FabTime Cycle Time Management Newsletter

Volume 21, No. 1 January 2020

Information

Mission: To discuss issues relating to proactive wafer fab cycle time management

Publisher: FabTime Inc. FabTime sells cycle time management software for wafer fab managers. New features in development right now include updates to make home page tabs more responsive to different screen sizes, and integration of a more flexible data table interface.

Editor: Jennifer Robinson

Contributors: Azizi (SilTerra Malaysia); Thomas Beeg (CREE/WOLFSPEED); Gerald Livingstone (TowerJazz Semiconductor); and Florin Oprea (Qorvo)

Keywords: Bottlenecks, Metrics and Goals, Reporting, Dispatching

Table of Contents

Welcome

Welcome to Volume 21, Number 1 of the FabTime Cycle Time Management Newsletter. We hope that you are all off to a great 2020 and believe that it is going to be a great year for the semiconductor industry.

Would you believe this is the 159th issue of the newsletter? In this issue we have an exciting announcement about a plan to make past newsletter issues available to subscribers. Our software tip of the month is about a starter set of charts for manufacturing supervisors. We have subscriber discussion about dispatch compliance, on-time delivery calculations, rework %, and queue time sub-states.

Inspired in part by the discussion on queue time sub-states, we focus our main article on cycle time bottlenecks: the tool groups that contribute the most queue time to cycle time in a fab. We describe methods for both identifying cycle time bottlenecks and analyzing them. We close with a brief summary of concrete recommendations for mitigating the primary contributing factors to cycle time bottlenecks. We welcome your feedback.

Thanks for reading – Jennifer

FabTime

Tel: (408) 549-9932
Fax: (408) 549-9941
www.FabTime.com
Sales@FabTime.com

Community News/Announcements

Plan to Make All Past Newsletter Issues Available to Subscribers

As part of a quest to make the FabTime newsletters more useful to our subscriber community, FabTime is working to provide password-protected access to all past issues (in PDF format) from our website for subscribers. Please stay tuned! We hope to have this in place by the next issue.

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

FabTime User Tip of the Month

Create A Starter Set of Charts for a Manufacturing Supervisor's Home Page

We had a question recently from a new software customer asking for a list of the first charts that manufacturing supervisors should add to their home pages. These are the charts that a manufacturing supervisor would look at every day at the start of each shift. We think it's a great idea to begin with a limited set of key metrics that are broadly familiar and useful. Here's the list that we shared with the customer:

- Starts
- Shipments
- Moves
- WIP (total and by segment of the line)
- WIP Turns (moves / average WIP)
- Scrap
- Idle WIP (WIP that has been in queue for more than some threshold, e.g. 12 hours)

These are core metrics that demonstrate the overall health of the fab. In all cases, these charts should be displayed relative to a goal, either an automated goal, or a manual chart stripe that indicates a goal region. Looking at the current data as well as the trend for the week gives a manufacturing supervisor a quick idea of the fab's progress.

Of course, there are lots of other charts that can be useful on a day-to-day basis: down tools, hot lots, due date performance, and holds come to mind. Different fabs will emphasize different things, depending on their size, customer base, product mix, etc. But, after working with fabs for 25+ years, we think that the list above is a good place to start!

Subscribe to the separate Tip of the Month email list (with additional discussion for customers only). Thanks!

Subscriber Discussion Forum

Dispatch Compliance (Issue 20.06)

Long-time subscriber **Azizi from SilTerra Malaysia** wrote in response to December's Dispatch Compliance article, saying: "I am sending you an update on an Alternative Approach to measure the application of a dispatching rule. Back in 2003, we observed wide range of cycle time variability, and one of the root causes identified was lack of consistent dispatching techniques used by operators. This happened during a time of transition when higher product mix was introduced to the fab. Individual business units would prioritize their own products to the level of dictating lot dispatching on the shop floor. Due to this problem, an RTD system was introduced as the new standard operating procedure. Operator performance was measured based on dispatch compliance. Doing this, we managed to reduce the lot variability 15% in the first half of the month. It later improved much more. We initially used a binary calculation but then improved into using accuracy levels subject to X-factor requirements and risk to due date commitments."

FabTime Response: It's great to hear about this example of immediate and sustained benefits from tracking dispatch compliance. Starting with a simpler binary calculation and then adding complexity along the way makes sense to us, too. We appreciate Azizi's feedback!

On-Time Delivery Calculation

Gerald Livingstone from TowerJazz Semiconductor wrote: "In the semiconductor industry, with all of your interactions, do you know how on-time delivery is measured? It is complicated by yield loss if a line yield is 96%, how can you ever get a score of 100%? How have you seen it done?"

FabTime Response: "What we do in our software is define on-time percent as the percentage of lots shipped on or before their due date. Any time we're looking at a list of lots that shipped, we compare the shipment date for each lot to the due date at the time of shipment and set a zero-one flag. Either it shipped on or before the due date or it didn't. Yield doesn't play into that calculation because the scrapped lots never ship. How about other subscribers? Do any of you calculate this differently? Do you take yield loss into account?

Rework %

Florin Oprea from Qorvo asked: Has FabTime written any articles or best-known methods about Rework % and how fabs are measuring this metric? We measure rework in different ways and trend it but there are differences between the technologies that our fabs are running. Our interest is to learn if indeed, somebody has created a BKM and they did a deep dive on this subject."

