FabTime Newsletter

Volume 24, No. 2 **April 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, April 25, 2023 – Vol. 24, No. 2

Contributors: Jason Sachs (EmbeddedRelated.com); Jay Maguire (Renesas); Jamie Potter (Flexciton)

Keywords: Bottlenecks; Tool Availability; Variability; Time Constrained Processing; Metrics and Goals

Table of Contents

Welcome

Welcome to Volume 24, Number 2 of the FabTime Cycle Time Management Newsletter. In this issue, we have an exciting staffing update about the promotion of Lara Nichols to become FabTime's President. We also have an announcement about our recent collaborative case study with Flexciton at the Fab Owners Alliance meeting, as well as the usual updates from Jennifer's LinkedIn. Our software tip of the month includes two ways of measuring the duration of downtime events, following up on a tip from the last issue.

We have a plethora of subscriber discussion in this issue, including calculating theoretical cycle time, X-Factor, and the value of a day's worth of cycle time, as well as managing bottlenecks. Our final subscriber question, about managing a process sequence with time constraints between steps inspired us to make time constrained processing the topic of this issue's main article. Jennifer studied time constrained systems for her PhD research and has been remiss in not delving into this complex topic sooner. She discusses capacity planning methods for time constrained systems, then moves on to operational issues, and closes with a few recommendations for coping with time constraints in practice. As always, we welcome your feedback.

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

Community News/Announcements

FabTime Staffing Update

We are excited to announce the promotion of Lara Nichols to President at FabTime Inc. Lara has been part of the FabTime family for more than 15 years, starting as an Engineer and most recently serving as Vice President of Engineering. Lara has a Master of Science in Computer Science from Cal Poly, San Luis Obispo. She has been instrumental in growing and developing the FabTime team. We are looking forward to Lara's leadership as FabTime moves forward in providing the best possible support for our customers.

FabTime's co-founders, Frank Chance and Jennifer Robinson, remain directors and full-time FabTime contributors. Frank is delighted that he will achieve a long-sought goal of focusing his time on enhancing FabTime's software and maximizing its value to customers in his new role as Software Architect. Jennifer will remain FabTime's Chief Operating Officer and will continue to focus on FabTime's newsletter, sales, and cycle time improvement efforts.

FabTime / Flexciton Joint Case Study at FOA

In February, FabTime's Lara Nichols and Flexciton's Jamie Potter presented a joint case study at the Fab Owners Alliance Collaborative Forum in Phoenix. The topic was **Intelligent Production Scheduling at Renesas Wafer Fab with Seamless Data Integration.** The Renesas sponsor for the project was Jay Maguire.

Renesas was already using FabTime's reporting dashboard in their fab in Palm Bay, Florida. The management team wished to conduct trials of Flexciton's intelligent scheduler to improve performance in their diffusion area. Renesas resources to support the project were limited. FabTime and Flexciton worked together to provide the necessary data needed for the scheduler via pull from FabTime's (already in place) on-site database. Key benefits of the joint FabTime/Flexciton solution included:

- IE resource-saving: The project did not require extra manhours from Renesas personnel.
- Time-saving: Flexciton's scheduling optimizer was deployed quickly using data from the FabTime database. Most required data was already there, while some was added for the project.
- Money-saving: Renesas did not have to purchase any additional hardware, because the Flexciton scheduler runs in a cloud environment, or additional SQL Server licenses, because data was stored on the FabTime database server.
- MES performance protection: There was no need to set up a second data pull from the fab MES.

Live trials of the Flexciton scheduler will be continuing at Renesas. Early results show significant reductions in time-link violations, number of batches run, and queue time for batch tools.

FabTime and Flexciton look forward to conducting more case studies in the future. For more information about the case study, or (for FabTime software customers) to set up a Flexciton trial for your fab, please contact Lara Nichols from FabTime or George Kopanias from Flexciton.

Another New Platform for Sending the Newsletter Issues

Last month FabTime Cycle Time Tip #4 was sent using an email automation platform called MailerLite. Some of our subscribers experienced issues receiving the cycle time tip through the new MailerLite system, so we will be sending the newsletter using Outlook (our old method) to those subscribers. If the formatting of the email you received with this newsletter issue appears different from past newsletters, you successfully received the newsletter using the new MailerLite system. We very much appreciate those of you who wrote to us to confirm receipt, or indicate non-receipt, of the previous tip. Thank you for your patience!

