INFICON
Inspired by visions. Proven by success.

# FabTime® Newsletter

Volume 25, No. 5 September 2024

## Information

**Publisher:** Acquired by INFICON in early 2024, FabTime has been helping fabs with cycle time and performance improvement since 1999. FabTime's flexible reporting software, cycle time management course, and this newsletter are now part of the INFICON Intelligent Manufacturing Systems (IMS) group.

**Editor:** Jennifer Robinson, Cycle Time Evangelist for INFICON

**Date:** Tuesday, September 17, 2024 – Vol. 25, No. 5

**Contributors:** Subramanian Pazhani (SAS Institute and NC State); Florin Oprea (Qorvo)

**Keywords:** Equipment Downtime; Metrics and Goals; Operating Curves

## Table of Contents



## Welcome

Welcome to Volume 25, Number 5 of the FabTime Cycle Time Management Newsletter. I hope you all had a lovely summer. In this issue we have announcements about two upcoming webinars, a new team member, and various upcoming in-person events. I also share highlights from my LinkedIn page, as usual. This month's FabTime software tip of the month is about understanding the slice (pareto) and cross (stack) options on FabTime charts. In the subscriber discussion forum, we link to a paper related to last month's main topic, impact of downtime variability, and share a question about generating the operating curve for a fab.

Our main article this month is about a new metric called (provisionally) **functional utilization**. This metric attempts to capture the portion of utilization that drives cycle time at the tool level by distinguishing between standby time with WIP waiting and standby time when no WIP is waiting.

A couple of administrative notes: we will be redirecting the legacy FabTime.com website to INIFCON.com in October. Past issues of the newsletter and other FabTime resources will be available from the INFICON website after the change. We will also be changing how the newsletter is distributed starting with the November issue. **You may need to re-subscribe.** See details below on what to expect.

Thanks for reading! – Jennifer

# Community News/Announcements

## FabTime, FPS, FabGuard Customers: Don't miss the IMS 2024 User Group Meeting


It's not too late to sign up for the INFICON IMS User Group Meeting/Smart Manufacturing Symposium, October 15 – 17 in Austin, Texas. The event will have multiple sessions for FabGuard®, FPS, and FabTime® over the course of three days.

This will be the FabTime team's first time participating in this event, and we hope that many of our customers will attend. Plan for lively discussions and collaboration on a variety of topics, including:

- **Special in-person 2-hour version of Jennifer's Cycle Time Management Class**
    - The full course is normally only available via Teams and costs $10,500/session for a site. Attending for free at the UGM is a great opportunity!
- Panel Discussion on Aligning Metrics Definitions between FabTime and FPS
- In Depth End User Presentations
- Sensor Applications and Technologies
- AI/ML and Smart Manufacturing
- Subfab and Facilities Management and Control
- Technology Roadmaps and New Capabilities

Customers of FabTime, FPS, and/or FabGuard products are welcome and encouraged to attend. Please contact Mike Neel for more information.

## See our SEMI FOA Fab Star webinar on Predicting the Future of Factories

**Holland Smith** from INFICON will be presenting a webinar on **September 18th at 8:00 am Pacific Time** as part of the SEMI FOA Fab Star series. The title is: "Silicon Crystal Balls: The Science, Math and Methods of Factory Future Prediction, Scheduling and Control." **Contact us to register.**

Anyone who has ever invested money in the stock market has a sense of what it is like to aim for positive outcomes amidst stochastic circumstances with no guarantees. Semiconductor factories share many characteristics with financial markets: complicated to model, governed by difficult mathematics and plagued by stochastic disruptions. Nevertheless, successful manufacturers must meet their commitments, and doing so efficiently inevitably involves future state forecasting – whether for capacity modeling, starts planning, scheduling, etc. This talk will describe the types of questions that can be sensibly asked about semiconductor factory futures and the techniques that are used to answer them, with particular focus on high-ROI production-worthy applications. It will also speculate on where and how "AI" may help.

## Join our INFICON webinar on Equipment Downtime Metrics


We received such great feedback to the previous newsletter main article about equipment downtime that we decided to turn it into a webinar, with the opportunity for questions and feedback. Please help us to spread the word! This webinar is open to all.

> Is equipment downtime a top contributor to cycle time in your fab? It is for most fabs. Downtime reduces buffer capacity, increases process time variability, and decreases the available number of qualified tools on a given shift. Understanding the factory physics underlying these effects suggests which metrics are most helpful for driving cycle time improvement. The INFICON FabTime reporting module puts these metrics, along with proactive alerts, at your fingertips, allowing you to mitigate the impact of equipment downtime and reduce overall cycle time in your fab.
>
> Join **Jennifer Robinson**, Cycle Time Evangelist for INFICON, for this webinar to learn to select the best equipment downtime metrics for wafer fab cycle time improvement.

Date: November 6, 2024, at 8:00 am Pacific Time. Register here.

## Changes to the distribution of the newsletter starting with the next issue

We will be moving this newsletter to the INFICON mailing system starting with the November issue.

**Depending on where you are located and when you subscribed, you may need to re-subscribe to continue receiving the newsletter after this issue.**

If that is the case for you, Jennifer will send you a separate email soon with the link to re-subscribe. **Please keep an eye out for that email**. We apologize for any inconvenience as we work to protect your personal data during this transition.

Here are changes that you'll see:

- Newsletter issues will be sent from an INFICON newsletter address, instead of directly from Jennifer's email.
- Instead of a PDF attachment, new issues will be sent as formatted emails with blurbs for each section. You'll be able to click through to the website to read the full text. As an example, you can view the main article from the July newsletter issue via this link: Improve Fab Cycle Time by Tracking the Right Equipment Reliability Metrics.

In related news, the archive of past newsletter issues will be moved to the INFICON website starting in October. You will still be able to download a full archive of all past issues in PDF format or download individual issues. If you would like to download past issues now, before the transition, the password to access the FabTime.com archive is FabTimeCommunity.

You can also now download the FabTime Operating Curve Spreadsheet, a useful Excel-based tool for exploring tradeoffs related to variability, utilization, and number of qualified tools, from the INFICON website. Other useful industrial engineering resources such as queueing formulas and a cycle time tutorial will be made available soon. There is also now a link on the INFICON website for requesting information about the FabTime Cycle Time Management Course.

We appreciate your patience as we make these changes. We are committed to maintaining the quality of the newsletter going forward and to making it ever more useful to subscribers. We welcome your questions and feedback.

## Join us in welcoming Paul Campbell to INFICON

INFICON is delighted to welcome **Paul Campbell** to the IMS Team. Paul has been working to improve operational excellence in the semiconductor industry for more than 20 years. Based in the United Kingdom, he worked as a senior Industrial Engineer for Atmel NTS and RFMD before joining MAX Industrial Engineering Group in 2008. After 11 years with MAX, including as VP of Operations for Europe, Paul launched his own consulting firm.

