Changing the Methodology for DFM
Moving from Design-Checker to Interactive, Informed Design Methodologies

Nolan Johnson
Product Marketing Manager
Sunstone Circuits

Abstract

The phrase "Design For Manufacture" strongly implies the use of manufacturing information while making of design decisions or, in other words, incorporating manufacturing knowledge during initial design. Unfortunately, most currently available PCB DFM tools don't fit into the design work flow until the PCB design is already complete. There are three nasty consequences to the current approach:

1. Designers are forced to make under-informed design decisions during layout, when DFM is easiest and least expensive to design in to the project.
2. When DFM errors are inevitably identified, they require significant rework, engineering effort and additional money.
3. The additional rework to fix back-end batch DFM increases the risk that your project will miss the market window.

New, more interactive approaches to DFM are becoming available to designers. Rather than postponing DFM checks until the end of the design, for example, the methodology described here enables designers to be notified (almost interactively) of any manufacturability issues as they design. By attending to DFM issues from the onset, designers are not only able to 'pass' DFM, but can also invest in optimizing the manufacturability of their designs even from the first prototype. The result can be a cleaner, more manufacturable design that qualifies for production faster and at lower cost, and that also produces higher yields in production.

In this paper, we present case studies and efficiency results, from their implementation of an interactive DFM model to support our manufacturing process.

Introduction

IC verification and PCB verification both have historically pushed forward the state of the art for Design for Manufacture. Twenty years ago, the state of the art was mostly traditional Design Rule knowledge – trace-and-space, shorts and opens – while today's complexities sometimes require a more detailed analysis of the design. In IC, the need to close the loop between parasitic effects and the original design's timing specifications is an example of this type of DFM analysis. In the PCB environment, while electrical rules and thermal requirements play an important part in the current trend technical development, most trends are toward panel-sized Lithography rules.

In this paper, we will examine reasons why the state of Design for Manufacturability (DFM) and Design for Yield (DFY) in the PCB design environment is not representative of the current stat of the art for Design for Manufacturability in general. The tools and processes that both bridge design and manufacture, and exist in widely-available, easy-to-acquire formats, are not sufficient for a fully-implemented DFM design flow.

Methodology

For this paper, we will be relying on collected industry research, interviews and discussions with PCB designers, and – to a limited degree – first-hand experience at Sunstone Circuits, especially on the manufacturing floor.

Discussion

The Role of DFM/DFY

DFM/DFY ultimately concentrates on enabling designers to hit the center of the allowed process windows. In so doing, minor process deviations will not jeopardize yield targets.[1]

Design teams are increasingly looking to DFM/DFY tools as a methodology to 1) capture and describe these process window relationships, 2) compare these relationships to their designs in quantifiable way, and 3) report this information back to the designer such that they can make use of it to do their jobs better

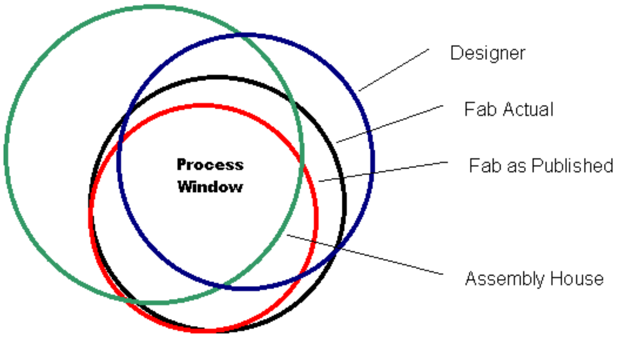

Figure 1: supply chain misalignments in process window can shrink the project's overall process window.

DFM/DFY's role is to provide a feedback loop for the engineering team. "While it is clear to most people in the industry that design and manufacture need to be more closely coupled than ever before, many remain unclear on the exact reasons for why Design-for-Yield (DFY) is important today; what the tie is to systematic yield loss; what additional investment is needed, if any, to make DFY a reality; and who stands to benefit from DFY."

