FabTime Cycle Time Management Newsletter

Volume 13, No. 4 September 2012

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. New features in the software this month include the addition of WIPState as standard filter (so that WIP Stacked Trend and Pareto can be sliced by WIPState, showing WIP broken out into Hold, Queue, Process, etc.).

Editor: Jennifer Robinson

Contributors: Mike Hillis (Spansion)

Table of Contents

Welcome

Welcome to Volume 13, Number 4 of the FabTime Cycle Time Management Newsletter! We hope that the arrival of fall finds you all well. Here at FabTime we are keeping busy with new customer installations (we recently kicked off installation for Site30), and continued modifications to our core product. Our general level of busyness has delayed this issue a bit, and we appreciate your patience.

In this installment we have an announcement about a new issue of Future Fab International. Our FabTime software tip of the month is about setting time windows for individual alerts. We have no new subscriber discussion topics. However, our main article this month was written by long-time newsletter subscriber Mike Hillis from Spansion. Mike's article is a detailed response to the topic of reentrant flow introduced in Issue 13.3. Mike discusses the way that variation in the fab makes it difficult to achieve planned output at reentrant tools (particularly nested reentrant tools, and particularly for fabs using critical ratio dispatching). He shares a solution that has worked for his company to keep WIP flowing linearly through the fab. I think that you'll find it interesting.

Thanks for reading – Jennifer

FabTime

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

Community News/Announcements

Issue 42 of Future Fab International is Now Available

The latest issue of Future Fab International, Issue 42, is now available for free download at http://www.future-fab.com/. One article that may be of interest to newsletter subscribers is:

Jackie Ferrell, Les Marshall, Chris Cartier, Jonathan Matthews, Toysha Walker, Alan Weber, and Lance Rist, "Wait Time Waste (WTW) Metrics, Methodology, and Support Tools," Future Fab International, Issue 42, 37-44, 2012.

This article is about an ISMI project to "develop metrics and methods to systematically measure time waste, prototype the metrics and methods, and then develop a SEMI standard ballot for industry adoption." Time waste, in the wafer-centric view of the article, is time during which nothing value-added is being done to the lot, analogous to the loss categories in FabTime's Overall WIP Effectiveness metric.

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

Apply Time Windows to Individual Alerts

It's always been possible to specify a shift for each alert in FabTime. This way you can ensure that you are only notified about your alerts during shifts when you are working. We've added an additional level of granularity, however. For each alert, you can now specify the time window during which that alert will be checked. For example, you might have something that only needs to be watched within 30 minutes of shift change. Or you might want to spot check something once a day while you're on vacation.

When you are creating a new alert, simply look for the input row that says:

Only check this alert between (hh:mm) [ ] and (hh:mm) [ ] (Use a 24-hour clock, e.g. enter 3:30pm as 15:30.)

Fill in your from and to times, using a 24 hour clock, and save the alert. You can also add this data to an existing alert, and you can change it at any time. FabTime will only check the alert (and only notify you) during the time window that you specify. Use of alert time windows helps in two ways. It cuts down on the number of notifications that you receive (ensuring that you only receive alerts when you want them), and it improves FabTime's performance (since server bandwidth isn't wasted checking alerts when they aren't of interest).

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

Subscriber Discussion Forum

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

We have no new subscriber discussion topics in this issue, although our main article this month was written by long-time friend and newsletter subscriber Mike Hillis.

Fab Variability and Reentrant Flow

By Mike Hillis (Spansion)

Inspired by the last newsletter issue, I would like to share some additional thoughts on reentrant flow. My fab has been battling this issue for years and we have worked our way into a place that seems to make sense and a difference.

First of all, I found the discussion interesting regarding how reentrant flow affects things like utilization, variability, etc. I agree there is a relationship. Taken as a given that we have sufficient capacity on board, the variability factor is still a major issue, but I would flip the formula on its head. Reentrant capacity is affected far more by tool variability (and utilization) than the tool utilization is affected by the nature of the process flow!

