Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 53(67), Fascicola 1, 2008

# An Algorithm for Automated Translation of Crosstalk Requirements into Physical Design Rules[1]

Marius Rangu[2]

**Abstract – Signal integrity is a major concern when designing printed circuit boards for high speed digital applications, and crosstalk is one of the most important issues. Crosstalk is influenced both by the routing geometry and the electrical parameters of the drivers and receivers on the board, and in order to keep crosstalk noise under control, minimum clearances must be enforced between sensitive and aggressive signal traces. However, the relationship between the crosstalk requirements ( in electrical terms – usually [mV] ) and the physical design rules (in geometrical terms – usually [mm] ) is not very obvious and in order to evaluate it, some form of analysis must be involved. This paper proposes an algorithm designed to automate this process, based on differential impedance equivalence, implemented as a SAX Basic script and embedded into PADS Layout Editor.**

**Keywords: PCB, PADS Layout, crosstalk, clearance, design rules, parallelism**

## I. INTRODUCTION

"Crosstalk is the transfer of pulse energy by the electromagnetic field from a source line to a victim line" [6]. Very often during the operation of printed circuit boards (PCBs), due to the inherent geometrical properties of the interconnection structure - parallel traces on parallel planes - the energy of a signal passing through a copper trace (aggressor) will be transferred on a neighbor trace, thus exhibiting crosstalk. It is the task of the PCB designer to control the amount of crosstalk admitted in such a manner that it will not drastically affect the performances of the circuit, and the mean to do this is to control the interconnection geometry.

The dominant geometrical factor influencing the crosstalk noise is the parallelism between adjacent traces, and since it can't be avoided it must be controlled in order to minimize its effects on signal integrity. There are two possible approaches:

- Hand calculations: done by a skilled engineer, hand calculations are capable to quickly give the PCB designer a general idea of the physical design rules that might keep the crosstalk noise under control. The physical design rules can then be communicated to the CAD software in terms of minimum clearance and / or parallelism between specific signals or signal classes.
- Pre-layout simulations: using a signal integrity analysis software (such as HyperLynx), the engineer may construct and simulate coupling models that are capable to give the PCB designer a more accurate idea of the physical design rules that might keep the crosstalk noise under control. The physical design rules can then be communicated to the CAD software in terms of minimum clearance and / or parallelism.

Both previous paragraphs ended with the same phrase. That is because both methodologies involve basically the same steps: using either hand calculations or a simulator, the PCB designer must translate the electrical requirements of the design into a set of physical design rules. This paper proposes a third approach, intended to speed up the process and make it less vulnerable to human mistakes (in terms of faulty calculations, inappropriate modeling or misinterpretation of results): an algorithm for automated translation of electrical crosstalk requirements into physical design rules.

## II. CROSSTALK ANALYSIS

Considering the two coupled transmission lines in fig. 1 and the signal source *Vs* generating a rising edge at *t=0*, it will take an amount of *TD* time to travel the aggressor line until it reaches the load Rs:

$$TD = X \cdot \sqrt{L_{11} \cdot C_{11}} \text{ [s]} \qquad (1)$$

, where X [m] is length of the aggressor trace, L11 [H/m] it's characteristic inductance and C11 [F/m] it's characteristic capacitance.

Each infinitesimal segment *x* of the lossless transmission line can be modeled as two coupled L-C circuits, as illustrated in figure 2.

[1] This paper is partly supported by Mentor Graphics, the EDA software provider for "Politehnica" University of Timisoara

[2] "Politehnica" University of Timisoara, Electronics and Telecommunication Faculty, 2, V. Parvan Blvd., 300223, Timisoara, Romania, email: marius.rangu@etc.upt.ro

