Coupling Mitigation in 3D Multiple-Stacked Devices

Pooria M.Yaghini, *Student Member, IEEE,* Ashkan Eghbal, *Student Member, IEEE,* Misagh Khayambashi, and Nader Bagherzadeh, *Fellow, IEEE,*

***Abstract*—3D multiple-stacked IC has been proposed to support energy efficiency for data center operations as DRAM scaling improves annually. A 3D multiple-stacked IC is a single package containing multiple dies, stacked together, using Through-Silicon Via (TSV) technology. Despite the advantages of 3D design, fault occurrence rate increases with feature size reduction of logic devices, which gets worse for 3D stacked designs. TSV coupling is one of the main reliability issues for 3D multiple-stacked IC data TSVs. It has large disruptive effects on signal integrity and transmission delay. In this paper, we first characterize the inductance parasitics in contemporary TSVs, then we analyze and present a classification for inductive coupling cases. Next we devise a coding algorithm to mitigate the TSV-to-TSV inductive coupling. The coding method controls the current flow direction in TSVs by adjusting the data bit streams at run-time to minimize the inductive coupling effects. After performing formal analyses on the efficiency scalability of devised algorithm, an enhanced approach supporting larger bus sizes is proposed. Our experimental results show that the proposed coding algorithm yields significant improvements while its hardware-implemented encoder results tangible latency, power consumption, and area.**

***Index Terms*—Reliability, 3D multiple-stacked IC, TSV, Coupling, Signal Integrity.**

I. Introduction

EXASCALE systems are expected to have approximately one billion processing elements by 2020 [1], [2]. Three Dimensional (3D) IC designs is considered as a viable solution for integrating more cores on a chip, while imposing smaller footprint area and better timing performance than 2D architecture [3].

Wire-bonding and flip-chip stacking have made their ways into mainstream semiconductor manufacturing in recent years, but they are not considered anymore for new generation of 3D integrations [4]. Through-Silicon Via (TSV) is currently more attractive, supporting better performance and integrated functionality. With 3D integration technology employing TSVs, the average and maximum distance among components on different dies will be substantially reduced, savings delay, power, and area factors [5]. Fault occurrence rate increases with the emergence of nano-scale circuits and designing more complex circuits on a chip, which will be more critical for 3D multiple-stacked IC designs. Reliable TSVs, as one of the major components in 3D multiple-stacked IC, are in demand. Many research groups have studied and proposed reliable TSVs [6]–[8]. The reliability of TSV interconnects is analyzed in [6]. The reliability aware TSV planning for the 3D multiple-stacked IChas been proposed in [7]. The key design for reliability challenges and possible solutions for TSV-based 3D IC integration are discussed in [8].

TSV coupling is one of the major issues in 3D multiple-stacked IC designs because of increased parasitic signals as compared to 2D ICs, which may result in delay or even mutual coupling between adjacent TSVs [9], [10]. The term TSV coupling refers to capacitive and inductive couplings among neighbor TSVs, which the latter is more critical in higher frequency data transmissions [11]. Electric field results in capacitance coupling and magnetic field is a source of inductive coupling. The impact of TSVs on SI in 3D ICs has been investigated in several articles [12], [13]. TSV-to-TSV inductive coupling is one of the fault sources exacerbating the Signal Integrity (SI) effect which causes two major issues. First, it increases the path delay due to Miller effect. Second, the coupling noise can result in logic function failure. An analytical model for the coupling capacitance between pairs of TSVs is presented in [12]. The TSV resiliency during manufacturing steps by investigating resistive open defects has been studied [14]. A complete set of self-consistent equations including self and coupling terms for resistance, capacitance and inductance of various TSV structures are presented in [13]. As a solution to TSV-to-TSV coupling issue increasing TSV distances, shielding the victim TSVs, inserting buffers at the victim net, decreasing the driver size at the aggressor net, and increasing the load at both victim and aggressor net have been suggested [10]. However, the last two suggestions have negative implications on timing performance, and others need high effort at post-design time.

