SEMICONDUCTOR MANUFACTURING SCHEDULING

In this article we will present the key ideas behind scheduling semiconductor manufacturing operations. In particular, we will concentrate on scheduling semiconductor wafer fabrication plants (fabs). Modern fabs require capital investment in plant and equipment of nearly $1 billion. This makes them the most costly manufacturing plants today. In addition, the semiconductor industry is extremely competitive, and the processes involved in wafer fabrication are exceedingly complex. As a consequence, efficient manufacturing is essential for economic success in this industry. One of the key components of efficient manufacturing of semiconductor wafers is good fab scheduling.

Scheduling Decisions

In the context of manufacturing systems, the term scheduling refers to the control of the flow of in-process material (commonly termed work in process inventory or WIP) on the factory floor. Typically, the flow is controlled so as to achieve production targets set by the production planning and control department, while attempting to optimize some performance measures.

Many fabs are run in *make-to-order* fashion. That is, production is initiated by customer orders. Other fabs, particularly those making memory (which is fast becoming a commodity), run as *make-to-stock.* Here production is according to plan, and customer orders are served from the finished goods inventory built up using the plan. In either case, the production planning and control department usually sets the production targets based on the number of outstanding customer orders (the backlog), or the level of the finished goods inventory required. These targets have to be met in a timely and efficient manner by the shop floor control (SFC) system. This system initiates production of new wafers and also tracks and controls the flow of work in process so as to achieve the production targets in the shortest possible time, while utilizing the plant, equipment and workforce in the most efficient manner. The decisions implemented by the SFC system are broadly characterized as scheduling.

Scheduling involves taking a variety of decisions regarding the flow of WIP on the shop floor. These decisions are taken based on planning as well as the current status of the fab. For example, deciding when to release a new set of wafers into the fab will be based on both the production targets set for that month, and the number of wafers of that type currently waiting to complete processing in the fab. Typically, scheduling a fab involves the following types of decisions.

- *Work release.* Deciding the release of a new set of wafers to begin processing in the fab. This involves both a timing decision, that is, when to release the new set of wafers, as well as a choice of type decision, that is, which one of the many types of wafers processed at the facility must be released next. The latter decision regarding the type of wafer released determines the *product-mix,* or the proportions of the various types of wafers in the fab. The different types of wafers may also differ from each other in the sequence of the operations that must be performed. For example, they may have different numbers of layers. In this case of multiple *processes,* all the scheduling decisions will have to take into account the different processes involved.
- *Routing.* More than one tool can be used to perform a particular step in the processing of a wafer. Deciding which one of the many tools capable of performing the step to send a wafer to, (called routing) is an important scheduling decision. Typically, these decisions are not planned in advance, but are taken dynamically when the wafer completes the previous processing step, based on which tools have failed, the workload on the tools, etc. In many fabs, many identical or similar tools are grouped together at stations. In that case, the routing decision involves deciding which one of the many tools at a station will be used to perform the given processing step.
- *Sequencing.* An important decision which influences both the flow of WIP as well as the utilization of the various machines involved in processing (typically called tools in fabs), is deciding when to work on a particular wafer at a particular tool. This involves developing a schedule for processing at that tool. Developing a detailed schedule which maps out exactly when each wafer will be processed at each tool is an onerous task, and the schedule can be thrown into complete disarray by the slightest variation in processing times, by tool failures, etc. As a result, detailed schedules are usually not implemented. Rather, rules are put in place which decide the sequence in which the wafers waiting to be processed at a given tool are taken up for processing by the tool. One way to implement such a sequence is to have rules for deciding which among the waiting wafers will be processed *next.* These rules are called *sequencing rules.* An example of a sequencing rule would be first-come-first-serve (FCFS) which picks that wafer which arrived first among those waiting for processing.
- *Lot sizing.* Wafers are released into the fabs in sets of a prescribed size known as lots. These wafers in the lot travel together between processing steps, although they may be processed individually at a tool. Deciding the size of a lot is one of the scheduling decisions.
- *Batching.* Some tools process more than one wafer at a time. Such tools are called batch tools. Deciding how many wafers to load at a given time into a batch tool is also one of the scheduling decisions.
- *Work-force scheduling.* The decisions involved in alloting operators to tools is also part of the scheduling function. There are sets of operators, and operators in a set can handle a set of tools. Also, operators may only be needed to load and unload wafers and to set-up a tool for a particular step. Deciding how to efficiently schedule the work-force subject to these constraints is an important scheduling function. The operator schedules are usually more static than the sequencing rules described above for scheduling wafers. They are similar to time-tables, and are usually made up in advance for a shift or longer periods.
- *Preventive maintenance scheduling.* The expensive and complex tools used in fabs require periodic preventive maintenance, in order to minimize the possibility of unplanned downtime due to tool failure. Scheduling preventive maintenance so that the production is minimally disrupted is important.

The scheduling decisions just outlined have to be made as to optimize the trade-offs between various performance measures of interest.

Performance Measures

In order to utilize invested capital properly, one must utilize plant and equipment efficiently. At the same time, one must not overload theplant with excessive work-in-process inventories. A basic requirement is to start filling as much of the plant's backlog as possible in a given period, in the shortest possible time. Thus, there are many dimensions of performance of a fab scheduling policy. First, let us characterize the various metrics by which the performance of a scheduling pol-

icy is measured. Then we will attempt to give a representative picture of the various trade-offs that exist between these performance measures, and how one has to juggle these conflicting dimensions of performance to schedule the fab efficiently.

The following performance metrics are typically used (1) to evaluate the efficacy of fabs as a whole, and scheduling policies in particular.

