Proceedings of the 2009 Winter Simulation Conference
M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, eds.

HEURISTIC BASED SCHEDULING SYSTEM FOR DIFFUSION IN SEMICONDUCTOR MANUFACTURING

Tanju Yurtsever

Freescale Semiconductor, Inc.
6501 William Cannon Drive West
Austin, TX, 78735, USA

Erhan Kutanoglu

The University of Texas at Austin
1 University Station C2200
Austin, TX 78712 USA

Jennifer Johns

Freescale Semiconductor, Inc.
6501 William Cannon Drive West
Austin, TX, 78735, USA

ABSTRACT

This paper addresses the scheduling of lots in a specific wafer fabrication area, called diffusion, where scheduling of lots interact with batching, equipment dedication and queue time constraints. Realizing the difficulty of solving the underlying mathematical program optimally, we develop a heuristic to regularly schedule the lots available in the area in real time. The paper explains the user interface and implementation issues as well as the details of the heuristic logic. The results obtained from production in a wafer fabrication facility to date show high user compliance, improved predictability and visibility of the overall schedule, and improved operational performance including reduced cycle times and queue time violations.

1 SEMICONDUCTOR MANUFACTURING OVERVIEW

Semiconductor manufacturing (see Figure 1) is modeled with the interaction of lots with equipment. In a 200-mm wafer fabrication facility (wafer fab for short), there are about 3500 lots that run through about 600 equipment. Due to the complexity of the process, advanced computer software is used to keep track of lot movements and equipment states within the fab. Further automation through computer-aided lot and recipe track-in (assignments of lots/recipes to specific equipment) is achieved with MES (Manufacturing Execution System).

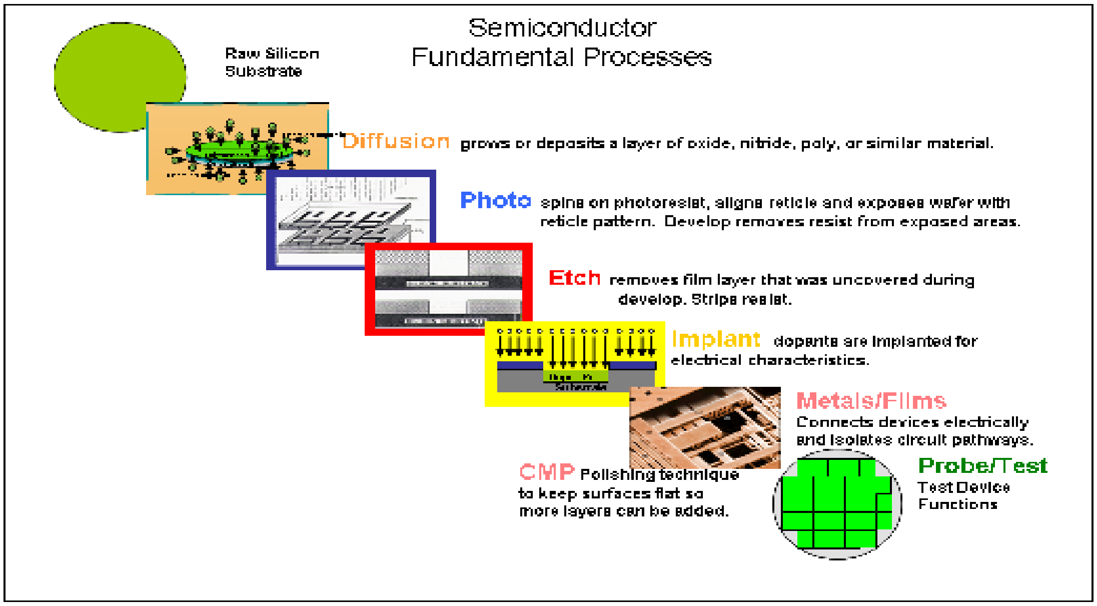

Semiconductor
Fundamental Processes
Raw Silicon
Substrate
Diffusion grows or deposits a layer of oxide, nitride, poly, or similar material.
Photo spins on photoresist, aligns reticle and exposes wafer with reticle pattern. Develop removes resist from exposed areas.
Etch removes film layer that was uncovered during develop. Strips resist.
Implant dopants are implanted for electrical characteristics.
Metals/Films
Connects devices electrically and isolates circuit pathways.
Probe/Test
Test Device
Functions
CMP Polishing technique to keep surfaces flat so more layers can be added.

Figure 1: Semiconductor process areas

978-1-4244-5771-7/09/$26.00 ©2009 IEEE

A lot contains 25 wafers of same type. Each lot follows a unique process flow (depending on the part type). A process flow consists of 300-500 process steps. Each process step uses an equipment type and a unique equipment recipe. Sometimes, a step may have alternative recipes and alternative equipment types.

Each step has an estimated process time and (maximum) batch size, based on the selected equipment and recipe. In diffusion, there is a time limit within which a group of steps must be completed which makes the scheduling problem even harder. These restrictions define the queue timer limits. The maximum batch size defines the maximum number of lots that can be run at the same time on equipment with a selected recipe.