Paul has helped many fabs to improve their operations over the years. We are excited to bring his extensive expertise to the INFICON team. Jennifer and Frank are particularly pleased. We worked closely with Paul on the FabTime installation at Atmel back in 2008 (when doing an installation required extended time onsite) and have remained friends ever since.

Paul will help us in a variety of areas. He'll start by working with our team to establish more standardized methods for measuring baseline factory performance. This will help our customers to understand productivity impacts from new software deployments, and to identify opportunities within the factory for further improvement.

## A second Women in FOA Brunch will be held at the October FOA meeting


Back by popular demand, there will be another Women in FOA brunch at the Q4 Fab Owners Alliance meeting. The brunch will be held on Wednesday, October 23rd from 11 to 3, followed by a team-building activity, prior to the main meeting on the 24th. The meetings will be held at the Texas Instruments fab in South Portland, Maine. To encourage attendance, the FOA will allow suppliers to bring a third attendee to the meetings without an additional fee, provided that third attendee is a woman. Jennifer will be representing INFICON at the brunch and looks forward to seeing you there.

## This Call for Papers for ASMC 2025 may also be of interest to subscribers

The call for papers for the 2025 Advanced Semiconductor Manufacturing Conference is now open. Submissions are due October 18, 2024. The conference will be held May 5-8 in Albany, NY. More details, including requested topics, are available at SEMI's ASMC 2025 Author Kit website.

## See 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:

**FabTime team members meet with the INFICON CEO**

- A reshare of a post by INFICON CEO **Oliver Wyrsch** about meeting several members of the FabTime team at a delightful dinner in Thousand Oaks, CA. Shown left to right: **Steve Lakeman, Jennifer Robinson, Oliver Wyrsch, Frank Chance, Evan Rozhon, Bailey Brouse, Lara Nichols,** and **Lauren Donahue**. As a largely virtual company, it was rare for even the original FabTime team to assemble in person. We appreciate Oliver for bringing us together this summer, and for the warm welcome to INFICON.
- A Reuters piece about the quality challenges that Intel is facing with Broadcom products, which are harming Intel's turnaround/foundry business plans. Or, as Jennifer said on LinkedIn, further confirmation that "yes, running a wafer fab foundry is challenging."
- A Semiconductor Engineering piece by **Byron Moyer** about how and why even in this time of megafabs, certain smaller fabs and assembly houses remain profitable and necessary (sensors, medical devices, defense, etc.). **Tim Brosnihan** from the SEMI FOA is quoted extensively, and there's quite a bit about product mix.

  This analogy at the end was especially good: "One can think of the marketplace as an orchard. The high-volume chipmakers and assembly firms can come through with automation that strips the easy-to-harvest fruit from the trees in high volumes and with high efficiency. But they can never get everything. So other companies can come in with specialty people and ladders to access the still-perfectly-good fruit left behind. As long as demand remains for all the fruit, both types of company can do well."
- A University of Texas article about the news that DARPA has selected the Texas Institute for Electronics at UT to "develop the next generation of high-performing semiconductor microsystems for the Department of Defense," awarding them $850M. Congratulations to the TIE team and especially to longtime friend of FabTime **Patrick Meyer**.
- A WSJ article by **Jiyoung Sohn** is about worker unrest in the semiconductor industry. It highlights a specific aspect of the labor shortage that we've been hearing more about lately: the critical role of maintenance engineers.

  "The 24-year-old **Kim (Jae-won)** is an engineer responsible for detecting equipment failures and replacing parts, among other tasks. Samsung's factories run around the clock, pumping out the kind of chips used in products such as smartphones, computers and data servers. A typical eight-hour shift often involves five maintenance issues or more, Kim said.

"The company says there is a high level of automation, but in reality, without maintenance engineers at work, dozens of errors can occur in a single step of the chip-making process," he said."

For more industry news, connect with Jennifer on LinkedIn.

# FabTime® Software Tip of the Month

## Understand Slice and Cross Options for Displaying Data

The "Slice" and "Cross" filters make the number of different charts you can create in FabTime essentially infinite. But you do have to know that this capability is there and know how to use it.

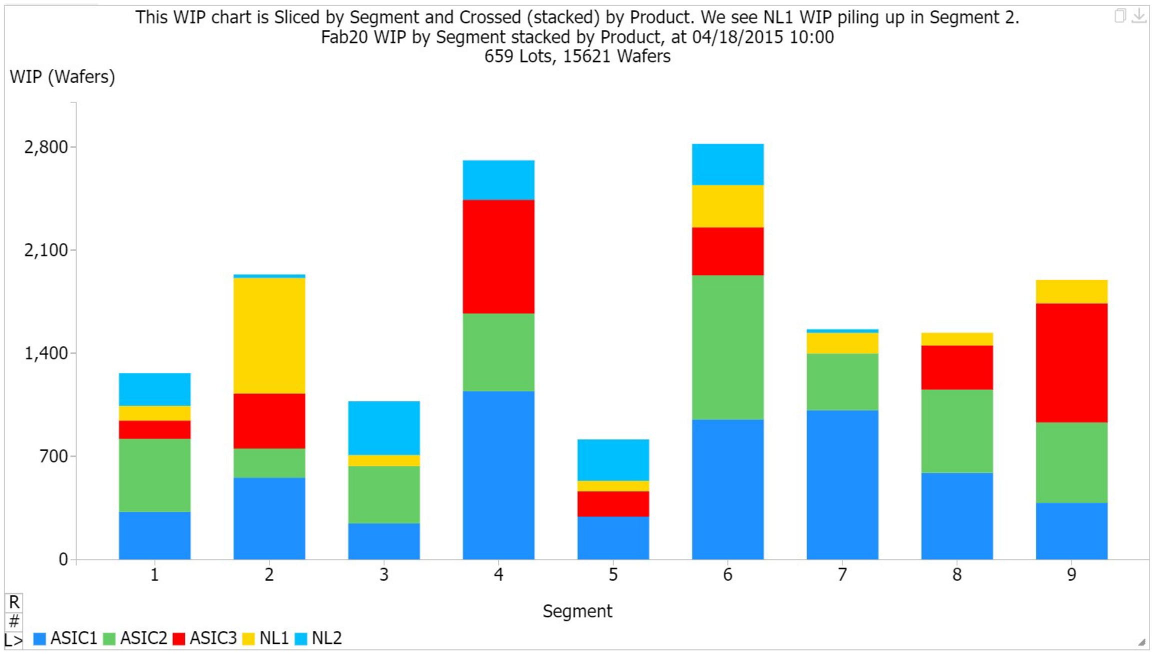
All Pareto charts in FabTime (Moves Pareto, WIP Pareto, Turns Pareto, etc.) have a "Slice" control on the detailed chart page. The Slice control lets you change what variable is shown on the x-axis. The default Slice is set to "Area" for most sites. You can change the Slice to any of up to 25 different variables on any chart, depending on how FabTime is configured for your site. The total data included in the chart doesn't change (e.g. the total number of moves for the chart's time period), but you choose how you slice it up along the x-axis. For example, let's look at the Operation Cycle Time Pareto Chart.

