FabTime Cycle Time Management Newsletter

Volume 20, No. 4 August 2019

Information

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

Publisher: FabTime Inc. FabTime sells cycle time management software for wafer fab managers. Features in development right now include new elapsed cycle time move detail and operation cycle time detail charts, as well as more detailed breakdown of data on other cycle time-related charts.

Editor: Jennifer Robinson

Contributors: David Carmichael from TowerJazz Semiconductor and Justice Stiles from Infineon Technologies

Keywords: Metrics and Goals; WIP Management; Semiconductor Industry

Table of Contents

Welcome

Welcome to Volume 20, Number 4 of the FabTime Cycle Time Management Newsletter. We hope you're all having a great summer. We have no community announcements in this issue, but we have shared links to a couple of recent news stories that we thought would be of interest to our subscriber community. Our software tip of the month is about viewing the distribution of WIP across the line, with additional detail about the breakdown of that WIP.

We have two extensive and thoughtful subscriber responses to the main topic of the previous issue: the metric WIP Hours (hours of WIP in queue per tool). As these responses have made the subscriber discussion both lengthy and technical, we have chosen to include a brief main article. Inspired by the recent burst of news about the 50th anniversary of the moon landing, we share some commentary about the impact of the US space program on the semiconductor industry. We welcome your feedback.

Thanks for reading – Jennifer

FabTime

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

Community News/Announcements

Two Recent Articles of Interest

While we have no formal announcements in this issue, we recently ran across two news stories that we thought would be of interest to our newsletter audience.

First up, via an AP report at Time.com:

"Codebreaker and computing pioneer Alan Turing has been chosen as the face of Britain's new 50 pound note, the Bank of England announced Monday.

Governor Mark Carney said Turing, who did ground-breaking work on computers and artificial intelligence, was "a giant on whose shoulders so many now stand."

During World War II Turing worked at the secret Bletchley Park code-breaking center, where he helped crack Nazi Germany's secret codes by creating the "Turing bombe," a forerunner of modern computers. He also developed the "Turing Test" to measure artificial intelligence."

We thought it was neat to see a computer scientist honored in this manner.

Second, as people who started our own careers working in wafer fab simulation modeling, Frank and Jennifer were both pleased to see simulation highlighted in a recent Wall Street Journal piece by Andy Kessler called Better Living Through Simulations:

Here's a snippet: "Simulations helped create the computer industry. The first U.S. computer, the Eniac, ran statistical simulations for hydrogen-bomb blasts—not something you can test in the lab. Every chip in your personal computer and phone is simulated well before it is manufactured in billion-dollar fabrication plants. Simulating weather patterns has led to better though never perfect forecasts."

Incidentally, Jennifer shares articles about business management, the semiconductor industry, and productivity improvement on her LinkedIn feed. Connect with her here:

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

FabTime User Tip of the Month

View the Distribution of WIP Across Your Line

Something that people often ask about in software demos is whether FabTime can show them where their WIP is in the line. These days, we recommend using the Stacked WIP Pareto chart for this purpose. The stacked version of the WIP pareto chart allows you to not only see the total WIP at each stage of the line, but also the type of WIP.

To see this information, generate the Stacked WIP Pareto chart from the chart list. You'll see both a "Slice" control and a "Cross" control. Both default to "Area." To see how your WIP is distributed across the line, you'll most likely want to change the "Slice" control to either "Segment" or "SubSegment." In most installations, segment is mapped to a consecutive sequence of steps in the line that is about a week long. Thus, a process flow with an eight-week cycle time would be broken into eight segments. Subsegments are smaller, representing some logical grouping of a linear group of steps. Fabs with short process flows may be able to slice the WIP pareto by operation. The important thing is to slice the pareto chart by a variable that is granular enough to give you a sense of the way the WIP is currently staged across the line.

You may need to modify the "Sort" controls to ensure that your segments (or subsegments or Operations are in order from start to finish. To do this, set the first sort drop-down to "N/A", and the second to either the attribute description (e.g. "SubSegment (Description)") or the attribute sequence. You can toggle between these to see which one looks right. Usually, you can just click the link for "Default Sort" to get the chart in object order.

