Lehigh University
Lehigh Preserve

Theses and Dissertations

1984

An evaluation of microelectronics manufacturing using SLAM II simulation /

Mikel Allan Koschmeder
Lehigh University

Follow this and additional works at: https://preserve.lehigh.edu/etd

Part of the Metallurgy Commons

Recommended Citation
Koschmeder, Mikel Allan, "An evaluation of microelectronics manufacturing using SLAM II simulation /" (1984). *Theses and Dissertations*. 4470.
https://preserve.lehigh.edu/etd/4470

This Thesis is brought to you for free and open access by Lehigh Preserve. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Lehigh Preserve. For more information, please contact preserve@lehigh.edu.

AN EVALUATION OF MICROELECTRONICS MANUFACTURING USING SLAM II SIMULATION

by

Mikel Allan Koschmeder

A Thesis

Presented to the Graduate Committee

of Lehigh University in Candidacy for the Degree of

Master of Science

in

Manufacturing Systems Engineering

Lehigh University

1984

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

12/14/85
(date)

Professor in Charge

Director of M.S.E. Department

Chairman of I.E. Department

TABLE OF CONTENTS

ABSTRACT 1
1.0 INTRODUCTION 3
1.1 THE PROBLEM 6
1.2 RELATED AREAS OF INVESTIGATION 7
1.3 WHAT'S DIFFERENT ABOUT MICROELECTRONICS MANUFACTURING . . 10
1.4 THE PROCESS 12
1.5 APPROACH TO ANALYZING THE PROCESS AND IT'S PROBLEMS . . 12

2.0 EXPERIMENTAL METHODS 16
2.1 MODELING IN GENERAL 16
2.2 CLASSIFICATION OF SIMULATION MODELS . . . 17
2.3 MODELING USING SIMULATION 18
2.4 THE SIMULATION PROCESS 20
2.5 MICROELECTRONICS SIMULATION MODELING PROCESS 24
2.5.1 Problem Definition 24
2.5.2 Model Formulation 25
2.5.3 Data Requirements 28
2.5.4 Computer Program Development . . . 29
2.5.5 SLAM Model Verification 31
2.5.6 SLAM Model Validation 32
2.5.7 Experimental Design 32
2.5.8 Simulation Model Exercising . . . 33
2.5.9 Results Analysis 33
2.5.10 Use of Results 34
2.5.11 Model Updates and Documentation . . 34

3.0 RESULTS 41
3.1 SERVICE TIME DISTRIBUTION 43
3.2 INPUT STARTS AND LINE PRODUCTIVITY . . . 45
3.3 REWORK LEVELS AND LINE PRODUCTIVITY . . . 47
3.4 UNPLANNED MAINTENANCE AND MEAN THROUGHPUT . 48

4.0 DISCUSSION OF RESULTS 57
4.1 SERVICE TIME IMPLICATIONS 57
4.2 INPUT START IMPLICATIONS 58
4.3 REWORK IMPLICATIONS 59
4.4 UNPLANNED MAINTENANCE IMPLICATIONS . . . 60

5.0 CONCLUSIONS AND SUMMARY 62

6.0 FUTURE AREAS OF STUDY 64

I REFERENCES 66

II APPENDICES

APPENDIX A 71
APPENDIX B 87
APPENDIX C 91

LIST OF TABLES

2.1 From-To Chart 35

A2.2 Mean Time Between Failure Assumptions . . . 72

A3.1 Constant Service Times Effect On The Production Parameters 73

A3.2 Exponential Service Times Effect On The Production Parameters 74

A3.3 Normal Service Times Effect On The Production Parameters 75

A3.4 Line Throughput Effect On The Production Parameters (with constant service times) . . 76

A3.5 Sector Resource Increase Effect On The Production Parameters 77

A3.6 1st Expose Rework Effect On The Production Parameters (Etch Rework=0%, 2nd Expose=0% . . 78

A3.7 Etch Rework Effect On The Production Parameters (1st Expose Rework=10%, 2nd Expose Rework=0%). 79

A3.8 Etch Rework Effect On The Production Parameters (1st Expose Rework=20%, 2nd Expose Rework=0%). 80

A3.9 Etch Rework Effect On The Production Parameters (1st Expose Rework=30%, 2nd Expose Rework=0%). 81

A3.10 2nd Expose Effect On The Production Parameters (1st Expose Rework=10%, Etch Rework=5%) . . 82

A3.11 2nd Expose Effect On The Production Parameters (1st Expose Rework=20%, Etch Rework=10%) . . 83

A3.12 2nd Expose Effect On The Production Parameters (1st Expose Rework=30%, Etch Rework=20%) . . 84

A3.13 Mean-Time-To-Repair Effect On The Production Parameters 85

A3.14 Mean-Time-To-Repair Effect On The Production Parameters (continued from A3.13) 86

LIST OF FIGURES

1.1 MOS Manufacturing Processes 14

1.2 Personality Process Flow 15

2.1 Graphical Representation of From-To Chart . . 36

2.2 Personality Process Flow 37

2.3 Process Flow 38

2.4 SLAM Network - Graphical Representation . . . 39

3.1 Cycle Time vs Release Rate 50

3.2 Normal Distribution of Service Times 51

3.3 Cycle Time vs Throughput 52

3.4 Cycle Time vs 1st Expose Rework 53

3.5 Cycle Time vs Etch Rework 54

3.6 Cycle Time vs 2nd Expose Rework 55

3.7 Cycle Time vs MTTR 56

ABSTRACT

Microelectronics manufacturing is a complex set of process interactions which can be described as flow-shop manufacturing with parallel machines. This study focuses on the microelectronics manufacturing process and attempts to clarify the interactions within the process using a SLAM simulation model.

The manufacturing processes modeled in this study consists of the first metal and first insulator layers of the interconnect process. The simulation experiments vary the service time distributions, input starts, rework levels and machine mean-time-to-repair to analyze the interactions with outputs which are measured as average cycle time, throughput and work-in-process.

The average line cycle time and work-in-process both increased when more variable service time distributions were used in the simulation. Increasing the total line start levels increased the throughput of the line until the capacity of the slowest tool was reached. Then, increasing starts only increased the work-in-process at the gating resource. Increasing rework levels also increased the line cycle time. However, rework was much more critical when the line was in a capacity-constrained

condition. The mean time to repair levels had a drastic effect on the variability of the line causing increased work-in-process and longer cycle times.

1.0 INTRODUCTION

Manufacturing productivity has long been a topic of great concern. Traditional productivity measurements and production improvement techniques (cycle time, throughput, line balancing, layout analysis, process flow analysis, etc) have been applied in areas such as assembly, metal cutting, material handling, metal forming, etc. However, little work seems to have been done in the relatively new area of microelectronics manufacturing.

Microelectronics manufacturing is difficult to describe since it is a hybrid of manufacturing types. In some ways microelectronics manufacturing resembles: 1) Batch processing - there are a large number of chips per wafer; 2) Group technology - all wafers have similar but different part numbers; and 3) Programmable automation or Flexible automation - there are many similar but different items (part numbers) similar due to the standardization of design, process, and equipment, but different due to the proliferation of different final part numbers.

In general, microelectronics manufacturing is a job-shop type of manufacturing since it processes each lot of silicon wafers, called a "job", through various machines in a predefined order [11]. The lots vary in size from one wafer to over 100 wafers per job. In addition, each

job may be comprised of wafers which require different processing steps or processing operations.

Typical processing steps of a simplified metal-oxide-semiconductor (MOS) wafer fabrication process are as shown in Figure 1.1 [24] (page 14). These processing operations create individual circuit components formed by impurity layers and the interconnection of these components formed by alternating conducting (metal) and insulating layers. The actual number of total individual processing steps may range anywhere from 100 to 400 steps.

The microelectronics manufacturing process is a combination of both batch and individual wafer processing; that is, the wafers progress through the manufacturing operations either as groups of wafers or else individually as single wafers. The actual processing, whether batch or piece, is completed simultaneously for each chip on a wafer. Since each wafer is populated with 50 to 300 individual chips, as many as 30,000 chips may be undergoing a given processing step at once.

In effect, microelectronics manufacturing is comprised of many different jobs which vary in lot size and wafer types. The processing steps are completed as batch processes, with some individual wafer processing mixed in. In addition, the processing is primarily

completed on automated machines which have relatively long cycle times compared to the human operations of loading, unloading, setup, etc.

Microelectronics manufacturing is affected by many variables within the manufacturing process, not to mention various process parameters such as time, temperature, thickness, electrical measurements, depth, etc. Some of the variables already mentioned are the lot or batch size, the number of process steps, the various interconnection patterns and the amount of process automation. Other major variables affecting the process include the following: rework levels, work-in-process levels, job releases into the line, machine downtimes, priorities assigned to the jobs, yields (percentage of good product made), process layout, operator training and staffing, operation schedule, safety requirements, plant shutdowns and overtime operations.

As can be seen from this short introduction, microelectronics manufacturing is a complex process with many interacting variables. The focus of this study is to analyze the microelectronics manufacturing process and determine some of the cause and effect relationships among interacting variables. The variables for analysis in this study are operation service time distributions, job releases (starts) into the line, rework levels, work-in-

process levels and unplanned machine downtime repairs.

The microelectronics manufacturing analysis begins with the observation that the manufacturing processing steps can be simplified and grouped into "sectors" of detailed steps. These sectors can be further understood using traditional methods of Process Flow analysis and From-To Chart analysis. At this point, a SLAM simulation model of a portion of the process is constructed to analyze the variable interactions. The simulation model is used to analyze the manufacturing system as a whole when changes are made to variables such as releases, reworks, service times, gates and unplanned maintenance. The effects are measured by observing the change in output variables such as cycle time, throughput and work-in-process. As stated, the simulation model is a means to show the cause and effect relationships among the interacting process variables.

1.1 THE PROBLEM

Microelectronics manufacturing as described in the introduction is a very complex and interactive type of manufacturing. The manufacturing complexity is created by the interaction of many variables within each process step. As expected, it is difficult to understand the true interaction among variables when analyzing this dynamic

manufacturing environment. This study focuses on using the "simulation process" to analyze and help understand the problem of cause and effect relationships among variables within the dynamic microelectronics manufacturing environment.

Specifically, the key process variables which will be analyzed are as follows: 1) Operation service time distributions; 2) Release level of starts into the process line; 3) Rework levels within an individual process; and 4) Mean-time-to-repair unplanned machine breakdowns. These variables will be analyzed by observing changes to the process output measurements of cycle time, throughput, and work-in-process.

1.2 RELATED AREAS OF INVESTIGATION

Extensive work has been done in analyzing the productivity of automated flow lines (transfer lines), manual flow lines and the many variations of job-shops. Although these types of manufacturing do not fit the microelectronics manufacturing process exactly, there are results from these works which do seem relevant.

Buzacott [5] determined an approximate solution to production capacity and in-process storage for a flow line or flow process system with two or more stations which produced a single product with no rework loops. The study

assumes exponential station processing times and random breakdowns. Buzacott's solution predicts the effect of adding in-process buffers (temporary product storage) on the utilization (measured as mean cycle time) of the line. The buffer capacity requirement for machine breakdowns was shown to be much larger than what is required for random processing times.

Buzacott's approximate solution considered one machine per stage and no reworks of a single product flow line. Ignall and Silver [14] looked at extending Buzacott's work to a two-stage system with multiple, automatic machines at each stage. They were able to show that increasing the number of machines per stage of the automated production line also increased the size of the storage buffer required for a given level of total output.

Freeman [10] investigated the productivity of automated production lines when interstage buffer storage was added. Freeman showed that line efficiency gains increased as interstage buffer capacity was increased. From his analysis, Freeman also generalized the following: 1) Buffer capacity requirements increased as the variance of the down time distribution increased; 2) The end of the line is more critical than the front of the line and thus requires more buffer storage for a given level of break downs at a stage; and 3) Poor allocation of large buffers

can completely negate their potential efficiency gaining ability.

Hillier and Boling [13] analyzed two, three, and four-stage production lines and found that variable operation times decreased the production rate of a line substantially. They also showed that unbalancing a production line with variable operation times can, in some cases, increase its efficiency. The unbalancing or assigning a lower average operation time to station(s) in the middle of a three or four-stage production line for optimal work allocation is known as the "Bowl Phenomenon."

Buxey, Slack and Wild reviewed manual production flow line system design [4]. They state that unstable (transient) conditions are introduced into the production line when disturbances such as machine breakdowns drastically change the storage buffer levels. Also, these transient conditions cause stations at the beginning of the line to suffer less than stations at the end of the line until steady state buffer levels are reached. Buxey, Slack and Wild recommend computer simulation as the most satisfactory approach to investigating transient conditions.

Gershwin and Berman [12] analyzed a two-stage flow-shop or transfer line production system. They found that as any machine becomes more productive due to increasing

repair and service rates or decreasing failure rates, the total system's production rate increases. Also, increasing buffer size increases the total line production rate to a limit of the production rate of the least productive machine (bottleneck).

Solberg analyzed a flow-shop with variable processing times and showed that the productive capacity of the system asymptotically approaches the capacity of the bottleneck station as the in-process inventory (buffers) is increased [23]. However, the cost of this additional capacity includes the cost of 1) maintaining a high in-process inventory, 2) increases in the average cycle time, 3) blocked servers and 4) general confusion.

A flow-shop production line subject to station breakdowns was modeled as a series of single-server queues by Altiok and Stidham [1]. Their study focused on the allocation of interstage buffer capacities to maximize total profit.

1.3 WHAT'S DIFFERENT ABOUT MICROELECTRONICS MANUFACTURING

Microelectronics can best be described as a job-shop type of manufacturing line where "jobs" of wafers flow through processing machines in a job dependent, predefined order. However, the jobs do flow through basically the

same sets of machines with only a few minor operation changes.

If the process is viewed from a machine set reference (i.e., group of basically identical tools which perform the same manufacturing operations), then the jobs look identical and flow through the same machine sets. Thus, the manufacturing is best considered as a flow-shop type of manufacturing with multiple tools in each machine set. By definition, flow-shops manufacture one basic product which follows the same path through the machines on the manufacturing floor [11].

The purpose of this study is the analysis of production variable interactions for the microelectronics flow-shop manufacturing process. Literature search revealed that analyses have been performed on production systems with only a few stages and then expanded to determine how these systems responded to machine breakdowns, work-in-process and variable processing times. However; analyses of flow-shop production systems with many stages each having multiple machines appears not to have been undertaken.

In summary, microelectronics flow-shop manufacturing utilizes highly automated machines that have long machine cycle times compared to the human operator activities of load, unload, and set-up. As expected, the machine

breakdown and repair rates become very important issues in this equipment-intensive manufacturing environment.

1.4 THE PROCESS

The micorelectronics manufacturing process for program logic arrays (PLA) or gate arrays can be viewed as two different and distinct processes. The first process is called "masterslice" and basically consists of successive photolithography and selective diffusion steps which create various semiconductor components (resistors, capacitors, transistors, diodes, etc.) on the surface of the semiconductor wafer.

The second process is known as "personality" which consists of processing steps that interconnect the discrete components into functioning electrical circuits and give them a unique nature. The personality process flow shown in Figure 1.2 (page 15) depicts the processing of alternating metal (conducting) and insulating layers. This study focuses on the first metal and first insulator layers of the "personality process."

1.5 APPROACH TO ANALYZING THE PROCESS AND IT'S PROBLEMS

The approach used in analyzing line productivity issues in the microelectronics manufacturing environment

was the "simulation process." First, general model building concepts are presented and then the simulation process is defined and explained. Finally, the microelectronics simulation process is presented and followed by in-depth sections covering the simulation results and discussion of the results.

As Mellichamp states [16], "simulation is nothing more than an efficient way of relating output to input." The simulation language used for this analysis was the simulation language for alternative modeling (SLAM) [19]. SLAM is a Fortran based simulation language maintained and distributed by Pritsker and Associates, Inc.

The manufacturing processes analyzed for this study were set up as a network in the SLAM simulation model. The model includes fifteen processing steps and is composed of multiple machines each having unplanned breakdowns, three major rework loops, and constant machine processing times.

