Proceedings of the 1996 Winter Simulation Conference
ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain

A COMPARISON STUDY OF THE LOGIC OF FOUR WAFER FABRICATION SIMULATORS

Scott J. Mason

Advanced Micro Devices
5204 East Ben White Boulevard, Mailstop 612
Austin, Texas 78741, U. S. A.

Paul A. Jensen

The University of Texas at Austin
Mechanical Engineering, Mailcode C2200
Austin, Texas 78712, U. S. A.

John W. Fowler

Industrial & Management Systems Engineering
Arizona State University
Tempe, Arizona 85287-5906, U. S. A.

ABSTRACT

Many semiconductor manufacturing companies use one of the following four simulators to aid in analyzing, planning, and operating their manufacturing facilities: Tyecin Systems' ManSim/X, AutoSimulations' AutoSched, Systems Modeling's Wafer Fabrication Template, or Chance Industrial Solution's Delphi, which Wright, Williams, and Kelly now licenses as Factory Explorer. A benchmark study of the four packages was conducted, using actual factory data. The packages each produce different results between replications of the data, due to the data's stochastic nature. The differences which exist between the simulation packages studied in this paper, along with the modeling assumptions required to model the data, caused differences in the experimental output metrics of interest: product cycle time and tool utilization parameters. Some possible reasons are each simulation package's batching logic, setup avoidance rules, and rework occurrence estimators.

1 INTRODUCTION

The semiconductor industry is a vital part of America's economy. With annual sales totaling billions of dollars, the manufacturers of these semiconductors work in a very competitive and lucrative arena. Though many different varieties of semiconductor chips exist, each manufacturer is faced with the enormous challenge of keeping up with this ever-changing industry's state-of-the-art technology. Products which are deemed advanced or new concepts can be outdated and replaced within two to three years. Companies which are best able to meet future customers' anticipated needs in a cost effective, timely manner will continue to thrive in the years to come.

There currently exists a variety of commercial, corporate, and university developed simulation and analytical tools for the analysis of wafer fabrication facilities (fabs). There have also been a number of strategies and rules developed to control the flow of semiconductor chips or product in a wafer fab. In order to investigate the possible merits of these strategies, many schedulers, planners, and manufacturing engineers have enlisted the aid of discrete event simulation and flow/queue analysis.

Without disrupting the current work-in-process in their own manufacturing facility, planners can perform "what-if" analyses using simulation and queuing models to find out whether a proposed change in the factory's schedule and/or toolset configuration will prove useful or profitable to the company in the long run. However, this type of analysis is a "point in time" look. Once the simulation analysis has been performed, it becomes obsolete as soon as the configuration or product mix of the factory changes. Indeed, this can occur on a daily basis in current fabs. Therefore, care must be taken to ensure the most current information from the fab is present in the simulation or queuing model.

Working with modeling and simulation practitioners from different semiconductor manufacturing companies around the world, it is evident that no standard semiconductor industry simulation package exists. However, the most widely used tools appear to be Tyecin Systems' ManSim/X, AutoSimulations' AutoSched, Systems Modeling's Wafer Fabrication Template, and Chance Industrial Solution's Delphi, which Wright, Williams, and Kelly now licenses as Factory Explorer.

This paper benchmarks the four factory performance analysis packages using actual semiconductor factory data. This data includes a list of the tools used in the semiconductor facility, the different types of products manufactured by the facility, as well as the process steps required to make each of the individual products. The same datasets are input into each package. Once the simulation models have been developed, simulation experiments are run under identical conditions in each package. After running the simulation experiments, a comparison of the output statistics of each of the packages is performed and explanations of both the similarities and differences in the results among the simulation software packages is given.

While there are many other attributes that could be used to compare the packages, only their predicted product cycle times and tool utilizations are examined. Other possible attributes include analysis run times, cost of the packages, ease of use, flexibility, etc. While these attributes were studied, they are not reported in this paper because results in these areas are generally obsolete by the time that they are published. This is due to the fact that most of the suppliers release at least two versions of their software each year. The authors have found, however, that the underlying way that basic elements (e.g. batching, setups, etc.) are handled does not change as often.

