



HyperLynx® 3D EM

MDSPICE USER'S MANUAL

Software Version 15.2

© 1999-2012 Mentor Graphics Corporation
All rights reserved.

This document contains information that is proprietary to Mentor Graphics Corporation. The original recipient of this document may duplicate this document in whole or in part for internal business purposes only, provided that this entire notice appears in all copies. In duplicating any part of this document, the recipient agrees to make every reasonable effort to prevent the unauthorized use and distribution of the proprietary information.

This document is for information and instruction purposes. Mentor Graphics reserves the right to make changes in specifications and other information contained in this publication without prior notice, and the reader should, in all cases, consult Mentor Graphics to determine whether any changes have been made.

The terms and conditions governing the sale and licensing of Mentor Graphics products are set forth in written agreements between Mentor Graphics and its customers. No representation or other affirmation of fact contained in this publication shall be deemed to be a warranty or give rise to any liability of Mentor Graphics whatsoever.

MENTOR GRAPHICS MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

MENTOR GRAPHICS SHALL NOT BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING BUT NOT LIMITED TO LOST PROFITS) ARISING OUT OF OR RELATED TO THIS PUBLICATION OR THE INFORMATION CONTAINED IN IT, EVEN IF MENTOR GRAPHICS CORPORATION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

RESTRICTED RIGHTS LEGEND 03/97

U.S. Government Restricted Rights. The SOFTWARE and documentation have been developed entirely at private expense and are commercial computer software provided with restricted rights. Use, duplication or disclosure by the U.S. Government or a U.S. Government subcontractor is subject to the restrictions set forth in the license agreement provided with the software pursuant to DFARS 227.7202-3(a) or as set forth in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clause at FAR 52.227-19, as applicable.

Contractor/manufacturer is:

Mentor Graphics Corporation

8005 S.W. Boeckman Road, Wilsonville, Oregon 97070-7777.

Telephone: 503.685.7000

Toll-Free Telephone: 800.592.2210

Website: www.mentor.com

SupportNet: supportnet.mentor.com/

Contact Your Technical Writer: [supportnet.mentor.com /doc_feedback_form](http://supportnet.mentor.com/doc_feedback_form)

TRADEMARKS: The trademarks, logos and service marks ("Marks") used herein are the property of Mentor Graphics Corporation or other third parties. No one is permitted to use these Marks without the prior written consent of Mentor Graphics or the respective third-party owner. The use herein of a third- party Mark is not an attempt to indicate Mentor Graphics as a source of a product, but is intended to indicate a product from, or associated with, a particular third party. A current list of Mentor Graphics' trademarks may be viewed at: www.mentor.com/trademarks.

End-User License Agreement: You can print a copy of the End-User License Agreement from: www.mentor.com/eula

Chapter 1 Introduction

Use of computer-aided design (CAD) programs has been a common practice in modern electronic design. According to the types of signals being processed, electronic circuits are divided into two groups: analog circuits and digital circuits. According to the frequency ranges of applications, they are divided into lumped element circuits and distributed circuits.

Typical analog circuits include those using magnitude modulation and frequency modulation used in radio sets, TV sets, RF and microwave circuits. Typical digital circuits are those logic circuits used extensively in the IC circuits in the computer industry.

Due to the differences in the circuits, time-domain SPICE simulators are normally used in the design of digital circuits and low frequency analog circuits. Microwave network simulators are normally used in the design of high frequency RF and microwave circuits. In the following sections, we will briefly illustrate the two types of simulators.

Section 1.1 Time-Domain SPICE Simulator

A typical low frequency linear circuit consists of mainly resistors, capacitors, inductors and mutual inductors. A resistor is defined as:

$$V = I R \quad (1-1)$$

Where R is the resistance of the resistor (unit: Ω); V is the voltage applied between the two terminals of the resistor (unit: V); I is the current through the resistor from the positive to the negative terminals (unit: A).

A capacitor is defined as:

$$I = C \frac{dV}{dt} \quad (1-2)$$

Where C is the capacitance of the capacitor (unit: F); dV/dt is the derivative of the voltage with respect to time in second; I is the current going through the capacitor.

An inductor is defined as:

$$V = L \frac{dI}{dt} \quad (1-3)$$

Where the L is the inductance of the inductor (unit: H); dI/dt is the derivative of the current with respect to time in second; V is the voltage applied on the inductor.

A mutual inductor is defined between two inductors:

$$V_1 = L_{11} \frac{dI_1}{dt} + M_{12} \frac{dI_2}{dt} \quad (1-4)$$

$$V_2 = M_{12} \frac{dI_1}{dt} + L_{22} \frac{dI_2}{dt} \quad (1-5)$$

Where L_{11} is the (self) inductance of 1st inductor; L_{22} is the (self) inductance of the 2nd inductor; M_{12} is the mutual inductance between the two inductors; I_1 and I_2 are the currents through the two inductors, respectively. V_1 and V_2 are the voltages applied on the two inductors, respectively. The alternative representation of a mutual inductance is

$$K_{12} = M_{12} / \sqrt{(L_{11} \cdot L_{22})} \quad (1-6)$$

A typical lumped element circuit consists of many R, C, L and K. They are connected together in some specific way to achieve some specific performance. Normally, they have multiple input and output terminal pairs. Each input or output terminal pair is considered as a port.

Each resistor, capacitor or inductor can be considered as a 1-port circuit. For such a 1-port circuit, the equation describing the voltage and current is a simple 1st order differential equation. We can analytically solve the time signal of voltage if we know the time signal of the current, or vice-versa.

For a practical circuit, we have many R, C, L and K in a complicated topology. The equations for the solution become a set of high order differential equations. Theoretically, we may still be able to solve the equations analytically. There are some reasons we do not want to solve the equations analytically: (1) The order of the differential equations become so high that it is tremendously difficult to solve them analytically. (2) Even we can solve it; the derivation will be different for different topology. (3) When non-linear circuit elements are included, we basically do not have any way to solve them analytically. Apparently, numerical solution is necessarily needed for a general circuit topology. For this reason, SPICE simulators are invented for such a purpose.

The principle of SPICE simulation is quite straight-forward. As we can see above, the equations governing the voltages and currents on the circuit elements are differential equations with respect to time. If we divide the time into discrete time steps and use finite difference to approximate the time derivatives, we will get approximated equations as:

For a resistor, we have

$$V(n) = R I(n) \quad (1-7)$$

Where $V(n)$ and $I(n)$ are the voltage and current at the No.n time step, respectively.

For a capacitor, we have

$$I(n) = C \cdot [V(n) - V(n-1)] / dt \quad (1-8)$$

Where dt is the time difference between the No.(n-1) and the No.n time steps.

For an inductor, we have

$$V(n) = L \cdot [I(n) - I(n-1)] / dt \quad (1-9)$$

For a mutual inductor, we have

$$V1(n) = L11 \cdot [I1(n) - I1(n-1)] / dt + M12 \cdot [I2(n) - I2(n-1)] \quad (1-10)$$

$$V2(n) = M12 \cdot [I1(n) - I1(n-1)] / dt + L22 \cdot [I2(n) - I2(n-1)] \quad (1-11)$$

Apparently, the equations governing the voltages and currents on the elements are converted from differential equations to matrix equations after we use finite differences to replace the derivatives. For a general N-port circuit, the process leads to a general matrix equation,

$$[V(n)]_{NX1} = [Z(n)]_{NXN} [I(n)]_{NX1} + [Cn]_{NX1} \quad (1-12)$$

Where $[V(n)]_{NX1}$ is the voltage vector at the No.n time step; $[I(n)]_{NX1}$ is the current vector at the No.n time step; $[Cn]_{NX1}$ is a constant vector known at the No.n time step; $[Z(n)]_{NXN}$ is a constant matrix known at the No.n time step.

In some sense, at the No.n time step, the $[Z(n)]_{NXN}$ and $[Cn]_{NX1}$ are known from the $[V(i)]_{NX1}$ and $[I(i)]_{NX1}$ with $i = 0, 1, \dots, n-1$. When the $[I(n)]_{NX1}$ is known, we will be able to solve $[I(n)]_{NX1}$. From the $[I(n)]_{NX1}$ and the solved $[V(n)]_{NX1}$, we will be able to calculate the $[Z(n+1)]_{NXN}$ and $[Cn+1]_{NX1}$. Repeating

the above process, we will be able to find the relationship between $[V(i)]_{NX1}$ and $[I(i)]_{NX1}$ with $i = 0, 1, \dots$. They are basically the solution of the equations in the time domain.

Apparently, the accuracy of the solution is related to the accuracy of representing the derivatives as finite differences. If we use fine enough time steps, we will be able to approximate a derivative using a finite difference as accurate as possible.

The above discussion is just on circuits consisting of linear elements. We can extend the SPICE time domain simulation theory to non-linear circuit easily. For a non-linear circuit, we are basically dealing with the so-called IV curves. A typical example is a diode. For a diode, the relationship between the current and voltage is defined as the IV curve:

$$I = F(V) \quad (1-13)$$

Where the V and I are the voltage and current on the diode, respectively. We can write,

$$dI/dV = F'(V) \quad (1-14)$$

Where $F'(V)$ is the derivative of the current with respect to the voltage. It is known when the $F(V)$ is given. If we use the finite difference to replace the derivative, we will have,

$$I(n) - I(n-1) = F'[V(n-1)] \cdot [V(n) - I(n-1)] \quad (1-15)$$

Obviously, equation (1-15) is in the same form as the equation (1-12). We can solve similar equations for any circuit consisting of R, C, L, M and diodes.

A diode is a two-node non-linear element. There are two other important types of non-linear elements: bipolar transistors and CMOS transistors. They are 3-node non-linear elements. For a typical bipolar transistor, we have

$$I_b = F1(V_{be}) \quad (1-16)$$

$$I_c = F2(I_b, V_{ce}) \quad (1-17)$$

Where I_b is the current on the base node; V_{be} is the voltage between the base and the emitter; I_c is the current on the collector; V_{ce} is the voltage between the collector and the emitter. For a typical CMOS, we have

$$I_g = 0 \quad (1-18)$$

$$I_{ds} = F3(V_{gs}, V_{ds}) \quad (1-19)$$

Where I_g is the current at the gate; I_{ds} is the current at the drain; V_{gs} is the voltage between the gate and the source. V_{ds} is the voltage between the drain and the source. In practical applications, there might be a small capacitor between the gate and the source. For such a case, we will end up with

$$I_g = F4(V_{gs}) \quad (1-20)$$

However, we can always model it as equations (1-18) and (1-19) with a shunt capacitor. Equations (1-19) and (1-20) are, in fact, equivalent to equations (1-16) and (1-17) even though the functions may not be the same. Anyway, we will consider the equations (1-16) and (1-17) more general form.

Equations (1-16) and (1-17) are non-linear equations. We can not solve them analytically. Again, we need to convert them into differential equations.

$$dI_b = F1'_{V_{be}}(V_{be}) dV_{be} \quad (1-21)$$

$$dI_c = F2'_{I_b}(I_b, V_{ce}) dI_b + F2'_{V_{ce}}(I_b, V_{ce}) dV_{be} \quad (1-22)$$

Where $F1'_{V_{be}}(V_{be})$ is the partial derivative with respect to dV_{be} ; $F2'_{I_b}(I_b, V_{ce})$ is the partial derivative with respect to I_b ; $F2'_{V_{be}}(I_b, V_{ce})$ is the partial derivative with respect to V_{be} .

By using finite differences to replace 1st order derivatives, we can convert equations (1-21) and (1-22) into a set of linear equations. From the linear equations, we will be able to find the relationship between the V's and I's for every element in the circuit.

As it can be seen, the time-domain SPICE simulation is quite straight-forward. There is no complicated process involved. The key concept is that we try to use finite difference to convert derivatives and non-linear IV curves into matrix equations. We solve the matrix equations for the time domain responses of the circuit.

Section 1.2 Frequency-Domain Network Simulator

In the time domain, the equations to be solved are differential equations. In the frequency domain, it is different.

For a non-linear circuit, if our input signal is a sinusoidal at a specific frequency, all the signals involved will be sinusoidal signals with the same frequency.

In the frequency domain, we can write down the frequency response of different lumped circuit elements easily.

For a resistor,

$$V = I R \quad (1-23)$$

For a capacitor,

$$I = j \omega C V \quad (1-24)$$

Where ω is the angular frequency.

For an inductor,

$$V = j \omega L I \quad (1-25)$$

For a mutual inductor,

$$V1 = j \omega L11 \cdot I1 + j \omega M12 \cdot I2 \quad (1-26)$$

$$V2 = j \omega M12 \cdot I1 + j \omega L22 \cdot I2 \quad (1-27)$$

Apparently, the equations to be solved are all linear equations. For a given topology consisting of R, L, C and M's, we can finally end up with network parameters of the circuit as,

$$[V]_{NXN} = [Z]_{NXN} [I]_{NXN} \quad (1-28)$$

Where the $[Z]_{NXN}$ is called the Z-matrix of the circuit.

A user should be notified that the $[Z]_{NXN}$ in equation (1-28) is not the same as the $[Z(n)]_{NXN}$ in equation (1-12). The $[Z]_{NXN}$ is a complex matrix for a specific frequency f (or $\omega = 2 \pi f$). The $[Z(n)]_{NXN}$ is a real matrix at a specific time.

We can also represent the circuit as a Y-matrix,

$$[I]_{NX1} = [Y]_{NXN} [V]_{NX1} \quad (1-29)$$

Apparently, assuming $[Z]_{NXN}^{-1}$ exists, we can get,

$$[Z]_{NXN}^{-1} = [Y]_{NXN} \quad (1-30)$$

There are cases either the Z or Y-matrix does not exist. For example, a short circuit does not have a Y-matirx. An open circuit does not have a Z-matrix. Due to the singularities, matrix manipulation based upon Z or Y-matrix needs to be very careful. For this reason, we would rather use the s-matrix (or s-parameters) for numerical calculation. S-parameters are the so-called scattering parameters. They are defined as,

$$[b]_{NX1} = [S]_{NXN} [a]_{NX1} \quad (1-31)$$

with

$$[V]_{NX1} = [\sqrt{Zc}]_{NXN} ([a]_{NX1} + [b]_{NX1}) \quad (1-32)$$

$$[I]_{NX1} = [1/\sqrt{Zc}]_{NXN} ([a]_{NX1} - [b]_{NX1}) \quad (1-33)$$

or

$$[a]_{NX1} = [1/\sqrt{Zc}]_{NXN} ([V]_{NX1} + [Zc]_{NXN} [I]_{NX1}) \quad (1-34)$$

$$[b]_{NX1} = [1/\sqrt{Zc}]_{NXN} ([V]_{NX1} - [Zc]_{NXN} [I]_{NX1}) \quad (1-35)$$

Where the $[a]_{NX1}$ and $[b]_{NX1}$ are the incident wave and reflected wave vectors, respectively; The $[Zc]_{NXN}$, $[\sqrt{Zc}]_{NXN}$ and $[1/\sqrt{Zc}]_{NXN}$ are diagonal matrices; Zc is the so-called normalization impedance for a specific port. From the properties of diagonal matrices, we have

$$[\sqrt{Zc}]_{NXN}^{-1} = [1/\sqrt{Zc}]_{NXN} \quad (1-36)$$

From equations (1-28), (1-29), (1-32) and (1-33), we can obtain

$$[Z]_{NXN} = [\sqrt{Zc}]_{NXN} (I + [S]_{NXN}) (I - [S]_{NXN})^{-1} [\sqrt{Zc}]_{NXN} \quad (1-37)$$

$$[Y]_{NXN} = [1/\sqrt{Zc}]_{NXN} (I - [S]_{NXN}) (I + [S]_{NXN})^{-1} [1/\sqrt{Zc}]_{NXN} \quad (1-38)$$

$$[S]_{NXN} = [1/\sqrt{Zc}]_{NXN} ([Z]_{NXN} - [Zc]_{NXN}) ([Z]_{NXN} + [Zc]_{NXN})^{-1} [\sqrt{Zc}]_{NXN} \quad (1-39)$$

Where I is the unit matrix (diagonal elements take the value of 1 and off-diagonal elements take the value of 0). Equations (1-30), (1-37)-(1-39) indicate that the Y, Z and S-parameters are interchangeable. In practical numerical simulation, S-parameters are preferred because they do not have singularities for regular circuits. For example, the s-parameter of an open circuit is 1, and the s-parameter of a short circuit is -1.

For non-linear circuits, the standard frequency domain procedure does not work. The reason is that when an input signal of a specific frequency f_1 is applied to a non-linear circuit, it will create other frequency components than the one at f_1 . The components include those at frequencies $2f_1, 3f_1, 4f_1, \dots$. When an input signals of two specific frequencies f_1 and f_2 is applied to a non-linear circuit, it will create components at $f_1, f_2, f_2-f_1, f_2+f_1, 2f_1, 2f_2, 2f_1-f_2, 2f_1+f_2, 2f_2-f_1, 2f_2+f_1, \dots$. Not only many different frequency components are created, the magnitude and phase of each component is even not linearly proportional to those of the input signals. All the $f_2-f_1, f_2+f_1, 2f_1, 2f_2, 2f_1-f_2, \dots$ are the so-called harmonics.

Obviously, the frequency domain solution of a non-linear circuit becomes a non-linear equation with infinite dimensions. There is no simple solution for it. To solve non-linear circuits in frequency domain, people have implemented the so-called harmonic balance simulators to tackle the problem. The idea is quite straight-forward. We first separate a circuit into the linear part and non-linear part (see Figure 1.1). The frequency response of the linear part can be represented using s-parameters. When an input signal consisting of one or multiple frequency components is applied to the circuit, the output signals in the linear circuit part can be solved. When the signal is applied to the non-linear part, its output signal contains

components of the base frequencies as well as all the harmonics. The goal of harmonic balance is try to perform an iterative procedure to get a stabilized solution with matching components between the linear and the non-linear part.

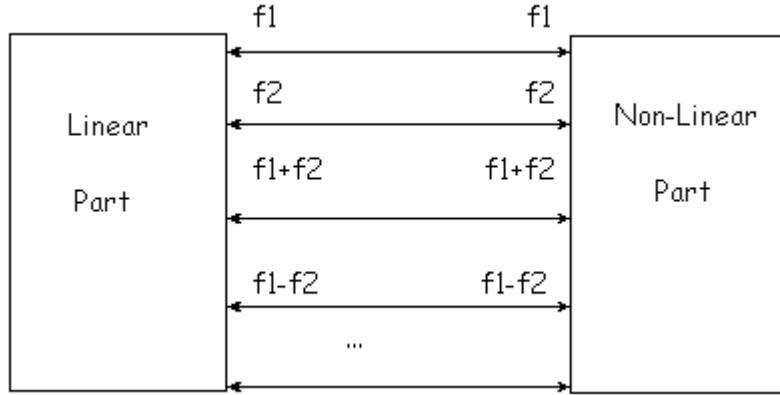


Figure 1.1 Harmonic balance illustration.

Apparently, the harmonic balance process is an optimization process with many unknowns. Theoretically, we have infinite number of unknowns. Practically, we consider the higher order components are much smaller and we can neglect them. The numerical solution becomes an optimization process with finite number of unknowns.

Section 1.3 Time-Domain Simulation vs. Frequency-Domain Simulation

In the above two sections, we have illustrated the principles of time-domain simulation and frequency domain simulation. The question is which simulation technique is better, time-domain or frequency domain. For linear circuits, frequency domain seems to be a better choice. The reasons are the following.

In the frequency domain, all the equations involved automatically become matrix equations. No approximation is involved. We may use the s-parameters at much less than one hundred frequency points to represent the circuit. For each frequency, we will solve a matrix once.

In the time domain, we have to finite differences to approximate derivatives. Approximation introduces errors. Also, the solution process in the time domain is a time-marching process. We need to find the time responses at tens, hundreds, thousands or even tens of thousands of points. For each point, we need to solve the matrix once. For a regular non-linear circuit, the frequency domain simulation should be much faster.

Another big advantage of frequency domain simulation over time domain simulation on linear circuit is that we can include s-parameter blocks into a frequency domain simulation easily. Frequency domain simulation is based upon s-parameters. If we have a sub-circuit and we know its s-matrix at all the frequencies of interest, we will be able to include the circuit into the simulation easily without loss of accuracy. On the other hand, time-domain SPICE simulation takes the R, L, C and M as the basic simulation elements. IV curves are handled using local approximation. Given a circuit consisting of R, L, C and M, we do not need to find its s-matrix, we can perform the SPICE simulation based upon the netlist of the R, L, C and M describing the circuit. The trouble is that standard SPICE can not handle a circuit described using s-parameter frequency response.

For low frequency circuits, we can always use R, L, C and M to represent them accurately. We do not need to care about s-parameters because we do not need to convert the R, L, C and M netlist into the s-

parameter frequency response. The problem is that, for high frequency circuits, we may not be able to provide accurate R, L, C and M netlist. At microwave frequency, a circuit is considered as a distributed circuit. We may not be able to use R, L, C and M to present a distributed circuit accurately.

For a typical microwave circuit, we use electromagnetic simulators (or frequency-domain microwave network simulators) to model it. We will get the s-parameter frequency response. If we want to get time domain frequency response on standard SPICE simulator, we have to convert the s-parameter frequency response into an R, L, C and M equivalent circuit. This process is called parameter extraction. The parameter extraction process is basically a multi-variable optimization process. Optimization process is an unsolved mathematical problem. We even do not know whether there is any solution or how many solutions exist. Even we get a solution, the solution is normally not perfect. It is only valid in some parameter range.

For the example of a transmission line, it is precisely described as

$$\frac{dz(z)}{dz} = -R i(z) - j\omega L i(z) \quad (1-40)$$

$$d i(z) / dz = -G v(z) - j\omega C v(z) \quad (1-41)$$

Where R is the series distributed inductance per unit length, L is the series distributed inductance per unit length, G is the shunt distributed conductance per unit length, C is the shunt distributed conductance per unit length.

Again, we can use finite difference to replace the derivatives in equations (1-40) and (1-41) to get two approximated equations,

$$V2 - V1 = -(R dz) I2 - j\omega (L dz) I2 \quad (1-42)$$

$$I2 - I1 = -(G dz) V1 - j\omega (C dz) V1 \quad (1-43)$$

The physical meaning of equations (1-42) and (1-43) is shown in Figure 1.2. When the dz is small, we can use equations (1-42) and (1-43) to approximate the equations (1-40) and (1-41). The transmission line circuit in Figure 1.2(a) is a distributed circuit. The equivalent circuit in Figure 1.2(b) is a lumped element circuit. We try to use the lumped element equivalent circuit (referred as LC-segment) to approximate the distributed circuit.

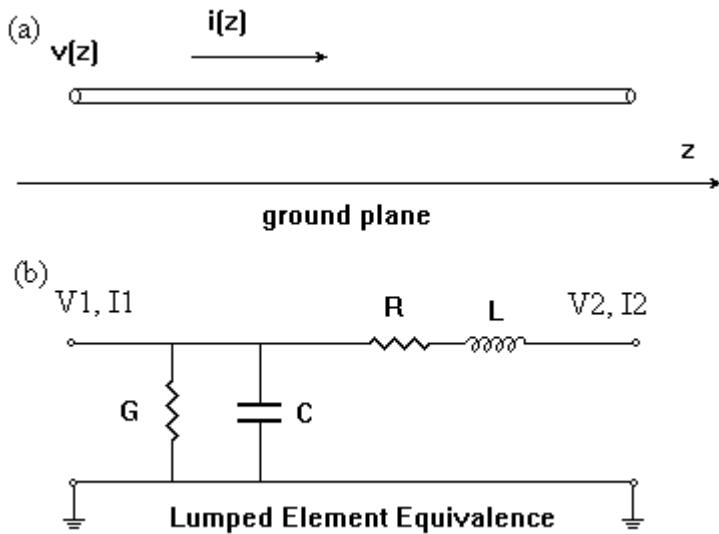


Figure 1.2 The lumped element equivalence of a short transmission line.

Under what conditions will the equations (1-42) and (1-43) be valid? The condition is that the dz is very small compared to wavelength of the highest frequency. When the length of the transmission line is not small compared to the wavelength, what can we do? A straight-forward solution is to use multiple LC segments to approximate the distributed circuit. As long as each segment is small compared to wavelength, the equivalence is valid. For an electrically long structure, we will expect to use many lumped elements in order to get a reasonably accurate equivalent circuit. For a complicated circuit, we may not even be able to find an accurate equivalent circuit. When we can not get an accurate lumped element equivalence of a circuit, we will not be able to use standard SPICE to perform a time-domain simulation on the circuit. On the other hand, it is not difficult to get accurate s-parameter frequency response of a complicated high frequency circuit using an electromagnetic simulator such as the IE3D package. By this point, the advantage of frequency-domain network simulator over time-domain SPICE simulator is quite clear in simulating linear circuit. The advantage of SPICE simulators is on the non-linear part.

Many microwave circuits can be considered as linear circuit. For digital circuits, they are almost exclusively non-linear circuit. As it can be seen, time-domain SPICE simulation on non-linear circuits is the same as it is on linear circuit. It is also of the simple form. No other approximation except replacing derivatives using finite differences is used. On the other hands, the harmonic balance process in frequency domain simulation in handling non-linear circuit is much more complicated than the frequency domain linear circuit simulation. Not only it is more complicated, it is also an optimization process. When optimization is involved, theoretically we can not guarantee that we have a solution and how good a solution will be. Another issue is the large number of unknowns resulted. For example, we have a circuit with one non-linear element. We apply a signal consisting of f_1 and f_2 . We will have the following frequency components to match: $f_1, f_2, f_1-f_2, f_1+f_2, 2f_1, 2f_2, 2f_1-f_2, 2f_1+f_2, f_1+2f_2, f_1-2f_2, \dots$ There are many components involved. In order to solve it accurately, we have to decide how which term is small and how many terms we want. When we apply 3 frequencies, we will end up with many more terms. If we have more than one non-linear element, each element may require the matching of the frequency components. Totally, we will end up with an optimization with many unknowns. Soon or later, we will not be able to solve it. Comparing the two methods, we can see time-domain SPICE simulation is much better than frequency domain simulation for non-linear circuits.

If we can combine the advantage of frequency domain network simulator on linear circuits and the advantage of time domain SPICE simulator on non-linear circuits into a single simulator, we will be able to come up with a more accurate, more efficient and more powerful tool.

Section 1.4 The MDSPICE Mixed Frequency-Domain and Time-Domain SPICE Simulator

The implementation of MDSPICE is inspired by the idea of combining the advantage of frequency domain network simulators and time-domain SPICE simulators. For a given circuit, we separate it into the linear part and the non-linear part, we can perform a frequency domain simulation on the linear part to get the s-parameter frequency response. Then, we can perform a time domain SPICE simulation for the whole circuit.

For the time domain simulation, every time step, we have to go through the netlist for the time response of every element. When the number of elements is big, we have to go through a large number of element and perform the time domain simulation on it. On the other hand, given a complicated topology, the complexity of s-parameters is only related to the number of ports. No matter how complicated a circuit is, the final s-parameters are only related to the total number of ports at some limited frequencies. If we can perform the time domain SPICE simulation based upon the s-parameters frequency response, we can save much computation effort. For distributed circuits, we even can not get the R, L, C and M circuit list, the only available information is the s-parameter frequency response. we have to perform time domain simulation directly on the s-parameter frequency response.

The time domain SPICE simulation on s-parameter frequency response is basically a convolution problem. It involves a complicated integration of time response over the complete time history. Accurate

and efficient time domain SPICE simulation based upon s-parameter is not a simple job. That might be the reason why traditional SPICE simulators do not have a solid implementation of it.

On the MDSPICE, we have implemented a solid time-domain simulation engine into it. Basically, the MDSPICE can take a s-parameter frequency response and convert it into time domain efficiently without loss of accuracy. The MDSPICE will accept the SPICE netlist consisting of R, L, C and M (or K). Besides, it will also accept s-parameter frequency response as an element. In the 1st release, we may only offer the linear part. We are going to implement the non-linear part as general IV curves. The non-linear part should be available soon because, as it is illustrated, the non-linear SPICE simulation, is basically the same as the linear part. As many people may know, any linear circuit can be represented as s-parameter frequency response. Any non-linear circuit can be represented as IV curves. After we implement the IV curves for non-linear circuits, the MDSPICE should be a complete tool for any type of accurate and efficient modeling.

The benefit of MDSPICE is multi-folded. First, it allows microwave engineers to perform accurate and efficient time domain analysis on microwave circuits. Time-domain analysis may reveal much information which frequency domain simulation can not provide. In fact, time-domain technique has been widely used in test equipment for calibration and other purposes.

The MDSPICE allows IC designers to perform accurate and efficient analysis on high speed digital circuits. Accurate analysis of high speed digital circuit using traditional SPICE simulator has been a difficult task due to the fact that accurate R, L, C and K models may not be available at high frequency. However, there is no difficulty to obtain the s-parameters of the circuit at very high frequency or clock rate. The capability of accurate and efficient time-domain simulation on s-parameters makes the MDSPICE a good choice for high speed digital circuit analysis.

Another problem facing traditional SPICE simulators is the netlist security. A digital circuit system consists of many different modules. Each module is designed and manufactured by a different company. In order to build the system, the manufacturer of each module must provide the SPICE model. Ideally, they should provide the netlist. The problem is that a SPICE netlist normally contains confidential information. How to present product information to customers without revealing confidential information has been a big issue in the digital design process. If we use s-parameters and IV curves to represent the design and use the MDSPICE to perform the SPICE simulation, we will be able to (1) keep the design topology confidential; (2) allow SPICE simulation for any input and loading condition. Given a SPICE netlist, we first separate them into a linear part and a non-linear part. The linear part can be simulated in the frequency domain to get the s-parameters frequency response. The non-linear elements are represented as IV curves. After the frequency domain simulation, we get the s-parameter frequency response that does not carry any topology information. It is safe to release the s-parameter frequency response to customers.

Another aspect the MDSPICE can help is the circuit reduction in the IC design. In an IC design, we can still represent a circuit as R, L, C and K and there is no need to use s-parameters. The problem is that large number of R, L, C and K elements are created. Traditional SPICE simulation on a large number of R, L, C and K is certainly not efficient. The idea of circuit reduction is that we try to use a circuit with fewer R, L, C and K elements to represent the circuit with larger number of R, L, C and K elements. As long as we can guarantee their time domain responses are close, we can use the circuit with fewer R, L, C and K elements in place of the circuit with larger number of R, L, C and K. The idea is certainly correct. However, how can we guarantee the circuit with fewer R, L, C and K will yield time response close to the circuit with large number of R, L, C and K. We will not be completely sure unless we perform a simulation on them and compare their responses. In the other word, we will know the two circuits have the same or similar time responses, we have spent the time to confirm it. Using MDSPICE, we can make the circuit reduction unnecessary. For a circuit with fixed number of port, we have the s-matrix of the same order. No matter how complicated a circuit's topology is, the resulting s-matrix is always of the same size. Given an R, L, C and K topology, the frequency domain simulation for the s-parameters should be much faster than the time-domain SPICE simulation. After the frequency domain simulation, we can save the s-parameter frequency response for further time domain SPICE simulation. When the number of ports is small, the time-domain SPICE simulation based upon the s-parameter frequency response takes no time.

In summary, the MDSPICE is a combination of frequency domain network simulator and a time domain SPICE simulator. It is not a simple integration of two different simulators. It really combines the advantage of both types of simulation to create a new type of simulator with higher efficiency, accuracy and capability. Starting from chapter 2, we will illustrate the use of the MDSPICE for different types of applications. Compared to other electromagnetic simulators, the MDSPICE is relatively simple and easy. Besides the MDSPICE.EXE, we also offer the utility program MODUA in the MDSPICE package. MODUA is basically a display program and a simple frequency domain network simulator. IE3D and FIDELITY users may have known that the MODUA is a very useful program. Before we implement all the features of MODUA into MDSPICE, we will include the MODUA as a utility program for MDSPICE.

Chapter 2. Simple Verification of the MDSPICE

In this chapter, we will demonstrate how the MDSPICE can be used to perform frequency domain and time domain simulation on linear circuits. As it is discussed in the chapter 1, it is normally more accurate and efficient to perform a simulation of a linear circuit in the frequency domain. Why we need to perform a time-domain SPICE simulation on a linear circuit? The answer is that time signal analysis may contain meaningful information that is hard to find from the frequency domain. For a linear circuit, time domain analysis is as important as frequency domain simulation.

Section 2.1 The Basic Setup in MDSPICE for a Simple Capacitor.

We start with a simple capacitor. We would like to use this simple example to verify the accuracy of the MDSPICE. The value of the capacitor is 1pF. The SPICE netlist is listed in Table 2.1. The netlist can be created using any ASCII editor such as the NOTEPAD or WORDPAT in the accessory of Windows 95/NT. The syntax is documented in the Appendix A. If you are not familiar with the SPICE netlist and the additional syntax of the MDSPICE, please read the Appendix A first. On the MDSPICE 2.X, we need to start from the SPICE netlist. Starting from MDSPICE 3.0, we can create the circuit easily from the integrated schematic editor.

Table 2.1 The SPICE netlist of a capacitor: RC1.LIB.

Line No.	Content	Explanation
1	Test on a RC circuit	The 1 st line is always a comment line on the netlist.
2	C1 1 2 1p	A capacitor C1 of - pf between node 1 and node 2
3	R1 2 0 50	A resistor R1 of 50-ohms between node 2 and ground.
3	.end	The notation for end of main circuit.

The netlist in RC1.LIB basically describes a RC circuit shown in Figure 2.1. The ground is denoted as “0” or “GND” in SPICE netlist. The sources and terminations (Vs1, Rs1, Vs2, and Rs2) are not included in the netlist. On MDSPICE, the user can define the sources and terminations before the simulation.

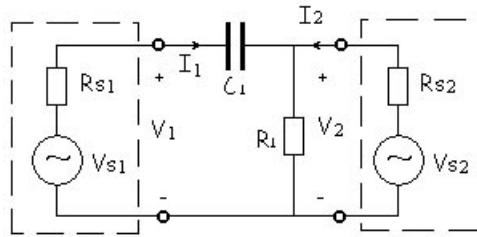


Figure 2.1 A capacitor is excited by a voltage source.

We are going to apply a voltage source (Vs1) defined in Table 2.2 to the circuit. Its waveform is shown in Figure 2.2 (each time step = 20 ps).

This is a simple example. We can solve the problem analytically. We use such a simple example to verify and introduce the MDSPICE. In the following, we will go through the detailed steps in simulating the simple capacitor using MDSPICE.

Table 2.2 The definition of the voltage source.

Formula	Domain
Vs = 0	T < 100 ps or T > 700 ps
Vs = (T-100) / 80 (V)	T > 100 ps & T < 180 ps
Vs = (T - 620) / 80 (V)	T > 620 ps & T < 700 ps

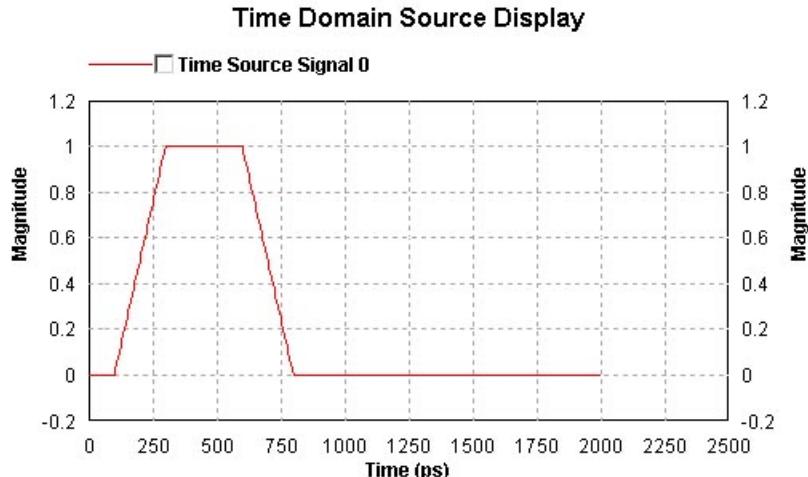


Figure 2.2 The input voltage waveform to be applied to the circuit.

- Step 1 Run IE3D Program Manager or ie3dpm.exe. IE3DPM is the security program. We have to run it before we run the application programs.
- Step 2 Run MDSPICE from the IE3D Program manager menu list.
- Step 3 We can create the circuit easily using the schematic editor on the MDSPICE 3.0. Since the circuit is created as a SPICE netlist, we can open or import it into MDSPICE. Select Open in File menu. MDSPICE prompts you for .mds file type.

Explanation:

The MDSPICE does not use the netlist file directly. It has its own format with the extension .mds. If we open a netlist (.lib) file, MDSPICE will automatically import the .lib file. There are 3 major types of files involved in a MDSPICE simulation: (1) the .mds file; (2) the .lib SPICE netlist file. Normally, we need the user to provide the .mds file. However, for compatibility consideration, the MDSPICE will accept the SPICE netlist files consisting R, L, C and K automatically. We are going to open the “C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\rc1.lib” file.

- Step 4 Change the directory to: “C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples”. Type “*.lib” in the edit field and enter to list the RC1.LIB file. Select the RC1.LIB file from the list.

Response:

The RC1.LIB file is opened and the content in the file is shown in Figure 2.3. The file RC1.LIB is a standard SPICE netlist. However, what is shown on MDSPICE 3.0 or in Figure 2.3 is not. It is special for MDSPICE. The MDSPICE netlist describes the connections between elements using the CONNECTION elements instead of nodes. In fact, it is less intuitive than the standard SPICE netlist. However, the CONNECTION based netlist is more for even more intuitive description of the circuit graphically. We have implemented schematic editing capability on MDSPICE and it is the reason why we choose the CONNECTION based netlist. The 1st line in the netlist is the comment. The 2nd line is the main circuit. We will not provide detail description on the MDSPICE netlist because it is really not necessary because we can display the circuit graphically.

- Step 5 Double clicking the 2nd line “S main P1(P1_T1) P2(P2_T1)”. MDSPICE will display two windows: the 1st window displaying the netlist and the 2nd window displaying the main circuit graphically (see Figure 2.4). We can see the main circuit is basically a capacitor, a resistor, a short-circuit (or ground) and two ports. This is basically the circuit topology we want to simulate.

Slightly different from what is shown in Figure 2.4, the values of the C1 and R1 will not be displayed on the circuit. To display the value of the C1 and R1, please select Display Detail Symbol in View menu and we will get the exact display in Figure 2.4.

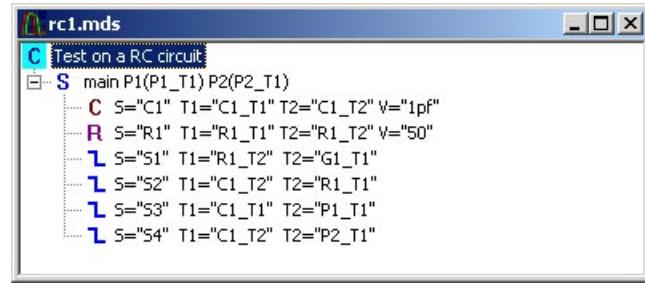


Figure 2.3 The display of RC1.LIB file on MDSPICE.

