FabTime Cycle Time Management Newsletter

Volume 20, No. 6 December 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. New features in development right now include a new WIP Hours by Tool chart to highlight bottlenecks and enhancements to the new Green to Green chart for tool downtime analysis.

Editor: Jennifer Robinson

Contributors: David Carmichael (TowerJazz Semiconductor); Hani Ofeck (TowerJazz Semiconductor); Brandon Paske and Sara Anderson from Seagate Technology; Justice Stiles (Infineon Technologies); and Patrick Sorenson (Microchip Technology)

Keywords: Dispatching; Metrics and Goals; Tool Qualification

Table of Contents

- Welcome
- Community News/Announcements
- FabTime User Tip of the Month – Generate A List of Tool Qualification Bottlenecks
- Subscriber Discussion Forum – Queue Time States; Tool Qualification; and Dispatch Compliance
- **Main Topic – Dispatch Compliance in Wafer Fabs**
- Current Subscribers

Welcome

Welcome to Volume 20, Number 6 of the FabTime Cycle Time Management Newsletter. We hope that the holiday season is treating you all well. A special welcome to our many new subscribers this month. The October issue was well-received, and we thank everyone who passed it along to their colleagues.

In this issue, we start with one announcement about an upcoming new software release and another highlighting some recent news shared by Jennifer on LinkedIn. Our FabTime user tip of the month is, following up with the previous issue, about how to generate a list of current tool qualification bottlenecks (operations that have WIP and only a small number of qualified tools).

We have a robust subscriber discussion forum this month, with a small correction to the previous issue and a new topic for which we are seeking input: breaking down queue time into sub-states based on operator unavailability, downtime, and lack of tool qualification. We also have several responses to a question we posed last month about dispatch compliance, leading into our new main article on that topic. Aggregating inputs from the literature, the subscriber community, and FabTime's customers, we identify three primary approaches to tracking and reporting dispatch compliance. As always, we welcome your feedback.

Best wishes for a happy, healthy, and productive 2020 to all of you – Jennifer and Frank

FabTime

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

Community News/Announcements

FabTime Patch 112

FabTime will shortly be releasing software Patch 112. Notable enhancements in this new release include:

1) Cycle Time Revamp

   a) Add Elapsed Cycle Time Move Details chart. This new chart will make it possible to drill into the step-by-step cycle time details underlying the Elapsed Cycle Time List chart.

   b) Add Operation Cycle Time Details Trend/Pareto Charts, displaying operation cycle time split into PreProcess, Process, PostProcess, Queue, Hold, Transport, and Other.

   c) Split PreProcess, Process, PostProcess, Queue, Hold, Transport, and Other Time into their own columns on all cycle time charts.

   d) Provide detailed rework and non-rework data table columns on all cycle time data tables.

2) New Tool Green-to-Green (G2G) List chart (as described in Issue 20.02).

3) Client-side support for fast hiding of data table columns.

4) Support for carrying hold comments forward across transactions while lots are on hold, including multiple simultaneous holds.

5) Ability to detect and resolve duplicate MES transactions before they are processed into FabTime.

6) New WIP Hours Trend, Pareto, and List charts (as described in Issue 20.03). These charts are useful for detecting temporary bottlenecks based on hours of work that is waiting to be processed.

We are grateful to our User Group for suggesting, prioritizing, and performing early testing of many of these improvements. We are also appreciative of the newsletter community for discussions that have helped guide these enhancements. Customers interested in installing Patch 112 should contact support@FabTime.com.

Building Community on LinkedIn

Jennifer continues to share articles about business management, the semiconductor industry, and productivity improvement on her LinkedIn feed. Recent posts have included:

- An opinion piece from the Wall Street Journal positing that America was wrong to think that we could continue innovating in the US while outsourcing our manufacturing capabilities. The authors argue that "the federal government must do more than invest in basic research; it must also fill the innovation deficit by creating a new infrastructure for R&D in engineering and manufacturing."
- News that for the first time ever, girls won all 5 of the top prizes in the Broadcom Inc. MASTERS STEM competition for middle school students.
- A question asking readers what they think are the most significant contributors to cycle time in wafer fabs, with various interesting responses. This thread may be included as part of a future newsletter article.
- And lots more…

