Proceedings of the 2004 Winter Simulation Conference
R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.

A HEURISTIC TO DETERMINE EQUIPMENT SETUP CHANGES BASED ON ESTIMATED LOT ARRIVALS IN A SEMICONDUCTOR FAB

Raja Sunkara

Corporate Information Services
National Semiconductor Corp.
2900 Semiconductor Drive
Santa Clara, CA 95051, U.S.A.

Ramesh Rao

Manufacturing Decision Tools
National Semiconductor Corp.
1111 West Bardin Road
Arlington, TX 76017, U.S.A.

ABSTRACT

Certain classes of tools used in the semiconductor industry require the tools to be setup differently in order to process different types of products. In cases of large setup times, it is important to minimize the number of setup changes in order to improve the overall equipment utilization. Minimizing the number of setup changes needs to be balanced with the restriction on the product queue times. Further, it is important that the cycle time of low volume products is not penalized in order to improve equipment utilization. This article presents a heuristic algorithm to determine the setup for each tool in a workstation based on the estimated arrival times of different products at the workstation. The approach described here takes into account the number of tools, their capability, and the expected workload for each setup over a predetermined horizon. The heuristic is independent of the product mix released into the line.

1 INTRODUCTION

The work presented in this article is an extension of earlier work presented by the authors (R. Sunkara and R. Rao 2003). The heuristic was developed for use in day-to-day operations at National Semiconductor's fab in Arlington Texas (NSTE). The heuristic was first implemented as part of the NSTE factory simulation model. For deployment on the floor the Real Time Dispatch utility of Brook's APF toolset (Brooks Automation, Inc. 2004a) is used. The implementation of the heuristic in the simulation model makes use of a framework for customizations built on top Brooks-PRI's AutoSchedAP simulation package (R. Sunkara and R. Rao 2003, Brooks Automation, Inc. 2004b). The framework provides flexibility in modeling floor rules and provides better analysis and information for decision-making.

Discrete event simulation modeling has become a widely recognized management tool by many manufacturing firms. Several companies have endorsed policies requiring some form of simulation evaluation before approving and committing new investments on production resources (A. W. M. Lung 1998). Many companies use simulation to address classical problems - production bottlenecks, shop floor layout, material transport, capacity balancing and cycle time planning. However, very few companies use simulation for the design, modification, and improvement work processes (N. Melão and M. Pidd 2003). Despite the general consensus that simulation is a powerful manufacturing system analysis tool, simulation models are usually developed as a one-time use analytical model except in cases where the simulation model is used for simulation based control (J. S. Smith, R. A. Wysk, D. T. Sturrok, S. E. Ramaswam, G. D. Smith, and S. B. Joshi 1994). or simulation based scheduling (C. M. Harmonoski 1995).

NSTE is a high volume wafer-manufacturing site producing a wide mix of products on many different types of technologies (CMOS, BICMOS, BIPOLAR. Etc…). Manufacturing at NSTE is heavily dependent on Real Time Dispatch (RTD) system and simulation. Besides implementing fab wide dispatch rules to control average cycle-time, RTD is also used to maintain 98th percentile cycle-time metrics, maintain line balance, ensure timely processing of hot lots, maximize utilization of constraint tools, increase utilization of batch tools, etc. Due to the complex nature of a semiconductor fab and the key role that RTD plays at NSTE, management requires that the impact and effectiveness of complex RTD rules be assessed, using simulation, prior to deployment on the floor. Simulation output is also used to drive RTD decisions and set daily plant goals. As a result, existing dispatch rules/policies used on the floor have to be incorporated in the simulation model.

The complex interactions between product mix, tool dedication, and process restrictions mandate a detailed simulation model to evaluate dispatch and scheduling policies. In addition to traditional scheduling and dispatching policies, The simulation model at NSTE is also used to determine policies guiding setup changes, tool configurations for clustered tools (ASM-FSI Photo Clusters), operator/operator certification requirements, etc. The simulation infrastructure used at NSTE is geared towards a mature FAB with an equipment base spanning several generations of tools with little or no automation. As a result, it is not possible to set up automated data collection directly from the tools and a process for keeping the data updated needs to be established. It is also extremely important that the data and the model be validated regularly. The simulation framework and modeling process developed at NSTE engages the entire organization to make this possible. Details on the NSTE modeling and simulation infrastructure are presented in (R. Sunkara and R. Rao 2003).

The heuristic algorithm presented in this article focuses on the problem of determining tool setups. The determination of setup for each tool in a workstation is based on the estimated arrival times of different products at the workstation. The heuristic is especially effective when there are multiple tools with large setup times. The approach described here takes into account the number of tools, their capability (allowed setups and other process restrictions), and the expected workload for each setup over a predetermined horizon. The heuristic is independent of the product mix released into the line. The problem is presented with reference to setup changes in the ion implant workstations, however, the heuristic is applicable to any toolset requiring large setup times.

