FabTime Newsletter

Volume 21, No. 5 September/October 2020

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: John Paul Gauci (JPG Consulting), Thomas Quarg (AMTC), David Carmichael (Tower Semiconductor Ltd.), Russell Barton (Penn State University), Hani Ofeck (Tower Semiconductor), and Justice Stiles (Infineon Technologies).

Date: Tuesday, September 22, 2020 – Vol. 21, No. 5

Keywords: Bottlenecks

Table of Contents

Welcome

Welcome to Volume 21, Number 5 of the FabTime Cycle Time Management Newsletter. We come to you this month from a smoky California, where FabTime's team is safe, well, and grateful for firefighters. We have no community announcements in this issue, but we do have a few industry news tidbits from Jennifer's LinkedIn feed. Our software tip of the month is about using the forecast arrivals charts in FabTime.

We received several detailed responses to last month's article about identifying short-term bottlenecks, including one that questioned the merits of looking at short-term bottlenecks in the first place. In our main article this month, we aggregate and respond to those contributions. We discuss potential extensions to the WIP Hours paradigm, additional methods for identifying short-term bottlenecks, and uses of short-term bottlenecks as indicator species to tease out underlying variability problems in the fab.

We welcome responses to the short-term bottlenecks discussion, as well as new questions for subscribers and/or suggestions for newsletter topics.

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 posts of relevance here have included:

- "This is good to see: "Headway Technologies (is expanding in Silicon Valley), providing a hopeful sign amid the coronavirus economic woes." Headway was the first customer for FabTime's software, a decision for which Frank and I will always be grateful. Glad to see them having good press. George Avalos, San Jose Mercury News, August 4, 2020.
- This seems plausible. "Communications giant Huawei Technologies Co. Ltd. is hurrying to build a wafer fab capable of running 45nm CMOS manufacturing process, according to unsubstantiated Chinese social network reports." Peter Clarke, EENews Analog, August 24, 2020.
- Good piece in today's Wall Street Journal about staying focused while working from home. I have personally found that noise-cancelling headphones and a white noise app are especially useful these days (when I'm not the only one working/learning from home). The Wall Street Journal, September 1, 2020.
- "Soaring pandemic-inspired demand for chips that power everything from communications and IT infrastructures to personal computing, gaming and healthcare electronics will drive an 8% increase in global fab equipment spending in 2020 and a 13% increase in 2021, SEMI announced today in its World Fab Forecast report. Rising demand for semiconductors for datacenter infrastructures and server storage along with the buildup of safety stock as U.S.-China trade tensions intensify are also contributing to this year's growth." SEMI, September 8, 2020.

For more industry news, connect with Jennifer on LinkedIn: http://www.linkedin.com/in/jenniferrobinsonfabtime

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

FabTime User Tip of the Month

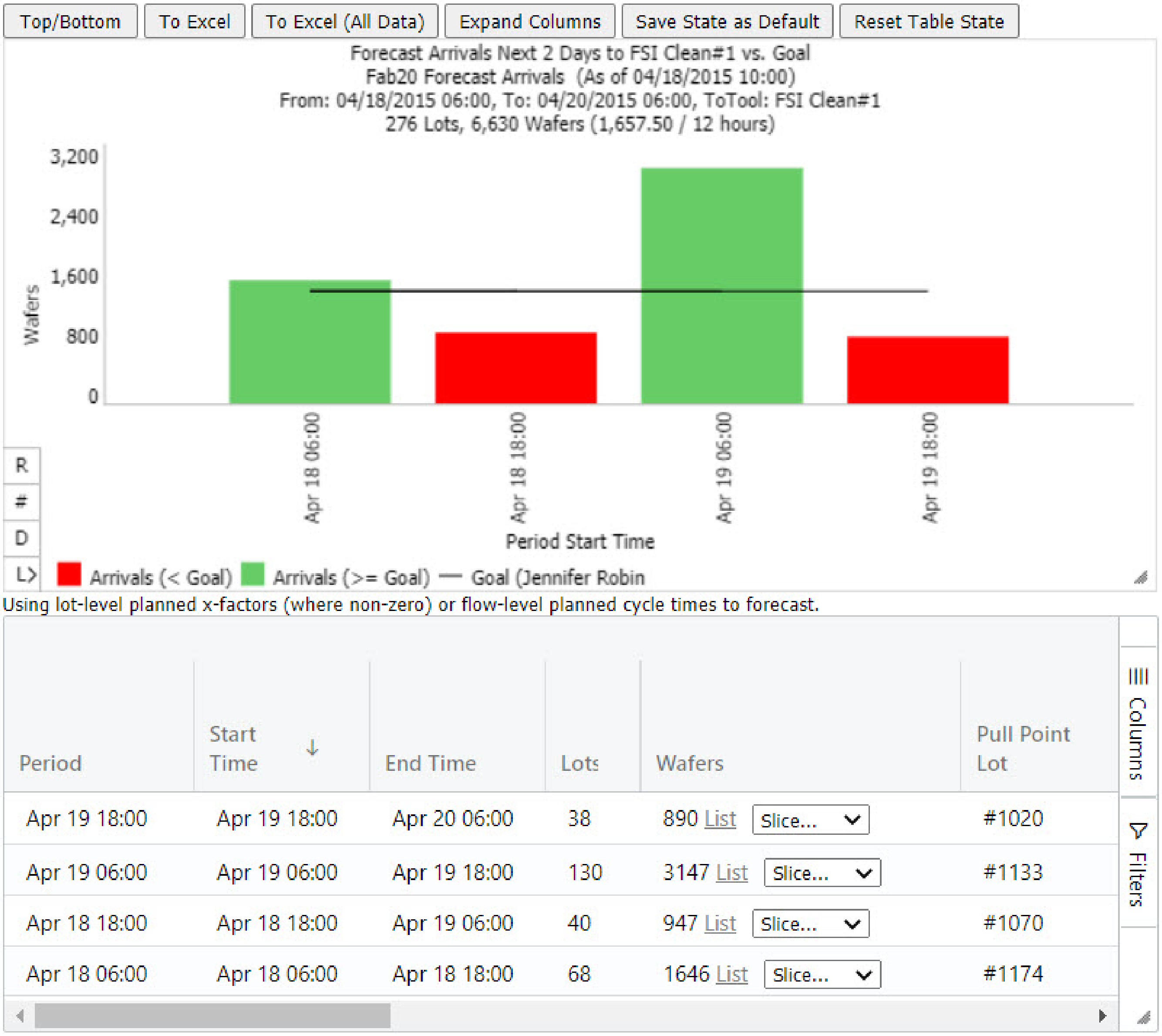
Forecast Arrivals to a Tool

There are a variety of ways to identify tools that are a current (or past) problem in the fab (WIP levels, inventory age, operation cycle times, WIP Hours, etc.). A FabTime chart that can help to predict when a tool or tool group is likely to become a problem over the next day or so is the Forecast Arrivals chart. This chart displays lots that are estimated to arrive at a target step, trended by time. To use this chart:

1. Expand "Forecast Charts" from the Charts page. Press the "Go" button next to "Forecast Arrivals Trend."
2. Modify the "From" date to the start of the current shift or some later date (e.g. the start of the next shift). Modify the "To" date to a time later in the future. Set the period length to 12 to see the expected arrivals by shift (or as preferred). You may optionally specify an as-of date in the "Date" filter. If you don't specify an as-of date, FabTime uses the latest time for the factory, which is normally what you would want to use (a forecast based on the most current information). If the as-of time falls within a period on the chart, FabTime shows the expected arrivals during the remainder of that period.
3. To see the forecast arrivals to a tool, enter the name of that tool in the ToTool filter. Use wildcards and/or comma separated lists to see the forecast arrivals to a group of tools. If the ToTool filter is specified but ToStep is left blank, then the target steps are all steps with one or more qualified tools

matching the ToTool filter. If ToStep is specified in addition to ToTool, then the target steps are all steps matching the ToStep filter, which also have one or more qualified tools matching the ToTool filter. (You can also specify just the ToStep filter. Step is used instead of operation because steps are unique in FabTime, but operation names can be repeated. Wildcards are allowed in specifying ToStep.)

4. Once you press "Go", FabTime will show the forecast arrivals during each time period on your chart. The forecast is based on projecting each lot forward from where it is at the as-of date and using planned cycle time values for the lot's next steps. These values are generated from your MES data link. See the chart help page for details. An example is shown below.


Top/Bottom
To Excel
To Excel (All Data)
Expand Columns
Save State as Default
Reset Table State
Forecast Arrivals Next 2 Days to FSI Clean#1 vs. Goal
Fab20 Forecast Arrivals (As of 04/18/2015 10:00)
From: 04/18/2015 06:00, To: 04/20/2015 06:00, ToTool: FSI Clean#1
276 Lots, 6,630 Wafers (1,657.50 / 12 hours)
Wafers
3,200
2,400
1,600
800
0
Apr 18 06:00
Apr 18 18:00
Apr 19 06:00
Apr 19 18:00
Period Start Time
R
#
D
L>
🟥 Arrivals (< Goal) 🟩 Arrivals (>= Goal) — Goal (Jennifer Robin
Columns
Filters

Using lot-level planned x-factors (where non-zero) or flow-level planned cycle times to forecast.

Period	Start Time ↓	End Time	Lots	Wafers	Pull Point Lot
Apr 19 18:00	Apr 19 18:00	Apr 20 06:00	38	890 List Slice...	#1020
Apr 19 06:00	Apr 19 06:00	Apr 19 18:00	130	3147 List Slice...	#1133
Apr 18 18:00	Apr 18 18:00	Apr 19 06:00	40	947 List Slice...	#1070
Apr 18 06:00	Apr 18 06:00	Apr 18 18:00	68	1646 List Slice...	#1174

5. You can also specify the XFactor filter (down towards the bottom of the main filter list) to experiment with what the arrivals would be if the XFactor for all lots was changed. In that case, the forecast cycle times will be computed as the planned theoretical times for the flow multiplied by this XFactor. This functionality could be useful if you wanted to, for example, estimate which hot lots might be expected to arrive to a tool. You could filter for priority and apply the hot lot XFactor.

6. Drill down from the data table via the "List" link in the Wafers column to view the list of lots expected. Use the Slice dropdowns to get to a Pareto version of this chart for a single time period.
7. Apply other filters as needed (e.g. to only see lots with a particular owner code or belonging to a particular customer).

The information displayed on the forecast charts is only as good as the planned cycle time data pulled from your MES and your tool qualification data for future steps. The charts do not use simulation or take tool downtime into account. They simply project each lot forward based on the planned cycle time at future steps. This means that the shorter the time window used to look forward, the more accurate the data will be. The list of lots that you see on the Forecast Arrivals List chart will probably not be a perfect prediction. However, the forecast arrival charts can give you an idea of places where a tool is likely to be swamped soon. 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

Identifying Short-Term Bottlenecks

John Paul Gauci from JPG Consulting wrote in response to last month's article about short-term bottlenecks, saying:

"I enjoyed your latest edition of FabTime. You talked about bottlenecks. We used to monitor load port utilization as means toward early detection of problems. Load port utilization is one of the (many) things that are tracked in fully automated fabs. Load port (LP) utilization is a way to identify when WIP is not flowing from operation to operation. This is more prevalent in fabs that run a high mix of part numbers. If some systems in a sector have high LP utilization while others don't, WIP may not be moving effectively. Turning off certain load ports on some systems can help drive the AHMS to load other tools that would be otherwise empty."

John also sent this link to a Winter Simulation Conference paper about AMHS metrics as used at TSMC:

> J. Tung, T. Sheen, M. Kao and C. H. Chen, "Optimization of AMHS Design for a Semiconductor Foundry Fab by Using Simulation Modeling," *Proceedings of the 2013 Winter Simulation Conference*, 2013. Available for download from the WinterSim archive.

Metrics used in the article to benchmark simulated performance of various AMHS configurations include tool load port service time, T2T (Tool-to-Tool) Ratio (quantity of direct transportation from tool to tool), FOUP cycle time, and FOUP total travel distance.

FabTime Response:
We haven't heard of using load port utilization as a technique for early detection of problems, but we think it sounds interesting. We have limited experience with fully automated fabs but wonder if other subscribers use this metric for short-term WIP management.