2 MOTIVATION

Dispatching used heavily in semiconductor manufacturing tries to customize a priority rule according to the current step and time, but even with enhancements for upstream and downstream data, current dispatching methods do not provide an easy way to coordinate decisions across equipment and time periods. One could argue that this issue is more prominent in diffusion than any other area. This is mainly because the diffusion processes are long and the equipment across multiple stages allows batching of lots of same type and process (recipe), which is essential for equipment utilization and throughput. Lot-based dispatching implemented one-stage at a time, despite significant diffusion-related enhancements, cannot easily handle batching across multiple wafer types and optimize decisions across batching equipment. Considering that lots of some wafer types must be scrapped if they spend more than a given time between certain batching stages (queue timer limits), coordination of decisions across stages of production becomes even more important. Therefore, we study this area in detail to develop batching/scheduling models and algorithms that will capture the essentials in coordinating decisions over time and across stages/equipment. To make this discussion more concrete and to show how we actually attack the ever-challenging dispatching/scheduling problems, we detail the diffusion area problem below first as a mathematical model, and then discuss our heuristic solution technique.

3 LITERATURE REVIEW

Although there exist many batching situations in manufacturing, two situations are most commonly studied from the scheduling point of view: batching lots sharing the same setup on an equipment (serial batching) and batching lots on an equipment that can process several lots simultaneously (parallel batching) (Potts and Kovalyov, 2000). In parallel batching, an equipment processes a batch of lots simultaneously like an oven. Different versions of the parallel batch scheduling problem are studied by Chandru et al. (1993a, 1993b), Uzsoy et al. (1994, 1995), Ghazvini and DuPont (1998), Dobson and Nambimadom (2001) and Azizoglu and Webster (2001).

As parallel batching is closely related to furnace operations seen in diffusion, there are specific studies motivated by diffusion issues. In this group, we list Mehta and Uzsoy (1998), Gurnani, Anupindi and Akella (1992), Balasubramanian et al. (2004), Perez et al. (2005), Monch et al. (2005), Weng and Leachman (1993), and Fowler, Hogg and Phillips (2000), majority of which use enhancements to dispatching rules to facilitate batching, and test the approaches with simulation.

Unlike parallel batching, lots in a batch are processed sequentially in serial batching, and each batch is preceded by a setup. The serial problem is first considered by Coffman et al. (1989, 1990), who show that the problem of minimizing total completion time with both identical and arbitrary lot processing times is polynomially solvable. Albers and Brucker (1993) show that the problem of minimizing the total weighted completion time is NP-hard unless the lot processing times are identical. The serial batch scheduling problem with the objective other than the total completion time is also studied (Hochbaum and Landy, 1994, Yuan and Yang, 2003). Unlike parallel batching, no limit on the batch size is considered in serial batching. One exception is a study by Cheng and Kovalyov (2001) who analyze a special case called bounded problem in which there is a limit on the number of lots that may be grouped together in a batch.

4 OPTIMIZATION MODEL

In this simplified version of the diffusion batching/scheduling problem, we assume that we have a set of lots I (indexed by i) to be scheduled over two stages of production, each with multiple equipment. The first stage is called sink (denoted by superscript S) and the second stage furnace (denoted by superscript F). The set of sinks is K^S, and furnaces K^F, with the overall equipment set denoted by K, all indexed by k. Set of process recipes is L, with the usual distinction between L^S and L^F, all indexed by l. Batch processing is allowed only of the lots of the same process recipe. The equipment and recipe combination determines the operation in the usual scheduling sense. Prioritization of lots is captured with π_i, larger values denoting high priority in the processing sequence. Processing time of sink recipe l per batch is p_l^S and the maximum batch size (or capacity) of sink when running recipe l is b_l^S. We similarly define p_l^F and b_l^F for furnaces. Although both stages allow batch processing, the actual operation is different between the two: Because a sink has multiple tanks, each can start taking in a new batch as soon as a tank is available (while processing batches in other tanks). Hence, the slowest tank time denoted by r_l^S determines the delay time between the starts of two consecutive batches at a sink. Furnaces operate like an oven, i.e., a batch has to be finished completely before the next one is loaded. Set J denotes the set of batches, which represent placeholders for lots. Without loss of generality and optimization, we assume that the batches are processed in the order of their index: batch 1 first, then batch 2, and so on. Hence, index j denotes not only a batch but also a position in the processing sequence at an equipment, which helps greatly in modeling and makes the overall problem to be filling the batches with lots (assignment of lots to batches) while making sure that dynamics of the problem are properly captured across lots, batches, recipes, and equipment.

We have the following binary decision variables:

$$X_{ijk}^S, X_{ijk}^F : 1 \text{ if lot } i \text{ is assigned to batch/position } j \text{ on sink (or furnace) } k$$

$$W_{jkl}^S, W_{jkl}^F : 1 \text{ if sink/furnace recipe } l \text{ is assigned to batch } j \text{ on sink (or furnace) } k$$

The other decision variables control the start and completion times:

$$S_{jk}^S, C_{jk}^S : \text{Start and completion times of batch } j \text{ on sink } k$$

$$S_{jkl}^F, C_{jk}^F : \text{Start and completion times of batch } j \text{ on furnace } k$$

$$B_i^S, E_i^S : \text{Start and completion times of batch } j \text{ on sink } k$$

$$B_i^F, E_i^F : \text{Start and completion times of batch } j \text{ on furnace } k$$

