ShieldUS: A Novel Design of Dynamic Shielding for Eliminating 3D TSV Crosstalk Coupling Noise

Yuan-Ying Chang*, Yoshi Shih-Chieh Huang*, Vijaykrishnan Narayanan+, and Chung-Ta King*

*Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
{elmo,yoshi,king}@cs.nthu.edu.tw
+Department of Computer Science and Engineering, Pennsylvania State University, University Park, USA
vijay@cse.psu.edu

Abstract— 3D IC is a promising technology to meet the demands of high throughput, high scalability, and low power consumption for future generation integrated circuits. One way to implement the 3D IC is to interconnect layers of two-dimensional (2D) IC with Through-Silicon Via (TSV), which shortens the signal lengths. Unfortunately, while TSVs are bundled together as a cluster, the crosstalk coupling noise may lead to transmission errors. As a result, the working frequency of TSVs has to be lowered to avoid the errors, leading to narrower bandwidth that TSVs can provide. In this paper, we first derive the crosstalk noise model from the perspective of 3D chip and then propose ShieldUS, a runtime data-to-TSVs remapping strategy. With ShieldUS, the transition patterns of data over TSVs are observed at runtime, and relatively stable bits will be mapped to the TSVs which act as shields to protect the other bits which have more fluctuations. We evaluate the performance of ShieldUS with address lines from real benchmark traces and data lines of different similarities. The results show that ShieldUS is accurate and flexible. We further study dynamic shielding and our design of Interval Equilibration Unit (IEU) can intelligently select suitable parameters for dynamic shielding, which makes dynamic shielding practical and does not need to predefine parameters. This also improves the practicability of ShieldUS.

I. INTRODUCTION

Three-dimensional Integrated Circuits (3D IC) is a promising technology which meets the demands of high throughput, scalability, and power consumption for future generation of ICs. Among several possible 3D IC implementations, die stacking is a feasible way to build 3D IC. Die stacking stacks several dies vertically and uses Through-Silicon Vias (TSV) to interconnect the different layers. Since the lengths of TSVs (in nano-scale) are quite shorter than traditional links, TSVs can work in extremely high frequency. Moreover, while integrating TSVs into 3D IC, TSVs are usually bundled together rather than used in isolation [11]. By using several bundles of TSVs, it is possible to provide high memory throughput, which is especially important for recent scalable memory design, such as Wide I/O DRAM [4].

Unfortunately, bundling TSVs causes the crosstalk coupling noise effect and may lower the performances of TSVs. Crosstalk coupling noise is a fundamental problem in VLSI design and becomes even more complicated in 3D chip. Previous works solve the crosstalk effect from different perspectives. Works such as active shielding [3], shields insertion [12, 9], and wire spacing [1], consider the problem from physical design. There are also works that focus on higher component-based considerations and clock frequency tuning [7]. Theoretical studies, i.e., coding theory, are also well researched [15, 10, 2]. 3D design using TSVs causes new issues, which have been studied extensively in recent years.

Nevertheless, for the TSV-to-TSV coupling noise problem, there is still large room to explore. Let us consider the difference of the crosstalk problem between 2D chip and 3D chip with TSVs. In the 2D case, wires are placed on a plane and the crosstalk is usually caused by the two immediate neighbors (i.e., left wire and right wire to the middle one). In the 3D case, TSVs are bundled and a TSV is surrounded by other TSVs. Consequently, a TSV needs to consider all the immediate neighbors from all directions.

In this paper, we first introduce a 3D crosstalk decoupling noise model, and then propose *ShieldUS*, a novel runtime data-to-TSVs remapping mechanism for shielding the transmission while considering the 3D crosstalk model. The basic idea of dynamic shielding is to observe the transmitted data through TSVs for a period of time to rank the similarities of the data, and select the least changing bits to the TSVs as shields. The remaining bits with more frequent changes can be safely transmitted via the normal TSVs. With our derived 3D model, the data-to-TSVs can be intelligent mapped for efficient transmissions. After applying ShieldUS, the performance improvement is up to 12%.

The contributions of this paper are as follows:

- A 3D crosstalk coupling noise model is proposed and applied in this paper.
- A novel method, ShieldUS, is proposed, which dynamically maps data-to-TSVs to minimize the crosstalk classes.
- Interval Equilibration Unit (IEU) is proposed for automatically finding an interval to observe transitions for better mapping at runtime.