1. Enter "Operation Cycle Time" in the FabTime Search box and select "Operation Cycle Time Pareto."
2. Change the date range to the current week (or other period of interest to you).
3. Scroll down to find the "Slice" control at the bottom of the big set of filters to the left of the chart and change it from "Area" to "ToolGroup". You will need to press the "Go" button. You can't just hit enter from inside a drop-down list.
4. The resulting chart will display the average operation cycle time for each Tool Group, sorted (by default) in descending order of cycle time.
5. If you choose a Slice variable that has many values, such as Operation, you may want to limit the number of values displayed on the chart. To do that, click "Top # Points" below the chart and data table and enter the number of points you would like to see on the chart. [In older versions of FabTime, the Points input box is located with the format controls in the lower-left corner of the screen.]
6. Be aware that not all Slice values are relevant for all charts. Sometimes a selection will show as "Undefined", if that attribute isn't mapped for the current chart. (E.g. Tool Model Number on a WIP chart). Sometimes a selection just doesn't make sense. Common ways that people Slice the Moves Pareto are by Area, Tool Group, Employee (Operator), Product, or Priority. People tend to Slice the WIP Pareto by Location, Priority, Segment, Product, etc. Ultimately, the choice is up to you.

The "Cross" control applies to all Stacked charts, whether they are Trend or Pareto charts. The Cross control lets you change what the data in each column is stacked by. Changing the Cross control won't change the height of any column on the Y-axis, but will change how the data is displayed up within the column. For example, let's look at the WIP Stacked Pareto chart.

1. Enter "WIP Stacked Pareto" in the FabTime Search box and select the chart.

2. Leave the date box blank to see the most recent WIP (or enter some past date if needed).
3. Scroll down to find the "Slice" and "Cross" controls at the bottom of the main set of filters. Change "Area" in the "Slice" filter to "Segment". Change "Area" in the "Cross" filter to "Product" (or "Route", "Priority", or "Owner" as you prefer). Press the "Go" button. The result for our demo server is shown below.




4. It is possible to edit the colors used for stacked charts at the site level (with a user-level permission setting). However, the selections apply for all users at the site. It is not currently possible to change the stacking colors on an individual chart.

By using the Slice and Cross drop-downs, you can generate a wide range of charts. Remember to add any chart that you wish to save to a Home Page Tab. We hope you find this tip useful in driving improvement efforts in your fab.

Please send your software questions or suggestions for software tips to **Elaine Jacobson**, the Program Manager of Customer Success for FabTime, Elaine.Jacobson@inficon.com. Thanks!

# Subscriber Discussion Forum

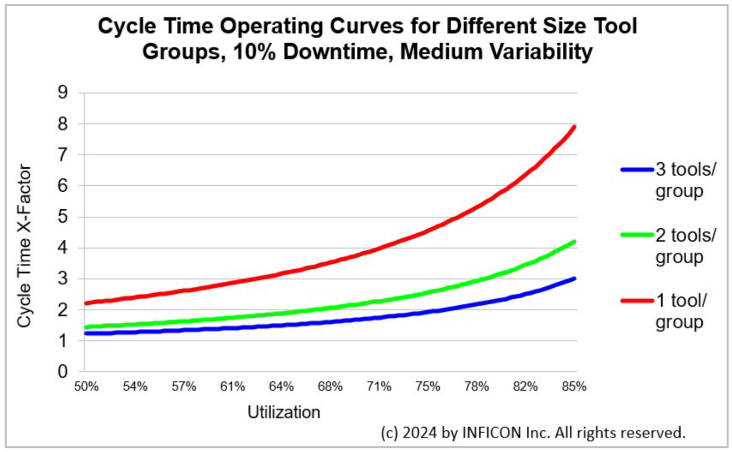
## Impact of Downtime Variability

Following up on the previous main article about equipment downtime, **Subramanian Pazhani of SAS Institute and NC State** was kind enough to send us a paper that he published on Thomas Beeg's Factory Physics and Automation blog. The author compares two tool groups to show that a tool group with lower utilization can have a higher queue time than a tool group with higher utilization, under certainly downtime variability characteristics. He first looks at two tool groups with the same availability and utilization, where one has a higher variability of repair times (and thus a higher queue time). He then looks at two tool groups with the same availability where one has shorter, more frequent downtimes and the second has longer, less frequent downtimes. Even though the utilization is lower for the second tool group, the cycle time is higher because of the significant impact of the long downtimes.

These results are consistent with what we discussed in the main article in the last newsletter (Improve Fab Cycle Time by Tracking the Right Equipment Reliability Metrics (Issue 25.04)) and with examples included in our Cycle Time Management Course. We recommend this paper to anyone interested in the topic – the examples are clear and the queueing formulas explained in more detail than was included in our newsletter article.

## Fab Utilization vs. X-Factor

**Florin Oprea from Qorvo** reached out on LinkedIn to ask: "Do you have a chart showing fab utilization vs. x-factor. I believe I've seen you publishing one before where a factory utilization of 85% was a good utilization target to attain 4X theoretical cycle time. Share please if you have this chart."




**Response:** We don't have a general operating curve (graph of x-factor vs. utilization) for the entire fab. We have a spreadsheet for generating operating curves for individual tool groups using queueing models. That one does show that in the presence of some (~10%) downtime and moderate arrival/process variability, a tool group with two tools will have an x-factor of ~4 at 85% utilization. See the image above for an illustration.

In a full fab, of course, you have a mix of tool group sizes and tool utilizations, but we have seen that 85% is a common utilization target for fabs because it keeps away from the steep part of the operating curve (except in the case of one-of-a-kind tools, where a lower utilization target may be necessary).

What happens in practice is that the x-factor is higher for some tool groups (more expensive tools where the utilization is pushed higher, one-of-a-kind tools, batch tools, tools with high variability and/or poor uptime performance) and lower for other tool groups (less expensive tools with excess capacity and tool groups with a high level of redundancy). But setting 85% as a general target has been a common practice in the industry for many years.

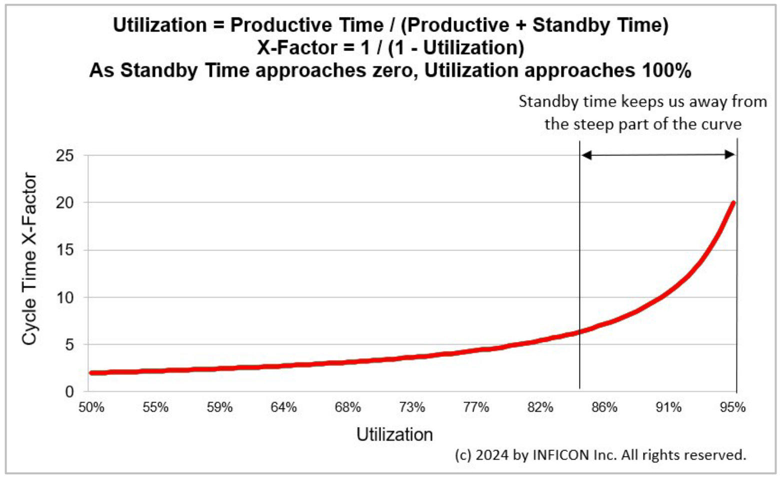
Subscribers can download the Operating Curve Spreadsheet from the FabTime Newsletter Archive. The password is FabTimeCommunity, and the link to download the spreadsheet is above the table of past issues.