2 SIMULATION PACKAGE INFORMATION

The four simulation packages investigated were selected on the basis of their wide-spread use in the semiconductor industry, their ability to allow the user to build simulation models through flat ASCII text files, and their availability to the authors.

AutoSched 4.0 is based on another one of AutoSimulation's modeling packages, AutoMod. AutoSched provides added functionalities which are not present in the standard AutoMod software. However, a user must own the AutoMod software before attempting to run AutoSched. AutoSched allows the users to perform discrete event simulations by simply populating required data fields, such as process flow, tool set, and operator information. Standard "rules" exist for processing jobsteps, but the user does have the ability to write custom step instructions for each individual jobstep to further customize the simulation model. Used in combination with AutoStat, the user can run multiple replications of his or her simulation model to get a better idea of the long run average cycle time of a particular product or the utilization of a certain tool group (AutoSimulations, Inc. 1993).

Delphi 8.20 is a C-based simulator which allows the user to create discrete event simulation models by simply creating a process file and a routing file which contain process flow, tool set, and operator information. Extensive simulation run support is offered in Delphi, with the user having many different command line options at his or her disposal for configuring a particular simulation experiment run or set of experimental replications. Detailed queuing analyses are provided in the Delphi output report, along with product cycle times, queue lengths at each work station, rework information, setup occurrences, and other quantities of interest (Chance 1994). The Delphi simulation engine has been incorporated into Wright, Williams, and Kelly's Factory Explorer.

The ManSim/X 3.4 factory performance simulator was specifically developed to analyze the performance of wafer fabrication facilities by Tyecin Systems. A heavy semiconductor nomenclature exists in the software package which helps the model builder, who is familiar with the jargon of the industry, to build simulation models rapidly. Users can build discrete event simulation models by simply populating required data fields, such as product and process flow information, tool set data, and operator information. Many different selectable output reports are also available to the user which provide a variety of output statistics, including tool, operator, process, and cost information (Tyecin Systems, Inc. 1994).

WFT 1.23 is an application specific template which attaches to Systems Modeling Corporation's simulation tool suite, Arena. Arena allows the user to create simulation models by directly using the SIMAN and CINEMA languages. However, the WFT was developed to help users build simulation models which are specific to the semiconductor industry without having to know the SIMAN and CINEMA simulation languages. The user specifies both the pertinent data to be used in the model, such as resources, process flows, and operators, and the "rules" to be followed at each jobstep. By specifying the data and the model "rules," the WFT modeler can build a model which very accurately depicts almost any operation present in his or her fab. Also, in addition to performing discrete event simulations, WFT has the ability to perform flow and queue analyses on a simulation model, providing the user with a "rough" estimate of the performance of the fab (Systems Modeling Corporation, 1994).

3 SIMULATION MODEL DEVELOPMENT

While each of the simulation packages being investigated was created to simulate operations in a production environment, the way in which these operations are modeled varies among the tools. Many different types of processes exist in a semiconductor fab. A major challenge in trying to create a discrete event simulation model of a semiconductor fab is to accurately emulate the wide variety of operations which take place in the actual fab.

Many specialized aspects of a particular manufacturing process exist which may not be directly modeled in a particular simulation tool: specialized process steps, operator breaks, scheduled downtimes, product- and/or sequence-dependent setups, and company proprietary lot release policies. However, it is the modeler who must make the proper assumptions and decisions to most accurately represent his or her actual fab. Therefore, a modeler must thoroughly understand the following aspects of a factory's operation: load rules and batching, process step rules and delays, wafer rework and scrap, setup, lot releases, sequencing/dispatching rules, simulation clock type, and machine downtimes.