Thomas Quarg from AMTC also wrote in response to the main article from the last issue, saying:

"Good article about WIP Hours … years ago my team and I defined something (details were patent-protected) called "Dynamic Kanban". The point, as you described, was counting the WIP Hours in front of a tool group by main technology or node. This has a little more granularity than total WIP Hours.

The WIP Hours in front of a tool group is building a "Pillar in hours". That Pillar is either increasing or decreasing, depending on the performance of the tool and the amount of upcoming WIP.

That Pillar generates a so-called "WIP Shadow" in the direction of upcoming WIP. The WIP Shadow indicates whether the WIP Hours are increasing or decreasing right now at a tool group. Where they are already likely to trend down, we don't need to focus as much, relative to places where they are expected to trend even higher.

We use the Shadow (= WIP at the steps ahead of the step with a high Pillar), to identify steps where we are going to de-prioritize certain activities, so that we don't keep feeding a tool that has too much WIP already. We can instead focus on processing WIP that is needed to prevent downstream bottlenecks from starving. The higher the WIP Hours value, the farther back the Shadow falls. If the Shadow is, for example, so long (or high) that we would have enough to work for the next shift, why should we fill that bubble further?

We then check every hour how high the bubble (Pillar) is and change the activities in front of it. This is Dynamic Kanban. An additional point is that this will be automated in the dispatch system, with rules for things like making sure high priority material is still processed.

We used two "views". One view was "layer" based (because of the expensive Litho Tools, which are usually bottlenecks). This meant counting the WIP Hours in a complete layer and not feeding that layer if it already had more than, for example, a shift could absorb. The other view was "operation" based with operations summarized / aggregated in a "main technology" or product group.

FabTime Response:

We appreciate Thomas sharing details of this application of WIP Hours as part of a dynamic Kanban system. We like the visual imagery of the "WIP Shadow" cast by the high level of WIP Hours at a step. We also like these refinements to our approach:

1. Considering not just total WIP Hours, but whether WIP Hours are increasing or decreasing, with more emphasis on places where they are increasing.
2. Consideration not just of total WIP Hours, but of WIP Hours by layer. This approach encompasses not just keeping tools busy but keeping tools busy processing the right WIP.
3. Using WIP Hours as a component in a Kanban or WIP smoothing approach.
4. We do think that, as Thomas noted, such an approach would need to be highly (and carefully) automated to be effective in a fab. We will give some thought to whether and how these ideas might inform our own approach to identifying and managing short-term bottlenecks.

David Carmichael of Tower Semiconductor Ltd. also wrote in response to the last issue, saying:

"I have heard reports that Eliyahu Goldratt is spinning in his grave due to your activities.

The whole concept of WIP hours and temporary bottlenecks is reinforcing the seat-of-pants attitude to driving a FAB that has the potential to do a great deal of damage to factory throughput. Piles of WIP do not always indicate a problem except in the minds of those who do not understand Theory of Constraints.

Here is the Theory of Constraints (TOC) philosophy (not the TOC process) as I understand it, and I know it works because I have seen it in action in production.

Every factory has a set of true bottlenecks. They can vary when major changes in product mix occur and can be impacted greatly by equipment downtime or failure to qualify for a process, but they are there and we should know where they are, and more importantly what their capacity is. At any point in time we should be able to say:

a) This list of TRUE constraint tools must be kept busy. True constraint tools have zero catch-up capacity.
b) The lots that must run on them are known AND WILL ACTUALLY BE RUN ON THEM.

c) The capacity of these tools, for the mix of products they must handle is known as is the time they will take to perform processing.

The criticality of these constraints will vary as some will be at their capacity limit and some a little below it. TRUE constraints in a factory are few in number and change only rarely. There may be other considerations that impact criticality such as Customer but maybe we shouldn't admit to that. Obviously, software is needed to keep track of the constraints and their criticality as no human has any chance of doing so.

Every constraint must have a buffer of the desired work sitting right in front of it and it must not be too large and just enough to prevent the tool set running dry. If the buffer at the constraint is being depleted below a target level, then we must accelerate the correct amount of work towards it and know how much to accelerate the work to maintain the buffer.

If every critical or near-to-critical constraint is always kept busy, then the factory is operating at its maximum throughput and cannot do any more. Double the starts and the throughput will not change (it can get worse). Reducing starts without limiting the throughput of the constraints, and WIP goes down along with cycle time. Factory output remains unchanged. The only local optimization that is required is at the true constraints of the factory. All other tools are of lesser importance because they have catch-up capacity.

I would be interested to hear your thoughts."

FabTime Response:

We're sure that Eli Goldratt would be happy to see your defense of his position on bottlenecks. And we certainly agree with you that a fab cannot do anything to increase beyond its maximum throughput, as defined by the capacity of the bottleneck or bottleneck tools.

However, we do think that tools that are not bottlenecks in the capacity/theory of constraints sense can still have an impact, in at least two ways.

1. Non-bottlenecks still contribute to cycle time, both directly and via the introduction of variability. This was last discussed in issue 10.09. Here's what we said in the conclusion of that article:

   > "Our point is very simple: actions that you take to improve cycle time at non-bottleneck tools generally will improve overall product cycle times. For operations located before the first visit to the bottleneck, or after the last visit to the bottleneck, the cycle time reduction leads to an essentially direct reduction in the overall cycle time. For intermediate operations the situation is less clear, but we believe that improvements here can sometimes improve cycle time dramatically, and in the worst case, will not make cycle time any worse. If you focus your efforts strictly on bottleneck tools, then, you miss out on many opportunities for improvement."

2. Having too much WIP pile up on a short-term basis at a non-bottleneck tool can prevent that tool from feeding the bottleneck in a timely manner. If this leads to any starvation of the bottleneck, that will reduce capacity of the fab. In theory, yes, you should be keeping the right level of WIP in front of the bottleneck to keep this from happening. In practice, however, fabs are highly variable environments. A long unscheduled downtime on a one-of-a-kind tool that feeds the bottleneck can have negative repercussions, no matter what the loading normally is on that tool. A long PM on a one of two like tools that normally are each 75% loaded … same thing.

We don't think we were saying not to always keep an eye on your long-term bottlenecks. But we think that in a high-variability environment like a wafer fab, it's also worth keeping an extra eye on other tools that are problematic right now because of current performance issues.