The remainder of the paper is organized as follows: Section 2 provides a brief problem description. Section 3 discusses our choice of simulation software and its advantages and limitations. Section 4 discusses user-defined enhancements, why we need them, our implementation approach, and functionality captured. The heuristic algorithm is presented in Section 5. Section 6 presents some of the observations we made during our studies using the simulation model. Figure 1 is presented in Section 7. We conclude the paper with a brief outline of future direction.

2 PROBLEM STATEMENT

The heuristic presented here was specifically developed as a solution to the following requirements for the ion implant area:

- Total of 9 implanters (5 belonging to family A, and 4 belonging to family B)
- Of the 4 possible setups {S1, S2, S3, S4} for Family A, S1 is allowed on two tools and S2 is allowed on the other 3 tools. S3 and S4 are allowed on all tools.
- A tool that is setup for S2 has to be setup for S3 before it can be setup for S4
- Not all products can be processed on Family B.
- Process material on Family B only if the workload on Family A would violate the maximum allowed queue time
- Tools in Family A achieve maximum throughput if we can stage multiple lots in front of the tool
- Minimize setup changes
- Maximum time a lot can wait for a tool to switch to the required setup is 8 hours
- For setup S3 there are two types of wafers (R and NR). The tool needs to run dummy wafers when switching from the R type wafers to the NR type wafers. Minimize the use of these dummy wafers
- The rule cannot be hardwired for a certain product mix but needs to self balance as the mix changes.

The heuristic developed ensures that tools are setup to best utilize the tools while trying to minimize time spent on setups, lot queue times, and tool idle times. The heuristic is used in conjunction with dispatch policies to satisfy requirements stated above.

3 SIMULATION SOFTWARE

The simulation software used at National Semiconductor Corporation (NSC) is the Brooks-PRI AutoSchedAP simulation package (Brooks Automation, Inc. 2004b). There were several reasons why AutoSchedAP was selected as the simulation engine. AutoSchedAP reads all the input data from tab delimited text files, which works well for data maintenance as opposed to a proprietary structured database. The software provides several standard rules and constructs that are geared towards the semiconductor industry. AutoSchedAP provides a flexible framework for user customizations that allows users model situations that are specific to individual factories. Customizations are also very well supported by Brooks-PRI. The standard features supporting operator modeling however leaves a lot to be desired. Additional features supporting the development of dispatch rules would be a welcome enhancement.

4 NATIONAL CUSTOMIZATION FRAMEWORK DETAILS

The framework for customizations presented in this section increases the turn-around time on the development and testing of new dispatch policies. The framework provides flexibility in modeling floor rules and provides better analysis and information for decision-making. AutoSchedAP provides a framework for the development of additional functionality; the National framework is built on top of the AutoschedAP framework to customize dynamic simulation data to meet the rule development and simulation reporting needs.

4.1 National Framework Design

AutoSchedAP is built using C++ class libraries. The AutoSchedAP framework forms the basis for most of the standard functionality in the simulation. AutoSchedAP provides many simple ways to customize features of the simulation engine, which does not require C++ development. Examples include optional fields in the model inputs and action lists, which allow the definition of a sequence of actions to be, take at different point in the simulation to attain a particular behavior. But some of the sophisticated enhancements like scheduling algorithms and dispatch policies will require additional C++ class library be developed.

The C++ class library developed in the case presented in this paper is referred to as National framework, see Figure 1. The National framework simplifies tracking information required to implement the dispatch rules. This takes into account information related to current factory dynamics. The AutoSchedAP framework provides the mechanisms to subscribe to various events (e.g. state changes, lot selection etc.); this feature was extensively used to update the factory data subject to the factory events. The National framework consists of Factory, Station Group, Station Family, Lot, Route and Setup etc. classes which capture different data elements which are updated subject to certain events in the simulation. For example, when a lot is released by a station the statistics of the current setup are decremented and the statistics for the next setup are incremented.

4.2 National Framework Functionality

One of the main features of the National framework is to enable dispatching decisions based on the factory status. In this section a subset of the National framework's functionality is described using the implant setup rule, outlined in Section. 2.

The framework provides mechanism to track WIP and classify it into several buckets. Some of the dispatching decisions are made looking at the WIP profile of current lots waiting to be processed by the equipment group along with the future lot arrivals. The future arrivals are broken down into hourly buckets for a predefined window of time (usually next 2 or 3 days) to provide finer granularity. This data is available to specific stations of interest as opposed to all stations in the factory to enable faster execution times. The WIP is identified by number of wafers and is further classified based on setup requirements, processing requirements, and process restrictions. WIP belonging to a particular setup is further classified by the wafer characteristics (e.g. R vs. NR wafers in implant). Most of the WIP tracking information is done at factory level for all equipment of interest.