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 WSJ piece about the various strings that the US government is attaching to CHIPS Act funding. "The Commerce Department said it would impose requirements to help ensure that billions of dollars in taxpayer funding is spent wisely and that the funding will meet national security goals to counter the technology advances by China… Some of the terms also reflected the administration's social and economic priorities, such as diverse workforce and the use of union labor." [LinkedIn Post.] See also this more recent piece about specific concerns expressed by TSMC, Samsung, and SK hynix. [LinkedIn Post.]
- Another WSJ piece that illustrates the different semiconductor demand situations faced by different market segments. "Growing sales of electric vehicles—which tend to use more semiconductors than their gas-powered counterparts—coupled with greater automation of all vehicles, have kept producers of chips for cars (including subscribing companies like NXP Semiconductors, Analog Devices and Renesas Electronics) busy." [LinkedIn Post.]
- A KSN.com news story that illustrates the way fab construction plans are migrating to new locations. Did you hear about EMP Shield INC's plans to build a $1.9 billion wafer fab in Burlington, Kansas? "The governor says that the facility will create more than 1,200 jobs averaging $66,000 annually." Hopefully they can find enough people to fill those jobs. [LinkedIn Post.]
- An obituary for Gordon Moore published in the San Jose Mercury News. Reading it was like reading a mini history of the semiconductor industry. Talk about a lasting impact! [LinkedIn Post.]
- A new US Semiconductor Ecosystem Map from the Semiconductor Industry Association. You can filter by existing/announced, facility activity, and industry segment (foundry, IDM, equipment, etc.). I haven't spent much time with it, but it seems like a nice resource. [LinkedIn Post.]
- A piece in the WSJ that discusses the issue that "Semiconductor companies seeking federal grants under the Chips Act could face a tough decision: take Washington's help to expand in the U.S., or preserve their ability to expand in China." Samsung Electronics, TSMC, and SK hynix, which already have extensive investments in China, could be particularly affected. [LinkedIn Post.]

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

Measure the Duration of Downtime Events

In the February issue of FabTime's cycle time management newsletter, we recommended that to improve cycle time, fabs should focus on reducing the total duration of unscheduled downtime events instead of on increasing mean time between failures. The idea here is that it is long periods of downtime of key tools that cause WIP to pile up and cycle time to increase. We had a follow-up discussion internally at FabTime about how our customers could use FabTime to identify (and hence work to improve) the tools experiencing the longest downtimes. Here are two ways to do this.

1. Use the Tool Downtime Duration CV Pareto Chart.

- From the FabTime Search box, generate the Tool Downtime Duration CV Pareto chart.
- Change the "Slice" drop-down to "ToolGroup", change the date range to the past four weeks or month, and apply a filter to either select all tools or all the tools in your chosen area.

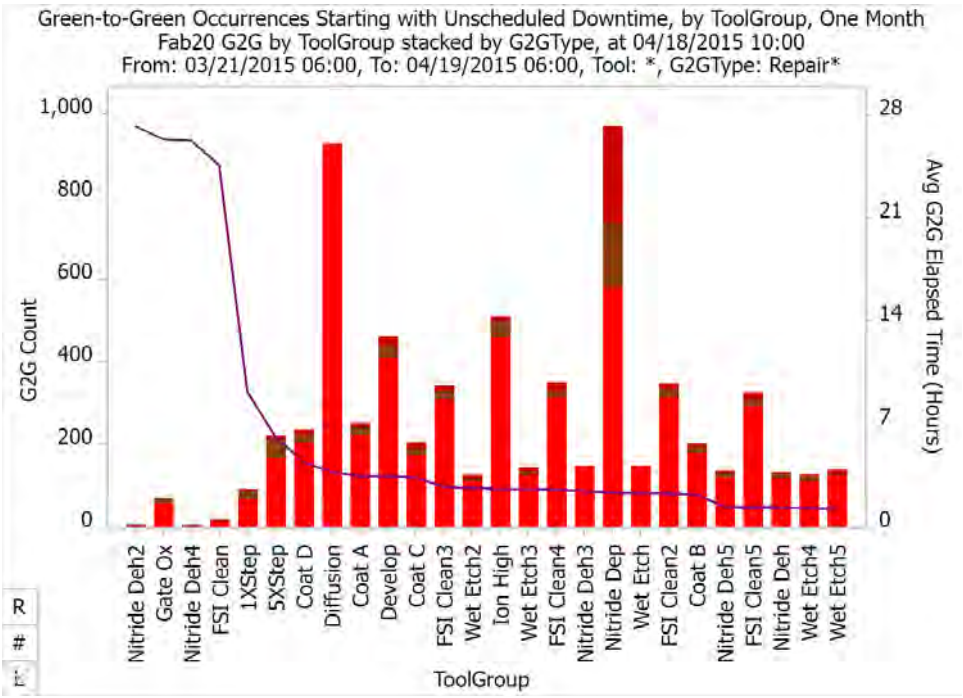
- The lines on the chart will show the average coefficient of variation for both unscheduled downtime events (the red line) and scheduled downtime events (the yellow line) for each tool group. The data for average downtime duration, however, will be in the AG-grid data table displayed with the chart.
    - Look for the column labeled "Unsched Down Duration (Hours)".
    - Click twice in the header for that column to sort the table in descending order by unscheduled downtime duration.
    - Click the three horizontal lines in the header and select "Pin Column | Pin Left" to move the column to the left-hand side of the data table, next to list of tool group names.
    - (Optional) Use the Edit Chart functionality to add bars to the chart showing the average scheduled and unscheduled downtime duration.
- Look for tool groups that have especially long unscheduled downtime durations. These are likely tools that also have high cycle times per visit, especially if they are one-of-a-kind tools. In the example below, the top four tool groups all experienced unscheduled downtime events that averaged more than 24 hours over the past month. These tools all warrant downtime improvement efforts (cross-training technicians, purchasing spare parts, etc.).

