FabTime Newsletter

Volume 24, No. 1 **February 2023**

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, February 14, 2023 – Vol. 24, No. 1

Contributors: Thomas Beeg (Fabmatics) and Hani Ofeck

Keywords: Performance Improvement; Fab Management; Hot Lots; Bottlenecks; Dynamic X-Factor

Table of Contents

Welcome

Happy Valentine's Day! Welcome to Volume 24, Number 1 of the FabTime Cycle Time Management Newsletter. In this issue, we have an announcement about a case study that we'll be presenting at the Fab Owners Alliance next week, an update about our efforts to change the platform that we use to send the newsletter, and an announcement about new training and informational materials for software customers. Our software tip of the month is about using the newly modified Dynamic X-Factor chart in FabTime. We have subscriber discussion about the best way to share alerts, how to identify bottlenecks in the fab, and whether to set a lower bound on the number of hot lots.

Our main article this month was inspired by the positive response that we've received to our new cycle time improvement tip emails. We shared a list of ten recommendations for improving fab cycle time two years ago, and today we introduce ten additional recommendations. While not all these recommendations will be relevant for all readers, we hope that each of you finds something useful in the discussion. If you have additional tips for improving fab cycle time, we would love to compile and share those in the future.

Thanks for reading! – Jennifer, Frank, Lara, and the FabTime Team

Community News/Announcements

Fab Owners Alliance Collaborative Forum Case Study

FabTime will be participating in a joint case study with Flexciton and Renesas Electronics Corporation at the February Fab Owners Alliance Collaborative Forum in Phoenix next week. The subject of the case study is **Time and Labour-Saving Implementation of Advanced Production Scheduling at Renesas Wafer Fab with Seamless Data Integration**. Please look for Jennifer Robinson and Lara Nichols if you are there.

New Platform for Sending the Newsletter Issues

As previously announced, we sent the January cycle time tip of the month email via a new third-party system. Our intent was to offer a more streamlined newsletter sign-up and account modification system. Unfortunately, the system we chose proved disappointing. We are currently researching other systems and are sending this issue via the old method.

One thing that would help to ensure that you receive future newsletters and tips is to right-click on the email to which this newsletter was attached and select:

Never Block Sender's Domain (@example.com)

This should help ensure that future tips and newsletters (depending on your subscription choices) will be delivered to your inbox. If you did not receive Cycle Time Tip #3 (sent January 11th), please email Jennifer.Robinson@FabTime.com to request a copy. We apologize for any inconvenience and appreciate your patience.

Updated FabTime Training Videos and Feature Showcase

FabTime software customers, we are pleased to announce that our Applications Programmer **Bailey Vandehei** has re-recorded most of our software training videos using the current FabTime user interface. You can view the updated videos on our demo server. If you work at a customer site and don't have an account on the demo server, just email Elaine.Jacobson@FabTime.com to request one.

We also have our first video subtitled in three different languages (Chinese, Japanese, and Czech), and are working to provide more subtitled material for our international sites as needed.

Finally, our Director of Customer Success **Elaine Jacobson** has prepared a detailed Feature Showcase for the newly released Patch 115. The feature showcase is a great way to learn about new FabTime features and share them with people at your site. Software customers can email Elaine.Jacobson@FabTime.com to request a copy.

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 couple of links to articles about new international wafer fabs under construction. One is by Teltonika in Lithuania (using Taiwanese technology). The other is a new foundry called Rapidus in Japan (partnering with IBM Research). The international landscape seems to be changing from week to week. It is definitely an exciting time in the semiconductor industry. [LinkedIn Post.]
- Also, via Semiconductor Engineering's Week in Review, an article about Wolfspeed's planned 200mm semiconductor fab in Germany, which will be the world's largest. "The company's first fab in Europe will be its most advanced, creating a breakthrough innovation Silicon Carbide development and production facility in the European Union to support growing demand for a wide variety of automotive, industrial and energy applications." [LinkedIn Post.]

- An announcement about a $1B domestic fab expansion by Analog Devices in Beaverton, OR. This is good to see! [LinkedIn Post.]
- In less optimistic news, a WSJ article about the pricing slump facing memory chip makers. "Memory-chip prices, which dropped steeply over the past year, are expected to keep falling in the first half of 2023, putting more pressure on an industry that has already cut investments and jobs." [LinkedIn Post.]
- A blog post about rework and cycle time published in late December by our friend **Thomas Beeg**. Thomas discusses various rework scenarios and includes numerical examples to conclude that "As long as rework rates are reasonably low and the time lots spend in the rework loop is short there is a small impact" on cycle time. The whole post is well worth a look! [LinkedIn Post.] See also another post from Thomas about Product Mix and Fab Performance, sure to be of interest to newsletter subscribers.

