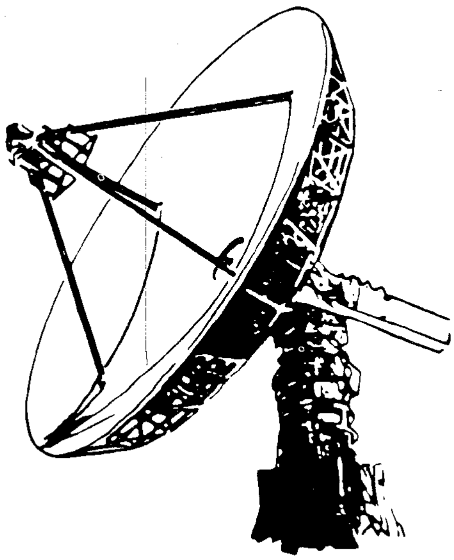
GENERALIZED DE-EMBEDDING TECHNIQUES

Roger D Pollard

Department of Electrical and Electronic Engineering
The University of Leeds
Leeds LS2 9JT, U.K.

Network Measurements Division
Hewlett-Packard Company
1400 Fountaingrove Parkway
Santa Rosa CA 95401



RF & Microwave Measurement Symposium and Exhibition

HEWLETT PACKARD

GENERALIZED DE-EMBEDDING TECHNIQUES

ABSTRACT

This paper discusses techniques for embedding and de-embedding the characteristics of two-port devices described by scattering parameters at microwave frequencies. Such techniques have wide applications in areas such as the measurement of the parameters of a semiconductor device mounted in a fixture, removal of the effects of cables, adapters, etc. and network analyzer calibration using variant error models.

The uses of the techniques are demonstrated with examples including the characterization of the HP85041A transistor test fixture.

Author:

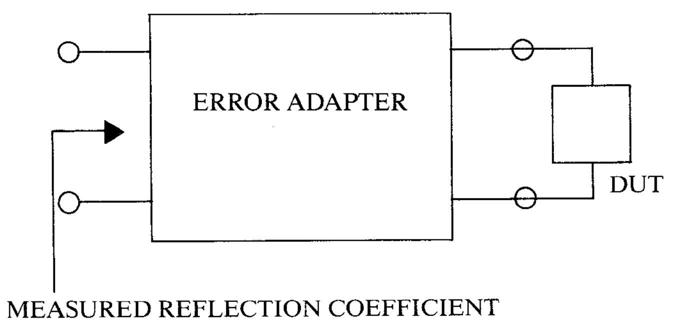
Dr Roger Pollard is a Senior Lecturer in the Department of Electrical and Electronic Engineering at the University of Leeds, England. He has been involved with research and teaching in microwave engineering for 15 years specializing in microwave measurements and active device characterization. He is a consultant for HP Network Measurements Division, Santa Rosa, CA. and has contributed to the HP8510 network analyzer (in particular the calibration and verification techniques), the HP85041A transistor test fixture and the 8510 millimeter wave measurement systems.

The process of measurement is that of de-embedding a device under test out of an environment. Similarly the design of a circuit or system can be considered to be embedding a device in a network to provide a desired set of terminal properties.

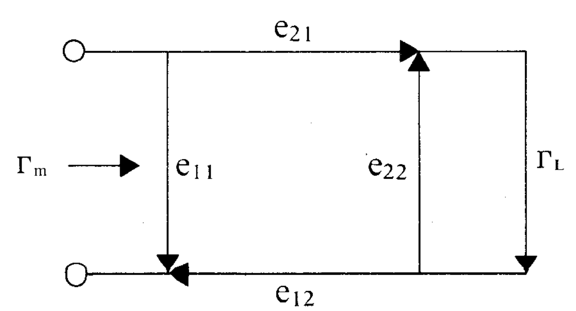
Generalized De-embedding Techniques

The simplest example in common use is one-port error correction in a microwave network analyzer. Here it is assumed that a two-terminal device under test (DUT) is connected to an ideal vector measuring instrument through a two-port network. The two-port network (known as the "error adapter") is assumed linear and can, in general, be characterized in terms of four scattering parameters.

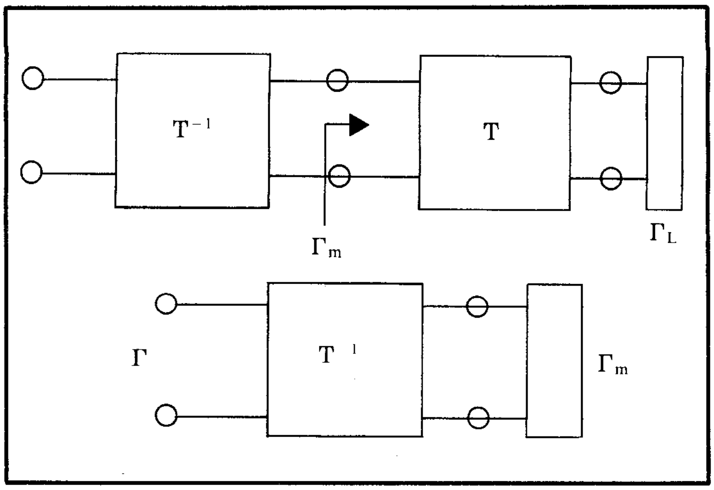
When making ratio measurements, only three independent terms are required to characterize the error adapter.