Connect with Jennifer here: http://www.linkedin.com/in/jenniferrobinsonfabtime

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

Generate A List of Tool Qualification Bottleneck

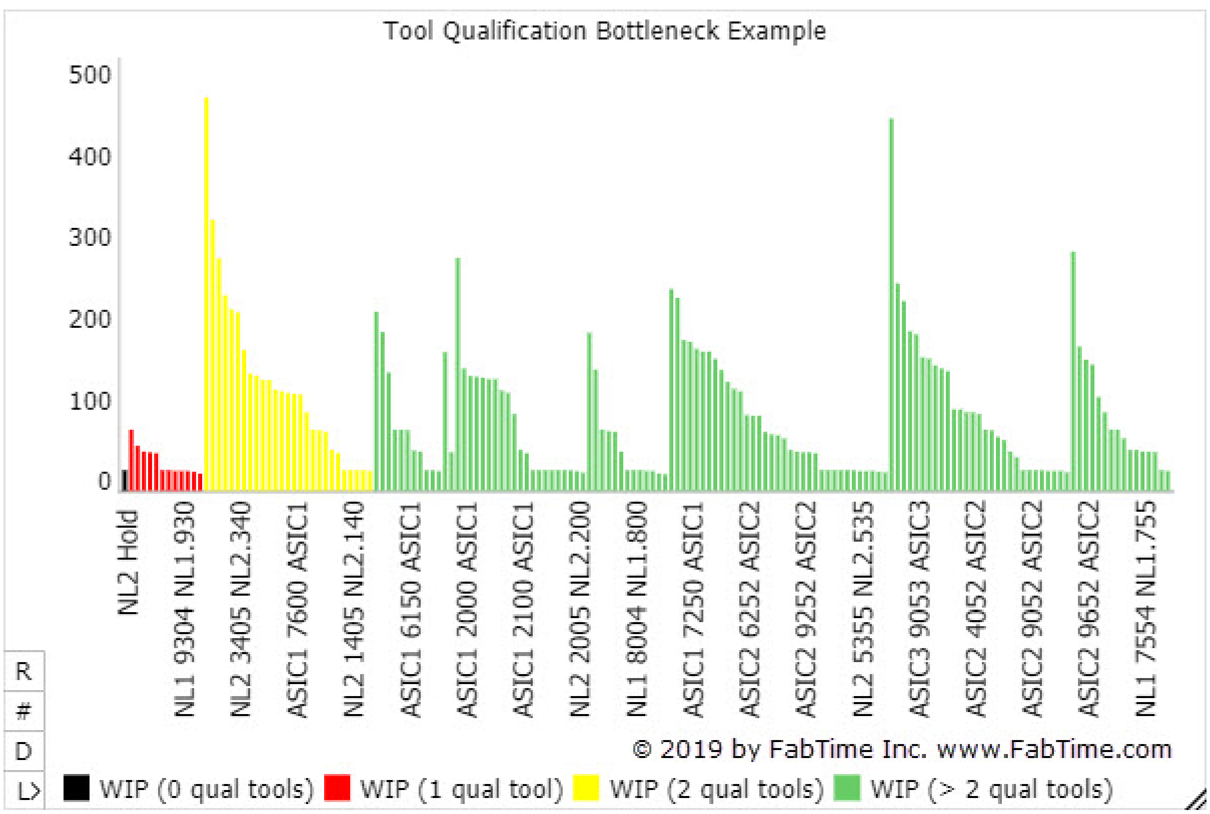
In the last FabTime newsletter, our main article was about understanding the impact of lack of qualified tools on fab cycle time. We explained how going from a single qualified tool for an operation to having a second qualified tool cuts cycle time through that operation by roughly 50% (at the same utilization rate). Going to three qualified tools decreases cycle time even further, with gradually diminishing improvement after that.

In this tip, we show how to generate a real-time list of tool qualification bottlenecks using FabTime. These are operations that have WIP waiting that have a low number of qualified tools. For fabs that have many one-of-a-kind tools, qualifying a second or third tool may not always be possible. However, we believe it's best to know where lack of backup is causing WIP to pile up. Over time, this data suggests strong candidates for capital purchases.

To look at this in FabTime, generate the Tool Qualification WIP Detail chart, as shown below.

By default, this chart shows a bar for each flow/step/qualification combination that has WIP. If you specify a flow filter, there will also be entries for steps with no WIP. Black bars are steps with no qualified tools, red are steps with one qualified tool, yellow are steps with two qualified tools, and green are steps with two or more qualified tools (sorted in order by number of tools). The steps of the most concern, of course, are black or red steps that have a high level of WIP.

To narrow in on the tool qualification bottlenecks, you can use the SQL filter to the left of the chart. Entering "QualToolCount <3" strips out all of the green bars and shows you just the steps with zero to two qualified tools. You can also use other filters (Area, ToolGroup, Flow) as needed. You can drill down to see the associated lot list for any of the rows in



the data table, or to see a history of the tool qualification information.

This chart is only available if your site's MES data link sends qualified tool data to FabTime. Qualified tools are listed at the flow/step/qual tool list level. The Help page includes more detail about how the data is stored and how the calculations work. If you display this chart for a time in the past, please note that qualified tool counts will be for the latest tool qualification data. FabTime stores the history of tool qualification changes to assist with troubleshooting but uses the latest tool qualification data for all computations.

We highly recommend that, unless you have a pure one-of-a-kind toolset, you regularly review this information for your fab. Because product mix changes so frequently in fabs, it is not uncommon for a new route to come online with many single path steps. Unless you have procedures in place to check, you could end up leaving such restrictions in place, leading to unexpectedly high cycle times.

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!

Subscriber Discussion Forum

Queue Time States

Jennifer was at the Fab Owners Alliance meeting at Broadcom in October. She talked with Sara Anderson and Brandon Paske from Seagate about whether it was possible to quantify the impact of lack of tool qualification on fab cycle time. This would be useful information for fabs in balancing yield vs. cycle time tradeoffs. It turns out to be difficult to quantify, however, because we can't run a controlled experiment on our fabs. We could estimate this using simulation, but it would take a very detailed model, and those are hard to keep up.

This got Jennifer thinking about whether we could break down queue time into different sub-states to better understand the contributors to cycle time. This could help to understand not just the impact of lack of tool qualification, but also the impact of downtime and operator unavailability. We came up with a first-pass proposal for doing this, discussed it with our ever-helpful User Group, and are now seeking feedback from the newsletter community.