Figure 1.1: MOS Manufacturing Processes

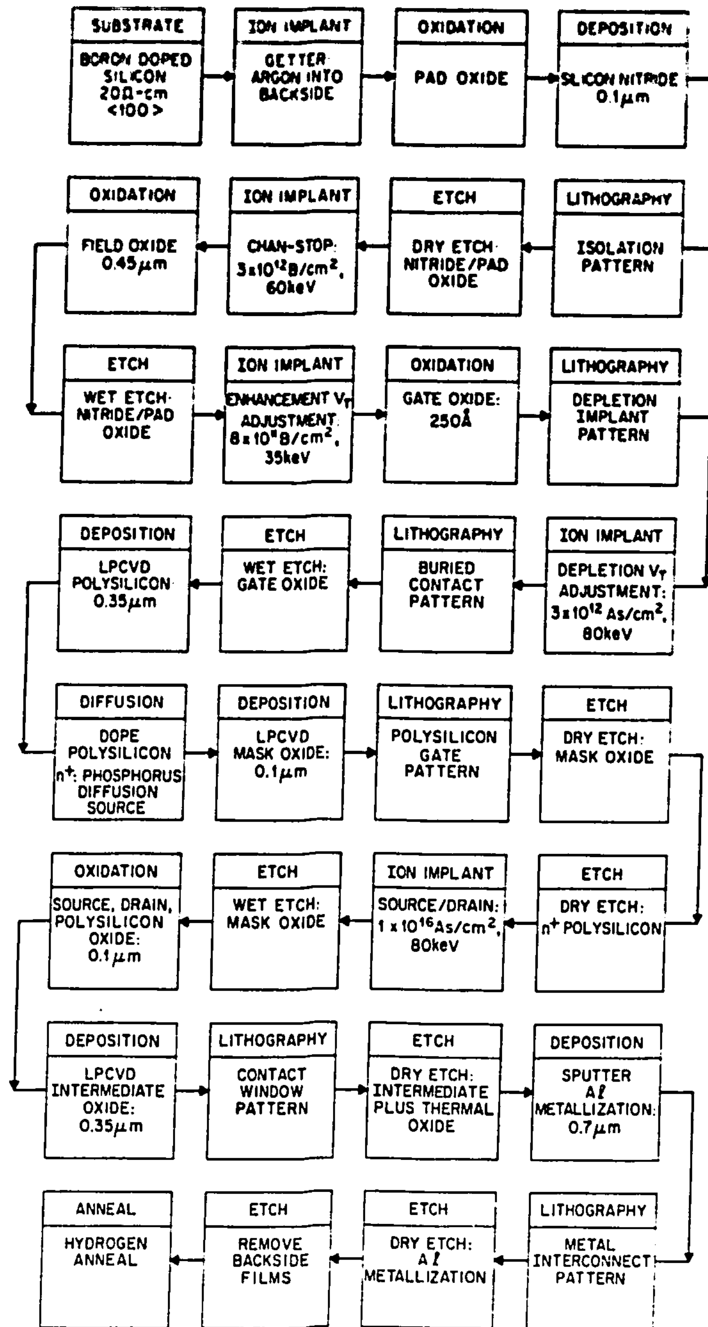
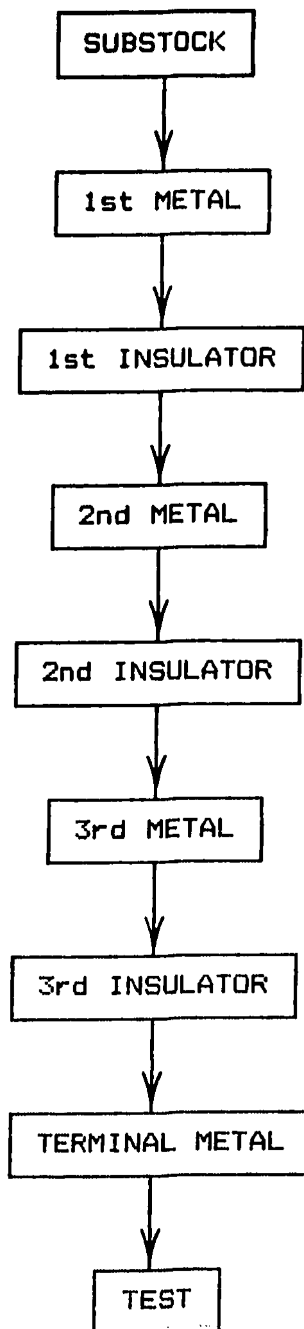


Fig. 1 Main steps in an n-channel, polysilicon-gate, MOS, IC process flow. *(After Siqusch et al., Ref. 7.)*

Figure 1.2: Personality Process Flow



2.0 EXPERIMENTAL METHODS

This section presents the experimental methods used in describing and simulating the microelectronics manufacturing process. First, general model building techniques are discussed, then more specific information used in building the semiconductor simulation is given.

2.1 MODELING IN GENERAL

Models are used in the physical sciences to describe entities or systems. Three types of models [8] used in the sciences are the following: a) physical or prototype models, b) symbolic or mathematical equation models and c) schematic or graphical models. Symbolic and schematic models are most useful for designing a systems simulation. Symbolic or mathematical models can be further classified as analytical or numerical. Analytical models are directly solvable using mathematical equations. Numerical models do not have direct solutions, but can be solved for specific numerical values of the model parameters by using iterative numerical methods; that is, each step in the solution gives a more accurate solution using the results of the previous step [15].

Other distinct types of models are static or dynamic and deterministic or stochastic. Static models are time

independent, while dynamic models change over time. Deterministic models have solutions which are determined by relationships between model variables. Contrarily, stochastic models have random variations in at least some of the model variable relationships.

The type of model used to describe the system is usually based on knowledge of the behavior of the system. Microelectronics processing is dynamic and it is characterized by many random variables (stochastic relationships). Thus, microelectronics processing can best be described by symbolic models.

Complex, large-scale systems such as microelectronics processing are difficult to model for the following reasons: 1) few fundamental laws are available, 2) the procedural elements are difficult to describe and represent, 3) the policy inputs are hard to quantify, 4) random components are significant elements, and 5) human decisions are part of the system [19]. Simulation modeling attempts to overcome these modeling constraints in order to describe complex systems.

2.2 CLASSIFICATION OF SIMULATION MODELS

The two types of classifications for simulation models are discrete and continuous. Discrete models simulate time in a stepwise manner while continuous models

simulate time in a smooth mathematical fashion. Further classifications of discrete models are time orientation and event orientation [17].

The discrete simulation is time-oriented if the clock representing simulated time is updated at regular time intervals (deterministic). Likewise, the discrete simulation is event-oriented if the clock representing simulated time is updated when the scheduled events occur (stochastic).

In summary, simulation models for a process such as microelectronics manufacturing can be classified as numerical, symbolic, deterministic or stochastic or both, dynamic and discrete.

2.3 MODELING USING SIMULATION

As stated previously, simulation modeling is a convenient and efficient way of relating output to input. When a system's inputs are known, a simulation model can be used to predict the system's output. In effect, the system's productivity (outputs divided by inputs using some common measurement) can be determined, analyzed, and compared.

Pritsker [21] states that information extracted from simulation models can be used to understand current

operations, understand and evaluate current productivity, initiate future designs, and formulate arguments for operational changes that can lead to productivity improvements. Pratt [20] identifies three reasons for using simulation modeling: 1) for planning resources, 2) to identify excesses and deficiencies in advance and 3) for comparing performance of alternate systems or arrangements.

Successful applications of simulation models have been in such diversified areas as manufacturing, transportation, communications, health care and the food industry [16]. A simulation model for a manufacturing plant in the Eaton Corporation was used to predict output, highlight obstructions to production and help justify capital equipment purchases [20]. Another example of a manufacturing application of simulation helped to increase the plant's machine utilization [6]. The most substantial benefit of this particular application was the increased understanding of the relative importance of the input resources of the operation. As a final example of simulation models in manufacturing, Nelson [18] used simulation to look at schedule demands during different time periods and then determine resource requirements for planning purposes.

Although significant contributions of simulation can

be cited in many different areas, this paper will focus on the microelectronics manufacturing application.

2.4 THE SIMULATION PROCESS

The simulation process proceeds in a step-like fashion covering the following activities [19]: 1) define the problem, 2) formulate the model, 3) gather required data, 4) develop the computer program, 5) verify the model, 6) validate the model, 7) design the experiments, 8) exercise the model, 9) analyze the model results, 10) use the model results to support management decisions and 11) update and document the model for changes in the system. Each step has important considerations which should not be overlooked.

The problem definition should include the goals and objectives of the simulation. Annino and Russel [3] state that the goal of a simulation project should never be "to model the ...". Modeling is not a goal, it is a means to achieving a goal. A successful simulation should have focused objectives which state what is to be learned about the system under study or what decisions will be based on the simulation results. As the simulation project proceeds, results may lead to more questions which will translate into additional simulation objectives. Therefore, problem definition may continue throughout the

duration of the project.

Before the model can be formulated, the characteristics of the system under study must be understood. Once the system is analyzed and understood, the designer can decide on the amount and level of detail to include in the model. Excessive detail increases the cost of the simulation in terms of computer run time and data collection cost. On the other hand, a broad study might be lower in cost, but may also limit the simulation to a very general model which will not satisfy the specific goals and objectives. As a result, the problem formulation step utilizes the problem definition to help determine the level of detail which is required for the simulation model.

The required input data must be specified and then gathered or assumed from some actual or proposed system. According to Mittra [17], the three types of data are timing, resource utilization and queuing, and historical. Timing data includes service times, system time allocated to various users, etc. Resource and queue data refers to the number of customers, waiting times, queue lengths, etc. Finally, historical data is represented by a chronological event trace of the simulation.

Developing the computer model consists of translating the model into the desired computer programming language.

Annino and Russell [3] state that the programming language should be English-like, self-documenting and readable by the user, who is primarily interested in the system under study, not in computer programming. Although many high-level computer simulation programs exist which can be used for systems modeling, it is not the intent of this paper to analyze all these simulation programs. A list of existing simulation languages would include the following: DYNAMO, GASP IV, GPSS, Q-GERT, SIMON, SIMPL/1, SLAM and others. As one would expect, programs are available which optimize certain applications; therefore, the simulation language should be chosen with a specific application in mind.

Verifying the model is merely the task of insuring that the computer simulation is performing in the desired manner. Essentially, verification is a program debugging step. Once the programmer is confident that the program is operating correctly, verification is complete.

Validation on the other hand is a check or correlation of simulation results with actual system performance. Schruben [22] suggests a validation procedure in which a manager familiar with the system, is presented with a shuffled collection of actual and simulated system outputs. The manager is then asked to identify the genuine documents. Schruben implies that the

model should be modified until the manager cannot identify the genuine documents. A very positive outcome of this method of validation was to increase communications between the users and the modelers.

Once the model has been validated, it is ready to be used as specified by the experimental design. The experimental design states the variables or factors that will be controlled in the simulation runs. The design also describes the degree of variation for each source variable, in order to establish relationships between independent (input) and dependent (output) factors.

After the experimental design is determined, the simulation model is ready to be exercised according to the goals and objectives of the simulation. Exercising the model, allows the modeler to determine the relationships between the system variables and the simulation outputs. Sensitivity-type analyses may also be done to determine how simulation outputs change with slight changes to variable inputs.

As simulation results become available, they should be analyzed to determine relationships between variables and simulation outputs. Statistical methods may be utilized to support the relationships between variables and outputs. An example of one such method is to determine the confidence intervals for the mean value of

variables in the simulation. Results can then be used as supporting evidence for making management decisions, stated in terms of a confidence level.

The final step in the simulation process, model documentation and updating, is easily accomplished, providing the initial model was successful. Model documentation should not be overlooked due to the high probability that system changes will surely require model updates to include new variable relationships within the system.

2.5 MICROELECTRONICS SIMULATION MODELING PROCESS

This section describes the simulation modeling process for the microelectronics manufacturing operation. Each step of the simulation process is uniquely described.

2.5.1 Problem Definition

The goal of the microelectronics manufacturing simulation is to analyze interacting variables within the process to better understand how the total process is affected by changes to the variables. The increased understanding will lead to optimizing the decisions relating to release starts, rework levels and unplanned maintenance repair activities.

2.5.2 Model Formulation

The model formulation was completed after an analysis of the manufacturing process was finished. The process analysis consisted of first determining whether all the processing steps needed to be included in the model or whether aggregate steps, known as sectors, would suffice. An expose sector, for example, might be composed of detailed steps such as: in-gate (record the lot arrival time) the wafer, load a wafer in the exposer, align the wafer, expose the wafer, unload the wafer, inspect the wafer, post bake the wafer, develop the wafer, clean the lot of wafers and out-gate (record the lot completion time) the wafers. Gathering service time, maintenance, equipment and product flow data on this level of detail was not possible due to the time constraints. Since data was available for the aggregated steps or sectors, sector level detail is the degree of detail which will be considered for this study.

Next, a "From-To" Chart was constructed from the process flow sector information. The From-To Chart, Table 2.1 (page 35), basically tallies the number of moves between different sectors for the following purposes [2]: 1) analyzing and visualizing material movement, 2) determining activity locations, 3) showing interdependency

of activities and 4) showing the volume of movement between activities. The activity volume information given in the From-To Chart can also be utilized in plant layout optimization programs.

The From-To Chart was expanded into a graphical representation of the sector moves as shown in Figure 2.1 (page 36). The graphical representation clearly shows the activity interaction volume between various sectors.

Even on a sector basis, the total process would be comprised of nearly 80 sector steps (the total of the row or column totals in the From-To Chart) which would be a very large model. Therefore, a process flow diagram was constructed of only the personality manufacturing processes on an aggregated sector basis as shown in Figure 2.2 (page 37). As can be seen from this diagram, alternate metal and insulator layers make up the final processing steps. In an effort to analyze a critical area of the manufacturing processes without being redundant, the simulation was limited to just one metal and one insulator layer for this analysis.

Finally, a sector process flow diagram was constructed for the the first metal and first insulator layers showing sector work sequence and rework loops. The diagram is shown in Figure 2.3 (page 38).

In addition to determining the level of activity

detail and the portion of the line to analyze, some operating assumptions were made to limit other operation detail. The following set of assumptions were used in conjunction with the sector process flow diagram for the purpose of setting up the simulation model.

1) A tool set is a group of identical tools.

2) A sector is a group of processing steps which utilize a major tool set during the processing time.

3) There is always a supply of wafers to be started at the beginning of the line. This assumption is valid because a sub-stock sector is positioned prior to the first metal operation.

4) There is sufficient space at the end of the line for receiving and storing finished wafers.

5) No adjustment is made to empty or even out the work-in-process at the end of the day, shift or week.

6) Transport time is assumed to be negligible; that is, it is very small and is internal to the waiting time at the next production station.

7) Labor is a relatively small operation cost compared to equipment operating costs. Therefore, the simulation will focus on maximizing the utilization of equipment, not labor. In other words, labor is assumed to be available when required.

8) Planned maintenance downtime will be ignored and

assumed to take place on weekends or off shifts.

9) Each tool in a sector tool set is subject to breakdowns which are random in occurrence and duration. This unplanned maintenance is described as high, medium, or low in occurrence, that is, 25 hours, 50 hours or 125 hours mean-time-between-failures (MTBF) respectively. The mean downtime duration or mean-time-to-repair (MTTR) is assumed to be two hours for all tools. Table A2.2 (page 72) in appendix A shows the MTBF assumptions for the tool sets.

10) The mean-time-to-repair and mean-time-between-failure for all tools are independent random variables described by exponential distributions. That is, MTTR and MTBF are random variables whose randomness in not dependent on previous or future values. Feller [6] and Fox and Zerbe [9] propose reasons for using exponential breakdowns and repairs.

2.5.3 Data Requirements

The types of data required for the microelectronics manufacturing simulation were timing data and resource utilization and queuing data. The following list contains the input data requirements:

1) The order and type of processing steps.

2) The approximate range of processing times.

3) The path of work-flow, including rework loops within the processing steps.

4) The approximate amount of equipment required.

5) Unplanned downtime and failure-rate-distribution data for the processing equipment.

6) The approximate ranges of time for downtime and failure rates.

7) The approximate volume of manufacturing per day. This volume is stated in terms of lots released per day or releases per day.

Most of the required input data was available from the actual manufacturing process modeled. However, the model data had to be fictitious for the purpose of maintaining confidentiality.

2.5.4 Computer Program Development

The simulation language selection for this study was based on the simulation feature requirements and the availablity of simulation models at Lehigh University. Microelectronics processing is basically a shop-flow process over multiple machines which makes a process and/or discrete event model desirable.

The SLAM simulation modeling language was selected based on its availability and its ability to handle the

event-oriented nature of semiconductor manufacturing. SLAM's network structure and pictorial representation allow easy translation of process model to computer code input.

The SLAM model for the microelectronics manufacturing process was constructed from the process flow diagram. Since the SLAM model is a graphical representation of the actual process, it was relatively easy to convert the process flow diagram into the SLAM graphical network model, Figure 2.4 (pages 39 and 40). The graphical model was then converted into a computer program as shown in appendix B. The SLAM symbols used in the graphical model are defined in appendix C.

The following assumptions were used in the SLAM model for simulating the microelectronics manufacturing:

1) Each of the parallel service channels consists of a work center which contains one machine to complete the associated service activity.

2) Each service channel has its own queue in which jobs are served on a first-in-first-out (FIFO) basis except rework jobs which have higher priority.

3) Servers are treated as resources.

4) The initial starting conditions are start empty and idle, then truncate initial statistics after 200 hours of production (200 hours is a result of analyzing output

from the model).

5) Simulation run length is 3000 hours of production based on stopping rules given in [19].

2.5.5 SLAM Model Verification

The SLAM model verification was done in two steps. The first step was to get the program running without errors and the second step was to get the program running as designed.

Program coding errors were not difficult to detect due to the SLAM error message routine. Fixing these types of errors is basically a trial and error task once the code error is flagged by SLAM and recognized by the programmer.

The major error problems encountered were with the SLAM main program system defaults. SLAM allows only a limited number of entities (things, pieces, etc., or jobs in this analysis) and attributes of these entities in the system at once. The defaults are very minimal so any medium-sized, realistic model would require a revised main program which sets these defaults to higher values. However, increasing the total simulation entities also increases memory requirements, account size and CPU times.

Once the program was running error free, attention

could be focused on obtaining quality output. Checking for correct output was accomplished by logically following entities through the process and comparing these results with the actual program output. The calculations for the correct release of entities into the system and the correct total entities through activities and rework loops were mathematical checks. Program output resulting from assumed distribution data, such as the number of unplanned equipment downtime failures, was assumed to be correct.

2.5.6 SLAM Model Validation

The program validation was done by comparing the SLAM output to the known actual line performance. However, since fictitious numbers were used in the simulation, the validation process could really only be partially successful; that is, the model might be slightly inaccurate, but not by orders of magnitude, and thus can be used for attaining this simulation's goals.

2.5.7 Experimental Design

The simulation experiments for the microelectronics analysis were designed to investigate four key variables: 1) operation service time distributions, 2) release level of starts into the process, 3) rework levels within the

process and 4) mean-time-to-repair unplanned machine breakdowns.

Each variable is analyzed separately while holding the other variables constant. The simulation experiments change the variable under analysis and then determine the change in system response in terms of average cycle times, average queue lengths and total throughputs.

2.5.8 Simulation Model Exercising

The simulation model was exercised over a period of approximately four weeks. The model input was edited on an IBM PC-XT personal computer, transmitted over telephone lines via Kermit protocol to the DECSYSTEM-20 computer, and then run on that mid-size computer. Each simulation run took about two minutes of computer CPU time for completion. After completion, the results were transmitted back to the personal computer via telephone lines and printed at the personal computer.