The error adapter and device under test can be represented as a flow-graph. This flowgraph is readily solved by conventional techniques to yield Γ_m in terms of Γ_L or vice versa. It is, however, interesting to re-cast the error adapter in the form of a T-matrix (chain-scattering matrix).



$$E = \begin{pmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{pmatrix}$$



We can then obtain Γ_m in terms of Γ_L. However if the network is now cascaded with the inverse T-matrix, we obtain automatically Γ_L in terms of Γ_m.

$$T = \begin{bmatrix} E_{RF} - E_{DF}E_{SF} & E_{DF} \\ -E_{SF} & 1 \end{bmatrix}$$

Using the conventional notation for the 3 terms of the one-port error adapter, its T-matrix is shown on the slide.

$$T^{-1} = \frac{1}{E_{RF}} \begin{bmatrix} 1 & -E_{DF} \\ E_{SF} & E_{RF} - E_{DF}E_{SF} \end{bmatrix}$$

This T-matrix is readily inverted to yield the result shown on the slide.

When the inverse matrix is cascaded with Γ_m the original measured reflection coefficient, we get the well-known result of the actual reflection coefficient of the DUT in terms of Γ_L and the error terms.

$$\Gamma_L = \frac{\Gamma_m - E_{DF}}{E_{SF}(\Gamma_m - E_{DF}) + E_{RF}}$$

The remaining problem is one of finding the values of E_{DF}, E_{SF}, E_{RF} by supplying "known" DUTs (calibration standards). Note that the calibration standards need to be known far more precisely than might be supposed (their values are measured through the error adapter and then effectively used a second time when the DUT is subsequently de-embedded). In the example case, any three devices may be used as calibration standards and the values of E_{DF}, E_{SF}, E_{RF} obtained from the solution of three linear simultaneous equations.

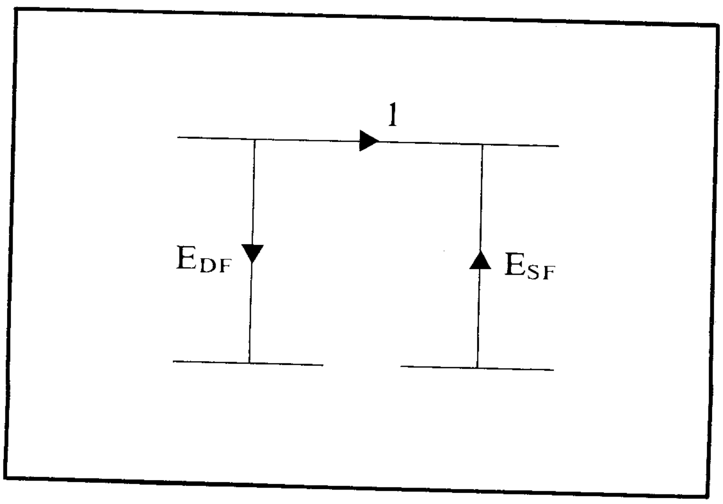
$$\Gamma_{si}(a - \Gamma_{mi}c) + b = \Gamma_{mi} \qquad i = 1, 2, 3$$

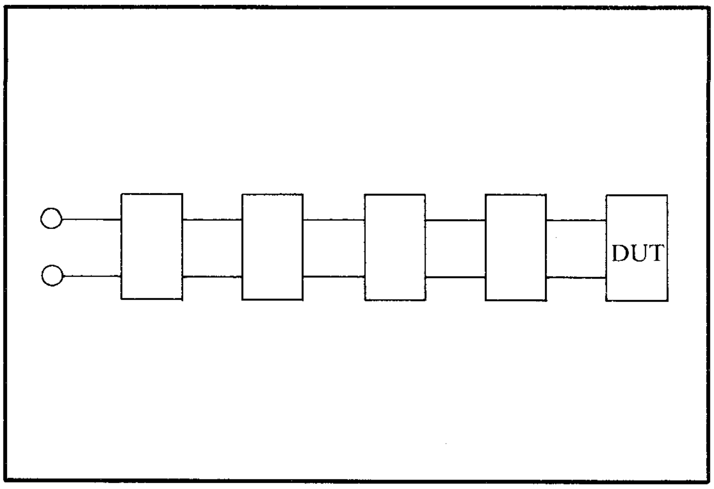
$$a = E_{RF} - E_{DF}E_{SF}$$

$$b = E_{DF}$$

$$c = -E_{SF}$$

It is critically important to realize that the accuracy of this technique is highly sensitive to the values of the parameters. Consider, for example, the limiting case where $E_{RF} = 0$ (there is no connection) and it becomes impossible to measure the DUT at all, even though the analysis will still yield a result.

