A Fast Word-Level Statistical Estimator of Intra-Bus Crosstalk

Suvodeep Gupta and Srinivas Katkoori
Department of Computer Science and Engineering
University of South Florida, Tampa, FL-33620
Email: {sgupta2, katkoori}@csee.usf.edu

Abstract

Given word-level statistics, namely mean, standard deviation, and lag-one temporal correlation of input data, we estimate the bit-level crosstalk probability on a system bus using a non-enumerative statistical approach. We introduce a sampling technique for fast evaluation of integrals during the estimation process. We had proposed two techniques previously - (a) a stream-based estimator that counts crosstalk events on a bus; and (b) a statistical enumeration technique that enumerates crosstalk-producing values on a bus and computes their occurrence probability. Both these techniques suffer from exponential time complexity with respect to the bus-width. In this work, we propose a statistical non-enumerative technique that has linear time complexity with respect to the bus-width. We achieve the linear complexity by resorting to: (1) manipulating the data stream to make the crosstalk-producing values contiguous and (2) sampling the distribution function and storing it as a lookup table. Experimental results for data streams from different data environments are presented, compared against the stream-based approach. Average errors of less than 12% are obtained for bus-widths ranging from 8b to 32b.

1. Introduction and problem formulation

In DSM technologies, the coupling capacitance between neighboring wires is significant compared to individual wire-to-substrate capacitances. For on-chip *buses*, the coupling capacitance has the dual properties of increasing delay as well as power consumption. These are the *crosstalk effects* which need to be accurately predicted for good designs.

Existing crosstalk *estimation* methods either use transient analysis techniques [5] or closed-form equations which capture the transient effects without explicitly analyzing them [4]. The transient analysis is computationally expensive while the alternative metrics rely on physical-level parameters such as victim-to-aggressor impedances and the input signal nature. Moreover, the objective functions of the estimation techniques have so far been the coupling noise *amplitude* and *pulse width*. Among system-level estimators, the Devgan metric, based on control theory, has been analyzed and improved upon by Kuhlmann et.al. in [6]. Vittal et.al. derive equations for the coupled noise voltage in [1] and reduce the coupled noise using layout techniques. The worst case crosstalk noise has been estimated in [3] using the maximum noise pulse height as the metric. It is shown in [5] that such a metric may be estimated incorrectly if the driver strengths of the affected lines are not accounted for.

Our approach differs from previous estimation techniques in that it addresses the intra-bus crosstalk problem at the high-level. The inherent advantage of design-space exploration at the high-level warrants the need for a fast crosstalk estimator during the process of high level synthesis. The main contributions of our work are as follows:

1. Given only *word-level statistics* of a system bus, we propose a purely statistical technique that estimates the probability of crosstalk on each bus line.
2. For efficient scaling of the proposed technique with large bus-widths, we introduce a *sampling* technique to evaluate definite integrals for discrete-valued random variables. This linearizes the time complexity of the estimation process with respect to the bus-width.
3. The proposed technique is independent of data stream lengths. Hence, the runtimes are considerably reduced while reasonable accuracy is maintained.

Two wires with coupling capacitance between them share an *aggressor-victim relationship* with one another i.e. a transition on one of the wires called the aggressor may cause spurious transitions or may adversely affect the timing on the other wire called the victim. Figure 1 [9] shows the victim wire *V* entrapped between the aggressors *A1* and *A2*.