David Carmichael's Additional Response:

"I have to disagree in a few places.

I have seen tremendous focus on non-constraint tool cycle time over the decades. The impact is exactly zero unless the lot is before the first or after the last constraint. It may well be zero if it is before the first constraint as lots just have to wait their turn at the constraint, i.e. they were started too early (starts goals are easiest to hit). Of course, people may forget that sometimes the constraint is in Sales and Marketing and not in the FAB.

By definition the constraints will define the throughput and no amount of local cycle time reduction will make any difference to that. Throughput = money (usually). If a line has more than one constraint then the smallest one wins and the larger capacity ones require less attention, though they still need to have buffers and be watched carefully.

My point was that allowing true constraint tools to run below capacity must not be acceptable. Your item #2 above must not be permissible and unfortunately very few people understand this. Each buffer should be big enough to reduce the chance of running dry to an acceptable level, remembering that if the constraint tool is idle, so is the line. This requires a system that will pull (you know drum-buffer-rope) the lots you need into the buffer. If you have two-of-a-kind tools, there is an increased risk of a constraint suddenly appearing, so you create a buffer for them even if you cannot sometimes keep it filled. Long PM's should not be random events. They are planned and so should WIP movement be planned, so that when tools become constraints for a short or long time, whichever like tools do exist are kept busy.

I have seen FAB's focus on Cycle Time and Moves exclusively for years. They never get better because they don't even know where their constraints are (or will be) and have no system to keep them busy. For me Cycle Time and Moves are the outcome of proper throughput management, not the root cause of anything.

I know how incredibly complex and variable FAB's can be but that is no excuse for not understanding the fundamentals. Buffers are the answer to FAB variability. Everything is about risk management and that should be the focus of the systems that support production."

FabTime Response:
Clearly, we have a philosophical difference as to whether it makes sense to manage cycle time at all. Given that FabTime's focus is on cycle time management, it seems we'll have to agree to disagree on that. Regarding a couple of specific points:

1. We do think that overall cycle time can be improved through changes at non-bottleneck tools that occur between visits to the bottleneck, even though some of that saved cycle time will be spent waiting longer for the bottleneck. This is because having the WIP waiting at the bottleneck, instead of at some other tool, may allow better decision-making at the bottleneck (reducing setups, for example).
2. We agree completely that avoiding starvation of the bottleneck is critical and have never said otherwise. David's reminder that "if the constraint tool is idle, so is the line" is a good one. But saying that something shouldn't happen and that something doesn't happen are not the same thing.

Perhaps other subscribers would care to weigh in on this debate. Two members of FabTime's user group share their thoughts further below.

Professor Russell Barton from Pennsylvania State University also wrote in response to the article about short-term bottlenecks:

"I wanted to tell you about two of my research projects that relate to monitoring and forecasting job progress and job completion times.

The first paper (with Jun Shu, published in *Production and Operations Management* in 2012) describes how to monitor timeliness and correctness of an object moving through a process. It is applied to supply chain (food products / Walmart) but also applies to a manufacturing lot. Here is the abstract:

> "Improvements in information technologies provide new opportunities to control and improve business processes based on real time performance data. A class of data we *call individualized trace data (ITD)* identifies the real-time status of individual entities as they move through execution processes, such as an individual product passing through a supply chain or a uniquely identified mortgage application going through an approval process. We develop a mathematical framework which we call *State-Identity-Time (SIT) Framework* to represent and manipulate ITD at multiple levels of aggregation for different managerial purposes. Using this framework, we design a pair of generic quality measures—timeliness and correctness—for the progress of entities through a supply chain. The timeliness and correctness metrics provide *behavioral visibility* that can help managers to grasp the dynamics of supply chain behavior that is distinct from *asset visibility* such as inventory. We develop special quality control methods using this framework to address the issue of overreaction that is common among managers faced with large volume of fast changing data. The SIT structure and its associated methods inform managers on *if, when*, and *where* to react. We illustrate our approach using simulations based on real RFID data from a Walmart RFID pilot project."

The second (Metamodel-Based Cycle Time Quantile Estimation for Real-Time Control of Manufacturing Systems, with Giulia Pedrielli at Arizona State University) shows that, using a simulation model of a process it may be possible to predict a completion time quantile for a job being released, given current work in process at each station. This paper has just been submitted for publication, so I can't publicly share extensive details right now, but the results are promising."

FabTime Response:

We read through the papers that Russell sent, and would be happy to send the first one to anyone interested. The second one offers promising results using meta-models to generate dramatically faster simulation-based estimates for cycle time quantiles. These models still incorporate fab complexities like shifting bottlenecks, lot priorities, and rework, which are all important for practical application of the methods. We will await the opportunity to share more details with you about that work in the future.

Regarding the first paper, we think that the idea of distinguishing between asset visibility (amount of WIP) and behavioral visibility (timeliness and correctness) makes sense. Bringing this framework to the fab, it's not simply a matter of WIP levels being high, but a matter of WIP being late, and of processing the right WIP. This ties in with WIP Hours, where we looked not just at the amount of WIP, but the amount of time it would take to process that WIP. That is, we are looking at the behavior of the WIP, not just the quantity of the WIP. As the authors note in the paper, "Properly monitored, behavioral visibility can provide early warning signals of problems that may or may not later emerge through asset visibility."

This also ties in with what Thomas said above, about pulling back on WIP that's being sent to already overloaded locations, and to what David said about proper throughput management in the fab.

The idea of avoiding overreaction in the presence of "large volume of fast changing data" also seems relevant to wafer fabs. The paper begins with the fact that many environments today are "data rich" and that "the data is at an unprecedented level of granularity". This is certainly true in wafer fabs. In the presence of this data, the authors recommend (as part of a larger framework) using Statistical Process Control (SPC) to "discriminate real change from a background of ordinary variation" in behavioral variables like cycle time.

Fabs are of course accustomed to using SPC to look at process variation. The framework in this paper applies SPC to different levels of supply chain performance, including metrics like operation cycle times within the fab. The authors share a simulated example of sales floor sojourn times in a Walmart and differentiate between a process going out of control (requiring intervention) and a process within normal variation (where intervention might just make things worse).