- *Line yield.* This is the fraction of the wafers started that emerge as completed defect-free wafers from the fab. This metric is influenced more by the maturity of the technology employed, and the quality control programs employed than by the scheduling policies. One can also talk of the yield of a specific processing step in a similar fashion.
- *Throughput rate.* Also called just *throughput,* this is the number of completed wafers exiting the fab in a given period, measured, for example, in wafers per day. If the line yield is 100%, then this is also the rate of *wafer starts* into the fab. In general, throughput rates are defined separately for each type of wafer processed in the fab. If there is more than one wafer type made in the fab, then the throughput rate is a vector with each component representing the throughput for the respective type.
- *Throughput capacity of a fab.* This is the maximum sustainable throughput rate of a fab operating under a given scheduling policy. This is a fundamental limit to the achievable performance of the fab, and is determined by the throughput capacity, that is, the maximum sustainable throughput rate, of each processing station considered in isolation. The throughput capacity of a fab is equal to the *smallest* of the throughput capacities of the individual stations. This is akin to the strength of a chain being determined by its weakest link. The station with the smallest throughput capacity is called the *bottleneck.* There may be more than bottleneck in a fab. Various factors determine the throughput capacity of an individual station and hence determine the throughput capacity of the fab:

  Product mix. If different types of wafers require different amounts of time to complete the processing steps at a station, then the throughput capacity of the station is determined by the relative proportions of these types, that is, the product mix.

  Yield. As explained earlier, the fraction of wafers that do not successfully complete a processing step do not contribute to throughput. Hence, the lower the yield of a processing step, the lower the throughput capacity of the station performing that step.

  Task time. Total time taken to perform all tasks involved in all the processing steps carried out at that particular station. The throughput capacity is inversely proportional to this time. The total task time consists of the time taken to load/unload the wafer (or a set of wafers in batch tools); the time taken to set-up for that particular processing step; and the time taken to perform the actual processing.

  Lot size. A tool which processes one wafer at a time (called a single-wafer tool) will perform as many load/unload and processing operations as there are wafers in the lot. However, since all the wafers in a lot undergo the same processing step, only one set-up needs to be performed for the entire lot. So the lot-size determines the total task time per wafer, and hence the throughput capacity.

  Number of tools. The throughput capacity of a station is proportional to the number of identical tools available at the station.

  Tool failures. The time available for processing by a tool is limited by failures and the consequent time to repair these tools. Such failures may be hard failures or just soft failures caused by a tool performance drifting out of its specified limits. These failures limit the number of wafers that these tools can process in a given period, and hence their throughput capacity. Failures in fabs are of two types. *Autonomous* failures which are independent of the usage of the tool (steppers, which perform the photolithography operations, typically fail in this fashion) and *operational* failures whose frequency is proportional to the usage of the tool (ion implanters typically fail at a rate proportional to the number of hours they are used).

  Preventive maintenance. The time available for processing by a tool is also limited by the duration of preventive maintenance carried out on the tool.

  Batching. The capacity of a batch tool can be fully realized only if the number of wafers loaded into the tool is equal to the maximum number of wafers the tool can handle. If, on the average, the number of wafers simultaneously processed by the tool is less than the maximum number it can handle, its throughput capacity is proportionally reduced.
- *Utilization.* The throughput rate in a fab cannot exceed the throughput capacity of the constraining station, the bottleneck. As a consequence other stations are *idle,* that is not being loaded/unloaded, being set-up, or processing, for a fraction of the time. Also, stations can be in a failed nonfunctional mode for part of the time. The fraction of the time a station is *not* idle is called the utilization of the station. In order to fully exploit the capital invested in obtaining the tools at that station, it is desirable to minimize the idleness of the station. Also, under most absorption costing based managerial accounting systems, the cost of goods sold is reduced by having high utilization (i.e., close to one).
- *Throughput time, lead time or cycle time.* All these terms are used to denote the total time taken by a wafer from when it is released into the fab to when it emerges from the fab as a completed wafer. This measures the responsiveness of the fab as well as its ability to achieve on-time delivery. Long throughput times also increase the time for which the wafer is exposed to potential contamination in the fab, and hence can result in lower yield.
- *Work in process (WIP) inventories.* WIP inventory is the number of wafers which are still in the fab at various stages of processing. WIP inventory represents working capital tied up in the fab. Large WIP inventories also result in slower detection of quality problems, and general sluggishness of the fab.

Relationships Between Performance Measures

The various performance measures already described, namely the throughput rate, utilization, throughput time, and WIP inventory are all related to each other. For the purpose of illustration, we will consider a fab making just one type of wafers with a line yield of 100%. Then, let the start rate of wafers equal the throughput rate equal λ.

When the throughput rate λ is fixed, the long term average WIP in the system L is directly proportional to the long term average lead time W. In fact,

$$L = \lambda W$$

This means that a high average WIP results from having a high average lead time and vice-versa. This is quite intuitive because we would expect a wafer entering a fab with a lot of WIP inventory built up in front of it to take longer to exit from a fab than a wafer entering a relatively empty fab. This relationship holds in great generality (for example, the relationship holds for each type of wafer made in a fab manufacturing many kinds of wafers), and is called *Little's law*. Little's law tells us that the goal for efficient fab management is the same for both WIP and lead time—reduce one without changing the throughput rate, and you automatically reduce the other.

The average lead time in a fab is proportional to the sum of the total task times for each of the processing steps required for completing the processing of the wafer. However, the constant of proportionality in this relationship, sometimes called the *actual-to-theoretical ratio,* is usually large. Typically, the sum of the task times is of the order of a few days, while the average lead time is of the order of a few weeks (actual-to-theoretical ratios in the range of 2.5–10 are common).