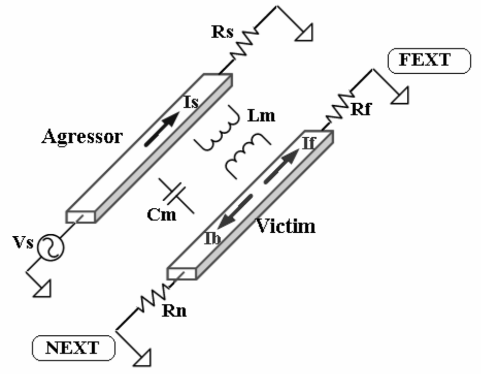


Fig. 1 General crosstalk model

The amount of crosstalk the aggressor signal *Vs* will cause to the victim due to electrical field coupling is a function of the mutual capacitance of the two traces (fig. 1), and the amount of crosstalk cause by the magnetic field coupling is a function of the mutual inductance of the two traces.

The mutual and coupling parasitics are expressed in a matrix form, as in eq. (2), and have the dominant effect on crosstalk noise. In order to evaluate the amount on noise the proximity of two signal traces will cause, the two matrices presented below must first be determined.

$$\begin{cases}[L]=\begin{bmatrix}L_{11} & L_{11}=L_m\\ L_{21}=L_n & L_{22}\end{bmatrix}\\ [C]=\begin{bmatrix}C_{11}=C_{10}+C_m & C_{12}=-C_m\\ C_{21}=-C_m & C_{22}=C_{20}+C_m\end{bmatrix}\end{cases} \qquad (2)$$

We'll further consider only the first transmission line in fig. 1 sourced by a signal generator and the second one passively connected to ground at both ends. The equivalent circuit of an infinitely small segment of the coupled transmission line is represented in fig. 2.

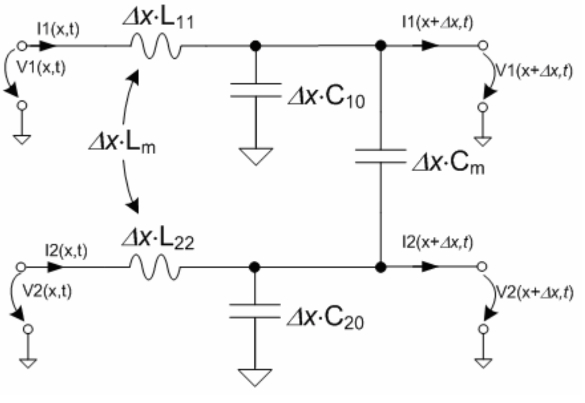


Fig. 2 Distributed model of two coupled transmission lines

Each coupling inductor in each infinitesimal segment Δx will act as a tiny voltage source (fig. 3.a), causing a voltage drop on the victim trace. The voltage induced on each segment of the victim line will cause a current to propagate backward, toward the near end of the trace

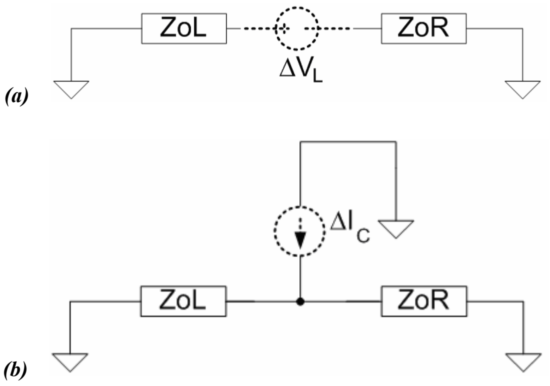


Fig. 3 Equivalent models for inductive (a) and capacitive (b) coupling

Each coupling capacitor in each infinitesimal segment *Δx* will act as a tiny current source (fig. 3.b), injecting a current in the victim trace. The current injected into on each segment of the victim line will propagate on both ends of the trace, causing a forward $I_f$ and a backward current $I_f$, as illustrated in fig. 1.

The current propagating backward represents the sum of the capacitive and the inductive effects, while the current propagating forward is the difference of the capacitive and inductive current, so the crosstalk will have different effects at the far end (FEXT) and at the near end (NEXT) of the victim trace.

