How Model Connection Interfaces (MCI) can be Integrated with Electronic Module Description (EMD)

Walter Katz
Signal Integrity Software, Inc.

January 24, 2013

A Model Connection Interfaces (MCI) is information contained within a SPICE subckt that describes the ports of the model. This port information may include reference designators, pin name, signal name, xy coordinates, etc. One example of an MCI is Model Connection Protocol (MCP). MCP is a proprietary and public. There are other formats that are being used or considered.

Electronic Module Description (EMD) contains instances of SPICE subckts. It also contains information on the ports, which typically contains a reference designator and pin name. Since the number of ports on these subckts can be large, and the tool that generates the SPICE subckt commonly uses different reference designators and pin names the generation of the EMD port information is problematic. This document describes how Model Makers, particularly IC Vendors, can automate this port mapping function.

I will use flip chip to explain the integration, wire bond is similar with some minor enhancements to the integration.

We are given a flip chip with a well-defined set of bump pads, and a corresponding package with a matching set of pins (top layer surface pads). In the package cad data base each of these pins has a reference designator, pin number, net name and xy coordinate. Similarly, in the IC CAD data base each of the bump pads has a reference designator, pin number, net name and xy coordinate. One must assume that there exists a one-to-one mapping of these two sets of package data base pins and IC data base bump pads. If there is not one, it is a trivial piece of code to determine a scale, mirror, rotation and xy shift to one of these two sets, and then automatically generate such a pin mapping. Certainly IC Vendors have this information available to them.

A user will use Tool A to generate a Package SPICE Subckt that includes as ports some number of external package cad data base pins and some number of package cad data base pins that mate to the flip chip bump pads. Similarly a user of will use Tool B to generate an On-Die SPICE Subckt that includes as ports some number of flip chip bump pads and on die buffer pads. In this context, User is an IC Vendor.

Let us now assume that the Package SPICE Subckt has an MCI section that defines the connection to the PCB board, and a connection to the Die. Similarly the On-Die SPICE Subckt will have an MCI section that defines the connection to the package, and the connections to the “IBIS” buffers.

In my vision of EMD, the IC Vendor would generate an EMD of the Package which would have a list of external Package Pins with Pin_names defined as the Footprint Pin_names in the PCB package data base. The package EMD would have one instance of the bare Die .ibs file. The Pin_names of the bare die .ibs file would be the same Pin_names as the Pin_names of the die footprint in the PCB package data base.

The IC Vendor now has all of the information required to automatically create an interface to the Package SPICE Subckt in the package EMD file and the On-Die SPICE Subckt in the .ibs file using the MCI in each of the SPICE circuits, and the information he already has described above:

In essence, the MCI in the subckts generate by Tools A and B, along with the data the IC Vendor must already know about the interface between the package and board, and package and die allows him to automatically generate an EMD wrapper in the EMD file.