FabTime Newsletter

Volume 22, No. 2 June 2021

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) letting them configure their own charts, so that they don't need assistance from IT for each new data request; and 2) including them in a community of people around the world working to improve fab operations.

Editor: Jennifer Robinson

Contributors: Mike Olewine (Sandia National Laboratories), Thomas Beeg from Cree/Wolfspeed; Jon Rossi

Date: Tuesday, June 8, 2021 – Vol. 22, No. 2

Keywords: Variability; Factory Behavior; Industry Cycles; Operators

Table of Contents

Welcome

Welcome to Volume 22, Number 2 of the FabTime Cycle Time Management Newsletter. In this issue we have a few links from Jennifer's LinkedIn, but no other announcements. Our software tip of the month is about using FabTime to identify possible instances of soft dedication in the fab. We have subscriber discussion about WIP Utilization %, sending SMS email to Japanese cell phones, breaking up maintenance events, managing SPC in a shared fab environment, and paying off technical debt.

Our main article was inspired by the many reports we've seen and heard lately of fabs facing capacity and cycle time challenges. We've spent 20+ years thinking and teaching about ways to improve cycle time for existing fabs. We've always closed our cycle time course with a list of our top ten recommendations. Most of these have been covered in past newsletters, but we thought there would be value in a single article that describes all ten. We hope you find these recommendations useful.

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

Community News/Announcements

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:

- An article about Semefab Ltd. and how Allan James started out in 1986 by raising £500,000 to buy a mothballed 4" #waferfab. Today Semefab operates 3 fabs and has 120 employees making MEMS devices.
- A piece by Steve Frezon from NXP Semiconductors about why and how it was such a major project to get their fabs back up and running after the recent weather-induced shutdown. Interesting reading for anyone wondering why the global chip shortage has no instant fix. (Via Tom Salmon from the FOA.)
- A new piece by Mark LaPedus in Semiconductor Engineering about the technology race in leading edge foundries. Mark goes through the history of the foundry market, with lots of detail about the technology nodes, Intel Corporation vs. TSMC, and more. This piece is well worth a look for those curious about the semiconductor industry.
- Various links about the chip shortage, including this Financial Times piece about UMC expanding mature chip capacity, this WSJ piece about various factors deepening the shortage, and this Wired piece about how the chip shortage is driving up television prices.

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® User Tip of the Month

Identify Cases of "Soft Dedication"

As discussed in more detail in the main article below, "soft dedication" occurs when a group of tools is officially qualified to run a particular operation, but in practice only a sub-set of that group is used (at least some of the time). Soft dedication is a hidden source of cycle time in wafer fabs. The question is: how do you use FabTime to find out where it is occurring?

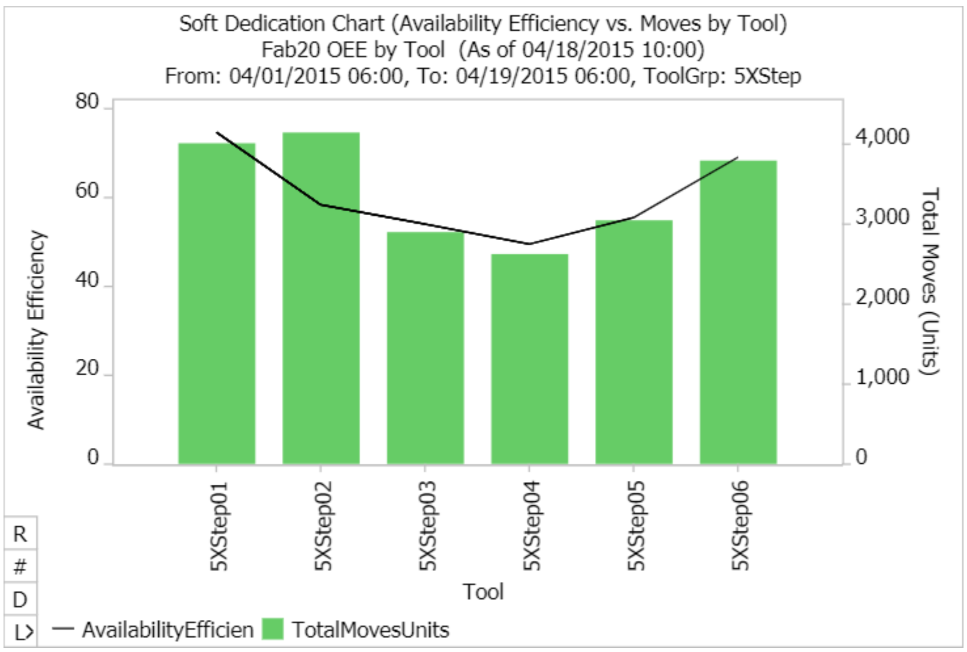
Soft dedication can be a tricky thing to tease out from reporting data. One place to start is to look for tool groups (or capacity types) where the total moves aren't balanced across tools. This gives you an idea of places where soft dedication may be causing an imbalance in tool utilizations.

