Global Wire Bus Configuration with Minimum Delay Uncertainty [*]

Li-Da Huang[1], Hung-Ming Chen[1], D.F. Wong[2]
[1]Department of Computer Sciences, University of Texas at Austin, Austin, TX 78712
[2]ECE Department, University of Illinois at Urbana-Champaign, Urbana, IL 61801 U.S.A.

Abstract

The gap between the advances and the utilization of the deep submicron (DSM) technology is increasing as the new generation of technology is introduced faster than ever. Signal integrity is one of the most important issues in overcoming this gap. With the increasing coupling capacitance between the high aspect ratio wires, the delay uncertainty is unpredictable in the current design flow. We present an algorithm to generate the global wire bus configuration with minimum delay uncertainty under timing constraints. The timing window information from the timing budget(or specified in IPs) is integrated with the modern accurate crosstalk noise models in the proposed algorithm. HSPICE simulations show that the algorithm is very effective and efficient when compared to the buffer insertion scheme with minimum delay. The standard deviation of the delay obtained from the Monte-Carlo simulation is improved up to 73%. This global wire bus configuration can be adopted in early wire planning to improve the timing closure problem and increase the accuracy of the timing budget.

1 Introduction

The advancement of the integrated circuit (IC) technology has been following the Moore's Law closely in the past few decades, and introduced a huge value in the electronic market. Based on the newly introduced/developed technologies in the leading foundry [21], the rate of scaling is even faster. New generation of technology is introduced every two years. This will lead to billions of transistors integrated on a single chip performing different tasks designed in IPs. However, without carefully designed and verified interconnects to connect these high speed devices, the advantage gained from DSM technologies can not be converted to profitable products. Signal integrity is one of the most important issues in the gap between the advances and the utilization of the modern technologies. To eliminate this gap, the interconnects need to be planned and verified to control the delay uncertainty and to avoid logic hazards. Hence the design flow is evolving into interconnect-centric methodology [8], or introducing a regular interconnect structure [16] to have a more predictable noise behavior.

Among the signal integrity problems, crosstalk noise is a particular concern because it can lead to delay and logic hazards. Noise analysis and avoidance techniques are necessary to control the delay uncertainty on signals and to reduce the faulty triggers because of the increasing coupling capacitance and the reduced noise margin [17]. In the current design methodology, most of the analysis is applied in the late physical design phase because the detailed layout data can only be gathered at that phase. If there is any timing violation, the design has to be modified. It is not rare that the iterations in the design flow can not generate a reliable design data base. This problem can be solved by providing a predictable routing source such as global wire buses.

Buffer insertion is an effective way to improve the delay on a long wire and to control the crosstalk effect [2, 1], but only the delay or switching factor is considered in [14]. The timing window and crosstalk noise are discussed in [23]. Estimation of the worst case crosstalk noise by aligning the aggressor signals is presented in [11, 6]. These methods usually need iterations to generate the results, and are adopted in static timing analysis. [3, 7] propose a more accurate crosstalk noise model that allows more complicated routing patterns. Recently, there are experimental chips with built-in measurement circuits to verify the accuracy of these proposed crosstalk models [4, 5].

In this paper, we will minimize the delay uncertainty by inserting buffers on a global wire bus under timing constraints. The information about timing windows, peak value of the crosstalk noise, and buffer locations are embedded into a graph. The best bus configuration is represented as the shortest path in that graph. The paper is organized as follows. Section 2 will review the crosstalk induced delay, and describe the problem formulation. The algorithm of global wire bus configuration with minimum delay uncertainty will be presented in section 3. The experimental results are included and discussed in section 4. Section 5 will conclude the paper.

2 Crosstalk Induced Delay and Problem Formulation

Due to the increase in coupling capacitance of overall net capacitance, the delay of a signal is now a dynamic phenomenon that depends on the activity of nearby signals. The aggressor signal generates crosstalk noise waveform on the victim wire. Since the noise waveform is superimposed on the waveform of the victim signal, the resultant waveform will reach the threshold of the repeater in different times depending on the relative switching time of the aggressor and victim signals. Fig. 1 shows the delay induced by the crosstalk noise waveform. This crosstalk induced delay is a complicated function of the peak value of the crosstalk noise, and the relative timing of the aggressor and victim signals. Hence it is hard to predict and represents as the delay uncertainty [15, 13, 10] in the timing of whole circuit. Global wire bus, which has been provided as a predictable routing sources in VLSI design, is our focus in this paper.

