FabTime Newsletter

Volume 23, No. 6 December 2022

Information

Publisher: FabTime Inc. FabTime sells cycle time management software for wafer fab managers. FabTime's mission is to help the people who run fabs improve performance by 1) helping them to understand the factors that drive fab performance and giving them the data to identify current improvement opportunities; 2) letting them control that data by setting parameters for their own charts, so they don't have to go back to IT every time they want a different piece of information; and 3) including them in a community of people around the world who are all working to drive better fab operations.

Editor: Jennifer Robinson

Date: Tuesday, December 13, 2022 – Vol. 23, No. 6

Contributors: Richard Goldhammmer (Micron Technology)

Keywords: Variability; Performance Improvement; Fab Management

Table of Contents

Welcome

Welcome to Volume 23, Number 6 of the FabTime Cycle Time Management Newsletter. In this issue, we have an important announcement about a new platform (MailChimp) that we'll be using to send the newsletter and tip emails starting in January. You may need to mark the messages as safe to avoid them being junked by your email program (and we would of course appreciate that very much!). We also have an announcement about the newest version release of FabTime's software, and a few news stories shared from Jennifer's LinkedIn.

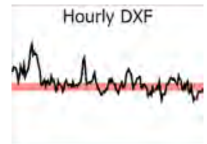
Our software tip of the month is about restricting the number of points displayed on a Pareto chart. Our subscriber discussion topics include green-to-green chart implementation, controlling variation in lot arrivals across factories, and (per last month's tip), separating maintenance events. A subscriber question about ranking of cycle time improvement tactics inspired us to share a proposed fab cycle time improvement framework as this month's main article. In this piece, we suggest what we think is a logical order for identifying cycle time improvement targets and then applying different strategies. We also include a list of the fab complexities that contribute to cycle time. We welcome your feedback, as always.

Wishing you a wonderful holiday season and a happy and productive 2023! – Jennifer, Frank, Lara, and the FabTime Team

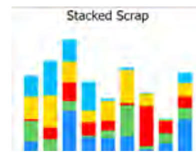
Community News/Announcements

FabTime Releases Patch 115

FabTime is pleased to announce the formal release of Patch 115 of our web-based cycle time improvement dashboard software. Following extensive testing by early adopters, the new version is now available to all sites with a current subscription or maintenance contract. Highlights of the new version include:



- Sparkline charts (very small charts with axes and most labels removed for legibility, allowing a dashboard to display many charts at once).
- New Column controls with the ability to control homepage tabs for more of a "dashboard" appearance: freeform, dynamic and fixed # of columns.
- Ability to set custom background images for homepage tabs.
- Ability to toggle homepage tab controls on or off for more focused display of charts.
- New Scrap Stacked Trend and Pareto charts.
- Functionality to copy and paste values from Excel or Notepad to filters, with commas automatically added.
- Fully updated FabTime User Guide with new topics and screen shots.



To find out more about the new version, please contact your internal FabTime sponsor or reach out to support@FabTime.com.

New Platform for Sending the Newsletter Issues

We are pleased to announce that we will be moving the newsletter distribution to a new platform, MailChimp. This change will allow us to add long-missing functionality to your subscriber experience, including one-click unsubscribe and the ability to directly update your email address. It will also make the process less time-intensive for Jennifer, so that she can focus on writing more content, rather than cleaning up bounce messages.

We will work to minimize change, but you may find that the issues look slightly different. You will need to click to download the PDF issues, instead of having them just arrive in your inbox. You may also need to check your spam filters the first time we send an issue using the new method and select "Never Block this Sender" to avoid future filtering. We thank you for your patience as we navigate this new system. Our target is to start with the January cycle time tip emails.

A Few Highlights from Jennifer's LinkedIn

Jennifer continues to share articles about business management, the semiconductor industry, and productivity improvement on her LinkedIn feed. Recent links have included:

- A widely reported story (MSN.com link here) about the wafer fab in Newport, Wales. After considerable back and forth over the past 2 years, the UK government has moved to block the fab's sale to Nexperia (owned by a Chinese company) on national security grounds. Workers at the fab have been making their objections known. Stay tuned… [LinkedIn Post.]
- A piece by Bloomberg's Mark Gurman reporting that "Apple is preparing to begin sourcing chips for its devices from a plant under construction in Arizona, marking a major step toward reducing the company's reliance on Asian production...Cook is likely referring to an Arizona factory that will be run by TSMC, Apple's exclusive chip-manufacturing partner. That plant is slated for a 2024 opening. And TSMC is already eyeing a second US facility, part of a broader push to increase chip production in the country." [LinkedIn Post.] For more on the second prospective TSMC fab see this

WSJ article. For a look at the challenges faced by TSMC's first Arizona fab (including "High costs, lack of trained personnel and unexpected construction snags") see this piece.