The National framework keeps track of the average wait time for the lots currently in the queue for the equipment under consideration. This helps in enforcing the operational policy of lots be routed to certain preferred group of equipment unless the current queue time at that group is greater than the predefined levels. Individual lot wait times at the current equipment are tracked to take care of process restriction (e.g. a lot can not wait more than 14 hours before next process). The dispatch rule looks at the wait time of each lot and makes an exception if a setup change is required to enforce process restrictions.

AutoSchedAP framework publications for an equipment state change event (e.g. process state to setup state) is used to keep note of elapsed time since the last time equipment changed its setup. This metric is taken into account in enforcing another operational policy that equipment cannot run for more than 30 hours on some specified setup.

When a tool finishes processing its current load the next lot selected is based on its current setup. The selection of next lot may trigger a new setup requirement. In order to minimize setups, the dispatch rule selects a lot to work on from the WIP and the framework acts as a mini MES to support dispatching decisions. The potential decisions may range anywhere from leaving the tool idle to changing the tool's setup. In most cases the current setup is changed subject to the WIP profile. In some cases the setup is not changed even though there are no lots waiting at the tool with the current setup requirements. Setup change decisions also take into account number of available tools, the state of each tool (e.g. Down, PM, Qual) and number of available tools with a particular setup. If a tool is in the down state, an estimated remaining down time is taken into account before making the setup decision. Most of the National framework features described are applicable to many dispatching situations.

5 THE SETUP HUERISTIC

Let n = total number of setups allowed at the workstation.
Let t = number of tools at the workstation.
Let h = the horizon or the time between subsequent runs of the algorithm (in hours).

We define the following:

$S = \{S_1, S_2, ..., S_n\}$ is the set of all possible setups allowed at the workstation.

$T = \{T_1, T_2, ..., T_n\}$ is the set of tools at the workstation.

$LRT(S_i)$ = Max workload allowed to build up for setup S_i at the workstaton before a tool has to be converted to S_i.

$WL(S_i,h)$ = Expected workload for setup S_i at the workstation over the horizon h.

$TL(h) = \sum_i WL(S_i,h) =$ total workstation workload.

$CS(T_k)$ = Current setup for tool T_k

$Count(S_i)$ = current number of tools on setup S_i.

$Estimated(S_i,h) = \dfrac{t \times WL(S_i,h)}{TL(h)}$ is the estimated number of tools required to process workload requiring setup S_i.

$Allowed(S_i,T_k)$ is true if T_k can be setup for S_i from $CS(T_k)$.

$Required(S_i,h)= Estimated(S_i,h) - Count(S_i)$ is the number of additional tools required over the next horizon. A negative requirement means that one or more tools currently on S_i may be changed to a different setup. The number of tools that can be changed is defined by $|Required(S_i,h)|$

$CanChange(T_k,S_i)$ is true if
$Required(CS(T_k),h) < 0$ and $Allowed(S_i,T_k)$

$PreviousCycle(S_i)$ is true if there was at least one tool setup on S_i during the previous horizon

$Set(T_k,S_i)$ sets T_k to setup S_i

Based on the above definitions the heuristic algorithm may be defined as follows:

```
For each Tk in T {
  For each Si in S {
    If (Required(Si,h) > 0 And CanChange(Tk,Si)) {

      If (Required(Si,h) > Required(CS(Tk),h) + 1) {
        Set(Tk,Si)
      }

      If (WL(Si,h) > LRT(Si) And Not PreviousCycle(Si)){
        Set(Tk,Si)
      }

    }
  }
}
```

5.1 Implementation Notes

The horizon h depends on several factors. In our implementation we based the horizon h on the following:

- Arrival Rates
- Low volume products
- Queue time constraints
- Average setup time
- Overall run rate (factory loading).

We also conducted several simulations on the impact of h and found that for our factory values between 6 and 10 best met our requirements.

The workload function $WL(S_i,h)$ may also be implemented is several ways. The simple implementation would be to total the processing times of all lots expected to arrive at the workstation over the horizon h. However, we found that adding a component of distance from the tool provides better results. One way to do this is by defining $WL(S_i,h)$ as follows:

$$WL(S_i,h) = \sum_{m=1}^{h}[WL(S_i,m)\times(h-m+1)\times(a/t)]$$

Where a = *average processing time per lot in hours*
t = *total number of tools at workstation*

The function $Allowed(S_i,T_k)$ is used to handle equipment dedication and other process/operational restrictions on setup changes. Like the horizon and workload function the implementation of this function will also depend on site policies, requirements, and goals. This function is also used to model interactions across workstations.

