# ESL, Back-annotating Crosstalk Fault Models into High-level Communication Links

Katayoon Basharkhah
*School of Electrical and Computer Engineering, College of Engineering, University of Tehran,*
Tehran, Iran
basharkhah.kt96@ut.ac.ir

Rezgar Sadeghi
*School of Electrical and Computer Engineering, College of Engineering, University of Tehran,*
Tehran, Iran
rr.sadeghi@ut.ac.ir

Nooshin Nosrati
*School of Electrical and Computer Engineering, College of Engineering, University of Tehran,*
Tehran, Iran
nosrati.nooshin@ut.ac.ir

Zainalabedin Navabi
*School of Electrical and Computer Engineering, College of Engineering, University of Tehran,*
Tehran, Iran
navabi@ut.ac.ir

***Abstract*— At the system-level, cores are put together using interconnects that we refer to as high-level communication links. This paper presents an abstract interconnect model for cores connecting to each other to estimate, and thus model, crosstalk noise resulting from the physical properties of interconnects. Such models consider the effects of adjacent wires on each other in the form of weighted transitions. Transition weights are extracted by DC analysis of interconnect SPICE models. These weights form our raw-models, which are then specialized by AC analysis of RLC interconnect models in a mixed-signal simulation environment. The latter analyses establish weight thresholds for glitch faults. Our simulations show that if we were to use only DC-based models for crosstalk faults, we would be over / under-estimating faults as compared with models that are specialized by AC simulation runs. For higher data rates, Specialized models perform an order of magnitude better than DC-based models for crosstalk fault detection.**

***Keywords*— ESL (Electronic System Level), Crosstalk fault, Fault modeling.***

## I. INTRODUCTION

With the GHz clock rates and requirements for faster data transfer, communication issues, e.g., dealing with various faults, have become important. As an example, the 100 Gbps Ethernet uses multiple Gbps channels in parallel that each one can be a potential aggressor or a potential victim, resulting in lines generating crosstalk faults for one another [1]. Affected by high operating frequencies, and requirements for faster transmission, are the on-chip bus-level communications between cores and processing elements.

Concern for faults in bus communications, is highlighted when designing at the system-level (ESL, or Electronic System Level), where low level details of interconnects are not evident and not directly accessible to the designer. This prevents designers from considering interconnect faults in their designs. Thus, making system-level designers unable to devise remedies or designing around interconnect noise and the resulting faults.

In order to correct this situation, or perhaps give a designer tools for wise selection of low-level physical details of communication links, i.e., interconnects, SPICE simulation is available. For high data rates and run-time simulation, SPICE is forbiddingly slow and impractical to use.

A more practical solution is to use SPICE with a limited set of test cases to obtain an abstract model and then use the abstract model in system-level simulations with the actual data. The challenge here is to obtain the models such that abstract models can reasonably estimate crosstalk and other faults caused by low-level physical properties. For this, some consideration must be taken into accounts. First, high-level models should contain process variation and technology information. Since these can lead to crosstalk increase. Second, high-level models should be application-based considering both frequency and data rate. This is where a learning process can help the model to be adapted to the special and dedicated system under design.

Models obtained as such can provide a fast evaluation of interconnects while enabling the designer to model complex high-level embedded architectures. Such models can help in generating different AC test patterns and examination of test solutions, i.e., making decisions regarding scan, BIST, and functional test. On the other side, there would be an opportunity for design space exploration. One can adopt different hardware reliability methods like hamming, error correction and detection in the high-level model for suppression of faults in noisy interconnects.

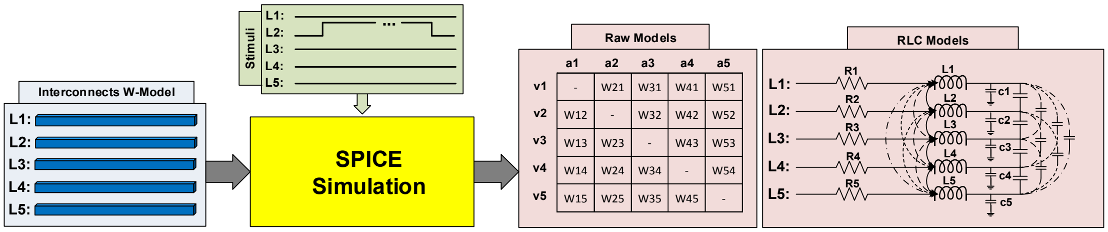
The subject of this work is to take low-level crosstalk faults that occur at the low-level and back-annotate them at the system-level to form a high-level crosstalk fault model. With this fault model, we will be able to run our simulations at the high-level, while using interconnect models that account for faults that occur at the low physical level. This paper proposes a method for back-annotating static and dynamic interconnect properties into high level interconnect models. The interconnects are modeled in SystemC for better integration of today's HDL-based design tools.

