Missouri S&T Library and Learning Resources

Scholars' Mine

Masters Theses

Student Theses and Dissertations

Fall 2010

The study of a model for via transition and the multi-layer via transition tool GUI design

Ketan Shringarpure

Follow this and additional works at: https://scholarsmine.mst.edu/masters_theses

 Part of the Electrical and Computer Engineering Commons

Department:

Recommended Citation

Shringarpure, Ketan, "The study of a model for via transition and the multi-layer via transition tool GUI design" (2010). *Masters Theses*. 4858.
https://scholarsmine.mst.edu/masters_theses/4858

This thesis is brought to you by Scholars' Mine, a service of the Missouri S&T Library and Learning Resources. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

THE STUDY OF A MODEL FOR VIA TRANSITION AND THE MUTLI-LAYER VIA TRANSITION TOOL GUI DESIGN

by

KETAN SHRINGARPURE

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

2010

Approved by

James L. Drewniak, Advisor
Jun Fan
David Pommerenke

ABSTRACT

One of the many challenges faced by engineers working with the present design scenario is to estimate the extent to which a signal, with significant high frequency content, is affected when it is routed on a printed circuit board. The printed circuit board routing will include the transition through the geometries like micro-strip or strip-line transmission lines, via transitions, and irregularities or asymmetries in the aforementioned geometries. One of these discontinuities, the via transition, results in the interaction of the signal on the via and the cavity (plane-pair) through which it passes. The via transition modeling will help characterize a block in the signal path. Section 1 explains the cavity model, and derives an expression for the impedance at a port in a rectangular cavity. The via to cavity connection, and the via capacitance calculation is explained. Then, five practical examples are used to show the model assembly in a circuit fashion, and the results are compared to the measurements.

This modeling approach has been automated and integrated into the Multilayer Via Transition Tool, a tool that models all the common PCB geometries and provides the results as network parameters for the user defined ports. This tool is used for performance analysis and design optimization for the high speed PCBs. The tool includes a basic graphic user interface and an engine. Section 2 explains the design methodology for the provided graphic user interface. It explains interface design from the basic set of user inputs required by the engine to run. This section also talks about the difficulties in implementing the interface, and the required improvements for a professional tool.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor, Dr. James L. Drewniak, for his invaluable support throughout my graduation and his guidance in my research work. I thank Dr. Jun Fan for guiding me through his knowledgeable advice for improvements. I appreciate Dr. David Pommerenke for serving as a member of my thesis committee. I am also grateful to all the members of the "Electromagnetic Compatibility Laboratory" at Missouri University of Science and Technology, for helping me through this learning process. I thank Dr. Giuseppe Selli for his exceptional help with my research work.

I would like to thank my parents, my brother and my entire family for their absolute support and unconditional encouragement throughout my education. I would like to thank my uncle, Mr. Nandkumar Shringarpure, for his constant encouragement to learn more and work towards excellence. I would also like to extend my warm appreciation to all my friends, for their unrelenting support and encouragement. Finally, I thank God Almighty, for making this possible.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Page

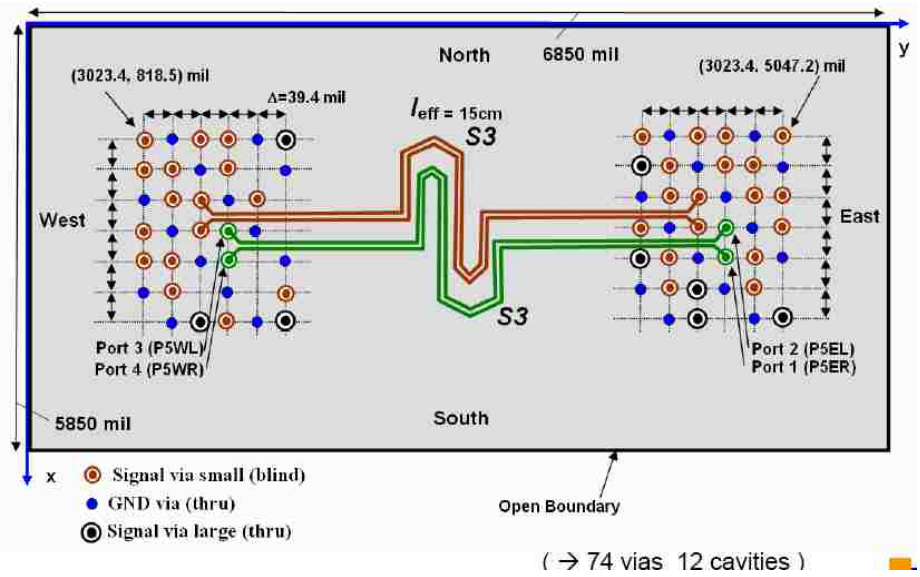
1. THE VIA MODEL

The performance of the printed circuit boards (PCBs) used in high-speed digital applications, if not carefully designed, may be significantly affected by power integrity and signal integrity issues. The power integrity issues may occur in the power distribution network (PDN) design which deals with providing a clean power and reference rails to the high speed integrated circuits (ICs) placed on the PCB. Recently, most ICs work with clock frequencies of more than several gigahertz. The switching in such ICs leads to fast transient currents on the power and ground rails, causing ripples in the supply voltage. Thus, noise can be introduced into a PDN, very easily. The IO signals of such ICs have to be routed on the same PCB, which may pick up the noise due to the coupling from the other signals or the power/reference rails, causing signal integrity issues too.

The PCBs used in the high-speed applications usually have several layers, many of which are complete planes or large area fills of power or reference nets. Such large metal planes form a parallel plate structure which may act as a waveguide for the high frequency noise coupled into the PDN. The noise in the PDN can cause signal integrity problems when coupled into signal nets, or lead to EMI issues when the noise couples in to a structure that can radiate. Thus, a challenge in PDN design is to make sure noise does not propagate easily along the PDN, and that it does not couple to other nets on the PCB. A common strategy followed by engineers is lowering the input impedance and transfer impedance of the PDN, for all the locations where noise can couple into the PDN.

Figure 1.1 shows a test vehicle used to study the performance of the signal links on a high -speed PCB. Top view shows the 74 vias placed in two via fields. The via fields include signal, reference and power net vias. The signal vias in two via fields are connected by the differential links on inner layers of the PCB. This is used as an illustration to show the complexity of high speed PCB designs. Such PCBs have several layers with metal planes or large area fills, which act as reference nets or power nets. The remaining layers are used for routing the signals. Due to the complexity of the implemented circuits and the size constraints, the signals have to be routed on two or more layers. A via is used for this transition of the signal from one layer to another. Also, the via is used to short circuit the reference planes or the power planes lying on different

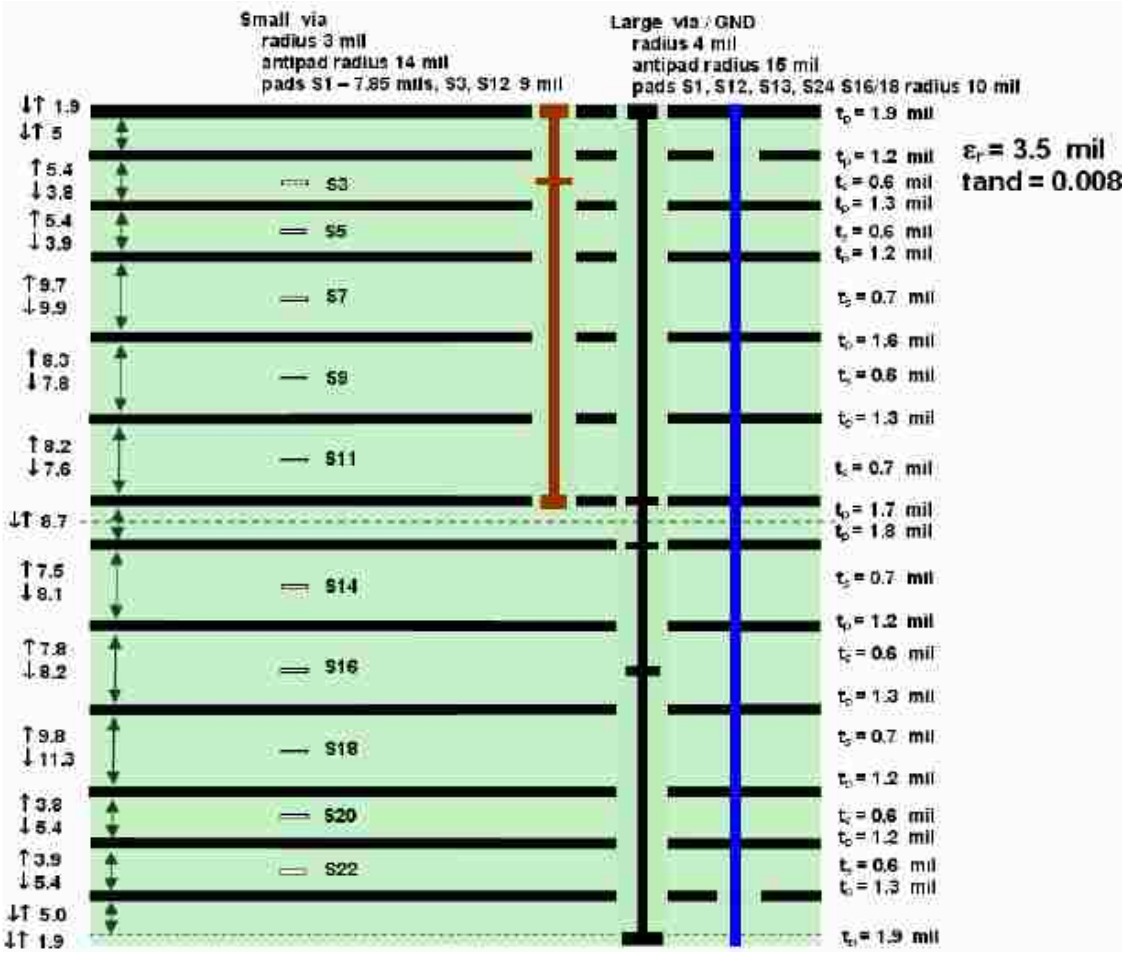
layers. At high frequencies, many such via are required to effectively short circuit the reference planes as the metal plane dimensions become electrically long, i.e., metal planes show distributed behavior. Thus, vias are used for connecting the metal with same net-name lying on different layers. These vias are hollow metallic cylinders, made by drilling holes in the PCB and plating these holes. Section 1.2 will further explain the geometry of the via. To address power integrity and signal integrity concerns, the engineers are required to perform EM analysis of the PCB geometry. For the PCB geometry, several modeling techniques like FDTD, FEM or PEEC, could be used. Many commercial full-wave modeling tools, which use one or more of such modeling techniques are available, e.g., CST Microwave Studio, Ansoft HFSS, Sigrity, etc. But the full-wave simulations require a lot of computational resources and also a very long simulation time.



(a)

Figure 1.1. A test board[1] for measurement and modeling designed to study real world PCB layouts: (a) Top view of 74 vias in two via fields connected with differential traces. (b) Cross section of PCB for understanding 24 layer stack up and the via dimensions for different via types.

[1] Image and design is courtesy Kevin Gu at IBM research centre and R. Rimolo – Donadio at Technical University Of Hamburg -Harberg.



(b)

Figure 1.1. A test board[2] for measurement and modeling designed to study real world PCB layouts (cont.)

This section explains an analytical model for the via transition, using the cavity model technique. The analytical formulation for the via transition, including the behavior of the metal planes, is used to model the common PCB geometries. The required computational time using the analytical formulation, is relatively very small, even for complex geometries, when compared to full-wave modeling techniques.

The model for via transition can be broken down into pieces which can be modeled individually based on the conventional electromagnetic theories and principles.

[2] Image and design is courtesy Kevin Gu at IBM research centre and R. Rimolo – Donadio at Technical University Of Hamburg -Harberg.

This section is organized as follows. Section 1.1 explains the cavity mode formulation and finds the solution for rectangular cavity based on Green's function. In section 1.2, the via to cavity connection is explained. The via is modeled as a circuit node connected to the cavity model network parameter block using capacitors for unconnected planes, which represent the via to plane capacitance. The via to plane capacitance is discussed in section 1.3. Section 1.4 explains how the single cavity structure is extended to a complete stack up. The final output of the model is in the form of a network parameters matrix between a given set of ports defined on the geometry. The network parameters help analyze the PCBs as a block in a link path and, allow comparison to measurement and full-wave simulation tool results. Section 1.5 uses a set of example cases for comparing modeling results to measurements. The results are for common geometries, so provide some insight into the physics of via transition. The model requires some effort, when each block is created and assembled in a circuit analysis tool. Section 1.6 explains how the complete process has been integrated into a single tool called the Multilayer Via Transition Tool (MVTT).

1.1. CAVITY MODEL

The cavity model is a modeling method used for a parallel plane pair geometry using planar circuits [1]. Planar circuits are electromagnetic circuits which are small compared to the wavelength only along one dimension. Based on this assumption, electromagnetic principles are used to find a model for the electric and magnetic fields present in the cavity. The cavity, in this via model, refers to two thin metal layers separated by a small distance. For cavity model application to printed circuit board, the basic structure is that of two arbitrarily shaped metal planes separated by a thin dielectric. Here, essentially, the dielectric thickness has to be smaller compared to the wavelength of electromagnetic waves, in order to form the planar circuit.

Figure 1.2 shows the cavity under consideration. Plane 1 and Plane 2 form the cavity of arbitrary shape separated by d. The cavity is bounded by the surface S with normal n and tangent $\hat{t}$. The dielectric thickness d is small compared to the wavelength (assuming the cavity is a planar circuit). Then, electric field is constant along the z direction. Due to the perfect electric conductor (PEC) boundary condition of top and

bottom conductor surfaces, the electric field only has a $\hat{z}$ component inside the cavity. Similarly, the tangential magnetic field has only the x and y components, i.e.,

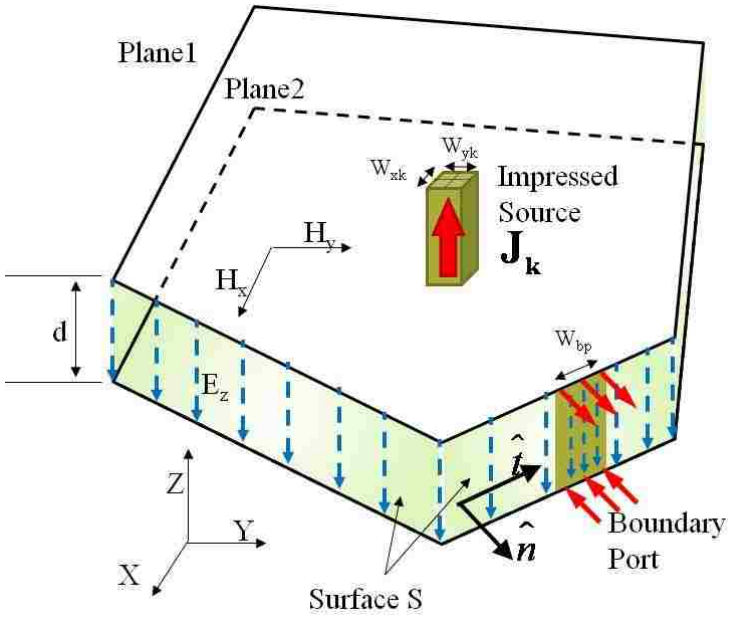


Figure 1.2. Electric and magnetic fields in a cavity formed by Plane1 and Plane2 separated by height d. Surface S forms a boundary along the open edge. A boundary port and impressed source are also shown.

$$E_x = E_y = 0;\ H_z = 0, \tag{1.1}$$

$$\frac{\partial}{\partial z} = 0. \tag{1.2}$$

The Maxwell's equations for a time harmonic electromagnetic fields are then

$$x\frac{\partial}{\partial y}E_z - y\frac{\partial}{\partial x}E_z = -j\omega\mu\ \ xH_x + yH_y \quad , \tag{1.3}$$

$$\frac{\partial}{\partial x}H_y - \frac{\partial}{\partial y}H_x = j\omega\varepsilon\ \ E_z \quad . \tag{1.4}$$

The wave equation can be obtained using (1.4) in the curl of (1.3), as

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \omega^2 \mu\varepsilon \right) E_z = 0 \ . \tag{1.5}$$

This is a two dimensional wave equation, which may be represented as

$$\nabla_t^2 + k^2 \ \ E_z = 0 \ , \tag{1.6}$$

where,

$$\nabla_t^{\ 2} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad , \tag{1.7}$$

$$k = \omega\sqrt{\mu\varepsilon} \qquad . \tag{1.8}$$

The boundary of the cavity along its edge, is considered ‘open’ in the region with no boundary ports. The physical meaning is that there is no current flowing outward from the cavity edge, i.e., from the ‘open boundary’ where a port is not defined. The current is on the bottom surface of the top conductor and the top surface of the bottom conductor. The magnetic field associated with this current lies in the x-y plane, in a direction normal to the direction of surface current. Along the edge of the cavity, the current normal to the edge is zero (open boundary), so it implies that the magnetic field tangential to the edge is also zero, but the magnetic field can exist in the direction normal to the edge. This condition can be characterized as a perfect magnetic conductor along the boundary or the cavity edge, such that magnetic field can only be perpendicular to the boundary surface. If $\hat{n}$ is an outward normal to the cavity edge, and $\hat{t}$ is the tangent to the cavity edge, as shown in Figure 1.2, the open boundary condition can be formulated as

$$\mathbf{J_n} = \hat{n} \times \mathbf{H}\big|_s = 0 \ ,$$

$$\mathbf{H_t} = \ 0 \, . \tag{1.9}$$

The relationship between the electric field and the magnetic field in the (1.3), Maxwell's equation in differential form for Faraday's law, shows that the gradient of electric field is always normal to the magnetic field, and is related by a non zero constant. Thus if the magnetic field is zero in a particular direction, (1.3) indicates that electric field gradient in that direction is zero, or the electric field remains constant moving in that direction, and vice versa. From this observation and the boundary condition that the tangential magnetic field is zero for the cavity edge, the electric field gradient in the direction normal to the edge of the cavity has to be zero, or

$$\mathbf{H_t} = 0 \Rightarrow \frac{\partial}{\partial n} E_z = 0 \quad . \tag{1.10}$$

The boundary condition on S enforces tangential electric field and normal magnetic field, with their components in the other direction to be zero. These conditions are enforced at the edge, and as no current is flowing out from the edge, there are no fringing electric field lines. Thus, the cavity model used here does not take into account the fringing field at the boundary.

The electric field is constant along the $\hat{z}$ direction, inside the cavity, so the potential difference between the top and bottom plates, defined as the integral of the electric field along the z direction at a location in the x-y plane, yields a product of the electric field intensity at that location and the cavity height, as

$$V = -\int \mathbf{E} \,.\, \hat{z}\, dz = -E_z d \ . \tag{1.11}$$

The relationship in (1.11) is important as it relates the field quantity to a circuit quantity, which is useful to define ports in the cavity model.

When the shape of the cavity is irregular, the segmentation approach is used to break down the irregular shape into simple polygons [2], for which the solution to the electric and the magnetic fields can be formulated easily. These simple polygons are connected together using boundary ports. The boundary ports are ports placed along the

edge of two cavities to be connected laterally, to enforce current and voltage continuity. The irregular shape is thus stitched together using these boundary ports, which are present on both the cavities at the corresponding locations. They are placed at an electrically small distance apart so that the field variations are contained in the continuity enforced at the ports. As shown in the Figure 1.2, the port current is defined as coming out of the cavity along the normal to the cavity edge on bottom surface of top plane, uniformly distributed across the width W_{bp} of the port. The return current is uniformly distributed along the width of the port on the top surface of the bottom plane. The voltage is constant along the width of the port. For boundary ports, the current and the voltage expressions in terms of the E field and surface current are

$$I = \int (\mathbf{J_s} \,.\, n) ds = (\mathbf{J_s} \,.\, n) W_{bp} \tag{1.12a}$$

$$V = -(\frac{d}{W_{bp}}) \int E_z ds \tag{1.12b}$$

The impressed current sources are used to model the vias present in the printed circuit board geometry. The via is a cylindrical PEC boundary inside the cavity, which is replaced by an impressed current source. The impressed current source $\mathbf{J_k}$ is a uniform current distribution oriented in the $\hat{z}$ direction, and placed at the via location (x_k, y_k). For a source with rectangular cross section, the dimensions are W_{xk} and W_{yk} along x and y direction respectively. As the current density is uniform for the x-y cross section of the source, the current flowing out of the source is

$$I_k = \int_{x_k - W_{xk}/2}^{x_k + W_{xk}/2} \int_{y_k - W_{yk}/2}^{y_k + W_{yk}/2} \mathbf{J_k} \,.\, \hat{z} \; dxdy \;\; , \tag{1.13}$$

$$I_k = \mathbf{J_k} \,.\, \hat{z} \;\; W_{x_k} W_{y_k} \;\; . \tag{1.14}$$

Also, the source voltage can be averaged over the area of the source which gives

$$V_k = \frac{1}{W_{x_k} W_{y_k}} \int_{x_k - W_{xk}/2}^{x_k + W_{xk}/2} \int_{y_k - W_{yk}/2}^{y_k + W_{yk}/2} -E_z(x,y)\, d\ \ dxdy \ . \tag{1.15}$$

These sources should appear in the Maxwell's equation for time harmonic fields used to find the wave equation. Then, the wave equation (1.6) becomes

$$\nabla_t^2 + k^2 \ \ E_z = j\omega\mu \ \ \mathbf{J_k} \,.\, \hat{z} \quad . \tag{1.16}$$

The solution is found for the open boundary condition, or a short boundary condition. For open boundary, the normal derivative of the electric field intensity in the cavity is zero along the boundary, and for the short boundary, the electric field intensity is zero, as given by

$$\frac{\partial}{\partial n} E_z(x,y) = 0 \ , \tag{1.17}$$

$$E_z = 0 \ , \tag{1.18}$$

for open and short boundary conditions, respectively. The solution to the wave equation in (1.16) has been found using the Green's function method. As an example, the solution for the open boundary rectangular cavity using the Green's function in series form is shown. The presented solution follows the standard methodology given in [3].

Consider the rectangular cavity with x, y, and z dimensions as a, b, and d respectively. The Green's function satisfies the auxiliary differential equation and boundary conditions as

$$\nabla_t^2 + k^2 \ \ G\ \ x, y; x', y' \ = \delta(x - x')\delta(y - y') \ , \tag{1.19}$$

$$\frac{\partial}{\partial n} G\ \ x, y; x', y' \ = 0 \ . \tag{1.20}$$

where n is the normal to the boundary. The Green's function is taken to be a series summation of an orthogonal basis function set $\psi_{mn}(x, y)$. Each basis function of this set will satisfy the auxiliary differential equation and the boundary condition applicable to the Green's function in equations (1.19) and (1.20) as

$$\nabla_t^2 + k_{mn}^2 \ \psi_{mn}(x, y) = 0 \ , \tag{1.21}$$

$$\frac{\partial}{\partial n}\psi_{mn}(x, y) = 0 \ . \tag{1.22}$$

The basis function can be written in the variable separable form by choosing

$$\psi_{mn}(x, y) = f(x)\, g(y) \ . \tag{1.23}$$

Using (1.23) in (1.21),

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + k_{mn}^2 \right) f(x)\, g(y) = 0,$$

$$\frac{1}{f(x)} \frac{\partial^2}{\partial x^2} f(x) + \frac{1}{g(y)} \frac{\partial^2}{\partial y^2} g(y) = -k_{mn}^2 \ .$$

Since f and g are functions of x and y alone, respectively, the differential equations can be separated out as,

$$\left.\begin{aligned} \frac{1}{f(x)} \frac{\partial^2}{\partial x^2} f(x) &= -k_x^2 \ , \\ \frac{1}{g(y)} \frac{\partial^2}{\partial y^2} g(y) &= -k_y^2 \ . \end{aligned}\right\} \tag{1.24}$$

Where,

$$k_{mn}^{\ 2} = k_x^{\ 2} + k_y^{\ 2} \quad . \tag{1.25}$$

The differential equations (1.24), have general solutions

$$f\ x\ = A\sin\ k_x x\ + B\cos\ k_x x \tag{1.26}$$

,

$$g\ \ y\ = C\sin\ k_y y\ + D\cos\ k_y y \quad . \tag{1.27}$$

To find the constants, the boundary conditions for the basis function set are used. The open boundary conditions are written more explicitly for each edge of the cavity, to solve for each constant. Then, at y = 0 and for x in [0, a],

$$\begin{gathered}\frac{\partial}{\partial y}\psi_{mn}\ \ x,0\ = 0 \\ \Rightarrow f(x)g'(0) = 0 \\ g'\ 0\ = k_y C\cos\ 0\ - k_y D\sin\ 0\ = 0 \\ \Rightarrow C = 0\ ,\end{gathered} \tag{1.28}$$

and at y = b and x in [0, a]

$$\begin{gathered}\frac{\partial}{\partial y}\psi_{mn}\ \ x,b\ = 0 \\ \Rightarrow f(x)g'(b) = 0 \\ g'\ b\ = -k_y D\sin\ k_y b\ = 0 \\ \Rightarrow k_y = \frac{n\pi}{b}; n = 0,1,2\ldots\ .\end{gathered} \tag{1.29}$$

Similarly, for x = 0 and x= b, results in,

$$A = 0\ , \tag{1.30}$$

$$k_x = \frac{m\pi}{a}; m = 0,1,2\ . \tag{1.31}$$

Then, the solution to the auxiliary differential equation (1.21) is

$$\psi_{mn}\ (x,y\ ) = BD\cos\left(\frac{m\pi}{a}x\right)\cos\left(\frac{n\pi}{b}y\right)\ . \tag{1.32}$$

The basis functions are orthogonal for x in [0, a] and y in [0, b], i.e.,

$$\int_0^a\int_0^b \psi_{mn}\ (x,y\ )\ \psi_{pq}\ (x,y\ )\ dydx = \begin{cases} 1 & \forall m = p; n = q \\ 0 & otherwise \end{cases} \tag{1.33}$$

$$BD^{\ 2}\int_0^a\left[\cos\left(\frac{m\pi}{a}x\right)\right]^2 dx\int_0^b\left[\cos\left(\frac{n\pi}{b}y\right)\right]^2 dy = 1\ , \tag{1.34}$$

and

$$\int_0^a\left[\cos\left(\frac{m\pi}{a}x\right)\right]^2 dx = \begin{cases} a & ;m = 0 \\ a/2 & ;m \neq 0 \end{cases}$$

$$\int_0^b\left[\cos\left(\frac{n\pi}{b}y\right)\right]^2 dy = \begin{cases} \mathrm{b} & ;n = 0 \\ b/2 & ;n \neq 0 \end{cases}\ .$$

Then,

$$BD = \sigma_m\sigma_n/\sqrt{ab}\ ,$$

and the basis function from (1.32) becomes,

$$\psi_{mn}\ (x,y\ ) = \frac{\sigma_m\sigma_n}{\sqrt{ab}}\cos\left(\frac{m\pi}{a}x\right)\cos\left(\frac{n\pi}{b}y\right)\ , \tag{1.35}$$

where

$$\left.\begin{aligned}\sigma_m &= \begin{cases} 1 & ;m=0 \\ \sqrt{2} & ;m\neq 0 \end{cases} \\ \sigma_n &= \begin{cases} 1 & ;n=0 \\ \sqrt{2} & ;n\neq 0 \end{cases}\end{aligned}\right\} . \tag{1.36}$$

The Greens function in series form can then be written using bilinear equation [3]

$$G\ x,y;x',y'\ = \sum_m \sum_n \frac{\psi_{mn}\ x',y'\ \psi_{mn}\ x,y}{k^2 - k_{mn}{}^2}\ , \tag{1.37}$$

or,

$$G\ x,y;x',y'\ = \sum_m \sum_n \frac{\sigma_m^2 \sigma_n^2}{ab} \frac{\cos\ k_x x'\ \cos\ k_y y'\ \cos\ k_x x\ \cos\ k_y y}{k^2 - \left[\ k_x{}^2 + \ k_y{}^2\right]}\ . \tag{1.38}$$

The solution to the differential equation (1.16) and boundary conditions (1.17) for a rectangular cavity can then be found from

$$E_z\ x,y\ = j\omega\mu \int_0^b \int_0^a G\ x,y;x',y'\ \ \mathbf{J}(x',y')\ .\ \hat{z}\ dx'dy'\ . \tag{1.39}$$

The solution to electric field in (1.39) can be used to get an impedance relationship between the voltages and the currents. For an N port cavity the Z parameter matrix is found by placing a current excitation at one port at a time and calculating the voltage at all other open circuited port locations, based on the definition of Z parameters for an N port system. Thus, in accordance to impressed source definition, a $\hat{z}$ directional source $\mathbf{J_i}$, is placed centered at (x_i, y_i), and is uniform across the cross section of dimensions W_{xi} and W_{yi}. Using (1.14) and (1.15) for the impressed source current and voltage respectively, the Z_{ij} is

$$Z_{ij} = \left.\frac{V_i}{I_j}\right|_{I_k=0;\forall k\neq j} = \frac{\dfrac{1}{W_{x_i}W_{y_i}}\displaystyle\int_{x_i-W_{xi}/2}^{x_i+W_{xi}/2}\int_{y_i-W_{yi}/2}^{y_i+W_{yi}/2} -E_z(x,y)\,d\;\;dxdy}{\displaystyle\int_{x_j-W_{xj}/2}^{x_j+W_{xj}/2}\int_{y_j-W_{yj}/2}^{y_j+W_{yj}/2} \mathbf{J_j}\,.\,\hat{z}\;\;dx'dy'}\;. \tag{1.40}$$