- A WSJ piece about the industry's flip from shortages to glut (though of course the real situation is much more complex). The Journal reports "The chip industry has pivoted hard from a clamor for higher output to cost cutting as it adjusts to a slump for semiconductors that has infected almost all parts of its business. Chip companies in recent weeks have instituted hiring freezes and layoffs, slashed capital spending plans, reduced factory output and warned investors of a stark reversal in their customers' buying habits... The strength of the boom is turning into extra pain in the current downturn." The party had to end sometime. [LinkedIn posts here and here with lots of discussion in the comments.]

For more industry news, connect with Jennifer on LinkedIn.

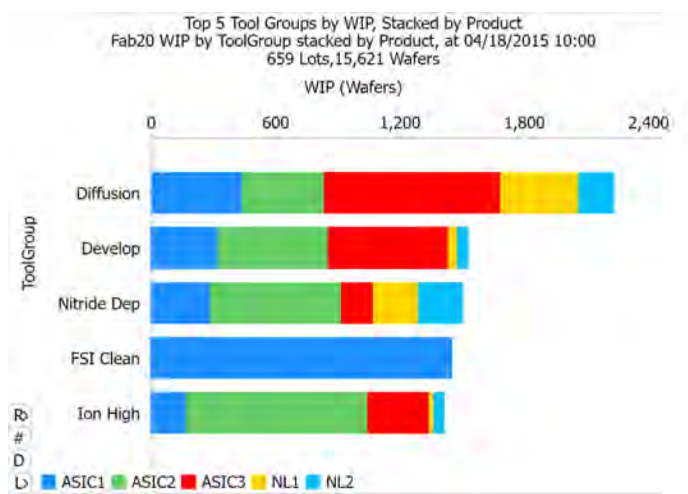
FabTime welcomes the opportunity to publish community announcements, including calls for papers. Send them to newsletter@FabTime.com.

FabTime® Software Tip of the Month

Restrict the Number of Points on a Pareto Chart

Pareto charts are a core type of chart in FabTime (together with trend charts and list charts). A Pareto chart takes that data for the specified time and divides it up according to the selected "Slice" attribute. For example, to look at WIP by Tool Group, we generate a WIP Pareto chart and set the "Slice" drop-down to "ToolGroup." However, there are probably many tool groups in your fab. It's likely, therefore, that you might only want to see the top 5 or 10 tool groups by quantity of WIP. To do this:

1. Generate the WIP Pareto chart and set the "Slice" drop-down (near the bottom of the big set of filters) to "ToolGroup."
2. Make sure that first drop-down under the "Sort Chart" section of the left-hand pane is set to "Units" with the box next to the drop-down checked (to sort in descending order).
3. Scroll down further to the "Formatting" section of the left-hand pane to find the "Points" input box. Enter "5" and press "Go" to restrict the chart to the top five tool groups (according to your sort order, set in step 2 above).
4. A recommended next step is to return to the "Formatting" section and enter a custom title in the "Title" input box, like "Top 5 Tool Groups by Current WIP." This will help reduce future confusion when someone looks at the chart and wonders about the other tools. Note that the "Points" control only applies to the chart image. The full set of tool groups will be included in the data table (unless you separately filter that).
5. We have also noticed that many people like to view Pareto charts with horizontal instead of vertical bars, as shown in this snapshot of a WIP Stacked Pareto. To change your chart's orientation, click the "R" box in the lower left-hand corner of the chart. This rotates the chart image to display horizontal bars on the y-axis. The "Points" control still applies. Click "R" again to return to vertical bars.



This tip applies to any Pareto chart, including stacked charts. If you are using a stacked chart, as in the example here, the "Points" control applies to the number of bars (the "Slice" drop-down) and not the number of items stacked on each bar (the choice of stacking variable is controlled by the "Cross" drop-down).

We hope you find this tip useful.

FabTime software customers can subscribe to the separate Tip of the Month email list (with additional discussion for customers only) here: http://www.fabtime.com/tip-of-the-month.php. Thanks!