We welcome the opportunity to publish subscriber discussion questions and responses. Simply send your contributions to Jennifer.Robinson@inficon.com.

# Main Article: A New Metric for the Functional Utilization that Drives Cycle Time

## Introduction

**Why a Standby Time buffer is needed for to avoid high CT**




Though there are several different definitions of utilization, Manufacturing Utilization, tracked as Productive Time / (Productive + Standby Time), is a primary driver of cycle time at the tool group level. Tool groups need to have some standby time as a buffer against the variability that is endemic in wafer fabs. Without that standby time, utilization rises close to 100%, and cycle time becomes unacceptably high, as shown in the operating curve above.

It is our belief, however, that this definition of manufacturing utilization is not always sufficient to capture the impact of standby time on cycle time. In this article, we propose a supplemental utilization metric that differentiates between standby time when there is no WIP waiting (which is helpful for cycle time) and standby time when there is WIP waiting (which is detrimental for cycle time). We are provisionally calling this new metric **Functional Utilization** and are seeking feedback from the newsletter community regarding this new terminology.

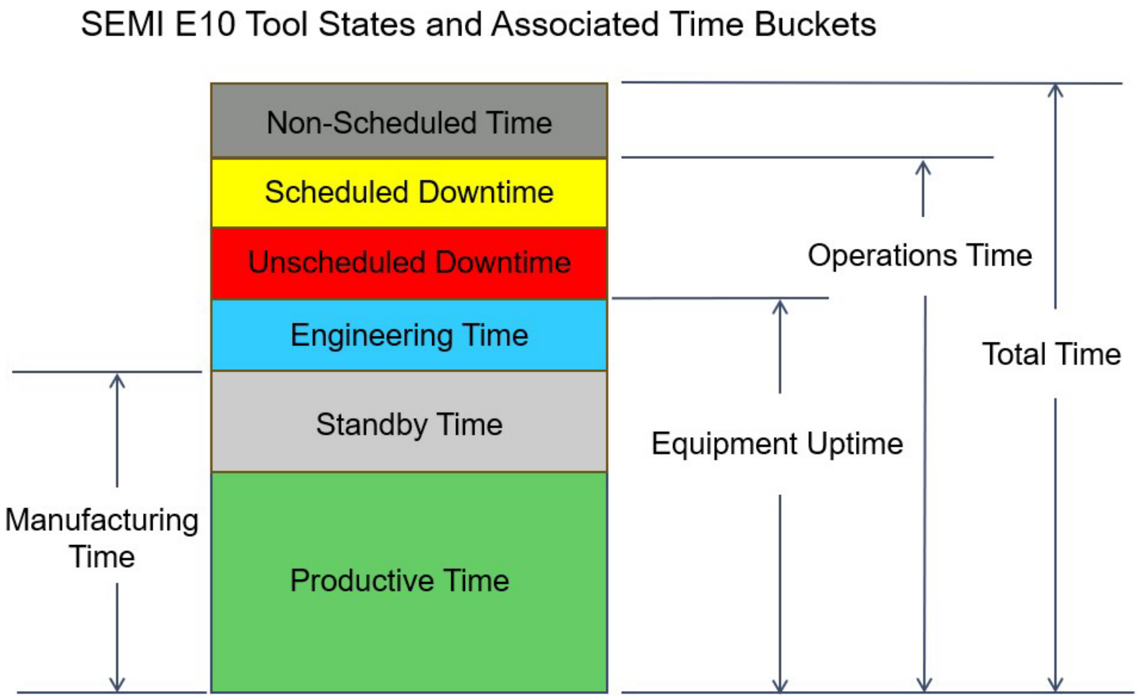
## Manufacturing utilization is a commonly used metric to track tool performance, one of several definitions of utilization

The FabTime and FPS Digital Twins historically included various definitions of tool utilization, each based on the percentage of productive time relative to a different, SEMI E10-defined time bucket. SEMI E10 is the SEMI Standard Specification for Definition and Measurement of Equipment Reliability, Availability, and Maintainability (RAM) and Utilization, available for purchase from SEMI's website. The SEMI-specified E10 tool states and time buckets are shown on the next page. The corresponding utilization definitions are:

- **Total Utilization %** = Productive Time / Total Time
- **Operational Utilization %** = Productive Time / Operational Time
- **Equipment Uptime Utilization %** = Productive Time / Equipment Uptime
- **Manufacturing Utilization %** = Productive Time / Manufacturing Time

Total utilization and operational utilization are defined in E10 as above. Equipment uptime utilization and manufacturing utilization are not specifically defined in E10, but are consistent with E10 and are widely used in the industry. FabTime has also long used a fifth utilization definition based on the breakdown of the E10 Standby state into two sub-states according to whether WIP is waiting:

- **WIP Utilization %** = Productive Time / (Productive Time + Standby-WIP-waiting Time)




FabTime previously referred to manufacturing utilization as simply utilization (and will be renaming the utilization metric in FabTime accordingly going forward). The reason that FabTime uses the manufacturing utilization definition is that we have always believed that this is the utilization that drives cycle time.

## We don't recommend targeting 100% manufacturing utilization because some standby time is needed to keep cycle time under control

If we try to run a tool for 100% of the time that it is available to manufacturing on a long-term basis, we will eventually run into trouble. This is because of variability. It is inevitable that at some point we'll be unable to keep the tool running wafers, maybe even just for a few minutes. There's no operator to load the tool during shift change. The tool was starved because of a lengthy upstream downtime. We held the tool idle for a hand-carry lot. And so on. If we try to run at 100% utilization of manufacturing time, and we lose a few minutes to variability, **we can never make that up**. This means that the queue for that tool will keep growing and growing.

Having a buffer of standby time protects us from this variability by giving the tool a chance to recover from these types of events. This is captured in the queueing formulas that we use to estimate x-factor. As has been shown previously in Issues 6.5 and 22.04, we can estimate x-factor for a one-of-a-kind tool as:

**X-Factor = 1 + [(Utilization / (1 – Utilization)) * Variability Factor]**

Let's look at what happens when we substitute for utilization using the definition of manufacturing utilization:

**Manufacturing Utilization = Productive Time / (Productive + Standby Time)**

Shortening Productive Time to P and Standby Time to S we have:

- Utilization = P/(P+S)

- (1 – Utilization) = [(P+S)/(P+S)] – [P/(P+S)] = S/(P+S)
- X-Factor = 1 + [(P/(P+S))/(S/(P+S))]*Variability Factor = 1 + (P/S)*Variability Factor

That is,

**X-Factor = 1 + (Productive Time*Variability Factor)/Standby Time**

This formulation makes it clear that when standby time is zero, we have:

X-Factor = 1 + (Productive Time * Variability Factor)/0

When we divide by zero, we get infinity. We'll never have truly infinite cycle time because we don't have infinite WIP. But we will see cycle time continuing to rise over time if we don't have any standby time to recover from variability.