[*]This work was partially supported by the National Science Foundation under grant CCR-0244236

We show the problem as follows.

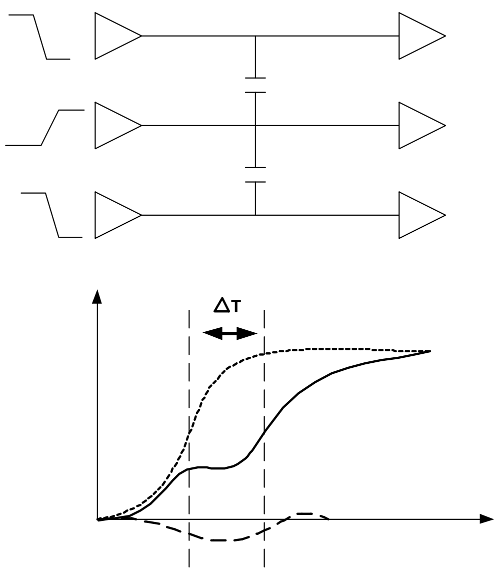


Figure 1: The crosstalk noise waveform is superimposed on the victim waveform. The resultant waveform will be delayed to toggle the threshold of the buffer. The timing difference between the initial waveform and the resultant waveform is the crosstalk induced delay.

Problem 2.1 *Given a bus containing n wires with different timing constraints and timing windows of input signals, insert buffers on each wire to meet the timing constraints with minimum delay uncertainty.*

We assume all wires in the same bus will be routed in the same conducting layer. Therefore, with the timing window provided or specified by the local blocks, each wire can have the timing information of its neighbors on both sides. Without considering the crosstalk induced delay, the number of buffers on each wire is within a range for a given timing constraint. Once the number of buffers on each wire has been decided, the configuration of the bus will be fixed. However, it is not obvious to find out which bus configuration has the minimum delay uncertainty because of the crosstalk induced delay. Intuitively, the buffers can be staggered to relieve the crosstalk by reducing the overlapping length of two parallel wires. This will increase the number of buffers, however it is unnecessary to avoid the crosstalk between two wires if the timing windows are not overlapping. In addition, the same number of buffers can not be inserted into every wire if the timing constraint is different on each wire. Staggered buffer insertion can not both minimize the delay uncertainty and satisfy all the timing constraints in a given bus. Furthermore, not all overlapping of two parallel wires will increase the crosstalk. In [22], a sensitive wire can reduce the crosstalk noise by having a slow switching wire as a neighbor. The slow switching wire becomes a protector instead of an aggressor. Therefore, the buffer insertion in a given bus with minimum delay uncertainty is a nontrivial problem.

2.1 CrossTalk Noise Signature CTNS

To evaluate the crosstalk effect within different timing windows, the crosstalk noise information is presented as a rectangular window on the time axis with V_{peak} height, which can be accurately computed as in [20]. This aggressor introduces the crosstalk noise with V_{peak} on the victim wire, which is represented as Fig. 2. Note that this representation is not the waveform of the crosstalk noise, instead it contains the information about timing of the aggressor and the peak value of the crosstalk noise. We call this representation the crosstalk noise signature($CTNS$). Fig. 3 has a victim wire with two wire segments on the aggressor. Each aggressor wire segment introduces a $CTNS$. If the width of the aggressor timing window is greater than the shift time(i.e., delay) between the two wire segments, the two $CTNS$s will be overlapping on the time axis. We can combine these two $CTNS$s into a bigger $CTNS$. Since the two wire segments are on the same wire, the occurrence of the crosstalk noise on the victim wire will not happen at the same time. The peak value of the crosstalk noise is the maximum of the two V_{peak} value within the overlapping timing window of the two $CTNS$s. Therefore, the combined $CTNS$ is actually the contour of the two overlapping $CTNS$s, as in Fig. 4.