The amount of effort required by a modeler to create and implement his or her simulation model can often determine whether or not the simulation package will again be used in the future. While many simulation packages provide the user with a graphical user interface (GUI), these GUIs often can become tedious and inefficient for building large simulation models. Indeed, pointing and clicking a mouse thousands of times in order to populate dialog box fields is a very time-consuming task.

To this end, the developers of most simulation packages have provided their users with the capability to read in and create simulation models from an ASCII flat file(s) as an alternative to building the model using the program's GUI. The ASCII file model building capability may also prove useful to modelers who have access to their facility's shop floor control system. Data from this system may be downloaded and subsequently formatted by the modeler in order to obtain most of the inputs required by the simulation package.

3.1 Testbed Datasets

A testbed dataset format was developed to provide planners, researchers, and software suppliers with actual data and models that can be used to benchmark their control strategies and software (Feigin et al. 1994). The testbed includes actual manufacturing data from several wafer fabrication facilities, organized into a standard format. The data sets do not include real product names, company names, or other nomenclature that could serve to identify the source of the data.

The goal in development of the fab level data sets was to include the minimum information necessary to model a factory. In support of this goal, the data sets were kept as simple as possible, specifically by including data but not modeling assumptions. The resulting data is given in five ASCII flat files, containing product routings and processing times, rework routings, equipment information, operator data, and product starts information. A comments file also exists which gives information about the dataset, such as tool setup requirements and product-type descriptions.

To date, six testbed datasets have been compiled. In Mason (1995), four of the datasets are modeled in the four simulation packages mentioned above. This paper presents the results for two of these datasets: the simplest dataset and the most complex dataset.

3.2 Modeling Assumptions

Conversion programs were written to convert the testbed datasets into the proper ASCII file formats required by each of the four simulation packages. Since not all of the testbed datasets' fields map directly into all of the software packages, certain modeling assumptions were made to most accurately model the specified data. However, some assumptions were made globally, applying to each of the four testbed datasets as they were modeled in all four simulation packages. The two most important global assumptions are described below.

The time between successive machine failures was assumed to be exponentially distributed, as was the time required to repair a machine. Another global assumption was that product lot starts were made uniformly, with a constant amount of time elapsing between each successive lot release into the factory.

4 DESIGN OF EXPERIMENTS

After the testbed datasets had been properly converted into simulation models for each one of the four packages being investigated in this paper, simulation replications were performed with all four testbed datasets. Prior to performing a simulation replication, Delphi performs a capacity analysis and calculates the maximum component input rate for each of the different product types which are present in the simulation data. This value was used in all models as the maximum allowable input rate. It was decided to perform experiments with three different levels for wafer starts per week, in terms of a percentage of the maximum allowable rate: 55%, 75%, and 95% of the maximum component input rate.

Each simulation model was run for a period of 1,000 days, with 25% of the run (250 days) being used as a "warm-up" period to account for initialization effects of starting the model with an empty and idle fab. After viewing WIP vs. Time curves for each of the simulation models, it was decided that 250 days was a sufficient length for the warm-up period, as each of the simulation models had achieved steady-state by this time. Deterministic lot releases into the fab were used in each of the simulation models, with each product type having its own unique, constant interarrival time.

For each of the three start rate levels, twenty replications were performed so that long term averages could be calculated for product cycle time and resource utilization, the two main quantities of interest in this paper. Once these quantities were obtained, comparisons were made between the four simulation packages for the same wafers per week start rate.

5 EXPERIMENTAL RESULTS AND ANALYSES

For both of the datasets investigated, the experimental tool results are given for the three tools which had the most number of lots waiting to be processed at the work station, i.e. tools with the largest average waiting queues. The percentage of time each tool spent processing lots (busy), incurring a setup (setup), failing due to unscheduled machine breakdown (down), and waiting to process a lot (idle) is given in the experimental tool data table.

Then, the average cycle time for each product type over the 20 simulation replications is given for both of the start rate levels. Some possible reasons are given for the similarities and differences between the outputs of the simulation packages being investigated for both of the datasets.