Once you have the pareto at the right level of granularity, you can change the "Cross" drop-down to stack the bars by your attribute of interest. You might stack by product, priority, owner code, etc. An example of WIP by subsegment stacked by product is shown below. We can quickly see WIP stacking up in segment 6 (subsegments starting with 6) and drill down for more information.

You can also vary this this chart by filtering to only see lots on hold, lots in queue, rework lots, inactive lots (lots that have been at their current operation for some time period), etc.

We hope that you find this tip useful.

If you have questions about this item, or any other FabTime software questions, just use the Feedback form inside FabTime's software. Subscribe to the separate Tip of the Month email list (with additional discussion for customers only). Thanks!

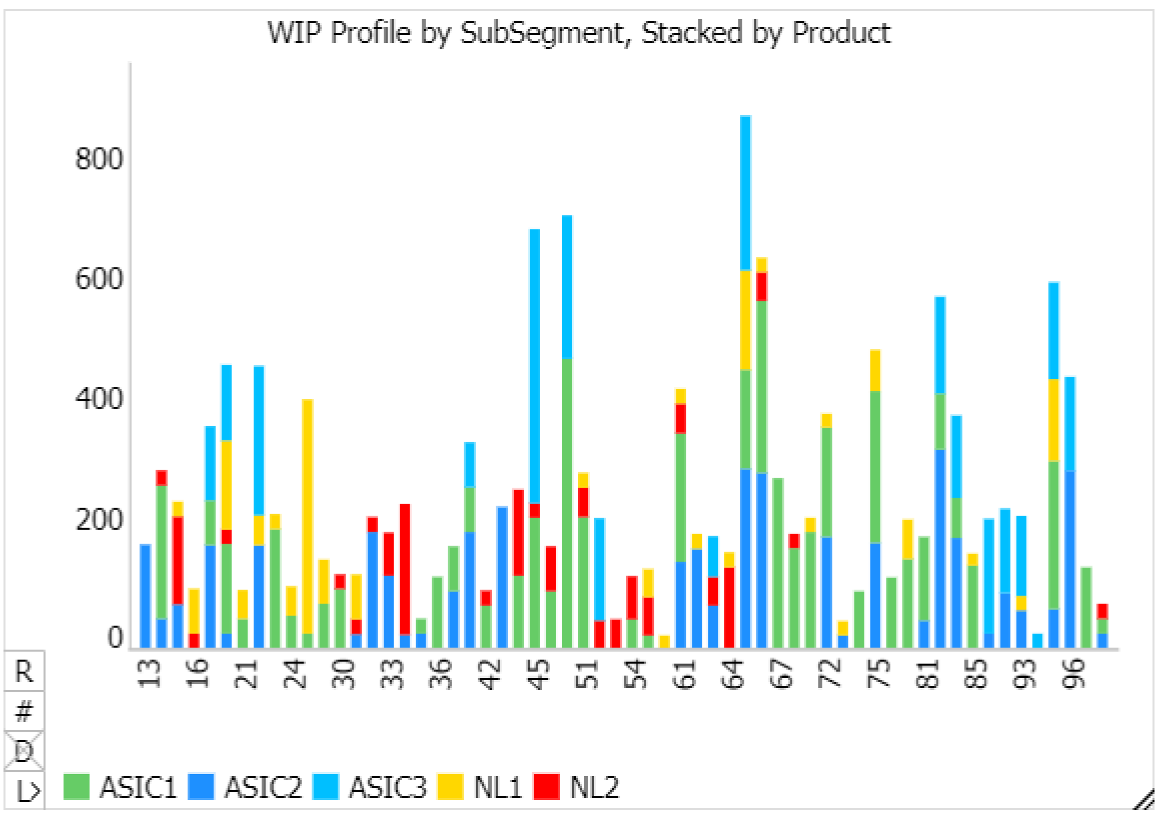

Figure 1. Example of WIP Profile by SubSegment, Stacked by Product

Subscriber Discussion Forum

Issue 20.03: WIP Hours

We received two detailed responses to the metric introduced in Issue 20.03, WIP Hours. We are including them below, with responses inline from FabTime (shown with gray highlighting).

