rule	Percentage of on time lots	Significance
VFEFS MFIFO	93 627	
VCRQ MFIFO	93 06	
VCR MFIFO	91 463	
VMUS MFIFO	90 56	
VFIFO MFIFO	85 843	
VEDD MFIFO	85 177	
VFEFS MEDD	52 577	
VFEFS MSPT	49 107	
VCR MSPT	44 407	
VCR MEDD	43 653	
VEDD MSPT	36 573	
VCRQ MEDD	34 633	
VFIFO MEDD	34 633	
VCRQ MSPT	33 093	
VFIFO MSPT	33 093	
VEDD MEDD	31 597	
VMUS MEDD	30 1	

Appendix L. Make-to-Order Fab Tukey Tests for Top 6 Dispatching Rule Combinations

This appendix contains the Tukey Tests performed on the make-to-order fab for the top six rule combinations in the dispatching rule experiment. Tables L.1 – L.5 show the tests for each performance measure. These tests are performed with only the rule combinations that are in the top significance group from the previous Tukey Tests.

Table L.1: Make-to-Order Fab Tukey Test on Average Cycle Time for Top 6 Rule Combinations

Dispatching Rule	Average Cycle Time (days)	Significance
VFEFS MFIFO	18.3267	
VMUS MFIFO	18.4767	
VCRQ MFIFO	18.4867	
VCR MFIFO	18.5133	
VFIFO MFIFO	18.6567	
VEDD MFIFO	18.67	

Table L.2: Make-to-Order Fab Tukey Test on Standard Deviation of Cycle Time for Top 6 Rule Combinations

Dispatching Rule	Standard Deviation of Cycle Time (days)	Significance
VCR MFIFO	4.2833	
VCRQ MFIFO	4.2933	
VFEFS MFIFO	4.32	
VMUS MFIFO	4.3367	
VFIFO MFIFO	4.3833	
VEDD MFIFO	4.5667	

Table L.3: Make-to-Order Fab Tukey Test on Average WIP for Top 6 Rule Combinations

Dispatching Rule	Average WIP (lots)	Significance
VFEFS MFIFO	244.683	
VMUS MFIFO	246.637	
VCRQ MFIFO	246.793	
VCR MFIFO	247.19	
VFIFO MFIFO	249.117	
VEDD MFIFO	249.297	

Table L.4: Make-to-Order Fab Tukey Test on Average X Theoretical Value for Top 6 Rule Combinations

Dispatching Rule	Average X Theoretical Value	Significance
VFEFS MFIFO	1.94667	
VCRQ MFIFO	1.96667	
VMUS MFIFO	1.96667	
VCR MFIFO	1.97	
VEDD MFIFO	1.98	
VFIFO MFIFO	1.98667	

Table L.5: Make-to-Order Fab Tukey Test on the Percentage of On Time Lots for Top 6 Rule Combinations

Dispatching Rule	Percentage of lots on time	Significance
VFEFS MFIFO	93.627	
VCRQ MFIFO	93.06	
VCR MFIFO	91.463	
VMUS MFIFO	90.56	
VFIFO MFIFO	85.843	
VEDD MFIFO	85.177	

Appendix M. Simulation Model Files

This appendix contains the simulation model files used for the experiments in this thesis. The files are included in the attached CDs. Each model attached is under a folder that contains several files that make up the model. Seven models are used for the capacity analysis experiment, which are all included under the "ASIC Fab – Capacity Analysis" folder. Table M.1 provides a list of the capacity analysis models.