The rest of paper is organized as follows. Section II provides a formal description of our targeted problem. In Section III,

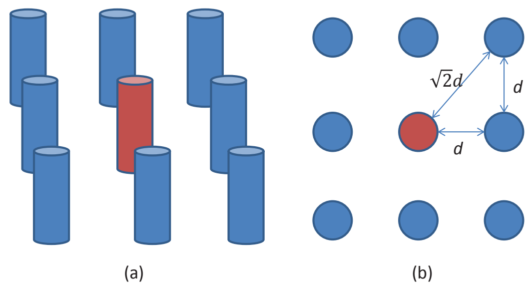

Fig. 1. (a) An illustration of the square of TSVs with 3D perspective and (b) the airscape.

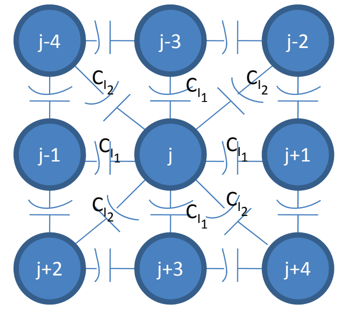

Fig. 2. A simple circuit model of a TSV bundle.

we introduce the 3D crosstalk noise model. Next, the design of ShieldUS is introduced. In Section V, we evaluate ShieldUS with synthetic traffic and real benchmarks, and investigate the performance of our interval equilibration unit. Then, related works are discussed in Section VI. Finally, conclusion is drawn in Section VII.

II. Problem Formulation

Assume a bundle of TSVs is formed by $M \times N$ TSVs, indexed as $TSV_{i,j}$, where $0 < i < M$ and $0 < j < N$, and also assume a program has total execution time T, and it is divided into several intervals, denoted as ordered set $I = \{I_1, I_2, I_3, ... I_H\}$. Then, denote the transmitted data as $D_{t \in I_k} = d_1, d_2, ..., d_{M \times N}$ at time t within the interval k.

Given a one-to-one mapping function $M : D \rightarrow TSV$, the mapping result at interval k is denoted as m_{I_k}. Also, given a function C which can judge the crosstalk class under the mapping result m_{I_k}, our goal is to find an ordered set $\{m_{I_1}, m_{I_2}, ..., m_{I_H}\}$ such that:

$$min \sum_{I_k \in I} \sum_{t \in I_k} C(m_{I_k}, D_t, D_{t-1})$$

III. 3D Crosstalk Noise Model

A. Preliminaries

Figure 1 shows a 3×3 square model. To the center TSV, there are two categories of interfering TSVs. The first category is for the TSVs of north, east, west, and south ones. The second category are corner ones. Assuming the distance of the first category is d, the distance of the second category is $\sqrt{2}d$. Without loss of generality, this model can be easily extended to any $N \times N$ square of TSVs, where $N \geq 3$.

B. Classes of Crosstalk in 3D chip

To calculate the crosstalk effects, the corresponding RC model is built as Figure 2 shows. C_L is the capacitance between a TSV and the substrate and C_I is the capacitance between two neighboring TSVs. Based on the circuit model in Figure 2, the following equation is derived by Ohm's law.

$$\begin{aligned}\frac{V_{dd} - V_j(t)}{R} &= C_L \frac{dV_j(t)}{dt} \\ &+ C_{I_1}(\frac{dV_{j,j-3}}{dt} + \frac{dV_{j,j-1}}{dt} + \frac{dV_{j,j+1}}{dt} + \frac{dV_{j,j+3}}{dt}) \\ &+ C_{I_2}(\frac{dV_{j,j-4}}{dt} + \frac{dV_{j,j-2}}{dt} + \frac{dV_{j,j+2}}{dt} + \frac{dV_{j,j+4}}{dt})\end{aligned} \quad (1)$$

where C_{I_1} and C_{I_2} are the interwire capacitances between the TSV j and direct neighbors and diagonal neighbors. The pitch of direct neighbors is shorter than diagonal neighbors, so C_{I_1} is larger than C_{I_2} according to the basic definition of capacitance. Here we assume a regular placement, i.e, the distance of C_{I_2} is $\sqrt{2}$ times that of the distance of C_{I_1}, and C_{I_1} is $\sqrt{2}$ times that of C_{I_2}. Note that it is dependent on technology parameters, such as the pitch between two TSV, the radius of a TSV, and the height of a TSV, etc. However, our model can be transformed by giving different technology parameters.