5.1 Dataset Four

The fourth testbed dataset is comprised of seven products. Products 1, 2, and 5 all share an identical process routing, each following the same identical set of 92 process steps. The remaining four products, 3, 4, 6, and 7, share an identical process routing as well, with each product following a 19 process step flow. Therefore, there are a total of 111 process steps in this dataset, with a total of 35 different tool groups being present. There are no operators in this dataset, no probability of rework at any of the jobsteps, and setup is not required at any process step. This dataset is the smallest and simplest of all of the testbed datasets.

All of the simulation packages being benchmarked were able to successfully read-in and run this dataset. Table 1 shows the experimental tool utilization data

Table 1: Dataset Four Experimental Tool Data

	AutoSched						Delphi					
	Tool	Busy	Setup	Down	Idle	Avg Q	Tool	Busy	Setup	Down	Idle	Avg Q
55%	DFC4	50.6	0.0	7.7	41.7	0.5	DFB3	77.2	0.0	7.2	15.6	0.5
	DFB3	78.7	0.0	7.5	13.8	0.5	DFC4	50.6	0.0	7.8	41.6	0.5
	ASM2	35.5	0.0	14.3	50.2	0.3	ASM2	35.5	0.0	16.8	47.7	0.3
75%	DFC4	69.0	0.0	7.5	23.5	1.5	DFC4	69.0	0.0	8.0	23.0	1.4
	DFB3	87.6	0.0	7.6	4.7	0.8	DFB3	89.3	0.0	7.1	3.6	0.7
	ASM2	48.4	0.0	14.5	37.1	0.7	ASM2	48.4	0.0	16.4	35.1	0.7
95%	DFC4	87.4	0.0	7.3	5.3	5.8	DFC4	87.4	0.0	7.6	5.0	6.6
	ASM2	61.3	0.0	14.1	24.5	1.5	ASM2	61.3	0.0	16.3	22.4	1.7
	D1_9	60.3	0.0	12.2	27.5	1.1	QLES	76.0	0.0	11.8	12.2	1.4
	ManSim						**WFT**					
	Tool	Busy	Setup	Down	Idle	Avg Q	Tool	Busy	Setup	Down	Idle	Avg Q
55%	DFB3	93.1	0.0	6.7	0.2	1.9	DFC4	50.6	0.0	7.3	42.1	0.5
	DFC4	50.6	0.0	7.3	42.1	0.4	DFB1	48.8	0.0	7.3	43.9	0.3
	ASM2	35.5	0.0	14.2	50.3	0.3	ASM2	35.5	0.0	14.3	50.2	0.3
75%	DFB3	93.2	0.0	6.7	0.1	2.4	DFC4	69.0	0.0	7.3	23.7	1.3
	DFC4	69.0	0.0	7.3	23.7	1.2	QLES	64.7	0.0	10.8	24.5	0.7
	ASM2	48.4	0.0	14.1	37.5	0.5	ASM2	48.4	0.0	14.8	36.7	0.6
95%	DFC4	87.4	0.0	7.2	5.4	5.3	DFC4	87.4	0.0	7.6	5.0	6.7
	DFB3	93.3	0.0	6.6	0.1	3.0	QLES	82.0	0.0	10.7	7.4	3.6
	ASM2	61.4	0.0	14.0	24.6	1.3	ASM2	61.3	0.0	15.0	23.6	1.3

obtained from the replications of Dataset Four. The results show good agreement between the downtime percentages of the four simulation packages. As expected, the start rate of the simulation model does not have an effect on machine downtime, as this is a random variable which is independent of start rate. However, in some simulation packages, such as WFT, the user can model machine downtimes based on the total number of wafers processed by a machine. This type of downtime is dependent on start rate by definition and would increase with start rate, but was not used in this benchmarking study.

