Reflection De-Embedding for High-Speed I/O Measurements

Sunil Sudhakaran, Daniel Lin
NVIDIA Corporation
Santa Clara, California 95050
ssudhakaran@nvidia.com

***Abstract*— Reflections in high-speed busses can corrupt signal measurements when the probe location is away from the DUT. This paper presents a methodology to remove such reflections. A mathematical formulation of the problem is presented along with solutions to commonly encountered topologies.**

Keywords—Signal Integrity, Probing, Measurements, De-embedding, Reflections, Memory Interface, GDDR, DDR, Serial I/O

I. Introduction

Post-silicon characterization is an integral part of high-speed I/O interface design as it helps validate the methodologies and pre-silicon design simulations. Post-silicon characterization can be broken into two categories: active and passive measurements. Active measurements involve characterizing the interface behavior and can be further classified into virtual eye measurements which rely upon adjusting controller & host settings in software to trace out the eye margin in contrast to analog oriented measurements which capture a signal of interest. Some examples of the latter include eye diagram and jitter measurements with an oscilloscope.

Taking jitter measurements of a high-speed signal is non-trivial. The oscilloscope, probe bandwidth, and measurement location all influence the quality of the measurement. For the measurement location, it should be noted that high-speed signals are electromagnetic waves. As a result, energy can be reflected due to mismatched terminations and parasitics. Consider Fig. 1 which shows a bounce diagram. The *x*-axis is the position vector, so the measurement location can be plotted as a vertical line along the position vector. If the load is not perfectly terminated, then the measurement waveform y_{obs} is distorted from the receiver waveform y_{rx} due to the reflection off of the load (Γ) resulting in a reverse wave suggesting it is optimal to probe as close to the receiver as possible.

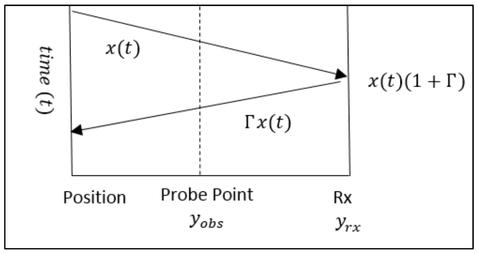


Fig. 1 Simple Bounce Diagram with Probe Point and Receiver

II. Existing Approaches & Limitations

To circumvent the probe point location problem, many serial interfaces have well-defined measurement specifications which include test-point locations (TP) where jitter and eye diagram measurements can be taken past the point of interest. Often times these test points are at a SMP connector and involve de-embedding of the connector fixture on each side to get the DUT response as depicted in Fig. 2. The mathematics behind the de-embedding in Fig. 2 has been widely discussed. As shown in (1), the typical approach is to convert the network to T-parameters and then use matrix inversion to find the desired DUT [2].

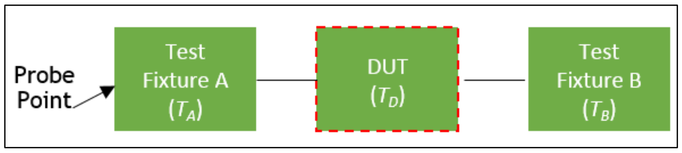


Fig. 2 De-Embedding Approach for DUT in a Test Fixture

$$[T_m] = [T_A][T_D][T_B] \rightarrow [T_D] = [T_A]^{-1}[T_m][T_B]^{-1} \quad (1)$$

Unfortunately, this approach is not applicable for some high-speed interfaces. Some examples of such interfaces include memory standards like LPDDR4 and GDDR5x along with storage I/O interfaces like eMMC. In such interfaces, the receiver test point is located at an inaccessible point in the host package. For GDDR5x, the closest suitable point is at the signal via at the backside of the board at the signal as shown in Fig. 3. The resulting measurement can be severely corrupted as shown in Fig. 4.