FabTime Response: We have not written about rework to date. A quick literature review didn't turn up anything that directly addressed the question either. We are thus raising this question to the subscriber community, to see if anyone has any thoughts to contribute on this subject.

Queue Time States

This is a follow up to a discussion that we started in the last newsletter. Over the past 20 years, FabTime has increased the granularity by which we break out cycle times in our wafer fab reporting dashboard. Initially, we only had access to operation move-outs, which allowed us to track total cycle time by operation. Gradually we added more detail. Customers who have the transactions available can now break cycle time into travel, hold, queue, pre-process, process, and post-process time.

What we're wondering now is whether we should try to break the queue time down into more detailed sub-states (as shown below) that reflect the reason that a lot is in queue. For example:

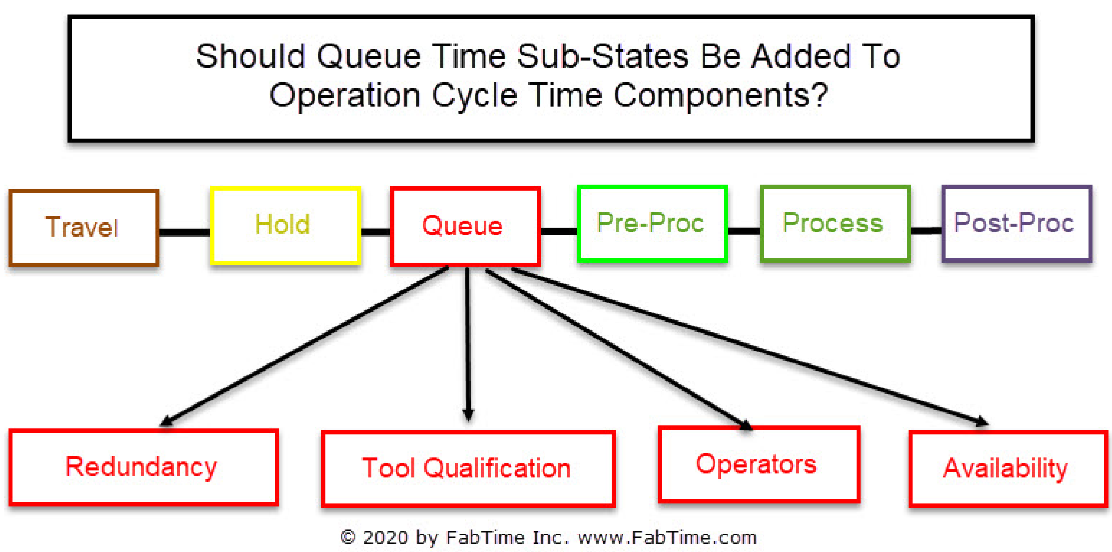
- Is there no tool available?
- Is there another tool available that is not qualified to run this operation?
- Is there no operator to load the tool?
- Is there no tool available because of equipment downtime?
- Etc.

It seems to us that having a better understanding of the reasons that lots are in queue, at least for bottleneck tools, would open opportunities for improvement. This is analogous to using OEE to analyze tool uptime. The better data we have about why tools are unavailable, the better we can focus equipment improvement efforts. Similarly, the more data we have about why lots are in queue, the more we should be able to focus cycle time improvement efforts. Do we need more of an emphasis on tool qualification, better ways to notify operators when a tool is ready to load, or something else?

FabTime raised this question of breaking queue time down into sub-states to Jennifer's connections on LinkedIn. You can read the discussion in the comments here. Summarizing the responses:

- Yes, people think this data is worth getting, even if it's going to take some extra effort to do so, particularly for less automated fabs.
- We should ensure alignment with the SEMI E10 and related standards, going beyond the existing standards where necessary.
- The quality of any results will depend on the quality of the input data, particularly where that input data is logged by people.
- We need to be careful not to do too much "spit balling" in the case of some of the complex and overlapping sub-states (e.g. it's lack of redundancy and tool downtime), because we could end up with a false sense of certainty, and less reliable data.
- We should make sure that effort put into understanding queue time contributors stays focused on tools that have the most overall impact, to avoid wasting scarce resources.

Thomas Beeg from CREE/ WOLFSPEED sent in this detailed response: "Thank you for bringing up the topic of lot wait time break down. In my personal opinion, looking more into the details of why lots wait adds significant value. The traditional focus on maximizing



OEE, if executed well, will lead to generally higher tool utilization. This is a risky approach if cycle time and on time delivery are important. Too often, a local optimization effort to reduce standby time, aka increasing OEE, will lead to additional fab bottlenecks and increase the chances of generating WIP waves. I think the only true, crystal clear indicator of fab performance that is measurable and not manipulatable is cycle time.

I propose to replace the traditional method of defining fab bottlenecks by looking for the highest OEE or utilization value, with one that looks for the tool groups with the highest lot wait time values. To understand where a fab has its true bottleneck tool groups, we should create a pareto chart of how much average wait time lots spend at each tool group.

This chart can be very telling about the general health of the fab. Typically, a handful of tool groups will show significantly higher wait times - these tool groups should be the focus in the first round of optimization efforts. Often, these tool groups have high utilization values. This is not a surprise. However, almost all the time there are also tool groups with not so high utilization values that generate significant wait times. These we would not detect by using an OEE based method.

Once the tool groups with high lot wait times are identified, the obvious next question is: why? Exactly here will the proposed break down of wait time categories help tremendously. This data will point us in the right direction, to where resources should be spent to improve. Often there are reasons outside of pure high tool utilization, things like: lot arrival patterns; hidden restrictions on recipe and/or chamber level; dispatching policies; wrong assumptions in planning about batching or cascading; availability of monitor wafers; and time link constraints, to name a few.

