# FabTime Cycle Time Management Newsletter

Volume 9, No. 8 October 2008

## 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 this month include the addition of number of holds, total hold time, number of moves, and moves per day of factory cycle time to shipped lot list data table, and the addition of employee to the scrap lot list data table.

**Editor:** Jennifer Robinson

**Contributors:** Professor Scott Mason (University of Arkansas)

## Table of Contents



## Welcome

Welcome to Volume 9, Number 8 of the FabTime Cycle Time Management Newsletter! I hope that this issue finds you all well, and weathering any recent financial storms safely. We've been keeping busy with customer installations and dispatching work. We're also doing early testing of a very cool custom chart functionality, which we expect to be a big hit with FabTime users.

In this issue, we have a community announcement about a deadline extension for abstracts for the Advanced Semiconductor Manufacturing Conference. Our FabTime software user tip of the month is about separating out the components of non-process time (queue time, hold time, etc.). We have no subscriber discussion this month. However, we do suspect that this month's main article will inspire some discussion for the next issue. We have asked occasional contributor Professor Scott Mason to write about tool state calculations for cluster tools. Professor Mason discusses two primary methods for estimating overall cluster tool performance, one based on logical rules and the other based on averages (possibly weighted) across chambers. He gives several examples, and shows through these examples how different the overall results can be depending on the calculation method used. He concludes that the wide disparity of results begs the question: how are cluster tool E10 states computed in your fab? We welcome your feedback, and we are happy to share it anonymously if you prefer.

Thanks for reading!—Jennifer

FabTime

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

# Community News/Announcements

### Call for Abstracts: Advanced Semiconductor Manufacturing Conference

Abstracts are due October 13 for the IEEE/SEMI Advanced Semiconductor Manufacturing Conference (ASMC 2009). The conference will be held in cooperation with ISSM, May 10-13, 2009 in Berlin, Germany. SEMI and IEEE have extended the deadline for submitting abstracts for the longest running-semiconductor technical conference—ASMC. ASMC 2009 focuses on the major issues that impact the performance of manufacturing facilities, including:

- Day-to-Day Fab Operations Challenges
- Defect Reduction
- Emerging Technologies, including PV/Solar
- Equipment Performance
- Factory Automation and Operations
- Line Utilization Issues
- Process and Materials Optimization Yield Improvement and Yield Modeling

The 2009 conference is co-chaired by Walter Schoenleber, Applied Materials and Brett Williams, ON Semiconductor. For the complete call for papers and guidelines, visit the ASMC homepage at www.semi.org/asmc2009.

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

# FabTime User Tip of the Month

### Breaking Out Average Queue Time from Other Non-Process Time

We had a question from a FabTime user the other day that we thought might be of interest to other users, too. Do you ever need to know average queue times, as separate from hold times? The Operation Cycle Time Trend and Pareto charts show per-operation cycle times, averaged across the time period or slice variable on the chart. The Operation Cycle Time Charts separate average process time from average non-process time (provided a BeginRun or StartProcess time was logged). The non-process time shown on the chart is the time from MoveOut of the prior operation until BeginRun at the current operation. As such, it includes queue time, hold time, travel time, etc. It can also, for automated tools, include post-process time, which is the time from EndRun to MoveOut.

FabTime does store this more detailed information (hold time, post-process time, etc.). You can find it in the data table on the Moves Lot List Chart. For each lot move, the data table breaks cycle time down into queue time, process time, post-process time, hold time, and other time. If you need to aggregate this data you can create the Moves Lot List chart for the time range and set of filters that you're interested in (e.g. for a particular operation). Then you can export the data to Excel and calculate the average for the column of interest (e.g. queue time or hold time). Just remember, before you do the Excel export, you will probably need to edit the number of rows displayed in the table to a large number (FabTime displays only the top 25 data rows by default - use the "Rows" control to change this). You may also wish to change the chart to display time in hours or minutes, instead of the default days. You can do this by changing the "U/M" drop-down to the left of the chart, near the bottom of the primary set of filters.

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 welcomes the opportunity to publish subscriber discussion questions and responses. Send your questions to Jennifer.Robinson@FabTime.com. We have no subscriber discussion this month.