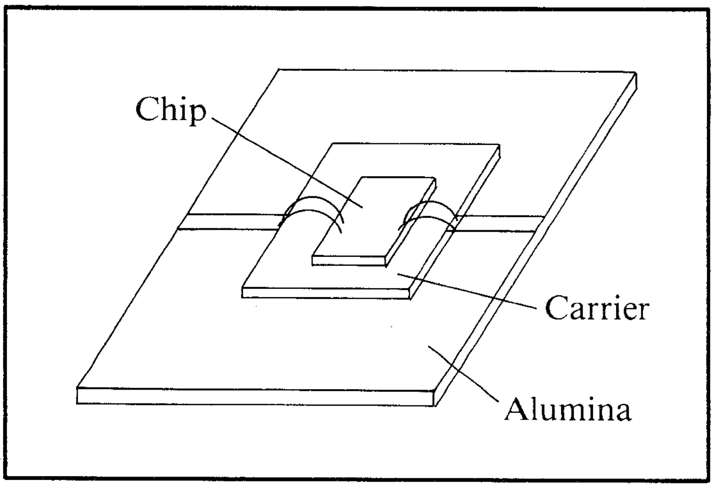




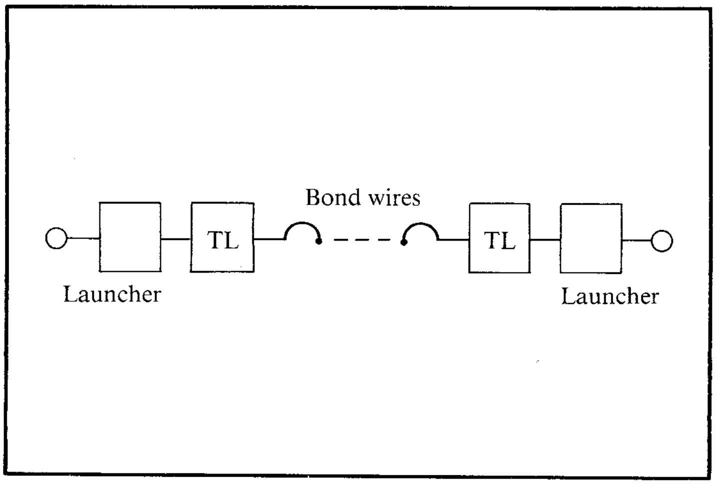
The argument outlined above can be extended to a whole range of problems involving cables, adapters, fixtures, etc. and a whole chain of networks can be cascaded.

It is necessary only to calculate the T-matrix for each element and to multiply the matrices to obtain the matrix for the total network.

Now we consider the problem of de-embedding in somewhat more general terms and look at some relevant examples.

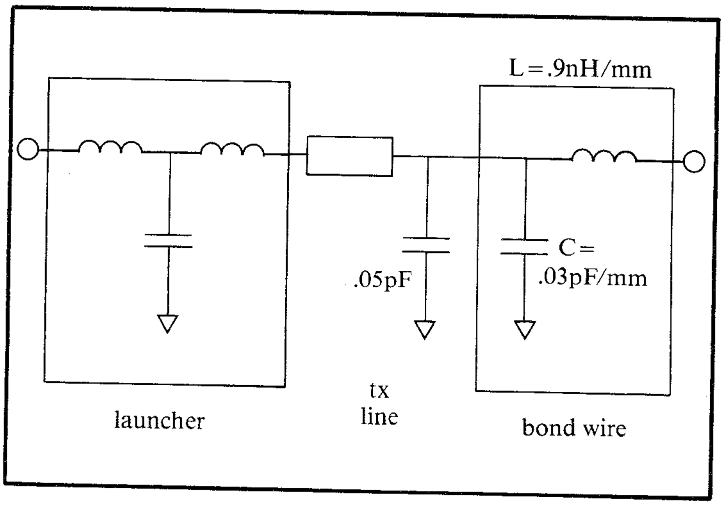


As a first example, consider the case shown here, where it is desired to measure the properties of a semiconductor chip mounted on a carrier on a microstrip substrate. Although, conceptually, it is possible to devise a means of providing calibration pieces on-chip, the difficulties of doing so and the inherent non-repeatability of bond-wires, launchers, etc. makes it very undesirable. In most instances where bondwires or non-precision connectors are involved these items cannot be part of the calibration scheme.



The most satisfactory approach is to model the discontinuities and then to de-embed the device under test from a measurement made at the test port connectors based on a highly reliable calibration scheme based on repeatable components with precision connectors.