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

Use (the Newly Modified) Dynamic X-Factor Chart to Monitor Speed of the Line

A FabTime metric that we recommend in our cycle time class is Dynamic X-Factor (DXF). We have recently modified our DXF calculation to make it more useful in real-world wafer fabs and wanted to introduce this metric to our software customers.

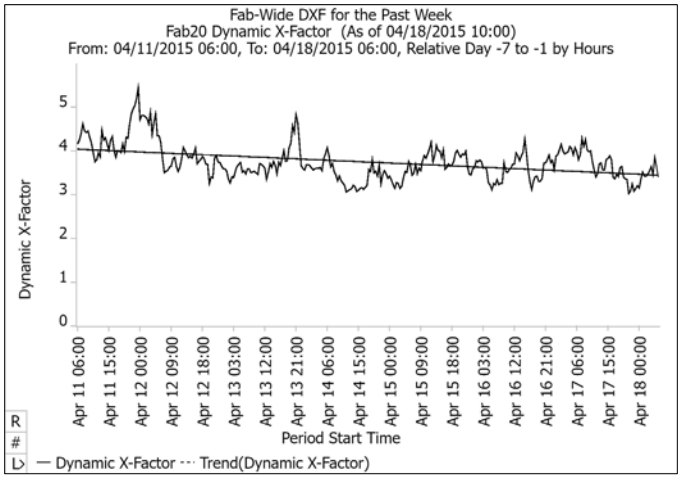
DXF was introduced by researchers from Yasu Semiconductor (S. Johnishi, K. Ozawa and N. Satoh) at the ISSM conference in 1992. DXF is a point estimate of production line speed, used to proactively identify deviation from cycle time goals. DXF is measured frequently (e.g., every hour) as the total WIP in the line divided by the WIP currently running on tools. DXF can be shown (see Issue 9.04 of FabTime's newsletter), when averaged over time, to equal the shipped lot cycle time x-factor (total cycle time / theoretical cycle time). However, to use DXF you don't need to know the theoretical cycle time. You just need to be able to determine whether lots are in process on tools. DXF is a forward-looking indicator. It estimates, based on current conditions, what future cycle time x-factor will be. More details can be found in Issues 4.08, 5.03, 9.04 and 15.05.

As initially implemented in FabTime, DXF was displayed as total WIP / in process WIP. The problem with this implementation was that "in process WIP" could include both rework lots and lots on hold. This distorted the DXF numbers and made them less useful. It wasn't possible to filter those out of the denominator without also filtering them out of the numerator. With Patch 115 of FabTime's software, we have remedied this defect. DXF now excludes rework lots and lots on hold from the denominator. If in process WIP is equal to zero at a tool, dynamic x-factor is set to total WIP.

To use the newly modified DXF chart in Patch 115:

- Enter "Dynamic" into the Search box and select Charts | Dynamic X-Factor Trend.
- Enter your period of interest (the default is the past week).
- Change the "Len" drop-down from "Days" to "Hours" and then change the "24" hours to "1" hour (or even .5 hours). This will give your DXF chart sufficient granularity for you to see periodic effects such as shift change. It will also avoid the distortions that might occur if you only measure DXF once a day and that measurement happens to occur at shift change. (Less WIP may be running on tools during shift change than is normally the case.) Hit "Enter" or press "Go" to update the chart.

- The resulting DXF chart will give you a visual representation of the cycle time x-factor and the variability in your overall fab from hour to hour. Adding a trend line can be helpful to see whether your cycle time is increasing or decreasing.
    - Click "Edit Chart" below the chart (and data table, if displayed). In the row that starts with DXF, click "Trend". You can also include a moving average by clicking that box. The chart will automatically update. An example is shown to the right.



- You can also filter the DXF chart by major product line, to estimate the future cycle time performance of your key products, or by priority class.
    - You can filter or Pareto DXF by area or tool group, but you can expect considerably more variability at that level. Also please remain aware, if using a Pareto version of the DXF chart, that you are only seeing a single point in time. We again advise against that point in time being chosen at shift change (unless your goal is to look at shift change behavior).