We wonder if our subscribers are using an SPC-type approach for monitoring manufacturing metrics in the fab, and whether this is something we should explore further in our software. We are grateful to Russell for sharing this research with us, and to all the contributors for making us think.

Responses from FabTime's User Group:

We shared an early version of this subscriber discussion section with several members of our User Group, as part of an ongoing discussion about improving reporting on short-term bottlenecks. They had some additional thoughts. **Hani Ofeck from Tower Semiconductor** said:

"If we look at each lot as a mini project that has a start and end time, any delay from that lot's target cycle time will have an effect on the end date. That is why it's important to find, locate and take care of short-term bottlenecks. The main complication is that there are many factors that can identify short-term bottlenecks.

Short-term bottlenecks tools may also be tools that combine high WIP with actual dedication problems (low redundancy). For example, suppose Flow A has 3 tools that can process Step A. If right now all three tools are unavailable, this means all WIP at Step A will be not run until one of the tools is back up.

There are many criteria that can be used to identify short-term bottlenecks:

- WIP above a planned capacity limit
- WIP Hours and Effective WIP Hours
- High rates of scrap or rework (indicating a quality issue)
- High WIP with Time Limitation between steps
- Current WIP with qualification rankings that indicate no path, one path, or two paths (depending on the situation in the fab)"

Justice Stiles from Infineon Technologies said:

"I think David Carmichael raises some fair points in the recent discussion of short-term bottlenecks; chiefly that it is the designed-in bottlenecks (gate tools for the purposes of this response to differentiate them from short-term bottleneck tools) that limit fab capacity and will have the largest effect on Cycle Time. That said, there are still good reasons to pay attention to short-term bottlenecks:

- In cases of time-constrained processing (i.e. timeouts) across multiple steps, feeding a short-term bottleneck is pointless and will cause unnecessary rework or potentially scrap in extreme cases. Rework can not only affect the on-time delivery (OTD) of a given lot, it creates unnecessary costs to the FAB.
- Short-term bottlenecks are the "indicator species" (to borrow a term from ecology) of other issues in the FAB that can lead to increased variability in Cycle Time and potentially missed OTD:
  - Too few qualified tools for a given operation relative to the demand for the operation or recipe (see FabTime's recent discussion in Issue 20.05 of the effect of the number of qualified tools on cycle time, which was quite good).
  - A loss of qualified tools due to unplanned down time or (hopefully not) poorly scheduled planned maintenance.
  - A change in product mix creating higher than predicted demand on a given toolset.
  - Lower than expected throughput (UPH) from a toolset causing the actual capacity of the tools to be lower than the predicted capacity.

In all the above cases, investigating and addressing the root cause of the ephemeral bottleneck yields long-term benefits to FAB stability and Cycle Time. As everyone agrees, wafer fab manufacturing is complicated. While gate-tools will be the key drivers of cycle time, that does not mean we should ignore short-term bottlenecks and the underlying issues they speak of."

FabTime Response:
We appreciate Hani and Justice's support for the importance of identifying short-term bottlenecks. We especially liked Hani's suggestions of additional criteria to identify them, and Justice's points about short-term bottlenecks as "indicator species". We will discuss their suggestions more as we flesh out some of these ideas further in our main article.

We are grateful to all the subscribers who took time to respond to this topic, expanding the discussion and showing us things that we either missed, or never knew about in the first place.

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

Further Thoughts on Short-Term Bottlenecks

Last month's main article about identifying short-term bottlenecks generated a robust response (see above). We hope you'll take time to read through the intelligent and varied contributions from our engaged subscribers. In this brief article, we share some follow-up thoughts derived from those responses.

First, a pair of philosophical questions:

1. Is it possible to improve overall cycle time by making improvements at tools that are not bottlenecks (aka gate tools) in a capacity sense?
2. Should fabs focus on identifying and managing short-term bottlenecks, in addition to managing gate tools?

FabTime believes that improvements in cycle time at tools that are not capacity bottlenecks can help with overall cycle time, even if those tools are visited in between visits to the "true" bottlenecks. While these improvements may be small compared with making improvements to the long-term bottlenecks (something that should certainly remain a focus), they may still, in the high-variability environment of a fab, make a difference.

Even if, like David Carmichael, you are skeptical of this claim (because any time saved at the other tools ends up added to cycle time at the gate tools), we believe there are still reasons to identify short-term bottlenecks. As Justice Stiles notes above, short-term bottlenecks can be an indicator of data problems or other issue that are driving up fab variability. We can think of short-term bottlenecks as an "indicator species", like OEE loss factors, helping to direct improvement efforts.

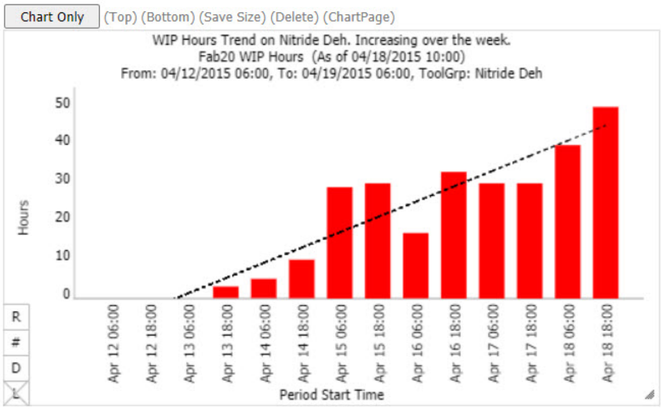
In this article we'll discuss potential extensions to the WIP Hours paradigm from our earlier article, additional methods for identifying short-term bottlenecks, and uses of short-term bottlenecks as indicator species. As always, we welcome your feedback.