But a wait time pareto will potentially also bring another interesting point to light: very often many of the true fab bottlenecks are known (even by only using OEE or utilization as the measurement). What we learn from the wait time pareto is that even though this handful of high wait time bottlenecks contribute a lot of wait time, this rarely amounts to more than 20-30% of the total wait time occurring throughout the fab. What that means is that 70-80% of the total wait time in the fab is generated on tool groups that are traditionally seen as non-bottlenecks and therefore do not get a lot of attention as candidates for improvement. I found in my years working on fab cycle time improvement that these tool groups often are easier to improve, simply because they have not experienced the attention of engineers in task forces, OEE projects, tiger team activities, or the like.

The question is sometimes asked: why should we work to improve non-bottlenecks, sometimes reducing their utilization even further? The general philosophy is, if we can reduce the overall fab cycle time by let's say 5 days by improving non-bottlenecks, we could start a few more wafers. This, of course, will increase utilization at all involved tool groups and therefore will increase the wait time especially at the higher utilized ones. However, as long as the wait time increase driven by the increased wafer starts is less than or equal to the reduction we got from improving the non-bottlenecks, we end up with the same (or lower) total fab cycle time, but higher wafer output. This is what I call the "Holy Grail" of all fab industrial engineering activity. Thoughts?"

FabTime agrees completely with Thomas' belief that we should look for bottlenecks based on wait time rather than based on utilization. We did ask Thomas a clarification question about what he meant by "a pareto chart of how much average wait time lots spend on each tool set."

FabTime's software includes two versions of this:

1. A metric that looks at lots that moved over some time period and records the queue time (and process time) incurred during the associated visit. (The Operation Cycle Time Pareto chart)

2. A metric that looks back at lots that have shipped and reports the total queue time and process time accrued over the lifetime of the lot across multiple visits. This chart, the Factory Cycle Time Contribution Pareto, shows the total queue time contributed by each tool group across multiple visits.

The Operation Cycle Time Pareto is an indicator of current cycle time problems, while the Factory Cycle Time Contribution Pareto highlights tool groups where cycle time builds up across multiple visits.

Thomas clarified his response by saying: "I actually think we would need both. We need a mid to long term pareto based on historical individual lot times to analyze where we lose systematically time. I think this should be based on a minimum of four weeks of data. Twelve weeks would be even better, but that depends on fab size, fab CT and volume. We also need a trend version of this chart that lets us drill down into one tool group, to see if the issue is chronic or a one-off event. We also to show these wait time sub components in a stacked chart by tool group, in near real time. This would enable people to react to problems as they occur, especially if most of the WIP is blocked, or bottlenecks are having an impact on some products / recipes."

FabTime also told Thomas that we thought he made an excellent point about the importance of improving wait time at non-bottlenecks. We talk about this a bit in our cycle time management class, and covered it in Issue 10.09, but we like how Thomas motivates this specifically around the "holy grail" of higher throughput at the same (or better) cycle time.

This discussion ties back to the 1990s MIMAC project where Jennifer and Frank first collaborated. The MIMAC project introduced the concept of "cycle time constrained capacity". The idea is that yes, you may have an absolute maximum capacity of 6,000 wafer starts per week based on running bottleneck equipment at 100% OEE, but you won't start 6,000 wafers each week because the resulting cycle time would be terrible. Thus, your capacity is effectively constrained by cycle time. If you currently start 5,500 wafers per week and achieve 3.2X cycle time, and a cycle time improvement effort enables you to start 5,600 wafers per week and still achieve 3.2X cycle time, you have increased your (cycle-time-constrained) capacity. [See J. W. Fowler, S. Brown, H. Gold, and A. Schoemig, "Measurable Improvements in Cycle-Time-Constrained Capacity," *Proceedings of the 6th IEEE/UCS/SEMI International Symposium on Semiconductor Manufacturing (ISSM)*, October 6-8, 1997, San Francisco, A21-A24. Available for download from FabTime's Website.]

We noted to Thomas our view that if we can improve non-bottlenecks, we may be able to reduce the variability that those tools contribute to the arrival stream for the capacity bottlenecks. This would potentially help even more. Thomas agreed, but reminded us that "some of the incoming variability does not really matter if a bottleneck has a high amount of WIP waiting." This, of course, is true. However, we do think that moving WIP from queues in front of lower utilization tools to queues in front of bottlenecks will allow better dispatching and batching decisions at the bottlenecks.

We appreciate Thomas' thoughtful contributions to this discussion. In fact, his thoughts inspired us to expand on the concept of cycle time bottlenecks as our main article.

Next steps on queue time sub-states: The FabTime team will discuss queue time sub-states further with our software User Group to decide where to prioritize this project in our development plan. We will subsequently launch an email discussion thread with interested customers to work through the implementation questions. (See last month's subscriber discussion forum for examples of some of the questions). Interested individuals from FabTime customer sites should contact us about participating.

We will share our findings in future issues. Our thanks to everyone who has participated in the discussion so far.

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

Finding and Analyzing Cycle Time Bottlenecks

Introduction

A new customer asked us recently about best practices for using FabTime's software to improve cycle time. We recommended that this customer start by identifying the tool groups currently contributing the most queue time.

