FACT SHEET

Table of Contents

High Level

Questions about ODB++ format scope

FACT SHEET

Table of Contents

Design-to-Manufacturing exchange processing

High level

1. Is ODB++ available to the whole industry?

Yes, ODB++ is available to all, both in terms of the format definition and implemented via commercially available interfaces and CAD/CAM tools. This is how it has been for over 15 years and will continue as such. Software vendors who want to implement the format in their commercial CAD/CAM products can gain access via the ODB++ Solutions Alliance; designers and manufacturers who want to understand the detail of the format can either view the native data in their applications or get a copy of the format specification by applying to the format custodian, or both.

2. Is ODB++ a standard?

In terms of a formal standard published by a standards-institution, no. However, ODB++ fits the category of de facto standard in the best tradition of formats and technologies that have reached "in effect" recognition as standards through high levels of adoption in day-to-day operations. Other examples of de facto standards are Adobe's PDF format, the USB interface from Intel, Microsoft and others, the Excellon NC format in the PCB industry, and Microsoft's Windows OS, to name just a few.

3. Who should implement ODB++, and why?

ODB++ can take a lot of cost, time-delay, and quality risk out of the PCB design-to-manufacturing transfer. By eliminating the need to work with multiple low-level files between PCB design and manufacturing, a lot of data manipulation is avoided and the fabrication and assembly process engineering work can begin faster and at a more automated level. Thus the designer gets his PCBs manufactured faster with lower supply chain risk, and the manufacturers reduce their operational costs.

4. Which CAD systems can support ODB++ output?

All of the well-known CAD vendors have ODB++ outputs. In most cases these interfaces are created and maintained via the OSA partnering program either in its current or previous forms.

5. Can all manufacturers accept ODB++?

The overwhelming majority (around 80%) of PCBs are fabricated through ODB++ compatible engineering processes. For assembly, the top-10 largest EMS companies are fully ODB++ compatible, and more than half of the top-50, and the number is growing steadily.

6. What viewers are available for ODB++?

Multiple ODB++ viewers are available in the market, some are free of charge, some are commercial products. Many of the viewers are integrated into the ODB++ output processors of the CAD or DFM systems.

7. Does ODB++ control the designer's intellectual property?

This is more of a question of the CAD/CAM tools rather than the format itself. ODB++ can carry net names, inner layer information, and component data up to the level you feel appropriate for transmission into manufacturing, according to how you use your CAD or DFM tool to create and manage the data. The intention of ODB++ is to give all possible information to manufacturing, in an intelligent and integrated data-structure, enabling the most efficient manufacturing.

8. Who owns the ODB++ format?

The format was invented by Valor Computerized Systems in 1995 and, following their acquisition by Mentor Graphics, ownership of the format was transferred accordingly. Mentor is committed to developing and maintaining the availability of ODB++ into the future, based on the same strategy as Valor in the early days of the format's history.

9. How will ODB++ be maintained for the future?

The format is currently at revision-7, with an update (revision-8) in the planning phase now. The format is developed to meet the emerging needs of the PCB interconnect industry on an evolutionary basis, according to the feedback from the thousands of users worldwide. The approach to implementation is "bottom-up", to ensure smooth transitions to deriving the value from the new versions of the format. First the format is implemented in the manufacturing tools, and only when leading manufacturers have become ready to work with the latest ODB++ version are the ODB++ output processors upgraded in the design systems.

Questions about ODB++ format scope

1. Variant designs? Does ODB++ handle them? How?

The ODB++ format itself enables the storage of variant data at the ODB++ step level and then at the lower component level through the use of system attributes. That is in the format definition. It is then up to the CAM or DFM application to utilize this functionality to manage the selection of the active variant.

2. BOM integration? Variable BOMs? No-pop? How is all that handled by ODB++?

The ODB++ format supports multiple BOMs and defines an active BOM. No-pop information that is part of the BOM is stored as part of the ODB++ data. This information can then be applied to the relevant component placements by a CAM application. The same CAM functionality enables the use of multiple BOMs by the user.

3. Attributes (surface finish, impedance, plugged vias, etc.) — how can that be handled by ODB++?

The ODB++ format enables the association of characteristics to data elements up and down the entire structure. This is typically done through the use of attributes.There are two types of attributes: system and user. System attributes are those that are defined as part of the ODB++ format. For example, there is an attribute "Capped Via" within the format definition for plugged vias.There are many other existing attributes already defined to support product model completeness. In addition, the user can define attributes expanding the ODB++ attribute definition independently of the format itself. This enables the transfer of design-related information not only internally, but with suppliers that are aware of these attributes and their meaning.