## II. RELATED WORKS

Several works have been proposed representing crosstalk fault models. Most of these models are created based on a low-level simulation of interconnects which is not suitable for complex SOC modeling. Some other works model the fault at a higher level but do not consider parameters like frequency and process variations. The work in [2], presents an HDL- level crosstalk defect model considering coupling capacity effects without applying the frequency in the model.

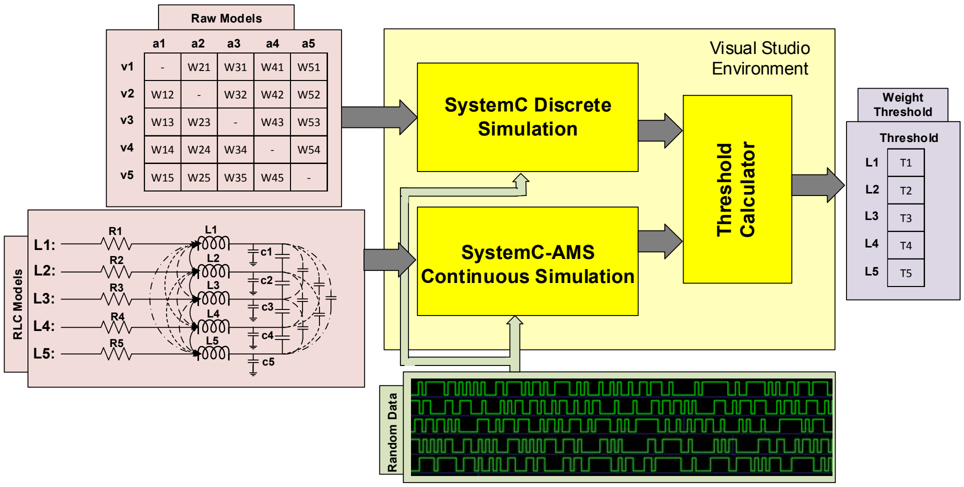
Work in [3] is primarily on testing, and it provides a mapping between system-level faults and actual gate-level faults and not on modeling for analysis of circuits for physical faults due to noise. The work presented in [4], on the other hand, is on a fault model which is based on MDSI fault modeling. MDSI [5] defines odd and even mode transitions to provide a crosstalk fault model. [4] builds upon MDSI and adds weights for a more realistic effect of aggressor transitions on the victim.

978-1-7281-5359-9/20/$31.00 ©2020 IEEE



| | a1 | a2 | a3 | a4 | a5 |
|---|---|---|---|---|---|
| v1 | - | W21 | W31 | W41 | W51 |
| v2 | W12 | - | W32 | W42 | W52 |
| v3 | W13 | W23 | - | W43 | W53 |
| v4 | W14 | W24 | W34 | - | W54 |
| v5 | W15 | W25 | W35 | W45 | - |

Fig. 1. SPICE Simulation for formation of Raw-Models and RLC extraction



| | a1 | a2 | a3 | a4 | a5 |
|---|---|---|---|---|---|
| v1 | - | W21 | W31 | W41 | W51 |
| v2 | W12 | - | W32 | W42 | W52 |
| v3 | W13 | W23 | - | W43 | W53 |
| v4 | W14 | W24 | W34 | - | W54 |
| v5 | W15 | W25 | W35 | W45 | - |

| | Threshold |
|---|---|
| L1 | T1 |
| L2 | T2 |
| L3 | T3 |
| L4 | T4 |
| L5 | T5 |

Fig. 2. Discrete and continuous simulation for calculation of Weight Thresholds

In [6], a failure mode and effects analysis are performed. A fault injection module is provided that evaluates the transaction information like address, data, and control signals, and based on the information, it decides if a fault event occurs if a specified condition for that fault is fulfilled. Although this work investigates the faults in communication and operational parts, no correspondence is made with low-level faults of interconnections.

Our work in this paper is for crosstalk fault models on interconnect. We consider aggressor-victim relations and focus on glitch faults. Although, the fault model is at the system level, low level physical interconnect parameters have been used for creating our model. Our models have parameters related to technology, interconnect length, width and line spacing. Data rate (bps) on interconnects plays a role in creating our models, thus consider frequency of data.

## III. Modeling Strategy

The main theme of this paper is to present crosstalk fault models for the interconnects. We start with a 5-line interconnect in a given technology, with a specific length, width, and other physical properties. We use SPICE and SystemC-AMS simulation to extract our interconnect models that we will use as our system-level interconnect models. This section gives an overview of our model extraction, the details of which will be presented in Section III & IV. Our model extraction procedure is a two-phase process. First we consider interconnect distances and perform DC analysis, and then, we consider more details of interconnects and perform AC analysis. Its effects are evaluated on all victim lines.