Table 1 gives a summary of the possible crosstalk effects. In Figure 2, spikes 2 and 3 indicate bootstrap noise on the

1530-1591/04 $20.00 (c) 2004 IEEE

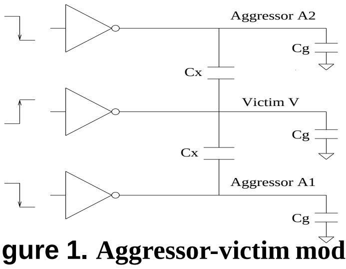

Figure 1. **Aggressor-victim model**

Aggressor transition	Victim transition	Crosstalk effect
0 → 1	0 → 0	Upward spike
1 → 0	0 → 0	Bootstrap spike
0 → 1	1 → 1	Bootstrap spike
1 → 0	1 → 1	Downward spike
0 → 1	0 → 1	Vic transition hastened
1 → 0	0 → 1	Vic transition delayed
1 → 0	1 → 0	Vic transition hastened
0 → 1	1 → 0	Vic transition delayed

Table 1. **Possible crosstalk effects**

victim while 1 and 4 indicate upward and downward spikes respectively. In Figure 3, regions 1 and 4 indicate hastening of the victim's transitions while 2 and 3 indicate the opposition to the victim's transitions. These plots were obtained using HSPICE by simulating a circuit with one aggressor and one victim.

Based on the number of crosstalk events on every line of the bus, we evaluate a probability that indicates the amount of crosstalk activity on that line. The advantage of our approach is that it gives the designer an idea as to which bus lines are more susceptible to crosstalk.

Section 2 summarizes two approaches to the crosstalk estimation problem, previously proposed by the authors [7], and explains their drawbacks which are alleviated by the currently proposed statistical approach. Section 3 presents the proposed word-level statistical estimator that linearizes the previously exponential complexity. Section 4 presents experimental results along with their analysis. Section 5 presents conclusions.

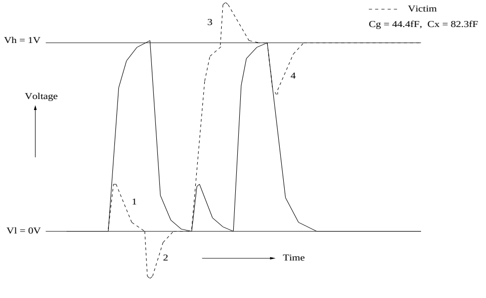

Figure 2. **Crosstalk spikes on victims**

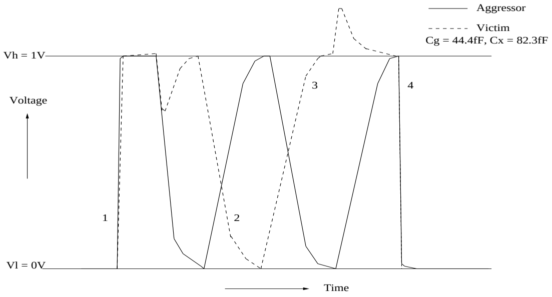

Figure 3. **Crosstalk delays in victims**

2. Stream-based and statistical enumerative techniques

The authors have previously proposed two separate techniques to estimate the crosstalk probability [7].

1. *Stream-based technique*: It computes a crosstalk probability for each bus line from the total number of crosstalk-producing patterns in the data stream and the length of the stream. This has the same accuracy as HSPICE but runs faster. The disadvantage is that the runtimes increase for long data streams.
2. *Statistical enumeration technique*: To overcome the disadvantage, this technique utilizes word-level statistics of the bus instead of data streams. It converts the lag-1 temporal characteristic of the data stream $x(n)$ to a spatial characteristic between $x(n-1)$ and $x(n)$ by concatenating them, thereby avoiding the difficult task of evaluating the temporal characteristic from word-level parameters. As an example, the lag-1 temporal correlation between $x(n-1)$ and $x(n)$ on a 8-bit bus is converted to the spatial characteristic of a 16-bit value $X(n)$ with $x(n-1)$ given by bits *b15-b8* and $x(n)$ given by bits *b7-b0* as shown in Figure 4. Now, if bits *b10-b8* and bits *b2-b0* are filled with crosstalk patterns, it means that $x(n-1)$ transitions to $x(n)$ such that crosstalk occurs on line *b1* of the bus *b7-b0*. Filling the remaining bits of $X(n)$ with either '0' or '1' yields a 16-bit value corresponding to crosstalk on line *b1*. The technique evaluates all such values and sums up their occurrence probabilities to compute the crosstalk probabilty for each line of the bus.