It's usually not enough to compare the moves per qualified tool, however, because different tools may have different short-term availabilities. What we really want is to look at moves per available hour. This is not a standard chart in FabTime (though perhaps it should be?), but you can create something similar using an edited version of the Tool OEE Pareto chart.

1. Generate the Tool OEE Pareto chart. Slice it by tool and filter for a key tool group or capacity type that you expect to be reasonably cross-qualified. Click "Chart Only" or "Left/Right" to the upper left of the chart so that the data table will not be below the chart.
2. Click the "Edit Chart" checkbox immediately below the chart. For each field in the resulting table, except "Object Plus Description", set the dropdown in the "On Chart" column to "Unused".
3. Click the "Show All Fields" checkbox below the chart. Set "On Chart" for "Availability Efficiency" to "Line" and set "On Chart" for "Total Move Units" to "Bar". For "Total Move Units" change the "Settings" drop-down to "Green" (or your color of choice) and the "Y-axis" to "Right-Y".

4. Type "Availability Efficiency" in the "Y Axis Title" field and "Total Moves (Wafers)" in the "Y2 Axis Title" field. Click "Go" immediately below the axis editing section. (The other chart editing elements should auto-update the chart, but this change does require using the "Go" button.)
5. Set the date range for the chart to be at least a week and press "Go" next to the date range fields.
6. (Optional) Add a custom title to the chart like "Soft Dedication Chart (Availability Efficiency vs. Moves)".
7. Add the chart to a home page tab to save it (so that you don't lose the work of setting it up).
8. Returning to the chart, look for mismatches across tools, where one or more tools has either a higher or lower move rate relative to availability. Test for different period lengths to confirm. Change the "ToolGrp" or "Tool" filter to explore soft dedication in other tool groups.

In the example below, we see that tool 5xStep02 has a higher move rate than the other tools relative to availability. In our example, we looked at 2 ½ weeks of data as well as four days of data, to similar results. This suggests an operator preference towards 5xStep02 relative to the other tools. More common is to see one or more tools that have a lower move rate than expected relative to their availability efficiency.



We realize that configuring this chart for the first time is a bit cumbersome. We have added an item to our development list to make it easier to remove all the fields from a chart with one click, and we will consider adding this as a standard FabTime chart. But the good news is that a) this exercise is good practice in editing charts (showing that you can make a chart showing anything included in the underlying chart data) and b) because you can save the chart, you only need go through the configuration exercise once. Of course, you can save different versions of the chart for different key tool groups, too, or use Tab Filters on your home page.

We hope that you find this tip useful.

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 (note new link). Thanks!

Subscriber Discussion Forum

WIP Utilization %

An anonymous subscriber asked: "Do you have any suggestions / recommendations to measure "WIP based utilization"?

My challenge is that given WIP fluctuates, I cannot always expect near 100% utilization over a 24-hour span, and I need a way to make sure my team is maximizing possible utilization at all times (if we only have WIP that supports 50% utilization for example, then we have to meet that 50%... 45% is loss opportunity).

I have been thinking about a WIP / Utilization ratio. I thought I would take the path of least resistance and ask you first."

FabTime Response: We wrote about a metric for this at the tool/tool group level back in Issue 5.05 of the newsletter and have had it in our software for many years. We call it WIP Utilization %. We define that as Productive Time / (Productive Time + Standby WIP Waiting Time), where Standby WIP Waiting Time is time that there is WIP in front of the tool, but the tool is not being run (vs. what we would call true standby time, when the tool is idle because there is no WIP).

WIP Utilization % will approach 100% if, whenever WIP is waiting, and a qualified tool is available, the WIP is processed as soon as possible. Driving WIP Utilization to 100% generally minimizes per-visit cycle times through the tool and helps to maximize shift-level throughput. This metric overcomes several shortcomings of the standard utilization definition as a shift-level metric for operators.

Follow up question from the subscriber: "How do you handle dedication? I have an available chamber, but Engineering has inhibited the WIP from running on it. To me, it counts as WIP time and goes into the denominator. You then force the discussion with Engineering about opening run paths.

FabTime Response: In our software, Standby WIP Waiting Time is enabled if the tool is in standby and the WIP transactions tell us there is WIP there that is qualified to run on that tool. In your dedication example, the WIP probably wouldn't count as being qualified to run on the chamber, so it would not show up as Standby WIP Waiting. Which means it wouldn't go in that denominator (unless it's some kind of "soft" restriction, such that the WIP does show up as qualified to run but people just know it's not supposed to run).

We do agree with you philosophically that it would be good if that time did go into the denominator, because of course we want to provide ammunition for discussion with the engineers about opening additional paths. But we're not sure how you would distinguish that in an automated way from the data.