The majority of the tools in the top three bottleneck list have identical busy percentages as calculated by each simulation package. The factory's bottleneck, tool DFC4, is correctly identified by each of the packages at the 95% start rate level, with its busy, down, and idle percentages, along with average queue length being similar for all four of the simulation packages. Each package obtains similar results for the factory's bottleneck, DFC4, with the range of values obtained for the tool's average queue length being less than 1.5 lots between all of the packages, from 5.3 lots to 6.7 lots. In fact, the range of average queue lengths for each of the tools is very small, within one to one and one-half lots. The simplicity of this dataset, having no rework, no operators, and no setup, allows each of the simulation packages to compute similar output performance measures for the given input data.

Table 2 shows the experimental cycle time data obtained from the simulation replications of this dataset. Each of the simulation packages computes a similar cycle time for each of the different start rates being investigated. In both AutoSched and ManSim/X, it is evident that the tie-breaking rule used to select between identical lots at a given step in the process flow is FIFO, as the product types which have identical process routings experience increasing cycle times according to their position in the products list. Product 5 has a longer cycle time than does product 2 because product 2 lots are released into the factory before product 5 lots, therefore always being the "lead" product type. Delphi and WFT, however, do not seem to have this sort of tie-breaking rule, as identical product's cycle times are not arranged in ascending order. However, all four packages are comparable in terms of their estimations of cycle time.

5.2 Dataset Five

The fifth testbed dataset is made up of 21 different product types. The 21 different process flows range in size from 108 to 259 steps, with the total dataset containing 3,798 jobsteps and 85 different tool groups. This dataset is the largest of the testbed datasets studied in this paper. There are four operator groups present in this dataset, but no rework is required at any of the jobsteps. Setup is required on some of the tools in this dataset, with two of the tools, 19 and 66, requiring setup avoidance rules.

One type of setup avoidance rule specifies that if a lot arrives at the work station which requires the same setup that the machine is currently configured for, process that lot immediately. Another rule uses the above logic, but also places a maximum time delay on a lot, saying that if a lot has been waiting longer than this specified maximum time, re-setup the machine for the new lot and process that lot. This could play an important part in determining when particular lots begin processing and how fully the machine is utilized.

All of the simulation packages being evaluated successfully read this dataset into memory except the evaluated version of WFT, Version 1.23. WFT was unable to read Dataset Five into memory due to its size and/or the program's memory limitations. However, WFT's author, Systems Modeling Corporation, has

Table 2: Dataset Four Experimental Cycle Time Data

	AutoSched			Delphi			ManSim			WFT		
Product	55%	75%	95%	55%	75%	95%	55%	75%	95%	55%	75%	95%
1	152.6	167.0	198.2	148.5	162.5	204.1	150.9	162.8	195.1	148.1	161.9	217.3
2	157.0	170.6	201.3	148.5	162.4	204.1	155.8	164.8	198.0	148.3	161.9	217.3
3	23.0	24.1	24.8	24.8	26.2	28.4	21.4	22.2	23.7	25.4	27.3	32.9
4	24.5	25.5	26.4	24.6	26.3	28.3	22.5	24.0	26.3	25.4	27.1	32.4
5	163.6	175.5	205.5	148.5	162.6	204.1	157.4	167.1	200.6	148.5	161.9	217.4
6	25.7	26.3	27.9	24.7	26.2	28.3	22.5	24.8	27.6	25.6	27.3	32.6
7	26.5	27.6	28.8	24.7	26.2	28.5	23.6	26.5	29.7	25.2	27.0	32.3

All times are specified in hours.

stated that this size and/or memory limitation has been eliminated in later versions of the software package. Table 3 shows the experimental tool data obtained from the simulation replications of this dataset. Batching tools dominate this dataset, with all of the tools being identified as bottlenecks by AutoSched and ManSim/X belonging to this tool type. The only serial tool which is present on the bottleneck tool list is tool 19 on the Delphi list.

The setup required of tool 19 places it on Delphi's list, as this setup directly impacts the tool's average queue length. However, this tool does not appear in the lists of the other two packages. Tool 19 has a group-dependent setup associated with it which was modeled in Delphi, but not in AutoSched or ManSim/X, as group-dependent setups were not modeled in the latter two simulation packages.