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

Formatting for Alerts

In our software, users can set up alerts whereby the user (and/or other designated recipients) will be notified if some condition in the fab is met. (A lot waited too long, a tool is down, etc.) Our alerts have historically been sent as short text email messages (which can be received as text messages). We are in the process of redesigning our alerts. While advising us on that change, **Hani Ofeck** shared some thoughts about alerts in general and suggested that we seek feedback from the newsletter community.

Hani has two points related to this, one to do with the format of the alerts when sent, and one to do with whether they should be sent separately at all, in some cases. We do feel that this discussion is relevant to anyone who uses alerts, whether they use FabTime's software or not. (See these two articles shared by Hani on the broader subject of alerts: 8 Rules of Great Email Notifications and Email Overload: Brain and Behavioral Responses...)

First, a quick poll. Which of the following display formats for an alert about a lot that waited too long is more useful (these are mock-ups prepared by Hani, shared with her permission)?

Text-based:

Or table-based:

[redacted] <FabTime@TowerSemi.com>
Sent: Tuesday, February 7, 2023 8:49 AM
To: Ofeck Hani <hanitas@towersemi.com>
Subject: Lot xxxxx is over HH hours - please check why

Time Stamp	Lot ID	Operation (Step)	Operation Desc	TIO (H) (Time in Operation)	Alert trigger	Alert Comments
2023-02-07 8:49	xxxxxxxx	xxx	nvbvavasv	xxxxx	xxxx	Lot xxxxx is over HH hours - please check why

From:
Hani Ofeck

Hani's contention is that the second format is more visual and readable.

Second, do we need to send out separate alerts for each triggered behavior, or would it be more useful in many cases to simply have a table that you check periodically throughout the day for any triggered alerts? We can see this being useful. We wouldn't want to receive emails related to every follow-up item in our to-do list, even those with due dates associated with them. We think there is probably room for both types of alerts: push-type alerts that go out via email and checklist type alerts that simply appear in a table somewhere. What do all of you think about this question? We will share any feedback in the next issue.

Identifying Bottlenecks

An **anonymous subscriber** wrote: "I would like to know if the topic of bottlenecks in production has been addressed in past Newsletters or Tips. Normally for bottleneck research and analysis we rely on the experience of someone who knows the process or product, but I would like to bring in a more data-driven approach. So, I'm asking where I can find advice on how to recognize and highlight bottlenecks in the production line."

A **subscriber from a different company** wrote a few days later to ask: "I wanted to check if you can share anything specific about how to identify bottlenecks, current bottlenecks and downstream or future bottlenecks and their management. I would really appreciate your help."

FabTime Response: In these newsletters, we have discussed three different types of bottlenecks.