[As a side note: There is also a problem with the Standby WIP Waiting Time data in cases where WIP is held up at an earlier step to avoid sending it downstream (e.g., held in front of the clean step to avoid time constraints at the next step). In this case you'll see lower WIP Utilization at the clean step. We find in practice that people understand this impact and interpret the data accordingly. No metric is perfect in all cases.]

Sending SMS Email to Japanese Cell Phones

FabTime's Frank Chance had a question and thought that the subscriber community might be able to help. We are wondering if anyone knows what the email address convention might be to send SMS messages to arrive as texts to cell phones for the Japanese carrier SOFTBANK. In the US, we know that it is possible to send messages to cellnumber@att.com, for example. Some of our customers use this functionality to send FabTime alerts to arrive as text message. However, we don't know what this convention is, or if such a thing exists, in Japan. If anyone has knowledge of this, we would be grateful.

Breaking up PMs

Mike Olewine from Sandia National Laboratories wrote in response to last month's issue "I can attest that shorter more frequent planned down-time is an effective tool. In the late eighties I worked for Ramtron at the NMB Semiconductor facility in Tateyama, Japan. They planned all maintenance to take place on Wednesday day shift. All tools went down during this time. The operators did cleans and wipe-downs of the surfaces while maintenance crews (usually from the tool vendors) did any PM's (weekly, monthly, quarterly, etc.) broken down to 8 hours once a week. This was an unusual fab because at the time they ran a single product, a very high-speed SRAM. They had no pellicles because the masks never left the lithography tools (they had exposure/track combinations for each mask level and a couple backup tools)."

Thomas Beeg from Cree/Wolfspeed wrote on LinkedIn: "This is a great topic and I think there is plenty of potential out there. Biggest challenge I see is, that often the maintenance team is measured with different KPI than lot cycle time and traditionally getting on a tool more often, even when it is with individually shorter down time duration, is a challenge. Looking forward to the discussion / feedback in the next newsletter."

Managing SPC in a Shared Fab Environment

Long-time subscriber Jon Rossi wrote: "There are many places, especially in universities, which share a fab among several users. I thought there may be some best practices relative to use of SPC in that situation. I can imagine a few approaches but was wondering if one or two have come across as superior and/or have been published or maybe even discussed in your archives. It may be one of those things so specialized that no general approach is appropriate."

FabTime Response:
We have worked with fabs that have needed to share information directly with different customers. We have an option in our software that allows designation of a specific filter for an individual user. That user is then only allowed to view charts that are populated with that filter (to only see their own data). We could see an approach like this being used whereby the SPC data is calculated separately for each fab user according to that user's products. It's not clear how effective an approach like that would be for SPC data, however. We unfortunately do not have any direct experience with that. Nor have we written about this in past newsletters. We are happy to raise the question with our subscriber community, however, to see if anyone has ideas or links to published results to share on this topic. Subscribers?

A Few Thoughts on "Technical Debt"

FabTime co-founder Frank Chance brought up the topic of technical debt on a recent meeting with our User Group. This term originated in the software development field, referring to when a coding team makes choices up front that speed delivery of a product but don't age well over time. Technical debt can result in something that is working but has limitations. It's not working the way you would design it now if you were starting from scratch, and it should be replaced or updated. Technical debt is often invisible to end users, however, which means that software companies often don't have an incentive to fix the underlying issue. As time passes, though, technical debt increasingly holds you back.

Our technical team has been working to balance projects that pay down technical debt against other priorities (like adding interesting new charts). For example, we migrated to a different provider for source code control. We are exploring options to outsource our current home-grown project ticketing system. We integrated a third-party engine for building data tables in FabTime, unlocking a wide range of new flexibility.

It struck me, Jennifer, that this concept can be applied to other areas of our business, and to my own personal life. For example, I struggled with regular internet outages until I finally worked with my provider to get a new modem and router. That project took quite a few hours over several days, but my family has more than recovered that time in increased productivity. The challenges, looking at other such projects, are finding those few hours (or many hours, in some cases) and overcoming inertia. I feel like the idea of technical debt might help me to make the leap on other projects. What will the payback period be for the time and money I invest now? I don't need that generator today, but how much will it be worth this fall, during the CA power shutoffs that seem inevitable?

Putting my sales hat on, I can see a parallel to installing a commercial reporting system like FabTime vs. continuing to use an older in-house system. It takes time and money to make such a change. And the forces of inertia are high (people rarely want to change the reports that they are accustomed to looking at). But what if by going through this effort you can save manufacturing management and IT staff time every day going forward? For years? What if by having access to FabTime charts, and the collective wisdom of

FabTime's user group, you can squeeze a bit of extra capacity out of your toolset, or bring down your cycle time? How much is that worth over the long-term? (Or over the short term, in today's environment?)

Of course, the "how much is that worth" question is frequently difficult or impossible to quantify. But for me, the concept of technical debt helps to at least get me out of my rut of only looking at incremental improvements. I'm imagining the appreciation of that technical debt over time, and what the benefit might be of paying it off now.