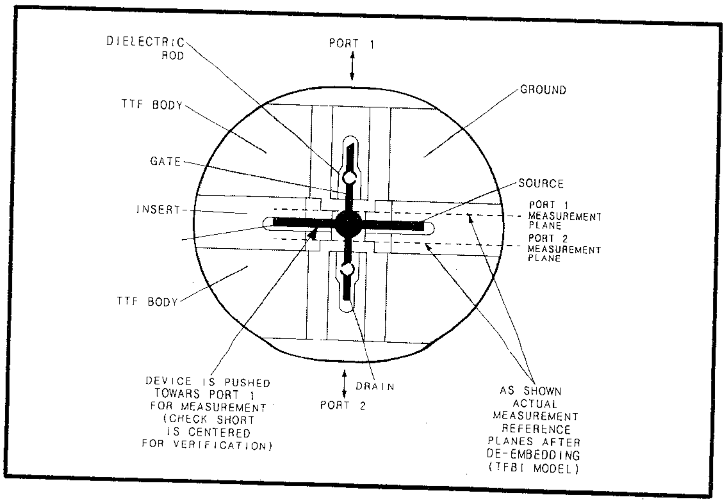
Each element can be modelled. The usual technique is measurement and optimization of the values of the elements using "Touchstone", "Super Compact" or similar. These values are used to create the T-matrix for the de-embedding process. The difficulty of creating high quality, repeatable standards in chip form means that better results are obtained by only using the launch once (for measurement) and not introducing additional uncertainty with a calibration step.

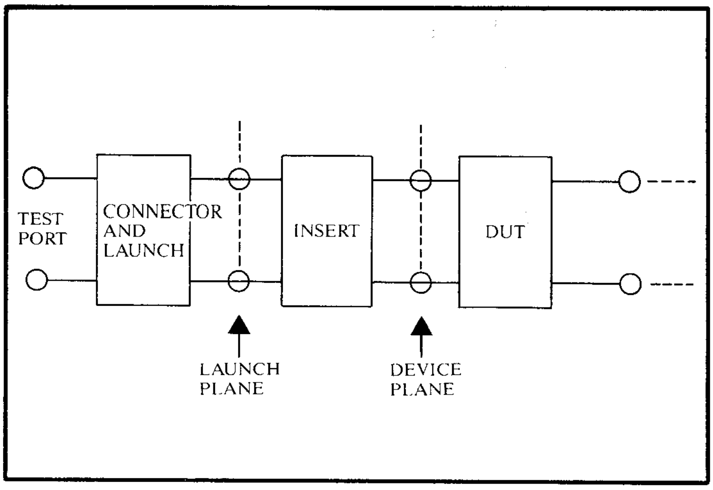


As a second example, let's look at the HP 85041A transistor test fixture. The essential problems are very similar to the situation described earlier.

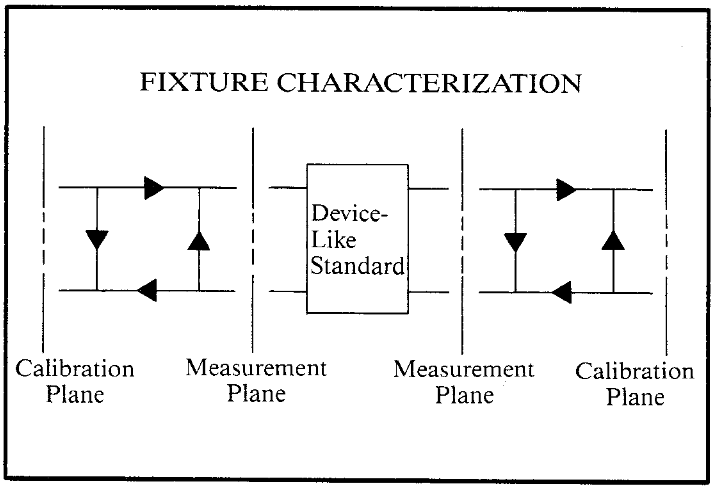
HP85041A
TRANSISTOR TEST
FIXTURE

The transistor to be characterized is necessarily connected to the test port via a set of discontinuities which includes the launch and the parasitics associated with locating the transistor leads in the insert. The approach taken in the HP 85014A software is to model these effects, to cascade the model with the HP 8510 error terms and replace the error terms in the HP 8510 with a new set which includes the fixture model.



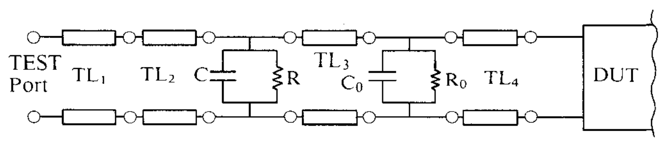


The original approach was to model only the discontinuity caused by the insert using a CLC pi network and to calibrate at the launch plane with open, short, load and thru. This approach appeared to be satisfactory with the HP 8409-series automatic network analyzers. The HP 8510 revealed that the repeatability of these (non-precision) items and the variability of transistors packaging introduced additional errors. The results are poor compared with those obtained by the use of precision (7mm) standards in conjunction with a model for the elements of a more sophisticated equivalent circuit for the launch and the insert.