Increasing the throughput rate in a fab results in increasing the utilization of each of the stations in the fab. The relationship between throughput rate, and consequently utilization of stations, and the average lead time in a fab is not so intuitive. Increasing utilization by a small amount when it is already close to one has the effect of increasing the actual-to-theoretical ratio and consequently the average lead time by a disproportionately large amount. The key driver for this nonlinear effect is *variability* (2). There are many sources of variability in a fab. There is variability in demand, and consequently in the wafer starts in a given period. There is variability in the time taken to complete the tasks for a particular step due to (1) variability in set-up, load/unload and processing times due to operator assists, (2) random machine failures and variability in the consequent repair time, and (3) variability in yield. The greater the degree of variability (as measured by the ratio of the standard deviation of the underlying distribution to the mean, commonly called the coefficient of variation) the greater the nonlinearity in the relationship. The effect of utilization and variability on lead time is representatively sketched in Fig. 1.

Figure 1 illustrates the difficult tradeoff that must be optimized while scheduling fabs. On one hand, we need to increase throughput and utilization as much as possible. On the other hand, we have to minimize the actual-to-theoretical ratio as much as possible. However, we cannot improve one of the objectives without giving up some of the other. This will be the recurring theme in the design of scheduling policies.

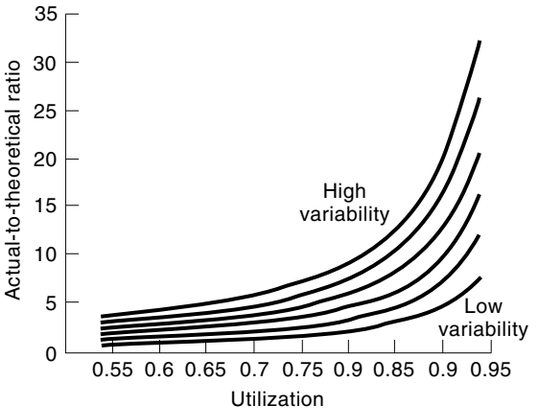

Figure 1. Relationship between utilization and cycle time.

Scheduling Difficulties Particular to Wafer Fabs

Wafer fabs have certain characteristics which make them particularly hard to schedule. The most important of these is the complexity of process flow. The production of wafers involves several hundred processing steps. These steps consist of similar operations which are repeated for each layer. For example, the expose step in photolithography is repeated for each of the 15 or so layers that form a very large scale integration (VLSI) chip. The economic necessity of reducing capital investment, as well as some technological requirements, force sharing of equipment between lots which differ in the layer being processed. That is, the same stepper may be used to expose wafers at different layers. As a consequence the flow of wafers in a fab is a complex re-entrant line. A representative re-entrant line is shown in Fig. 2. It is seen that wafers repeatedly return to the same station for the processing of subsequent layers. In the standard manufacturing process spectrum, the re-entrant line topology of wafer fabs places them somewhere in between classical line flow manufacturing systems and classical job shops.

This re-entrant nature of the fab makes local decision-making suboptimal. In fact, a reasonable decision from the local perspective of an individual station can prove disastrous from the global perspective of the entire fab. In the next section we motivate the design of global policies using an example of a system where a policy designed from a greedy local perspective proves disastrous.

The other characteristic of wafer fabs that makes them hard to schedule (3) is the diversity of equipment. The equipment (or tools, as they are often called) vary widely. Some of the tools process wafers one at a time. Such serial processing could involve significant set-up and changeover times. Other tools called batch tools process a batch of wafers at the same time. An example of a batch tool is a well-drive furnace. This tool is used to drive implanted impurities to various depths by heating. Such a batch tool will have to be scheduled intelligently so as to ensure that its capacity to process many wafers simultaneously is maximally utilized. Tools are not completely reliable, and they fail periodically. Some tools, especially those operating in a high vacuum environment like

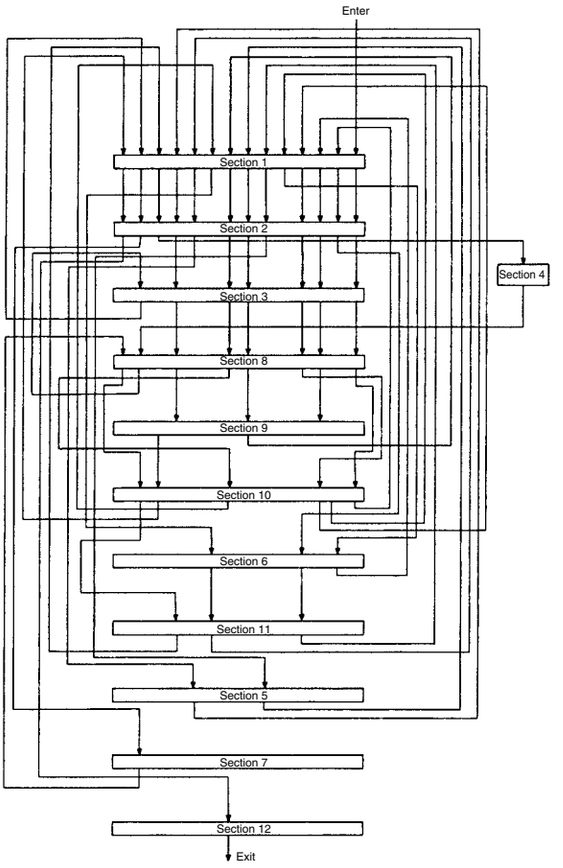

Figure 2. A representative re-entrant line.

physical vapor deposition (used to deposit metal on the surface of a wafer), may take a long time to repair when they fail. The effect of these failures on the rest of the fab must be taken into account when designing scheduling policies.