David Carmichael from TowerJazz Semiconductor said: "Your attempt to take on "WIP hours" is a very brave one. I don't think WIP hours is a particularly good name for the concept as you are really using a UPH value (that you probably don't have) to allocate lots to tools, using algorithms you probably don't have either.

FabTime: Regarding this being a UPH metric, we do fill in with planned process time where UPH data is not available. For some sites, we think that will often be the case. We're not sure that it would make sense for us to change the name of WIP Hours at this point, having been talking about it like this for a while with our customers, but we do take the point, and would welcome other suggestions.

In our MES we have a simplified form of model for minimum processing time on a tool. We can define the time to load and unload the tool as well as the actual processing time. This can be on a per-lot or per-wafer basis and can even vary from tool to tool for the same recipe.

Even with all of this we still fall short because of all the unknowns:

1. When will a tool come back up or go down? We do not even know the latter if there is a PM scheduled on the tool, and we never know the former.

FabTime: This is always a problem in trying to project things forward in a fab. We handle this in the WIP Hours calculations by not trying to predict it at all. The WIP Hours metric just looks at the WIP in queue for the tool, whether the tool is up or down. The WIP Hours per Effective Tool metric doesn't calculate WIP Hours for down tools at all. It's basically a point in time metric, so we don't even try to say when the tool will come back. The "real" answer is of course somewhere in between the two values.

2. In a chamber-based tool can the lot be run if one of the chambers is down? In the case of some tools this is not an issue and at the other end of the spectrum some recipes require specific capability on at least one tool chamber to be able to run at all.

FabTime: We only have a few sites that have recipe-chamber-requirements data to the point where we can know the impact of a chamber being down, and even in those cases the impact of the change is a yes/no flag of whether the recipe can be run at all, not the impact on UPH of the chamber being down.

For the WIP Hours, we left some flexibility in the design so that "tool % up (effective)" is a percentage variable rather than a 0/1 flag. We were thinking that we would add site-specific hooks so that sites that *can* quantify partial loss of capability – or the loss of capability for a recipe due to a chamber being down -- could override this value. Then that number is taken into account in the "WIP Hours per Effective Tool" computation. It's not perfect, but it may still be useful.

3. We have developed a 10-step look-ahead with a recommendation for each tool to be used at every step. This is only an approximation of course as tools can not only be up or down but their capabilities and qualifications can change in a very dynamic way. In addition, some tools do not actually have the ability to run a particular tool recipe either because of some limitation of the tool or the fact that the recipe has never been loaded onto that tool even though it appears to have the required capability. There are other very dynamic constraints we have to consider including max wait times between process

steps and the desire of the fab to run identical lots sequentially despite their very different Critical ratios.

FabTime: We did experiment with this type of look-ahead scheduling with one of our customers. But as you note, there are a LOT of complexities in practice. We don't expect that we would get to that level of granularity with the WIP Hours metric, given that different customers are at different levels of detail in their data.

Historical analysis can be used to estimate some of these things, but it must be very extensive, is very complex and not very accurate. The human factor must always be considered too, and this randomizes some of the results in unexpected ways.

I wish you luck modelling all of this in sufficient detail and ensuring your systems have up-to-date information about all of these factors. I hope we can get some ideas about how to improve from your experience."

FabTime: We appreciate your input regarding how you are handling similar issues and hope that working together across FabTime's customers and subscribers, we can make these sorts of metrics useful in improving fab cycle time.

Justice Stiles from Infineon Technologies added: "As always, interesting discussion in the newsletter. Since you actively solicited feedback on the WIP Hours metric, and it's a topic I've put a fair amount of thought into, I thought I would respond. As it happens, we implemented exactly this metric at our site as part of a larger fab pilot system that we used for ensuring wafers stop in safe locations during the run-up to a fab shutdown. The purpose was exactly what you describe with your current project, the ability to identify ephemeral bottlenecks. As I'm sure you're aware, you can't just stop a wafer at an arbitrary position in the line for a prolonged period without a penalty in the form of rework or diminished yield. We built the system so we did not attempt to move wafers out of a safe stopping location if they had no chance of making it to the next safe location due to said bottleneck.