A particular lot may be assigned a group ID which is then used for determining the need for a specific type of tool setup. This group-dependent setup is 45 minutes in duration, which definitely would increase the number of lots and the amount of time which they have to wait at the tool before it is properly setup. This type of setup obviously plays an important part in determining when particular lots are worked and how often the machine is busy processing actual lots.

The experimental cycle time data obtained from the simulation replications for Dataset Five are presented in Table 4. Surprisingly, AutoSched shows little variation in cycle time as the start rate is increased, especially when compared to the other simulation packages. The cycle times calculated by AutoSched for the 55% input rate case are very close to the values arrived at for the 95% case. While the AutoSched bottleneck tool lists show relatively small average queue lengths, the batching efficiencies, a measure of how "full" a batch is, are significantly increasing in AutoSched, allowing for more products to be processed at a time, thus keeping the overall product cycle time down. This, combined with the fact that group-dependent setups were not modeled in AutoSched and ManSim/X, keeps the two packages' cycle times below the results obtained in Delphi.

The ManSim/X and Delphi cycle time numbers increase significantly when going from the 75% to the 95% start rate. Setup on tool 19 is driving Delphi's cycle times up at higher input rates, especially due to the previously mentioned group-dependent setups, while tool 77's zero idle time is causing the increased cycle time in the ManSim/X results. The cycle time results also follow the expected trend based on the average queue sizes present in each of the simulation models according to Little's Law: increased queue lengths (WIP) leads to an increase in cycle time.

Another possible reason for Delphi having longer cycle times than the other simulation packages is that Delphi has both group- and sequence-dependent setup modeling capability. The testbed datasets used in this benchmarking study contain both types of setup. Consequently, both types of setup were modeled in Delphi. AutoSched's and ManSim/X's simulation models contain sequence-dependent setups, but not group-dependent setups. This suggests lower cycle times for these two packages due to the decrease in the amount of setups which must be performed.

Table 3: Dataset Five Experimental Tool Data

	AutoSched						Delphi						ManSim					
	Tool	Busy	Setup	Down	Idle	Avg Q	Tool	Busy	Setup	Down	Idle	Avg Q	Tool	Busy	Setup	Down	Idle	Avg Q
55%	77	91.8	0.0	5.0	3.1	1.4	32	96.0	0.0	4.0	0.0	2.1	32	95.8	0.0	4.1	0.1	3.9
	32	95.3	0.0	4.6	0.1	1.1	34	93.3	0.0	4.4	2.3	1.7	77	94.3	0.0	5.0	0.7	3.9
	75	82.8	0.0	5.0	12.2	1.0	89	96.6	0.0	0.0	3.4	1.5	75	90.6	0.0	5.0	4.4	2.2
75%	77	93.9	0.0	5.0	1.1	2.1	19	38.0	51.4	6.7	3.9	4.3	77	95.0	0.0	4.9	0.1	6.2
	75	87.4	0.0	5.0	7.6	1.6	32	95.9	0.0	4.0	0.2	2.7	32	95.6	0.0	4.1	0.3	5.6
	32	95.3	0.0	4.6	0.1	1.5	89	98.5	0.0	0.0	1.5	2.4	89	97.4	0.0	0.0	2.6	4.9
95%	77	94.6	0.0	5.0	0.4	3.2	19	48.1	45.1	6.7	0.1	12.2	77	95.1	0.0	4.9	0.0	9.2
	75	91.0	0.0	4.9	4.1	2.3	4	89.0	0.0	4.3	5.8	5.9	89	92.8	0.0	0.0	7.2	8.7
	32	95.6	0.0	4.4	0.1	1.9	77	94.4	0.0	5.2	0.4	4.3	32	95.1	0.0	4.1	0.8	7.7