Queue states come down to understanding, where possible, why a lot is in queue. As with most things in fabs (see dispatch compliance discussion below), this gets complex when we get into the specifics. But it does seem possible to get at least a rough estimate. Here's our first pass-proposal:

The lot is in queue because:

1) Two or more tools are qualified to run the lot and
    a) At least one other tool is available but not running. The corresponding tool state is standby-WIP-waiting. The queue

state would be something like "no operator to load the WIP."

b) All qualified tools are busy. This is the classic understanding of queue time, "other tools busy."

c) At least one other qualified tool is down. This queue state would be "in queue for a down tool."

2) Only one tool is qualified to run this lot and

a) It is a one-of-a-kind tool and is busy or down or in standby-WIP-waiting as above. In this case, we'll use the 3 states above (1.a to 1.c).

b) There are other tools in the tool group, but they are not qualified to run this flow/step combination and

i) At least one of those other similar tools is available. The state here will be "in queue because of lack of tool qualification."

ii) None of the other tools are available. Here we probably just default to 2.a, for simplicity. It's hard to blame this queue time on lack of tool qualification when the other tool isn't even available.

Assumptions and open questions:

- The queue state changes every time there is a new information available.
- We'll only consider (for now) the cost of having a single qualified tool. We won't look at potential benefits of going from two to three, three to four, etc. This is to manage complexity, and because the largest "bang for the buck" is in going from single path to dual path.
- We are attributing standby-WIP-waiting to there not being an operator to load the tool. In practice, WIP is sometimes held at an earlier step for other reasons, as when we delay the clean step until the furnace is ready. To capture that, however, we would need to log a different WIP state. Everything else here can be determined (even if it's an approximation) from the existing WIP and tool state transactions.
- We're assuming all the tools in a tool group could be qualified to run the same operations. This is not true. In practice we could use a different tool group mapping. In FabTime there's an attribute for capacity type. We could use this, or maybe there would need to be a "qualification group" attribute for tools similar enough that we would expect to be able to cross-qualify them.
- There's a question about how to handle large queues. All the lots aren't really in queue right now because there's a down tool. Only the lot at the front of the queue is technically waiting for this downtime. Except … downtimes can be really long relative to process time, so maybe they are all in queue because of the downtime. What do you all think?

We have been working over the years to break down overall cycle time into more granular buckets (queue time, hold time, transport time, etc.), as outlined in our Patch 112 announcement above. Here we look specifically at breaking the (often large) queue time bucket into smaller sub-categories. The more detail we have about how lots are spending their time, and why, the more easily we can work on improvement.

What do subscribers think:

- Is this worth doing?
- Are we missing anything major in thinking about this?
- Have you ever tried to do this in practice, either systematically in your reporting, or as a one-time analysis for a key lot?

We welcome your feedback and hope to write about this in a future newsletter.

Correction to Tool Qualification Article

FabTime would like to **thank an anonymous subscriber** who kindly brought to our attention a typo from the main article in the October issue of the newsletter that significantly changed the meaning of a statement. We had written:

"One other point about tool dedication is that "soft constraints" can also arise. Soft constraints are places where tools are dedicated in practice, even if there is such official restriction."

What the text should have said (correction in red bold upper case) was:

"One other point about tool dedication is that "soft constraints" can also arise. Soft constraints are places where tools are dedicated in practice, even if there is **NO** such official restriction."

Soft constraints, by our definition, are unofficial, and usually arise due to layout issues or operator preferences.

The subscriber added this important point: "One additional comment on this topic…

You wrote "Soft constraints, by our definition, are unofficial, and usually arise due to layout issues or operator preferences." Of course, these are absolutely correct and are consistent with what I have seen.

There is a third reason that I have observed that I would call "engineering preferences." In this case, the process engineers don't want to officially disqualify (or inhibit) the tool or chamber, but provide unofficial direction to the operators to not run a certain recipe on a certain tool or chamber because it isn't viewed as producing quite the same level of quality as the best tool or chamber in the group and that as long as WIP levels are acceptable certain chambers or tools are preferred to others."

FabTime Response: This is very interesting. This type of "unofficial direction" is one of the things that makes dispatching and scheduling challenging in wafer fabs. Not every restriction or operating practice is captured in the MES. In this case, we can see the upside. Yields can be improved, but because the restriction isn't official, operators have flexibility in the event of large queues building up. The problem with this approach, of course, is that it can lead to unexpected cycle time problems. The people making delivery commitments for the fab don't know about these "engineering preferences" and may expect cycle time to be lower than it is in practice.

Fabs! Always interesting!

Dispatch Compliance

We received several responses to our previous question about dispatch compliance. Rather than responding here, we will save FabTime's response to these contributions for our main article below. We are grateful to all who took time to respond and share with the subscriber community.

David Carmichael from TowerJazz Semiconductor wrote: "We have tried many different approaches to this problem over the years, but none were fully satisfactory due to the real-time nature of the Dispatch lists, the equipment state and the Operator's decision.

We now do have one that works.

When our automation system tracks a lot into a tool it examines the Dispatch List that was just displayed and ranks where on the list the tracked-in lot was entered. If not on the list, then that is recorded. This data is then stored in our MES during track-in so that we can get an accurate measurement of compliance.