This seemed to us to tie directly to Thomas Beeg's remarks in the subscriber discussion forum above. Thomas said: "I propose to replace the traditional method of defining fab bottlenecks by looking for the highest OEE or utilization value, with one that looks for the tool groups with the highest lot wait time values." We like this idea very much and are using this article to expand on the concept of what we will call cycle time bottlenecks.

In this article, we discuss methods for: finding cycle time bottlenecks; analyzing why these tool groups are contributing so much to cycle time; and mitigating the impact of cycle time bottlenecks.

Finding the Cycle Time Bottlenecks in Your Fab

Let's begin by defining cycle time bottlenecks as the tool groups that are contributing the most queue time for lots in a factory. As discussed above, we recommend two primary ways to identify these tool groups.

1. Generate a pareto of operation cycle times by tool group over the short-term (48 to 168 hours, depending on your transaction volumes). Sort the resulting chart in descending order by queue time. The tool groups at the top of the list are those that have contributed the most queue time to lots that moved during this time period.

2. Generate a pareto of shipped lot cycle time contribution by tool group for all the lots that shipped during the past week or month (again, depending on your transaction volumes). This is a chart that sums up the contribution from each tool group to the total shipped lot cycle time, accumulating across all visits to the tool group. Again, sort in descending order by the queue time component of the cycle time. The tool groups at the top of the list are those that have contributed the most queue time across visits for the lots that shipped during the time period.

What we're looking for in these two charts are tool groups that stand out by contributing a significant chunk of queue time. Technically these are queue time bottlenecks, not cycle time bottlenecks.

However, because cycle time is the overall metric that we are trying to improve, we prefer the term cycle time bottlenecks.

It's true that certain long process steps (e.g. furnace operations) may contribute significantly to overall cycle time. [See Grewal et. al.] However, there are usually process reasons for this that are outside the scope of manufacturing improvement efforts.

It's also true that things like holds and travel time contribute to overall cycle time. However, our focus for today is on identifying the tool groups that have the most impact on overall cycle time where we can make a difference in terms of operating practices. For that, we look at the tool groups where queue time is accumulating.

We could also include post-process time in such an analysis (where the lot has finished processing but has not been moved out.) This can be worth tackling, especially for high utilization tools. However, our experience has been that this time is smaller in magnitude than queue time. In many fabs, the post-process time is not broken out separately, but rather captured in process time variability.

A few notes:

- As Thomas noted in his remarks, the cycle time bottlenecks will likely overlap with the capacity bottlenecks in the fab. However, there will almost surely be tool groups that are not heavily loaded overall that are also contributing significant queue time. These can be excellent and previously untapped targets for cycle time improvement efforts.

- Sometimes you know that the queue time at a given tool is really queue time for some downstream tool, as when lots are held at a clean step prior to being sent to a batch tool. This dynamic is hard to capture systematically in your reporting, but it is something to keep in mind as you compile your list of candidates for improvement efforts.

- Another real-time indicator of cycle time problems is a pareto of WIP in queue by the age of each lot (how long the lot has been at the current operation). This is an excellent operational chart to look at to decide what lots should be processed this shift. Over time, however, the same tool groups should rise to the top of this list that already show up on the operation cycle time pareto. Because this chart is a point-in-time estimate, it is less helpful for identifying systematic contributors to cycle time.

Analyzing the Causes for the Top Cycle Time Bottlenecks

Once you have a list of cycle time bottlenecks, the next step is to select the top candidates and dig into the causes of the queue time. The most obvious contributors are:

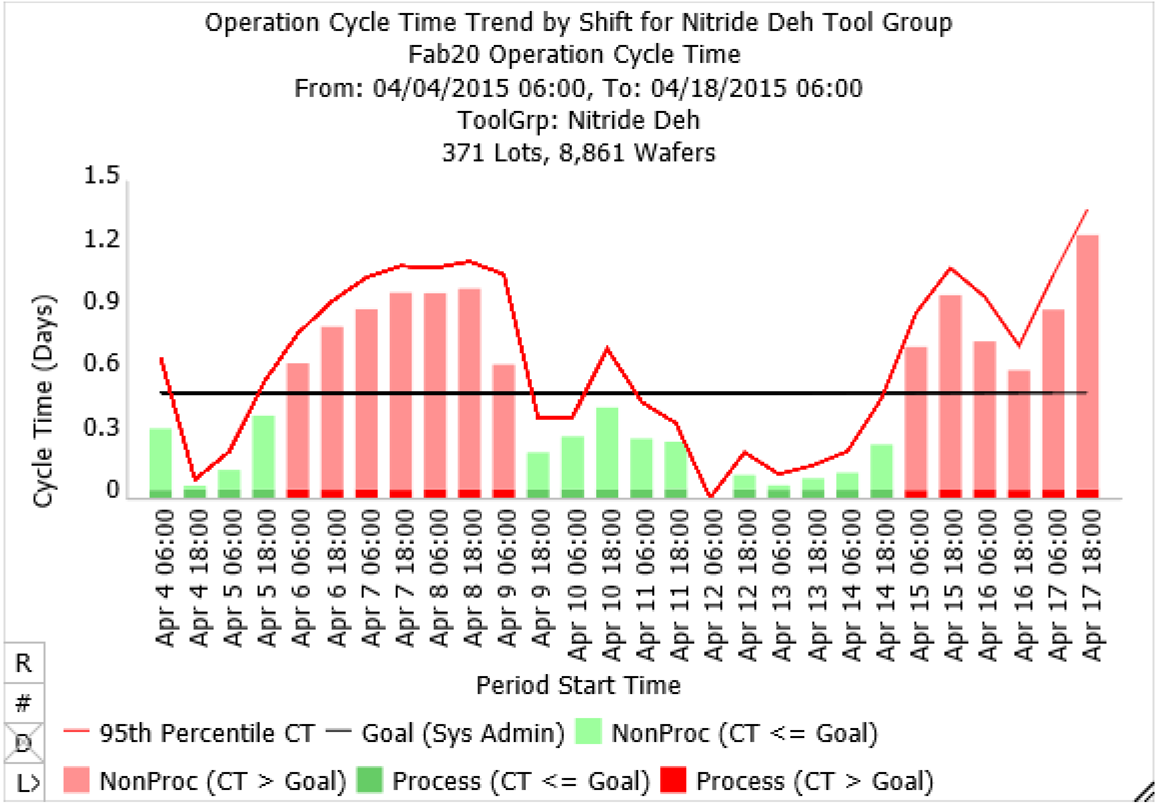
- Utilization
- Downtime
- Arrival Variability
- Lack of Operators
- Process Time Variability
- Lack of Redundancy/Qualification