## A. *Forward Crosstalk (FEXT)*

Since each segment of the victim trace generates a forward-propagating crosstalk pulse that will travel with the same speed as the aggressor signal, they will both arrive at the far end at the same time, TD calculated with (2). The effect of forward propagation is incremental, so that the far-end-crosstalk (FEXT) signal will be a short pulse of about the rise time of the aggressor signal, with an amplitude proportional with the coupling length, as depicted in fig 4.

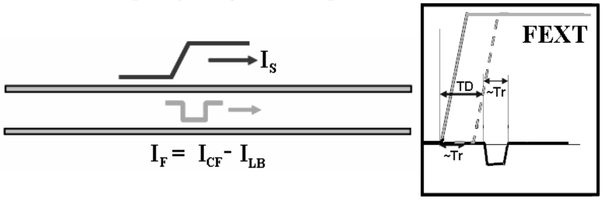


Fig. 4 Forward crosstalk and FEXT waveforms

Subtracting the forward-propagating wave of the circuit in figure 3.a (inductive) from that of the circuit in figure 3.b (capacitive) gives:

$$\frac{\partial v_F(x,t)}{\partial x}=\frac{1}{2}\cdot\left(C_m Zo-\frac{L_m}{Zo}\right)\cdot\frac{\partial v_s(x,t)}{\partial t} \qquad (3)$$

, which gives the amplitude of the noise signal at the far end [4]:

$$V_F=\frac{1}{2}\cdot X\sqrt{LC}\cdot\left(C_m Zo-\frac{L_m}{Zo}\right)\cdot\frac{\Delta Vs}{Tr} \qquad (4)$$

, where $X$ is the length of the coupling region, $L = L_{11} = L_{12}$ the inductance of the transmission lines (considered to have the same geometry), $C = C_{10} + C_m = C_{20} + C_m$ the capacitance of the transmission lines and  the rise time of the aggressor signal.

The sign of the FEXT signal in (4) depends on which of the inductive or capacitive coupling dominates, and is negative for most microstrip transmission lines and zero for symmetrical stripline [5].

### B. *Backward Crosstalk (NEXT)*

The backward-propagating pulse generated by crosstalk will have a constant effect on the near end of the victim, lasting about two times the propagation time of the lines (the time required by the pulse generated by the rightmost segment to reach the left end), as depicted in figure 5.

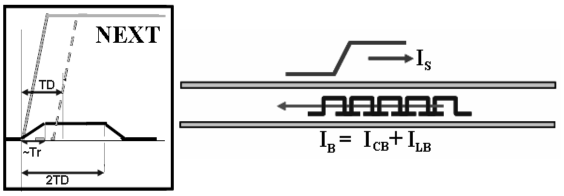


Fig. 5 Forward crosstalk and FEXT waveforms

Adding the backward-propagating waves of the circuits in figure 3.a (inductive) and 3.b (capacitive) gives:

$$\frac{\partial v_B(x,t)}{\partial x} = \frac{1}{2}\cdot\left(C_m Zo + \frac{L_m}{Zo}\right)\cdot\frac{\partial v_s(x,t)}{\partial t} \qquad (5)$$

, which gives the amplitude of the noise signal at the near end [4]:

$$V_B = \frac{1}{4}\cdot\left(C_m Zo + \frac{L_m}{Zo}\right)\cdot\Delta Vs \qquad (6)$$

The amplitude of the NEXT signal does not depend on the length of the coupling region or the rise time of the signal. This is true as long as the round trip of the coupling region ($2\cdot TD$) is larger than the rise time of the aggressor ($Tr$), otherwise the NEXT signal will not have enough time to fully develop [1], and eq. (6) must be adjusted, considering the propagation time from (1), to:

$$V_B^{short} = \frac{2TD}{Tr}\cdot V_F = \frac{1}{2}\cdot\frac{X\sqrt{LC}}{Tr}\cdot\left(C_m Zo + \frac{L_m}{Zo}\right)\cdot\Delta Vs \qquad (7)$$

For short coupling regions the far-end (eq. 4) and near-end (eq. 7) crosstalk depends on the same parameters.