# Tool State Calculations for Cluster Tools in Fabs

## Introduction

**Written by Professor Scott Mason from the University of Arkansas, with contributions by Jennifer Robinson from FabTime**

In this article we examine calculating tool state-related performance metrics for cluster tools. Based on our review of available SEMI standards and discussions with FabTime customer sites, we provide possible methods for calculating these performance measures. We hope this initial discussion motivates further responses from readers with respect to how this is currently done in practice, as our initial conversations did not reveal any one dominant method of computing cluster tool mainframe tool state metrics.

## Background

In an earlier newsletter (Issue 8.06), we provided an overview of cluster tools, presenting various definitions of what these rapidly emerging toolsets are and how they function in the fab. Multi-functional cluster tools are comprised of a number of configurable processing chambers surrounding a central robotic handling unit. Srinivasan (1998) effectively defines a cluster tool as "an integrated, environmentally isolated manufacturing system consisting of process, transport, and cassette modules mechanically linked together."

While the incorporation of cluster tools into a wafer fab can increase throughput and improve floor space utilization, the failure of a single processing chamber or the central mainframe can cause an entire cluster tool to be down, depending on the configuration of the chambers. In this article, we hope to start a discussion on how cluster tool chamber states are monitored and measured in practice. This discussion is motivated by our desire to understand how overall cluster tool SEMI E10 tool states are calculated in practice and how these values can be used to determine overall equipment efficiency (OEE) for cluster tools.

SEMI E10 defines single-path cluster tools as "a cluster tool with only one process flow path (as used)." More common in practice, however, are multi-path cluster tools containing "more than one independent process flow path (e.g., multiple load ports/load-locks, multiple process chambers of the same type." In this article, we focus our discussions and calculations on the multi-path cluster tools, as the more simplified single path cluster tool can be analyzed using the proposed multi-path tool approaches. With this in mind, we now present alternatives for measuring cluster tool performance.

## Tool State Calculations and OEE for Cluster Tools

Section 6 in SEMI Standard E79-0304 presents definitions and formulas for productivity assessment of flexible-sequence (i.e., multi-path) cluster tools. The SEMI approach is to examine each individual processing chamber/module separately, and then determine the "productivity performance for the entire flexible-sequence cluster tool … as the aggregate productivity performance of its individual processing modules." In addition, Note 5 suggests that "evaluation of flexible-sequence cluster tool productivity does not necessarily apply to the evaluation of flexible-sequence cluster tool RAM." In other words, while cluster tool OEE is apparently computed by aggregating individual processing chamber/module performance, this is not the recommended approach/strategy for computing overall E10 states for the cluster tool. This is our motivating question in this article: How are/should tool states be calculated for a cluster tool "main frame"?

## A Potential Approach: Transaction-Based Logic Rules

At any point in time, an operator/fab personnel/MES can observe and record the current SEMI E10 state of each cluster tool chamber (e.g., productive, standby, engineering, scheduled down, unscheduled down, or non-scheduled). Then, based on the state of each chamber, the overall cluster tool mainframe's state could be established based on the current state for all of its component parts via some sort of logic or lookup table. Finally, any time a module/chamber's state changes via a tool-related MES transaction, the state of the overall cluster tool would also change. Over time, this would result in various percent occurrences of the overall cluster tool mainframe being in each of the SEMI E10 states.

For example, consider the following possible set of logic rules for a three-chamber, multi-path cluster tool. (Please note that we're not at all proposing this particular set of logic rules. It's been our experience that different sites apply these types of rules differently, and we are presenting our own made-up logic for illustration):

- If two or more chambers are productive, the tool is productive, regardless of the state of the other chamber.
- Else If only one chamber is productive, then the tool is 50% productive.
  - If the other two chambers are in the same state, then the tool is 50% in that state.
  - If either of the other two chambers is engineering, then the tool is 50% engineering.
  - Else If either of the other two chambers is scheduled down, then the tool is 50% scheduled down.
  - Else If either of the other two chambers is unscheduled down, then the tool is 50% unscheduled down.
- Else If none of the chambers is productive, if any one state has a majority, the tool is 100% in that state.
  - Else if none of the chambers has a majority, and any chamber is in engineering, then the tool is 100% in engineering.
  - Else if any chamber is scheduled down, then the tool is 100% scheduled down.
  - Else if any chamber is unscheduled down, then the tool is 100% unscheduled down.