A question I did not see listed among your open questions: How do you plan to normalize tools and WIP Hours in cases where the tools of a given tool set run multiple operations, but not all tools are qualified for the same subset of operations? To illustrate:

Say we have three etch tools Etch1 Etch2, and Etch3.

Additionally, suppose we have six etch operations that run on this tool set, call them operations A through F.

Suppose not every tool in the set runs all six operations.

- Etch1 runs all six operations
- Etch2 doesn't run operations A and B
- Etch3 only runs operations B, C and D

Now assume there are 500 wafers queued in front of this toolset in 50 wafer lots (10 lots).

- Three lots require operation A
- Two lots require operation B
- Two lots require operation C
- One lot requires Operation D
- One lot requires Operation E
- One lot requires Operation F

Further assume that for the purpose of WIP Hours each of the operations requires a non-trivially different amount of processing time.

Given the scenario above, how do you go about normalizing tools and calculating WIP Hours? Potentially further complicating matters, quite often a tool can run an operation for some lots but not others as restrictions are imposed on a tool that affect some products but not others.

For what it's worth, our approach was to slice by operation and use the notion of fractional tools for this calculation. The basics of the algorithm are as follows:

For each operation:

1. Take all tools that could process an operation of interest and find all lots queued for the operation.

2. Find all other tools that could process any of those same lots.

This gives us our initial tool set, a theoretical maximum number of tools that could be available to distribute the workload for this operation between.

> For each tool in the initial toolset for this operation:
>
> 1. Sum the Process Time (PT) of all lots queued before that tool (whether the lot requires our operation of interest or not, this is our TotalListPT).
>
> 2. Sum the PT of the lots that require the operation of interest (our InterestListPT).
>
> 3. Take the fraction of InterestListPT / TotalListPT.
>
> 4. The resulting fraction is the current proportional tool dedication to our interest list. If every lot queued before a tool requires the operation of interest, the number is 1, if half require the operation of interest then the returned fraction would be 0.5.
>
> Note that this fractional dedication applies to only one tool. Since the other tools in the initial list can process other operations, the lots queued before those tools will be different; each tool will most likely yield a different fractional dedication for our operation of interest.
>
> Repeat this process for each tool that can process the operation(s) of interest to get the proportional tool dedication per operation.

Repeat the entire above process for all operations of interest.

This process was tailored to a more specific goal than the generalized reporting you're looking to create, but the approach may be of use. Sliced slightly differently, the same data generated by the algorithm allowed us to determine which tools we were most reliant on for a given operation and ipso facto would create the greatest issues if the tool were taken down or were to go down unexpectedly. In the example I gave above, it would become apparent only Etch1 can run operation A, or if you were to lose Etch1 or Etch2 you would lose 50% of your capacity for operations E and F. If these tools were it to be idled prematurely, or were one of them to go down, the fab would be left with a potentially impassible bottleneck, even though the tool would appear to be only one of a set of three.

We also wrangled with the question "Will the calculations be able to look forward in time to include lots that are scheduled to arrive?" We experimented with this but were never satisfied with the results as it can be very difficult to predict with accuracy what exact lot will be queued before a given tool or operation very far in the future. There are just too many variables that cannot be controlled (tools go down, a lot requires rework, a lot goes on hold etc.) In the end we declared the attempt to predict what lots would be at a tool in the future quixotic and rather decided to only look at the current state of affairs with our tools and update the data frequently. If you can find a way to reliably make this prediction, my hat goes off to you and your team! For our purposes, we could sum the current expected queue time at each operation between the current operation and the next safe operation, use a time buffer (say we don't move if the total current queue time is within 120% of the remaining time), and the result actually worked extremely well under both high and lower fab loading conditions.

Hopefully, some of this can be of use to you and your team, as your goal is broader than ours. I think the new report will be a very useful addition to FabTime.

FabTime: Thank you for the detailed notes! We also struggled with the question of including future arrivals, and decided not to tackle that, as it would likely just introduce noise into the calculations and make them harder to validate.