The solution for $E_z(x, y)$ with only one source J_j is

$$E_z\;\;x,y\;\big|_{\mathbf{J}=\mathbf{J_j}} = j\omega\mu\int_0^b\int_0^a G\;\;x,y;x',y'\;\;\;\mathbf{J_j}(x',y')\,.\,\hat{z}\;\;dx'dy'\;\;\;, \tag{1.41}$$

$$E_z\;\;x,y\;\big|_{\mathbf{J}=\mathbf{J_j}} = j\omega\mu\;\;\mathbf{J_j}\,.\,\hat{z}\int_{y_j-W_{yj}/2}^{y_j+W_{yj}/2}\int_{x_j-W_{xj}/2}^{x_j+W_{xj}/2} G\;\;x,y;x',y'\;\;dx'dy'\;\;. \tag{1.42}$$

Using Green's function series expression from (1.38) in (1.42), E_z is

$$E_z\;\;x,y\;\big|_{\mathbf{J}=\mathbf{J_j}} = j\omega\mu\;\;\mathbf{J_j}\,.\,\hat{z}\;\;\frac{1}{ab}\sum_m\sum_n \frac{4\sigma_m\sigma_n}{k_x\;\;k_y}\frac{\left[\begin{array}{l}\cos\;\;k_x x\;\cos\;\;k_y y\;\cos\;\;k_x x_j\;\cdot\\ \cos\;\;k_y y_j\;\;\sin\left(\dfrac{k_x W_{xj}}{2}\right)\sin\left(\dfrac{k_y W_{yj}}{2}\right)\end{array}\right]}{k^2-\left[\;\;k_x\;^2+\;k_y\;^2\right]}\;. \tag{1.43}$$

Using (1.43) in (1.40), the impedance is

$$Z_{ij} = -\frac{j\omega\mu d\,\dfrac{1}{ab}}{W_{x_j}W_{y_j}W_{x_i}W_{y_i}}\sum_m\sum_n\left[\frac{4\sigma_m\sigma_n}{k_x\;\;k_y}\right]^2\frac{\left[\begin{array}{l}\cos\;\;k_x x_i\;\cos\;\;k_y y_i\;\cos\;\;k_x x_j\;\cos\;\;k_y y_j\\ \sin\left(\dfrac{k_x W_{xi}}{2}\right)\sin\left(\dfrac{k_y W_{yi}}{2}\right)\sin\left(\dfrac{k_x W_{xj}}{2}\right)\sin\left(\dfrac{k_y W_{yj}}{2}\right)\end{array}\right]}{k^2-\left[\;\;k_x\;^2+\;k_y\;^2\right]}\;,$$

$$Z_{ij} = j\omega\mu d \frac{1}{ab} \sum_{m=0}^{m=\infty} \sum_{n=0}^{n=\infty} \frac{\left[\begin{array}{l} \sigma_m^2 \sigma_n^2 \cos\ k_x x_i\ \cos\ k_y y_i\ \cos\ k_x x_j\ \cos\ k_y y_j\ . \\ \operatorname{sinc}\left(\frac{k_x W_{xi}}{2}\right) \operatorname{sinc}\left(\frac{k_y W_{yi}}{2}\right) \operatorname{sinc}\left(\frac{k_x W_{xj}}{2}\right) \operatorname{sinc}\left(\frac{k_y W_{yj}}{2}\right) \end{array} \right]}{\left[k_x^{\ 2} + k_y^{\ 2} \right] - k^2} \quad (1.44)$$

,

where,

$$\sigma_m = \begin{cases} 1 & ; m = 0 \\ \sqrt{2} & ; m \neq 0 \end{cases},$$

$$\sigma_n = \begin{cases} 1 & ; n = 0 \\ \sqrt{2} & ; n \neq 0 \end{cases},$$

$$k_x = m\pi / a\ ,$$

$$k_y = n\pi / b\ .$$

The m and n, in the impedance equation (1.44), stand for the $TM_{z0,mn}$ modes with functional variation in the x and y directions, respectively. The equation for Z_{ij} shows its dependence on the location of the source and the observation port, and their dimensions, the cavity dimensions, the material properties, and the frequency at which the impedance is calculated. The low frequency behavior of the cavity is like a parallel plate capacitor which is given by the m=0 and n=0 mode. As frequency is increased, the distributed behavior of the cavity can be seen by the standing waves formed from the field reflections from the boundary. These resonances, related to the cavity dimensions appear in the self and transfer impedance as alternating poles and zeros. These are the propagating modes. Depending on the highest frequency of simulation and the cavity dimensions, only a certain set of modes will act as the propagating modes. The remaining modes are the evanescent modes which contribute to the self and mutual inductance. The transfer impedance is less sensitive to these evanescent modes, compared to the self impedance. Apart from rectangular geometry, similar expressions can be derived for triangular and circular shaped cavities as well [4], [5].

1.2. THE VIA TO CAVITY CONNECTION

The via is an interconnect used between different layers of a printed circuit board (PCB). It connects or short circuits the copper regions of the same nets, lying on different layers. The geometry of a via is like a hollow metallic cylinder. The via is also known as a plated through hole (PTH). To create the via geometry, as a step in the manufacturing process of a PCB, holes are drilled at the specific locations on the PCB, and then plated with metal. Along the hole, the layers where metal is present in the form of area fills or trace pads, will form a connection to the plating of the hole. For the layers where no connection has to be made, the metal is etched away from the location of the hole. This may be done before the holes are drilled during the etching process, wherein the excess or undesired metal is selectively etched away. This region around the via with metal etched away to avoid connection is known as the anti-pad. A signal via connecting two traces is shown in the Figure 1.3, as it passes through two signal reference planes.

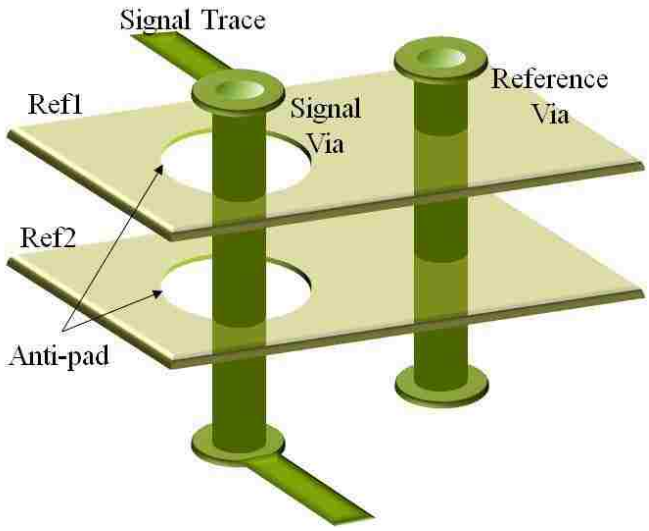


Figure 1.3. PCB geometry showing a signal via connecting two traces and a reference via. The two reference planes shown have anti-pads for signal via and no anti-pads for the reference via, thus controlling the connection to the via.

The metal on the reference layers is etched out at the location of the signal via, to ensure no connection. Thus two anti-pads are formed at the reference layers at the signal

via location. Also, at the layer where the via is connected to the trace, pads are placed to ensure the trace-to-via connection while accounting for the drill location tolerance. The reference via has no anti-pads, and the via directly connects to the metal on reference layers.

The printed circuit boards built for RF applications or high-speed digital applications will usually have multiple metal layers, many of which are full plane layers formed of power or reference nets. Multiple vias may pass through these planes, some of which may be connected to the planes, and rest would have anti-pads around them. The high frequency currents on the vias are the surface currents oriented in the vertical or $\hat{z}$ direction, as shown in the Figure 1.4. When a via and plane are not connected, i.e., an anti-pad is present at the plane layer, then the magnetic fields generated by the via currents will induce radial currents on the unconnected plane. For a via connected to the plane, depending on the current direction, the via current is either spreading into that plane or necking down from the plane into the via. These currents will cause magnetic fields within the parallel plate structure.

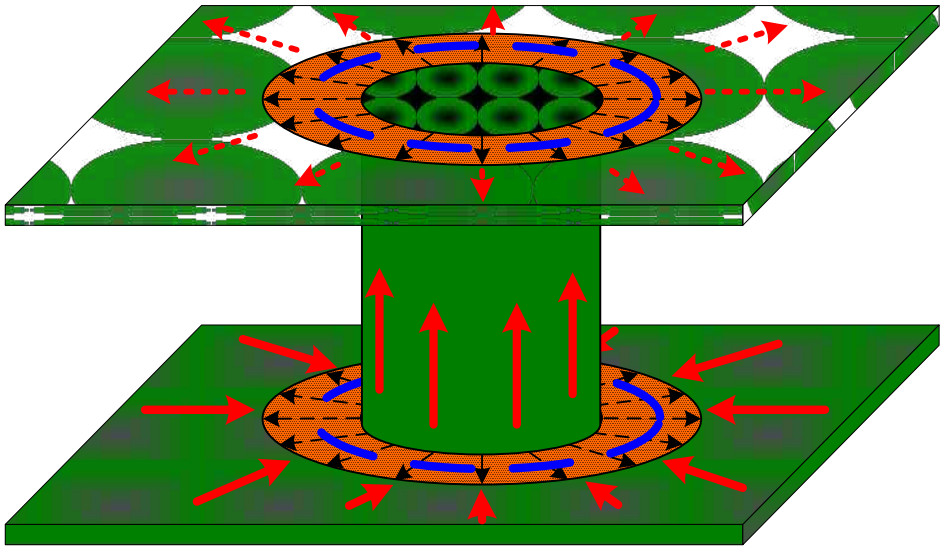


Figure 1.4. The via – cavity geometry with the electric field map and the magnetic field map is shown. The antipad has radial electric field and concentric magnetic field like a coaxial transmission line. The magnetic field induces radial current on the cavity planes in opposite directions.

The Figure 1.5 also shows the impressed current source placed in a cavity. The impressed current source forces a radial current on both planes, in opposite directions, and also results in a magnetic field which is concentric with the impressed source. The nature of the fields and the surface currents on the planes for the impressed current source shown in Figure 1.5 is the same as that for and the via geometry shown in Figure 1.4, outside the anti-pad region. Thus, the cavity model with an impressed current source at each via location can be used to model the plane-pair behavior. The expression for self and transfer impedance for any number of impressed current sources placed in a cavity was formulated in Section 1.1. As the Z parameters can be used as a network parameter block with a defined relationship between the voltages and currents for each port, the model including complete plane-pair behavior can be used for circuit analysis. Such an impedance block is referred to as the Zpp block or plane-pair impedance block in a cavity model. The Zpp block will have as many ports as the number of vias in the cavity. To perform circuit analysis, the via has to be connected to the Zpp block.

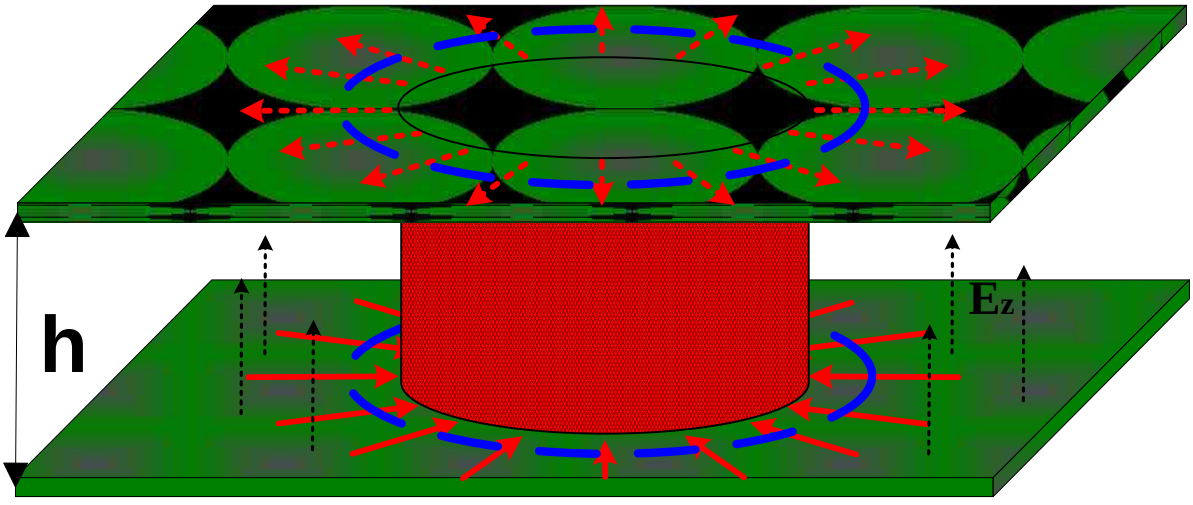


Figure 1.5. The magnetic field and the surface currents on the planes due to the impressed current source placed in the cavity at a via location. The surface current is shown for bottom surface of top layer and top surface of bottom layer. The magnetic field shown is inside the cavity.

The Zpp block has two terminals for each port. The two terminals of a Zpp port correspond to the terminals of the impressed current source in the cavity model. The terminal is a geometry feature where the voltage and current are defined. The impressed current source is defined between the two planes. The port voltage used in the formulation is the average voltage across the impressed source cross section and the port current is the current through the impressed source, which is the same as the current injected into a plane (spreading current) or the current necking into the source. Thus the terminals of this source lie, one on each plane, at the outer edge of the source, i.e. the circumference of the source, on each plane forms a terminal. The via, an electrically small hollow PEC cylinder, is considered as a single node for the purpose of circuit analysis. When the via is connected to a plane, this node is shorted to the Zpp port terminal for that plane. When an anti-pad is present between the via and a plane, there will be some capacitance between the via barrel and the plane surfaces. In this case, a capacitor is placed between Zpp port terminal for that plane and the via node. Section 1.3 will explain more about the via to plane capacitance calculation.

The Figure 1.6 shows three different geometry configurations, from left to right, for a via passing through a cavity, and the circuit connection of the via to the cavity port below the corresponding configuration. In the first configuration, a signal via passes through two reference planes, there is an anti-pad on both layers, and thus Zpp terminal to via node connection has a capacitor for each Zpp terminal. In the second geometry, a reference (REF) via passes through the power (PWR) and reference (REF) cavity, it is connected to the reference plane but has an anti-pad at the power plane. For the circuit model, the Zpp port terminal for reference plane is shorted to via, but the power plane terminal in connected to the via node through a capacitor. The third geometry is a shorting via, connected to both planes of the cavity. The Zpp port terminals are both connected to the via node, and thus short circuited.

All geometries discussed, were single-cavity geometries. This one cavity geometry will form the basic building block for analyzing complex stack ups. In most cases, many cavities are stacked vertically to form a multi-layered structure. Current can flow along the via surface from one cavity to another, where an anti-pad exists. Across the anti-pad, for current flowing along the via outer surface, the electric field is radial

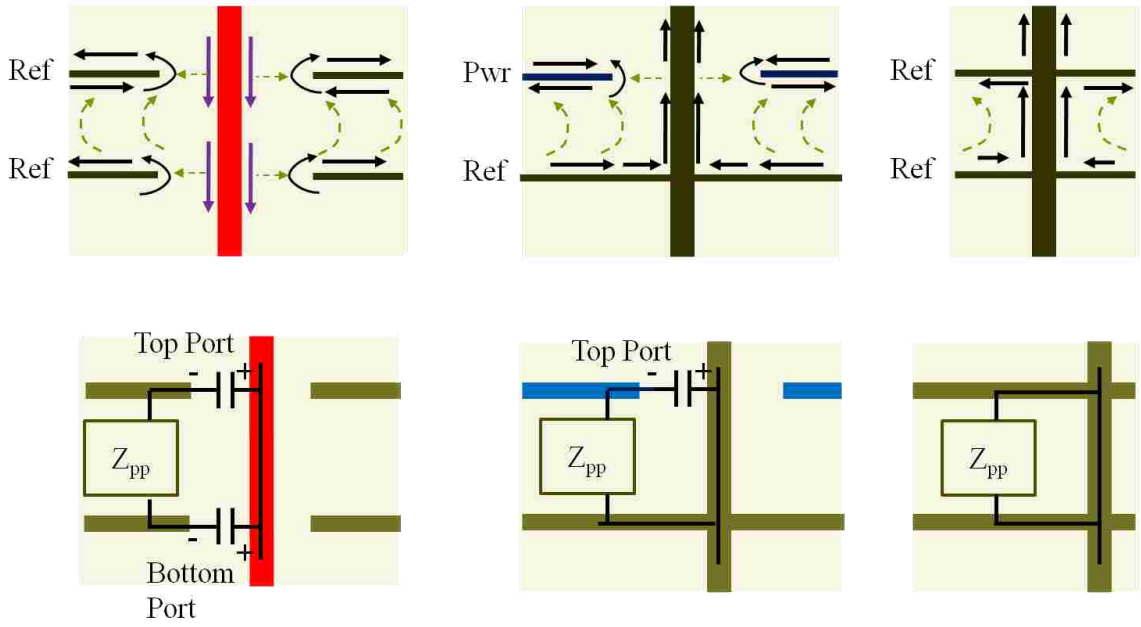


Figure 1.6. Zpp port to via connection is shown for three different geometry configurations. Each geometry with the current map and the circuit model is shown below it. If an anti-pad exists between the via and the plane, then the via is connected to the Zpp port terminal through a capacitor, else the via is short circuited to the Zpp port terminal.

from the via to the anti-pad edge, and the magnetic field goes in concentric circles around the via barrel, as shown in Figure 1.4. The field map represents those of a coaxial transmission line. Thus the voltage across the anti-pad and the current flowing through it can be well defined, allowing a port definition at this location. This port is calle a via port. The positive terminal is the outer surface of via barrel at the anti-pad, i.e., anti-pad inner edge, and the negative terminal is the anti-pad outer edge at the plane. These ports exist at each anti-pad, across the capacitor, e.g., Figure 1.6 shows the model with via ports at each anti-pad. The via port acts as an external port for characterizing the via geometry, and is also used to connect this model to other models by enforcing voltage and current continuity. These ports are used for modeling the multilayered structures, where the cavities are connected vertically at these via ports, as shown in Section 1.4.

1.3. CALCULATING THE VIA TO PLATE CAPACITANCE

As explained in Section 1.2, the via connection to the cavity model Zpp is through a capacitor, where the anti-pad occurs. The capacitance between the via and the unconnected plane can be divided into different parts as shown in Figure 1.7. The accuracy of the capacitance calculation will affect the final result of the network parameters, depending on the geometry being analyzed. This makes the capacitance calculation important for the entire model.

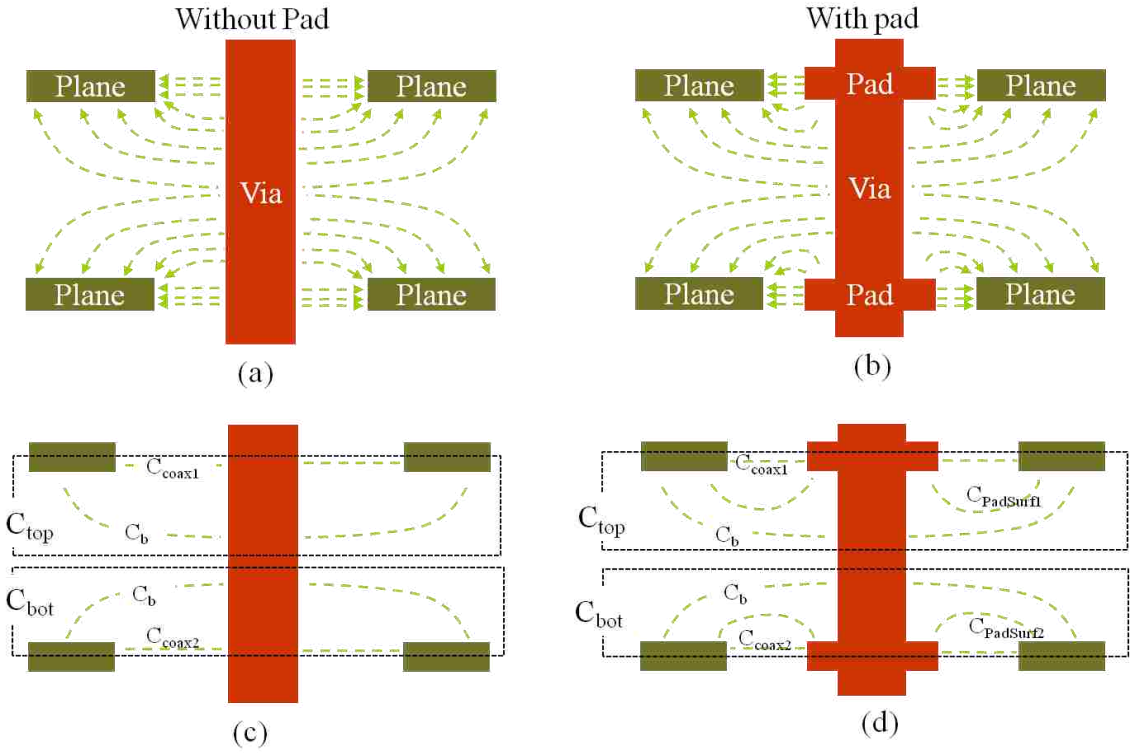


Figure 1.7. Electric flux lines between the via and the planes of the cavity are shown for via without pad and with pad cases. The capacitance is grouped C_{top} with top plane and C_{bot} with bottom plane. Both groups have one capacitance from via barrel to plane horizontal surface and another capacitance between the curved surfaces of the anti-pad.

As per Figure 1.7, one part of the capacitance exists between the via barrel and the top plane's bottom surface, and similarly between the via barrel and the bottom plane's top surface, named as C_b. This part depends on the anti-pad size, via drill size,

and the height of the cavity formed by this plane with other planes above and below it. There is another part of the capacitance in the anti-pad region, C_{coax}, which can be calculated as coaxial capacitance, which depends on the plane thickness, anti-pad size, and via drill size. When a pad is present on the via at the plane layer, the capacitance from the top and the bottom surfaces of the pad, to the plane surface, $C_{padsurf}$, needs to be accounted for.

In Figure 1.7a and Figure 1.7b, the geometry for and electric flux map is shown for case without pad and with pad respectively. In the Figure 1.7c and Figure 1.7d the capacitance is divided into pieces. The capacitances are calculated as individual pieces and then added up to get the value of capacitor to be connected together cavity port and the via. The capacitance between via and top terminal of the cavity port is given in equation (1.45). The capacitance between bottom plane terminal of the cavity port and via is given in the equation (1.46).

$$C_{top} = C_b + \frac{1}{2} C_{coax1} \tag{1.45}$$

$$C_{bot} = C_b + \frac{1}{2} C_{coax2} \tag{1.46}$$

The same two capacitances, for the case with pad is given as follows.

$$C_{top} = C_b + \frac{1}{2} C_{coax1} + C_{PadSurf1} \tag{1.47}$$

$$C_{bot} = C_b + \frac{1}{2} C_{coax2} + C_{PadSurf2} \tag{1.48}$$

The capacitance for the coaxial region is shared by the cavities above and below the plane. There is a capacitor connecting the via to each cavity impedance block. So, half of the anti-pad capacitance value is used in each capacitance. Eventually, they appear in parallel and get added up.

For fast computation an analytical formulation [6] was used. This method provided the required accuracy for via barrel to plane capacitance geometry. However the

result was found to be less accurate when a pad would be present on the via, at a plane layer. Also, the capacitance from a signal layer pad to the planes was not accounted for in this formulation. Thus to improve the accuracy of the capacitance calculation, a new approach used. A 2D FEM tool [7] was used. The FEM method, would use cylindrical symmetry to get the capacitance between the via and the planes. This method accounts for the capacitance due to pads on any layers.

1.4. VIA MODEL FOR MULTILAYERED PCBS

To extend the model to multi-layered structures, segmentation approach is used [2]. The practical PCB has many cavities of irregular shapes and formed over a multilayer stack up. The coupling between vertically stacked cavities is through the voids in the metal planes, like anti-pads or splits (gaps). The voids in the planes may also result in cavities between the non adjacent layers.

The cavities which are stacked vertically couple through the anti-pad region. For connecting cavities in the vertical direction, a via port is defined in every via anti-pad region, between the cavities to be connected. Section 1.2 explains in detail how a via ports is defined at the anti-pad. Corresponding ports from two cavities are connected together, to enforce voltage and current continuity between cavities stacked on top of each other. Due to the current and voltage continuity at the circuit level, there is electric and magnetic field continuity at the field level, as the voltage and currents at the ports are well defined from the fields. As the continuity is enforced the coupling through the anti-pad is accounted for in the model.

The Figure 1.8 shows the connection of vertically stacked cavities. The geometry shown has two cavities with three vias, one each of the type signal, reference and power. The stack up has a power plane between two reference planes. The circuit model for each cavity is shown to the right of the geometry. The vias are connected to the network parameter block (Zpp) port terminals through a capacitor, where anti-pad is present and short circuited where a connection is present from the via to the plane. Thus individual circuit models of each cavity are formed. These individual circuit models can be connected together at the via ports defined across the same anti-pad, in the geometry.

Then, a final circuit model is obtained for the entire (two cavity) geometry, which can be simulated to get the required network parameter matrix.

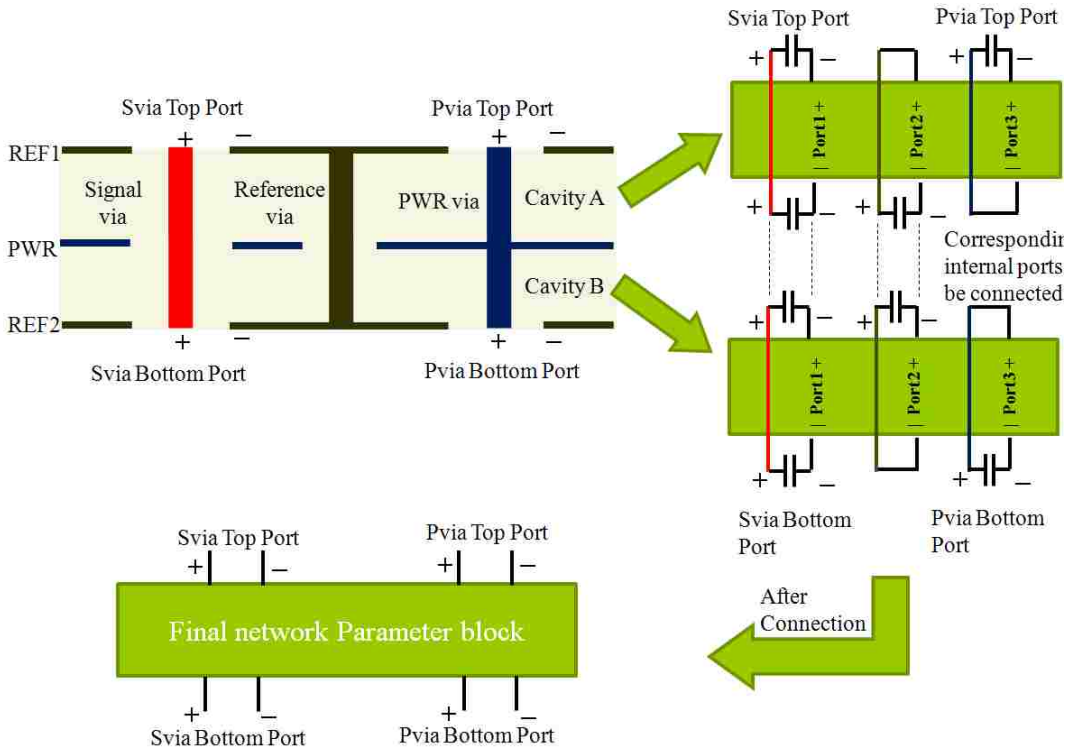


Figure 1.8. Connecting the cavities stacked vertically. Geometry shown has two cavities with three vias. The two cavities are modeled individually first and then connected at the corresponding via ports of the same anti-pads to get the complete multilayer model.

The irregular shaped cavities are segmented into regular shapes like rectangles and triangles. There exist closed form expressions for the cavities with regular shapes. Many boundary ports are defined along the boundary of these regular shaped cavities and they are solved for the impedance matrix. These cavities are then assembled together by connecting the corresponding boundary ports defined along the edges of these cavities. Thus continuity is enforced along the edges for these cavities using boundary ports which stitch together the smaller cavities to form the irregular shaped cavity. Thus it is possible to connect the cavities laterally and vertically. These techniques are used together to model the entire multilayered printed circuit board.

1.5. MODELING EXAMPLES

The results of this modeling technique are compared with measurements for practical geometries, designed and implemented by Giuseppe Selli. The geometry structures are multilayered via structures which were designed to model practical PCB scenarios. Following subsection explains the geometry of these test vehicles, the calculation of the impedance matrix and the anti-pad capacitance, the spice like model assembly in ADS, and validation of the modeling results to some practical measurements. Some expectations from the model for the given geometry and inferences from the comparison are also shared. The stack up is shown in Figure 1.9.