I believe this may be the only way to get good compliance information as, if the automation is not used (assuming this is allowed), then no compliance entry is made."

An anonymous subscriber wrote: "We do have a dispatch compliance metric. People did use the dispatch list, but the complication arises when we try to achieve 100% compliance. I think you would agree that the environment of the fab is constantly changing. So, with those changes, we need to keep our dispatch list up to date with all the new requirements that come along. Given the time between development and implementation and even fixes to potential bugs introduced on the newest dispatch version, it could be bad for people to have to comply 100%. We found it best to leave some wiggle room for our compliance to hover around 80% to allow that 20% be the human aspect of catching irregularities with dispatch lists."

Justice Stiles from Infineon Technologies wrote: "The question put out to subscribers for the next newsletter regarding dispatch compliance is almost a case study in questions that seem simple superficially but can become a rabbit hole once you dig into them. There are a lot of ways you could potentially define the concept of dispatch compliance, and just as many metrics you might choose to measure it by. For our fab, we've taken a relatively simple approach defined by the following rules:

1) A move is either compliant or it is not. The relationship is strictly binary.
2) A compliant move is one that conforms to our specified standards of dispatch use:
    a) A Dispatch List was generated for the specific tool loaded before a lot was tracked in.
    b) The operator tracked in a lot or batch of lots that was within the accepted range of run orders from the dispatch system.
        i) Currently the accepted run order range is 1 through 3.
        ii) This means if the operator tracked in any lot or batch of lots with run orders 1, 2, or 3, the move is considered compliant.
3) All other moves are considered non-compliant.
4) Compliance is measured as a percentage of compliant moves to total moves.
5) We set a goal for the percentage of compliant moves for the fab.
    a) Currently our compliance goal is 85% compliance.
    b) The reason we do not expect 100% compliance is dispatch can't account for every possible externality such as instructions by a fab supervisor to prioritize a certain lot or product type, or a request by an engineer or Process Tech to run a given lot out of order.

Our primary metric then reduces to a dispatch compliance percentage that we can then slice by area, sub-area, operation, operator, shift etc. to get a handle on how well our operators are using the dispatch system."

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

Dispatch Compliance in Wafer Fabs

Introduction

A subscriber wrote to us recently to ask if we had ever published an article about measuring dispatch compliance. We had not and thought that it was a good idea. What our research on the topic has uncovered is three different types of compliance measures, made somewhat murkier by variations in terminology. Here, we share our results as part of our continuing quest to help spread knowledge across the industry and help drive performance improvement.

Background: Dispatching and the Need for Compliance Metrics

Historically, wafer fabs have relied heavily on the use of dispatch lists. A dispatch list is simply a list of all the lots that can be processed on a tool right now, sorted in some order. That sorting order can consider information from other areas of the fab (e.g. downstream WIP levels or tool status information). However, the dispatch list itself is generated locally for each tool and is updated in near-real time based on the most current available information.

Dispatching is different from scheduling, in which we look ahead across the day or the shift and across multiple tool groups to make a larger plan based on some optimal or near-optimal objective function. Scheduling is challenging in wafer fabs because of the high degree of variation and the large quantity of data involved. Scheduling has been increasing in use in fabs as it becomes computationally more feasible. However, local dispatch lists remain common, and operator compliance to those lists is the subject of this article.

Dispatch lists are typically displayed for the operator at each tool. The operator might see the full dispatch list or might only see some sub-set representing the top few lots on the dispatch list. Ideally, the operator selects a lot to process from the dispatch list and loads it onto the tool. In some cases, the operator is expected to log a dispatch transaction. In other cases, the dispatch transactions are created automatically and sent to the MES.

In our experience, it is unusual to require the operator to always select the first lot on the list. Rather, operators have some discretion regarding which lot is chosen. This is where dispatch compliance comes in. Dispatch lists reflect fab management priorities. Typically, the people running the fab have chosen dispatch rules that they would like to see followed. They (almost always) want the high priority lots processed first. They may want the lots that are behind schedule boosted to get them back on target. They may want to smooth WIP. They may want to prioritize lots for a customer or prioritize lots that will ship soon. The possibilities are many and complex, and often vary for different areas of the fab.

Meanwhile, the operators have priorities of their own. As just one example, they may be assessed based on the completion of move targets. Even if this is not officially the case, human nature dictates that people feel a sense of accomplishment when they can complete a series of activities. That is, they may assess themselves based on their moves per shift. If this is the case, the operator will prefer starting a lot of the same recipe over, say, waiting to do a setup first. And what if the first lot on the dispatch list isn't even there yet? What then? There are many reasons why, in practice, operators don't always select the first lot on the dispatch list.

This tension between the priorities of fab management, as expressed by the dispatch rule, and local preferences and constraints, as experienced by the operators, drives the need for dispatch compliance metrics.