Our work in this paper is for crosstalk fault models on interconnects. We consider aggressor-victim relations and focus on glitch faults. Although the fault model is at the system level, low level physical interconnect parameters have been used in the creation of the model. Our model has parameters related to technology, interconnect length, width, and line separations. Data rate (bps) on interconnects plays a role in creation of our models, thus consider frequency of data.

### A. Considering Line Distances (Raw model)

The first phase of crosstalk fault model extraction considers the aggressor-victim relation between interconnect lines. An aggressor places a positive or negative transition on a line, and its effects are evaluated on all victim lines. We start with SPICE W-models for the interconnects for a specific technology, using specific line widths, lengths, and distances between the lines.

As shown in Fig. 1, such a model is used in SPICE simulation (left of Fig. 1) and is subject to single line transition DC analysis. In the five-line interconnect shown, every line becomes a single aggressor and a transition is placed on it. The transition is given enough time to propagate and to cause glitches on other lines (victims). The amplitude of the glitches on the victims is considered as the weight of the glitch caused by the aggressor on the individual victims. This process is performed for aggressor positive and negative transitions and is repeated for every line (1 to 5) playing the role of an aggressor. The result of this SPICE DC analysis is a weight table with positive and negative entries caused by aggressor a on victim v. For a 5-line interconnect this becomes a 5×5 table that we refer to as a weight table, the rows of this table are the victims, and its columns are the aggressors. This table with the associated interconnect line is our *Raw model* for the interconnects, crosstalk fault.

### B. Considering Data Rates (Specialized model)

The *Raw models* discussed above consider single transitions, perform DC analysis, and translate DC effects to imposed weights on victims. This *Raw model* doesn't have any timing information, rate of change of data and simultaneous multiple transitions. The next phase of model extraction adds these effects to the interconnect *Raw model*. We refer to this phase as a model

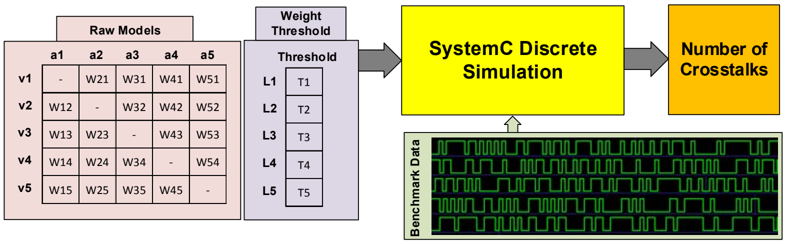


Fig. 3. Using the *Specialized model* in discrete simulation

specialization that specializes the *Raw models* for the specific frequency of use.

As shown in Fig. 2, model specialization uses mixed-mode SystemC, SystemC-AMS simulation environment. Interconnect *Raw model* is used as input of SystemC discrete simulation, and the interconnect RLC model of the specific technology is used for the input of SystemC-AMS continuous simulation. Random data with the frequency of the actual data that will be applied to the interconnects, is applied to both models simultaneously.

The discrete simulation (SystemC simulation) produces sum of weights imposed on each interconnect line as a result of data applied to the interconnect lines. On the other hand, the continuous simulation results of SystemC-AMS produce glitches that are based on the specific technology, physical properties, and technology threshold values. A threshold calculation block written in SystemC considers these simulations and decides what weight ranges on victim lines correspond to the actual glitches of the RLC interconnect models. The minimum of all weights that are marked by the continuous simulation as glitches is marked as the weight threshold for that line.

In effect, this is a learning process for the *Raw model* to decide what weight thresholds to use for crosstalk glitches. As shown in Fig. 2, the result of this process adds a weight threshold table to the weight table. The resulting table has specialized the *Raw model* for the operation frequency. This model becomes the interconnect crosstalk fault model in SystemC discrete simulations.

### *C. Running with Actual Data*

Fig. 3 shows the specialized interconnect crosstalk fault model used in a discrete simulation environment. This model is incorporated in a SystemC model for interconnects and used with actual data. Note that detailed low-level RLC simulations are not needed here, the effect of which is already included in the *Specialized model*s.

## IV. CREATING RAW MODELS

For an interconnect with L lines, the *Raw Model* of the interconnect is an L×L table of weights. The rows of the table are formed by L victims and the columns are aggressors. A table entry represents the effect of aggressor *v* on victim *a* when only that particular aggressor makes a transition. The weights in the table represent the strength of the effect. Two weight values for a positive and negative transition exist for every entry.