Finally, many fabs operate in a make-to-order environment. That is, they begin production only to satisfy an outstanding customer order. There is usually a due-date before which the customer order needs to be filled. In such fabs, poor yield can cause serious scheduling difficulties. A lot destined for a customer being scrapped near the very end of its processing, when it is also very close to its due-date, will result in the scheduling rules having to be broken to expedite a new lot to fill the order. If such exceptions are frequent, the design of scheduling policies for the fab will have to take the poor yield and scrapping into account, further complicating an already difficult task.

SCHEDULING SEMICONDUCTOR FABS

In this section, we discuss a representative set of scheduling rules which have been developed in particular for scheduling semiconductor fabs. We shall do this by first introducing various rules which have been used for general job shop type manufacturing systems. Then we shall point out the difficulties with using these policies in a semiconductor fab, and thus motivate the need for designing policies specially for semiconductor fabs. We begin by describing scheduling rules which have been used for many years in job shop manufacturing settings. Some of these rules continue to be used in wafer fabs today.

Common Sequencing Rules

There are a wide variety of sequencing rules which have been developed for general job shop type manufacturing systems (4). Most of these rules have been developed using heuristics which attempt to control either (1) the configuration of the WIP inventory and/or (2) the material flows within the manufacturing shop floor, as a way of attempting to optimize the trade-off between throughput, lead time, and WIP inventory on the shop. As a brief introduction to the vast array of available rules, we present a short list of representative sequencing rules and the rationale behind each of them.

To recall, sequencing rules are policies which decide which of the wafer lots waiting for processing at a tool is to be processed next at that tool. The one all of us understand and know about is the first in first out (FIFO) rule. This rule picks that lot which has waited at the tool the longest for service. Another popular policy is the shortest processing time rule. This rule picks that wafer lot which has the least amount of processing time requirement from that tool. The rationale is that one wishes to get the short jobs out to the next processing step as quickly as possible. Alternately one can think of getting lots out of the *entire system* as soon as possible. This is motivated by the desire to reduce the throughput time of the jobs. One way to try and achieve this is to choose a scheduling rule which picks that lot for processing which has the least amount of total processing left before it exits the entire system. This rule is called the shortest remaining processing time rule. On the other hand, one can argue that at any station, attention must be given to that lot that requires the maximum amount of work from the tool, before attending to shorter jobs. This results in the longest processing time rule.

Another set of flow control based sequencing rules are the least slack policies. These policies take into account the due dates of the wafer lots, and give priority to those policies to those lots that have the least amount of slack, that is, that are closest to the due dates (or the most past due). Another sequencing rule that is used both in classical job shops and wafer fabs is the critical ratio rule. In this rule, one gives priority to the lot with the smallest ratio of slack time to the number of remaining processing steps. As one can imagine, the number of such heuristics is tremendous. Rather than attempt an exhaustive survey, we will conclude with one more interesting heuristic sequencing rule. The rules we have described above all attempt to regulate the flow of wafer lots on the fab floor. We can also think of policies which attempt to regulate the inventory levels at each of the stations. One heuristic which does this is the least work next queue rule, where priority is given to the wafer lot that, on completion of processing, will join the queue of waiting lots at the next processing step which has the least amount of work waiting to be done. Thus, this rule attempts to regulate WIP inventories at the next, downstream station and provide work for that station which is most likely to be starved.

Some sequencing rules are designed to mitigate the impact of set-ups. One way to minimize the impact of set-ups is to serve all the wafer lots which can be processed using the current set-up, until no more such lots are available for pro-

cessing at the tool, before switching to processing another type of wafer and thus having to do a set-up. This is the serve to exhaustion or clearing rule.

Another set of commonly used scheduling rules worth discussing are the batching rules. Recall that the batching decision involves deciding when and how many wafers to load into a batch tool for simultaneous processing. The trade-off is whether to start as soon as possible and possibly run the batch tool with fewer wafers than the tool is capable of handling simultaneously or to wait until enough wafers have accumulated at the tool to fully utilize the capacity of the tool, at the risk of increasing the delay experienced by the wafers. One commonly used batching rule is the limited look-ahead rule, where one waits to see if there are any wafers arriving in the near future (up to a limited time horizon) before loading and starting up the batch tool.

Common Work Release Policies

Scheduling policies also attempt to regulate the flow of work onto the factory floor in an attempt to optimize the trade-off between throughput rate and cycle-time discussed in the previous section. In this subsection we discuss some common release policies to illustrate the various issues which must be grappled with in designing such policies.

In designing release policies, one must try and achieve the throughput rate required to achieve the quotas set by the production planning and control function (or, equivalently, to make sure that the backlog of customer orders does not grow without bound) while still maintaining a small amount of WIP in the fab and keeping the mean cycle time small. One can just release work into the system as it arrives and thus buildup WIP on the fab floor. Arguably, it is better to keep the inventories on paper, that is, as a pending order waiting to be released onto the fab floor than as WIP in the fab. Then these pending orders can be released along with the required raw material (in this case, a raw wafer) according to some mechanism which improves the performance of the fab (see Fig. 3). Although the order spends some time in the paper queue, and thus increases the time taken to fill that order, it is hoped that the decreased cycle time on the fab floor due to the release control mechanism will more than compensate for this.