On the question of how to handle similar-but-not-identical tools, we believe we have that covered in our calculations. We think that our approach is equivalent to what you describe above, but we'd have to work out a detailed example to be sure. In our case, we're going to have a second "details" chart that shows lot/tool pairs – it will show all WIP that matches the filters, and then for each lot there will be one row for each qualified tool. The WIP hours for the lot is calculated by averaging the WIP hours across its qualified tools, and then dividing by the number of qualified tools – this fractional portion of WIP hours is then allocated across the qualified tools.

We're glad that your approach has worked well for your purpose of better managing WIP during shutdowns, and we hope that other subscribers will benefit from reading about this method.

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

How the Space Program Launched the Semiconductor Industry

As we have quite a bit of technical detail in this month's subscriber discussion forum, we thought that a brief main article would be appropriate.

July 20th, 2019 marked the 50th anniversary of the first time that humans walked on the moon. The quest for the moon landing, announced by John F. Kennedy in 1961, was accomplished through eight years of technological and managerial effort. It is recognized today as one of the United States' greatest achievements.

What is less well known is the impact that the space program had on the technology industry, in terms of both hardware and software. The space program in essence launched the semiconductor industry.

This piece draws extensively on recent news stories. Extended quotations are indented for clarity.

Hardware

Getting ahead in the space race required computing technology. The money that the US government was willing to invest dramatically accelerated the pace of development of that technology, changing the world.

Writing in the Wall Street Journal on July 14th, **Robert Lee Hotz** discussed the computers that were the brains of the Apollo lunar landers. He says: "These moonshot machines were the world's first general-purpose, portable, digital computers, the first to fly and the first on which human lives directly depended."

Hotz also describes the influence of the space program on the development of computer chips, saying:

> "It was Mr. (Eldon) Hall who gambled on using a then-untried device called an integrated circuit to make a computer small enough to fit in a space

capsule, robust enough to survive a Saturn V rocket launch, and fast enough to monitor or control 200 spacecraft systems at the same time.

At his urging, the Apollo program became the first and single largest consumer of the semiconductor chips, buying a million or more of them, some 60% of all the integrated circuits produced in the U.S. between 1962 and 1967, according to Mr. Hall's purchasing records. The first computer chips tested by MIT cost $1,000 each. By the time astronauts landed on the moon, the price had dropped to $15 apiece, his records show. It set a pattern of innovation, quality control and price-cutting that persists in the computer business to this day. "It kicked off the integrated-circuit-technology industry," says Mr. Hall, 96, who now lives in Florida. "That was the creation of Silicon Valley.""

A 2018 article by **Phil Goldstein** in FedTech Magazine, How the Government Helped Spur the Microchip Industry, goes into more detail about how the needs of the space program affected the nascent semiconductor industry. He says:

> "According to (a **1988 paper by Anna) Slomovic**, "two government procurement decisions were responsible for moving integrated circuits into large-scale production."
>
> First, in 1962, NASA announced that its prototype Apollo guidance computer would use integrated circuits from Fairchild. Not long after that, the Air Force announced the use of integrated circuits in the Minuteman II missile guidance package... Fairchild was a major supplier (to NASA), shipping about 100,000 devices for the Apollo space program in 1964 alone."

The semiconductor industry was already in place and would almost certainly have advanced regardless of the space race. But there's no question that the boost provided by the US government kicked things into a higher gear. It's interesting (if difficult) to imagine what the world might be like today if semiconductors had developed more slowly.

Software

Similarly, putting people on the moon required software. Programs were needed to operate the new technology. Some of the earliest programmers (including women) worked on the space program.

In another WSJ article on July 14th, Robert Lee Hotz discusses the importance of the brand new field of software development to the space program. He recounts an incident in which the entire success of the project depended on software, writing:

> "Five times the onboard computer signaled an emergency like none Armstrong and crewmate Buzz Aldrin had practiced.
>
> In that moment, the lives of two astronauts, the efforts of more than 300,000 technicians, the labor of eight years at a cost of $25 billion, and the pride of a nation depended on a few lines of pioneering computer code."

This was revolutionary.

Hotz also discusses the way that the space program started out relying mainly on hardware, but gradually shifted to also rely upon software. Once people discovered that "Code was cheaper, more adaptable and, most important, weightless" they started using it more broadly in the project. As we well know today, there was no turning back.