I will digress for a minute to talk about capacity planning. We have become effective at calculating our required capital to meet output needs. In other words, we can pretty well figure out just what needs to go through each tool group. The algorithms used are sophisticated enough to accommodate the issues brought up in the previous article. Recipe differences, set up times, etc. are all included and give us a utilization target that can be compared to historical availability data. Taken together we can fairly accurately identify when we need to add capital. It may be a static model, but it works for us.

However, we find that where we have heavily reentrant tools, their capacity is affected by line and tool performance variability as discussed below.

Nested Reentrant Flow

The area I am managing now has a relatively complex scenario. Not only are there tool sets caught up in a reentrant flow, they are nested. What I mean by that is that product will visit tools more than once in a line segment (a portion of the line that consists of roughly one week of planned cycle time). Product will then visit the entire segment again, sometimes more than once. To see this more clearly, let's look at the visits for one line segment (each letter is a tool type and usually one of several in a tool group):

The process flow goes like this:

A → B → C → A → B → C → A → C

Clearly reentrant. Then, in a different week of planned cycle time, we execute this a second time. In one flow it is done a third time!

The tools in question are in the back end of the line, and we often end up in a situation mentioned in the original article. We tend to prioritize the very last part of the line (the last visit to this sequence of tools) to make the current shipment target, at the expense of the lots waiting at earlier segments. The impact of this will be discussed in more detail below.

How Tool Variability and Utilization Affect Reentrant Capacity

Now comes the capacity planning and availability/utilization piece. The static model suggests there is sufficient capacity available at reasonable utilization targets to handle this nested reentrant scenario. However, variability in tool uptime and variability in arrivals together act to reduce effective reentrant capacity.

The impact of variability in tool uptime on capacity is not rocket science, of course. If the tools in question exhibit uncharacteristic downtime, it can really throw the whole flow into disarray. Downstream constraint or near-constraint operations run dry, line imbalance grows, shipments are missed and I get more gray hair. Consistent tool uptime is important and necessary for maintaining good linearity (and is of course an organizational focus), but it is not sufficient.

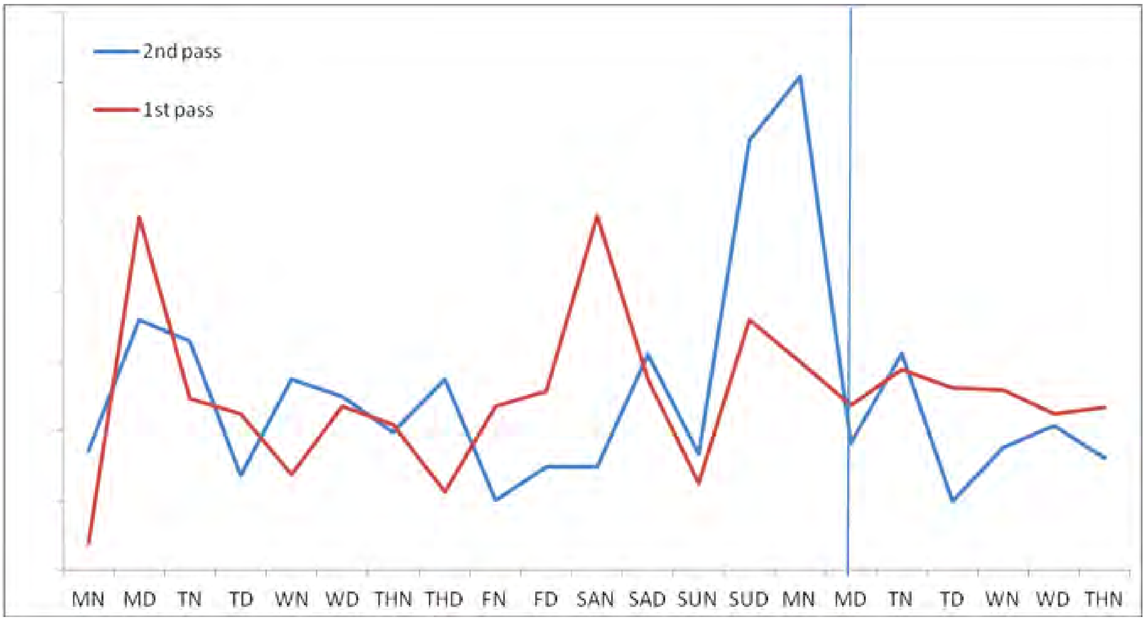
Certainly, if the upstream availability was perfect and the feed was forever constant, this would make managing the nested reentrant tools easier. However, in practice the entire line is subject to perturbations. Variability in supply is inherent in the life of a fab manager. We will always have those times when key tools run dry due to issues upstream. For tools that are heavily "reentrant encumbered" it can really be an issue.

