FabTime Cycle Time Management Newsletter

Volume 5, No. 7 August 2004

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 current version (6.0) include the ability to mix FabTime slide show charts and external web content using a web-screen-saver, and an HTML archive of all past FabTime newsletters.

Editor: Jennifer Robinson

Contributors: James Ignizio (Intel); Jimmy Martin (Analog Devices)

Table of Contents

Welcome

Welcome to Volume 5, Number 7 of the FabTime Cycle Time Management Newsletter! In this issue we are pleased to announce the publication of one of our papers in the latest issue of Semiconductor Fabtech magazine. We also have a call for papers for a special journal issue on operational control of wafer production. This issue includes subscriber discussion related to capacity planning, Dynamic X-Factor, WIP Utilization, and metrics for measuring the effect of tool downtime. This last discussion topic inspired us to think about quantifying the effect of downtime on both shipments and cycle times in our main article.

If we could eliminate downtime from our fabs, we could increase throughput (where the constraint tools have any downtime at all), and improve cycle time at the same time. In this article, we make a first pass at quantifying this impact more formally, by measuring the increased operation-level cycle time for lots that are in queue when a tool goes down. We believe that better understanding the cycle time cost from specific downtime events could be helpful for fabs in deciding where to focus tool improvement efforts. We hope that you find this article interesting, and 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

FabTime Article in the Current Issue of Semiconductor Fabtech

The following paper was published in the Twenty-Second Edition of Semiconductor Fabtech Magazine.

J. Robinson, "Best Practices for Wafer Fab Cycle Time Management – Tool Uptime," *Semiconductor FabTech, 22nd Edition*, 2004, 29-32.

Abstract: FabTime is focused on improving wafer fab cycle times through consulting, training, and our web-based digital dashboard software. For the past two years, we have conducted an informal survey of the roadblocks to great fab cycle time performance. To date, the top response has been "equipment downtime". In this article, we discuss several best practices for improving fab cycle time by focusing on tool uptime. We begin by summarizing the ways in which equipment downtime events (scheduled and unscheduled) affect cycle time, with emphasis on tool utilization and variability. We then offer suggestions for mitigating these effects. We conclude by proposing a series of uptime-related metrics that, if improved, will tend to improve cycle time, especially for bottleneck and single-path tools. You can subscribe to Fabtech online, at www.semiconductorfabtech.com/.

Call for Papers: The International Journal "Production Planning & Control" Special Issue on Operational Control of Wafer Production

The Production Planning & Control: The Management of Operations international journal plans to publish a special issue on Operational Control of Wafer Production.

This special issue will give the opportunity of putting together high-quality papers in the area of production control with special emphasis on wafer production. Examples of the subject matter of the papers include, but are not limited to, the following:

- Comprehensive reviews and surveys that give an integrated view of the past and present contributions in the operational control aspects of wafer production with insight on the future research needs.
- Development of appropriate tools and techniques for solving various arising the planning, scheduling and control of wafer production.
- Application papers (i.e., actual or potential applications of various operational control techniques to wafer production as a whole or to any of its phases).
- Description and evaluation of software packages available to solve operational control problems in wafer production.

All manuscripts will be promptly and carefully refereed to be published in late 2005 or early 2006. Authors should adhere to Instructions for Authors for the Production Planning & Control journal available at www.tandf.co.uk/journals-/authors/tppcauth.asp when preparing their manuscripts that should be submitted electronically in PS, PDF or DOC formats to one of the following guest editors of this special issue no later than Jan 20, 2005:

Dr. Jatinder N. D. Gupta
College of Administrative Science
University of Alabama in Huntsville
Huntsville, Al 35899, USA
Tel: (256)-824-6593
e-mail: guptaj@uah.edu

Dr. Ruben Ruiz Garcia
Depart. de Estadistica e Investigacion Operativa Aplicadas y Calidad
Universidad Politecnica de Valencia
Camino de Vera s/n (Edificio I-3)
Valencia, 46021, Spain
Tel: (+34)-96-387-70-07, 74946
e-mail: rruiz@eio.upv.es

FabTime welcomes the opportunity to publish community announcements. Send them to newsletter@FabTime.com.