Subscriber Discussion Forum

Results from Implementing Green-to-Green Charts

We introduced Green-to-Green (G2G) charts in Issue 20.02. A G2G instance measures the total time that a tool is down, grouping together any logged sub-periods (waiting for parts, waiting for tech, etc.). This total unavailable time (time between two good, or "green" states) is what drives cycle time. **An anonymous participant** in a recent session of our cycle time management class asked if we knew of any published results from anyone who has implemented G2G charts. We were not able to find any published articles on this topic and thought that we would ask our subscriber community. If you have any published results on G2G, or any anecdotal results that you would be willing to share in the next newsletter, we would appreciate your feedback.

Ranking of Cycle Time Improvement Tactics

An anonymous participant from another FabTime cycle time improvement class wrote in recently with this question:

"In our fab, over the past year or so we have reduced the global level of WIP by about 15% in order to improve our cycle time (applying Little's Law). This works well!

Now we are putting in place other actions to maintain the level of activity while we have this lower level of WIP (as Little's Law is not enough to improve cycle time), mainly based on your course recommendations. We are working:

- To increase the number of qualified tools for each operation.
- To reduce tool saturation (improving OEE or transferring some processes to less saturated tools).
- To reduce downtime variability of tools.
- To reduce the hold rate.

We would like to estimate potential cycle time gain of each action, in order to rank these actions (pareto). Stakeholders would like more clarity on the expected impact of their efforts. Do you have formulas or empirical rules to calculate potential cycle time improvement? I remember 50% reduction for single-path eradication and 20-30% for dual-path. But I do not remember anything concrete about reducing hold rate for example."

FabTime Response: Congratulations on your WIP and cycle time reduction efforts so far! Little's Law does apply in practice, doesn't it?

Regarding your other actions, we would rank tool qualification at the top of the list, at least if you have single or dual path operations. As you recalled, you can get a 50% reduction in cycle time per visit through an operation if you go from one to two, and another 20-30% as you go from two to three. We can't say what your overall reduction in fab cycle time will be because it depends on how many operations are affected. As to the other three, we think you have them listed above in the right order in terms of where to focus first.

We felt that your broader question of how to quantify and rank the different potential cycle time reduction approaches for a specific fab warranted a more in-depth response. We have written about this in our main article below.

Controlling Variation in Lot Arrivals Across Factories

An **anonymous subscriber** wrote: "I am wondering if you have any journal articles or other reading material about how to control variation between locations in our supply chain (e.g. arrivals to post-fab from the wafer fab, arrivals to assembly and test from post-fab, or arrivals to internal storage facilities). I see high variation and high utilization causing significant inventory issues. However, I am not sure how to improve variation across factories.

I have plotted data showing (Actual CT – Commit CT) delta and can see cases where lots are being shipped earlier than their commit dates. This causes lots to pile up in storage banks. I understand that some tolerance is needed in cycle time plans, but the question is how big the tolerance should be. Is there any standard or best practice?

Also, when I deep dive to part-level performance I notice a wide range of results, because most of the sites are committing to cycle time targets that are based on 85th percentile of historical lot performance."

FabTime Response: We wrote back in Issue 7.04 about metrics for variability in lot-to-lot cycle times. We also wrote in Issue 18.03 about doing analysis of why lots miss their commit dates. That article was focused more on lots that are late than lots that are early, but we believe that some of the same techniques would apply.

We don't have an answer to your question of how big the deviation should be in lot-to-lot cycle times. Obviously, we'd like it to be as small as possible. There is some research on which dispatch rules tend to be better for tightening the distribution of cycle times. See the "Further Reading" section in Issue 7.04. We think you have the right first step of measuring and reporting that difference. It might be interesting to look at what is different about the fabs that have a lower spread vs. the ones with the widest spread. Is there a difference in how these fabs do their dispatch? Is there a difference in mix or fab size? We regret that we don't have a more definitive answer. We are including this question in the newsletter to see if anyone has recommendations to share.

Separating Maintenance Events (Cycle Time Tip #2)

Another anonymous subscriber replied to validate last month's cycle time improvement tip about separating maintenance events. He said: "Having managed all the equipment at a 300mm fab in the past, I think that grouping maintenance events is a bad idea. The single biggest issue to focus on is coming back up quickly from a PM event. If a tool is down for scheduled maintenance, get it done and back up ASAP. It's risky to pile on extra work to a tool down for PM, since other PMs that you add might end up in an unplanned maintenance state. My experience supports your graph."