The technique used to arrive at values for the model elements employs a set of components inserted at the device plane. These include a short circuit the shape and size of a packaged transistor, sections of transistor lead and a very thin insert to characterize the discontinuity as well as improved versions of the calibration components used in the earlier technique.

The model shown in the slide is fitted to the measured data using optimization techniques.



$R = R_0 = 0$

$C = 14$ FF

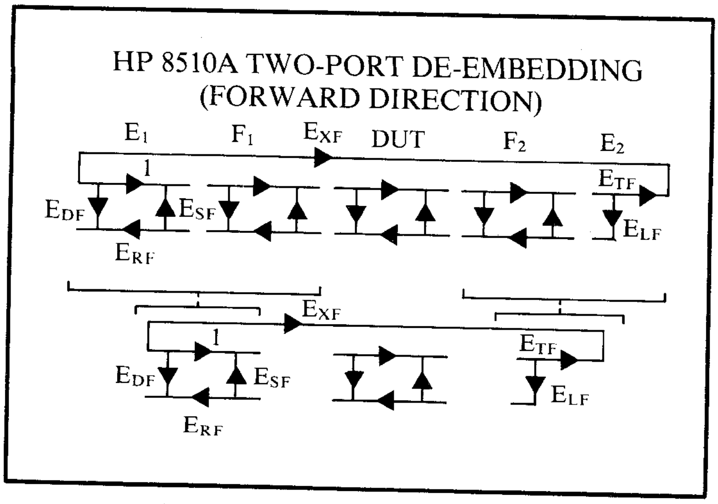
$C_0 = 125$ FF

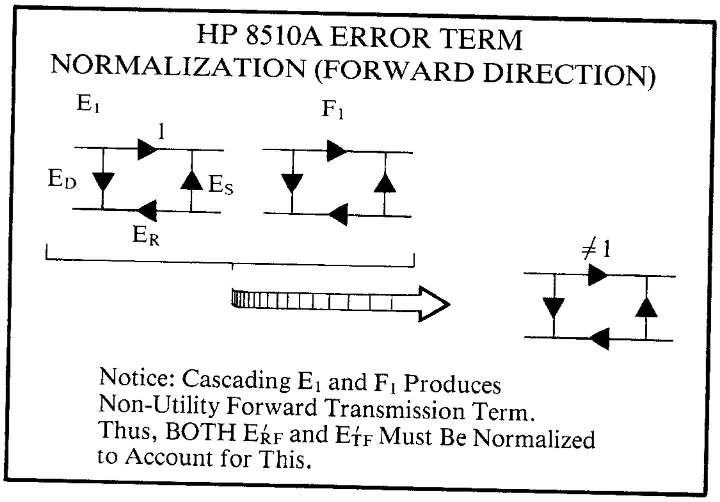
T_l	1	2	3	4	UNITS
z0	50.0	51.4	50.5	240	Ω
l	1.6	0.35	0.25	0.03	Cm
α	.003	.005	.003	.06	neper

All transmission lines are specified in terms of Z_O, length and loss (assumed small but non-zero).

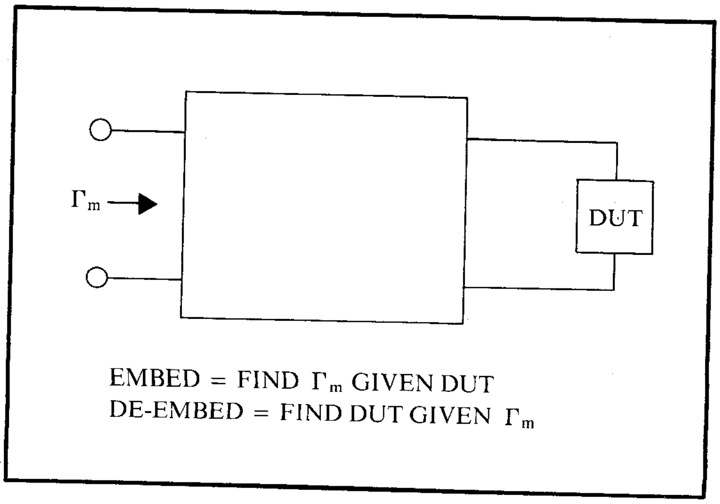
Using this basic principle, it is not a difficult matter to determine the appropriate element values for a model for any custom insert designed for device package styles not supported by the HP 85041A. The major consideration effecting ultimate accuracy is the care taken to ensure good mechanical compatability of the insert with the fixture body and precision manufacture to ensure repeatability.