Table M.1: Simulation Model Files Attached for Capacity Analysis of ASIC Fab

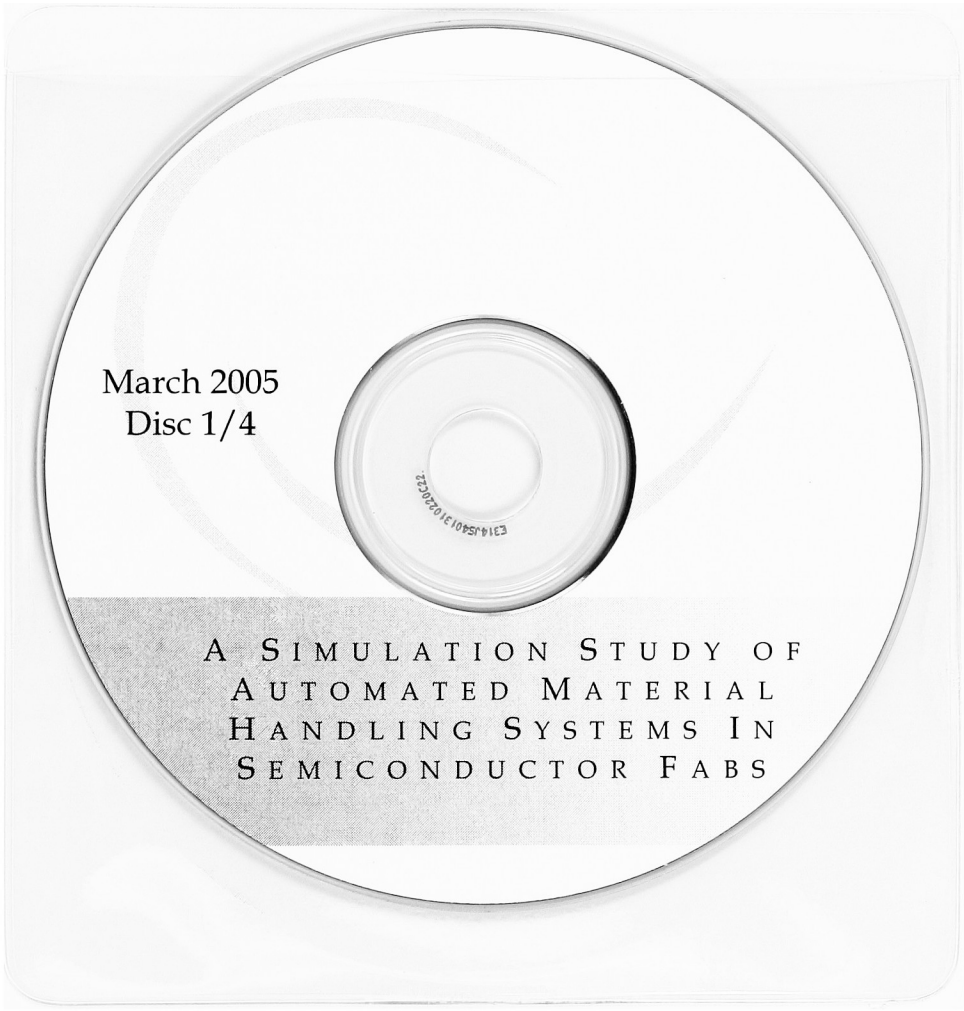
Model Folder Name	Description
ASIC Fab - Capacity Analysis - 50	AMHS bottleneck system run at a release rate of 50%
ASIC Fab - Capacity Analysis - 60	AMHS bottleneck system run at a release rate of 60%
ASIC Fab - Capacity Analysis - 70	AMHS bottleneck system run at a release rate of 70%
ASIC Fab - Capacity Analysis - 80	AMHS bottleneck system run at a release rate of 80%
ASIC Fab - Capacity Analysis - 90	AMHS bottleneck system run at a release rate of 90%
ASIC Fab - Capacity Analysis - 100	AMHS bottleneck system run at a release rate of 100%
ASIC Fab - Capacity Analysis - 110	AMHS bottleneck system run at a release rate of 110%

Seventy models are included for the dispatching rules experiment, 35 for the ASIC fab and 35 for the make-to-order fab. The models for the ASIC fab are in the "ASIC Fab – Dispatching Rules" folder, and the models for the make-to-order fab are in the "Make-to-Order Fab – Dispatching Rules" folder. The models are labeled by the dispatching rules that the fab is set up to run. The same dispatching rule scenarios are included for both fabs. Table M.2 provides a list of the models used for the dispatching rules experiment. Each model in the table is included for both the ASIC and make-to-order fabs.