ToolGroup	Unsched Down Duration (Hours)	Unsched Down Duration CV	Unsched Down Count
Nitride Deh2 Trend Slice...	26.59	0.41	8
Nitride Deh4 Trend Slice...	25.88	0.29	6
Gate Ox Trend Slice...	24.83	0.4	75
FSI Clean Trend Slice...	24.43	0.43	19
1XStep Trend Slice...	7.86	1.72	115
5XStep Trend Slice...	4.13	0.97	334

2. Use the Green-to-Green Chart.

- From the FabTime Search box, generate the Tool Green-to-Green (G2G) Stacked Pareto chart. Green-to-Green (G2G) time is the elapsed time from when a tool goes down (to unavailable status for scheduled or unscheduled maintenance) to when it comes back up again (available status). It's called "Green-to-Green" time because it measures the elapsed time between two good states (with green color indicating as good). The main concept of this metric is to be able to see (visually) the elapsed time between two available slots and know by the G2G types what happened and whether the unavailable time involved scheduled or unscheduled maintenance work.
- As above, change the "Slice" drop-down to "ToolGroup", change the date range to the past four weeks or month, and apply a filter to either select all tools or all the tools in your chosen area.
- In the "G2GType" filter, type "Repair*" (be sure to include the *). This will generate data for all green-to-green events that began with an unscheduled downtime but may have also had some period of scheduled downtime included. It will also group together multiple downtime events that occur consecutively (e.g. "waiting for parts", "waiting for tech", "performing the repair."
    - (Optional) Sort the chart by "AvgG2GElapsedTime" in descending order, using the sort controls located just below the big set of filters to the left of the chart.

- The bars on this chart indicate the number of G2G events, while the line (against the right-hand y-axis) indicates the average G2G duration. In the chart below, we see that the same four tool groups are outliers as the four shown in the data table above, with average downtime duration exceeding 24 hours. This chart also makes clear that there were fewer downtime events for these tools (fortunately!). However, the long unscheduled downtime periods probably caused considerable pain in the fab.



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

Value of a Day's Worth of Cycle Time

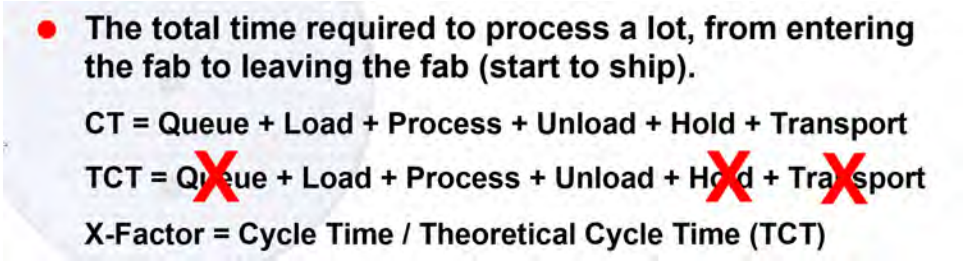
A **connection on LinkedIn** asked recently: "Do you have a calculation for the value of a day's worth of cycle time?"

FabTime Response: We do have a spreadsheet tool that we developed many years ago to estimate this. It's obviously quite dependent on your inputs. The spreadsheet is available for download from our website. Use of the spreadsheet was described in Issue 3.05, available for download by newsletter subscribers from the FabTime newsletter archive (password: FabTimeCommunity).