A table entry, (*w*+, *w*-), at row *a* and column *v* represents the effect of positive (*w*+) and negative (*w*-) transition of *a* on *v*. the *w*+ and *w*- values are extracted from static SPICE simulation in a given technology.

### *A. Raw Model Principles*

*Raw model* weights are considered for an aggressor transition on a victim. We define a transition interval as the time needed for an aggressor transition to propagate to all its victims. SPICE simulations are done statically (DC analysis) to allow all propagations to occur.

We use Fig. 4 to illustrate how transition effects are to be calculated and what role they play in the calculation of weights. Shown in this figure are hypothetical effects of several aggressors making positive transitions of a single victim. The victim is Line 1, and the aggressors are Lines 2 to 5 that are distanced closest to the farthest to Line 1 in that order. A transition on an aggressor that is closer to the victim, e.g., Line 2, has a larger effect on Line 1, and its effect appears faster (smaller delay) than aggressors that are farther from the victim.

In our calculation of aggressor effects, we consider the amplitude of a glitch as a weight on the victim that is distributed in the entire duration of a transition interval. During the run time, the total effect of all aggressors making a transition in a transition interval is considered as the superposition of the individual aggressor's weights on the victim.

Fig. 5 shows a scenario in which several aggressors make transitions in a transition interval, and how their effects are considered on their common victim (Line 1). As discussed, the

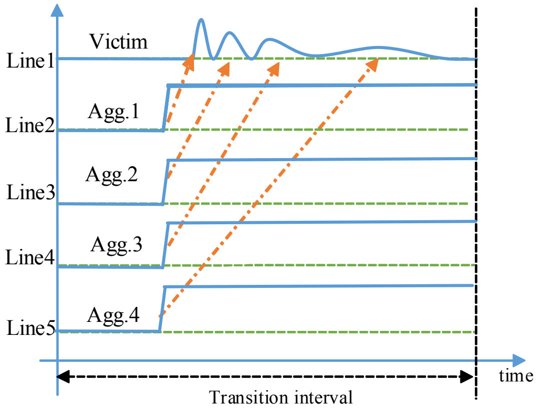


Fig. 4. Aggressor effects in transition interval.

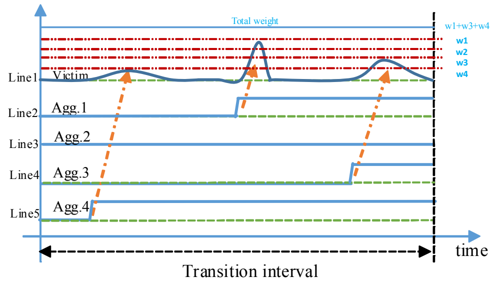


Fig. 5. Superposition of transition effects (using weight of *Raw Model*)

*Raw Model* provides a weight for a victim corresponding to every aggressor transition (shown in the upper part of Fig. 5). In the scenario shown, position weights of aggressors 1, 3 and 5 are added together to calculate the total weight effect on Line 1 (Victim 1).

Fig. 5 shows that regardless of when a transition occurs in a *transition interval*, its weight effect on the victim is a constant that is decided by our *Raw Model* table entries. This is consistent with the DC modeling that is the base of the *Raw Models.*

### B. Obtaining Weights with SPICE Simulation,

The above discussion illustrated what we refer to as *Raw Models*, weights, and *transition intervals*. Furthermore, we showed how *Raw Model* weights are to be calculated, and how they are utilized in calculation of the aggressor transition effects on a victim. This section discusses SPICE simulations for obtaining weight values.

For calculation of the *Raw Model* weights, we used 5-Line SPICE W-interconnect model. The W parameters were extracted for higher metal layers connecting Intellectual Property (IP) cores of SoCs. The technologies considered were 180 nm, 65 nm, and 32 nm. Since we are focusing on higher metal layers, the interconnects that are considered are major bus lines that are placed parallel to each other. This placement alleviates consideration of geometries of interconnects from glitch effects, thus simplifies our analysis.

SPICE simulation runs that follow consider 180 nm technology for five interconnect lines. Timing and glitch amplitudes shown are for this technology only. For a five-Line simulation, every line is designated as the victim, and in eight independent simulation runs positive and negative transitions are imposed on the other four lines. The eight simulation runs account for four positive and four negative transitions. All together we will calculate 40 positive and negative weight values that constitute our *Raw Model.*

Fig. 6 shows two simulation results for a) Line 1 being the victim and a positive transition occurs on Line 2, and b) Line 1 is the victim and a positive transition occurs on Line 3. Note here that line orders are the same as those shown in Fig. 4 and The waveforms in Fig. 6 confirm the distance based transition effects that we discussed in relation to Fig. 4 and Fig. 5.