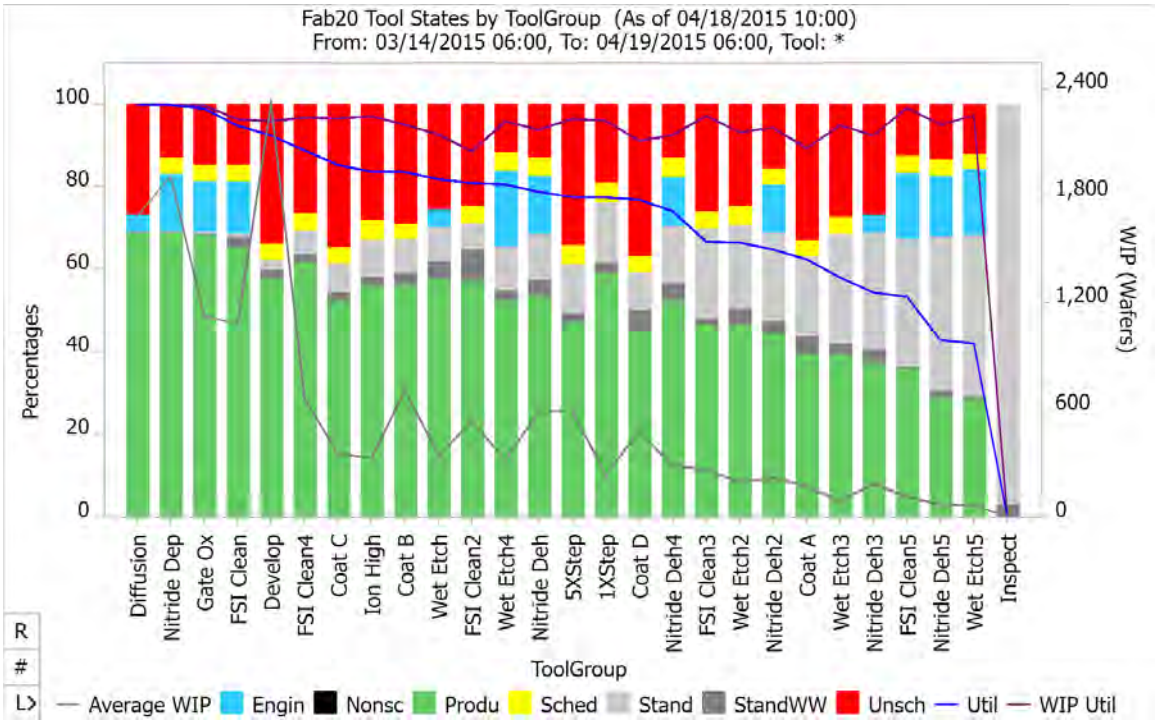
- **Traditional capacity bottlenecks** are the tools in the fab that, over a long-term basis, are planned with the smallest buffer of unused capacity. Their utilization of manufacturing time ((productive time / (productive time + standby time)) approaches 100%. Capacity bottlenecks are typically the most expensive tools in the fab.
- **Cycle time bottlenecks** are tools that may not be bottlenecks in the sense of being overloaded, but for some reason have higher cycle time per visit than other tools (perhaps due to poor reliability or high arrival variability). See Issue 21.01 and the subscriber discussion forum in Issue 21.02.
- **Short-term bottlenecks** are tools that are not overloaded over the long term but become bottlenecks on a short-term basis. This is often because of availability issues but can also be due to product mix changes or engineering requirements. In Issues 21.04 and 21.05 we defined a short-term bottleneck as a tool or tool group for which, over the next 12-24 hours, required capacity is likely to exceed available capacity. We recommended the WIP Hours metric to identify short-term bottlenecks.

The second subscriber question above introduces a fourth type of bottleneck:

- **Future bottlenecks** are tools that are likely to become bottlenecks in the future due to high expected arriving WIP. Product mix changes or starts ramps could drive long-term future bottlenecks, while short-term future bottlenecks are likely to result more from variability.

To find traditional capacity bottlenecks (tools that have the highest utilization of available time, FabTime customers can use the Tool State Pareto Chart, sliced by Tool (or ToolGroup), over a relatively long time period (maybe a month), sorted in descending order by Utilization. The capacity bottlenecks have the

highest utilization (the blue line in the example below). Here utilization is defined as utilization of time that the tool is available to manufacturing = productive time / (productive time + standby time).



These above-referenced issues on short-term and cycle time bottlenecks can be downloaded here: FabTime Latest Newsletter Publication. The current password is: FabTimeCommunity

To identify future bottlenecks, we must know what WIP is expected to arrive at downstream tools and how the required process time for that WIP compares with the expected available time. Short-term simulation can be used for this purpose but requires quite a bit of upkeep of detailed data to be useful.

In our software, we do offer predictions of which lots are expected to arrive to downstream tools or operations. We make these predictions based on where the lots are now and what their planned process times are for future steps. These predictions consider whether a lot is in queue or in process at the current step, but do not consider WIP levels or tool availability. Thus, the best they can do for future bottleneck identification is show tools that are likely, based on current WIP levels, to have high WIP levels soon.

Setting a Lower Bound for the Number of Hot Lots

An **anonymous subscriber** wrote: "Today, I would like to ask your opinion about the portion of WIP with priority. Have you any guidelines or benchmarks to ensure good cycle time? We know it is better to respect a maximum of 10% of WIP in priority, but we are currently running well below that level. Some on our team would like to put a fictional priority on lots that are not late to increase the quantity of priority WIP closer to 10%. Are there any guidelines for a minimum portion of priority WIP?"

FabTime Response: Regarding priority lots, our recommendation is:

- No more than 1-2 ultra hand-carry lots at one time in the fab (to achieve the best possible cycle time for those lots). With 5, you may find that they interfere with one another, or that holding tools for them leads to capacity losses (and higher cycle time, as you know) on your bottleneck tools.
- No more than 5% priority lots if the goal is to minimize the impact on the regular lot cycle time. We previously recommended 10% here, but fabs we work with have moved this threshold down over

the years closer to 5%, except for special cases (e.g., some of the WIP is make-to-stock and some is make-to-order, so you prioritize the make the order above the make to stock).