Calculation of Theoretical Cycle Time

A participant from one of our cycle time classes wrote to ask: "I was looking at x-factor benchmarking and trying to understand how different fabs calculate theoretical cycle time. There are some fabs that include handling and transportation time and some fabs don't. Do you have any useful data on this? What's accepted in the industry? Shall we use median or minimum for a specific population?"

FabTime Response: What we have on this in our course is: Cycle Time is:



Our view is that you should include time that's necessary to process the wafer, including load and unload times (because you need to load and unload to process the wafers). You might include a lower load time for tools with chaining, but usually there would still be something to include. For transport time we recommend either excluding it or including the bare minimum time (what it would take to transport a hand carry hot lot between steps, for example). Many fabs don't track transport time – the transport time just is lumped in with the queue time for the next step. If you have good transport time data, you could include a minimum amount.

Regarding process time, it's common for people to use the actual process time, but of course the correct thing to include here is the theoretical process time. People sometimes use actuals because that's easier and because the difference between actual and theoretical process time is usually small compared to the queue time.

We don't think there is much standardization across the industry on this, though people probably have company standards. The SEMI E79 standard includes:

> "5.2.46 *theoretical production time per unit (THT)* — the minimum rate of time per unit to complete processing, given the specified recipe, equipment system design, continuous operation, and no efficiency losses.
>
> 5.3.7 *optimized-recipe theoretical production time per unit (ORTHT)* — the theoretical production time per unit required to process a given recipe assuming the recipe specification is optimized for minimum theoretical production time. ORTHT is based on minimum durations for the objective processing steps (e.g., implant time for ion implanters) plus minimum allowances for any additional supporting process steps (e.g., heating, cooling, gas stabilization) that are deemed absolutely necessary. ORTHT shall be defined to be less than or equal to the corresponding theoretical production time per unit (THT) used in calculating OEE."

That is, for each recipe, you determine the theoretical production time based on the optimal UPH rate for that tool/recipe.

In practice, we think that most people do use some kind of historical data based on actual process times. It's simplest to just use the average, but we can see why it would make sense to use the minimum. The problem with using the minimum in practice is that there may be logging errors, where people log move in and move out at the same time, such that the recorded process time is zero. So, you'd want to have some method for excluding values below some threshold close to zero.

The idea is to come up with a realistic best case cycle time that's what it would take to process a lot under normal conditions, if that lot never had to wait in queue, and incurred only minimal transport times. For new flows, it may be easiest to do this using UPH rates for tools, while for more established flows where there is historical data, it makes sense to use that data.

Do any other subscribers have any comments on this?

Measurement of X-Factor

We also had a question from **another anonymous subscriber** about how fabs measure x-factor.

FabTime Response: Newsletter Issue 9.04 compared calculations for x-factor with those for dynamic x-factor. Here's how we calculate x-factor in our software for shipped lots:

FabTime Help

GUIDE HELP NEWSLETTERS VIDEOS TIPS SERVICES EXIT Search FabTime

SHIPPED LOT X-FACTOR LIST CHART

This chart displays a list of Shipped Lots and their calculated X-Factor. Shipped Lot X-Factor is factory cycle time divided by process time. This article documents the calculations, data filters and chart series for this chart, and provides general data table tips.

CALCULATIONS

Shipped Lot X-Factor is factory cycle time divided by process time. Factory cycle time is elapsed time from lot start to lot ship. If there is no start transaction for a lot, factory cycle time is elapsed time from first transaction to lot ship. For process time, FabTime uses values set by the site at lot-by-lot level (if available). If lot-by-lot values are not set by a site, FabTime uses the sum of planned process times from flow/step data (if available) for non-rework moves. If planned process time data is not available, FabTime estimated process time based on the sum of actual process times for non-rework moves.

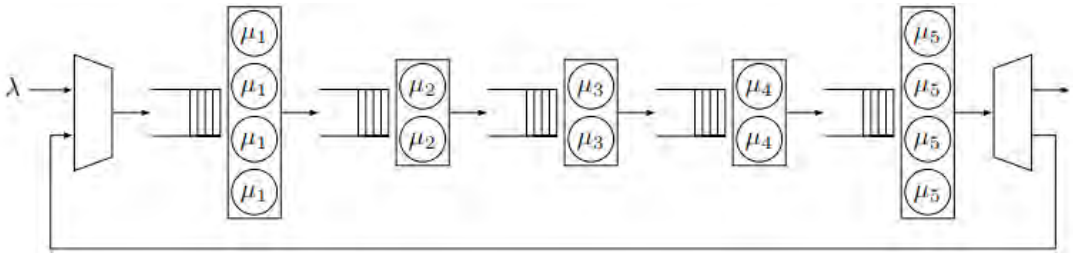
As you can see, it's not completely straight forward in practice to take cycle time / theoretical cycle time. That theoretical cycle time could be planned or just a sum of process times.