The weight calculation applies the glitch amplitude and applies it to the entire transition interval. Thus the weights of our *Raw Model* for row 1 column 2, and row 1 column 3 become as follows:

$$W_{12}^{+} = 0.15 \;; \quad W_{13}^{+} = 0.02$$

Other weights (W+ and W-) values are extracted from SPICE simulation runs similarly.

### C. Discrete Simulation Model

The *Raw Model* for the interconnect fault extracted as discussed above has been incorporated in a SystemC channel [7]. The SystemC channel that represents the 5-Line interconnect looks at transitions on its source side and finds an algebraic sum of weights that correspond to the lines having positive or negative transition. Here, a nested for-loop considers every line as a victim and examines the other lines for transitions.

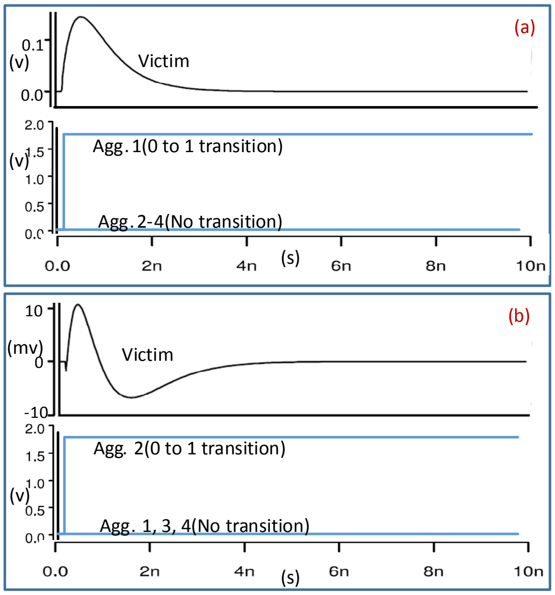


Fig. 6. SPICE simulations, effects on victim.

## V. Specializing Raw Models

The above section showed the creation of *Raw Models* that consider signal transitions in transition intervals that are long enough to allow all the resulting victim propagations to occur. This DC analysis resulted in weight tables that we refer to as *Raw Models*. These models are technology-dependent, but do not consider the frequency of operations or the data rate.

This section performs AC analysis and specializes our *Raw Models* to consider rate of data transmitted over the interconnects. The result becomes our Specialized models.

The simulation setup for such analysis is that of Fig. 2, and as shown, the result of the simulation is extraction of weight threshold for every line of the interconnect. Together, the weight table (*Raw Model*) and weight threshold table form the Specialized model.

The interconnect model used for the continuous simulation part of mixed simulation is based on the RLC models extracted from SPICE interconnect W-model shown in Fig. 1. The RLC model is segmented based on the interconnect lengths. The number of segments is shown based on technology and the desired accuracy. The segmented RLC model is formatted according to SystemC-AMS and used in our analysis. Figure 7 shows the continuous simulation part using the segmented model.

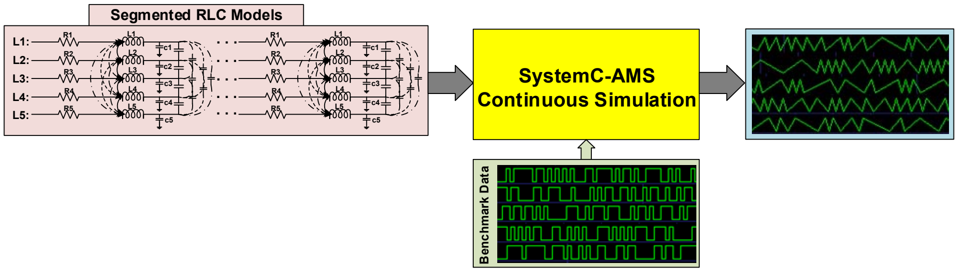
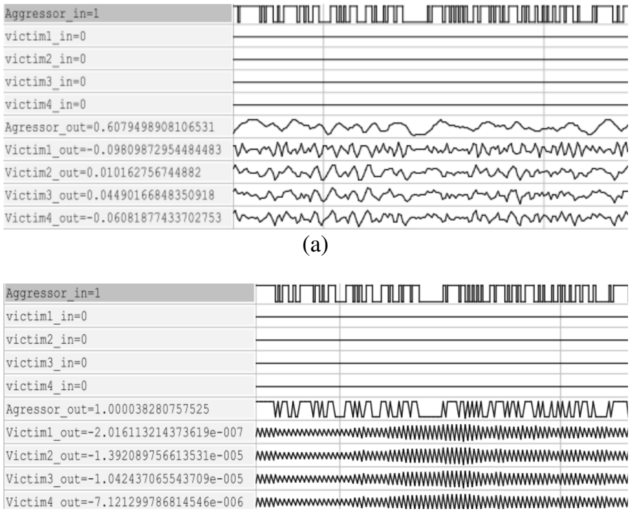