Richard Goldhammmer from Micron wrote: "That is an excellent tip. What is convenient for engineering is not always best for cycle time and can create additional WIP. The short, frequent PM is more favorable than one long occurrence. I would further clarify your cycle time contributor to "un-forecasted" equipment downtime which should include appending work to a PM. A buffer is kept ahead of bottleneck steps and un-forecasted downtime drives the buffer higher. Forecasting maintenance to a bottleneck tool based on when the buffer is low or empty, makes sense to me."

FabTime Response: We are glad to hear that our recommendations are in line with your experience. You make an excellent additional point about the fact that when you append extra work to a PM, that is "un-forecasted" time, making the PM longer than manufacturing was planning. We agree completely that planning ahead to do maintenance on bottleneck tools when the WIP buffer is low makes a lot of sense.

Taking that plan and then disrupting it by adding unplanned work seems like a recipe for both frustration and extra cycle time.

Richard's response inspired a related thought from software development that would support more focused maintenance events apart from the impact on cycle time: troubleshooting when something goes wrong. After a large batch of code changes in a software product, if a customer reports a problem, it's natural to ask – did one of the changes cause the reported problem? If there's a single change, or a small number of changes, it's often easier to confirm or refute the hypothesis that the change was the root cause for the bug report. Do any subscribers know of published studies on this possible benefit of shorter, more focused PMs for manufacturing equipment?

There is also the question of scheduled maintenance vs condition-based maintenance. There are studies from the aviation industry that argue in favor of condition-based maintenance rather than time-based maintenance. See, for example, this article, which says:

> "These findings contradict the traditional belief that reliability is predominantly age-related, and that the more often an item is overhauled or replaced, the less likely it is to fail. RCM studies show clearly that unless there is a dominant age-related failure mode (e.g., metal fatigue), age limits and scheduled overhauls do little or nothing to improve reliability."

FabTime welcomes the opportunity to publish subscriber discussion questions and responses. Simply send your contributions to Jennifer.Robinson@FabTime.com.

Main Article: A Fab Cycle Time Improvement Framework

Introduction

In the 23 years that we've been publishing this newsletter (!), we've made many different recommendations for improving wafer fab cycle time. Many of these recommendations are also covered in our cycle time management course. The recommendations vary both in ease of implementation and in amount of potential improvement for any given fab.

A subscriber who participated in a recent session of our course asked a follow-up question (see the Subscriber Discussion Forum above) that we thought would be of interest to the broader newsletter community. He asked, in essence:

- How do you estimate the potential cycle time improvement from different recommendations/strategies to determine what will have the greatest benefit for a particular fab?

We've talked previously about how you decide which tools to focus on (see Issues 21.01, 21.04, and 21.05, for example), but how do you decide which initiatives matter the most for your fab?

The truth is that fabs are extremely complex systems containing many sources of cycle time. For the most part, it's not possible to predict the expected cycle time improvement from any single approach in isolation. However, in thinking about this question, we realized that we do have a sense, after working with fabs for many years, of a logical order in which to introduce improvement initiatives. In this article, we discuss the contributors to fab cycle time and propose a framework for implementing cycle time improvement recommendations for your fab.

Contributors to Fab Cycle Time

As outlined in Issue 22.04: Fundamental Drivers of Wafer Fab Cycle Time, and in Session 1 of our cycle time management class, there are three fundamental conditions that increase cycle time in a fab (or any manufacturing system). They are:

1. High utilization of equipment (having a low buffer of standby time).
2. Variability in arrivals to tools and/or in lot-to-lot process times at tools.
3. An insufficient number of tools for a given operation, due to either small or one-of-a-kind tool groups or to process restrictions/constraints.

There are many factors in wafer fabs that contribute to cycle time. These include:

- Downtime (scheduled and unscheduled)
- Product mix
- Holds
- Setups
- Time constraints between process steps
- Lot release
- Operator unavailability
- Scheduling/dispatching
- Engineering use of tools
- Running development lots in a production fab
- Hot lots
- Reentrant flow
- Scrap
- Rework
- Manual lot transport between steps (via carts or runners)
- Automated material handling systems
- Batch processing
- Soft dedication/operator preferences for tools

All these complexities affect cycle time via their impact on one or more of the fundamentals. For example, downtime reduces buffer capacity on tools, driving up utilization. Downtimes contribute to variability and temporarily reduce the number of available tools per tool group. Setups take up capacity on tools, also reducing buffer capacity. They also effectively increase the process time of the lot at the front of the queue when a setup occurs, increasing effective process time variability.