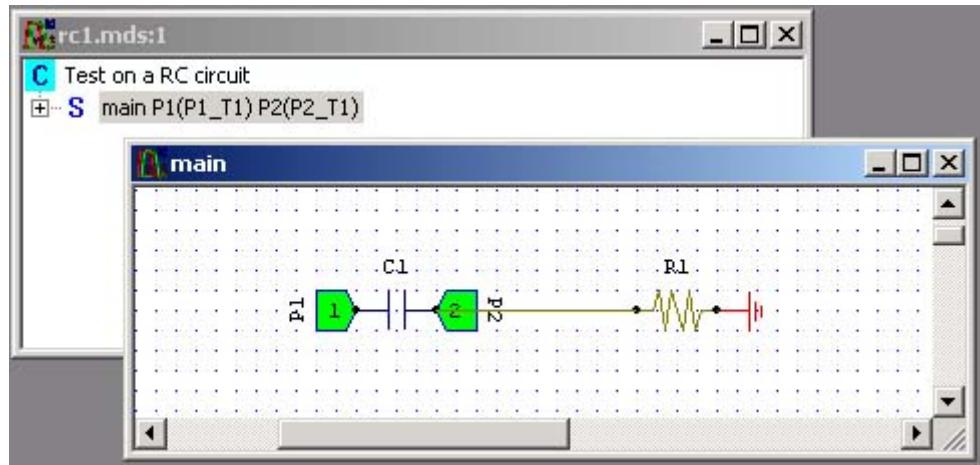


Figure 2.4 The graphic display of the main circuit in RC1.LIB.

Step 6 Select Save As and save the file as: C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\rc1.mds" file. We are going to simulate the structure in frequency domain.

Section 2.2 Defining Time Domain and Frequency Domain Parameters.

Step 1 Select Time Domain Parameters in Parameters menu. MDSPICE will prompt you for the time domain parameters. We will get the dialog similar to the one displayed in Figure 2.7. The default Time Step is 2.5 ps. Select Change by Factor button. Change the "Increase by a Factor" from "N/A" to "2" and select OK (see Figure 2.5). Basically, we want to increase the Time Step from 2.5 ps to 5 ps. The Time Step is the time segment for each step in the SPICE simulation.

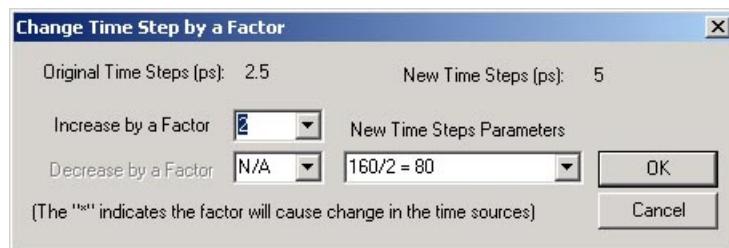


Figure 2.5 The Change Time Step by a Factor dialog.

Explanation:

By default, we set Time Divisions Per Minimum Period as 20. When the Time Step = 5 ps, MDSPICE automatically calculate the Highest Frequency of the time signals we should use is 10 GHz. If we go beyond it, we may end up with accuracy problem. In order to perform a time domain simulation with signals beyond the 10 GHz, we have to reduce the Time Step value.

- Step 2 Click at the Add button. MDSPICE will list all the different time domain waveforms available and prompt you to select the Source Type. Select the “One Clock Waveform” from the list and select OK. MDSPICE will prompt you for the “Single Clock Waveform” parameters.
- Step 3 Enter the parameters in the dialog as shown in Figure 2.6. There are two columns in the dialog. The 1st column is for the data entry. It is based upon the count of time steps. The 2nd column is for the real time value. For this example, we entered:

Start Time = 20 (counts of time steps) = 100 ps (for real time)
 Ramp Time = 40 (counts of time steps) = 200 ps (for real time)
 Pulse Width = 140 (counts of time steps) = 700 ps (for real time)
 Magnitude = 1

Another interesting parameter is the Rise Time. The Rise Time is defined as the time interval from 10% of the pulse height to 90% of the pulse height. It should be 80% of the Ramp Time. Therefore, when the Ramp Time = 40 (counts of time steps) = 200 ps, we have

Rise Time = 32 (counts of time steps) = 160 ps (for real time).

The magnitude is entered as 1. However, the value here is not the final one. The real magnitude for the excitation will also be determined by other factor.

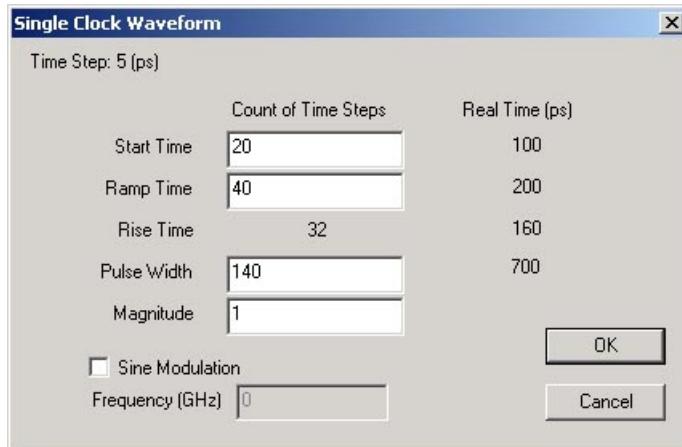


Figure 2.6 The Single Clock Waveform dialog.

- Step 4 Select OK to finish the One Clock Waveform definition. The waveform is listed in the Time Domain Parameters dialog box (see Figure 2.7). We can define more than 1 waveform. However, we just need to define for this example. Select OK to continue.
- Step 5 Select Frequency Domain Parameters in Parameters menu. You will be prompted to define the frequency range. For traditional SPICE, we do not need to enter the frequency parameters. For MDSPICE, we are working on both frequency and time domains. In order to solve the time domain response, we first need to solve the frequency domain response of the circuit first. The frequency range should cover the spectrum of the time signals to be used. For our case, we have

defined the Time Step = 5 ps with corresponding Highest Frequency = 10 GHz. Also, the input pulse contains DC component. At least, we should cover the frequency range from 0 to 10 GHz. We will define 201 frequency points from 0 to 10 GHz.

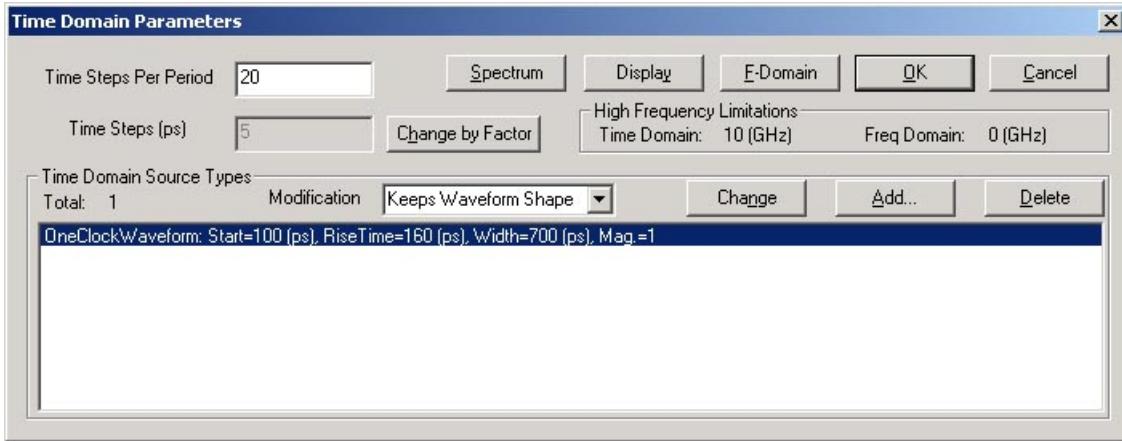


Figure 2.7 The Time Domain Parameters after the single clock waveform is defined.

- Step 6 Enter “0” for the Start (GHz) , “10” for the End (GHz) and “201” for the Number. Select Enter to continue. Total 200 frequency points starting from 0.05 to 10 GHz are entered into the list.

Explanation:

The frequency responses of some circuits elements may have singularities at 0 frequency (or DC). Normally, we avoid frequency simulation at 0 frequency even though we allow you to define a frequency very close to DC. On the MDSPICE, we allow you to enter multiple ranges of frequencies if necessary. For this example, we just need a uniform frequency from 0.05 to 10 GHz.

- Step 7 Select OK to finish the Frequency Domain Parameters.

Explanation:

We have defined all the necessary parameters. We are ready to perform a Frequency Domain Simulation or Time Domain Simulation. In fact, if we just want to perform a frequency domain simulation, we do not need to define the Time Domain Parameters. For the time domain simulation, it is always necessary to define the Time Domain and Frequency Domain parameters.

To perform a frequency domain simulation, please select Frequency Domain Simulation in Process menu. The MDSPICE will perform a frequency domain simulation on the SPICE netlist. After the simulation, it will store the frequency response inside the program. You can select the Save S-Parameters in File menu to save the frequency domain result.

To perform a time domain simulation, you can select the Time Domain Simulation in Process menu. It is always suggested that a user performs the Source Spectrum Analysis in Process menu before he starts a Time Domain Simulation. The Source Spectrum Analysis is not necessary. It is just for you to make sure the spectrum of the source signals falls into the range of the frequency range defined in Frequency Domain Parameters. If a large portion of the spectrum is outside the frequency range defined in Frequency Domain Parameters, the accuracy of our time domain simulation will be affected.

Section 2.3 Time Domain Analysis and Spectrum Analysis on the Sources.

- Step 1 Select Time Domain Sources in Display menu. MDSPICE will list the sources for the user to choose for the time domain display. Click at the check-box for the only time source, or the “OneClockWaveform...” to select it (see Figure 2.8). Select OK to continue. The time-domain display of the source is shown in Figure 2.2. Close the time-domain source display window.

Step 2 Select Source Spectrum Analysis in Process menu. Enter the frequency range for the spectrum as Start Frequency = 0, End Frequency = 35 GHz and Number of Frequencies = 141. Select OK to continue. The MDSPICE performs the Fourier transform on the source waveform. Then, it displays the results in a dialog box shown in Figure 2.9.

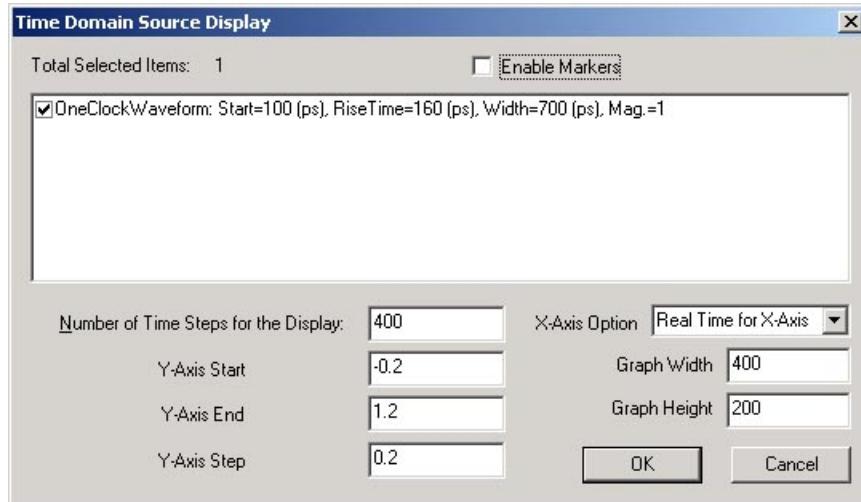


Figure 2.8 The selection dialog for displaying time domain sources.

Explanation:

For the No.0 source or the only source defined for this circuit, 99.9902% of the power spectrum is inside 0 to 35 GHz. Please note that the “Inband Power” is selected as “99.7%” by default. MDSPICE shows that 99.7% of the power spectrum is between 0 and 3.5 GHz. If you want to know the frequency range for a specified amount of power spectrum, you can select a pre-defined value in “Inband Power”. MDSPICE will show you the bandwidth containing the specified power spectrum.

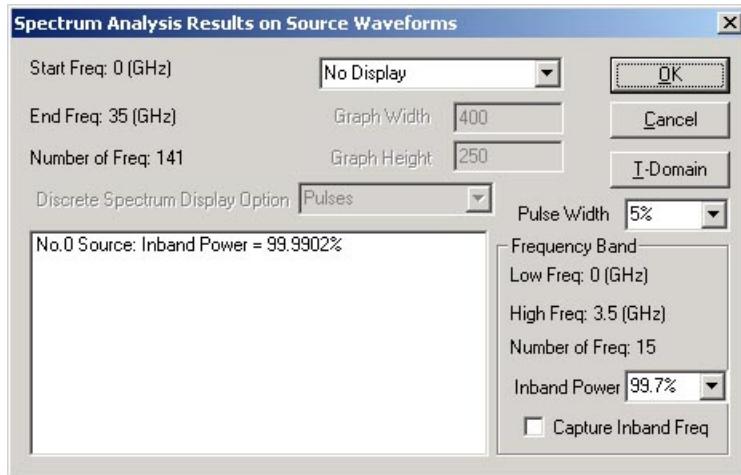
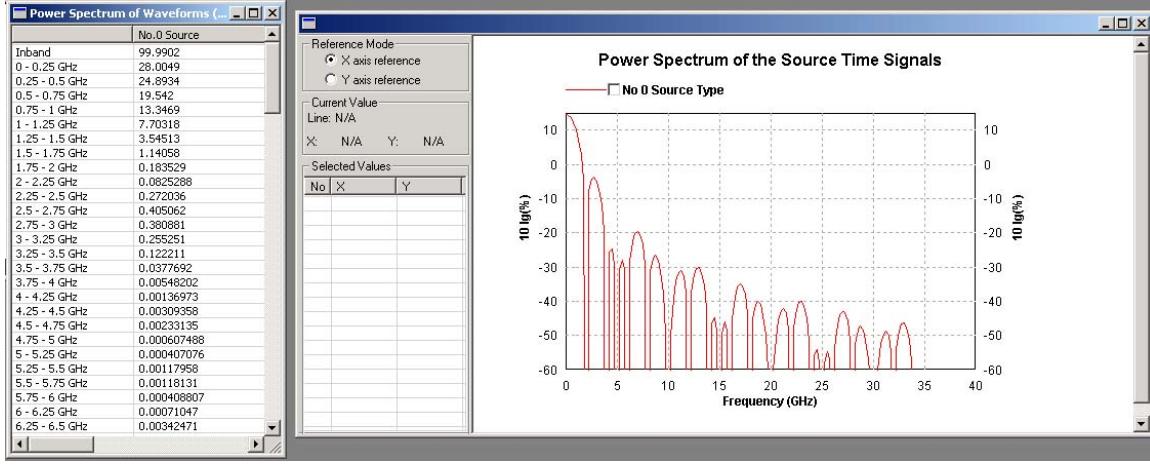


Figure 2.9 The spectrum analysis results on the source waveforms.

Another feature is that you can choose 1 of the 4 options for the spectrum analysis: (1) No Display; (2) Spectrum in Table; (3) Spectrum in Graph; (4) Spectrum in Graph (Log). By default, the option is No Display. If you choose Spectrum in Table, MDSPICE will display the spectrum distribution in a table (see Figure 2.10a). If you choose Spectrum in Graph, the MDSPICE will display the spectrum as curves in a cartesian coordinate system. If you choose Spectrum in Graph

(Log), the MDSPICE will display the spectrum as curves in log scale in a cartesian coordinate system (see Figure 2.10b). For the particular OneClockWaveform defined, 99.7% of the power is below 3.5 GHz. We can feel safe to use the source for our time domain simulation because we have defined 200 frequency points between 0.05 and 10 GHz in the Frequency Domain Parameters.



Starting from MDSPICE 3.0, a user is allowed to define markers in the Cartesian graphs and measure the values at and between the markers. We will not show the features. You can try it by yourself.

Section 2.4 Frequency-Domain Simulation and Display.

- Step 1** Select Frequency Time Domain Simulation in Process menu. MDSPICE will perform a frequency domain simulation on the circuit from 0.05 to 10 GHz. After the simulation, MDSPICE will prompt you to define the graphs for display (see Figure 2.11). There are 3 sections in the dialog: The 1st section, “Graphs to be displayed” allows a user to define multiple display graphs for the s-parameters in the 2nd and 3rd sections. The 2nd section contains the Internal S-parameters for the main circuit and all the sub-circuit involved in the design. The 3rd section contains the External S-parameters. A user is allowed to add external s-parameter files in Touchstone format into the dialog for simultaneous display of internal and external s-parameters for comparison purposes.

Step 2 Select the “New” button above the “Graphs to be displayed” section. MDSPICE will prompt you for the Graph Type. Select S-Parameters and select OK. MDSPICE will prompt you for the “Display Selection”.

Step 3 Select the “Root=Main Circuit” in the 1st list box. Check the “dB” for “S(1,1)” and “S(2,1)”, and “Ang” for “S(1,1)” and “S(2,1)” for 4 items (see Figure 2.12). If we have multiple s-parameters, we can select them from the 1st list box and select its items for display. For this circuit, we have only one set of s-parameters and we want to display its S(1,1) and S(2,1). Select OK to continue. The Parameter Display dialog shows that there is one graph “No.1: Title = S-Parameter Display, Total = 4 curves”. Select OK. A new window is opened and it displays the s-parameter graph (see Figure 2.13). We are displaying dB[S(1,1)], dB[S(2,1)], Ang[S(1,1)] and Ang[S(2,1)] on one graph. The 4 items use different scales and units. The scale on the left is using the unit of dB and it is for dB[S(1,1)] and dB[S(2,1)]. The scale on the right is using the unit of degree and it is for Ang[S(1,1)] and Ang[S(2,1)]. The user is allowed to define some markers for constant X or constant Y on some curves. The X and Y-coordinates for the markers are listed on the left hand side. The user can find the differences (dX and dY) between any two markers (see Figure 2.13).

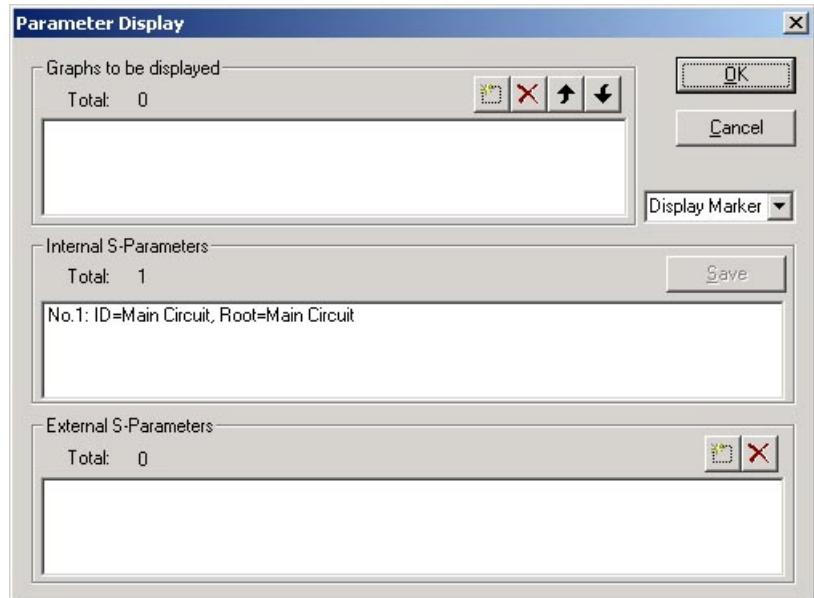


Figure 2.11 The s-parameter display dialog.

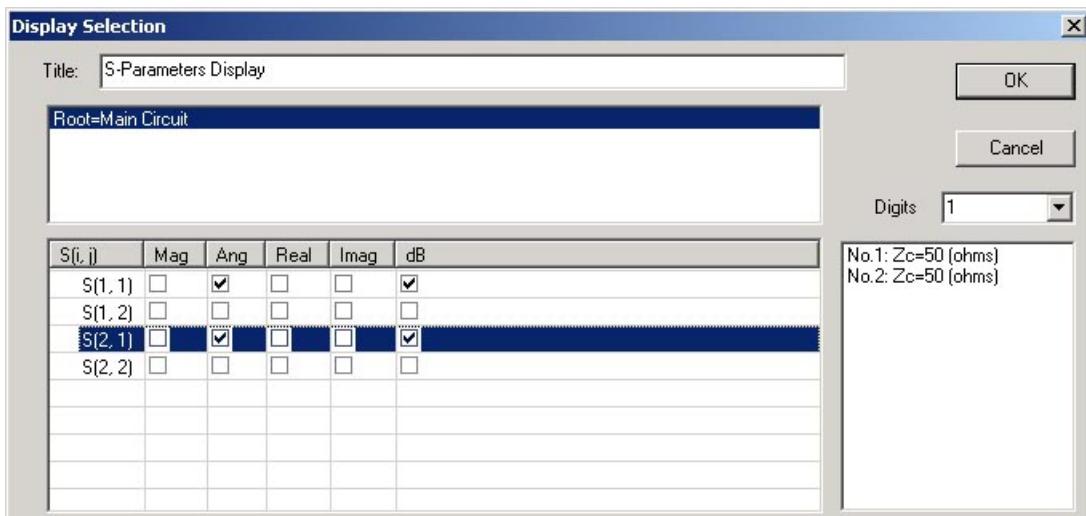


Figure 2.12 The dialog for selecting the s-parameter items for display.

Section 2.5 Time-Domain Simulation and Display.

A user should understand that frequency domain simulation is always necessary. However, explicit frequency domain simulation is not necessary. In the other word, we may not need to perform the frequency domain simulation explicitly as discussed in Section 2.4. We can go directly into section 2.5 and perform the time domain simulation. However, when we do the time domain simulation, the MDSPICE will perform a frequency domain simulation internally anyway. In this section, we will discuss the time-domain simulation.

Step 1 Select Time Domain Simulation in Process menu. The SPICE Simulation Setup dialog comes up (see Figure 2.14). If you have not performed a frequency domain simulation, the MDSPICE will perform the frequency domain simulation first and come up with the SPICE Simulation Setup dialog after the frequency domain simulation.

Traditional SPICE allows users to define the excitation and termination in the SPICE netlist. The MDSPICE does not do it. The reason is that we can perform a frequency domain

simulation independent of the excitation and termination. We can define the excitation and termination at the time we perform the time domain simulation.

On the MDSPICE, you can define multiple sets of excitations. Each excitation can have different types of sources and internal resistors for the ports. For an N-port structure, we can define N-sets of linearly independent excitations. Perform the time domain simulation on the N-sets of excitations. Then, we can extract the s-parameters frequency response out from the time-domain simulation results. We can compare the frequency response extracted from the time-domain simulation results and the frequency response from the frequency domain simulation for consistency test. Normally, we do not expect them to be completely the same. However, they should be close.

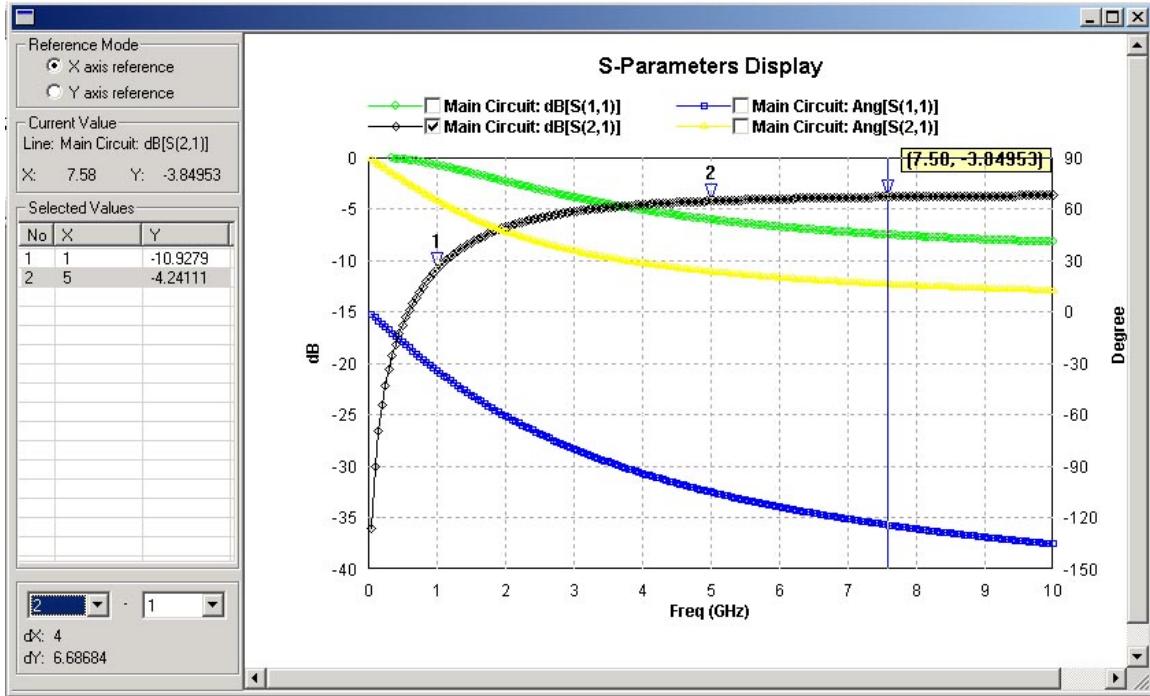


Figure 2.13 The s-parameters of the RC circuit and the markers.

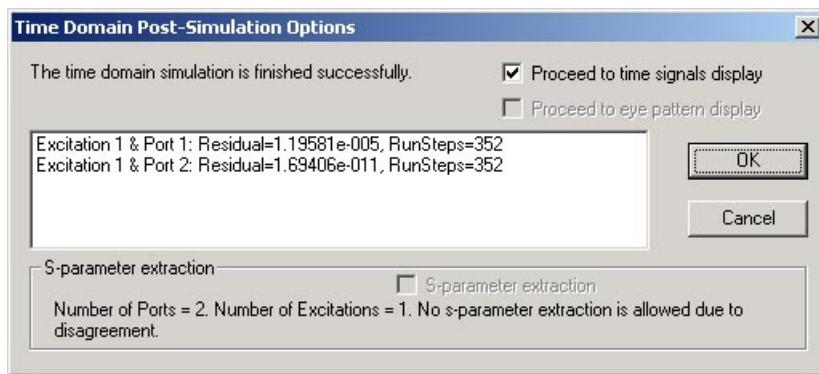


Figure 2.14 The Time-Domain Simulation Setup dialog.

- Step 2 Select the “No.1 Port: VoltageSource...” from the list. Select the “No.1 Port: VoltageSource...” from the list box. Enter Source Amplitude = 1 and Internal Resistor = 10. Select the “No.2 Port: Voltage Source...” from the list box. Enter Source Amplitude = 0 and Internal Resistor = 0. We should see the exact picture as shown in Figure 2.14. Basically, we have defined are Vs1 = 1, Rs1

$= 10$ ohms, $Vs2 = 0$, $Rs2 = 0$ ohm in Figure 2.1. We are exciting the port 1 and see the responses of the circuit at port 1 and port 2.

In Figure 2.14, MDSPICE shows you the Source Spectrum = 99.9811% InBand and it is above the warning limit of 99.7%, which is arbitrarily defined. Majority of the excitation source power spectrum is within the frequency ranges defined for this simulation. We should feel safe to perform the time domain simulation. MDSPICE is a s-parameter, frequency domain based time domain simulator. We need to make sure the bandwidth of the frequency domain s-parameters cover the spectrum of the source in order to yield high accuracy results.

There are other options in the setup dialog. By default, the “Pre-Process” option is “Divided”. It means that we will treat to perform the simulation based upon individual s-parameters in each sub-circuit. Assume we have some R, L, C and K, as well as some s-parameter file elements in a sub-circuit. Each s-parameter file will be treated as one s-parameter element. However, all the R, L, C and K will be grouped together and will be simulated for the s-parameters to be considered as one s-parameter element in the MDSPICE time-domain simulation. Simulating a sub-circuit as individual s-parameters normally can preserve high accuracy time domain simulation and it is suggested. For the other hand, the option “Combined” will try to get the s-parameters of the complete circuit and perform the time-domain simulation on the final s-parameters. Normally, its accuracy is not as good as the “Divided” option. Users are always suggested to use the “Divided” option.

The “Max Length” option of the simulation of coupled transmission lines (TLN). Starting from MDSPICE 3.0, we have built in frequency dependent, lossy or lossless coupled TLN models for high accuracy time-domain simulation. Internally, we also want to divide the coupled TLN as smaller sections for higher accuracy. The “Max Length” defines the maximum electrical length in terms of waveguide wavelength we will use to divide the coupled TLN for time domain simulation. Typical range of “Max Length” is from 0.25 to 0.5.

Starting from MDSPICE 3.0, we offer two time domain schemes, “Regular” and “Advanced”. The “Regular” scheme is the scheme implemented in the MDSPICE 2.X. It is very robust and it is reasonably high accuracy. However, it normally requires us to divide the long TLN into smaller segments in order to maintain high accuracy and meet the causality conditions. On the MDSPICE 3.0, we have implemented the “Advanced” scheme. The “Advanced” scheme is also a very stable scheme and it offers improved accuracy. Normally, it does not require the electrical length of the s-parameters to be very small and its results normally meet the causality condition very well. The user is suggested to use the “Advanced” scheme in normal condition.

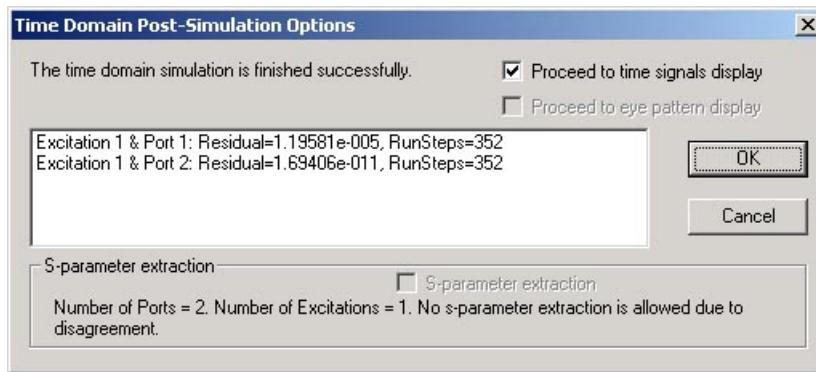


Figure 2.15 The Time-Domain Post-Simulation Options dialog.

MDSPICE automatically detects the minimum time steps and assume the maximum time steps the simulation may need to perform. For normal simulation, the user does not need to change it. However, for ultra long TLN, the delay from the input to the output may be very large. If the

value for minimum time steps is too small, MDSPICE may detect a fake convergence before the time signals reach the outputs. The user can increase the minimum time steps appropriately for the simulation of ultra long TLN. The Convergence Criteria are defined as “0.001”. It means that the MDSPICE will consider the simulation results are converged when the residual is less than 0.1% of the peak value. The Divergence Criteria is defined as “10000”. When the signals are more than 10000 times higher than the source’s peak value, we will consider the simulation is diverging and the simulation will be aborted.

- Step 3 Select OK to start the simulation. MDSPICE will take no time to finish the simulation and show you the Time-Domain Post-Simulation Options dialog (see Figure 2.15). It shows you that the residual for the Port 1 for the No.1 Excitation is $1.19581e-5$ and the residual for the Port 2 of the No.1 Excitation is $1.69406e-11$. The simulation takes 352 time steps. The low residuals are indications of convergence.

Section 2.6 Post-Processing on Time-Domain Simulation Results.

- Step 1 Make sure the “Proceed to time domain display” is checked. Select OK to continue. The MDSPICE will prompt you to select the display items (see Figure 2.16). Select the “Voltage” and “Current” for “No.1: Port 1”. Select the “Voltage” and “Current” for “No.1: Port 2”. The “No.1” is means it is the No.1 Excitation. For this simulation, we have only one excitation. We want to display the voltage and current at both ports 1 and 2 for the excitation.

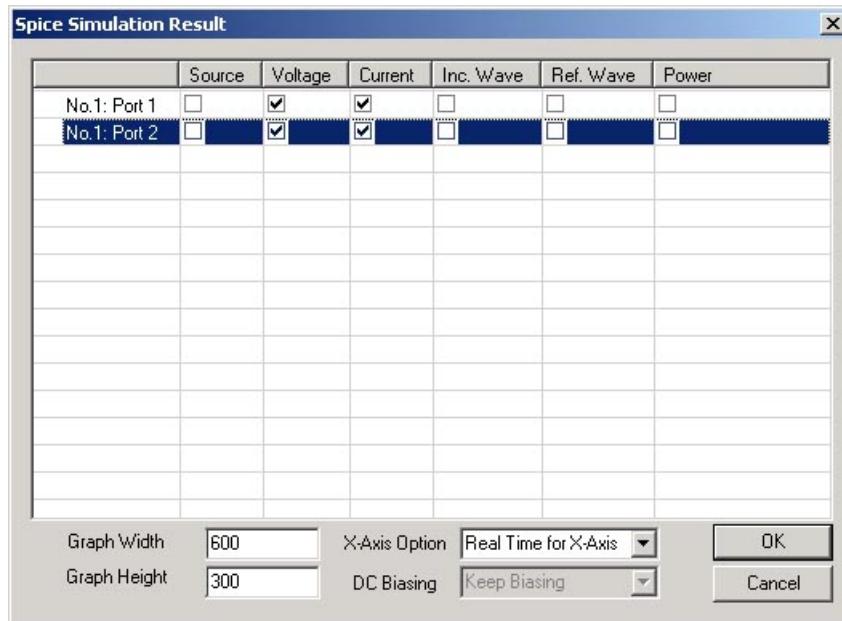


Figure 2.16 The dialog for selecting the display items.

- Step 2 Select OK and the MDSPICE will display a graph showing the V1, I1, V2 and I2 (see Figure 2.17). We are showing both the V and I and they are using different scales: V’s on the left scale and I’s on the right scale. It is interesting to note that $V2 = 0$ because the $Vs2 = 0$ and $Rs2 = 0$. It is basically short-circuited in port 2. The I2 is out of phase compared to I1 because its direction is opposite to that of I1.
- Step 3 Select Browse Graph Data in View menu. MDSPICE will invoke Internet Explorer to list the data for the time-domain signals. The user can get the data for any data on the graphs of MDSPICE by selecting Browse Graph Data in View.

Step 4 Select Time-Domain Result Spectrum in Display menu. MDSPICE will prompt you for the frequency range. Select OK to accept the default range from 0 to 12 GHz for 121 steps. Select OK. MDSPICE will perform the spectrum analysis for the results. Then, it will prompt you to select the items to be displayed.

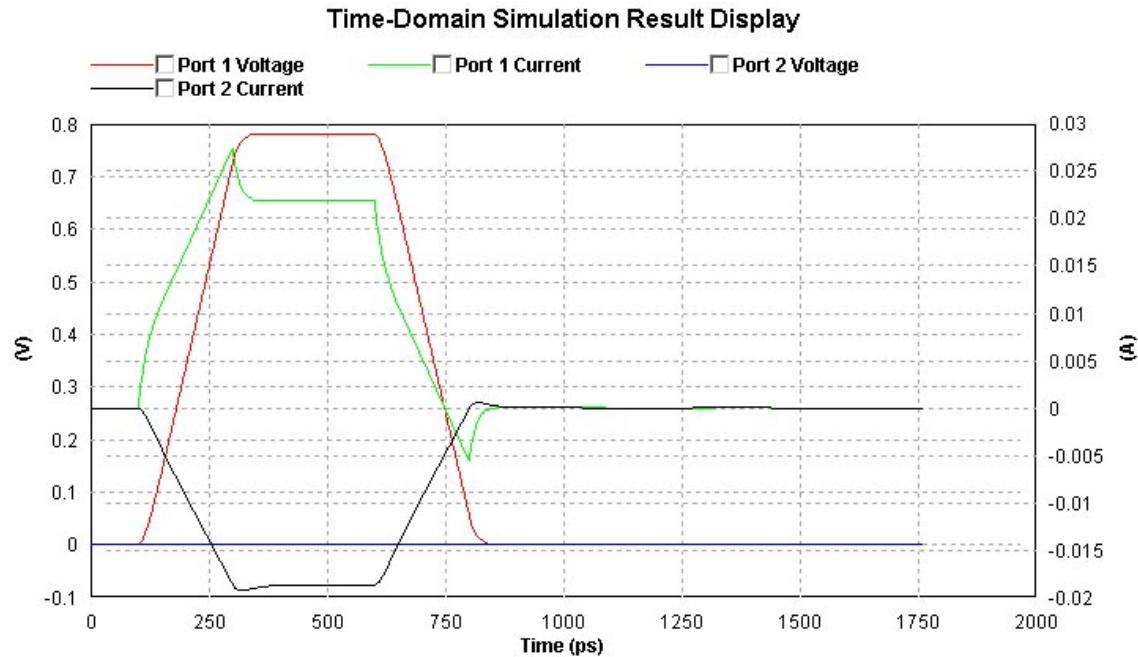


Figure 2.17 The Time-Domain Simulation Results.

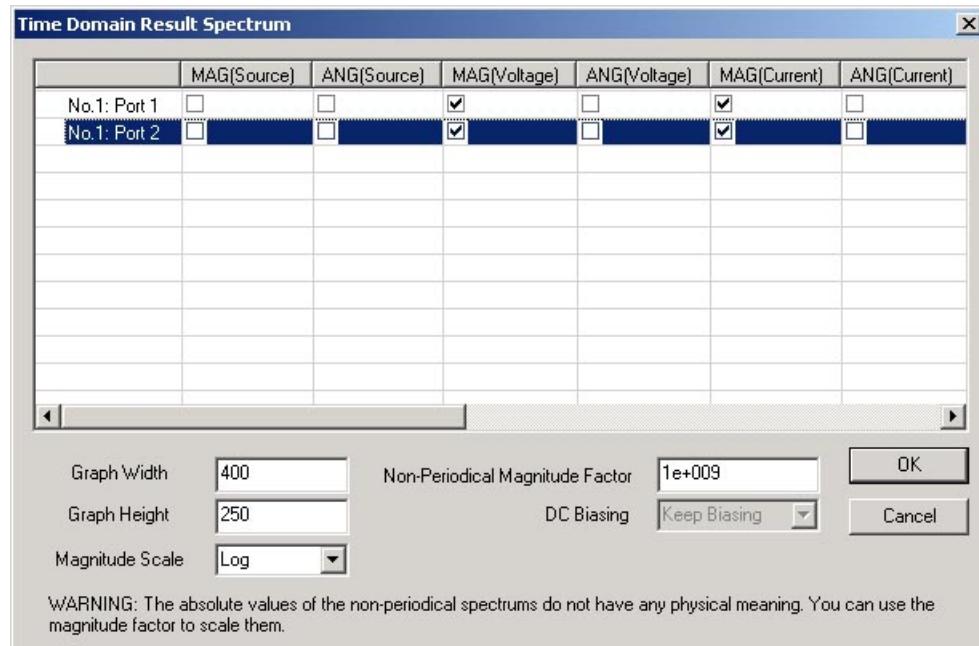


Figure 2.18 The spectrum display selection dialog.

Step 5 The results we are going to display are the frequency domain results. We can display both magnitude and phase. Please select MAGNITUDE(Voltage) and MAGNITUDE(Current) for both ports 1 and 2 (Figure 2.18). Select OK. MDSPICE will display the spectrum of the signals in

Figure 2.19. Again, the V's are using the left scale and the I's are using the right scale. The V2 is out of range. Its dB value should be negative infinity. We set it to -2000 in the display.

