I·Connect007
GOOD FOR THE INDUSTRY

Roundtable Discussion

App Notes and Fab Notes

I·007e RESOURCES

Roundtable Discussion: App Notes and Fab Notes

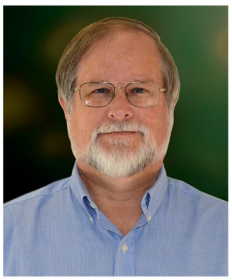
A Lively and insightful discussion from experts in the industry.

Moderated by Andy Shaughnessy

Editor, PCBDesign007.com

and *Design007 Magazine*

Recorded September 2020



Dana Korf
Korf Consultancy

Jen Kolar
Monsoon Solutions

Kelly Dack
PCEA

Mark Thompson
Monsoon Solutions

Meet the Participants

Dana Korf is the founder of Korf Consultancy and former director of PCB technology for Huawei. In his columns, he focuses on helping designers create better data packages.

Jen Kolar is the VP of engineering at Monsoon Solutions—a Bellevue, Washington, design bureau known for having their data right every time, which, as we found out, is not always the case in our segment.

Kelly Dack an I-Connect007 columnist, a long-time designer and CID instructor in addition to his day job with a Washington State assembly provider; like all of you, much of his job involves resolving issues with bad or incomplete data.

Mark Thompson recently joined Monsoon Solutions as a senior PCB technologist. As an I-Connect007 columnist and CAM expert, he writes one of our more popular columns on DFM issues and design data.

Introduction

I recently invited four of our recent contributors—Dana Korf, Jen Kolar, Mark Thompson, and Kelly Dack—to review the June and August 2020 issues of *Design007 Magazine*, which covered app notes and fab notes, respectively. In this wide-ranging roundtable, we discuss some of the ongoing challenges related to incomplete and inaccurate design data and why communication can preclude many of these problems. What follows is the transcript from this conversation.

REALTIME with... I·Connect007

Dana Korf
Korf Consultancy

Jen Kolar
Monsoon Solutions

Kelly Dack
PCEA

Mark Thompson
Monsoon Solutions

App Notes & Fab Notes Roundtable

A lively and insightful discussion from experts in the idustry.

Andy Shaughnessy: I'd like to welcome you all to the first *Real Time with… I-Connect007 Roundtable*. Thank you for taking time out of your busy day to join us. Usually, the Real Time with… program features video interviews at trade shows, but at today's roundtable discussion, we're turning the camera around, and we've asked the four of you to review a pair of recent 2020 editions of *Design007 Magazine*—the app notes issue in June, and the August fab notes issue. You all spend a lot of time working with this type of data, so we'd really like to get your feedback.

I think it's fair to say that data seems to be an ongoing issue for our readers. Dana, why don't you go first. What did you think of the topics? Were you surprised by anything you learned?

Dana Korf: Yes, that was an interesting combination: app notes and fab notes. Those are two totally different things, but in reality, they actually are kind of linked when you think about it. The app notes issue in June was interesting. All of the bases were covered, but you have to realize that the purpose of app notes and datasheets is to sell products. You create a device, you put a bunch of specs out there along with some test methods or how to program it, etc., and then someone sits down and decides, "Here's one way to use it." A lot of the comments said, "They're all wrong." Well, the app notes are probably not all technically wrong; they're just not applicable to every conceivable way of using the device.

And one thing I liked was with Max Maxfield's article. His whole concept was "Trust but verify." It doesn't matter what you get: "Let me look at it. Let's

make sure they're right. Let's make sure I'm using it right." Way back when I first was out designing minicomputers many years ago, and Motorola came out with the MECL 10K application note—I still have it—it was how to use ECL when everyone was using TTL at the time, and it became the Bible for everyone, saying, "Here's what's different." A lot of people learned transmission line theory off that app note document because that's all that was out there. That was actually fairly thorough at the time.

The real linkage between fab notes and the app notes is when, as a front-end engineer, you open up a fab print in a data package. You're reading all of the notes, and you try to find the specifications that are referenced. You're trying to figure out, "How is this person using this board? Are they using the right materials there? Is the pad-to-line spacing correct?" You're trying to guess at the application because they're not all the same. You don't want to embed in too much cost, but you don't want to remove so much cost that it may not work. It may not be reliable now, so it's very interesting that there's kind of a linkage there.