FabTime User Tip of the Month

Set a Default Start Time for New Charts

When you create trend charts in FabTime, you typically want the periods displayed on the chart to coincide with your fab's shift schedule. That is, you want the periods on new charts that you generate to default to start and end at 6:00 am and 6:00 pm, or 7:00 am and 7:00 pm, or whatever is appropriate for your fab, and for your shift. In most cases, this will already be set correctly for you. However, if you work on night shift and you would like to change the default value to be 6:00 pm, or 7:00 pm, or if your charts are defaulting the periods to noon and midnight, then you can change this preference.

To do this, click on the link on your home page left pane for "Email, Passwords, etc." (or, for older versions of the software, "Change My Preferences"). In the text box labeled "Start of Day (Hour):" enter the period start time that you would like to use. That is, for 6:00 am, enter "6". For 7:00 pm, enter "19". If your shifts do not start on the hour, you can use a decimal start time, e.g. for 8:30 am, enter 8.5. Enter your FabTime login password next to the "Password" text box, and click on the "Change" button. The next time you create a new chart from the Chart List, FabTime will use this value as the default period start time. Note that changing this setting will not affect any charts that you already have on your home page.

If you have any questions about this feature (or any other software-related issues), just use the Feedback form in the software.

Subscriber Discussion Forum

FabTime Survey Question: The Oldest Continuously Operating Fab

We were out to dinner with a customer the other night, and talking about the ownership history of a particular fab. During the conversation, we got to wondering, "what is the oldest continuously operating fab?" And we thought that we would pose the question to our subscribers. If you know of a fab that's still operating (regardless of ownership changes), that's been around for a long time, please let us know. We'll go through the responses and report back. Let us know if the fab is owned by a company or is a university lab of some sort, and we'll report back in both categories. Thanks!

Capacity Planning for Tool Sets with Non-Identical Tools

James Ignizio (Intel) wrote: "I've been evaluating the formulas/models used by various firms to compute tool set capacity. So far I've been surprised to discover that none of them are correct for the general case. As such, I wonder if you or your readers could provide (and explain) their Capacity Assessment models/formulas for a tool set in which:

1. There are multiple tools (of different "flavors")
2. The tool set supports multiple operations.

3. Not all tools are qualified for all operations.
4. For a given operation the process times are different according to the specific flavor of tool."

FabTime Response: The formulas do indeed get complicated when you have that general case. We've been working on this with one of our customers, but it gets pretty detailed. We would be happy to include it as a subscriber discussion question, however, to see if any of the other subscribers have anything to contribute.

WIP Utilization, Dynamic X-Factor, and Metrics for the Effect of Tool Downtime

Jimmy Martin (Analog Devices) wrote in response to two recent newsletter topics, and also introduced a new topic. His comments are below.

"**WIP Utilisation (Issue 5.5):** Here at Analog Devices we have been using WIP Utilisation for the past 6 months as a metric on all tools. We have the metric divided into two tool categories: Batch Tools and Wafer Based Tools. At wafer based tools we monitor the WIP in front of the tool and use it to calculate the WIP utilisation of the tool. On batch tools (i.e. furnaces) we use the WIP waiting at the clean step, WIP running on the clean step, and WIP waiting at the furnaces as the WIP Waiting for calculating the WIP utilisation of the furnace. We have found this works very well as it encourages the optimisation of the clean step to ensure the constraint tool does not sit idle. It also has indicated problems at certain bottleneck tools where we were losing significant capacity due to lack of optimisation and conflicts at the feed tool.

Dynamic X-Factor (Issues 4.8 and 5.3): At Analog we use a very similar metric which we call Dynamic Run Time. Definition: (WIP in Process) / Total WIP, where WIP in Process is WIP that is running on tools but not waiting for unload. We find this a good indicator of cycle time in real-time. We can correlate this directly to the output cycle time of the fab. There is typically 2 weeks lag time between the dynamic run time and Fab output cycle time. Typical ratio Dynamic Run Time ratio to x-factors are 50%: 3.2X, or 40%: 4X.

Metrics for the Effect of Tool Downtime: Here at Analog we are looking at what different metrics we can use to indicate the effect of tools down on manufacturing outputs. Does FabTime have any suggestions for such metrics?"