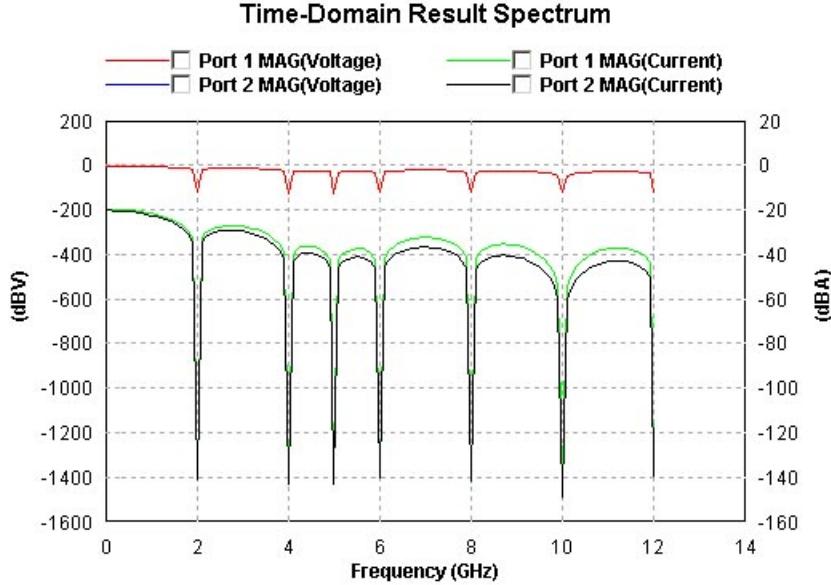


Figure 2.19 The spectrum display for the signals.

Section 2.7 Self-Consistency Checking.

The time-domain results we got in Section 2.5 look very good. However, a user should understand that the SPICE simulation is not based upon traditional formulas. The time domain simulation is not applied to the R and C elements directly. Instead, we simulate the R and C in frequency domain to get the s-parameters of the circuit. Then, we perform the time domain simulation on the resulting s-parameters. The question is whether we can prove the results are accurate. Certainly, we can compare the results with traditional SPICE simulation results and we will know the results are accurate. However, it is not necessary to do that because the MDSPICE is a self-consistent simulator. Apparently, we can compare the s-parameters of the RC circuit obtained directly from frequency domain simulation, and the s-parameters based upon the MDSPICE time-domain simulation. If the s-parameters from the two ways agree, it will be a proof for the accuracy of the time-domain simulation.

The circuit under investigation is a 2-port circuit. We need 2 linearly-independent excitation combinations to solve the s-parameters.

- Step 1 Select Time-Domain Simulation in Process menu. Select the ADD button to add the 2nd excitation. The default 2nd excitation is: No.1 Port, Voltge Source, Mag=0, R = 50 and No.2 Port: Voltage Source, Mag = 1, R=0. Select OK multiple times until the MDSPICE starts the 2nd simulation. It will finish the two SPICE simulations on two different excitations in less than one second. It will show the residuals for the Excitation 1 & Port 1, Excitation 1 & Port 2, Excitation 2 & Port 1, and Excitation 2 & Port 2. All of them are below 1e-5 and good convergence is achieved. The user can select OK to do the display. However, this time, we are not interested in the time domain signal display. We will try to extract the s-parameters of the circuit based upon the time domain responses of the two linearly independent excitations.
- Step 2 Uncheck the “Proceed to time domain display”. Check the S-Parameter Extraction. MDSPICE will go directly to s-parameter extraction and prompt you the 200 frequency points from 0.05 to 10 GHz. Select OK. MDSPICE will prompt you the S-Parameter File Name. Check Display Extracted S-Parameters (see Figure 2.20). Then, select OK to accept other default settings. MDSPICE will perform the extraction. After the extraction, it will save the data into file:

“C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\rc1.sp”. Then, it will add the “rc1.sp” file into the External S-Parameters list and invoke the Display S-Parameters dialog. You will notice that the frequency domain simulated s-parameter is listed in the Internal S-Parameters list as “No.1: ID=Main Circuit, Root=Main Circuit”, while the time-domain extracted s-parameter is listed in the External S-Parameters list as “No.1: ID=Extracted, Root=Extracted” (see Figure 2.21).



Figure 2.20 The S-Parameter File dialog.

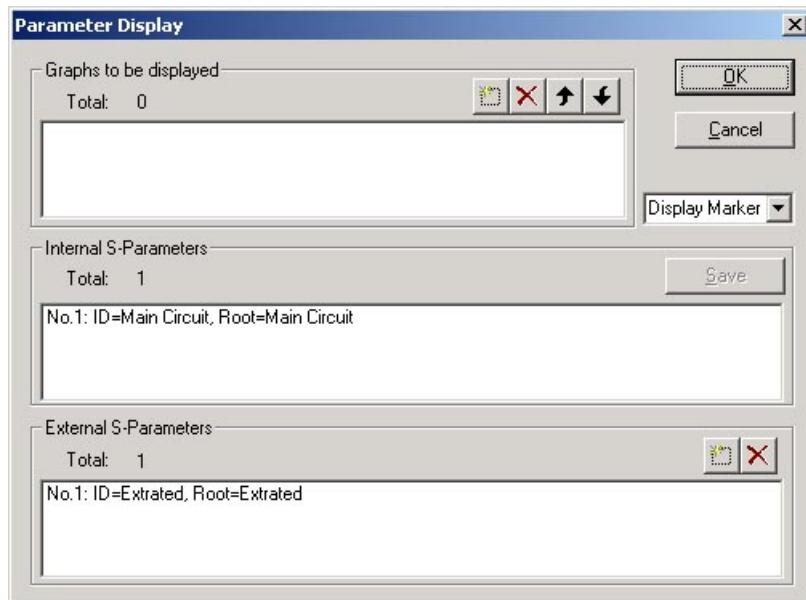


Figure 2.21 The Display S-Parameters dialog.

- Step 3** Select the New Button on the top right of the list box for “Graphs to be displayed”. MDSPICE will prompt you for the Graph Type. Select “S-Parameters” and select OK. MDSPICE will prompt you to select the display items for the available s-parameters. There are two s-parameters available: (1) Root=Main Circuit; (2) Root=Extracted (see Figure 2.22). Please select the “Root=Main Circuit” and check the items for dB and Ang of S(1,1) and S(2,1). Select the “Root=Extracted” and check the items for dB and Ang of S(1,1) and S(2,1) (see Figure 2.22). Select OK. The graph will be added into the list for “Graph to be displayed”. It shows “No.1: Title = S-Parameters Display, Total = 8 Curves”. Select OK to continue. MDSPICE will compare the frequency domain s-parameters “Main Circuit” and the time domain s-parameters “Extracted” in one graph. (see Figure 2.23). As you see, it is a perfect agreement between the two s-parameters except at some isolated frequency points. There are some sudden jumps at some isolated frequency points. They are caused by the fact that the time domain waveform, or stimulus has 0 spectrum density at those frequency points. The extraction is not valid at those points with 0 spectrum density. By now, we can say that the time-domain responses we got in Figure 2.17 must be accurate ones. If they are not accurate, we should not get the perfect agreement between the two s-parameters obtained using different methods.

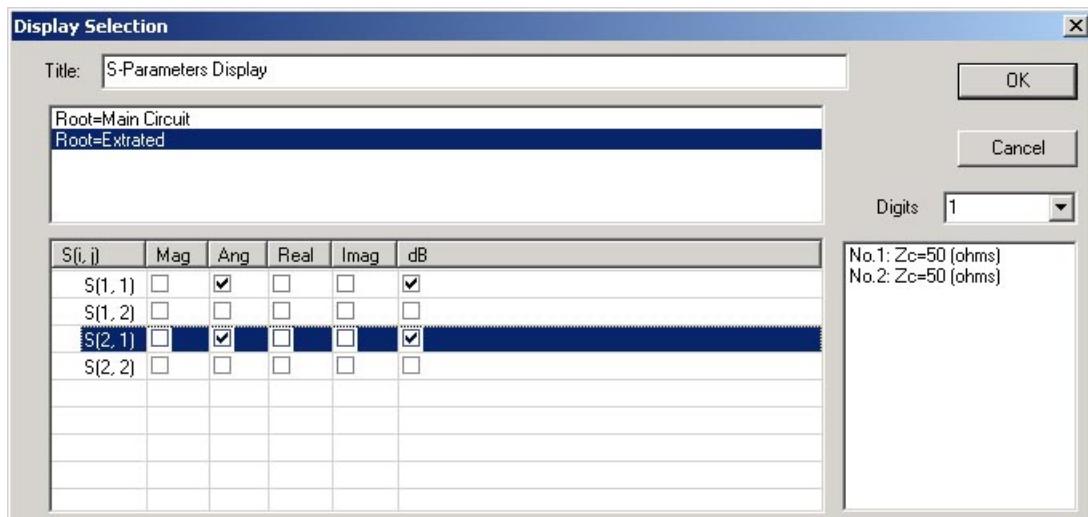


Figure 2.22 The Display Selection dialog for selecting the display items.

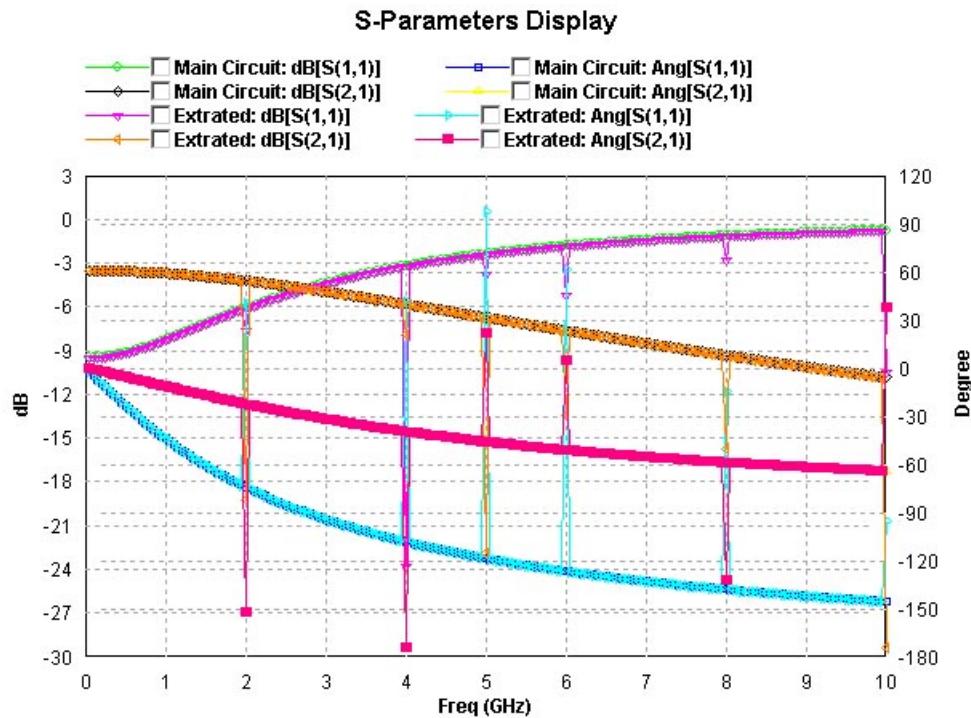


Figure 2.23 The comparison between the frequency domain and time domain s-parameters.

Chapter 3. Modeling of Frequency Dependent Circuits and Transmission Lines

In the last chapter, we have demonstrated the frequency-domain and time-domain simulation on a simple capacitor circuit. As time-domain simulation is concern, all SPICE simulators are supposed to do a good job on it. We used the example to illustrate the steps in using the MDSPICE and to verify the accuracy of the MDSPICE time-domain simulation.

In this chapter, we will demonstrate the applications of MDSPICE in modeling lossy transmission lines. We will compare the difference between the SPICE simulation using RLC netlist and the SPICE simulation using the s-parameter frequency response. From the comparison, we will prove the advantage of the MDSPICE simulator over traditional SPICE simulators.

Section 3.1 Modeling of a Lossy Transmission Line in Frequency Domain.

The structure we are going to model is a coupled stripline differential structure shown in Figure 3.1. The bottom surface to the bottom ground distance is $h_1 = 10$ mils. The trace thickness is $h_2 = 1.4$ mils. The top surface to the top ground distance is $h_3 = 8$ mils. There are three dielectric layers involved. The h_1 layer has the dielectric properties $\epsilon_1 = 4.3$ and $\tan\delta_1 = 0.002$. The h_2 layer has the dielectric properties $\epsilon_2 = 4.1$ and $\tan\delta_2 = 0.002$. The h_3 layer has the dielectric properties $\epsilon_3 = 4.3$ and $\tan\delta_3 = 0.002$. The grounds and the traces have the conductivity of $2e+7$ S/m. The cross-section of each trace is not of the rectangular shape. The bottom width of the trace is 7 mils, and the top width of the trace is 6.4 mils. The length of the transmission line is 500-mils. It is about 0.45 waveguide wavelength at 20 GHz. We are going to put a 100-ohm load and a capacitor of value to be determined at the end of the load (as shown in Figure 3.2). The differential stimulus is shown in Figure 3.3.

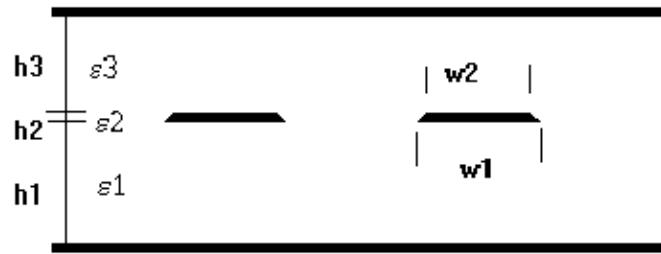


Figure 3.1 The coupled stripline structure.

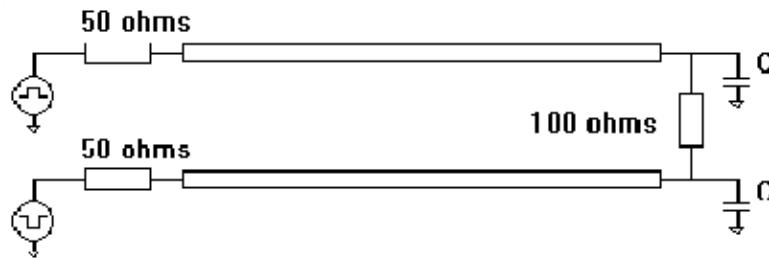


Figure 3.2 The differential feed configuration.

Accurate frequency-domain modeling of such a structure can be obtained from the IE3D Electromagnetic Simulation and Optimization Package. The geometry file is saved in c:\program files\mdspice\samples\st1.geo. On the IE3D, we do not need to simulate the complete structure. Due to symmetry, we only need to simulate half of the structure by placing an electric wall on the symmetric plane. The top view of the IE3D geometry is shown in Figure 3.4. Those users have the IE3D can open it and check it. The thick (red) line in the upper top view is the electric symmetric plane. It is simulated from 0.05 to 20 GHz and the results are saved into C:\MentorGraphics\<latest_release>\IE3D\

SDD_HOME\IE3D\mdspice\samples\st1.sp. We would be interested in checking the s-parameters of the half structure. We will show it to you later.

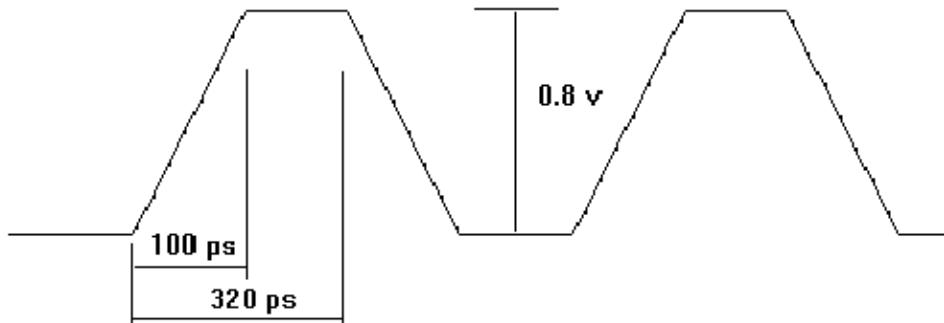


Figure 3.3 The differential stimulus of the transmission line.



Figure 3.4 The top view of the model of the transmission line on the IE3D.

Section 3.2 Using the Schematic Editor.

- Step 1 Run MDSPICE. Select New in File menu to start a new project. MDSPICE will create an empty document on the window. We will call this window as the object list window. Click the right mouse button on the object list window. A pop-up menu comes up. Select Add Main Circuit. The Main Sub-Circuit “main” is created in the object list window (see Figure 3.5).



Figure 3.5 The “main” circuit in the object list window.

- Step 2 Click the right mouse button on the object list window to bring up the pop-up window again. Select Add SubCircuit. A SubCircuit “SubCkt2” is created in the list (see Figure 3.6). We are going to add the call to the s-parameters “st1.sp” into “SubCkt2”.

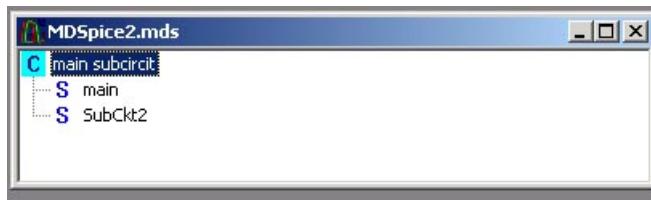


Figure 3.6 The “main” and “SubCkt2” in the object list window.

- Step 3 Double-click at the line “SubCkt2” in the object window list. A schematic window is opened for the “SubCkt2”. Also, you will notice that there are many circuit elements such as R, C, L, CK and SP, etc appear on the toolbars. They correspond to the objects in the Element menu. We can select those elements and add them into the opened schematic editor.

- Step 4 Select the tool bar “sp”. It is equivalent to selecting Parameter Files in the Element menu. The Parameter Files dialog comes up with nothing in the list. Select Add button. The Add Parameter File dialog comes up. Select File Name button. Then, select the file “C:\MentorGraphics\<latest_release>\IE3D\SDD_HOME\IE3D\mdspice\samples\st1.sp”. The information of the file is shown in the dialog (see Figure 3.7). It is a 2-port s-parameters with frequency from 0.05 to 20 GHz.

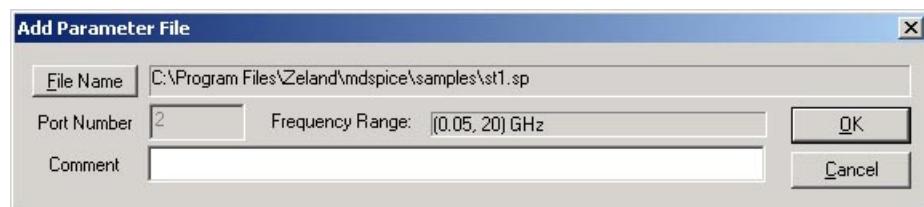


Figure 3.7 The information on the “st1.sp”.

- Step 5** Select OK. The file “st1.sp” is added into the list in Parameter Files dialog. Select the only item or the “st1.sp” in the list box (see Figure 3.8). Select OK to continue. A black box is flying with the mouse on the “SubCkt2” window. Locate it somewhere and click the left mouse button to drop it. Its 4-corners are still high-lighted, meaning it is still being selected (Figure 3.9). Please understand that we can grab the selected object and re-locate it on the schematic editor. You should also notice that a line for “st1.sp” object is added into the “SubCkt2” in the object list window.

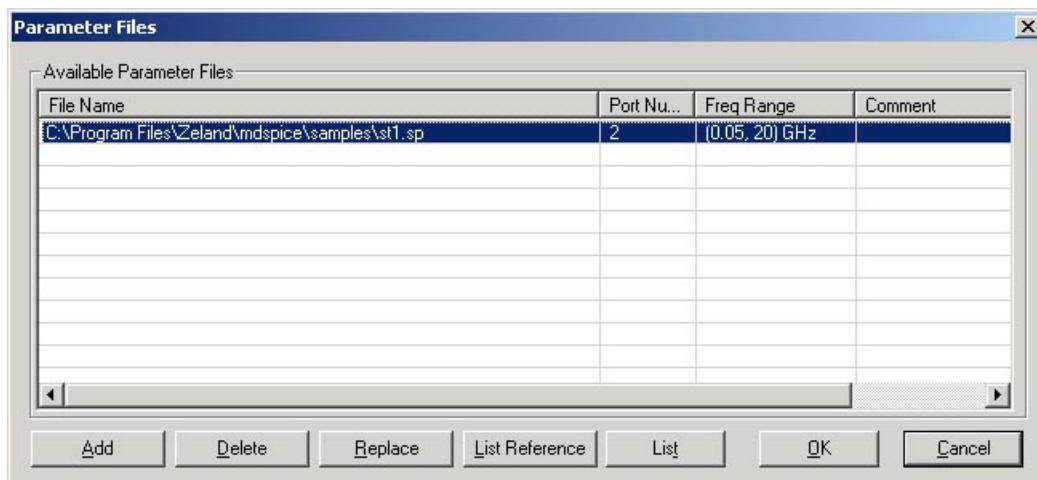


Figure 3.8 The Parameter Files dialog after the “st1.sp” is added.

- Step 6** Select the Define All Ports bitmap in the tool bar, or select Define All Port menu item in the Element menu. Click at the “SubCkt2” window. MDSpice detects 2-spare terminals and connect 2-ports to them (see Figure 3.10). You will see two connections objects are created. They represent the connections between the ports and the terminals which are not shown in Figure 3.10.

Step 7 We have defined the “SubCkt2” now. If necessary, we can add more elements and we can also define more sub-circuits. For now, we just want to call the “SubCkt2” from the “main”. Double-click at the “main” in the object list window to open the schematic editor for “main”. It is empty.

Step 8 Select the “Insert a subcircuit” bitmap in the tool bar. It is equivalent to selecting SubCircuit menu item in the Element menu. The Select SubCircuit dialog comes up and the “SubCkt2” is the only sub-circuits in the list (see Figure 3.11). Select “SubCkt2” and select OK. A black box “X1” is

flying with the mouse. Click somewhere on the “main” to drop the object. Again, the 4-corners of the object are still high-lighted meaning it is being selected.

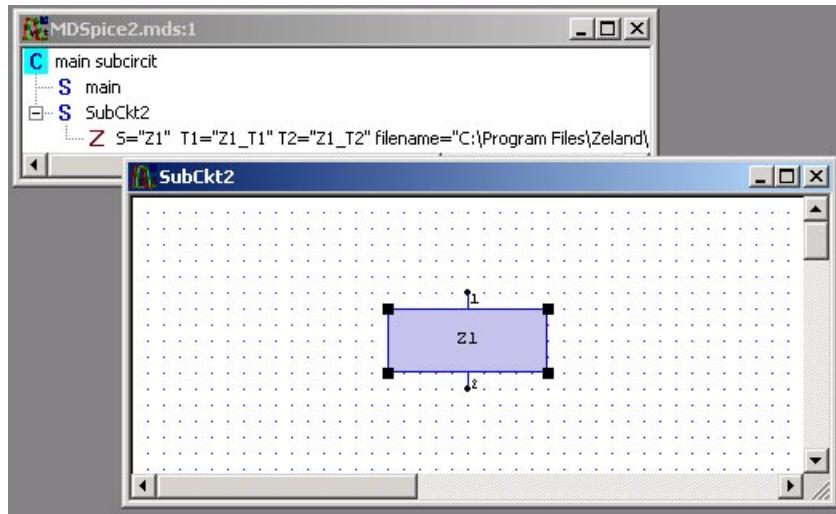


Figure 3.9 The added “st1.sp” object in the schematic editor.

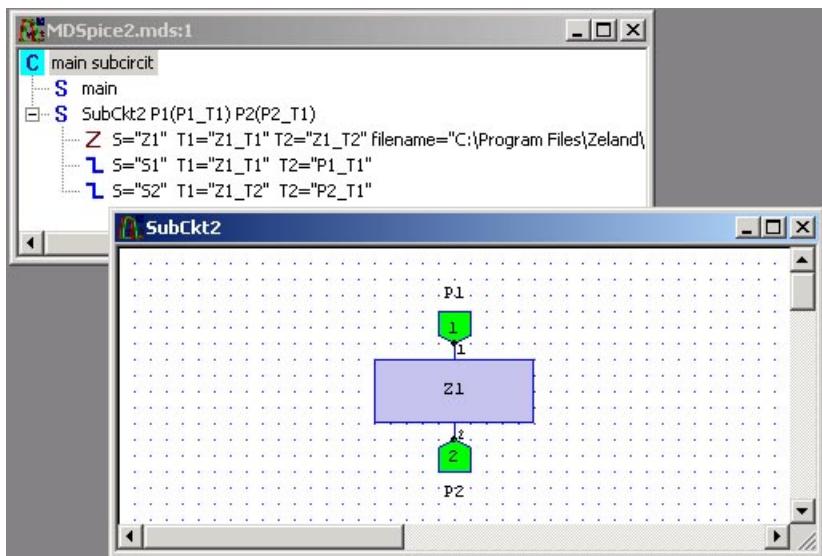


Figure 3.10 The “SubCkt2” with the “st1.sp” and the two-ports defined.

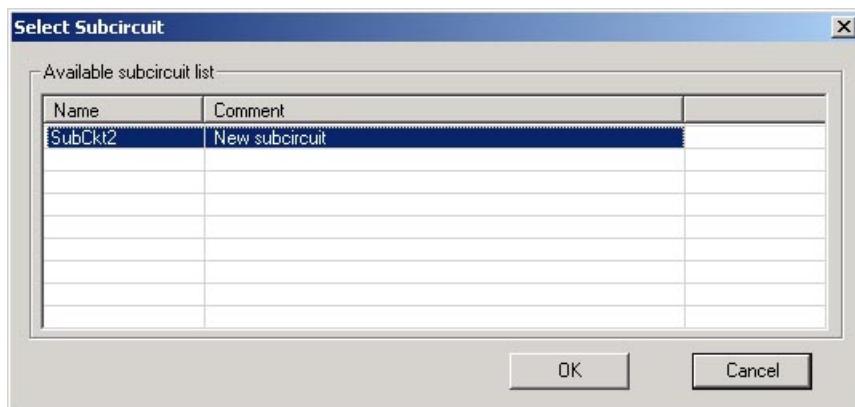


Figure 3.11 The Select SubCircuit dialog.

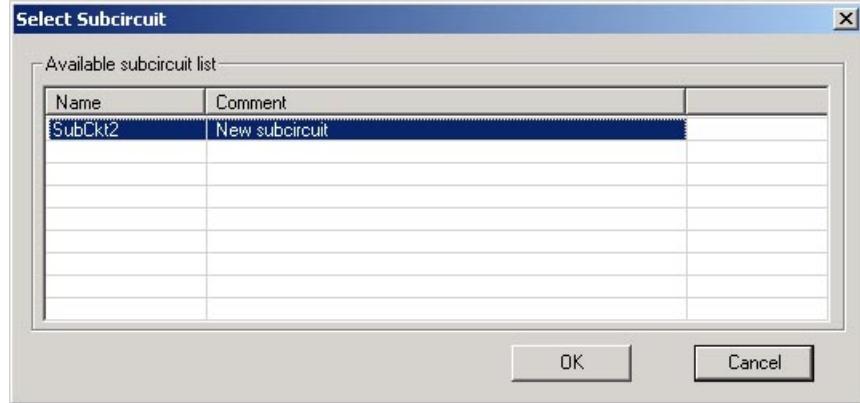


Figure 3.12 The windows with the “X1” call and ports defined in “main”.

Step 9 Select “Define All Ports” bitmap in the tool bar (or the Element menu). Click at the “main” window. Select OK when MDSpice prompts you for the 2-spare terminals. The 2-terminals of the X1 call are defined with two ports. We will have 3 objects in the “main” in the object list window and we will get the picture in Figure 3.12. The “X1” call is a call to the sub-circuit “SubCkt2”, and the “SubCkt2” has the “Z1” call to the s-parameter file “st1.sp”. The circuit is what we want.

Section 3.3 Defining the Frequency Domain and Time Domain Parameters.

Step 1 Select Frequency Domain Parameters in Parameters menu. MDSpice automatically detects the frequency points in the “st1.sp” and list them. Select OK to accept the default frequency parameters.

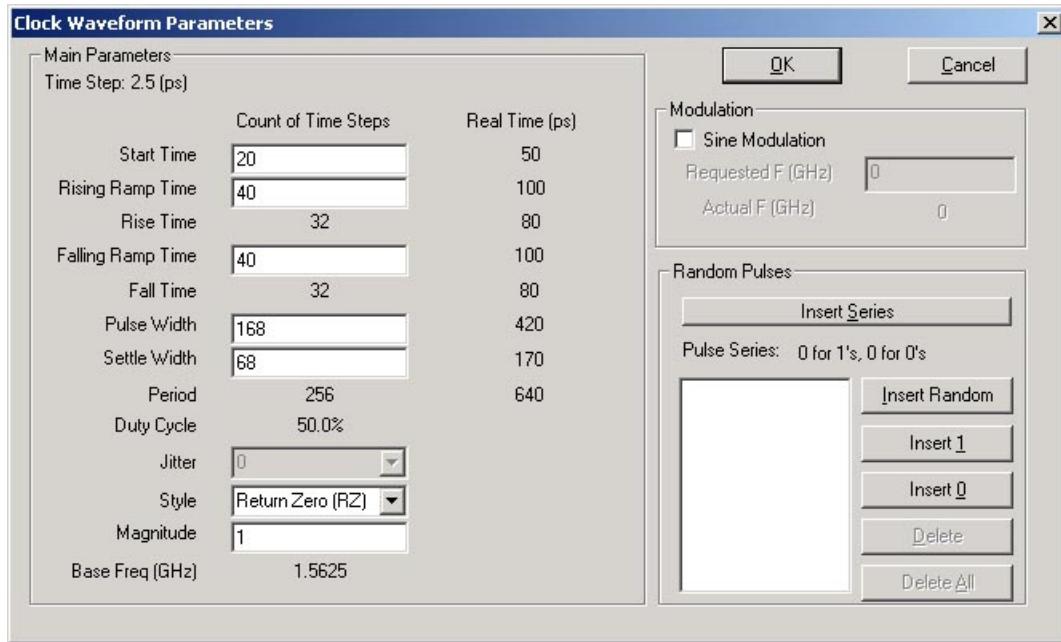


Figure 3.13 The Clock Waveform dialog after the parameters are entered.

Step 2 Select Time Domain Parameters in Parameters menu. The Time Domain Parameters dialog comes up. Select Add button. MDSpice lists the waveform types available. Select Clock Waveform

from the list. Select OK to continue. The Clock Waveform Parameters dialog comes up. Enter the parameters as shown in Figure 3.13. We are going to apply the periodical clock waveform with 80-ps rise time to the circuit and see its response. The style of the clock waveform is Return-Zero. We have the option to define the style of Not-Return-Zero.

- Step 3 Select OK and the waveform will be added into the Time Domain Parameters list box (see Figure 3.14).

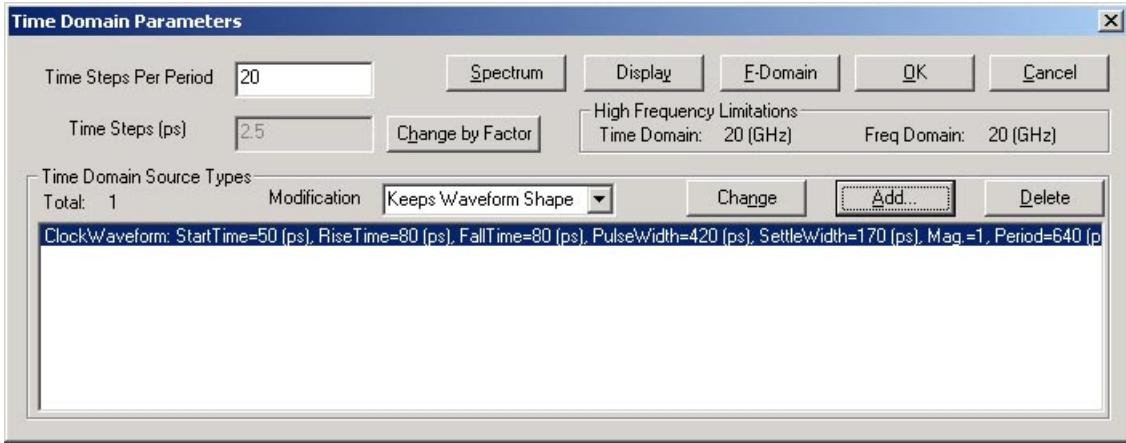


Figure 3.14. The Time Domain Parameters dialog after the Clock Waveform is added.

- Step 4 Select Display button. MDSpice will save the Time Domain Parameters and goes into the Time Domain Sources dialog in Display menu. Select the “ClockWaveform...” or only one in the list. Select OK. The ClockWaveform will be displayed in Figure 3.15.

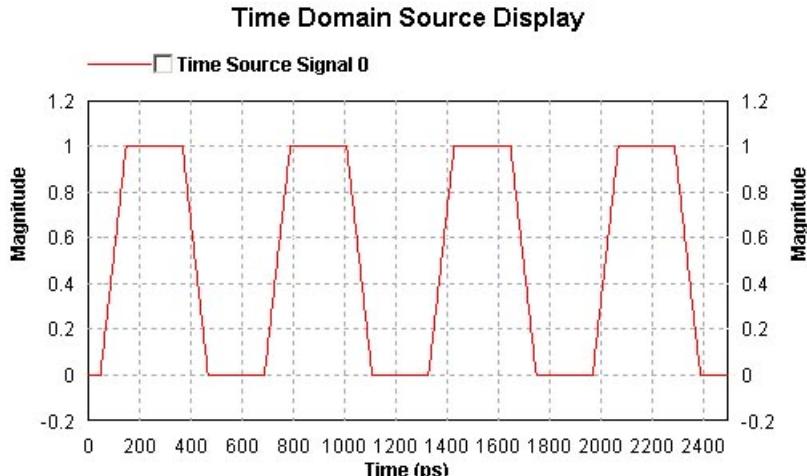


Figure 3.15. The ClockWaveform defined.

- Step 5 Select Save As and save the file as: C:\MentorGraphics\<latest_release>\IE3D\SDD_HOME\IE3D\mdspice\samples\st1a.mds.

Section 3.4 Frequency Domain and Time Domain Simulations.

- Step 1 We are ready to perform a time-domain simulation. However, we may be curious to know how the s-parameters look like. Select Frequency Domain Simulation in Process menu. MDSpice simulate the circuit and list the s-parameters “st1.sp”, “SubCkt2” and “main” in the Internal S-Parameters list. They are in fact the same s-parameters because the “st1.sp” is the only real

element in the circuit. Select New on the top right of the “Graph to be displayed” list box. Select S-Parameters for the “Graph Type”. Select OK to continue. MDSPICE will prompt the user to select the displayed items.

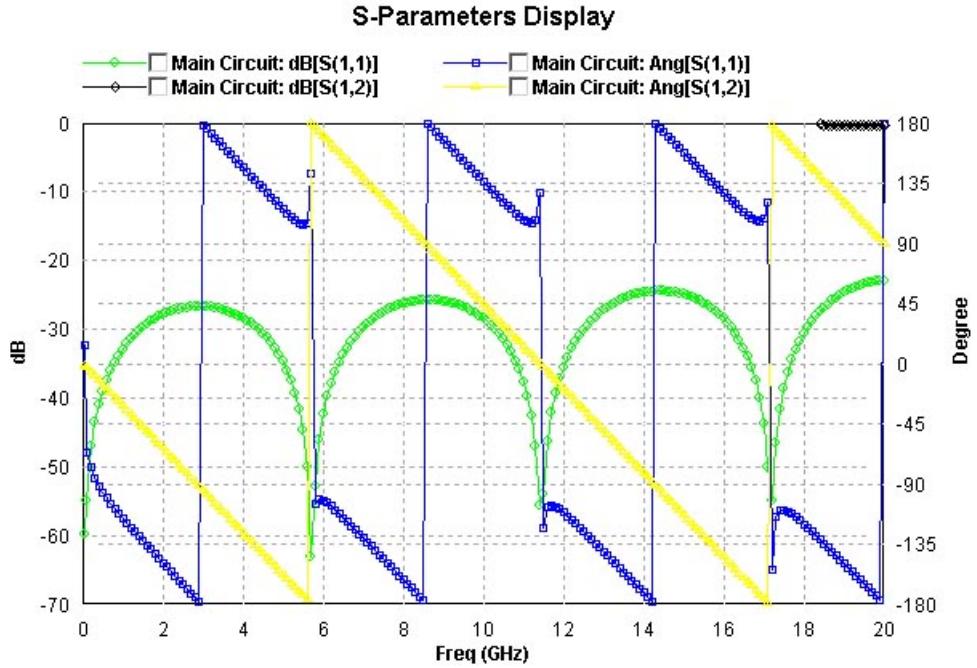


Figure 3.16 The displayed s-parameters of “st1.sp”.

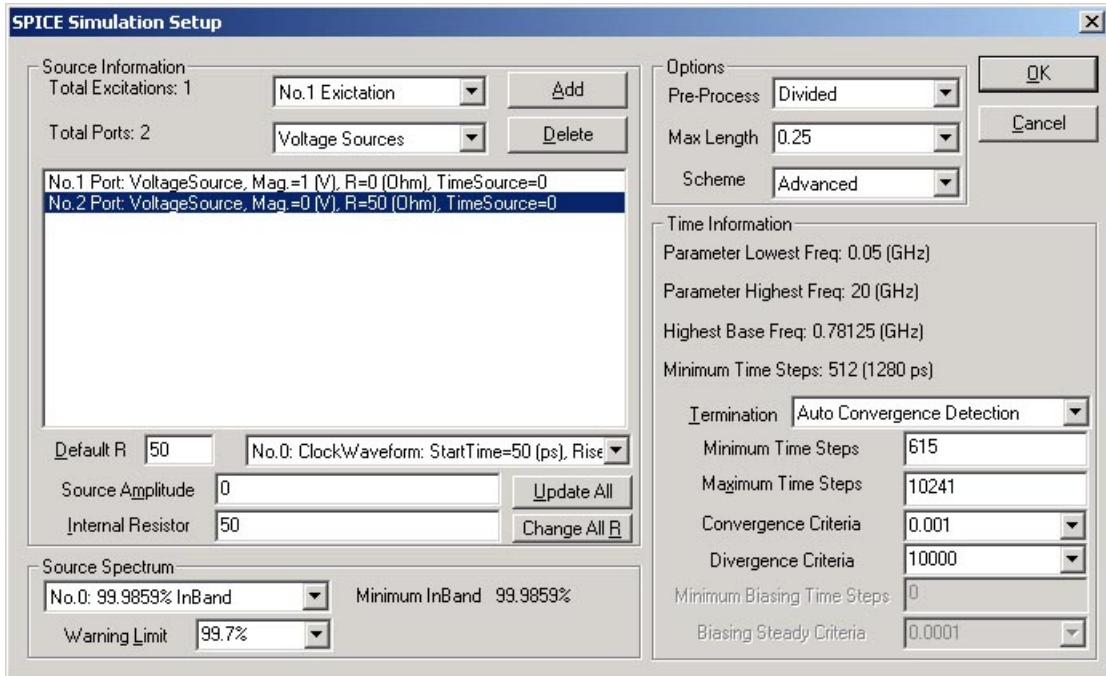


Figure 3.17 The SPICE Simulation Setup dialog by Step 3.