Do other subscribers have any comments on the calculation of x-factor for shipped lots?

Managing Bottlenecks

EmbeddedRelated.com blogger Jason Sachs sent in a question about how fabs manage bottlenecks. He said: "I'm somewhat familiar with the Theory of Constraints (TOC) philosophy as it pertains to ensuring there is one bottleneck and managing that bottleneck. My question is how much does having only one bottleneck matter?

Here is a sample system I have tried analyzing through simulation:



I've modeled a production line with:

- Five tool groups (1-5); group 1 and 5 have four tools in parallel, and the rest have two tools in parallel.
- Arrival rate λ is exponentially distributed.
- Processing times are exponentially distributed, parameterized by two processing rates and some perturbations.
  - $\mu_1 = (\mu_B / 4) * (1 + \delta_1)$
  - $\mu_2 = (\mu_A / 2) * (1 + \delta_2)$
  - $\mu_3 = (\mu_B / 2) * (1 + \delta_3)$
  - $\mu_4 = (\mu_B / 2) * (1 + \delta_4)$
  - $\mu_5 = (\mu_B / 4) * (1 + \delta_5)$
- The line is re-entrant, requiring N=10 cycles.

The intent is for tool group 2 to be the bottleneck with $\mu_A = 1$ per second, and all other tool groups to be run at some other rate $\mu_B \geq \mu_A$, and make small perturbations in each tool group's rate to see what the sensitivity is to overall cycle time.

I ran a bunch of simulations at various values of λ near full utilization (λ = 0.08 (utilization ρ = 0.8), 0.085, 0.09, 0.095, 0.097) and have reached the following preliminary conclusions:

- If $\mu_B = \mu_A$, then sensitivity is roughly equal among all tool groups: varying processing rates by 1-2% affects the cycle time about the same.
- If $\mu_B > \mu_A$, then sensitivity of the non-bottleneck groups drops fairly quickly, depending on utilization; at $\lambda = 0.08$ the drop-off is moderate: if $\mu_B = 1.1*\mu_A$ then the sensitivity of non-bottleneck groups is about half of the bottleneck; at higher utilization the drop-off is much steeper -- $\lambda = 0.095$ if $\mu_b = 1.05*\mu_A$ then the sensitivity of non-bottleneck groups is about 1/4 of the bottleneck.

This seems to confirm the idea of managing the bottleneck.

But does it really matter if the bottleneck bounces around? How much human effort in a fab is required to "manage a bottleneck," and is it that much easier/less risky/less expensive to have only 1 known bottleneck vs. having more than one?

I realize that simulations are only a rough approximation to reality, so I would appreciate any real-world insight that you or your subscribers may be able to provide. I would especially like to better understand what it means for a bottleneck to be well-managed, compared to other tools in the fab, and what kind of levers fab managers are adjusting on a day-to-day basis to manage bottlenecks. How much extra effort or cost is required to manage a bottleneck, compared to non-bottleneck tools? What would make it easier or more beneficial to have one tool group that is known to be the dominant capacity bottleneck?

FabTime Response: This seems to us to be one of the many cases in which the theory offers useful lessons but requires some flexibility to implement in the complex environment of a fab. It can be easier to run a manufacturing system that has a single, unchanging bottleneck, because you focus your day-to-day attention on running that tool efficiently, including by making dispatch decisions at other tools to keep the bottleneck from starving.

The TOC advice to subordinate everything else to the bottleneck can help you to maximize the throughput from the factory. The goal is to ensure that the bottleneck is never idle and minimize any wasted time on the bottleneck. As one example, if you keep a relatively large buffer of WIP in front of the bottleneck, you can minimize setups (though at a potential cost of cycle time for low volume recipes). You can also focus availability improvement programs on the bottleneck, and work to ensure that you don't have any forced idle time due to lack of operator. OEE is a good metric for bottleneck tools, because it focuses on getting as many wafers as possible through the tool, with loss factors to dig into to identify targets for further improvement.

However, it's been our experience that many fabs don't have a single bottleneck, but rather have several tool groups that become the bottleneck at different times. The bottleneck can shift due to product mix changes, engineering requirements, and/or tool availability issues. Of course, the bottleneck can also change if new capacity is added at the previous bottleneck.