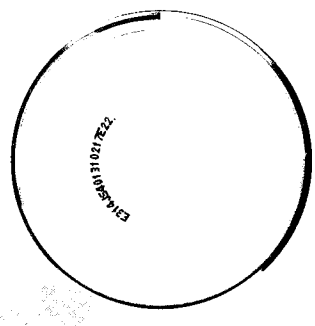
Table M.2: Simulation Model Files Attached for Dispatching Rules Experiment for both ASIC and Make-to-Order Fabs

Model Folder Name	Description
Vehicle CR Machine CR	Fab run with critical ratio for the vehicle dispatching rule and critical ratio for the machine dispatching rule
Vehicle CR Machine CRQ	Fab run with critical ratio for the vehicle dispatching rule and critical ratio including queue time for the machine dispatching rule
Vehicle CR Machine EDD	Fab run with critical ratio for the vehicle dispatching rule and earliest due date for the machine dispatching rule
Vehicle CR Machine FIFO	Fab run with critical ratio for the vehicle dispatching rule and first in first out for the machine dispatching rule
Vehicle CR Machine SPT	Fab run with critical ratio for the vehicle dispatching rule and shortest processing time for the machine dispatching rule
Vehicle CRQ - Machine CR	Fab run with critical ratio including queue time for the vehicle dispatching rule and critical ratio for the machine dispatching rule
Vehicle CRQ - Machine CRQ	Fab run with critical ratio including queue time for the vehicle dispatching rule and critical ratio including queue time for the machine dispatching rule
Vehicle CRQ - Machine EDD	Fab run with critical ratio including queue time for the vehicle dispatching rule and earliest due date for the machine dispatching rule
Vehicle CRQ - Machine FIFO	Fab run with critical ratio including queue time for the vehicle dispatching rule and first in first out for the machine dispatching rule
Vehicle CRQ Machine SPT	Fab run with critical ratio including queue time for the vehicle dispatching rule and shortest processing time for the machine dispatching rule
Vehicle EDD - Machine CR	Fab run with earliest due date for the vehicle dispatching rule and critical ratio for the machine dispatching rule
Vehicle EDD - Machine CRQ	Fab run with earliest due date for the vehicle dispatching rule and critical ratio including queue time for the machine dispatching rule
Vehicle EDD - Machine EDD	Fab run with earliest due date for the vehicle dispatching rule and earliest due date for the machine dispatching rule
Vehicle EDD - Machine FIFO	Fab run with earliest due date for the vehicle dispatching rule and first in first out for the machine dispatching rule
Vehicle EDD - Machine SPT	Fab run with earliest due date for the vehicle dispatching rule and shortest processing time for the machine dispatching rule
Vehicle FEFS - Machine CR	Fab run with first encountered first served for the vehicle dispatching rule and critical ratio for the machine dispatching rule
Vehicle FEFS - Machine CRQ	Fab run with first encountered first served for the vehicle dispatching rule and critical ratio including queue time for the machine dispatching rule
Vehicle FEFS - Machine EDD	Fab run with first encountered first served for the vehicle dispatching rule and earliest due date for the machine dispatching rule
Vehicle FEFS - Machine FIFO	Fab run with first encountered first served for the vehicle dispatching rule and first in first out for the machine dispatching rule
Vehicle FEFS - Machine SPT	Fab run with first encountered first served for the vehicle dispatching rule and shortest processing time for the machine dispatching rule