FabTime Response: One thing that we're doing in this area in our software is adding a WIP line to our current tool state trend chart. This lets users filter to say, generate a list of tools that are down, and have WIP waiting in queue in front of them. Then they could perhaps filter further, to find the tools that are down, where WIP has been waiting for more than 6 hours, for example, and flag those for immediate attention. A logical extension to this, to make it into more of a "metric", would be to think about something analogous to WIP Utilization Percentage for downtime, where instead of looking at the standby time, we look at the unscheduled downtime. That is, we break the unscheduled downtime into Down WIP Waiting or Down No WIP Waiting. Down WIP Waiting time is basically a direct addition to cycle time, since the WIP could be being processed if the tool was not down. Similarly for tools doing maintenance while WIP is waiting, vs. when no WIP is waiting. Another option would be to in some way roll this up across the lots in the fab. We discuss this further in the article that follows, and we particularly welcome feedback from other subscribers regarding this issue.

Quantifying the Effect of Tool Downtime

Introduction

In our subscriber discussion forum this month, Jimmy Martin from Analog Devices asked us about metrics for quantifying the effect of tool downtime on manufacturing outputs. We thought that this was a good question, and we've chosen to address it in detail in this month's main article. We first look at the impact of tool downtime on shipments from the fab, and then look in more detail at the impact on cycle time.

Impact of Tool Downtime on Fab Shipments

The most obvious effect from tool downtime comes when looking at top constraint tools in the fab. Constraint tools are the tools that are planned at the highest utilization, and limit fab throughput. For tools that are heavily utilized, downtime can translate directly into a loss in shipments. For example, suppose that the bottleneck (most highly loaded) toolgroup in the fab has four tools. These tools are planned to spend 10% of their time down (16.8 hours per tool per week), and 10% of their time idle (also 16.8 hours per tool per week), with the remaining 80% of their time (168 - 16.8 - 16.8 = 134.4 hours) spent processing wafers. Across the four tools, this means that 537.6 hours/week are scheduled for processing, and that this process time is required to make shipment targets.

But now suppose one of the tools has a catastrophic downtime, and is down for the entire week. Even if we run the other three tools full out, with no idle time, and no downtime, we still have only 168*3 = 504 hours available for processing on the bottleneck tool group. This is 33.6 hours less than planned. If the tools process 10 wafers per hour, then we'll build up a deficit of 336 wafers that were supposed to be processed through the constraint toolgroup this week, but were not. In this particular example, the constraint toolgroup has some planned idle time, and so we may be able to make up this time during the next week. But suppose the constraint toolgroup is planned to run with little or no idle time. In that case, expected downtime translates even more directly into a loss in shipments.

Downtime on non-constraint tools that feed the bottleneck tool groups can also translate into a loss in shipments if it leads to forced idle time on the constraint tools. Measuring this time could help to focus attention on maintaining adequate WIP buffers in front of the constraints.

Impact of Tool Downtime on Cycle Time

Whether or not a tool downtime event will impact fab cycle times depends on whether or not there is WIP waiting for the down tool. So, a first pass in looking at the impact of downtime on cycle time would be to separate the downtime into "downtime with WIP waiting" vs. "downtime no WIP waiting." (This is analogous to our discussion in Issue 5.5 about standby time with WIP waiting vs. standby time without WIP waiting.)

However, we can't just take the length of the "downtime with WIP waiting" and multiply by the average WIP during that time, and call that the cycle time penalty, because some of the WIP would be waiting anyway, even if the tool were available (because the tool would be busy processing other WIP). What we would like to know is the amount of EXTRA cycle time incurred by the WIP due to the downtime. One way to measure this would be to simulate the situation both ways (with and without the downtime), measure the average cycle time in each case, and take the difference. This is somewhat painful to do on an ongoing basis. However, we will look at some simple examples manually, to see if any general patterns or observations emerge.

The Simplest Case: Single-Path, Per-Lot Tool

Here we look at a one-of-a-kind, or single-path, tool that processes WIP one lot at a time. For simplicity, assume the following:

- The tool is down at 8am, and the downtime lasts until 8pm.
- The WIP at 8am is 24 lots.
- The 24 lots all arrived exactly at 8am when the tool went down. (We assume this because any queue time the lots have already incurred will be independent of what happens from 8am forward.)
- No WIP arrives during the downtime. (We are disregarding this WIP for simplicity.)
- The process time is a constant one hour per lot.