Our queue time sub-states proposal discussed above is an attempt to simplify this analysis. However, even without formal sub-states, most fabs have plenty of other data that can uncover root causes. We recommend starting with the above list, which is sorted in approximate order of data availability.

Here's a sample process, with examples from our demonstration server:

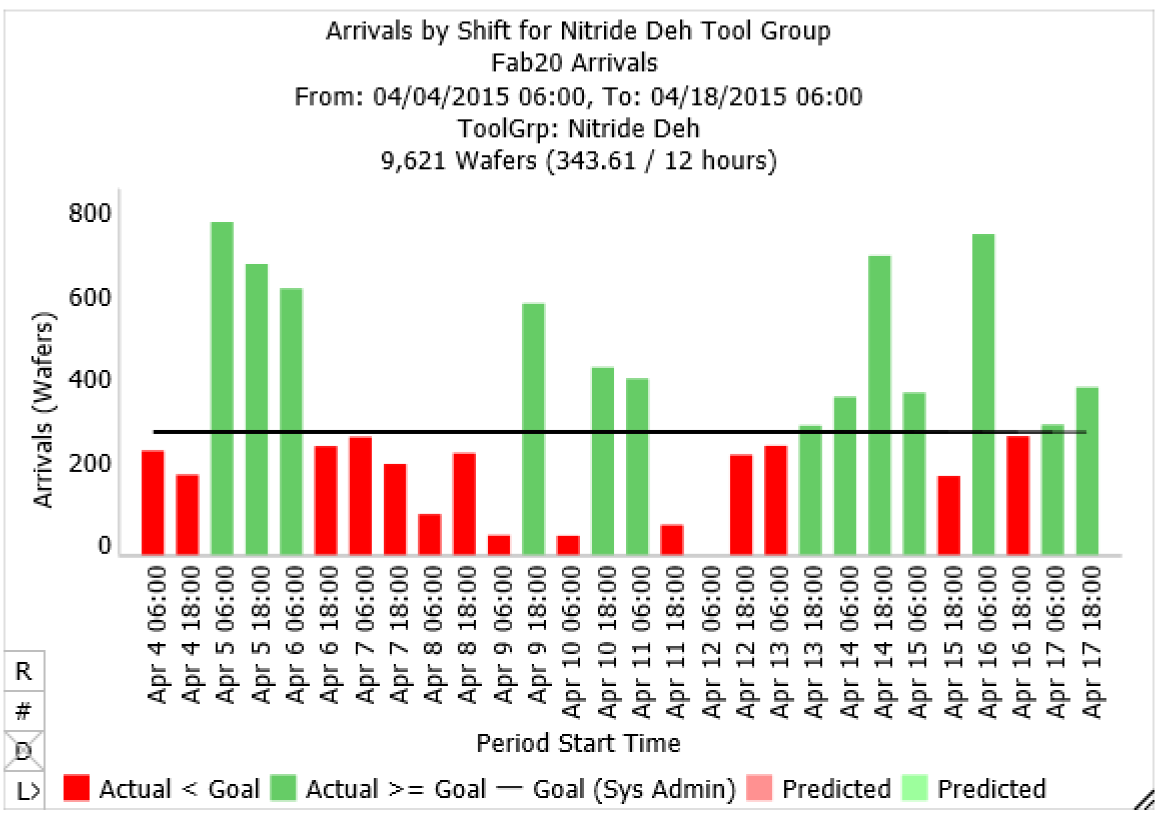
1. Look at the operation cycle time trend through the tool group by shift or by day to look for patterns.

In this example, we've selected a Nitride Deh tool group that appears in the top cycle time bottleneck lists in terms of both operation-level and cumulative cycle time. Looking at a trend in operation-level cycle time through this tool group by shift over



two weeks (shown above), we see that the cycle time tends to rise and fall over periods of a few days. The average utilization of the tool group (which contains two tools) during the time period is 76.2%, with 16.4% standby time.

2. Look below at the arrivals trend to the tool group for the same time periods. Look for variation. Is it systematic or more random?



Here we can see that arrival variability contributes to this tool group being a cycle time bottleneck, though further analysis would be needed to understand the reasons for the lumpy arrivals. The periods with low arrivals correspond approximately to the periods with less queue time, as common sense (and the theory) would predict. [See issues 4.05, 7.08, and 16.05 for more on arrival variability.]