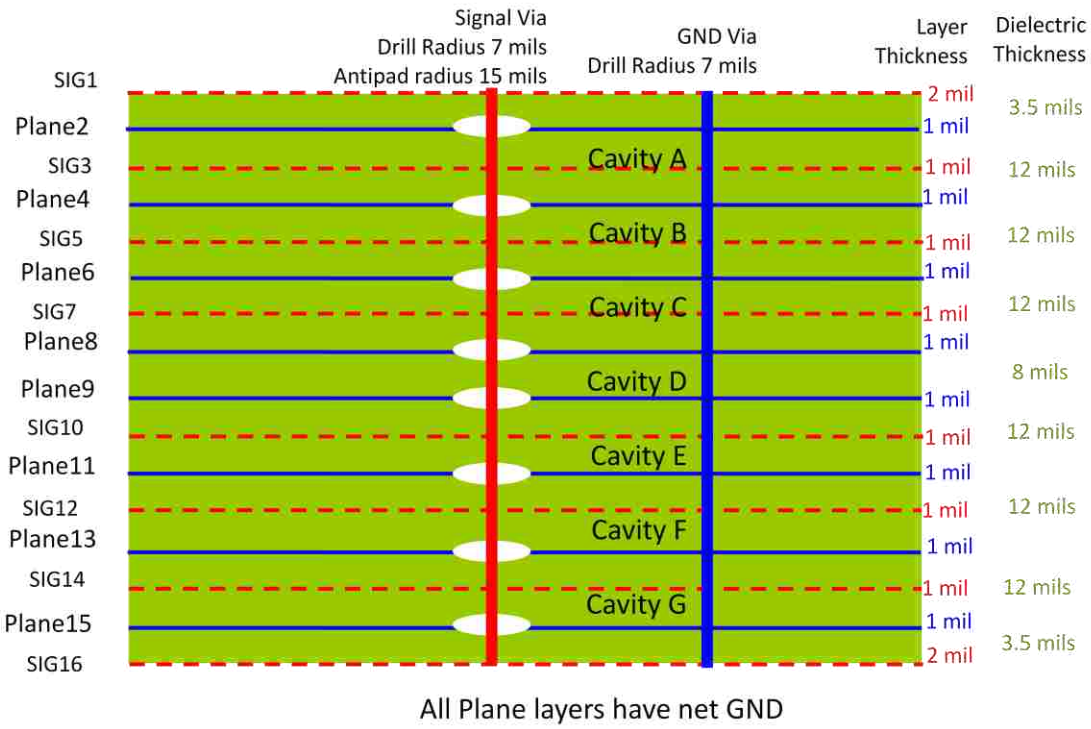


Figure 1.9. Common stack up for all test vehicles is shown. It has eight plane layer forming seven cavities. All cavities are 12 mils thick except center cavity, of 8 mil thickness.

There are 7 cavities, formed by 8 plane layers. Center cavity has 8 mil thickness, and does not contain a signal layer. All other cavities have 12 mil thickness and contain

signal layer for trace connection to the signal vias. All plane layers are ground layers and connected by a shorting via cage, as shown in top view of one of the structures in Figure 1.10. The cage is formed by two rows of vias which are offset by half via pitch, i.e., offset by 20 mils. The signal layers include a trace which connects these shorting via rows. Thus, the shorting via cage forms a PEC boundary around the cavities. The PEC boundary will help isolate individual geometries, and help define the spatial simulation domain for the cavity model.

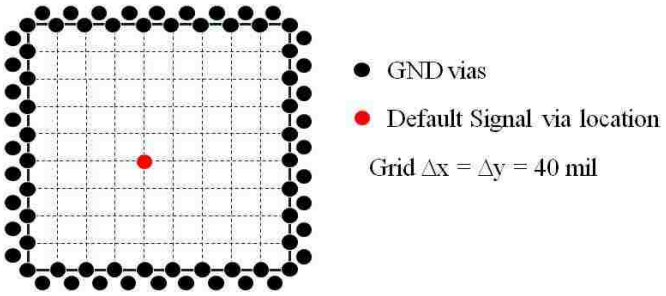


Figure 1.10. A GND via cage is formed by two rows of vias offset by 20 mils. The vias inside the cage lie only the grid locations with pitch 40 mils. The default signal via location is shown by a blue dot.

The via structures under study are placed close to the center of this cavity. There are 5 via structures of interest which will be used for simulation and measurement comparison. These will be referred to as Example 1, Example 2, and so on till Example 5. All vias in the geometry are placed on a grid with pitch 40 mils, as shown in Figure 1.10. The geometry for each case is shown from Figure 1.11 onwards. The Example 1 geometry, shown in Figure 1.11, has one signal via placed close to the center of the cavity, with a trace connection in the top cavity, cavity A and the bottom cavity, cavity B. The measurement is across two ports, from a trace port in cavity A to a trace port in cavity G. The Example 2, geometry shown in Figure 1.12, has one signal via as in the case of Example 1 but, the traces are connected to the signal via in the top cavity, cavity A and the second cavity, cavity B. The measurement is performed at two trace ports in cavity A and cavity B. The signal via, in this case, has a long stub, and its effect is

expected to be seen in the simulation results and measurements. The Example 3 geometry is shown in Figure 1.13. This case is the same as Example 1 with a GND via placed close to the signal via. The Example 3 has the same signal via and traces as the Example 1. The results from measurement and simulations show the effect of the GND via placed close to the signal via when compared to Example 1.

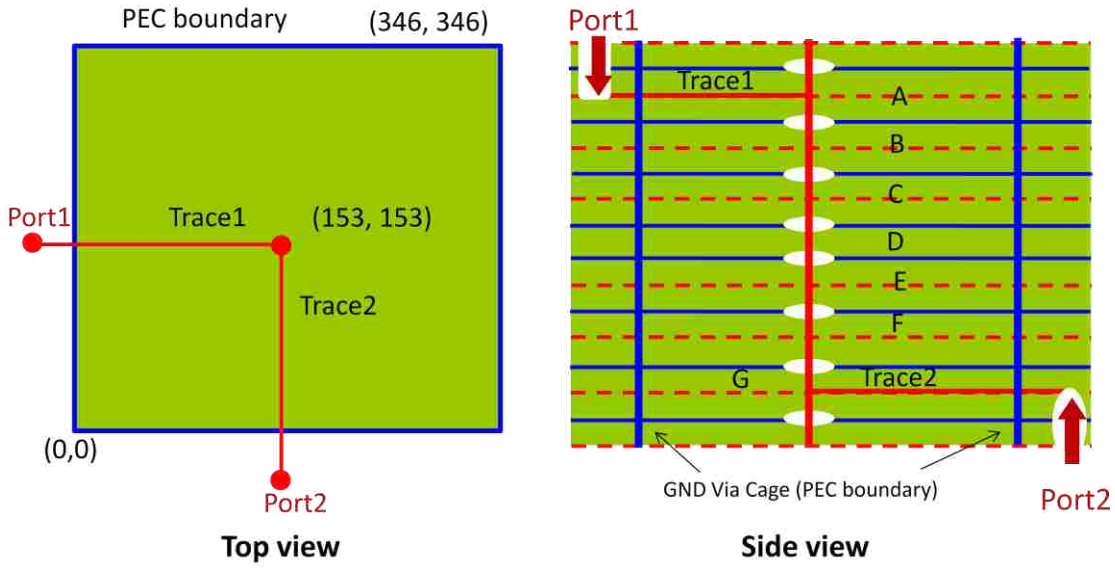


Figure 1.11. Example 1 geometry shows one signal via with trace connection in top and bottom cavity.

The Example 4 and 5 are used for four port measurements. In Example 4, shown in Figure 1.14, has two signal vias, placed near the center of the cavity. The signal vias are connected to a trace each in the top cavity, cavity A and the bottom cavity, cavity G. The measurement is performed at the trace ports in the top and bottom cavities. The measurement will give insertion loss and return loss as in the previous cases, and also the near end cross talk (NEXT) and far end cross talk (FEXT). Example 5 geometry is shown in Figure 1.15. The geometry in this case is the Example 4 geometry with two GND vias placed close to the signal vias. The same measurements as in the case of Example 4 are performed.

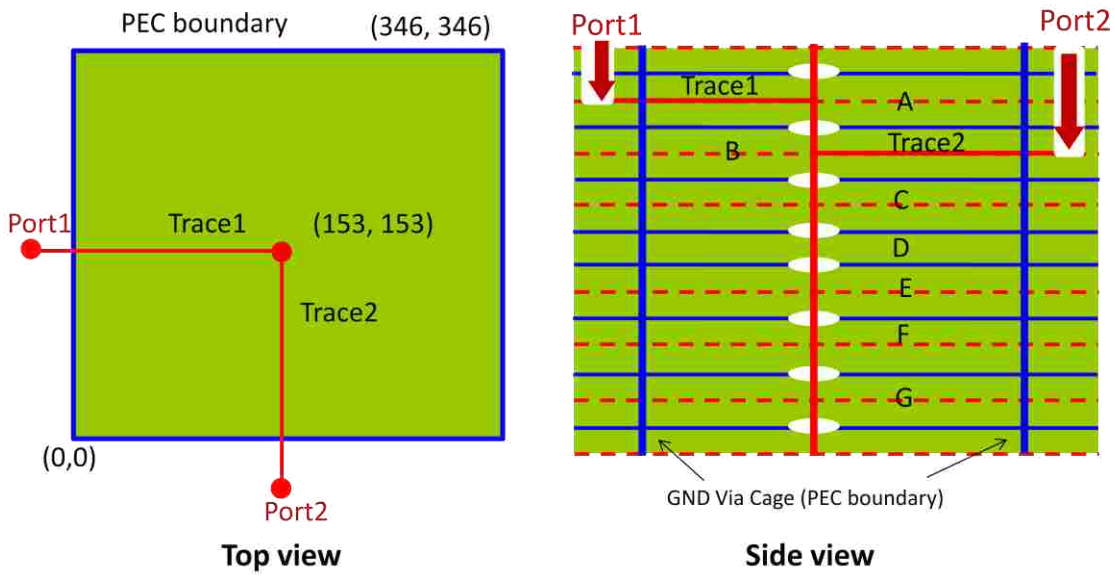


Figure 1.12. Example 2 geometry shows one signal via connected to traces in first and second cavities. This case has a long stub as compared to Example 1.

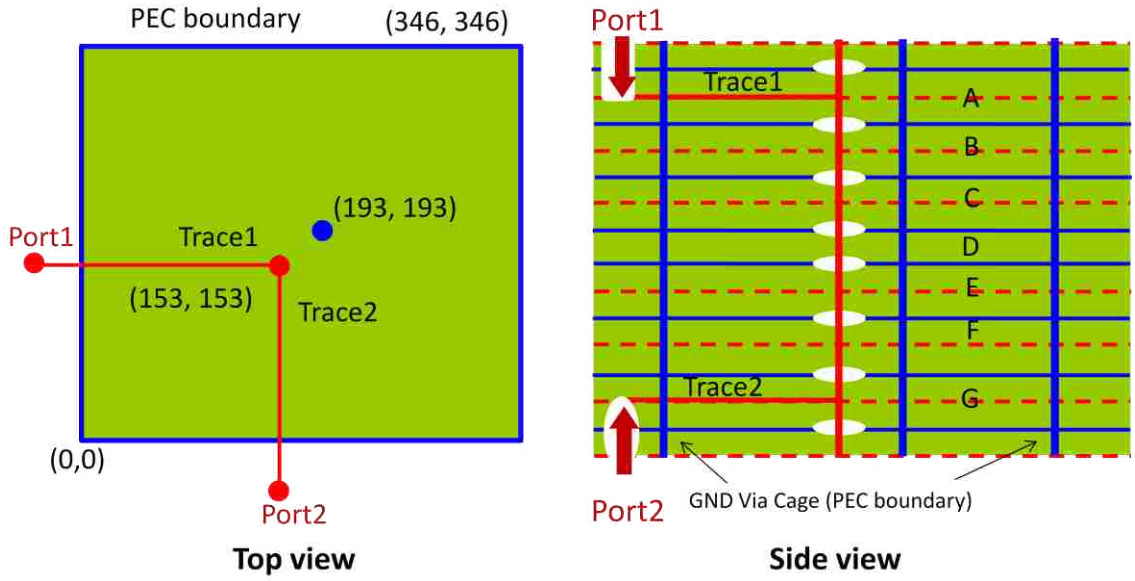


Figure 1.13. Example 3 geometry a signal via and a signal return via location. The signal transition is from top cavity to bottom cavity. This is Example 1 geometry with a close signal return via.

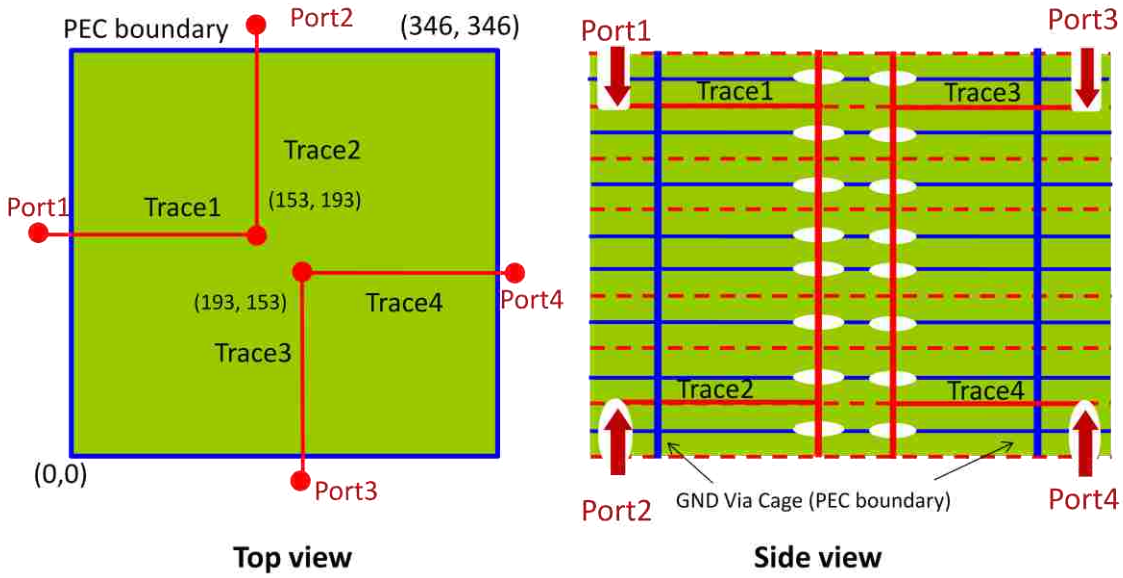


Figure 1.14. Example 4 geometry shows two signal vias with trace connection in top and bottom cavities. The traces are orthogonal in orientation.

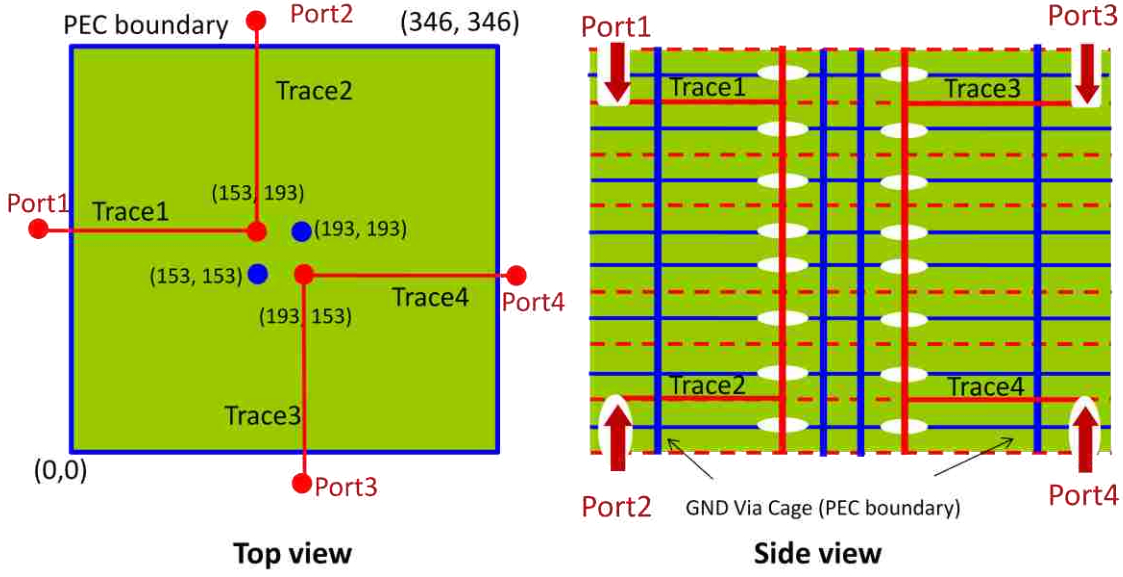


Figure 1.15. Example 5 geometry shows the same geometry as Example 4, with two signal return vias.

The expression for PEC boundary Zpp is similar to the PMC boundary, except the boundary condition that, E field has to be zero along the boundary. The PEC boundary Z_{ij} expression is

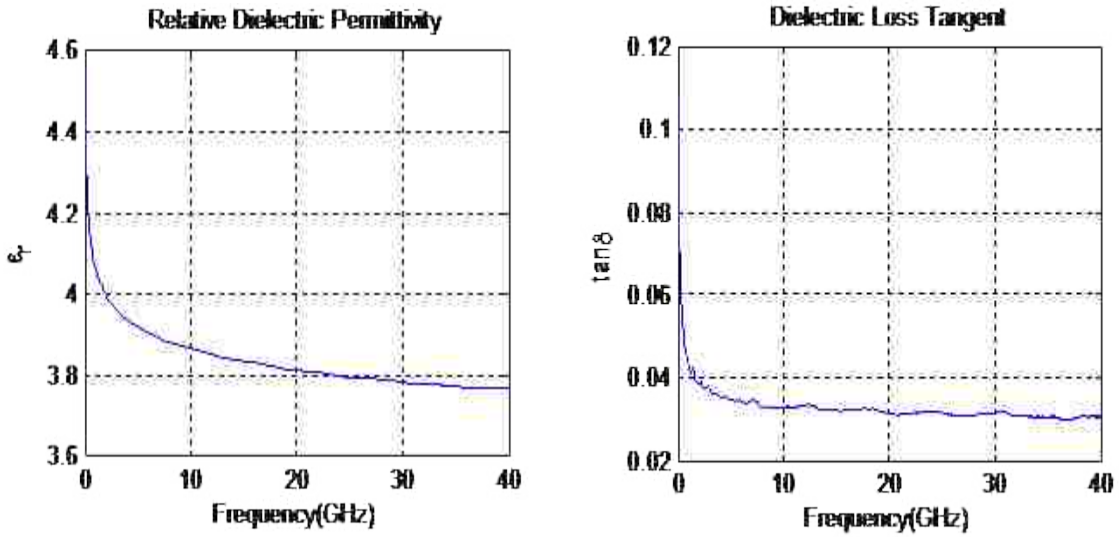
$$Z_{ij} = -j\omega\mu d \frac{1}{ab} \sum_{m=0}^{m=\infty} \sum_{n=0}^{n=\infty} \frac{\left[\begin{array}{l} \sigma_m^2 \sigma_n^2 \sin\ k_x x_i\ \sin\ k_y y_i\ \sin\ k_x x_j\ \sin\ k_y y_j\ . \\ \operatorname{sinc}\left(\frac{k_x W_{xi}}{2}\right) \operatorname{sinc}\left(\frac{k_y W_{yi}}{2}\right) \operatorname{sinc}\left(\frac{k_x W_{xj}}{2}\right) \operatorname{sinc}\left(\frac{k_y W_{yj}}{2}\right) \end{array} \right]}{k^2 - \left[k_x^{\ 2} + k_y^{\ 2} \right]} \quad (1.49)$$

Where,

$$\sigma_m = \begin{cases} 1 & ; m = 0 \\ \sqrt{2} & ; m \neq 0 \end{cases}$$

$$\sigma_n = \begin{cases} 1 & ; n = 0 \\ \sqrt{2} & ; n \neq 0 \end{cases} .$$

EZPP 118 with PEC boundary is used for Zpp calculation for all models. The boundary is placed at the inner edge of the GND via cage. The dielectric properties of the material were extracted from the measurements by Giuseppe Selli, are plotted in Figure 1.16. These frequency dependent values of relative permittivity and loss tangent are used as input parameters to EZpp during the calculation. The Z parameters for the cavity ports for Example 1 through Example 5 are shown in Figure 1.17. The Zpp is expected to start with an inductive 20dB per decade slope as the boundary is PEC boundary. The Z_{11} for the Example 1 and Example 3 show the effect of a shorting via, close to observation port. This is seen on comparing the parts (a) and (b) of the Figure 1.17. The inductance at low frequency is reduced when a GND via is placed near the observation port. Example 4 geometry has no GND vias placed close to the signal vias. In Example 5 two GND vias are placed next the signal vias. The same effect is seen when comparing the Zpp in parts (c) and (d) of the Figure 1.17. The self inductance and the transfer inductance at low frequency are both reduced when the two GND vias are placed close to the signal vias.

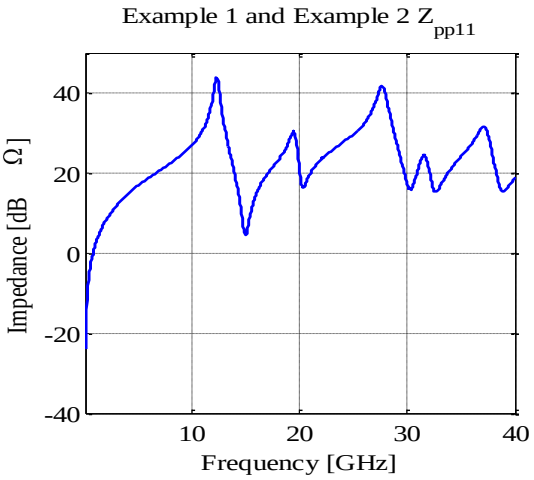


(a) Dielectric Permittivity (b) Dielectric Loss Tangent

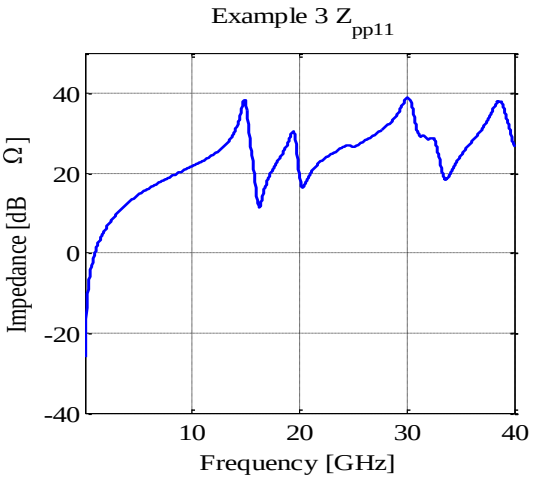
Figure 1.16. Dielectric properties of the test vehicles extracted from measurements.

The capacitance values for the model are calculated with CST EM studio and Hanfeng Wang's 2D FEM code. The two sets of values are very close and the simulation results were found to be the same for both sets. The capacitors from adjacent cavities are lumped, so only three values of capacitors are required. The capacitors connecting the cavity terminal to via at the top plane and bottom plane have capacitance, due to barrel, pads, and top and bottom stubs, equal to 104fF. The capacitance for the centre cavity port terminals to the via, which is smallest due to the shorter center cavity height, is 89fF. The capacitors connecting the via to all other cavity port terminals is 99fF.

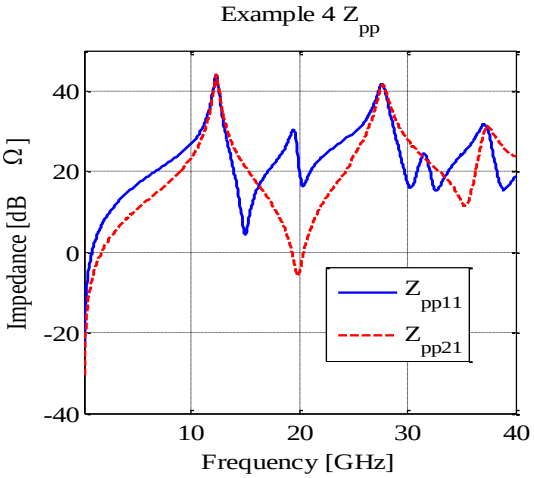
ADS is used for connecting the model together in a network fashion. The circuit model used for each example will change according to geometry being modeled. The Zpp is converted into S parameters, and used as S parameter blocks in the ADS network. As both terminals are needed, a differential design kit is used in ADS to access both the terminals of each port in Spp (derived from Zpp) network block. The model for Example 1 and Example 2 are shown in Figure 1.18 and Figure 1.19, respectively. The Zpp blocks used here have a single port as there is just one via. The model for Example 3 is shown in Figure 1.20, which has the Zpp blocks with two ports. In the model, the second via is a GND via which is modeled by shorting both terminals of that via port.



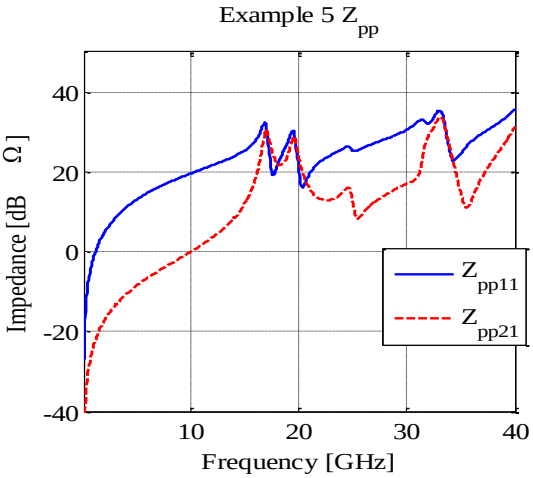
(a) Example 1and Example2



(c) Example 3



(c) Example 4



(d) Example 5

Figure 1.17. Z parameters for Zpp of cavity used in (a)Example 1 and Example 2, (b) Example 3, (c)Example 4, and (d) Example 5. As the cavity boundary is PEC, the low frequency behavior in inductive. The effect of close shorting via is seen as reduction in low frequency inductance from Example 1 to Example 3 and also from Example 4 to Example 5.

The difference between the Example 1 and the Example 2 models is the shifting of trace connection from last cavity to the second cavity. This is in direct correspondence to the geometry change between the two cases.

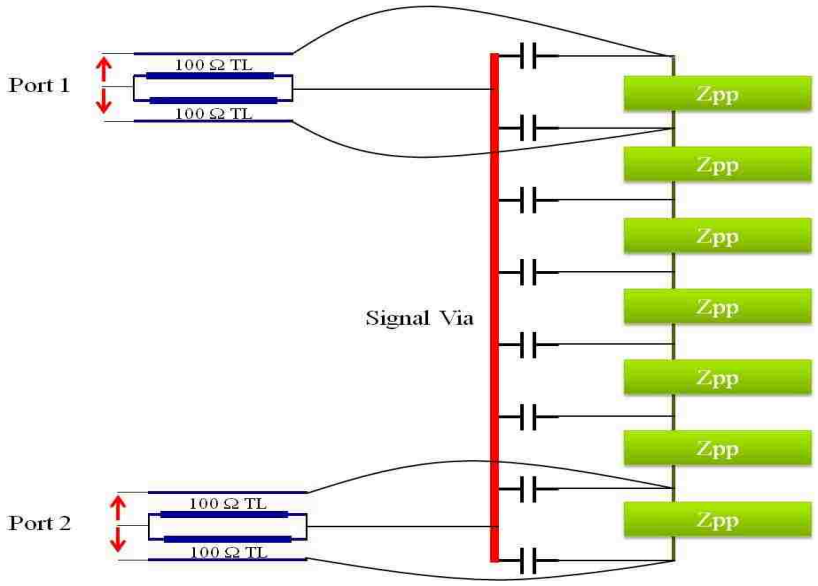


Figure 1.18. The circuit model for Example 1 shows Zpp blocks with one port, connected to the via. The traces, implemented using two inverted 100 Ω microstrips, in top and bottom cavities are connected in the Ad hoc fashion.

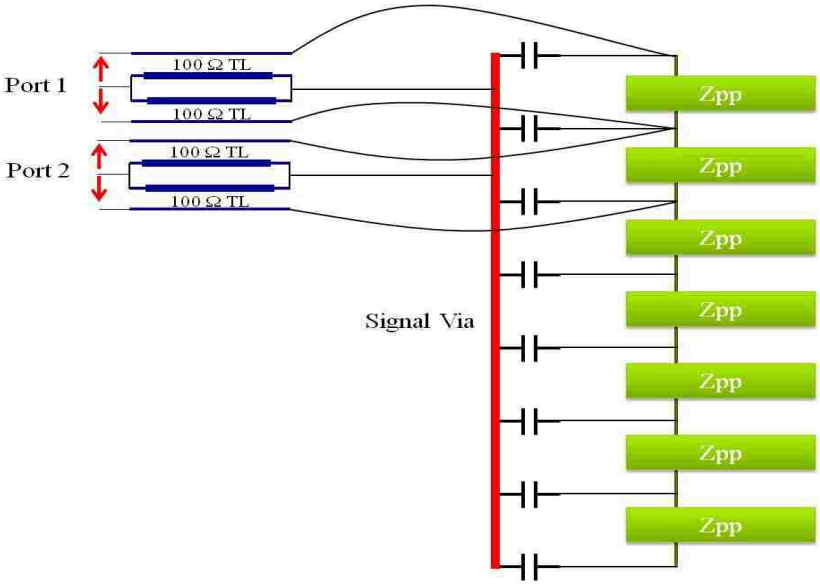


Figure 1.19. The circuit model for Example 2 shown is similar to Example 1 circuit except, the trace from bottom cavity is moved up to the second cavity. Like Example 1, the Zpp blocks have one port connected to the signal via and the traces implemented using two inverted 100 Ω microstrips.