Then, the equation of signal delay of TSV affected by crosstalk inferred from the above equation is listed as follows:

$$\begin{aligned}\tau_j(\alpha) &= -RC_L(1 + \lambda_1(4 - \delta_{j,j-3} - \delta_{j,j-1} - \delta_{j,j+1} - \delta_{j,j+3}) \\ &+ \lambda_2(4 - \delta_{j,j-4} - \delta_{j,j-2} - \delta_{j,j+2} - \delta_{j,j+4}))\end{aligned} \quad (2)$$

where

$$\begin{cases} \lambda_1 = \frac{C_{I_1}}{C_L} \\ \lambda_2 = \frac{C_{I_2}}{C_L} \\ \Delta V_j = V_j(t^+) - V_j(t^-) \\ \Delta V_{j,k} = \Delta V_j - \Delta V_k \\ \delta_{j,k} = sgn(\Delta V_j)\frac{\Delta V_k}{V_{dd}}, \delta_{j,k} \in 0, \pm 1 \end{cases}$$

To compute the voltage transition, $V_j(t^+)$ and $V_j(t^-)$ mean the voltages at the time before the transition and after the transition respectively (i.e. $V_j(t^-) = 0$ and $V_j(t^+) = V_{dd}$). Besides, the normalized relative voltage change $\delta_{j,k}$ is used to simplify in the Equation 2. Then, we can define the effective capacitance of the jth TSV as follows:

$$\begin{aligned}C_{eff,j} &= C_L(1 + \lambda_1(4 - \delta_{j,j-3} - \delta_{j,j-1} - \delta_{j,j+1} - \delta_{j,j+3}) \\ &+ \lambda_2(4 - \delta_{j,j-4} - \delta_{j,j-2} - \delta_{j,j+2} - \delta_{j,j+4}))\end{aligned} \quad (3)$$

Therefore, Equation 2 can be rewritten as follows:

$$\tau_j(\alpha) = kV_{dd}C_{eff,j} \quad (4)$$

TABLE I
THE CLASSES OF TSV CROSSTALK. NOTE THAT NOT ALL CLASSES ARE SHOWN DUE TO THE LENGTH.

Class	C_{eff}	Transition patterns
1	C_L	000000000 $\rightarrow$ 111111111
2	$C_L(1+\lambda_2)$	011111111 $\rightarrow$ 000000000
3	$C_L(1+\lambda_1)$	101111111 $\rightarrow$ 000000000
4	$C_L(1+2\lambda_2)$	010111111 $\rightarrow$ 000000000
5	$C_L(1+\lambda_1+\lambda_2)$	001111111 $\rightarrow$ 000000000
6	$C_L(1+2\lambda_1)$	101011111 $\rightarrow$ 000000000
⋮		
81	$C_L(1+8\lambda_1+8\lambda_2)$	000010000 $\rightarrow$ 111101111

From Equation 4, the signal delay of TSV j is dominated by the $C_{eff,j}$. Besides, we can derive the upper and lower bound of $C_{eff,j}$ (as Equation 5 shows) because the possible values of $\delta_{j,k}$ are -1,0 and 1.

$$C_L \leq C_{eff,j} \leq C_L(1+8\lambda_1+8\lambda_2) \tag{5}$$

According to Equation 5, the table of the transition patterns can be derived listed in Table I.

IV. DESIGN OF SHIELDUS

A. Overview

In the previous section, the number of noise class has been calculated as 81 different classes. In this section, we aim at minimizing the occurrence of higher class transmissions and maximizing the occurrence of lower class transmissions.

Figure 3 shows the overview of the design of ShieldUS. N-bit data are transmitted to the crossbar and ShieldUS simultaneously. ShieldUS will send control to crossbar according to the recent history to shuffle the position of data bits. Recall that the main purpose of shuffling the data is to lower the number of class of crosstalk noise. After shuffling, ShieldUS will send message to generator to decide how many *short cycles* required by this transmission. Note that ShieldUS assumes *variable cycle transmission* [7], in which multiple short clock cycles are used instead of using a long single cycle which is based on the worst-case to avoid crosstalk. This assumption is suitable for TSVs since the lengths of TSVs (nano-scale) are quite shorter than traditional links (such as bus/NoC links), and therefore TSVs are able to provide higher frequencies to realize multiple short cycles for transmissions.

