Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

SIMULATING TEST PROGRAM METHODS IN SEMICONDUCTOR ASSEMBLY TEST FACTORIES

Chad D. DeJong

Intel Corporation
5000 West Chandler Boulevard
Mail Stop CH3-113
Chandler, AZ 85226, U.S.A.

ABSTRACT

Significant opportunities for improvement in semiconductor Assembly/Test (A/T) manufacturing reside in the Test areas. These Test areas can very often be the system constraint, due to complex testing policies, bin-to-order mapping, and cost. A very difficult problem for both researchers and manufacturers is to determine the best methods for assigning test programs for lots on these test equipment. To answer these problems, Intel has produced dynamic discrete event simulation models that consider multiple wafer types, multiple end products, multiple test program methods, and binning policies of end products according to the tested performance of the die. This model does not require modeling specific manufacturing equipment and operator activities, only detailed logic of test program and binning policies. The quantitative output data from this model provides the relative decision support necessary to determine what methods work best for Intel, given other costs and business drivers.

1 PROBLEM STATEMENT

This paper is motivated by the problem of assigning test programs for lots to use on assembly/test manufacturing test equipment, which is often the system constraint. This problem requires a decision support tool that can predict quantitative success indicators for multiple test program assignment methods. The test program assignments, along with binning policies, will directly impact Intel's success in satisfying customer orders with the right product on the date requested.

2 INTRODUCTION AND BACKGROUND

Semiconductor manufacturing consists of four major sections: wafer fabrication, wafer sort (or probe), assembly and test. Wafer fabrication and sort, typically called front-end, is where integrated circuits (die) are developed on a silicon wafer through several hundred process steps. At sort, the die are tested and marked electronically. During A/T, typically called back-end, the good die are separated, packaged, run through various testing procedures, and finally packed for shipment. At this point, the die are ready to be shipped to customers and/or warehouses. This paper focuses on back-end manufacturing, and specifically, how lots can best be routed through the test operations. Intel primarily uses a functional layout for its back-end tools (Hilton et al., 1996).

Productivity improvements in the test area are very complex. The test area is very often the system constraint, depending on product mix. There are multiple reasons for this, including complex test program policies, binning policies, and of course cost. The test programs can be described as follows. Each lot is assigned exactly one test program, which is the range of speeds against which each lot or die will be tested. During test operations, the lot is tested against each speed in the test program, and the result is a match between the lot performance and a speed (bin) within the test program. The test program setting depends on a variety of factors, such as the lot and die types, current and future customer orders, and current work-in-process (WIP) in line. Binning policies define to what speed the die is actually assigned. It depends on the results of the testing, and customer orders pending. Test programs and binning policies directly impact Intel's on-time delivery (OTD) indicators, which measure success in satisfying customer orders.

Matching product to customer orders in a timely manner is an obvious priority for semiconductor factories. In an environment where customer lead times are continuously decreasing, the two major hurdles are marketing projections and manufacturing variability. Test programs and the timing of assigning them have implications in both regions. During the time a lot spends in assembly processing, customer orders may change. Also, due to natural manufacturing variability, a single lot may yield the same quality microprocessor, but at a range of bins, or clock speeds. While silicon manufacturing is a precise process, it is not exact. These two hurdles make it very difficult to produce exactly to customer orders, and increase the importance of the test program assignment method.

The problems of satisfying customer orders within a typical A/T plant has received significant attention in recent years, as product offerings have become increasingly diverse, and customers continue to demand shorter lead times. It can be decomposed into two parts, selecting which orders to fill, and assigning die to orders (Fowler et al., 2000 and Knutson et al., 1998). The two parts are related to classic knapsack problems. However, the solutions provide local or "snapshot" answers. Extending the problem out over multiple time periods or across multiple products, it has been shown that bin covering in these cases is NP-hard (Carlyle, 1999). Intel, along with industry practitioners, must resort to heuristics and other methods for satisfying daily demand. This daily demand is scrutinized frequently, through backward product mapping. This software determines die-to-order mappings and inventory release schedules from order due dates, and is used to reevaluate Intel's manufacturing execution plans.