Step 2 Select Root = Main Circuit. Select “dB” and “Ang” for S(1,1) and S(2,1) for 4-items. Select OK. The defined graph is added into the list. Select OK to continue. The s-parameters for “st1.sp” are displayed in Figure 3.16. From the up and down of the dB[S(1,1)] and the phase delay in the

$\text{Ang}[\text{S}(2,1)]$, we can see it is about 1.75 waveguide wavelength long at 20 GHz. Simulating such a long transmission line and guarantee the causality condition is not a simple task. It certainly requires a robust and hi-fidelity time domain engine. We will give it a try on the MDSPICE.

- Step 3 Select Time Domain Simulation in Process menu. The SPICE Simulation Setup dialog comes up. Select the “No.1 Port: VoltageSource...”. Change the Source Amplitude = 1 and Internal Resistor = 0. WE are applying all the voltage into the port 1 of “main”. Select the “No.2 Port: VoltageSource...”. Change the Source Amplitude = 0 and Internal Resistor = 50-ohms. We will get dialog shown in Figure 3.17.

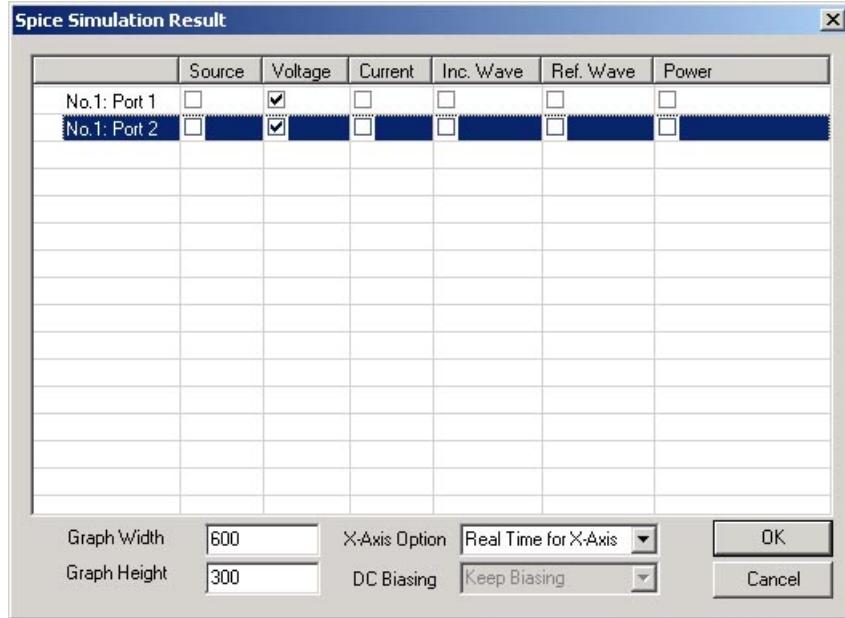


Figure 3.18 The SPICE Simulation Result dialog.

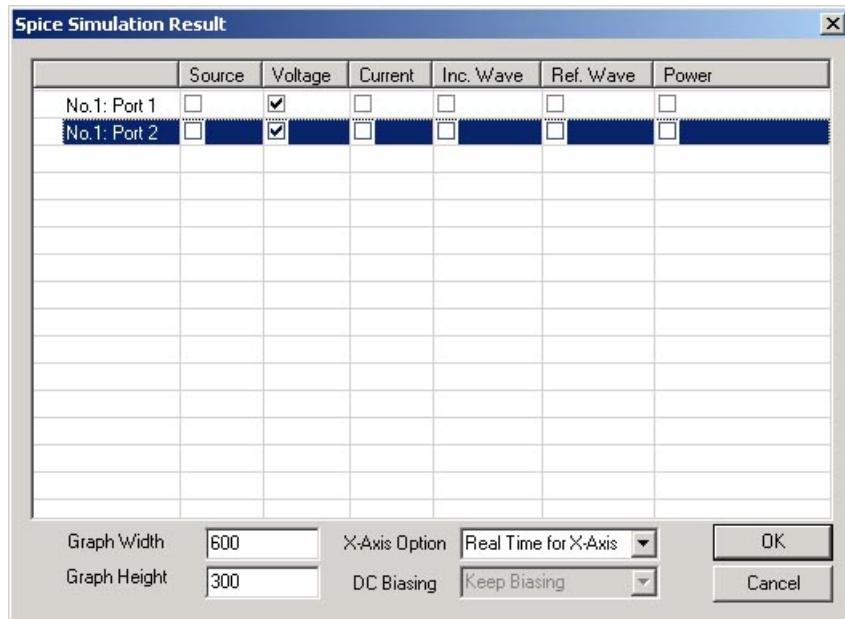


Figure 3.19 The input and output waveform on the “st1.sp”.

- Step 4 Make sure the Scheme is “Advanced”. A user should understand that the “Advanced” scheme is a very robust and high accuracy scheme implemented into the MDSPICE 3.0. The other option is the “Regular” scheme implemented in the MDSPICE 2.2. If the user wants to try the “Regular” scheme, it is suggested that the user should not use the s-parameters longer than $\frac{1}{2}$ of the waveguide wavelength at the highest frequency. Time-domain simulation is very sensitive to any numerical error because it will accumulate. For long transmission line, the numerical error can easily accumulate to create significant error in the time domain signal. For ultra long transmission lines, we should always cut the transmission line into shorter sections in order to reduce the accumulation of error. However, using the “Advanced” scheme, we should be able to accept relative long sections of transmission lines.
- Step 5 Select OK to accept the other default settings and start the simulation. MDSPICE will take less than 1 second to finish the simulation. As it is shown in Figure 3.18, the residuals are below 9.5e-5 and the signals are well converged. The simulation takes 2176 time steps. Select OK to proceed to time signal display. MDSPICE will prompt you to select the display items. Select “Voltage” for “No.1: Port 1” and “No.1: Port 2” (see Figure 3.18). Select OK. The time signals are displayed in Figure 3.19.

Section 3.5 Setting Markers and Measuring Time Delay.

- Step 1 Select Graph Parameters in View menu. Change the parameters as: Start = 0, End = 1000, Step = 100. Select OK to accept the other default settings. The time signals are zoomed in the section from 0 to 1000 ps (see Figure 3.20). As you can see, there is a clear delay in the “Port 2 Voltage”. There is some very small ripple (below 1%) before the pulse reaches the port 2.
- Step 2 Select the check box on the legend “Port 1 Voltage”. Move the mouse to the red line for the input waveform (Port 1 Voltage) until the Y-value reaches Y = 0.5. Click the left mouse button to locate the 1st marker at (X, Y) = (100, 0.5). Select the check box on the legend “Port 2 Voltage”. Move the mouse to the green line for the output waveform (Port 2 Voltage) until the Y-value reaches about Y = 0.5. It is difficult to reach exact Y = 0.5. It does not matter and we should be able to set it later. Click the left mouse button to locate the 2nd marker at about Y = 0.5.
- Step 3 The 2nd marker can not be exactly at Y = 0.5. Click at the No.2 marker’s X and Y-values on the table on the left of the window. The dialog “Number 2 marker property” comes up. Select the Y-value and change it to 0.5. Select OK. The No.2 marker’s X and Y-values are changed to (X, Y) = (187.019, 0.5).
- Step 4 On the left bottom, please select the “2” in the 1st combo box. We will see the display dX: 87.019 and dY: 0 (see Figure 3.20). It means that the differences between markers 1 and 2 are dX = 87.019 and dY = 0. Basically, we are measuring the time delay between the “Port 2 Voltage” and the “Port 1 Voltage” and the average voltage points. The time delay is 87.019 ps for the 500-mil long transmission line. It is about 174.038 ps for 1-inch long transmission line. It is a typical value for PCB using FR4.



Figure 3.20 The zoomed display with markers for measurement.

Section 3.6 Modeling of Ultra Long Transmission Lines.

We have demonstrated how we can model a section of transmission line. In practical applications, we may encounter very long transmission lines. For the st1.geo, the length is 500-mils, it is possible that we may have a transmission line of length 1-foot or more. Can we still use one single s-parameter file to model it? We may or may not. If possible, we would like to use cascaded sections of shorter transmission lines to model it. When we use shorter transmission lines, we will be able to reduce the accumulation of numerical errors. In this section, we will try to model a 1-foot transmission line.

For the “SubCkt2”, it consists of the “st1.sp” of 500-mils long. We can construct a main circuit consisting of a series sub-circuits rooting from the X-calls to the “st1.sp”. The sub-circuits are listed in Table 3.1 and the file is saved in “C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\st1b.mds” file.

Table 3.1 The sub-circuits in the st1b.mds file.

SubCircuit	Contents	Equivalent Length
SubCkt2	1-call to st1.sp	500-mils
SubCkt3	2-calls to SubCkt2	1000-mils
SubCkt4	2-calls to SubCkt3	2000-mils
SubCkt5	2-calls to SubCkt4	4000-mils
SubCkt6	2-calls to SubCkt5	8000-mils
Main	1-call to SubCkt6 and 1-call to SubCkt5	12000-mils or 1-foot

Step 1 Open the C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\st1b.mds file. Check its contents against what is documented in Table 3.1 so that you understand how we build up the 1-foot transmission line based upon the multiple-level sub-circuits.

Step 2 Select Time Domain Simulation in Process menu. Select OK to accept the default settings. The simulation finishes in couple seconds. The residuals are below 0.001 and the total time steps are 6272. Select OK to proceed to time signal display. Select the “Port 1 Voltage” and “Port 2 Voltage”. Select OK to continue. Select Graph Parameters in View menu and adjust the End = 4000 and the Step = 400. Select OK. We should get the display in Figure 3.21. You can see the output waveform does not appear after the delay of 2114.58 ps. The delay is equivalent to about 176.2 ps per inch. It agrees with well the simulation result of 500-mil long transmission line. You can also see the reduced magnitude in the output waveform compared to the input waveform.

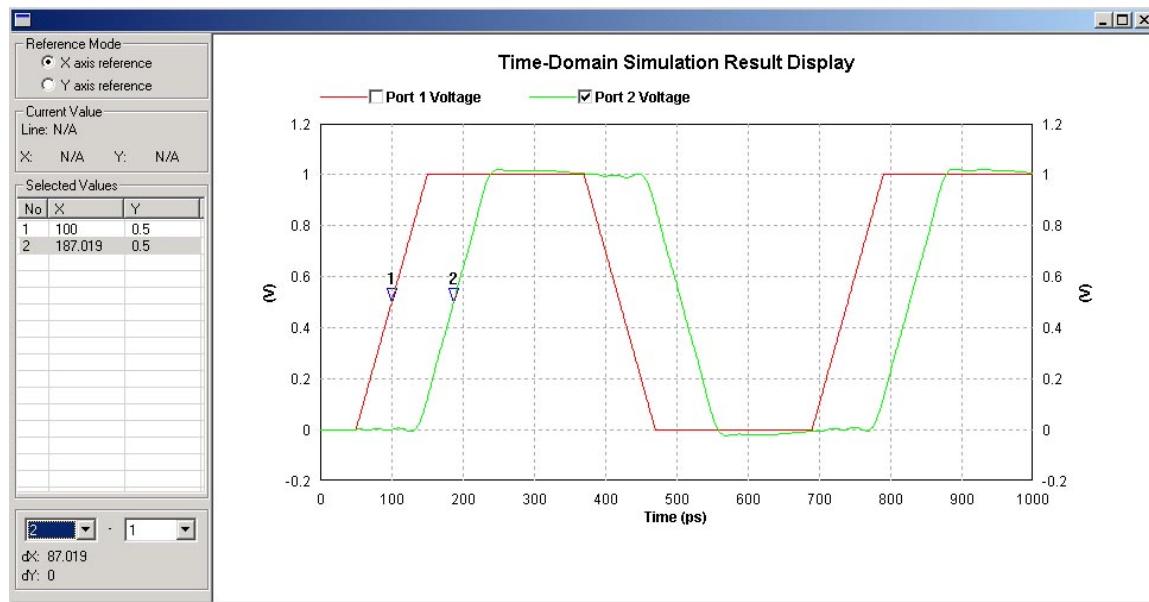


Figure 3.21 The time delay display for the ultra-long transmission line.

Section 3.7 Eye Pattern Display.

In our above simulations, we always assume the clock waveforms are periodical waveforms. The pulses are of the same width. In practical applications, the clock waveforms may not be periodical. They might take the on or off randomly. Also, the pulse width may change slightly with jitters. On the MDSPICE, we can define the clock waveforms with random series and jitters.

Step 1 While the st1b.mds is opened, select the Time-Domain Parameters to open the dialog. Double-click at the only waveform: ClockWaveform... in the list to edit it. (see Figure 3.22) . Select Insert Series. MDSPICE will prompt the users to enter the length of the series and whether the user wants the series to be randomly created. Enter 200 for the length of the series and select Random for the Magnitude. Select OK to continue. The random process on MDSPICE creates ninety-nine 1's and one hundred and one 0's (It may differ in your practical use). Change the Jitter from 0 to 2. It means that the jitters in the waveform may vary from -2 to 2 time steps or -5 to 5 ps) in the pulses. You will see there is a percentage following each 1's or 0's (see Figure 3.25). It means the jitter for the pulse is offset by the percentage of the total maximum size (2 time steps). Change the Style from Return Zero (RZ) to Not Return Zero (NRZ). We will get the exact picture in Figure 3.22. Select OK to finish the change. Select OK to exit the Time Domain Parameters dialog.

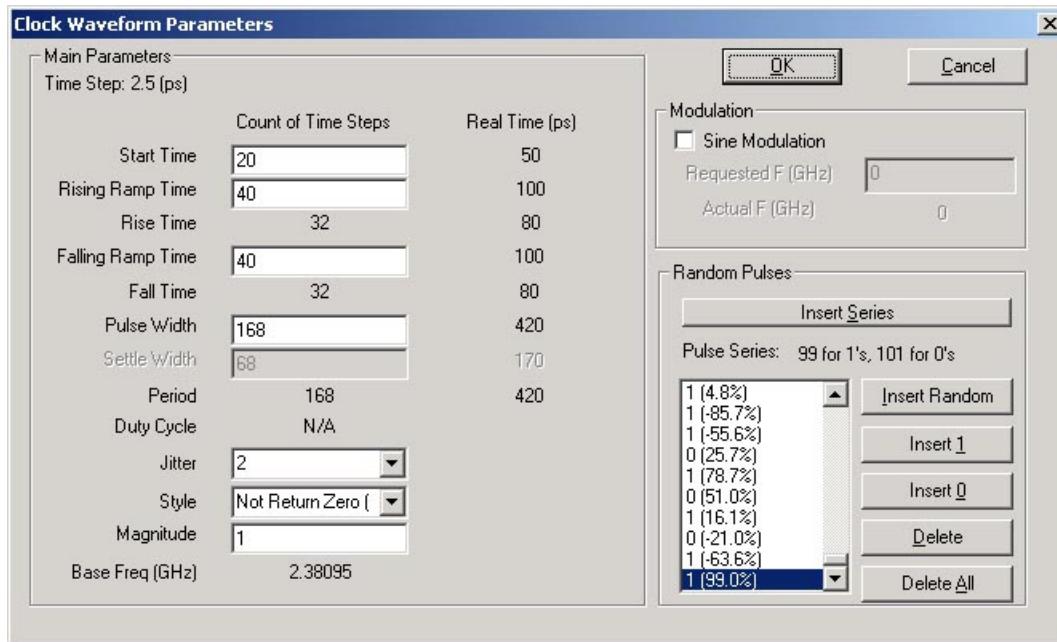


Figure 3.22 The Clock Waveform with random series and jitters.

Step 2 Save the file as “C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples \st1c.mds”. Select Time-Domain Simulation in Process menu. Select OK to accept the default settings. MDSPICE will take a few seconds to finish the 33621 time steps. After the simulation, MDSPICE shows that the residuals are below 0.001. This time, MDSPICE will set the default to Proceed to Eye Pattern Display. For those users who want to see the time signal display, you can also check the Proceed to Time Signal Display. For the discussion below, we will focus on the eye-pattern display. Select OK to continue. Select “Port 1 Voltage” and “Port 2 Voltage”. Select OK and the eye pattern will be displayed. There is some slight shift.

Step 3 Select Graph Parameters in View menu. Change the Start from 0 to 98. Select OK and the eye-pattern will be centered (see Figure 3.23). The eye-pattern is a measure how good the transmission line is. We only demonstrate how to get the eye-pattern. Detailed discussion on the meaning of the eye-pattern is out of the scope of this manual. The file containing the jittered random signals is saved in the C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\st1c.mds. If you could not produce it from the st1b.mds, you can open the file for practice.

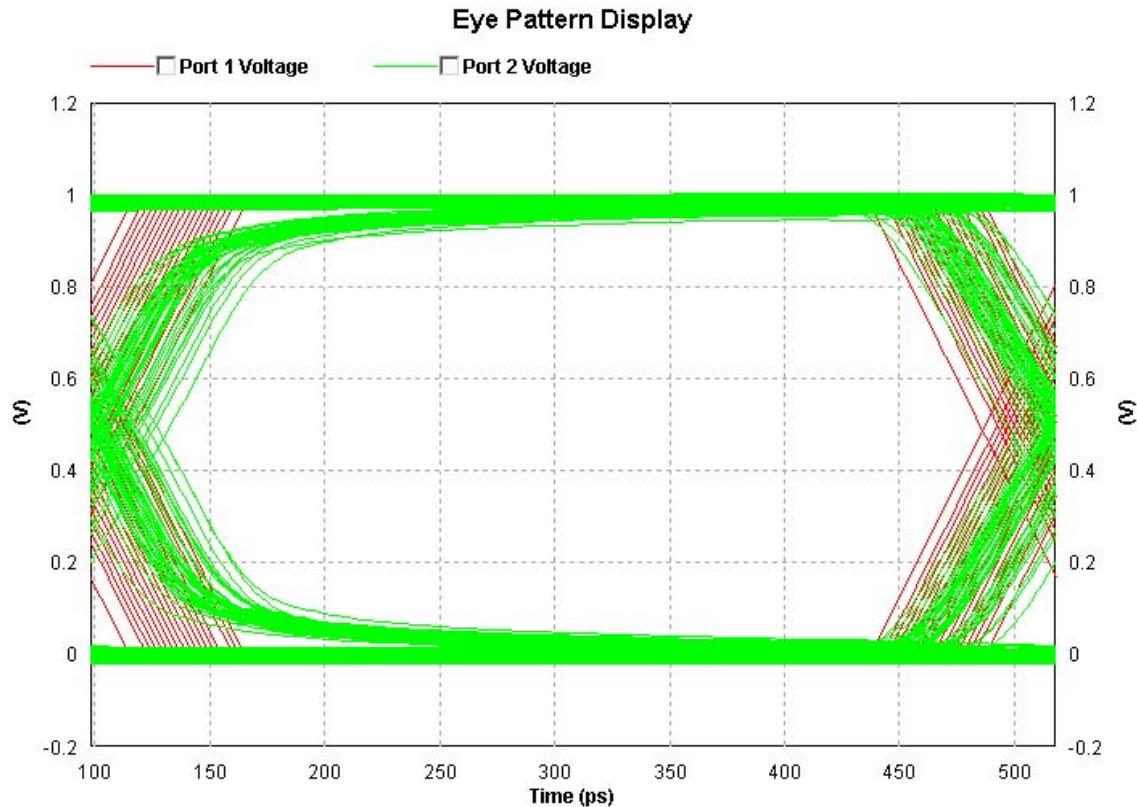


Figure 3.23 The eye-pattern display on the ultra long transmission line.

Section 3.8 Extracting Frequency Dependent RLGC of Transmission Lines from S-Parameters.

In our above examples, we have shown you how we can perform the time-domain simulation on ultra long transmission lines by modeling them as s-parameters. Such a method does not have any obvious limitation and we can apply the same method to any s-parameter elements.

As coupled and frequency dependent transmission lines are concerned, we may not be able to get the s-parameters of the transmission lines directly. In practical applications, we may have the distributed R, L, G and C of a transmission line only. Also, the length of the transmission line may not be a known value until the simulation time. We only know the distributed R, L, G and C. The length can be determined any time later. Certainly, theoretically we can implement a scheme to calculate the s-parameters of the transmission line with known distributed R, L, G and C. Then, we can perform the time domain simulation on the resulting s-parameters of the transmission line. Such a process is fully implemented and automated into the MDSPICE. On the MDSPICE 3.0, we have implemented the frequency dependent transmission line model. We will show the user how we can use the frequency dependent transmission line model for convenient frequency and time domain simulation.

The example we are going to use is similar to the one shown in Figure 3.1. This time, we will not consider it as a differential line. We will consider it as coupled stripline structure with 4-ports (ports 1 and 2 at one end and ports 3 and 4 at the other end). A short section of the transmission lines is created on the

IE3D in C:\MentorGraphics\<latest_release>\IE3D\SDD_HOME\IE3D\mdspice\samples\st2.geo. The length is 50-mils. Based upon the s-parameters of the short section of transmission lines, we are able to find the distributed R, L, G and C on the MODUA, an accessory to IE3D and MDSPICE. The s-parameters of the st2.geo is saved in: C:\MentorGraphics\<latest_release>\IE3D\SDD_HOME\IE3D\mdspice\samples\st2.sp. The 1st step is to extract the distributed R, L, G and C from the st2.sp.

Step 1 Run MODUA. Select Display Parameters Module. Select the file st2.sp in directory C:\MentorGraphics\<latest_release>\IE3D\SDD_HOME\IE3D\mdspice\samples. The s-parameters of the st2.sp will be displayed on MODUA. MODUA is a simple circuit simulator with many useful utilities. We will just show you how we can extract frequency dependent RLGC from MODUA. For more information on MODUA, please read the IE3D manual.

Step 2 The s-parameter file has 7 frequency points (0.1, 0.5, 1, 2, 3, 5 and 10 GHz). It does not show a smooth curve on MODUA (see Figure 3.24). We can't apply the time-domain simulation directly on the s-parameters because the MDSPICE requires the s-parameters to be smooth on a wide frequency range. However, we can try to extract the distributed RLGC from the few frequency points. The distributed RLGC should be close to constants and we do not need to use many sample points.

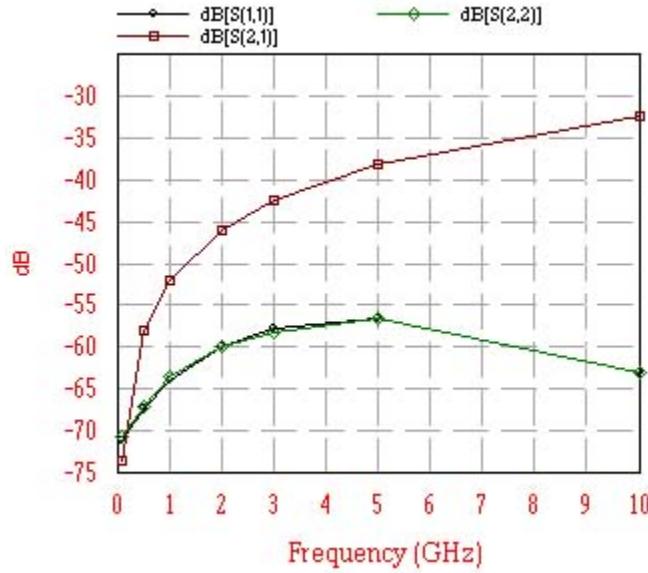


Figure 3.24 The s-parameters of st2.sp with few frequency points.

Step 3 Select LC-Equivalent in Process menu of MODUA. MODUA will prompt you for the frequency list. Select OK. MODUA will prompt you with a dialog “Multiple Frequency LC-Equivalence”. Select OK to continue. MODUA will prompt you for the port style definition.

For this equivalence, we can't change the port style (orders and patterns of the ports in the coupled transmission lines). The port styles are always ports 1 and 2 on one side of the TLN and ports 3 and 4 are on the other side of the TLN. It is the general requirement for multiple frequency LC-equivalent circuit extraction on the MODUA. Given N-number of coupled traces, we require the ports 1-N to be on one side and the ports (N+1) to (2N) to be on the other side for multiple frequency extraction. For single frequency extraction, we do allow the user to have the options.

Another option is the Shunt R. If you want to neglect the shunt R's in order to reduce the number of elements in the equivalent circuit, you can select the “Without Shunt R” option. The default is “With Shunt R” and it is of the best accuracy.

Step 4 Select OK. MODUA will prompt you for Saving the RLGC Matrix (see Figure 3.25). MODUA will try to save the extracted RLGC from the s-parameters. However, the s-parameters (in Agilent/EESOF Touchstone format) do not carry the information about the length of the TLN. The user should enter the length information correctly if he wants to use the extracted circuit in the MDSPICE simulator. For our case, it is a 50-mil long coupled TLN. Please select Unit = mil and Length = 50 (see Figure 3.25).

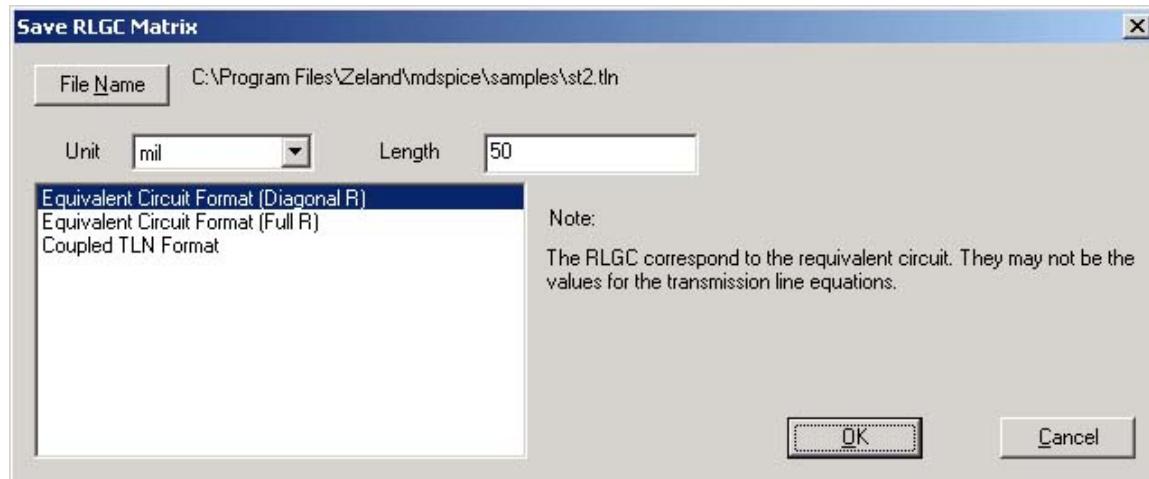


Figure 3.25 The Save for RLGC Matrix dialog.

Step 5 There are 3 types of output file formats: (1) Equivalent Circuit Format (Diagonal R); (2) Equivalent Circuit Format (Full-R); (3) Coupled TLN Format. The (1) and (2) are the values for the true topology. For the LC-Equivalent, we normally only create diagonal R for the series R. However, in case the user wants to put them into the full matrix format, the user can choose (2) and the off-diagonal series R's will be set to 0. For the (3), the shunt R's and shunt C's are not the values of the true topology. They are in the Maxwellian form. Interested users can try to save the file into different format. Each format has an ID in the file: (1) ID=2646; (2) ID=2647; (3) ID=2648. For this case, please select Equivalent Circuit Format (Diagonal). Select OK. MDSPICE will extract the equivalent circuit and prompt you. Select OK. The saved RLGC equivalent circuit: st2.tln will be opened on the Notepad (see Table 3.2).

Step 6 As you can see, the TLN is too long at 10 GHz. We have two ways to go: (1) Reduce the length of the st2.geo and re-simulate it for the s-parameters and the equivalent circuit. (2) Remove the 10 GHz from the frequency list. We may not need the 10 GHz point because the distributed RLGC values are kind of constant at high enough frequency. If we remove the 10 GHz, it is assumed that the RLGC values are constant above 5 GHz in our case.

Step 7 Select Frequencies in Control menu of MODUA. Make sure all the frequency points except the 10 GHz are checked. Select OK to continue. Select LC-Equivalent in Process again. You will see the 10 GHz frequency is no longer in the list. Select OK. MODUA will prompt you for the Shunt R Option. Make sure it is "With Shunt R" and select OK. MODUA will prompt you for the Save RLGC Matrix dialog. Make sure the Unit is "mil" and the Length is "50". Select OK. MODUA may still warn you that the Error is 0.28... exceeding 0.25. It is slightly over 0.25 and it should be ok. Select OK to continue. The equivalent circuit is saved into: st2.tln and it is opened on Notepad. Please check the data. You will see that the L(i,j) and C(i,j) are close to constant of frequency. Certainly, there is some frequency dependency in them, and the frequency dependency is critical to the accuracy of wide band SPICE simulation. Also, the user should understand that the RLGC values in the .tln file are the values per meter no matter what the user selects for the Unit in the extraction.

Section 3.9 Alternative Way to Simulate Dependent RLGC of Transmission Lines

What we have extracted in the last section are the RLCG of the equivalent circuit for the short coupled transmission line. When it is short enough (small enough Error in the extraction), we can use the extracted RLCG values as the distributed RLCG values.

After we obtained the distributed RLCG values for general coupled transmission lines, we will be able to perform SPICE simulation based upon the values in the .tln file.

Table 3.2 The RLGC Equivalent Circuit (Diagonal) file:

2646	7	2	1	2.5400000000e-005
Equivalent Circuit Parameters: The error function for 5 GHz and higher frequency is quite high (0.28626>0.25)...(cont. in one line)				
Error Factor				
Freq (GHz)	0.1	0.5	1	2
Error	0.0056327	0.028068	0.056062	0.1122
			3	5
			0.16898	0.28626
				0.62257
Series R (ohm)				
Freq (GHz)	0.1	0.5	1	2
R(1)	0.0002532	0.00049644	0.00073884	0.0010439
R(2)	0.00019397	0.00052356	0.00080495	0.0010421
			3	5
			0.0011925	0.0014842
				0.0012565
			10	
			0.0011262	0.0014842
				0.0012565
L (nH)				
Freq (GHz)	0.1	0.5	1	2
L(1,1)	0.0093628	0.0090433	0.0089801	0.0088899
L(1,2)	0.00036051	0.00040424	0.00040227	0.00039583
L(2,1)	0.00036051	0.00040424	0.00040227	0.00039583
L(2,2)	0.0094533	0.0090472	0.0089812	0.0088899
			3	5
			0.0087798	0.0084653
				0.0071389
			10	
			0.00039021	0.00037671
				0.0003182
			0.00039021	0.00037671
				0.0003182
			0.00084653	0.0071389
Shunt R (ohm)				
Freq (GHz)	0.1	0.5	1	2
R(1,1)	2.3987e+008	4.7056e+007	2.5998e+007	1.1414e+007
R(1,2)	6.5995e+009	1.0229e+009	5.0169e+008	2.5916e+008
R(2,1)	6.5995e+009	1.0229e+009	5.0169e+008	2.5916e+008
R(2,2)	2.3989e+008	4.7049e+007	3.0216e+007	1.1415e+007
			3	5
			6.3093e+006	4.0289e+006
				1.4115e+006
			10	
			1.5755e+008	1.1817e+008
				1.0269e+008
			1.5755e+008	1.1817e+008
				1.0269e+008
			6.087e+006	4.0289e+006
				1.4115e+006
C (pF)				
Freq (Ghz)	0.1	0.5	1	2
C(1,1)	0.003314	0.0033722	0.003378	0.0033862
C(1,2)	0.00011935	0.00015562	0.00015709	0.00015626
C(2,1)	0.00011935	0.00015562	0.00015709	0.00015626
C(2,2)	0.003314	0.0033722	0.0033773	0.0033862
			3	5
			0.0034033	0.0034667
				0.0037915
			0.00015653	0.00015996
				0.00017512
			0.00015653	0.00015996
				0.00017512
			0.0034046	0.0034667
				0.0037915
Explanation:				
No.1 Line: The “2646” means that it is in Equivalent Circuit Format (Diagonal). The “7” means that it has 7 frequency points. The “2” means that there are two coupled lines. “1” means that it is in “mil” (“0” means “mm” and “2” means “micron”). The “2.5400e-5” is the length of the unit in m.				
The user should pay attention to the Error. It is a value determines how good the equivalent circuit is at a specific point. Normally, its value should not exceed 0.25. For this circuit, it is 0.286 at 5 GHz and it is at the start point for accuracy degrade. For the 10 GHz, the error function is 0.62257 and it is high.				
Other lines should be straight-forward. A user can create his own files while keeping the format the same. The strings in the files are not critical. Each continuous string (without blanks) will be considered as a string when the MDSPICE 3.0 parses the file.				
For the series R and L, and shunt R and C, they are the values per unit length. For this example, they are the values per mils (or per 2.5400e-5 m).				

Step 1 Run MDSPICE. Open the C:\MentorGraphics\<latest_release>\IE3D\SDD_HOME\IE3D\mdspice\samples\st1a.mds file. Save it as st1d.mds file. We should have the object list window.

Step 2 Click at the “SubCkt2” in the list to open the schematic editor. You will see a 2-port black box “Z1” with two ports defined on its two terminals.

- Step 3 Click at the black box “Z1” to select it. Then, hit “Del” button or select “Delete” in Edit menu to delete the element. The two ports, which are dependent elements, will also be deleted automatically.
- Step 4 Select the bitmap “Insert a Lossy TLN” or select LossyTLN in the Element menu. The Lossy TLN Files dialog comes up. It is very similar to the Parameters Files dialog we discussed earlier.
- Step 5 Select Add button. MDSPICE will prompt you to Add TLN File. Select File Name button. Then, select the st2.tln file. Select OK to add it. MDSPICE will detect it is a 4-port transmission line. By default, the ports 1 and 2 of the TLN are the terminals 1 and 2 of the element. The ports 3 and 4 of the TLN are the terminals 3 and 4 of the element. It is unchangeable. The frequency range is from 0.1 to 5 GHz. Select OK. The TLN file is added into the list.
- Step 6 Click at the only added TLN file in the list. Select OK. The element representing the call to the TLN is flying with the mouse. Click at somewhere to drop the element labeled as “M1”. Its 4-corners are still high-lighted, meaning it is being selected. Double-click at the element “M1”. Its properties are displayed in a dialog. You will see the Length = 0 mm. We want to change it to some value in mil and we need to exit first.
- Step 7 Select OK to continue. Select Optional Parameters in Parameters menu. Change the selection for Length Unit from “mm” to “mil”. Select OK to close the dialog.
- Step 8 Double-click at the “M1” element. Its properties are listed again. Change Length from “0” mil to “100” mils. Select OK to continue.
- Step 9 Select Define All Ports bitmap from the toolbar or select Define All Ports in the Element menu. Click at the “SubCkt2” window. MDSPICE will detect 4-spare terminals. Select YES and MDSPICE will put the ports 1, 2, 3 and 4 on the terminals 1, 2, 3 and 4 of “M1” sequentially.
- Step 10 Go to the object list window (or the main window). Double click the “main” in the list to open the schematic editor on it. Select “X1” element. It is the call to the “SubCkt2”. We have changed the “SubCkt2” from 2-ports to 4-ports. We need to redefine it. Select Delete in Edit menu to delete it.
- Step 11 Select “Insert a subcircuit” bitmap in the toolbar or select SubCircuit in Element menu. MDSPICE will prompt you to select an pre-defined SubCircuit. “SubCkt2” is the only one in the list. Click at the “SubCkt2” and select OK. The black box for “SubCkt2” is flying with the mouse. It is a 4-terminal element. Click at somewhere to drop it.

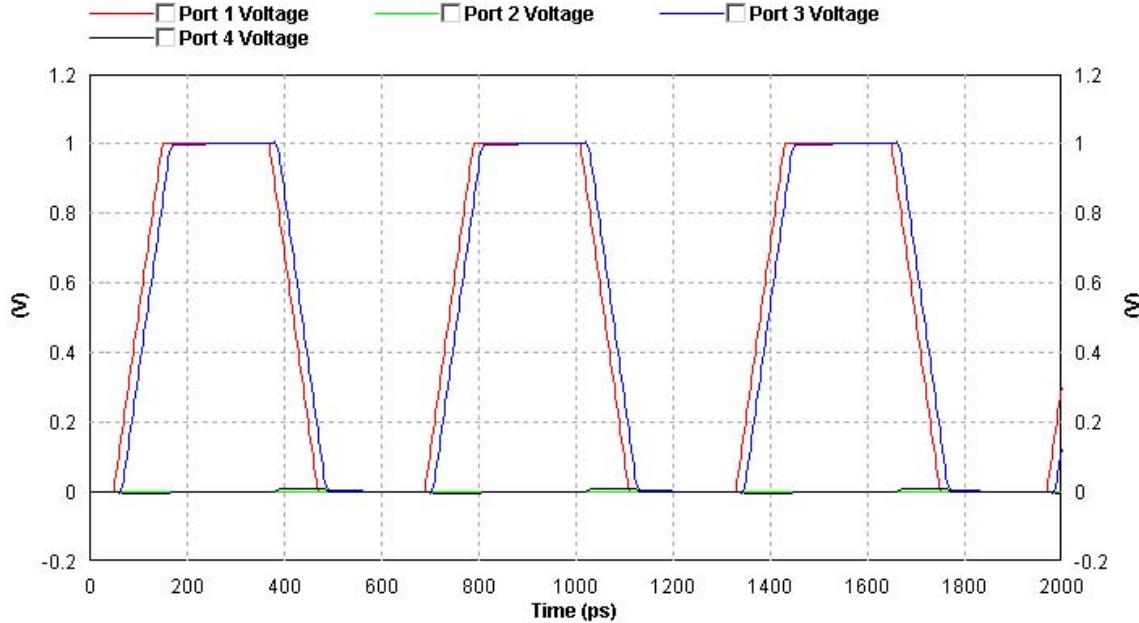


Figure 3.26 The delay and cross-talks on coupled short TLNs.

- Step 12 Select the bitmap “Define All Ports” or select the Define All Ports in Element menu. Click at the “main” schematic window. Four ports are defined on the 4-terminals of the “X1” element, a call to the “SubCkt2” element. Select Save in File menu to save the change. By now, we have changed the topology while we keep the other parameters are unchanged.
- Step 13 Select Time Domain Simulation in Process menu. Define the No.1 Port as: Amplitude = 1, Source Resistor Rs1 = 0. Define the Rs2 = Rs3 = Rs4 = 50 ohms for the ports 2, 3 and 4. Select OK to continue. MDSPIKE will simulate the circuit in couple seconds. It will show the convergence is excellent in the post-simulation dialog. Select OK. MDSPIKE will prompt you for the display of simulation results. Select the voltage for No.1 Ports 1, 2, 3 and 4 (4 items). Select OK. MDSPIKE will show you the complete time domain results. It is a very wide time period. Please select Graph Parameters in View menu. Change the Start = 0, End = 2000, Step = 200. Select OK. The display will zoom in the 0 to 2000 ps period (see Figure 3.26). As you can see, there is little delay between port 1 and port 3. Also, the cross-talk levels at ports 2 and 4 are quite low. The results are expected because the TLN is very short (100 mils). You may want to see how the length of the TLN affect the delay and the cross-talk levels.
- Step 14 Go to the object list window (main window) of st1d.mds. Double-click at the element “M1” in the “SubCkt2” in the list. MDSPIKE will show you the properties of the M1 element. Change the length from 100 mils to 2000 mils or 2-inches. Select OK to close it. Select Save As and save the file as: C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\st1e.mds. Select Time Domain Simulation in Process to simulate it. It will take tens of seconds to simulate the circuit. Select OK to continue in the post-simulation dialog. Select Voltage for No.1 Ports 1, 2, 3 and 4 for display. Select Graph Parameters and change the Start = 0, End = 3000, Step = 300. Select OK and we will see the zoomed graph in Figure 3.27. As you can see, there is a significant delay (>320 ps) at the port 3. The waveform at port 3 is also deformed. The cross-talk at port 4 is quite significant. However, the cross-talk at port 2 is still small. If you display the s-parameters of the coupled TLNs, you will understand the reason. As it is shown in Figure 3.28, the $|S_{21}|$ is below -45 dB over the whole frequency range. However, $|S_{41}|$ increases steadily and peaks at almost 0 dB at 16 GHz. For coupled long TLNs, there will be some frequency points that we will see a complete through from the port 1 to the port 4 when the TLNs are long enough. From the time domain, the coupled waveform in port 4 repeats faster than the waveforms in port 1 and 3. Certainly, it should contain relatively more high frequency components. The good thing about

MDSPICE is that it can yield results in both frequency domain and time domain and reveal more information to the users. We will end this chapter here.