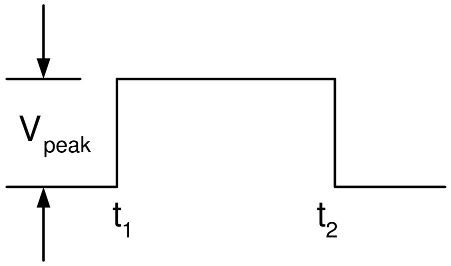


Figure 2: The crosstalk noise signature($CTNS$) on a victim wire segment is defined as a pulse with V_{peak} on time (t_1, t_2) from one aggressor wire segment. (t_1, t_2) is the timing window of the aggressor signal. V_{peak} is the peak value of the crosstalk spike on the victim wire.

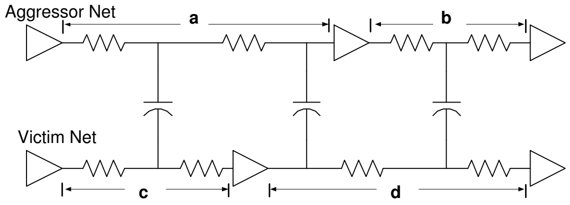


Figure 3: Two wire segments on both aggressor wire and victim wire. The wire segment d on the victim wire is overlapping with two aggressor wire segments a and b.

2.2 Delay Uncertainty Index DUI

After constructing the $CTNS$s of all wire segments on the victim wire as in Fig. 4, the timing window of the input signal on the victim wire is calculated for each wire segments on the victim wire. Since the arrival time of the signal in the later buffer stage equals to the arrival time of the signal in the previous buffer stage plus the shift time [12], the timing window of each wire segment on the victim wire is

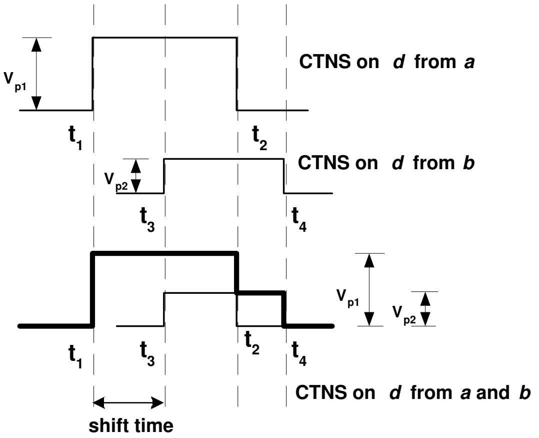


Figure 4: Signal on each aggressor wire segment introduces a $CTNS$ on the victim wire segment d. Because of the delay on the aggressor wire, the shift time is the timing difference between both $CTNS$s. The combined $CTNS$ on the second victim wire segment is the contour of both $CTNS$s overlapping with each other.

actually the shifted timing window of the input signal. The shifted timing window on each wire segment forms the mask of the $CTNS$ on the corresponding wire segment. Only the portion of the $CTNS$ in the mask will introduce crosstalk induced delay. In order to simplify the description, we assume the signal on the aggressor is switching in the opposite direction of the signal on the victim wire. If the mask is overlapping with the $CTNS$, it is possible that the crosstalk will introduce delay. The area, A, of the $CTNS$ within the mask is proportional to the delay uncertainty, since the crosstalk noise is within the timing window of the signal on the victim wire as shown in Fig. 6. The masks of the later wire segments on the victim wire will have a bigger coverage on the $CTNS$ than the masks on the previous wire segments, as in Fig. 5. Assuming the arrival time is equally distributed in the timing window, the area can be weighted with the probability of both the aggressor and victim signals that are in the mask range. The delay uncertainty index (DUI) of a victim wire segment is defined as:

$$I = A \cdot \frac{t_A}{t_{mask}} \cdot \frac{t_A}{t_{CTNS}} \qquad (1)$$

The higher value of I means it is more possible that the crosstalk noise will introduce delay. The DUI of the victim wire with m wire segments is the sum of the DUIs on all wire segments.

$$I_{net} = \sum_{i=0}^{m-1} I_i \qquad (2)$$

In the next section, we will propose an algorithm to minimize the DUI.

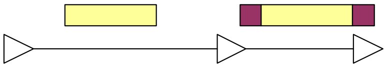


Figure 5: The mask (timing window) of the later wire segment on the victim wire has a bigger coverage than the mask on the previous wire segment.