This is a particular problem for reentrant tools because of the tendency to prioritize the back end of the line. If, for example, I run out of WIP at any one of the "visits" throughout the segments, a hole develops. This creates a situation where capacity is "robbed" (not borrowed, it's like borrowing a sip of water. You can't return it!) to make it up. This generates an oscillation in regards to which operations are allocated to the tools. This is not always immediately obvious. It may take two weeks for the full impact of the oscillations to show up. There is a strong tendency to allocate more than the calculated run rate on the shared tools to the last segment, and not running or greatly limiting the output of the penultimate segment. This becomes self-reinforcing. On a non-reentrant tool, you can run like crazy and catch up with no further impact, but not so on the reentrant tool. Of course, this is greatly exacerbated in the nested reentrant scenario.

This becomes a utilization issue that is external to the tool itself. Forced idle time can be killer in these situations. Quite often, the interruptions in linearity occur well upstream and can be difficult to overcome using standard dispatching criteria. This is a critical component of managing tool sets that are heavily reentrant. What we really need, as a tactical solution, is a way to manage these tools to have linear output even when provided with a non-linear feed.

Dispatching for a Nested Reentrant Tool Set

This brings us to dispatching rules. In a classic critical ratio approach, the system wants to drive the material that is the most behind the hardest. As you can imagine, this can easily create the situation described above: push this week (the last visit to the nested operations) and we'll catch up next week later. This behavior results in some pretty wild oscillations that tend to be self-reinforcing (think standing wave) and very difficult to get out of. For example, you might see something that looks like the graph at the top of the next page.



Here the red line is per-shift activities for one segment (e.g. MN = Monday night). The other line is the next pass through the same set of tools. The nested reentrant nature of the process was driving us to run one segment or the other heavily at different times. The problem with this is that unless you do something to break the cycle, you never get out of it. Only after you settle the oscillations are you able to run approximately the same volume through each segment all the time (enhancing linearity of shipments).

Notice that the blue lines and the red lines tend to run in opposite directions or have notable gaps. I will talk about the last six points shortly. In any event, the point is we tend to have rather robust oscillations. This plays havoc with the profile and cycle time, especially at the earlier operations. This is a very real problem with nested reentrant tool groups.

So what to do about it? Since the number of tools is determined by capacity analysis, tool availability is a function of continuous improvement and line non-linearity is a fact of life, what is a Manufacturing Operations Manager to do? The answer ended up being in the dispatching methodology.

It seemed to me that the standing wave was never going to go away if we just pushed harder. We had to have something that was structurally different that would drive fab linearity more effectively. The approach I proposed and tested was to establish a drum to attenuate the effect of the critical ratio problem.

In the example above, the decision was made to start manually overriding the system and force a balance between the 1st and 2nd pass instances of the reentrant tool sets. Over the course of three days (starting at the vertical line in the figure) the oscillation was much softer and our linearity of output improved. We have more internal data on this, but the point is that the idea yielded the intended consequences.

So the question became: how do I do this regularly in the system? I negotiated the implementation of a modified drum across the tool set. The modification agreed upon was that a drum was set at a percentage of the linear demand (rather than the full run rate). We calculate a fraction of the run rate to use as a drum. Then we set that as a minimum target in our dispatching system. Each shift, the factory system will set a goal for this quantity. The dispatching system then selects the correct lots to meet that requirement and sends them to the tool to be processed.

My recommendation was to set the system to 80% of demand but I had to settle for less. The head room was to make sure we could still accelerate the back end if we needed to catch up. However, rather than a digital response the factory system slows down the 1st pass output by some percentage but never cuts it off altogether.