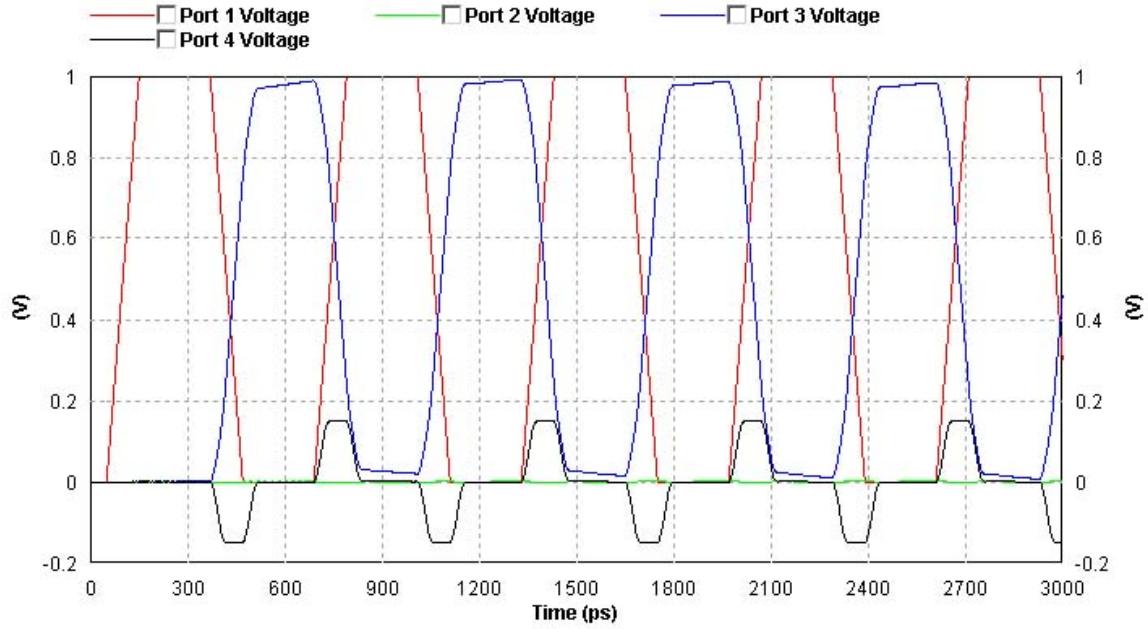


Figure 3.27. The delay, deformation and cross-talks on coupled long TLNs.

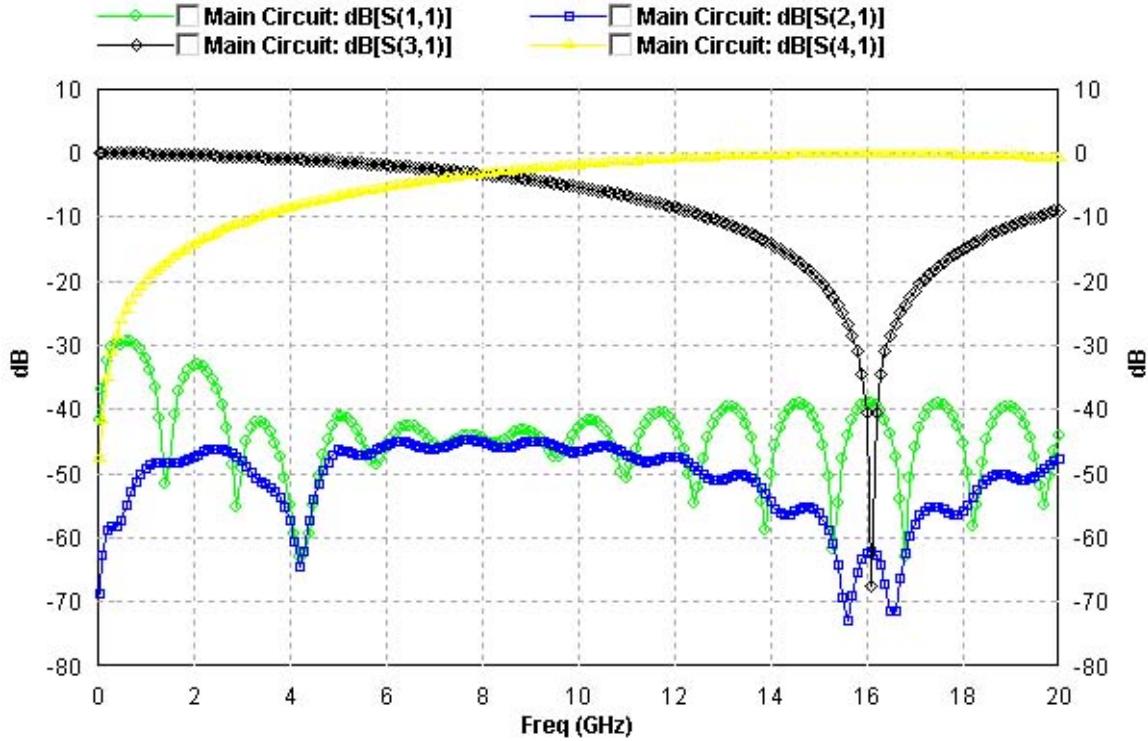


Figure 3.28. The S-parameters of the coupled long TLNs.

Chapter 4. Modeling of Interconnects

The use of the MDSPICE is quite simple. Modeling of interconnects is the same as modeling transmission lines. In this chapter, we will demonstrate two more examples of the MDSPICE in modeling interconnects.

Section 4.1 Modeling of a Coupled Wire Bond Structure.

The structure we are going to model is a coupled wire bond structure shown in Figure 4.1. The ground plane is on $z = 0$ mm. The bond-pads are on $z = 0.25$ mm. The maximum height of the wire bonds is 6 mm. It is relative long. However, we try to use this example to demonstrate some high frequency effects. The center to center separation of the bond pads for the same wire bond is 6 mm. The center to center separation of the bond pads for the adjacent bond pads is 1 mm. The radius of the wire bond is 0.1 mm.

No SPICE simulator can provide accurate models for the wire bonds. Accurate modeling of the wire bonds can be obtained using electromagnetic simulators. What is shown in Figure 4.1 is the geometry in the IE3D Electromagnetic Simulation and Optimization Package. Building such a structure on the IE3D takes couple minutes. We simulate the structure from 0.05 and 0.1 to 20 GHz with 0.1 GHz step for 201 frequency points.

The electromagnetic simulation took less than a minute to finish the s-parameter simulation on a modern PC. If you are an IE3D user, you can try to build the structure and simulate it. We will not go through the IE3D simulation process on the wire bond structure. The geometry file of the structure is saved into “C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\bw2.geo” and the frequency response is saved into “C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\bw2.sp”. We will use the results directly.

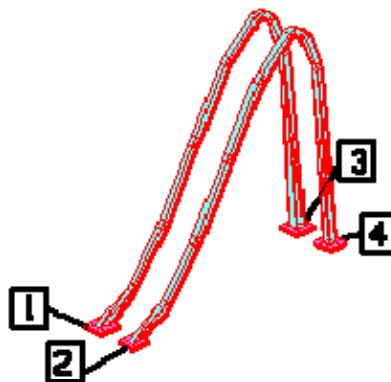


Figure 4.1 A structure consisting of 2 coupled wire bonds.

Section 4.2 Construction of the Model and RLC Extraction of the Structure.

Construction of the circuit is very simple. However, we even do not need to start from the beginning. We can modify the st1d.mds file created in the last chapter for it.

Step 1 Run MDSPICE. Open the file “C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\st1d.mds”. We will have the object list window. Double click at the “SubCkt2” in the list. The “SubCkt2” schematic window is opened. The 4-port TLN module is displayed on it.

Step 2 Click at the 4-port TLN module and select Delete in Edit menu to delete it. We are going to add the 4-port s-parameters module in place of the 4-port TLN module.

Step 3 Select “Add a parameter file” bitmap in the tool bar or Parameter File in Element menu. The st1.sp file is still there. We are not interested in the st1.sp file. Please select Add button. MDSPICE will prompt you for adding a new s-parameter file. Select File Name and select the C:\MentorGraphics\<latest_release>\IE3D\SDD_HOME\IE3D\mdspice\samples\wb2.sp file from the list. MDSPICE will show it has 4-ports and its frequency range is from 0.05 to 20 GHz. Select OK to add the file into the list. Select item “wb2.sp” from the list, and select OK. A 4-terminal element is flying with the mouse. Click anywhere to locate it and it is an Z-call “Z1” to the “wb2.sp”.

Step 4 Select Define All Ports in Element menu. Click at the “SubCkt2” schematic window. MDSPICE will prompt there are 4-spare terminals. Select OK and 4-ports are connected to the 4-spare terminals on the “Z1” element.

Step 5 Select Frequency Domain Parameters in Parameters to let MDSPICE to automatically detect the frequency points in the “wb2.sp”. Select OK to accept the default frequency points.

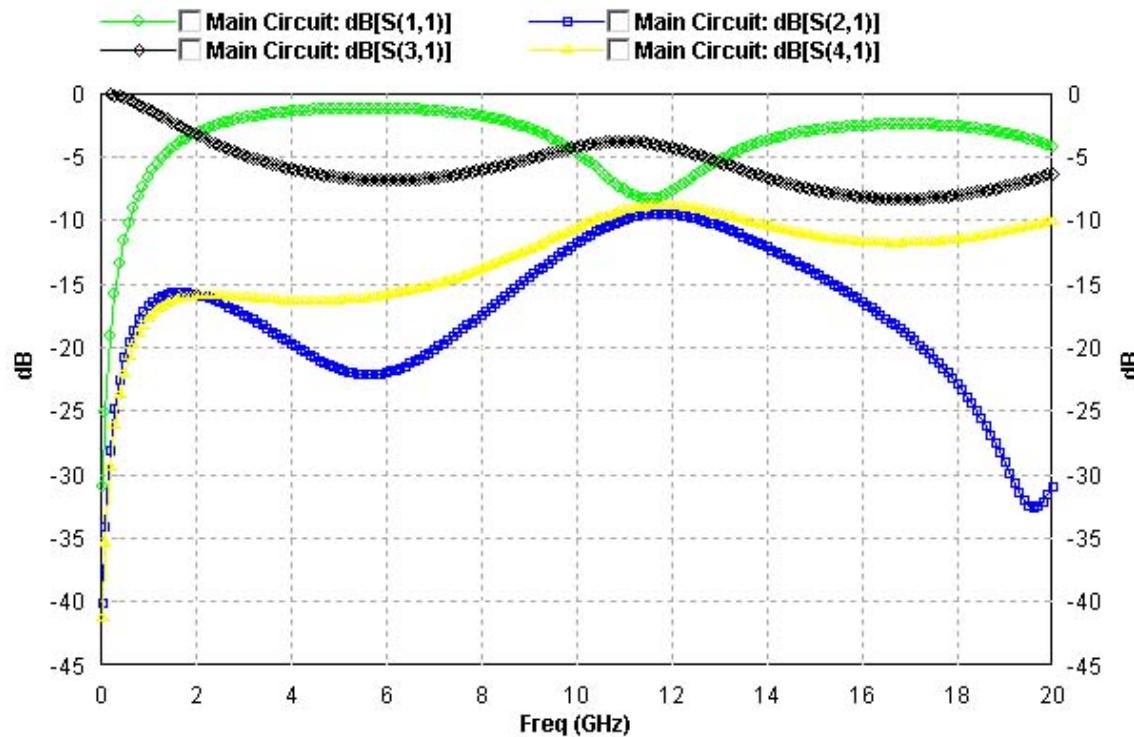


Figure 4.2 The frequency responses of the coupled wire bonds.

Step 6 Select Save As and save the file as: C:\MentorGraphics\<latest_release>\IE3D\SDD_HOME\IE3D\mdspice\samples\wb2a.mds file. Define a graph to display the dB[S(1,1)], dB[S(2,1)], dB[S(3,1)] and dB[S(4,1)] of “main”. The results are shown in Figure 4.2. At low frequency, the wire bonds show the properties of an inductor. However, a resonance is observed at about 11 GHz under the conditions that all the ports are terminated with 50-ohms. The resonance can be shifted with different loading conditions.

In order to perform a traditional SPICE simulation on the circuit, we need to convert the s-parameters into an RLGC equivalent circuit. The question is what frequency range should we expect the equivalent circuit to work. If you require the equivalent circuit working from 0 to 20 GHz, our answer is that it is extremely difficult to get an equivalent circuit for the structure working in so wide a frequency range. If you require an equivalent circuit working at low frequency range, you will be able to do it. The question is how high we can go.

Step 7 Select Parameter Extraction in Process menu. The Find LC-Equivalent Circuit dialog comes up. The user should be informed that the feature for finding LC-equivalent circuit is implemented onto the MODUA. However, the one on MODUA is only good at single frequency. The one on MDSPICE does optimization over a frequency range.

Step 8 Select MDSPICE FILE button. You will be prompted to change the default file name. Change the file name to: C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\wb2am1.mds. You will see the Intermediate File name is also changed to: C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\wb2am1.lib. This file is the resulting equivalent circuit in standard SPICE netlist format.

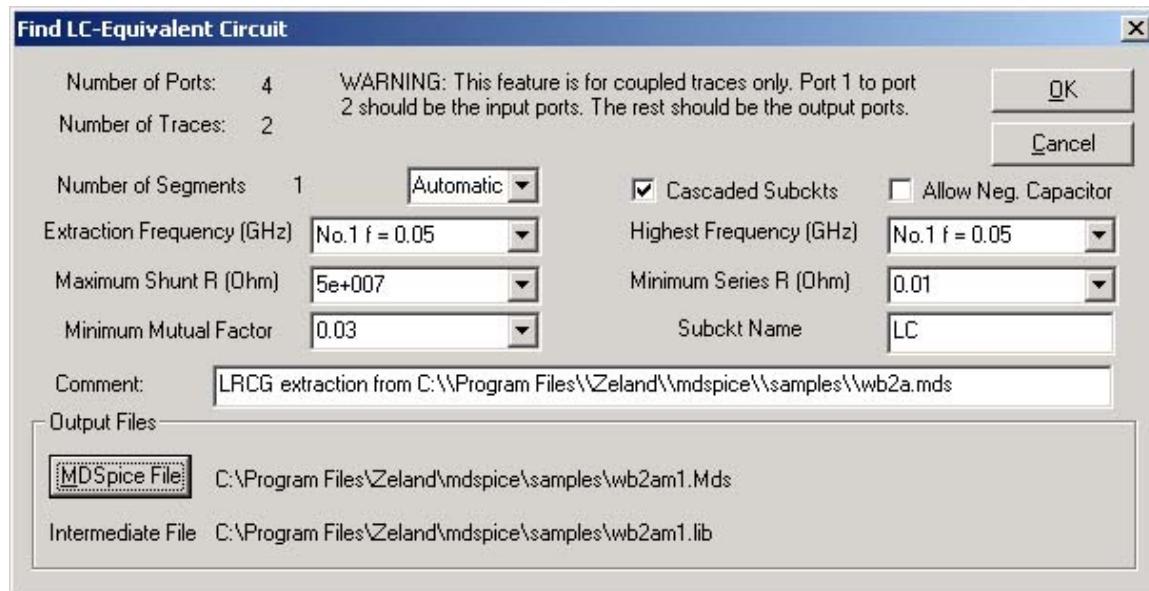


Figure 4.3 The Find LC-Equivalent Circuit dialog after Step 8.

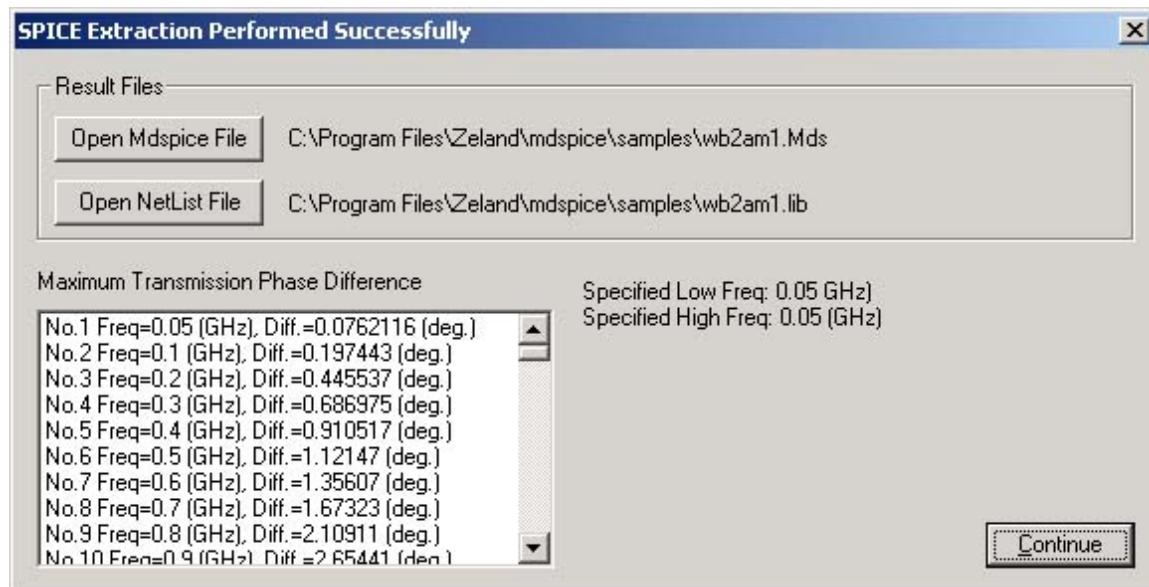


Figure 4.4 The post-extraction dialog.

- Step 9 Please note that the Extraction Frequency = No.1: $f = 0.05$ GHz and the Highest Frequency = No.1: $f = 0.05$ GHz. It means that we will extract it at 0.05 GHz and we want it to be good up to 0.05 GHz. MDSPICE automatically detects that we only need 1-segment for the LC-circuit (Number of Segments = 1). Select OK. MDSPICE will perform the extraction Then, it will show the maximum transmission phase difference in the post-extraction window. As you can see from Figure 4.4, the phase difference between the original s-parameters and the extracted circuit (or the “Diff” in the list box) is increasing with frequency, meaning that the extracted circuit matches the original s-parameters only good below some frequency point.
- Step 10 Select Continue. The equivalent circuit will be saved as: wb2am1.mds, and the standard SPICE netlist is saved in: wb2am1.lib. The “Diff” in the post-extraction dialog only gives you a rough estimate how good the extracted circuit matches the original s-parameters. If you want to check more carefully on the agreement, we should compare the s-parameters of “wb2am1.mds” with the “wb2.sp”. We will do so later.
- Step 11 Select Parameter Extraction in Process menu. The Find LC-Equivalent Circuit dialog comes up again. This time, please select Highest Frequency = No.31: $f = 3$ GHz. MDSPICE will automatically set the Number of Segments = 10. Select MDSPICE FILE button and change the file name to: C:\MentorGraphics\<latest_release>\IE3D\SDD_HOME\IE3D\mdspice\samples\wb2am2.mds. The SPICE netlist is changed to wb2am2.lib too. Select OK. MDSPICE will start the extraction with optimization. It will take a few seconds. Then, it will come up with the post-extraction dialog. You will see the “Diff” or the maximum transmission phase difference is much reduced. For the 1-segment equivalent circuit, “Diff=1.12 degrees” at 0.5 GHz. For the 10-segment equivalent circuit, :”Diff=0.256 degrees” at 0.5 GHz. Select Continue. MDSPICE will save the extracted circuits.

Section 4.3 Verification of RLC Equivalent Circuit

We have extracted the equivalent circuit in the last section. We would like to check how good the equivalent circuit matches the original s-parameters.

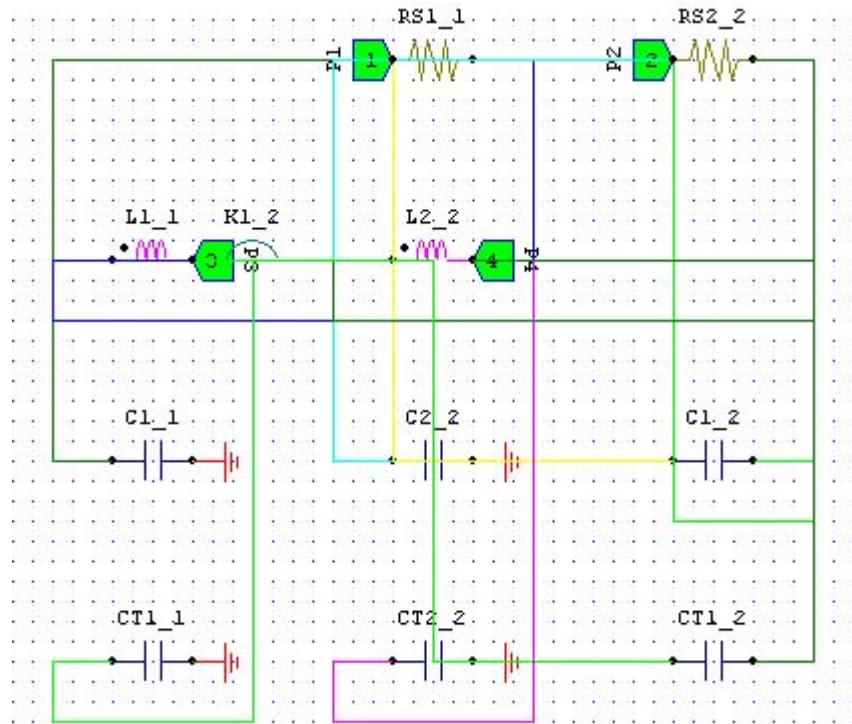


Figure 4.5 The 1-segment equivalent circuit.

Step 1 Run MDSPICE. Open the file “C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\wb2am1.mds”, or the 1-segment equivalent circuit. You will have a long list in the object list window. If you want to know how the circuit topology looks like, please click at the “LC” sub-circuit, the 1-segment equivalent circuit as a sub-circuit is shown in Figure 4.5. If you want to show the values of the RLCG elements, please select Display Detail Symbols in View menu. MDSPICE will show all the detail about the elements (see Figure 4.6).

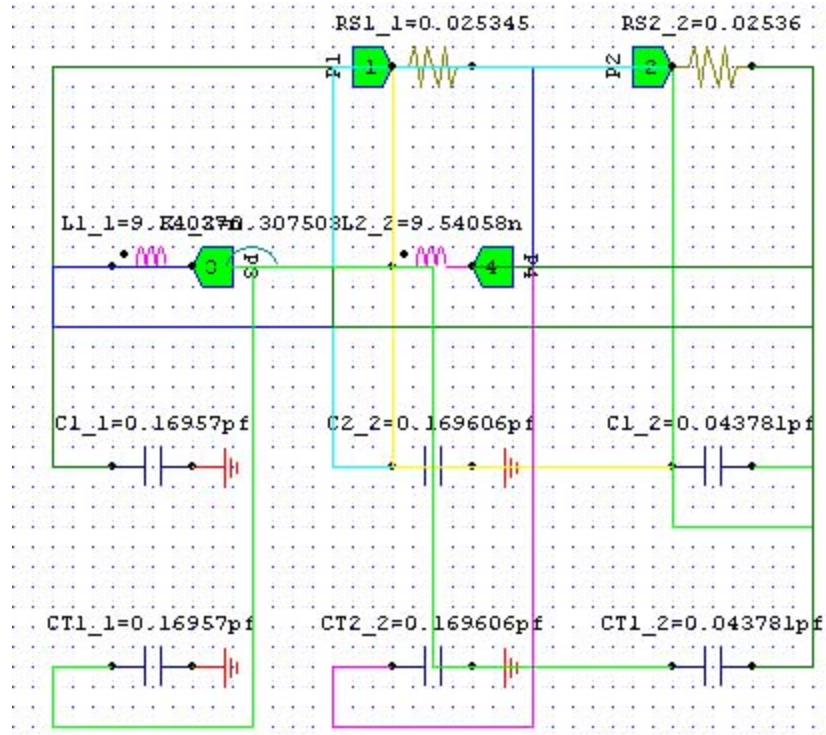


Figure 4.6 The 1-segment RLCG circuit with detail on elements.

- Step 2 Select S-Parameters in Display menu. The Parameter Display dialog comes up. We can define some graphs to display the results. However, we just want to save the s-parameters of the circuit first. We will compare them later. Select the “main circuit” in the Internal S-Parameters. You will notice the Save button is activated. Select Save and enter the name: wb2am1.sp to save the results of the 1-segment into file: C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\wb2am1.sp. Select OK to exit the Display S-Parameters dialog.
- Step 3 Select Open and select the file name: C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\wb2am2.mds file. The content of the file is listed in the object list window. You will see there are 10 sub-circuits: LC0, LC1, LC2,..., LC9. Each of them is similar to the LC in the wb2am1.mds. Also, their RLGC values are not the same.
- Step 4 Select S-Parameters in Display menu. It will take MDSPICE couple seconds to finish the simulation of the complete circuit. Then, the Parameter Display dialog comes up with the s-parameters of all the sub-circuits and the main circuit of wb2am2.mds listed in the Internal S-Parameters list box.
- Step 5 Select Add button on the External S-Parameters list box (on the upper right corner of the list box). Select the “wb2.sp” to add it into the list. It is the original s-parameters. Select Add button on the External S-Parameters list box again. Select “wb2am1.sp”. The s-parameter file for the 1-segment is added into the list. By now, we have the 10-segment s-parameters denoted as “Main Circuit” in Internal S-Parameters. The original s-parameters and the 1-segment s-parameters are denoted as “wb2” and “wb2am1”, respectively in the External Parameters. We are going to define a graph to compare them.
- Step 6 Select Add button in the “Graph to be displayed” list box. Select S-Parameters. Select the “Root=Main Circuit”. Select the dB[S(1,1)], dB[S(2,1)], dB[S(4,1)] and the Ang[S(3,1)]. Select “Root=wb2” and Select the dB[S(1,1)], dB[S(2,1)], dB[S(4,1)] and the Ang[S(3,1)]. Select “Root=wb2am1” and Select the dB[S(1,1)], dB[S(2,1)], dB[S(4,1)] and the Ang[S(3,1)]. Select

OK until you get the display graph. Figure 4.7 shows the comparison. As you can see, the 10-segment “Main Circuit” matches the original s-parameters quite well up to 8 GHz, while the 1-segment “wb2am1” matches the original s-parameters quite well up to 3 GHz. At low frequency, the 1-segment equivalent circuit is a good description of the circuit. However, if we want to expand the applicable frequency range of the circuit, we do need more segments. Can we use even more segments to match the original s-parameters up to 20 GHz? The answer is that there is a limit. Up to some degree, more segments will not help to match the original s-parameters better.

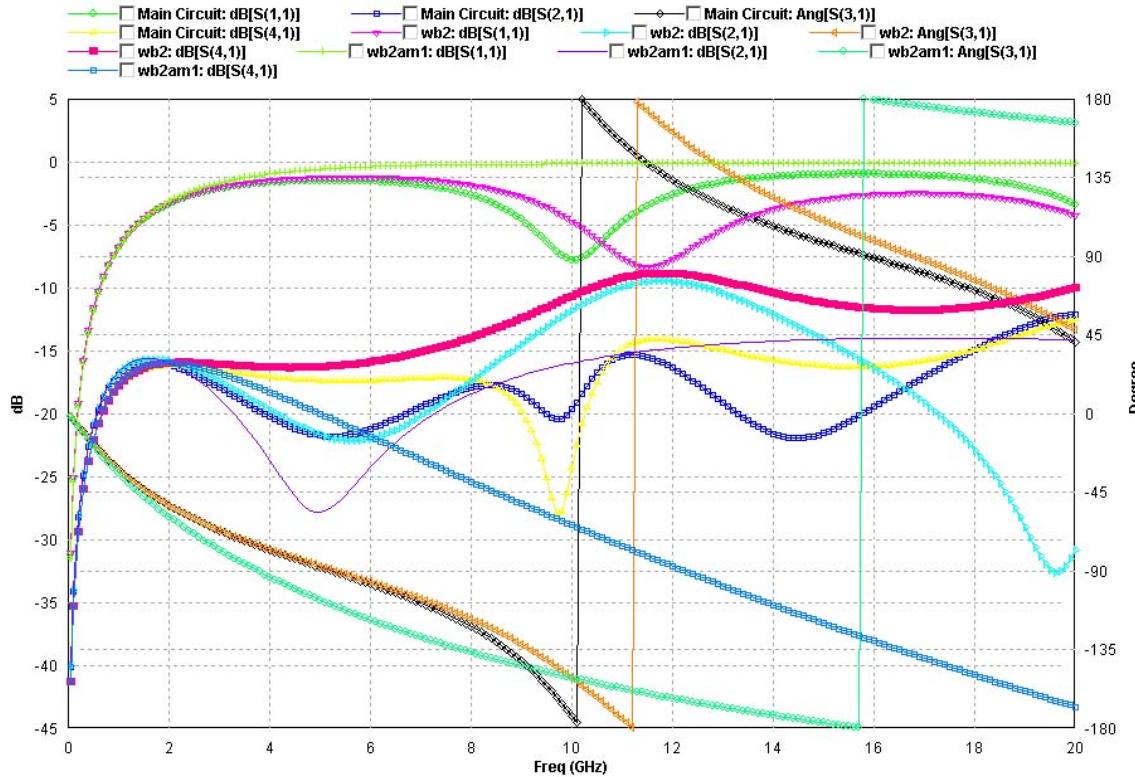


Figure 4.7 The comparison between the original, 1-segment and 10-segment equivalent circuits.

Section 4.4 Time Domain Simulation on the Coupled Wire Bonds

We can use the extracted RLGC circuit in a standard SPICE simulator. The results should be good if we do not apply high clock rate stimulus on it. However, to preserve high accuracy for high clock rate signals, we should use the MDSPICE to simulate the circuit on the original s-parameters.

Step 1 Run **MDSPICE** Open C:\MentorGraphics\<latest_release>\IE3D\SDD_HOME\IE3D\mdspice\samples\wb2.sp. Select Time Domain Parameters in Parameters dialog. Select the only item “Clock Waveform...” in the list. Select Add button and select the “One Clock Waveform” in the list. Enter the parameters as shown in Figure 4.8. The Rise Time of the clock waveform is 128 time steps and 320 ps. Select OK until you exit from the time domain parameters dialog. The waveform of the excitation source is shown in Figure 4.9.

Step 2 Select Time Domain Simulation in Process menu. The SPICE simulation setup dialog comes up. Select the Add button 3 times to define total 4 excitations. Select “No.1 Excitation”, “No.2 Excitation”, “No. 3 Excitation” and “No. 4 Excitation” and define the amplitude and internal resistor as shown in Table We have defined 4 excitations to the structure (see Figure 4.5). The excitations are listed in Table 4.1. For the excitation 1, we apply the source to the port 1 with no internal resistor. The ports 2, 3 and 4 are terminated with 50-ohms. The termination impedance is

changed to 200, 800 and 3200 ohms in the excitations 2, 3 and 4. We will simulate the circuit with 4 different excitations and compare their results.

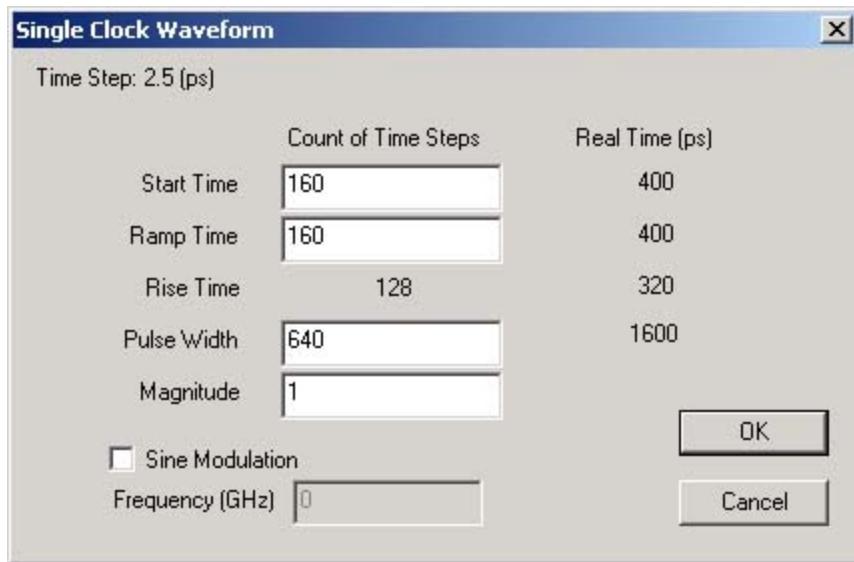


Figure 4.8 The “One Clock Waveform” defined.

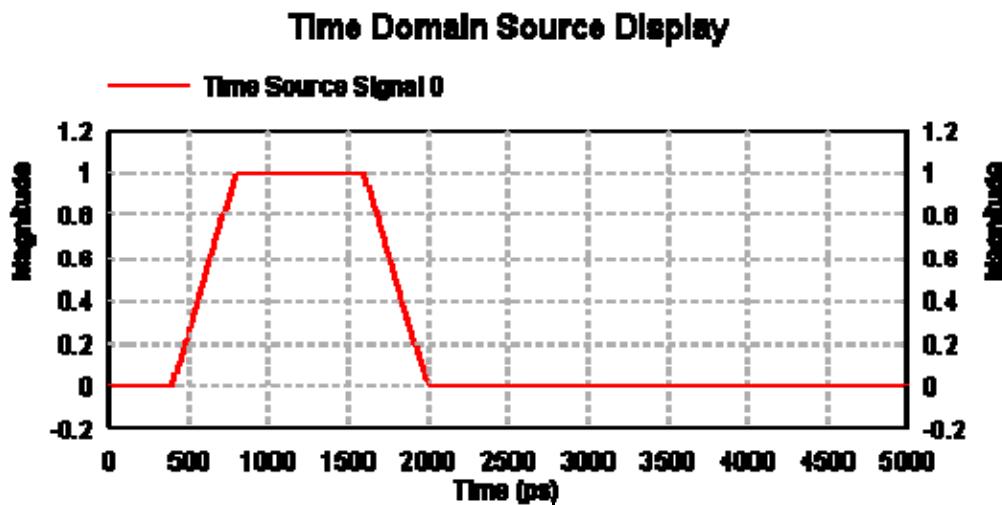


Figure 4.9 The waveform of the excitation source.

Table 4.1 The 4-different excitation combinations.

	Port 1	Port 2	Port 3	Port 4
Excitation 1	Mag=1, R=0	Mag = 0, R = 50	Mag = 0, R = 50	Mag = 0, R = 50
Excitation 2	Mag=1, R=0	Mag = 0, R = 200	Mag = 0, R = 200	Mag = 0, R = 200
Excitation 3	Mag=1, R=0	Mag = 0, R = 800	Mag = 0, R = 800	Mag = 0, R = 800
Excitation 4	Mag=1, R=0	Mag = 0, R = 3200	Mag = 0, R = 3200	Mag = 0, R = 3200

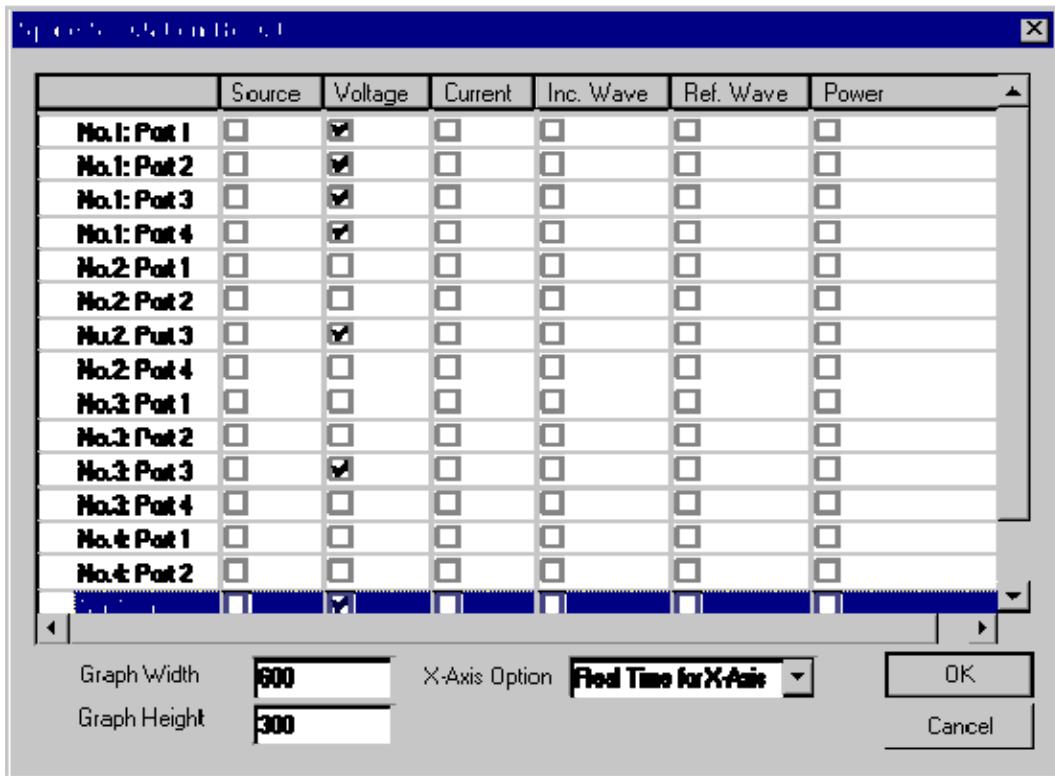


Figure 4.10 The selection of items to be displayed.