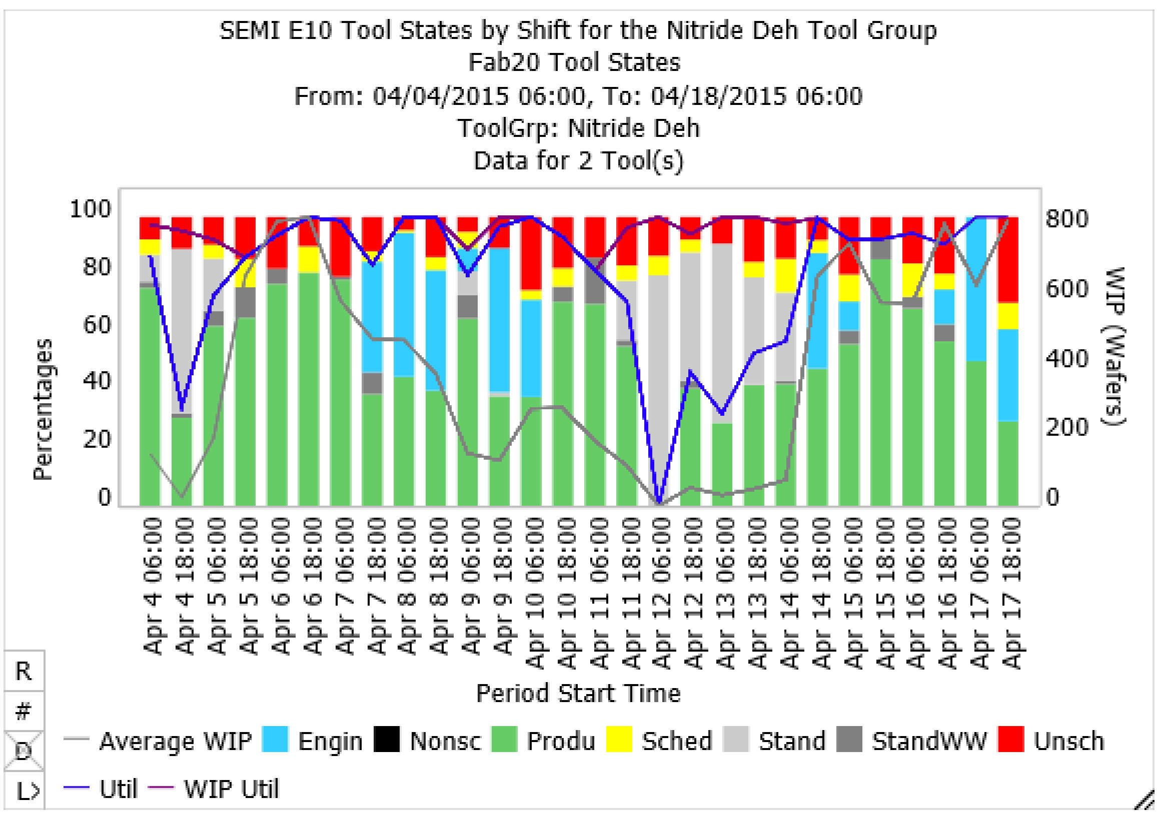
3. Look at the SEMI E10 tool state trend chart for the same time periods. Is the tool group always heavily loaded? Is there never much standby time? In that case, utilization may be the biggest driver of cycle time through the tool group.

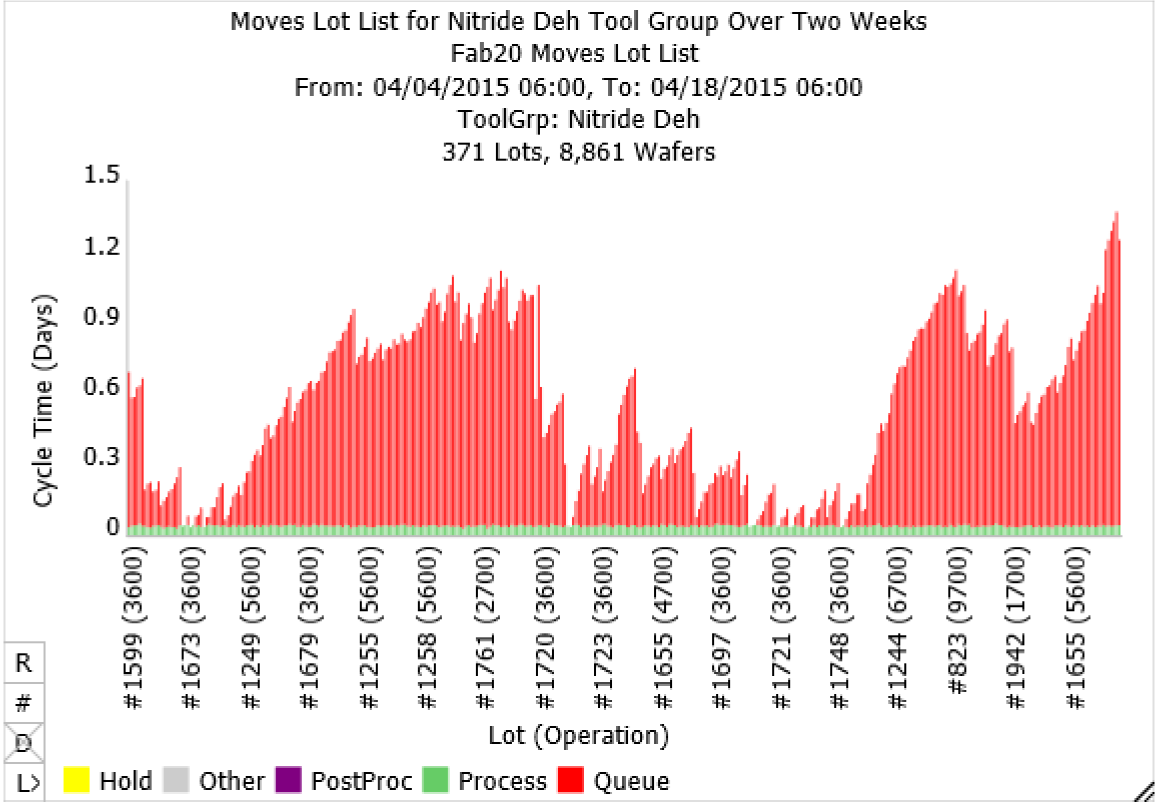
Can you identify time when tools in the group were available and WIP was waiting, but the tool wasn't running? This time, which FabTime calls standby-WIP-waiting, may be an indicator of staffing issues.