After introducing the overview of ShieldUS and its assumed transmission mechanism, in the following paragraphs the detailed design are discussed, including the dynamic data-to-TSVs remapping strategy and the interval equilibration strategy.

B. Dynamic Data-to-TSVs Remapping

For simplifying the discussion, we illustrate dynamic shielding with 3×3 TSVs. It can be generalized to any $N\times N$ array. As discussed in B, there are 81 classes noise classes for a TSV

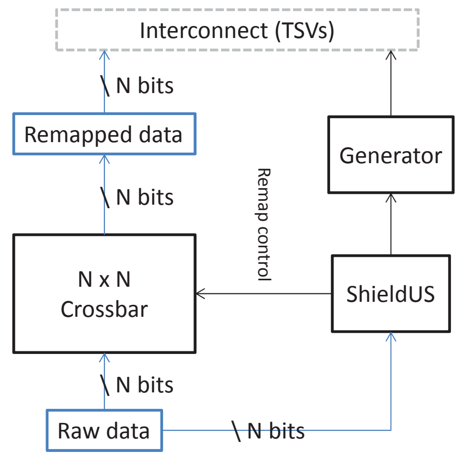

Fig. 3. An overview of the design of ShieldUS.

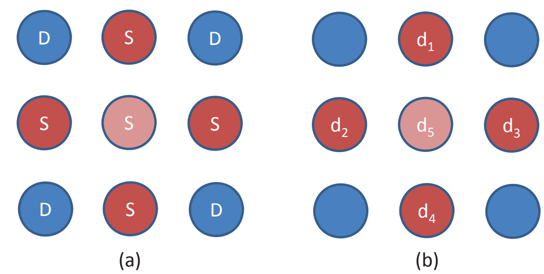

Fig. 4. (a) An illustration of 3×3 array and (b) the data mapping.

surrounding by 8 TSVs. We can organize the 3×3 TSVs as Figure 4a shows, where **D** stands for the data bits and **S** for shields. In this case, 9 TSVs contain 4 data TSVs and 5 shields. Note that the center TSV also causes crosstalk, but relatively weaker crosstalk effect comparing to the other 4 TSVs, so we used a lighter color to distinguish.

Consider the data width is 9, denoted as $d_9d_8d_7d_6d_5d_4d_3d_2d_1$, we firstly check the similarity in a given period and rank them. For simplifying the discussion, we assume the rank of similarity from highest to lowest is d_1 to d_9. Since the top 4 have the highest similarity, the 4 bits are mapped to the shield cells, and the fifth one is mapped to the center cell, as Figure 4b shows. The rest four TSVs can be used as normal data TSVs, and the remaining bits are mapped to them. Note that the procedure is operated every I cycles. Each time I cycles pass, the mapping is reset according to the recent observation.

In this procedure above tries to minimize the number of noise class. However, since the noise class varies in different periods, the variable cycle transmission is used [7]. That is, precalculate the required number of cycles of each noise degree for the sender side.

B.1 Equilibrating interval for remapping

Setting a fixed interval for remapping is not practical since each program has their own program behaviors. Therefore, we

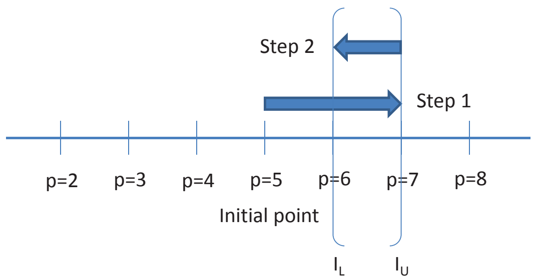

Fig. 5. An illustration of equilibrating the interval for remapping.

integrate *Interval Equilibration Unit* (IEU) into the dynamic shielding. The idea of adapting to a suitable interval is widely-used in design of wired and wireless network, similar but not the same to backoff algorithms [5].

In IEU, the initial interval is set as C^p, where C is a user-defined base number for identifying the aggressiveness of each trial, and p is the number for exponentially increasing or decreasing the base number while previous adapting interval has gained benefits, i.e., lower crosstalk noise.