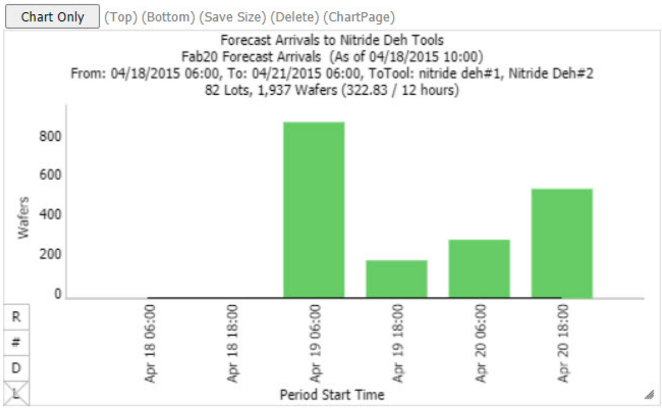
Extensions to WIP Hours as a Metric

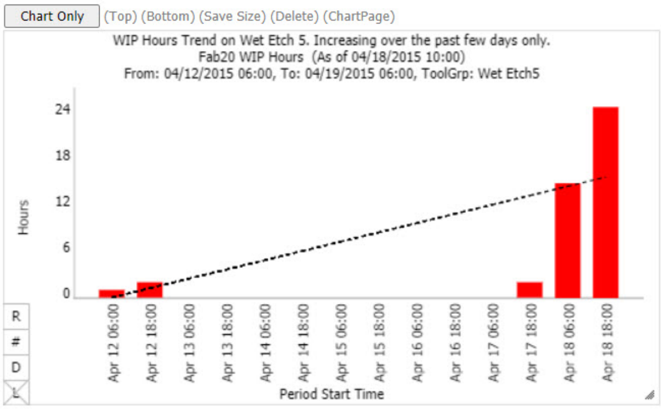
In our previous issue, we suggested the WIP Hours metric as a means of identifying short-term bottlenecks. WIP Hours measures the estimated hours of required processing time for WIP in queue at a tool. Effective WIP Hours is an enhancement of WIP Hours that accounts for the number of tools that are currently available. Tools that have WIP Hours (or Effective WIP Hours) greater than the length of a shift may be worthy of extra attention. Here are a few potential extensions to the WIP Hours paradigm based on our subscribers' input.

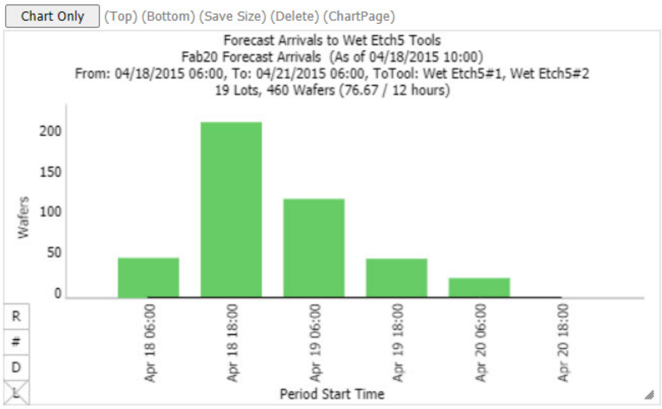
<span style="color:red">**Increase or Decrease:**</span> Thomas Quarg's description of his "Dynamic Kanban" project suggested to us that knowing whether WIP Hours was increasing or decreasing at a tool would be a useful additional indicator of tools worthy of attention. Determining this will likely require a bit of nuance, however. We can certainly look at the WIP Hours values for a tool over the past few hours or shifts and see if they are trending up or down. But deciding exactly what time window to use for that is a bit subjective. One thing that we think might be useful here is to look at the Forecast Arrivals to each tool over the next couple of shifts. Where WIP Hours are high and a significant amount of new WIP is expected, tools could require extra focus.

Here's a quick illustration, continuing our example from the previous issue. Suppose we've identified two tool groups as short-term bottlenecks based on WIP Hours: Nitride Deh and Wet Etch5. Let's look at the WIP Hours Trend (charts on the left) and Forecast Arrivals Trend to those tools by shift. Each is a tool group with two similarly qualified tools, such that the WIP Hours per Tool would be half of the value shown on the Y axis.


WIP Hours Trend on Nitride Deh. Increasing over the week.
Fab20 WIP Hours (As of 04/18/2015 10:00)
From: 04/12/2015 06:00, To: 04/19/2015 06:00, ToolGrp: Nitride Deh


Forecast Arrivals to Nitride Deh Tools
Fab20 Forecast Arrivals (As of 04/18/2015 10:00)
From: 04/18/2015 06:00, To: 04/21/2015 06:00, ToTool: nitride deh#1, Nitride Deh#2
82 Lots, 1,937 Wafers (322.83 / 12 hours)


WIP Hours Trend on Wet Etch 5. Increasing over the past few days only.
Fab20 WIP Hours (As of 04/18/2015 10:00)
From: 04/12/2015 06:00, To: 04/19/2015 06:00, ToolGrp: Wet Etch5


Forecast Arrivals to Wet Etch5 Tools
Fab20 Forecast Arrivals (As of 04/18/2015 10:00)
From: 04/18/2015 06:00, To: 04/21/2015 06:00, ToTool: Wet Etch5#1, Wet Etch5#2
19 Lots, 460 Wafers (76.67 / 12 hours)

Here we see that while Nitride Deh has a longer-lasting trend of increased WIP Hours, Wet Etch5 has more WIP expected to arrive over the next 24 hours. Which tool group is more likely to be able to work off the hours of WIP already in queue before more arrivals come will depend on how well the tools perform. Both tool groups warrant efforts to keep them running smoothly.

The Forecast Arrivals charts in FabTime are based on planned cycle time data by step, rather than on actual conditions. A further enhancement to this approach would be to use a method like the one described in Russell Barton's second paper above to use metamodels that predict future lot progress based on current conditions. We'll await more details of those methods in the future. Another potential enhancement would be to project WIP Hours forward, based on the forecast arrivals. Whether this would be accurate enough to be useful is an open question.

WIP Hours by Layer or Operation. Thomas also mentioned that his dynamic Kanban approach considered layers or operations, to make better decisions regarding which WIP to process first. The idea was, for example, to not feed a layer if it already had more hours of WIP than could be absorbed by the shift. We can envision a stacked version of the WIP Hours chart that would help with this, and consequently allow WIP Hours to inform dispatching and scheduling decisions in general.

Time Constraints. Both Hani Ofeck and Justice Stiles mentioned the challenge of short-term bottlenecks that are inside time constraint regions. If using WIP Hours to identify short-term bottlenecks, it seems to us that the threshold for declaring a tool a problem would be lower for tools inside constraint regions. This is another case (as with layers that already have more hours of WIP than can be absorbed during the shift) where the short-term bottlenecks list may impact dispatching and scheduling decisions. If a tool inside a constraint region already has more hours of WIP than can be processed before WIP starts to time out, release of lots into that region should be restricted. We have not written extensively about time constraint regions in the newsletter but did discuss dispatching for time constraint regions back in Issue 11.05.