One comment that Jennifer made in her article, which I thought was pretty good—which I see works really well with a lot of especially large organizations—is having a series of notes, and you check off the ones that work. As I always tell people, I've never seen a board layout person intentionally layout a board wrong. I don't believe that person exists, but they may not know what they're doing and, in particular, the board shops don't tell designers what their capability really is. What are the real rules? They're trying to guess. Too many people consider it to be IP. "I don't want to tell my customer what we can do because they might tell somebody else. "Well, there are only so many ways you can build a board. Sorry, people.

The biggest problem with notes and drawings and data, especially using unintelligent data like Gerber, is that they conflict all of the time. There are multiple mentions about cut-and-paste but cutting and pasting wrong. I've seen that probably a million times, and they don't realize that's not applicable, or they put in too many notes that don't make sense. Or, more importantly, they don't put enough in, and especially around impedance and reliability.

I guess that's what the big guessing game is going to be. The CAD tools don't review all of the rules that boards can have. The board shops don't tell the CAD tools what rules they have, so it's a big guessing game on both ends. Now, when you overlay non-intelligent data where you send out conflicting documentation, then everyone has to try to guess what they really meant and which one's right. And as anyone who's been in a board shop knows, when you guess what you think they think, you pay for the scrap because

you're going to guess wrong, but you only do that one time. Hopefully, you only do it one time. I thought that was an interesting marriage between the app notes and the fab notes. It's kind of similar as, in the case of a component, you're trying to tell the customer how to use it, and they're trying to guess how to use it. In our case, you're trying to tell a fabricator how to build a board, and they're trying to guess how are you going to use it? It's very interesting. They're so similar in theme, in my opinion.

Shaughnessy: Jen, what do you think?

Jennifer Kolar: Sure. Actually, one of the things that you just said there, Dana, was curious to me: about the fab notes either having too much or too little data. I've tended to start taking the approach with people that I would rather have too little because then the fab vendor will ask questions versus having too much that's incorrect will cause confusion or cause something to be built incorrectly. We do boards where sometimes we didn't do the initial version of a board, and then we're now doing the revision to it, so we've inherited notes. Or we're starting the layout, and most designers will find some board with a similar stackup, grab that, start with that, strip everything out and keep going.

Well, that sometimes means notes get copied, or a lot of crap gets left in. Or, when we're doing a lot of the builds we do for our customers, it will ultimately go to production in large manufacturing for consumer electronics. That means they had different notes about different testing and all these things that really don't apply to, say, a proto build, so I'd be curious what your view is. Is it better to have too much, knowing that it could be incorrect or too little? Then, you get to that guessing game where obviously the right balance is having exactly what is needed—no more, no less. You rarely get there.

Korf: Yes, that's a good question. I wrote an article on this a while ago for the magazine. I'm a simple person. As a fabricator, I just want to build what you send me. I don't want to have to ask anything. Send it to me, and I build it and ship it back to you. That's where the world is, but your comment is correct. Over the 40-plus years, we've kind of tailored around an environment where we're expected to ask questions, so you're expecting questions to be asked, be it from cost reductions, whatever.

Per your point, though, I would prefer to have few notes that are right and leave off the ones that are wrong, for sure. I had an example a couple of years ago where the customer just said, "I want an FR-4," so we build a mid-Tg FR-4. They got the boards after the third prototype run and came back and said, "Oh, we wanted a high Tg of FR-4." "Well, why didn't you tell us that? You just said FR-4." We ended up paying for the scrap because we didn't do what

they were thinking, and we're supposed to know what they think, so to me, always quality is always number one, so if you're not sure, leave it off. You can always figure it out later.

They're going from prototype to production. Obviously, that's always a big problem because proto shops are paid for speed and not necessarily focused on high yield, where buying shops are focused on yield and definitely not necessarily speed. We always struggle trying to get stuff out fast through a million-square-foot factory, so one customer says, "I prefer fewer notes that are right versus more notes that may be wrong or not applicable."

Kolar: Perfect, and that's what we definitely emphasize. The other thing you said that was interesting to me was about the app notes. In our world, because we work for so many different customers as a service bureau, often, a lot of these are high-end, cutting-edge products. A lot of times, we have no idea how they're going to be used. Our customers aren't going anywhere. We mostly work with them for quite a while. We figure it out where we tend to have a sense, but we don't ask, and in a lot of cases, they won't say. If we say, "Hey, what is this going to be used in?" They can't tell us. That's information that isn't allowed to be public even if you want and need it. That can be a tricky thing, and I've definitely seen this in the fab world where, "This area was intentionally voided of copper," and a fab shop then goes and puts thieving in.