2.5.9 Results Analysis

The results of the microelectronics simulation are presented in the next section of this study. In addition to presenting the results, a discussion and analysis of the results is also presented in a later section.

2.5.10 Use of Results

The results of the microelectronics simulation will be presented to operating management of the actual line modeled. It is desired that the results of this simulation will aid in the understanding and control of some of the operating characteristics of microelectronics manufacturing.

2.5.11 Model Updates and Documentation

Programming updates to the microelectronics model in this study have already been considered. However, due to the time constraint, the model will not be updated for this analysis. After making multiple simulation runs and becoming more familiar with the SLAM simulation language, there are some additional features which would make this model easier to use. One such update would be to create global variables for the input releases, the rework volume percentages and the MTTR.

The microelectronics simulation model is well documented due to the process flow analysis, the SLAM graphical description and the SLAM program given in this study. Any program updates to the model, as well as updates due to the process changing, should be documented if the model is to be understood in the future.

Table 2.1: From-To Chart

FROM \ TO	Diffusion	RIE	Apply	Expose	Ion Implant	LPCVD	EPI	Plat. Evap.	Wet Stat.	Test	Expose	Apply	RIE	Metals	Pln. Qtz.	Comp. Ins.	RIE Via	Lift Off	Wet Stat.	Std. Qtz.	Test	Sinter	TOTALS
Diffusion			4			2	1		1														8
RIE	3																						3
Apply				7																			7
Expose									7														7
Ion Implant			1						1														2
LPCVD			2																				2
Epi	1																						1
Plat. Evap.									1														1
Wet Stat.	2	3			2			1		3													11
Test									2			1											3
Expose																			6				6
Apply											6												6
RIE														2									2
Metals												1						2					3
Pln. Qtz.																1							1
Comp. Ins.																			1				1
RIE Via																			1				1
Lift Off															1				1				2
Wet Stat.												2	2	1			1			1		1	8
Std. Qtz.												1										1	2
Test																				1			1
Sinter												1									1		2
TOTALS	6	3	7	7	2	2	1	1	12	3	6	6	2	3	1	1	1	2	9	2	1	2	79

Table 2.1: From-To Chart

FROM \ TO	Diffusion	RIE	Apply	Expose	Ion Implant	LPCVD	EPI	Plat. Evap.	Wet Stat.	Test	Expose	Apply	RIE	Metals	Pln. Qtz.	Comp. Ins.	RIE Via	Lift Off	Wet Stat.	Std. Qtz.	Test	Sinter	TOTALS
Diffusion			4			2	1		1														8
RIE	3																						3
Apply				7																			7
Expose									7														7
Ion Implant			1						1														2
LPCVD			2																				2
Epi	1																						1
Plat. Evap.									1														1
Wet Stat.	2	3			2			1		3													11
Test									2			1											3
Expose																			6				6
Apply											6												6
RIE														2									2
Metals												1						2					3
Pln. Qtz.																1							1
Comp. Ins.																			1				1
RIE Via																			1				1
Lift Off															1				1				2
Wet Stat.												2	2	1			1			1		1	8
Std. Qtz.												1										1	2
Test																				1			1
Sinter												1									1		2
TOTALS	6	3	7	7	2	2	1	1	12	3	6	6	2	3	1	1	1	2	9	2	1	2	79

Figure 2.1: Graphical Representation of From-To Chart

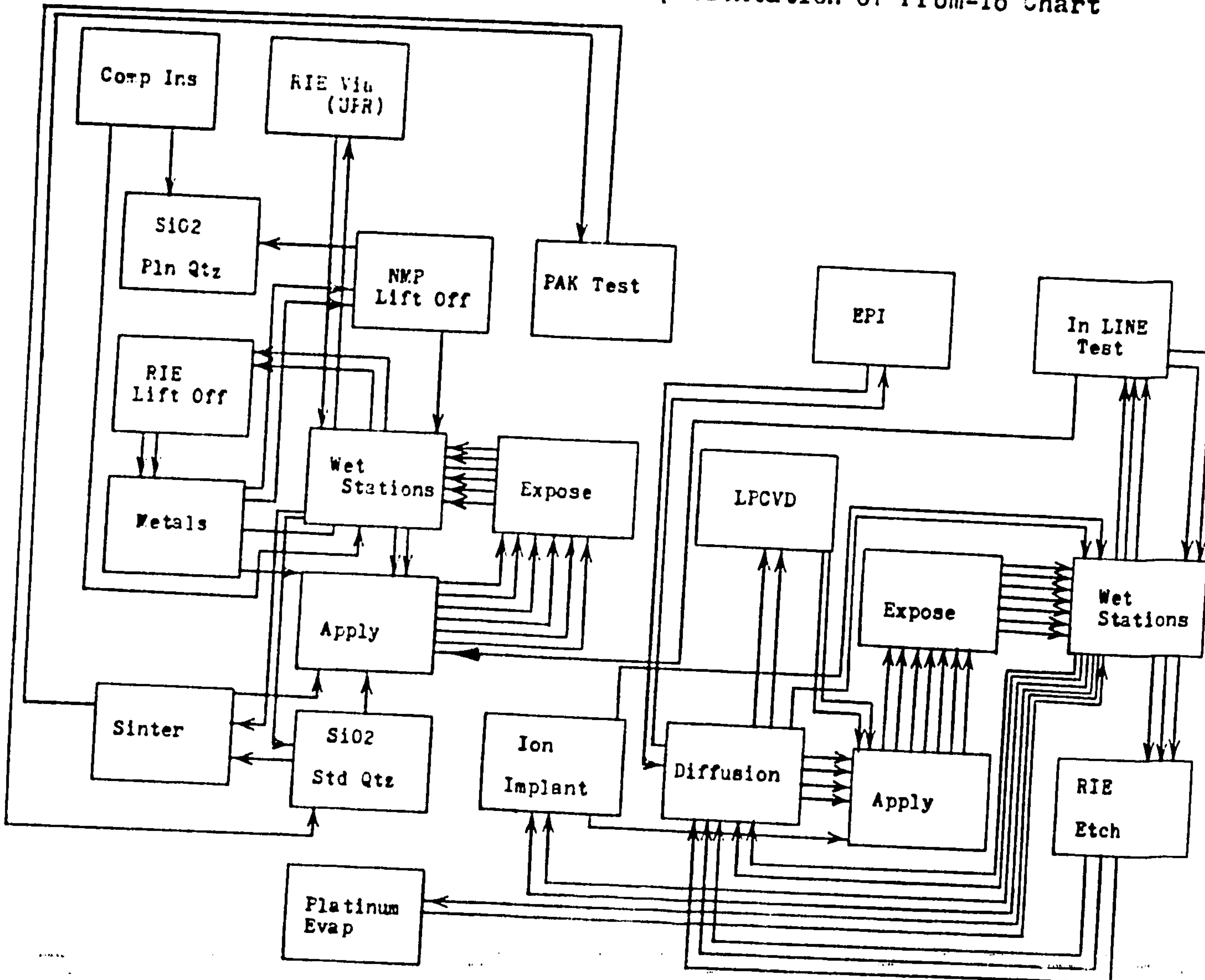


Figure 2.2: Personality Process Flow

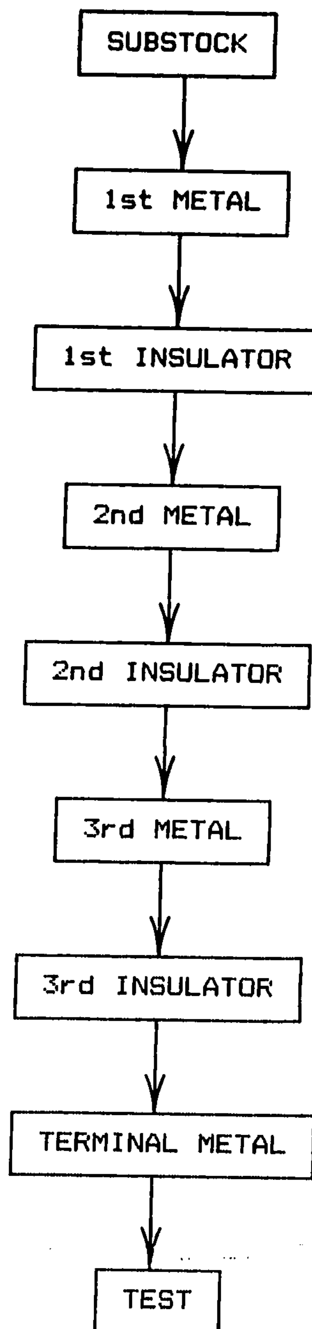


Figure 2.3: Process Flow

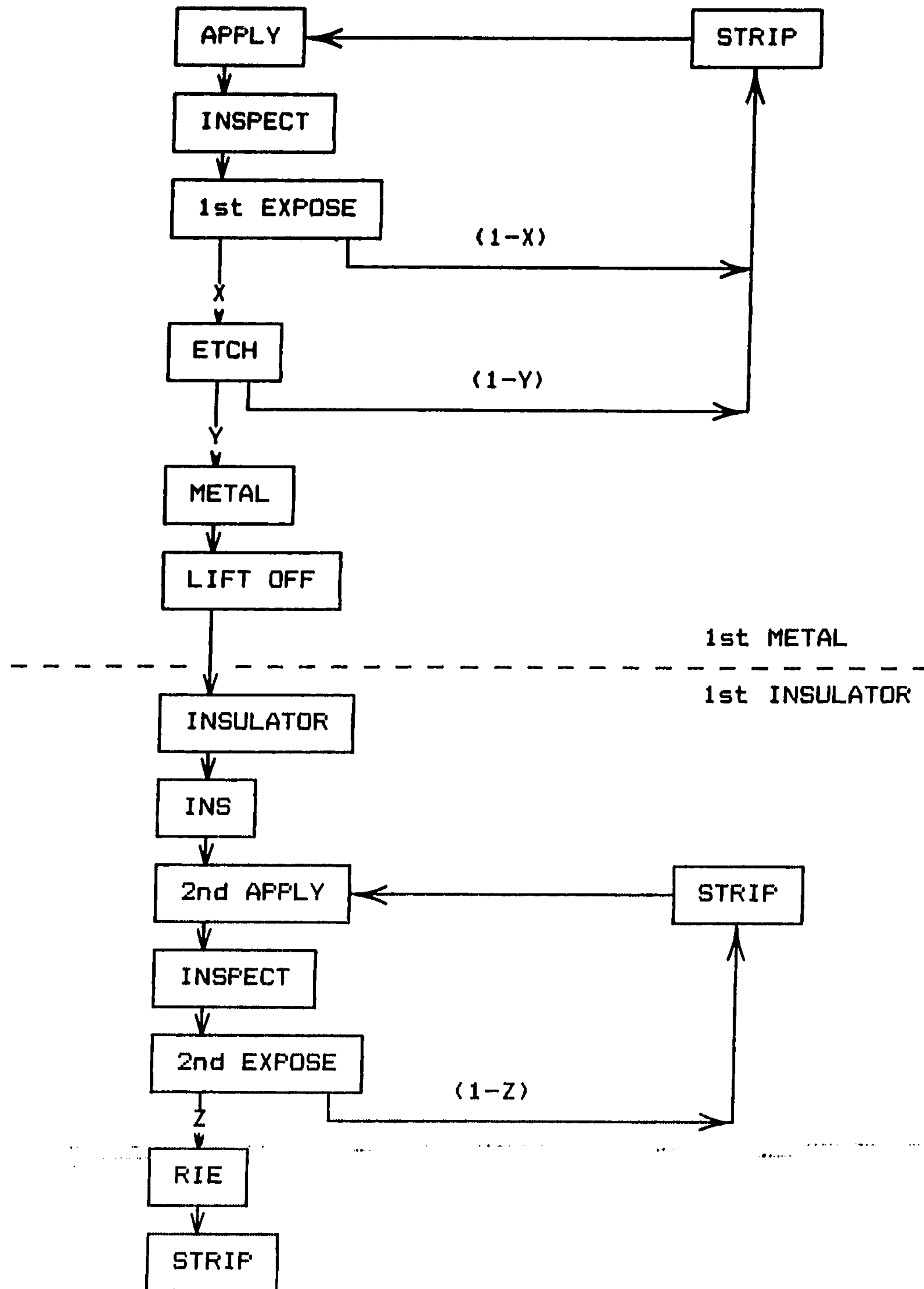


Figure 2.4: SLAM Network - Graphical Representation

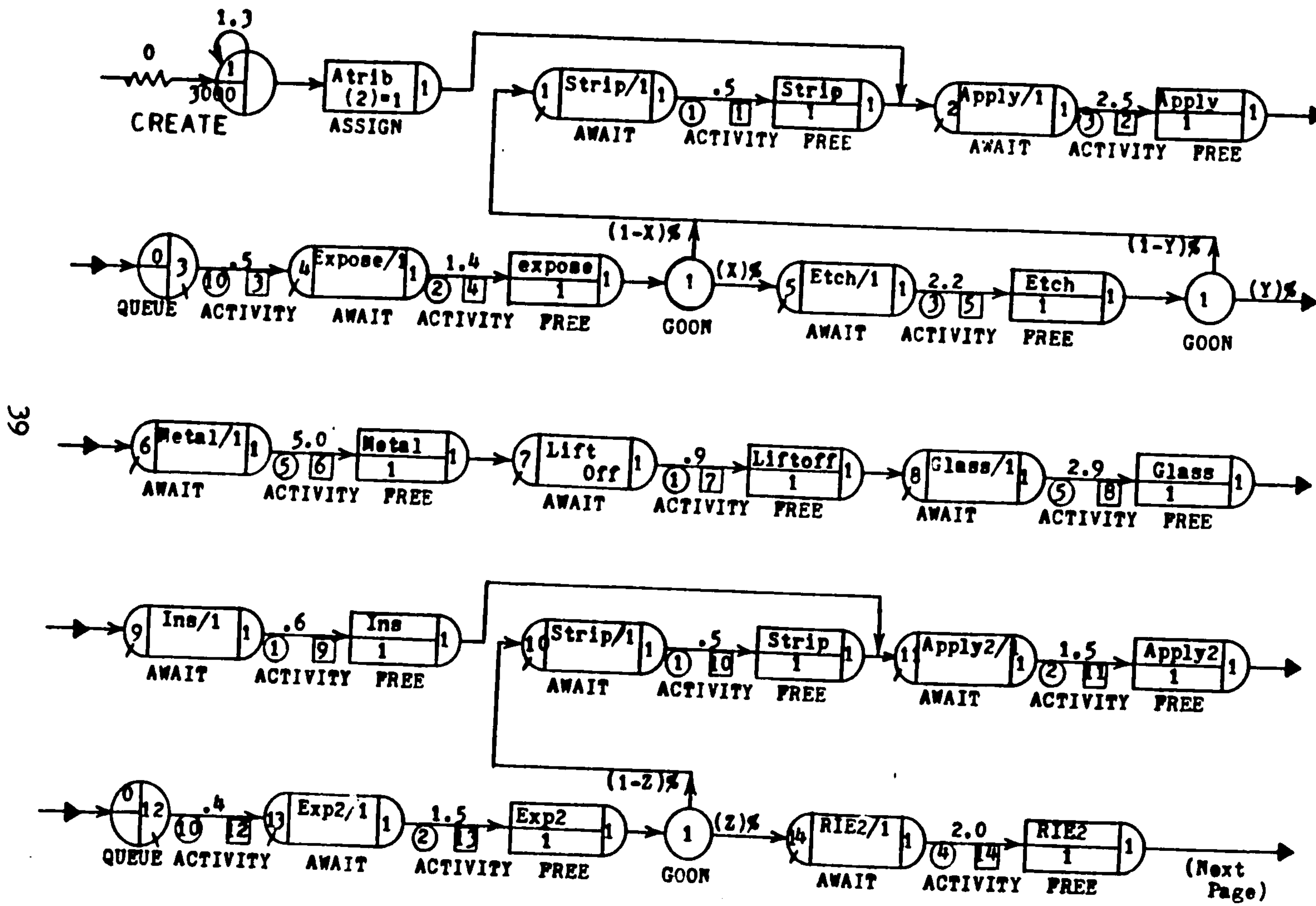
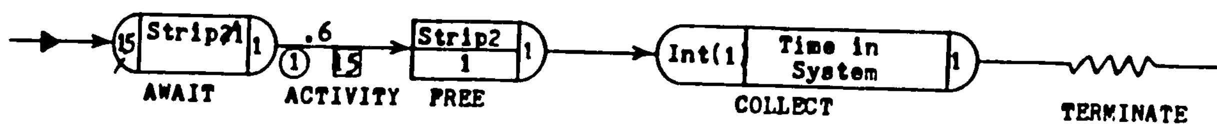
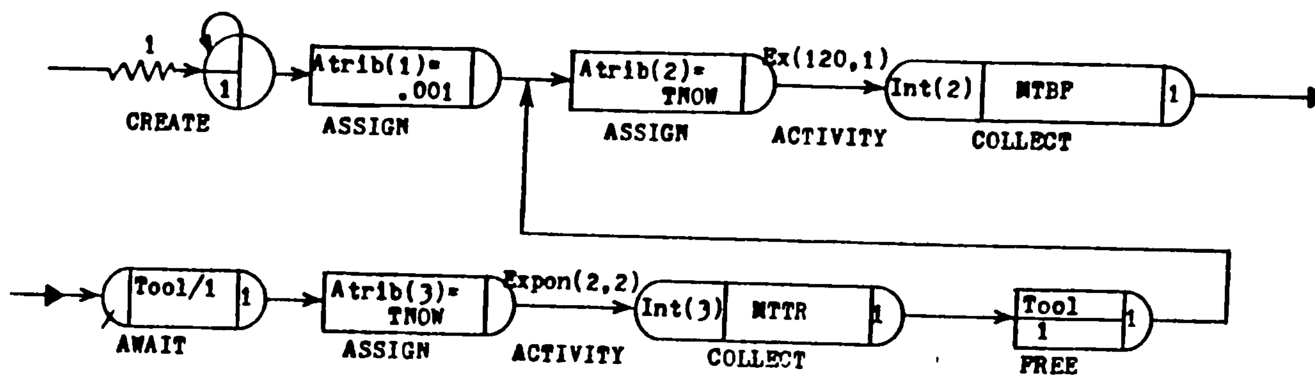


Figure 2.4: SLAM Network - Graphical Representation (cont.)