Had the tool been up at 8am, cycle times in this case would be: 1 hour, 2 hours, 3 hours, ..., 24 hours. The first lot has no queue time, and a one hour process time. The last lot has to wait while the previous 23 lots are processed, so has 23 hours of queue time, and then one hour of process time, for a total cycle time of 24 hours.

With the tool down for 12 hours, cycle times are 13 hours, 14 hours, 15 hours, ..., 36 hours. The first lot waits in queue for the 12 hour downtime, and then spends one hour being processed, for a total cycle time of 13 hours. The last lot spends 12 hours waiting for the down tool, then 23 hours waiting for the other lots to process, then one hour being processed, for a total of 12 + 23 + 1 = 36 hours.

That is, all 24 lots had their cycle time increased by 12 hours due to the down-time, and the total cycle time created by the downtime is 12 hours/lot * 24 lots = 288 hours. Similarly, if there had been 30 lots in queue at 8am, then the total cycle time created by the 12 hour downtime would be 12 hours/lot * 30 lots = 360 hours. In both cases, we divide this cycle time across all the lots in the fab to get an average cycle time impact due to the downtime of this tool.

Observations from the Single Path Case

In this case, **all** of the WIP in queue when the tool goes down is impacted by the entire downtime, including lots that would not even have been processed during the next 12 hours. This was non-intuitive to us at first, but it does make sense if you think about it. On a single-path tool, the 20th lot cannot process until the 19th lot finishes, which cannot process until the 18th lot finishes, and so on... This goes all the way to the first lot, which is delayed by 12 hours, thus delaying the lot behind it by 12 hours, and so on... back to the 20th lot.

A More Complex Example: Dual Path, Per-Lot Tools

Suppose now that we are looking at a similar example, but the operation in question is qualified to run on two per-lot tools, T1 and T2. Assume the following:

- T1 is down at 8am, and stays down until 8pm.
- 24 lots arrive exactly at 8am, and T2 is available to start processing at this time.
- No WIP arrives during the downtime. (We are disregarding this WIP for simplicity.)
- The process time is a constant one hour per lot.
- Sufficient operators are present to load the tools as required (no operator delays included in this example).

Had T1 been up at 8am, and had both T1 and T2 remained up during the entire shift, then the cycle times for the 24 lots in queue would be: 1 hour (1st lot finishes on T1 at 9am), 1 hour (2nd lot finishes on T2 at 9am), 2 hours, 2 hours, ... 12 hours, 12 hours. The total cycle time incurred by the 24 lots, then, would be 1 + 1 + 2 + 2 + 3 + 3 + 4 + 4 + 5 + 5 + 6 + 6 + 7 + 7 + 8 + 8 + 9 + 9 + 10 + 10 + 11 + 11 + 12 +

12 = 156 hours, or an average of 6.5 hours per lot across the 24 lots.

With T1 down for 12 hours, the cycle times are 1 hour (1st lot finishes on T2), 2 hours, 3 hours, ... 12 hours (12th lot finishes on T2), 13 hours (13th lot finishes on T1), 13 hours (14th lot finishes on T2), 14 hours, 14 hours, 15 hours, 15 hours, 16, 16, 17, 17, 18, 18. That is, the first 12 lots are processed one at a time on T2, and then the remaining 12 lots are processed 2 at a time, one on T1 and one on T2. The total cycle time incurred by the 24 lots is: 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 + 12 + 13 + 13 + 14 + 14 + 15 + 15 + 16 + 16 + 17 + 17 + 18 + 18 = 264 hours, or an average of 11 hours per lot.

Therefore, the total cycle time created by downtime in this example is 264 hours - 156 hours = 108 hours, or 4.5 hours per lot across the 24 lots.

Observations from the Dual Path Case

- The impact of downtime on lot cycle time is significantly higher for single-path tools. When a second tool is available, the impact of downtime is much less dramatic than for the single path case (4.5 hours/lot vs. the 12 hours/lot from the first example).
- This improvement comes from the fact that half of the WIP runs through the second tool with essentially no cycle time impact. We also have quicker processing once T1 is back up, because there are two tools available.
- We believe that it should be possible to formulate this to get a simple equation based on L=number of lots in queue at downtime, N=number of tools, and PT= average per lot process time.
- With a simple formula, it would be possible to record the impact on cycle time of tool downtime just by looking at WIP in queue at the start of the downtime, the number of qualified tools, and the process time. We open this up in particular to our academic readers. We think that this could make an excellent student project.