## III. PHYSICAL ANALYSIS

In order to evaluate the crosstalk noise generated by a specific routing geometry, first the parasitics matrices presented in (2) must be evaluated. In order to solve this problem without the use of an electromagnetic field solver, the equivalent impedances of differential propagation modes are considered. In the odd mode currents travels in different directions, so the magnetic fields will differentiate, while the electric fields will add, which gives the equivalent impedance of two coupled transmission lines in odd propagation mode:

$$Z_{odd} = \sqrt{\frac{L_{11} - L_{12}}{C_{11} + C_{12}}} \qquad (8)$$

In the even mode currents travels in the same direction, so the magnetic fields will add, while the electric fields will differentiate, which gives the equivalent impedance of two coupled transmission lines in even propagation mode:

$$Z_{even} = \sqrt{\frac{L_{11} + L_{12}}{C_{11} - C_{12}}} \qquad (9)$$

Similarly, the propagation times in odd and even propagation modes may be expressed as functions of inductive and capacitive parasitics, as:

$$TD_{odd} = \sqrt{(L_{11} - L_{12})\cdot(C_{11} + C_{12})} \qquad (10)$$

$$TD_{even} = \sqrt{(L_{11} + L_{12})\cdot(C_{11} - C_{12})} \qquad (11)$$

The same impedances presented in eq. (8,9) and propagation times presented in (10,11) may be estimated analytically, based on the equivalent dielectric constant of the structure formed by two signal traces in odd and even propagation modes. Bahl and Garg gives in [2] an analytic estimation for differential impedance and propagation time of microstrip transmission lines, while Cohn gives in [3] an analytic estimation for stripline configuration. Due to their complexity, th formulas given in [2] and [3] are not reproduced in this paper, but implemented as part of the crosstalk calculation functions presented further in paragraph IV.

Since $Z_{odd}$, $Z_{even}$, $TD_{odd}$ and $TD_{even}$ may be determined using Bahl and Cohn solutions, eq. (8) and (9), together with the expression of propagation time, may be used to determine the elements of the parasitics matrices:

$$\begin{cases} C_{11} = \frac{1}{2}\cdot\left(\frac{TD_{even}}{Z_{even}} + \frac{TD_{odd}}{Z_{odd}}\right) \\ C_{12} = \frac{1}{2}\cdot\left(\frac{TD_{even}}{Z_{even}} - \frac{TD_{odd}}{Z_{odd}}\right) \\ L_{11} = \frac{1}{2}\cdot\left(TD_{even}\cdot Z_{even} + TD_{odd}\cdot Z_{odd}\right) \\ L_{12} = \frac{1}{2}\cdot\left(TD_{even}\cdot Z_{even} - TD_{odd}\cdot Z_{odd}\right) \end{cases} \qquad (12)$$

## IV. DESIGN RULE TRANSLATION: ELECTRICAL → PHYSICAL

Both forward and backward crosstalk depends on the following geometrical parameters:

- ***S*** = the spacing between traces, restricted to a minimum value by the PCB fabrication technology
- ***W*** = the width of the traces: imposed by characteristic impedance or current requirement for the trace
- ***H*** = the distance to the reference plane, determined by the layer stackup of the board
- ***X*** = the length of the coupling region

Only two parameters, namely S and X, are major contributors in crosstalk. Among those S has a technological minimum and, because of the common practice of channel routing, it should be an integer multiple of W. This greatly simplifies the problem because it becomes practical to avoid solving eq. (4) and (7) in S. The only important parameter that must be controlled is X, the length of the coupled traces, which defines a parallelism rule in a CAD environment.