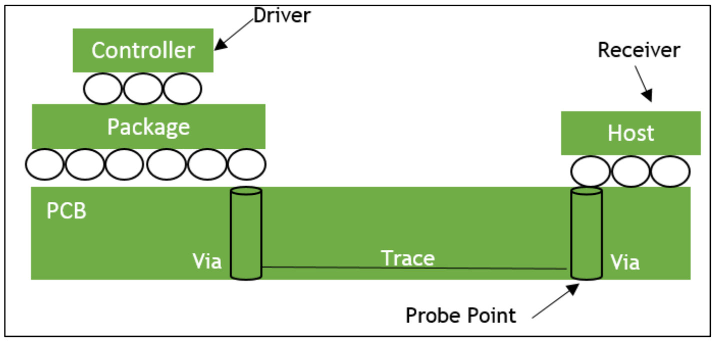


Fig. 3. Measurement Accessibility in Graphics Memory System

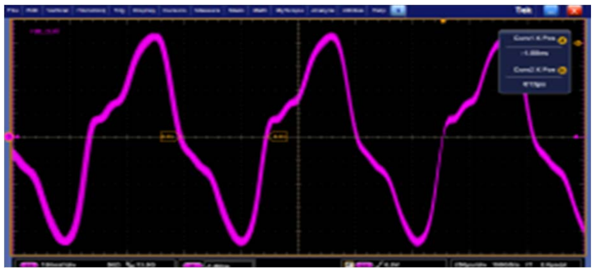


Fig. 4. Measured Waveform Corrupted by Reflections in GDDR5x Bus

III. Proposed Reflection De-embedding Algorithms

As Fig. 4 shows, measurements in such an environment will likely not match what the actual receiver sees. As a result, it is desirable to find a way to remove such reflections analytically. This section introduces an approach on how to derive a mathematical formulation for two commonly used topologies. It should be noted that the framework presented could be applied to other more complex topologies.

A. Single Load Case Deriviation

A mathematical formulation for the interconnect in Fig. 4 is shown below in Fig. 5. The interconnect consists of two channel models represented as S-Parameters.

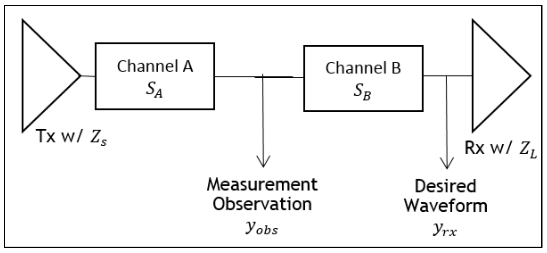


Fig. 5. Block Diagram of Single Load Case

The problem can simply be stated as given y_{obs} can we solve for y_{rx}? A convenient way to visualize the problem is to employ the use of signal flow graphs (SFG). SFG has many applications and are usually applied where a physical system governed by a set of linear equations needs to be modeled. Fig. 6 shows the SFG representation of the network depicted in Fig. 5.

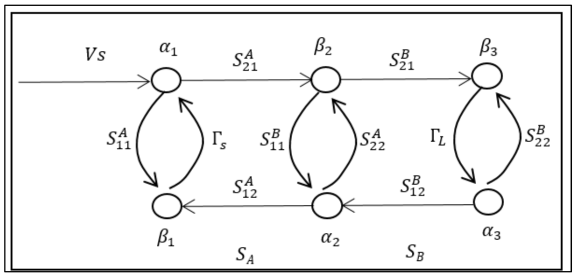


Fig. 6. Signal Flow Graph for Single Load Network

Fig. 6. identifies the waves entering (α_i) and exiting (β_i) from each node in the system. The unity branches (gain of 1) between the entering and existing waves at the middle junction are omitted for succinctness. The measured output and desired output can now be written out as equations based on the SFG as shown in (2)

$$Y_{rx} = \beta_3 + \alpha_3$$
$$Y_{obs} = \beta_2 + \alpha_2 \tag{2}$$

One approach is to use algebraic substitution with equations for all of the nodes. Each node can be represented as an equation consisting of a linear sum of all the input arcs. Equation (3) shows this derivation where Γ_L is the reflection coefficient

$$Y_{obs} = \beta_2 + \alpha_2$$
$$= \beta_2 + S^B_{11}\beta_2 + S^B_{12}\,\alpha_3$$
$$=(S^B_{11}+1)\beta_2+S^B_{12}\Gamma_L\beta_3$$
$$=\frac{\beta_3(1-S^B_{22}\Gamma_L)(S^B_{11}+1)}{S^B_{21}}+S^B_{12}\Gamma_L\beta_3$$
$$\beta_3 = \frac{y_{obs}}{\left(\frac{(1-S^B_{22}\Gamma_L)(S^B_{11}+1)}{S^B_{21}}+S^B_{12}\Gamma_L\right)} \tag{3}$$

To relate Y_{obs} and Y_{rx}, β_2 needs to be related to β_3 (4)

$$\beta_3 = S^B_{21}\beta_2 + S^B_{22}\alpha_3$$
$$\beta_2 = \frac{\beta_3(1-S^B_{22}\Gamma_L)}{S^B_{21}} \tag{4}$$