In discussing this further with Frank, we wondered about technical debt in the broader "buy vs. build" discussion. Is building a software system yourself always a recipe for accumulating technical debt? Perhaps not. If you buy a product and the vendor goes out of business, or the quality goes downhill, you might still end up with technical debt. But building it yourself certainly accumulates more technical debt in some cases. The example that strikes us is gigantic Excel spreadsheets that fabs use for capacity planning. These typically have all sorts of complex formulas that are a mystery to the people who rely on the spreadsheets' output. They represent a *lot* of technical debt, and when the employee who built a particular spreadsheet retires, the bill comes due. A similar issue lurks in some dispatch systems.

There is, of course, a larger parallel discussion of technical debt in buy vs. build for wafer fabrication. We know many people who are operating older fabs that aren't set up the way they would be if they were built from scratch today. That's technical debt, writ large.

Have any of you used this concept, in your work or personal lives? Has it helped? We welcome your feedback.

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

10 Recommendations for Fab Cycle Time Improvement

Background

The semiconductor industry is currently experiencing considerable strain. As other segments of the manufacturing industry, particularly the automotive sector, recover from the pandemic, demand for chips has spiked. Readers of this newsletter understand well the core relationship between cycle time and capacity utilization in a fab. In the presence of variability (which we will always have in fabs), cycle time increases rapidly as a fab approaches 100% utilization of the bottleneck(s). Under current demand pressures, fab cycle times are increasing in many sectors. Lead times (which include even getting your order into the fab at this point) have increased from four to eight weeks to 24 to 52 weeks in some categories (per a Nikkei Asia analysis re-shared by Jennifer on LinkedIn).

While many efforts are ongoing to add capacity (either by building new fabs or by adding equipment to existing fabs), this is not a quick process. Fabs are expensive and complex and can take years to build from scratch. We did read recently about a small fab that is planning to expand capacity by doubling staff, in order to start running 24 hours a day, but this is an unusual case given that most fabs already run 7 by 24 schedules. Thus, efforts to improve fab cycle time by making operational changes are of increased value right now. These types of improvements, which center on reducing variability to move to a more favorable operating curve, are at the heart of FabTime's educational outreach via our cycle time management course and this newsletter.

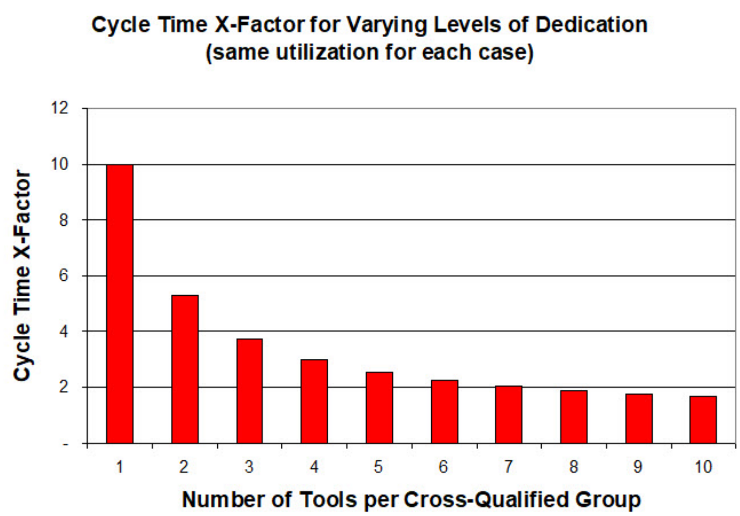
For many years, we've been concluding our course with a slide that lists our top ten recommendations for improving fab cycle time. These recommendations, sorted in "bang for the buck" order, recap topics discussed in more detail throughout the course. Jennifer recently shared the list on LinkedIn, where it generated some great discussion. In this article, we add a bit of detail and context to the items on the list.

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.

Identify and eliminate single path operations (if possible).

As discussed most recently in Issue 20.05, one of the fundamental drivers of cycle time at the tool group level is number of qualified tools. This is because, in the presence of variability, having another potential path to be processed significantly reduces the chance that a lot will have to wait. When there is only a single path for an operation, any variability at that tool impacts all lots in queue, whether this variability is in the form of unscheduled downtime, scheduled downtime, engineering time, operator delays, or long process times of other lots. Having a second path, at the same utilization level, decreases the average cycle time through an operation by approximately 50%. Adding a third path decreases the cycle time by another 20-30%, with the relative improvement tapering off after that.



This behavior is illustrated by the graph to the left, which shows varying levels of cross-qualification for 10 operations and 10 tools. On the left is full dedication (one tool per operation). On the right is full cross-qualification (any of the 10 tools can run any of the 10 operations).

Sometimes, of course, a fab will have true one-of-a-kind tools. In this case, there is nothing to be done about those single paths (except to prioritize such tools for any capacity expansion projects and work in the meantime to minimize variability on those tools).