For this article, we looked at dispatch compliance ideas from three sources:

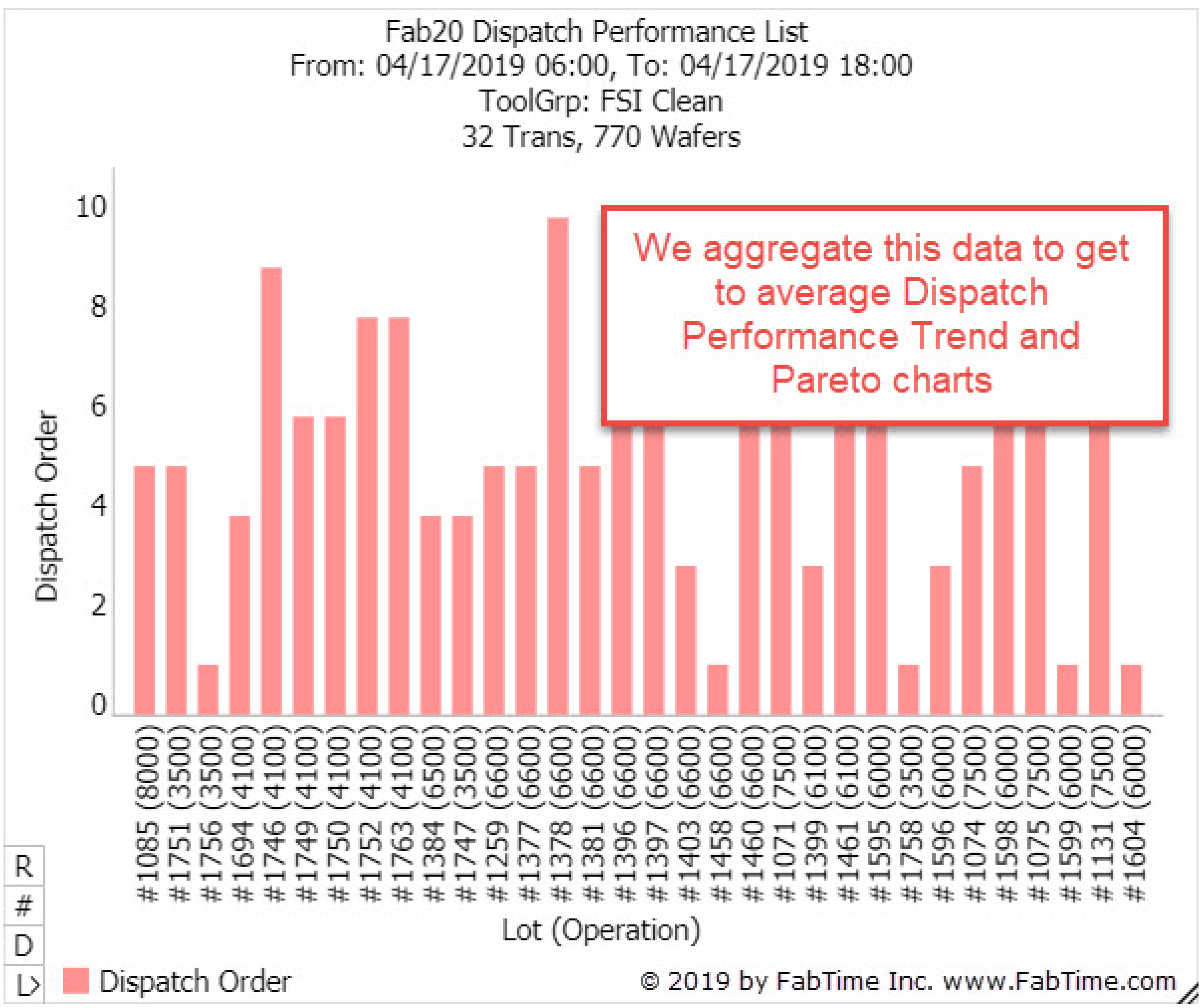
1. Our customers, whose requests have influenced the compliance metrics currently available in our software.
2. Published literature.
3. FabTime newsletter subscribers, whose inputs are shared above.

We will discuss ideas from each of these sources and then summarize the results.

What FabTime Does Now

FabTime's primary dispatch compliance metric is Dispatch Performance, aggregated from Dispatch Order. Each dispatch transaction in FabTime includes the dispatch order, or position of the lot on the dispatch list at the time it was moved into a tool. The Dispatch Performance List (example below) shows the individual transactions, while the Dispatch Performance Trend reports average Dispatch Order over time, and the Dispatch Performance Pareto allows the slicing of average Dispatch Order by area, tool, operation, priority, employee, etc. Dispatch performance that is close to one means that operators usually select the first lot on the list, while higher numbers mean that operators are selecting lots from farther back in the list.

For most dispatching sites, FabTime is configured to automatically compute the order of lots on dispatch lists and generate a dispatch transaction containing this information each time a lot is tracked into a tool. Alternatively, the MES -> FabTime data link may report dispatch transactions to FabTime. In the latter case, FabTime can display a list of Move In transactions for which there was no corresponding dispatch transaction. These generally indicate that a dispatch list was not generated prior to the operator starting the lot. The dispatch performance charts



include a % Dispatch line indicating the percentage of lots that had a dispatch transaction.

FabTime also maintains a secondary dispatch compliance metric based on the lot's relative placement in the dispatch list. We call this metric Dispatch Precision. This metric was suggested by **Hani Ofeck from TowerJazz Semiconductor**. Hani also assisted FabTime with advice during development. Dispatch precision for a lot/tool is:

100% (1.0 - (DispatchOrder-1)*(1.0 / MAX(DispatchOrder on the most recent dispatch list where the lot appears for the tool)))

If the first lot on the dispatch list is selected, then dispatch precision is 100%. If the last lot on the dispatch list is selected, dispatch precision is 100%/MAX(DispatchOrder).