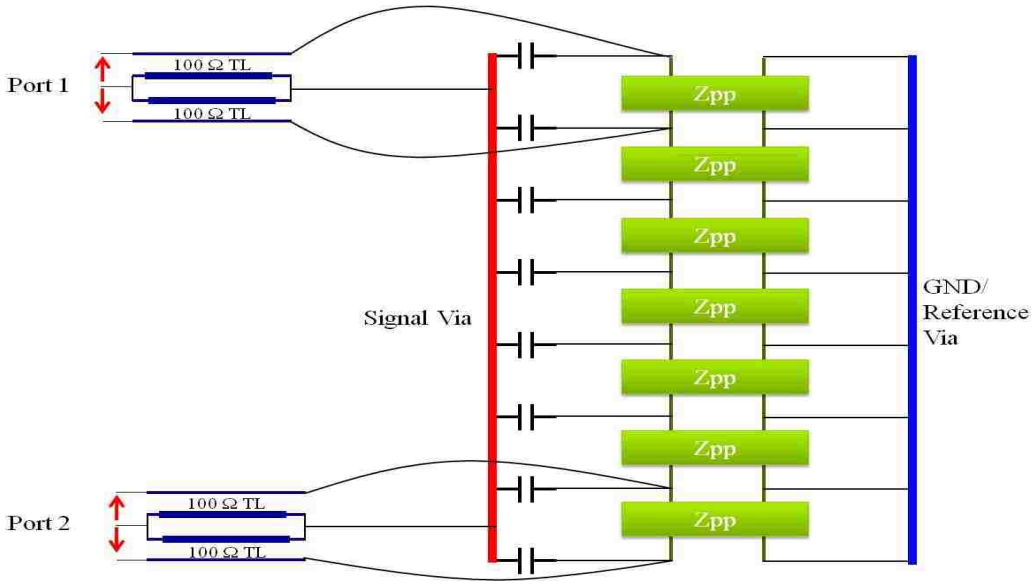


Figure 1.20. The model for Example 3, shown, is the Example 1 circuit model, with the Zpp blocks with two ports. For Example 3, a ground via connected to a second Zpp port.

The Example 4 model, shown in Figure 1.21, also uses a two port network parameter block, but both vias are signal vias and are connected to the traces in the top and bottom cavities. The Figure 1.22 shows the model for Example 5, which has two signal and two GND vias, with the 4 port S parameter block. The port terminals for shorting vias or GND vias are connected together. Signal vias are connected in the same manner as in the Example 4.

The model also contains traces which are transmission line sections used from ADS component library. The stripline model and its connection were implemented by Giuseppe Selli. The stripline is modeled as two inverted microstrip transmission lines. Two ports of microstrips have their positive terminals shorted together, which is the trace terminal, and is connected to the signal via. The reference terminals of the microstrips are the same as the planes of Zpp and so these are connected to respective plane terminals of network parameter block. The signal via has an anti-pad with both planes, so the via is also connected to the planes with capacitors as explained in previous sub-sections. The microstrip traces have an impedance of 100Ω, so in effect they model the 50Ω trace. Each microstrip is given the same AC source so as to excite the two traces symmetrically.

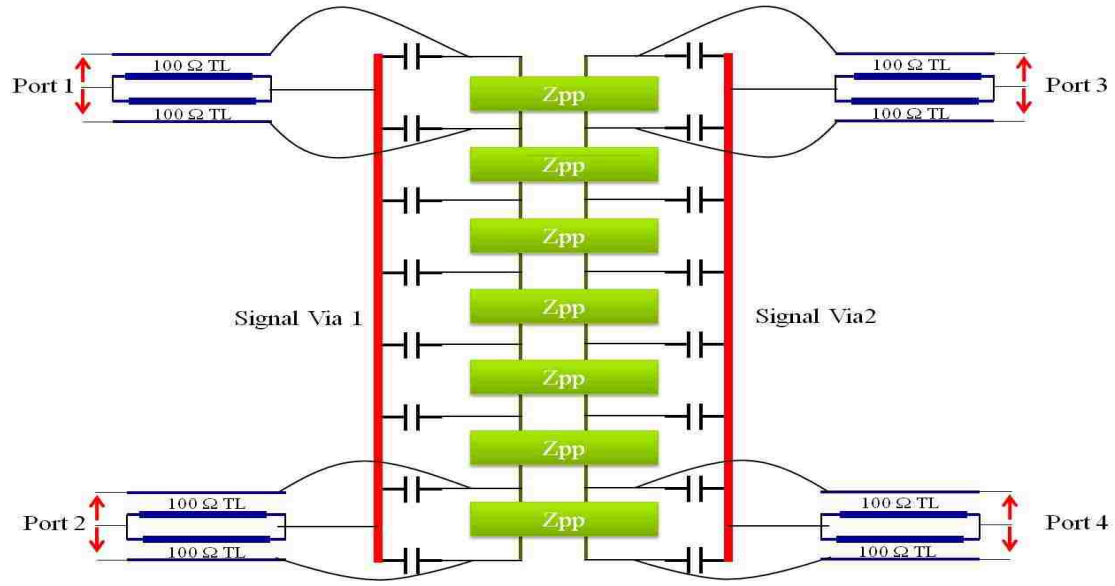


Figure 1.21. The circuit model for Example 4 shows Zpp blocks with two ports, connected to two signal vias. Traces are connected to both the signal vias at top and bottom cavities and, implemented in a similar manner as Example1.

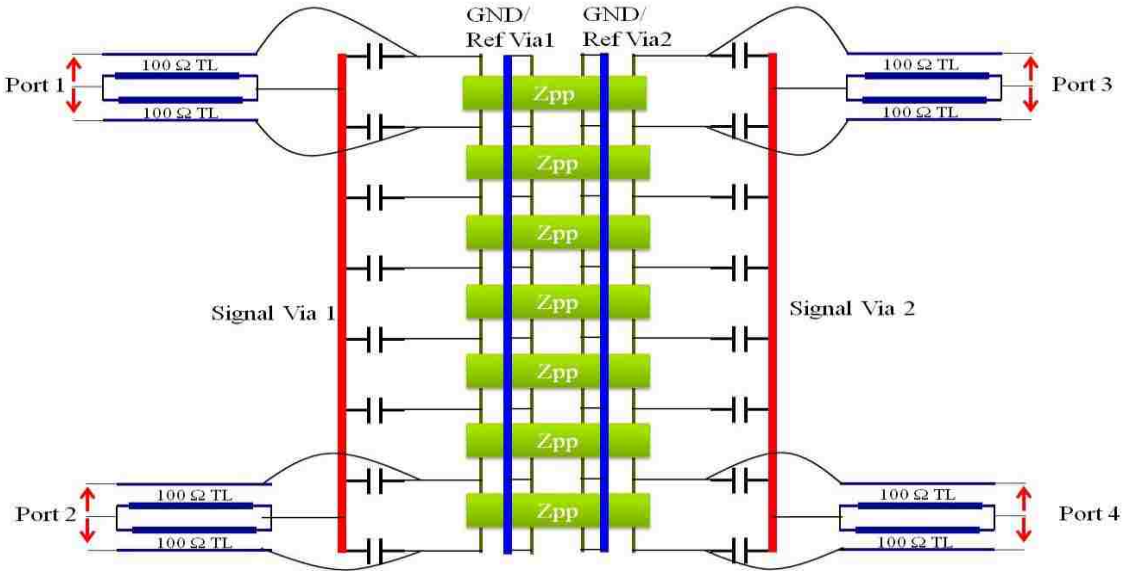
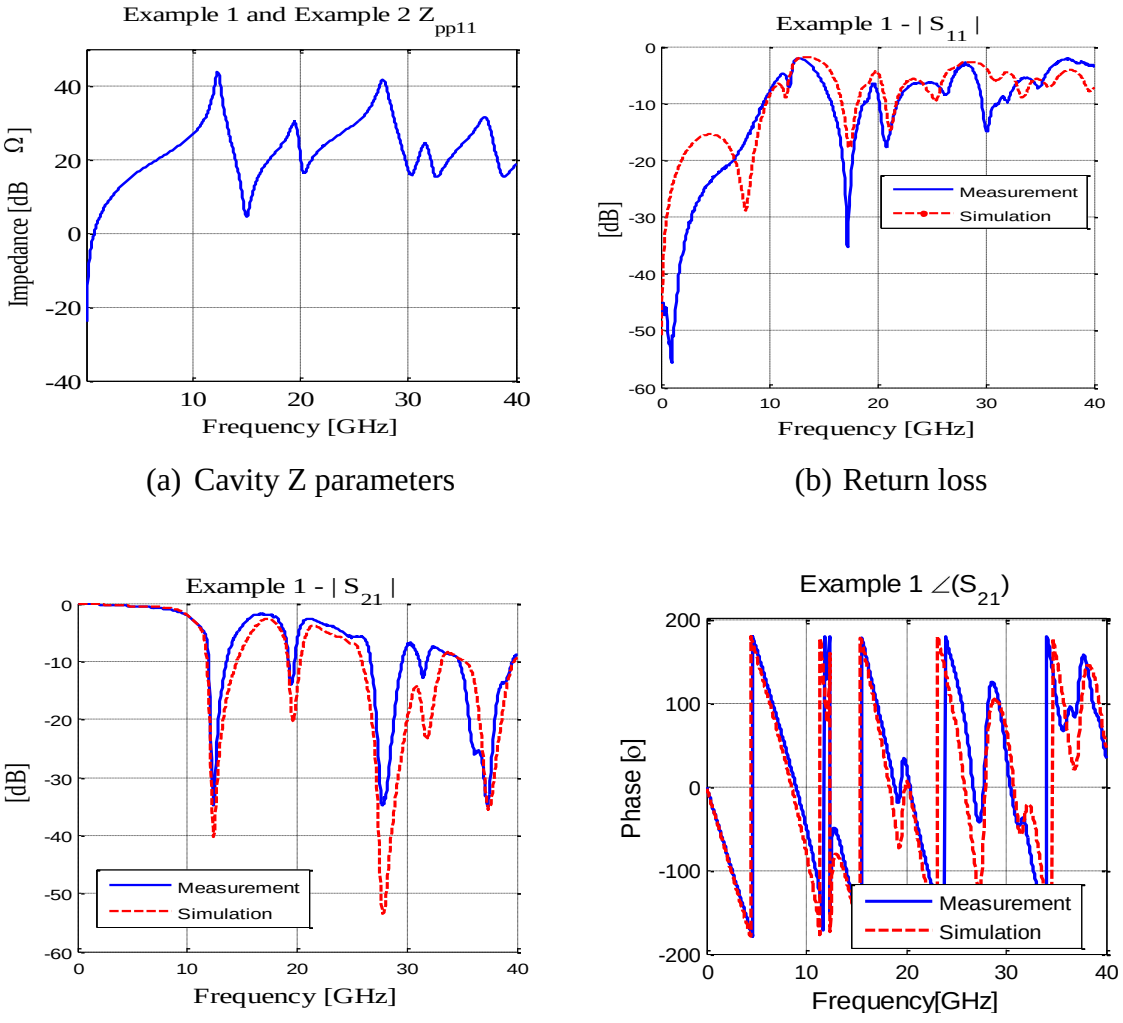


Figure 1.22. The model for Example 5, shown, has the Example 4 model, with the four port Zpp blocks, where two ports are used to connect to ground vias, and two ports are connected to signal vias as in Example 4.

The insertion loss for the Example 1, shown in Figure 1.23, and the Example 3, shown in Figure 1.25, match relatively well compared to the Example 2, shown in Figure 1.24. The return loss comparison has not been successful, for low frequencies. It is observed that the impedance of the cavity, shows poles at which the cavity will offer a very high impedance. From the models in Figure 1.18, the Zpp occurs in series for the return current path of the via current.



(a) Cavity Z parameters

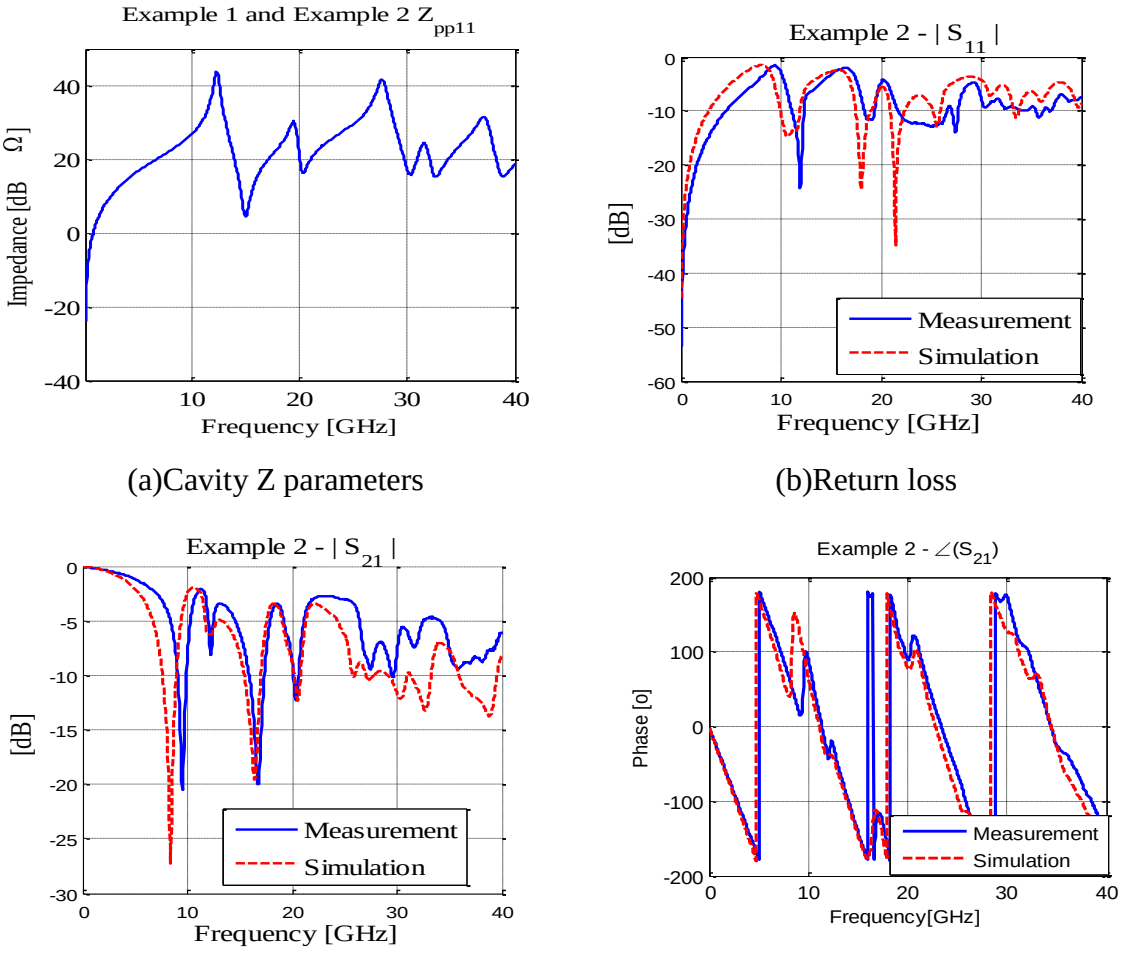
(b) Return loss

(c) Insertion Loss

(d) Phase(S_{21})

Figure 1.23. Example 1 results from measurement and simulation are shown: (a)Zpp from cavity model,(b)Return loss, (c)Insertion loss (d)Phase of S21. It shows the dependence of return loss peaks, and insertion loss nulls on the corresponding poles of cavity Zpp.

At the frequencies where a pole occurs in cavity impedance, the signal will see high impedance in the return path, causing a loss in the transmitted signal from one port to another. This is observed by the corresponding nulls in insertion loss. Also, the return loss shows peaks at these frequencies, indicating the reflection of the signal from the high impedance path.

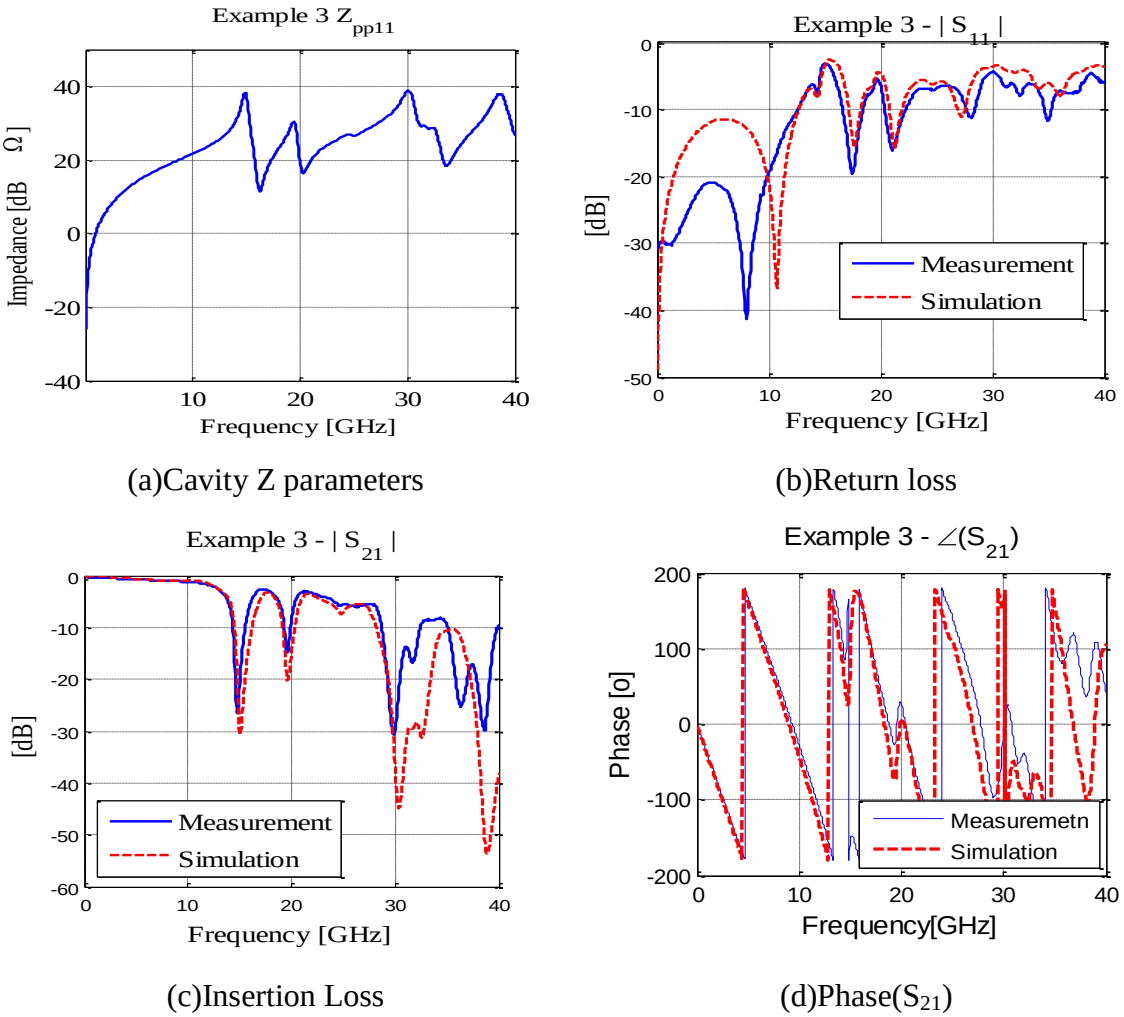


(a)Cavity Z parameters (b)Return loss

(c)Insertion Loss (d)Phase(S_{21})

Figure 1.24. Example 2 results from measurement and simulation are shown: (a)Zpp from cavity model,(b)Return loss, (c)Insertion loss (d)Phase of S21. A large stub resonance and less effect of Zpp on results, as compared to Example 1 can be observed. This can be explained by Example 2 geometry having a large stub and signal transition from first cavity to second only, as compared to no stub and top to bottom cavity transition in Example 1.

The measurement and the simulation results agree well at this resonance and confirm the physical reasoning. The Example 2 simulation results and measurements show a very strong null in insertion loss and corresponding peak in return loss, which occurs due to the via stub from present in the geometry. The observation that the poles of cavity impedance cause nulls in the insertion loss, holds true for Example 2. The effect of cavity impedance has reduced in this case, as there are less number of cavities in series path from one port to the other.



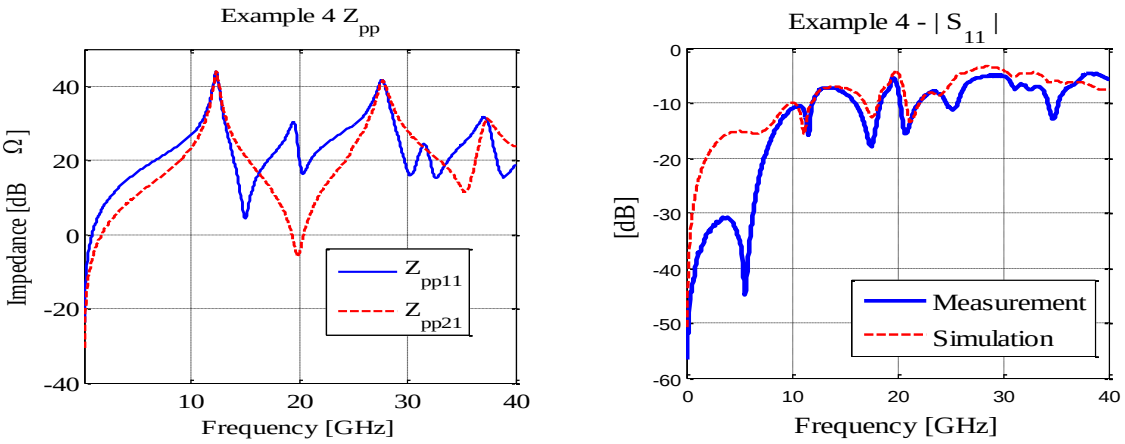
(a)Cavity Z parameters (b)Return loss

(c)Insertion Loss (d)Phase(S_{21})

Figure 1.25. Example 3 results from measurement and simulation are shown: (a)Zpp from cavity model,(b)Return loss, (c)Insertion loss (d)Phase of S_{21}. The Example 3 is the same as the Example 1 geometry with a ground via, and this is seen in the reduced low frequency inductance in Zpp and improved bandwidth seen in the insertion loss, as compared to Example 1 results.

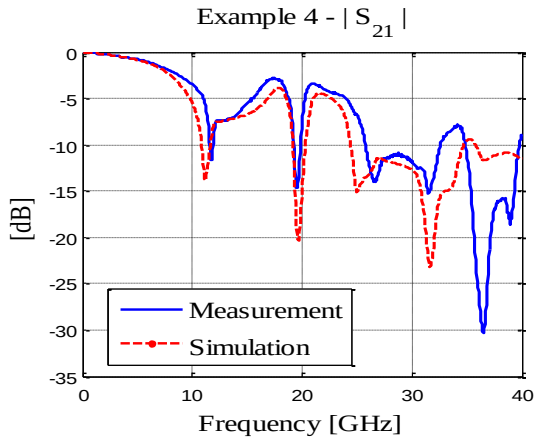
The Example 3 can be compared to the Example 1 as the only difference is the return via placed close the signal via. The Example 1, cavity impedance shows an inductive region in low frequency. This is caused by the conduction current in the cavity travelling along the planes to the PEC boundary, and then returning on the other plane. The return current is conductive till the frequency reaches the first pole in cavity impedance, past which the return current is displacement current through cavity capacitance. When a shorting via is added close to the signal via, majority of the conduction current will use the shorting via as return current path instead of the PEC boundary, thus reducing the return current inductance and pushing the pole to a higher frequency. In terms of link path, this change will increase the bandwidth of the channel, as evident from the results in Figure 1.25.

The Example 4 and Example 5 are four port networks and the comparison results for modeling and measurements are shown in Figure 1.26 and Figure 1.27 respectively. The return loss and insertion loss for Example 4 and Example5 will be similar to the Example 1 and Example 3 results. The same trend of improvement in the insertion loss is seen from Example 4 to Example 5 due to the change in the return current path.

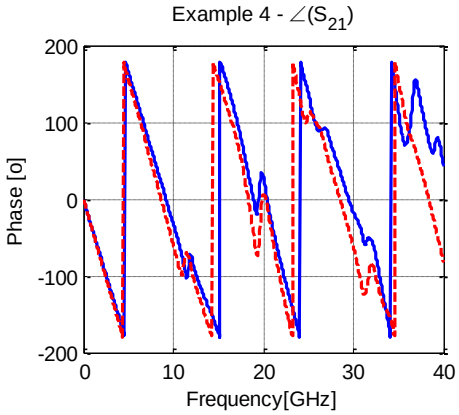


(a)Cavity Z parameters (b)Return loss

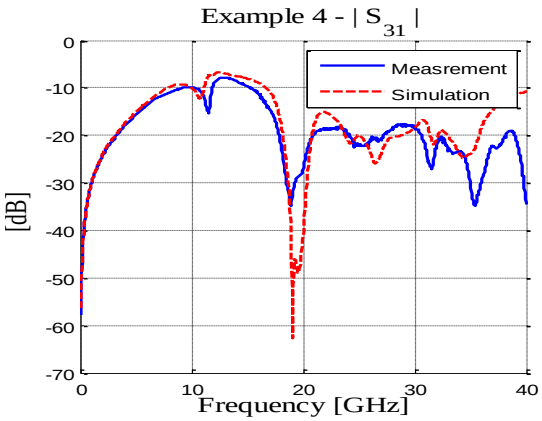
Figure 1.26. Example 4 results from measurement and simulation are shown: (a)Zpp from cavity model,(b)Return loss, (c)Insertion loss, (d)Phase of S21, (e)NEXT, (f)FEXT. The insertion and return loss are dependent on the cavity Z_{pp11}, and the NEXT and FEXT are more dependent on cavity Z_{pp21}.



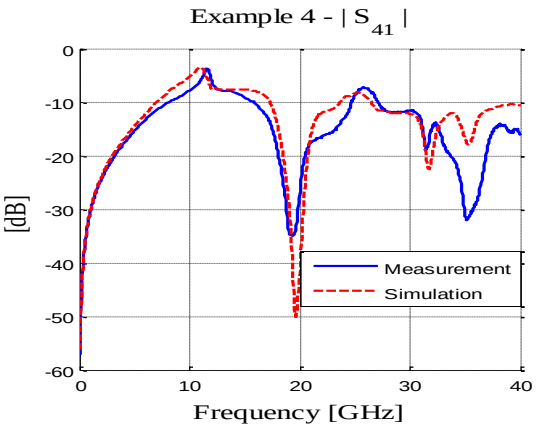
(c)Insertion Loss |S21|



(d)Phase(S_{21})



(e)NEXT (S_{31})



(f)FEXT (S_{41})

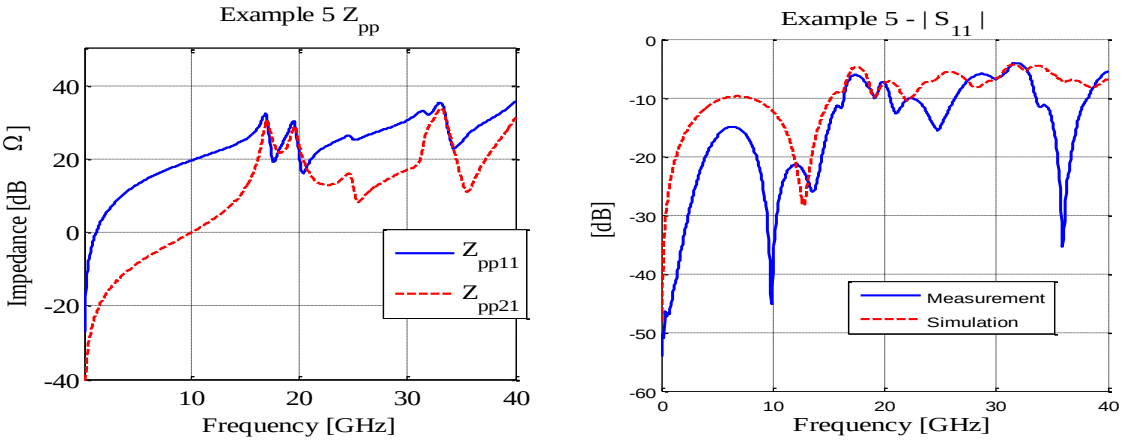
Figure 1.26. Example 4 results from measurement and simulation are shown (cont.)

In this case, the improvement is seen easily as there are two close return current vias instead of one. The effect of self impedance of cavity port at the via location, on the return loss and insertion loss is due to the same physics as explained for Example 1 and Example 3.