For example, assuming $C = 10$ and initial interval is set as $C^5 = 100,000$, after the first interval passes, IEU tries $C^6 = 1,000,000$ as the longer interval to explore the similarity. If the result is better than the previous interval, the following interval will be set as $C^7 = 10,000,000$. However, if the larger interval results in worse case for two times in a row, the upper bound of the interval is then decided. Then, the same function is executed but oppositely for finding the lower bound of the interval. Then, the range of interval will be decided, denoted as $[I_L, I_U]$.

Note that the above procedure will be run repeatedly until $I_L = I_U$. Since each time either the upper bound or the lower bound will be narrower, finally the interval will be a fixed value. Figure 5 shows an illustration of this equilibration strategy.

Nevertheless, even a single program may have different program phases at different times [6], so the history may need to be reset and IEU requires to be run periodically for re-adapting to the internal program phase changing. In Section V, related evaluations are discussed.

V. Evaluation

In our evaluation, the number of TSV is 49 and TSVs are arranged by a 7×7 square. The resulting class of each transmission is the largest one which decides the required time for this transmission. There are 82 classes in total and in which the 81 classes are derived from Section III, and the additional class (class 0) stands for the case of no transition, such as $00000000 \rightarrow 00000000$.

The signal delay is proportional to C_{eff} by Equation 4 mentioned in Section III. To accurately estimate improved performance, we set λ_1 and λ_2 as 5.54 and 3.92 [12] to get the delay time for each class. The related technology parameters are listed Table II. If we set one clock cycle as the delay time of class 0, the frequency must be very high because the delay time of class 82 is 76 times larger than class 0. Therefore, we adopt the delay time of class 11 as one clock cycle, and then we group several classes into a group. The first 11 classes including class 0 are grouped into Group 1, denoted as $Group_1 = (0, 10)$. The remaining groups are $Group_2 = (11, 30)$, $Group_3 = (31, 55)$, $Group_4 = (56, 74)$ and $Group_5 = (75, 81)$. Therefore, we set the transmission time for $Group_1$ as one cycle, $Group_2$ as two cycles and so on.

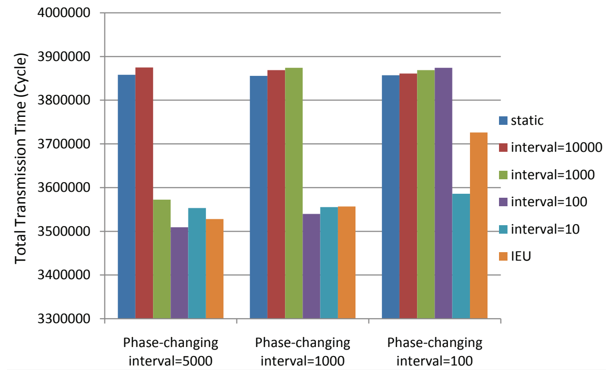

Fig. 6. Performance evaluation under different phase-changing interval and capturing interval of ShieldUS.

A. Mechanism Validation

In this experiment, we firstly validate the mechanism of ShieldUS with synthesized traffic. Since ShieldUS is a dynamic shielding strategy, among the 49 TSVs, we periodically select 16 of them to have higher transition probability, and 33 of them have lower transition probability. That is, 33 data-bits are more suitable to be the shields. The transition probabilities change with different phase-changing intervals, as Figure 6 show, i.e., 5000, 1000, and 100. Besides ShieldUS, we also implement static shielding, i.e., selecting fixed bits as shields, which uses the most significant 33 bits as the shields.

As Figure 6 shows, ShieldUS outperforms static shielding, especially when the capturing interval is smaller than phase-changing interval. This is because once the phase changes, ShieldUS requires a period to learn the history. The results validate that ShieldUS has the ability to learn the patterns from history and successfully selects the suitable bits as shields.

Another important observation is the intervals. With different interval settings, the performance is dramatically affected, as Figure 6 shows, while the interval is set as 100 in the first two cases (phase-changing interval=5000 and 1000), the transmission time can be improved by 10% and 5% on average compared with the interval set to 10000 and 1000, respectively.