But the fab said, "Wait. This is unbalanced." They didn't know that that was intentional. I think that's a tricky one from our perspective or from my perspective. A lot of times, we don't have any way to give great data on the application. I am curious: What would be the most important thing to know where a fab vendor would come back and say, "You need to rethink this"? We have customers asking us all of the time, "What materials should we use? Will this work?" Where do you draw the line of what is your responsibility to the fab vendor to essentially be a design engineer? They designed your product versus build what they specify and have them do the upfront legwork.

Korf: The safest way is you see a lot of especially large companies or design organizations who say, "You can do no edits without my approval," except for edge compensation and things like that. Me as a fabricator, if I'm not sure, I train everyone to ask. If you're not sure, ask. The more experienced people will see a clearance area, and it's on every layer. Well, maybe there's not supposed to be anything there, but the board's not going to be fabricatable. It's going to delaminate.

The short answer, and everybody here knows I could talk for hours on minute little subjects, so I'll try to keep this short for Andy. If you're going to do

an edit, ask. If you're a designer, require them to show you what you're doing, so you could make sure, maybe you have to go back to your customer, and in Kelly's case, a lot of times they come through EMS shops, so you might affect an assembly where a designer has no clue about assembly, what their rules are, let alone going from fab to assembler to layout person to designer. No one really knows.

Kelly Dack: Thank you for noting that, Dana. Oh, my goodness, Maybe Mark should go next, and then I'll chime in on that. That's a great point.

Mark Thompson: I'm really here only to talk about accurate fab notes because I don't really know much about app notes. I think this is literally my first design foray, so I'm learning about design, and I had sort of a PCB 101 yesterday with Darin Dix, one of the owners here. I wanted to say Dan Warren, our note template maker, has done a great job adding all of the potential fab notes. Of course, the designer is responsible for calling out the notes that are not applicable to that particular board. Generally, it's a two-tier process. Not only is the designer responsible for that, but also there's a review process, and the review person actually goes through the same notes to make sure that they're correct.

From where I sit, here are the notes that are important, the first of which would be a board outline with dimensions and data. If you're not providing overall dimensions, make sure the Gerber board outline is present with a dimensioned hole or a feature to be able to place the board outline at the fabricator. The material type was 4101\ unless the customer has specified such as P370HR or Rogers 4350, but don't call out the material name. Call it out by the spec. Call it out as 4101\ that number. A drill drawing should include a drill table with sizes, plating status, quantity, and tolerances. A unique symbol for each size should be used.

I remember, years ago, some designers used circles for all of the tools. It makes it really difficult for a quality control person to verify the hole sizes in quantities. As an example, use, for instance, an 038", 040", and 042". They're all so close. If you just denote it as a circle, it makes it extremely difficult for a QC person to be able to say, "The quantities are wrong, they drilled it the wrong size, etc., etc." Use a unique symbol. Denote if the hole is part of a blind or a buried scenario and do that in the stackup. This last one's not a necessity, but it's helpful. If it's a via, note it as a via; if it's a component hole; note it as a component hole.

Lastly, with regard to drill drawings, don't mix imperial with metric units in your drawings. That's something that happens all of the time. If you're going to have metric callouts, stick with metric callouts. Don't mix inch and metric.

Stick to one. The next thing would be a bow and twist callout. Here you can simply state, “Bow and twist needs to be 2% over the entire length of the board.” Or, you can use ANSI symbology, like a flatness or a position symbol. That works great.

Now, with stackup information, what you’re looking for is overall thickness, dielectrics, and copper weight on a stackup. If it’s a four-layer board and there is no stackup information, a fabricator is going to assume it’s going to be top, ground, power, and bottom layer. If you have a different organization and you have a different layer stack than that, please note it. Lines buried should be depicted on the stackup.

It’s helpful to use the same names. For instance, if you’re calling it +15 Vo, but in your stackup, you’re calling it power, the fabricator doesn’t necessarily know the difference between those, so make sure that the layer names and the description match.

Another example would be blind vias on layers one to two, but the stackup portion may show blinds on one to three. Sometimes it’s just a simple error. As Jen mentioned, maybe the parts started as blinds one to two and then got changed to one to three for various impedance reasons. They wanted to keep the impedance line size at a reasonable size, so they changed it to a one-to-three from a one-to-two type of scenario. It makes sense.