Ignoring the nonnegativity/binary restrictions on the variables, we write the overall model:

$$\min \sum\nolimits_{i \in I} \pi_i E_i^F \tag{1}$$

subject to

$$S_{jk}^S \ge S_{j-1,k}^S + \sum\nolimits_{l \in L_k^S} r_l W_{j-1,k,l}^S, S_{jk}^F \ge C_{j-1,k}^F \quad \forall j \in J, k \in K \tag{2}$$

$$C_{jk}^S = S_{jk}^S + \sum\nolimits_{l \in L_k^S} p_l^S W_{j,k,l}^S, C_{jk}^F = S_{jk+}^F \sum\nolimits_{l \in L_k^F} p_l^F W_{jkl}^F \quad \forall j \in J, k \in K \tag{3}$$

$$\sum\nolimits_{j \in J} \sum\nolimits_{k \in K_l^S} X_{ijk}^S = 1, \sum\nolimits_{j \in J} \sum\nolimits_{k \in K_l^F} X_{ijk}^F = 1 \quad \forall l \in L, i \in I \tag{4}$$

$$\sum\nolimits_{i \in I_l^S} X_{ijk}^S \le b^S W_{jkl}^S, \sum\nolimits_{i \in I_l^F} X_{ijk}^F \le b^F W_{jkl}^F \quad \forall l \in L \tag{5}$$

$$\sum\nolimits_{l \in L_l^S} W_{jkl}^S \le 1, \sum\nolimits_{l \in L_k^F} W_{jkl}^F \le 1 \quad \forall j \in J, k \in K \tag{6}$$

$$X_{ijk}^S \le W_{jkl}^S, X_{ijk}^F \le W_{jkl}^F \quad \forall l \in L, i \in I, j \in J, k \in K \tag{7}$$

$$B_i^S \ge S_{jk}^S - M(1 - X_{ijk}^S), B_i^F \ge S_{jk}^F - M(1 - X_{ijk}^F) \quad \forall i \in I, j \in J, k \in K \tag{8}$$

$$E_i^S = B_i^S + p_l^S, E_i^F = B_i^F + p_l^F \quad \forall l \in L, i \in I \tag{9}$$

$$B_i^F \ge E_i^S \quad \forall i \in I \tag{10}$$

Here, the objective function (1) seeks to finish lots as quickly possible, weighting the lots according to their priorities. Constraints (2-9) are grouped according to their stages, the left for sink constraints, and the right for the furnace constraints. Constraints (2) find the batch start times, taking advantage of the presequenced batches in the order of their corresponding indices. Sinks have a delay between starting times of consecutive batches according to the recipe-based slowest tank time, while batches at furnaces cannot start until the previous batch is completed. Constraints (3) compute the batch completion times, considering the recipe-equipment specific processing times of batches. Constraints (4) make sure that each lot is assigned to exactly one batch on an equipment in each stage. Constraints (5) ensure that the number of lots in a batch does not exceed the maximum batch size (sink/furnace capacity in number of lots). Constraints (6) state that at most one recipe can be assigned to a batch on an equipment. Constraints (7) make sure that a lot of a certain recipe is assigned to a batch only if the batch is of that recipe. Constraints (8) align the lot and batch start and completion times at both stages without needing nonlinear terms. Here, the big-M constants can be tightly calculated to strengthen the LP bounds. Constraints (9) compute lot completion times given the lot start times at both stages. Constraints (10) make sure that a lot starts processing at a furnace after it is completed at the sink. For queue timer lots, we can add constraints of the form $B_i^F \le B_i^S + q_i$, where q_i is the queue timer limit for lot i.

Note that the actual diffusion batching/scheduling problem is more complicated, mainly due to longer process flows (routings) potentially different across product types (some with sink-to-furnace (S-F) flows as presented here, but others with 3 or more stages such as S-S-F, S-F-F, and S-F-F-S-F flows). Also note that batching compatibilities between product types change as they move in diffusion with changing recipes across multiple steps, making the product type less relevant for batching purposes. Finally, equipment maximum batch sizes/capacities (especially furnaces) depend on assigned recipes, which is again different from the models considered in the literature.

Solving this optimization problem (even with its simplified flow) to optimality with exact methods is extremely challenging due to its size and overall complexity. As mentioned before, even the single-stage batching problems are NP-hard, which gives no hope for finding efficient exact algorithms for this model. Hence, we develop a heuristic method that finds fast (but not necessarily optimal) schedules. The heuristic implementation at Freescale Semiconductor (called FaST, Fab Scheduling Tool) is explained next.

5 SCHEDULING HEURISTIC

As discussed earlier in the paper, due to the size and nature of this problem, optimization tools were not feasible so a heuristic algorithm is developed to solve this large scale complex problem.

The heuristic procedure has three distinct objectives while it schedules batches through the equipment:

- Maximize equipment utilization by assigning highly ranked full batches and minimize the idle time between each batch run
- Minimize the total queue (waiting) time of highly ranked lots which will also minimize the cycle times
- Minimize the amount of expired lot timers by scheduling lots in timer loops on time

The heuristic is a batch-oriented and furnace driven algorithm. At each iteration, it tries to find the best batch based on the above objectives. The algorithm ends when the entire equipment capacity is used to schedule the current WIP within a user defined interval.