Fig. 7. Continuous AC simulation environment



```
Aggressor_in=1
victim1_in=0
victim2_in=0
victim3_in=0
victim4_in=0
Agressor_out=0.6079498908106531
Victim1_out=-0.09809872954484483
Victim2_out=0.010162756744882
Victim3_out=0.04490166848350918
Victim4_out=-0.06081877433702753
```

(a)

```
Aggressor_in=1
victim1_in=0
victim2_in=0
victim3_in=0
victim4_in=0
Agressor_out=1.000038280757525
Victim1_out=-2.016113214373619e-007
Victim2_out=-1.392089756613531e-005
Victim3_out=-1.042437065543709e-005
Victim4 out=-7.121299786814546e-006
```

(b)

Fig. 8. SystemC-AMS simulation of aggressor-victim transition

### *A. Frequency Dependency*

Because DC models allow for all propagations to occur, glitch amplitudes obtained as such do not represent what occurs at the higher data rates. For higher data rates, a transition, e.g., a positive one, that is closely followed by an opposite transition for the next data set, will both affect the glitch on a victim.

We have illustrated this using SystemC-AMS simulations at two extreme data rates of 1 PS and 1 US. The simulations use our segmented RLC models (as in Fig. 7) to study data rate effects on glitch amplitude due to crosstalk noise. The scenario played here is a single aggressor producing glitches on four victims, where Victim 1 is the closest to the aggressor, and Victim 4 is the farthest. As shown in Fig. 8, at 1 PS (Fig. 8.a) glitch amplitudes on the victims are much higher than those of the 1 US data rate (Fig. 8.b).

This shows the importance of data rate in crosstalk noise, which should be considered in any model we use for crosstalk faults. Although SystemC-AMS continuous simulations with appropriate low-level RLC interconnect models can be used to show the crosstalk faults that occur, this simulation is extremely slow and inefficient for large actual data simulations.

We propose to extract frequency dependencies from sample data SystemC-AMS simulation runs and incorporate them into our *Raw Models*. This alleviates actual data simulation runs from depending on slow SystemC-AMS simulations.

Raw Models

| | a1 | a2 | a3 | a4 | a5 |
|---|---|---|---|---|---|
| v1 | - | 0.627 | 0.241 | 0.0997 | 0.0514 |
| v2 | 0.627 | - | 0.627 | 0.241 | 0.0997 |
| v3 | 0.241 | 0.627 | - | 0.627 | 0.241 |
| v4 | 0.0997 | 0.241 | 0.627 | - | 0.627 |
| v5 | 0.0514 | 0.0997 | 0.241 | 0.627 | - |

Weight Threshold

| | Threshold |
|---|---|
| L1 | 0.0514955 |
| L2 | 0. 099724 |
| L3 | 0.2419136 |
| L4 | 0.2419136 |
| L5 | 0.0997245 |

Fig. 9. *Specialized model* for 45 ns technology, 100 Mega

### *B. Calculation of Weight Threshold*

The peak voltage imposed on a victim decides on the occurrence of a crosstalk fault on the victim. On the other hand, our *Raw Models* facilitate calculation of the total weight on a victim as a result of an aggressor transition.

Specializing an interconnect *Raw Model* to account for a given operating frequency (data rate) is done by calculating a weight threshold that appears as a one-dimensional array of positive and negative threshold values corresponding to every line of the interconnect System.

Calculation of weight thresholds is done by a continuous SystemC-AMS simulation alongside a discrete SystemC simulation. While the continuous simulation decides exactly when the crosstalk faults occur, the discrete simulation calculates the corresponding weights. The comparison of these two results (Fig. 2) determines the minimum total weight on a victim that corresponds to an actual glitch.