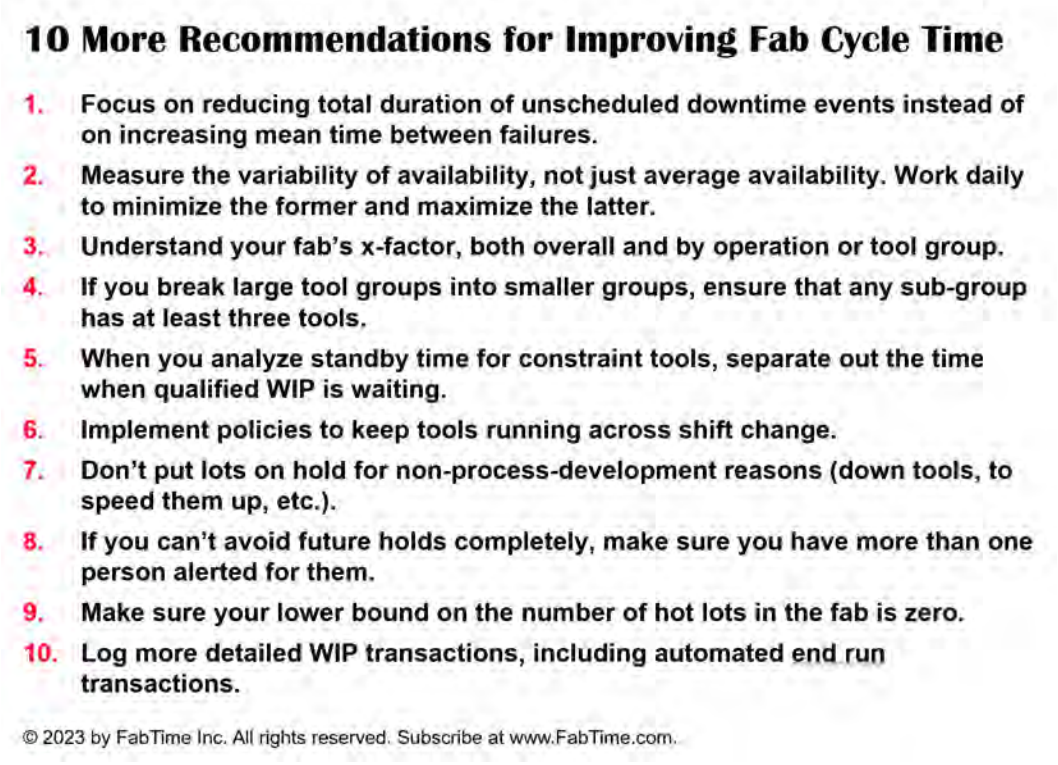
We don't believe there is any lower bound on the quantity of priority WIP. The less the better. If you are not having late lots that need to be prioritized, that's great! It means you are doing something right (or … your utilizations are down because of softening demand and fewer starts, which is less positive, but this is still no reason to artificially add hot lots). In theory, priority lots that are not hand-carry don't affect average cycle time. They just shift queue time from the higher priority lots to the lower priority lots. In practice, however, priority lots add variability, particularly at tools with large batch sizes or setups. The ideal target that we would recommend for hot lots is zero (though we recognize that some small number of hot lots is usually necessary for business reasons). See also our most recent newsletter about hot lots, Issue 19.03.

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

Main Article: 10 More Recommendations for Improving Fab Cycle Time

Introduction

Two years ago, we published a list of our top ten recommendations for improving wafer fab cycle time (see Issue 22.02). We continue to stand by these suggestions. More recently, we started sharing brief cycle time improvement tips to this distribution list every other month, alternating those with the full newsletters. Feedback for the new cycle time tips has been positive and appears to us to reflect an appetite for short, actionable improvement tips. Therefore, we offer in this article ten more recommendations for improving fab cycle time. These are distilled from our experience and our cycle time management course. [More details about the course can be found on our website.]



Each new recommendation is briefly described below, with references to more detailed newsletter articles where applicable. You may see some of these published separately as cycle time tips in the future.

Our Old Recommendations

For those who missed it, our previous list of ten recommendations for cycle time improvement is shown below.