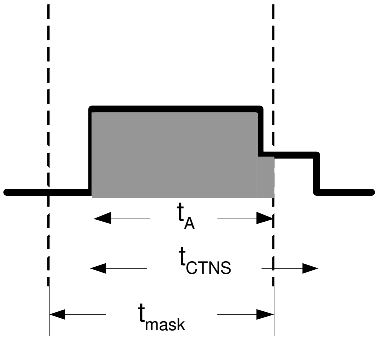


Figure 6: The combined $CTNS$ on a victim wire segment and the mask. The area A in shadow is part of delay uncertainty model.

3 The Algorithm $BCMDUI$: Bus Configuration with Minimum Delay Uncertainty Index

To simplify the presentation, we assume only one kind of buffer and wire size will be used in minimizing the delay uncertainty of a given global wire bus. This algorithm can be easily extended to include any buffer library and wire library. The bus configuration is defined as the locations of the inserted buffers. This algorithm will find the bus configuration with the minimum DUI under given timing constraints. There are two phases in the proposed algorithm. The first phase is to find out the number of buffers equally distributed on each wire of the bus. The second phase is to search for the best offset of the buffer locations given from the first phase to further decrease the DUI.

Because each wire has different timing constraints, the ranges of the buffer numbers are different for each wire. Given a timing constraint on an unbuffered wire, there exists several buffer insertion schemes that meet the timing constraint. Fig. 7 is the simulation result for the equally distributed buffer insertions on a wire with different lengths. We can have $\{2, \cdots, 5\}$ buffers on the $20mm$ wire with the delay less than $1.8ns$. In [1], with only one buffer and one width of wire in the buffer and wire libraries, the equally distributed buffer insertion schemes give the minimum Elmore delay with the given buffer numbers. We define the specifications of a global wire bus as follows:

1. n is the number of the wires on the bus B. w_i is the i^{th} wire of B, $0 \leq i < n$.
2. DT_i is the delay timing constraint on w_i.
3. tr_i^0 is the transition time of the input signal on w_i.
4. $TW_i = \{t_{af}, t_{al}\}$ is the timing window for wire i with t_{af} and t_{al} being the first arrival time and last arrival time of input signal on w_i respectively.

Given wire w_i with the delay timing constraint DT_i, the feasible buffer number range($FBNR$) R_i for w_i is $\{r_{min}, r_{max}\}$ where $r_{min} \leq r_{max}$, both are non-negative integers. Since the buffers are equally distributed on w_i, the buffer locations are well defined for a given $r \in R_i$. There are $r+1$ wire segments on w_i separated by r buffers. The transition time tr_i^j and shift-time st_i^j of j^{th} segment on w_i with r buffers can be calculated as in [12]. Assuming w_i and w_{i+1}

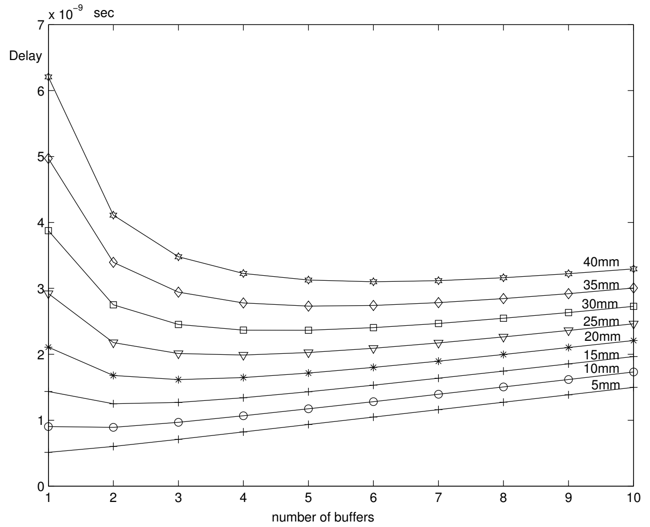


Figure 7: HSPICE simulation results for wires with lengths ranged from $5mm$ to $40mm$. The buffers are equally distributed on the wires. Given the timing constraints on a fixed length wire, the feasible buffer number range($FBNR$) is defined as the numbers of the inserted buffers without violating the timing constraint.