Although the statistical enumerative approach improves the run times significantly, compared to the stream-based approach as shown in Table 2, its complexity is $O(2^{2m})$ where *m* is the width of the data bus. This arises from the fact that for every line of the bus, all values containing

Bus-width	Length	Stream-based	Enumerator
8b	3000	77s	7.2s
8b	6500	196s	7.8s
10b	9500	262s	190s
10b	14500	337s	191s

Table 2. Str-based vs Stat enumerator run times

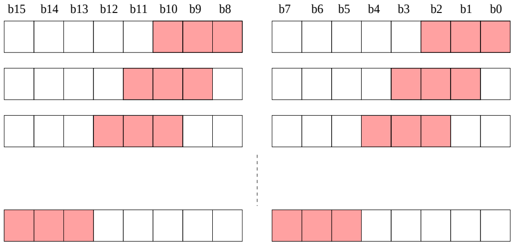

Figure 4. Discontinuous crosstalk windows

crosstalk-producing patterns need to be enumerated in order to estimate their probability. The execution time of the algorithm rises with increase in the bus-width as is evident from Table 2.

3. Proposed statistical non-enumerative approach

Let us assume that the signal $x(n)$ on the bus has a normal distribution. There is no restriction on the distribution of $x(n)$ and the normal distribution is assumed as a general case [2]. Typically, the probability distribution of $x(n)$ can be estimated from the ARMA signal generation models. We assume that the distribution of $x(n)$ is known beforehand [8]. The probability p_i of the ith bit b_i in the data word on the bus is the probability that the ith line of the bus is a 1. This is given by $p_i = Pr(x(n) \in \zeta_i) = \sum_{j \in \zeta_i} \frac{1}{\sigma\sqrt{2\pi}} e^{-(j-\mu)^2/2\sigma^2}$, where ζ_i is the set of all elements in ζ whose ith bit is 1 [8]. The lag-1 temporal correlation of a data stream $x(n)$ is dependent on the cross-covariance of $x(n-1)$ and $x(n)$. The cross-covariance is independent of the mean and standard deviation of $x(n)$ except for specific cases. Thus, it is difficult to accurately predict the temporal characteristic of a data stream $x(n)$ from word-level μ and σ. Besides, it is cumbersome to compute the same from bit-level information. Hence, the proposed technique transforms the temporal characteristic of the data stream $x(n)$ to the spatial characteristic of $x(n-1)$ and $x(n)$ using a concatenation procedure depicted in Figure 5. For a m-bit wide on-chip bus, the concatenated data stream $X(n)$ can be expressed in terms of the component streams $x(n-1)$ and $x(n)$ using the following equation:

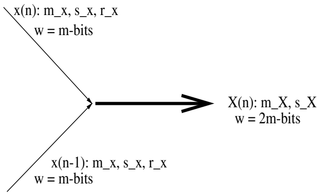

Figure 5. Concatenation

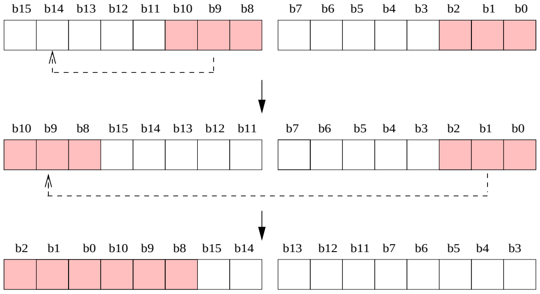

Figure 6. Continuous crosstalk windows using CRS

$$X(n) = 2^m x(n-1) + x(n) \tag{1}$$

We present analytical equations to compute the mean and standard deviation of the compound stream $X(n)$ from the statistics of the original stream $x(n)$ as shown below.

$$\mu_X = (2^m + 1)\mu_x \tag{2}$$

$$\sigma_X = \sigma_x \sqrt{2^{2m} + 1 + 2^{m+1}\rho_x} \tag{3}$$

From equation (1), we observe that concatenation is a linear operation involving a multiplication and an addition operation. Any linear operations on normal distributions results in another normal distribution with a different mean and standard deviation [10]. Thus, the concatenated data stream $X(n)$ has a normal distribution with mean μ_X and standard deviation σ_X. We utilize these parameters in the next step of the algorithm.