How Far Have We Come?

Everyone who works in the semiconductor industry knows about the rapid pace of technological improvements in computer chips. And everyone who owns a smart phone knows how dramatically cell phone technology has improved in recent years. It's still instructive to consider how minimal the technology used by NASA was compared with the tools we have today.

A July 16th article by **Alexis C. Madrigal** in The Atlantic notes that "if you compare the computing power that NASA used (in 1969) with any common device, from a watch to a greeting card to a microwave, it induces technological vertigo. Michio Kaku, the physicist and popular author, put it like this: "Today, your cell phone has more computer power than all of NASA back in 1969, when it placed two astronauts on the moon."

Looking at the changes in software, Robert Lee Hotz says: "The Apollo computer eventually required about 145,000 lines of code in all, compared with about 62 million lines of code required today to operate Facebook and more than two billion lines of code for Google." The efficiency with which the Apollo systems were coded is a marvel.

And, in another piece for FedTech Magazine, Phil Goldstein summarizes more detailed performance comparisons from various sources:

> "In terms of memory, the ACG (Apollo Guidance Computer) held "2,048 words of erasable magnetic core memory and 36 kilowords of read-only core memory, with a cycle time of 11.72 microseconds," ExtremeTech reports. That core memory works out to 32,768 bits of RAM or 72KB (equal to 589,824 bits) of ROM.
>
> How does that compare to a modern smartphone? As Cult of Mac notes, an iPhone with 4 gigabytes of RAM (that's 34,359,738,368 bits) has more than 1 million times more memory than the AGC, and a 512GB iPhone has 7 million times more memory.
>
> The AGC did not have a powerful processor by today's standards, operating at a speed of 0.043 megahertz. University of Nottingham computer science professor Graham Kendall writes for The Conversation that the processor in the latest iPhone is estimated to run at about 2490 MHz, meaning it has over 100,000 times the processing power of the AGC."

It's hard to even wrap one's head around 7 million times more memory, isn't it? Maybe in 2069 our grandkids will look back and try to imagine a communication device that only has 512GB of memory.

Conclusions

The recent coverage of the Apollo moon landing has been widespread and varied. There are a host of stories to be told, about the people who participated and the people who were awed by the accomplishment. But as admittedly geeky members of the semiconductor industry, we were struck by the technological aspects of the story. The space race played a major role in setting the world on a high-tech trajectory. This path has led us to profound advances in transportation, communication, medicine, and more. Thanks, NASA (and associated supplier) engineers!

Further Reading

Phil Goldstein, "The Tech Behind Apollo 11's Guidance Computer," *FedTech Magazine*, July 18, 2019.

Phil Goldstein, "How the Government Helped Spur the Microchip Industry," *FedTech Magazine*, September 11, 2018.

Robert Lee Hotz, "An Apollo Spacecraft Computer Is Brought Back to Life," *The Wall Street Journal*, July 14, 2019.

Robert Lee Hotz, "Apollo 11 Had a Hidden Hero: Software", *The Wall Street Journal*, July 14, 2019.

Alexis C. Madrigal, "Your Smart Toaster Can't Hold a Candle to the Apollo Computer," *The Atlantic*. July 16, 2019. /

Anna Slomovic, "Anteing Up: The Government's Role in the Microelectronics Industry," December 16, 1988.

Subscriber List

Total number of subscribers: 2677

Top 20 subscribing companies:

- ON Semiconductor (212)
- Infineon Technologies (144)
- Micron Technology, Inc. (118)
- Intel Corporation (111)
- GlobalFoundries (93)
- Maxim Integrated Products, Inc. (87)
- NXP Semiconductors (77)
- Carsem M Sdn Bhd (69)
- Microchip Technology (69)
- Skyworks Solutions, Inc. (65)
- STMicroelectronics (62)
- Western Digital Corporation Inc. (59)
- Texas Instruments (54)
- Seagate Technology (52)
- X-FAB Inc. (49)
- TDK (47)
- Analog Devices (40)
- Zymergen (34)
- Cree, Inc. (33)
- Honeywell (30)

Top 5 subscribing universities:

- Arizona State University (8)
- Virginia Tech (7)
- Ben-Gurion University (6)
- Ecole des Mines de Saint-Etienne (EMSE) (6)
- Nanyang Technological University (6)

New companies and universities this month:

- Aerospace Semiconductor
- Amazon
- General Dynamics Mission Systems Canada
- Magneti Marelli
- MHP Management
- Qimonda
- Redlen Technologies

Sampler Set of Other Subscribing Companies and Universities:

- Axcelis Technologies (1)
- CSMC (1)
- DEE - Politecnico di Bari (1)
- Dresden University of Applied Sciences (1)
- Elettranova Engineering (1)
- Elmos Semiconductors AG (6)
- Feng Chia University (1)
- FlipChip International (8)
- Industrial Ventilation, Inc. (1)
- InPress Media Group, LLC (1)
- Micross Components (1)
- Powerex, Inc. (1)
- Productivity Partners Ltd. (1)
- Semtech (3)
- Silevo/Solar City (2)
- SV Microwave (1)
- TR Control Solutions UK (1)
- Truesense Imaging (9)
- Ulvac (1)
- Winbond (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 FabTime or its products by any individual or his or her company.

There is no charge to subscribe and receive the current issue of the newsletter each month. Past issues of the newsletter are currently only available to customers of FabTime's web-based digital dashboard software or cycle time management course.

To subscribe to the newsletter, send email to newsletter@FabTime.com, or use the form at www.FabTime.com/newsletter-subscribe.php. To unsubscribe, send email to newsletter@FabTime.com with "Unsubscribe" in the subject. FabTime will not, under any circumstances, give your email address or other contact information to anyone outside of FabTime without your permission.

FabTime® Software for Assembly and Test

"Instead of spending time preparing reports, shift facilitators can get the data they need quickly from FabTime, and then spend their time making real improvements."
Mike Hillis
Cycle Time and Line Yield Improvement Manager
Spansion Fab 25

FabTime Subscription

One low monthly price includes

- Software installation and real-time connect to your MES
- End user and system administrator training
- Unlimited users via your Intranet.
- Software maintenance and regular upgrades (approx. 4 per year, via our no-downtime patch system)
- Add-on dispatching and planning module for a slightly higher monthly fee

Interested?

Contact FabTime for technical details and/or a web-based demonstration.

FabTime Inc.
Phone: +1 (408) 549-9932
Fax: +1 (408) 549-9941
Email: Sales@FabTime.com
Web: www.FabTime.com

FabTime's Web-Based Dashboard is Fully Applicable for Assembly & Test Facilities

- Do your customers (internal or external) want more visibility into your factory?
- Is it difficult to look at trends in equipment performance, or tie equipment performance to throughput and cycle time?
- Does your factory lack real-time reporting?

FabTime can help. FabTime saves your management team time daily by turning MES data into information, via a real-time web-based dashboard that includes lot dispatching. FabTime saves your IT staff time by breaking the cycle of custom-developed reports. Most importantly, FabTime can help your company to increase revenue by reducing cycle times up to 20% for regular lots, and even more for high-priority lots.

Although FabTime was originally designed for front-end manufacturing, you can use FabTime for your assembly or test facility. You simply need to have a transaction-based manufacturing execution system. FabTime can link to all commercial systems commonly used in the industry (e.g. WorkStream, Promis, Eyelit, Mesa, FactoryWorks) or can link to internally developed systems. FabTime can pull data from multiple databases if needed (e.g. WIP transactions from the MES, tool transactions from another system). FabTime is currently being implemented in two assembly and test facilities, with no major technical hurdles.

FabTime Applicability for Back-End Factories

FabTime handles lot merging and splitting, with full tracking of overall cycle times.

All chart quantities (moves, WIP, etc.) can be displayed as die, with data tables formatted for readability of large quantity values.

Custom assembly and test parameters (applicable to WIP or tool state transactions) can be mapped.

Custom site-specific reports for wire bond area have been developed for customers (die and component placements, etc.).

Custom dispatch factors allow for incorporation of back-end-specific data used in dispatch decisions (e.g. availability of boards, and minimization of sequence-dependent setups).