The geometry in Example 4 and Example 5 is ideal for studying coupling between vias in the cavity. The traces connecting the vias are routed orthogonally to reduce any coupling through the traces. Hence, the NEXT and FEXT will depend on the mutual coupling of the signal vias inside the cavity. The transfer impedance of the cavity, from port at one signal via to another, will dominate the coupling between the two vias. The

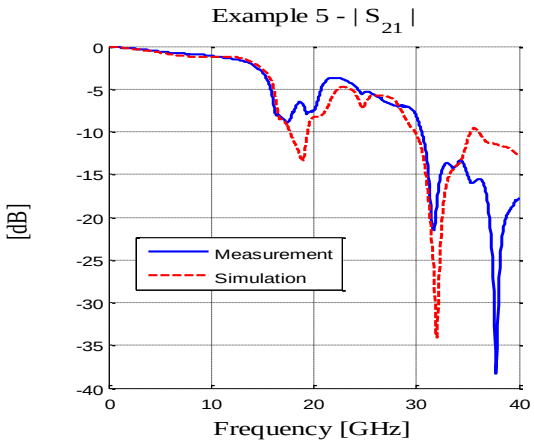
results for the Example 4 show that the zeros in the transfer impedance of the cavity, translate into zeros in the NEXT shown by S_{31} and FEXT shown by S_{41}. For the poles of transfer impedance, there exist poles in FEXT. On the other hand, NEXT is seen not to depend completely on the transfer impedance, as the poles are not reproduced here. Then as two signal return, or shorting vias are added close to the signal vias, for the Example 5, the self and transfer impedance changes drastically, and so the effect is seen in NEXT and FEXT also

The results of the simulation and measurement show similar trends for all cases. Some differences in return loss are caused for all cases due to the measurement techniques or physical phenomena not accounted for, in this particular modeling technique. The agreement for most cases is proof that the model had accounted for all the relevant physics involved in the via transition.

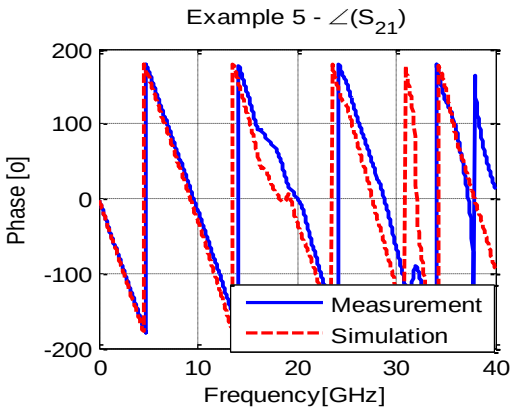


(a)Cavity Z parameters (b)Return loss

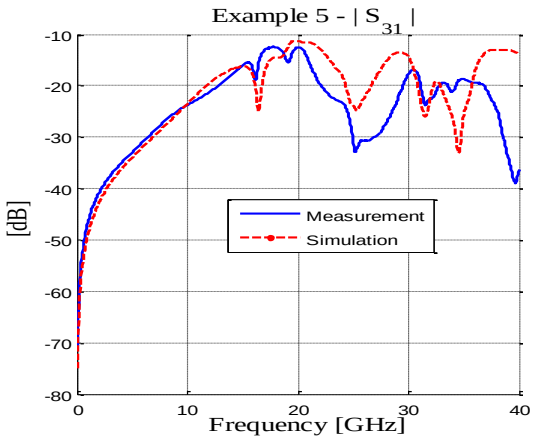
Figure 1.27. Example 5 results from measurement and simulation are shown: (a)Zpp from cavity model,(b)Return loss, (c)Insertion loss, (d)Phase of S21, (e)NEXT, (f)FEXT. The Example 5 geometry is same as the Example 4 geometry with two close return vias. Thus the improvement in bandwidth and reduced influence of Zpp can be observed when compared to the Example 4 results.



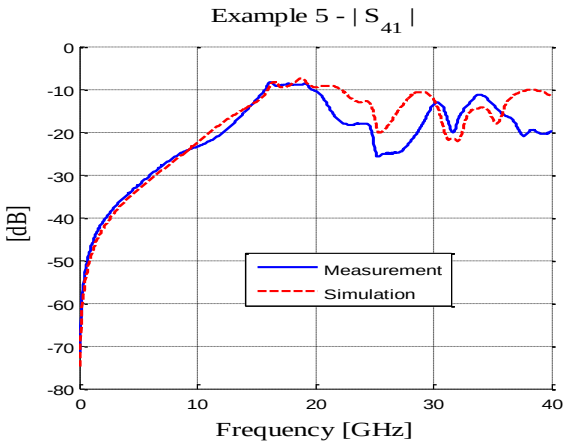
(c)Insertion Loss |S21|



(d)Phase(S_{21})



(e)NEXT (S_{31})



(f)FEXT (S_{41})

Figure 1.27. Example 5 results from measurement and simulation are shown (cont.)

1.6. NEED FOR A TOOL

The comparison of measurements with modeling results shows that the model can be used to obtain network parameters for complex geometries present on the PCBs. The method is quicker than a full wave model and gives a good comparison to measurement. Thus, it can be used for checking a design before having it built and measured. An accurate model for via and cavities, will find a variety of applications. But the complex calculations and assembly in a spice like software will require some familiarization with the process.

To make the via model, available to the design engineers, as a fast calculator or a quick utility to check their designs, a tool called Multilayer Via Transition Tool (MVTT) has been development at the EMC laboratory at Missouri S&T, by Zhenwei Yu and Dr Jun Fan. The tool can read a text file with all geometry information and user requirements. It then calculates the impedance matrix for each cavity, with all the required set of internal and boundary ports. Then the anti-pad capacitance is calculated and vias are connected based on netname information from the text file. The tool then uses voltage and current continuity at the ports to snap together the entire model. At this stage, it is able to connect multi-conductor transmission lines and external components to the model. Thus the whole process of modeling multilayered PCBs with multiple vias, has been automated.

The MVTT has reduced the burden on the user to perform each step of the modeling process. The user has to provide the geometry needed to be modeled. The Section 2 will explain in detail about a graphic user interface of this tool.

2. MVTT GRAPHIC USER INTERFACE

The Multilayer Via Transition Tool (MVTT) is a quick computational utility which can be used to analyze a complex PCB geometry very fast as compared to other tools present in the industry today. The speed and accuracy of the tool are two very important features which give it an advantage over other similar tools. From the user point of view, the time invested in building the geometry in the tool and effectiveness with which the user utilizes the computational ability of the tool are also very important and directly dependent on the design and functionality of the User Interface (UI). Thus the development of the UI becomes very important from the point of view of application and widespread utility of the tool. This section explains the process of designing the graphic user interface (GUI) for the MVTT in Matlab. This section deals only with design and is not intended to serve as a user manual. For this purpose, the 'Quick Start Guide' for the MVTT GUI is placed in the appendix.

2.1. INSIGHT INTO GUI DEVELOPMENT FOR MVTT

As explained in Section 1.6, the MVTT is a fast calculator of network parameters, for a PCB geometry. The MVTT is by itself a design aid, which can replace the full-wave tools for the application with some limitations. To be as successful, the MVTT has to go through an engineered tool development. An integral part of this tool development is User Interface (UI) development, as the usefulness of this tool depends to a great extent on the convenience with which a user can communicate with this tool. There are various options to simplify the communication process. Some tools used command prompt, which is a more primitive user interface. Some tools still use text files, which are mostly the SPICE based tools. But the most user friendly one is the Graphic User Interface (GUI).

The MVTT engine (also referred to as the 'kernel' which is the main computational part of the tool) was designed to work with a text file input. Thus, the engine is designed to directly run with an input file, without a GUI. The text file is, however, difficult to build from scratch for a new user and thus a GUI would make it more convenient for the user to run the tool. Thus, at this stage, a simple GUI was

required, which could easily interpret the user inputs and output a text file, which could be used with the engine. But, a simple GUI would mean it may not support all the features of the engine. Thus, the features deemed too difficult to implement in the basic GUI, would still be accessible through the text file interface, which was considered to be convenient for an advanced user.

The MVTT development was initiated in Matlab. For the computational part, or engine, using the Matlab programming environment is the obvious choice with the numerous scientific toolboxes present in the coding environment. Matlab has a Graphic User Interface Design Environment (GUIDE), a Java based utility, which is helpful in building simple GUIs. It provides some convenience in creating a GUI but does not offer complete control to the programmer. As the engine is Matlab based, having a Matlab based GUI can help overcome any interface issues. Thus, Matlab was chosen to start the GUI development.

For the MVTT, all the inputs are required up front before any computation can be performed. Then the results are available to the user after the engine finishes its calculations. These results are available for viewing in the GUI. Thus, the tool flow could be represented as shown in Figure 2.1. First the geometry definition is completed by the user. Here the user interacts with the GUI alone. Then in the next step, the GUI generates a text file with the user inputs which is passed on to the engine. The engine then computes the desired results, and outputs the touchstone network parameter files. Then, these results are made available to the user through the GUI. This flow also allows a simple text file interface to the engine.

2.2. GUI DESIGN

The most basic aim of the GUI is to accept all the inputs (requirements) from the user and then pass them over to engine. This set of requirements can then be arranged into a flow which helps to simplify the process of accepting inputs. To obtain these set of requirements, it is essential to understand the projected functionality of the engine. An understanding of the engine would present a scope of the expected user inputs and the fine tuning required by the user to better control the results from the engine. Figure 2.2 shows the functionality of by the Multilayer via transition tool.

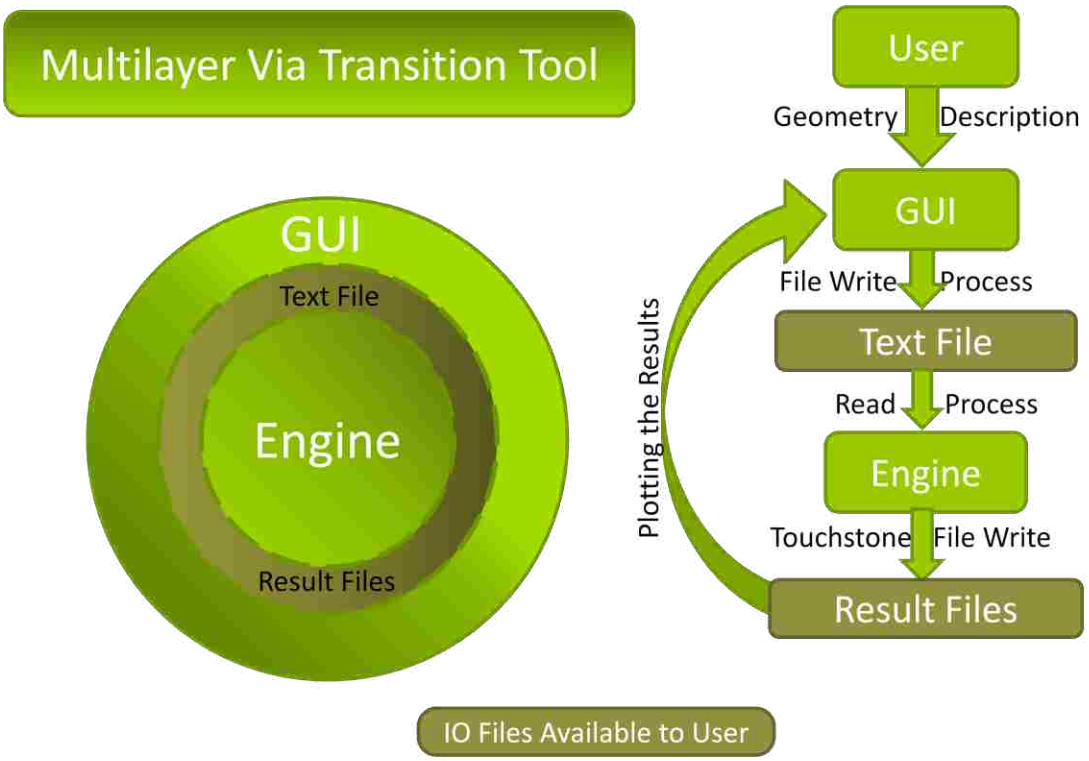


Figure 2.1. Flow used by the MVTT is shown. The tool has an engine and a GUI developed such that they can be independently used. The user provides the geometry description to the GUI, which then generates a text file in a format read by the engine. After processing, the engine writes the network parameter file in touchstone format which user can view using the GUI.

The GUI for such a tool, as MVTT, can become very complicated due to the vast scope of inputs and with the different features of the engine, available to tweak the simulation. The GUI is required to be organized in a manner that the user finds it intuitive while navigating through it. For this reason the organization of the GUI is very close to the physical geometry features and processes involved in the PCB manufacturing.

The inputs can be grouped into sets based on geometry features present in the PCB, which the user wants to simulate, as well as, ports and settings. The grouping of inputs required by the tool is,

- Geometry
    - Stackup
    - Padstack

- Vias
- Traces
- Components

- Port Definitions
- Simulation settings
- Advanced Settings

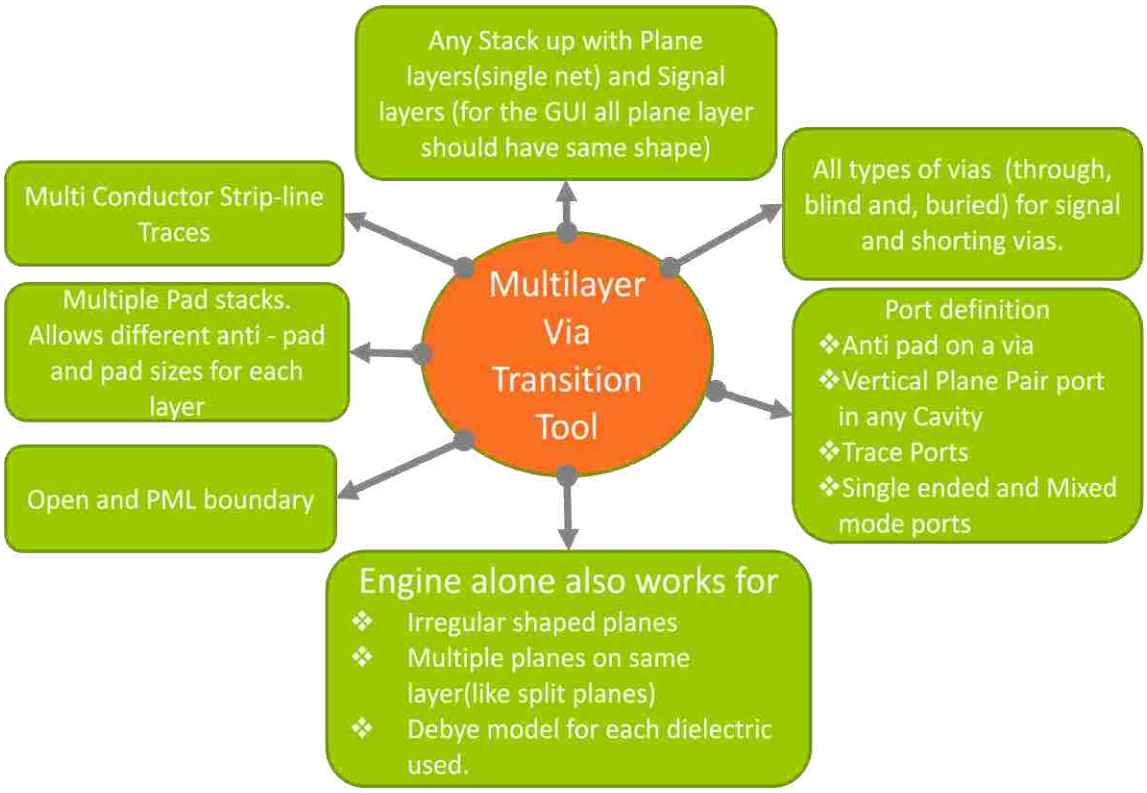


Figure 2.2. Functionality of the MVTT is shown. The tool can process the listed geometries from a PCB. Results are calculated for the different kinds of user defined ports as shown. It allows for simulation with Open or PML boundary. The engine features, not implemented in the GUI are also listed.

The GUI can be segregated into a sequence of screens, each with a set of inputs, which the user has to fill. A main window or governing screen is provided which helps the user keep track of the current status, in the process of completing the entire geometry

description. The GUI would then be able to guide the user intuitively through the entire process, including simulations settings and advanced settings. Figure 2.3 shows an ideal flow that the GUI would require the user to follow when setting up a simulation from scratch.

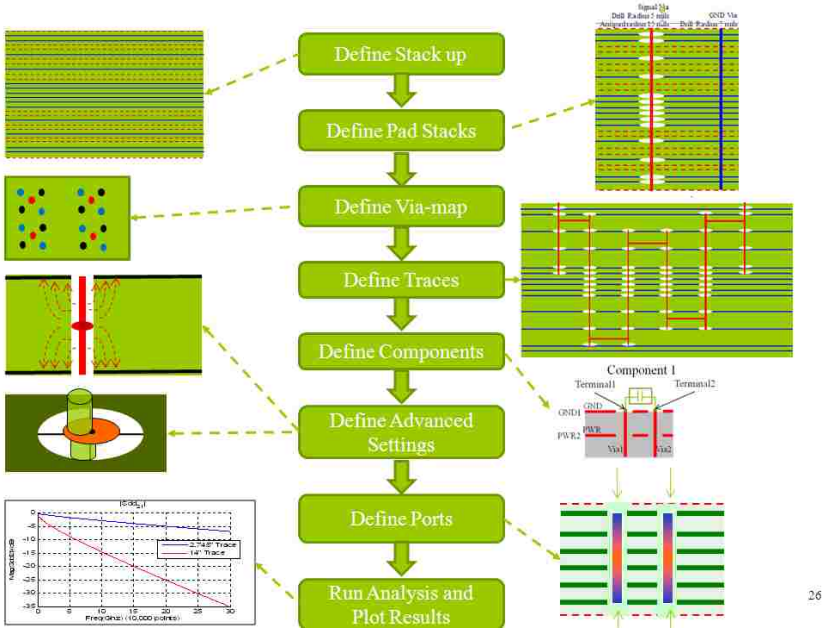


Figure 2.3. Ideal flow of the GUI.

Although these inputs (which are reasonably interdependent) can be handled one set at a time, the central idea is facilitating user convenience by relating the input sets together. For example, when the stack up is defined, each layer gets classified as a signal or a plane layer and also gets a name. Later when defining the traces, the user gets to choose the trace layer location from a set of signal layers already defined and named in the stack up definition window. This flow allows for a defined procedure to use the GUI from geometry description to viewing results.

The following part of this subsection is used to describe the organization used for each screen. The text would also describe how each screen will be simplified with default values, images, and other aids which are used for clarity, speed up, and/or add convenience to this process.

2.2.1. The Main Window. Figure 2.4 shows the blank main window which appears when the tool is started afresh. A core function of the main window is to convey the flow of the tool to the user and ensuring that this flow is being followed.

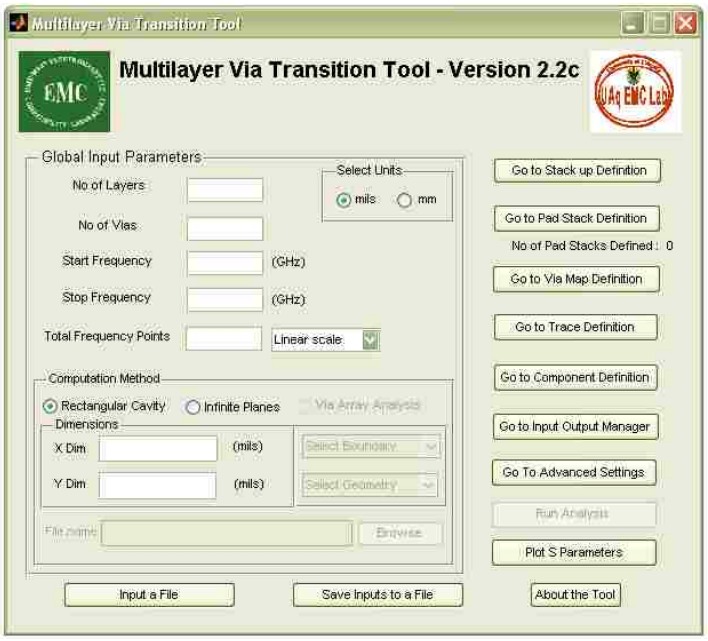


Figure 2.4. Blank 'Main' Window. This window will be used to guide the user through the process of providing all inputs. It is used to access all other windows of the GUI and so is used to control the flow of the GUI.

The 'Main Window' contains buttons which link to all other windows (screens) of the tool. Then, it becomes easier to control which screen the user proceeds to, next, in the flow. There are indicators provided adjacent to each button, to show the user whether all the inputs on that screen are entered. If the user tries to skip a window, then, the requested window is not shown, but the user is reminded about the incomplete window. This is also important since the data from the previous windows is required on consequent ones, so they cannot be displayed otherwise.

Some inputs, classified as simulation settings, are accepted in this window. These are general settings which may be required in any of the linked windows. Also, the

number of layers and number of vias are the two inputs required to display the stack up and via map windows correctly.

2.2.2. Stack Up Window. The 'Stack up Definition' window is accessed by clicking on the 'Go To Stack up' button from the main window. This window is designed to accept all the properties of metal and dielectric layers present in the stack up as required by the engine. The 'Number of layers' input on the main window, will provide number of metal layers present on the PCB, number of dielectric layer is assumed one less than the metal layers. The metal properties accepted are layer type, layer name, a netname and metal thickness. The dielectric properties accepted are dielectric constants, loss tangents, and, thickness of dielectrics. As a provision for future requirements, a disabled button for importing the 'debye model' for the dielectric is provided. This feature is not present in the GUI but is available for the engine use alone.

Figure 2.5 shows a default 'Stack up Definition' window. The window has been organized in a way to show the values in a table fashion but on the image of a PCB itself. The layers get a color depending on their type and dielectrics are shown in green. When

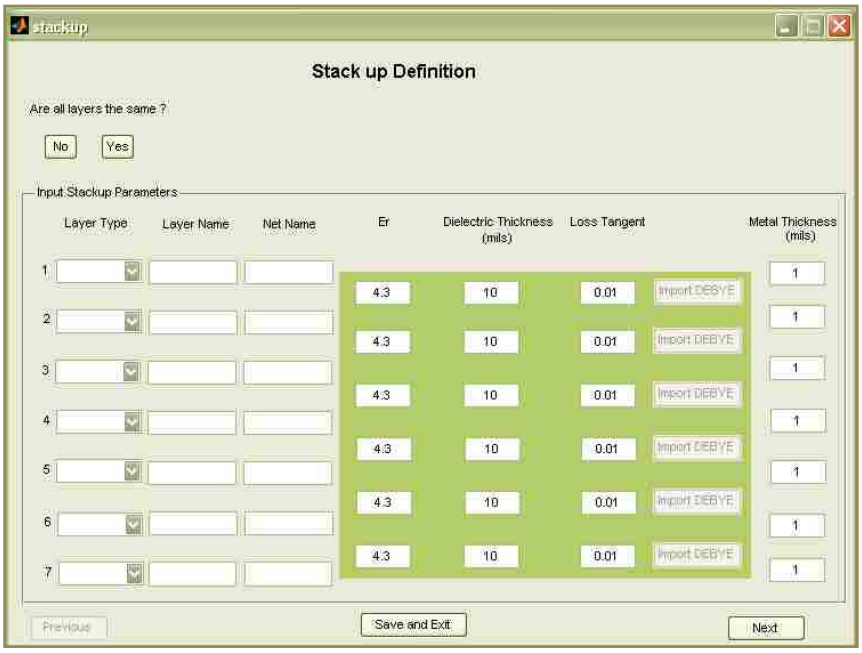


Figure 2.5. Stack Up Definition Window. This window accepts the inputs about the layer thickness and material properties of each metal and dielectric layer.

there are more than seven metal layers, more than one stack up definition windows are used. The overflow is divided in sets of six metal layers until the last layer is reached. Each consequent stack up definition window will have a dielectric layer preceding the metal layer.

Apart from the visual aid, there are default values provided for most properties of the metals and dielectric layers. The metal layers will get default names and net names when their type is selected. The metal layers of type ‘signal’ will not get a net name since they may have multiple nets routed over them. Also the user is allowed to set a default values at the top of the first window, and apply them to all dielectric and metal layers.

At the bottom of the window are three buttons provided, namely ‘Back’, ‘Next’ and ‘Save and Exit’. The ‘Back’ and ‘Next’ are used to navigate through the stack up, if more than seven metal layers are present. When the user is on the last window, the ‘Next’ button changes to ‘Finish’ which signifies that the stack up definition is completed. The ‘Save and Exit’ button is used to save any changes made to the stack up and return to the main window. This feature is necessary if the user does not wish to complete the entire stack up and exit the window as is, or even when the user just wants to change some values in the stack up and exit to main window immediately.

2.2.3. Pad Stack Window. A set of values for the anti-pad diameter, the pad diameter and drill diameter for each layer together form one pad stack. Many such pad stacks can be defined, and referred to by their user-defined unique names. These are used as templates for the vias, such that when vias are being defined in the ‘Via Map Definition’ window, a pad stack is assigned for this via to follow. Figure 2.6 shows the pad stack Window.

Similar to the ‘Stack up Definition’ window, the data is accepted in sets of seven layers from first layer to the last layer with a separate window for each set. Default can be set and applied to all layers on the first window itself. There is a table like format of display for anti-pad diameter and drill diameter, verses the layers. Since the drill diameter remains same for all layers, it is accepted only once for all layers for each pad stack.

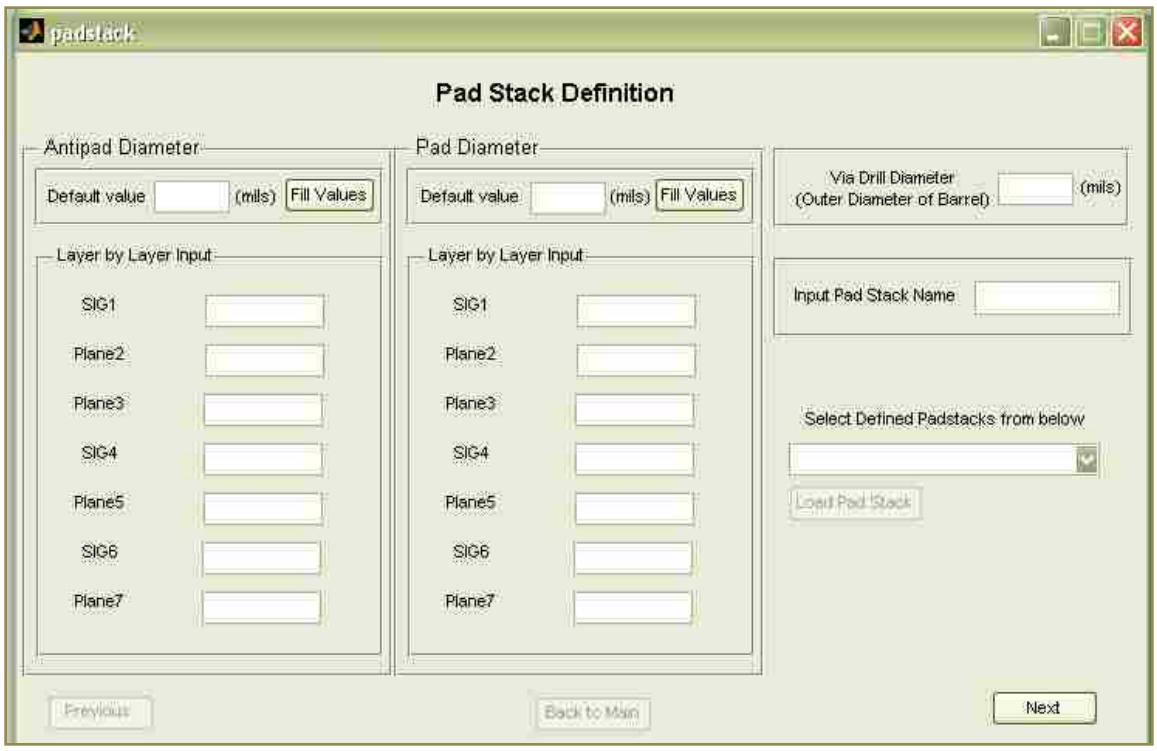


Figure 2.6. Pad Stack Definition Window. It accepts all the dimensions of via related geometry like anitpad, pad and via drill.

Each time the user clicks on the ‘Go to Pad Stack Definition’ button on the main window, a new pad stack definition is initialized. To finish defining this pad stack, the user has to proceed untill the final pad stack window, which will display a ‘Finish’ button, clicking on which completes the definition and returns user to main window. Once the user enters the pad stack window, only one pad stack can be added or a saved one modified. If user needs to modify a saved pad stack, instead of defining a new one, the ‘Load Saved Pad stack’ provides for it. To define several pad stacks, user has to define them one at a time, clicking on ‘Go to Pad Stack Definition’ every time a new one pad stack is to be added.

2.2.4. Via Map Window. Figure 2.7 shows the ‘Via Map Definition’ window which pops up when the ‘Go to Via Map Definition’ button is clicked on the main window. This window accepts inputs with regard to via type and location. Again, the table like representation of information is used with a property input along each row, and every column represents a via. To have a unique handle to each via for later reference, a

name is assigned automatically to each via. The default name assigned has the format ‘VIA<via index>’. For instance, the first via is ‘VIA1’. The via index is the number assigned to the via which is displayed in the top row of this window.