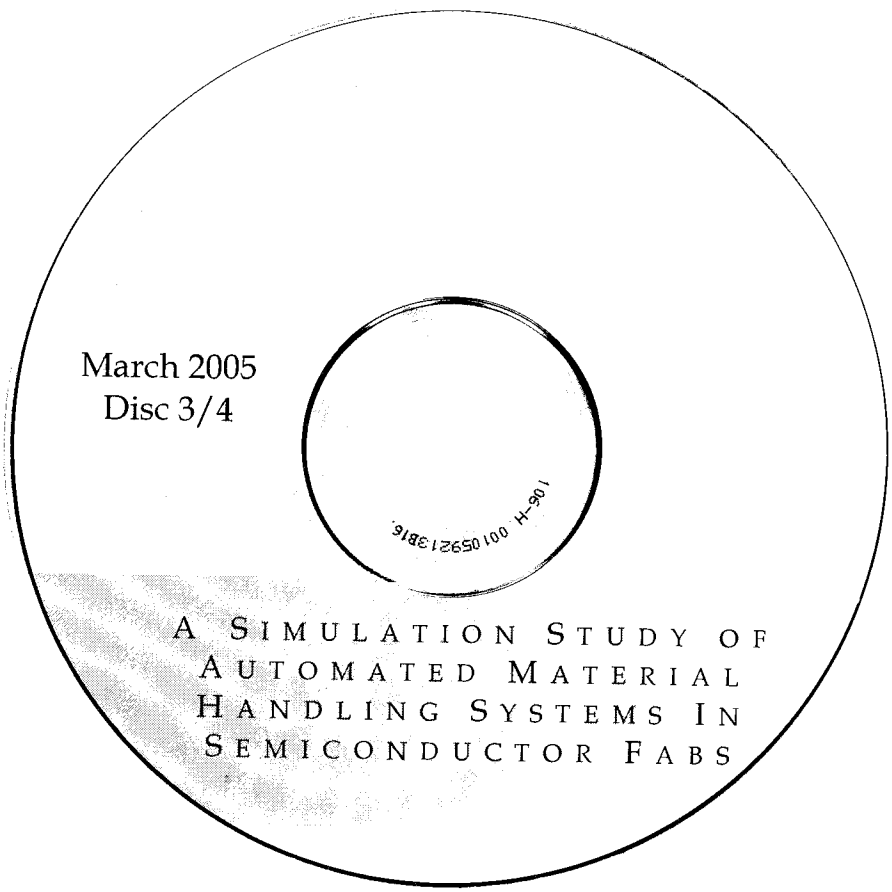
Vehicle FIFO - Machine CR	Fab run with first in first out for the vehicle dispatching rule and critical ratio for the machine dispatching rule
Vehicle FIFO - Machine CRQ	Fab run with first in first out for the vehicle dispatching rule and critical ratio including queue time for the machine dispatching rule
Vehicle FIFO - Machine EDD	Fab run with first in first out for the vehicle dispatching rule and earliest due date for the machine dispatching rule
Vehicle FIFO - Machine FIFO	Fab run with first in first out for the vehicle dispatching rule and first in first out for the machine dispatching rule
Vehicle FIFO - Machine SPT	Fab run with first in first out for the vehicle dispatching rule and shortest processing time for the machine dispatching rule
Vehicle MUS Machine CR	Fab run with most utilized station for the vehicle dispatching rule and critical ratio for the machine dispatching rule
Vehicle MUS Machine CRQ	Fab run with most utilized station for the vehicle dispatching rule and critical ratio including queue time for the machine dispatching rule
Vehicle MUS - Machine EDD	Fab run with most utilized station for the vehicle dispatching rule and earliest due date for the machine dispatching rule
Vehicle MUS - Machine FIFO	Fab run with most utilized station for the vehicle dispatching rule and first in first out for the machine dispatching rule
Vehicle MUS - Machine SPT	Fab run with most utilized station for the vehicle dispatching rule and shortest processing time for the machine dispatching rule
Vehicle STT Machine CR	Fab run with shortest travel time for the vehicle dispatching rule and critical ratio for the machine dispatching rule
Vehicle STT Machine CRQ	Fab run with shortest travel time for the vehicle dispatching rule and critical ratio including queue time for the machine dispatching rule
Vehicle STT Machine EDD	Fab run with shortest travel time for the vehicle dispatching rule and earliest due date for the machine dispatching rule
Vehicle STT Machine FIFO	Fab run with shortest travel time for the vehicle dispatching rule and first in first out for the machine dispatching rule
Vehicle STT Machine SPT	Fab run with shortest travel time for the vehicle dispatching rule and shortest processing time for the machine dispatching rule