MTBF and MTTR Network



3.0 RESULTS

This section presents the results of the microelectronics simulation study. The SLAM simulation model for the microelectronics processing was run on Lehigh University's DECSYSTEM-20 computer. The SLAM model was then exercised over a time period of approximately four weeks.

The initial simulation runs were used for verification and validation. Once the simulation model had been verified and validated, it was ready to be exercised according to the objective of learning more about the microelectronics process parameter interactions.

The modeled production line performance measurements were computed by the SLAM simulation program. The output measurements used for comparing various interactions included: 1) average cycle time - the average time jobs spent from entry to exit, 2) throughput - the total number of jobs that were processed until finished and 3) average queue length - the average number of jobs waiting for service at each particular sector. The performance measurements are plotted as graphs at the end of this section and are also listed in tabular format in appendix A.

The first set of actual simulation runs was made to substantiate the argument for constant processing

(service) times. Simulation runs were made with constant service times, exponentially distributed service times and normally distributed service times.

The next set of simulation runs analyzed the input start affect on the production output. The production output was measured in terms of throughput (the number of completed jobs) and the average cycle time (mean time jobs spent in the line). In addition, the input wafer starts were increased beyond the line's capacity limit, to investigate the idea of gating tool production capacity (bottleneck capacity).

The simulation model was also used to determine how line rework levels affected the average line cycle time. Rework levels were varied, holding the line input starts constant, in multiple simulation runs to show the effect on average cycle time. As expected, increasing rework increased the average cycle time of jobs released into the process.

Another set of simulation runs was completed to show the interaction of unplanned machine maintenance mean-time-to-repair with the line productivity. Increasing the unplanned maintenance MTTR had the effect of adding variability to the line, which caused longer average cycle times.

3.1 SERVICE TIME DISTRIBUTION

Simulation runs made with various service time distributions resulted in changing the line variability thus changing the line's efficiency. The constant service times added the least to line variability, measured in terms of average sector queue lengths. Consequently, as more variable service time distributions were used, the line variability increased.

The next distributions of service times used in this simulation were exponential distributions with means equal to the raw processing time of the activities. The exponential distribution implies a higher degree of variability than what probably exists in most production line operations [13]. However, the general effect of using variable service time distributions will be indicated, even if results are exaggerated due to the exponential distribution.

Figure 3.1 (page 50), extracted from Tables A3.1 and A3.2 (pages 73 and 74), compares the average cycle time for the constant and exponential service distributions at various release rates. The constant and exponential service distributions indicate the lower and upper bounds of the variability for an optimally designed production line. As can be seen, the exponential service times added

large amounts of variability to the line. The variability caused excessive work-in-process at each activity, which increased the average queue lengths and cycle time of the processing line.

Lastly, normal distributions were used with means equal to the constant service times or raw processing times and standard deviations equal to 10%, 20%, 40% and 50% of the mean. Increasing the service time variability (larger standard deviation) caused increased line variability resulting in longer average sector queues and increased average line cycle times.

Figure 3.2 (page 51) and Table A3.3 (page 75) show how the average cycle time increased as the standard deviation of the normal distribution of service times was increased. The smallest standard deviation, 10% of the mean, was used to check for small variations in the processing times. The standard deviation value of 20% of the mean was used to represent the limit of how this process could vary in processing times. Finally, the standard deviation values of 40% and 50% of the mean were used to determine the influence of unreasonable variation which would probably not be seen in this process. However, these high standard deviations were of interest for viewing extremes.

The resulting average line variability for the

constant, normal, and exponential service time distributions was increased respectively. However, the difference between the constant service times and the normal service time distribution with small standard deviations (the most realistic cases) had a minimal affect on the average cycle time for the line. Thus, the proposal for using constant service times for the microelectronics simulation appears to be reasonable.

3.2 INPUT STARTS AND LINE PRODUCTIVITY

The effect of increasing the line input starts was measured on the basis of average cycle time and total line throughput. The input starts were increased from a capacity-unconstrained condition to a capacity-constrained condition; the capacity became constrained at the release rate of approximately .8 jobs per hour. The resulting line degradation was observed at the gating tool (sector 2) where work-in-process built up at an increasing rate as starts were raised above the gating tool's capacity.

As shown in Figure 3.3 (page 52) and Table A3.4 (page 76), total throughput increased as starts increased, until a point where work-in-process began building in front of the gating sector. At this point, increasing the input starts only served to increase the gating sector's work-in-process. As a result, sectors in front of the gating

sector were operating faster than the gating sector. Accordingly, sectors following the gating sector were limited to a maximum operating rate of the gating sector's output rate. The average queue lengths increased as the release rate was increased, until sector 2 reached capacity. When sector 2 reached capacity, the work-in-process built up and continued to build in front of that sector.

The effect of adding an additional gating sector tool resource was also considered. When an additional simulation run was made with an extra gating sector tool resource, the result was either of the following: 1) a shift of the work-in-process to the next minimum capacity or gating tool, or 2) a shift to a capacity-unconstrained condition where only normal work-in-process built up in front of the work sectors.

Specifically, a sector 2 tool resource was added when the release rate was one job per hour and a huge queue had, on previous runs, built up in front of sector 2. The additional resource at sector 2 caused that sector to become capacity-unconstrained. However, sector 4 then became the gating sector and a queue of jobs built up in front of that sector. Table A3.5 (page 77) shows the comparison of average queue lengths for the capacity-constrained condition at sector 2 and sector 4 after the

additional resource was added.

3.3 REWORK LEVELS AND LINE PRODUCTIVITY

The rework levels for the three major rework loops were varied to see the effect on the line's average cycle time and throughput. First, the 1st expose rework was increased from 7.5 to 20% with the other rework loops held at 0% rework. This resulted in increasing average line cycle time and decreasing the throughput. As Figure 3.4 (page 53) and Table A3.6 (page 78) show, the cycle time did not change too drastically, because the capacity was such that it did not become constrained.

Next, the etch rework was increased from 2.5 to 10% while holding the 1st expose rework at 10% and the 2nd expose rework at 0%. Since the capacity was not constrained, the effects were a slight increase in cycle time and relatively no change in the line throughput. Figure 3.5 (page 54) and Tables A3.7, A3.8 and A3.9 (pages 79-81) show the cycle time increases due to increasing rework levels for the etch sector.

The etch rework was also increased from 10 to 20% and from 0 to 20% while holding the 1st expose rework at 20% then 30% and the 2nd expose rework at 0% and 0% resspectively. In these cases, the capacity became constrained and queues formed at the gating sector. As

the capacity became constrained, the cycle times increased drastically and throughputs decreased accordingly.

Finally, the 2nd expose rework was increased from 7.5 to 20% and from 5 to 20% while the 1st expose rework was held at 10% and 20% and etch rework was held at 5% and 10% respectively. These cases represented capacity-unconstrained conditions which resulted in slightly increased cycle times and slightly decreased throughputs. The increased cycle time due to 2nd expose rework can be seen in Figure 3.6 (page 55) and Tables A3.10, A3.11 and A3.12 (pages 82-84).

The 2nd expose rework was also increased from 10 to 20% while holding the 1st expose rework at 30% and the etch rework at 20%. This case represented a capacity-constrained ("gated") condition which had a high average cycle time and decreased throughputs.

The effect of increasing rework in any of the rework loops increased the total cycle time of the line. In addition, the increased cycle time decreased throughputs in most cases and caused capacity-constrained conditions in some of the cases.

3.4 UNPLANNED MAINTENANCE AND MEAN THROUGH PUTS

The total cycle time for the microelectronics

distributions and a variable mean for the MTTR exponential distributions. The MTTR was varied from one hour to nine hours which resulted in adding a huge variability to the work-in-process at each sector queue. Figure 3.7 (page 56) and Tables A3.13 and A3.14 (pages 85 and 86) show how the increased work-in-process variability increased the average cycle time of the line which resulted in decreased total throughput.

Specifically, increases of one and two hours in the MTTR for the tools in each sector had a drastic affect on the performance of the line. Referring to Figure 3.7, MTTR of 1 to 2.5 hours minimally affects the cycle time. However, MTTR's of 3 hours and higher really begin to degrade the line cycle time due to a capacity-constrained condition forming at sector 2. The added variability was immense compared to the seemingly small amount of increase in the MTTR.