The trends in PCB DFM are not unique. IC is undergoing similar challenges in lining up software technology with the current state of the art in manufacturing. For example, in Chip Design Magazine (June/July 2006), Behrooz Zahiri, speaking about IC DFM issues, writes:

[1] Nolan Johnson, Printed Circuit Design & Fabrication, December, 2007

“To employ an effective DFM/DFY solution, IC designers must address manufacturing or yield problems caused by the catastrophic or parametric failures that are either systematic (feature-driven) or statistical (random) in nature. Unfortunately, existing design flows don't adequately address these DFM or DFY problems for 90-nm and below technology nodes.

“A … design flow is needed in which all of the design and analysis engines are DFM/DFY-aware.”

Zahiri goes on to write:

In the past, the design and manufacturing worlds have been treated as separate, distinct entities. Until now, designers have been shielded from the intricacies of the fabrication process by the use of "design rules" and "recommended rules" provided by the foundry. In earlier technology nodes, designers could safely assume that the chip could be manufactured if they and their tools rigorously met these rules. Any yield problems were considered the foundry's responsibility and were addressed by improving the capabilities of the fabrication process or bringing that process under tighter control…these rules no longer reflect the underlying physics of the fabrication process. [2]

As IC designers and software developers continue to push on the capabilities of IC verification, further merging manufacturing knowledge with the design process, PCB designers face similar needs and pressures. These pressure dynamics, driven by increasing layer counts, decreasing spacing clearances, more complex packaging of components (small SMTs, BGAs, etc.), the interrelationships and impact of manufacturing issues are now so important as to require significant pre-planning and forecasting to ensure a working end product. Zahari’s article points the direction of the path that PCB DFM/DFY will shortly be following.

It makes sense, then to spend some time looking at how we got to this point, so as to gain a better perspective of where we’re going.

DFM – AN EDA Timeline

PCB design checking tools have been present since the early 1980’s; IC design checking tools can trace their roots to programs that emerged out of R&D teams in the late 1970’s and 1980’s. Given the close relationship between chip design and PCB board design, a number of trends have tracked closely between the two disciplines though the majority of the innovation has taken place in the IC sector first, then propagated to the PCB tools space.

DFM Error Types

DFM Errors are commonly categorized by a two-by-two matrix, as follows:

- Catastrophic – board design is non-functional
- Parametric – board functions, but doesn’t perform to specification
- Systemic – functionality does not work as planned in a consistent and reproducible way
- Statistical – failures are seemingly random or does not correlate to another condition

	Catastrophic	Parametric
Systemic	Board Design errors Functional Spec errors PCB layer mapping Assembly errors	Functional Spec errors Component mismatch Timing errors in PCB
Statistical	Design errors process yield issues Component specs	Assembly placement Process window /yield

Figure 2: Manufacturing Failure types, by category

Design for Yield issues tend to concentrate on the Statistical errors; DFM tends to concentrate on the Systemic errors. Geometric DRC tends to concentrate on identifying systemic/catastrophic errors.

[2] Zahiri, Behrooz, “True DFM/DFY Solutions Require More Than OPC and DRC”, **Chip Design,** June/July 2006

Factors Influencing DFM Trends

Internet Access

The availability of network drops of all types has become practically ubiquitous, particularly in North American technology facilities. According to Internet World Stats, (see figure 3) which tracks internet usage and population statistics, Internet penetration in North America is at 70%, well beyond the point of 'critical mass' for acceptance as described by HS Dent's S-Curve of product adoption[3]. Using Dent's model, we can see that only the continent of Africa has yet to reach Dent's point of critical mass.

World Internet Usage And Population Statistics						
World Regions	Population (2007 est.)	Population % of World	Internet Usage, Latest Data	% Population (Penetration)	Usage % of World	Usage Growth 2000-2007
Africa	933,448,292	14.2%	43,995,700	4.7%	3.5%	874.6%
Asia	3,712,527,624	56.5%	459,476,825	12.3%	36.9%	302.0%
Europe	809,624,686	12.3%	337,878,613	41.7%	27.2%	221.5%
Middle East	193,452,727	2.9%	33,510,500	17.3%	2.7%	920.2%
North America	334,538,018	5.1%	234,788,864	70.2%	18.9%	117.2%
Latin America/ Caribbean	556,606,627	8.5%	115,759,709	20.8%	9.3%	540.7%
Oceana/Australia	34,468,443	0.5%	19,039,390	55.2%	1.5%	149.9%
WORLD TOTAL	6,574,666,417	100.0%	1,244,449,601	18.9%	100.0%	244.7%

Figure 3: Internet access can increasingly be expected in corporate environments worldwide

This global growth in Internet access has a number of influences on the development of software, including:

- New distribution models using digital/Internet delivery over physical media
- ASP-based or "on the net" tools such as Google Office, SalesForce, etc.
- Use of a distributed client/server database storage/access model rather than relying on data local to the user's client PC.
- Improved technical support and communications.