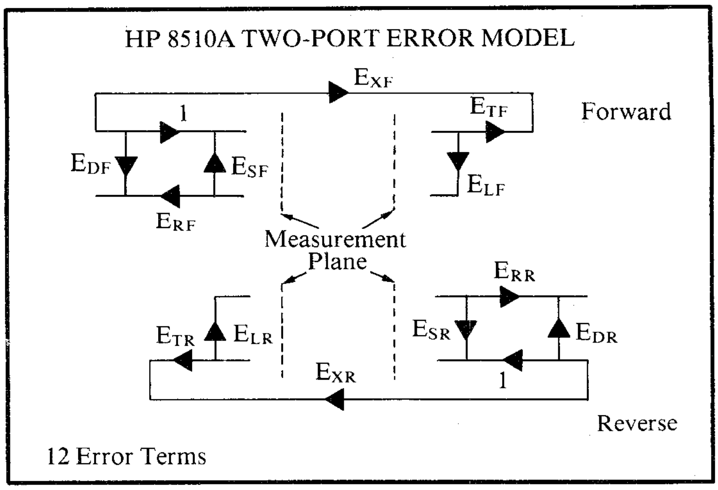
The approach used in the HP 85014A software adds a slight extra degree of complication because of the use of a 12-term error model in the HP 8510 network analyzer. The model requires E_{RF} and E_{TF} in the forward direction (E_{RR} and E_{TR} in the reverse direction) to be normalized. This process is straightforward (see "De-embedded Measurements using the HP 8510 Microwave Network Analyzer", Glenn Elmore) and provides almost real-time measurements of de-embedded transistor S-parameters.



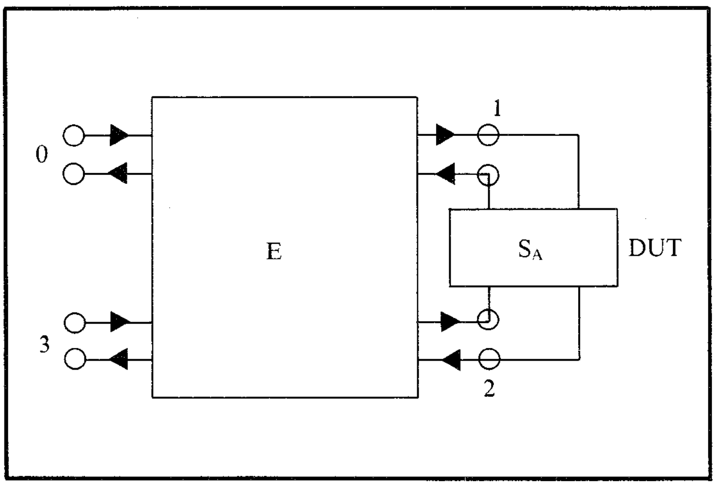


This approach can be used either for de-embedding (error removal to get at the actual value of the unknown device) or embedding (design of a circuit in which to place a known device).





As discussed above, this approach is quite straightforward for simple cascades (no matter how many components or ports are involved but becomes more complicated when there are inter-port couplings. This is illustrated in the slide which shows the two-port (12-term) error model used by the HP 8510. If there are any additional inter-port couplings associated with the devices connected between the measurement planes, the strategy described earlier will break down.



The generic solution for the two-port DUT is to use a four-port network to model the error adapters at both ports and the inter-port couplings. The essence of the arrangement is shown in the diagram; ports 0 and 3 are the measurement ports, the two-port device under test ($\mathbf{S}_A$) is connected between ports 1 and 2.

$$\begin{bmatrix} b_0 \\ b_3 \end{bmatrix} = S_m \begin{bmatrix} a_0 \\ a_3 \end{bmatrix}$$

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = S_A \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

The arrangement can, again, be described in terms of scattering parameters.

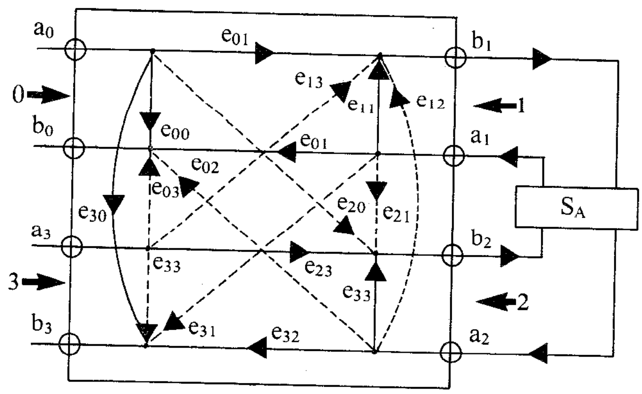
The error adapted is now completely described in terms of a 4x4 matrix with complex entries.