Now, let's follow a numeric example. By observing the state of each of the three chambers over the last 24 hours, applying these logic rules results in the "Mainframe" status below:

| Time | Chamber 1 | Chamber 2 | Chamber 3 | Mainframe |
|---|---|---|---|---|
| 10/1/08 12:00 AM | Productive | Productive | Productive | Productive |
| 10/1/08 1:46 AM | Productive | Productive | Unsched DT | Productive |
| 10/1/08 1:57 AM | Productive | Standby | Unsched DT | Half Prod, Half Unsched DT |
| 10/1/08 3:18 AM | Productive | Unsched DT | Unsched DT | Half Prod, Half Unsched DT |
| 10/1/08 3:46 AM | Schedule DT | Unsched DT | Unsched DT | Unsched DT |
| 10/1/08 4:05 AM | Unsched DT | Unsched DT | Unsched DT | Unsched DT |
| 10/1/08 4:39 AM | Productive | Unsched DT | Unsched DT | Half Prod, Half Unsched DT |
| 10/1/08 5:40 AM | Standby | Unsched DT | Unsched DT | Unsched DT |
| 10/1/08 5:47 AM | Standby | Unsched DT | Productive | Half Prod, Half Unsched DT |
| 10/1/08 6:09 AM | Standby | Unsched DT | Schedule DT | Sched DT |
| 10/1/08 6:50 AM | Productive | Unsched DT | Schedule DT | Half Prod, Half Sched DT |
| 10/1/08 7:29 AM | Productive | Unsched DT | Standby | Half Prod, Half Unsched DT |
| 10/1/08 8:12 AM | Productive | Standby | Standby | Half Prod, Half Standby |
| 10/1/08 8:57 AM | Productive | Standby | Productive | Productive |
| 10/1/08 10:42 AM | Engineering | Standby | Productive | Half Prod, Half Engineering |
| 10/1/08 12:07 PM | Engineering | Productive | Productive | Productive |
| 10/1/08 1:01 PM | Engineering | Unsched DT | Productive | Half Prod, Half Engineering |
| 10/1/08 2:01 PM | Productive | Unsched DT | Productive | Productive |
| 10/1/08 3:27 PM | Schedule DT | Unsched DT | Productive | Half Prod, Half Sched DT |
| 10/1/08 4:33 PM | Schedule DT | Engineering | Productive | Half Prod, Half Sched DT |
| 10/1/08 5:55 PM | Schedule DT | Schedule DT | Productive | Half Prod, Half Sched DT |
| 10/1/08 6:00 PM | Schedule DT | Schedule DT | Schedule DT | Sched DT |
| 10/1/08 7:22 PM | Engineering | Schedule DT | Schedule DT | Sched DT |
| 10/1/08 7:54 PM | Engineering | Schedule DT | Productive | Half Prod, Half Engineering |
| 10/1/08 8:02 PM | Engineering | Schedule DT | Standby | Engineering |
| 10/1/08 9:25 PM | Productive | Schedule DT | Standby | Half Prod, Half Sched DT |
| 10/1/08 10:47 PM | Productive | Standby | Standby | Half Prod, Half Standby |
| 10/2/08 12:00 AM | Productive | Standby | Unsched DT | Half Prod, Half Unsched DT |

Notice that applying the logic rules in this manner results in very little standby time for this tool (since the tool is only standby if two or more chambers are in a standby state. This illustrates the way that such logical rules can be applied to bias results toward particular tool states. It is also interesting to note that the application of our logic rules is process flow/product mix independent —- we are simply observing the states of the individual chambers, and then applying our logic rules to get the cluster tool's mainframe state.

Based on these mainframe categorizations, we can sum up the time the tool (mainframe) spent in each state as follows:

| State | Productive | Standby | Sched DT | Unsched DT | Engineering |
|---|---|---|---|---|---|
| Time | 12:36:18 | 1:02:06 | 4:48:18 | 3:50:24 | 1:42:54 |
| Pct Time | 52.5% | 4.3% | 20.0% | 16.0% | 7.1% |

**Editor's Note:** Of course, there are many different possible sets of logic rules that could be applied. For example, we could have had a more complex set of logic rules, with things like "If Chamber 1 and Chamber 2 are Productive, then the tool is Productive" or "If Chamber 1 is in Unscheduled DT, then the whole tool is in Unscheduled DT, even if Chamber 2 is Productive." The more complex the set of logic rules, however, the more difficult they will be to implement in reporting software, and the more at-risk they will be for logical errors.