In [15], a coding scheme has been suggested for a matrix of TSVs, reducing the maximum capacitive crosstalk by 25% in a mesh of TSV with size of $3 \times n$. However, the impact of inductive coupling on SI has not been evaluated yet. The goal of this paper is to investigate the reliability of 3D multiple-stacked IC against the inductive TSV-to-TSV coupling and to propose a scalable inductive coupling aware coding. The proposed algorithm in this paper is intended to support two major 3D device categories. The devised baseline algorithm [16] targets the first 3D device category which consists of designs with low TSV concentration (less than 100 TSVs) such as 3D NoC [17]–[19]. An enhanced scheme is proposed to support architectures with high TSV concentration (around 500 TSVs) such as 3D DRAM memories (Hybrid Cube Memory (HMC) [20]–[22]), as shown in Fig. 1. Our experimental results show that the

The authors are with the Center for Pervasive Communications and Computing at Department of Electrical Engineering and Computer Science, University of California, Irvine, CA, 92697 USA e-mail: (pooriam@uci.edu; aeghbal@uci.edu; mkhayamb@uci.edu; nader@uci.edu).

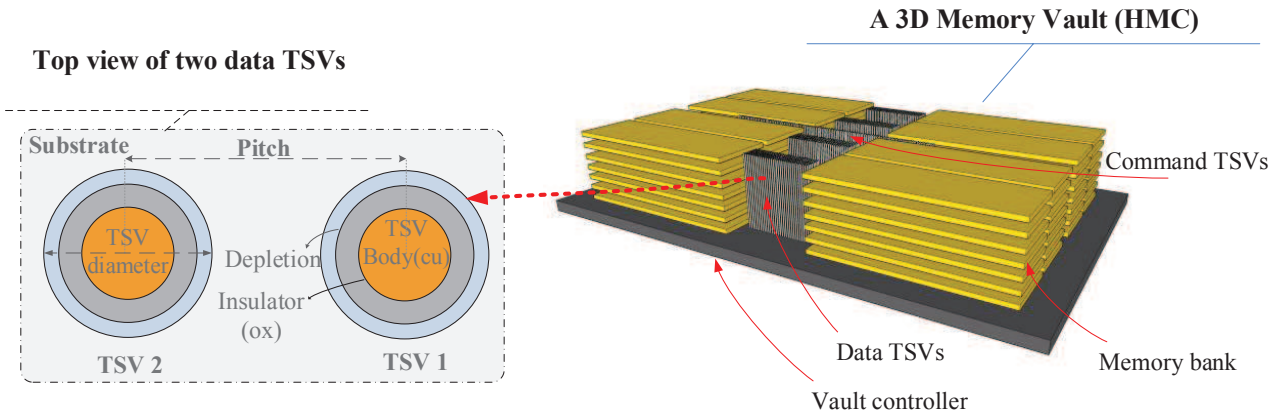


Fig. 1. 3D multiple-stacked IC vault, vertically interconnected by TSV bus in 3D integration technology

proposed coding algorithm yields significant improvements while its hardware-implemented encoder depicts tangible latency, power consumption, and area.

The major contributions of this article are:

- To analyze the reliability issue of inductive TSV-to-TSV coupling fault effect within a 3D multiple-stacked IC. Also providing an analytical failure (data corruption) estimation of TSV links caused by inductive TSV coupling effect.
- To devise a method to minimize the effect of magnetic field caused by TSVs, including an analytical analysis to demonstrate the strength of proposed technique.
- To present a scalable coding approach with modest information redundancy overhead for implementing the proposed technique in large-scale 3D multiple-stacked ICs.
- To prove the efficiency in mitigating TSV-to-TSV inductive coupling and justify the scalability and practicality of our proposed schemes through concrete experiments and hardware implementation and synthesis.