From a modeling standpoint, Intel increasingly uses modeling and simulation methods for operational decision support. The method presented here uses a discrete event simulation model to compare these heuristics and methods, and their ability to improve OTD relative to each other. The model considers multiple wafer types from front-end manufacturing, multiple end products, and the binning policies of end products according to the tested performance of the die.

Two test program assignment methods were compared on the basis of timing (when should the test program be set?). The first test program method (TP1) sets the test program for a lot as soon as it is dispatched from an initial inventory point to begin processing in the A/T process. The second test program method (TP2) is set only when the lot has actually reached the test operation, after all assembly processing is complete. Please see Figure 1 for a graphical description.

In either case, it is important for these assignments to be automated, since this is a frequent activity for the factory planners. The actual assignment of a specific test program is done via reference tables, obtained from heuristic or other empirical methods within the factories. The test programs are limited to four bins (speed ranges), although a specific die may support a larger range of speeds.

Given the above business environment, some would conclude that setting the test program as late as possible (using TP2) will allow the most flexibility in responding to market and manufacturing variability. However, a potential weakness of TP2 is that when determining dispatch requirements for future demand, the planning system does not have knowledge of any defined test program assignment. Therefore, the planning system can only make decisions based on forecasted natural binning. The answer is certainly non-trivial. The problem remained to quantify what type of relative OTD performance either method would provide. The quantitative data would ultimately drive what method works best, given costs and other business drivers.

3 MODELING METHODOLOGY

3.1 Project Approach

The simulation team worked directly with backend planners and manufacturing personnel to determine project deliverables, timing requirements, and level setting on the

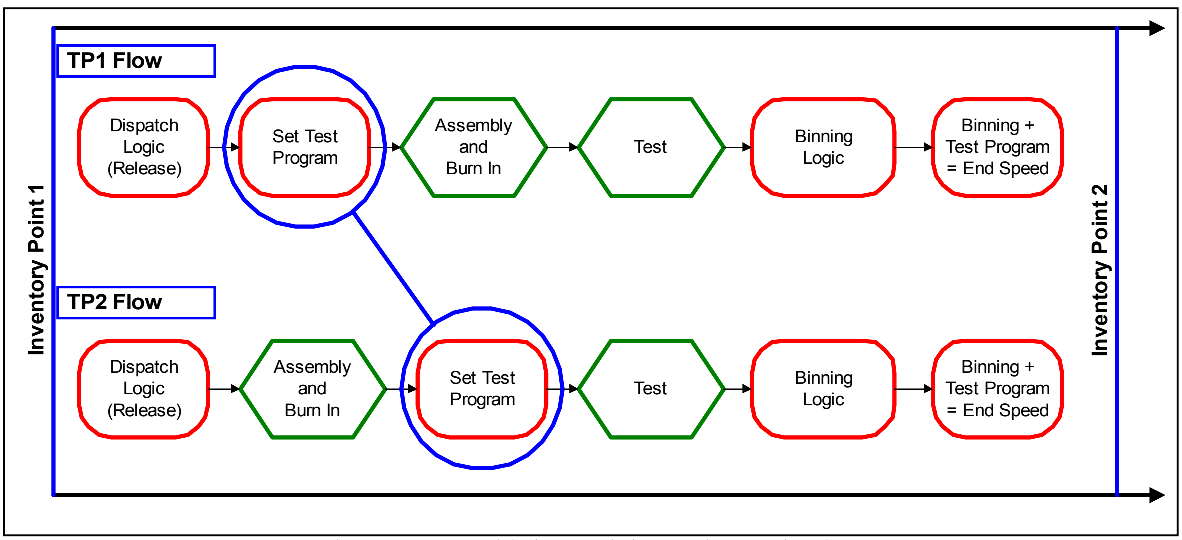

Figure 1: Assembly/Test High Level Generic Flow

model granularity. The team decided on a model that maintained simplicity, while still providing enough detail to provide credible results to aid in decision support (Domaschke et al., 1998).