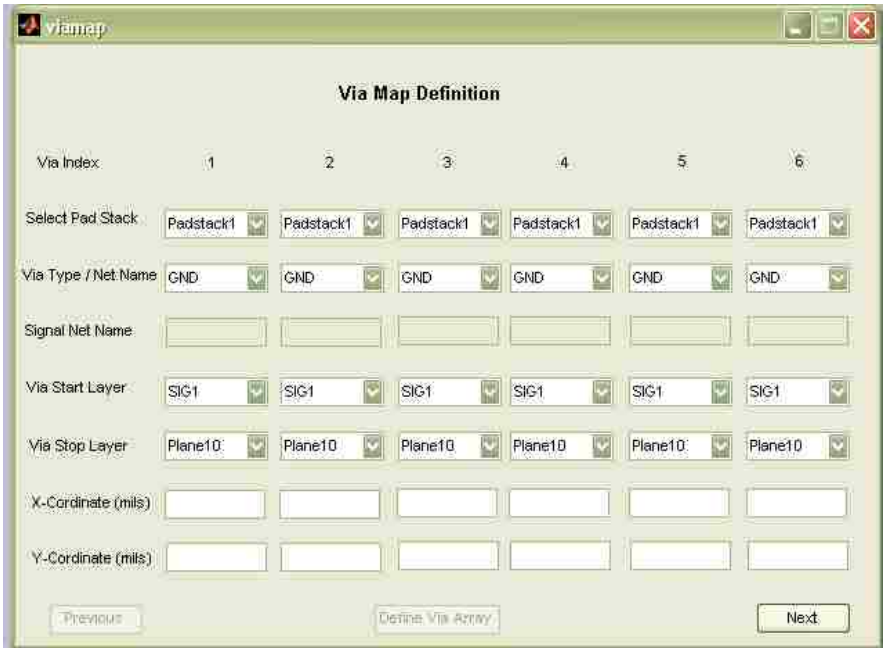


Figure 2.7. Via Map Definition Window. This window accepts the specifics for each via, like type, net name, pad stack and location.

The dimensions for the via in terms of drill diameter, pad diameter and anti-pad diameter for each layer, are through the pad stacks specified in the previous window. These are referenced by the names assigned by the user, chosen in the second row. The vertical extent of the via is represented by the via start layer and the via stop layer, which is relevant when vias are buried or blind. The default for the start layer for via is the first layer and stop layer is the last layer. The layers being referred to are metal layers available from the ‘Stack up Definition’ window. The restriction, however, is that the via has to start or stop at a plane layer. This comes from the model used by the engine where the geometry is segmented into cavities at the plane layers. The last two rows, require the via location in terms of its x and y coordinates.

The via connection to plane layers or no connection (anti-pad), is interpreted by the tool based on the net-name specified for the via. The net-name is specified in the third and fourth rows of this window. If the via is connected to a plane layer, it has the same net-name as the fill on that plane. The via type chosen in third row is the plane net name referenced from the stack up definition. If the via does not connect to any of the plane layers, then it is assumed to be a signal via (‘sig’ is chosen in via type), and the unique net-name is accepted in fourth row.

2.2.5. Trace Definition Window. The ‘Trace Definition’ window, shown in Figure 2.8, is used to define the transmission lines present in the geometry. The flow of this window is controlled to facilitate the user in providing the required information about the trace. ‘Trace’ is interpreted as a multi-conductor transmission line, by the tool. Each ‘Trace’ may have a number of coupled stripline or microstrip transmission lines which are referred to individually as ‘Strips’. The strips forming the multi-conductor ‘Trace’ must lie on the same signal layer. The trace control panel at the top left allows the user to navigate through traces already defined, or to add new traces.

To start with the definition of a new Trace, the user is required to click on the ‘Add New Trace’ button. The window now requires the user to name the trace and fix the number of strips forming the trace. The signal layer on which all these strips lie is also chosen here. The signal layers are referenced from the stack up definition. Once this information is filled, the user may click on the ‘Proceed’ button which leads to the next set of inputs. The GUI is designed to accept the multi-conductor transmission line definition as S parameter, per unit length RLGC parameters, or with the cross section geometry. The current version only accepts the definition in terms of S parameters. The number of strips input is used to select a touchstone file which defines the trace, with the required number of ports. The strip definition section allows user to define each strip individually. The inputs accepted here are strip names, strip net names, strip dimensions, start and stop coordinates, port mapping with said touchstone file. Each strip can be defined and saved individually, and a different strip is selected by clicking on the name in the list provided.

For user convenience, default names for trace and strip are provided, but can be changed. In strip definitions, net name is chosen from a list of signal net names referenced from via map, thus reducing user burden of entering same net name.

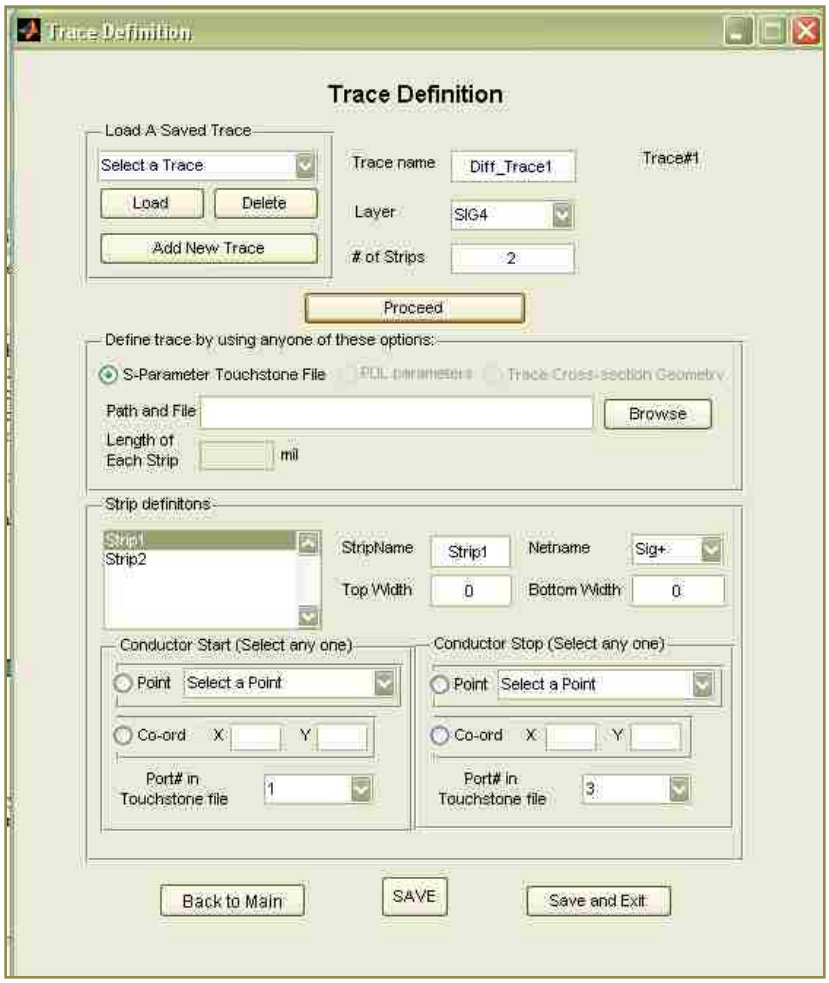


Figure 2.8. The Trace Definition Window. This window allows the user to define a mutli-conductor transmission line, providing all inputs as per the requirement of the engine. The image shows a trace with two strips defined by S parameters. The user would have to specify the start and stop of each strip and port mapping to the touchstone file.

Other inputs like port mapping and start stop locations are facilitated by drop down menus which allow the user to choose from a set of possible values. Strip start/stop, usually, at a via location, which is one option and otherwise at a new point, which is

updated to the ‘available points’ set immediately. The objective has been to reduce user burden to recall coordinates and make errors in repeating data entry. ‘Save and Exit’ and ‘Back to Main’ button are used to return to the main window after having saved the changes and without saving respectively. To save changes and still remain on the same window, the ‘Save’ button can be used.

2.2.6. Component Definition. The Component definition window is similar to ‘Trace Definition’ window, in its flow control and layout. The Component control panel will allow the user to navigate through the defined components, add new components, or delete components. The components are classified as RLC series, RLC parallel, and S parameter block. The lumped element components are allowed to connect to the geometry by defining two terminals, whereas the S parameter block may have any number of ports which may be left connected to geometry, kept open or used as a new port in the results. Figure 2.9 shows the component window with a new component definition.

When the user adds a new component, it is named, and the type is chosen. Accordingly, if it is the RLC series or the RLC parallel type component, then the user can proceed to define the component values and the connection. The R, L and C values will be used in the model only if they are enabled in the window. The connection type available is ‘Connect to PCB’ only. If the component is the S parameter block type, then the user would have to provide the touchstone file and then define the connection of each port. S parameter blocks need the port connection type specified for each port and then the port connection information is specified, if required. The port terminal connection is not required if the port has to be left open, or if it is to be used as an IO port present in final tool results.

The connection when selected as ‘Connect to PCB’, a two terminal connection to PCB has to be defined. A drop down menu is available for each terminal which lists any via terminals at top or bottom plane layers with anti pads around them. The net name of the top or bottom layer may be selected if the terminal connects to the plane directly.

User is allowed to change the ‘component type’ input when required and save the changes. The ‘Done’ button is used to save a new component or save changes to previously saved components. The ‘Back to Main’ button is used to exit to the main

window as is. The component definitions, left incomplete, may be finished later by accessing the window again.

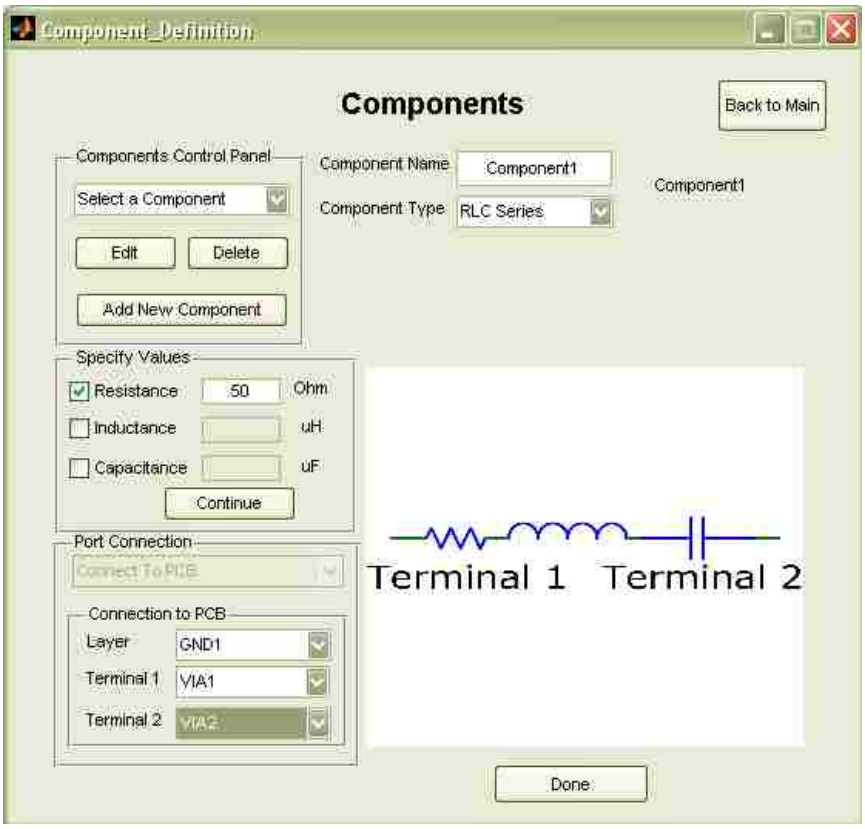


Figure 2.9. Component Window which shows a 50 Ohm resistor definition between the Via 1 and Via 2 at the GND1 layer.

2.2.7. IO Manager Window. The IO Manager is accessed after the entire geometry has been defined in the tool. The formulations explained in section 1 will be used to get network parameters between the user defined ports. This window is designed to help define each port, as required by the engine. The 'Input Output Manager' window is shown in Figure 2.10, as it looks when defining a single ended 'Via port'.

The user is allowed to define three basic types of ports, namely single ended, differential and planepair port. The single ended port can be a via port or a trace port. The via port is defined across the anti-pad of a via, where the engine can obtain well defined current and voltage relations. The trace port is a port defined at the end of a transmission

line. The ‘Differential port’ is in reality a mixed mode port, defined from two single ended ports (both via ports or both trace ports). The planepair port is a vertical probe placed in any cavity is used with the purpose of quantifying the energy coupling to a plane pair.

The user gets to define each port in an intuitive manner. As in the previous two windows, the flow is well controlled by the GUI. The window contains a Port control panel, used to add a new port, navigate through previously defined ports, or delete a port.

Figure 2.10. The Input Output Manager window, showing the definition of a single ended via port using the via and the plane name.

When adding a new port the GUI follows basic steps: name and select port type, select port geometry type and then the particular geometry for port. For planepair port, the flow is slightly different. In this case, the user selects the port type as ‘Planepair port’, then chooses two consecutive planes, and enters the coordinate information for the port location.

The single ended via port needs the user to select a via and then a reference plane at which the via has an anti-pad. In the case of a differential via port, the user selects two vias and then a reference layer at which both vias have anti-pads. Similarly for the trace port the user selects a trace, then a strip in that trace, and then the start/stop of the strip whichever is open and the reference plane for the port. For the differential trace port, user will select two strips of a trace, then their start/stop locations with a reference plane for the port.

As the entire geometry has been defined with a name for each feature, it is very convenient for the user to refer to each via, trace, strip, or plane with their user defined names available in the drop down lists. The flow helps user choose a port which is present on the geometry, and if no port can be defined with a geometry chosen, then the user gets an error message before saving the port. The 'Done' button at the end of the port definition will also perform an error check to avoid the repetition of ports. The 'Back to Main' button will allow the user to exit at any stage. There is a user convenience feature accessed through the 'Arrange ports' button, that is used to change the order of ports, if required. A new window pops up, where the ports can be re-ordered and, if saved, this order is reflected in the port control panel when the control returns to the 'Input Output Manger' window. The user also selects the network parameter output type required from the tool by selecting S or Z or both. A touchstone file of selected type is outputted by the engine, when the analysis is complete.

2.2.8. Advanced Settings Window. The 'Advanced Settings' window, an MVTT special feature, is used to tweak the tool settings and improve the simulations. Figure 2.11 shows the 'Advanced Setting' window. This is meant for an advanced user, with a reasonable understanding of the engine which will run the analysis. This window not only has small inputs like port size for planepair ports, and the ports per wavelength setting for cavity model, but also has buttons to access more advanced features like adding via – anti-pad offset for misalignment or adding extra capacitance to account for fringing, pads, etc.

Figure 2.11. Advanced Settings Window. This window provides the user with access to the advanced settings of the tool, like changing capacitance values at an anti-pad, or adding an offset between the via and anti-pad.

2.2.9. Run Analysis. The user may click the 'Run Analysis' button on the main window, after the geometry description and the port definition is complete. The GUI writes a text file compiling all the inputs from the user together. The GUI calls the engine and passes the filename and the path of the input file. The engine reads the input file, begins its computation, and passes the control back to the GUI when the computation is over. The GUI displays a pop-up message for the user indicating that the tool has finished its processing, and the path to the results' directory. The path to results' directory is just a convenience, so user can access them immediately. Figure 2.12 shows an example of a pop-up which occurs when the tool finishes its analysis.

2.2.10. Plot S Parameters. The user can view the results computed by the tool by clicking on the 'Plot S Parameters' button in main window. A window that pops up is S manager window, designed and programmed by Francesco De Paulis as a part of the Link Path Analyzer [8]. The output S parameters (or even Z parameters) can be plotted in this window which is a quick plotting tool for files in the touchstone format. It also has utility

to compare parameters from different files. Figure 2.13 shows the S manager window and a touchstone file plot.

Figure 2.12. Processing complete popup which occurs when engine finishes its processing and writes results into the respective files.

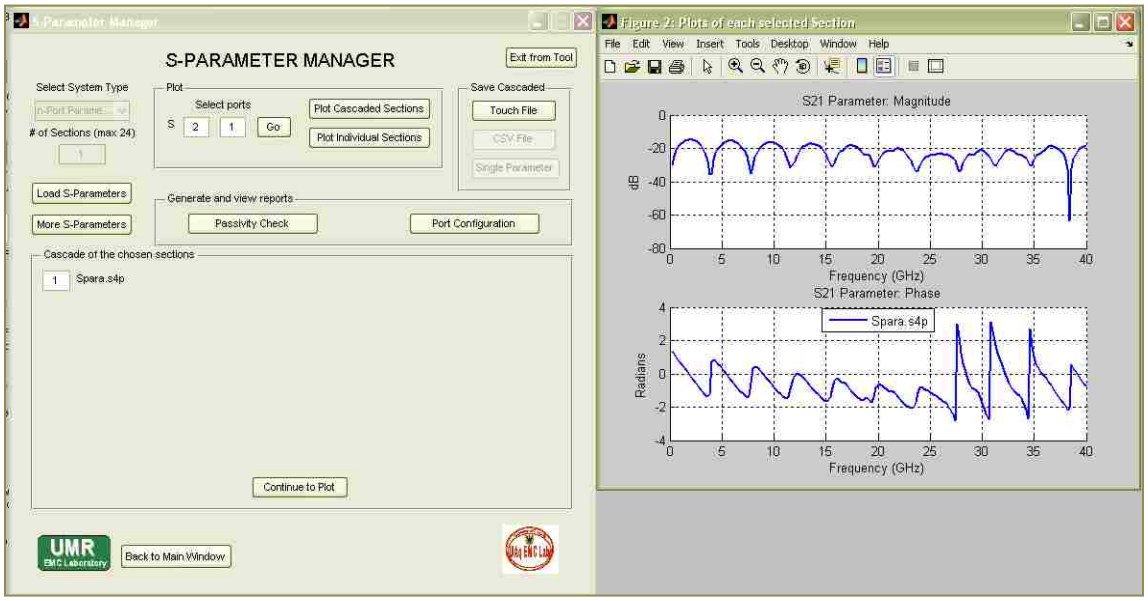


Figure 2.13. S Manager Window used to plot and compare the results which are given as outputs when the tool finishes processing.

2.3. DISCUSSION ABOUT ISSUES IN UI DESIGN AND DEPLOYMENT

The Graphic User Interface development for the MVTT has provided a very good experience in terms of the user interface design and implementation. During the

implementation phase, many issues were faced, which were not anticipated in the design phase. Some of these issues would be discussed in this subsection. One of the major issues throughout the tool development was the error checking. The different types of error checking methods, and the difficulties in implementing them are explained with respect to the MVTT GUI, in the first part of this subsection. The second issue discussed here is a deployment issue specific to Matlab as a tool development language.

2.3.1. Error Checking Necessity And Implementation. The GUI design discussed in the section 2.2, has a very long flow or procedure. During this entire flow, many sets of inputs are accepted from the user and these sets inputs are interrelated with each other. The inputs are related with the user defined names assigned to each geometry feature defined in the tool. This helps the user to setup the geometry with relative ease. If a new input is invalid or if changes made in some inputs, invalidates some other set of related inputs, then this could crash the GUI. If the GUI accepts such invalid inputs and passes them on to the engine, there is a good chance it will crash or give erroneous results. Hence it is essential to check continuously the information provided by the user for its validity and any changes in the information for its effect on the information already entered in the tool.

There are two types of error checks implemented for each window. The first type of error check is with respect to the validity of information provided in the current window, e.g., validity of the numerical values or repetition of the inputs. The second type is where a new information or the changes to any available information are checked for their effect on other dependent geometry features. The second type is relatively very difficult for implementation. Also all references to a particular input are to be accounted for, as the tool will crash in another window due to a change in this window, making it difficult even to debug. Apart from these two error checks a global error check is also provided for checking changes in the main window inputs.

When the user inputs invalid information for an input not provided before, the tool prompts the user and does not store this invalid data. If this occurs when there user is changing previously inputted information, then care has to be taken to not change the previous information, until a valid new input is provided. Also GUI flow has to allow the user to exit the window without saving the changes made in it. The same is also forced

when the changes in inputs in one window affects the information present in other windows. In this case the user may also be given a choice to delete invalidated information from other windows. This has to be preceded with a warning so as to make the changes clear to the user.

Based on the above guidelines, there are several lines of code written for error check in the MVTT GUI. Even so, some situations still occur where the GUI will crash unexpectedly. The code is being improved continuously as these situations arise. A testing pattern is to be designed to check for these situations. The most common difficulty faced is to check for the effect of the changes in the initial geometry inputs to the other geometry features, defined using these previous inputs. For instance, the changes in the layer type in the stack up window will affect the traces or the port definitions. If there are already traces defined with a reference layer or ports defined with a reference layer, then changing that plane layer to signal layer will cause a chain effect of errors in the back end data structures. This has to be taken care of very effectively, as it may lead to a crash as soon as the GUI tries to write the input file for the engine.

2.3.2. Deployment Of Matlab Based Tool. The tool developed in the Matlab programming environment will be sent out for use at a different location. The tool may be used on computers with a different operating system and which may or may not have a Matlab installation. The version of the installed Matlab, if any, also cannot be assumed the same Hence Matlab has a deployment procedure required to be performed to ensure the Matlab based tool will function correctly on any machine. Matlab has a system library called Matlab Compiler Runtime (MCR) which is needed to be installed prior to running the compiled Matlab code on any machine.

The only situation, in which the tool (compiled executable) will run on a user machine without installing the MCR, is if the user machine has exactly the same Matlab version installed on it with the compiler toolbox and all other required Matlab toolboxes installed and licensed, as the programmer's machine. If the resident Matlab version is different, or if the compiler toolbox or any other toolbox is not present, the compiled tool will not be able to run correctly and may crash. Thus, it is always better to install the MCR before using the tool.

The MCR has to be installed on the user machine once and then the tool can be run as many times as necessary. The version of the MCR to be installed is also important. When the programmer compiles the code, Matlab will use the installed compiler toolbox to create an executable file. Corresponding to this compiler, an MCR installer is provided by Matlab or can be extracted from the compiler toolbox. The executable file will work with only that version of MCR pre-installed on the user machine. Thus, MCR version is defined by the compiler toolbox used by programmer to create the executable.

Multiple versions of the MCR installer may be present on a machine on which the tool will be used. As long as the required version of the MCR is installed correctly, this should not cause any problems. Each version of MCR will add a path, pointing to its installation directory, to the system or environment variable, 'path'. If the required MCR is not installed correctly, then the path variable may not be configured for it. In such case an error may occur which would suggest that the MCR '*.dll' files were not found. In this event, reinstallation of that particular MCR installer is needed. Such errors can also occur if the system has a corrupted path variable, in which case reinstallation may not solve the problem, but fixing the path variable manually could help.

To understand the effects of the presence of different versions of Matlab Component Runtime (MCR versions), various testing procedures were followed on the EMC Lab computers. These tests have helped gain insight into the requirements for being able to successfully deploy a tool and then be able to maintain and upgrade it over the long run. These testing procedures included installing various versions of MCR on a single machine (v7.5, v7.6, v7.7 and v7.9). Then various stand alone applications, which required each of these MCR versions installed earlier, were executed. The reasons for errors, if any, were investigated and Matlab – Mathworks support was also invoked to better understanding the procedures involved. The same testing was carried out on several machines with different version of 'Windows' operating system and different Matlab versions to understand machine dependency or the role of local (resident) Matlab. This experiment confirmed that, if the required MCR is installed correctly, and the system has a clean path variable, there would be no issues faced in running the executable files.

2.4. FUTURE SCOPE FOR TOOL DEVELOPMENT

For the Multilayer Via Transition Tool, the engine has many capabilities not utilized by the GUI. For instance, in the GUI all layers have a rectangular shape and same size, which is not necessarily the case in a real PCB geometry. For a real PCB, there may be many area fills on each layer with irregular shapes. These irregular shaped planes form many cavities which extent laterally as well as vertically. The GUI needs to be more flexible for defining the irregular shapes on various layers. The engine is well equipped to handle such cases, and it could give user a more 'close to reality' simulation results for such cases. There are more features in the engine like using a dispersive model for dielectric properties, having multiple dielectrics between layers, etc which can be added to the GUI too.

Also, there are several aspects of the GUI that can be improved. Due to its use for some practical cases, some limitations in the Matlab GUI design have been identified. Lot of these come from the code design of the GUI and may be eliminated by just improving the code design itself. For instance, inserting and deleting layers can be complicated in the GUI. Usually it would be very pain stacking if multiple via are present. Majority of these code design improvements are required to add flexibility to the GUI in making changes to the already existing geometry data. Better code design will assure a more robust GUI, compared to the present version.

Some of these limitations, new features and code design, also depend on the coding environment. It is expected that a change to a more professional GUI development environment will not just make the GUI more user friendly, but also the design and implementation a lot more convenient for the developers. Thus, a choice of coding environment has to be made and some target features, apart from the ones already implemented, should be accounted for in the final GUI development.

Tool development process consists of the user interface development which can facilitate the effective application of the developed product. The GUI development, like any other software, follows the basic life cycle of design, programming and testing. The design phase should be able to account for or satisfy all the requirements (which should remain relatively constant). Only then, the subsequent phases can be completed successfully. The Matlab based GUI development is a step in the design phase which

helps to realize the complexity of design requirements. It also helps in understanding the features necessary in the GUI to meet all the requirements and the usability of each of these features. It acts like proof of concept for the design phase by its short implementation time.

The final GUI developed based on this design can then, not just satisfy the requirements, but also help to simplify the process of accepting inputs from the user, in a very robust manner. This initial development can be considered a big step in the design of the final GUI which will follow.

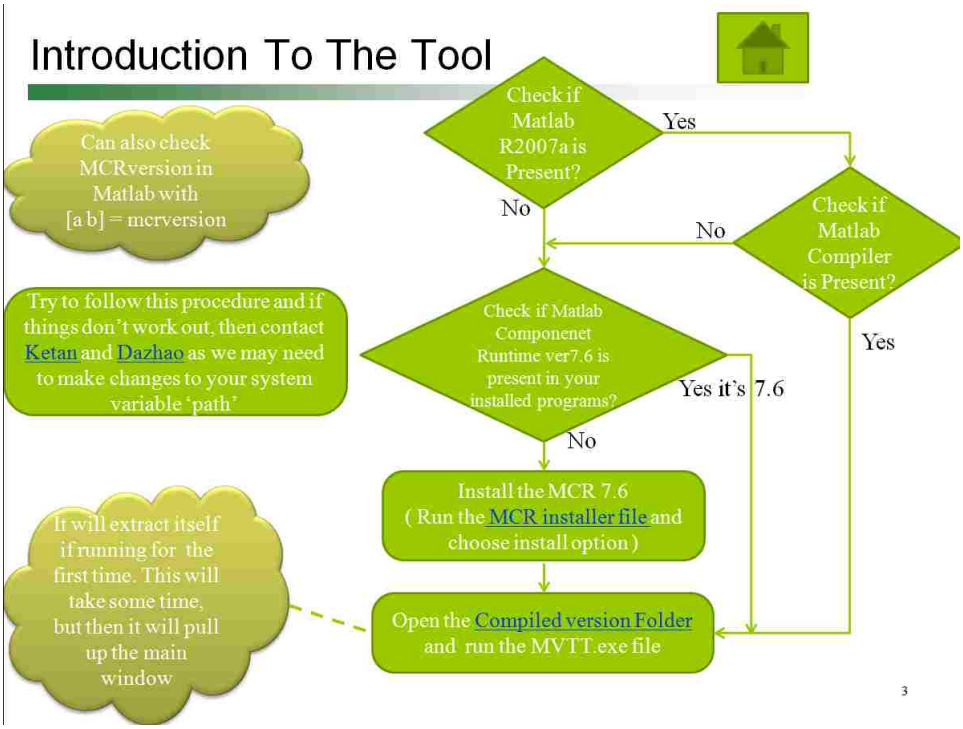
APPENDIX

THE QUICK START GUIDE FOR MVTT

The Quick Start Guide
for
Multilayer Via Transition Tool v2.2c

Outline of Quick Start Guide

Introduction To The Tool



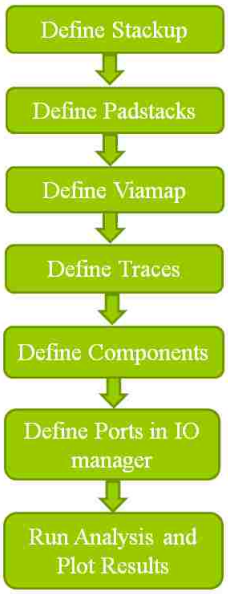
3

Introduction to the Tool - Flow

- We need to follow the flow of this tool. At least for building the geometry correctly, but once it is created, we can make changes to the parameters defined earlier.
- So we need to enter these sets of information in this same order
  1. Stackup
  2. Padstack
  3. Viamap
  4. Traces
  5. Component
  6. IOManager
- Then we can run analysis and use the Plot button to launch S manager and then check our results.