10 Recommendations for Improving Fab Cycle Time

1. Identify and eliminate single path operations (if possible).
2. Check for soft-dedication due to operator preferences.
3. Reduce transfer batch sizes between steps.
4. Run batch tools under a greedy policy.
5. Separate maintenance events instead of grouping them.
6. Minimize the number of distinct tools for which each operator is responsible, and stagger break schedules.
7. Reduce the number of hot lots in the fab, especially hand-carry lots.
8. Smooth the flow of arrivals into the fab.
9. Check setup avoidance policies to make sure that low volume lots aren't waiting too long, especially on non-bottlenecks.
10. Make dispatching decisions to keep critical downstream tools from starving.

Now, on to the new recommendations!

Focus on reducing total duration of unscheduled downtime events instead of on increasing mean time between failures.

What affects cycle time is not how frequently tools go down (to a point, at least), but rather how long they are unavailable when they do go down. This is particularly true for one-of-a-kind tools. Anyone who has worked in a fab, particularly a smaller or older fab, knows how painful it is to have a key tool unavailable for days at a time. WIP bubbles pile up, causing operational problems that last much longer than the downtime event itself.

Rather than focusing on the mean time between failures, we recommend driving for high overall availability (see below) while reducing the duration of downtime events. A good metric to use for this is Green-to-Green (G2G), which measures the total time from when the tool becomes unavailable until it is available once again, grouping together any sub-states like waiting for technician or waiting for parts. See Issue 20.02 for details about G2G.

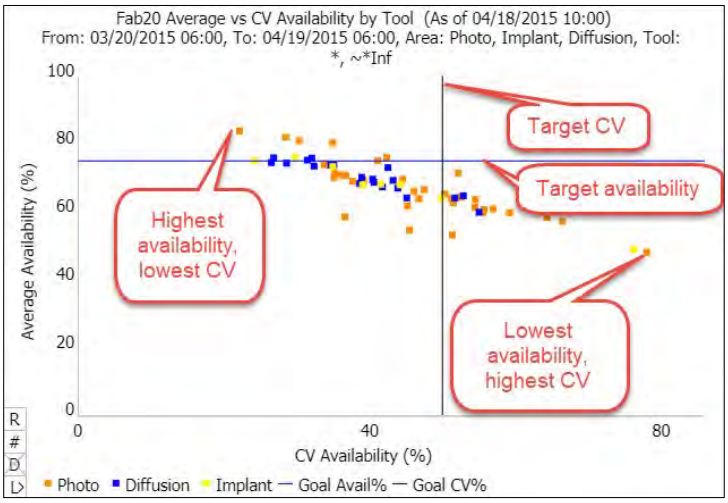
Measure the variability of availability, not just average availability. Work daily to minimize the former and maximize the latter.

Increasing tool availability is a well-known key to improving fab cycle times. Whenever unavailable time is converted to standby time on a given tool, that tool has more of a buffer to recover from variability. Improving availability on key tools helps fabs to move away from the steep part of the operating curve, and achieve lower cycle times. Improving availability can also help fabs to increase throughput and become more profitable, though that is not our primary topic today.

For fabs that are focused on cycle time improvement, it makes sense to move beyond focusing on average availability to also consider the variability of the availability itself. We don't just want tools with high average availability over the course of the year. We want tools that can be relied upon day-to-day and shift-to-shift to be available when they are needed.

Availability variability can be measured using:

- Coefficient of variation of the availability;
- Coefficient of variation of repair times or of G2G times; or
- A proxy metric like A20/A80.



The important thing is to track something, ideally in a visual way that captures not just average availability but variation in availability from tool-to-tool and from day-to-day. An example of a quadrant chart showing availability variability is shown to the right. [This chart was implemented in FabTime's software at the request of a customer who worked with a consultant specializing in fab performance improvement.] The more you can reduce availability variability, especially for bottleneck tools, the better your overall cycle times will be.

Note also that even on lower-utilization tools, availability variability sends arrival variability downstream. When the tool comes back up from a long downtime, a big burst of WIP is sent to previously starved downstream tools. See Issues 4.02 and 18.04 for details.

Understand your fab's x-factor, both overall and by operation or tool group.

If you want to improve a fab's cycle time, it helps to understand how much improvement opportunity is available. FabTime has long recommended the use of X-Factor (total cycle time / theoretical cycle time) to track cycle time performance. We particularly encourage fabs to use Dynamic X-Factor (DXF, see the software tip of the month above), which can give a forward look at expected X-Factor without requiring access to theoretical cycle time data.

X-factor is also useful at the operation level, aggregated by tool group, to identify the tools that, on a short-term basis, are contributing the most to cycle time. These are the tools to start with for cycle time improvement initiatives, as outlined in Issues 21.01 and 23.06.