However, it is often the case that tool qualification choices lead to single path operations. Efforts to qualify a second tool are of high value, reducing average cycle time through the operation by about 50%. This is a low cost, high benefit solution. FabTime's number one recommendation for improving cycle time without adding capacity is to identify single path operations and work to qualify at least one more tool for each. (For more detail, see Issues 20.05 and 9.01.)

Check for soft dedication due to operator preferences.

Sometimes, even when your engineers qualify multiple tools for every operation, a closer look at the data reveals “soft dedication”. This is when multiple tools are qualified to run a particular operation, but the operators for some reason prefer one tool over the other(s). This can happen because of:

- Layout issues (the other qualified tools require more travel time)
- Tool capability differences (one tool is faster, or easier to use)
- Data communication issues (not everyone knows that another tool has been qualified)

What can happen in this case is a load imbalance across the tools. One tool ends up being operated at a higher utilization than necessary, while another has excess capacity. Because cycle time increases non-linearly with utilization (especially as you get closer to 100% utilization), this leads to increases in overall cycle time. It’s a hidden source of cycle time.

To identify soft dedication, start by looking at the moves relative to availability efficiency by tool for key tool groups or capacity types. If a tool group is fully cross-qualified, you can expect this to be similar across tools. Where one tool seems to be used more (or less) than the others relative to availability, there may be soft dedication occurring. An unexpected difference may be cause for further investigation. An example is shown above in the Tip of the Month section of the newsletter.

It’s important to add that just because the operators are doing something that isn’t planned for in your capacity model doesn’t mean that they are wrong. The reason for the soft dedication may be a good one. We once worked with a fab where we observed that one tool had a consistently lower than expected number of moves, after adjusting for availability. When we asked about it, our contact said “Well, that tool is actually in a separate building from the other tools.” We said something along the lines of “Aha!”. In that example, we recommended modifying the capacity planning model to reflect the soft dedication that was occurring in practice.

Reduce transfer batch sizes between steps.

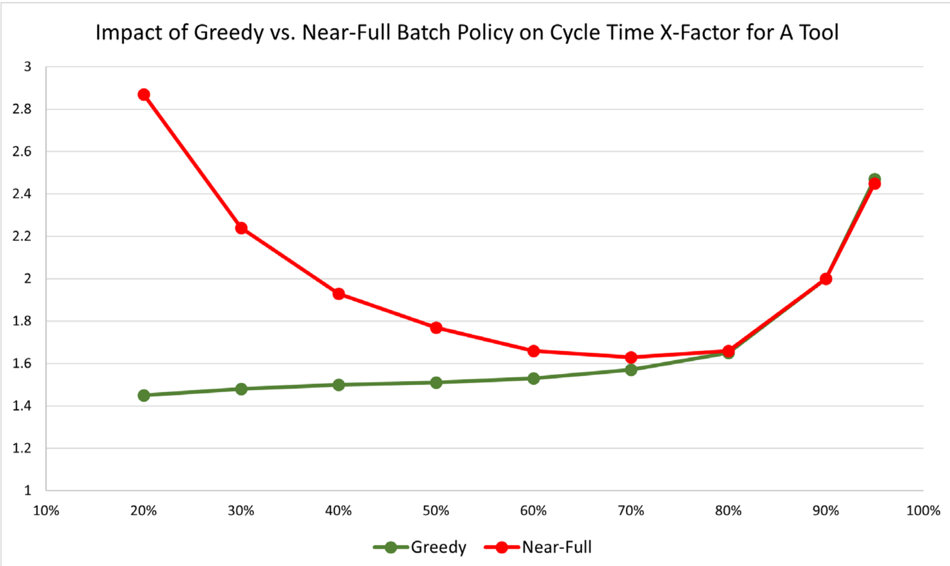
In less automated fabs, transfer batching can be another hidden source of cycle time problems. Where carts are used to transfer lots between steps, operators have an understandable bias towards efficiency. They often wait until a cart is full before transferring any of the lots to the next step. While this is efficient in terms of reducing steps for the operators, it is terrible for cycle time. Queue time is introduced as the lots wait upstream, especially for the first lot added to the cart. Then, when the lots are unloaded, a bubble of WIP arrives all at once to the next step. This arrival variability introduces queue time downstream. We have worked with fabs that reduced the overall cycle time of a fab by purchasing smaller carts, or by hiring dedicated runners to transport lots. These solutions aren’t free, but they are low-cost relative to the potential cycle time reduction that comes from removing this variability. See Issue 9.02 for more details.

Run batch tools under a greedy policy.

Most wafer fabs have batch operations at which up to eight lots can be processed at one time, where the process time is the same regardless of the number of lots in the batch. Often the process time for the batch is lengthy, as in furnace operations. Because of the long process times, there is an incentive to run these tools with full batches. That way, the capacity for the unused slots on the tool is not wasted. When a batch tool is available and has a partial batch ready to go, there’s a question of whether to go ahead and start the batch or wait for more lots to fill the batch. Some fabs have policies in place to wait for a full (or nearly full) batch.