Substituting (3) into (4) below allows for Y_{rx} to be expressed as a linear function of Y_{obs}.

$$Y_{rx} = \beta_3 + \alpha_3$$
$$Y_{rx} = \beta_3(1+\Gamma_L)$$
$$=\frac{Y_{obs}(1+\Gamma_L)}{\left(\frac{(1-S^B_{22}\Gamma_L)(S^B_{11}+1)}{S^B_{21}}+S^B_{12}\Gamma_L\right)} = Y_{obs}H \tag{5}$$

The time domain output y_{rx} can be derived (6) based on taking the FFT of the measurement Y_{obs} and applying the transfer function from (5). It should be noted that H does not involve the S-matrix from the transmitter to the probe point.

$$y_{rx} = FFT^{-1}\{Y_{obs}H\} \tag{6}$$

B. Double Load Case Derivation

A similar approach can be applied for a double-load case. Examples of such topologies are memory interfaces which are DDP (dual-die package) and consist of one ball (BGA) being routed to two die openings as shown in Fig. 7. This system includes an interposer for measurement which needs to be included in the model.

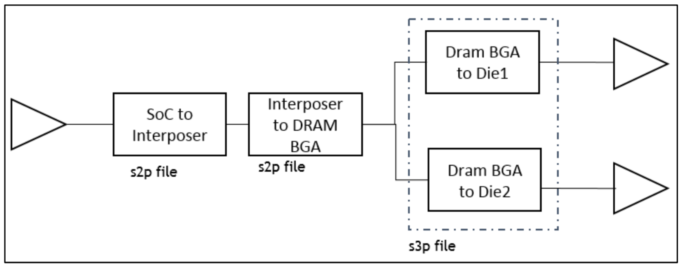


Fig. 7 Block Diagram of Double-Load Case

The corresponding SFG to the network in Fig. 7 is shown in Fig. 8. Note that the S-parameter file representing the double load case is an s3p file instead of an s2p file.

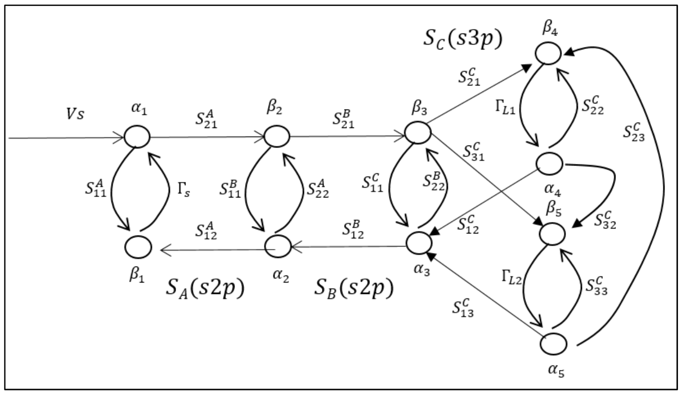


Fig. 8. Signal Flow Graph for Double-Load Case

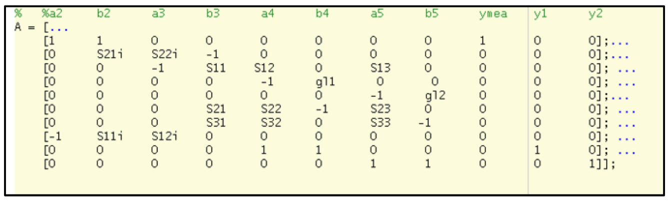
The SFG in Fig. 8. is more complicated to solve by hand, so a computer program was used. The first step is to list out all of the governing equations. Equation (7) shows the signals of interest.

$$y_{rx1} = \beta_4 + \alpha_4,\ y_{rx2} = \beta_5 + \alpha_5,\ y_{obs} = \beta_2 + \alpha_2 \qquad (7)$$

The equations for all of the nodes are shown in (8)

$$\begin{aligned}
\alpha_2 &= S_{11}^B\beta_2 + S_{12}^B\,\alpha_3 \\
\alpha_3 &= S_{11}^C\beta_3 + S_{12}^C\beta_4 + S_{13}^C\beta_5 \\
\beta_3 &= S_{21}^B\beta_2 + S_{22}^B\alpha_3 \\
\alpha_4 &= \Gamma_{L1}\beta_4 \\
\beta_4 &= S_{21}^C\beta_3 + S_{22}^C\alpha_4 + S_{23}^C\alpha_5 \\
\alpha_5 &= \Gamma_{L2}\beta_5 \\
\beta_5 &= S_{31}^C\beta_3 + S_{33}^C\alpha_5 + S_{32}^C\alpha_4
\end{aligned} \qquad (8)$$