Preliminary Notes on the Formulas

Let:

L = WIP in queue for some set of qualified tools

DT = Length of the downtime on the first tool in the group

N = Number of qualified tools

PT = Per lot process time for lots in queue (assume identical for all lots, all tools)

If $N = 1$, then, as in the above example, the average queue time incurred per lot due to the downtime is equal to DT (the length of the downtime). The total cycle time caused by the downtime = $L * DT$. Note that this is independent of PT (for the single tool case).

If $N = 2$, then we have one tool available during the downtime. Assume that this second tool does not go down itself during the time that the L lots are present.

$L * PT$ = process time required to process the WIP currently in queue.

If $L * PT < DT$, then we can process all the WIP in queue on the second tool during the downtime of the first tool. Then we need to compare what the cycle time would have been if the lots were processed on both tools to what it would be just being processed on the second tool.

For a single tool processing L lots with process time PT for each lot, the average cycle time is (from the formula for the sum of a finite integer series):

Avg. CT (one tool) = $((L+1)/2)*PT$

and the total cycle time for all the lots (one tool) is $L*((L+1)/2)*PT$.

For two tools processing L lots, if L is an even number, then we assume that each tool processes one-half of the lots, and we can substitute $L/2$ for L in the above formula, to get:

Avg. CT (two tools) = $(((L/2)+1)/2)*PT$

and the total cycle time (two tools) is $L*(((L/2)+1)/2)*PT$.

Therefore, the difference in total cycle time, for the case where we can process all of the WIP on the second tool during the downtime, and we have an even amount of WIP, is:

$L*((L+1)/2)*PT - L*(((L/2)+1)/2)*PT$

$= L*PT * ((L+1)/2 - (((L/2)+1)/2))$

$= L*PT * ((L+1)-((L/2)+1))/2$

$= L*PT * L/4$

So, if in a modification to the above example the WIP was 12 lots, and the process time was 1 hour, then we could process all of the WIP on T2 during the downtime of T1. In this case, the total cycle time to process the 12 lots on the two tools (if there were no downtime) would be:

$L*(((L/2)+1)/2)*PT = 12*((6+1)/2)*1 = 12*3.5 = 42$ hours.

The cycle time with the downtime, to process all of the WIP on T2, would be:

$L*((L+1)/2)*PT = 12*((12+1)/2)*1 = 12*6.5 = 78$ hours.

The difference in total cycle time is:

$L*PT*L/4 = 12*1*12/4 = 144/4 = 36 = 78-42$.

The difference in average cycle time is $36/12 = 3$ hours/lot.

These formulas need to be extended to the more general case, but the methodology would be similar.

How Does this Translate to Shipped Lot Cycle Time?

This type of formula will measure the impact on cycle time through a particular operation for lots already in queue at the start of the downtime. There is no guarantee that this per-operation cycle time will translate directly into shipped lot cycle time. For example, if the tools in the above examples feed some downstream operation with a huge queue in front of it, then their total cycle time through both operations might remain the same, regardless of the downtime. However, we believe that any improvements in per-operation cycle times are likely to translate into shipped lot cycle times because:

a) The downstream tool might not have a queue in front of it, and so the lots will keep moving through the fab.

b) Even if the downstream tool is a bottleneck, it's better to keep the WIP in front of the bottleneck rather than sitting in front of other operations. This reduces the chance of the bottleneck starving, and improves dispatching choices at the bottleneck.

With throughput improvement projects, you have to focus on the bottleneck tools. One of the nice things about cycle time improvement programs is that you can generally improve cycle time by reducing queue time anywhere in the fab.

Conclusions

We all know that if we could eliminate downtime from our fabs, we could increase throughput (where the constraint tools have any downtime at all), and improve cycle time at the same time. Down tools lead to WIP bubbles, increased variability, and in general, queue time for lots that are waiting for tools to come back up. In this article, we have made a first pass at quantifying this impact more formally, by measuring the increased operation-level cycle time for lots that are in queue when a tool goes down. Better understanding the cycle time cost from specific downtime events could be helpful for fabs in deciding where to focus tool improvement efforts. It will also likely point to the disproportionate effect of downtime on single path tools, and perhaps provide further justification for tool flexibility / cross-qualification projects.

Acknowledgement

We thank Jimmy Martin of Analog Devices for his subscriber discussion question, which inspired us to write this article.