One common release control mechanism used in general job shops is deterministic release. Here the orders are released onto the shop floor only at periodic intervals. This has the advantage of removing one potential source of variability from the system. This is an example of a release policy which attempts to regulate flows in the system. One could also conceive of release policies which attempt to regulate WIP on the fab floor. One such policy is the CONWIP (2) policy, also known as the Closed Loop release policy (5). CONWIP (which

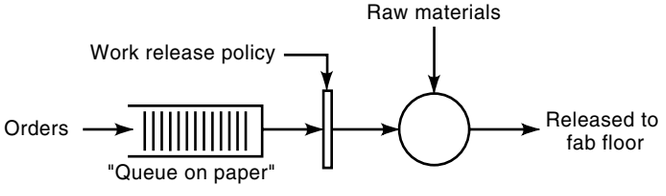

Figure 3. The lot release architecture.

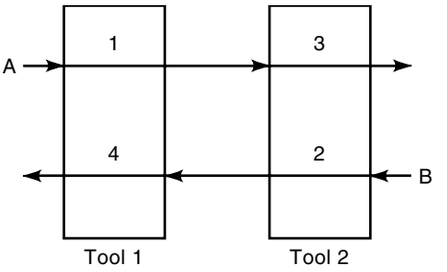

Figure 4. Stylized example of re-entrant line.

stands for constant WIP) explicitly controls the amount of WIP inventory on floor. It maintains the level of total WIP constant. This policy is usually implemented by releasing a new wafer lot onto the floor only when a completed lot of the same type leaves the floor. One attempts to match the throughput rate required to achieve the production targets by increasing the constant level of WIP being maintained. Increasing the WIP usually increases the throughput rate, but it also increases the cycle time as well. Thus a balance needs to be struck between the allowed WIP level and the target throughput rate.

An extension of this policy is to explicitly maintain the level of WIP constant at every processing step. One way to do this is to allow a transfer of a lot from one process step to its succeeding processing step only when the succeeding step completes a transfer. That is, the downstream step *pulls* work in from the upstream step as it completes and delivers its own work further downstream. This method of WIP control was popularized by the Japanese automobile industry (6) and is called the Kanban system.

In fabs with one clearly identified bottleneck step, one can release work into the fab such that the WIP upstream of the bottleneck step is held constant in a fashion similar to CONWIP. The rest of the steps downstream of the bottleneck can be paced by the bottleneck. This release mechanism is called drum-buffer-rope and was popularized by Eliyahu Goldratt. In re-entrant lines, where the bottleneck resource is revisited for many process steps, this rule has to be suitably adapted. Rather than discuss this further, we will discuss an alternative approach to allowing the bottleneck to pace work release into the fab in the next section.

Motivation for Designing Policies for Fabs

In this subsection we will motivate the need for designing scheduling policies especially for wafer fab scheduling, using a highly idealized re-entrant segment of a fab. This example is motivated by examples presented in (7,8). Consider the re-entrant line segment shown in Fig. 4. This can be seen as a highly idealized caricature of a segment of a fab, with two single wafer tools performing four processing steps (1, 2, 3, and 4) on two types of lots, A and B. Steps 1 and 3 are required to complete processing for type A lots, and 2 and 4 for type B lots (there are two processes in this fab). Processing steps 1 and 4 are performed on tool 1 and steps 2 and 3 on tool 2. Assume for simplicity that the lot size is 1, that is, there is 1 wafer per lot. The processing times for steps 1 and 3 are exactly 1 h and those for 2 and 4 are variable with a mean processing time of 10 min.

In the spirit of the shortest remaining processing time rule, processing steps 4 and 3 are given priority at tools 1 and 2 respectively, since they correspond to exit steps. The release policy is deterministic, and lots of both type A and type B wafers are released into the system periodically at 75 min intervals. The total WIP in the system is plotted versus time for a simulation run which is plotted in Fig. 5.

As we can see from Fig. 5, the WIP inventory increases without bound. This is definitely not what could be predicted from a naive analysis of the situation presented here. For example, each pair of wafers of type A and B entering the system brings with it 70 min worth of work for tool 1 (since step 1 takes 1 h and step 2 takes 10 min) and since wafer pairs come in every 75 min, one expects that tool 1 will be capable of handling this work and would be busy about 70/75 or 93.3% of the time. but this is not the case. The reason for this bizarre behavior is the highly re-entrant nature of the flow, combined with a poor choice of scheduling policies. The priority policy causes alternative blocking and starvation of the tools, resulting in WIP increasing without bound because the tools loose too large a fraction of their time being starved for work to be able to complete the workload imposed on them.

Although this example is in a very simple setting, it's moral carries over to real fabs—a naive choice of scheduling policies combined with re-entrant line flow could result in very nonintuitive and undesirable behavior. This motivates the need for better policy design for scheduling wafer fabs, which take the special features of the wafer fab into account.

The Workload Regulation Release Policy

In this and the next subsection we will present two policies which have been specially designed for scheduling semiconductor wafer fabs. In the next section we present the results for a simulation case study of scheduling a wafer fab where these two policies are compared against the common policies described in the previous section. This, we hope, will convince the reader of the benefits of designing policies specially for fabs. We begin by describing a work release policy which is due to Wein (9).

The key to input regulation, that is, deciding when to release wafer lots onto the fab is the idea of a bottleneck. The bottleneck is that station (or stations) in a fab which is utilized the most under a given set of throughput rates and a given product mix. A fab may have more than one bottleneck.

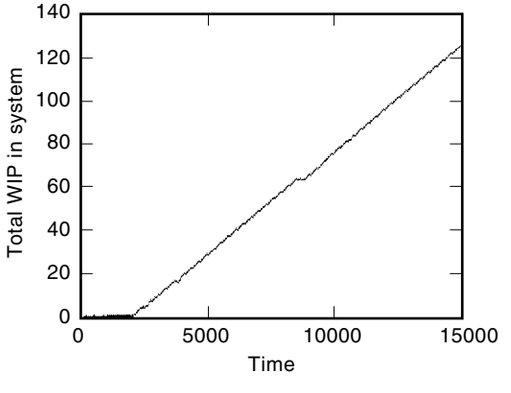

Figure 5. The total WIP trajectory in the example.