If you break large tool groups into smaller groups, ensure that any sub-group has at least three tools.

Some fabs that have large tool groups break them into smaller sub-groups, with the tools in each group running similar recipes. This can be helpful in reducing setups as well as reducing process time variability. However, it is important if you do this to ensure that any sub-group has at least three, or preferably four, tools in the group. This is because tool groups with only one or two tools have significantly higher average cycle time per visit. See Issues 20.05 and 22.04.

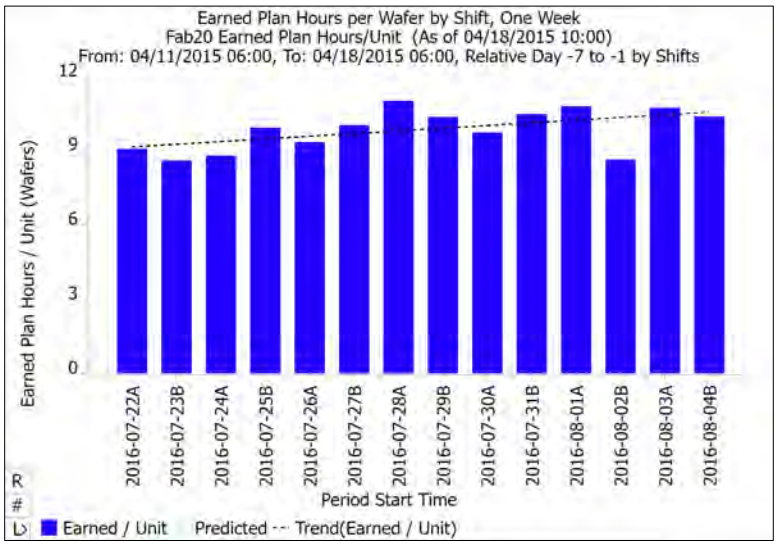
It's also important to ensure that the average utilization across the sub-groups is consistent. We don't want to create two sub-groups, one with average utilization of 75% and the other with average utilization of 95%. The latter will have much higher cycle times than necessary, outweighing any benefits from reduced process time variability.

When you analyze standby time for constraint tools, separate out the time when qualified WIP is waiting.

This was the topic of Cycle Time Tip #3. The idea is that not all standby time is created equal. Sometimes a tool will display as being in a standby state while there is qualified WIP waiting in front of the tool. In our

software, we call this Standby-WIP-Waiting. We report it separately from Standby-Other (time when the tool is available but is not running because there is no WIP to run). Standby-WIP-Waiting time is **not** buffer capacity that can be used to recover from variability. Instead, Standby-WIP-Waiting time represents a capacity loss, forced idle time on a tool. Often, it occurs because there is no operator there to load the tool, though there can be other causes such as WIP staging. Identifying Standby-WIP-Waiting time when it occurs on constraint tools is a first step in reducing it and improving cycle time.

Implement policies to keep tools running across shift change.



Shift changes can be a hidden source of forced idle time on tools. When tracking moves, there is an incentive to finish short production runs toward the end of a shift. This may mean that tools are left idle at shift change, because the incentive is to track work out of tools, not to keep tools working. This leads to low moves at the start of the next shift. We worked several years ago with Cypress Semiconductor (now SkyWater Technologies) to implement a metric called Earned Plan Hours (EPH).

The EPH metric attempts to counteract undesirable behavior by giving credit when lots are tracked into tools, and as processing occurs on tools. When a lot is tracked into a tool, credit is given for the planned queue time of the lot at its current flow/step. As a lot is processed, credit is given for elapsed processing time, up to the end of the planned process time or shift change, whichever comes first. EPH per wafer should always equal the shift length if things are going according to plan. If less, as shown in the example above, the factory is falling behind. See Issue 14.01. Fabs can also use Dynamic X-Factor to identify shifts or areas that do a better job than others keeping WIP running over shift change.

Don't put lots on hold for non-process-development reasons (down tools, to speed them up, etc.).

Engineering holds may be unavoidable for fabs developing new processes. We have learned from discussion with many fabs, however, that it is not uncommon to put lots on hold for reasons other than process development. Sometimes a lot is put on hold because the only tool that can be used to process it is down waiting for parts, or because a reticle is unavailable.