It concludes that having a suitable interval is an important parameter in ShieldUS. However, selecting a suitable interval for any program is not possible since every program has its own best one. Therefore, IEU which is able to select a suitable interval intelligently is also evaluated and compared with the fixed interval settings in this experiment. In Figure 6, the performance results of IEU are the best or the second best in all the cases and improved by 8.5% compared with static mapping.

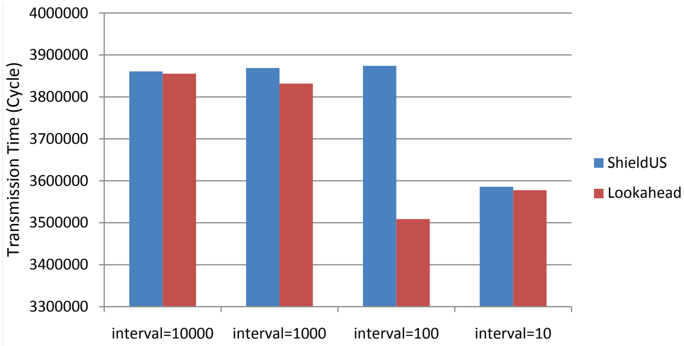

Fig. 7. Accuracy of ShieldUS. Comparing to the ideal shields selecting by an oracle method. Note that the intervals here represent the visions of looking behind and looking ahead

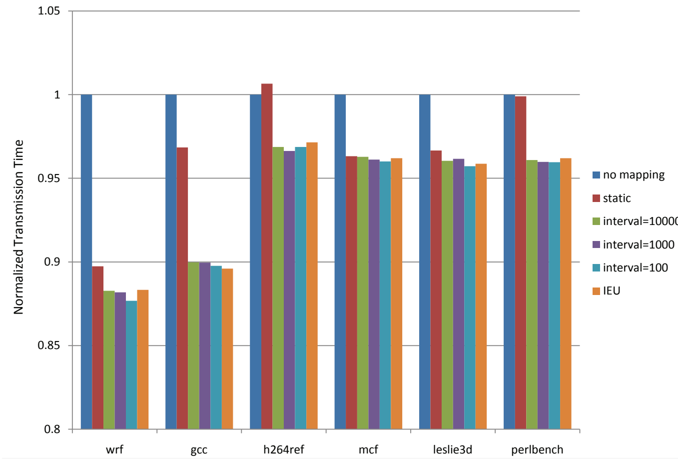

Fig. 8. Evaluation with real benchmark traces and the performance results are normalized to no mapping.

B. Accuracy Comparison with Oracle Method

In this experiment, we evaluate the accuracy of ShieldUS. An oracle method is used to look ahead the incoming traffic in advance. With looking ahead, the shields are selected guaranteeing *minimal transitions* in the incoming period. Note that ShieldUS is a looking behind strategy, by comparing to the results of looking ahead, the accuracy of ShieldUS can be evaluated. Note that the oracle method is impractical in real cases since the incoming traffic cannot be known in advance; however, it sets a performance upper bound for evaluating our strategy.

We select the result with phase-changing interval equal to 100 as Figure 7 shows. The interval here stands for lengths of looking behind and looking ahead. It shows that ShieldUS has almost the same result as the oracle method except when the interval is 100. It is because that phase-changing interval matches the looking ahead interval. However, it is impractical as mentioned above. Moreover, it again shows that a suitable interval strongly affects the effectiveness regardless ShieldUS or oracle method.

C. Real Benchmark Traces

After using synthesized traffic for observing the behaviors in depth, we use real benchmark traces to evaluate the effectiveness. The traces are the data transitions in address bus line, captured from `perlbench`, `gcc`, `mcf`, `leslie3d`, `h264ref`, and `wrf`. Note that the transition in address bus line is highly regular (achieving 99% similarity) which has been found in many previous works, such as [12]. Therefore, selecting the MSBs as the shields is expected to have good results. Here we evaluate ShieldUS to see if it can perform as good as static shielding.

As Figure 8 shows, the first observation is that the performance result of static shielding is worse than *no mapping* in `h264ref`, but ShieldUS can always outperform *no mapping*. The second observation is that ShieldUS ties with static shielding in `mcf`; however, it outperforms static shielding in other benchmarks and it can improve performance by 8% compared to static shielding in `gcc`.