The specific back-end operations with regard to manufacturing equipment and operations were not modeled. Instead, the project team decided to treat these as stochastic model parameters, and focus on die-to-order mapping, WIP release, and WIP binning policies at two critical manufacturing steps. The steps between these points are constant with a stochastic throughput time, and were not required to provide relative decision support between competing policies.

3.2 Software

Due to the simplistic method for modeling the processing operations, the only real requirement was a language which 1) could comprehend the logic of the competing policies, 2) was easy to use, 3) very fast, and 4) inexpensive. The team decided on Microsoft Visual Basic 6.0. An added benefit to using this software is that it easily interfaces to spreadsheet inputs for all team members to use.

The challenge with using this software was updating the simulation clock throughout several product mapping reevaluation cycles, across several weeks of projected customer demand orders. This was handled through a time increment input from the model user at the beginning of the simulation, and a demand order file input.

3.3 Data Requirements and Outputs

Although processing tools and operators were not modeled, there were still significant data inputs required to drive the logic of the dispatching, testing, and binning policies.

Inputs are 1) the various product sets from front-end manufacturing, 2) test program rules, 3) binning probabilities and rules, 4) customer order files, 5) forecasted and historical bin ranges and percentages for each product set, and 6) component part yield percentages and throughput time distributions. A design of experiments analysis was set up to compare two types of customer order files (weekly and daily), two test program assignment look-up tables (based on factory heuristics), and finally the TP1 vs. TP2 assignment timing.

The outputs were much simpler. The project team was primarily concerned with one success indicator, OTD performance. However, we also tracked WIP that, at end of processing, ultimately had no specific customer orders to satisfy (these die would have to be held in inventory). As a secondary metric, we would try to minimize these situations.

3.4 Model Logic

The general model logic can be thought of as two parts – setting up the model run, and executing the model. To set up the model, the user must input four pieces of information through graphical user interfaces, 1) the test program assignment method, 2) the backward product mapping reevaluation frequency, 3) the number of simulation replications, and 4) the level of debugging print messages to print out. To execute the model, the simulation simply runs through the required replications, resetting model variables and data fields appropriately. Through any one product mapping reevaluation, four things must be checked.

First, the initial inventory point must be verified, and all inventory levels are updated based on incoming component parts that may have arrived since the last reset.

Second, all WIP in the model that is ready to run through the tester (based on its stochastic processing time) is checked, and logically executed through the tester operations, based on its test program setting and binning policies. After the testing logic is complete, the die are logically routed to appropriate finish inventory points.

Third, the model iterates through all customer orders, and determines which require WIP to be dispatched from the initial inventory point, based on expected A/T throughput time parameters. Before doing this, the model first checks existing WIP in the line (and projected binning), and die already existing in inventory that could satisfy this demand. If no inventory or WIP exists, the model selects which component parts in the initial inventory point is best suited to meet this demand, checks that enough inventory exists, and then dispatches the appropriate amounts. If insufficient inventory exists to meet this demand at this time, then a demand backlog is recorded. This backlog will be met at a later date.

The fourth and final activity is to check all customer orders now due, and determine which orders can be met through existing finished inventory levels. At this time, OTD performance indicators are recorded for each customer order.

3.5 Model Verification

The project team worked with planners to verify model logic and results, obtain realistic customer order files and test program look-up tables, plan model improvements, and look for follow-up analyses.

During the simulation itself, three key experimental conditions must be agreed upon by the project team: how long to run the simulation, when to begin collecting output metrics (determine the warm-up period), and number of simulation replications. Of course, the end recommendations depended on the complexity of the model input parameters. In general, the warm-up period was several weeks, the model run length was several months, and the replications ranged from thirty to fifty.