The rest of this article is organized as follows: A brief introduction of 3D multiple-stacked IC architecture is provided in Section II. Inductive TSV-to-TSV coupling is investigated in Section III. Section IV presents an inductive coupling analysis. The baseline algorithm coding and its improvement evaluation are discussed in Sections V and VI, respectively. The enhanced algorithm coding scheme and its coupling mitigation efficiency and hardware implementation are provided in Section VII. Section VIII overviews the related work. Finally, Section IX delivers some conclusion remarks.

II. 3D Multiple-Stacked IC Architecture

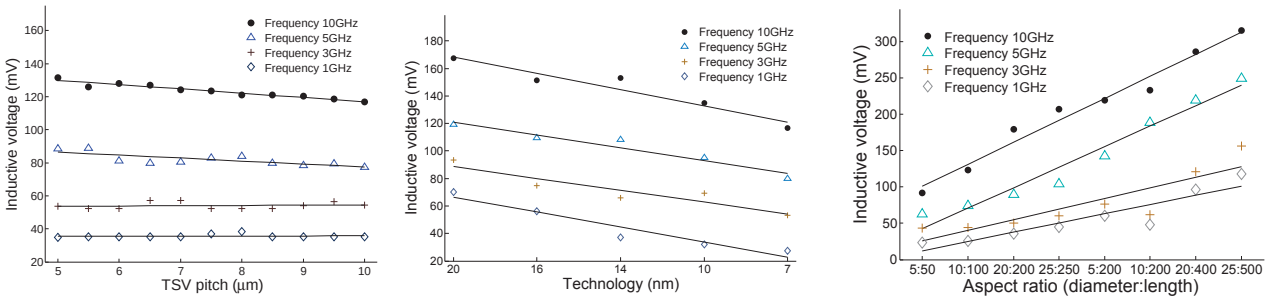
The advent of 3D or vertical integration is a promising path to boost scalability and power/performance characteristics to extend capabilities of modern integrated circuits [23]–[25]. These capabilities are inherent to 3D ICs, resulting in considerably shorter interconnecting wires in the vertical direction. 3D integration supports new opportunities by providing feasible and cost effective approaches for integrating heterogeneous cores to realize future computer systems. It supports heterogeneous stacking because different types of components can be fabricated separately, and silicon layers can be implemented with different technologies. One of the most promising technologies for 3D IC integration is the notion of TSVs [26], pillars manufactured across thinned silicon substrates to establish inter-die connectivity after die bonding. Salient TSV features include: fine pitches, high densities, and high compatibility with the standard CMOS process. 3D ICs are interconnected with high-density short and thin TSVs, supporting low-level integration and superior to existing solutions. Multiple layers of 2D planar designs are stacked on each other and are vertically interconnected by high-density short and thin TSVs in 3D integration technologies.

Micro-bumps are the interfaces between TSVs and 2D designs. The minimum TSV depth normally is about 40-$100\mu m$ which is projected to reach 30-$40\mu m$ by 2018. A copper TSV in standard Si-bulk technology is expected to have minimum via diameter of 2-$4\mu m$, 1:20 minimum aspect ratio, 4-$7\mu m$ minimum via pitch, $0.5\mu m$ oxide thickness (t_{ox}), and there can be up to 2-8 dies per stack [27]. It is important to note that the TSV process is independent from the technology node used for the 2D chip and it does not scale. TSV diameters and pitches are two to three orders larger than transistor channel gate lengths. Furthermore, in order to reach a high yield rate, manufacturers typically impose a minimum TSV density policy to maintain the planarity of the wafer during chemical and mechanical polishing [28]. For example, Tezzaron requires that at least one TSV in every $250\mu m \times 250\mu m$ area [29].

III. Inductive TSV-to-TSV Coupling

The impact of TSVs on future 3D ICs is still unknown [30]. However, chip warpage, TSV coupling, and thermal stress are known as main causes of TSV failure [9], [26].