Note that the transmitted value in address bus line is quite stable and thus the period for observation does not dominate the results. Although the impacts of interval setting is not obvious in this experiment, we can still observe that the optimal interval setting is not the same in these benchmarks. For example, the optimal interval in `wrf` and `leslie3d` is 100, but 1000 in `h264ref`.

In this experiment, IEU is also evaluated and compared with the fixed interval settings. As Figure 8 shows, IEU has up to 12% and 7% improvements in performance compared to no mapping and static mapping, respectively.

TABLE II
THE TECHNOLOGY PARAMETERS.

Parameter	Technology notes
Technology	45nm
Width (nm)	102.5
Space (nm)	102
Thickness (nm)	235.75
Height (nm)	215.25
Dielectric constant	2.3

VI. RELATED WORKS

In [14, 8, 18], the effect of crosstalk in 3D TSVs is evaluated. However, the assumed models consider the coupling noise with two kinds of TSVs, i.e., aggressor and victim, which simplify the the coupling noise problem. More general 3D models are given, such as [17], but most of them are from the view point of physical design, and our work solves the problem considering the frequencies of occurrences of each bit-line, which actually is a content-aware solution and independent of physical wiring considerations.

In [12], they also investigate the problem from a content-driven perspective. They found that the changed part of data (usually the upper part, i.e., MSBs) can be used as shields for the lower part (i.e., LSBs). That is, interleave the upper bits to the lower bits. In our work, we also check the similarity from the content perspective, but we periodically check the similarity in a given period and dynamically dispatch them as shields, rather than a design time decision.

In [7], it pointed out that the previous works usually use longest cycle as the working frequency to transmit data. Since the crosstalk noise can be classified to 6 groups, from no noise to high noise, previous works usually used the worst case (the 6th group) to decide the working frequency, which results in low performance. Therefore, they proposed *variable cycle* to send data. Instead of using a long cycle, several shorter cycles are used since the lengths of TSVs are quite shorter than traditional links, providing higher frequencies to realize multiple short cycles. In this paper, we applied the idea that with different noised data, different cycles should be used to maximize the performance.

In [13], the idea of shuffling the bus line to minimize the power consumption is proposed. Similarly, in [16], it pointed out an idea that data can be rotated regularly to balance the thermal distribution. Inspired by the previous works, we improved the idea to dynamic remapping to select the shields rather than just rotating the bits regularly.

VII. Conclusion

In this paper, we firstly discuss the 3D crosstalk coupling noise model, and then propose *ShieldUS*, a novel runtime data-to-TSVs remapping mechanism for shielding the transmission. The core idea of ShieldUS is that the transmitted data in a period are observed and the most unchanged bits are mapped to the TSVs as shields based on the 3D crosstalk model. With our novel Interval Equilibration Unit (IEU) in ShieldUS, the interval for remapping data-to-TSVs can be automatically sandwiched in between an upper bound and a lower bound and finally fixed to a value. The experiments are threefold: First, we validate ShieldUS with synthesized traffic to show that ShieldUS can successfully select the suitable bits to map to TSVs as shields. Second, we evaluate the accuracy by comparing with an oracle method, which shows that ShieldUS has almost the same performance as oracle method. Finally, we evaluate IEU, an intelligent unit for finding a suitable interval for observing the behaviors of TSVs, which makes dynamic shielding practical and does not need to predefine parameters, and therefore improves the practicability of ShieldUS.

Our future works are twofold: First, the crossbar for shuffling the bits cannot be ignored since it grows as the number of bits to shuffle. We are considering a two-stage crossbar to eliminate the overhead. Second, the details of the implementation are going to be discussed in the future work.

VIII. Acknowledgements

This work was supported in part by NSF grants 1213052,1147388, 1017277, 0903432 and 0916887 as well as Industrial Technology Research Institute and National Science Council grant NSC 101-2220-E-007-025.

References

[1] R. Arunachalam, E. Acar, and S. Nassif. Optimal shielding/spacing metrics for low power design. In *VLSI, 2003. Proceedings. IEEE Computer Society Annual Symposium on*, pages 167–172. IEEE, 2003.

[2] K. Karmarkar and S. Tragoudas. Scalable codeword generation for coupled buses. In *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '10, pages 729–734, 3001 Leuven, Belgium, Belgium, 2010. European Design and Automation Association.