are next to each other with R_i and R_{i+1} as the $FBNR$s respectively, the $I^{(i,i+1)}_{(x,y)}$ (defined as the DUI of the w_i as the aggressor, w_{i+1} as the victim for $x \in R_i$ and $y \in R_{i+1}$) can be calculated according to section 2. The $I^{(i+1,i)}_{(y,x)}$ for w_{i+1} as the aggressor and w_i as the victim can be obtained likewise. Then $I(i, i+1)_{(x,y)}$, which is the DUI for w_i with x buffers and w_{i+1} with y buffers, can be defined as the linear combination of $I^{(i,i+1)}_{(x,y)}$ and $I^{(i+1,i)}_{(y,x)}$:

$$I(i, i+1)_{(x,y)} = \alpha \cdot I^{(i,i+1)}_{(x,y)} + \beta \cdot I^{(i+1,i)}_{(y,x)} \qquad (3)$$

where $\alpha + \beta = 1$. For example, $I(1, 2)_{(3,4)}$ is the DUI for w_1 with 3 buffers and w_2 with 4 buffers. It is the linear combination of $I^{(1,2)}_{(3,4)}$ and $I^{(2,1)}_{(4,3)}$ with w_1 and w_2 as the aggressors respectively. By adjusting the weights, some critical wires can have more controllability on delay uncertainty.

Given global wire bus B with n wires ordered as $w_0, \cdots, w_{n-1}$, we first construct the graph $G(V, E)$ as:

1. $V = \{s, t\} \cup \{v^i_r\}$, where s, t represent the source node and sink node respectively. v^i_r is the node added for each wire $w_i \in B$ with each integer number $r \in R_i$, where R_i is the FBNR for w_i.

2. $E = \{(s, v^0_x)\} \cup \{v^i_y, v^{i+1}_z\} \cup \{v^{n-1}_w, t)\}$, for each $i \in \{0, \cdots, n-1\}$, $x \in R_0, y \in R_i, z \in R_{i+1}, w \in R_{n-1}$

3. Edge cost: $C(s, v^0_x) = 0, C(v^n_w, t) = 0$, for other edges in $\{v^i_y, v^{i+1}_z\}$, each cost is assigned as the $I(i, i+1)_{y,z}$.

Fig. 8 is the simple example that illustrates the graph constructed for a bus containing 3 wires with $R_0 = \{3, 4, 5\}$, $R_1 = \{3, 4, 5, 6\}$, and $R_2 = \{3, 4\}$. We apply the shortest path algorithm from s to t on G. The nodes on the shortest path are the bus configuration of equally distributed buffers with minimum DUI.

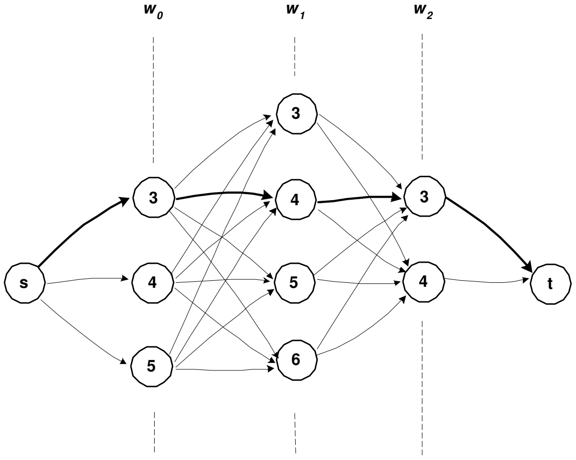


Figure 8: The graph for bus with 3 wires. There are three options({3, 4, 5}) of the number of buffers inserted in w_0, four options({3, 4, 5, 6}) for w_1, and two options({3, 4}) for w_2. The highlighted path is the shortest path representing the bus configuration with minimum delay uncertainty index with equally distributed buffers.

Based on the result of the first phase, the positions of the buffers on each wire are allowed to offset its position within the independent feasible region(IFR) defined in [19] or any other reasonable region without violating the timing constraint [9, 18]. Assuming each buffer can be relocated to the middle of the left region, right region, or its original position, there are three possible positions for each buffer in the first phase output. Fig. 9 is an example of buffer locations in the same graph with offsets on w_0. We encode the buffer locations within the region as 0 for offset to the left, 1 for original position, and 2 for offset to the right. The configuration of the buffered path on each wire can be encoded into a number χ as follows:

$$\chi = \sum_{i=0}^{m-1} code(i) \cdot 3^i \qquad (4)$$

$0 \leq \chi \leq 3^m - 1$, where m is the number of buffers obtained from the first phase, i is the i^{th} buffer on that wire, and $code(i) \in \{0, 1, 2\}$ is the offset code of buffer i. We then construct another graph $G_F(V_F, E_F)$ as the same procedure when constructing $G(V, E)$ except that the node v^i_χ now represents the χ^{th} configuration on w_i.