To scale the crosstalk estimation solution efficiently for larger bus-widths, the next step of the algorithm involves shifting the disjoint crosstalk windows to the MSB positions so that they are adjacent to one another as shown in Figure 6. This is done using the *Circular Right Shift (CRS)* operation which obeys the following equation:

$$x'(n) = \lfloor x(n)/2 \rfloor + 2^{m-1}[x(n) \ mod \ 2] \tag{4}$$

where $x'(n)$ is obtained by shifting $x(n)$ once to the right in a circular fashion. From equation 4, we observe that the *CRS* operation also preserves the normal properties of the distribution because of its linear nature.

Each *CRS* operation causes the statistics of a data stream to change. Using the above *CRS* equation, we derive analytical equations that relate the mean and standard deviation of the shifted data stream $x'(n)$ to those of the original stream $x(n)$ as shown:

$$\begin{aligned} E[x'(n)] &= E\lfloor x(n)/2 \rfloor + 2^{m-1}E[x(n)\ mod\ 2] \\ \mu_{x'} &= \frac{\mu_x}{2} + 2^{m-1}\eta \end{aligned} \tag{5}$$

$$\begin{aligned} \sigma^2_{x'} &= E[x'^2] - \mu^2_{x'} \\ &= E[\frac{x(n)^2}{4} + 2^{2(m-1)}[x(n)\ mod\ 2]^2 \\ &\quad + 2^{m-1}x(n)[x(n)\ mod\ 2]] \\ &\quad - \frac{\mu_x^2}{4} - 2^{m-1}\eta\mu_x - 2^{2(m-1)}\eta^2 \\ &= \frac{E[x(n)^2]}{4} + 2^{2(m-1)}\eta + 2^{m-1}\mu_x\eta - \frac{\mu_x^2}{4} \\ &\quad - 2^{m-1}\eta\mu_x - 2^{2(m-1)}\eta^2 \\ &= \frac{\sigma_x^2}{4} + 2^{2(m-1)}(\eta - \eta^2) \end{aligned} \tag{6}$$

where $\eta = E[x(n) mod 2]$ is the bit probability of the current LSB in $x(n)$.

This CRS operation is done in two stages to create continuous crosstalk windows as shown in Figure 6. It is first applied to the left half of $X(n)$ to get a modified $2m$-bit value $X_1(n)$ Now, the CRS operation is again performed on $X_1(n)$ to get the transformed value $X_2(n)$. This makes all the bits in a given crosstalk pattern adjacent to each other and located in the *MSB* region of $X_2(n)$. Consequently, dispersed values in the original $X(n)$ map to continuous values in $X_2(n)$.

Consider a 4-bit bus as an example. The compound word $X(n)$ is 8-bits (*b7-b0*) wide. Now, if bits *b6-b4* = ‘000’ and bits *b2-b0* = ‘011’, it corresponds to a transition that causes crosstalk on *b1*. By substituting ‘00’ for bits *b7 b3*, we obtain the value 3 which represents a crosstalk producing transition for *b1*. Using the CRS technique, we now shift the crosstalk pattern to the MSB position of the 8-bit bus. The pattern now reads ‘011000’. By substituting ‘00’ for bits *b1 b0*, we obtain the value 96. Thus, the disjoint values $\{3, 11, 131, 139\}$ in $X(n)$ map to the continuous values $\{96, 97, 98, 99\}$ in $X_2(n)$. As shown in Figure 7, exhaustive enumeration of values *v1, v2, v3*, and *v4* reduces to the integral between limits *l1* and *l2*. This modification reduces the complexity of the algorithm from exponential (in the disjoint case) to linear (in the continuous case) with respect to the bus-width *m*.