Note that the possibility of effects due to switches to reverse the direction of signal flow to measure reverse parameters can be accounted for by a further **E** matrix for each physical arrangement.

$$\begin{bmatrix} b_0 \\ b_3 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} e_{00} & e_{01} & e_{02} & e_{03} \\ e_{10} & e_{11} & e_{12} & e_{13} \\ e_{20} & e_{21} & e_{22} & e_{23} \\ e_{30} & e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} a_0 \\ a_3 \\ a_1 \\ a_2 \end{bmatrix}$$

The equations are as shown in the slide. It is straightforward matrix algebra to solve $\mathbf{S}_m$ and **E** for $\mathbf{S}_A$ and embed or de-embed a cascade of these four-ports.

To illustrate the generality of the approach, consider the usual 12-term error model interpreted in terms of the four-port error adapter. All 16 terms are shown on the slide, the six terms which appear in the forward direction are shown in solid.



The relation between the conventional 12-term error model notation and the elements of the matrix is shown here.

In filling in the entries in the 4x4 matrix, note how many of the slots are zero. The normalization discussed above in connection with de-embedding and re-inserting the 12 error terms into the 8510 model is now seen to appear naturally in this treatment.

$$e_{20} = e_{02} = e_{31} = e_{13} = e_{12} = e_{21} = 0$$

$$e_{30} = E_X \qquad e_{22} = E_L \qquad e_{32} = E_S$$

$$e_{01} = F_R \qquad e_{00} = E_D \qquad e_{11} = E_S$$

$$e_{10} = e_{23} = 1$$

$$e_{33} = 0 \qquad e_{03} = 0$$

A

$$E = \begin{bmatrix} e_{00} & e_{01} & e_{02} & e_{03} \\ e_{10} & e_{11} & e_{12} & e_{13} \\ e_{20} & e_{21} & e_{22} & e_{23} \\ e_{30} & e_{31} & e_{32} & e_{33} \end{bmatrix} = \begin{bmatrix} E_1 & E_2 \\ E_3 & E_4 \end{bmatrix} = \begin{bmatrix} E_D & 0 & E_R & 0 \\ E_X & 0 & 0 & E_T \\ 1 & 0 & E_S & 0 \\ 0 & 1 & 0 & E_L \end{bmatrix}$$

B

$$T_1 = E_2 - E_1 E_3^{-1} E_4$$

$$T_2 = E_3^{-1}$$

$$T_3 = E_3^{-1} E_4$$

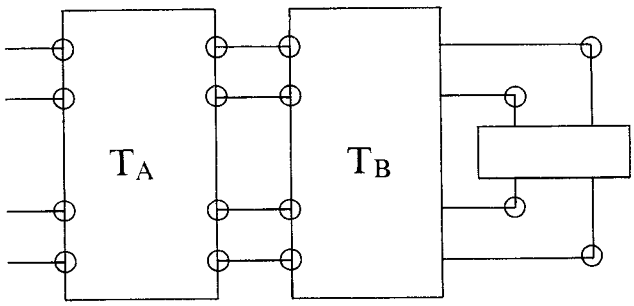
$$T_4 = E_3^{-1}$$

$$T = E_3^{-1} \begin{bmatrix} E_2 E_3 - E_1 E_4 & E_1 \\ -E_4 & 1 \end{bmatrix}$$

A It is convenient at this stage to partition the matrix.

B We now need the T-matrix for this network. Note that the solution is similar to the earlier case and viewing the large matrix in its partitioned form the result can be derived looking identical to the earlier result except that the elements of the T-matrix are themselves matrices.

It is interesting that the solution only requires the inversion of a single 2x2 complex matrix ($\mathbf{E}_3$).



$$T = T_A \cdot T_B$$

These 4x4 T-matrices are readily cascaded and inverted, thus we can now

- o De-embed a two-port DUT
- o Embed a two-port DUT
- o Add or subtract line length at each or all of the four ports.
- o Selectively model parts of the network at one or more ports; this allows dealings with non-insertable DUTs.

$$\begin{bmatrix} b_0 \\ b_3 \end{bmatrix} = S_m \begin{bmatrix} a_0 \\ a_3 \end{bmatrix}$$

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = S_A \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

Now solving the equations, noting carefully the directions of the arrows on the flowgraphs, we have S_M (embedding) ...

... and S_A (de-embedding).

Although these complex matrix equations appear complicated, they are readily implemented on a desk-top computer (see Appendix).