4 RESULTS AND IMPLICATIONS

In the first set of experiments, the team decided to test two policy inputs against customer orders. The two policy inputs were: the timing of the test program assignment (TP1 vs. TP2) and the test program determination method (LT1 vs. LT2). The customer order files also had two types, weekly vs. daily. A design of experiments was formulated with three binary design parameters, providing eight model run results.

The first step was to analyze the impact of when to assign the test program, or compare TP1 vs. TP2 method. Again, TP1 assigns the test program to the lot when dispatched from inventory, while TP2 assigns the test program only after assembly processing. The OTD performance indicator was compared for both test program lookup tables and both customer order file types. Figure 2 clearly shows that TP2 delivers better OTD performance in every scenario. This benefit is more pronounced when using more granular customer order schedules, regardless of the assignment lookup table used. The model shows that the benefits associated with TP2, such as allowing the test program to reflect changes in WIP and customer orders, outweigh the negative of not having test program information of the lot while it completes assembly and burn-in operations.

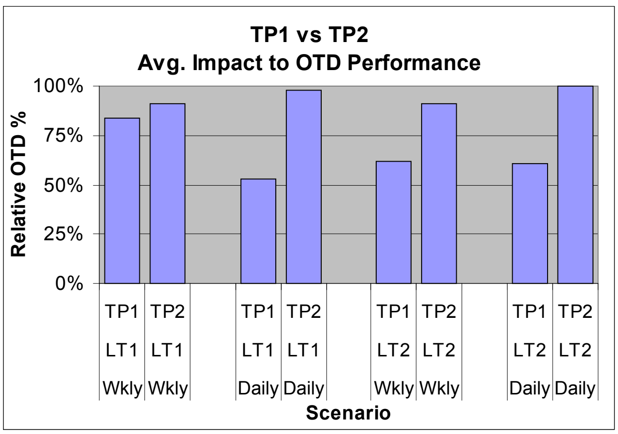

Figure 2: TP1 vs. TP2 Impact to OTD Performance

Next, the model reported OTD performance keeping all variables fixed except for the customer order granularity (daily vs. weekly). The results are shown in Figure 3. In the weekly scenario, most, but not all, customer orders occurred on one particular day of the week. In the daily scenario, the orders were not truly daily, but they did occur at various days throughout any given week. The customer order granularity greatly impacts the model in terms of being able to look ahead to future demands for planning purposes.

Figure 3 indicates the extent of OTD performance sensitivity to the customer order files. When test programs are assigned as they approach the tester (TP2), results are better overall. Again, the model shows a pronounced benefit to using the more granular customer orders (daily) along with the TP2 method. However, if test programs are assigned early (TP1), the result is inconclusive, depending on the assignment lookup heuristic used.

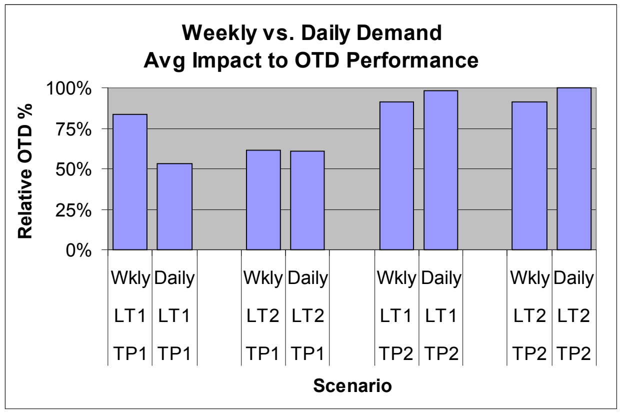

Figure 3: Customer Order Granularity Impact to OTD Performance

Next, the model reported OTD performance keeping all variables fixed except for the test program heuristic lookup method. The results are shown in Figure 4.