Closing Questions for FabTime Subscribers

Do you have any metrics that you use to quantify the impact of downtime on fab shipments, or on fab cycle time?

Subscriber List

Total number of subscribers: 1656, from 396 companies and universities. 25 consultants.

Top 10 subscribing companies:

- Analog Devices (81)
- Intel Corporation (80)
- Motorola Corporation (53)
- Infineon Technologies (51)
- STMicroelectronics (51)
- Philips (43)
- Micron Technology (41)
- Seagate Technology (41)
- Texas Instruments (39)
- AMD/Spansion (35)

Top 5 subscribing universities:

- Arizona State University (10)
- Virginia Tech (10)
- Georgia Tech (6)
- Nanyang Technological University (6)
- University of California – Berkeley (6)

New companies and universities this month:

- Ashok Leyland Ltd.
- Aviza Technology
- Banc of America Securities
- Silicon Strategies
- Spectra Physics
- Synopsys Inc.

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 available for a small fee from FabTime's Amazon zShop, at www.amazon.com-/shops/fabtime.

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® Cycle Time Management Software

"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
AMD Fab 25

FabTime Installation

One fixed price includes

- Site license, unlimited users.
- Implementation & training.
- Software maintenance.

Pilot Project – Analyze your data with FabTime

For $4950, FabTime will

- Identify key contributors.
- Benchmark common metrics.
- Review results at your site.

Interested?

Contact FabTime for technical details or a pilot project quote.

FabTime Inc.
325M Sharon Park Drive #219
Menlo Park, California 94025
Phone: +1 (408) 549-9932
Fax: +1 (408) 549-9941
Email: Sales@FabTime.com
Web: www.FabTime.com

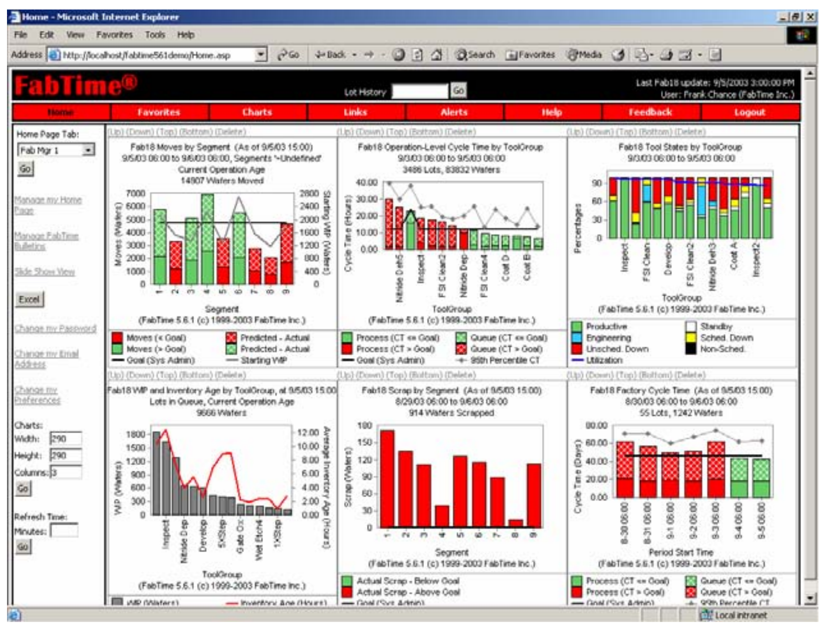
Do you have the best possible information?

- Are your supervisors swamped with daily reports, but lacking real-time information?
- Is it difficult to link equipment performance to cycle time?
- Does each new cycle time analysis require IT resources?

FabTime is a digital dashboard for your fab. In real-time, it provides a comprehensive view of fab performance data – everything you need for proactive management of cycle time. FabTime is designed for hands-on use by managers and supervisors, unlike traditional reporting tools, which were designed for programmers.

A Web-Based Digital Dashboard

"I use FabTime every day, and so do the supervisors who report to me. The data that I need is right on my home page where I need it when I come in every morning."
Jim Wright
Production Manager
Headway Technologies



FabTime Benefits

- Cut production cycle times by 10%, hot lot cycle times by 20%.
- Focus improvement efforts on the tools that inflate cycle time.
- Improve supervisor productivity – cut reporting time by 50%.