3.1. Evaluation of the definite integral using sampling

For large bus-widths, the bounds of the definite integrals are far apart. Since the word-level values are discrete in nature, we propose a *sampling* technique to evaluate the integrals in such cases. This provides a fast and accurate solution.

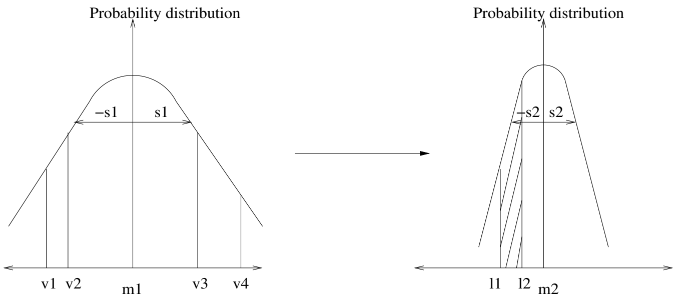

Figure 7. Enumeration to integral transformation

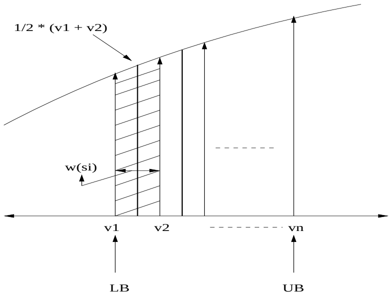

Figure 8. The sampling technique

Each interval corresponding to an integral is sampled a specific number of times. The value obtained by substituting each sample into the probability distribution function is stored in a *look-up table (LUT)*. The interval is now split up into sub-intervals whose width is given by $w_{si} = \frac{ub-lb+1}{n}$ where the integral is bounded by [*lb, ub*] and sampled *n* times. This is illustrated in Figure 8. Thus, each sub-interval is bounded by values which are stored in the LUT. We evaluate the integral for each sub-interval by taking the product of the median for the sub-interval and width w_{si}. If the values stored in the LUT are v_1, v_2, v_3, ..., v_n, the integral I_j for the j*th* sub-interval is $I_j = \frac{v_j+v_{j+1}}{2} * w_{si}$.

The integrals for the remaining sub-intervals are also evaluated in a similar manner. We then compute the integral I_{int} for the interval [*lb, ub*] by summing up the integrals for all the sub-intervals. Thus, $I_{int} = \sum_{j=1}^{n-1} I_j$

A summary of the procedure is provided in the pseudocode. First, the statistics of the compound and shifted words are computed analytically from those of the input stream. Next, the occurrence probability p_{cp} of all crosstalk-producing words is computed. Such words are formed by concatenating each 6-bit crosstalk template instance with the remaining bits which can change from all zeros to all ones. The occurrence probability of these contiguous words

```
Input ← $\mu$, $\sigma$, $\rho$ of m-bit $x(n)$
Compute $\mu_X$, $\sigma_X$ of concatenated 2m-bit $X(n)$
Compute $\mu_{X_2}$, $\sigma_{X_2}$ of 2m-bit $X_2(n)$
LB ← [ctalk_pattern, 00..0]
UB ← [ctalk_pattern, 11..1]
$p_{cp}$ ← Integrate prob.distribution function within bounds [UB, LB] using sampling
Accumulate $p_{cp}$; Repeat for order-1 crosstalk patterns
Output $p_{cp}$
Repeat for all bus-lines of $x(n)$
```

Non-enumerative pseudocode

Bit-width	Data env	ARMA equation
8	SIG 1	x(n)=75γ(n)+200
16	SIG 2	x(n)=250γ(n)+56*10^3
32	SIG 3	x(n)=$10^6$$\gamma$(n)+0.5x(n-1)+5*$10^8$

Table 3. **Data environments**

is obtained by integrating the input distribution function between the extreme bounds. For example, the lower bound *LB* is obtained by concatenating a template instance and zeros i.e. [ctalk_pattern, 00..0]. The sum of p_{cp} over all the template instances gives the crosstalk probability of a bus line.