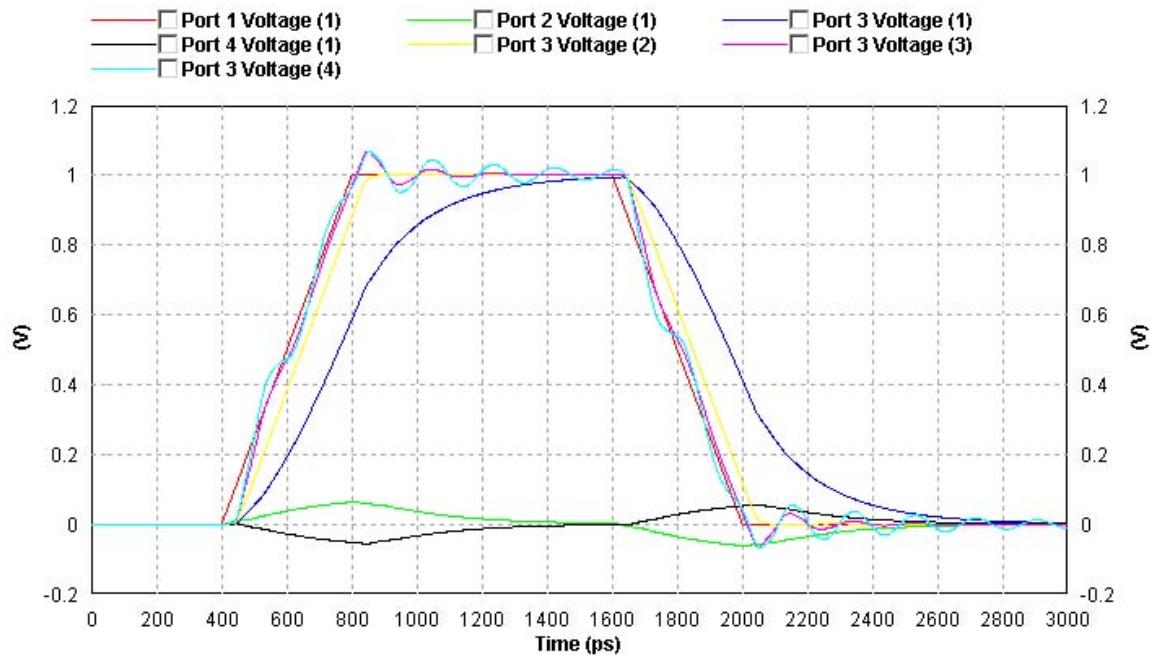


Figure 4.11 The time responses at the ports for different excitations with rise time of 320 ps.

Step 3 Select OK. The simulation on all the 4 different excitations will be done in less than one second. Select OK to proceed to time domain display. Select the following items: No.1: Port 1 Voltage, No.1: Port 2 Voltage, No.1: Port 3 Voltage, No.1: Port 4 Voltage, No.2: Port 3 Voltage, No.3: Port 3 Voltage and No.4: Port 3 Voltage (see Figure 4.10). No.1: Port 1 is the excitation waveform.

No.1: Port 3, No.2: Port3, No.3: Port 3, and No.3: Port 4 are the waveforms of the transmitted voltage at the port 3 for the 4 different excitations and loads. No.1: Port 2 and No.1: Port 4 are the coupling between the port 1 and the port 2(4) for the 1st excitation. Select OK to continue. MDSPICE will display the time signals in Figure 4.11. We can also select the Display Time Domain Result Spectrum and we can get the picture in Figure 4.12.

As you can see, for the No.1 excitation with 50-ohm terminations (or $R = 50$ ohms), the output signal at port 3 is quite damp. With increasing value of resistance for the terminations, we can see the output signal at port 3 is kind of approaching the input waveform. However, we can see there are some ripples in the waveforms when the R value becomes big. The ripples are caused by the resonance around 5 GHz for the wire bonds. On the spectrum display, we can see that the output spectrum at port 3 is significantly higher than the input spectrum at about 4 to 6 GHz.

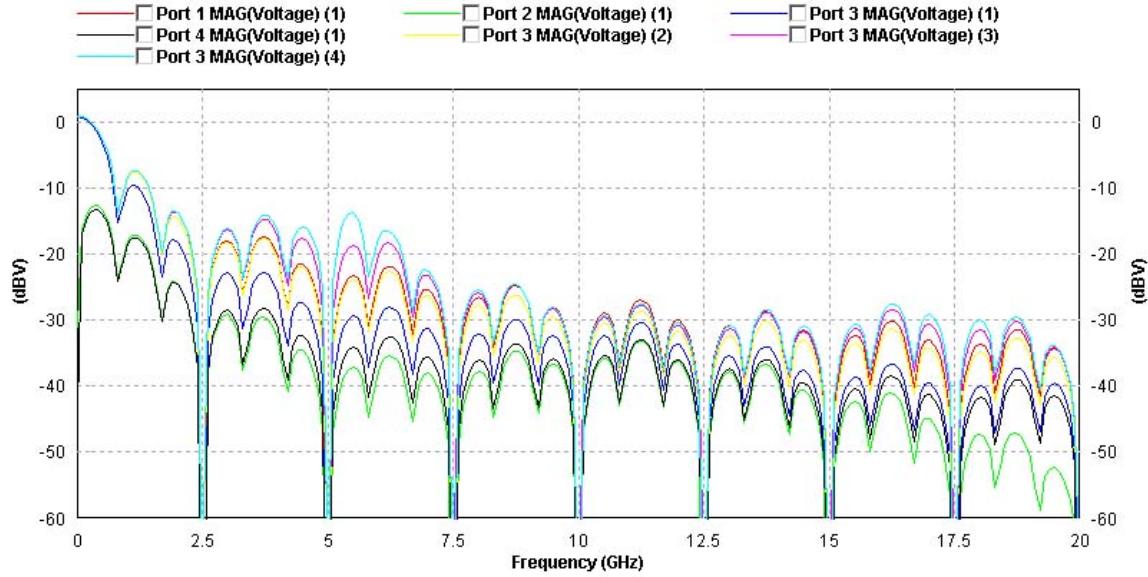


Figure 4.12 The spectrum of the time signals for different excitations with rise time of 320 ps.

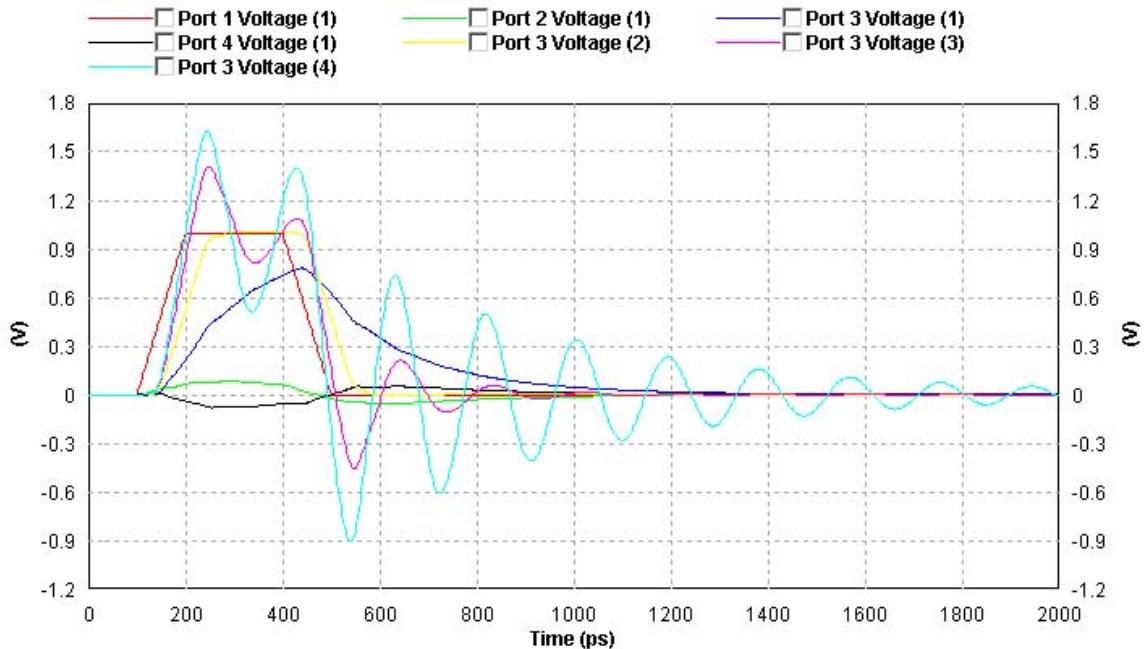


Figure 4.13 The time signals with excitations of rise time 80 ps.

- Step 4 Select Time Domain Parameters in the Parameters menu. Select the only time source in the list box for the Time Domain Source Types list box. Select the Change button. Select Decrease by Factor of 4. Select OK. You will see the Rise Time of the time source is changed from 320 ps to 80 ps. Select OK to continue.
- Step 5 Select Time Domain Simulation in Process menu to simulate it again. Display time domain responses in Figure 4.13 and the spectrum of the time signals in Figure 4.14. As you can, there are many oscillations in the time domain responses. From the spectrum analysis, we can see the spectrum of the Port 3 Mag(Voltage) for excitation 4 is much higher than the input spectrum at about 5.3 GHz. Some resonance must be going on at about 5.3 GHz. We can perform a frequency domain investigation on it.

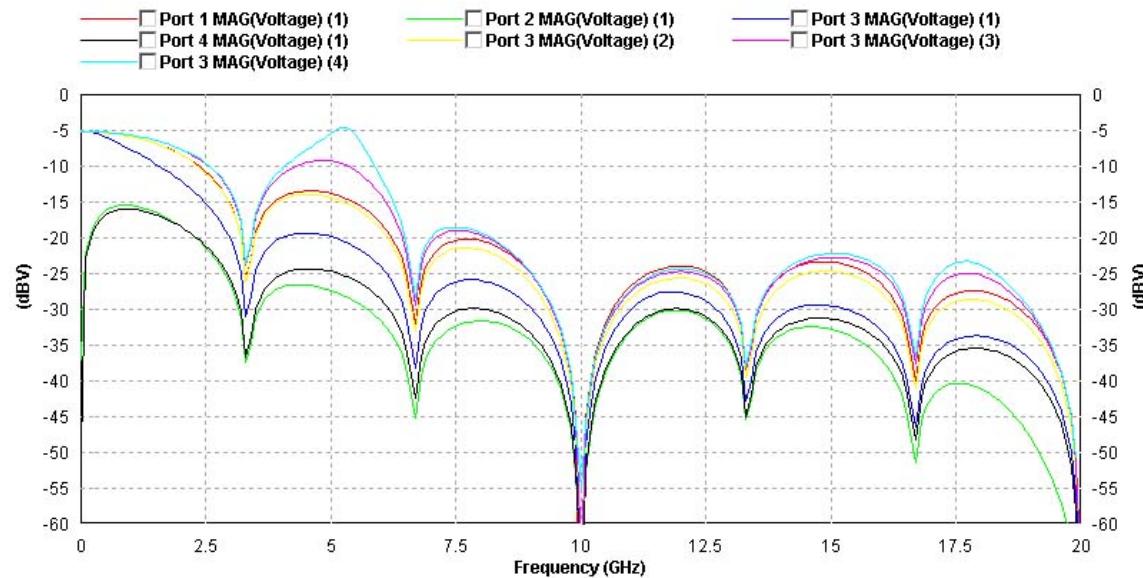


Figure 4.14 The output spectrum of the wire bonds with the input signals of rise time 80 ps.

Section 4.5 Frequency-Domain Investigation on the Resonance.

- Step 1. Run MODUA. Select Display Parameter Module. Select the file C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\wb2.sp. MODUA will display the s-parameters of the file in Figure 4.2. Select Display Toggle. We will see the picture in Figure 4.15a. Click at the ports 2, 3 and 4 to select them. Select Delete Module in Edit menu of MODUA to delete the selected 3 ports. The resulting circuit is shown in Figure 4.15b. Select Termination in Element menu. You will be prompted for the Termination Resistor. Please enter “3200” ohms for it. MODUA is setup to take the termination you are going to define. You will see a moving module is following your cursor.
- Step 2 Click the right mouse button once. The moving module rotate 90 degrees. This step is not necessary on MODUA. It is suggest for clearer display purpose on MODUA. It will not affect the end results. Click at the terminal 2 of module “wb2”. A termination resistor of 3.2K is connected to the terminal 2 of module “wb2”.
- Step 3 Click at the terminal 3 of module “wb2”. Another identical termination is created at terminal 3 of “wb2”. Click at the terminal 4 of module “wb2”. The 3rd termination of 3.2K is connected to the terminal 4 of “wb2”. Click the left mouse button any empty spot in the MODUA window. MODUA will prompt you that “No terminal is selected” and whether you want to repeat. Select “No” to exit the adding termination mode. You should get the display in Figure 4.15c.

Explanation:

The procedure discussed above is a faster way to exit the adding element mode. It is equivalent to selecting the Exit Element in Element menu of MODUA. We have finished defining the terminations at ports 2, 3 and 4. We are ready to find the s-parameters or the input impedance of the resulting 1-port structure.

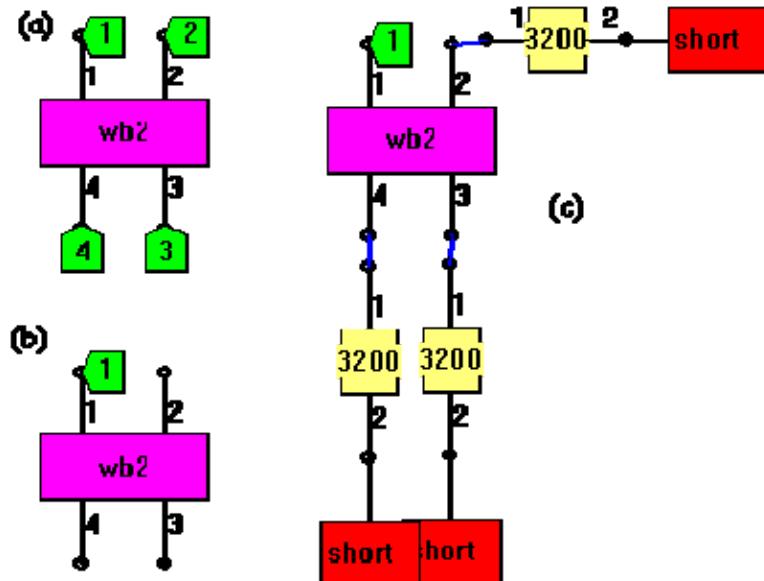


Figure 4.15 The display on MODUA in various steps.

- Step 4 Select Simulate in Process menu of MODUA. After it finishes, please select OK to continue. Select Define Display Graph in Control menu. Select “dB and Phase of S-Parameters”. Select OK and you will be prompted for the display items. Select “dB[S(1,1)]” and select OK. The dB[S(1,1)] frequency response is shown in Figure 4.16. A resonance is clearly seen at about 5.3 GHz. The frequency domain analysis confirms that there is a resonance at 5.3 GHz.

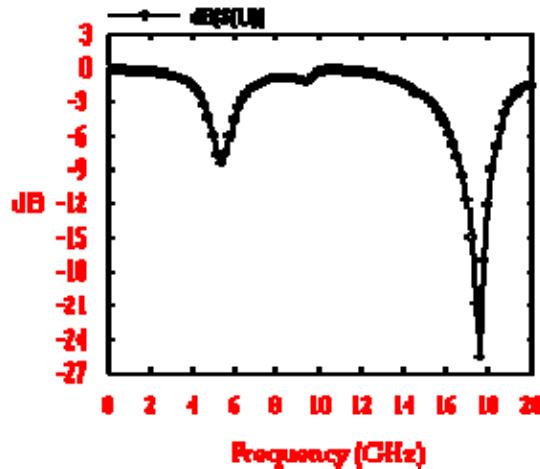


Figure 4.16 The dB[S(1,1)] of “wb2.sp” when the other 3 ports are terminated with 3.2K resistors.

- Step 5 Select Simulate and Find Excitation in Process menu of MODUA. MODUA will simulate the structure again. It prompts you to enter the excitation at port 1. We are going to find the relationship between the input voltage and the output voltages in the frequency domain.

- Step 6 Select the excitation at port 1. Select Change Source button. Type the Voltage Mag = 1 (V) and the Load Re(Z) = 0 ohms. Select Enter and we will get the dialog shown in Figure 4.17. Select OK to continue. MODUA displays all the information about the voltage and current at every terminal of the modules on MODUA.

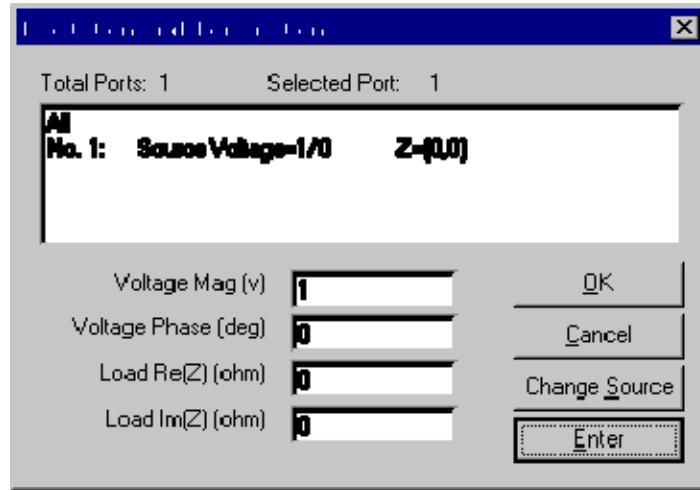


Figure 4.17. The excitation after the change in Step 7.

- Step 7 Select Save Excitation in File menu and select OK to save it into the default file: "wb2.ect". The file format is similar to what is displayed on MODUA. You can use an ASCII editor to open it. Basically, the file contains all the information about the input and output. "wb2.sp" is the 1st independent module. The three 3.2K resistors are the 2nd, 3rd and 4th independent modules. The No.1 terminal of the 1st independent module is the port 1 of "wb2.sp". The No.3 terminal of the 1st independent module is the port 3 of "wb2.sp". We document the voltages at port 1 to port 4 of "wb2.sp" at different frequencies in Table 4.2 based upon the information in the "wb2.ect" file. You may get slightly different results for the trailing digits depending upon which version of the IE3D you are using to create the s-parameters.

As it can be seen, the voltage at port 1 is the excitation. It is kept at 1 V. The output voltage at port 3 is always larger than 1 V from 0.2 to 6 GHz. At 5.4 GHz, the voltage at port 3 is more than 9 times as big as the voltage at port 1. This observation confirms what we have observed in the spectrum analysis of the time domain results in earlier sections.

Interested users can also perform the time domain simulations on the extracted RLCG circuits. The simulation on RLGC will be much slower. Also, you will not see the resonance at 5.3 GHz for the 1-segment RLGC because the resonance is not in its frequency responses. With increasing clock rate, you will see the difference between the time signals between the s-parameters based and the RLGC circuit based simulations.

Section 4.6 Summary on Linear Time Domain Simulation of the MDSPICE.

We have demonstrated the capability of the MDSPICE in modeling linear circuits in the last 3 chapters. As it can be seen, the MDSPICE yields robust and accurate time domain simulation. As the MDSPICE takes a different approach than traditional SPICE simulator, its performance and characteristics are quite different from those of the traditional SPICE simulator.

In an MDSPICE simulation, we are required to provide the frequency response of each element in the netlist. Normally, we should guarantee 99.9% of the power spectrum to be inside the power spectrum. Most signals involved in ICs and semi-conductor circuits have rich low frequency spectrum. We should have the frequency responses covering all the detail at low frequency range, in order to get accurate and robust time domain simulations on the MDSPICE.

Table 4.2 The frequency domain voltages at ports 1, 2, 3 and 4 at the specified excitation.

Frequency (GHz)	Port 1 (V)	Port 2 (V)	Port 3 (V)	Port 4 (V)
0.2	1.000	0.157	1.004	0.157
0.4	1.000	0.251	1.009	0.252
0.6	1.000	0.298	1.018	0.300
0.8	1.000	0.321	1.029	0.326
1.0	1.000	0.334	1.045	0.343
1.2	1.000	0.342	1.064	0.357
1.4	1.000	0.348	1.088	0.369
1.6	1.000	0.353	1.117	0.382
1.8	1.000	0.358	1.151	0.396
2.0	1.000	0.362	1.192	0.412
2.2	1.000	0.366	1.240	0.430
2.4	1.000	0.371	1.296	0.451
2.6	1.000	0.375	1.363	0.475
2.8	1.000	0.381	1.442	0.504
3.0	1.000	0.387	1.536	0.537
3.2	1.000	0.394	1.651	0.578
3.4	1.000	0.402	1.790	0.628
3.6	1.000	0.411	1.963	0.690
3.8	1.000	0.423	2.180	0.769
4.0	1.000	0.437	2.462	0.870
4.2	1.000	0.457	2.836	1.006
4.4	1.000	0.484	3.352	1.193
4.6	1.000	0.522	4.099	1.466
4.8	1.000	0.580	5.234	1.881
5.0	1.000	0.670	7.007	2.533
5.2	1.000	0.772	9.326	3.395
5.4	1.000	0.742	9.917	3.640
5.6	1.000	0.575	7.775	2.880

Chapter 5. Non-Linear Simulation on MDSPICE

Starting from version 2.1, we offer the non-linear modeling capability on the MDSPICE. Non-linear models can be used for either digital or microwave applications. In this chapter, we will discuss the modeling of non-linear circuits on MDSPICE.

Section 5.1 Non-Linear Models on MDSPICE

Non-linear elements are modeled as IV curves. For a typical N-type CMOS transistor, normally we will have the IV curves and properties shown in Figure 5.1 and Table 5.1. On the MDSPICE, we use a XML based file to describe it. A typical example is shown in Table 5.2.

Table 5.1 The properties of N-type CMOS.

Properties and Formulas	Explanation
$Id = 0$ when $V_{gs} < V_{thd}$ $Id = \beta / 2 (V_{ds} - V_{thd})^2, V_{ds} > V_{gs} - V_{thd} > 0$ $Id = \beta [(V_{ds} - V_{thd}) - V_{ds} / 2] V_{ds}, V_{ds} < V_{gs} - V_{thd}, V_{gs} > V_{thd}$	V_{thd} is the threshold voltage. Its typical value is 0.8.

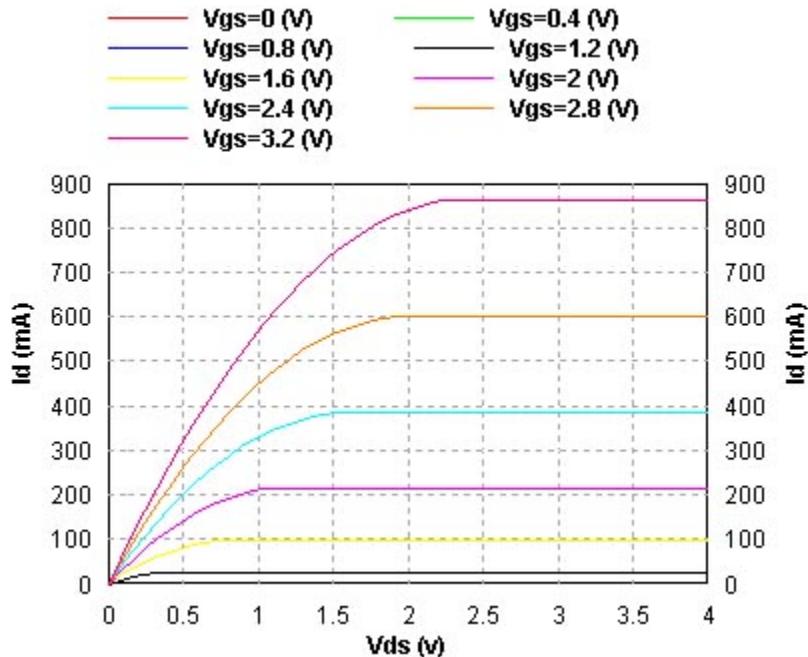


Figure 5.1 The IV curves of a power CMOS transistor (N-type).

Table 5.2 The file syntax for an NMOS transistor on MDSPICE.

The C:\MentorGraphics\<latest_release>\IE3D\SDD_HOME\IE3D\mdspice\samples\nmos1.txt file	Description
<pre> <IE3D filetype="iv" version="1.0" nodes="3" type="NMOS"> <bias> 1.0 1.2 0.0 </bias> <beta> 0.3 </beta> <threshold> 0.8 </threshold> </IE3D></pre>	The nodes = “3” means it is a transistor. The type = “NMOS” means that it is a CMOS N-type device. The threshold is 0.8 v. We also provide the biasing voltages: Vg = V1 = 1 V, Vd = V2 = 1.2 V and V3 = Vs = 0 V. In case, no biasing is provided in the simulation, MDSPICE will use the values for biasing.

For a P-type CMOS transistor, we have the simple model shown in Figure 5.2 and Table 5.3. A typical file is shown in Table 5.4.

Table 5.3 The properties of P-type CMOS.

Properties and Formulas	Explanation
$Id = 0$ when $V_{gs} > V_{thd}$	V_{thd} is the threshold voltage. Its typical value is -0.8.
$Id = -\beta / 2 (V_{ds} - V_{thd})^2, V_{ds} < V_{gs} - V_{thd} < 0$	
$Id = -\beta [(V_{ds} - V_{thd}) - V_{ds} / 2] V_{ds}, V_{ds} < V_{gs} - V_{thd}, V_{gs} < V_{thd}$	

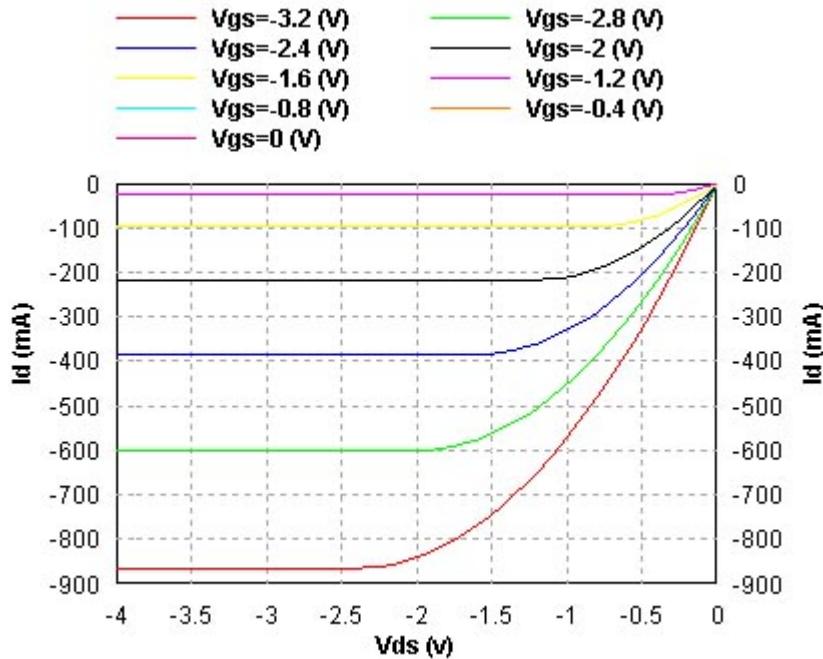


Figure 5.2 The IV curves of a power CMOS transistor (P-type).

Table 5.4 The file syntax for a PMOS transistor on MDSPICE.

The C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\pmos1.txt file	Description
<pre><I filetype="iv" version="1.0" nodes="3" type="NMOS"> <bias> -1.0 -1.2 0.0 </bias> <beta> 0.3 </beta> <threshold> 0.8 </threshold> </IE3D></pre>	<p>The nodes = "3" means it is a transistor. The type = "PMOS" means that it is a CMOS P-type device. The threshold is 0.8 v. We always use the absolute value of the threshold voltage. We also provide the biasing voltages: Vg = V1 = -1 V, Vd = V2 = -1.2 V and V3 = Vs = 0 V. In case, no biasing is provided in the simulation, MDSPICE will use the values for biasing.</p>

Another non-linear device implemented into MDSPICE is the diode. The properties of the diode are shown in Table 5.5 and Figure 5.3. The file syntax on a diode is shown in Table 5.6.

The non-linear element models implemented into the MDSPICE 2.1 are still limited. We welcome users to provide the IV curves for those devices they want to be implemented into the MDSPICE. We will

try our best to implement them as soon as possible in case there is such a request. We will go for some examples of non-linear simulation.

Table 5.5 The properties of a diode.

Properties and Formulas	Explanation
$I_d = 0$ when $V_d < 0$ $I_d = I_s \exp[V_d/(N_e V_t)]$, $V_d > 0$	I_s is the saturation current. N_e is the emission coefficient. V_t is the thermal voltage.

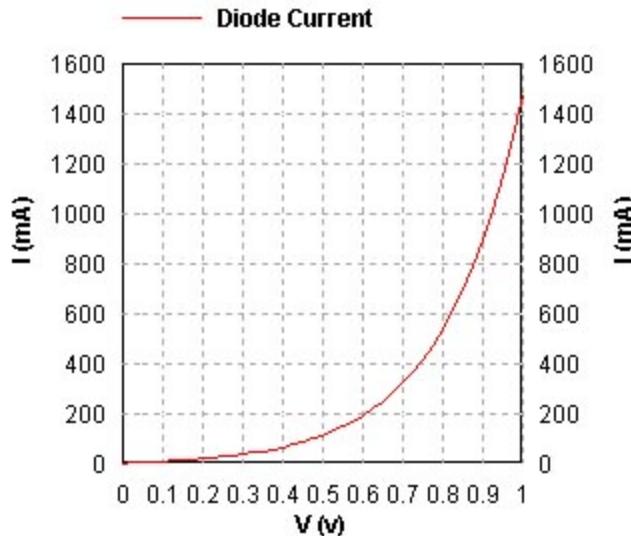


Figure 5.3 The IV curve of a diode.

Table 5.6 The file syntax for a diode on MDSPICE.

The C:\MentorGraphics\<latest_release>IE3D \SDD_HOME\IE3D\mdspice\samples\diode1.txt	Description
<IE3D filetype="iv" version="1.0" nodes="2" type="NMOS"> <bias> 0.3 0.0 </bias> <thermalvoltage> 0.2 </thermalvoltage> <saturationcurrent> 0.01 </saturationcurrent> <emissioncoefficient> 1.0 </emissioncoefficient> </IE3D>	The nodes = "2" means it is a diode. The type can be either "NMOS" or "PMOS" and it will not affect the properties. The type has to be there. The bias voltage at the positive terminal is 0.3 V and bias voltage at the negative terminal is 0 V. They will be used only when there is no biasing for the circuit.

Section 5.2 Non-Linear Simulation with Real-Time DC Biasing.

Non-linear devices are normally working at some DC biasing. In frequency domain simulation, we normally assume some pre-defined biasing. On the MDSPICE, we can define some biasing in the IV curve files describing the devices (see Tables 5.2, 5.4 and 5.6). However, it is not necessary to do so on the MDSPICE. We can define the biasing as the true excitation. We will discuss an examples using pre-defined biasing later. Here, we will discuss the case with the DC biasing as the excitation. The example circuit is shown in Figure 5.4. The non-linear device is an N-type CMOS transistor. The $R_1 = 3$ K ohms, $R_2 = 2$ K ohms and $R_3 = 100$ -ohms are the biasing resistors. The dashed line rectangle is the external circuit. The $R_4 = 2$ K ohms is the internal resistor of the source 1. V_2 is the DC-biasing voltage source of 4 V connected to terminal 2. $R_5 = 100$ ohm is the load at the terminal 3. Two sources are defined for the circuit (see Figure 5.5). The No.0 source is the Clock Waveform with rise time of 240 ps (see Figure 5.6). The magnitude of

the source is 1 volt. The No.1 source is the DC biasing source. Its start time is from No.0 time step. It ramps up to 4 volt smoothly in 20 steps (see Figure 5.7). Before the DC biasing stabilizes, the other sources will not be started in a time domain simulation.

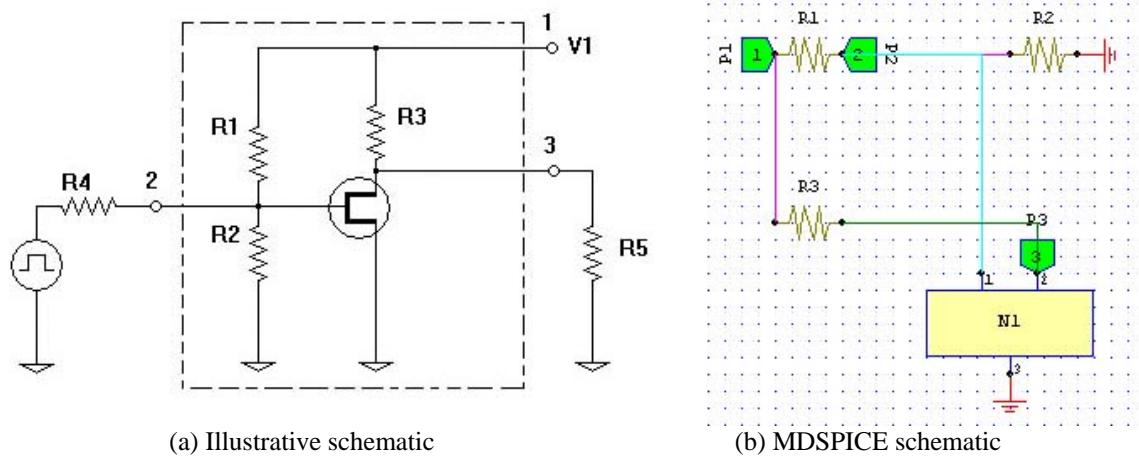


Figure 5.4 The circuit to be analyzed.

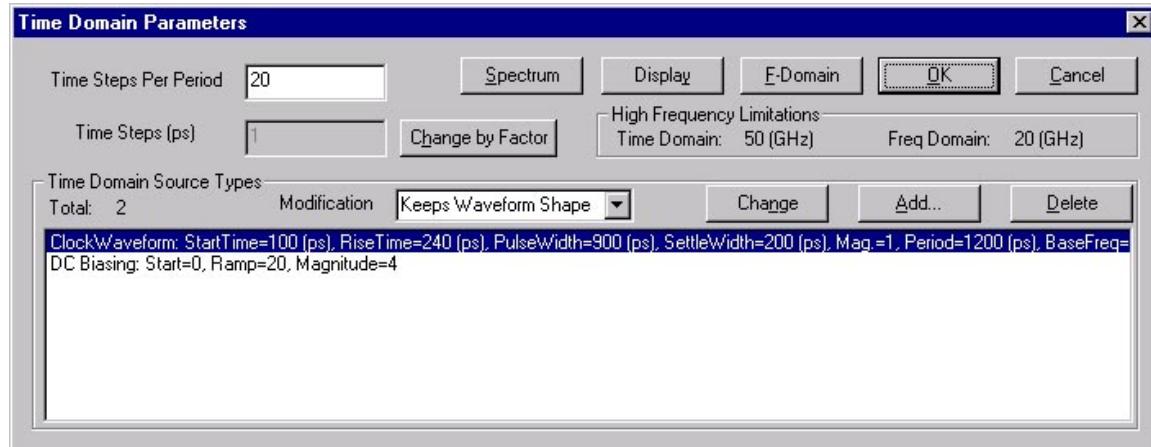


Figure 5.5 The time sources defined for the non-linear circuit.

Step 1 Run MDSPICE. Open C:\MentorGraphics\<latest_release>\IE3D\SDD_HOME\IE3D\mdspice\samples\nmos1b.mds file. Select Time Domain Simulation in Process menu. The SPICE Simulation Setup dialog comes up. The time sources are defined on each terminal. The terminal 1 is connected to the No.1 time source with amplitude = 1 and internal resistor = 0. The No.1 time source is basically the DC biasing of 4 volts. The terminal 2 is connected to the No.0 time source, which is the stimulus, with amplitude = 1 and internal resistor of 2000-ohms. The internal resistor of 2000-ohms is the R4 in Figure 5.4 and it is part of the DC biasing system. Terminal 3 is connected to the No.0 time source with magnitude = 0, meaning there is no source on it. The internal resistor is 50-ohms. It is the R5 in Figure 5.4 and it is also part of the DC biasing system.

There are some new parameters in the SPICE Simulation Setup dialog: Minimum Biasing Time Steps = 20 and the Biasing Steady Criteria = 0.0001. It means that the circuit will start with the DC biasing first. It will take at least 20 time steps for the biasing. When the biasing becomes

steady, it will start the stimulus. The criteria for the steady state are 0.0001 meaning the detected residual is below 0.0001 of the DC voltage. In the SPICE Simulation Setup dialog in Figure 5.8, it shows that the No.1 source has 87.1129% of the spectrum in the bandwidth. It is for the DC biasing signal in finite time length. We just do not need to worry about it.

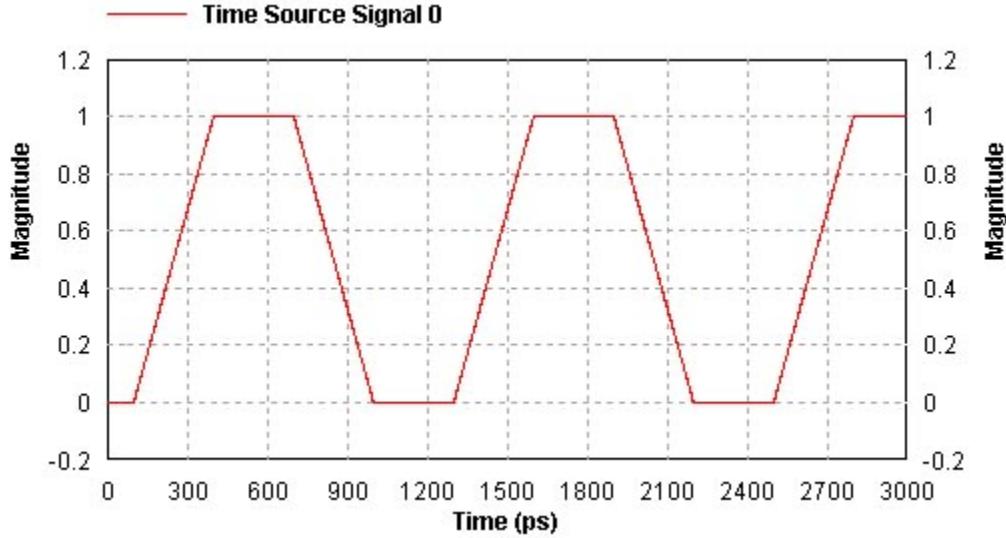


Figure 5.6 The Clock Waveform for the stimulus.

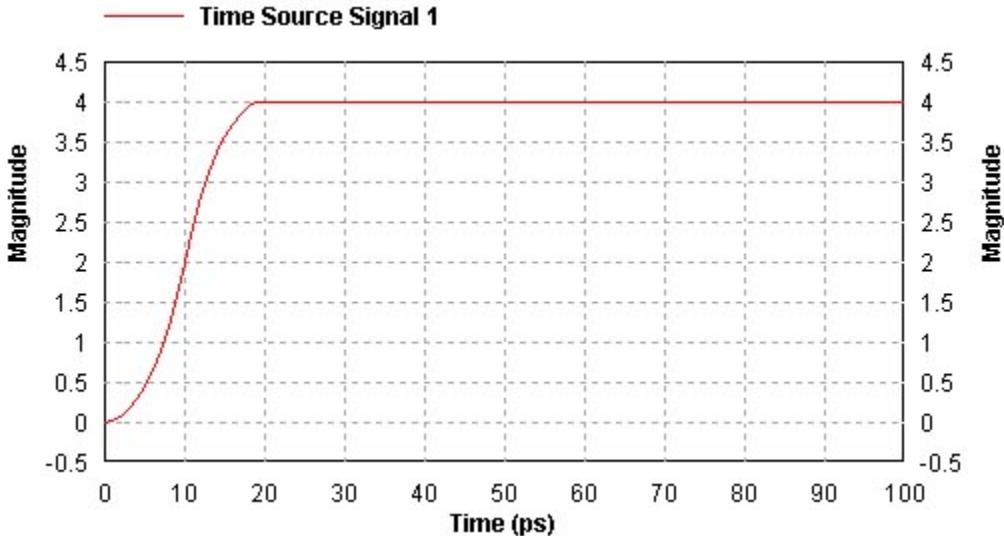


Figure 5.7 The smooth ramp up in the DC biasing.