Dispatch Precision is useful when asking the question "Is running the 5th lot on the dispatch list ok?" If there are 5 lots on the list, and the operator runs the 5th lot, our answer is probably "No". But if there are 50 lots on the list, and the operator runs the 5th lot, our answer is probably "Yes, good enough." Comparing these two examples, running the 5th lot out of 5 lots has a dispatch precision of 20%. Running the 5th lot out of 50 lots has a dispatch precision of 92%. As with the previously outlined dispatch performance metric, dispatch precision can be displayed in a list form, or averaged and displayed in trend or pareto versions.

What the Literature Says

We identified a relatively small number of published papers related to dispatch compliance, listed in the Further Reading section below. We looked at three of these in detail and share highlights from those papers here.

Bernd Waschneck, Thomas Altenmüller, Thomas Bauernhansl, and Andreas Kyek (2018) share a case study of dispatch compliance in a wafer fab and examine factors contributing to low dispatch compliance. They define both Absolute Compliance (where was the lot on the dispatch list when it was processed, first, second, third, etc.) and Relative Compliance (where the length of the queue is also considered). For relative compliance, they define compliance as 100% if the first lot on the dispatch list is chosen. Where there are two or more lots on the list, compliance ranges from zero (choosing the last lot on the list) up to 100%.

Looking at data from three fabs with different characteristics, they find that absolute compliance is similar across the three fabs and varies with queue length (the longer the queue, the worse the absolute compliance). Relative compliance varies much less with queue length, except for very short and very long queues. The authors also look at relative compliance for different classes of dispatch rules and types of tools. We refer those interested to the full paper for details. The authors stress the importance of looking for root causes in cases where dispatch compliance is poor, in order to improve the dispatch rules going forward.

Gißrau and Rose (2013) assess the performance of a combined dispatch rule that they had proposed in an earlier paper. Part of their assessment is in terms of dispatch compliance. They define a Compliance Score that is based on the position of the selected lot relative to the length of the dispatch list. If the first lot on the dispatch list is selected, the compliance score is 1. Otherwise, the score varies between zero and 1, with zero being the last lot in the dispatch list. They also define an Absolute Dispatch Compliance value that is 1 if the first lot in the dispatch list (or the first batch) is selected and zero otherwise. They define Average Sort Index as the average lot position taken from the dispatch list.

Looking at the compliance data over a test period, they find the actual compliance score to vary between .55 and .68, lower than the target of .75. They cite several reasons for this lower than target score, including:

- Manual transport (exact location of the lot isn't known).
- Organization of the WIP, resulting in operators taking lots in FIFO order from those stored nearby.
- Process requirements of cluster tools that are not fully documented in the MES (hence not accounted for in the dispatch lists).
- The "conditioning state" of the tool, which seems to be some variant of a setup, leading to operators choosing lots of the current conditioning state rather than changing conditioning.

[Gißrau and Rose also look at how long the dispatch list takes to generate, noting that values of above 3 seconds are not accepted by the operating staff. Of this list generation time, the most time is taken up in "detecting the right lots for the equipment rather than by the dispatch algorithm itself." We found this an interesting corollary to our recent work on tool qualification, though it's not directly related to dispatch compliance.]

Madan Chakravarthi, Chih Ming Chan, and Muralitharan Subramanian (2005) present a case study of Chartered Semiconductor's joint venture wafer fab that realized benefits from a dispatch system by focusing on dispatch compliance. The authors define compliance according to the rank of the lot at the time that it is selected and break the ranks into groups: "Rank 1-2, 3-5, 6-10, > 10. (A lot that had a dispatch rank 4, if selected would fall into Rank 3-5 category)." They track the percentage of moves falling into each rank category for a time period and display these using bar charts.

They further distinguish between compliance at lot selection and lot track-in, noting that "trackin may not happen immediately after lot selection especially if there is a large number of lots queuing in front of the tool." The authors categorize compliance at lot track-in into unconstrained compliance and constrained compliance. The former is the compliance to the tool group-level dispatch list, while the latter is the compliance to a list generated for the individual tool, taking process restrictions into account. Constrained compliance at lot track-in seems to most closely match the other dispatch discussion in this article.

Chakravarthi et. al. also use something called native compliance to record the rank of the lot according to what the dispatch list would have been if the global dispatch rule for the fab was used. The idea is to measure how additional local rules are diluting the fab's overall objectives. They further use the difference between constrained and unconstrained compliance as an indicator of the degree of tool dedication, with a large gap indicating a high degree of dedication. We appreciate this approach of using variations in the dispatch compliance metrics to look at other effects.

In looking at this research overall, we can see the need for some standardization across dispatch compliance metrics. What Waschneck et. al. define as Absolute Compliance (position in the list) is what Gißrau and Rose define as Sort Index, what Chakravarthi et. al. call Rank, and what FabTime calls Dispatch Order. Meanwhile, what Gißrau and Rose call Absolute Compliance is something else, a 0,1 variable. What Waschneck et. al. call Relative Compliance is what Gißrau and Rose call Compliance Score and is similar to FabTime's Dispatch Precision.