4. Experimental Results

We compare the proposed non-enumerative statistical crosstalk estimation technique with the stream-based technique. Table 3 shows the data environments modeled using ARMA models [8]. Such ARMA models are often used to represent speech and video signals. The proposed approach is general enough to handle any data environment. The white noise factor γ(n) has a standard normal distribution.

Tables 4 - 6 compare the crosstalk probabilities as computed by the statistical non-enumerative approach against those obtained from the stream-based estimator for bus-widths ranging from 8 bits to 32 bits. It may be noted that although the probability error in line 25 for the 32-bit bus is very high, it differs only in the second decimal place. In practice, it has only 9% chance of crosstalk which we estimate to be negligible. For the lower bits, a correction factor of +20% (due to underestimation) can be incorporated into the estimator since overestimation of the crosstalk probability leads to conservative designs.

Bus line	Stream-based	Stat.estimator	Err(%)
1	0.51	0.51	0.0
2	0.62	0.58	6.4
3	0.63	0.59	6.3
4	0.65	0.58	10.7
5	0.62	0.57	8.0
6	0.63	0.57	9.5
7	0.51	0.69	35.2
8	0.49	0.49	0.0

Average error: 9.5%

Table 4. **Crosstalk probability for 8-bit bus**

Bus line	Stream-based	Stat.estimator	Err(%)
1	0.53	0.53	0.0
2	0.61	0.70	14.7
3	0.66	0.70	6.1
4	0.63	0.69	9.5
5	0.63	0.69	9.5
6	0.62	0.68	9.7
7	0.62	0.68	9.7
8	0.65	0.68	4.6
9	0.60	0.75	25.0
10	0.51	0.54	5.9
11	0.46	0.46	0.0
12	0.15	0.15	0.0
13	0.00	0.00	0.0
14	0.00	0.00	0.0
15	0.00	0.00	0.0
16	0.00	0.00	0.0

Average error: 5.9%

Table 5. **Crosstalk probability for 16-bit bus**

The runtimes for the proposed non-enumerative technique are compared to those of the stream-based estimator for different bus-widths in Table 7. It is to be noted that with increase in the data stream length, the difference in the runtimes becomes even more significant.

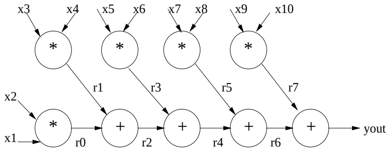

Figure 9. **FIR filter**

Increasing the number of samples increases the accuracy as well as the runtimes. It is observed that beyond a certain number of samples, the increase in accuracy is insignificant as compared to the run times. Hence, the number of samples to be used for each interval should be discreetly selected. Table 8 shows the effects of the number of samples on the runtimes.

We use the non-enumerative statistical procedure to estimate the crosstalk affecting the edges of a finite-impulse response (FIR) filter [8] shown in Figure 9. The average estimation error for each edge, computed with respect to the stream-based estimator, is shown in Table 9.