Next is controlled impedance information. The four biggies that everybody goes by are: What tray sizes are being involved? Where does it reside? What’s the threshold? What’s the tolerance? Now, those four things are very important, but one thing that you would definitely want to do is avoid using the same line size for all of the same. If you had single-ended, 90 ohms, 100 ohms, and you used a 0.005” line for all three of those scenarios, it’s kind of a problem because the CAM operator has to uniquely pick those out to be able to resize those for edge compensations. God forbid they have to resize them to actually make some sort of a heinous resizing to meet the impedance, but just for the edge compensations alone, having all three of them at the same size is kind of problematic.

Sometimes at Prototron, we would have the quadruple whammy, where people would use that same line size for copper pour as well. You’d have single-ended structures, 90 ohms, 100 ohms, and all of the copper pour was done at a 0.005” size. It made it darn near impossible for the CAM operator to be able to pick out those sizes.

Next would be board class and specifications, Class II, etc., and then logo and date code placement. If you have a location where you can’t actually

have a logo and date code due to EMI issues or if it's going to be buried by a component when it goes to assembly, call it out. It can be as simple as any free area on the solder side.

Solder mask information should include the type, color, size, and sometimes thickness, depending on your application. Then there's silk information, type, color, and size. None of this is critical information. Out of those two things, the color of the solder mask and the color of the silkscreen are going to be the most critical things. If the fabricator can't figure out that they have two silkscreens top and bottom, they've got a problem anyway. With plating thickness requirements, if it's going to be more than your standard one-ounce plate-up, a fabricator should know. They're going to need to know if it's going to be a two-ounce plate-up, a three-ounce plate-up, whatever the case may be.

With testing requirements, a simple note about standard electrical tests or HASS testing or netlist testing is very beneficial. Now, if you're going to provide ODB++ data, the CAM operator and the fab shop should automatically run it on a netlist. If they got a netlist present, they're going to need to run that netlist to make sure that there's no mismatches.

Lastly, if you're not going to allow the fab shop to panelize, provide some panelization requirements: rail size, location, fiducial size, fiducial location, and tooling hole location. Even in situations where you may have part overhang at assembly, if you can, denote an area where there's a cut-out in the panel to be able to deal with the part overhang. Again, Dan Warren, our notes guy here, is really super, and he leaves all of the applicable notes in, and it's the designer's responsibility to call out all of those that don't actually apply to that job.

Kolar: What Mark's talking about there is that we have some standard templates that we have built, and I was actually thinking about Dana's article on total quality management as pertaining to exactly this. We have a lot of detailed manuals and processes and documents, but then we've had trouble getting people to implement and follow them because there's a lot. What we've done is create shorter stripped-down versions of them. "Here's your checklist for everything you need to make sure to do before you release the board. Here are your standard assembly notes. Here's how we release the package."

We've taken that and made simplified versions that people can quickly follow, and then if they have a deeper question or they're newer to the field, like Mark is to the layout side of things, then there's much more detail that we can go into and really understand the theory of why. We're really big into

"why." One of the things that I was thinking about while reading Dana's article was looking at making sure that what you're doing is actually working. It had me rethinking that and looking through and thinking, "Okay, where could we be doing better? How can we get more measurements of actual real data to figure out what's working and what's not with our process?"

Dack: That is right. I really thought that the two articles that we're reviewing here worked really well together because right now, we've had really good explanations of how to decipher fab notes for a bare PCB or an assembly, but diving down into it, when it gets to app notes and datasheets and the confusion between can we trust them or not, this is all the same theme. I'm realizing that there's a good reason for putting these two together.

From the standpoint of the issue on app notes and the datasheets, I really appreciated some of the statements made regarding the two because they do fit in with bare board design. Rick Hartley made the point by quoting Lee Ritchey in saying, "Don't trust app notes unless they've been verified."

Thompson: That's right.

Dack: And Lee always says, "I'm not saying don't trust them; I'm saying don't trust them without verifying. Trust and verify." And in the fabrication notes that Mark's gone through and described in detail, there are so many things that go wrong. We like to say in design, fabrication, assembly, "There are a thousand ways to die." Dana makes the point, if you trust without verifying or asking the questions, get ready because you may be in for a surprise.