In order that the fab be able to handle the demand placed on it, it is necessary that every tool in the system be utilized less than one hundred percent of the time. As we have seen from Fig. 2, utilizing the bottlenecks too close to capacity can result in excessively long cycle times in the presence of variability. So the capacity of the entire fab is determined by the level of utilization of the bottlenecks when there is an upper limit to the acceptable average cycle times. This leads to the idea of cycle time constrained capacity, where the inability to accept very long mean cycle times restricts the permissible throughput rate.

It is intuitive that the bottlenecks should determine the flow of work into the system. On the one hand, we want to make sure that a bottleneck is never starved for work as it is among the critical resources in the system. Such starvation will lead to a later bunching up of subsequent lots, and thus to higher cycle times. On the other hand we do not wish to buildup excessive WIP in front of the bottleneck, thus leading to excessively long cycle times. The workload regulation release policy achieves this balance by releasing new work into the system only when the total work in the system, remaining to be done by the bottleneck tools in order to get rid of all of the current WIP in the system, is in a particular configuration. The particular choice of the WIP configuration can be chosen in many ways, and each one of them leads to a different workload release policy.

For simplicity we shall present the workload regulation policies in the setting of a fab with a single process and single product type. In a single bottleneck fab, one can choose to release work into the fab only when the total work which must be completed by the bottleneck, in order to get rid of all of the current WIP in the system, is less than a threshold A. We shall call this policy workload regulation policy WR(A). The total work yet to be completed by the bottleneck M can be calculated as

$$M = \sum_{i=1}^{S} m_i X_{i,j}$$

where S is the number of processing steps, m_i is the amount of work to be done by the bottleneck on a lot in processing step i before it exits the system, and X_i is the number of lots currently at processing step i. The $WR(A)$ policy then releases work into the system only when $M \leq A$. The choice of A determines the throughput rate which will be sustained under this release policy, and thus will have to be tuned to match the rate required to ensure that the backlog of orders does not grow without bound.

When there is more than one bottleneck in the fab, we can adapt the workload regulation policy described above for the one bottleneck case in many ways. First, we could just replace the workload M by the sum of the workloads for each of the bottlenecks, and then pick a new threshold A which reflects this as well. This approach does not differentiate between the bottlenecks and so the interactions between the bottlenecks are ignored. This may not be such a good idea. Alternately, we could replace the single index A by multiple indices where we explicitly track the workload for each of the machines and compare these against individual thresholds and release work into the system when any one of the workloads falls below its respective threshold.

The Fluctuation Smoothing Sequencing Rules

Having discussed a release policy designed specifically for semiconductor fabs, let us now discuss a sequencing rule which was also designed especially for semiconductor by Kumar and co-workers (10).

The sequencing rule we discuss is a variant of the least slack rule described in the previous section. The main idea here is that the due-date associated with a lot can be modified by the sequencing rule in such a way that the overall performance of the fab, as measured by the average cycle time, is improved. Setting overly critical due-dates has the effect of disrupting the flow of lots in a fab. Lots with extremely low slack in the due-dates, commonly called *hot lots,* adversely affect the performance of the fab as a whole. They receive priority at every step, and as a consequence the majority of the lots in fab still awaiting processing suffer. It is worth examining whether the benefits gained from getting the hot lot out on time outweight the increased lead time suffered by the majority of lots. In a fab producing a small variety of parts, a case can be made that improving the overall performance of the fab in terms of average cycle time and WIP will improve the due date performance of the individual lots as well, especially when the due-dates are set in a rational fashion. This is the philosophy adopted in designing the sequencing rule we are about to discuss.

Suppose we were to ignore the actual due dates on the lots, and instead set due dates for each lot with the aim of improving overall fab performance. Some lots will be completed later than their due dates. However, if the original due dates were picked in a rational fashion, with every due-date being set as the date on which the order was placed plus a quoted lead time, and if the orders were released into the fab in the order in which they were received, then reducing the average cycle time would reduce the average lateness of a lot as well. So we could ignore the original due-dates in this case. The question now to be addressed is: what should the new due-dates set by the sequencing rule be?

We have seen that variability induces congestion and delay in manufacturing. One source of variability is the variability in the flows. In particular, it is the variability in the time between consecutive arrivals to every station in the fab. We propose a scheme for setting due dates which will simultaneously reduce burstiness of arrivals to each processing step, thus reducing variability in the flows. We do this by setting a due-date for reaching *each* processing step. Suppose λ is the target throughput rate, that is, the mean rate of release of new lots into the fab. For the nth lot being released into the fab, we can set the due date to reach step k as $d_k(n) = n/\lambda$. Then, if we reduce the variance of the lateness in reaching step k, that is, make lots uniformly early or late, we will reduce the burstiness of arrivals to step k. Let us now turn to reducing the variance of lateness in reaching step k. Suppose $e_k(n)$ is the time at which the nth part arrives at step k. The lateness of the nth lot in reaching step $l_k(n)$ is given by

$$l_k(n) = e_k(n) - d_k(n)$$

We will attempt to reduce the variance of lateness by implementing a variant of the least slack scheduling rule at each step i where we define slack of the nth part in reaching step k as

$$s_k(n) = d_k(n) - \zeta_{k,i}$$

where $\zeta_{k,i}$ is an estimate of the time remaining for a lot currently in step i until it reaches buffer k. If $\zeta_{k,i}$ is accurate, this results in a fair policy which attempts to make all lots arriving at step k equally early or late. We can also achieve the same results by implementing a least slack policy at each step i with slack for the nth lot defined as

$$s_i(n) = \frac{n}{\lambda} - \zeta_i$$

where ζ_i is an estimate of the time remaining until exit from the system for a lot currently in step i. This version of the least slack policy is independent of the choice of the step k. If we have accurate estimates of the delay parameters ζ_i, we hope to reduce the variability of arrivals to each step k and thus reduce the consequent delays, and hence the average cycle time in the fab. This sequencing rule is called the fluctuation smoothing policy for mean cycle time.