Eq. (4,7,12), together with the analytical estimation of odd and even mode impedances and propagation times, are used to determine the parallelism rule for a specific set of electrical and geometrical data. The algorithm "**PARAX"** presented in fig 6, applied to a pair of nets from a PCB database, requires two predefined groups of IC terminals, one named "Alpha" and the other "Beta", each containing at least one terminal from each net. Assuming all the relevant information is contained in the PCB database, the algorithm will calculate the parameters $X_{\max}$ and $S$ (as an integer multiple of $S_{\min}$) required to keep the crosstalk below the maximum allowed values specified in the database, considering both NEXT and FEXT for every driver in the group as an aggressor. PARAX assumes that each driver pin in the PCB database is defined by $\Delta Vs$ and *Tr*, each receiver by a maximum noise accepted at the input *Vnoise*, each net by a trace width *W* and the set of available routing layers. It also assumes that the PCB stackup is defined, with known geometry, $\varepsilon_r$ and layers association (routing / power / ground).

The algorithm uses two functions:

- **VB(**driver, victim, S**)**, which calculates the worst case NEXT voltage for the specified driver and victim pins, with the spacing S.
- **XF(**driver, victim, VF**)**, which calculates the coupling length X for a given driver and victim and a given FEXT voltage, VF

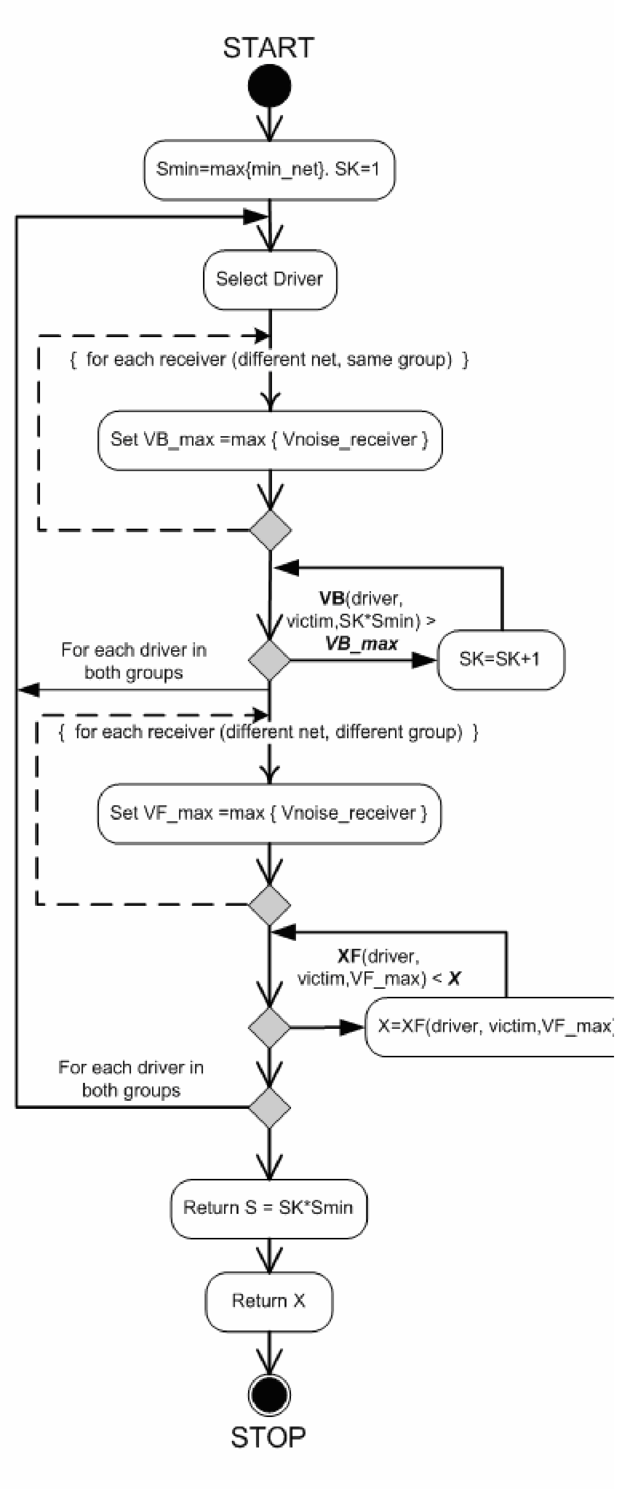

START
Smin=max{min_net}. SK=1
Select Driver
{ for each receiver (different net, same group) }
Set VB_max =max { Vnoise_receiver }
VB(driver, victim,SK*Smin) > VB_max
SK=SK+1
For each driver in both groups
{ for each receiver (different net, different group) }
Set VF_max =max { Vnoise_receiver }
XF(driver, victim,VF_max) < X
X=XF(driver, victim,VF_max
For each driver in both groups
Return S = SK*Smin
Return X
STOP

Fig. 6 PARAX algorithm flowchart