## Another Approach: Historical-Based Calculations

Instead of applying logic rules to transactional data, another approach is to calculate chamber-specific performance metrics over some period of time, and then aggregate the chambers' values in some manner. This is the approach that coincides with the SEMI E79-0304 standard. We will examine two potential methods of aggregation in the example below.

Examining the three-chamber cluster tool's performance over the same 24 hour period as in Approach 1, we compute the following total amounts of time each chamber spent in each of the five SEMI E10 states of interest:

| | Productive | Standby | Schedule DT | Unsched DT | Engineering |
|---|---|---|---|---|---|
| Chamber 1 | 12:39 | 1:09 | 4:14 | 0:34 | 5:22 |
| Chamber 2 | 2:51 | 6:30 | 4:52 | 8:25 | 1:21 |
| Chamber 3 | 11:18 | 5:25 | 3:15 | 4:01 | 0:00 |

These total amounts of time correspond to the following percentage breakdown (with each chamber having the same 24 total hours of time):

| | Productive | Standby | Schedule DT | Unsched DT | Engineering |
|---|---|---|---|---|---|
| Chamber 1 | 52.7% | 4.8% | 17.7% | 2.4% | 22.4% |
| Chamber 2 | 11.9% | 27.1% | 20.3% | 35.1% | 5.7% |
| Chamber 3 | 47.1% | 22.6% | 13.5% | 16.7% | 0.0% |

Now, we consider two different aggregation schemes for determining overall cluster tool mainframe performance. First, if all three chambers are considered equal/equivalent, a simple average across all three chambers for each E10 state results in the following:

| | Productive | Standby | Schedule DT | Unsched DT | Engineering |
|---|---|---|---|---|---|
| Mainframe | 37.2% | 18.2% | 17.2% | 18.1% | 9.4% |

However, consider the case when Chambers 2 and 3 are identical, but are both different from Chamber 1. In this case, it may be more appropriate to average Chambers 2 and 3 together first, and then average these values with Chamber 1. In effect, this is computing a weighted average of the three chambers with 50% weight placed on Chamber 1 and 25% weight placed on each of Chambers 2 and 3. Performing this calculation results in the following aggregated cluster tool mainframe performance metrics:

| | **Productive** | **Standby** | **Schedule DT** | **Unsched DT** | **Engineering** |
|---|---|---|---|---|---|
| Mainframe | 41.1% | 14.8% | 17.3% | 14.1% | 12.6% |

Clearly, other weighting schemes could also be used when aggregating cluster tool chamber performance metrics.

## Conclusions

In this article, we have reviewed different methods for computing cluster tool mainframe tool state metrics. Conversations with industry personnel suggest that no single method is used in practice to compute cluster tool mainframe tool states. In fact, one company did not wish to disclose their method as it is closely guarded intellectual property. Consider the three methods suggested in this article in terms of the resulting performance metrics for the same cluster tool:

| Method | **Productive** | **Standby** | **Schedule DT** | **Unsched DT** | **Engineering** |
|---|---|---|---|---|---|
| Logic Rule | 52.5% | 4.3% | 20.0% | 16.0% | 7.1% |
| Average | 37.2% | 18.2% | 17.2% | 18.1% | 9.4% |
| Wtd. Avg. | 41.1% | 14.8% | 17.3% | 14.1% | 12.6% |

Given such a disparity of results, this begs the question: how are cluster tool E10 states computed in your fab? One of these three approaches? A different method? We are not advocating using only a 24 hour period of time to compute these metrics, as this was only used for illustration purposes. However, the fact that the Productive state varies by over 15% across the three methods and the Standby state varies from 4% to 15% is reason enough to hopefully motivate a discussion among newsletter readers as to methods and best practices for calculating cluster tool mainframe tool states.

## Closing Questions for FabTime Subscribers

How do you compute cluster tool chamber tool state performance measures? How do you use the chamber tool state metrics to compute values for the entire cluster tool mainframe? How are these values used in determining OEE values for your cluster tools?

## Further Reading


- SEMI E10-0304, 2004, Specification for definition and measurement of equipment reliability, availability, and maintainability (RAM).
- SEMI E79-0304, 2004, Specification for definition and measurement of equipment productivity.
- Srinivasan R.S., 1998, Modeling and performance analysis of cluster tools using Petri nets, *IEEE Transactions on Semiconductor Manufacturing*, 11(3), 394-403.