- Step 2 Select OK to start the time domain simulation. It will finish the simulation in a second. Then, it comes up with the Time Domain Post-Simulation Option dialog (see Figure 5.9). The dialog displays a new parameters: Biasing = 29. It means that it takes about 29 time steps to build up the biasing for the non-linear circuit.
- Step 3 Select OK to start the display. MDSpice will prompt the SPICE Simulation Result dialog (see Figure 5.10). Please note that there is a combo box called DC Biasing at the right corner. The default selection is Remove Biasing. If we select Remove Biasing, we will display the offset values of the V and I at each port from the steady biasing.

Step 4 Change the DC biasing to Keep Biasing. We want to see the complete time signals first. Select No.1: Port 2, Voltage and Current, No.1: Port 3, Voltage and Current. Select OK to continue. MDSpice will display the time signals as shown in Figure 5.11.

Step 5 As you can see, the Port 2 Current is very small. The Port 3 Voltage and Current are slightly deformed due to the non-linearity. Port 3 Voltage is slightly larger than the Port 2 Voltage. The Port 3 Current is much larger than the Port 2 Current. It is the amplification of the input signal. We can not take the Fourier transform to get accurate spectrum for the structure due to the presence of the DC biasing. Only the spectrum of the signals with the DC biasing removed has meaning. Before we do that, we want to check what the biasing voltage and current are.

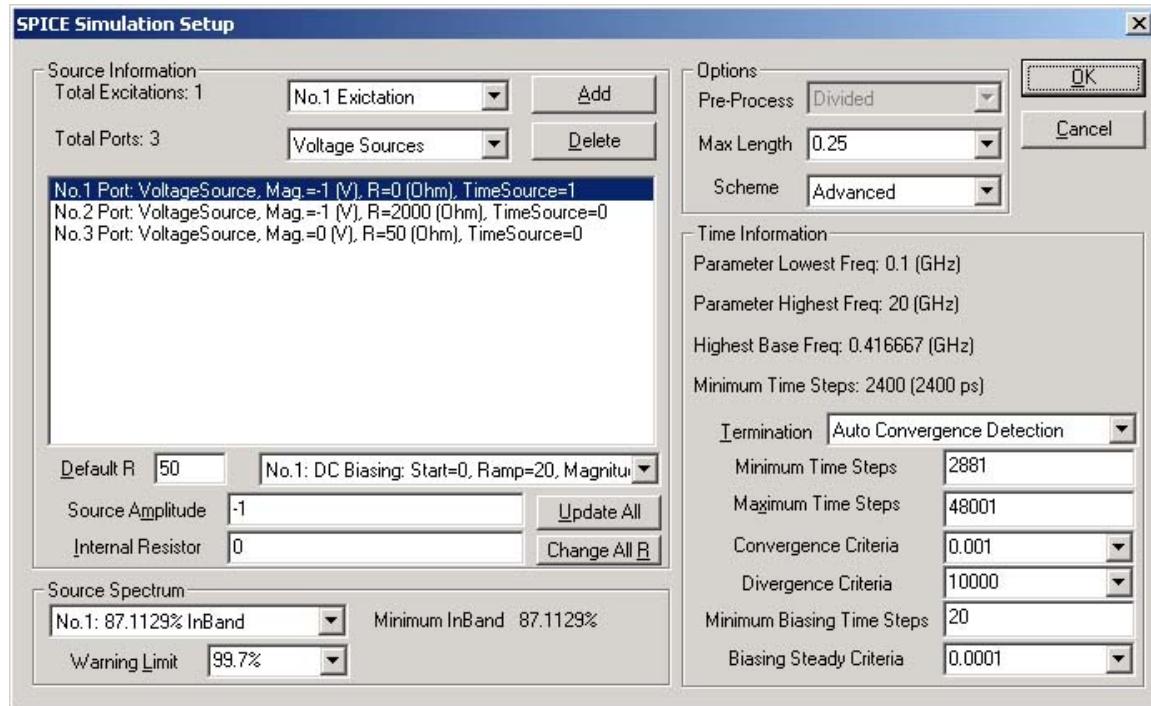


Figure 5.8 The SPICE Simulation Setup dialog.

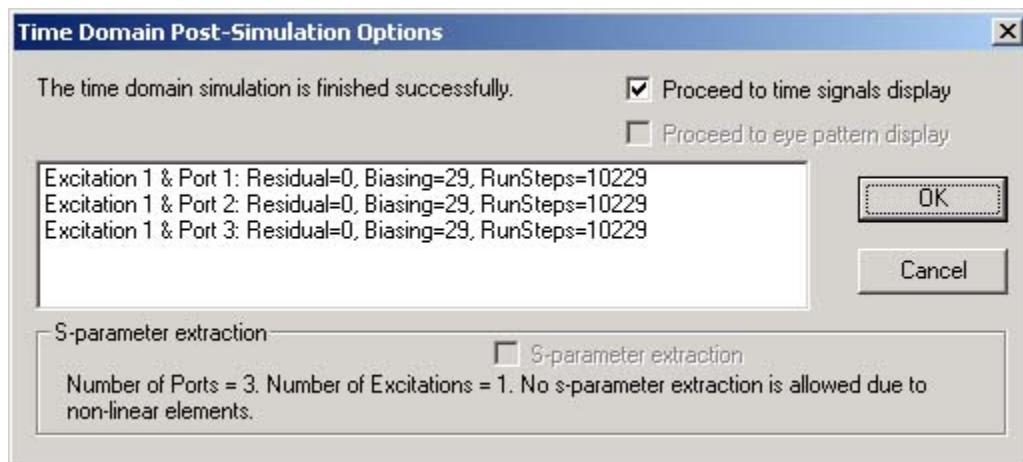


Figure 5.9 The Time Domain Post-Simulation Options dialog with the biasing parameter.

Step 6 Select DC Biasing Voltage and Current in Display menu. MDSpice will show the DC biasing steady state voltage and current at each port to you (see Figure 5.12). At port 1, the DC biasing steady voltage is 4 V and the DC biasing steady current is 0.0296667 A. At port 2, the DC biasing steady voltage is 1 volt and the DC biasing steady current is -0.0005 A. At port 3, the DC biasing steady voltage is 1.13333 volt, and the current is -0.0226667 A. Select Close to close the window.

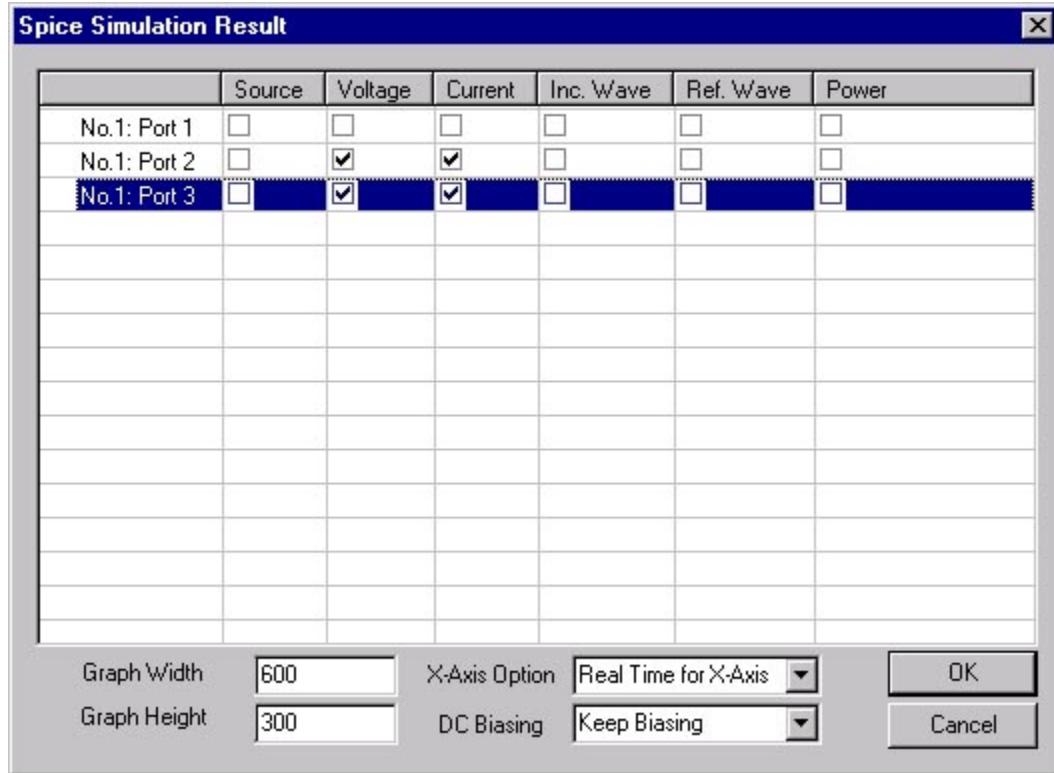


Figure 5.10 The SPICE Simulation Result dialog.

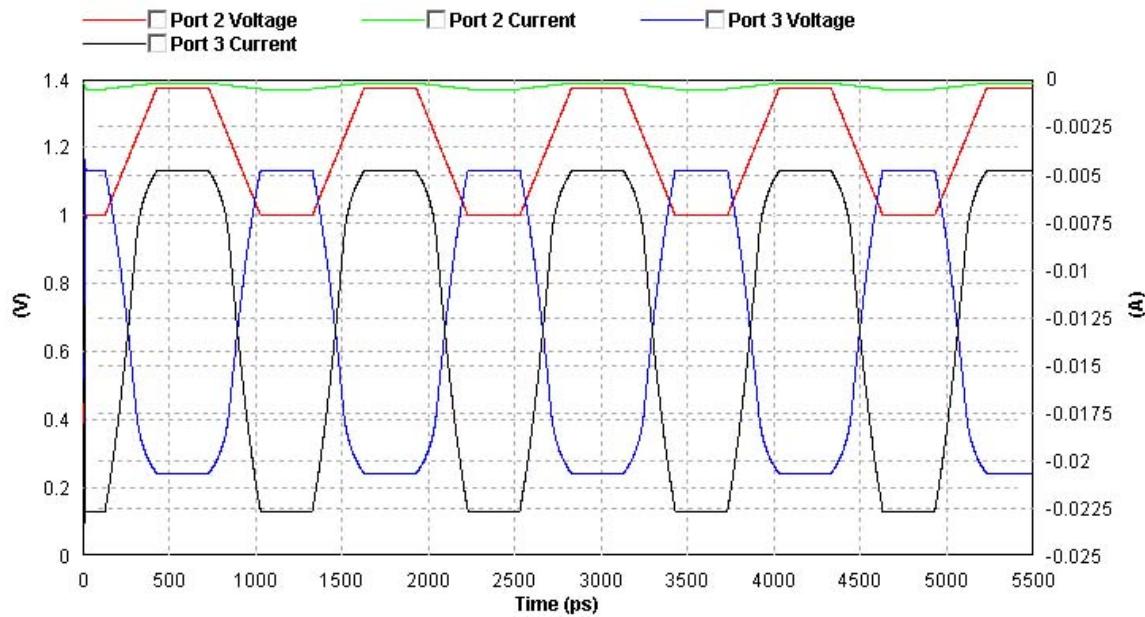


Figure 5.11 The complete time signal including the biasing.

Step 7 Select Time Domain Simulation Results in Display menu. Choose the No1: Port 1 Voltage and Current, No.2: Port 2 Voltage and Current in the list box. Change the DC Biasing from Keep Biasing to Remove Biasing. Select OK to continue. The time signals with the DC biasing removed are shown in Figure 5.13. This time, all the waveform's base is at 0. They are the off-set values from the DC biasing steady state values.

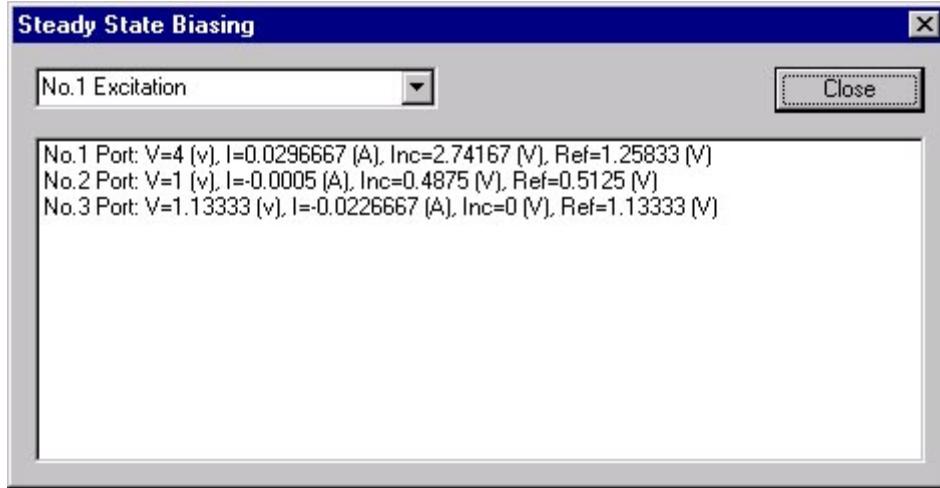


Figure 5.12 The steady state DC biasing voltage and current.

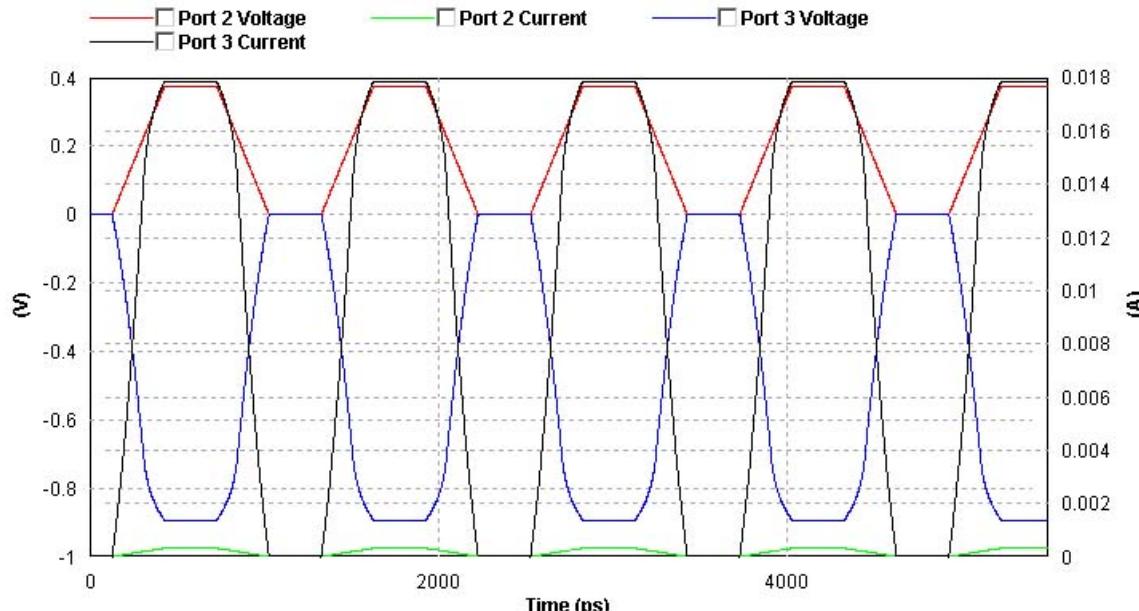


Figure 5.13. The time signals with the DC biasing removed.

Step 8 Select Time Domain Result Spectrum in Display menu. Define the frequency range from -10 to 10 GHz. Select OK. MDSpice will prompt you for the items to be displayed. Remember to make sure the DC Biasing combo box is set to Remove Biasing. The spectrum with DC biasing kept normally does not make much sense. Select the No.1: Port 2 Mag(Voltage), Mag (Current), the No.1: Port 3 Mag(Voltage) and Mag(Current). Select OK to continue. The MDSpice will display the spectrum of the stimulus and the output signals with the DC biasing removed (see Figure 5.14). As you can see, the spectrum of the output signals at port 3 is higher than that of the input signals at port 2. The spectrum is supposed to be discrete. We are displaying it as some lines connecting the discrete points. You should also notice that the output signal does have spectrum at

points where the input signal does not have spectrum ($f=1.66667, 3.3333, 3, \dots$ GHz). These are the non-linear effects.

The corresponding p-type circuit is saved in C:\MentorGraphics\<latest_release>\IE3D\SDD_HOME\IE3D\mdspice\samples\pmos1b.mds. Its setting is completely the opposite of the nmos1b.mds. You will get exactly opposite results. Interested users can try it out. We will not discuss it here.

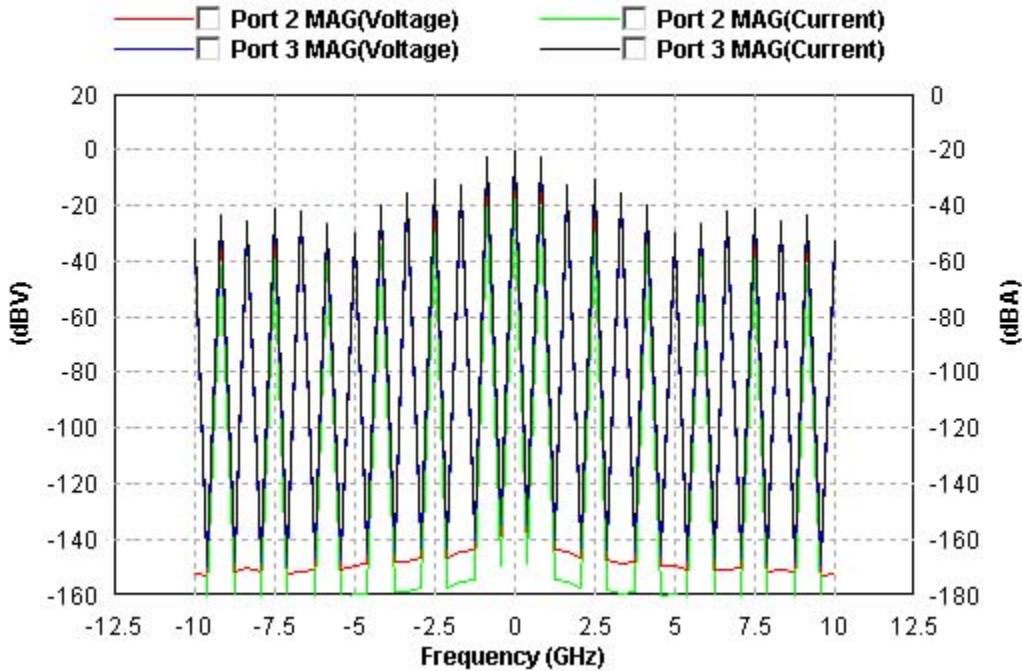


Figure 5.14 The input and output spectrum of the circuit (DC biasing removed).

Section 5.3 Non-Linear Simulation with Pre-Defined DC Biasing.

We have discussed how we can excite a circuit using DC biasing and stimulus. Sometime, we may not need to do the DC biasing. We can use pre-defined DC biasing to perform a time-domain simulation. As it is shown in Table 5.4 and Table 5.6, there is a block in the IV file to describe the DC biasing. In our real-time DC biasing simulation discussed in the above section, the DC biasing from the IV file is automatically dropped. However, if we do not define any DC biasing source in our simulation, the pre-defined DC biasing from the IV file will be used.

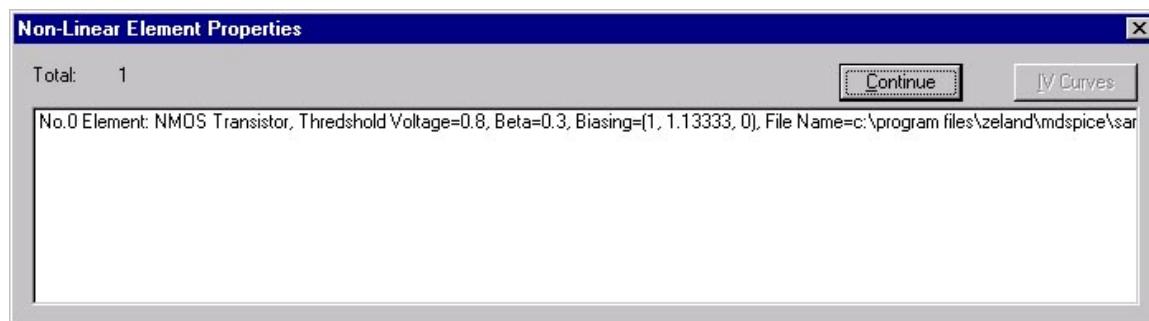


Figure 5.15 The Non-Linear Element Properties dialog.

- Step 1 Open C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\nmos2b.mds file. The content in nmos2b.lib file is almost identical to that in the nmos1b.lib file except the IV file name for the non-linear element is replaced by C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\nmos2.txt file.
- Step 2 Select Non-Linear Element IV Curves in Display menu. MDSPICE will show all the non-linear circuit's properties to you (see Figure 5.15). For this example, the only non-linear circuit element is nmos2.txt. As you can see, its DC biasing is exactly like the one in Figure 5.12. Select Continue.
- Step 3 Select Time Domain Simulation in Process menu. Change the excitation to what is shown in Figure 5.16. Only the No.0 Time Source is used. The No.1 Time Source which is the DC biasing is not defined. When no DC biasing source is defined, MDSPICE will automatically use the pre-defined DC biasing inside each non-linear circuit. Select OK to continue. MDSPICE will finish the simulation in a second. It will prompt you to display the time domain results. This time, you will not see the "Biasing=..." in the list box because there is no biasing procedure in this simulation. The DC biasing is pre-defined. Select OK to continue. MDSPICE will prompt you to select the items. You will also see the DC Biasing combo box is grayed out. What you can see is only the results without the biasing. Select the No.1: Port 1 Voltage and Current, No.1: Port 2 Voltage and Current. Select OK. You will see the exact picture in Figure 5.13.

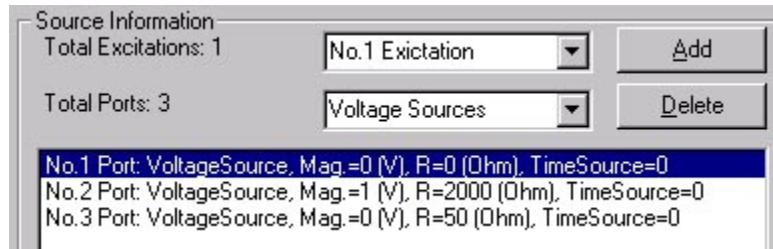


Figure 5.16 The excitation defined in the SPICE Simulation Setup.

The user should understand that, if he uses the pre-defined DC biasing, he needs to provide the correct one. MDSPICE will not check whether the DC biasing is reasonable for you.

Section 5.4 Harmonic Analysis of Non-Linear Circuits

As we have seen, non-linear circuits can create frequency spectrum in those ranges where the input signal does not have (see Figure 5.14). In the above examples, we have used wide band clock signals to excite the circuits. We do not identify the rules for the harmonics created. Let's try it using some narrow bandwidth signal.

- Step 1 Open C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\nmos1c.mds file. The nmos1c.lib file is exactly the same as the nmos1b.lib file. Select Time Domain Parameters. You will see the No.0 Time Source is replaced by a MultiGaussian waveform. There are 2 narrow band Gaussian pulses are defined in the waveform: (1) f0 = 2 GHz , 3dB bandwidth = 0.05 GHz, Magnitude = 0.5. (2) f0 = 3 GHz, 3 dB bandwidth = 0.05 GHz, Magnitude = 0.5. They will be overlapped together. Select OK to continue.
- Step 2 Select Time Domain Simulation. The SPICE Simulation Setup dialog comes up. The excitation source is the same as in the nmos1b.mds simulation. However, the No.0 time source is already changed from the Clock Waveform to the MultiGaussian waveform. The simulation will take seconds. Select OK multiple times to display the No.1: Port 1 Voltage and Current, No.1: Port 2 Voltage and Current with the DC biasing removed. It will take some seconds for the graph to be

displayed due to the large number of data points. You will see some Gaussian like waveforms. However, the waveforms can not tell much information to you.

- Step 3 Select Time Domain Result Spectrum in Display menu. Change the range as: Start Frequency = 0, End Frequency = 10 GHz, and Number of Frequency = 201. Select No.1: Port 1 Mag(Voltage) and Mag(Current), No.1: Port 2 Mag(Voltage) and Mag(Current). MDSPICE will take some seconds to perform the FFT on the time signals. Then, MDSPICE will display the spectrum as shown in Figure 5.17. For the input signals at Port 2, we can see two narrow band Gaussian pulses: one at 2 GHz and the other at 3 GHz. For the output signals at Port 3, we will see many harmonics at different frequencies ($f=0, 1, 2, 3, 4, 5, 6, \dots$ GHz).

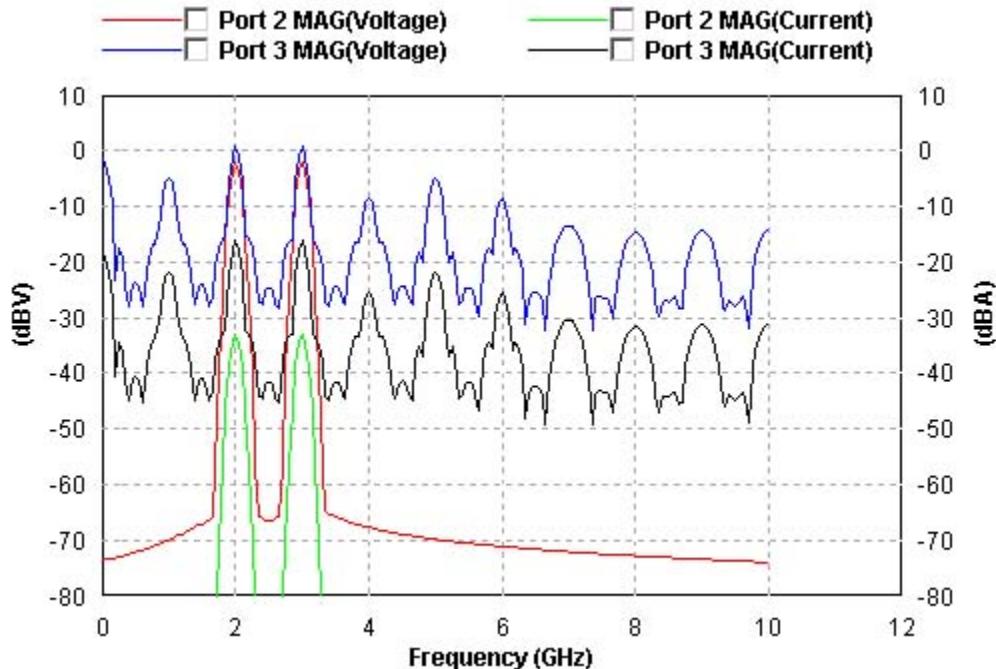


Figure 5.17 The narrow band signals at input and the harmonics created at the output.

All the harmonics are created by the non-linear device. If there is no non-linear device in the circuit, the output waveform spectrum will be very close to the Gaussian spectrum distribution at the input. Filters can be used to filter out unwanted spectrum. We have built more examples with filters and non-linear devices. Saved in C:\MentorGraphics\<latest_release>\IE3D\SDD_HOME\IE3D\mdspice\samples\nmos1f.mds (nmos1f.lib) file is an example. The circuit is illustrated in Figure 5.18. There is a filter at the output. The IE3D geometry of the filter is saved in C:\MentorGraphics\<latest_release>\IE3D\SDD_HOME\IE3D\mdspice\samples\bpf3a.geo and the simulation results are saved in: bpf3a.sp. The pass-band of the filter is at 3 GHz. There are multiple other pass-bands at 6 and 9 GHz, etc. (see Figure 5.19).

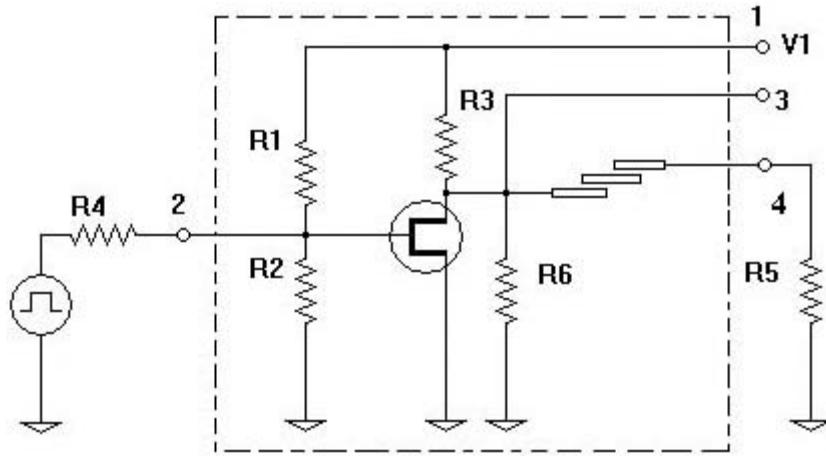


Figure 5.18 The non-linear circuit with a filter connected before the output.

Simulating such a circuit needs us to be a little bit more careful. We can go ahead to simulate it. However, the DC biasing procedure is changed completely. Due to the presence of the filter, it takes many time steps (about 9500 for the nmos1f.mds file) for the DC biasing to become steady. If we use the DC biasing removed time signals to perform the Fourier transform, we may not get very clean results. The reason is that there might be some small residuals in the time responses even after the DC biasing is detected to be stable for highly resonant structures. For this example, if we perform the Fourier transform on the DC biasing removed signals, we will see high level of noise at the DC frequency at the terminal 3. There should be little DC signal at terminal 3 because of the filter is basically open circuit for DC. In order to get high accuracy results, we may want to define the Bias Steady Criteria to a smaller number (default 0.0001) or simulate the structure with pre-defined DC biasing by removing any DC biasing source.

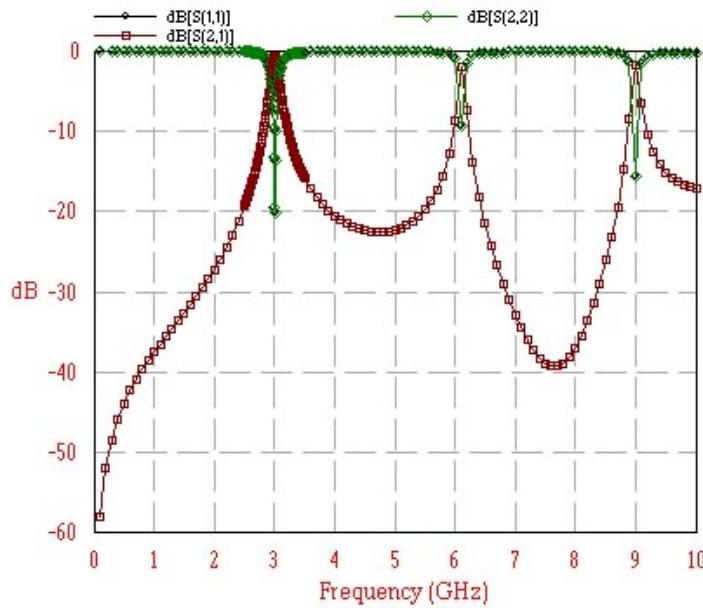


Figure 5.19 The frequency response of the filter.

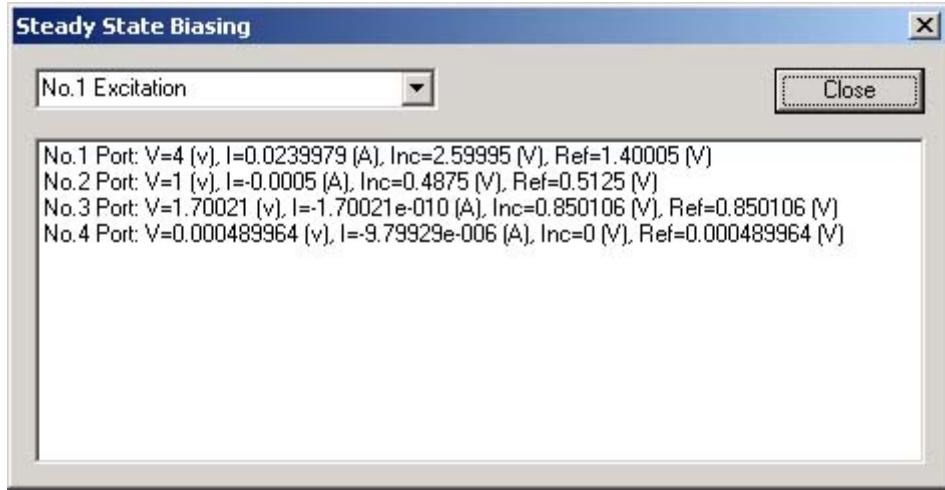


Figure 5.20 The steady state biasing dialog.

In Figure 5.18, we purposely define the terminal 3 for the main circuit. We will not put any practical load there. We just use it to find out the DC biasing voltage at the drain terminal of the transistor when the DC biasing reaches steady. When we setup the simulation, we should set the internal resistor for port 3 as $1.0e+10$ ohms, meaning it is open circuit. After the simulation of nmos1f.mds file, please select the Biassing Voltage and Current in Display menu to check the biasing voltages at the 3 terminals of the non-linear device (see Figure 5.20). The gate terminal is 1 V. The sink terminal is connected to ground and it should be 0. The drain voltage at terminal 3 is 1.70021 V. Theoretically, the biasing or steady state voltage at port 4 should be 0 because it is open circuit at DC for the band-pass filter. However, numerically, we can not guarantee it. It is indicated that the voltage at terminal 4 is 0.000489964 V, which is quite close to 0 anyway. The power spectrum of the simulation is shown in Figure 5.21. You may notice there is some noise on the floor of the spectrum. The noise is more due to the fact that we did not complete remove the biasing effect or the residual in the biasing. To remove the noise, we can define smaller Biassing Steady Criteria (default value of 0.0001). Another way is to start a simulation with the pre-defined biasing.

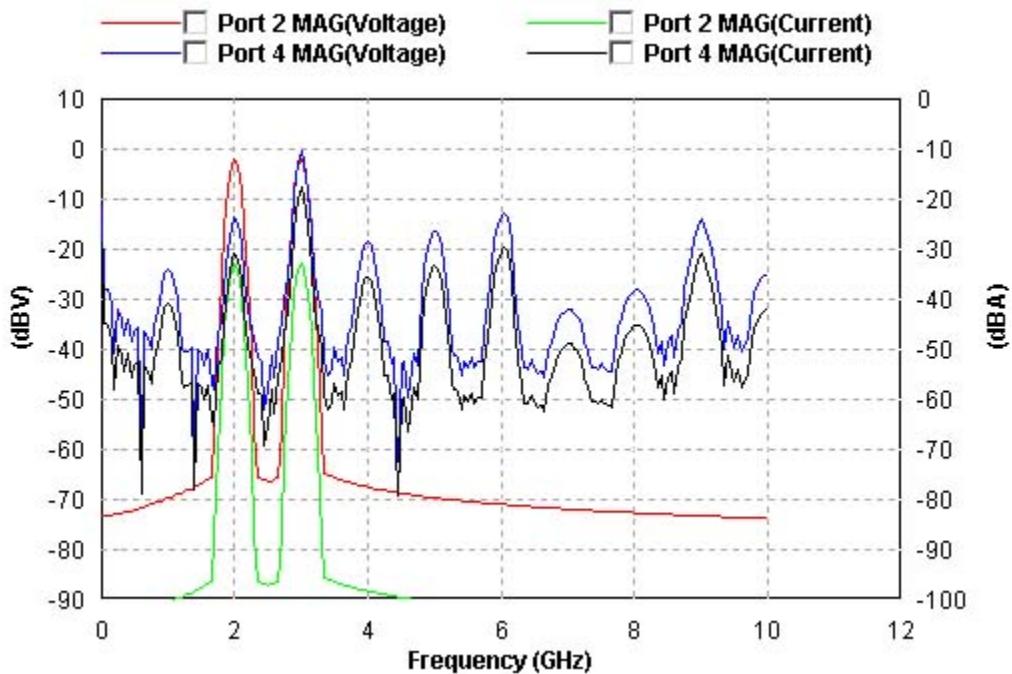


Figure 5.21 The spectrum of the amplifier with the band-pass filter with real time DC-biasing.

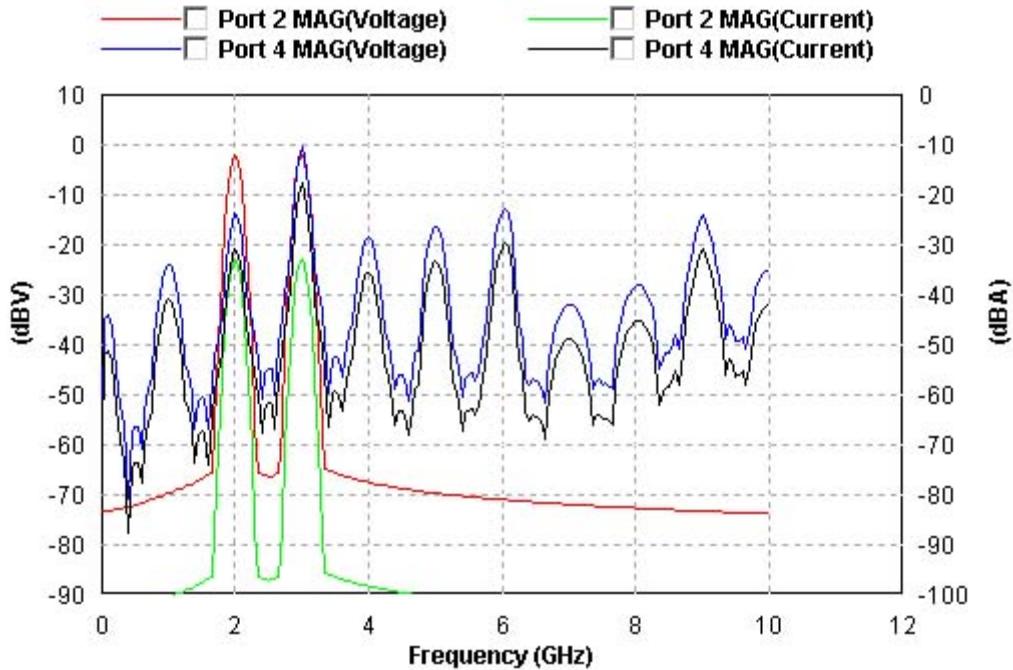


Figure 5.22 The spectrum of the amplifier with the band-pass filter with pre-defined biasing.

We have created another non-linear device: C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\nmos3.txt file with the pre-defined DC biasing from the results of C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\nmos1f.mds. We will create another MDSPICE file C:\MentorGraphics\<latest_release>IE3D\SDD_HOME\IE3D\mdspice\samples\nmos3f.mds file. This time, we will simulate it without the DC Biasing source. The Fourier transform based upon the results of nmos3f.mds using pre-defined DC biasing will be much cleaner especially at the low frequency end (see Figure 5.22).