Figure 3.1

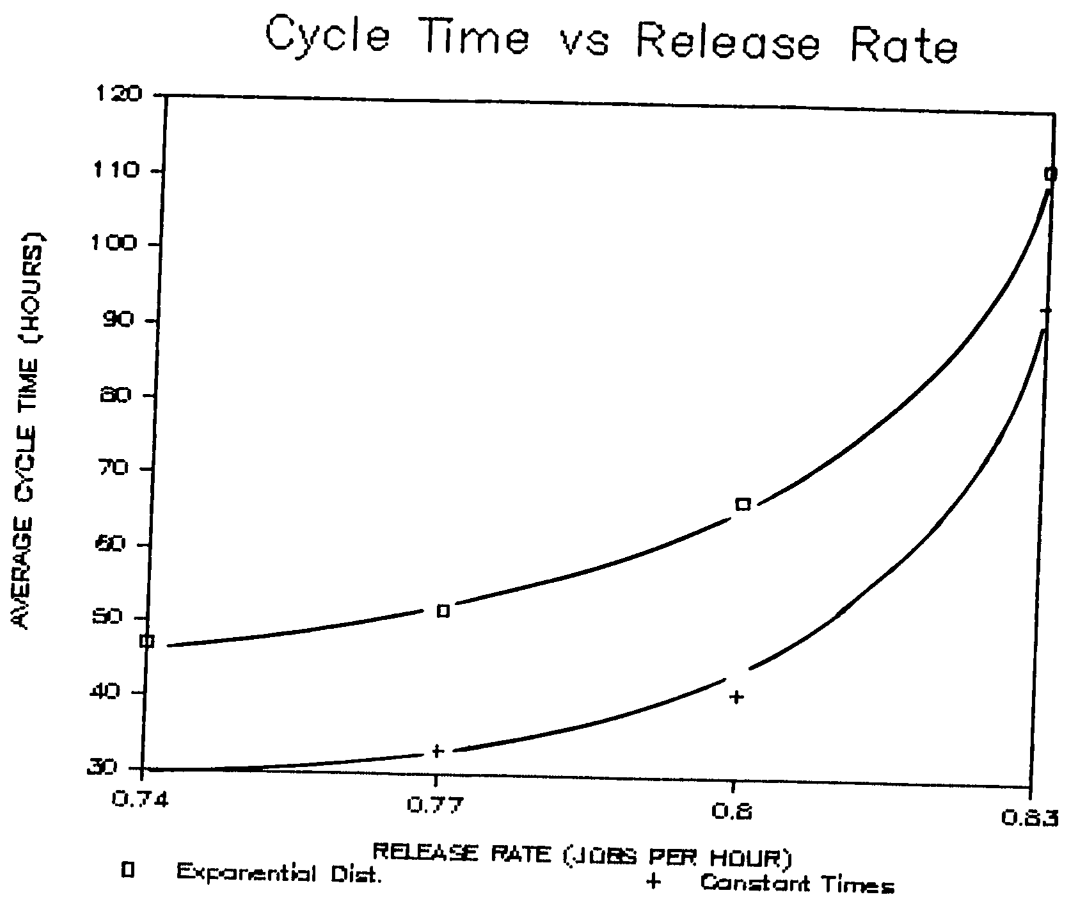


Figure 3.1

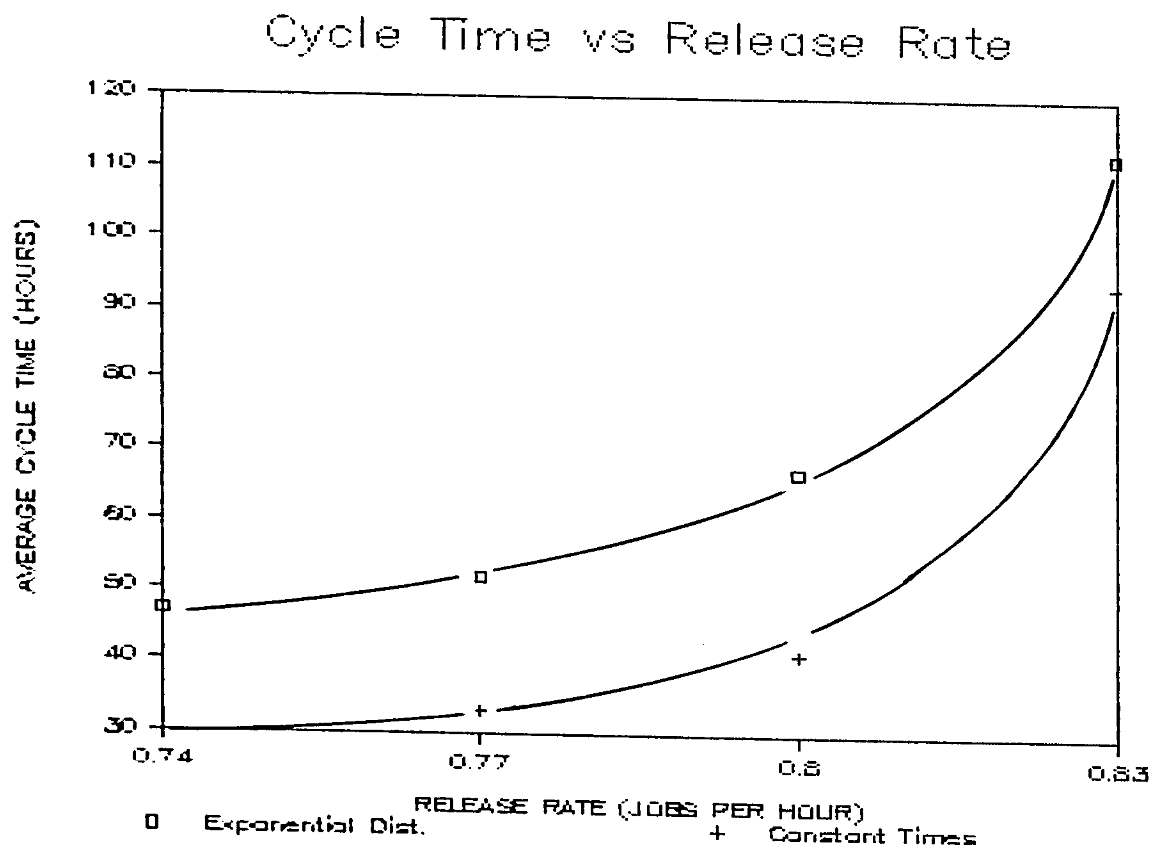


Figure 3.2

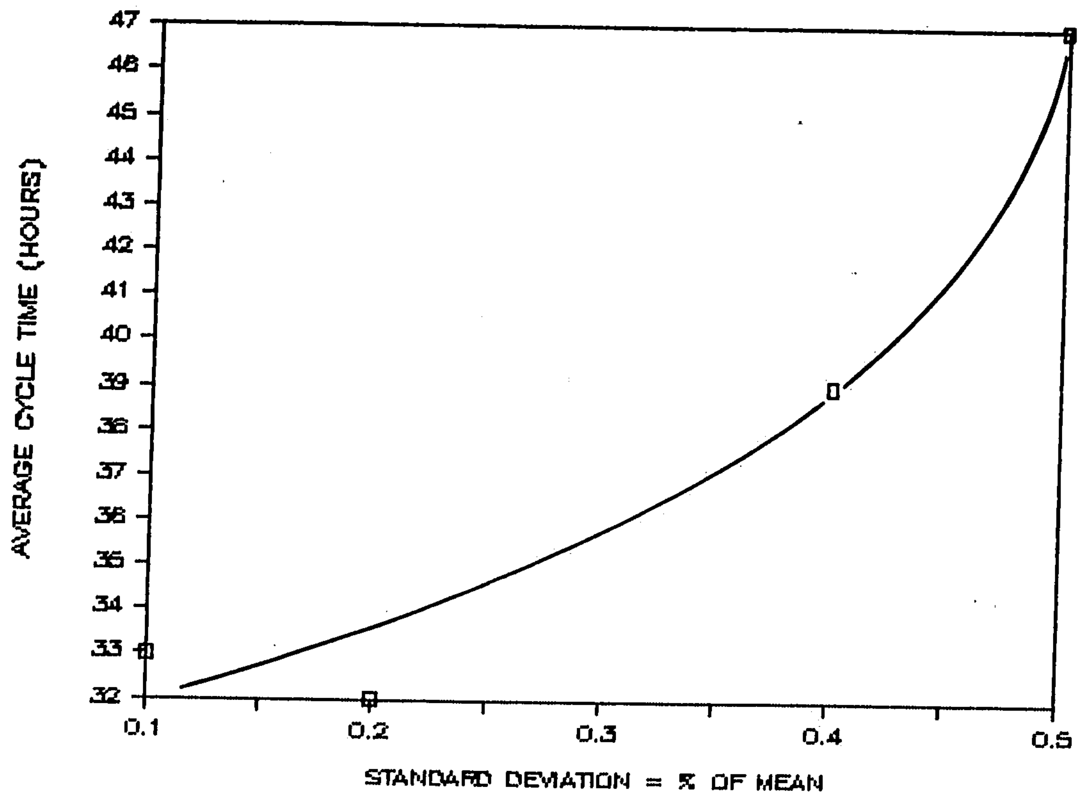


Figure 3.2

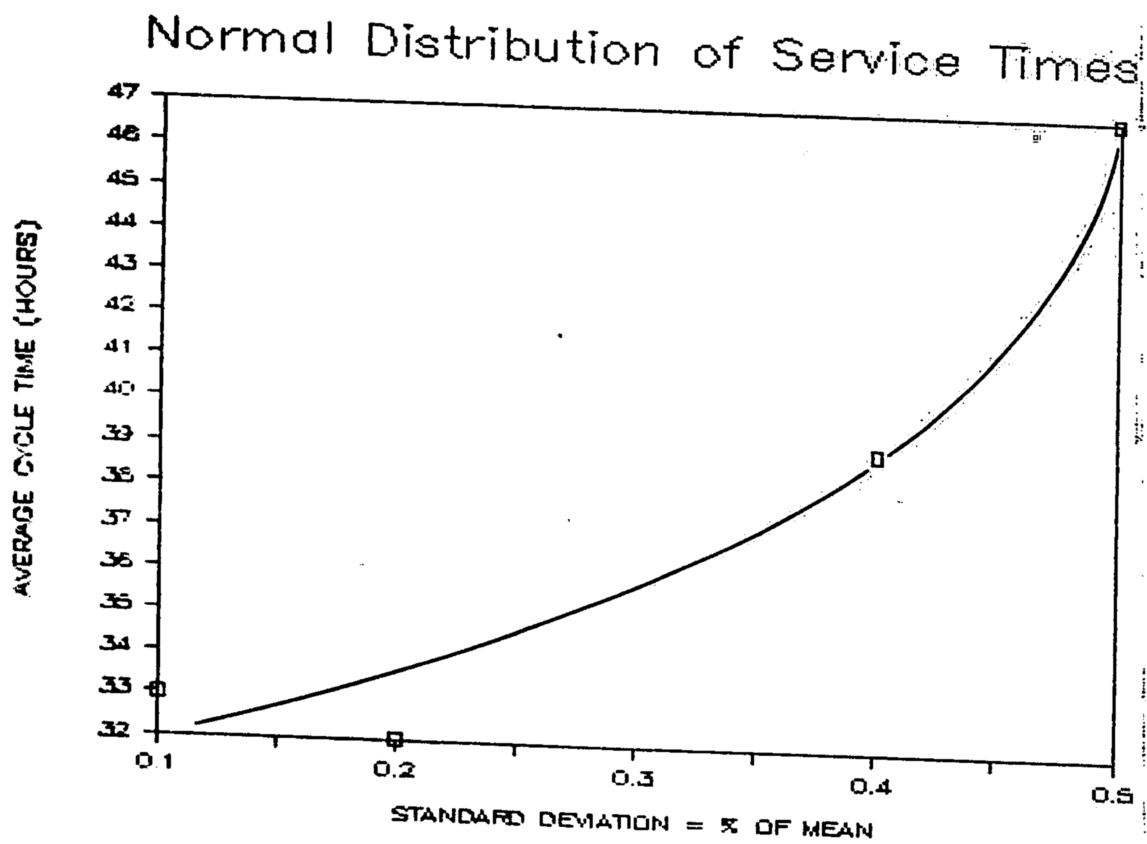


Figure 3.3

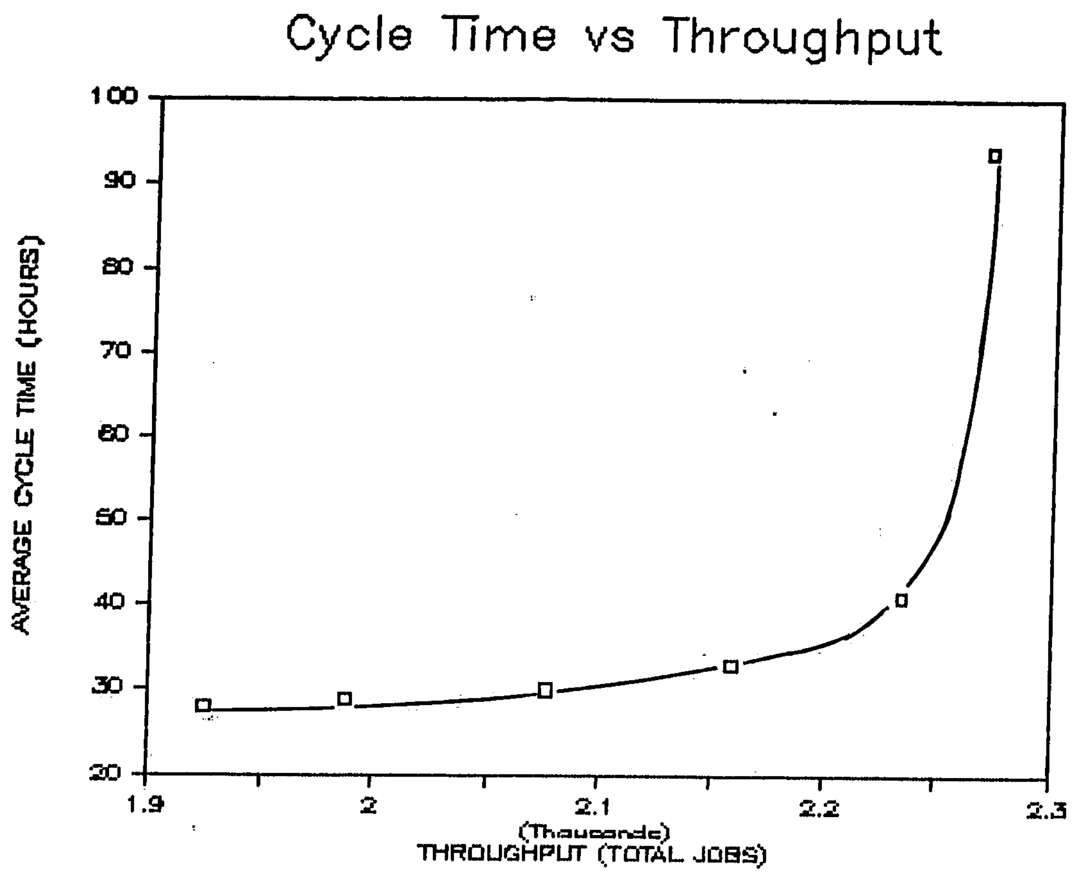


Figure 3.3

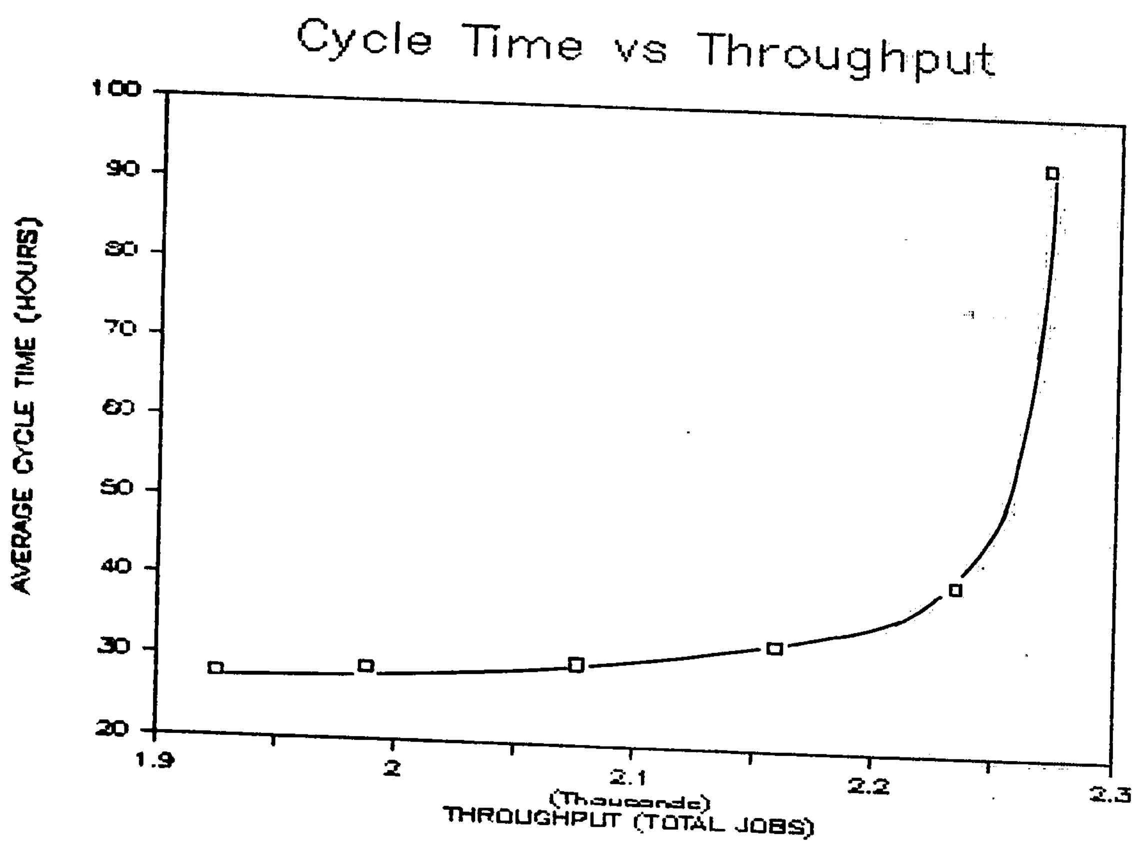


Figure 3.4

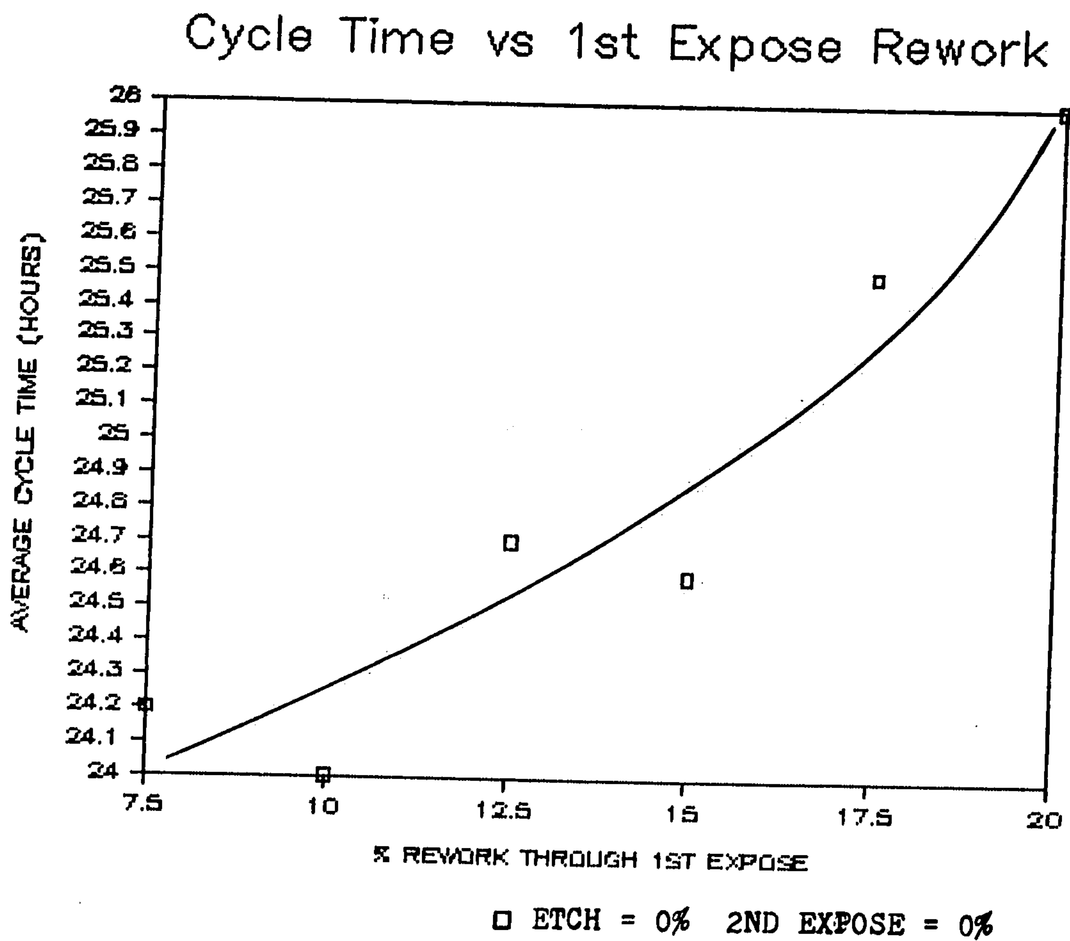


Figure 3.4

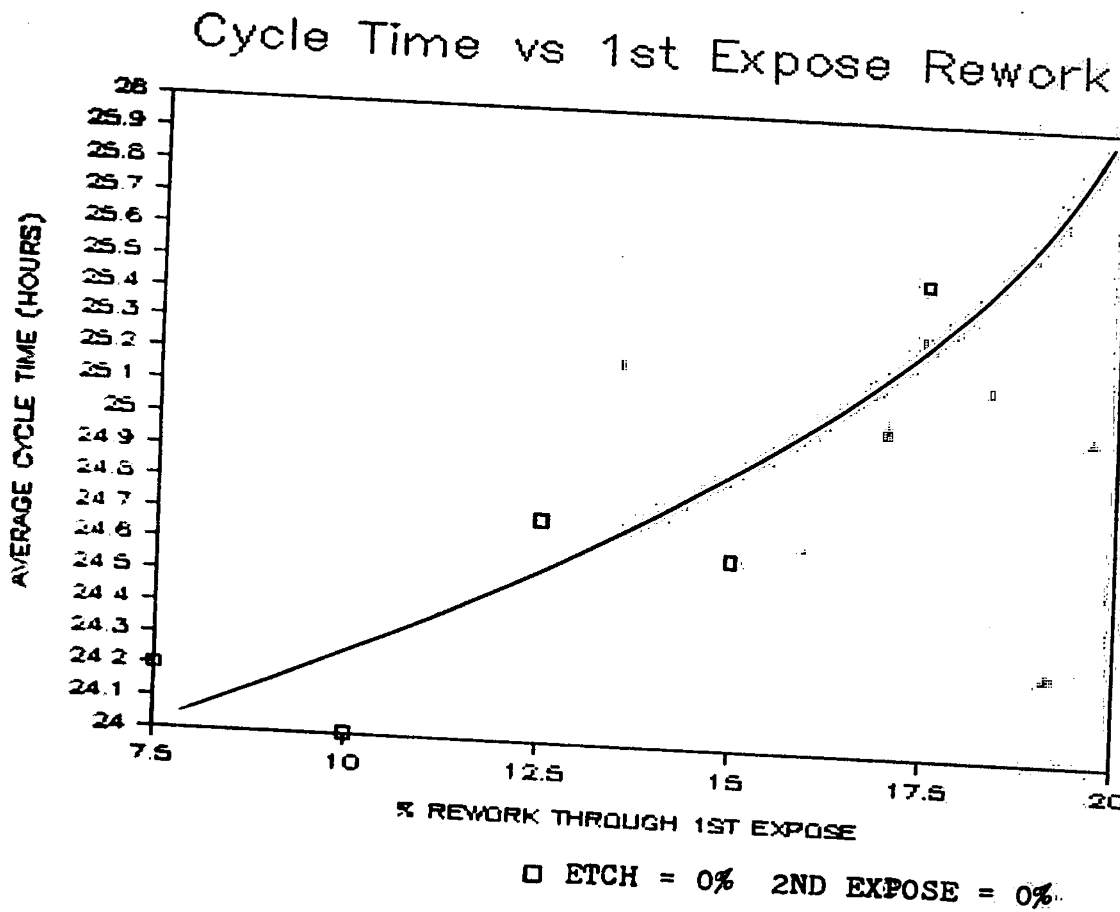


Figure 3.5

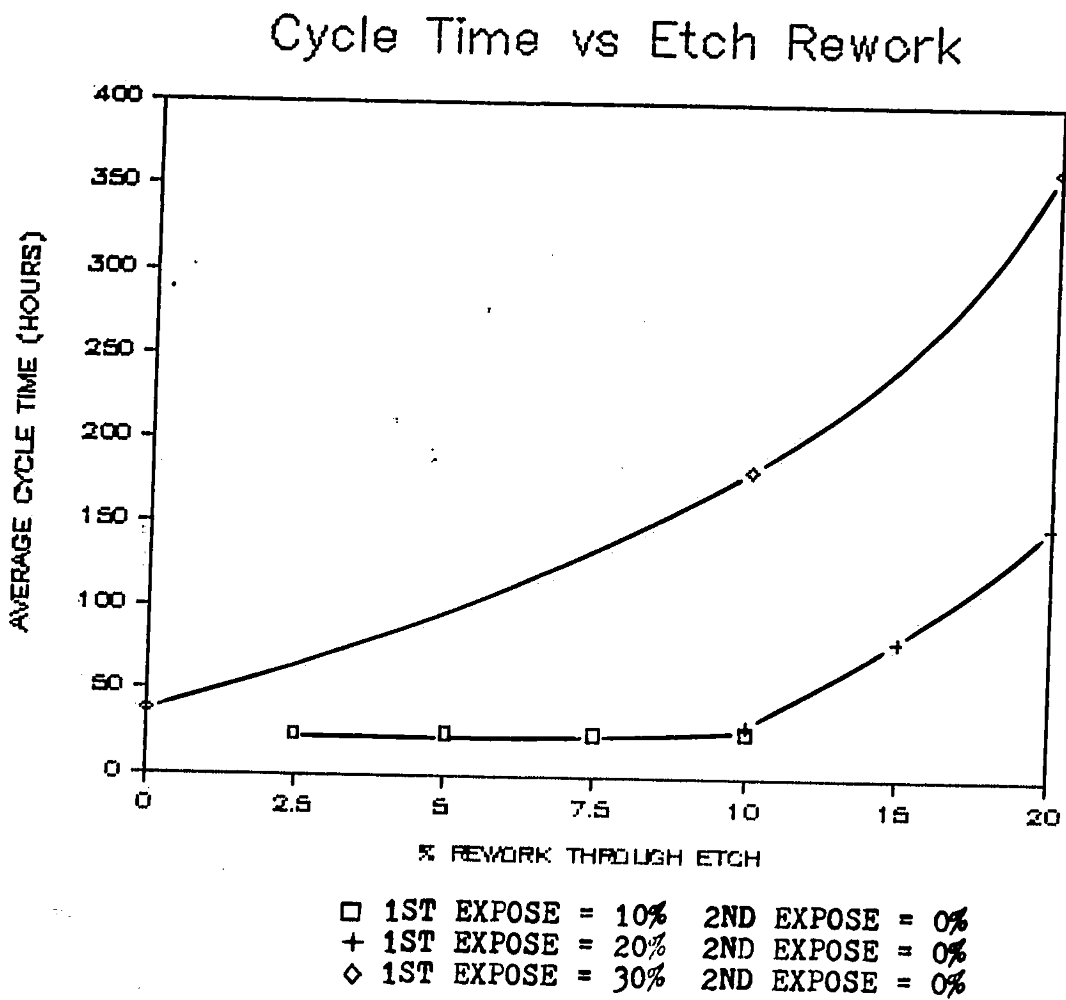


Figure 3.5

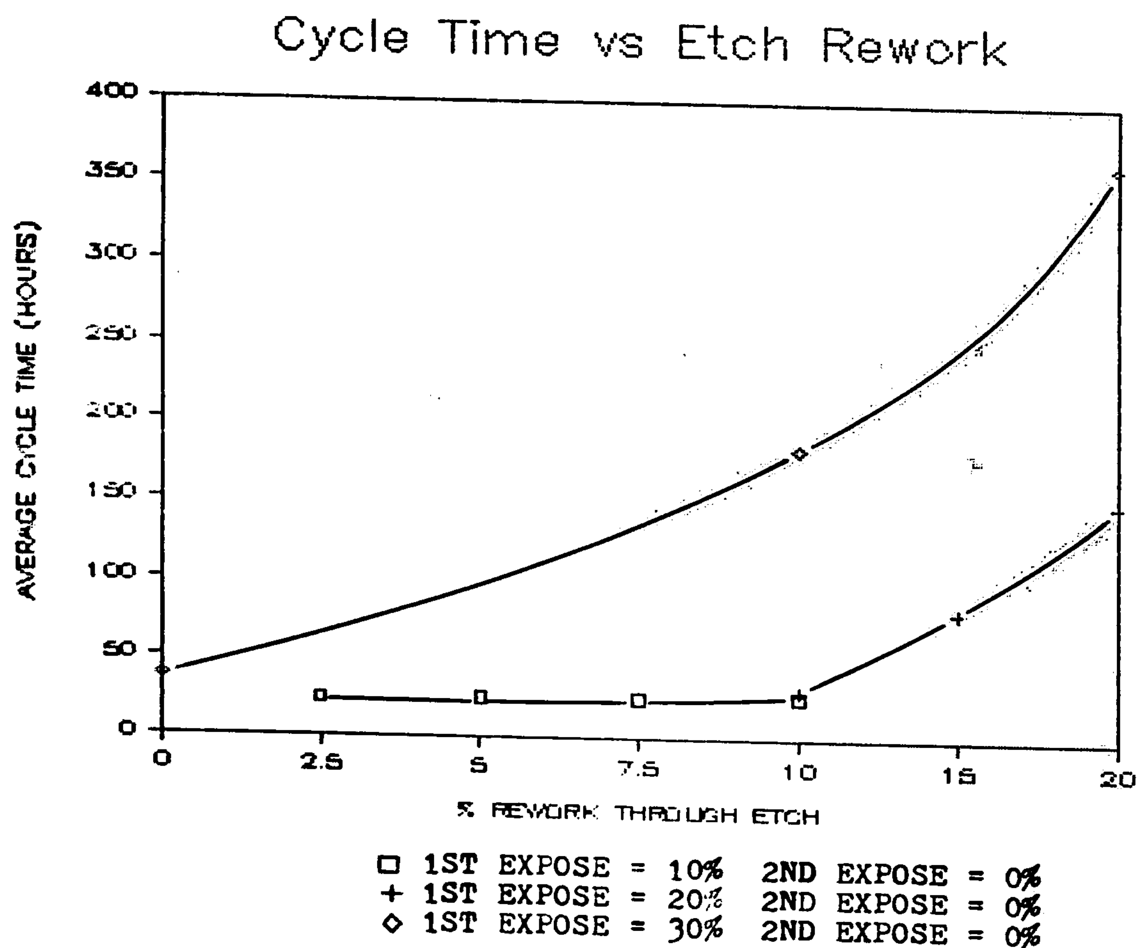


Figure 3.6

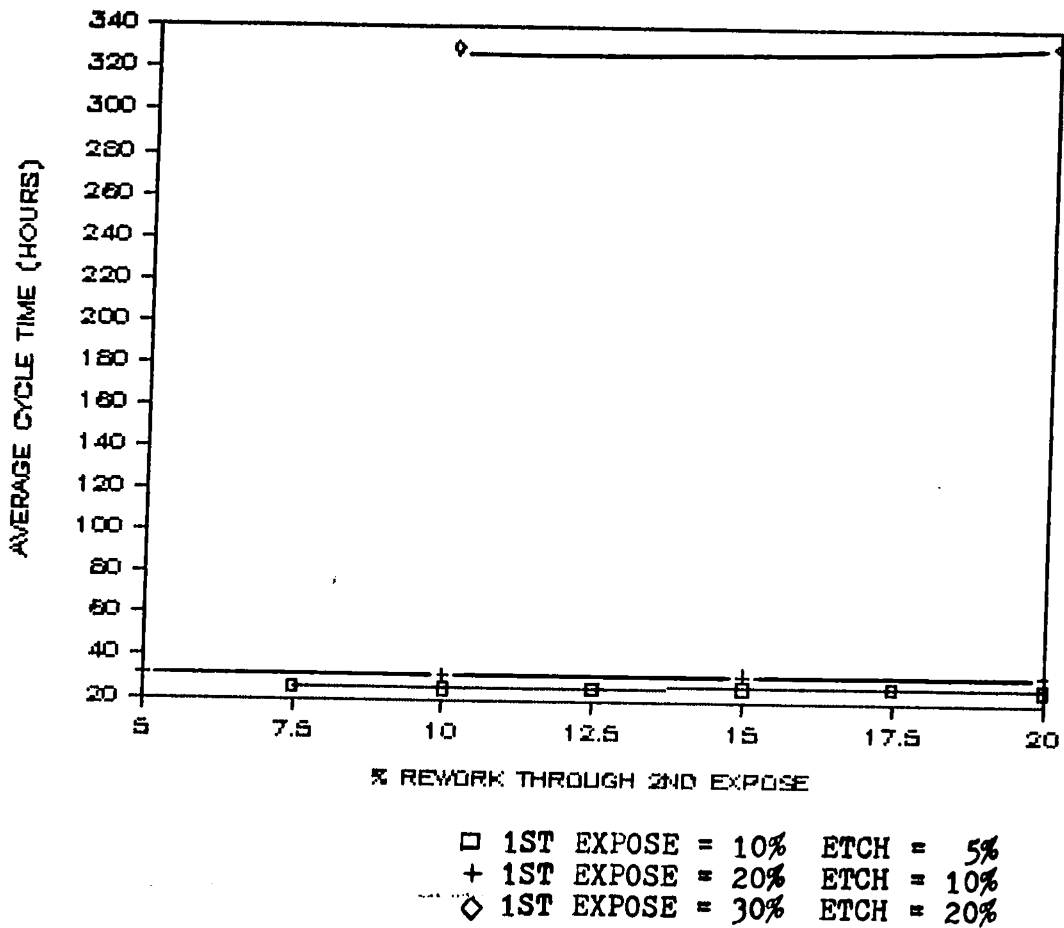


Figure 3.6

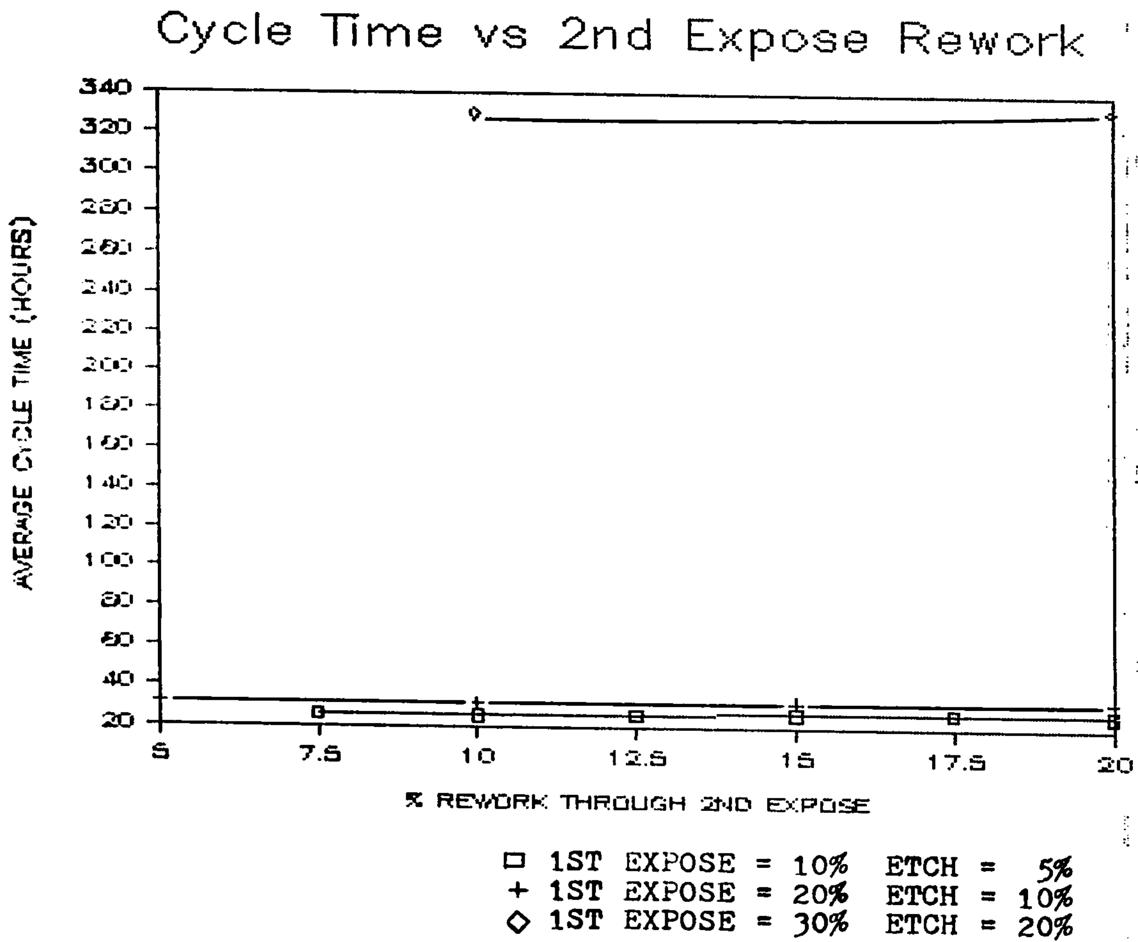


Figure 3.7

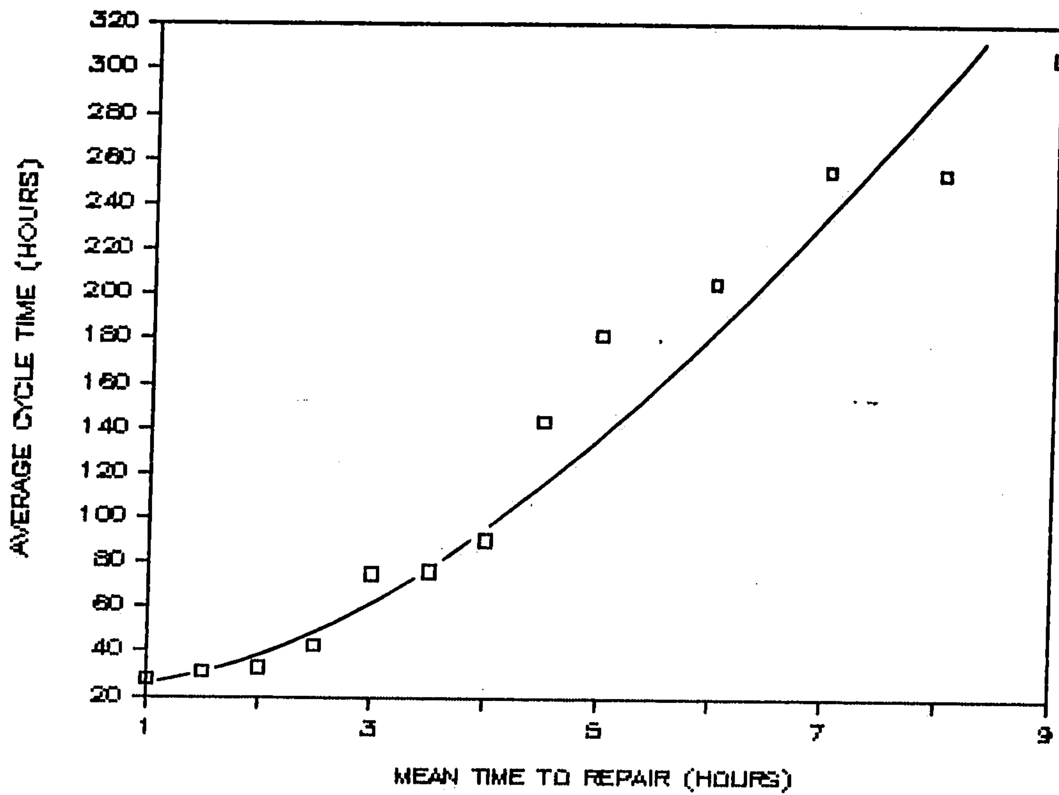
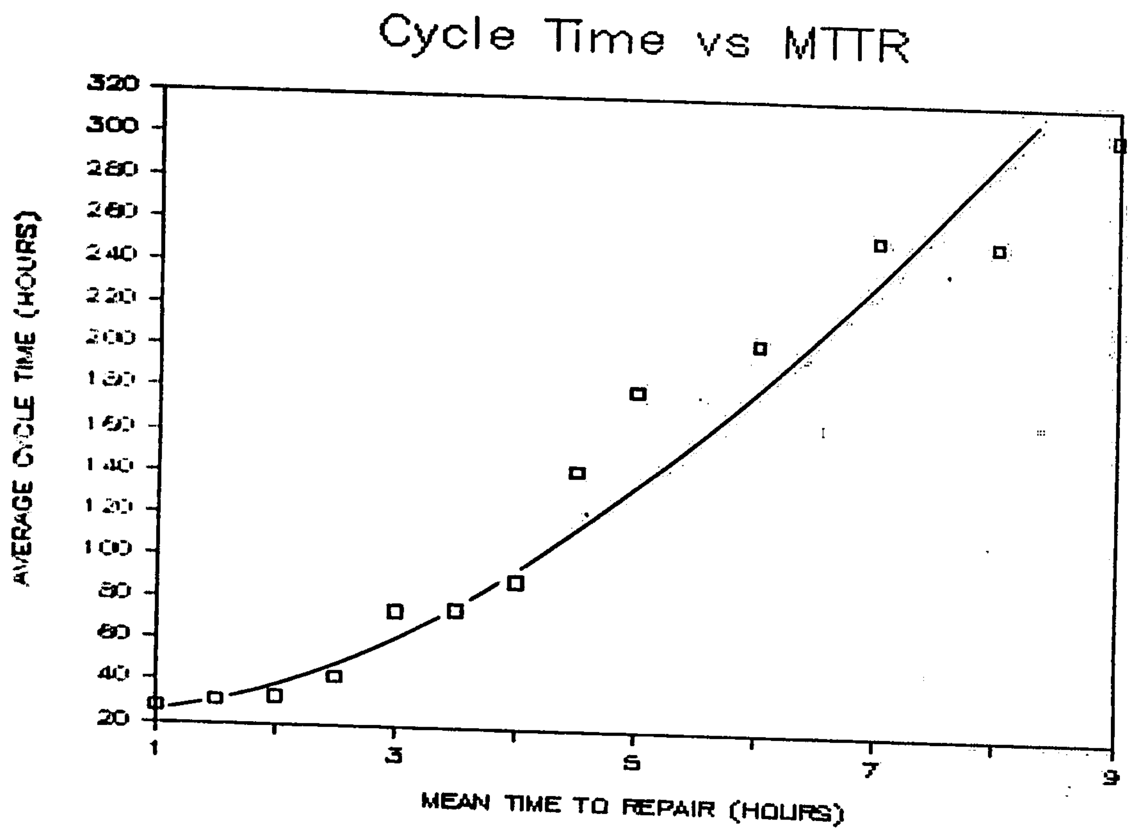


Figure 3.7



4.0 DISCUSSION OF RESULTS

This section analyzes the results of the microelectronics manufacturing simulation and seeks implications, lessons and generalizations that can be applied to real-world manufacturing. Each of the result sections will be individually revisited and analyzed.

4.1 SERVICE TIME IMPLICATIONS

The variability of the service time distributions resulted in directly increasing the variability of the total microelectronics processing line. The average line cycle time and work-in-process both increased when more variable service time distributions were used in the simulation. The major implication here is that the line would require additional in-process inventory (i.e. floor space) to accommodate the increased work-in-process for the more variable service time distribution situation.