**[Side Note:** It may be possible to run the constraint tool in the fab close to 100% utilization by managing the buffer in front of that tool (so that it never starves), while gating starts into the fab to avoid overloading. This must be done carefully because a) the constraint sometimes shifts over time and b) in the presence of reentrant flow, balancing the WIP level in front of the tool is quite complex.]

In general, the larger the productive time relative to the standby time, at a given level of variability, the higher the cycle time will be. This is why most tools need to have some quantity of standby time to recover from variability (as we've been saying in our cycle time class for many years). But this is not the whole story.

## Manufacturing utilization doesn't tell the whole story, because not all standby time is created equal

The above definition of manufacturing utilization conflates two different kinds of standby time. In FabTime these are called **standby-WIP-waiting** and **standby-other**. Standby-WIP-waiting is also called "Standby With WIP" (FPS Dashboard) or "Standby when product is available" (the SEMI E79 Specification for Definition and Measurement of Equipment Productivity). Whatever it's called, it's time that the tool is available, and has WIP waiting to be processed, but is not being run for some reason. Reasons for standby-WIP-waiting include:

- **Lack of Operators:** There is no operator there to load the tool.
- **Hot Lots:** The tool is being held for an expected hand-carry lot.
- **Lot Transport:** The MES says that WIP is available, but the operators can't find it.
- **Batch Loading Rules:** The operator is waiting for more lots before starting a non-full batch.
- **Time Constraints**: WIP is being held at an upstream step until operators are sure it can make it through a time link loop without violating a constraint. (See Issue 22.02: Managing Time Constraints between Process Steps in Wafer Fabs)
- **Setup Minimization Rules:** The operator is waiting for another lot with a matching setup ID to avoid doing a setup on the tool.

Standby-other is time that the tool is available to manufacturing and is not being run because there is no WIP there. Standby-other is also called "Standby No WIP." Reasons for standby-other include:

- **Short-Term WIP Fluctuation**: The is no WIP at this tool because the WIP has been held up somewhere else (e.g. waiting for a down tool).

- **Planned Capacity Buffer:** The capacity plan for this tool calls for some percentage of idle time. The size of this buffer will be influenced by tool granularity. We can't buy 2.6 tools, so we buy three tools. The tool group will have extra idle time unless the mix changes or start rate increases.

It should be noted that the SEMI E10 Specification does not distinguish between these two types of standby time, though it does state that standby time can include periods such as "no operator available." The E79 Specification does not distinguish between types of standby time for the standard OEE definitions. E79 does, however, include the concept of excluding "No Product Time" from productivity loss as part of a supplemental productivity metric called Production Equipment Efficiency (PEE). Under PEE, Equipment Downtime when no product is available is also excluded.

We believe that standby-WIP-waiting time and standby-other should be tracked separately because they have different impacts on cycle time. Standby-WIP-waiting is essentially a capacity loss. It's time that the tool could be running wafers, but is not. Standby-other, in contrast, is (for the most part) a buffer that allows us to recover from variability. We normally want to decrease standby-WIP-waiting time, because it represents lost capacity. However, if we want good cycle time, we may want to increase standby-other, at least by enough to keep us away from the steep part of the operating curve.

**One other point about standby-WIP-waiting**: there is some room for debate about how time is allocated between standby-WIP-waiting and standby-other. Suppose there is one lot waiting, and two tools are available. Should the time be counted as standby-WIP-waiting for both tools, or just one of the tools? If the latter, how do we decide which one? We will defer detailed discussion of these decisions to a future issue (and to our upcoming in-person User Group Meeting).

## WIP Utilization is a metric for driving Standby-WIP-Waiting time to zero

FabTime worked previously with a customer for our cycle time class to define a metric called **WIP Utilization** (see Issue 5.05: WIP Utilization Percentage). WIP Utilization was designed to incentivize the manufacturing team to reduce standby-WIP-waiting time. It was defined as:

**WIP Utilization = Productive Time / (Productive Time + Standby-WIP-Waiting Time)**

The nice thing about WIP utilization is that if we can reduce standby-WIP-waiting time to zero, then WIP utilization will always be 100%. This makes it a cleaner metric for driving improvement than standard manufacturing utilization, where we have "well, keep it below 100%, but don't let it go too low because that's not cost-effective." We always want to drive WIP utilization to 100%.

WIP utilization is available on all FabTime Tool State charts. An example is shown at the top of the next page. WIP utilization, the purple line near the top of the chart, is 100% for the first few shifts, because there is no standby-WIP-waiting time. WIP utilization drops later in the week for shifts that have standby-WIP-waiting time (the dark gray on the chart).

For the April 17th day shift (shown in bold in the table on the next page), nearly all the standby time is standby-WIP-waiting, and the WIP utilization is nearly equal to the manufacturing utilization.