There are two problems with this practice. First, it hides the true reason for the lot's delay. We won't know from the data that better spare parts or reticle management policies might have avoided the delay. The second problem is that we may not have policies that automatically take the lot back off hold when the tool or reticle or whatever is available. Lots can thus spend more time on hold than they need to. It's our recommendation that lots not be put on hold when the hold masks the true reason for the delay. We have even heard of cases where fabs put lots on hold to draw attention to them, with the goal of speeding them up. This seems counter-productive to us and is not recommended. See Issue 23.04.

If you can't avoid future holds completely, make sure you have more than one person alerted for them.

Another phenomenon that we've observed is the "future hold." This is when an engineer specifies that a lot be placed on hold when it reaches some future step. These future holds may again be unavoidable for process reasons. The problem with them is that when the future hold arises, the engineer who placed the

hold might not be available. Engineers are people, not tools. They don't work 7x24. They go on vacation. They get sick. Therefore, our recommendation is that fabs establish a policy that any future hold has a designated backup engineer who can disposition the lot and get it moving again.

Make sure your lower bound on the number of hot lots in the fab is zero.

This tip was inspired by a question from the subscriber discussion forum above. A subscriber reported that people on his or her team "would like to put a fictional priority on lots that are not late in order to increase the quantity of priority WIP closer to (the fab target of) 10%." We'll just say here that while hot lots may be necessary, they are not free. They increase variability in the fab, and thus drive up cycle time. While target upper bounds for hot lots can vary (usually between 5% and 10%, except in special cases), the target lower bound on the number of hot lots in a fab should be zero. See the subscriber discussion forum above for more information.

Log more detailed WIP transactions, including automated end run transactions.

Another hidden source of forced idle time on tools occurs in less automated fabs, when the tools do not send an automatic signal when a lot finishes processing. The lot appears as "in process" in such fabs' systems until the operator logs a move out transaction. For fabs that are operator-constrained, this time can be significant. This causes the reported process time to be inflated and reduces the standby time that the tool can use to recover from variability. Dynamic X-Factor charts will also be less accurate where this occurs, meaning that projected cycle times from DXF will be biased low.

In general, more detailed logging of WIP-related transactions (automatically recording end run transactions, differentiating between move out transactions at one step and arrival transactions at the next, and differentiating between arrival and begin run transactions) enables better understanding of lot cycle times and highlights opportunities for improvement. This item is last on the list not because it isn't important but because it is relatively difficult to implement.

Conclusions

Even after 30 years working in the industry (for Jennifer and Frank, at least), we are still learning about painful complexities and quirky behaviors in fabs. The upside of that is that there are many things that we can recommend that people in fabs do, or stop doing, to improve cycle time. We summarized ten of these ideas two years ago. The positive response to our new bi-monthly cycle time tips emails has inspired us to share ten more here.

Every fab is on a different place in their journey to understand and improve cycle time. Each of these tips will be more relevant for some fabs, and less relevant for others. But it is our hope, if you have taken the time to read this far, that you have found some tidbit that will be useful for you. As always, we welcome your feedback (and your tips for improving fab cycle time!).

Closing Questions for Newsletter Subscribers

What other tips do you have for improving cycle time in a wafer fab? Do you agree with the ideas presented here? Disagree? Or find them irrelevant because everything is perfect in your fab? Please let us know what you think.

Further Reading

- J. Robinson and F. Chance, "10 Recommendations for Fab Cycle Time Improvement," FabTime Newsletter, Vol. 22, No. 2, 2021. This issue, together with all other past issues referenced above, is available for download from the FabTime Newsletter Archive. The current password is FabTimeCommunity.

Subscriber List

Total number of subscribers: 2885

Top 20 subscribing companies:

- Onsemi (178)
- Analog Devices (143)
- Infineon Technologies (141)
- Intel Corporation (140)
- Micron Technology, Inc. (118)
- Microchip Technology (101)
- GlobalFoundries (90)
- NXP Semiconductors (80)
- STMicroelectronics (69)
- Skyworks Solutions, Inc. (62)
- Texas Instruments (60)
- Seagate Technology (57)
- Western Digital Corporation Inc. (56)
- X-FAB Inc. (55)
- Carsem M Sdn Bhd (53)
- Wolfspeed, Inc. (42)
- Qualcomm (37)
- Tower Semiconductor (31)
- Applied Materials Corporation (30)
- ASML (30)

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:

- Arcadia Cold Storage Logistics
- Boston Scientific
- Brooks Instrument
- Deloitte

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.