I like this idea of perfection at the foundation of design. It's not always achievable because of all of the manufacturing variables and the material variables. This is why I think these articles bring up such a good point. A lot of it is subjective. A lot of it's dependent on where the product is going to be used, and as designers, we have to aim design toward end-use, so materials, production, processes up front are all what we lay down so easily in our design database is not necessarily easy for our manufacturers. We designers had better understand every jot and tittle of those notes because if we don't, we are going to literally stop production downstream.

Korf: Mark touched on a couple of good points. One was the fab print that doesn't match the data provided because of revision controller changes. That's why I'm a big fan of moving away from Gerber data and moving to ODB++ with the inheritance issues, and then on to IPC-2581. If you're looking at 2581, you see a lot of duplicate data. You don't need a lot of these notes. I don't need an impedance table because every trace is attributed with a model, the impedance, the tolerance, the reference planes. I don't need a

drill table because, oftentimes, the hole counts never match anyway.

Thompson: The intelligence is embedded in the design.

Korf: Yes, it's in the design, and every CAM tool can count the holes. That's pretty simple stuff. You don't need a netlist because the netlist was invented for 274D data back when the capture table was in a different file.

Thompson: Exactly.

Korf: You actually don't need it anymore with intelligent data, ODB or 2581, but we still use it, and then netlists come in wrong a lot of times. One of the big issues is, "Well, they're intentionally shorted." But the netlist says they aren't, and their comment was like using generic materials—like slide sheets—from IPC. We all know everyone says, "Well, the Dk values are all wrong, and none of them are right." Actually, if you look at the datasheet, they're all correct. We use the test method that they use. The problem is, they don't give us the data versus the test method, which is a TDR that the board shops use.

Thompson: There you go.

Korf: Every board shop has to figure out, "What's the right one? How does it match what I'm doing?" Then, you have to come up with various ways to guess the right answer or have it match their process anyway, so I think you get revision control issues, and not just copy from the other design, but the copy from Rev A to Rev B can pass the wrong issues forward because they're in a hurry, or they never fix the issues from Rev A into Rev B.

Kolar: A number of times I'll see, especially on jobs that we don't design but that we just manufacture, that the vendor comes back and I look through the list of EQs from the fab vendor that says, "We're going to do the same as last time. We'll ignore this same as last time. Change this thing the same as last time." You just want to smack the designer and say, "Why are you making the fab shop do this? Why aren't you learning from it?"

Dack: I'm so glad you mentioned that, Jen. Transitioning over to the EMS world from the fabrication world, from the design world—I've been a designer throughout, but it's just been fascinating to move into working for a fab shop. For the summer of 2015, I worked side by side with Mark and learned so much. Now, I'm at an EMS provider over here in eastern Washington, and I'm learning something every day.

Regarding Dana's point about the intelligent data, I was a believer—and I still am, in a sense—that Gerber data and drill files and the things we traditionally have sent as a manufacturing data package are complete enough, and

that is a popular opinion out there. I've heard that 90% of the boards built are still using Gerber data and Excellon drill files. As long as it includes an IPC-D-356 netlist, 90% of the boards are still built fine.

But what I learned while talking with Dana during the August interview was the confusion factor. Talking with Dana, my challenge to him was, "Wait a second. With intelligent data, if I need purple solder masks, how am I going to do that?" There are still some things that intelligent data cannot define. Evidently, Dana says we can do things like that, so that was really interesting to me, and I've been making an effort to look into providing ODB and 2581 data.

Another interesting thing is that in working for an EMS provider, I see data and designs from different customers, and I see that everyone's doing it differently. With the circular drill symbols, for example, I thought I was the only one having to deal with this stuff. It's got to be ironed out, and maybe intelligent data and intelligent design methodologies are coming. I see a lot of that on the horizon, so it's very exciting.

Thompson: One last thing I'd like to say is back to what Rick Hartley mentioned: Don't trust anything in your career or in your design that you can't test empirically. That was such a great statement. I believe he says that at the start of his two-day tutorials, but I love that. That's such a great comment.

Shaughnessy: Didn't Ronald Reagan make that famous during detente? "Trust but verify."

Thompson: Trust and verify; there you go.

Korf: If you look at most CAM mistakes that they actually make on the production floor, you'll look at the scrap and say, "You have the highest-paid people, you have a million dollars' worth of tools. How did you let that go through?" That's why they have four revisions of software every year. You can't always predict what combination of weird attributes and data that you can't check, so you've got to add another check or another combination of checks to try and figure out all of this stuff.