4

Flow



- Stackup: this window will take information about each layer and the dielectric in between the layers
- Padstacks: Each padstack asks for anitpad and pad diameter at each layer and the drill diameter of the Via. Atleast one such padstack has to be defined before proceding to viamap.
- Viamap: This window will acquire informtion about each via(padstack used, start, stop, netname, location)
- Traces. This window will ask for complete trace information.
- Components : This window will ask for information regarding the external components connected to the PCB
- IOmanager: Here we select the ports from a list of all possible ports for the given geometry and arrange them in the order that we want. We can also add plane pair ports which is a vertical port with a terminal on each full layer of a cavity.
- Run and Plot: We can now(only after selecting and arranging the ports in IO manager) Run the analysis. A new file is written(in a folder with project name and time stamp) and passed to the engine . Here the engine saves the results as s or z parameters within same folder and an extension of s_para or z_para for identifying type of file

5

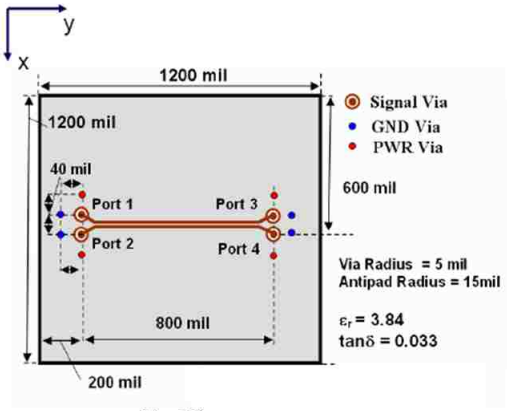
How to use this Tool

- **Project Concept:** We use this concept of Project such that the tool will have a project name 'Untitled' for new geometries until they are saved. Once saved or if a geometry is read from a file, then the file name is take up as the project name and the path as the location to save results. When the tool runs, it creates a sub directory with the project name and a time stamp and then saves input text files and all chosen output files into this sub directory.
- **Creating a new geometry:** We can from scratch build a geometry in the GUI and write a text file to save it.
- **Using a previously created geometry(Loading a saved Geometry file) :** A user-written or GUI-written text file can be read in, and the tool used to check the geometry described by this file as also to make changes to this geometry and then save them back.
- **Defining the ports:** Further for the geometry created, we can define ports(Via ports, differential ports and trace ports) and even plane pair ports. Then we can arrange these to get the S or Z para result with numbering as required
- **Advanced Settings**: These are features used to teak the geometry details and thus get a closer model of the actual geometry. For example, capacitance from pads on signal layers.
- **Running the tool and Checking results:** Running the tool will save the geometry to a file and saves result to file with name with 'Results_S_para' or 'Results_Z_para' appended to it. This can be read in from the S manager (Plot results Button) and used to plot the curves required. These files along with CSV versions are saved inside a subfolder having the timestamp in its name.

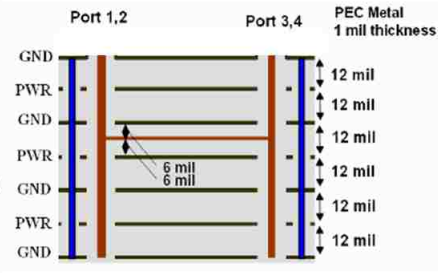
6

How to use this Tool - Creating a New Ge ry

- We will try to create this geometry as an example along our geometry building exercise. It is named 'Example_case4.txt' in provided examples. This is a geometry given by Kevin Gu and Renato Donadio from their DesignCon paper.



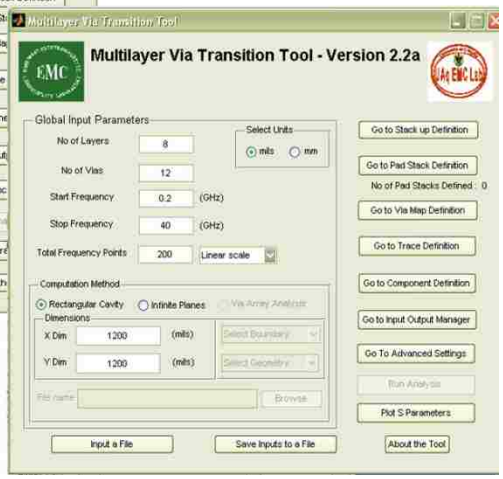
Top View



Side View

7

Main Window fill up

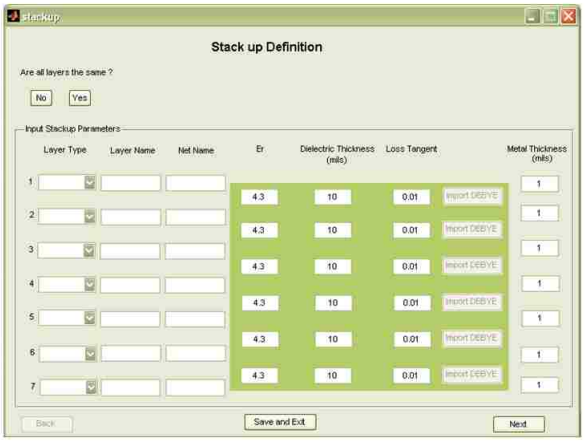


Here we have an Option of selecting from computational methods

1. Rectangular Cavity
2. Infinite Planes (No board dimensions required)

Stackup

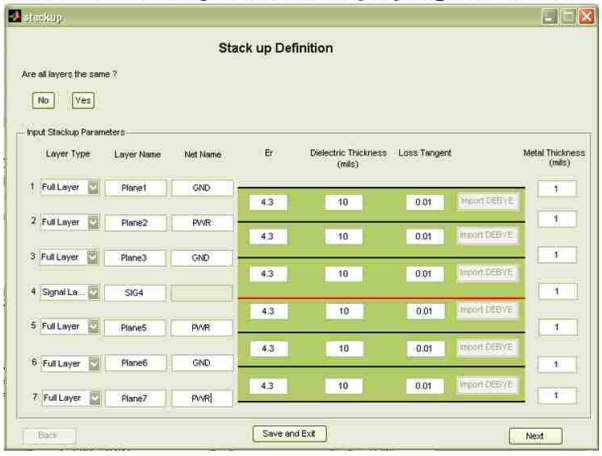
- Click on "Go To Stack up" to access the stack up window.
- This is *just the first 7 layers*, if more than 7 layers are required for the geometry, then we get the rest in sets of 6 layers in the following windows as we click on 'Next'. This also contains default values for the material properties
- So, for our example geometry, our remaining one layer will just come as a single layer on the next window. Now we will fill this window and then proceed to next one.



9

Filling the Stackup

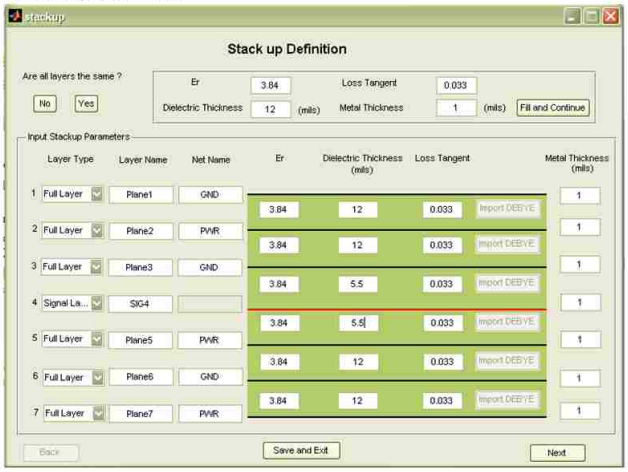
- To start with, we can select the Layer type for all layers according to our geometry
- You will observe that each of the layers are automatically given a unique name and if it's a full layer then given a net name. Now you can rename and change the net name of power layer as we have done here. ***Please do not give a net-name for full plane as 'SIG'***
- ***If renaming, please give unique names to Layers***
- Next we have to fill the properties of dielectrics and layers. We can go about in 2 ways:
  - Fill individually the remaining boxes
  - Or make use of the default functionality
- Here we go about the second the way to explore more functionality. So click on Yes where it says all layers are the same.



10

Filling up the Stack up

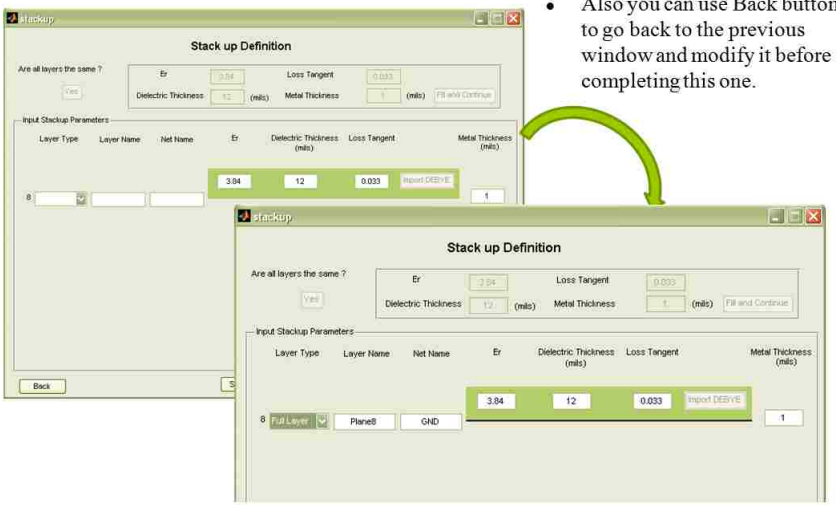
- Now just fill up the default values we would want for each of these properties. And click on '**Fill and Continue**'
- This will fill up all the remaining fields and will also be default for the rest of the windows.
- Now you can modify the fields which you want different from the default values
- We will continue with the values set here.
- At any point, you need to stop , then just hit 'Save and Exit' and your work will be saved in a variable. But if you are exiting the tool, then you need to save inputs to a file from the main window
- To proceed , click on next button. We have to fill till the finish to complete the entire stack



11

Filling up the Stack up

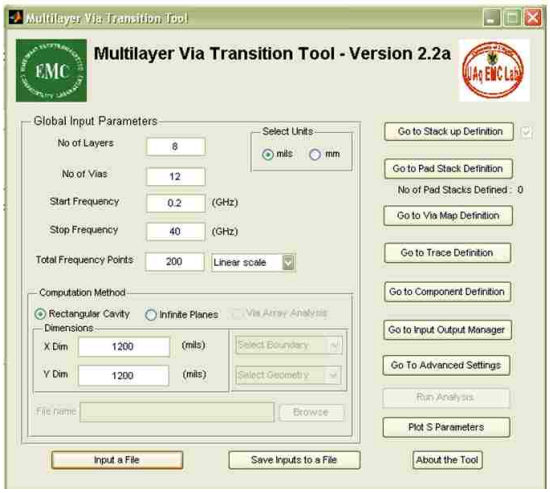
- Observe that the default values are filled for the fields on all stack up windows and they can be modified only on the 1st window of stack up. But modifying individual values in between is ok.
- Now we just change the last layer type to Full layer and Click Finish to complete our description of stack up for the GUI
- Also you can use Back button to go back to the previous window and modify it before completing this one.



12

Stack up is defined: Change visible on Main

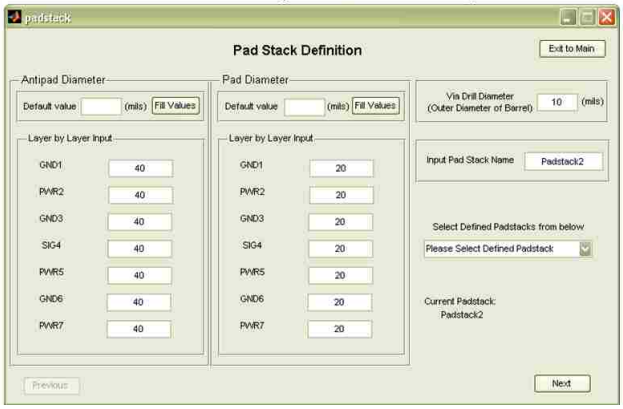
- Observe the tick mark or check box on the right of the 'Go to Stack up' is checked indicating that we have finished this part of the GUI.
- Now we can go on to Padstack and define a new padstack. Please click on 'Go To Padstack'



13

Defining a Padstack

- We can define only one Padstack at a time and in it 7 layers per window. So we have to traverse through all the windows to complete the padstack information for layers and hit finish button on the last window to exit.
- We will define a padstack Padstack2 with the following specs:
  - Drill size (This is the Barrel outer Diameter) = 10 mils
  - Antipads = 40 mils and Pads = 20mils(As we d not have Pads we set Pad Diameter = Drill Diameter)
  - As values are for all layers, we can use default.
  - Then we can also change individual values if required.

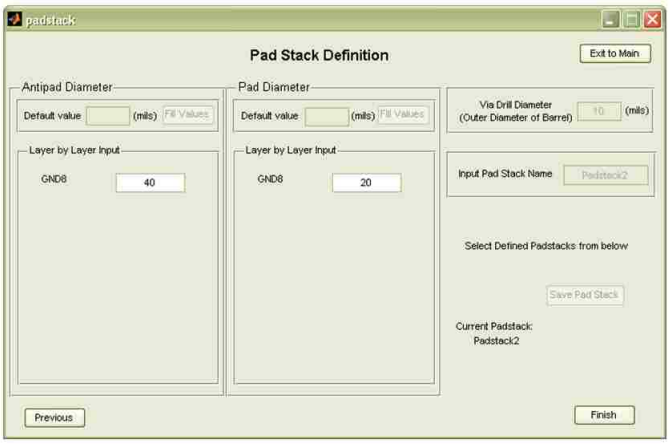


- To use default, enter value in default field and click on Fill values.
- Then we can modify the individual values which differ from the default
- After this we will click next to fill data for next set of layers, which in our case is just 1 layer

14

Defining a Padstack

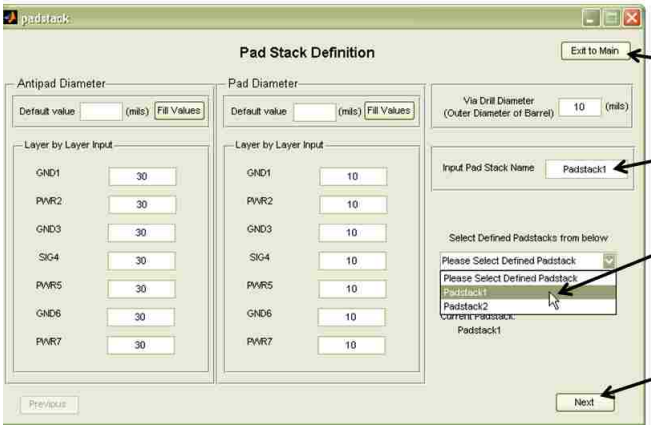
- Exit Button Has been added to the Padstack, and the close button is disabled. But if the Exit button is used, the pad stack that the user defined will not be saved. Only finish button will save the pad stack that the user defined.
- But if we enter this window and decide that we do not want a new padstack we can just load any previously defined padtack.



15

Loading a saved Padstack

- Click on 'Go to Padstack Definition' to open a new padstack window. But if we do not want to define a new padstack now (or simply to leave the padstack definition window), so we load a saved padstack.
- Use the drop down menu on the right to select the padstack 'Padstack1'. The saved padstack 'Padstack1' will be automatically loaded.
- This will load the data of the old padstack and we can modify it if need be. After loading if we want to exit we can just go on clicking on the next button till we exit the padstack definition and come to the main window.



Using Exit button to exit is also OK

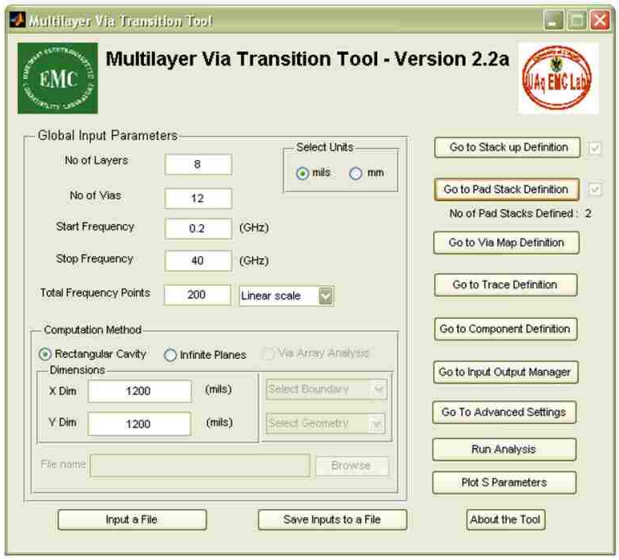
Indicate the current padstack name

Select a saved Padstack: Padstack1

Use next till the last window with the 'Finish' to exit

16

Padstack Defined: Main window chang



We can observe the checbox next to padstack button comes up whenever we define atleast 1 padstack.

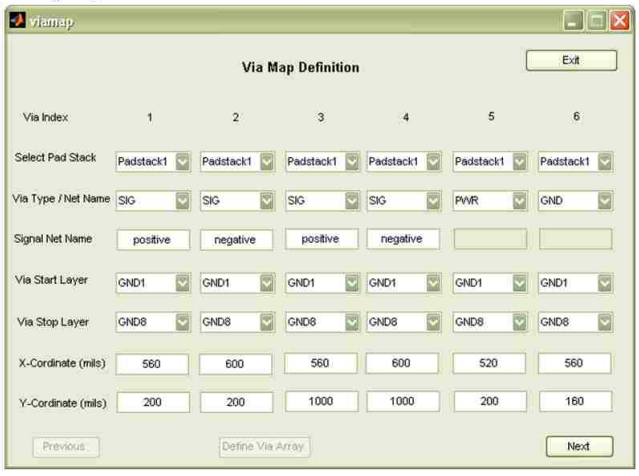
Also we the displays the number of padstacks defined.

Next click on 'Go to Via Map Definition' to input informaiton about the Vias

17

Via Map Definition

- The Via Map window lists 6 vias at a time in a column-wise fashion. We see the first 6 vias here and the next 6 vias will appear in the next window.
- The close button is disabled. User can exit by clicking the Exit button on the up right corner.



The via map definition consists of parameters like padstack to use, type of via, via netname, via startlayer viastop layer, and coordinates of via.

18

Via Map Definition

- **Padstack**: We start with the padstack selection using the top row of dropdowns which list the names of the defined padstacks. When more than one padstack is defined we need to tell the GUI which padstack to use for each via. For this, default is that it uses the first padstack that was defined. So in our case the 'Padstack1' is the only one in the list and so its already selected.
- **Via-type and Net names**:
  - **Via shorted to any of the Full planes (GND or PWR in our case):** We have to choose the net name of the Full plane to which it connects. Now it will have the same net name as chosen, so the Engine will consider it shorted to the full plane of that net.
  - **Via is not connected to any full plane**: It is implied that this is a *signal via*, so we choose the viatype to be SIG. Now that we have made it clear it's a signal via, we must give it a net name. This is filled in the edit box below the drop down menu.
- **Via Start layer and Stop Layer**: Then we can see that the next 2 rows are Via start and Via stops. These drop downs will list all the internal full layers and the top and bottom layers. We can select one in each drop down selection as a layer where the Via Starts and Via Stops. But understand that our formulation does not allow the start or stop of any via on a signal layer inside the PCB (on the surface this is fine).
- **Co-ordinates**: Now the last part is the Via co-ordinates. We need to give the x and y co-ordinates for the via in chosen dimensional units.
- The filled up Via Map window is shown on next slide.

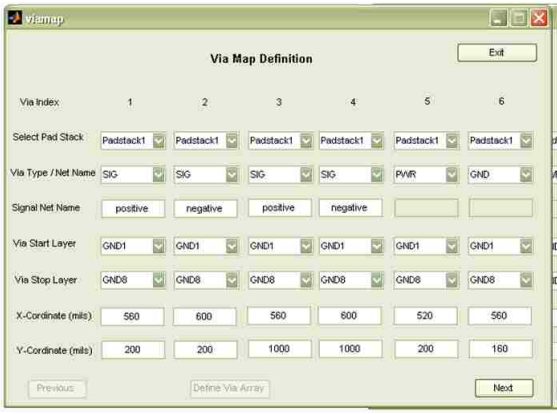
19

Via map window

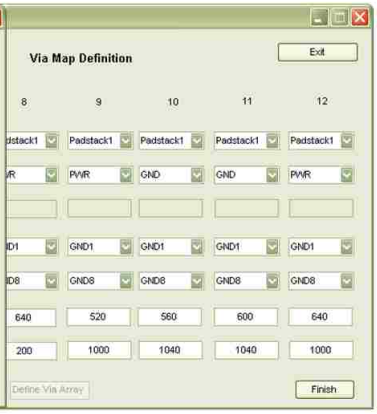
For our exampleWe can Consider the first 4 signal Vias as Signal Vias (Nets Positive and Negative). And the rest as GND and PWR vias according to the coordinates

Follow the shown image of Via Map Windows to get the same via topology.

1st Window



2nd Window

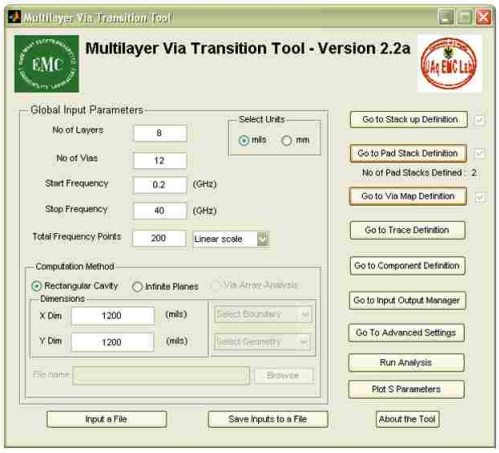


So we take Via 12 as left end of Trace, 3 and 4 as right end. Vias 5&8 are PWR and 6&7are GND for Left Side. Similarly 9 &12, and 10&11 are PWR and GND respectively for Right Side

20

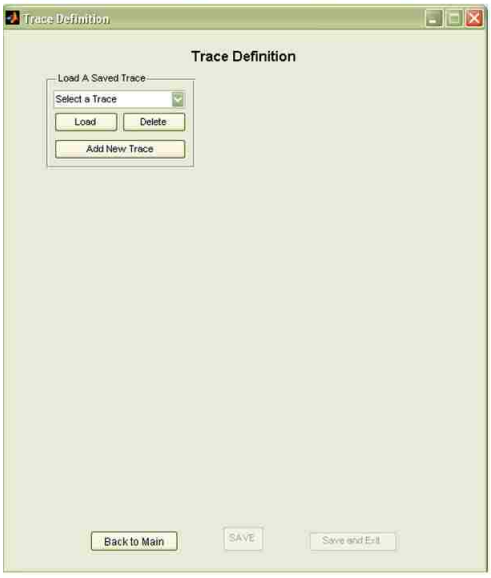
Via map defined:

- We can similarly finish the next page of the Via map
- On coming back to main we will notice the checkbox indicating the completion of Via map.
- Now we can go on to Traces and define a new differential trace. Please click on 'Go To Trace Definition'



21

Trace Definition Window



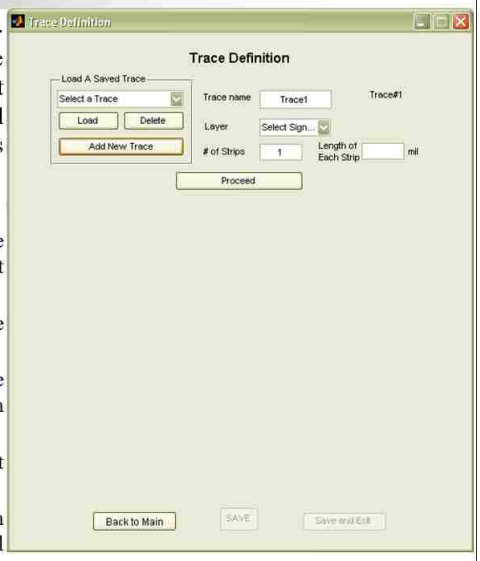
- When we enter the Trace Definition window, we will find a blank page.
- We can select an already defined trace and then click on
  - 'Load' to edit it or check its details.
  - 'Delete' to just delete this trace
- 'Back to Main' is provided to return to main window. If you are defining a trace, then the latest changes will not be saved.

22

Defining a new Trace : Overall data

- ***When we say Trace***, we refer to a multi-conductor transmission line. So each trace can have a number of coupled microstrips(not in this version) or striplines, all on one signal layer. The number of strips refers to this number of microstrips or striplines.
- ***Click on 'Add new Trace'***
  - A set of details appear which are required to be filled out initially before proceeding to the next part of the trace definition.
  - Name: A default name is given to our Trace. We will change it to 'Diff_Trace1'
  - Layer: It indicates the signal layer on which the Trace lies. Only One signal Layer has been defined Sig4 so we select it.
  - As we will define a Differential Trace, we put #strips as 2.
  - Length: It states the effective Length of the each component strip(assumed equal for all), useful when PUL parameters or cross section is stated.

When done, click on 'Proceed'

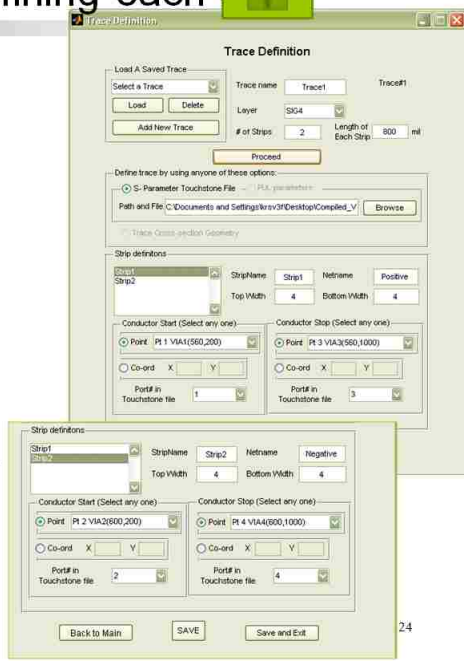


23

Defining a new trace: Defining each

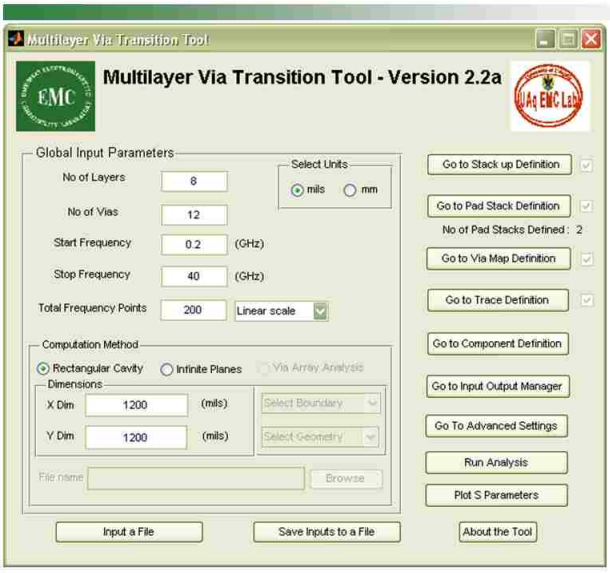
- Filename: Here we enter the filename of the touchstone file which contains the parameters for the coupled striplines we are defining. If a text file is read to load the present geometry and this file was saved in a different computer, PLEASE assign the touchstone file name ***again***. A relative path will be used later to avoid this issue.
- Each strip has a name, netname, start & stop point, and start stop port. It is selected one at a time from the listbox and then defined.
- Netname has to be the same as the vias(if its connected to a via.) its connected to.
- Start/stop points can be selected from a list, or a new location can be provided.
- Start/Stop port refer to the particular port in the touchstone file which corresponds to this particular end.

When completed filling all information for 2 strips, we may click on 'Save and Exit' or 'Save' and then use 'Back to Main' to return to main window.



24

Trace is defined



Notice the check marked next to Traces indicating that atleast one trace is defined and no trace definition is incomplete.

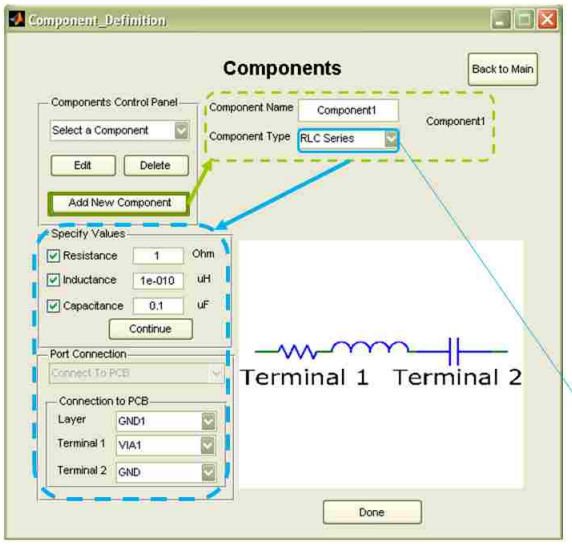
Now we have the entire Geometry except the components created in the GUI.

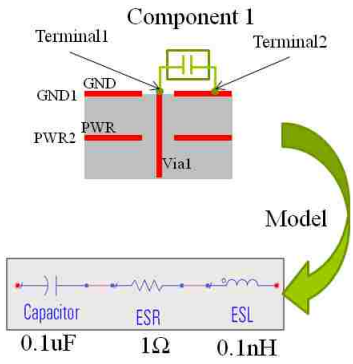
Click on 'Go to Component Definition'.