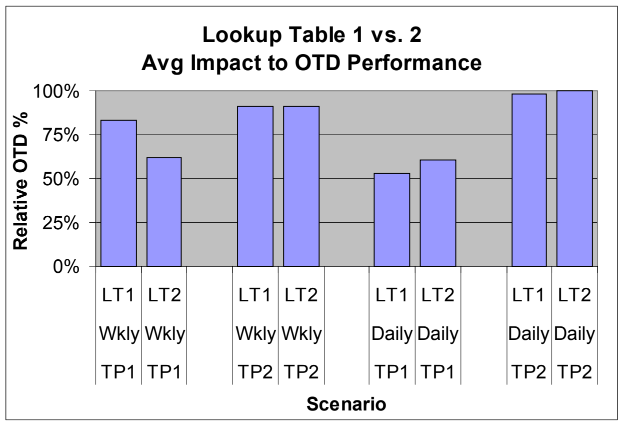

Figure 4: Test Program Lookup Impact to OTD Performance

Figure 4 indicates no clear winner for the best lookup table heuristic for assigning test programs, regardless of when this happens, or what format the customer orders are in. Also, there was not much difference in the OTD results for all comparisons. This was not a surprise to the team, as the two test program lookup methods were very similar except in a few situations. This would indicate there are opportunities for improvement in these heuristics.

The next set of experiments involved analyzing the lot binning policy decisions. An example of this concerns looking ahead to future demand. The problems can be stated as such: how far ahead in the future should customer orders be considered when assigning WIP at the tester equipment to bins? Does it make sense to look ahead several days, or will that simply cause more thrashing of binning assignments? Are there sensible cut-off points for looking ahead to future demand, depending on the nature of the customer orders?

Essentially, Figures 5 and 6 indicate that there is a benefit to looking ahead to future demand. As expected, the model shows that the benefit is more linear as the customer orders become more granular. Figure 5 indicates that there were very few customer orders between two and six days after the standard order day, so there is no incremental benefit in considering these days. Analyzing the customer order input file to the model verified this result. Figure 6 shows that there is no clear cutoff point in considering future customer demand, but the benefit is roughly linear.

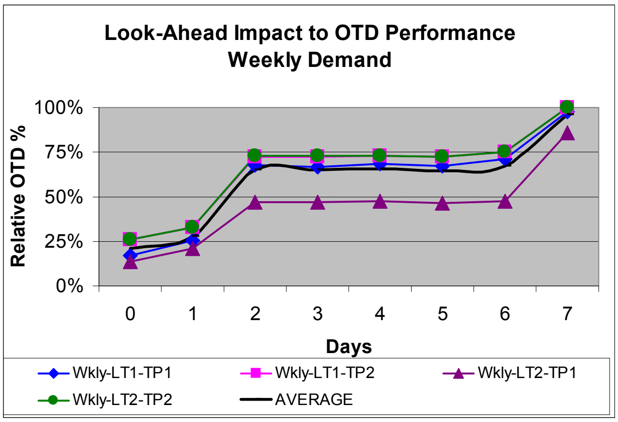

Figure 5: Look-Ahead Benefit to OTD Performance with Weekly Customer Orders

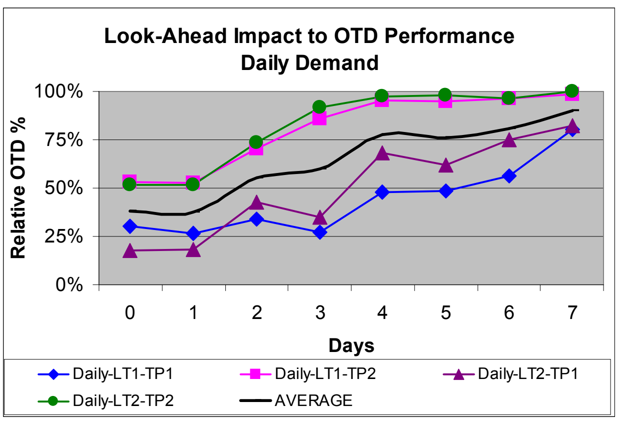

Figure 6: Look-Ahead Benefit to OTD Performance with Daily Customer Orders

5 CONCLUSIONS

The model and modeling approach provided the capability to answer test program assignment and binning policy level questions, as intended. We were able to develop an approach for answering these types of questions by simulating the processing equipment, operators, and materials as "black-box" stochastic entities, while using detailed decision logic functions at critical points in the overall process flow. Granted, this does not allow Intel to make decisions based on tool or functional level criteria, such as throughput time, WIP variability, operator loadings, or a number of other important metrics. However, it does provide the relative decision-making ability required for this analysis, and opportunities for much more policy decision support.