The problem with this approach is that it drives up cycle time for the lots that are already there (as in the transfer batch case discussed above). Then, after a full batch is run, the batch tool sends highly variable arrivals downstream. The arrival pattern downstream looks like "nothing, nothing, nothing, nothing, … big batch of eight lots". Often the lots must wait again downstream.

Full batch policies are a particular problem for tools that are not highly utilized – there's a higher probability of having to wait a long time to fill the batch. There, a greedy policy (just run the tool if there is anything there to run) can generate significantly lower cycle times. An example, generated using simulation, is shown below.



When the tool is highly utilized, it doesn't matter whether you run a greedy or near-full batch policy, because the batches usually are full (the lines overlap).

When you run a greedy policy at those high utilizations, even if you occasionally run a less than full batch, there will be a full batch waiting by the time the partial batch is finished. Batches are usually run close to full.

The greedy policy also works well at low utilizations. The near-full batch policy, on the other hand, is terrible at low utilizations, introducing a considerable amount of unnecessary cycle time. Therefore, we recommend ensuring that your batch tools are being run under a greedy policy, or at least not a full batch. See Issue 9.03 for more details, including a full-fab simulation example.

Separate maintenance events instead of grouping them.

Downtime is a major source of both capacity loss and variability in wafer fabs. Fabs undertake many downtime reduction programs, of course. But one recommendation that we like to make that is not always intuitive for people is to separate maintenance events and other periods of unavailable time.

While it can be tempting to group smaller maintenance activities together, or to group them in with other downtime events, this is generally counterproductive for cycle time. What's best for cycle time is to have each period of unavailable time be as short as possible, particularly for one-of-a-kind tools, to keep lots moving through the tool smoothly, and prevent WIP bubbles from building up. For cycle time, then, it's better to break PM activity into the smallest possible chunks and make the tool available for production in between.

Clearly, there are limits to this approach, depending on the qualification time required to bring a tool back up, staffing issues, etc. However, it may be worth checking your PM schedules, to see where you may be introducing more variability into the fab than needed. Tracking average and maximum time offline for scheduled downtime, rather than tracking the time between events, is a very good place to start. See last month's newsletter, Issue 22.01, for more detail.

Minimize the number of distinct tools for which each operator is responsible, and stagger break schedules.

Another often hidden source of cycle time in wafer fabs is operator delay. Any time an operator is responsible for more than one tool, there is a chance that multiple tools will require the operator's attention at one time. Lot 1 is ready to load on Tool A while Lot 7 is ready to unload from Tool C. Because people can't be in two places at one time, either Lot 1 or Lot 7 will incur extra cycle time. Tool A or Tool C will also lose some capacity. Operator delay can push a tool to a steeper place on the operating curve, generating a higher-than-expected cycle time per visit.

One way to get a sense of where this is happening is to report what we call Standby WIP Waiting Time. Standby WIP Waiting is time when a tool is available and WIP is waiting but the tool is not being run. You can't rely on operators to log a tool into a Standby WIP Waiting state, but you can infer it by knowing when WIP has moved out of the previous operation and is now in queue for the current operation. (See the discussion about WIP Utilization % in this month's Subscriber Discussion Forum above for more detail.)

Another metric available in some fabs is post-processing time, where the lot has completed processing but not been moved out from the tool. Not all fabs log the end-run transaction as separate from the move out, but those that do can record post-process time.

Once you know where operator delay may be causing cycle time problems, you can check whether you have operators who are spread too thin, managing too many tools. We've had various discussions in the newsletter over the years about whether and how to measure operator utilization. But what we do know is that in the presence of high variability, the more tools you have waiting for a single operator, the more likely you are to incur operator delay.

As a related recommendation, staggering break schedules is best for cycle time (though we imagine that most fabs are doing this already). Dynamic X-Factor can give you a fab-level indictor for where break schedules may be causing trouble (see Issue 9.04). For more about operators and cycle time, see Issues 4.06 and 7.02.

Reduce the number of hot lots in the fab, especially hand-carry lots.