4. Pos/neg merging — Can it be handled by ODB++?

Yes, the very definition of ODB++ was created to support the merging of historical Gerber 274D/X data into single layers.

5. Is ODB++ suitable for RF?

Yes, the ODB++ format has the ability to transfer RF product models into manufacturing.

6. Does ODB++ "contain" drawings? Are drawings needed? How will the inspection process work if there are no drawings?

The ODB++ data structure can carry any drawings that you may generate. But overall, the strategy of ODB++ is to contain all data that is normally contained in drawings but to carry it embedded in the product-model itself as intelligent interlinked data. After that, it becomes a matter for your CAM or DFM software as to how to render that data for any purpose, including inspection. Having said that, if you want to generate drawings from the ODB++ core-data, you can do that too. In the end, we see the drawings as a potential output from the ODB++ data, not the other way around.

7. With ODB++, how would I deliver a reference netlist to my fabricator? Today I send IPC356; would I have to continue to do that?

The ODB++ format contains the EDA reference netlist as part of the ODB++ format. There is no need for an IPC 356 file as long as the ODB++ has been comprehensively generated by the CAD system. To protect IP, some applications provide a means to neutralize the net names into numerical nets. This conveys the conductivity requirements without exposing design-specific net names. In addition, unlike Gerber where all copper features are really unrelated, ODB++ copper features, drills, etc. are all tagged with the assigned net name. This greatly improves communication between the designer and suppliers when problems are located.

Design-to-Manufacturing exchange process:

1. What is the best way to plan to implement ODB++ between my design flow and my manufacturers? What are the steps to follow? What can go wrong, and how can I ensure that I do it right?

At a high level, the steps to moving from traditional CAM files based transfer to an intelligent ODB++ data exchange flow begins first with your organization committing to make the change. Similar to any other process improvement, the internal commitment must come first. It is too easy to fall back on old habits.Once the internal commitment is made, you need to inform and get the commitment of your suppliers to participate in your plan to improve the data exchange process. Imposing these changes on suppliers may result in failure just through misunderstanding. Instead, agreeing together that, for the better of the overall process, moving to intelligent ODB++ will simplify the transfer of the product model into manufacturing is essential to success.

Now to begin actual implementation. To be sure that your organization is comfortable with the ODB++ exchange, it is advisable to establish a process of checking the ODB++ against the files currently produced by your existing methods. Some CAD output processors or DFM solutions have the ability to compare the ODB++ generated by your EDA system to the traditional data exchange process automatically. Performing either of this operation will increase your organization's confidence in the ODB++ process.

Your supplier, at this time, will most likely want both the ODB++ and the traditional CAM files. Both will probably need to be provided, but your instructions to the supplier should be to create the finished product based on the ODB++ content only. The traditional CAM files are provided for the same reason as earlier: to build confidence at the supplier. Your supplier will want to perform similar operations that you have supporting their interest in providing you with a valid and quality product.

Over a short period of time, you and your suppliers will build confidence in the intelligent ODB++ data exchange process and you will be able to stop providing the traditional CAM files. The design transfer of the product model will completely covered by the single content contained within the ODB++ format.

2. What will the manufacturer do with the ODB++ data I send him? How will he use it to program his machines?

He will have software tools, DFM and CAM systems, that can work with ODB++ instead of the old-style traditional formats such as Gerber, placement lists etc. Using his CAM software, he will generate all the programs and instructions for his factory-floor operations. The advantage of using ODB++ in this way is that he avoids having to reverse-engineer multiple files into his software first, thus he gets to the "production ready" stage much faster and with less risk of mistakes along the way.

3. How many ODB++ users are there worldwide? How can I be sure that my suppliers can accept ODB++? How can we check that? How can we persuade a supplier to work with ODB++?

We estimate that about 80% of the world's PCBs are fabricated with CAM systems that accept ODB++ as the primary format, thus eliminating the need to reverse-engineer the product data at the manufacturing level. For PCB assembly and test, over half of the top-50 contract manufacturers worldwide, and all of the top-10, are ODB++ compatible in their engineering processes. The key to persuading a manufacturer to accept ODB++ is to state your goal of switching to the ODB++ method, get acceptance of that from all stakeholders, including communicating how you propose to go about it (deliver ODB++ and the old-style files for a while, but always insist that the ODB++ is the master definition of the product to be manufactured).