What the Subscriber Community Said

Looking at the responses recorded above from subscribers we have:

- One company (TowerJazz Semiconductor) that uses the same approach that FabTime uses: report dispatch performance in terms of where the lot was on the associated dispatch list at the time of the Move In, using automatically generated transactions to get the most detailed results.

- One company that didn't specify the format of their compliance metric in detail but noted that they only expect about 80% compliance from operators, vs. 100%.

- One company (Infineon Technologies in Temecula) that uses a binary system to declare a move as compliant or not. This is like Gißrau and Rose's Absolute Dispatch Compliance metric, except that in Infineon's case, the move is declared compliant if it is in positions 1 through 3 on the dispatch list. In the Gißrau and Rose paper, a lot had to be in position 1 to be absolutely compliant. Even after allowing lots up to the third spot on the list, Infineon also only requires 85% compliance to the metric, to allow for "possible externality such as instructions by a fab supervisor to prioritize a certain lot or product type, or a request by an engineer or Process Tech to run a given lot out of order." As with FabTime's compliance metrics, Infineon's binary metric can be sliced by various attributes, trended, etc.

Putting It All Together

Putting all the above together, we see that there are three general approaches to dispatch compliance, though the naming of these varies. We can track:

1. The order of the selected lot on the dispatch list that was displayed (or would have been displayed) when the lot was selected. Here values close to 1 are best, with the scores getting higher and higher as operators track lots farther down on the list. This order may be further broken into groups and graphed as a stacked bar chart.
2. The relative order of the lot on the dispatch list after considering the length of the list. A score of 1 (or 100%) is given when the first (or only) lot on the dispatch list is selected. A low score is given when the last lot on the list is selected. The score otherwise scales between its minimum value and 1 (or 100%).
3. A binary variable that labels a move as compliant or not. A compliant move could be defined to always be the first lot on the list but could also be defined more flexibly as the first to third lot on the list, or otherwise.

In all cases, the dispatch transaction could be for a single lot or for a batch. In all cases, the metrics can be aggregated and trended over time or sliced by other attributes (operator, tool group, etc.). In no case did anyone report requiring operators to be 100% compliant.

Here are a few reasons why operators don't always select the first lot on the dispatch list:

- The lot might not even be there yet, especially if lots are transported manually.

- The operator might have local knowledge that is not codified into the system (soft constraints or other preferences).

- The operator might have received short-term instructions from a supervisor or a process engineer that have not been codified into the system (e.g. "We want to drive up shipments because tomorrow is the end of the quarter" or "That operation runs better on this other tool")

- The operator's personal incentives might differ from the priorities of the dispatch list (driving individual moves, reducing setups, not wanting to go track down a lot that isn't nearby, etc.).

Doubtless there are other reasons. Fabs are very complex environments.

Conclusions

As Justice Stiles noted in the subscriber discussion forum, dispatch compliance is one of the many things in fabs that sound simple but become more complex once you dig into the details. Telling operators to just run the first lot on the dispatch list seems like a reasonable idea on the surface but requiring 100% compliance to that is impractical for many reasons.

Our experience and our research here suggest that there are three primary approaches for measuring dispatch compliance: report the order that the lot was in on the dispatch list (possibly using ranges to make the results more discrete); weight the dispatch order by the length of the dispatch list; or use a binary metric that labels moves as compliant or not. We currently use the first two approaches in FabTime, but not the third. We think that this third approach, particularly with the added flexibility included by Infineon Technologies in marking the first through third lots as compliant, is especially promising. We also think that some additional work is needed to standardize the terminology for these three approaches across the semiconductor industry, to make it easier to talk about them across companies.

We would like to close by highlighting a point made by Waschneck et. al. The purpose of dispatch compliance shouldn't be to penalize operators for not selecting the first lot on the list. Where dispatch compliance is poor, there is always a reason. While further training of the operators may be needed in some cases, poor dispatch compliance scores provide an opportunity for the fab to improve dispatch rules (and to better document informal rules in the MES) going forward.

Closing Questions for Newsletter Subscribers

Do you think that the three general approaches outlined here are sufficient for tracking dispatch compliance? Do you have any thoughts about naming conventions on these metrics? How is dispatch compliance different in more automated fabs?

Further Reading

- M. Chakravarthi, C. M. Chan, and M. Subramanian, "Compliance to Dispatch Rules in a Wafer Fab," *Proceedings of the 2005 Modeling and Analysis of Semiconductor Manufacturing (MASM) Conference*, 2005. A copy of this paper was kindly sent to us by the first author.

- M. A. Chik, I Ahmad, I., and M. Y. Jamaluddin, "An Alternative Approach to Measures the Application of Dispatching Rule in the Wafer Foundry," *Proceedings of the 2004 IEEE International Conference on Semiconductor Electronics*, 2004. We have not seen this paper ourselves, but Waschneck et. al. report that this paper found after the introduction of a dispatch compliance monitoring system, "the cycle time variability could be reduced by 15%."

- M. Gißrau (X-FAB Dresden) and O. Rose (Universität der Bundeswehr München), "Practical Assessment of a Combined Dispatching Policy at a High-Mix Low-Volume Asic Facility," *Proceedings of the 2013 Winter Simulation Conference*, R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, eds., December 2013. Available for download here.

- B. Waschneck, T. Altenmüller, T. Bauernhansl, and A. Kyek, "Case Study on Operator Compliance to Scheduling Decisions in Semiconductor Manufacturing," Conference Paper shared by the authors on ResearchGate, 2018. Available for download here.