It is difficult to quantify the impact of any one of these factors on overall cycle time, because fabs are so complex and so different from one another. For a small, high-mix fab, hold reduction could be important. For an operator-constrained fab, operator unavailability might be more significant. Further, many of these factors interact with one another, making decisions about improvement programs challenging. For example, process restrictions may well improve yields, cutting back on the process time variability that comes from different lot sizes. But process restrictions that lead to single or dual path operations increase cycle time.

That said, there are things we know from working with fabs over the past 30 years:

- When asked about the top contributor to their fab's cycle time, more people mention downtime than anything else.
- Single path operations are not a problem for everyone, but where they occur, they increase cycle time significantly.
- Product mix and holds have increased in complexity and prevalence over the years.
- Delays due to lack of operators can cause forced idle time, reducing buffer capacity and driving up cycle time. Lack of operators has been a challenge for many fabs over the past two years. However, that bottleneck seems to be easing.
- Many fabs have worked successfully over the years to reduce the quantity of hot lots. Hold reduction efforts don't appear to have progressed as far.

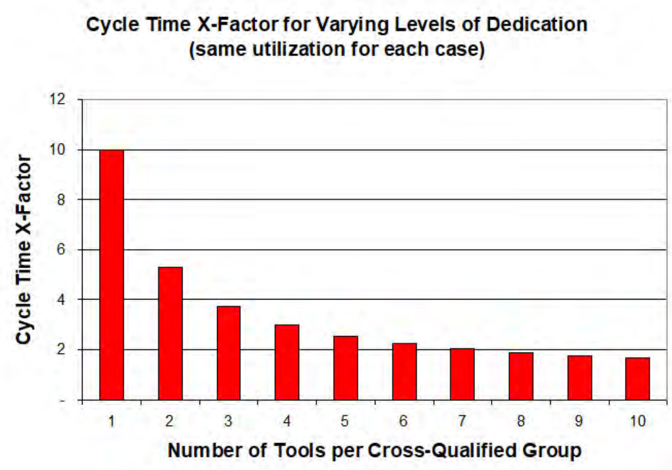
These observations inform our framework for tackling wafer fab cycle time improvement project, proposed below.

A Proposed Cycle Time Improvement Framework

In the conclusion to our cycle time class, we group our top recommendations according to the three fundamentals. That grouping is helpful in summarizing the material from the course. However, as our subscriber question above demonstrates, this still doesn't tell a specific fab where to start and where to focus. Here, we attempt to rectify that omission. We welcome your feedback.

Step 1: Focus on Number of Qualified Tools

If your fab has any operations that have only one or two qualified tools (that are not one-of-a-kind or two-of-a-kind tools), the highest "bang for the buck" thing that you can do to improve cycle time is work to change that. For every operation where you can get from one qualified tool to two, you'll reduce that operation's cycle time by about 50%. For every operation where you can get from two to three, you'll likely reduce that operation's cycle time by another 20% to 30%. The benefit diminishes after that, as shown below. See Issue 20.05: The Impact of Tool Qualification on Cycle Time for more detail.



Once you are in a situation where you either have no more single or dual path operations, or only have single or dual path operations on tools that are true one-of-a-kind or two-of-a-kind tools (see issue 23.05: Managing One-of-a-Kind Tools for suggestions here), move on to Step 2.

Step 2: Find the Current Cycle Time Bottlenecks

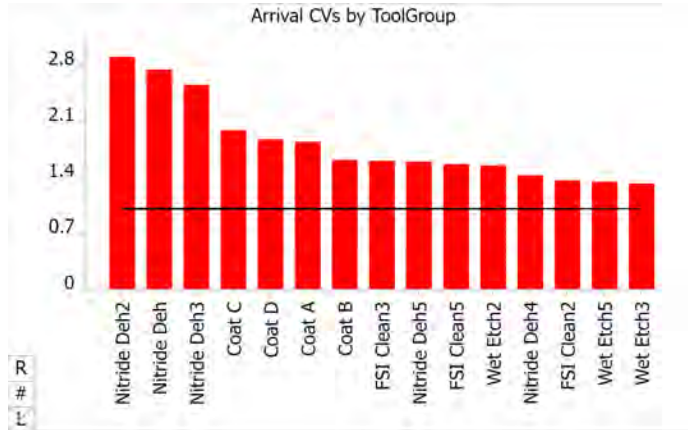
Measure the operation level cycle times by tool group (over at least two weeks) and look for tool groups that have a high per-visit X-Factor (or just a large amount of queue time). See Issue 21.01: Finding and Analyzing Cycle Time Bottlenecks. Also consider using the WIP Hours metric that we defined in Issue 20.03. Identify a candidate list of up to 10 tool groups that right now are driving up your fab's cycle time.