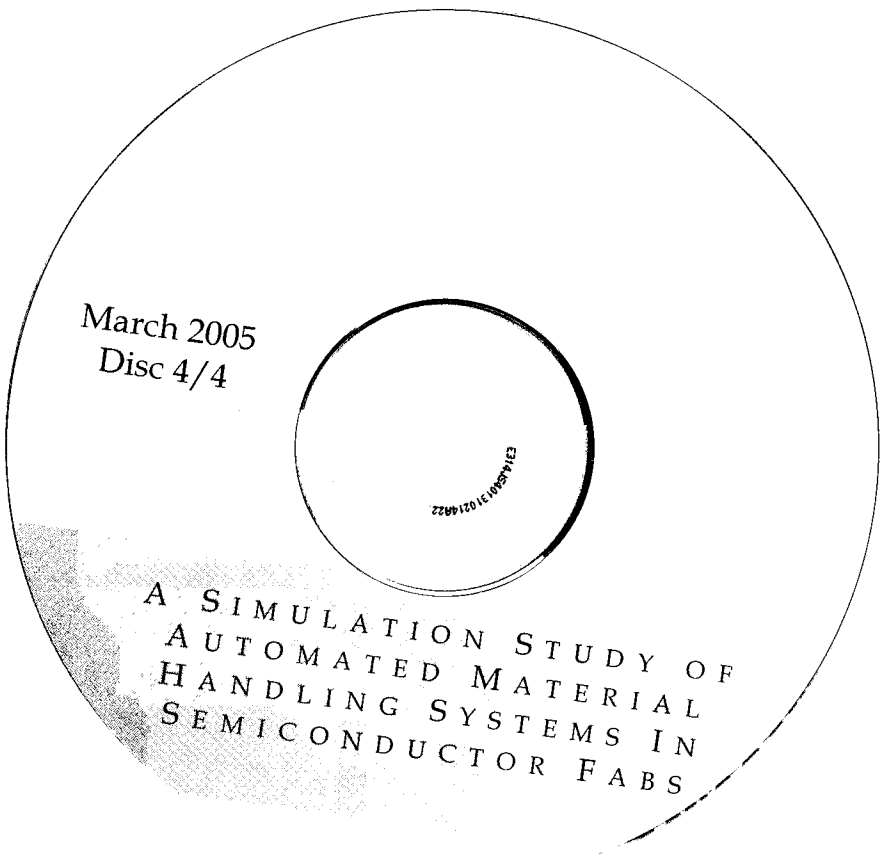

March 2005
Disc 1/4
A SIMULATION STUDY OF
AUTOMATED MATERIAL
HANDLING SYSTEMS IN
SEMICONDUCTOR FABS

March 2005
Disc 2/4



A Simulation Study of Automated Material Handling Systems in Semiconductor Fabs


March 2005
Disc 3/4
A SIMULATION STUDY OF
AUTOMATED MATERIAL
HANDLING SYSTEMS IN
SEMICONDUCTOR FABS


March 2005
Disc 4/4
A SIMULATION STUDY OF
AUTOMATED MATERIAL
HANDLING SYSTEMS IN
SEMICONDUCTOR FABS