At any iteration, the best batch is defined to be the batch that has the highest number of most important lots that will introduce the minimum idle time to the equipment it is assigned to. Having a full batch will satisfy the equipment utilization objective together with introducing the minimum idle time on a tool between batches. While forming batches, using a global dispatch rule that also considers the total queue time of a lot together with its overall importance to pick the lots to be assigned to batches will address the cycle time objective. If the heuristic finds an important lot on a timer loop, it will give precedence to these batches which will satisfy the minimization of the lot timer objective.

A rather sophisticated and multi-stage search algorithm is used while choosing the best batch of lots and best equipment for the best batch. During these assignments, all the constraints are evaluated one by one to keep the solution feasible.

There were several decision variables that affect the quality of the solution. There is a trade-off at certain cases such as:

- *Amount of time a batch should wait for an incoming lot or lots*, i.e: there may be 2 lots available to be processed at time *t* but if we hold the batch for 30 more minutes, two more lots can be added. There is a tradeoff between trying to minimize the idle time on equipment by not waiting for incoming lots and increasing the batch size.
- *Physical lot location*: A lot can be assigned to an equipment that will be available within a certain time in its current bay (i.e., physical area in which similar equipment types are grouped) or it can be assigned to an already idle equipment in a different bay. There is a tradeoff between waiting in the current bay versus moving to a different bay that has an idle equipment.
- *Equipment selection for a batch*. If a batch can be scheduled on more than one equipment, the algorithm has to choose one so that there are more available assignment alternatives for unscheduled batches. Equipment assignment to a batch is also hard decision and requires advanced algorithms. In this heuristic approach, equipment with the most process restrictions is selected to maximize the possibilities for the other assignments. For example, if equipment A can run 10 lots and equipment B can run 5 lots, if a lot can be processed on both A and B, the algorithm will choose equipment B to increase the possibility of assigning an equipment (say A) for incoming lots (10 of which are counted in the equipment selection) when needed.
- *Fixing the schedule*. Due to real time and dynamic behavior of the system, if the algorithm is free to reschedule all available lots at its each run, the schedule may change dramatically. Changing the schedule regardless of what had been decided in the previous runs may be disruptive, considering the staging of batches, setting up equipment, etc. For example, operators stage the lots in front of the equipment based on the schedule and they prefer to minimize the moving of lots between areas and equipment in response to schedule changes. Therefore, it is not practical to run the

schedule from scratch every time. The algorithm uses a rule-based decision system to decide which batch to equipment pairings from previous runs should be fixed (kept as is) or which fixed pairings should be released.

- *Buffer time between consecutive process steps*. It is not practical to assume that a lot will be processed on its next step right after it completes its current step. The algorithm assigns a buffer time between consecutive process steps so operators will have enough time to move the lots between equipment.
- *Queue timer lot flag time*. Lots with less time to expire their queue timer loops are prioritized higher. Lots with less time to expire than a certain threshold are expedited over other timer or non-timer lots.

To understand the underlying tradeoffs between different settings of these parameters and their interactions with each other, we have conducted a parametric computational study of over 100,000 runs with sampled actual data. The results showed that some of these parameters were critical and others were not. The critical parameters were set according to their impact on the scheduling objectives. We have been continuously observing the impacts of these parameters and adjust them based on inventory, equipment availability and product mix.

5.1 System Design

The algorithm explained in the previous section is supported by other components in the system design which provides data and user interface functionalities. The overall system is called Fab Scheduling Tool (FaST) which consists of three main components, Data Extraction, Scheduling Heuristic (algorithm engine), and Gantt-Chart based schedule viewer (Figure 2).