In the next section we will provide some evidence of the efficacy of the release policy and sequencing rule presented thus far. We will present a simulation case study of a representative wafer fab, and establish that the workload regulation release policy in combination with the fluctuation smoothing policy for mean cycle time does outperform many of the release policies and sequencing rules described in the previous section.

A Case Study

In this subsection, we present excerpts from a simulation case study of an R&D fab carried out first by Wein (9) and later by Kumar et al. (10). The fab has a single process comprising 172 operations carried out at 24 stations, each consisting of one or more identical tools or machines. Many of these stations are visited more than once.

As before, let λ be the target rate of release of wafer into the fab. The variability in the system is both in actual processing time (usually due to the involvement of an operator whose task times are not deterministic) as well as due to random failures of the machines. If MPT is the mean processing time, $MTBF$ the mean time between failures and $MTTR$ the mean time to repair, the utilization of each station (measured in hours of work per hour) is given by

$$\text{utilization} = \left[\frac{\lambda(\text{no. of visits})(MPT)}{\text{no. of machines}} + \frac{MTTR}{MTTR + MTBF}\right]$$

The data for each of the stations is presented in Table 1.

The target throughput desired to be achieved is $\lambda = 0.0236$ lots per hour. At this rate, the fab has one bottleneck, Station 14, which is utilized over 90% of the time. Three release policies described in the previous section are compared: deterministic release, the CONWIP release rule, and the workload regulation policy $WR(A)$ with A being the threshold for the work at Station 14 below which additional wafers are released into the system. Both the CONWIP level and the threshold are chosen so as to achieve the target throughput.

The fluctuation smoothing policy for mean cycle time (FSMCT) is compared against the first in first out (FIFO) se-

Table 1. Data for R&D Fab

Station	Machine Count	No. of Visits	MPT	MTBF	MTTR	Utilization (%)
1	2	19	1.55	42.18	2.22	39.8
2	2	5	4.98	101.11	10.00	38.4
3	2	5	5.45	113.25	5.21	37.0
4	1	3	4.68	103.74	12.56	43.9
5	1	1	6.14	100.55	6.99	21.0
6	1	2	7.76	113.25	5.21	41.4
7	1	1	6.23	16.78	4.38	35.4
8	1	3	4.35	13.22	3.43	51.4
9	1	2	4.71	10.59	3.74	48.3
10	1	3	4.05	47.53	12.71	49.8
11	1	1	7.86	52.67	19.78	46.2
12	1	2	6.10	72.57	9.43	40.3
13	4	13	4.23	22.37	1.15	37.3
14	3	12	7.82	21.76	4.81	91.9
15	1	15	0.87	387.2	12.80	34.0
16	2	11	2.96	∞	—	38.4
17	1	10	1.56	119.20	1.57	38.1
18	1	4	3.59	∞	—	33.9
19	2	2	13.88	46.38	17.32	60.1
20	1	2	5.41	36.58	9.49	46.1
21	2	4	7.58	36.58	9.49	56.4
22	2	21	1.04	118.92	1.08	26.7
23	2	23	1.09	∞	—	29.6
24	2	8	3.86	55.18	12.86	55.3

quencing rule and the shortest expected remaining processing time (SRPT) rule described in the previous section under each of the release policies already described. The performance metric used in the mean cycle time of wafers in the fab. The results are tabulated in Table 2.

It is evident that the combination of the workload regulation policy in combination with the FSMCT sequencing rule outperforms all other combinations of policies. Although we have not presented the exhaustive set of results that the authors cited have obtained, it can be seen that a carefully designed scheduling policy can result in substantial improvement in the performance of a fab, which in an industry as competitive and capital intensive as wafer fabrication can translate to substantial financial gains.

We do not want to leave the reader with the impression that these policies which have been designed and tuned using simulation studies can just be picked up and immediately implemented in a real production fab leading to instantaneous improvement in performance. So, in the next section, we discuss the implementation issues involved in scheduling wafer fabs.

IMPLEMENTATION

In this section we present some of the difficulties that must be dealt with before a scheduling policy can be successfully implemented in a fab. Then we present a generic example of commercially available software that allow us to overcome these difficulties, abstracted from a recent survey for SEMATECH (11).

Table 2. Cycle Time Performance Comparisons of R&D Fab

Policies	FIFO	SRPT	FSMCT
Deterministic	261.67	280.34	234.97
CONWIP	301.59	297.43	271.12
Workload Reg.	253.93	273.35	229.66

Difficulties with Implementing Scheduling Policies

Among all the difficulties with implementing scheduling policies in wafer fabs, the most important one is the need for information. Most scheduling policies have some informational requirements. Even the simple FIFO policy requires that the order of arrivals to a particular tool be known. Of course, this can be easily obtained by simply stacking the lots in the order in which they arrived. The shortest processing time and the shortest remaining processing time rules require that an accurate estimate of the time taken to perform each processing step be known. The workload regulation release policy and the least work next queue rule require the knowledge of the WIP at each of the processing steps at each instant of time, in addition to the processing time information. The FSMCT policy requires knowledge of the processing times, as well as an estimate of the time remaining until each wafer lot, at each processing step, exits from the system. Thus there is a need in most policies to know the parameters of the process like the processing steps and processing times, as well as to track the WIP on the shop floor.