Kolar: I think that's one of my concerns, and we primarily output ODB data, and that's primarily what we prefer to work with. Some of our customers will just provide Gerbers, but the further you get down to an intelligent package that's completely self-contained, you start losing some of those human checks. You know, "I'm reading this, but I'm seeing that," and you're relying more and more on the designer to be really detailed and accurate and put in all of those attributes correctly. That's just one of the things. For exam-

ple, there are things that front-end designers can do, say, in Altium. If you're using Altium in schematics, you can put in a lot of annotations and all of the nets that can then propagate straight into the layout, and your rules can come from the schematic in some cases, but most people don't bother to take the time to do that on the front-end schematic.

Just as an example, designers often don't update their stackups. Or they try to remember, "Oh yeah, I need to regenerate that drill chart. I need to re-generate this right after I'm done with that one last tweak and adding my ground stitching." It's an area of concern. "How easy is it to cut and paste it or leave in the wrong data? What will be the default values that people need to remember to change?" I think we need to figure this out; how do we do it in a way that still keeps the checks and balances?

Dack: Jen, I have a question. I know the Seattle area is just teeming with new engineers, and this is changing the face of the traditional designer-engineer relationship. Would you say we've moved to close to 100% of designs being captured on the front end by the same engineer who's going to be laying the board out?

Kolar and Korf *(simultaneously)***:** No.

Dack: Really?

Kolar: There's still very much this sense in the industry, although it is changing, that layout is just connecting the dots, that any electrical engineer should be able to do layout, and that is so not true. At our company, over half of our designers are electrical engineers who choose to be layout engineers, and they bring that knowledge with them and treat you and the layout as an engineering project itself.

What we see at the company is that initially, they'll have their electrical engineers also trying to do layout, and what happens is they start slipping schedule, they start not understanding what they're doing with the tool. They don't have the depth of knowledge or anything apart from manufacturing. They hand that over the wall, and they've designed something that's completely non-manufacturable. Or they get part-way down, and then they call us and say, "Help. We're running out of time. We need our engineers to be working on these other projects. They're stuck doing layout." I would say, for some of them, the answer would be yes, but I would say for a lot, it's not even 50%.

Dack: That's interesting because we were talking about designing on the front end and how helpful it is to add as much as we can to the schematic on the front end. Mounting holes, mechanical information, we're seeing more

and more of that is being added to the schematic. However, without DFM, and not taking into account the effects of adding physical mechanical constraints, it's going to cause a problem on the back end of the layout. Testability and test points are some things that come to mind. A lot of engineers will physically put in a test point with no reference to how large the board area is, or the stackup, or how much real estate there is to add those test points. It's a fail from the beginning.

We merge in the data only to find that the board has to be twice as big to shoot all the test points or that it's breaking out from a fine-pitch BGA. Oh, this is a classic where an engineer will define the trace widths for a specific impedance stackup, which will require a certain trace width, but that trace width won't be able to break out of the BGA. We have a habit as designers of undoing everything in a traditional flow. I'd like to see them ask more questions.

Korf: I spent many years in Asia, and we would see a design by a French person who learned English from someone in Scotland who writes a fab note in what they think is proper English. It comes to China to a Mandarin-speaking person who learned English in Hong Kong who learned it from someone in Singapore. You get the translations, and someone in France doesn't know what "green oil" is. Everyone in China knows what green oil is, but no one in France knows that green oil is actually solder mask. They would come to me and say, "Dana, you speak good English. What's this note mean?" I'd say, "I have no clue." Even if I had three beers, I wouldn't know what that note says at all. And to Mark's point about English and metric, boy, how confusing when you get them mixed together? You know you're going to have a problem at that point.

Thompson: I think a lot of times that really started at the fabrication level. The fabricators are the last holdout in this whole equation who wants to still go with imperial units. They're still lost in a world of inches, even though five or six years ago, the drill sizes automatically went to metric, so you can't even buy any imperial sizes anymore. It's crazy that they still hang on to that.

Shaughnessy: Do you all think we'll ever go fully to metric? It seems like metric would be ideal since the rest of the world is metric, right?

Korf: Only one country is the holdout, and we're sitting in it.

Kolar: Also, some small country in Africa, I think.