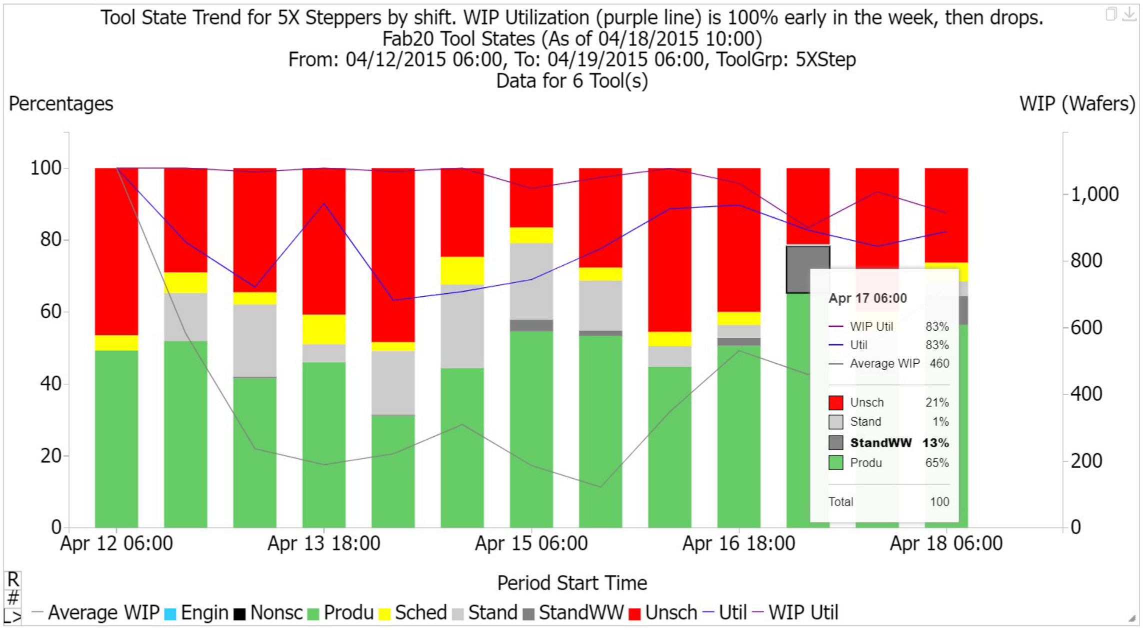



**Table 1. WIP Utilization and Manufacturing Utilization for 5XStepper Tool Group**

| Start Time | Avg WIP | Productive Pct | Stand-WIP-Waiting Pct | Standby Other Pct | WIP Utiliz. Pct | Mfg. Utiliz. Pct |
|---|---|---|---|---|---|---|
| 2015-04-12 06:00 | 1079.26 | 49.28 | 0.00 | 0.00 | 100.00 | 100.00 |
| 2015-04-12 18:00 | 583.31 | 51.92 | 0.00 | 13.40 | 100.00 | 79.48 |
| 2015-04-13 06:00 | 237.67 | 41.60 | 0.44 | 20.09 | 98.95 | 66.95 |
| 2015-04-13 18:00 | 189.58 | 46.04 | 0.00 | 5.02 | 100.00 | 90.16 |
| 2015-04-14 06:00 | 222.04 | 31.13 | 0.30 | 17.75 | 99.04 | 63.29 |
| 2015-04-14 18:00 | 310.45 | 44.44 | 0.00 | 23.22 | 100.00 | 65.69 |
| 2015-04-15 06:00 | 187.10 | 54.63 | 3.29 | 21.23 | 94.32 | 69.03 |
| 2015-04-15 18:00 | 122.28 | 53.38 | 1.44 | 13.96 | 97.38 | 77.62 |
| 2015-04-16 06:00 | 348.43 | 44.79 | 0.05 | 5.67 | 99.90 | 88.68 |
| 2015-04-16 18:00 | 531.92 | 50.60 | 2.22 | 3.59 | 95.79 | 89.70 |
| **2015-04-17 06:00** | **460.07** | **65.30** | **12.99** | **0.60** | **83.41** | **82.78** |
| 2015-04-17 18:00 | 566.39 | 42.94 | 3.01 | 8.91 | 93.45 | 78.27 |
| 2015-04-18 06:00 | 721.05 | 56.46 | 8.06 | 4.03 | 87.51 | 82.37 |

Increasing WIP utilization by reducing standby-WIP-waiting time can help with cycle time improvement efforts. In the above example, cycle time per visit would likely have been better on April 17$^{th}$ if that 12.99% standby-WIP-waiting was converted to standby-other. However, WIP utilization alone won't help us to distinguish between the left-most shift (April 12, 6:00), where there is no standby time, and the next few shifts, where there is standby time. WIP Utilization is high for all those shifts. Notice that the WIP is much higher for the first shift, however, and then decreases over the next few days, as the amount of standby-other time increases from zero.

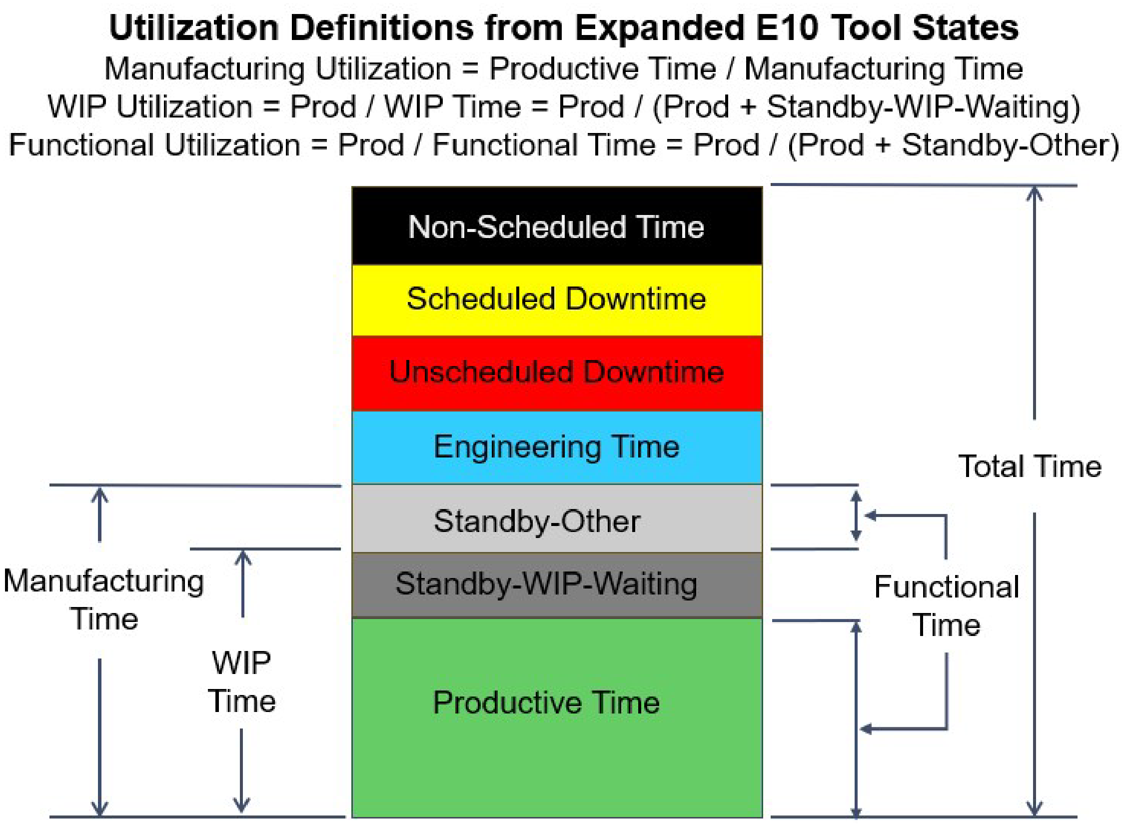
## To understand what cycle time we can expect from a tool group, we need a utilization definition that is based on standby time when no WIP is waiting

We propose a supplemental utilization metric, which we are provisionally calling **Functional Utilization**. This metric is based on productive time relative to the time that the tool is available to manufacturing and not lost to operational issues such as lack of operator availability or holding tools for hot lots. We'll define:

**Functional Time = Productive Time + Standby-Other**

And:

**Functional Utilization = Productive Time / Functional Time**




For consistency, we'll also define:

**WIP Time = Productive Time + Standby-WIP-Waiting Time**

And:

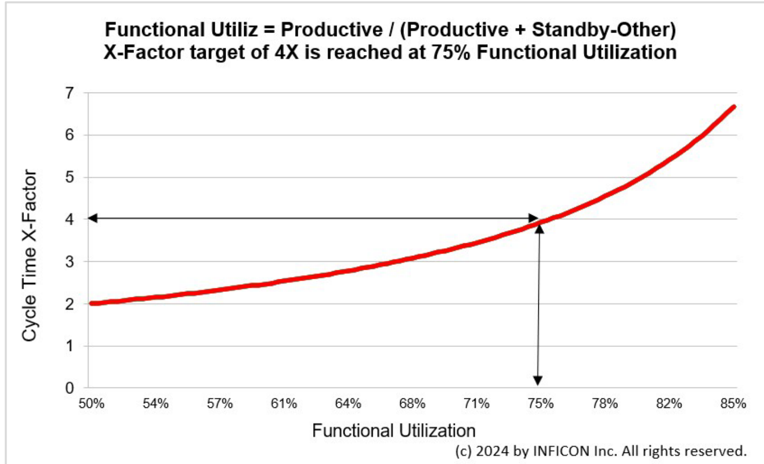
**WIP Utilization = Productive Time / WIP Time**

This is the same definition used for WIP utilization above. We've just added a name for the time bucket that consists of productive time plus standby-WIP-waiting time.

But it's functional utilization that's new, and that drives cycle time. When functional utilization is 100%, cycle time will likely be very high for a given tool group.

## So how would we use functional utilization?

We can use functional utilization as we used manufacturing utilization in queueing formulas. When we do this, we're treating the standby-WIP-waiting time as a capacity loss, instead of treating it like standby time. We can thus use the FabTime Operating Curve Spreadsheet to set targets for the required functional utilization to reach a given cycle time goal. An example for a one-of-a-kind tool is shown below.




We can also use functional utilization as an overall indicator for how good a cycle time a fab will be likely to achieve. We would expect a high value for functional utilization for the most expensive bottlenecks tools. But if most of the tools in the fab have a functional utilization above 85%, it's likely that overall cycle times will be high (especially for fabs that lack tool redundancy). If many tool groups in the fab have low values for functional utilization, the fab is unlikely to be cycle time constrained (but may struggle with profitability). It's possible that increases in average functional utilization across the full fab toolset would correlate with increased cycle time, but we would want to see actual data on this.

We can also compare actual functional utilization over time to the planned utilization for a tool group. If functional utilization is much higher than planned utilization, this suggests that the fab is losing significant capacity due to operational issues or poor availability performance. We're not getting as much standby time that we can use as a buffer for variability as we expected.

Returning to the Tool State Trend example shown above, in the WIP utilization section, let's look at the additional information that functional utilization (right-most column) gives us about expected cycle time.

**Table 2. Functional Utilization added to WIP and Manufacturing Utilization Columns**

| Start Time | Avg WIP | Productive Pct | Stand-WIP-Waiting Pct | Standby Other Pct | WIP Utiliz. Pct | Mfg. Utiliz. Pct | Functional Utiliz. Pct |
|---|---|---|---|---|---|---|---|
| 2015-04-12 06:00 | 1079.26 | 49.28 | 0.00 | 0.00 | 100.00 | 100.00 | 100.00 |
| 2015-04-12 18:00 | 583.31 | 51.92 | 0.00 | 13.40 | 100.00 | 79.48 | 79.48 |
| 2015-04-13 06:00 | 237.67 | 41.60 | 0.44 | 20.09 | 98.95 | 66.95 | 67.43 |
| 2015-04-13 18:00 | 189.58 | 46.04 | 0.00 | 5.02 | 100.00 | 90.16 | 90.16 |
| 2015-04-14 06:00 | 222.04 | 31.13 | 0.30 | 17.75 | 99.04 | 63.29 | 63.68 |
| 2015-04-14 18:00 | 310.45 | 44.44 | 0.00 | 23.22 | 100.00 | 65.69 | 65.69 |
| 2015-04-15 06:00 | 187.10 | 54.63 | 3.29 | 21.23 | 94.32 | 69.03 | 72.02 |
| 2015-04-15 18:00 | 122.28 | 53.38 | 1.44 | 13.96 | 97.38 | 77.62 | 79.27 |
| 2015-04-16 06:00 | 348.43 | 44.79 | 0.05 | 5.67 | 99.90 | 88.68 | 88.76 |
| 2015-04-16 18:00 | 531.92 | 50.60 | 2.22 | 3.59 | 95.79 | 89.70 | 93.38 |
| **2015-04-17 06:00** | **460.07** | **65.30** | **12.99** | **0.60** | **83.41** | **82.78** | **99.09** |
| 2015-04-17 18:00 | 566.39 | 42.94 | 3.01 | 8.91 | 93.45 | 78.27 | 82.81 |
| 2015-04-18 06:00 | 721.05 | 56.46 | 8.06 | 4.03 | 87.51 | 82.37 | 93.34 |

Here we see that WIP (a proxy for cycle time) is decreasing as functional utilization decreases and increasing as functional utilization increases. Manufacturing utilization follows a similar trend early in the week, but misses the impact of the April 17th morning shift (shown in bold), where the functional utilization rises to 99%.

## We likely wouldn't use functional utilization as a KPI to drive daily performance

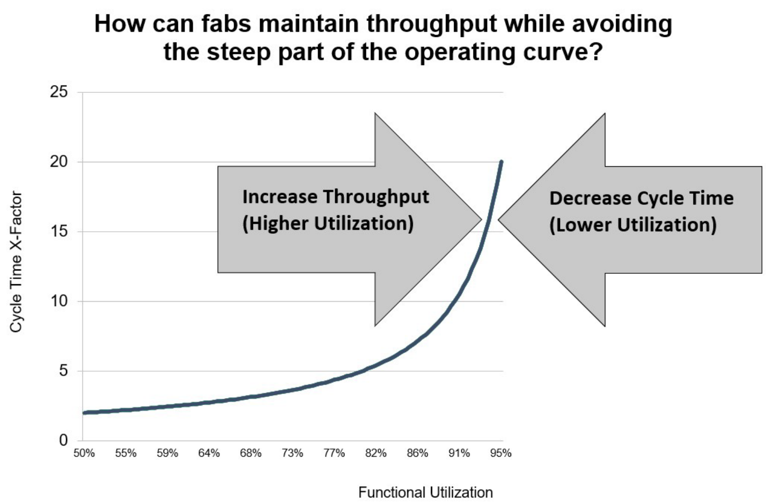
Unlike WIP utilization, functional utilization does suffer the problem of lacking a clear target. We don't want it to be 100% (over the long term, at least), because that means that cycle times will be very high. But we equally don't want it to be zero, because then we aren't processing any wafers. In practice, we'd like functional utilization to be as low as possible while still meeting throughput goals.