Applications that rely upon databases that are either large, or highly-shared, have demonstrated the most opportunity to benefit from a client/server internet connection, including:

People Directories – online phone directories, social networking sites
Sales CRM applications – Salesforce.com, Microsoft CRM
Search Engines – google, yahoo, etc.
Parts Catalogs – parts distributors, parts database providers..

Changes in Software Distribution

According to a report by Macrovision, Key Trends in Software Pricing And Licensing, from November 2007, the key issues for software vendors are providing [on a scale of 1 – 5]:

• Software as a Service	[3.4]
• Integration of Applications	[3.4]
• Electronic Software Delivery	[3.3]

In contrast, migrating to Windows Vista rated a 2.5.

In the business software industry as a whole, companies are pushing vendors to provide mechanisms that ensure software licensing compliance and auditing. It is reasonable to assume that companies are actively working to minimize their legal exposure due to improperly licensed tools upon which their business might operate. Given the large amount of human effort involved in manually monitoring every computer in a large corporate environment, compliance and auditing concerns only make good sense.

[3] H.S. Dent Foundation, 'The S-Curve: Predicting Product Life Cycles and Reacting Ahead of Your Competition', www.hsdent.com/scurve.ph

This data supports the decision by some PCB design teams to use open source and pay-for-use design tools. After all, any software model that eliminates the need for licensing is compliant.

Decline in North American PCB Sales

IPC reports that, between 2000 and 2005, the number of North American PCB fabrication facilities has decreased from around 800 to around 350. This consolidation is a result of the following market trends:

- Movement of PCB fabrication work to Asia: Asia now produces 80.3% of the rigid PCBs worldwide.[4] In 1984, the US represented 42% of worldwide production; in 2006, the US represents 9%.

- Fabricator resistance to new business models: PCB fabrication firms tend to use their technical expertise as their primary marketing tool, relying on word-of-mouth and local/regional awareness to generate business. As overseas manufacturing successfully undercuts North American prices, fabs find themselves unable to retain profitable clients. Overall, these manufacturing firms cannot count marketing and sales as core competencies in their business model. This lack of a strong sales function results in the inability to replace business as quickly as it is lost to the overseas competition.

Increase in PCB Design Starts

IPC research states in a number of reports that rigid PCB design starts are on the increase overall. When we couple the overall decline in sales, with an increase in design starts, we can identify two key trends in the North American PCB design market:

1. PCB prototyping continues to be an active market segment.
2. Overseas manufacturing is not well-suited to supporting prototype work

DFM Tool Implementation Types

During the research for this paper, we identified three major implementation styles for DFM tools, as well as some variants. In this section, we will detail these implementation categories and the variants we identified

Standalone Client

Implemented as an executable installed directly to the user’s PC or network, a standalone tool’s architecture parallels the traditional batch-run tools. Whether checking IC or PCB designs, the standalone tool is fed two main input files: the design artwork and the rules specifications against which to check. Errors are then reported in a text-based report, with an optional graphical overlay of some fashion.

In the process conducting this research, we encountered some standalone tools under development that use an interactive interface to define the manufacturing rules as the user submits the design for fabrication. Whether the process rules are defined interactively or defined in a rules file, the basic premise is the same: the DFM program needs an artwork file and a set of rules to check against.