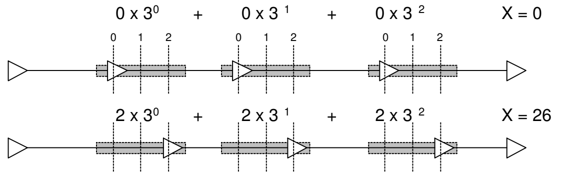

0 x 3^0 + 0 x 3 ^{1} + 0 x 3 ^{2}
X = 0
0 1 2
0 1 2
0 1 2
2 x 3^0 + 2 x 3 ^{1} + 2 x 3 ^{2}
X = 26

Figure 9: The buffer position with offset in the feasible region is encoded as an integer. In this example, three buffers are inserted into the wire. The different buffer positions are encoded as integers from 0 to 26.

Fig. 10 illustrates the $G_F(V_F, E_F)$ constructed from the result of Fig. 8. The best offsets of the buffer locations will be obtained from shortest path from s to t on $G_F(V_F, E_F)$ as in Fig. 10. The resultant buffer positions are shown in Fig. 11. The summary of the algorithm $BCMDUI$ is in Fig. 12.

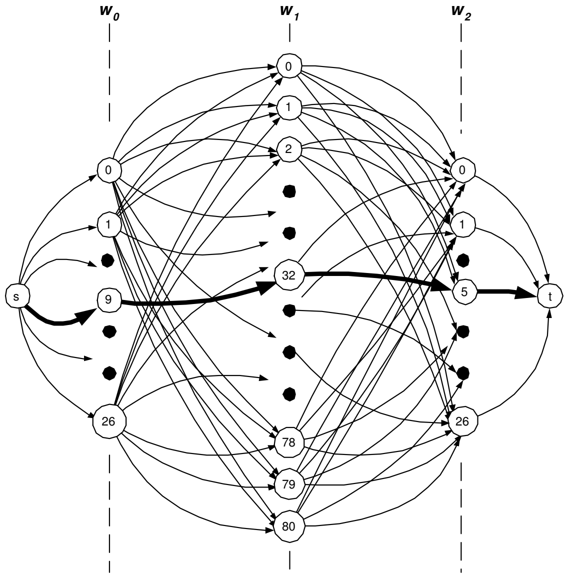
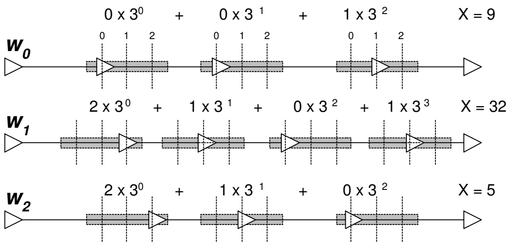


Figure 10: The graph for finding the best offset of the buffer position. Each node represents an offset code, χ, defined in Fig. 9. The shortest path corresponds the best bus configuration with offsets based on Fig. 8.



$0 \times 3^0 + 0 \times 3^1 + 1 \times 3^2$ X = 9
0 1 2
w_0
$2 \times 3^0 + 1 \times 3^1 + 0 \times 3^2 + 1 \times 3^3$ X = 32
w_1
$2 \times 3^0 + 1 \times 3^1 + 0 \times 3^2$ X = 5
w_2

Figure 11: The resultant buffer positions from the solution of Fig. 10.

4 Experimental Results and Discussion

The algorithm $BCMDUI$ is implemented in C++ programming language, and tested on PentiumIII(750MHZ, 256 MB DRAM) machines. The delay uncertainty is obtained by the Monte-Carlo simulations in HSPICE on Sun Sparc Ultra10(440MHZ, 256 MB DRAM). The technology parameters in Table 1 are from $TSMC$ 180nm process.