Table 4: Dataset Five Experimental Cycle Time Data

	AutoSched			Delphi			ManSim		
Product	55%	75%	95%	55%	75%	95%	55%	75%	95%
1	163.3	164.9	166.0	272.6	303.8	377.7	173.2	192.5	272.2
2	177.6	180.1	181.6	329.2	358.2	428.2	190.9	207.1	246.0
3	166.2	170.0	172.6	285.5	319.4	398.7	181.2	202.3	287.6
4	198.4	198.4	202.2	342.7	375.2	452.2	221.5	237.9	286.6
5	183.2	185.8	188.5	345.9	390.3	488.7	200.9	223.0	327.2
6	197.0	198.1	199.6	418.1	460.4	557.0	223.9	238.3	285.9
7	200.5	202.0	205.8	331.6	371.4	466.4	222.5	246.3	313.5
8	210.7	212.9	214.5	399.7	436.9	530.5	233.7	251.5	298.1
9	196.6	197.7	197.6	266.2	301.1	384.1	220.3	242.6	354.7
10	154.0	156.8	159.6	304.6	343.4	434.3	171.2	191.9	277.8
11	283.8	291.4	303.2	369.1	405.9	493.9	397.8	452.6	532.9
12	176.2	179.3	183.6	441.4	488.5	615.6	198.3	222.2	284.2
13	292.5	302.3	315.9	291.0	325.9	409.7	390.6	442.2	540.8
14	187.9	190.5	193.3	350.6	384.8	468.3	210.4	225.6	267.8
15	283.3	293.4	307.0	423.0	471.8	598.4	359.3	395.4	454.8
16	351.1	361.6	379.7	498.6	551.9	689.6	502.7	583.1	758.1
17	306.9	316.4	330.7	445.9	493.0	622.7	401.3	452.6	542.2
18	99.6	101.4	101.9	247.6	272.5	335.6	107.7	120.0	147.4
19	104.0	109.2	110.3	213.8	243.1	302.4	116.9	135.4	204.1
20	103.6	103.8	106.0	254.9	280.9	343.4	113.2	125.4	156.7
21	110.8	111.4	113.2	208.0	233.4	295.8	119.9	133.6	197.3

All times are specified in hours.

6 CONCLUSIONS AND COMMENTS

Dr. George Box is widely credited with saying that, "All simulation models are wrong, some are useful." Simulation models are "wrong" in the sense that they are only computer representations of real systems. Through the use of dispatching rules, random number generators, and sampling from statistical distributions, simulation models attempt to imitate the actual dynamic processes found in a manufacturing facility. "Useful" can be interpreted as does the simulation model accomplish the tasks which the modeler intended it to perform? Does it answer the question(s) which the modeler was asking? The accuracy and functionality of a simulation model determines both how wrong is "wrong" and how useful is "useful."

A simulation package can produce different results between replications of the same model, due to its stochastic nature and the enormous number of events which occur in the model: lot moves, random number sampling, resource contingencies, lot rework and scrap, and forming of batches. The different ways in which each of the simulation packages studied in this paper handle these events, coupled with the modeling assumptions required to model the testbed datasets, caused differences in the experimental results. While the simulation packages' results are comparable for the smallest, simplest dataset, Dataset Four, a wide range of values for both product cycle time and machine utilization were observed among the software packages for Dataset Five.

Some possible reasons for these discrepancies which were discussed are each simulation package's batching logic, setup avoidance rules, and rework occurrence estimators. The results confirm that batching efficiency, a measure of how "full" each batch is, increases with start rate as expected. A noticeable increase in the batching efficiencies of the tools helped to keep the tool's busy percentage fairly constant and average queue lengths at the batching tools somewhat low. This also in turn can help to shorten the product's average cycle time as more lots experience concurrent processing with other lots at one or more places in the process flow. The setup avoidance rules used by the simulation packages dictate the frequency of required setups, thus affecting both machine utilization and overall product cycle time. Finally, the rework occurrence estimators used by the simulation packages, being either deterministic or probabilistic, directly affect the number of lots which must pass through a given job step and therefore, a given work station.