Shaughnessy: We had to learn it at school. I remember having to learn 88 kilometers per hour was 55 miles per hour. It was on all the speed limit signs.

Korf: If engineers in America were asked, “Would you rather be metric or English?” I think most of us would reply metric. It’s easier to write that 10, you know?

Dack: We need a PCB design app note that starts with, “Thou shalt design in metric.”

Kolar: Especially when you’re getting to really tiny, tiny devices, all of the components for that are designed in metric. If you do a translation, if you take your layout, you have a really, really tiny piece like boards, really fine pitch parts, you lay it in metric and if you do a translation to imperial, you’re going to start getting rounding errors and data shifts that actually matter.

Dack: You bring up another facet of app notes that might be missing, and that is vendor specification. We, as designers, deal in nominal numbers in our CAD layout. We talk about a 0.005” line or 0.127 metric, or dimension line. But we, as designers, have to realize that’s the fabricator’s job; that’s a target for the fabricator to hit.

All these compensations—something I learned working with Mark over the years at Prototron— required to hit that number are subject to a vast number of variables: copper weight, edge factors, compensation factors, and outer/inner layers. There’s a ton of stuff that goes on, and designers need to know that so they can pick the right width, not only for impedance and electrical performance but for manufacturability. Don’t do that on two-ounce or three-ounce copper.

Kolar: That’s another thing. The bridge with designers is to reach back out to the front-end engineer, to question them, to not be afraid to push back and say, “No, this doesn’t make sense. Let’s look at this. Have you thought about this?” We encourage all our designers to push back and to question and to challenge, and that a lot of our customers have come to really rely on that.

Korf: When I’d have my front-end engineers, the senior ones would go out and visit a designer, sit down, and see that they’re not all idiots. Look at all the constraints they have to worry about to trap those two dots together because they have no idea how it conflicts.

Kolar: You have to be willing to push back. You have to go through and explain the pure basic physics of space and how it’s going to work.

Korf: You talk about app notes’ influence on design. There’s one really famous large microprocessor company that writes app notes for each of their processors. Suddenly, a few years ago, we started seeing a specific material from Taiwan being used by everyone who used that chip. We couldn’t

understand why, but that supplier did all of their modeling using material from one supplier, who made it free for them to do all of their testing, and they wrote that material in all of their app notes. So, everyone who used that chip was using that material because they had no clue that another material, which happened to be way cheaper, could also be used. You can see an influence of app notes down to the layout, down to the notes, down to the fab sometimes.

Dack: Yep, marketing mission accomplished.

Korf: We worked for that company. The marketing person is a good friend of mine. I said, "Good job."

Shaughnessy: Wow, I'm still intrigued by Jen's earlier comment about incomplete data being better than wrong data. I never really thought about it like that, but you would rather get an incomplete data package than one that's wrong because at least it's not wrong, right?

Korf: Yes, as long as the board works for the customer because that says people or the designer know what those answers are when they lay the board out.

Kolar: It's fascinating to listen to some of what you're describing from the fab vendors' side because it's so different than how we work with our layouts; we don't just throw things over the wall. By the time we send a design to fabrication, we've been iterating in the stackup and we've fine-tuned the impedances as much as possible. We know what's available for materials. We know what the clearances are. We've tried to fine-tune it that so that when it gets handed to the shop, it's already using a stackup that was pre-approved by that vendor.

Korf: You're lucky. You know who your suppliers are when you start. Most designers have no clue because someone at one of your customers doesn't know who Kelly's EMS shop is going to buy from. They have no idea.

Kolar: That's where we have an advantage; we have project managers who work directly with our designers. If we're doing the layout, they get the iteration early and up front, and early DFM up front. Even if we didn't do the design and we're doing the manufacturing, we'll still iterate back and forth. Our customers will sometimes say, "Hey, we're doing design in-house, but we're going to have you build it. Can you get a stackup that works for this? Can you verify these clearances are possible? Tell us what via sizes we need to use."

Dack: That makes me think back to why incomplete information is better than complete information that's wrong. Regarding thermal specifications

on FR-4 materials, often, we'll get conditions where the designer specifies, "Thou shalt make this out of the cheapest material," because they're trying to hit a design-for-cost target, so make it out of the cheapest material. They may specify Tg 130 material on a 10-layer board with very fine vias and fine features that's double-sided assembly, and additionally has all different mix of technologies on it. In other words, it's going to go through lots and lots of thermal cycles. Tg 130 material is not the right material for that. They should have left the material less specific and let somebody else fill in the blank like the EMS provider who knows how many thermal cycles it's going to go through and then pushed back with a justification for spending more on the material. That's a far better way to go.