Click on 'Go Component Definition', to proceed to component definition

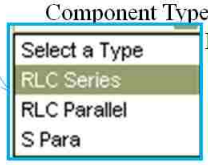
25

Using Component Window to Add a Series RLC Component



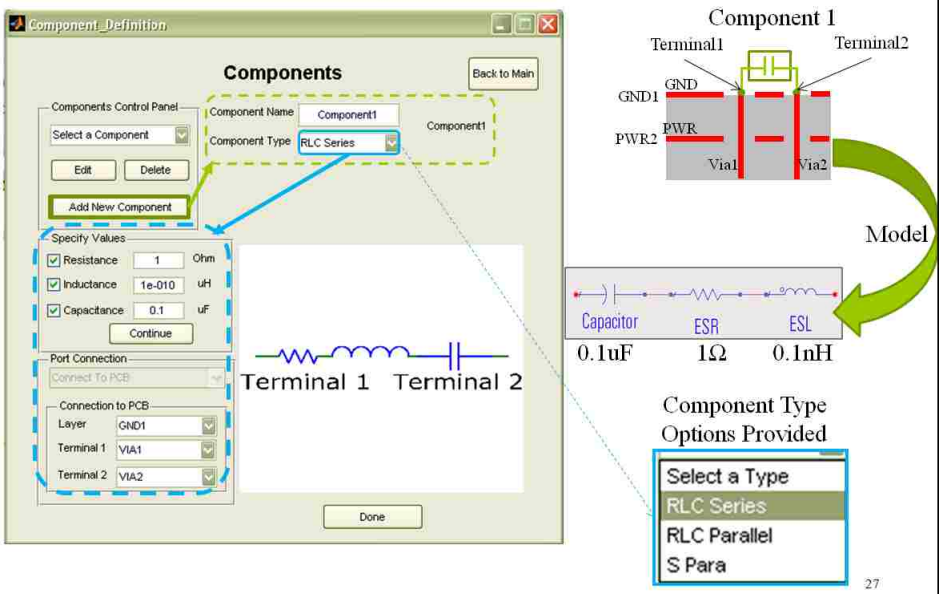


Component Type Options Provided



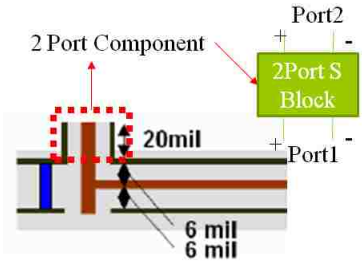
26

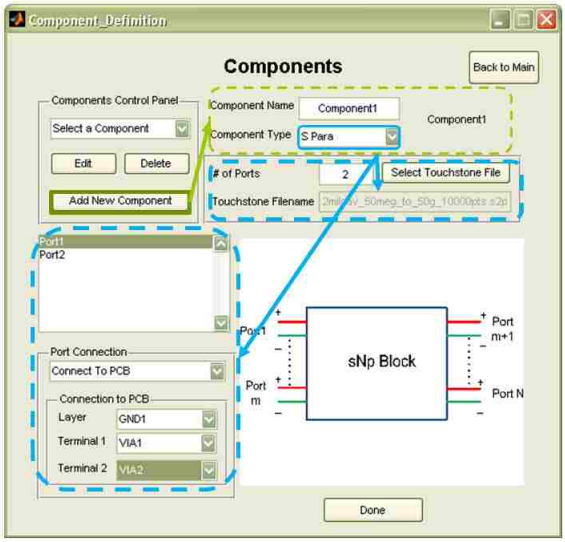
Using Component Window to Add a Series RLC Component



27

Using Component Window to Add a S Block Component - Connection of Port1

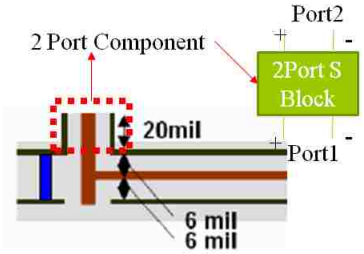




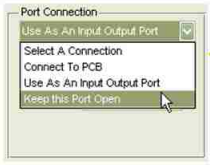
- Bottom port lets say Port 1 should be connected toVia1 and Plane GND around the Antipad
- Top Port, say Port 2, should be used as the IO port

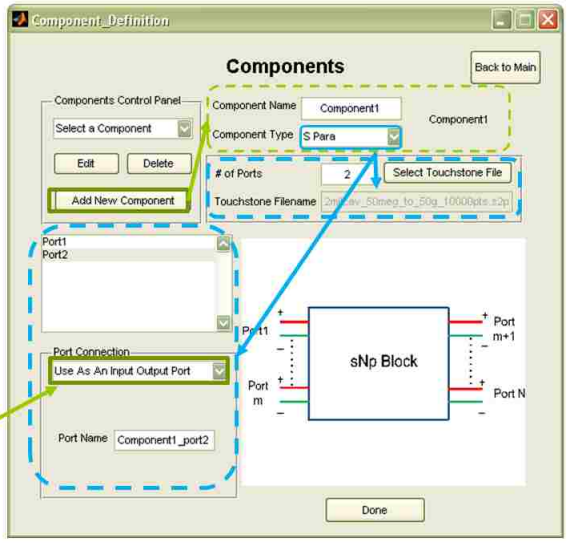
28

Using Component Window to Add a S Blo
Component - Connection of Port2



- Bottom port lets say Port 1 should be connected toVia1 and Plane GND around the Antipad
- Top Port, say Port 2, should be used as the IO port





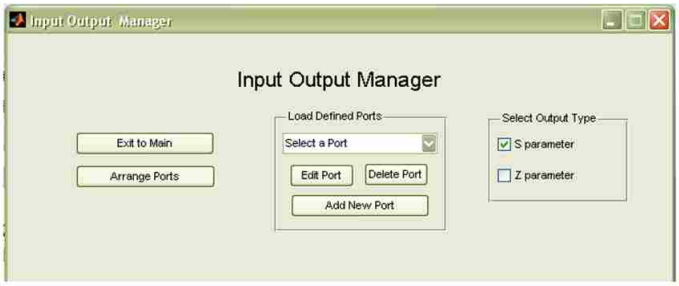
29

IO Manager Window

- IO Manager window also opens up as a blank window just like the Trace window.
- In this window, we can
  - Define new ports
  - Edit ports
  - Delete ports
  - Arrange ports
  - Select type of output expected

We have 3 types of ports

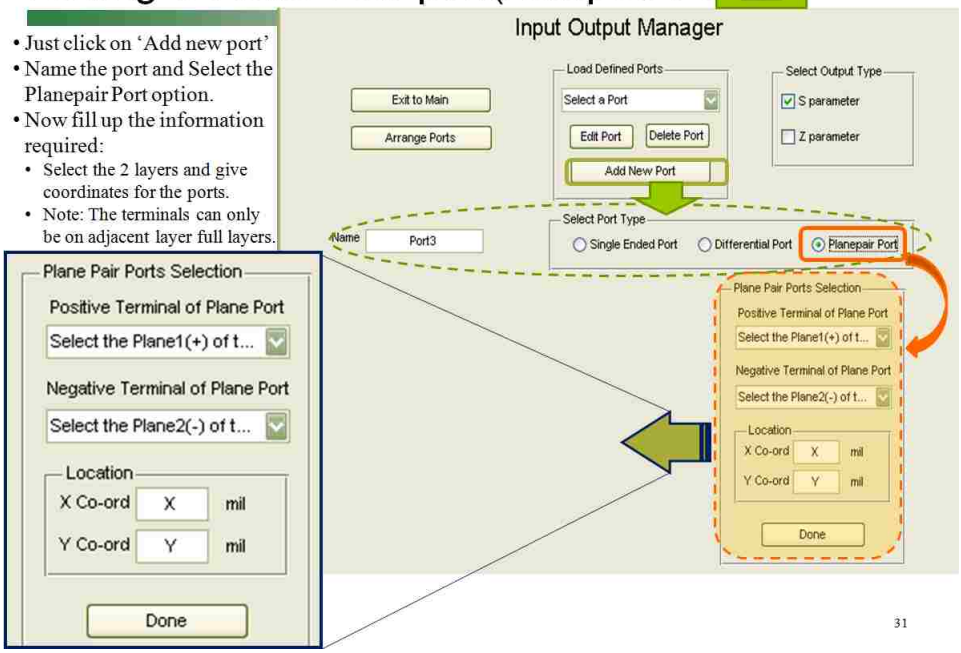
- **Planepair ports:** A vertical lumped port set up anywhere in a cavity.
- **Single Ended :** A Single via port or trace port defined as a Via and a Reference Layer or an open end of a Trace strip and one of its Reference Layers respectively.
- **Differential Port:** Has 2 complete Single Ended ViaPorts or Trace Ports, but the Reference layer has to be common.



30

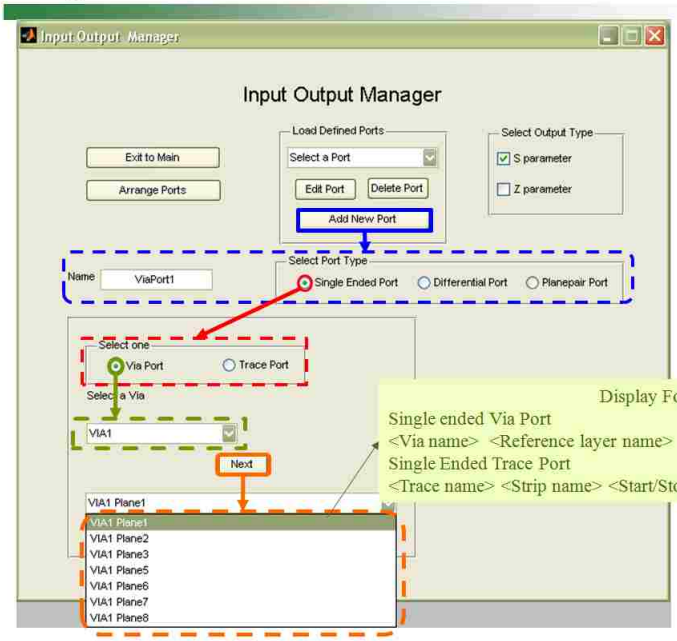
Adding a Plane Pair port(Lumped Po

- Just click on 'Add new port'
- Name the port and Select the Planepair Port option.
- Now fill up the information required:
  - Select the 2 layers and give coordinates for the ports.
  - Note: The terminals can only be on adjacent layer full layers.



31

Single Ended Port - Procedure

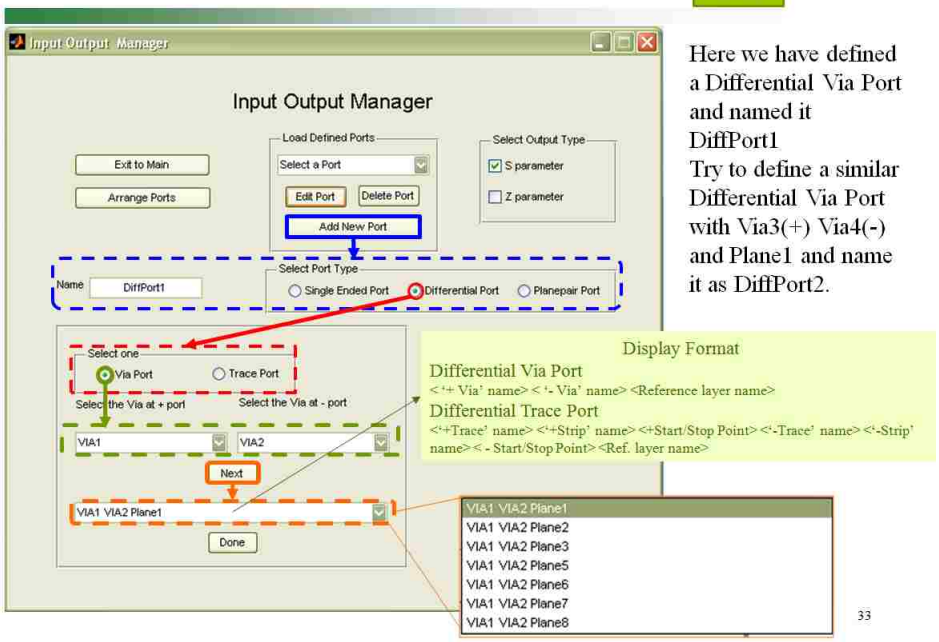


The Diagram shows a flow to define ViaPort1.
For our example, we can similarly define Ports ViaPort2, ViaPort3, and ViaPort4, as ports between vias VIA2, VIA3, and VIA4 and the Plane1 respectively.

Display Format
Single ended Via Port
<Via name> <Reference layer name>
Single Ended Trace Port
<Trace name> <Strip name> <Start/Stop Point> <Ref. layer name>

32

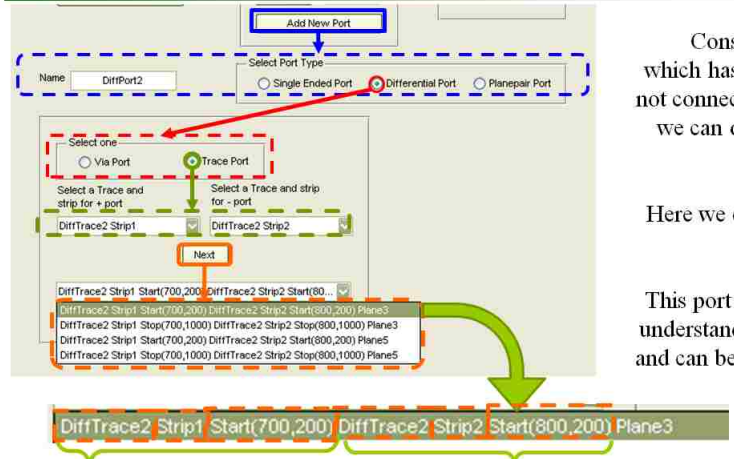
Differential Via Port



Here we have defined a Differential Via Port and named it DiffPort1
Try to define a similar Differential Via Port with Via3(+) Via4(-) and Plane1 and name it as DiffPort2.

33

Differential Trace Port Example



Consider a DiffTrace2 which has open ends and is not connected to Vias. Then we can define Trace Ports on these ends.

Here we consider one such case .

This port was just added to understand Diff Trace Ports and can be deleted from our example later.

Positive Port
Trace – DiffTrace2
Strip – Strip1
Point – Start of Selected Strip @ (700,200)

Negative Port
Trace – DiffTrace2
Strip – Strip2
Point – Start of Selected Strip @ (800,200)

Common Reference Layer name – Plane3

34

Arranging the Ports -1

- Now if we are done defining the ports, we can arrange the ports in the numerical order we want (if they are not already in the correct order).
- By order we refer to the Port Numbering which will be used in the Output touchstone files and *.csv files.

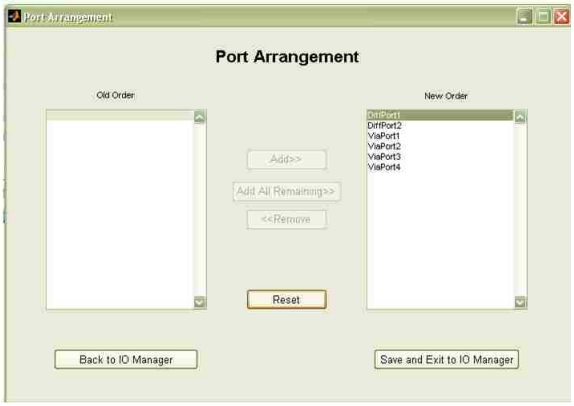
w window with 2 lists :

- Old Order: Which contains a list of port names in the present order.
- New Order : Which is blank initially but will finally contain the order in which ports are arranged

35

Arranging the Ports -2

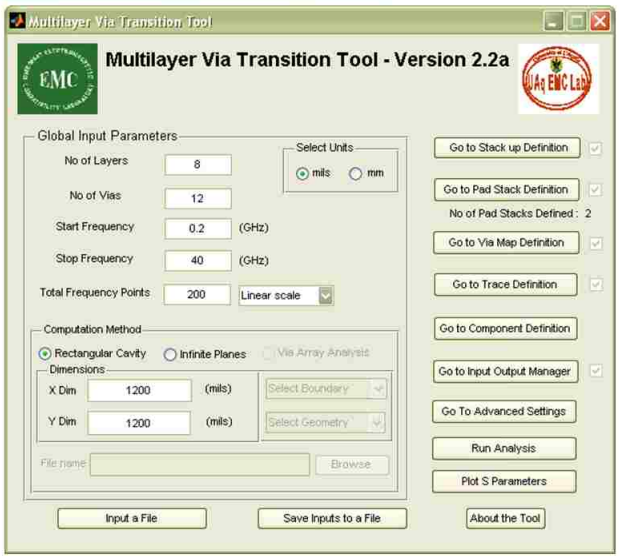
- We will rearrange as follows
  1. Select DiffPort1 and click on Add.(We see both lists modified)
  2. Select DiffPort2 and click on Add.(Now we see 2 Diff ports on the new List, Single ended ports remain in the old list)
  3. Click on add all remaining. (all ports are copied to the other list).We get the same order as we have this screenshot.



- *Back to IO manager:* If no changes are to be saved then use this button to return to IO manager. It can be used at any stage.
- *Save and Exit IO Manager:* This will save the order of ports as that in the new Order list and then take us back to IO manager

36

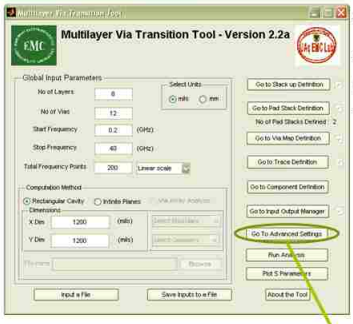
The Port Definition is complete



- We observe a that now there is a check mark next to IO manager.
- Also 'Run Analysis' button is now Enabled for use. So we can Run the Tools and Check the results.
- **We can with the current engine only do Single Ended Ports. The differential ports were for illustration only, and we delete them before we 'Run Analysis'**

37

Advanced Settings Window

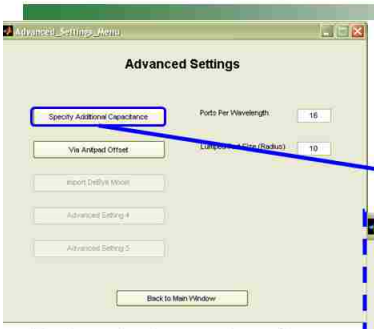


The information on this window and its sub – windows is not necessary information for the kernel to run the simulation. A new user may skip this window and its sub – windows.

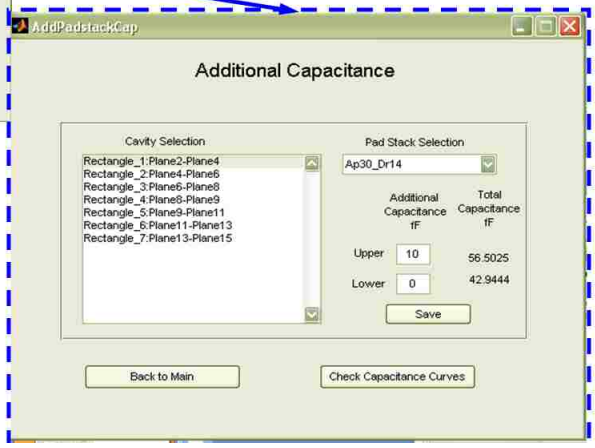
- It allows the user to tweak the simulation parameters like Ports per wavelength, or Lumped port size, etc.
- It also lets the user access the Additional Capacitance and Via Anti-pad Offset window

38

Additional Capacitance Window

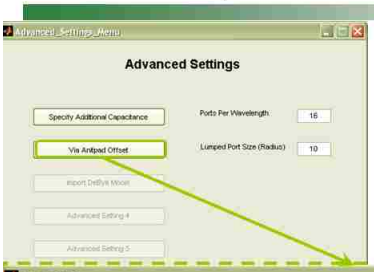


This extra capacitance is used for compensation of the capacitance not accounted for by the kernel, viz, the capacitance caused by the Pad on a signal layer or due to fringing fields on top or bottom layer. Default value is zero

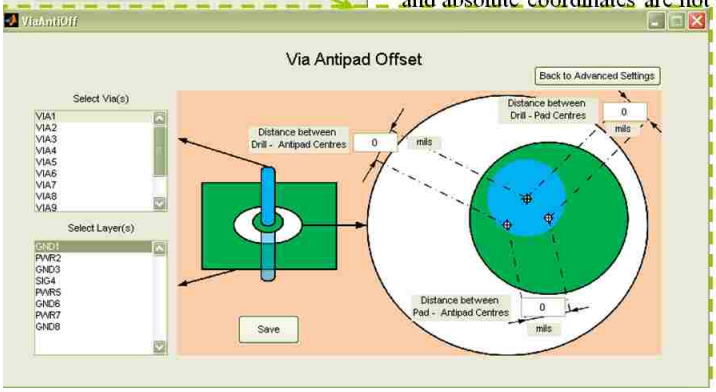


- Each cavity has a value of extra capacitance for each Pad stack
- Multiple Selection of Cavities is allowed.
- Total capacitance is sum of $C_{from\ Formula}$ and $C_{additional}$
- User should click on 'Save' to store the changed/new value for Additional Capacitance.

Via – Anti-pad Offset Window

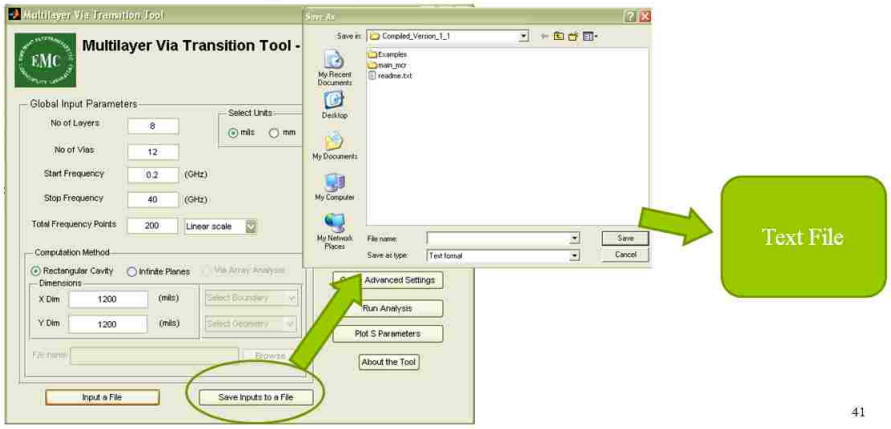


- If there exists a misalignment between the via drill center and the anti-pad center, then this helps to account for the change in capacitance due the such irregular geometry.
- To generalize, we ask user to provide 3 offsets, drill-pad, drill-antipad, and pad-antipad. Default values are zero.
- As this geometrical irregularity does not affect the other features of this tool, only relative distances are enough and absolute coordinates are not required.



Saving geometry to file

- To save the geometry to a txt file, at any stage just return to the main window and click on the button 'Save Inputs to File'.
- A Save as dialog box pops up, and you give the name and location of file to save as or replace.
- The file written is the geometry file in txt format.



41

Loading a saved Geometry file

- To load a text file, we click on 'Input a File' button.
- In the open file dialog box which appears, we select the particular file we require to be loaded.
- This will load the file into the GUI. We can see the information in the file in the Stack up, Pad stack, Via map, or IO manager, as much information was present in the input file.

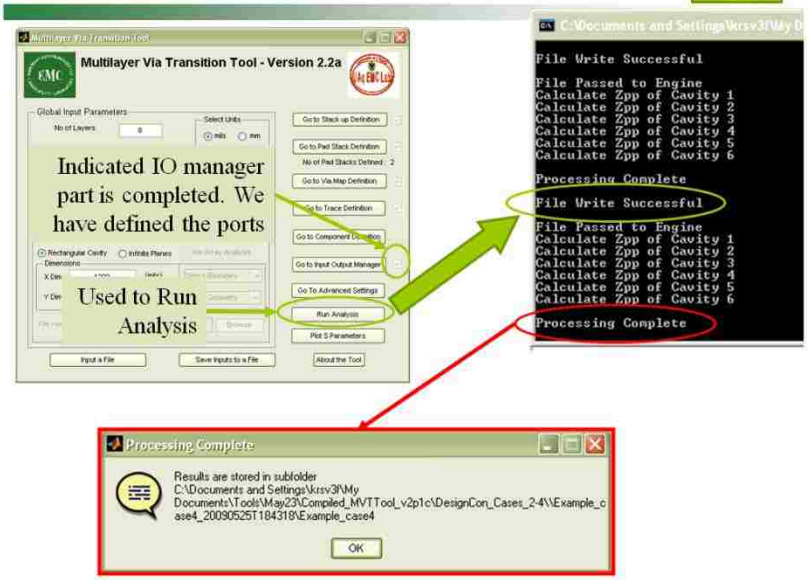
42

Run Analysis -1

- After completing the port definition in IO manager, we are now ready to run analysis.
- Here we just click on 'Run analysis', and the Tool will then
  - Create a folder at the same location as a project file.
  - This folder has the name as project name appended with a date and time stamp as '_YYYYMMDD7HHMMSS'.
  - The it writes a new geometry text file inside the above mentioned folder named as 'Inputfile.txt'
  - This file is then passed on to the Engine.
- The results of the analysis will be saved in a S parameter or Z parameter touchstone and csv files with the name 'result' suffixed with '_S_para' or '_Z_para' and stored in the same folder.
- The command prompt or the Black console window indicates the progress in the analysis.
  - When the file is written, it says – File write Successful
  - Then it say - File passed to Engine
  - Then this display is controlled by the Engine till we get a message saying – Processing complete- This Indicates GUI has the control and processing is over.

43

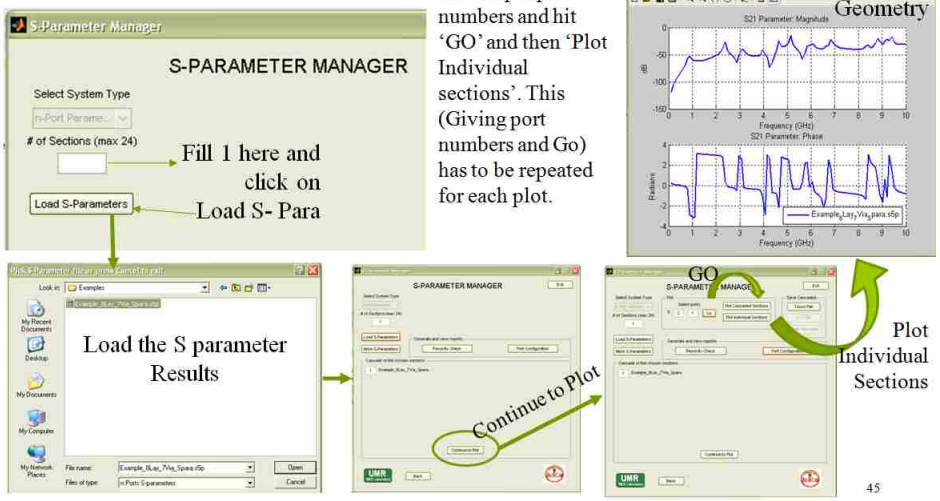
Run Analysis -2



44

Plotting the Results

- Click on Plot results and the S-para Manager comes up.
- Fill # of sections as 1 and load the S Para from same location as where we saved the geometry file. Then click on Continue to plot.
- Then input port numbers and hit 'GO' and then 'Plot Individual sections'. This (Giving port numbers and Go) has to be repeated for each plot.



45

Limitations of the Tool

- By Limitations we mean the situation in the tool which might lead to an error.
- The Major limitation found in this tool is while using the RED CLOSE BUTTON(X)
  - This causes an error in the GUI.
  - We have added an Exit button on each sub window and disabled the close button on most of them.
- For the rest of the tool, we will keep adding here, but for now:
  - Please give Unique names to Layers and do not give net-name for a Full layer as 'SIG'
  - Give Unique names to the all ports.
  - Vias are named automatically as 'VIA<index of via>'. If you need to change this you can change the text file and load it but be sure to change all instances of the Via. The engine uses it but the tool does not display the name
  - Changes in Text files : For any change in the Text file(you may use the replace all feature of whichever text editor you use) please be consistent throughout the file and replace all instances of the changed layer name or netname.

46

BIBLIOGRAPHY

[1] T. Okoshi, *Planar Circuits for Microwaves and Lightwaves*, Berlin, Germany: Springer-Verlag, 1985, ch. 2.

[2] T. Okoshi, Y. Uehara, and T. Takeuchi, "The segmentation method–an approach to the analysis of microwave planar circuits," *IEEE Trans. Microw. Theory Tech.*,vol. MTT-24, 662-668, Oct. 1976.

[3] C. Balanis, *Advanced Engineering Electromagnetics*, New York, USA: John Wiley & Sons, Inc.1989, ch. 14.

[4] C. Schuster, G. Selli, Y. H. Kwark, M. B. Ritter, J. L. Drewniak, "Progress in representation and validation of physics-based via models," in *Proc. 11th IEEE Workshop Signal Propagation on Interconnects.*,May 2007, Genova, Italy.

[5] C. Wang, J. Mao, G. Selli, S. Luan, L. Zhang, J. Fan, D. J. Pommerenke, R. E. Dubroff, and J. L. Drewniak, "An efficient approach for power delivery network design with closed-form expressions for parasitic interconnect inductance," *IEEE Transaction on Advanced Packaging*, vol. 29, no. 2, pp. 320- 334, August 2006.

[6] Y. Zhang, J. Fan, G. Selli, M. Cocchini, and D. P. Francesco, "Analytical Evaluation of Via-Plate Capacitance for Multilayer Printed Circuit Boards and Packages," *IEEE Trans. on Microwave Theory Techn.*, vol. 56, no. 9, pp. 2118 – 2128, September 2008.

[7] H. Wang, Y. Zhang, J. L. Drewniak, J. Fan, B. Archambeault, "Capacitance Calculation for Offset Via Structures Using an Integral Approximation Approach Based on Finite Element Method," *43rd International Symposium on Microelectronics (IMAPS)*, Raleigh, USA, 2010.

[8] F. De Paulis, J. Fan, "Link Path Design on a Block – by – Block Basis," *IEEE International Symposium on EMC*, 2008.

VITA

Ketan Shringarpure was born on October 14th, 1985 in Mumbai, India. He completed his Bachelor's degree in Electronics and Telecommunication from Thakur College of Engineering and Technology at Mumbai University, in August 2007. He joined the Electromagnetic Compatibility Laboratory at the Missouri University of Science and Technology, in Jan 2008, as a graduate research assistant. He received his Master's Degree in Electrical Engineering from Missouri University of Science and Technology in Dec 2010.