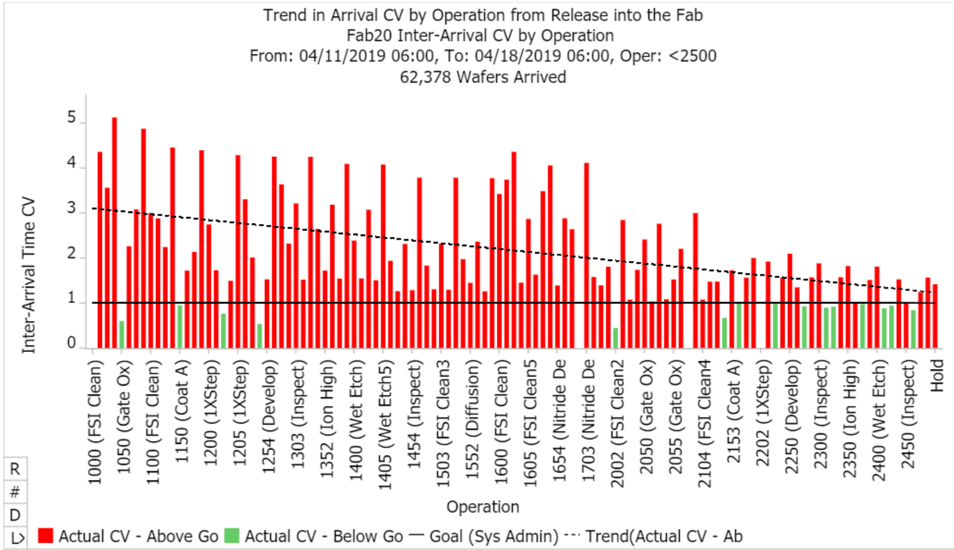
Hot lots are a fact of life in wafer fabs, perhaps even more so now, in the wake of the chip shortage. Hot lots allow a fab to get some small number of lots out with a relatively low cycle time, without having to reduce fab utilization in the way that would be necessary to reduce the overall fab cycle time. But hot lots are not free (despite what some in the organization might believe). Hot lots add variability, which increases cycle time. The greater the number of hot lots you have, the more likely they are to interfere with one another. You can even get into a vicious cycle where you make more lots hot, and the added variability drives up cycle time, leading to a need for even more hot lots.

Front-of-the-line hot lots are less of a problem than hand-carry lots. Front-of-the-line, or regular, hot lots are prioritized ahead of other lots in your dispatch system. However, you typically don't hold tools idle for them, or break setups. In the best case, front-of-the-line hot lots move queue time from the hot lots to the regular lots, without changing the overall average cycle time. In practice, some capacity is probably still lost at batch tools and tools with setups, which is why we recommend keeping the quantity of regular hot lots below 5% of total WIP.

Hand carry lots (also called rocket lots, platinum lots, and many other names) are more disruptive. Any time you hold tools idle ahead of a lot that is expected, you lose capacity. Similarly for breaking setups, running batch tools nearly empty when full batches of other lots could have been run, etc. Hand carry lots also typically require extra management attention, which is thus diverted from other activities. Our recommendation, if you must have hand-carry lots, is to have no more than one or two in the fab at the same time. See Issue 19.03 for details.

Smooth the flow of arrivals into the fab.

At the tool level, arrival variability can be a significant contributor to cycle time. Performance is much better under a steady pattern of arrivals than a lumpy pattern of arrivals. Arrival variability is less of an issue for major bottlenecks that always have a large queue waiting but can degrade cycle time performance quite a bit for tools with moderate utilization. The discussions above about batch transfer and batch processing were centered in part around minimizing arrival variability throughout the process flow.



A source of arrival variability that is more directly controllable is lot releases into the fab. For cycle time, it's generally better to release fewer lots at a time steadily throughout the day, rather than releasing a large batch of lots into the fab at one time.

While the arrival variability from lumpy lot release policies does dampen as WIP makes its way through the process, variability from lot release can have a poor impact on cycle time at operations early in the process flow. An example from our demo model is shown. The coefficient of variability of arrivals is shown to each operation, in operation order. This data is noisy (reflecting sources of variability such as batch tools) but shows a clear trend downward over the set of operations included.

Some fabs do release lots in batches specifically timed to optimize loading of early batch tools. This makes sense if those batch tools are highly utilized. But, as discussed above, this strategy may be introducing unnecessary variability in cases where the batch tools are more lightly loaded overall. See Issue 7.08 for more detail.

Check setup avoidance policies to make sure that low volume lots aren't waiting too long, especially on non-bottlenecks.

Some tools, such as implanters, require significant setups when changed over from running one type of process to another. Setup avoidance policies are dispatch policies that attempt to minimize that setup time. In the simplest implementation, a setup avoidance policy will prioritize lots of the same recipe over other lots. If there is a lot in queue that doesn't require a setup, that lot will be processed. The setup will only take place if there are no lots of the current recipe waiting. In some cases, operators may even go further, waiting until other lots of the same recipe arrive, so as not to incur the setup time.

Setup avoidance policies are important in minimizing lost capacity on key tools. Setup time is non-value added. Operators who are measured based on moves have an incentive to do as few setups as possible. However, what can happen in the presence of setup avoidance policies is that lots from low volume routes can wait … forever.

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. See Issue 6.07.

Make dispatching decisions to keep critical downstream tools from starving.

This recommendation is last on our list not because it is not important, but because it requires more effort to implement. Dispatch decisions (which lot to process next on each tool) are normally local, as compared to scheduling solutions, which attempt to optimize across the entire fab. A useful first step in making dispatch decisions a bit more global is to incorporate downstream information into the decision with the goal of keeping critical downstream tools from starving, and balancing WIP across the fab.