[4] IPC, World PCB Production and Laminate Market Report for the year 2006, November 2007

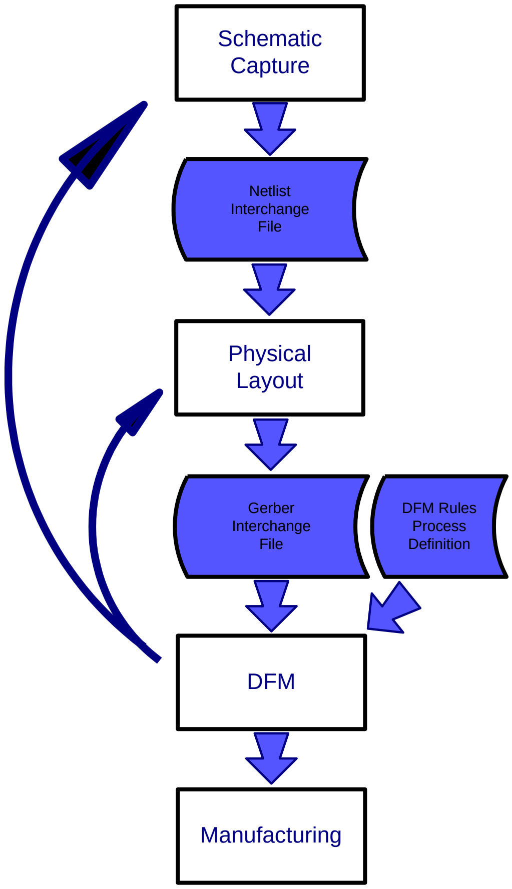

Figure 4: Design process flow utilizing back-end standalone DFM tools

As in IC design verification, it is common for standalone DFM to offer design post-processing functionality. These post-processors are designed to fix design issues without further involvement from the engineer. Common examples in IC might be geometry enhancement (LithoCAD); examples in PCB include functions like:

- Sliver Fill
- Copper Balancing
- Line Width Optimization (post-processed neck-downs)
- Neck-Down Repair

Strengths & Weaknesses

Generally speaking, standalone tools are where the highest performance algorithms for analysis can be found. This is true for IC as well as for PCB, and extends into other areas of design expertise as well (router technology, etc.) For software development firms with core strength in algorithms, the most successful competitive strategy is to craft a standalone tool and then provide interfaces and bolt-ons to minimize the effect of the detour their standalone tool creates in the design flow. Given the highly compute-intensive nature of the algorithms used in standalone tools, it is not uncommon for checks on large PCB designs to as many as two CPU hours for completion; in IC, verification jobs sometimes can run for days to reach completion.

As figure 4 illustrates, standalone tools tend to fit in at the end of the design flow. Typically, this form of DFM tool is employed at the end of the design phase, when the PCB is (almost) complete. Though DFM can certainly be run against in-process designs, verifying the interactions with outside modules once the design is flattened into a finished whole are often left to the end of the process.

By definition, then, the standalone DFM tool will identify what can be expected to be a large number of design 1) violations, 2) warnings, and 3) suggestions. The design team has three common strategies they can employ to manage the information returned:

1. Fewer full-functionality full-design checking runs; longer results-review phases
2. Run full-functionality checks on portions of the design; use one full-design run at the end for a final check
3. More iterations using focused-functionality runs; more (but shorter) design reviews.

Option 1 is the most thorough, as violations will not be masked by the design hierarchy or by a design in-process. Unfortunately, Option 1 also creates the most errors overall and increases the potential that a meaningful error might get lost inside a long list of spurious violations. In addition, Option 1 means that errors will be identified late in the design cycle; the design may already be quite optimized for space, meaning the potential rip-up and re-route of significant portion of the design. These sorts of fixes can be resource expensive and high-risk of introducing a schematic vs. layout mismatch.

Option 2 is a common technique for checking hierarchical or modular designs, especially common immediately before a module revision is checked back in to the development library. Though the rules checked are comprehensive, any violations between the interior of this subcircuit and the interior of a neighboring subcircuit will probably be missed. The result is that engineers only partially trust this module-by-module approach completely, and run at least one series of Option 1 checks at the end of the design cycle to ensure a clean design.

Option 3 was quite commonplace in IC as a pseudo-interactive design check methodology. The availability of interactive IC verification toolsets tends to minimize the ongoing effectiveness of this option.

Web-Based Tools

Web-based tools can have a variety of implementation styles, mainly batch-oriented, but with varying levels of interaction with the user.

Three implementation variations were identified in the course of this research. Some are currently available.