Note direction of arrows!

$$\begin{bmatrix} b_0 \\ b_3 \end{bmatrix} = E_1 \begin{bmatrix} a_0 \\ a_3 \end{bmatrix} + E_2 \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = E_1 \begin{bmatrix} a_0 \\ a_3 \end{bmatrix} + E_2 S_A \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = E_3 \begin{bmatrix} a_0 \\ a_3 \end{bmatrix} + E_4 \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = E_3 \begin{bmatrix} a_0 \\ a_3 \end{bmatrix} + E_4 S_A \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

Thus $[I - E_4 S_A] \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = E_3 \begin{bmatrix} a_0 \\ a_3 \end{bmatrix}$

$$S_M = E_1 + E_2 S_A (I - E_4 S_A)^{-1} E_3$$

and $S_A = [E_3 (S_M - E_1)^{-1} E_2 + E_4]^{-1}$

APPENDIX

Implementation of complex matrix math in HP Series 200 BASIC

* Consider the complex arrays

$$C = A + jB$$
$$D = E + jF$$

then $C * D = (AE - BF) + j(AF + BE)$

Provides complex matrix multiplication using real arrays.

* Consider C in terms of the partitioned matrix

$$C = \begin{bmatrix} A & -B \\ B & A \end{bmatrix}$$

then $C^{-1} = (A + BA^{-1}B)^{-1} + j(B + AB^{-1}A)^{-1}$

and note that

$$(B + AB^{-1}A)^{-1} = -A^{-1}B(A + BA^{-1}B)^{-1}$$

```
10      !-----------------------------------------------------------------------
20      !
30      !              Skeleton program to demonstrate Embedding
40      !               and De-embedding algorithm using 4-port
50      !
60      !-----------------------------------------------------------------------
70      OPTION BASE 1
80      COM E1_r(2,2),E1_i(2,2),E2_r(2,2),E2_i(2,2)        !  Complex arrays
90      COM E3_r(2,2),E3_i(2,2),E4_r(2,2),E4_i(2,2)
100     COM Sm_r(2,2),Sm_i(2,2),Sa_r(2,2),Sa_i(2,2)
110     !
120     DIM I_re(2,2),I_im(2,2)                             !  Identify matrix
130     DIM T_re(2,2),T_im(2,2),T1_re(2,2),T1_im(2,2)       !  Temporary
140     !
150     MAT I_re= IDN                                       !  Set up [I]
160     MAT I_im= (0)
170     !
180     !    Sm = e1*Sa*INV(I-e4*Sa)*e3                          DE-EMBED
190     !
200     Mul(E4_r(*),E4_i(*),Sa_r(*),Sa_i(*),T_re(*),T_im(*))
210     MAT T_re= I_re-T_re
220     MAT T_im= I_im-T_im
230     Inv(T_re(*),T_im(*),T1_re(*),T1_im(*))
240     Mul(Sa_r(*),Sa_i(*),T1_re(*),T1_im(*),T_re(*),T_im(*))
250     Mul(E2_r(*),E2_i(*),T_re(*),T_im(*),T1_re(*),T1_im(*))
260     Mul(T1_re(*),T1_im(*),E3_r(*),E3_i(*),T_re(*),T_im(*))
270     MAT Sm_r= E1_r+T_re
280     MAT Sm_i= E1_i+T_im
290     !
300     !     Sa = INV(e3*INV(Sm-e1)*e2+e4)                      EMBED
310     !
320     MAT T_re= Sm_r-E1_r
330     MAT T_im= Sm_i-E1_i
340     Inv(T_re(*),T_im(*),T1_re(*),T1_im(*))
350     Mul(E3_r(*),E3_i(*),T1_re(*),T1_im(*),T_re(*),T_im(*))
360     Mul(T_re(*),T_im(*),E2_r(*),E2_i(*),T1_re(*),T1_im(*))
370     MAT T_re= T1_re+E4_r
380     MAT T_im= T1_im+E4_i
390     Inv(T_re(*),T_im(*),Sa_r(*),Sa_i(*))
400     !
410     END
420     !
430     !******************** SUBROUTINES ************************************
440     !
450     SUB Mul(A(*),B(*),E(*),F(*),X(*),Y(*))    ! Complex matrix multiplication
460       OPTION BASE 1
470       DIM T(2,2),T1(2,2)                       ! Temporaries
480       !
490       MAT T= A*E
500       MAT T1= B*F
510       MAT X= T-T1                              ! Real part
520       MAT T= A*F
530       MAT T1= B*E
540       MAT Y= T+T1                              ! Imaginary part
550     SUBEND
560     !
570     !-----------------------------------------------------------------------
580     !
590     SUB Inv(A(*),B(*),D(*),E(*))               !  Complex matrix inversion
600       OPTION BASE 1
610       DIM T(2,2),T1(2,2),T2(2,2)               !  Temporary
620       !
630       MAT T2= INV(A)
640       MAT T= B*T2
650       MAT T1= T*B
660       MAT T= A+T1
670       MAT D= INV(T)                            !  Real part
680       MAT T1= B*D
690       MAT T= T2*T1
700       MAT E= (-1)*T                            !  Imaginary part
710     SUBEND
```

HEWLETT PACKARD

5954 - 1578

April 1986

Printed in UK