Bus line	Stream-based	Stat.estimator	Err(%)
1	0.31	0.27	12.9
2	0.50	0.52	4.0
3	0.59	0.46	22.0
4	0.65	0.48	26.2
5	0.62	0.49	20.9
6	0.61	0.49	19.7
7	0.64	0.50	21.8
8	0.64	0.50	21.8
9	0.64	0.50	21.8
10	0.61	0.50	18.0
11	0.62	0.47	24.2
12	0.62	0.49	20.9
13	0.61	0.49	19.7
14	0.62	0.50	19.3
15	0.59	0.50	15.2
16	0.63	0.50	20.6
17	0.63	0.50	20.6
18	0.64	0.50	21.9
19	0.60	0.50	16.7
20	0.64	0.50	21.9
21	0.57	0.50	12.3
22	0.53	0.44	17.0
23	0.35	0.47	34.3
24	0.13	0.16	23.0
25	0.09	0.01	88.9
26	0.00	0.00	0.0
27	0.00	0.00	0.0
28	0.00	0.00	0.0
29	0.00	0.00	0.0
30	0.00	0.00	0.0
31	0.00	0.00	0.0
32	0.00	0.00	0.0

Average error: 17.6%

Table 6. **Crosstalk probability for 32-bit bus**

Bus-width	Length	Stream-based	Stat estimator
8b	1000	22.4s	0.4s
16b	1000	24.3s	1.2s
32b	1000	33.0s	0.5s

Table 7. **Str-based vs Stat estimator run times**

Bus-width	s=10	s=5000	s=50000
Runtime(s)	0.2	2.6	10.0
Error(%)	19.4	17.6	17.6

Table 8. **Effect of samples for 32b bus**

5. Conclusions

We presented a non-enumerative statistical technique to evaluate bit-level crosstalk probability within a system bus from the word-level statistics of input data. We introduced a sampling technique to quickly evaluate definite integrals of discrete random variables during the estimation process. The technique reduces the estimation complexity from exponential to linear with respect to the bus-width. The following work will integrate a floorplanner and a global router with this technique in order to predict inter-bus crosstalk effects.

References

[1] A. Vittal, M. M. Sadowska. “Crosstalk Reduction for VLSI”. *ITCAD*, 16(3):290–298, March 1997.

[2] J. H. Satyanarayana, and K. K. Parhi. “Theoretical Analysis of Word-Level Switching Activity in the Presence of Glitching and Correlation”. *IEEE Tran. on VLSI*, April 2000, 8(2).

[3] K. T. Tang and E. G. Friedman. “Peak Crosstalk Noise Estimation in CMOS VLSI Circuits”. In *Proceedings of ICECS*, pages 1539–1542, 1999.

[4] L. H. Chen and M. M. Sadowska. “Closed-Form Crosstalk Noise Metrics for Physical Design Applications”. In *DATE*, pages 1–8, 2002.

[5] M. Becer and I. N. Hajj. “An Analytical Model for Delay and Crosstalk Estimation with Application to Decoupling”. In *Proceedings of ISQED*, pages 51–57, 2000.

[6] M. Kuhlmann, S. Sapatnekar. “Exact and Efficient Crosstalk Estimation”. *ITCAD*, 20(7):858–866, July 2001.

[7] S. Gupta and S. Katkoori. “Intra Bus Crosstalk Estimation Using Word-Level Statistics”. In *17th International Conference on VLSI Design*, 2004.

[8] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj. “Analytical Estimation of Signal Transition Activity from Word-Level Statistics”. *ITCAD*, 16(7):718–733, July 1997.

[9] T. K. Tien, S. C. Chang, and T. K. Tsai. “Crosstalk Alleviation for Dynamic PLAs”. *ITCAD*, 21(12):1416–1424, December 2002.

[10] V.K. Rohatgi. *“An Introduction to Probability and Statistics”*. Wiley, 2001.

Edge_id	Avg. err(%)	Edge_id	Avg. err(%)
x1	8.7	yout	15.6
x2	8.7	r0	8.7
x3	9.0	r1	11.4
x4	8.8	r2	15.5
x5	9.3	r3	11.6
x6	8.7	r4	21.0
x7	9.3	r5	9.2
x8	9.0	r6	30.4
x9	8.7	r7	12.0
x10	9.4	-	-

Table 9. **Avg. crosstalk estimation error - FIR**