Korf: That's a good point. When lead-free first came out, all of a sudden, we all learned the hard way that, "Oh, it's very sensitive to the lead-free profile, good temperatures, and everything."

The industry morphed into giving this idea now to the fabricator. "Here's my cycle we asked for. What's your maximum reflow cycle? What's your maximum temperature? How many cycles are you going through, including rework?" That has a huge impact on materials, especially when you get to higher layer counts. A lot of these won't work, and with HDI, it's absolutely critical because materials will fall apart while you're building the board, let alone when you're trying to assemble it later on.

Shaughnessy: Do you all just want to offer any final words of advice for new designers or fabricators?

Dack: I'll chime in. This theme of trusting but verifying really rounds out to a lot of things we've said here. Designers without years and years of experience are often not qualified to apply the app notes. They need a lot of experience to know what to trust or if they can trust it. I think that's the main theme. Trust and verify is a huge message that was brought forth in these two issues. I think it all boils down to education.

Kolar: I think the other is to really understand the end-to-end process. When you're a LAN engineer, step out of that a little and own the whole process in terms of thinking, "Okay, when the data's coming in, does this have everything I need? How will this be used downstream? Will this be able to be manufactured, assembled, tested, used?" Really take the time to step back and think through that to make sure you're not making some obvious mistakes.

Do some upfront planning. For example, to your point about, "Hey, once we bring the schematic in, we find out we need something twice as big." On a complex project, do some initial floor planning. Get a sense of the area you

need. Do some initial discussions with your fab vendors so that, as you said, the fab shop is the expert in the materials they want to use. Iterate with them to find out the right stackup, and then implement that in your design so that when you hand it back, you're not just throwing garbage over the wall blindly, but you're working as a team with your front end engineer and your vendors.

Korf: I'll second Jennifer's point, especially when it comes to flex and rigid-flex. I agree with what Jen is saying, and I really emphasize this on flex and rigid-flex, especially when a rigid designer is suddenly doing a flex board. It's a different world.

Dack: Dana points out that it's still a different world. It's like PCB design, but it's flex design, and there are so many different rules. Would you guys agree they have very different rules? It's still the laws of physics, but the materials are so different.

Korf: You're using boards that are meant to bend versus boards that aren't meant to bend. I say yes.

Kolar: No. I would say that means you have to think about things completely differently when you're doing the layout for either one of them, but ultimately, you still got a layout engineer who is doing that work and often needs to go back and forth and work on either. We see a lot of different types of rigid-flex designs now, as well as simple ribbon cables. You have those, you have that mix, so I think it is legitimate to have them be combined.

Dack: Mark, are many customers coming seeking specialized prototype designs for flex?

Thompson: Yes. Actually, that's an interesting question because many times, not designers, but predominantly engineers would say something like, "I want to jump on this bandwagon because I read this article about this new technology. I'm a big fan of the fewer moving parts, the better." Earlier, Jen said, "Here at Monsoon, we say the why." I can assure you I'm driving my design trainer crazy with all of my "why" questions because I'll go in and I'll be building design rules, for instance, and I'll see that, "I busted my 0.2-millimeter space value." I can just take it in and change it to something smaller, but is that correct? Did I do that correctly? That's where I say, "Why?" I think a lot of people are just jumping on the bandwagon, but there is a modicum of folks who are edging more toward flex and rigid-flex.

Shaughnessy: We're seeing a lot of people who say they're being forced into using flex.

Kolar: Yep. They're seeing that more and more, which will be interesting to see for long-term reliability. We like to say that flex boards really don't want to exist, but they just seem very fragile, and so we'll have to see how that works for long-term reliability.

Dack: I've designed a few flex circuits that didn't want to exist, and it's really hard. When they don't want to exist, you just have to keep at it, and time stretch.

Shaughnessy: Thank you all for doing this. Maybe we can get together live sometime and talk to each other in person over a cold beverage.

Thompson: Sounds good to me.

Kolar: Thank you for the opportunity.

Korf: Any time.

Dack: This was great. Thanks to this great group of people.

Related Content

- *Real Time with... I-Connect007: The App Notes and Fab Notes Roundtable* (PCB Design007.com, September 2020)