4. Are there differences between how ODB++ is used in software tools from different suppliers?

Certainly; the same goes for every data format out there. The way the data is used is controlled by the software (CAD, CAM, DFM, etc). The key point in favor of ODB++ here is that there is a partnering program available to support all software vendors in best-practice implementation of the format. All software vendors are welcome to join the ODB++ Solutions Alliance partnering program to ensure that the ODB++ format is properly understood and implemented in the best way possible. If you know of a software vendor whose ODB++ interface needs improvement, please ask the vendor to get in contact with the OSA partnering program – they will be welcomed.

5. Should a designer create different ODB++ for different purposes, or will the same ODB++ meet the needs of all manufacturing operations?

In principle, you can use one ODB++ data package to drive fabrication, assembly and test of the PCB. Send the same data to all the manufacturers (fabricators and assemblers) so you know that they all work from the same master version of the product you want manufactured. You will have fewer configuration control headaches that way! The key is to ensure that the ODB++ data generated in the first place is complete and meets the needs of all aspects of the manufacturing process. If, after that, you do want to make reduced ODB++ data sets for specific manufacturers, you can do that as well. Your CAD vendor should be able to support you with instructions on how to achieve that.

6. How can I protect the functional data of my design in ODB++, for example the net-names? How can I protect my design from being reverse engineered by a competitor? How can I control what data is included in the ODB++ and what is excluded?

The ODB++ format itself is an enabler. The applications that generate the ODB++ format product model need to be able to provide the intellectual property ("IP") protection the user finds necessary. Best-in-class ODB++ format CAD export capabilities enable the export of fabrication and assembly data separately or combined. In addition, there are more particular controls such as neutralizing CAD net names with generic names or simply excluding the net information altogether or, if desired, to provide the specific CAD component information or simply component representation with reference designator, etc. All of these controls are application based as requested by ODB++ users and not format specific.

In any case, not all information needed to reverse engineer a product to the design level is part of the ODB++ format specification. For example, specific constraints, routing rules, timing rules, etc. are not part of the ODB++ format because they are not required for manufacturing. Therefore reverse engineering a design from ODB++ is not a lot different than trying to reverse engineer a design from Gerber, Excellon and a netlist file, all of which is possible but only to a very limited extent.

7. In the future, will the old-style formats like Gerber and Excellon still be needed? If so, why? If somebody still needs Gerber, can they output it from ODB++?

If all manufacturing software in the design-manufacturing chain is ODB++ compatible, and all designers only send ODB++ to their manufacturers, then there will be no need to output Gerber and Excellon from CAD systems anymore. But Gerber and Excellon will always be extractable from an ODB++ file by using conversion software – just as most CAM systems can read ODB++ and output Gerber today. But, to emphasize the point – there will be no need to do that if all manufacturers can accept ODB++.

8. What are the reasons why a company might not use ODB++? What stops them?

The most common reason is lack of knowledge about the opportunity to switch over to ODB++, or lack of knowledge how to do it. That usually results in "playing it safe" and sticking with the old tried-and-tested methods (the time consuming and risky method of Gerber/Excellon/Netlist, etc.). After these factors, the next possibility is that the manufacturer may not have software that can accept ODB++; but that is a rare event after so many years of ODB++ being out there in the market.

9. How can I check whether my production machines can accept ODB++?

If you are referring to the CAM systems that generate the machine-level files that you take to your machines (like drilling, pick-and-place or test programs) then the ODB++ input capability should be quite visible in the user-menus of that software. If you cannot see the ODB++ input capability alongside the ability to read in Gerber or a BOM, you should contact your CAM software vendor and inquire whether you can expect an ODB++ interface in an upcoming update of the application. If they want assistance with understanding how to make the implementation they will be welcome to join the ODB++ Solutions Alliance.

10. My CAD does not produce ODB++ today. How can I get ODB++ supported on my CAD system? Do I have to write the software myself with scripts?

Most CAD vendors either have provided an ODB++ output already, or would be willing to do so in response to customer-demand. In all cases CAD vendors are welcome to join the ODB++ Solutions Alliance to help them create and maintain a good quality ODB++ output. There should be no need for you to write the software yourself – normally it is something you would expect to get from your CAD supplier.