Further complicating this question is the fact that there are different types of bottlenecks. In the subscriber discussion forum of the previous issue of the newsletter, Issue 24.01, we discussed several different types and ways to identify them. These included:

- Traditional capacity bottlenecks
- Cycle time bottlenecks
- Short-term bottlenecks
- Future bottlenecks

At any given time, there should be a single tool group that is the top bottleneck (the constraint) in a long-term capacity planning sense (though there might well be one or more other tool groups that are near-bottlenecks). This is the tool group with the highest planned utilization. But every time the product mix changes, this designation might change. And every time a near-bottleneck tool goes down, that tool group could become the bottleneck on a (hopefully) short-term basis.

One other important point is that because there are so many tools in the fab, and because process flows are both long and reentrant, the performance of non-bottleneck tools does matter for cycle time improvement. The capacity planning bottlenecks limits fab throughput, but cycle time is accumulated across many tool groups. See Issue 10.09 for more details about this. What this means in practice is that even if you had a clearly defined single bottleneck tool group in a wafer fab, focusing all your efforts on that tool group would not be enough to ensure great cycle time through the fab.

Bottom line: it would be somewhat easier to manage a fab with a single, known bottleneck. But this is not the world that most people who run fabs live in. Do other subscribers have anything to add here?

Managing Lightly Loaded Single Thread Tool/Process Sequences with Time Constraints

An **anonymous** subscriber wrote asking for suggestions for managing lightly loaded single thread tool/process sequences to get reasonable cycle times. He wrote:

"We have a pilot line for a new technology that is completely single threaded. There are five re-entrant loops, each of which includes three different queue time limits between process steps. The flow looks like this:

- A
- B
- C
- B
- C
- D – 12 hour queue time to E
- E – 12 hour queue time to F
- F – 24 hour queue time to A
- A

Currently we are going through optimization and product qualifications. Based on our current low run rate, our traditional static tool capacity algorithms show that we only need a max of ~15-20% tool utilization to run the necessary wafers. Traditional tool availability tabulations are meaningless, as the tools are idle most of the time. It only matters if the tool is up when the lot gets there, and of course the delivery rate is highly variable. This is complicated by the queue time limits. In a couple of places, if the next tool is down, you can't run the existing tool due to the queue time limits. Hence the cycle time is higher than we would like (especially given the low tool utilizations).

Two of the tools subject to the queue time limits (E and F) have been particularly problematic for "uptime when needed." The brute force method at our disposal is to task the maintenance people with "must be up 100%," but that's not very functionally helpful. Do you have any articles or items in your toolbox that might help us?"

FabTime Response: That does sound like quite a tricky situation. We wrote last year about managing one-of-a-kind tools (see Issue 23.05). You might find some ideas there. But it sounds like the real problem here is the combination of downtime and queue time limits. Managing time limits between steps is something we haven't written about in the newsletter (until today – see below). In our dispatch module we have a reservation system, such that when you start the lot on one tool, it reserves a place in the dispatch list at the next tool, to avoid missing the time constraint. But that isn't helpful if the next tool is down.

Rather than focusing on trying for 100% uptime on the tools, what we think you should focus on is minimizing the duration of the unavailable times. When the tool is down, whether for scheduled or unscheduled downtime, the most important thing in your situation is to get it back up quickly, so that the total unavailable time is short. See Issues 22.01 and 20.02.

We also wonder if there is something you could do with alerting of the maintenance team. When you start a lot at tool D, alert the maintenance team that tool E will need to be up and running within 12 hours + process time. Similarly, when you start a lot at tool E, you notify the maintenance team that tool F will need to be up and running within 12 hours + process time.

Do other subscribers have any suggestions to add here? Has anyone tried alerts in this situation?

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

Main Article: Managing Time Constraints between Process Steps in Wafer Fabs

Introduction

Back in the late 1990s, Jennifer did her PhD dissertation on capacity planning for semiconductor fabs with time constraints between process steps. Her research used queueing models, simulation, and (for the most complex systems) fluid models. She published a paper documenting some of the results in the 1999 Winter Simulation Conference Proceedings (available for download from FabTime's website). Since then, we haven't referred to that work very often, but we've continued to learn about the complexities and realities of fabs.

Recently, two different newsletter subscribers have written to ask for help managing systems with time limits between process steps (see the Subscriber Discussion Forum above, and the one in Issue 23.03 from last year). Others have asked about this over the years, and we decided that it was time to address this complexity of wafer fabrication in a new article.