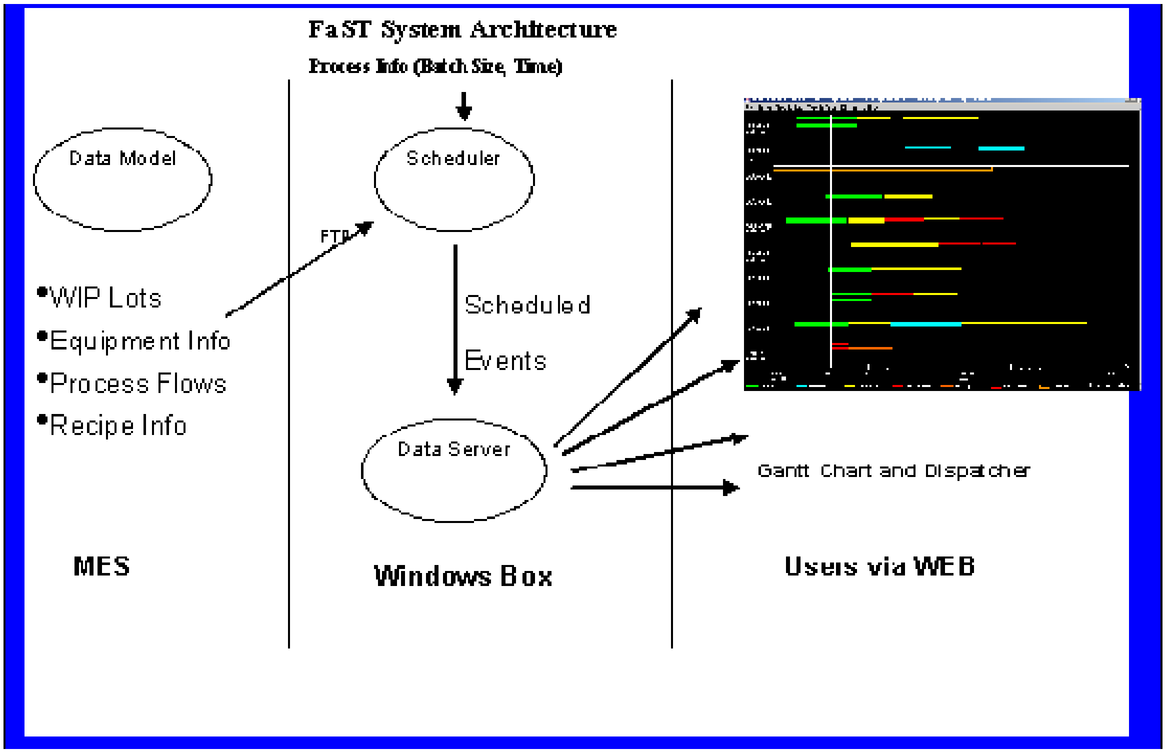


Figure 2: High-level overview of the FaST architecture

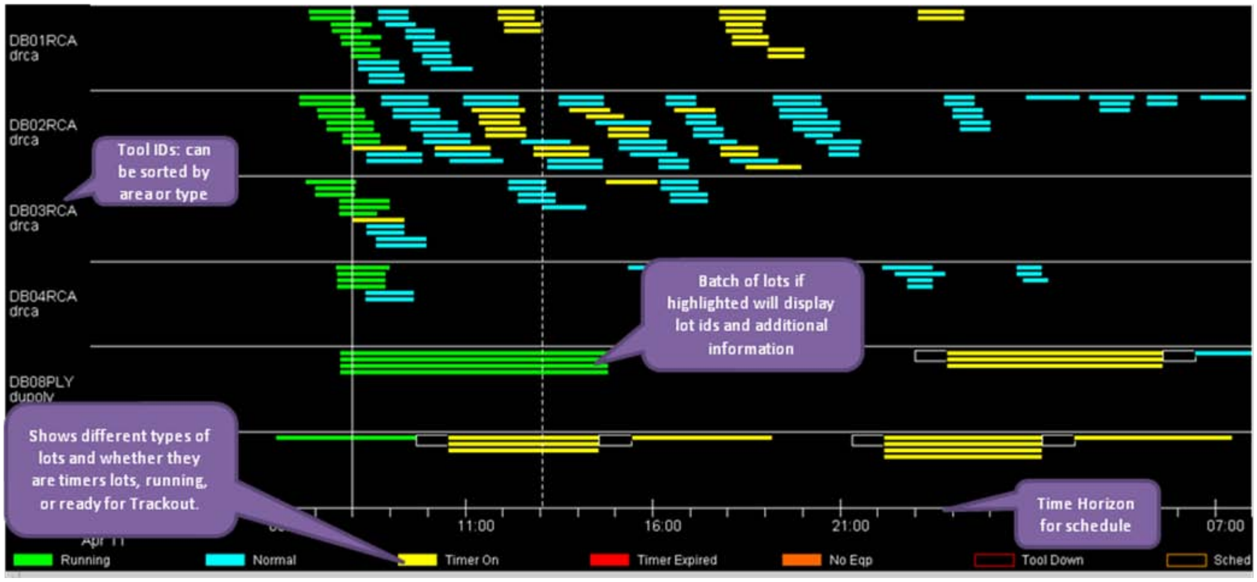
Due to the complicated nature of semiconductor manufacturing, advanced Manufacturing Execution Systems (MES) are used for lot and equipment tracking, equipment integration and automation. Computerized and automated systems provide ways to facilitate paperless manufacturing and its management. MES is part of this system that keeps track of people, lot and equipment interactions. FaST's input data is directly extracted from the MES. The data contains current Work-In-Process (WIP) lot snapshot, lot process flows (routings), equipment states (up, down, etc.), expected equipment up and down times, equipment maintenance information (preventive maintenance schedules), equipment recipe states (qualified or enables recipes), process times and (maximum) batch sizes. This data is being extracted in real time and it is fed to the Scheduling Engine.

Above data is maintained by the MES so that data always represents reality which enables FaST to be a real-time scheduling system.

Only WIP lots in the diffusion area or queued to arrive to diffusion are used in the extract. Each lot has average of 5 to 6 processing steps in diffusion. We group the processing steps to be sink, furnace, or other (such as inspection, sorters). Each step can be processed using one of 3 to 4 alternate equipment recipes on one of 3 to 5 equipment. There are about 20 equipment types with over 100 equipment within diffusion. Each equipment has 5 to 10 qualified process recipes. Equipment may have scheduled maintenance tasks or unscheduled down times. It takes about 90 seconds to extract the data required to feed the scheduler.