Other Ways to Identify Short-Term Bottlenecks

Several new ideas for identifying short-term bottlenecks arose from the subscriber discussion above.

- We could see a control-chart type of approach, as discussed in Russell Barton's contribution above, for identifying tools for which scrap or rework rates are higher than normal. While such tools may not be bottlenecks in a capacity sense, they absolutely fit our larger goal of looking for the tools that require special attention right now.
- Hani Ofeck offered a reminder that in addition to using WIP Hours, setting targets based on WIP levels can be a useful practice. This is especially true given that calculating WIP Hours is a bit complex. It might also be possible to use a control chart-type approach to automatically highlight situations where WIP levels are higher than usual. The challenge would be setting appropriate and useful WIP levels in the presence of fab variability. Using inventory age (how long each lot has been at its current operation) also makes sense here, as was discussed in the previous issue.
- John Paul Gauci offered the suggestion, new to us, to use load port utilization to identify when WIP is not flowing well from operation to operation. This metric is most relevant in fully automated fabs, with which FabTime has less experience. See also the Winter Simulation Conference paper that John recommended for other AMHS-related metrics.

We neglected to reference in the last issue an earlier article (Issue 4.09) that FabTime published, way back in 2003, about identifying temporary bottlenecks in the fab. Readers interested in this topic will find discussions of a few additional metrics like A80, and a very early introduction to WIP Hours.

Investigating and Exploring Root Causes

We think there is value in the idea, suggested by Justice above, of using short-term bottlenecks as a starting point for improvement efforts, particularly in the areas of tool qualification, unavailable time, and planning data.

Tool Qualification: The appearance of one or more tools from a tool group on the short-term bottlenecks list, where there is enough capacity overall, can be a red flag regarding qualification issues (as mentioned by both Hani and Justice). In particular, short-term bottlenecks may indicate soft constraints, where the planning models suggest a redundancy that is not used in practice. If you see high WIP Hours even though one or more tools in the group has been idle recently, there may be something worth looking into regarding operator preferences or layout issues. In general, investigations into tool qualification issues can be of significant benefit for cycle time improvement efforts. See Issue 20.05 for more details.

Maintenance Scheduling: A key driver of short-term bottlenecks is extended periods of unavailable time on one or more tools in a tool group. While unscheduled downtime is sometimes unavoidable, efforts to understand the causes of such events can be important. Where poorly scheduled maintenance events or blocks of engineering time lead to short-term bottlenecks, investigation is always a good idea. See Issue

12.04 for a discussion about the impact of PM scheduling on cycle time. See Issue 20.02 for the introduction of the Green-to-Green metric, which captures the total time that tools are unavailable to manufacturing.

Planning Data: When fabs plan their capacity, they know which tools are expected to be bottlenecks. Managers should, as discussed by David Carmichael above, make every effort to keep such tools busy. In practice, however, those plans are based on assumptions about product mix, UPH rates, and scrap and rework rates. When a tool shows up regularly on the short-term bottlenecks list, this may be an indication that either the planning data is incorrect, or the product mix has shifted without the plan being updated. Communication between the manufacturing team and production planners or industrial engineers is recommended.

Conclusions

In the previous newsletter issue, we discussed the identification of short-term bottlenecks in a wafer fab. We defined a short-term bottleneck as tool or tool group for which, over the next 12-24 hours, required capacity is likely to exceed available capacity. We focused primarily on use of the WIP Hours metric for identifying these tools, but also reviewed several other techniques. That article generated quite a bit of feedback, some positive and some more skeptical. Each of the contributions added to our own understanding of the topic, leading us to create this follow-up article.

In this issue, we distill key open questions and enhancements to the short-term bottleneck paradigm. We share several potential extensions to the WIP Hours metric, summarize additional methods for identifying short-term bottlenecks, and highlight some key uses of short-term bottlenecks for identifying underlying problems in the fab.

In a perfect, Eli Goldratt-consistent world, nothing described in this article would be necessary. Fabs would identify a couple of key constraint tools and subordinate everything else to keeping those tools running. In practice, the variability in a fab can make things more complex. Tools running single path operations go down. Product mix changes. Rework rates are higher than expected for a new process running through a time constraint loop. Operator preferences for certain tools differ from the expectations of the capacity model. And so on. We hope that the ideas outlined in this pair of newsletter issues will help you to identify these temporary bottlenecks quickly when they arise and respond accordingly.

We plan to work with our User Group to expand our software to better reflect the insights shared here. We welcome additional feedback.

Closing Questions for Newsletter Subscribers

Is it possible to improve overall cycle time by making improvements at tools that are not bottlenecks in a capacity sense? Should fabs focus any effort on identifying and managing short-term bottlenecks? What other extensions to WIP Hours as a metric do you think are needed? What other methods do you use to identify short-term bottlenecks?

Further Reading

All past issues of FabTime's newsletter are available to subscribers for download from our website. Contact Jennifer.Robinson@FabTime.com for the current password.

- J. Robinson and F. Chance, "Identifying Temporary Bottlenecks in the Fab," *FabTime Newsletter*, Vol. 4, No. 9, 2003.
- J. Robinson and F. Chance, "Improving Factory Cycle Time through Improvements at Non-Bottleneck Tools," *FabTime Newsletter*, Vol. 10, No. 9, 2009.
- J. Robinson and F. Chance, "Time Constraints and Reverse Dispatch in Fabs," *FabTime Newsletter*, Vol. 11, No. 5, 2010.