Is there sometimes standby time, but high variability in the periods of unavailable time for the tool group? In this case, efforts to work on reducing the variability of the downtime might be the most productive.

Returning to our example, we see below that the periods of high cycle time (April 6-9 and 15-18) correspond roughly with periods experiencing a high level of engineering time (the blue bars). During the very intervals when this tool group was experiencing higher arrivals than usual, the tools were also subject to extensive periods of engineering time. Also, note the dark gray sections spread throughout the chart. These represent time periods where at least one tool was available, and there was qualified WIP waiting, but the tool was not running. This may indicate that no operator was available to load the tool.

We can also take a quick look at the process time variability for the tool group, by looking at the detailed move transactions (shown at the top of the next page). In this case, it's clear visually that any process time variability is dwarfed by other effects.





We could also look at whether lack of tool qualification is affecting the tool group. In this demonstration case, all the operations processed on the tool group have at least two qualified tools. Adding a third qualified tool would obviously improve performance, particularly if it reduced the periods of unavailability due to engineering time. Comparing the number of qualified tools for recipes run through the tool group with the number of similar tools that could have been qualified is an excellent (though potentially time-consuming) idea. As noted in Issue 20.05, lack of tool qualification is often a significant and hidden source of queue time. Identifying your cycle time bottlenecks, and then checking those for process restriction issues, is a great place to start.

In summary, what we see in this example is a tool group that has enough capacity on average, but is regularly impacted by arrival variability, downtime (particularly engineering time), and (probably) lack of operators. These factors combine to generate a pattern of oscillating per visit cycle times – sometimes fine and sometimes not. If this tool group is visited multiple times, the cumulative impact on overall cycle time of the fab will be significant, even though the overall utilization for this tool (72.6% over this time frame) is not particularly high. A similar analysis could be conducted for any cycle time bottleneck.

Mitigating the Impact of Your Cycle Time Bottlenecks

Once you have your list of cycle time bottlenecks, and you've spent time understanding the specific causes of queue time for the most critical ones, the next step is to mitigate their impact. Exactly what to do is going to depend on your specific situation, of course. But here are a few general recommendations. These recommendations are distilled from past newsletter articles and our one-day cycle time management course.

Reduce Downtime Variability by:

- Separating PMs instead of grouping them (longer periods of unavailable time contribute much more variability, and hence queue time, than shorter periods, even for the same total amount of unavailable time).

- Breaking up the time that tools are unavailable because engineers are using them for experiments.
- Stocking more spare parts or increasing service contracts for key cycle time bottlenecks.
- Establishing policies restricting taking tools down for maintenance or engineering when other tools in the group are already unavailable and/or WIP levels are high.

Reduce Arrival Variability by:

- Applying a greedy policy to your batch tools (don't wait for full batches where utilization is low). (Issues 2.1 and 9.3)
- For less automated fabs where carts are used for manual lot transfer between steps, considering smaller carts or runners to reduce transfer batch sizes. (7.08 and 16.05)
- Modifying lot release policies (to release smaller numbers of lots at once) if cycle time bottlenecks arise early in the process flow.

Improve Operator Availability by:

- Minimizing the number of different tools that each operator or technician monitors at one time (for cycle time bottlenecks). (4.06 and 7.02)
- Staggering break schedules for cycle time bottlenecks.

Improve Effective Process Time Variability by:

- Reducing the number of hot lots in the fab, especially if your cycle time bottlenecks are ever held idle for anticipated hot lots. (19.03)
- Tracking post-process time (where the operation is finished but move out doesn't occur right away) and creating better signals to unload the tool.
- Minimizing setups. (6.07)
- Eliminating "future holds", where an engineer can specify that a lot will be held at a tool at some future date (by which time the engineer might be on vacation). (6.06)

Also:

- Look for recipes that have only one or two qualified tools. Where possible, qualify more (20.05).
- Look for soft constraints (where technically multiple tools are qualified but in practice some of those tools are rarely used). (20.05)
- Consider cycle time bottlenecks for potential capacity expansion projects. If a tool group is a major contributor to cycle time, it may be worth adding capacity even though the tool group is not the most heavily utilized in the fab. This is especially true for batch tools, which can contribute disproportionately to cycle time. If you have cycle time bottlenecks that are true one-of-a-kind tools, these should be among your top candidates for capacity expansion.