Extracted input data is moved to a Windows Server where the core scheduling engine runs. The scheduling engine is a heuristic based scheduler that is developed in Java and it can run on any platform that supports Java. The scheduler runs each time a new input data is provided by the data extraction component. The heuristic uses about 30 seconds of CPU to schedule 2000 events (pairs of equipment and lot assignments). The scheduling horizon is set to be 24 hours. FaST creates a scheduled event list as an output to be sent to the schedule viewer module. (This output is also integrated to the existing dispatching applications so that the operators can follow the schedule with the standard MES interface that shows the operators the order of lots that need to be run on the shop floor)

Schedule viewer is the 3rd component of the system (see Figure 3 for an example screen snapshot). Gantt chart based schedule viewer is a Web-based GUI application that displays the scheduled events generated by the scheduler. The viewer helps supervisors and shift administrators to visualize the generated schedule. On this chart, we group lots in five classes, lots currently running on an equipment, non-timer lots, timer lots, expired timer lots, and lots whose down stream equipment is not available, and each group is shown with a different color.



Tool IDs: can be sorted by area or type

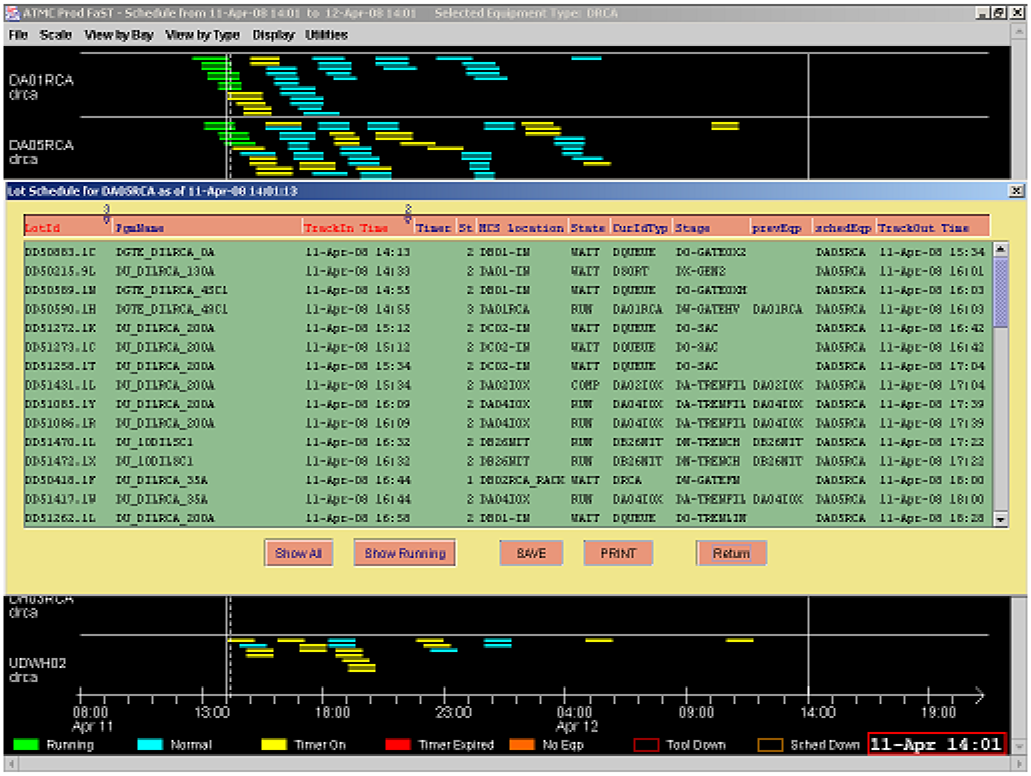
Batch of lots if highlighted will display lot ids and additional information

Shows different types of lots and whether they are timers lots, running, or ready for Trackout.

Time Horizon for schedule

Figure 3: Gantt chart showing schedule for next 24 hours

Gantt chart has been also a very powerful tool to debug the algorithm. This makes it possible to validate the heuristic logic by following the flow of each batch. It also helps management and supervisors validate the accuracy of the heuristic. Gantt chart has several supporting reports to show scheduled batches on each equipment or equipment type (see Figure 4 for an example), available equipment recipes, expected equipment up or down times, current WIP displays, equipment history and individual lot flows.