A good target for functional utilization for non-bottleneck tools that have redundancy is probably 85%. For one-of-a-kind tools, 75% might be better. For true bottleneck tools, the most expensive tools in the fab, we may want to push that functional utilization closer to 90 or 95%. But if we want to improve cycle time, we want to continuously drive those functional utilization numbers down.

All of this makes functional utilization problematic as a primary day-to-day metric. We view it instead as a supplemental metric that we can use to understand why cycle time is higher than anticipated, and to drive cycle time improvement efforts.

## How can we decrease functional utilization (increase the standby time buffer) without sacrificing throughput?

The goal here is to maintain throughput, but convert other losses into standby time when there is no WIP waiting. There are several ways to do this:




- **Improve equipment uptime** by reducing scheduled and especially unscheduled downtime.
- **Reduce the quantity of engineering time** (or at least schedule it to take place when no WIP is waiting).
- **Reduce standby-WIP-waiting time**. Here are a few ideas (most taken from our cycle time management course):
    - Cross-train operators and stagger break schedules, to reduce delays due to lack of operators.
    - Reduce the number of hot lots, especially hand-carry lots.
    - Run batch tools under a greedy policy.
    - Ensure that setup avoidance policies only apply when there is WIP with a matching setup ID that is already in queue.
    - Use a Smart Scheduler that can handle time constraints between process steps.
- **Process the same quantity of wafers in a shorter time to reduce productive time by eliminating speed losses and other non-value-added time** (e.g. processing rework wafers

or waiting for an operator to unload a tool) that gets recorded as productive time. That is, get the same number of wafers through the tool in a smaller amount of productive time.

## Isn't this just OEE?

Not quite. The recommendations in the previous section do tie to OEE loss factors. However, standard OEE does not include a loss factor specific to standby-WIP-waiting time. (Or rather, it includes a loss factor for all standby time, whether WIP is waiting or not.) PEE does have a loss factor for this, but it also only considers downtime a loss if there is WIP waiting. Downtime when there is no WIP waiting is certainly better than downtime with WIP waiting, and PEE incentivizes the maintenance organization to do PMs when no WIP is waiting, which is a good thing. But for cycle time, we'd still rather have that time as standby time than downtime, so that we can use it to recover from variability.

PEE is accounting for time that a tool isn't busy because there's no WIP there, and saying "OK, that's not a loss factor." But what it isn't saying is that for optimal cycle time performance we *require* some time that the tool is in a standby state and has no WIP waiting.

OEE and PEE are important frameworks for working to eliminate capacity losses. PEE extends upon OEE by understanding that there is a difference in standby time depending on whether WIP is waiting. **But neither OEE nor PEE is focused specifically on reducing cycle time.** To reduce cycle time, we feel that there is benefit in this additional metric of functional utilization, which we try to drive down for cycle time improvement.

Another way to look at this is just to say that from a cycle time perspective, we don't want standby-other to be driven to zero. Most of the time, at least, we need some positive buffer of standby time, ranging from 5% to 25% of the functional time.

## Is functional utilization the best name to use for this metric?

We considered other names for functional utilization. We thought of:

- **Effective utilization**. This was rejected because "effective" is similar to and easily confused with "efficiency."
- **Line-limited utilization**. This one had some strong internal advocacy but was rejected because we also wanted to have a corresponding time bucket on the E10 Tool State-based chart, and the name "line-limited time" to represent productive time + standby-other didn't seem accurate.
- **Constrained utilization**. This one was too easily confused with constraint terminology (the top bottleneck in the fab).
- **Buffered utilization**. This one just didn't sound right.
- **Net utilization**. This was our second choice, but we found "functional" to be more descriptive.

We also considered whether the metric should just be the quantity of standby-other time. That is, what we want to encourage is having a buffer of standby time when no WIP is waiting, so why not just use that as the metric? However, we ultimately decided that having functional utilization as a parallel companion to WIP utilization had value. We also liked the idea that functional utilization drives the function that makes up the operating curve. Finally, one can think of standby-WIP-waiting time as time that is dysfunctional and is excluded from functional time.

We spent quite a bit of time discussing this metric internally. These discussions led to our decision to form a Metrics Team within INFICON to align our definitions of other metrics between the FabTime and FPS software products. The INFICON Metrics Team (led by Jennifer Robinson) welcomes feedback from all of you and looks forward to the metrics alignment panel that we will be hosting at our upcoming in-person User Group Meeting.

## Conclusions

There are various utilization definitions relative to the SEMI E10 tool states that are used for different purposes. The FabTime software historically used manufacturing utilization, which is productive time divided by the time that a tool is available to the manufacturing organization. But what we've come to realize is that this definition isn't sufficiently granular to fully explain cycle time. That's because not all standby time is created equal.

We normally want some standby time as a buffer against variability, to keep us away from the steep part of the operating curve. Standby time when WIP is waiting for the tool is more a capacity loss than a buffer. Its presence increases cycle time. Not only that, standby-WIP-waiting is a hidden capacity loss, because it's conflated with standby time when WIP isn't waiting.

In this article, we have proposed a supplemental definition of utilization that we are calling functional utilization. Functional utilization looks at productive time out of the usable manufacturing time, which is recorded as productive time plus standby time with no WIP waiting. This is the utilization definition that truly drives cycle time. As functional utilization approaches 100%, cycle time will likely be very high. While we don't, of course, want functional utilization to be zero, we have included suggestions for reducing it to a reasonable target range.

## Closing Questions for Subscribers

Do any of you use a metric like the one that we're calling functional utilization? If so, what do you call it? Do you distinguish between standby-WIP-waiting and standby-other (or standby-no-WIP)? Do you have targets for maximum utilization (or a standby time buffer) for different types of tools?

# Subscriber List

**Total number of subscribers: 2911**

**Top 20 subscribing companies:**

- onsemi (137)
- Intel (135)
- Micron Technology (113)
- Infineon (112)
- Analog Devices (105)
- Microchip Technology (97)
- GlobalFoundries (82)
- NXP (72)
- STMicroelectronic (66)
- Skyworks Solutions (64)
- Texas Instruments (60)
- Western Digital (53)
- Seagate Technology (52)
- X-FAB (52)
- Wolfspeed (51)
- Carsem M Sdn Bhd (38)
- Qualcomm (35)
- Applied Materials (34)
- Tower Semiconductor (34)
- ASML (33)

**Top 4 subscribing universities:**

- Arizona State University (7)
- Ben Gurion University of the Negev (5)
- Ecole des Mines de St. Etienne (EMSE) (3)
- Nanyang Technological University (3)

**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 INFICON 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 the FabTime website. To request the current password, e-mail your request to Jennifer.Robinson@inficon.com. To subscribe to the newsletter, please visit the FabTime website. To unsubscribe, reply to your newsletter e-mail with "Unsubscribe" in the subject or visit this Unsubscribe Page (fabtime.com). INFICON will not, under any circumstances, give your e-mail address or other contact information to anyone outside of INFICON without your permission.