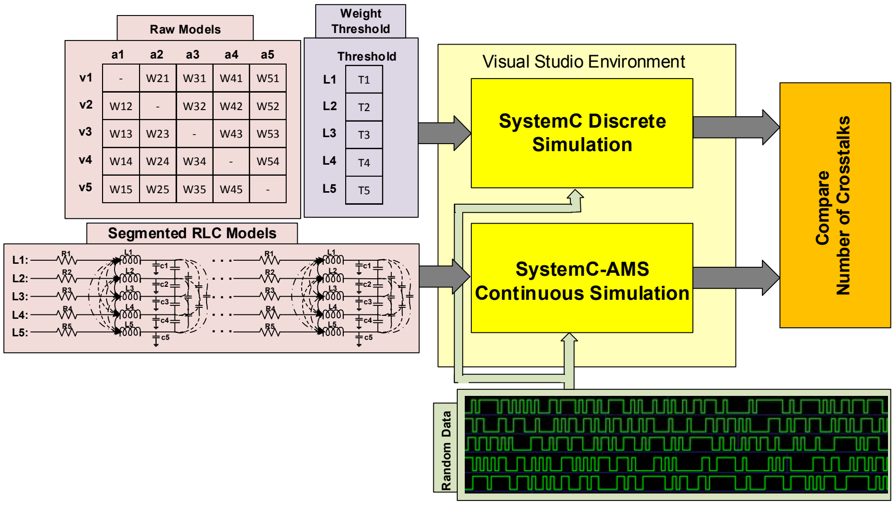
This simulation uses a sample data for both simulations and is considered a learning process for the discrete simulation model (*Raw Model*) to be appended with weight thresholds that are to be regarded as glitches. Fig. 9 shows the *Specialized model* of a 5-line interconnect for 45 nm technology operating at 50 MHz frequency. While the weight table is fixed for this technology, the weight thresholds vary depending on the frequency or data rate.

## VI. EXPERIMENTAL RESULTS

This section describes the experimental setup based on the previous discussions.

### A. *Simulations for Raw Models*

As shown in Fig. 10 two separate simulations are performed


Raw Models
Weight Threshold
Threshold
Segmented RLC Models
L1:
L2:
L3:
L4:
L5:
R1
R2
R3
R4
R5
L1
L2
L3
L4
L5
c1
c2
c3
c4
c5
Visual Studio Environment
SystemC Discrete Simulation
SystemC-AMS Continuous Simulation
Compare Number of Crosstalks
Random Data

| | a1 | a2 | a3 | a4 | a5 |
|---|---|---|---|---|---|
| v1 | - | W21 | W31 | W41 | W51 |
| v2 | W12 | - | W32 | W42 | W52 |
| v3 | W13 | W23 | - | W43 | W53 |
| v4 | W14 | W24 | W34 | - | W54 |
| v5 | W15 | W25 | W35 | W45 | - |

| | Threshold |
|---|---|
| L1 | T1 |
| L2 | T2 |
| L3 | T3 |
| L4 | T4 |
| L5 | T5 |

Fig. 10. Simulation setup for comparing *Specialized models* with segmented RLC models

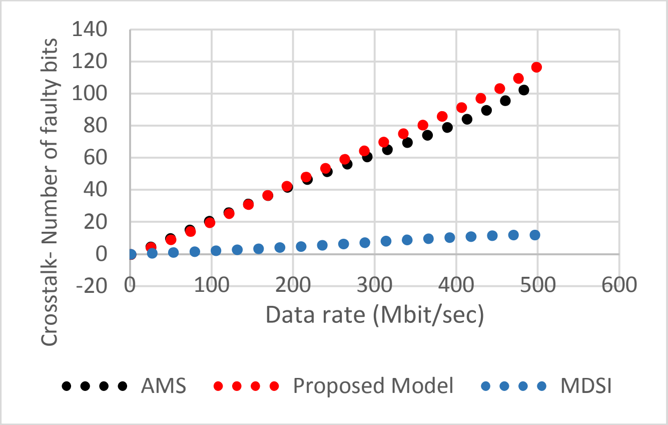


Fig. 11. Comparison of proposed, AMS and MDSI models

to verify our proposed crosstalk fault model, i.e., specialized model. In one simulation our specialized model is used with SystemC discrete simulation engine, and in the other simulation the segmented RLC interconnect model is used with SystemC-AMS continuous simulation engine. The specialized and the RLC models are both for the 180 nm technology. The same data input is used for both simulations. The data is applied to both engines at rates between 0 and 500 Mbps. The two simulation runs count the number of crosstalk faults reported by each simulation engine.

Fault count results are shown in Fig. 11. In addition, we are comparing our results with MDSI that is a well-established crosstalk fault model. In Fig. 11, the red dotted line represents the number of crosstalk faults for different data rates (Mbit/sec) in SystemC-AMS simulations. The blue line shows the same for the MDSI model. As shown, the number of crosstalk faults in the MDSI model is not affected by increasing data rates and frequency. On the other hand, the actual RLC models show a significant linear increase in crosstalk faults when moving toward higher data rates. The simulation results for the proposed model (Specialized model) is also shown in this figure by the black dotted line. Fig. 11 shows that our Specialized model for this technology very closely follows the actual RLC continuous simulation. Our model shows an absolute deviation of 0.08 as compared with the RLC models. We show that deviation between RLC and Specialized models increases for higher data rates. This is due to accumulation of transition effects, from one data set to the next for higher frequencies.