Step 3: Analyze the Cycle Time Bottlenecks to Understand Root Causes

For each of the cycle time bottlenecks, the next step is to determine the likely root causes. This will help to inform which of Steps 4 through 7 are likely to be of most benefit for your fab. Here are a few avenues to explore.

- Is there a very low amount of standby time? Make sure you're looking at true standby time, not standby time when there is WIP waiting, because that's more of an operator delay. If you have a tool group that has a high cycle time per visit, and a low amount of true standby time (5% or less), focus on increasing buffer capacity for this tool group, as outlined in Step 4.
- Measure Green-to-Green time if possible (see Issue 20.02: A Metric for Green-to-Green (G2G) Analysis) and look for the tool groups that have long periods of unavailable time (whether from scheduled or unscheduled downtime events). Look especially at tool groups with long downtime that are smaller tool groups. This is where downtime variability is likely to be causing the most problems. These are tool groups where you should focus on reducing downtime variability, as discussed in Step 5.

- Is there a high amount of arrival variability? Ideally you would look at the coefficient of variation of the time between arrivals to each tool group of interest (see example to the right). However, a proxy for this is to look at a graph of arrivals per shift and see where that is highly variable. See if you can dig into the data to understand why the arrival variability is high at key tool groups. Are they fed by a large batch tool? Are they downstream from a tool with a lot of downtime variability? Are carts being used for manual transfer of lots to this tool? Once you have an idea about the sources of the arrival variability, move on to Step 6.



Step 4: Increase Buffer Capacity at Key Tools

Because of variability in the fab, it's important for all tool groups to have some standby time. This standby time provides a buffer against variability and keeps cycle time from getting too high. If you have tool groups with little or no standby time, they are likely cycle time bottlenecks. Recommendations for improvement here include:

- **Maximize availability**. These are the tools for which it is most important to focus downtime reduction efforts. Any unavailable time that can be converted to standby time results in a lower effective utilization (for the same amount of process time) and reduces cycle time.
- **Minimize forced idle time**. Forced idle time is time when the tool is available and has WIP that could be run, but for some reason is not running. Often, though not always, forced idle time results from operator unavailability (see Issue 22.05: Managing Operators During a Staffing Shortage). Standby-WIP-waiting time (a category that we report on our Tool State Trend and Pareto charts) is a useful indicator of forced idle time. Post-process time (time between when a tool logs an End Run event and an operator logs a Move Out from the tool) also indicates a hidden forced idle time. Here, the lost capacity will be hidden in inflated process times. In addition to operator unavailability, forced idle time can result from hand carry lots (the tool is held idle to wait for an arriving hand carry) and from time constraints between process steps (hold the lot at the upstream tool until we are sure it can make it through the downstream step). Dispatch decisions that starve downstream bottleneck tools also effectively force idle time. Suggestions for reducing forced idle time include:
    - Minimize the number of distinct tools for which each operator is responsible, and stagger break schedules.
    - Reduce the number of hot lots in the fab, especially hand-carry lots that cause setup changes or force idle capacity.
    - Make dispatching decisions to keep critical downstream tools from starving.
    - Make sure that lots aren't sitting on hold unnecessarily when they could be in process. For example, make sure that a backup is designated for future holds, in case the original engineer is unavailable when the lot reaches the future hold step. (See Issue 23.04: Cycle Time and Holds Revisited.) This can create a type of forced idle time, if a tool could have been busy processing the lot.
- **Transfer processes to other tool groups**. Some fabs that have significant tool redundancy have different tool groups that contain similar tools. A larger tool group may be broken into two smaller groups. This can make sense provided the tool groups aren't too small, but it can also lead to

unbalanced utilizations between the sub-groups, particularly if product mix changes. In such cases, it may be necessary to transfer operations between sub-groups. It is always a good idea in these situations (especially for high mix fabs) to keep a close eye on the utilization of each sub-group, and make changes as necessary.

- **Find and reduce soft dedication**. Soft dedication is common in fabs. It occurs when the capacity model and tool qualification data in the MES say that a tool group is fully cross-qualified, but the operators in fact have preferences for certain tools over others. This can result in unbalanced utilizations across a tool group, where one tool is used more than expected, and others less. To identify soft dedication, plot the moves and availability efficiency of each tool in the group, and look for mismatches. Where this is taking place, the operators doubtless have reasons (one tool is in a less convenient location, a newer tool is faster than the other tools in the group, etc.). However, if soft dedication is causing cycle time bottlenecks, intervention on the part of management may be required.