Conclusions

For fabs that care about maintaining cycle time, it makes sense to focus not just on improving capacity bottlenecks but also on improving cycle time bottlenecks. In this article we follow Thomas Beeg's recommendation and define cycle time bottlenecks as the tool groups that are contributing the most queue time to cycle time in a fab. We describe methods for both identifying cycle time bottlenecks and analyzing them. We close with a brief summary of concrete recommendations for mitigating the primary contributing factors to cycle time bottlenecks.

Closing Questions for Newsletter Subscribers

Do you track cycle time bottlenecks in your fab? Do they tend to align with capacity bottlenecks, or do you find that other tool groups also drive up queue time? Do you spend time on cycle time improvement efforts for tools that are not capacity bottlenecks?

Acknowledgements

FabTime would like to thank Thomas Beeg from CREE/WOLFSPEED for his subscriber discussion remarks, which helped to motivate this article.

Further Reading

- J. Robinson and F. Chance, "Impact of Batch Size Decision Rules on Cycle Time (Issue 2.1)," *FabTime Newsletter*, 2001.
- J. Robinson and F. Chance, "Arrival Variability and Cycle Time (Issue 4.05)," *FabTime Newsletter*, 2003.
- J. Robinson and F. Chance, "In-Depth Guide to Operators and Cycle Time (Issue 4.06)," *FabTime Newsletter*, 2003.
- J. Robinson and F. Chance, "Quantifying the Effect of Tool Downtime (Issue 5.07)," *FabTime Newsletter*, 2001.
- J. Robinson and F. Chance, "The Three Fundamental Drivers of Fab Cycle Time (Issue 6.05)," *FabTime Newsletter,* 2005.
- J. Robinson and F. Chance, "Cycle Time and Holds (Issue 6.06)," *FabTime Newsletter,* 2005.
- J. Robinson and F. Chance, "Setup Avoidance and Dispatching (Issue 6.07)," *FabTime Newsletter,* 2005.
- J. Robinson and F. Chance, "Operator Variability and Cycle Time (Issue 7.02)," *FabTime Newsletter,* 2005.
- J. Robinson and F. Chance, "Cycle Time Metrics Baseline (Issue 7.03)," *FabTime Newsletter*, 2006.
- J. Robinson and F. Chance, "Ways that Fabs Create Arrival Variability (and Cycle Time) (Issue 7.08)," *FabTime Newsletter*, 2006.
- J. Robinson and F. Chance, "Batch Loading Policies for Wafer Fabs (Issue 9.03)," *FabTime Newsletter*, 2008.
- J. Robinson and F. Chance, "Improving Factory Cycle Time through Improvements at Non-Bottleneck Tools (Issue 10.09)," *FabTime Newsletter*, 2009.
- J. Robinson and F. Chance, "Identifying Sources of Arrival Variability (Issue 16.05)," *FabTime Newsletter*, 2015.
- J. Robinson and F. Chance, "Cycle Time and Hot Lots: Updated (Issue 19.03)," *FabTime Newsletter*, 2018.
- J. Robinson and F. Chance, "The Impact of Tool Qualification on Cycle Time (Issue 20.05)," *FabTime Newsletter*, 2019.
- N. S. Grewal, A. C. Bruska, T. M. Wulf, and J. K. Robinson, "Integrating Targeted Cycle-Time Reduction Into The Capital Planning Process," *Proceedings of the 1998 Winter Simulation Conference*, Washington, DC, 1005-1010, 1998. Available for download from FabTime's website.

As mentioned in the announcements section above, FabTime will be making all of the past newsletters available for direct download by subscribers soon. If you would like any of these issues in the meantime, please send your request to newsletter@FabTime.com.

Subscriber List

Total number of subscribers: 2783

Top 20 subscribing companies:

- ON Semiconductor (218)
- Infineon Technologies (158)
- Intel Corporation (117)
- Micron Technology, Inc. (116)
- GlobalFoundries (101)
- Maxim Integrated Products, Inc. (89)
- NXP Semiconductors (80)
- Carsem M Sdn Bhd (70)
- Microchip Technology (70)
- Skyworks Solutions, Inc. (64)
- STMicroelectronics (64)
- Western Digital Corporation Inc. (62)
- Texas Instruments (53)
- Seagate Technology (52)
- X-FAB Inc. (49)
- Qualcomm (47)
- Analog Devices (42)
- TowerJazz Semiconductor Ltd. (34)
- Cree / Wolfspeed (33)
- Zymergen (32)

Top 3 subscribing universities:

- Arizona State University (8)
- Ecole des Mines de Saint-Etienne (EMSE) (7)
- Virginia Tech (7)

New companies and universities this month:

- font
- Ball Aerospace
- Clas-SiC Wafer Fab
- Denselight Semiconductors
- GTA Semiconductor
- MangoGem
- McKnight Consulting
- Othman and Partners, LLP
- PWB Interconnect Solutions
- SK Hynix
- The Improvement Partners
- Transphorm Inc.
- Varex Imaging
- Vienna University of Economics and Business

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 and receive the current issue of the newsletter each month. Past issues of the newsletter are currently only available to customers of FabTime's web-based digital dashboard software or cycle time management course.

To subscribe to the newsletter, send email to newsletter@FabTime.com, or use the form at www.FabTime.com/newsletter-subscribe.php. 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.