As stated earlier, PARAX can only be applied to a pair of nets and requires that each IC pin on both nets must be assigned, based on proximity, to one of two possible groups, **Alpha** or **Beta**, as depicted in fig. 7.

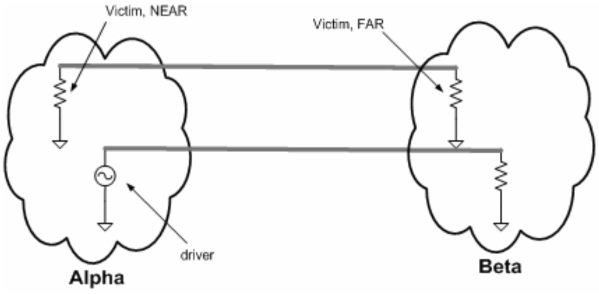


Fig. 7 PARAX partitioning of nets in two groups

The partitioning is used to identify each possible near and far end victims and analyze each case at one time, finally generating the parallelism values that meets the worst case demands.

PARAX will return the *Smin* value as the smallest multiple of the maximum clearance specified for each net, that satisfy the condition:

$$V_B(S_{\min}) < Vnoise_{\max imum\_allowed} \qquad (13)$$

, for every victim on the same group as the driver and for every driver from both nets.

In order to avoid numerical solving of an equation of type "$e^{a1\cdot x+b1} + e^{a2\cdot x+b2} = c$" the algorithm calculates $V_B$ several times until it finds the optimum value.

PARAX will also return the *X* value, according to the pre-determined *Smin*, that satisfy the condition:

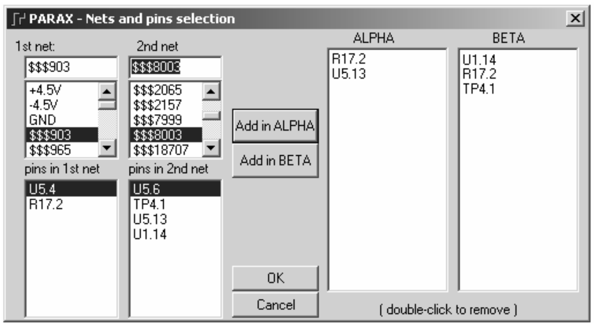
$$V_F(X, S_{\min}) < Vnoise_{\max imum\_allowed} \qquad (14)$$

,for every victim on the same group as the driver and for every driver from both nets.

The calculations are made iteratively, traversing each receiver of the other net for each driver in the net pair. Considering that each pair has *n* pins and half of them are drivers, the complexity of the algorithm is $\mathrm{O}\left(\frac{n^2}{4}\right)$. However, since on most practical applications there is only one driver and a reduced (except for clock signals) number of receivers, it is not computationally intensive.

## IV. IMPLEMENTATION AND RESULTS

The algorithm presented above was implemented as a Basic Script into PADS Layout from Mentor Graphics. The main interface, presented in figure 8, allows the user to select a pair of nets and to partition the IC pins into the two groups, ALPHA and BETA, according to their anticipated positioning on the board.