The Problem

In wafer fabrication, it is not uncommon to have time constraints between process steps. That is, a lot must start processing at a later operation within some time window after completing processing at an earlier operation. A common example is a bake step that must be completed within some time window of the prior clean operation. Such constraints are also called queue time limits, time bound sequences, and time links. They are typically put in place by process engineers to improve yields in the fab. If the time constraint is violated, the lot must go back to the first operation in the sequence for reprocessing. When this happens, capacity is lost at tools running the first operation (increasing cycle time for all lots that use that tool), and cycle time of the reprocessed lot is increased by the reprocessing time. There is also increased variability in the arrival rate to the later operation.

There can be intervening steps between the initial and final operation of a time constraint loop. These multi-step systems are particularly difficult to manage. Even a system involving two steps holds considerable complexity.

To accurately plan capacity for tools subject to time constraints, we need a way to estimate the percentage of lots that will be reprocessed. Operationally, fabs need to make dispatch decisions that minimize the chance of the time constraint being violated. In this article, we will discuss capacity planning methods, then move on to operational issues, and close with a few recommendations for coping with time constraints in practice.

Capacity Planning with Time Constraints

The capacity of a system is the maximum feasible arrival rate of work to the system, or, equivalently, the maximum achievable throughput rate of the system. The behavior of a time constrained system with more than two operations is difficult to predict except at very low tool utilizations. In the low utilization case, lots flow through with few delays, and are rarely sent back for reprocessing. At higher arrival rates, or for highly

variable systems, time constrained systems can rapidly become unstable. Once a few lots are delayed enough to be sent back for reprocessing, these lots increase the arrival rate to the earlier tools. This in turn increases queuing delays and makes it more likely that other lots will be sent back. A "vicious cycle" ensues, making predicting system capacity difficult. Time constraint loops with intervening steps are difficult to predict and difficult to operate. We recommend avoiding them as much as possible. For the remainder of this article, we will focus on time constraints between two operations.

Understanding the capacity of a time-constrained system, even with only two operations, requires understanding the distribution of lot cycle times (to understand the probability of any given lot being sent back). This distributional data would not typically be included in spreadsheet models. These models usually include, at best, static data such as average cycle times. Even queuing models rarely capture the entire distribution. Therefore, to fully understand the behavior of a time-constrained system, capacity planners would typically need to use simulation.

In her earlier research, Jennifer developed a simple approximation for the probability of lots being reprocessed based on queuing formulas for time constrained systems involving two operations. She compared results from the approximation with results from a discrete event simulation for various system parameters. The approximation performed well in predicting the probability of reprocessing for highly variable systems. It provided an upper bound that could be included in spreadsheet capacity models. In the interest of space and complexity minimization, we will send those of you who are interested in the details of the approximation to the WinterSim paper.

Here are two final notes on the capacity planning research:

1. The primary issue addressed here is the capacity impact on the first tool in the time constraint link. The loading of the final tool doesn't change due to reprocessing (because lots are reprocessed only at the earlier operation), though the arrival distribution to the downstream tool might change.
2. In her research, Jennifer did not capture the impact of downtime. While this was a reasonable and perhaps necessary omission in a research context, equipment downtime in practice has a huge impact on the operation of time constraint sequences.

Operating Practices and Questions

It's all well and good to have a model for predicting the average percentage of lots that will need to be reprocessed due to time constraint violations. In practice, what people running a fab want to do is ensure that lots are sent back for reprocessing as rarely as possible. Reprocessing adds to lot cycle time, uses extra consumables, and takes up extra capacity.

What we commonly see is fabs holding lots at the first operation in a time constraint loop, and only processing at that operation when they have reasonable confidence that the lot will make it all the way through without violating the time constraint. Policies for starting a lot on the upstream tool might look something like this:

- Don't start lots at the upstream operation when the tool needed for the downstream operation is unavailable; and
- Don't start lots at the upstream operation when the WIP in front of the downstream operation is too high.

But of course, fabs being fabs, even these relatively straightforward policies become complex the closer we look. Questions must be addressed, such as:

- How high is "too high" for WIP at the downstream tool? How can we tell?
- What dispatch rule is followed at the downstream tool? In the reentrant environment of the fab, how do we prioritize lots within time constraint loops at the downstream step over other lots that

might be there for other steps? What happens if a hot lot comes through? Are we prepared to violate the time constraint for other lots to get a hand-carry lot through?

    - In our dispatch module we have a reservation system, such that a spot can be reserved on the dispatch list for the downstream tool when the lot is processed on the upstream tool. But, of course, the reservation is no guarantee that the lot will arrive in time.