The selection of the constant and exponential service time distributions resulted in capturing the extremes for possible line variability. An interesting result of using the various service time distributions is that the average cycle times for the normal service times (which were all made at a release rate of .77 jobs per hour) fell between average cycle times for the constant and exponential

service times. This makes sense, since it was already shown that added variability increased the average cycle time of the line.

The service time distribution variability suggests that if the processing times are only slightly variable, then the assumption of using constant service times is reasonable. As in the case with the normal distribution with a standard deviation equal to 10-20% of the mean, the average cycle time for the line did not seem to change at all. In fact, there might be a benefit due to adding only slight variability to the line. This may be a prime area for further analysis.

Logically, the constant service times make sense for these types of manufacturing processes since they are mainly composed of machine cycle times. However, it must be remembered that this study only considers two processing sectors and that even slight variability could be a major factor when considering the total processing line.

4.2 INPUT START IMPLICATIONS

The effect of increasing total line release levels was an increase in the total throughput of the line up to a point. Once the capacity of the slowest or gating tool was reached, increasing total releases only caused an

increase in the work-in-process for that gating tool.

At this point an additional gating sector resource could be added if the higher release schedule is permanent. However, in a capacity-constrained condition, this would only shift the work-in-process to the next slowest gating sector. In a capacity-unconstrained condition, the total throughput would be increased.

Another option for reducing this new work-in-process, would be to schedule periodic overtime for the gating sector to reduce the work-in-process. However, these pertubations would lead to additional variability which may cause excessive work-in-process variability further down the line. Moreover, scheduling overtime for the entire rest of the line would probably not be an economical solution since a capital expenditure for an additional gated resource may cost less.

The total average cycle time appeared to increase only slightly when input releases were increased in the capacity-unconstrained region. This result seems logical and indicates that until a gating sector is reached, the additional throughput can be attained with little affect on the line cycle time.

4.3 REWORK IMPLICATIONS

The effect of increasing the rework within the

microelectronics processing caused an increase in the average cycle time of the jobs. As rework was increased to higher levels, cycle time increased mildly unless the rework caused a sector to reach its capacity. At this point, the work-in-process would build as before and cause excessive cycle time increases.

The increased rework through 1st expose and etch caused a capacity-constrained condition as the rework reached a crucial level. Although the rework through 2nd expose did not cause a capacity constrained-condition, rework levels above 20% probably would have created a gating tool and then capacity also would have been limited.

The major implication with rework is that rework, should not be filling the line's extra capacity, if there is any. It would seem reasonable to want to limit rework levels, especially where capacity is a factor. Minor levels of rework may be acceptable for short durations if capacity is not constrained. However, in general it would seem best to eliminate rework completely, if possible, and use that capacity for additional production.

4.4 UNPLANNED MAINTENANCE IMPLICATIONS

The unplanned maintenance mean-time-to-repair had a

drastic effect on the variability of the processing line. As MTTR was increased, work-in-process throughout the line increased which also caused longer cycle times. Minor increases in the MTTR seemed to have only small affects on the line productivity while larger increases in MTTR caused major changes. In addition, if the increases in MTTR were large enough to cause a capacity-constrained condition, then the work-in-process would again build up behind the gating sector.

There appears to be large amounts of leverage with unplanned maintenance MTTR. The simulation showed how small variations to MTTR can cause major line variations. Although it is unlikely that MTTR for every machine on the floor would change at the same time, on the average, slight MTTR improvements appear to carry major capacity implications.

5.0 CONCLUSIONS AND SUMMARY

Microelectronics manufacturing is a complex set of process interactions which can be described as flow-shop manufacturing with parallel machines. This study utilized a SLAM simulation model to help clarify the interactions between the inputs (service time distributions, input starts, rework levels and unplanned maintenance mean-time-to-repair levels) and outputs (average job cycle time, throughput and work-in-process) of the microelectronics manufacturing process.

As more variable service time distributions were used in the simulation, the average cycle time and work-in-process increased. However, slightly variable service times (normal distributions with standard deviations equal to 10% to 20% of the mean service time value) seemed to have little affect on the line cycle time. Since the service times are mostly comprised of machine time which could only be slightly variable, it seems reasonable to use the constant service times for this study.

Increasing the job release levels likewise increased the throughput of the line until the capacity of the gating tool was reached. The limited capacity was asymptotically reached at the expense of drastically increasing average cycle times for the line. This

suggests that there exits an optimal release level, at some point just short of the capacity of the gating tool. If this level is surpassed, work-in-process builds in front of the gating tool.

Average cycle time also increased when rework levels were increased throughout the line. Small amounts of rework occurring when the line was in a capacity-unconstrained condition caused the average cycle time to increase slightly. However, when the line was in a capacity-constrained condition, this rework had more of a multiplying effect on the cycle time.

The unplanned maintenance mean-time-to-repair levels caused very drastic changes in the work-in-process and average line cycle time. Even in a capacity-unconstrained condition, MTTR changes of 1 hour radically changed the average cycle time, throughput and work-in-process of the line.

In summary, parameters adding variability to the manufacturing process appeared to increase the average cycle time and work-in-process of the line in every case. Additionally, the throughput of the line was also decreased as cycle times increased.

6.0 FUTURE AREAS OF STUDY

The microelectronics simulation study was an excellent educational vehicle for learning the SLAM simulation language, the microelectronics process and some of the interactions within the microelectronics process itself. This study covered two of the six major personality processing sectors and was able to show product flow including interactive affects due to changing service times and input starts, varying the amounts of rework and altering the mean-time-to-repair for unplanned equipment failures.

Additional simulation work with this manufacturing process should probably include the remaining personality sectors and possibly the masterslice sectors. A model for the total personality line would be useful for analyzing total personality cycle time, determining process gating tools, anticipating the maximum work-in-process, determining utilization of equipment and analyzing total capacity. Moreover, the same type of information could be obtained from a model of the masterslice portion of the manufacturing process.

A major factor to consider when increasing the size of this model, will be whether the simulation language can handle the increased size. In addition, the simulation run-time will increase, which may cause

problems.

In addition to increasing the scope of the model to include more of the process, including other information about the current model would be useful. The SLAM model could be improved to include such things as preventive maintenance downtime, variable product jobs, variable job sizes, priorities, scrap and yield.

Another related area of further study would be to optimize the physical floor layout with a computer layout program. Some of the simulation outputs, such as queue lengths and waiting times, would be useful inputs to most layout programs. Likewise, it would be of great interest to see how the tool layout might affect the line performance measurements in a simulation model.

REFERENCES

[1] Altiok, Tayfur, and Stidham, Shaler JR, "The Allocation of Interstage Buffer Capacities in Production Lines," _IIE Transactions_, 15, 4, December 1983, pp. 292-299.

[2] Apple, James M. _Plant Layout and Material Handling._ New York: John Wiley and Sons, Inc., 1977.

[3] Annino, Joseph S. and Russell, Edward C., "The Ten Most Frequent Causes of Simulation Analysis Failure and How to Avoid Them," _Simulation_, June 1979, pp. 137-140.

[4] Buxey, G. M., Slack N. D., and Wild, Ray, "Production Flow Line System Design - A Review," _AIIE Transactions_, 5, 1, March 1973, pp. 37-47.

[5] Buzacott, John A., "The Effect of Station Breakdowns and Random Processing Times on the Capacity of Flow Lines with In-Process Storage," _AIIE Transactions_, 4, 4, December 1972, pp. 308-312.

[6] Dissen, Richard, and Hancock, Walton, and Merten, Alan, "An Example of Simulation to Improve Plant Productivity," AIIE Transactions, 9, 1, March 1977, pp. 2-10.

[7] Feller, William. An Introduction to Probability Theory and its Applications. New York: John Wiley and Sons, Inc. 1950.

[8] Fishman, George S. Concepts and Methods in Discrete Event Digital Simulation. New York: John Wiley and Sons, 1973.

[9] Fox, Richard J. and Zerbe Darell R. "Some Practical System Availability Calcualtions," AIIE Transactions, 6, 3, September 1974, pp. 228-234.

[10] Freeman, Michael C., "The Effects of Breakdowns and Interstage Storage on Production Line Capacity," The Journal of Industrial Engineering, 15, 4, July-August 1964, pp. 194-200.

[11] French, Simon. Sequencing and Scheduling. West Sussex, England: Ellis Harwood, 1982.

[12] Gershwin, S. B., and Berman, Oded, "Analysis of Transfer Lines Consisting of Two Unreliable Machines with Random Processing Times and Finite Storage Buffers," AIIE Transactions, 13, 1, March 1981, pp. 2-10.

[13] Hillier, Frederick S. and Boling, Ronald W. "The Effect fo Some Design Factors on the Efficiency of Production Lines with Variable Operation Times," The Journal of Industrial Engineering, 17, 12, December 1966, pp. 651-658.

[14] Ignall, Edward, and Silver, Alvin, "The Output of a Two-Stage System with Unreliable Machines and Limited Storage," AIIE Transactions, 9, 2, June 1977, pp. 183-188.

[15] Kleijnen, Jack P. C. Statistical Techniques in Simulation-Part 1. New York: Marcel Dekker, Inc. 1974.

[16] Mellichamp, Joseph M., "Simulation Models Are A Flexible, Efficient Aid To Productivity Improvement Efforts," Industrial Engineering, August 1984, pp. 78-85.

[17] Mittra, Sitansu S., "Discrete System Simulation Concepts," Simulation, September 1984, pp. 142-144.

[18] Nelson, Elaine A., "The Use of a Generic Simulation Model in Evaluating Manufacturing Processes," Computer and Industrial Engineering, 7, 3, 1983, pp. 217-224.

[19] Pegdon, Claude D. and Pritsker, Alan B. Introduction to Simulation and Slam. New York: Halston Press, 1979.

[20] Pratt, Charles A., "Simulation Tools For Manufacturing," Simulation, October 1982, pp. 143-144.

[21] Pritsker, A. Alan B., "Models Yield Keys To Productivity Problems, Solutions," Industrial Engineering, October 1983, pp. 83-87.

[22] Schruben, Lee W., "Establishing The Credibility of Simulations," Simulation, March 1980, pp. 101-105.

[23] Solberg, James J. "Capacity Planning with a Stochastic Workflow Model," AIIE Transactions, 13, 2, June 1981, pp. 116-122.

[24] Sze, S. M. VLSI Technology. New York: McGraw Hill Book Company, 1983.

APPENDIX A

Table A2.2: Mean Time Between Failure Assumptions.

MTBF (Hours)	25	50	125
Equipment			
Apply	x		
Expose	x		
Composite Insulator		x	
RIE		x	
Metal Evaporators		x	
Lift Off		x	
Insulator Deposition			x
Etch			x

Table A3.1: Constant Service Times Effect On The Production Parameters.

Release Rate (j/hr)	.74	.77	.80	.83
Cycle Time (hrs)	30.00	33.00	41.00	94.00
Throughput	2076.00	2159.00	2234.00	2271.00

Sector	Average Queue Length			
1	.04	.05	.06	.06
2	1.43	2.84	9.16	53.76
3	.00	.00	.00	.00
4	.40	.49	.71	.65
5	.09	.10	.11	.08
6	.21	.30	.44	.46
7	.32	.39	.68	.55
8	.00	.00	.00	.00
9	.25	.20	.15	.06
10	.00	.00	.00	.00
11	.66	1.13	1.04	1.09
12	.00	.00	.00	.00
13	.44	.53	.92	.80
14	.00	.00	.00	.00
15	.10	.10	.10	.10

Table A3.2: Exponential Service Times Effect on The Production Parameters.

Release Rate (j/hr)	.74	.77	.80	.83
Cycle Time (hrs)	47.00	52.00	67.00	112.00
Throughput	2077.00	2159.00	2231.00	2221.00

Sector	Average Queue Length			
1	.12	.17	.11	.15
2	4.07	6.38	16.21	58.86
3	.00	.00	.00	.00
4	2.79	2.29	3.75	3.35
5	.63	.96	1.25	1.26
6	1.96	2.24	2.75	2.76
7	1.64	1.86	2.39	1.98
8	.08	.09	.09	.08
9	.54	.55	.71	.76
10	.00	.00	.00	.00
11	1.94	3.27	3.08	4.25
12	.00	.00	.00	.00
13	2.12	2.48	3.02	4.13
14	.06	.05	.07	.06
15	.60	.60	.43	.59

Table A3.3: Normal Service Times Effect On The Production Parameters.

Std. Deviation	10.00	20.00	40.00	50.00
Cycle Time (hrs)	33.00	32.00	39.00	47.00
Throughput	2153.00	2153.00	2156.00	2152.00

Sector	Average Queue Length			
1	.06	.05	.06	.05
2	3.16	2.63	5.83	10.29
3	.00	.00	.00	.00
4	.60	.46	1.00	1.21
5	.07	.11	.23	.34
6	.38	.40	.59	.92
7	.45	.47	.71	.99
8	.01	.01	.01	.01
9	.11	.16	.29	.33
10	.00	.00	.00	.00
11	.70	.78	1.09	1.55
12	.00	.00	.00	.00
13	.56	.63	.96	1.05
14	.00	.00	.01	.02
15	.21	.15	.24	.28

Table A3.4: Line Throughput Effect on Production Parameters (with constant service times).

Release Rate (j/hr)	.69	.71	.77	.83
Cycle Time (hrs)	28.10	29.20	33.00	94.00
Throughput	1925.00	1988.00	2159.00	2271.00

Sector	Average Queue Length			
1	.04	.04	.05	.06
2	.58	1.02	2.84	53.76
3	.00	.00	.00	.00
4	.32	.29	.49	.65
5	.04	.04	.10	.08
6	.12	.13	.30	.46
7	.23	.27	.39	.55
8	.00	.00	.00	.00
9	.12	.16	.20	.06
10	.00	.00	.00	.00
11	.38	.46	1.13	1.09
12	.00	.00	.00	.00
13	.26	.54	.53	.80
14	.00	.00	.00	.00
15	.12	.13	.10	.14

Table A3.5: Sector Resource Increase Effect On The Production Parameters.

		Sector # 2 (+ 1 Resource)
Release Rate (jobs/hr)	1.00	1.00
Cycle Time (hrs)	337.00	113.00
Throughput (jobs)	2244.00	2668.00

Sector	Average Queue Length	
1	.12	.07
2	313.67	.94
3	.00	.00
4	.72	57.91
5	.11	.41
6	.70	13.22
7	.77	1.73
8	.00	.00
9	.16	.23
10	.00	.00
11	1.19	4.36
12	.00	.00
13	.60	8.34
14	.00	.01
15	.33	.20

Table A3.6: 1st Expose Rework Effect On The Production Parameters. (Etch Rework=0%, 2nd Expose Rework=0%)

Rework (%)	10.00	12.50	15.00	20.00
Cycle Time (hrs)	24.00	26.70	26.60	26.10
Throughput	2153.00	2154.00	2151.00	2147.00

Sector	Average Queue Length			
1	.00	.00	.00	.01
2	.11	.20	.24	.58
3	.00	.00	.00	.00
4	.10	.12	.11	.27
5	.04	.03	.02	.03
6	.12	.14	.12	.15
7	.36	.36	.40	.63
8	.00	.00	.00	.00
9	.11	.18	.13	.16
10	.00	.00	.00	.00
11	.15	.23	.17	.25
12	.00	.00	.00	.00
13	.09	.14	.19	.17
14	.00	.00	.00	.00
15	.13	.17	.09	.13

Table A3.7: Etch Rework Effect On The Production Parameters. (1st Expose Rework=10%, 2nd Expose Rework=0%)

Rework (%)	2.50	5.00	7.50	10.00
Cycle Time (hrs)	24.50	25.10	25.20	26.20
Throughput	2155.00	2154.00	2154.00	2157.00

Sector	Average Queue Length			
1	.01	.01	.02	.01
2	.18	.28	.38	.51
3	.00	.00	.00	.00
4	.15	.14	.13	.29
5	.04	.04	.04	.05
6	.13	.14	.15	.24
7	.39	.37	.44	.57
8	.00	.00	.00	.00
9	.16	.20	.10	.16
10	.00	.00	.00	.00
11	.12	.18	.14	.18
12	.00	.00	.00	.00
13	.16	.14	.14	.13
14	.00	.00	.00	.00
15	.09	.18	.11	.12

Table A3.8: Etch Rework Effect On The Production Parameters. (1st Expose Rework=20%, 2nd Expose Rework=0%)

Rework (%)	10.00	15.00	20.00
Cycle Time (hrs)	30.00	81.30	150.00
Throughput	2162.00	2103.00	2011.00

Sector	Average Queue Length		
1	.01	.03	.01
2	2.79	41.76	95.11
3	.00	.00	.00
4	.51	.54	.62
5	.06	.10	.12
6	.35	.24	.16
7	.39	.39	.34
8	.00	.00	.00
9	.13	.18	.10
10	.00	.00	.00
11	.14	.26	.18
12	.00	.00	.00
13	.09	.19	.17
14	.00	.00	.00
15	.21	.19	.12

Table A3.9: Etch Rework Effect On The Production Parameters. (1st Expose Rework=30%, 2nd Expose Rework=0%)

Rework (%)	00.00	10.00	20.00
Cycle Time (hrs)	38.30	183.20	362.40
Throughput	2162.00	1952.00	1706.00

Sector	Average Queue Length		
1	.04	.03	.07
2	9.21	121.65	258.82
3	.00	.00	.00
4	.48	.69	.81
5	.06	.07	.05
6	.37	.19	.06
7	.38	.48	.22
8	.00	.00	.00
9	.13	.12	.11
10	.00	.00	.00
11	.13	.17	.10
12	.00	.00	.00
13	.13	.12	.09
14	.00	.00	.00
15	.16	.14	.10

Table A3.10: 2nd Expose Effect On The Production Parameters. (1st Expose Rework=10%, Etch Rework=5%)

Rework (%)	10.00	12.50	15.00	17.50
Cycle Time (hrs)	25.30	25.90	26.40	26.90
Throughput	2154.00	2153.00	2154.00	2149.00

Sector	Average Queue Length			
1	.02	.02	.02	.03
2	.29	.36	.28	.35
3	.00	.00	.00	.00
4	.12	.18	.16	.23
5	.04	.03	.04	.04
6	.13	.16	.15	.24
7	.30	.32	.29	.44
8	.00	.00	.00	.00
9	.05	.10	.24	.12
10	.00	.00	.00	.00
11	.28	.43	.69	.50
12	.00	.00	.00	.00
13	.18	.20	.30	.44
14	.00	.00	.00	.00
15	.15	.12	.08	.19

Table A3.11: 2nd Expose Effect On The Production Parameters. (1st Expose Rework=20%, Etch Rework=10%)

Rework (%)	5.00	10.00	15.00	20.00
Cycle Time (hrs)	31.60	31.50	32.40	33.00
Throughput	2151.00	2165.00	2155.00	2159.00

Sector	Average Queue Length			
1	.02	.03	.03	.06
2	3.62	3.08	3.58	2.84
3	.00	.00	.00	.00
4	.53	.53	.48	.49
5	.07	.12	.06	.10
6	.32	.35	.30	.30
7	.51	.31	.39	.39
8	.00	.00	.00	.00
9	.05	.11	.12	.20
10	.00	.00	.00	.00
11	.30	.26	.50	1.13
12	.00	.00	.00	.00
13	.25	.41	.38	.53
14	.00	.00	.00	.00
15	.15	.20	.18	.10

Table A3.12: 2nd Expose Effect On The Production Parameters. (1st Expose Rework=30%, Etch Rework=20%)

Rework (%)	10.00	20.00
Cycle Time (hrs)	329.60	333.30
Throughput	1745.00	1763.00

Sector	Average Queue Length	
1	.05	.11
2	235.59	234.81
3	.00	.00
4	.60	.70
5	.04	.06
6	.08	.09
7	.21	.30
8	.00	.00
9	.19	.18
10	.00	.00
11	.22	.55
12	.00	.00
13	.10	.22
14	.00	.00
15	.13	.08

Table A3.13: Mean-Time-To-Repair Effect on Production Parameters.