PARAX - Nets and pins selection

1st net: $$$903

2nd net: $$$8003

+4.5V
-4.5V
GND
$$$903
$$$965

$$$2065
$$$2157
$$$7999
$$$8003
$$$18707

pins in 1st net: U5.4, R17.2

pins in 2nd net: U5.6, TP4.1, U5.13, U1.14

ALPHA: R17.2, U5.13

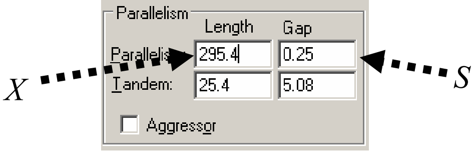
BETA: U1.14, R17.2, TP4.1

Fig. 8 Interface of the PADS Layout implementation of PARAX

Since the algorithm act on the physical design rule of parallelism, the positions of the pins may change, so the partitioning can't be made automatic.

After net selection and pin partitioning, upon pressing the "OK" button, the program will check that the database is consistent with the requirements presented in the previous paragraph. Although all the information needed can be extracted from different electric models, such as industry-standard IBIS or Mentor Graphics MOD, for the purpose of this implementation , all the data required was extracted manually and specified as component parameters.

After the database consistency check, if all the required information are available, the program will run the PARAX algorithm presented in fig. 6. The two values returned, S and X, are then written into the design rules database of the PCB project file. The parallelism rules are available through the main GUI of PADS Layout, so the values calculated by PARAX can be further adjusted if necessary.



Parallelism

| | Length | Gap |
|---|---|---|
| Parallelism: | 295.4 | 0.25 |
| Tandem: | 25.4 | 5.08 |

□ Aggressor

Fig. 9 Parallelism constraints accessed through PADS GUI

For evaluation purposes, we compared the results presented by our implementation with the results obtained using HyperLynx Simulations, for the simple circuit presented in figure 9.a. The driver was XC1701, a PROM for Xilinx with $\Delta Vs = 4.5$ [V] (loaded) and $Tr = 0.5$ [ns]. To avoid reflections, both the aggressor and the victim were terminated on both ends. The typical waveform obtained during simulations are presented in figure 9.b.

The results for NEXT evaluation, for an arbitrary value of X (which does not influence the results) are presented in figure 11.

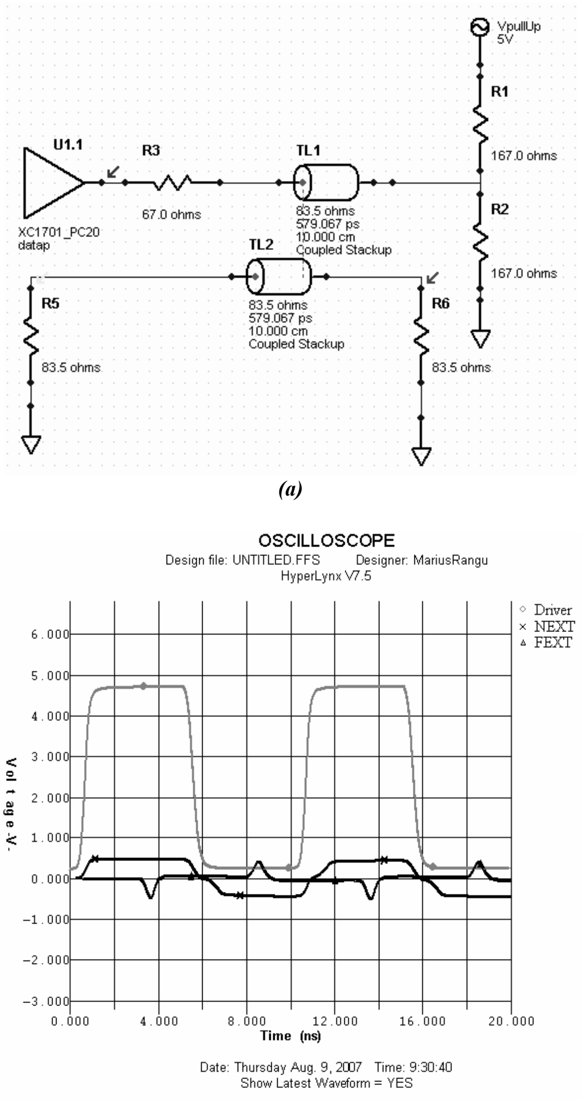


Fig. 10 Simulations using HyperLynx LineSim: (a) Circuit schematic (b) Typical waveforms