- J. Robinson and F. Chance, "PM Scheduling and Cycle Time," *FabTime Newsletter*, Vol. 12, No. 4, 2011.
- J. Robinson and F. Chance, "A Metric for Green-to-Green (G2G) Analysis," *FabTime Newsletter*, Vol. 20, No. 2, 2019.
- J. Robinson and F. Chance, "The Impact of Tool Qualification on Cycle Time," *FabTime Newsletter*, Vol. 20, No. 5, 2019.
- J. Robinson and F. Chance, "Identifying Short-Term Bottlenecks," *FabTime Newsletter*, Vol. 21, No. 4, 2020.
- J. Shu and R. R. Barton, "Managing Supply Chain Execution: Monitoring Timeliness and Correctness via Individualized Trace Data," *Production and Operations Management* 21, 715-729, 2012. Available from https://doi.org/10.1111/j.1937-5956.2012.01353.x.
- J. Tung, T. Sheen, M. Kao and C. H. Chen, "Optimization of AMHS Design for a Semiconductor Foundry Fab by Using Simulation Modeling," Proceedings of the 2013 Winter Simulation Conference, 2013. Available for download from the WinterSim archive.

Acknowledgements

FabTime is grateful to John Paul Gauci (JPG Consulting), Thomas Quarg (AMTC), David Carmichael (Tower Semiconductor Ltd.), Russell Barton (Penn State University), Hani Ofeck (Tower Semiconductor), and Justice Stiles (Infineon Technologies) for their contributions, all of which informed this article. We are also grateful to our User Group for additional discussions regarding bottlenecks.

Subscriber List

Total number of subscribers: 2819

Top 20 subscribing companies:

- ON Semiconductor (223)
- Infineon Technologies (153)
- Micron Technology (117)
- Intel Corporation (115)
- GlobalFoundries (101)
- Maxim Integrated Products (85)
- NXP Semiconductors (80)
- Carsem M Sdn Bhd (70)
- Microchip Technology (70)
- Skyworks Solutions, Inc. (66)
- STMicroelectronics (66)
- Western Digital Corporation (63)
- Texas Instruments (56)
- Seagate Technology (50)
- X-FAB Inc. (50)
- Qualcomm (41)
- Analog Devices (40)
- Tower Semiconductor (34)
- Cree / Wolfspeed (32)
- Honeywell (31)

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:

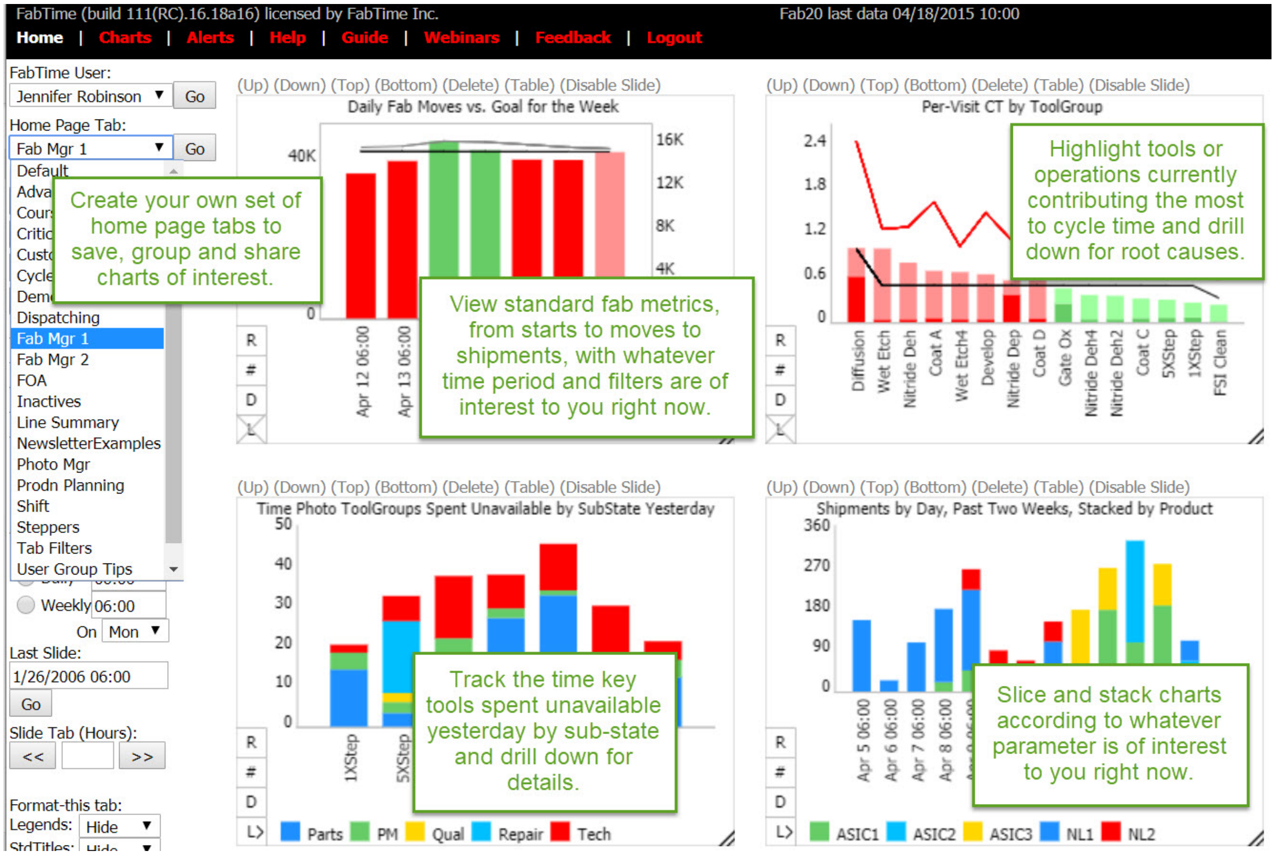
- EV Group
- NVIDIA
- Photon Sciences
- TE Connectivity
- Vanguard International Semiconductor

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.



FabTime (build 111(RC).16.18a16) licensed by FabTime Inc.

Fab20 last data 04/18/2015 10:00

FabTime User: Jennifer Robinson

Home Page Tab: Fab Mgr 1

Weekly 06:00 On Mon

Last Slide: 1/26/2006 06:00

Legends: Hide

Create your own set of home page tabs to save, group and share charts of interest.

View standard fab metrics, from starts to moves to shipments, with whatever time period and filters are of interest to you right now.

Highlight tools or operations currently contributing the most to cycle time and drill down for root causes.

Track the time key tools spent unavailable yesterday by sub-state and drill down for details.

Slice and stack charts according to whatever parameter is of interest to you right now.