Lot Schedule for DA05RCA as of 11-Apr-08 14:01:13

LotId	PgmName	TrackIn Time	Timer	St	MES Location	State	DurIdTyp	Stage	prevEqp	schedEqp	TrackOut Time
DD50883.1C	DGTE_DILRCA_DA	11-Apr-08 14:13		2	DB01-IN	WAIT	DQUEUE	DO-GATEOX2		DA05RCA	11-Apr-08 15:34
DD50215.9L	DU_DILRCA_130A	11-Apr-08 14:33		2	DA01-IN	WAIT	DSORT	DX-GEN2		DA05RCA	11-Apr-08 16:01
DD50589.1M	DGTE_DILRCA_45C1	11-Apr-08 14:55		2	DB01-IN	WAIT	DQUEUE	DO-GATEOXM		DA05RCA	11-Apr-08 16:03
DD50590.1H	DGTE_DILRCA_48C1	11-Apr-08 14:55		3	DA01RCA	RUN	DA01RCA	DW-GATEHV	DA01RCA	DA05RCA	11-Apr-08 16:03
DD51272.1K	DU_DILRCA_200A	11-Apr-08 15:12		2	DC02-IN	WAIT	DQUEUE	DO-SAC		DA05RCA	11-Apr-08 16:42
DD51273.1C	DU_DILRCA_200A	11-Apr-08 15:12		2	DC02-IN	WAIT	DQUEUE	DO-SAC		DA05RCA	11-Apr-08 16:42
DD51258.1T	DU_DILRCA_200A	11-Apr-08 15:34		2	DC02-IN	WAIT	DQUEUE	DO-SAC		DA05RCA	11-Apr-08 17:04
DD51431.1L	DU_DILRCA_200A	11-Apr-08 15:34		2	DA02IOX	COMP	DA02IOX	DA-TRENFIL	DA02IOX	DA05RCA	11-Apr-08 17:04
DD51085.1Y	DU_DILRCA_200A	11-Apr-08 16:09		2	DA04IOX	RUN	DA04IOX	DA-TRENFIL	DA04IOX	DA05RCA	11-Apr-08 17:39
DD51086.1R	DU_DILRCA_200A	11-Apr-08 16:09		2	DA04IOX	RUN	DA04IOX	DA-TRENFIL	DA04IOX	DA05RCA	11-Apr-08 17:39
DD51470.1L	DU_10DIL5C1	11-Apr-08 16:32		2	DB26NIT	RUN	DB26NIT	DN-TRENCH	DB26NIT	DA05RCA	11-Apr-08 17:22
DD51472.1X	DU_10DIL8C1	11-Apr-08 16:32		2	DB26NIT	RUN	DB26NIT	DN-TRENCH	DB26NIT	DA05RCA	11-Apr-08 17:22
DD50418.1F	DU_DILRCA_35A	11-Apr-08 16:44		1	DB02RCA_RACK	WAIT	DRCA	DW-GATEFN		DA05RCA	11-Apr-08 18:00
DD51417.1W	DU_DILRCA_35A	11-Apr-08 16:44		2	DA04IOX	RUN	DA04IOX	DA-TRENFIL	DA04IOX	DA05RCA	11-Apr-08 18:00
DD51262.1L	DU_DILRCA_200A	11-Apr-08 16:58		2	DB01-IN	WAIT	DQUEUE	DO-TRENLIN		DA05RCA	11-Apr-08 18:28

Figure 4: Display of detailed lot schedule for a specific equipment

5.2 Results to Date

FaST has been designed and developed over 2 years at Freescale Semiconductor, Austin, TX. The project had over 40 human resources from IT, Manufacturing, Equipment Engineering, Process Engineering and Industrial Engineering for requirements definition, deployment, and implementation. The FaST application has been deployed to one factory at Freescale with the intent to rollout to additional factories. We have been able to cover 90% of the factory requirements. The benefits recognized since May 2008 are:

- 10% increase in throughput and a 25% decrease in inline cycle time in the area.
- A reduction in end user time trying to decide which lots to run next.
- The mean time between runs on the furnaces has been reduced by more than 10%.
- The users have better visibility into what lots are coming to their equipment and when lots will be ready to unload allowing them to reduce idle time.

We were also able to release 4 resources that previously were dedicated to forming batches and scheduling them manually and use them for other projects. Within a couple of days of deployment we had other areas asking for a scheduling system. Within a couple weeks we had all internal factories requesting the same system. This system opened a new era for how scheduling and the data that now becomes available within seconds (that took hours to process before) can be used in manufacturing. One challenge that we were able to convert to an opportunity has been the critique that the project deployment received from people who were resistant to significant changes to existing scheduling protocols and procedures. In this project, even those people are now giving ideas to the development team for next steps and enhancements. In fact, the system has been so favorably/enthusiastically received that over 100 additional enhancement requests for extending the system capability in specific ways have been submitted by a combination of the manufacturing and engineering teams in just a few weeks following the deployment. Most of these have been addressed in newer versions that included these enhancements and bug fixes.

5.3 Next Steps

The team will engage in a phase II version of diffusion scheduling allowing us to cover the remaining 10% of scenarios as well as provide additional focus on maintenance tasks that have to be completed on equipment. After phase II is completed the team will continue to rollout the new FaST application to the other factories at Freescale (FSL). Below is a list of the future enhancements planned for phase II:

- Schedule when an equipment should be taken down for preventative maintenance. Since the scheduler knows when lots are scheduled to run on each equipment, it can opportunistically schedule maintenance tasks between batch runs.
- Schedule lots that will be coming into diffusion in the next few hours.
- Enhance algorithm to reduce recipes becoming unqualified because they have not run on an equipment within a certain number of days.
- Enhance algorithm to look at WIP levels outside of diffusion and push lots to areas that need WIP.
- Integrate scheduling with the material handling system so that lots can be moved to their scheduled area seamlessly.

ACKNOWLEDGMENTS

We would like to acknowledge Freescale ATMC Manufacturing Leadership Team and Freescale CIM Leadership Team for their support during the life of this project.

REFERENCES

Albers, S., and P. Brucker. 1993. The complexity of one-equipment batching problems. *Discrete Applied Mathematics* 47(2):87-107.

Azizoglu, M. and S. Webster. 2001. Scheduling a batch processing equipment with incompatible job families. *Computers and Industrial Engineering* 39(3):325-335.

Balasubramanian, H., L. Monch, J.F. Fowler, and M.E. Pfund. Genetic algorithm based scheduling of parallel batch equipment with incompatible lot families to minimize total weighted tardiness. *International Journal of Production Research* 42(8):1621-1638.

Chandru, V., C.Y. Lee, and R. Uzsoy. 1993. Minimizing total completion time on a batch processing equipment with job families. *Operations Research Letters* 13(2):61-65.