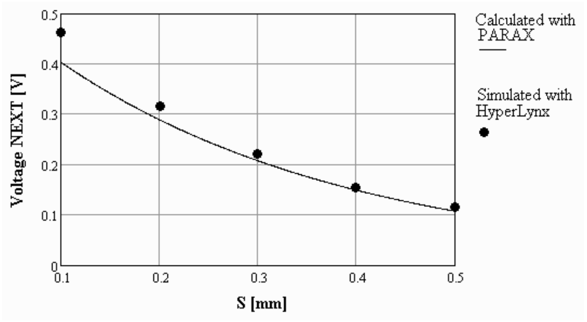


Fig. 11 NEXT comparison of results: HyperLynx vs. PARAX

The voltage noise at the far end is influenced both by *X* and *S*, and the results are presented in table 1, in the form PARAX / Hyperlynx.

| X [cm] | V FEXT [mV] | | |
|---|---|---|---|
| | S=0.15mm | S=0.3mm | S=0.5mm |
| 1 | 3 / 4.4 | 24 / 25 | 13 / 13 |
| 10 | 135 / 153 | 155 / 158 | 129 / 135 |
| 20 | 240 / 251 | 230 / 261 | 250 / 250 |
| 30 | 303 / 337 | 370 / 377 | 329 / 341 |
| 40 | 388 / 403 | 443 / 453 | 402 / 424 |
| 50 | 404 / 451 | 499 / 512 | 465 / 487 |

Table 1 Numerical results for FEXT, in the form PARAX result / HyperLynx Result

## V. CONCLUSIONS

The paper presented an algorithm for fast estimation of crosstalk noise at the near and far end of a victim PCB trace, taking into consideration the characteristics of the driver and the coupling geometry. The main advantage of the algorithm and its implementation is that it allows the PCB designer to specify crosstalk constraints not in geometrical but in electrical terms, as the maximum noise admitted at the input of the receiving circuits, which greatly simplifies the PCB design process for high speed applications.

The crosstalk is underestimated within a 20 percents, mostly because it does not take into considerations the reflections at the ends of the traces. Although it does not provide an accuracy comparable to that of a dedicated simulator (such as HyperLynx), it provides instant results within reasonable limits, and thus may prove to be a valuable design tool.

## REFERENCES

[1] D. Brooks, *Crosstalk Coupling: Single Ended vs. Differential*, Mentor Graphics Technical Paper, 2005, http://www.ultracad.com/mentor/crosstalk_coupling.pdf

[2] I. J. Bahl, R. Garg, *Simple and accurate formulas for a microstrip with finite strip thickness*, Proc. Of IEEE, Vol: 65, Issue: 11, 1977, 1611- 1612

[3] S.B. Cohn, *Characteristic Impedances of Broadside Coupled Strip Transmission Line*, IEEE Trans on MTT, Vol. 8, Nr. 6, 1960, pp 633-637

[4] J. DeFalco, *Reflection and Crosstalk in Logic Circuit Interconnections*, IEEE Spectrum, Vol. 7, July 1970, pp. 44-50

[5] S. Hall, G. Hall, J. McCall, *High-Speed Digital System Design – A Handbook of Interconnect theory and Design Practices*, J. Wiley & Sons, 2000

[6] IPC-D-317A, *Design Guidelines Guideliness for Electronic Packaging Utilizing High-Speed Technique*, IPC, January 1995

[7] H. Johnson, M. Graham, *High-Speed Signal Popagation:Advanced Black Magic*, Prentice Hall, 2003

[8] J. Zhao, *Signal Integrity*, Santa Clara University Lecture Notes, 2006

[9] *** PADS Layout Automation Server Help, Mentor Graphics, 2005