[3] H. Kaul, D. Sylvester, and D. Blaauw. Active shields: a new approach to shielding global wires. In *Proceedings of the 12th ACM Great Lakes symposium on VLSI*, pages 112–117. ACM, 2002.

[4] J.-S. Kim, C. S. Oh, H. Lee, D. Lee, H.-R. Hwang, S. Hwang, B. Na, J. Moon, J.-G. Kim, H. Park, J.-W. Ryu, K. Park, S.-K. Kang, S.-Y. Kim, H. Kim, J.-M. Bang, H. Cho, M. Jang, C. Han, J.-B. Lee, K. Kyung, J.-S. Choi, and Y.-H. Jun. A 1.2v 12.8gb/s 2gb mobile wide-i/o dram with 4x128 i/os using tsv-based stacking. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pages 496 –498, feb. 2011.

[5] B.-J. Kwak, N.-O. Song, and L. Miller. Performance analysis of exponential backoff. *Networking, IEEE/ACM Transactions on*, 13(2):343 – 355, april 2005.

[6] J. Lau, S. Schoenmackers, and B. Calder. Transition phase classification and prediction. In *High-Performance Computer Architecture, 2005. HPCA-11. 11th International Symposium on*, pages 278 – 289, feb. 2005.

[7] L. Li, N. Vijaykrishnan, M. Kandemir, and M. Irwin. A crosstalk aware interconnect with variable cycle transmission. In *Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings*, volume 1, pages 102 – 107 Vol.1, feb. 2004.

[8] C. Liu, T. Song, J. Cho, J. Kim, J. Kim, and S. K. Lim. Full-chip tsv-to-tsv coupling analysis and optimization in 3d ic. In *DAC*, pages 783–788, 2011.

[9] J. Ma and L. He. Formulae and applications of interconnect estimation considering shield insertion and net ordering. In *Computer Aided Design, 2001. ICCAD 2001. IEEE/ACM International Conference on*, pages 327–332. IEEE, 2001.

[10] M. Mutyam. Selective shielding: a crosstalk-free bus encoding technique. In *Computer-Aided Design, 2007. ICCAD 2007. IEEE/ACM International Conference on*, pages 618 –621, nov. 2007.

[11] S. Pasricha. Exploring serial vertical interconnects for 3d ics. In *Design Automation Conference, 2009. DAC'09. 46th ACM/IEEE*, pages 581–586. IEEE, 2009.

[12] N. Satyanarayana, A. Vinaya Babu, and M. Mutyam. Delay-efficient bus encoding techniques. *Microprocess. Microsyst.*, 33(5-6):365–373, Aug. 2009.

[13] Y. Shin and T. Sakurai. Coupling-driven bus design for low-power application-specific systems. In *Design Automation Conference, 2001. Proceedings*, pages 750 – 753, 2001.

[14] T. Song, C. Liu, D. H. Kim, S. K. Lim, J. Cho, J. Kim, J. Pak, S. Ahn, J. Kim, and K. Yoon. Analysis of tsv-to-tsv coupling with high-impedance termination in 3d ics. In *ISQED*, pages 122–128, 2011.

[15] B. Victor and K. Keutzer. Bus encoding to prevent crosstalk delay. In *Proceedings of the 2001 IEEE/ACM international conference on Computer-aided design*, ICCAD '01, pages 57–63, Piscataway, NJ, USA, 2001. IEEE Press.

[16] F. Wang, Y. Xie, N. Vijaykrishnan, and M. J. Irwin. On-chip bus thermal analysis and optimization. In *Proceedings of the conference on Design, automation and test in Europe: Proceedings*, DATE '06, pages 850–855, 3001 Leuven, Belgium, Belgium, 2006. European Design and Automation Association.

[17] R. Weerasekera, M. Grange, D. Pamunuwa, H. Tenhunen, and L.-R. Zheng. Compact modelling of through-silicon vias (tsvs) in three-dimensional (3-d) integrated circuits. In *3D System Integration, 2009. 3DIC 2009. IEEE International Conference on*, pages 1 –8, sept. 2009.

[18] Z. Xu, A. Beece, D. Zhang, Q. Chen, K. neng Chen, K. Rose, and J.-Q. Lu. Crosstalk evaluation, suppression and modeling in 3d through-strata-via (tsv) network. In *3D Systems Integration Conference (3DIC), 2010 IEEE International*, pages 1 –8, nov. 2010.