The non-inverted buffer consists of two cascaded inverters that the W/L ratio of the second stage is five times the first stage. These SPICE simulations in Table 2 show that the delay uncertainties were decreased a lot when compared to the bus configuration with minimum Elmore delay on each wire. The standard deviations of the delays are improved from 39% up to 73%. Without considering the switching activity, minimum Elmore delay buffer insertion scheme can not guarantee the minimum delay because of the crosstalk induced delay. The average delay on the minimum Elmore delay buffer insertion is larger than the result from $BCMDUI$ in each case. Table 3 compares the results between the staggered buffer insertion and $BCMDUI$. Since the delay constraints for wires on the bus are not fully considered, the buffers are inserted more than necessary in the staggered buffer insertion scheme. The average delay is increased as compared to $BCMDUI$. In addition, even though the standard deviation of the delay has been reduced by staggered buffer insertion, the reduction is still limited because of not considering the timing windows of the signals.

Algorithm $BCMDUI(B, DT)$

```
1. Calculate the $FBNR$ of each wire on $B$ with given DT.
2. Construct graph $G(V, E)$.
3. Calculate the $DUI$s between each wire on $B$, and assign the internal edge cost as the $DUI$s
4. Obtain the shortest path from $s$ to $t$ on $G$.
5. According the output from step 4, construct $G_F(V_F, E_F)$
6. Obtain the shortest path from $s$ to $t$ on $G_F$.
7. Output the nodes on the shortest path from step 6, which represent the bus configuration with the minimum $DUI$.
```

Figure 12: Summary of Algorithm $BCMDUI$

name	description	value
width	the M5 line width(um)	0.28
space	the space between two M5 lines	0.28
Cc	the coupling capacitance(fF/um)	926
Ca	the area capacitance(fF/um)	118
r	sheet resistance(Ω/sq)	0.078
Cb	input capacitance of buffer(fF)	21
Rb	output resistance of buffer(Ω)	70

Table 1: Technology parameters

5 Conclusions

The DUI proposed in this paper is the information containing the peak voltage of crosstalk noise and the timing windows of both victim and aggressor. Based on DUI, the graph representing the bus configurations embedded with DUI is constructed and the shortest path gives the bus configuration with minimum DUI. The offsets of the buffer locations are considered in the second phase of $BCMDUI$ to further reduce the delay uncertainties. Experimental results show the delay uncertainties were decreased significantly as compared to bus with minimum delay.

References

[1] C. Alpert and A. Devgan. "Wire Segmenting for Improved Buffer Insertion". In *Proceedings IEEE/ACM Design Automation Conference*, pages 588–593, 1997.

[2] C.J. Alpert, A. Devgan, and S.T. Quay. "Buffer Insertion for Noise and Delay Optimization". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(11):1633–1645, November 1999.

test	len	delay	min Elmore delay		BCMDUI		impv(%)
	(mm)	cnstr(ns)	avg(ns)	std(ps)	avg(ns)	std(ps)	
$T1$	20	1.95	1.91	189	1.80	136	39%
$T2$	20	2.5	1.91	189	2.05	121	56%
$T3$	40	3.5	3.48	642	3.22	427	50%
$T4$	40	4.0	3.48	642	3.1	370	73%

Table 2: Results of Monte-Carlo simulations for various global wire buses. The BCMDUI results are compared to the buffer insertion that minimizes Elmore delay without considering crosstalk.

test	len	delay	Staggered buf inser.		BCMDUI		impv(%)
	(mm)	cnstr(ns)	avg(ns)	std(ps)	avg(ns)	std(ps)	
$T5$	20	1.95	2.37	144	1.80	136	6%
$T6$	20	2.5	2.37	144	2.05	121	19%
$T7$	40	3.5	3.63	437	3.22	427	2.3%
$T8$	40	4.0	3.63	437	3.1	370	18%

Table 3: The staggered buffer insertion can effectively reduce the standard deviation of the delay, but the mean value of the delay might violate the timing constraint.

[3] M.R. Becer, D. Blaauw, V. Zolotov, R. Panda, and I.N. Hajj. "Analysis of Noise Avoidance Techniques in DSM Interconnects using a Complete Crosstalk Noise Model". In *Proceedings Design, Automation and Test in Europe*, pages 456–463, 2002.