# Subscriber List

**Total number of subscribers:** 2855, from 476 companies and universities. 22 consultants.

**Top 20 subscribing companies:**

- Maxim Integrated Products, Inc. (233)
- Intel Corporation (155)
- Micron Technology, Inc. (81)
- Chartered Semiconductor Mfg (80)
- X-FAB Inc. (73)
- Western Digital Corporation (67)
- Analog Devices (65)
- Freescale Semiconductor (61)
- Infineon Technologies (61)
- ON Semiconductor (58)
- Texas Instruments (58)
- International Rectifier (55)
- NEC Electronics (54)
- TECH Semiconductor Singapore (53)
- Cypress Semiconductor (52)
- STMicroelectronics (50)
- IBM (45)
- NXP Semiconductors (44)
- ATMEL (36)
- Spansion (36)

**Top 3 subscribing universities:**

- Virginia Tech (11)
- Ben Gurion Univ. of the Negev (8)
- Nanyang Technological University (8)

**New companies and universities this month:**

- Finisar
- Fort Dodge Animal Health
- GE Healthcare
- Nanosolar

**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 Capacity Planning Module

## CP Configuration

We offer our dispatching and planning modules together for a single, fixed monthly fee (on top of your regular FabTime subscription). This includes:

- Identification of the source of any additional data needed for the planning module.
- Automation of the process of importing the additional data into FabTime.
- Validation against client data.

## Interested?

Contact FabTime for more information, or for a quote.

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

## Do you need to answer questions like:

- Given a target product mix, do we need any new tools?
- Given the tools that we have, and the products that we are running, how many wafers can we expect to produce?
- Given our existing set of products and tools, what happens if the product mix changes? Where can we expect bottlenecks?

## Are you tired of maintaining a standalone capacity planning spreadsheet?

FabTime's capacity planning module leverages the data already stored in the FabTime digital dashboard software, to make it easier to build capacity planning scenarios. The only required manual inputs are:

- Weekly ships per product.
- Product line yield percentages.

FabTime uses route information from the fab MES and calculates UPH data (tool speed) based on actual performance. FabTime also uses tool uptime performance to estimate availability (though this can be overridden). These inputs are used to generate predicted utilization percentages for each capacity type. Detailed intermediate calculations (UPH, tool productive time, tool rework percentage, etc.) are also available (an example for one tool is shown below). All outputs can be easily exported to Excel.

## Capacity Planning Module Benefits

- Eliminate the need to maintain offline capacity planning models.
- Automatically update capacity planning data to reflect new conditions (process flows, tool uptime characteristics).
- Quickly run scenarios to anticipate (and avoid) bottlenecks caused by product mix changes.

| C Type | Output | Value | Notes |
|---|---|---|---|
| 1XStep | Rework Moves/Week | 21 | 2004-09-06 10:00:00 to 2004-11-15 10:00:00 |
| 1XStep | Total Moves/Week | 12310 | 2004-09-06 10:00:00 to 2004-11-15 10:00:00 |
| 1XStep | Rework Ratio | 0 | Rework Ratio = Rework Moves / Total Moves. |
| 1XStep | Productive% | 61 | 2004-09-06 10:00:00 to 2004-11-15 10:00:00 |
| 1XStep | Availability% | 76.26 | Availability = Productive% + Standby%. |
| 1XStep | Historic Utilization% | 79.99 | Utilization (Mfg efficiency) = Productive% / Availability%. |
| 1XStep | Productive(Rework)% | 0.1 | Productive(Rework)=Productive% * ReworkRatio. |
| 1XStep | Net Availability% | 76.15 | Net availability% = Availability% - Productive(Rework)%. |
| 1XStep | Arrivals (Units/Hour) | 79.36 | Based on total plan WGR=2025 |
| 1XStep | Tool Quantity | 8 | 1XStep#1 ... 1XStep#8 |
| 1XStep | UPH | 15.02 | UPH = (TotalMoves/ToolQty) / (Productive% * 168) |
| 1XStep | Required Hours/Day | 126.84 | Required hours = 24 * HourlyArrivalRate / UPH |
| 1XStep | Predicted Utilization% | 86.75 | Util = 100 * ReqdHours / (24 * NetAvail * ToolQty / 100) |
| 1XStep | Max WGR | 2334.22 | MaxWGR = PlanWGR / PredictedUtilization |
| 1XStep | Historic WGR | 2457.8 | (Non Rework Moves) / (OperationCount / ProductCount). |