As you can see from either Figure 5.21 or 5.22, the spectrum at other than 3, 6 and 9 GHz is much suppressed. The power spectrum is significantly affected by the filter. In the nmos1c.mds simulation, we will get the spectrum to be equal at 2 and 3 GHz. For the nmos3f.mds simulation, the spectrum at 2 GHz is about 13 dB lower than 3 GHz (exact value obtained from the Browse Graph Data in View menu). From the filter response, the S(2,1) at 2 GHz is about 26 dB lower than the S(2,1) at 3 GHz. We do not expect the difference to be identical because the internal working mechanism of the whole non-linear circuit will be quite different from the filter. Basically, the non-linear device creates the power at different spectrum. The power at 3 GHz will pass through the filter immediately. Only a small portion of the power at 2 GHz will pass through the filter. Majority of the power at 2 GHz will be reflected back to the transistor and other components of the circuits. The power will get reflected back again and more power at the 2 GHz will pass through the filter. Such a process will repeat and, eventually, we will get the power spectrum at 2 GHz higher than expected from the filter's response.

Traditionally, harmonic balance has been the only choice for microwave non-linear circuit design. Its basic principle is that we separate the linear parts and the non-linear parts. We find the s-parameters of the linear part in a wide frequency range. Then, we apply steady sinusoidal waveforms with selected frequency point(s) into the system. The sinusoidal waveforms will create harmonics in the non-linear circuit. For example, if the sinusoidal waveform contains frequencies f_1 and f_2 . Then, there will be harmonic frequencies at f_1-f_2 , f_1+f_2 , $2f_1$, $2f_2$, $2f_1-f_2$, $2f_1+f_2$, $-f_1+2f_2$, f_1+2f_2 , ... etc. The objective is to match the magnitude and phase of each frequency at the connecting point between the linear and non-linear parts. The matching process is basically a non-linear multiple variable minimization or optimization procedure. Mathematically, there is no guaranteed solution and there is no guaranteed accuracy.

Furthermore, for a transistor, there will be 3 connecting points between the linear and non-linear circuit. If we have a complicated circuit with N-transistors, we will end up with $3NM$ number of unknowns, M is the number of harmonics used in the optimization. Theoretically, there are infinite numbers of harmonics created in a circuit. A user has to decide which harmonics to be included and which one is not included in the optimization process. When multiple frequency points are involved, the number of M can be quite a large number. Eventually, we will end up with a global optimization of many unknowns ($3NM$ with large N and M values). It becomes a challenging problem to solve.

On the other hand, using a time-domain approach employed in the MDSPICE, we do not need to solve any non-linear equations. What we need to do is like the above example. We just excite the circuit with the specified waveforms with multiple frequency components. We monitor the output waveforms and perform the Fourier transform to get the output spectrum. That is the exact result we want. Clearly, the MDSPICE opens the door for a better alternative of microwave non-linear circuit design.

The circuits described in this chapter do not represent any design. They are just examples to demonstrate how the MDSPICE works for non-linear circuits. More practical examples will be included in future release.

Section 5.5 Summary on Non-Linear Circuit Simulation.

The non-linear models implemented into the MDSPICE are limited. However, it is very simple to implement more models into the MDSPICE. What we need are the behavior model of a device or the IV curves. For a CMOS transistor, we need the I_d vs. V_{ds} curves with given V_{gs} values. For a BJT transistor, we need the I_b vs. V_{be} curve, and the I_c vs. V_{ce} curves with given V_{be} values. When such a curves are given, it is just a simple task to implement it into the MDSPICE. We welcome users to provide such models so that we can implement more models into the MDSPICE to meet the needs of our customers.

Appendix A The Syntax of MDSPICE Net-List

The MDSPICE can accept traditional SPICE net-list. However, there are many different elements in different SPICE simulators. It is difficult to accept all different elements in all different SPICE simulators. Our compatibility is mainly limited to those commonly used elements such as resistors, capacitors, inductors and mutual inductors. The Table A.1 documents the syntax for the different elements acceptable in the MDSPICE. Content inside “<” and “>” are optional. The possible units are documented in Table A.2. We will keep the users informed whether an element is compatible with the traditional SPICE simulators.

Table A.1 The syntax for the elements in MDSPICE.

Element	Syntax	Explanation
Resistor	Rname n1 n2 value<unit> Example 1: The following line defines a resistor “R1” connected between nodes “1” and “1b”. Its value is 1000 ohms: R1 1 1b 1k Example 2: The following line defines the same resistor: R1 1 1b 1000	The 1 st character “R” is the key word for resistor. The “name” can be a string without blanks. The “n1” and “n2” are the two nodes of the resistor. Each can be a string without blanks. The “value” is a floating number. The unit is optional. The default is ohm. The choice of the unit is documented in Table A.2. The syntax of resistor is compatible with traditional SPICE simulators.
Capacitor	Cname n1 n2 value<unit> Example 1: The following line defines a capacitor “C2” connected between nodes “2” and “3”. Its value is 1.2 pF: C2 2 3 1.2p	The 1 st character “C” is the key word for capacitor. Other information is the same as that in resistor. The default unit is F (farad). Please note that “F” is also used as the unit for “1.0e-15” for the values of the elements. The syntax of capacitor is compatible with traditional SPICE simulator.
Inductor	Lname n1 n2 value<unit> Example 1: The following line defines an inductor “L3” connected between nodes “4” and “5”. Its value is 12 nH: L3 4 5 12n	The 1 st character “L” is the key word for inductor. Other information is the same as that in resistor. The default unit is H (henry). The syntax of inductor is compatible with traditional SPICE.
Mutual Inductor	Kname Lname1 Lname2 value Example 1: The following line defines a mutual inductor “K1” between inductors Lone and Ltwo. K1 Lone Ltwo 0.8	The 1 st character “K” is the key word for mutual inductor. A mutual inductor is defined between two inductors. The value of the mutual inductor is the coupling coefficient between the two inductors. It must be between -1 and 1. The value of K is defined as: $K = M12/\sqrt{(L1 \cdot L2)}$ The sign of K is an indication of direction. The syntax of mutual inductor is compatible with traditional SPICE.
Comment	*this is a comment line. It can be anything.	The 1 st character “*” is the key word for comment line. A comment line can happen any where in the netlist. The comment syntax is compatible with traditional spice.

Sub-Circuit	<pre>.subckt n1 n2 n3 my_sub_ckt_1ends my_sub_ckt_1</pre> <p>Example 1: The following section defines a sub-circuit. It consists of one resistor, two inductors and two capacitors. The resistor “R1” is defined between nodes “1” and “2”. The inductor “L1” is defined between two nodes “2” and “a”. The inductor “L2” is defined between two nodes “a” and “b”. The capacitor “C2” is defined between node “1” and ground. The capacitor “C3” is defined between node “a” and ground. The nodes “1”, “2”, “a” and “b” are also the ports of the sub-circuit.</p> <pre>.subckt 1 2 a b an_rlc_circuit R1 1 2 100 L1 2 a 10n C2 1 0 1p C3 a 0 2p L2 a b 5n .ends an_rlc_circuit</pre>	A SPICE file may contains one main circuit (discussed later) and multiple sub-circuits. The sub-circuits can be called by the main circuit or other sub-circuit. However, they can not form a loop. For example, sub-circuit 1 calls sub-circuit 2. Sub-circuit 2 calls sub-circuit 3. Sub-circuit 3 can not call sub-circuit 1. The “.subckt” is the key word for the beginning of the sub-circuit. The “.ends” is the key word for the end of the sub-circuit. The sub-circuit name “my_sub_ckt_1” can be any string without blanks. The “n1”, “n2” and “n3” are the names of the ports of the sub-circuit. You may have one or more ports for a sub-circuit. A sub-circuit is referenced by the main circuit or other sub-circuits using the XCall defined below. The syntax of sub-circuit is compatible with traditional SPICE.
Xcall	<p>Xname n1 n2 n3 sub_ckt_name</p> <p>Example 1: The following line defines an XCall reference to the sub-circuit defined in the example for sub-circuit. Please note that the number of ports in the XCall is 4, matching the number of ports in the sub-circuit “sub_ckt_name”. However, the name for each node can be different.</p> <pre>X2 3 4 5 6 an_rlc_circuit</pre>	An XCall is used to reference a sub-circuit from the main circuit or another sub-circuit. It is important to match the number of ports (or nodes) in the XCall to the number of ports (or nodes) in the sub-circuit referenced. However, we do not need to match the name of each port. The name of a port in the XCall belongs to the node (or port) in the sub-circuit. The name of the corresponding port in the sub-circuit belongs to the sub-circuit. The syntax of XCall is compatible with traditional SPICE.
ZCall	<p>Zname n1 n2 n3 “parameter_name”</p> <p>Example 1: The following line defines a ZCall reference to a 4-port s-parameter file created by IE3D.</p> <pre>Z3 1 3 2 4 "c:\mydir\coupled wires.sp"</pre>	The 1 st character “Z” is the key word for the ZCall. A ZCall is basically a reference to an parameter file in IE3D format documented in Appendix B. (compatible with HP/EEsof touchstone format). The parameter file name can contains blanks. We have to use “” to enclose the parameter file name. The number of nodes (or ports) should match the number of ports defined in the parameter file. The ZCall is a new element type in MDSPICE. It is not compatible with traditional SPICE.
Main Circuit	<p>The 1st section of the file. Its end is denoted by a line “.end”.</p> <p>Example 1: The following text file defines a complete circuit with a sub-circuit. The</p>	By default, the 1 st section of a .lib file is always the main circuit. A SPICE file may contains one main circuit and multiple sub-circuits. The nodes in the main circuits automatically become the

	<p>sub-circuit is called twice using different call names inside the main circuit. The nodes “1”, “2”, “3”, “4”, “a”, “b” and “c” become the ports of the main circuit automatically:</p> <pre>The 1st line is always comment line. *I want to show another comment here. R1 1 a 10.0 X1 1 2 3 4 an_rlc_circuit Z1 2 a 3 b "c:\mydir\coupled_wires.sp" X2 a b c 0 an_rlc_circuit .end .subckt 1 2 a b an_rlc_circuit R1 1 2 100 L1 2 a 10n C2 1 0 1p C3 a 0 2p L2 a b 5n .ends an_rlc_circuit</pre>	ports of the main circuit.
Non-Linear Elements	<p>Nname n1 n2 n3 “iv_curve_file”</p> <p>Example 1: The following line defines a diode connected to nodes “a” and “b”.</p> <pre>N1 a b "the_iv_curve_of_a_diode"</pre> <p>Example 2: The following line defines a transistor connected between nodes “a”, “b” and “c”:</p> <pre>N2 a b c "the_iv_curves"</pre>	The 1 st character “N” is the key word for non-linear elements. A non-linear element may have two, three or more nodes. The properties of a non-linear element is defined by the file “iv_curve_file” and the syntax of the file “iv_curve_file” is documented in Appendix C. The syntax of Non-Linear Elements in MDSPICE is not compatible with traditional SPICE.

Table A.2 The units used in the SPICE netlist.

Symbol	T	G	Meg	K	m	mil	U	N	P	F
Unit	1e+12	1e+9	1e+6	1e+3	1e-3	25.4e-6	1e-6	1e-9	1e-12	1e-15

Appendix B The Syntax of Parameter Files

The MDSPICE can accept s-parameter frequency responses. The s-parameter frequency responses are normally obtained using electromagnetic simulators such as the IE3D and FIDELITY. They normally stored in files. Table B.1 shows the syntax for the s-parameter file (z or y-parameter files).

Table B.1 The syntax for the elements in MDSPICE.

Section	Syntax	Explanation
Header	Syntax 1: #GHz S MA R 50 Syntax 2: #GHz S RI R 50 Syntax 3: #GHz Z RI R 50 Syntax 4: #GHz Y MA R 50	The “#” is the keyword for the header. “GHz” means that the frequency is in GHz. You can use “HZ”, “KHZ” or “MHZ”. “S” means the data is s-parameters. “Z” for z-parameters and “Y” for y-parameters. “MA” means that the complex data is represented as magnitude and phase in degree. The “RI” means that the data is presented as real and imaginary parts. “R 50” means that the s-parameters are normalized to 50-ohms. Other normalization may not be accepted. The “R 50” does not mean anything for z-parameters and y-parameters.
Comment	!Any line starts with exclamation	“Any line starts with “!” is considered as a comment line.
Port	! Nport = 3	The number of ports is stored in a comment line in order to be compatible with the HP/EEsof touchstone format, in which no port number is record in the file. Normally, it is the 2 nd comment line after the header. This line may be omitted. However, if it is in it, you need to make sure it matches the frequency data block discussed next.
Frequency Data	For 1-Port Freq Mag(S11) Ang(S11) For 2-Ports Freq Mag(S11) Ang(S11) Mag(S21) Ang(S21) Mag(S12) Ang(S12) Mag(S22) Ang(S22) For 3- or More Ports Freq Mag(S11) Ang(S11) Mag(S12) Ang(S12) Mag(S13) Ang(S13) Mag(S21) Ang(S21) Mag(S22) Ang(S22) Mag(S23) Ang(S23) Mag(S31) Ang(S31) Mag(S32) Ang(S32) Mag(S33) Ang(S33)	Each set of frequency data contains a matrix of s-parameters. The 1 st value is always the frequency. Then, it goes through the matrix element list with magnitude and phase (or real part and imaginary part depending upon the definition in the header). Be careful of the order of the matrix elements. The 2-port case goes column by column. The more port cases go row by row. Certainly, the A set of frequency data can occupy multiple lines in the ASCII file. You can list as many sets of frequency data as you like.

Appendix C The Syntax of IV Curve Files

Non-linear elements are modeled as IV-curves in the MDSPICE. At the time of this writing, we only implemented 3 types of non-linear devices into the MDSPICE. They are discussed in the Section 5.1 of Chapter 5. We welcome users to provide the Iv curves for different devices. We will try to document more examples in the near future.

Appendix D Different Excitation Sources

There are 3 types of excitation sources available in MDSPICE: Voltage Source, Current Source and Wave Source. The formulations for the 3 excitation sources are documented in equations (D-1), (D-2) and (D-3).

For the Voltage Source,

$$V = Z_{in} / (Z_{in} + Z_0) V_0 \quad (D-1)$$

For the Current Source,

$$I = Y_{in} / (Y_{in} + Y_0) I_0 \quad (D-2)$$

For the Wave Source,

$$a = a_0 + \Gamma b \quad (D-3)$$

Where V is the voltage applied to the port; $Z_{in} = V/I$ is the input impedance at the port; Z_0 is the internal impedance of the source; V_0 is the source voltage of the Voltage Source; $Y_{in} = I/V$ is the input admittance at the port; $Y_0 = 1/Z_0$ is the internal admittance of the source; I_0 is the source current of the Current Source; a is the incident wave; b is the reflected wave; a_0 is the source incident wave of the Wave Source; Γ is the reflection coefficient of the internal impedance. For time domain simulation, we have to limit the Z_0 to be a non-negative value.

In fact, the three types of sources are convertible under some conditions. In the MDSPICE, we always assume the Wave Source is using 50-ohm normalization. We also assume define the incident wave “a” and the reflected waves “b” as:

$$V = a + b \quad (D-4)$$

$$I = (a - b) / 50 \quad (D-5)$$

For such a definition, the unit of “a” and “b” is always “V”.

We can prove after some derivation that,

$$\Gamma = (Z_0 - 50) / (Z_0 + 50) \quad (D-4)$$

$$Y_0 = 1 / Z_0 \quad (D-5)$$

$$V_0 = I_0 \cdot Z_0 \quad (D-6)$$

$$a_0 = V_0 \cdot 50 / (Z_0 + 50) \quad (D-7)$$

If we want to apply a constant voltage to a port, we should choose the VoltageSource with Internal Resistor = 0. If we want to apply a constant current to a port, we should choose the CurrentSource with Internal Resistor = 1.0e+15 (infinity). If we want to apply a constant incident wave to a port, we should choose WaveSource with Internal Resistor = 50 ohms ($\Gamma = 0$).

End-User License Agreement

The latest version of the End-User License Agreement is available on-line at:
www.mentor.com/eula

IMPORTANT INFORMATION

USE OF ALL SOFTWARE IS SUBJECT TO LICENSE RESTRICTIONS. CAREFULLY READ THIS LICENSE AGREEMENT BEFORE USING THE PRODUCTS. USE OF SOFTWARE INDICATES CUSTOMER'S COMPLETE AND UNCONDITIONAL ACCEPTANCE OF THE TERMS AND CONDITIONS SET FORTH IN THIS AGREEMENT. ANY ADDITIONAL OR DIFFERENT PURCHASE ORDER TERMS AND CONDITIONS SHALL NOT APPLY.

END-USER LICENSE AGREEMENT ("Agreement")

This is a legal agreement concerning the use of Software (as defined in Section GRANT OF LICENSE) and hardware (collectively "Products") between the company acquiring the Products ("Customer"), and the Mentor Graphics entity that issued the corresponding quotation or, if no quotation was issued, the applicable local Mentor Graphics entity ("Mentor Graphics"). Except for license agreements related to the subject matter of this license agreement which are physically signed by Customer and an authorized representative of Mentor Graphics, this Agreement and the applicable quotation contain the parties' entire understanding relating to the subject matter and supersede all prior or contemporaneous agreements. If Customer does not agree to these terms and conditions, promptly return or, in the case of Software received electronically, certify destruction of Software and all accompanying items within five days after receipt of Software and receive a full refund of any license fee paid.

1. ORDERS, FEES AND PAYMENT.

- 1.1. To the extent Customer (or if agreed by Mentor Graphics, Customer's appointed third party buying agent) places and Mentor Graphics accepts purchase orders pursuant to this Agreement ("Order(s)'), each Order will constitute a contract between Customer and Mentor Graphics, which shall be governed solely and exclusively by the terms and conditions of this Agreement, any applicable addenda and the applicable quotation, whether or not these documents are referenced on the Order. Any additional or conflicting terms and conditions appearing on an Order will not be effective unless agreed in writing by an authorized representative of Customer and Mentor Graphics.
- 1.2. Amounts invoiced will be paid, in the currency specified on the applicable invoice, within 30 days from the date of such invoice. Any past due invoices will be subject to the imposition of interest charges in the amount of one and one-half percent per month or the applicable legal rate currently in effect, whichever is lower. Prices do not include freight, insurance, customs duties, taxes or other similar charges, which Mentor Graphics will state separately in the applicable invoice(s). Unless timely provided with a valid certificate of exemption or other evidence that items are not taxable, Mentor Graphics will invoice Customer for all applicable taxes including, but not limited to, VAT, GST, sales tax and service tax. Customer will make all payments free and clear of, and without reduction for, any withholding or other taxes; any such taxes imposed on payments by Customer hereunder will be Customer's sole responsibility. If Customer appoints a third party to place purchase orders and/or make payments on Customer's behalf, Customer shall be liable for payment under Orders placed by such third party in the event of default.
- 1.3. All Products are delivered FCA factory (Incoterms 2000), freight prepaid and invoiced to Customer, except Software delivered electronically, which shall be deemed delivered when made available to Customer for download. Mentor Graphics retains a security interest in all Products delivered under this Agreement, to secure payment of the purchase price of such Products, and Customer agrees to sign any documents that Mentor Graphics determines to be necessary or convenient for use in filing or perfecting such security interest. Mentor Graphics' delivery of Software by electronic means is subject to Customer's provision of both a primary and an alternate e-mail address.

2. **GRANT OF LICENSE.** The software installed, downloaded, or otherwise acquired by Customer under this Agreement, including any updates, modifications, revisions, copies, documentation and design data (“Software”) are copyrighted, trade secret and confidential information of Mentor Graphics or its licensors, who maintain exclusive title to all Software and retain all rights not expressly granted by this Agreement. Mentor Graphics grants to Customer, subject to payment of applicable license fees, a nontransferable, nonexclusive license to use Software solely: (a) in machine-readable, object-code form (except as provided in Subsection If any Software or portions thereof are provided in source code form, Customer will use the source code only to correct so); (b) for Customer’s internal business purposes; (c) for the term of the license; and (d) on the computer hardware and at the site authorized by Mentor Graphics. A site is restricted to a one-half mile (800 meter) radius. Customer may have Software temporarily used by an employee for telecommuting purposes from locations other than a Customer office, such as the employee’s residence, an airport or hotel, provided that such employee’s primary place of employment is the site where the Software is authorized for use. Mentor Graphics’ standard policies and programs, which vary depending on Software, license fees paid or services purchased, apply to the following: (a) relocation of Software; (b) use of Software, which may be limited, for example, to execution of a single session by a single user on the authorized hardware or for a restricted period of time (such limitations may be technically implemented through the use of authorization codes or similar devices); and (c) support services provided, including eligibility to receive telephone support, updates, modifications, and revisions. For the avoidance of doubt, if Customer requests any change or enhancement to Software, whether in the course of receiving support or consulting services, evaluating Software, performing beta testing or otherwise, any inventions, product improvements, modifications or developments made by Mentor Graphics (at Mentor Graphics’ sole discretion) will be the exclusive property of Mentor Graphics.
 3. **ESC SOFTWARE.** If Customer purchases a license to use development or prototyping tools of Mentor Graphics’ Embedded Software Channel (“ESC”), Mentor Graphics grants to Customer a nontransferable, nonexclusive license to reproduce and distribute executable files created using ESC compilers, including the ESC run-time libraries distributed with ESC C and C++ compiler Software that are linked into a composite program as an integral part of Customer’s compiled computer program, provided that Customer distributes these files only in conjunction with Customer’s compiled computer program. Mentor Graphics does NOT grant Customer any right to duplicate, incorporate or embed copies of Mentor Graphics’ real-time operating systems or other embedded software products into Customer’s products or applications without first signing or otherwise agreeing to a separate agreement with Mentor Graphics for such purpose.
4. **BETA CODE.**
 - 4.1. Portions or all of certain Software may contain code for experimental testing and evaluation (“Beta Code”), which may not be used without Mentor Graphics’ explicit authorization. Upon Mentor Graphics’ authorization, Mentor Graphics grants to Customer a temporary, nontransferable, nonexclusive license for experimental use to test and evaluate the Beta Code without charge for a limited period of time specified by Mentor Graphics. This grant and Customer’s use of the Beta Code shall not be construed as marketing or offering to sell a license to the Beta Code, which Mentor Graphics may choose not to release commercially in any form.
 - 4.2. If Mentor Graphics authorizes Customer to use the Beta Code, Customer agrees to evaluate and test the Beta Code under normal conditions as directed by Mentor Graphics. Customer will contact Mentor Graphics periodically during Customer’s use of the Beta Code to discuss any malfunctions or suggested improvements. Upon completion of Customer’s evaluation and testing, Customer will send to Mentor Graphics a written evaluation of the Beta Code, including its strengths, weaknesses and recommended improvements.
 - 4.3. Customer agrees to maintain Beta Code in confidence and shall restrict access to the Beta Code, including the methods and concepts utilized therein, solely to those employees and Customer location(s) authorized by Mentor Graphics to perform beta testing. Customer agrees that any written evaluations and all inventions, product improvements, modifications or developments that Mentor Graphics conceived or made during or subsequent to this Agreement, including those based partly or wholly on Customer’s feedback, will be the exclusive property of Mentor Graphics. Mentor Graphics will have exclusive rights, title and interest in all such property. The provisions of this Subsection Customer agrees to maintain Beta Code in confidence and shall restrict access to the Beta Code, including the methods and co shall survive termination of this Agreement.
 5. **RESTRICTIONS ON USE.**

- 5.1. Customer may copy Software only as reasonably necessary to support the authorized use. Each copy must include all notices and legends embedded in Software and affixed to its medium and container as received from Mentor Graphics. All copies shall remain the property of Mentor Graphics or its licensors. Customer shall maintain a record of the number and primary location of all copies of Software, including copies merged with other software, and shall make those records available to Mentor Graphics upon request. Customer shall not make Products available in any form to any person other than Customer's employees and on-site contractors, excluding Mentor Graphics competitors, whose job performance requires access and who are under obligations of confidentiality. Customer shall take appropriate action to protect the confidentiality of Products and ensure that any person permitted access does not disclose or use it except as permitted by this Agreement. Customer shall give Mentor Graphics written notice of any unauthorized disclosure or use of the Products as soon as Customer learns or becomes aware of such unauthorized disclosure or use. Except as otherwise permitted for purposes of interoperability as specified by applicable and mandatory local law, Customer shall not reverse-assemble, reverse-compile, reverse-engineer or in any way derive any source code from Software. Log files, data files, rule files and script files generated by or for the Software (collectively "Files"), including without limitation files containing Standard Verification Rule Format ("SVRF") and Tcl Verification Format ("TVF") which are Mentor Graphics' proprietary syntaxes for expressing process rules, constitute or include confidential information of Mentor Graphics. Customer may share Files with third parties, excluding Mentor Graphics competitors, provided that the confidentiality of such Files is protected by written agreement at least as well as Customer protects other information of a similar nature or importance, but in any case with at least reasonable care. Customer may use Files containing SVRF or TVF only with Mentor Graphics products. Under no circumstances shall Customer use Software or Files or allow their use for the purpose of developing, enhancing or marketing any product that is in any way competitive with Software, or disclose to any third party the results of, or information pertaining to, any benchmark.
 - 5.2. If any Software or portions thereof are provided in source code form, Customer will use the source code only to correct software errors and enhance or modify the Software for the authorized use. Customer shall not disclose or permit disclosure of source code, in whole or in part, including any of its methods or concepts, to anyone except Customer's employees or contractors, excluding Mentor Graphics competitors, with a need to know. Customer shall not copy or compile source code in any manner except to support this authorized use.
 - 5.3. Customer may not assign this Agreement or the rights and duties under it, or relocate, sublicense or otherwise transfer the Products, whether by operation of law or otherwise ("Attempted Transfer"), without Mentor Graphics' prior written consent and payment of Mentor Graphics' then-current applicable relocation and/or transfer fees. Any Attempted Transfer without Mentor Graphics' prior written consent shall be a material breach of this Agreement and may, at Mentor Graphics' option, result in the immediate termination of the Agreement and/or the licenses granted under this Agreement. The terms of this Agreement, including without limitation the licensing and assignment provisions, shall be binding upon Customer's permitted successors in interest and assigns.
 - 5.4. The provisions of this Section **RESTRICTIONS ON USE** shall survive the termination of this Agreement.
6. **SUPPORT SERVICES.** To the extent Customer purchases support services, Mentor Graphics will provide Customer updates and technical support for the Products, at the Customer site(s) for which support is purchased, in accordance with Mentor Graphics' then current End-User Support Terms located at <http://supportnet.mentor.com/about/legal/>.
 7. **AUTOMATIC CHECK FOR UPDATES; PRIVACY.** Technological measures in Software may communicate with servers of Mentor Graphics or its contractors for the purpose of checking for and notifying the user of updates and to ensure that the Software in use is licensed in compliance with this Agreement. Mentor Graphics will not collect any personally identifiable data in this process and will not disclose any data collected to any third party without the prior written consent of Customer, except to Mentor Graphics' outside attorneys or as may be required by a court of competent jurisdiction.
8. **LIMITED WARRANTY.**
 - 8.1. Mentor Graphics warrants that during the warranty period its standard, generally supported Products, when properly installed, will substantially conform to the functional specifications set forth in the

applicable user manual. Mentor Graphics does not warrant that Products will meet Customer's requirements or that operation of Products will be uninterrupted or error free. The warranty period is 90 days starting on the 15th day after delivery or upon installation, whichever first occurs. Customer must notify Mentor Graphics in writing of any nonconformity within the warranty period. For the avoidance of doubt, this warranty applies only to the initial shipment of Software under an Order and does not renew or reset, for example, with the delivery of (a) Software updates or (b) authorization codes or alternate Software under a transaction involving Software re-mix. This warranty shall not be valid if Products have been subject to misuse, unauthorized modification or improper installation. MENTOR GRAPHICS' ENTIRE LIABILITY AND CUSTOMER'S EXCLUSIVE REMEDY SHALL BE, AT MENTOR GRAPHICS' OPTION, EITHER (A) REFUND OF THE PRICE PAID UPON RETURN OF THE PRODUCTS TO MENTOR GRAPHICS OR (B) MODIFICATION OR REPLACEMENT OF THE PRODUCTS THAT DO NOT MEET THIS LIMITED WARRANTY, PROVIDED CUSTOMER HAS OTHERWISE COMPLIED WITH THIS AGREEMENT. MENTOR GRAPHICS MAKES NO WARRANTIES WITH RESPECT TO: (A) SERVICES; (B) PRODUCTS PROVIDED AT NO CHARGE; OR (C) BETA CODE; ALL OF WHICH ARE PROVIDED "AS IS."

- 8.2. THE WARRANTIES SET FORTH IN THIS SECTION **LIMITED WARRANTY** ARE EXCLUSIVE. NEITHER MENTOR GRAPHICS NOR ITS LICENSORS MAKE ANY OTHER WARRANTIES EXPRESS, IMPLIED OR STATUTORY, WITH RESPECT TO PRODUCTS PROVIDED UNDER THIS AGREEMENT. MENTOR GRAPHICS AND ITS LICENSORS SPECIFICALLY DISCLAIM ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF INTELLECTUAL PROPERTY.
9. **LIMITATION OF LIABILITY.** EXCEPT WHERE THIS EXCLUSION OR RESTRICTION OF LIABILITY WOULD BE VOID OR INEFFECTIVE UNDER APPLICABLE LAW, IN NO EVENT SHALL MENTOR GRAPHICS OR ITS LICENSORS BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES (INCLUDING LOST PROFITS OR SAVINGS) WHETHER BASED ON CONTRACT, TORT OR ANY OTHER LEGAL THEORY, EVEN IF MENTOR GRAPHICS OR ITS LICENSORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL MENTOR GRAPHICS' OR ITS LICENSORS' LIABILITY UNDER THIS AGREEMENT EXCEED THE AMOUNT RECEIVED FROM CUSTOMER FOR THE HARDWARE, SOFTWARE LICENSE OR SERVICE GIVING RISE TO THE CLAIM. IN THE CASE WHERE NO AMOUNT WAS PAID, MENTOR GRAPHICS AND ITS LICENSORS SHALL HAVE NO LIABILITY FOR ANY DAMAGES WHATSOEVER. THE PROVISIONS OF THIS SECTION **LIMITATION OF LIABILITY** SHALL SURVIVE THE TERMINATION OF THIS AGREEMENT.
10. **HAZARDOUS APPLICATIONS.** CUSTOMER ACKNOWLEDGES IT IS SOLELY RESPONSIBLE FOR TESTING ITS PRODUCTS USED IN APPLICATIONS WHERE THE FAILURE OR INACCURACY OF ITS PRODUCTS MIGHT RESULT IN DEATH OR PERSONAL INJURY ("HAZARDOUS APPLICATIONS"). NEITHER MENTOR GRAPHICS NOR ITS LICENSORS SHALL BE LIABLE FOR ANY DAMAGES RESULTING FROM OR IN CONNECTION WITH THE USE OF MENTOR GRAPHICS PRODUCTS IN OR FOR HAZARDOUS APPLICATIONS. THE PROVISIONS OF THIS SECTION **HAZARDOUS APPLICATIONS** SHALL SURVIVE THE TERMINATION OF THIS AGREEMENT.
11. **INDEMNIFICATION.** CUSTOMER AGREES TO INDEMNIFY AND HOLD HARMLESS MENTOR GRAPHICS AND ITS LICENSORS FROM ANY CLAIMS, LOSS, COST, DAMAGE, EXPENSE OR LIABILITY, INCLUDING ATTORNEYS' FEES, ARISING OUT OF OR IN CONNECTION WITH THE USE OF PRODUCTS AS DESCRIBED IN SECTION **HAZARDOUS APPLICATIONS**. THE PROVISIONS OF THIS SECTION **INDEMNIFICATION** SHALL SURVIVE THE TERMINATION OF THIS AGREEMENT.
12. **INFRINGEMENT.**
 - 12.1. Mentor Graphics will defend or settle, at its option and expense, any action brought against Customer in the United States, Canada, Japan, or member state of the European Union which alleges that any standard, generally supported Product acquired by Customer hereunder infringes a patent or copyright or misappropriates a trade secret in such jurisdiction. Mentor Graphics will pay costs and damages finally awarded against Customer that are attributable to the action. Customer understands and agrees that as conditions to Mentor Graphics' obligations under this section Customer must: (a) notify Mentor Graphics promptly in writing of the action; (b) provide Mentor Graphics all reasonable information and

- assistance to settle or defend the action; and (c) grant Mentor Graphics sole authority and control of the defense or settlement of the action.
- 12.2. If a claim is made under Subsection Mentor Graphics will defend or settle, at its option and expense, any action brought against Customer in the United States, Ca Mentor Graphics may, at its option and expense, (a) replace or modify the Product so that it becomes noninfringing; (b) procure for Customer the right to continue using the Product; or (c) require the return of the Product and refund to Customer any purchase price or license fee paid, less a reasonable allowance for use.
- 12.3. Mentor Graphics has no liability to Customer if the action is based upon: (a) the combination of Software or hardware with any product not furnished by Mentor Graphics; (b) the modification of the Product other than by Mentor Graphics; (c) the use of other than a current unaltered release of Software; (d) the use of the Product as part of an infringing process; (e) a product that Customer makes, uses, or sells; (f) any Beta Code or Product provided at no charge; (g) any software provided by Mentor Graphics' licensors who do not provide such indemnification to Mentor Graphics' customers; or (h) infringement by Customer that is deemed willful. In the case of (h), Customer shall reimburse Mentor Graphics for its reasonable attorney fees and other costs related to the action.
- 12.4. **THIS SECTION INFRINGEMENT IS SUBJECT TO SECTION LIMITATION OF LIABILITY ABOVE AND STATES THE ENTIRE LIABILITY OF MENTOR GRAPHICS AND ITS LICENSORS FOR DEFENSE, SETTLEMENT AND DAMAGES, AND CUSTOMER'S SOLE AND EXCLUSIVE REMEDY, WITH RESPECT TO ANY ALLEGED PATENT OR COPYRIGHT INFRINGEMENT OR TRADE SECRET MISAPPROPRIATION BY ANY PRODUCT PROVIDED UNDER THIS AGREEMENT.**
13. **TERMINATION AND EFFECT OF TERMINATION.** If a Software license was provided for limited term use, such license will automatically terminate at the end of the authorized term.
- 13.1. Mentor Graphics may terminate this Agreement and/or any license granted under this Agreement immediately upon written notice if Customer: (a) exceeds the scope of the license or otherwise fails to comply with the licensing or confidentiality provisions of this Agreement, or (b) becomes insolvent, files a bankruptcy petition, institutes proceedings for liquidation or winding up or enters into an agreement to assign its assets for the benefit of creditors. For any other material breach of any provision of this Agreement, Mentor Graphics may terminate this Agreement and/or any license granted under this Agreement upon 30 days written notice if Customer fails to cure the breach within the 30 day notice period. Termination of this Agreement or any license granted hereunder will not affect Customer's obligation to pay for Products shipped or licenses granted prior to the termination, which amounts shall be payable immediately upon the date of termination.
- 13.2. Upon termination of this Agreement, the rights and obligations of the parties shall cease except as expressly set forth in this Agreement. Upon termination, Customer shall ensure that all use of the affected Products ceases, and shall return hardware and either return to Mentor Graphics or destroy Software in Customer's possession, including all copies and documentation, and certify in writing to Mentor Graphics within ten business days of the termination date that Customer no longer possesses any of the affected Products or copies of Software in any form.
14. **EXPORT.** The Products provided hereunder are subject to regulation by local laws and United States government agencies, which prohibit export or diversion of certain products and information about the products to certain countries and certain persons. Customer agrees that it will not export Products in any manner without first obtaining all necessary approval from appropriate local and United States government agencies.
15. **U.S. GOVERNMENT LICENSE RIGHTS.** Software was developed entirely at private expense. All Software is commercial computer software within the meaning of the applicable acquisition regulations. Accordingly, pursuant to US FAR 48 CFR 12.212 and DFAR 48 CFR 227.7202, use, duplication and disclosure of the Software by or for the U.S. Government or a U.S. Government subcontractor is subject solely to the terms and conditions set forth in this Agreement, except for provisions which are contrary to applicable mandatory federal laws.

16. **THIRD PARTY BENEFICIARY.** Mentor Graphics Corporation, Mentor Graphics (Ireland) Limited, Microsoft Corporation and other licensors may be third party beneficiaries of this Agreement with the right to enforce the obligations set forth herein.
17. **REVIEW OF LICENSE USAGE.** Customer will monitor the access to and use of Software. With prior written notice and during Customer's normal business hours, Mentor Graphics may engage an internationally recognized accounting firm to review Customer's software monitoring system and records deemed relevant by the internationally recognized accounting firm to confirm Customer's compliance with the terms of this Agreement or U.S. or other local export laws. Such review may include FLEXlm or FLEXnet (or successor product) report log files that Customer shall capture and provide at Mentor Graphics' request. Customer shall make records available in electronic format and shall fully cooperate with data gathering to support the license review. Mentor Graphics shall bear the expense of any such review unless a material non-compliance is revealed. Mentor Graphics shall treat as confidential information all information gained as a result of any request or review and shall only use or disclose such information as required by law or to enforce its rights under this Agreement. The provisions of this Section **REVIEW OF LICENSE USAGE** shall survive the termination of this Agreement.
18. **CONTROLLING LAW, JURISDICTION AND DISPUTE RESOLUTION.** The owners of certain Mentor Graphics intellectual property licensed under this Agreement are located in Ireland and the United States. To promote consistency around the world, disputes shall be resolved as follows: excluding conflict of laws rules, this Agreement shall be governed by and construed under the laws of the State of Oregon, USA, if Customer is located in North or South America, and the laws of Ireland if Customer is located outside of North or South America. All disputes arising out of or in relation to this Agreement shall be submitted to the exclusive jurisdiction of the courts of Portland, Oregon when the laws of Oregon apply, or Dublin, Ireland when the laws of Ireland apply. Notwithstanding the foregoing, all disputes in Asia arising out of or in relation to this Agreement shall be resolved by arbitration in Singapore before a single arbitrator to be appointed by the chairman of the Singapore International Arbitration Centre ("SIAC") to be conducted in the English language, in accordance with the Arbitration Rules of the SIAC in effect at the time of the dispute, which rules are deemed to be incorporated by reference in this section. This section shall not restrict Mentor Graphics' right to bring an action against Customer in the jurisdiction where Customer's place of business is located. The United Nations Convention on Contracts for the International Sale of Goods does not apply to this Agreement.
19. **SEVERABILITY.** If any provision of this Agreement is held by a court of competent jurisdiction to be void, invalid, unenforceable or illegal, such provision shall be severed from this Agreement and the remaining provisions will remain in full force and effect.
20. **MISCELLANEOUS.** This Agreement contains the parties' entire understanding relating to its subject matter and supersedes all prior or contemporaneous agreements, including but not limited to any purchase order terms and conditions. Some Software may contain code distributed under a third party license agreement that may provide additional rights to Customer. Please see the applicable Software documentation for details. This Agreement may only be modified in writing by authorized representatives of the parties. Waiver of terms or excuse of breach must be in writing and shall not constitute subsequent consent, waiver or excuse.

Rev. 100615, Part No. 246066