This benchmarking project was definitely a larger undertaking than at first expected, mostly due to the amount of time required to decipher the nuances of each simulation package and the huge amounts of data which must be accounted for in a simulation model. Due to the size of simulation model data, there is no doubt that simulation packages which allow the user to ASCII load-in simulation model data are the only practical way to build factory-sized simulation models.

Even after the model data has been read into the simulation package, great care must be taken during the large amounts of time necessary to check the data's integrity and the simulation model's assumptions into the software package. As described in section 3.1, various assumptions were required when building the simulation models for the software packages being studied in this paper. A careless oversight or misunderstanding of the simulation package's assumptions and/or methodologies can produce simulation results which are of no use to the modeler and might even have detrimental effects on a factory if used to make policy decisions.

This paper represents the culmination of one and one-half years of modeling, simulation, experimentation, and analysis. During this time, newer releases of each of the simulation packages evaluated in this paper have become available to modelers, in effect making the results in this paper obsolete. This paper is not intended to provide "the answer" for all of the questions associated with the modeling and simulation of manufacturing facilities. Rather, it is a point-in-time comparison study of the basic logic in four state-of-the-art simulation packages which have been used to model semiconductor fabs. The author encourages the reader to perform additional simulation experiments to gain a better understanding of both his or her own modeling needs and requirements, and to get a feel for the types of simulation packages which are available to modelers today.

REFERENCES

AutoSimulations, Inc. 1993. *AutoSched and the Simulator System.* Bountiful, Utah: AutoSimulations, Inc.

Chance, F. 1994. *Delphi: Integrated Capacity Analysis and Factory Simulation.* Version 8.20. Dublin, California: Chance Industrial Solutions.

Feigin, G., J. W. Fowler, J. Robinson, and R. Leachman. 1994. Semiconductor Wafer Manufacturing Data Format Specification. SEMATECH, Austin, Texas.

Mason, S. 1995. A Comparison Study of Four Factory Performance Simulators. Master's Report, Mechanical Engineering Department, The University of Texas at Austin, Austin, Texas.

Systems Modeling Corporation. 1994. *Wafer Fabrication Reference Guide.* Sewickley, Pennsylvania: Systems Modeling Corporation.

Tyecin Systems, Inc. 1994. *ManSim/X User's Manual.* Release 3.4. Los Altos, California: Tyecin Systems, Inc.

AUTHOR BIOGRAPHIES

SCOTT J. MASON is an Operational Modeling Engineer at Advanced Micro Devices (AMD) in Austin, Texas, specializing in cost and capacity modeling. He has a B.S. degree in Mechanical Engineering and an M.S. in Operations Research and Industrial Engineering, both from The University of Texas at Austin. Prior to working at AMD, he was an independent simulation consultant to SEMATECH.

PAUL A. JENSEN is the Cullen Trust Professor of Mechanical Engineering. He has B.S. and M.S. degrees in electrical engineering from the University of Illinois and the University of Pittsburgh, and the Ph.D. in operations research and industrial engineering from Johns Hopkins University. He has taught at the University of Texas since 1967. In addition to prior employment with the Westinghouse Electric Corp., he has practical experience from a number of consulting engagements. He is the co-author with J. Wesley Barnes of the book Network Flow Programming, the winner of the 1980 Book-of-the-Year Award from the Institute of Industrial Engineers.

JOHN W. FOWLER is an Assistant Professor in the Industrial and Management Systems Engineering Department at Arizona State University. Prior to his current position, he was a Senior Member of Technical Staff in the Modeling, CAD, and Statistical Methods Division of SEMATECH and an Adjunct Assistant Professor in the Graduate Program in Operations of the Mechanical Engineering Department at The University of Texas at Austin. He received his Ph.D. in Industrial Engineering from Texas A&M University and spent the last 1.5 years of his doctoral studies as an intern at Advanced Micro Devices. His research interests include modeling, analysis, and control of manufacturing (especially semiconductor) systems. He is a member of IIE, INFORMS, and SCS.