6 OBSERVATIONS

In the case of the implant area operations presented here, the range of setup times reported in the simulation model had dropped from a range of 20-25% to 5-9% reflecting the floor operations more accurately. Further, the Queue times reported in front of the tools also dropped to acceptable limits (from a range of 6-9 hours to a range of 3-7 hours).

When a factory is loaded less than 75% we found (via simulations) that the model is more sensitive to the selection of the horizon h. Larger values of h resulted in excessive Queue and Idle times. The same applies to the selection of h when there are certain setups catering to low volume products. In the algorithm presented above the second if statement, <u>If ($WL(S_i,h) > LRT(S_i)$ And Not $PreviousCycle(S_i)$){ $Set(T_k,S_i)$},</u> takes care setups catering to low volume products, however, a large value of h will result in a longer period of time before low volume products are processed.

Equally important to low volume products is the determination of $LRT(S_i)$. This value of $LRT(S_i)$ is determined by the mix, volume, and Queue time restrictions. We derived the value of $LRT(S_i)$ based on a combination of operation procedure, start rate/mix, and simulation analysis.

7 FIGURES AND TABLES

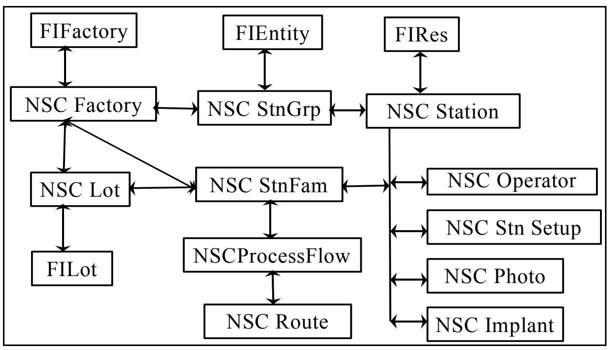

Figure 1: Sample National Framework Classes

8 FUTURE DIRECTION

We are currently working on methods to dynamically determine when to invoke the algorithm and a more formal way to evaluate the value of *h* and $LRT(S_i)$. We are also experimenting with different ways of calculating $WL(S_i,h)$ so tool dedication and mix are taken into account eliminating the need for $LRT(S_i)$ as an input.

REFERENCES

Brooks Automation, Inc. Auto Simulations Division,. 2004a. APF Reporting, Dispatching, and system administration User's Guides

Brooks Automation, Inc. Auto Simulations Division,. 2004b. AutoschedAP User's Guide

C. M. Harmonoski (1995). Simulation-based real time scheduling: Review of recent developments. *Proceedings of the 1995 Winter Simulation Conference*, 220-225.

A. W. M. Lung (1998). Benefits of a Systematic Approach for Problem Analysis on Manufacturing Simulation Modeling. *International Conference on SIMULATION* No. 457: 271-276.

N. Melão and M. Pidd (2003). Use of business process simulation: A survey of practitioners. *Journal of the operational Research Society* Vol. 54, No. 1: 2-10.

J. S. Smith, R. A. Wysk, D. T. Sturrok, S. E. Ramaswam, G. D. Smith, and S. B. Joshi (1994). Discrete Event Simulation for Shop Floor Control. *Proceedings of the 1994 Winter Simulation Conference*, 962-969.

R. Sunkara and R. Rao (2003). Use Of Discrete Event Simulation To Analyze Dispatch Policies Of An Equipment Group In Semiconductor Fab. *Proceedings of the 2003 Winter Simulation Conference*, 1474-9.

AUTHOR BIOGRAPHIES

RAJA SUNKARA is a Principal Systems Engineer in Corporate Information Services Division of National Semiconductor Corp. headquarters at Santa Clara, California. He has been working in the semiconductor industry for the past 10 years in the areas of Productivity Improvement, Enterprise Resource Planning, Business Process Automation and Information Services. He received MS in Industrial Engineering, MBA and BS in Mechanical Engineering. (e-mail `<Raja.Sunkara@nsc.com>`)

RAMESH RAO is a Manufacturing Engineering Manager at National Semiconductor Corp. at Arlington Texas where he heads the Manufacturing Decision Tools group. He has been with National Semiconductor Corp. at Arlington Texas since 1996. Prior to that he was at the University of Virginia where he received his Ms and PhD Degrees. His interests include simulation, RTD, algorithms, factory analysis tools, e-Manufacturing, factory automation architectures, and work process automation. (e-mail `<Ramesh.Rao@nsc.com>`)