Step 5: Improve Variability from Downtime

As mentioned above, downtime is recognized as one of the largest sources of cycle time in most fabs. Improving overall availability increases buffer capacity, as discussed in Step 4, and is a long-time focus of fab improvement efforts. What is sometimes less well-understood is the impact of downtime variability on cycle time. To improve cycle time, it's best to minimize that variability. To do this:

- Separate maintenance events instead of grouping them (see Issue 22.01: On Breaking Up PMs and Other Unavailable Periods, as well as Cycle Time Tip 002).
- Focus downtime improvement programs on reducing the duration of repair times, particularly for the cycle time bottlenecks. This might mean purchasing additional spare parts or re-allocating equipment technicians where long total unavailable times are driving high cycle time. Reducing the duration of downtimes will have the highest impact at one-of-a-kind tools.
- Report not just availability but availability variability (see Issue 18.04: Measuring Variability of Availability). Look at hours of unscheduled downtime broken up by sub-state, to understand how much time is spent waiting for parts, waiting for technicians, etc., to help identify the best paths to improvement.

Step 6: Improve Arrival Variability

Variability changes the shape of the operating curve of cycle time x-factor vs. utilization. Reducing arrival or process time variability reduces cycle time. In the queueing formulas that we teach in our cycle time class, arrival variability and process time variability, if equal, have the same impact on cycle time. Usually, however, there is more arrival variability than process time variability in a fab. This is because process time variability often gets sent downstream as arrival variability to the next step. Meanwhile, other aspects of semiconductor manufacturing contribute additional arrival variability. Efforts to reduce arrival variability should focus on the following areas.

- **Batch processing:** Batch tools, such as furnaces, are tools that can process multiple lots at one time. The arrival pattern downstream from a batch tool consists of long periods of no arrivals, and then a burst of several arrivals at once. This variable arrival pattern contributes to high cycle times downstream from batch tools. Batch process time is usually independent of the number of lots in the batch. This leads to a decision when a partial batch is available for processing, but not a full batch. Some fabs have policies that require waiting for some minimum batch size before processing. This is fine for tools that are heavily loaded, because most batches on these tools end up full or nearly full anyway. However, requiring full batches on less heavily loaded tools increases queue time for the lots that must wait and sends extra arrival variability downstream. We recommend working to eliminate minimum batch size constraints on batch tools that are not heavily loaded. (See Issue 9.03: Batch Loading Policies for Wafer Fabs.)

- **Lot transfer between steps:** A similar issue to batch processing arises when carts are used to transfer lots between steps. Operators, in an understandable effort to increase their own efficiency, often wait until a cart is full before moving it. This means that lots wait on the cart, and then wait again when sent downstream, as they wait for other lots on the cart. Due to operator constraints, it may be necessary to use carts in this way. However, we note that one possibility for reducing arrival variability in the fab is to get smaller carts for lot transfer or set policies that discourage waiting until a cart is full before moving it. (See Issue 9.02: Manual Lot Transfer in Wafer Fabs.)
- **Lot release into the fab:** Lots are typically released into the fab in batches (groups of lots). The less frequent those lot releases are, the more variable the arrival pattern to the early operations in the process flow. If you find that arrival variability is high at early steps in the flow, consider more frequent, smaller lot releases.

Step 7: Improve Process Time Variability

While not usually as pronounced as arrival variability, process time variability can also be considerable in fabs, especially high mix fabs. The simplest type of process time variability involves running recipes with different process times on the same tool. Things like downtime and setups also contribute to effective process time variability. From the perspective of a lot at the front of the queue and ready to go, a downtime or a setup looks like an increase in process time. Operator unavailability to unload tools also contributes to process time variability, as do changes in lot size due to scrap and rework. Steps to help reduce process time variability include:

- Work on process simplification. The more similar your recipes (and process times) are, the better for cycle time. There are other benefits from this approach, too, in terms of learning curves and a more robust response to product mix changes.
- Reduce setups and rework. One thing to be careful of when reducing setups, however, is to check setup avoidance policies to make sure that low volume lots aren't waiting too long, especially on non-bottlenecks. We recommend having some threshold in place by which, if a lot waits more than some amount of time, a setup is performed, even if there are lots with the current recipe in queue. This is especially important for non-bottlenecks, where the loss of capacity from doing the setup isn't a major problem anyway. We also recommend having policies in place that discourage waiting for a lot with a matching recipe when there are lots already there that could be processed (after a setup). This is analogous to the greedy vs. full batch decision and introduces some forced idle time.
- Consider breaking up large tool groups into smaller groups and assign recipes with similar process times (and/or that reduce setups). This must be done with caution, however. The tool groups shouldn't be smaller than 3-4 tools, and it's important to balance utilization across such sub-groups, as discussed above.
- Automate reporting of End Run transactions so that it's possible to separate post-process time from true process time. Where this is significant, look at assigning/cross training more operators.