Extension across the Factory

The modified drum seemed to be pretty successfully implemented at the target tool set, so we rolled the drum out across several key parts of the line. We focused on locations that had to deal with reentrant operations on tool sets that were challenged by capacity, variability or some other detractor.

The implementation of the modified drum has been halting, but we are progressing. Different tool sets or line segments may have different drum ratios. Some parts of the line, running by critical ratio is fine as there is sufficient extra capacity or stable tool sets such that this is not an issue. The payoff is in the tool sets I have been going on about. We have found that once we get more linear, the drum has less and less impact and the critical ratio calculations work just fine. This fit with our thinking that line imbalance caused by perturbations really drove us. Now the drum helps smooth things out pretty quickly.

Are there problems? Sure. It is tough to do this the first time around. It might cause a blip in output. However, subsequent periods will be much smoother. Picking the correct ratio is also a bit of an art form. I haven't had much of a chance to think about how to set this in a more scientific manner. (Feedback welcome!) The drum can be over ridden by priority lots, so one must be very alert and observant to make sure this practice doesn't get out of hand and negate the power of the drum.

Conclusions

Reentrant flow is one of the things that make wafer fabs so challenging to manage. I believe that the issue is not so much that the reentrant nature of the flow increases fab variability, but rather that variation in the fab makes it difficult to achieve planned output at reentrant tools. This is particularly a problem with nested reentrant tools, where lots revisit the same tools as part of different segments of the process flow.

The tendency of critical ratio dispatching systems to prioritize lots that are late in the process flow can lead to extreme oscillations in moves by segment on these nested reentrant tools. These oscillations are self-reinforcing, and tend to get worse unless something is done to break the cycle. At our fab, we have implemented a modified drum system for our reentrant tools that forces the system to run at least some percentage of the lots at the earlier segments during each shift. This helps us to keep our WIP flowing linearly through the fab and mitigate the impact of variation on our reentrant capacity.

Closing Question for FabTime Subscribers?

Has anyone else tried a modified drum like this, to mitigate the impact of variability on reentrant capacity? If so, do you have any thoughts on how to set the correct ratio for the drum?

Acknowledgements

The author would like to thank Spansion's Marty Halm for taking time to proofread this article.

Subscriber List

Total number of subscribers: 2742, from 439 companies and universities.

Top 20 subscribing companies:

- Intel Corporation (148)
- Maxim Integrated Products, Inc. (146)
- Micron Technology, Inc. (113)
- Texas Instruments (83)
- Carsem M Sdn Bhd (80)
- International Rectifier (69)
- Western Digital Corporation (69)
- X-FAB Inc. (66)
- ON Semiconductor (65)
- TECH Semiconductor Singapore (60)
- GLOBALFOUNDRIES (59)
- STMicroelectronics (57)
- Analog Devices (52)
- Freescale Semiconductor (52)
- IBM (52)
- Infineon Technologies (50)
- Fairchild Semiconductor (49)
- Skyworks Solutions, Inc. (47)
- Telefunken Semiconductors (46)
- Seagate Technology (39)

Top 4 subscribing universities:

- Ecole des Mines de Saint-Etienne (EMSE) (13)
- Arizona State (8)
- Nanyang Technological University (8)
- Virginia Tech (7)

New companies and universities this month:

- micMAC Global Inc.
- Ocean Thin Films
- Oracle America Inc.
- PCH International
- Sapphire Automation
- SRI International

Sampler set of 20 other subscribing companies and universities:

- 3M Company (3)
- Affymetrix (1)
- Amkor (7)
- Analysis Group (1)
- ASML (16)
- CIMETECH International Inc. (1)
- EM Microelectronic Company (1)
- Fraunhofer (2)
- Intellion (1)
- McKinsey (1)
- Michigan State University (1)
- Mutah University (1)
- National Taiwan University (2)
- National University of Singapore (2)
- Spansion (17)
- Structural Integrity (1)
- Syracuse University (1)
- Vanguard International Semiconductor Corporation (2)
- Visa (1)
- Vishay (6)

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.htm. 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.
- Specific reports for wire bond area are in process (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).