Acknowledgements

This article was inspired by a question from newsletter subscriber Patrick Sorenson from Microchip Technology. We are also grateful to the customers who have worked with us on our dispatch system, and to the subscribers who contributed to the discussion above.

Subscriber List

Total number of subscribers: 2748

Top 20 subscribing companies:

- ON Semiconductor (212)
- Infineon Technologies (156)
- Micron Technology, Inc. (121)
- Intel Corporation (117)
- GlobalFoundries (97)
- Maxim Integrated Products, Inc. (89)
- NXP Semiconductors (81)
- Microchip Technology (70)
- Carsem M Sdn Bhd (69)
- Skyworks Solutions, Inc. (65)
- STMicroelectronics (63)
- Western Digital Corporation (61)
- Texas Instruments (53)
- Seagate Technology (52)
- TDK (50)
- X-FAB Inc. (49)
- Analog Devices (41)
- TowerJazz Semiconductor Ltd. (34)
- Zymergen (34)
- Cree, Inc. (33)

Top 4 subscribing universities:

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

New companies and universities this month:

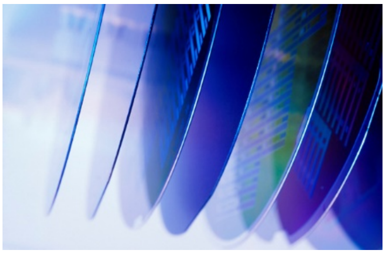
- BISTel
- Daikin
- Ionbond IHI Group
- Pacific Science
- PJP Tech Korea
- Romaric
- SMI Pressure Sensors
- Teledyne Micralyne
- University of Cincinnati

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® Dispatching Module



Dispatch Configuration and Support

We offer our dispatching module for a single, fixed annual fee (on top of your regular FabTime subscription). This includes:

- Dispatch rule configuration via user-friendly web-based interface for standard factors
- Training.
- Dispatch list feed to the MES (if applicable).
- Support and upgrades.

Custom dispatch rules and consulting from our dispatching expert available for additional fee

Dispatch Factors

- Batch code at the current tool.
- Lot priority.
- Downstream tool priority.
- Current tool FIFO.
- Current tool idle time.
- Downstream batch efficiency.
- Critical ratio.
- Earliest-due-date.
- Current step processing time.
- Remaining processing time.
- Current step qualified tool count
- WIP level or staging time at downstream tools.

Interested?

Contact FabTime for details.

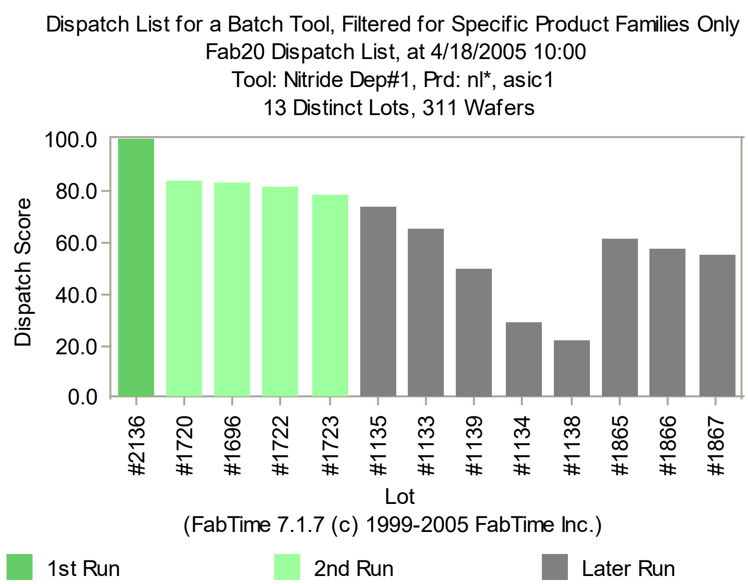
FabTime Inc.
Phone: +1 (408) 549-9932
Fax: +1 (408) 549-9941
Email: Sales@FabTime.com
Web: www.FabTime.com

Do your operators make the best possible dispatching decisions?

- Do you struggle to balance lot priorities and due dates with tool utilization and moves goals?
- Do your critical bottleneck tools ever starve?
- Do you use standard dispatch rules, but feel that your fab's situation is more complex, requiring custom blended rules? Do you know how well your fab executes your dispatch strategy?

FabTime's dispatching module is an add-on to our **web-based digital dashboard software**. At any point, for any tool in your fab, FabTime will show you the list of all lots qualified to run on that tool. This list will be ordered by the dispatching logic that your site has selected for that tool. This logic can use standard dispatch rules such as Priority-FIFO and Critical Ratio. However, you can also create custom dispatching logic using any combination of dispatch factors (shown to the left).

You can display dispatch lists in FabTime, and/or export them back to your MES. FabTime also includes a dispatch reservation system to hold downstream tools when a lot is started on an upstream tool, as well as dispatch performance reporting. FabTime also includes an optional **short-interval scheduler**.



FabTime Dispatching Module Benefits

- Ensure that wafers needed by management are in fact the wafers that are run, while requiring less manual intervention on the part of management.
- Improve delivery to schedule, and the display of performance to schedule.
- Document the dispatching logic used by the best operators and make this available to all shifts.