Web-Based Batch: these tools are accessed only through an email connection. In this implementation, users email their Gerbers to the tool operator. The tool operator then runs the check with a rule deck that the operator maintains. The error report is then emailed back to the user.[5]

Web-Based Interactive: in this implementation, the user installs a client application on their local computer to perform DFM checks. The tool developer maintains all appropriate rule definitions files on a web server. Each design check begins by a query for the latest version of the user-specified rule deck. If there is a newer process file, it is downloaded and used to perform the verification run.
Our research did not identify any web-based interactive tools at this time.

ASP Model: In this implementation, executable and the process description reside on a server, accessed via a browser (or equivalent). Users then upload their design files, select the process definitions to use for verification, and execute the verification on the server's CPU. This model is similar to that of Google Maps, salesforce.com, and others. Advantages are that any new functionality deployed is distributed immediately to all users of the tool, unlike client systems which require users to be notified (in some fashion) of an update to their local data.

Interactive Tools

Interactive Tools are, for the purposes of this paper, defined as tools that run native to the design environment, or can be invoked as a plug-in from the design environment.

A detailed discussion of hardware and software quality assurance is outside the scope of this paper. It will be assumed, however, that the reader is familiar with the concept that the cost of fixing a design flaw increases in geometric proportion to how late in the design cycle the flaw is identified. In other words, find the errors early and they're cheaper to fix.

[5] Old-hand computer scientists will shudder at how much this process reminds them of handing their FORTRAN card decks over to a mainframe operator at the computer center.

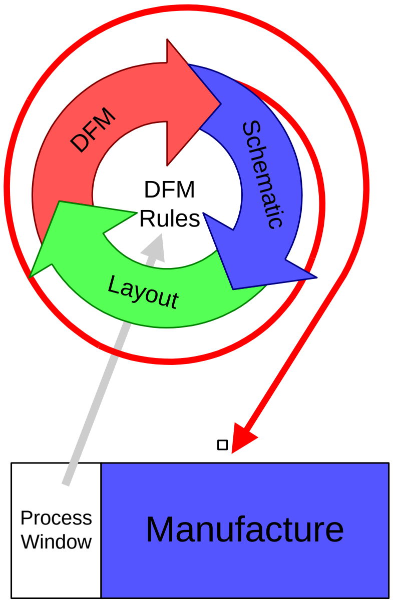

Figure 5: Design process flow with DFM analysis integrated in the design process

It is this very thinking that drives the development of interactive DFM tools: give engineers insight into how their design choices will impact the yield of their design, and engineers will use that information to make more robust design decisions (or, at least, take well-informed calculated risks with the process window limits).

In general terms, an interactive solution is built up from a checker engine or executable, directly accessible from the editing tool. This checker engine may run in real time, or function as a command that the user explicitly invokes as needed. Checker Engines are typically a general-purpose tool, using a parameter file to define the specifics for a process.

These parameter files - also sometimes called rules files or rule decks – can vary in the comprehensiveness or power of the language. Some tools provide rule specifications as simple pre-defined numerical parameters (table-driven), others (especially in IC design) provide a full-scale programming language type environment for highly sophisticated rules specification. There is a correlation overall between the power of the rules language and the overall sophistication of the rules checked.

Strengths & Weaknesses

Because the interactive DFM tools reside on the client work station, they certainly insulate the user from internet access outages. There are limits, however, to the amount of data accessible to local storage constraints.

Client-resident executables may require updates for bug fixes or enhancements. Because of the critical nature of rule checking software, it is in the best interest of the user and the fabricator to ensure the most comprehensive, defect-free software is used to perform the verification. For client-resident executables, this may require some sort of version-polling or notification mechanism.

Client-resident tools can create support and version mismatch issues for service providers relying on the accuracy of the information in the database. If the database contents are relatively static, then client-resident data is a manageable solution. If, however, data might change often (parts lead times from a distributor, for example), then a server-hosted database might be a better solution overall, even though it tends to exclude non-internet users from access.

Case Studies

PCB Editor Rule Decks

Most PCB design tools provide some form of design rule checking functionality. This functionality can be a simple as constraint parameters for the router. Or, they can be more comprehensive, encompassing electrical rule checks, timing or impedance issues, and other non-outer design issues. How design rule errors are reported varies from tool to tool, and will not be examined in this paper.