To solve this system of equations, an augmented matrix was used as shown in Fig. 9. The solution was derived by reducing the matrix to reduced row echelon form. The solution is omitted as the expression is quite long.



```
%   %a2     b2      a3      b3      a4      b4      a5      b5      ymea    y1      y2
A = [...
    [1      1       0       0       0       0       0       0       1       0       0];...
    [0      S21i    S22i    -1      0       0       0       0       0       0       0];...
    [0      0       -1      S11     S12     0       S13     0       0       0       0]; ...
    [0      0       0       0        -1     g11      0       0      0       0       0]; ...
    [0      0       0       0       0       0       -1      g12     0       0       0];...
    [0      0       0       S21     S22     -1      S23     0       0       0       0]; ...
    [0      0       0       S31     S32     0       S33     -1      0       0       0]; ...
    [-1     S11i    S12i    0       0       0       0       0       0       0       0]; ...
    [0      0       0       0       1       1       0       0       0       1       0]; ...
    [0      0       0       0       0       0       1       1       0       0       1]];
```

Fig. 9. Augmented Matrix to Solve Double-Load Case

IV. Application of Proposed Algorithms

A. *Single Load Case Simulation & Measurement Results*

Corresponding to Fig.5, a spice simulation of a GDDR5 channel with a probe point in the middle was performed. Fig. 9 shows the comparison of the original y_{obs}, the spice waveform corresponding to y_{rx}, and the reconstructed y_{rx} from y_{obs}. The mean square error for the fit is below 100uV.

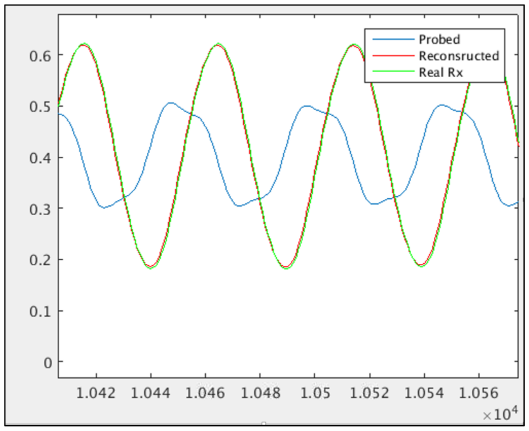
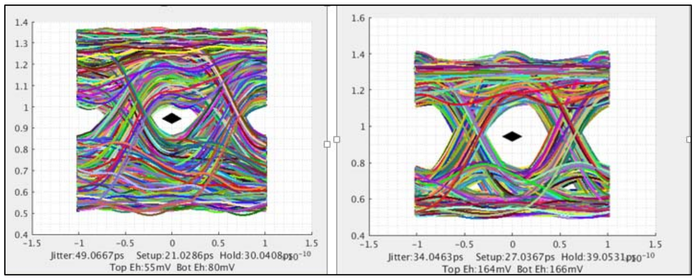


Fig. 10. Single Load Transient Simulation Case Results

The single load algorithm was applied to a measured GDDR5x eye diagram. Fig. 11(a) is the raw scope data while Fig. 11(b) shows the application of the algorithm



(a) (b)

Fig 11. Single Load Case Algorithm Applied to Measured Data

B. *Double Load Case Simulation & Measurement Results*

Fig. 12 shows the algorithm performance in simulation with respect to the double load system in Fig. 8 for one of the loads while the second load (rank) is unterminated. The match shows good correlation between the reconstructed and the actual waveforms.

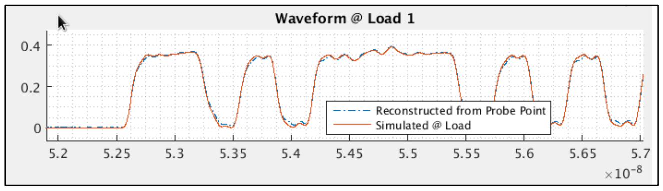


Fig 12. Double Load Case Algorithm Applied to LPDDR4x DDP Simulation

Acknowledgment

The authors would like to acknowledge Emily Lan and Lamar Tatro who helped collect the measurements. Also, they would like to acknowledge Chih-Wei Tsai who generated a lot of the channel models.

References

[1] D. M. Pozar, Microwave Engineering, 2005, Wiley

[2] "Agilent De-embedding and Embedding S-Parameter Networks Using a Vector Network Analyzer," Application Note 13