Chandru, V., C.Y. Lee, and R. Uzsoy. 1993. Minimizing total completion time on batch processing equipment. *International Journal of Production Research* 31(9):2097-2121.

Cheng, T.C.E. and M.Y. Kovalyov. 2001. Single equipment batch scheduling with sequential lot processing. *IIE Transactions* 33(5):413-420.

Coffman, E.G., A. Nozari, and M. Yannakakis. 1989. Optimal scheduling of products with two subassemblies on a single equipment. *Operations Research* 37(3):426-436.

Coffman, E.G., M. Yannakakis, M.J. Magazine, and C.A. Santos. Batch sizing and lot sequencing on a single equipment. *Annals of Operations Research*, 26(1-4):135-147.

Dobson, G., and R.S. Nambimadom. 2001. The batch loading and scheduling problem*.* *Operations Research* 49(1):52-65.

Fowler, J.W., G.L. Hogg, and D.T. Phillips. 2000. Control of multiproduct bulk server diffusion/oxidation processes. Part 2: Multiple servers. *IIE Transactions* 32(2):167-176.

Ghazvini, F.J. and L. DuPont. 1998. Minimizing mean flow times criteria on a single batch processing equipment with non-identical lots sizes. *International Journal of Production Economics* 55(3):273-280.

Gurnani, H., R. Anupindi, and R. Akella. 1992. Control of batch processing systems in semiconductor wafer fabrication facilities. *IEEE Transactions on Semiconductor Manufacturing* 5(4):319-328.

Hochbaum, D.S., and D. Landy. 1994. Scheduling with batching: Minimizing the weighted number of tardy lots. *Operations Research Letters* 16(2):79-86.

Horiguch, K., N. Raghavan, R. Uzsoy, and S.Venkateswaran. Finite-capacity production planning algorithms for a semiconductor wafer fabrication facility. *International Journal of Production Research* 39(5):825-842.

Mehta, S.V., and R. Uzsoy. 1998. Minimizing total tardiness on a batch processing equipment with incompatible lot families. *IIE Transactions* 30(2):165-178.

Monch, L., H. Balasubramanian, J.F. Fowler, and M.E. Pfund. Heuristic scheduling of lots on parallel batch equipment with incompatible lot families and unequal ready times. *Computers and Operations Research* 32(11):2731-2750.

Perez, I.C., J.W. Fowler, and W.M. Carlyle. 2005, Minimizing total weighted tardiness on a single batch process equipment with incompatible job families. *Computers and Operations Research* 32(2):327-341.

Potts, C.N., and M.Y. Kovalyov. 2000. Scheduling with batching: A review. *European Journal of Operational Researc*h 120(2):228-249.

Tanrisever, F. and E. Kutanoglu. 2007. Forming and scheduling lots with capacitated constraints in semiconductor manufacturing: Single equipment problem. *Annals of Operations Research* 159(1):5-24.

Uzsoy, R. 1994. Scheduling a single batch processing equipment with non-identical lot sizes. *International Journal of Production Research* 32(7):1615-1635.

Uzsoy, R. 1995. Scheduling batch processing equipment with incompatible lot families. *International Journal of Production Research* 33(10):2685-2708.

Weng, W.W., and R.C. Leachman. 1993. An improved methodology for real-time production decisions at batch-process work stations. *IEEE Transactions on Semiconductor Manufacturing* 6(3):219-225.

Yuan, J.J., A.F. Yang, and T.C.E. Cheng. A note on the single equipment serial batching scheduling problem to minimize maximum lateness with identical processing times. *European Journal of Operational Research* 158(2):525-528.

AUTHOR BIOGRAPHIES

TANJU YURTSEVER is a Senior CIM Systems Architect at Freescale Semiconductor. He received his B.S. and M.S. in Industrial Engineering from Middle East Technical University, Ankara, Turkey and M. Eng. in Information Systems from University of Toronto, Canada. He has designed and developed Real Time Graphical Manufacturing Monitoring System (GraMMS), Simulation based Dispatching System and advanced features for Semiconductor Manufacturing Execution Systems. Last two years, he is concentrated on heuristic based scheduling systems for Semiconductor Manufacturing. His email is <tanju.yurtsever@freescale.com>.

ERHAN KUTANOGLU is an Associate Professor in the Operations Research and Industrial Engineering Graduate Program in the Department of Mechanical Engineering at The University of Texas at Austin. Before his current position, he worked as Operations Research Analyst at IBM Global Services, and taught at the University of Arkansas. He received his Ph.D. in Industrial Engineering at Lehigh University, and his M.S. and B.S. degrees, both in IE, at Bilkent University, Turkey. His research interests include large-scale optimization and scalable, distributed, and game-theoretic models and algorithms with applications to scheduling, inventory, logistics network design, and supply chain problems. His current industrial focus is post-sales service parts logistics and semiconductor manufacturing. His email is <erhank@me.utexas.edu>.

JENNIFER A. JOHNS is a Service Deliver Manager and Software Engineer at Freescale Semiconductor. She graduated from Texas A&M in 1999 with a degree in Information and Operations Management. The past 9 years she has been implementing and developing productivity improvement projects in the semiconductor industry. Her email is <jennifer.johns@freescale.com>