One DFM strategy is to implement rule decks as loadable files that configure the editor's design rule checking functions. The idea behind this technique is that, if a design is built to a detailed process definition for a target fabrication facility, then the design will have fewer inadvertent process violations, and will have a higher yield overall. Fabs could even go so far as to guarantee the manufacturability of a design if built to the 'golden' rule deck from the outset.

Though this strategy has been possible for quite some time, it has been difficult to implement because of the difficulties in crossing knowledge across the conceptual gap between design and manufacture.

As mentioned earlier in this paper, IC's GDSII and PCB's Gerber data formats, most commonly used for 'tapeout', essentially capture only artwork to be used by the available mask generation equipment. GDSII and Gerber, as file formats, do not lend themselves to communicating detailed electrical information or process/manufacturing assumptions.

Further complicating the development of interactive DFM has been the manufacturing sector's apparent unwillingness to develop and support rule decks for their manufacturing processes. Manufacturers have largely stopped at publishing process capabilities as readable documents, but leave the implementation of a rule deck up to the designer. One argument historically used for not supporting PCB design tools with rule decks has been the complexity of writing rule decks for so many potential CAD tools.

While designers certainly design with the intention of adhering to the manufacturing fab's process window, the designer is perhaps the least-knowledgeable person for developing rules to implement the fab's processes. Whereas the PCB fabs balk at developing rules for as many as a dozen PCB design environments, designers understandably resist implementing rule decks for multiple potential fabs. Given that there are 350+ fabs still operating in North America (down from 800+ in 2001), designers often give up on establishing trustworthy rule characterizations for their editing environments, resulting in a tendency to perform more design spins, more pre-fabrication analysis and tooling, and longer lead times.

In late 2007, Sunstone Circuits began pilot testing rule decks for third-party PCB editing environments. These rule decks were developed by Sunstone's tooling experts and written so as to capture as much detail about Sunstone's manufacturing process as possible. Beta versions of these rule decks generated an enormous amount of customer interest and activity. Sunstone estimates an increase of ___ new customers per day just from pre-release testing activity.

Software Cost[6]:	$7,000
SW Maintenance (per year):	$700
DFM Functionality Cost:	$0
DFM Process Integration:	$0

Using no-charge, certified rule decks from a reputable fabricator, design teams using this approach need to realize a reduction of about six proto turns per year to improve the ROI on their toolsets. For small teams with limited resources, these increased efficiencies may enable them to purchase more software or more advanced toolset add-ons.

Interactive DRC inside PCB123

PCB123, hereafter referred to as "the tool", is a PCB design tool developed, maintained and distributed by Sunstone Circuits. A central feature in this tool is the interactive DRC/DFM checking environment inside the tool.

PCB123's interactive DRC will:

- Flag trace/space violations real-time
- Mark some electrical connections on-demand
- Require users to review and 'sign off' on violations prior to the placement of an order

These functions help ensure that designers have the feedback needed to make informed design choices at the decision making point. The tool will, for example, flag a pair of shorted traces immediately and interactively. Designers get instant feedback that their last route is in error. Additionally, as the user backs the short out, the tool removes the error indicator. The tool also performs full-design rules checks at logical points throughout the design process (immediately before submitting the design to Sunstone as an order, for example)

[6] A composite number representing entry-level first-tier tools and fully-configured second-tier systems.

The tool development team continues work refining and expanding the DRC functions within the tool.

Software Cost:	$0
SW Maintenance:	$0
DFM Functionality Cost:	$0
DFM Process Integration:	$0

In this example, the customer's up-front cost is truly nothing. By accessing detailed process violation reporting during the design creation phase, design teams begin realizing a ROI upside with the first prototype design turn they eliminate. In fact, many design teams see a 3X improvement in the probability that their first prototype turn will be functional.

Standalone DFM tools

Standalone DFM solution share an architectural similarity with standalone IC verification and standalone DFM tools, including an overall executive program (ERF Manager), a comprehensive rules specification and job-control language (DFM Programming Environment) and a portfolio of add-on modules to extend overall DFM capabilities.