MTTR (hrs)	1.00	2.00	2.50	3.00
Cycle Time (hrs)	27.90	33.00	42.70	75.10
Throughput	2154.00	2159.00	2165.00	2145.00

Sector	Average Queue Length			
1	.05	.06	.05	.06
2	1.19	2.83	9.53	32.04
3	.00	.00	.00	.00
4	.18	.49	.77	1.06
5	.03	.10	.12	.21
6	.16	.30	.49	.65
7	.21	.39	.36	1.33
8	.00	.00	.00	.00
9	.03	.20	.19	.18
10	.00	.00	.00	.00
11	.33	1.13	.91	1.41
12	.00	.00	.00	.00
13	.13	.53	1.00	1.46
14	.00	.00	.00	.00
15	.05	.10	.28	.26

Table A3.14 Mean-Time-To-Repair Effect on Production Parameters (continued from A3.13).

MTTR (hrs)	4.00	5.00	6.00	8.00
Cycle Time (hrs)	90.00	182.00	205.00	254.00
Throughput	2083.00	2007.00	1965.00	1932.00

Sector	Average Queue Length			
1	.19	.14	.18	.46
2	41.57	109.04	110.67	127.11
3	.00	.00	.00	.00
4	1.35	2.12	8.77	8.56
5	.16	.57	.39	1.51
6	.53	.66	1.56	1.60
7	1.36	1.22	.94	2.58
8	.00	.00	.00	.00
9	.57	.66	1.04	3.38
10	.00	.00	.00	.00
11	3.30	4.36	4.88	11.21
12	.00	.00	.00	.00
13	1.52	3.32	7.97	18.37
14	.00	.00	.00	.00
15	.57	.58	1.04	2.12

APPENDIX B

SLAM NETWORK FOR MICROELECTRONICS SIMULATION MODEL

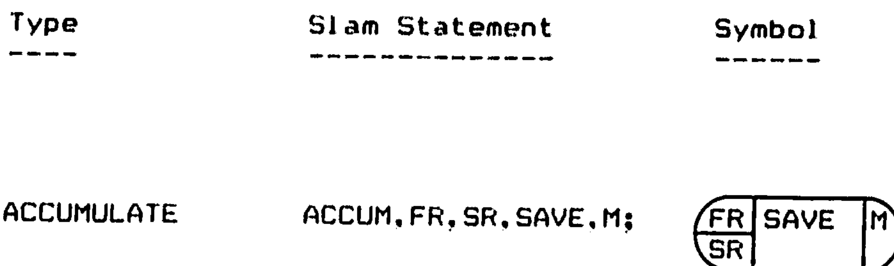
```
GEN, M. A. KOSCHMEDER, KLREUK SECTORS,10/27/84,1,YES;
LIMITS,18,3,4000;
NETWORK;
        RESOURCE/STRIP(1),1;                RESOURCES
        RESOURCE/APPLY1(3),2;
        RESOURCE/EXPOSE(2),4;
        RESOURCE/ETCH(3),5;
        RESOURCE/METAL(5),6;
        RESOURCE/LIFTOFF(1),7;
        RESOURCE/INSULATOR(5),8;
        RESOURCE/INS(1),9;
        RESOURCE/APPLY2(2),11;
        RESOURCE/EXP2(2),13;
        RESOURCE/RIE2(4),14;
        RESOURCE/STRIP2(1),15;
        CREATE,1.3,0,1,3000;
        ASSIGN,ATRIB(2)=1;
        ACT,,,STRT;
RWK1    AWAIT(1),STRIP/1;                   STRIP
        ACT/1,,.5;
        FREE,STRIP;
STRT    AWAIT(2),APPLY1/1;                  APPLY
        ACT/2,2.5;
        FREE,APPLY1;
        QUEUE(3);                           INSP
        ACT(10)/3,.5;
        AWAIT(4),EXPOSE/1;                  EXPOSE
        ACT/4,1.4;
        FREE,EXPOSE;
        GOON;                               REWORK (1-X)%
        ACT,,.8,SECD;
        ACT,,.2,RWK1;
SECD    AWAIT(5),ETCH/1;                    ETCH
        ACT/5,2.2;
        FREE,ETCH;
        GOON;                               REWORK (1-Y)%
        ACT,,.90,THRD;
        ACT,,.10,RWK1;
THRD    AWAIT(6),METAL/1;                   METAL
        ACT/6,5.0;
        FREE,METAL;
        AWAIT(7),LIFTOFF/1;                 LIFTOFF
```

```
          ACT/7,.9;
          FREE,LIFTOFF;
          AWAIT(8),INSULATOR/1;                 INSULATOR
          ACT/8,2.9;
          FREE,INSULATOR
          AWAIT(9),INS/1;
          ACT/9,.6;
          FREE,INS;
          ACT,,,SKIP;
RWK2      AWAIT(1),STRIP/1;                     STRIP
          ACT/10,.5;
          FREE,STRIP;
SKIP      AWAIT(11),APPLY2;                     APPLY
          ACT/11,1.5;
          FREE,APPLY2;
          QUEUE(12);                            INSPECT
          ACT(10)/12,.4;
          AWAIT(13),EXP2;
          ACT/13,1.5;                           EXPOSE
          FREE,EXP2;
          GOON;                                 REWORK (1-Z)%
          ACT,,.80,FOTH;
          ACT,,.20,RWK2;
FOTH      AWAIT(14),RIE2;                       RIE ETCH
          ACT/14,2.0;
          FREE,RIE2;
          AWAIT(15),STRIP2;                     STRIP
          ACT/15,.6;
          FREE,STRIP2;
          COLCT,INT(1),TIME IN SYSTEM;
          TERM;
          CREATE,,1,,1;
          ASSIGN,ATRIB(1)=.001;
DOWN      ASSIGN,ATRIB(2)=TNOW;
          ACT,EXPON(120,1);                     MTBF
          COLCT,INT(2),STRIP MTBF;
          AWAIT(1),STRIP/1;
          ASSIGN,ATRIB(3)=TNOW;
          ACT,EXPON(2,2);                       MTTR
          COLCT,INT(3),STRIP MTTR;
          FREE,STRIP;
          ACT,,,DOWN;
          TERM;
          CREATE,,1,,1;
          ASSIGN,ATRIB(1)=.001;
DOW2      ASSIGN,ATRIB(2)=TNOW;
          ACT,EXPON(25,3);                      MTBF
          COLCT,INT(2),APPLY1 MTBF;
          AWAIT(2),APPLY1/1;
          ASSIGN,ATRIB(3)=TNOW;
```

```
        ACT,EXPON(2,4);                          MTTR
        COLCT,INT(3),APPLY1 MTTR;
        FREE,APPLY1;
        ACT,,,DOW2;
        TERM;
        END;
PRIORITY,1,LVF(1);                              FILE PRIORITIES
PRIORITY,2,LVF(1);
PRIORITY,4,LVF(1);
PRIORITY,5,LVF(1);
PRIORITY,6,LVF(1);
PRIORITY,7,LVF(1);
PRIORITY,8,LVF(1);
PRIORITY,9,LVF(1);
PRIORITY,11,LVF(1);
PRIORITY,13,LVF(1);
PRIORITY,14,LVF(1);
PRIORITY,15,LVF(1);
MONTR,CLEAR,200;
INIT,0,3000;
FIN;
```

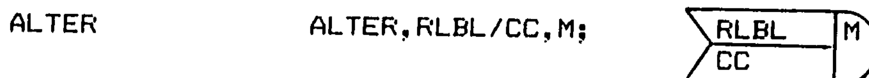
APPENDIX C

SLAM NETWORK SYMBOLS



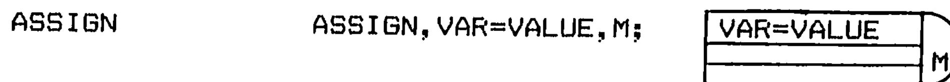
Type	Slam Statement	Symbol
ACCUMULATE	ACCUM,FR,SR,SAVE,M;	

The ACCUM node combines activities by specifying a release mechanism. FR is the number of arrivals for the first release. SR is the number of arrivals for subsequent releases. SAVE is the attribute holding criterion for entities. M is the maximum emanating activities.



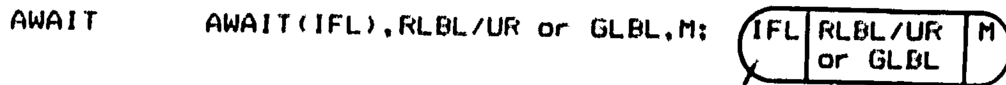
Type	Slam Statement	Symbol
ALTER	ALTER,RLBL/CC,M;	

The ALTER node changes the capacity of resource RLBL by CC units. M is the maximum emanating activities.

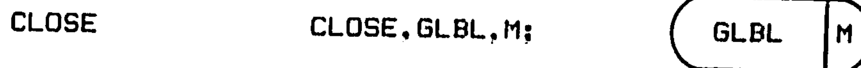


Type	Slam Statement	Symbol
ASSIGN	ASSIGN,VAR=VALUE,M;	

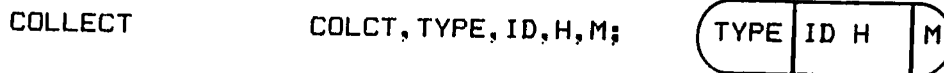
The ASSIGN node assigns values to slam variables as each entity arrivals to the node. M is the maximum emanating activities.



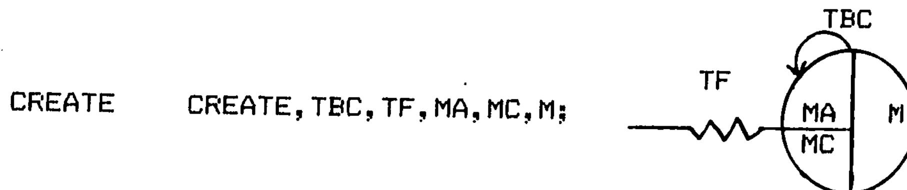
The AWAIT node delays entities in file IFL based on availability of UR units of resource RLBL or the status of gate GLBL. M is the maximum emanating activities.



The CLOSE node changes the status of gate GLBL to closed.



The COLCT node collects statistics on entities or variables arriving at nodes. TYPE specifies the type of statistics to be recorded. ID is an identifier for output purposes. H specifies parameters for output reports. M is the maximum emanating activities.

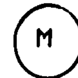


The CREATE node generates entities. The time of the first release is TF. The time between releases is TBC. The maximum number of releases is MC. The time of the creation is stored in attribute MA. M is the maximum emanating activities.

FREE FREE,RLBL/UF,M;

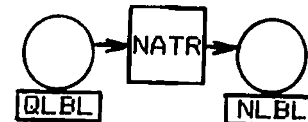
The FREE node releases UF units of resource RLBL. Freed units are made available to entities waiting at await and preempt nodes.

GOON GOON,M;



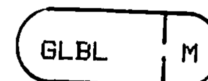
The GOON node provides a continuation node where every entering entity passes directly through the node.

MATCH MATCH,NATR,QLBL/NLBL, repeats..;



The MATCH node delays movement of entities. When a match on attribute NATR occurs, each matched entity is released from ity QUEUE node (QLBL) to the node labled NLBL.

OPEN OPEN,GLBL,M;



The OPEN node changes the status of gate GLBL to open.

PREEMPT PREEMPT(IFL)/PR,RLBL,SNLBL,NATR,M;

IFL | PR | RLBL / NATR | M

SNLBL

The PREEMPT node preempts resources seized by entities at await nodes. Priorities PR can be assigned to the preempted entities. Attribute NATR stores remaining activity time. Preempted entities are routed to the node labeled NLBL.

QUEUE QUEUE(IFL),IQ,QC,BLOCK or BALK,SLBL;

IQ / QC | IFL

The QUEUE node delays entities in file IFL until a server is available. Queue initially contains IQ entities. Queue capacity is QL. For multiple queues, SLBL is the label of the associated select node.

SELECT SELECT,QSR,SSR,BLOCK or BALK,QLBL'S;

QSR | SSR

The select node selects from queues (QLBL's) and available servers based on the queue selection rule (QSR) and the server selection rule (SSR).

TERMINATE TERM,TC;

TC

The TERM node ends the simulation by destroying entities through the TC entity which stops the simulation.

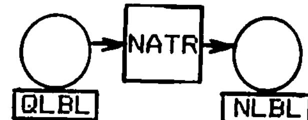
FREE FREE,RLBL/UF,M;

The FREE node releases UF units of resource RLBL. Freed units are made available to entities waiting at await and preempt nodes.

GOON GOON,M;

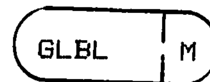
The GOON node provides a continuation node where every entering entity passes directly through the node.

MATCH MATCH,NATR,QLBL/NLBL, repeats..;



The MATCH node delays movement of entities. When a match on attribute NATR occurs, each matched entity is released from ity QUEUE node (QLBL) to the node labled NLBL.

OPEN OPEN,GLBL,M;

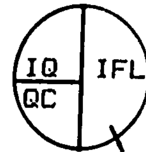


The OPEN node changes the status of gate GLBL to open.

PREEMPT   PREEMPT(IFL)/PR,RLBL,SNLBL,NATR,M;

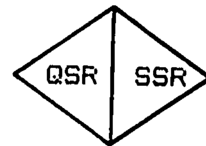
The PREEMPT node preempts resources seized by entities at await nodes. Priorities PR can be assigned to the preempted entities. Attribute NATR stores remaining activity time. Preempted entities are routed to the node labeled NLBL.

QUEUE   QUEUE(IFL),IQ,QC,BLOCK or BALK,SLBL;



The QUEUE node delays entities in file IFL until a server is available. Queue initially contains IQ entities. Queue capacity is QL. For multiple queues, SLBL is the label of the associated select node.

SELECT   SELECT,QSR,SSR,BLOCK or BALK,QLBL'S;



The select node selects from queues (QLBL's) and available servers based on the queue selection rule (QSR) and the server selection rule (SSR).

TERMINATE   TERM,TC;

The TERM node ends the simulation by destroying entities through the TC entity which stops the simulation.

OTHER SLAM SYMBOLS

ACTIVITY   ACT(N)/A,duration,PROB or COND,NLBL;   DUR. PROB → (N) [A]

The ACTIVITY node is used to delay entities for a specified duration or for probabilistic (PROB) or conditional (COND) branching. The number of multiple servers is given by N. Statistics are provided on the activity if it is labeled with an activity number A. Non-sequential routing is accomplished by specifying a node label NLBL.

RESOURCE   RESOURCE/RLBL(IRC),IFL's;

RLBL (IRC)	IFL	IFL

The RESOURCE block defines a resource labeled RLBL with an initial capacity of IRC. The await or preempt nodes desiring units of the resource are listed by their file numbers IFL's which are given in increasing priority order.

GATE   GATE/GLBL,OPEN or CLOSED,IFL's;

GLBL	OPEN or CLOSED	IFL	IFL

The GATE node is used to initially label gates GLBL as OPEN or CLOSED. Await nodes where entities are queued for gate operations are referenced by their file numbers (IFL's).

Vita

Mikel Allan Koschmeder was born in Readlyn, Iowa on April 15, 1958 to Erwin John and Carol Ann Koschmeder. He attended Iowa State University in Ames, Iowa from 1976 through 1981. As part of his undergraduate training, Mikel also co-oped for the Kellogg Cereal Company in Omaha, Nebraska. In May 1981, he received a B.S. in Industrial Engineering from Iowa State University. From June 1981 until present, Mikel has worked as an industrial engineer for the IBM Corporation in East Fishkill, New York.