- What if there are multiple downstream tools that could be used, and only one of them is down?
- What if the downstream tool is unavailable because of a PM, where we know when the PM is scheduled to finish? Should we count on that? Do we have any information to tell us when a tool in unscheduled downtime might come back up?
- What if the downstream tool is a batch tool? How hard should we try to get enough lots through the upstream tool to form a batch?

It matters what the queue time limit is relative to the downstream operation's process time, too, because that impacts how much of a queue we'll allow to accumulate. If the time constraint is 24 hours, and the downstream step process time is 1 hour, we can afford to let multiple lots through. We would normally want to do that to avoid starving the downstream tool, especially if the upstream tool has reliability issues. However, in this case we wouldn't want to let more than 24 lots wait at one time (and probably fewer because the downstream tool could go into unscheduled downtime in the next 24 hours).

Recommendations for Minimizing Time Limit Expiration

Here are a few suggestions for minimizing the possibility of lots violating their time constraints.

- In the subscriber discussion question about time constraints above, the fab attempted to require 100% uptime on the critical tools but acknowledged that such a requirement was not "functionally helpful." Our suggestion is instead to focus maintenance teams on **reducing the duration of any repair or PM time** at the downstream tool. It's not total availability that matters here, but minimizing the chance of the tool not being available when the time constrained lot arrives. This ties back to one of our general recommendations for improving fab cycle time, and this month's software tip above.
- Managing time constraint loops is also an excellent case for **using alerts**. In our software, alerts are set by individuals based on lot or tool status and are sent as short emails or text messages. We recommend setting an alert for any lots at the downstream step that are nearing their time constraint, so that these don't slip through the cracks.
- Communication, via alerts or some other method, should also be used to ensure that:
    - Once a lot has started in a time constraint loop, no new PMs are scheduled; and
    - Once a PM is scheduled, at least a longer PM, no new lots are started into the constraint loop.
- **Qualifying additional tools for the downstream step** in a time constraint link will also, naturally, reduce the probability of violating the time limit. The probability of two tools going down at the same time is much smaller than the probability of just one tool going down. As with everything else in the fab, managing time constraints is hardest in the case of one-of-a-kind tools or single path operations.
- Consider **using the WIP Hours metric**. WIP Hours (see Issue 20.03) measures the estimated hours of required processing time for WIP in queue at a tool. If you track WIP Hours (as we do in our software), you could set an alert for when WIP Hours at the downstream tool exceeds some value X (close to but not equal to the time constraint, perhaps considering tool availability in some

way). You could then stop releasing WIP from the upstream tool, and re-start later based on some other trigger (WIP Hours went below value Y).

Conclusions

Time constrained processing is a complex topic, but one that many fabs manage on a day-to-day basis. This article has merely scratched the surface. We have defined the problem, introduced a method for including the probability of reprocessing in capacity planning spreadsheets, and discussed some operational issues and questions arising from time constraints. We have shared a few ideas for minimizing the probability of time constraint violation in practice. We are particularly interested in the question of whether the relatively new metric WIP Hours could be used to help set triggers for starting and stopping the release of lots into the time constraint loop. If any readers have experience in doing this, we would love to hear your feedback.

Closing Question for Newsletter Subscribers

How does your fab manage time constraints between process steps? Do you use alerts or a measure of WIP Hours? Do you have any suggestions for other readers?

Further Reading

- J. K. Robinson and R. Giglio, "Capacity Planning for Semiconductor Wafer Fabrication with Time Constraints between Operations." In *Proceedings of the 1999 Winter Simulation Conference*, ed. P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, 880-887. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 1999. Download PDF (58 KB)
- Several past FabTime newsletters are also referenced in this article. They are all available to subscribers for download from the FabTime Newsletter Archive. The current password is FabTimeCommunity.

Subscriber List

Total number of subscribers: 2795

Top 20 subscribing companies:

- Onsemi (138)
- Intel (137)
- Infineon (133)
- Micron Technology (124)
- Analog Devices (118)
- Microchip Technology (98)
- Globalfoundries (87)
- NXP (77)
- STMicroelectronics 69
- Skyworks Solutions (66)
- Texas Instruments (65)
- Western Digital Technologies (57)
- Seagate Technology (54)
- X-FAB (47)
- Carsem M Sdn Bhd (44)
- Wolfspeed (42)
- Qualcomm (38)
- Tower Semiconductor (37)
- Applied Materials (33)
- ASML (32)

Top 3 subscribing universities:

- Ecole des Mines de Saint-Etienne (EMSE) (7)
- Arizona State University (6)
- Ben Gurion University of the Negev (5)

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.