[4] S. Delmas Bendhia, F. Caignet, and E. Sicard. "On Chip Crosstalk Characterization on Deep Submicron Buses". In *International Caracas Conference on Devices, Circuits and Systems*, pages C70/1–C70/5, 2000.

[5] F. Caignet, S. Delmas-Bendhia, and E. Sicard. "The Challenge of Signal Integrity in Deep-Submicrometer CMOS Technology". *Proceedings of the IEEE*, 89(4):556–573, April 2001.

[6] L.H. Chen and M. Marek-Sadowska. "Aggressor Alignment for Worst-Case Crosstalk Noise". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 20(5):612–621, May 2001.

[7] L.H. Chen and M. Marek-Sadowska. "Closed-Form Crosstalk Noise Metrics for Physical Design Applications". In *Proceedings Design, Automation and Test in Europe*, pages 812–819, 2002.

[8] J. Cong. "An Interconnect-Centric Design Flow for Nanometer Technologies". *Proceedings of the IEEE*, 89(4):505–528, April 2001.

[9] J. Cong, T. Kong, and Z. Pan. "Buffer Block Planning for Interconnect Planning and Prediction". *IEEE Transactions on Very Large Scale Integration Systems*, 9(6):929–937, December 2001.

[10] J.A. Davis and J.D. Meindl. "Length, Scaling, and Material Dependence of Crosstalk between Distributed RC Interconnects". In *IEEE International Interconnect Technology Conference*, pages 227–229, 1999.

[11] P.D. Gross, R. Arunachalam, K. Rajagopal, and L.T. Pileggi. "Determination of Worst-Case Aggressor Alignment for Delay Calculation". In *Proceedings IEEE/ACM International Conference on Computer-Aided Design*, pages 212–219, 1998.

[12] L. Huang, M. Lai, Y. Gao, and D.F. Wong. "Maze Routing with Buffer Insertion Under Transition Time Constraints". In *Proceedings Design, Automation and Test in Europe*, pages 702–707, 2002.

[13] I. H.-R. Jiang, S.-R. Pan, Y.-W. Chang, and J.-Y. Jou. "Optimal Reliable Crosstalk-Driven Interconnect Optimization". In *Proceedings International Symposium on Physical Design*, pages 128–133, 2000.

[14] A.B. Kahng, S. Muddu, and E. Sarto. "On Switch Factor Based Analysis of Coupled RC Interconnects". In *Proceedings IEEE/ACM Design Automation Conference*, pages 79–84, 2000.

[15] A.B. Kahng, S. Muddu, and D. Vidhani. "Noise and Delay Uncertainty Studies for Coupled RC Interconnects". In *ASIC/SOC Conference*, pages 3–8, 1999.

[16] S.P. Khatri, A. Mehrotra, R.K. Brayton, A. Sangiovanni-Vincentelli, and R.H.J.M. Otten. "A Novel VLSI Layout Fabric for Deep Sub-Micron Applications". In *Proceedings IEEE/ACM Design Automation Conference*, pages 491–496, 1999.

[17] D.A. Kirkpatrick and A.L. Sangiovanni-Vincentelli. "Techniques for Crosstalk Avoidance in the Physical Design of High-Performance Digital Systems". In *Proceedings IEEE/ACM International Conference on Computer-Aided Design*, pages 616–619, 1994.

[18] P. Sarkar and C.-K. Koh. "Repeater Block Planning under Simultaneous Delay and Transition Time Constraints". In *Proceedings Design, Automation and Test in Europe*, pages 540–544, 2001.

[19] P. Sarkar, V. Sundararaman, and C.-K. Koh. "Routability-Driven Repeater Block Planning for Interconnect-Centric Floorplanning". In *Proceedings International Symposium on Physical Design*, pages 186–191, 2000.

[20] D. Sylvester and C. Hu. "Analytical Modeling and Characterization of Deep-Submicrometer Interconnect". *Proceedings of the IEEE*, 89(5):634–664, May 2001.

[21] TSMC. In *U.S. Technology Symposium*, 2002.

[22] A. Vittal and M. Marek-Sadowska. "Crosstalk Reduction for VLSI". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 16(3):290–298, March 1997.

[23] T. Xiao and M. Marek-Sadowska. "Efficient Delay Calculation in Presence of Crosstalk". In *IEEE International Symposium on Quality Electronic Design*, pages 491–497, 2000.