One vendor's solution targets the enterprise-level tooling environment; their tool reflects that concentration:

- CAM translation
- Multiple Interchange Databases
- Supply Chain Tools – BOM and Parts
- Tools to maintain or reclaim design intent from legacy files
- Bare Board Verification Analysis (manufacturability)
- Multiple Add-On Options, including:
    - HDI & HDI Microvia handling
    - Sliver Filling automation
    - Etch Compensation automation
    - Copper Balancing automation
    - Pin Holes repair
    - Line Widths (neck-down generator)
    - Neck-Down Repair
    - Line Unification

The first-tier standalone PCB DFM tools are comprehensive, complete and well regarded in the marketplace. As with many large-scale, complex standalone tool environments, DFM can be highly effective, yet the maintenance and optimization of all the interrelating functional blocks often requires dedicated staff, resources, design volume and infrastructure in order to enable it to run smoothly.

While certainly extremely powerful, the tool requires significant setup to ensure each module interacts with its process neighbors appropriately. This work is often much more involving than users might initially expect. Setting up the parameters for an enterprise-sized solution will require integrating:

- Multiple bare board suppliers
- Multiple processes and their variations
- Multiple CAD tool environments
- Module-by-module understanding of the inputs and outputs.

Purchase Price:	$60,000
Process Integration:	$150,000
3 people, 6 months	
Total Purchase	**$210,000**

Ongoing Maintenance:	$75,000/year
SW Support (@10%/yr)	$6,000/year
Annual cost	$81,000/year

If we assume a five year amortization, the total costs for a standalone system might approach $534,000 including licensing, maintenance and staff to maintain the system.

If we concentrate solely on the first-order function of bare board fabrication as the method for achieving ROI for the standalone tool, and assume an average order size of $350 per prototype order, then a design team will need to reduce the total number of prototype orders by 305 per year, or 25 fewer orders per month.

A more detailed analysis might include components/assembly in the analysis. Assuming an average of $1,500 per turn in assembly costs, the design team must reduce overall design turns by 58 per year, or nearly five designs per month

And if we further include hours of labor saved by the design team for each design turn eliminated. If a design turn requires an average of three days effort by a $100,000/year engineer, then an eliminated design turn represents $1,500 in first-order (the order) and second-order (the labor) costs. In this example, a firm must realize a reduction of 31 turns a year, or roughly three turns a month.

Unless a design team also maintains their own fabrication line – quite unusual these days – a standalone tool makes most sense as a initial verification and panelization tool for contract fabrication facilities

Final Conclusion/Summary

The importance of DFM and DFY in the design of PCBs continues to increase over time, driven by increasing design complexities, and the need to support increased use of overseas production facilities. Designers are increasingly seeking DFM solutions that are easy, intuitive and integrated into the actual design cycle.

The need for standalone back-end DFM tools will continue, as these tools are especially well suited for use by contract fabricators to increase and improve overall production yields. Standalone DFM tools, however, are not a cost-effective solution for design teams themselves. The result is that relying on back-end DFM solutions will increasingly become a bottleneck in the design process. Design teams must either delay DFM until submitting the design to the board fab, or find alternative methods of performing DFM.

Given that most design teams use a PCB design tool that supports design rule checks, we see in increase in the number of designers seeking known-correct rule decks for their PCB design tools. Designers find themselves caught between the need for detailed and accurate rules definitions, and the lack of definitions available direct from fabricators. We expect to see in increase in designer demand for rule decks that fully implement a fabricator’s process characteristics.

Internet-based DFM applications, based on what we found in the marketplace to date, are usually integrated with an order quoting mechanism. Users of these tools will receive process verification along with the quote details for their design. Some of these applications will likely emerge as brokerage firm front-ends, or as a mechanism to ‘comparison shop’ multiple fabrication services from a single online location. Like standalone DFM tools, internet-based DFM solutions are optimized for checking a completed design. Manufacturing problems reported during this phase often can be expensive or difficult to resolve.

As back-end DFM increasingly becomes a design bottleneck, designers will look for alternative solutions that are both cost effective and timely with information. Interactive DFM tools will move into prominence within the industry. When a designer can get as-you-go reporting of a process violation, they can fix that violation immediately and inexpensively. IC designers have reported significant increases in design robustness and overall yields through the added information interactive DFM brings to their design process.