Step 8: Consider Other Fab-Wide Initiatives

One shortcoming of the approach described here is that by focusing on specific cycle time contributors, we may be under-emphasizing fab-wide initiatives, including:

- Reduce holds, especially future holds. This will reduce variability and help reduce the cycle time of individual lots.
- Enhance dispatch rules to take into account non-local information (to spread out WIP, avoid feeding WIP to down tools, etc.).

Conclusions

Fabs are uniquely complex environments, with no shortage of factors that increase cycle time. We have listed some of these factors above, though we have doubtless missed a few. This complexity can make it challenging to even know where to start with improvement programs. For this article, we have taken a step back to think about how the various recommendations that we have made over the years could fit together to provide a logical sequence for cycle time improvement efforts.

We stand by our top recommendation for improving fab cycle time, increasing the number of qualified tools for each operation, as a first step. After that, we think it makes sense to identify the tools that are contributing the most to fab cycle time, and then analyze them to get a sense of likely root causes. After that, we would focus first on increasing buffer capacity for these cycle time bottlenecks, then on reducing variability from downtime, then on arrival variability, and then on process time variability. We have included other fab-wide efforts such as hold reduction and enhancing dispatch rules as a final step in the framework, through these projects could be undertaken at any point.

Of course, this sequence, and the recommendations therein, will not be a perfect fit for every fab. But we believe that some of you may benefit from thinking about cycle time improvement in this sequential way. We've also captured several of our previous recommendations and included references to more detailed articles where available. As always, we welcome your feedback.

Closing Questions for Newsletter Subscribers

How would you modify this cycle time improvement framework? What are we missing or over-simplifying? Are there other factors that drive cycle time in wafer fabs that are not included in the list above? Should we add this framework to our cycle time management course?

Further Reading

We have referenced many past issues of this newsletter in the text above. Subscribers can download any or all past newsletter issues from our website. The password is FabTimeCommunity (case-sensitive, all one word). For information about our remote cycle time management course, where we cover many of the topics touched on here in more detail, please fill out the form on our website.

Subscriber List

Total number of subscribers: 2861

Top 20 subscribing companies:

- Onsemi (174)
- Analog Devices (includes Maxim) (144)
- Intel Corporation (137)
- Infineon Technologies (136)
- Micron Technology, Inc. (116)
- Microchip Technology (100)
- GlobalFoundries (91)
- NXP Semiconductors (81)
- STMicroelectronics (69)
- Skyworks Solutions, Inc. (64)
- Texas Instruments (60)
- Seagate Technology (56)
- Western Digital Corporation Inc. (56)
- Carsem M Sdn Bhd (54)
- X-FAB Inc. (54)
- Wolfspeed, Inc. (44)
- Qualcomm (38)
- Tower Semiconductor (30)
- Applied Materials Corporation (29)
- ASML (28)
- Hitachi Energy Ltd. (28)
- Polar Semiconductor (28)

Top 3 subscribing universities:

- Ecole des Mines de Saint-Etienne (EMSE) (7)
- Arizona State University (5)
- Ben Gurion University of the Negev (5)

New companies and universities this month:

- ASYS Automatic Systems
- MaxLinear
- Miko
- NAURA Akrion Inc.
- Oregon State University
- Polariton
- TÜBİTAK

Note: Inclusion in the subscriber profile for this newsletter indicates an interest, on the part of individual subscribers, in cycle time management. It does not imply any endorsement of FabTime or its products by any individual or his or her company.

There is no charge to subscribe to the newsletter. Past issues of the newsletter are now available in PDF for download by newsletter subscribers from FabTime's website. To request the current password, email your request to Jennifer.Robinson@FabTime.com. To subscribe to the newsletter, send email to newsletter@FabTime.com, or visit our website. To unsubscribe, send email to newsletter@FabTime.com with "Unsubscribe" in the subject. FabTime will not, under any circumstances, give your email address or other contact information to anyone outside of FabTime without your permission.

FabTime® Software: If you would like more information about our web-based dashboard for improving fab cycle times, please visit our website. A sample home page and a sample page from FabTime's new Charts menu are shown below.