These difficulties are further exacerbated by the dynamically changing environment in the fab. Tools are constantly failing, and their status needs to be monitored. The capacity of the overall fab, and the capacity of each station is also constantly changing, because of changes in yield. Yield improves as more is learnt about the process. This is particularly true when a new processing technology is implemented, and the fab is slowly ramped up to full production as processing bugs are ironed out. The scheduling policies have to be constantly tuned during this phase. Another factor which contributes to the dynamic nature of the fab environment is the change in product mix. The product lifetime in the semiconductor industry is only a small multiple of the cycle times in the fab. As a consequence, the mix of products being made in a fab changes constantly. The scheduling policies have to take this into account. For example, this means that in implementing workload regulation policies, we have to keep track of the bottlenecks as they might dynamically change as the product mix changes.

To summarize, the WIP in the fab must be constantly tracked, the processing equipment, yield,and product mix monitored, and the scheduling policies have to be periodically tuned to realize the maximum benefits of implementing the scheduling policies. All of these point toward the need for a computerized system with custom software. In the next section we will briefly describe such a system.

Scheduling Software

This subsection is based on a recent SEMATECH survey (11), which discusses a wide variety of commercial scheduling software packages in great detail. Rather than attempt to provide an exhaustive list of available packages, we will profile a generic package, whose modules exist in many of the commercial packages, as an illustrative example of what is available on the market.

A typical shop floor control package contains various modules that interact and perform the various functions required for efficient shop floor control. The lot scheduling module performs the scheduling function we have discussed in this article. This module is a real time system which performs lot sequencing and lot release among a host of other functions. It interacts with the other modules in the package such as the WIP tracking module to track the current status of the various lots, the resource tracking module to obtain the status of operators and equipment, thus providing the needed information for implementation of the various policies we have described in this article. It also forecasts lot completion times, thus for example, allowing us to track the delay estimates required for implementing FSMCT. It also provides statistics like WIP levels and resource utilization levels. Thus, we can obtain the needed information for identifying and tracking bottlenecks, and thus facilitating the implementation of the workload regulation release policies. It also keeps track of defects to allow adjustment of the yield estimates.

Thus, we can see that the implementation difficulties pointed out in the previous section can be mitigated to a large extent using appropriate software. However, there are costs of acquiring, implementing, and maintaining the software, but these costs are insignificant in comparison with the large capital investment in a wafer fab. Hence shop floor control software is quite prevalent in the semiconductor industry.

SUMMARY

In this article, we have described the scheduling function in semiconductor wafer fabs, and identified the key trade-off's to be evaluated in designing scheduling policies. We have surveyed some the sequencing rules and release polices used in semiconductor manufacturing, and presented examples of policies specially designed for wafer fabs. We have discussed the possible benefits of using such policies, and the issues involved in implementing them in a fab.

Several other detailed issues arise. We have not discussed the issues of routing, lotsizing, and batching in any detail. We have also restricted attention to sequencing rules and not discussed the more general scenario of schedule development which is essential for workforce scheduling and scheduling preventive maintenance.

Although this has been a limited introduction to the subject, the issues described here are sufficient for the reader to get acquainted with the basic ideas behind scheduling semiconductor manufacturing.

BIBLIOGRAPHY

1. R. C. Leachman and D. A. Hodges, Benchmarking semiconductor manufacturing, *IEEE Trans. Semicond. Manuf.,* **9**: 158–169, 1996.
2. W. J. Hopp and M. L. Spearman, *Factory Physics,* Chicago: Irwin, 1996.
3. R. Uzsoy, C.-Y. Lee, and L. A. Martin-Vega, A review of production planning and scheduling models in the semiconductor industry Part I, *IIE Trans. Scheduling Logistics,* **24** (4): 47–60, 1992, and Part II: Shop floor control, *IIE Trans. Scheduling Logistics,* **26**: 44–45, 1994.
4. S. S. Panwalker and W. Iskander, A survey of scheduling rules, *Operations Research,* **25** (1): 45–61, 1977.
5. C. R. Glassey and M. Resende, Closed-loop job release control for VLSI circuit manufacturing, *IEEE Trans. Semicond. Manuf.,* **1**: 147–153, 1988.
6. R. J. Schonberger, *Japanese Manufacturing Techniques,* New York: The Free Press, 1982.
7. A. N. Rybko and A. L. Stolyar, On the ergodicity of stochastic processes describing open queueing networks, *Problemy Peredachi Informatsii,* **28**: 2–26, 1991.
8. P. R. Kumar and T. I. Seidman, Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems, *IEEE Trans. Autom. Control,* **AC-35**: 289–298, 1990.
9. L. M. Wein, Scheduling semiconductor wafer fabrication, *IEEE Trans. Semicond. Manuf.,* **1**: 115–130, 1988.
10. S. C. H. Lu, D. Ramaswamy, and P. R. Kumar, Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants, *IEEE Trans. Semicond. Manuf.,* **7**: 374–388, 1994.
11. M. Arguello and E. Schorn, A survey of manufacturing scheduling software, *SEMATECH Technology Transfer,* 95012685A-XFR, 1995.

P. R. KUMAR
University of Illinois at Urbana-Champaign
SUNIL KUMAR
Stanford University