Specifically, this model clearly indicated direction for when to assign the test programs to lots dispatched to begin processing in A/T. The benefit to postpone this assignment until the lots have reached the actual test equipment is most pronounced when customer order files are more granular. The model also indicates that the two program assignment heuristics modeled are relatively insensitive to assignment timing, and customer order files. In terms of WIP binning policies, the model was able to quantify the benefit of looking ahead to future demand, before assigning WIP to specific bins, or speed ranges.

6 NEXT STEPS

As always, the model results presented here are subject to the inputs. We have shown OTD performance is sensitive to customer orders. Clearly, Intel would like to analyze additional customer order what-if scenarios, as these do change quite regularly. Moreover, the test program assignment heuristics presented indicated some stability across policy changes, but also opportunity for improvement. The model format allows the flexibility to add logic here.

ACKNOWLEDGMENTS

The author would like to thank the following individuals, who helped this work come to fruition: Mike O'Brien, German Ham, and Joaquim Guedes for conceptual brainstorming, model generation and verification assistance.

REFERENCES

Carlyle, M., Knutson, K. and Fowler, J. 1999. Bin covering algorithms in the second stage of the lot to order matching problem. Department of Industrial Engineering, Arizona State University.

Domashke, J., Brown, S., Liebl, F. and Robinson, J. 1998. Effective implementation of cycle time reduction strategies for semiconductor back-end manufacturing. *Proceedings of the 1998 Winter Simulation Conference*, 985-992.

Fowler, J.W., Knutson, K., and Carlyle, W.M. 2000. Comparison and Evaluation of Lot-To-Order Matching Policies for a Semiconductor Assembly and Test Facility. *International Journal of Production Research*, 38(8):1841-1853.

Fowler, J.W., Aguilar, R.A. and Flores, C.A. 1997. A Simulation Methodology for Evaluating the Impact of Layout on the performance of Assembly/ Packaging/ Test Operations. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, Part C: Manufacturing.

Hilton, C., Mazenko, G., Solomon, L., Kempf, K. 1996. Assembly Floor Layout and Operation: Quantifying the differences. *Proceedings IEEE 5th International Symposium Semiconductor Manufacturing.*

Hopp, W.J. and Spearman, M. 1996. *Factory Physics*. Chicago, Illinois: Richard D. Irwin.

Knutson, K., Kempf, K., Fowler, J. and Carlyle, M. 1998. Lot-to-order matching for a semiconductor assembly and test facility. Department of Industrial Engineering, Arizona State University.

Law, A.M. and W.D. Kelton. 1991. *Simulation Modeling & Analysis*, Second Edition. New York: McGraw-Hill, Inc.

AUTHOR BIOGRAPHY

CHAD D. DEJONG is a Sr. Software Engineer with the Operational Decision Support Technology group at Intel Corporation, in Chandler, AZ. He has designed simulation models and various decision support tools for Intel projects over the last 4 years. He earned a B.S. in Industrial and Operations Engineering from the University of Michigan, Ann Arbor. He also earned an M.S. in Industrial and Systems Engineering from Georgia Institute of Technology, and completed the Management of Technology certificate program. His previous professional background includes hospital and health care system consulting, automotive component manufacturing, and environmental engineering. Current research and professional interests in the semiconductor industry are in whole factory capacity and operations modeling, supply chain simulation, and model design and execution time reduction. Chad can be reached via email at `<chad.d.dejong@intel.com>`.