### B. *Simulations Using Specialized Models*

To investigate the performance of the proposed Specialized model, SPEC CPU2006 benchmarks [8] that include *gcc*, *gromacs*, *milc*, *perbench* and *torno* transfer data sets were tested separately in high level specialized and SystemC-AMS analog models. Fig. 12 shows the number of crosstalks for this benchmark. The simulation was performed for 45 nm technology and 5000 sample data. Based on this chart, continuous AMS and the proposed model are near in the number of crosstalk with a percentage error of 0.04. These models are also compared in terms of simulation runtime in TABLE I. Simulation of the Specialized model has a much shorter runtime than the SystemC-AMS electrical model. Compare for example 88.4 seconds with the fastest SystemC-AMS results of 4885.7 seconds. This happens for all transmission data rates and is more significant at the higher data rates. SystemC-AMS electrical models are from 55 to 1800 times slower depending on the frequency. This faster performance is along with the advantage of back annotating low-level effects and considering the effect of data transition rate.

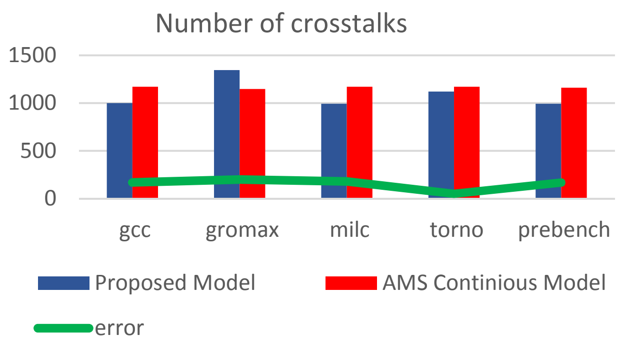


Fig. 12. Results for benchmark CPU2006

TABLE I
Runtime SystemC versus AMS

| Data rate Mbps | Runtime (second) | |
|---|---|---|
| | SystemC | SystemC-AMS |
| 1 | 88.481 | 4886 |
| 10 | 88.481 | 10537 |
| 50 | 88.481 | 17863 |
| 100 | 88.481 | 89314 |
| 500 | 88.481 | 159069 |

## I. Conclusion

In this paper we have shown creation of an abstract interconnect model that considers crosstalk faults. We have shown that our models run several orders of magnitude faster than RLC interconnect models. The penalty we are paying is less than 10% but increases with higher data rates. Our models are based on basic DC interconnect models that are trained for various frequencies.

Our models allow system level designers to be able to plan their design for avoiding crosstalk faults or building mechanisms in their designs to recover from faults when they occur. The Specialized models discussed here can be implemented in any HDL based discrete simulation engine. In works related to this, we have built SystemC channels that incorporate such models. A simple C++ program with access to the weight and weight threshold tables can easily implement our crosstalk fault model.

## References

[1] P. Papadopoulos, A. Raman, Y. Koutsoyannopoulos, N. Provatas and M. Abadir, "Challenges and trends in SOC Electromagnetic (EM) Crosstalk," *2017 IEEE 2nd International Verification and Security Workshop (IVSW)*, Thessaloniki, 2017, pp. 63-69.

[2] M. Cuviello, S. Dey, X. Bai, and Y. Zhao, “Fault Modeling and Simulation for Crosstalk in System-on-Chip Interconnects, ” *Proc. Int. Conf. Computer-aided Design*, 1999, pp. 297–303.

[3] F. Javaheri, M. Namaki-Shoushtari, P. Kamranfar, and Z. Navabi, “Mapping Transaction Level Faults to Stuck-At Faults in Communication Hardware,” in *Test Symposium (ATS), 2011 20th Asian*, 2011, pp. 114–119.

[4] R. Sadeghi, N. Nosrati, K. Basharkhah, and Z. Navabi, “Back-annotation of Interconnect Physical Properties for System-Level Crosstalk Modeling,” in *Test Symposium (ETS), 2019 24th IEEE European*, 2019.

[5] S. Chun, Y. Kim, and S. Kang, “MDSI: Signal integrity interconnect fault modeling and testing for SOCs,” *J. Electron. Test.*, vol. 23, no. 4, pp. 357–362, 2007.

[6] R. Camposano, “Automating system implementation from system specification,” *Talk Synopsys Univ. Day Towaard Syst. Silicon, Aachen*, 1997.

[7] N. Nosrati, K. Basharkhah, R. Sadeghi, and Z. Navabi, “An ESL Environment for Modeling Electrical Interconnect Faults,” in *2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2019.

[8] J. L. Henning, “SPEC CPU2006 benchmark descriptions”, *SIGARCH Computer Architecture News*, vol. 34, no. 4, pp. 1-17, 2006.