WIP smoothing or line balance policies look some number of steps downstream for each lot in queue and prioritize lots that have the least WIP ahead of them. We have done some research (see Issue 15.01) that showed that in the presence of extended downtimes, a Critical Ratio (CR) dispatch rule (common in fabs) can lead to undesirable behavior – WIP bubbles that oscillate between the front and back of the line.

If you can instead focus on keeping your line balanced (roughly equal WIP throughout segments or sub-segments), you may find your fab easier to manage. You'll see fewer WIP bubbles, fewer starved bottlenecks, and lower cycle time variability. This lower cycle time variability will end up helping with on-time delivery. You can also still use CR as a secondary dispatch factor, of course, but we recommend that line balance be a primary goal of your dispatch policies.

In general, once you have tackled the low-hanging fruit from recommendations 1 through 9, a logical next step is to move from local dispatch rules to rules that incorporate downstream information into upstream decisions.

Conclusions

Fabs are usually struggling with competing pressure to reduce cycle time and increase tool utilization. In the current chip shortage, these pressures have been exacerbated. While there are efforts underway around the globe to add wafer fab capacity, these initiatives (particularly those involving new fab construction) take time.

The good news is that it is possible to reduce fab cycle time by reducing variability. When successful, variability reduction gives fabs a choice. They can achieve better cycle time, or they can push to a slightly steeper place on the operating curve and squeeze out a bit of extra throughput. These days, even a small improvement in either would be welcomed by most fabs.

In this article, we have shared ten recommendations for improving fab cycle time through changes in operating practices. Most of these recommendations are relatively low cost, with the possible exception of recommendation 10 (modifying dispatch rules to include downstream information). They rest on understanding the way that decisions made by people in the fab, from lot release to PM scheduling, affect variability.

Most of these recommendations will not be surprising to people who understand fabs. But we hope that they will serve as a reminder, or possibly as ammunition to help spur changes. We welcome your questions and feedback.

Closing Questions for Newsletter Subscribers

What do you think of this list of recommendations for improving fab cycle time without purchasing equipment? What would you add or remove from the list?

Further Reading

The FabTime newsletters listed below cover the topics from this article in more detail. They are listed in the order in which they were discussed above. These issues are all available to FabTime newsletter subscribers from our website at https://www.fabtime.com/newsletter-archives.php. The current password for newsletter download is "FabTimeCommunity" (no quotation marks).

These topics are also discussed in our half day remote cycle time management course. More information about the course is available on our website or by contacting Jennifer.Robinson@FabTime.com.

- The Impact of Tool Qualification on Cycle Time (Issue 20.05)
- Our Top Recommendation for Cycle Time Improvement: Tackle Single Path Operations (Issue 9.01) (includes discussion of soft dedication)
- Manual Lot Transfer in Wafer Fabs (Issue 9.02)
- Batch Loading Policies for Wafer Fabs (Issue 9.03)
- On Breaking Up PMs and Other Unavailable Periods (Issue 22.01)
- In-Depth Guide to Operators and Cycle Time (Issue 4.06)
- Operator Variability and Cycle Time (Issue 7.02)
- Dynamic X-Factor and Shipped Lot X-Factor (Issue 9.04)
- Cycle Time and Hot Lots: Updated (Issue 19.03)
- Ways that Fabs Create Arrival Variability (and Cycle Time) (Issue 7.08)
- Setup Avoidance and Dispatching (Issue 6.07)
- Dispatching and Line Balance (Issue 15.01)

Subscriber List

Total number of subscribers: 2851

Top 20 subscribing companies:

- ON Semiconductor (215)
- Infineon Technologies (158)
- Intel Corporation (119)
- Micron Technology, Inc. (116)
- GlobalFoundries (102)
- Maxim Integrated Products, Inc. (84)
- NXP Semiconductors (84)
- Microchip Technology (71)
- Carsem M Sdn Bhd (70)
- Skyworks Solutions, Inc. (65)
- STMicroelectronics (65)
- Western Digital Corporation Inc. (64)
- Seagate Technology (55)
- Texas Instruments (53)
- X-FAB Inc. (50)
- Analog Devices (47)
- Qualcomm (39)
- Cree / Wolfspeed (38)
- Tower Semiconductor (32)
- Honeywell (30)

Top 3 subscribing universities:

- Ecole des Mines de Saint-Etienne (EMSE) (9)
- Arizona State University (8)
- Virginia Tech (7)

New companies and universities this month:

- ABCsWorld Consulting
- Abriso Jiffy
- Banneker Partners
- Confoe
- Cornel Reed AB
- Critical Manufacturing
- EMCORE
- Ericsson
- Flex Logix Technologies
- PCB Technologies Ltd.
- Rizzoli Consulting
- RTX Collins Aerospace
- University of Applied Science Dresden

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, or use the Contact form.

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.

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 is shown below.