The term TSV coupling refers to capacitive and inductive couplings among neighboring TSVs. Electric field results in capacitance coupling and magnetic field is a source of inductive coupling. The capacitance coupling between TSVs depends on the permittivity of the oxide, TSV geometry, the arrangement of surrounding TSVs and body contacts places. The capacitance coupling is influenced by the arrangement of the TSVs while the inductive coupling



(a) TSV Pitch (b) Technology (c) TSV aspect ratio

Fig. 2. Inductive coupling SPICE simulation results

is slightly dependent on the distance of neighbor TSVs as inductive coupling has longer range of effect than capacitance coupling noises [31]. Inductive coupling among neighboring TSVs is more critical in higher frequency data transmissions [32], and long TSVs which is considered in this article. Processors with higher operating frequency are emerging as the process technology is scaling down; an example is the IBM 5.2GHz multiprocessor [33].

A. Inductive Coupling Characteristics

To characterize the effect of inductive coupling, a 3x3 matrix of TSVs is modeled at circuit-level with Synopsys HSPICE. The middle TSV is assumed to be the victim and the other 8 are the aggressors. In simulations, the top end of each TSV is connected to the output of an inverter, which drives the input of another inverter connected to the bottom of the TSV. The coupled TSV structure is modeled based on [10], [34] as a lumped RLC circuit. The circuit is composed of a series TSV resistance R_{TSV} and inductance L_{TSV}, parallel silicon substrate resistance R_{si} and capacitance C_{si}, and silicon dioxide capacitance C_{ox} around the TSV. These components describe the relationship between the signal TSVs. The values of circuit elements are obtained using analytic equations based on the dimensions of the structure, such as oxide thickness t_{ox}, silicon substrate height h_{si}, TSV radius r_{TSV}, and TSV pitch P_{TSV} and by material properties like dielectric constant ϵ and resistivity ρ.

$$R_{si} = \rho_{si} \frac{\cosh^{-1}\left[\frac{P_{TSV}}{2r_{TSV}}\right]}{\pi h_{si}} \tag{1}$$

where

$$\rho_{si} = 0.0012T^2 - 0.0352T + 10 \tag{2}$$

and

$$C_{si} = \epsilon_{si} \frac{\pi h_{si}}{\cosh^{-1}\left[\frac{P_{TSV}}{2r_{TSV}}\right]} \tag{3}$$

$$C_{ox} = \epsilon_{ox} \frac{2\pi h_{si}}{\ln \frac{r_{TSV}+t_{ox}}{r_{TSV}}]} \tag{4}$$

where

$$\epsilon_{ox} = 0.016T + 3.6 \tag{5}$$

TSV inductance [35] is also derived through partial self-inductance and mutual inductance. Partial self-inductance depends on the diameter and length of TSV and is expressed as:

$$L_{TSV_{self}} = \frac{\mu_0 l_{TSV}}{2\pi}[ln(\frac{2l_{TSV}}{r_{TSV}}) - \frac{3}{4}] \tag{6}$$

$$L_{TSV_{Mutual}} = \frac{\mu_0 l_{TSV}}{2\pi}[ln(\frac{l_{TSV}}{P_{TSV}} + \sqrt{1+(\frac{l_{TSV}}{P_{TSV}})^2} - \sqrt{1+(\frac{P_{TSV}}{l_{TSV}})^2} + \frac{P_{TSV}}{l_{TSV}}] \tag{7}$$

where μ_0 is the permeability of free space given by $4\pi \cdot 10^{-7}$.

Predictive Technology Model (PTM) [36] FinFET transistor models are employed to implement inverters in this experiment. The worst-case induced voltage on the victim TSV is reported for different TSV pitches (Fig. 2(a)), process technologies (Fig. 2(b)), and TSV aspect ratio (Fig. 2(c)) over different frequencies. The simulation parameter values are chosen according to ITRS [27] interconnect report, as shown in Table I.

Table I
SIMULATION CONFIGURATIONS IN FIG. 2

Figure	Technology	Length	Pitch	Diameter
2(a)	$14nm$	$100\mu m$	$8\mu m$	$4\mu m$
2(b)	$20, 16, 14, 10, 7nm$	$100\mu m$	$8\mu m$	$4\mu m$
2(c)	$14nm$	$10-500\mu m$	$9-29\mu m$	$5-25\mu m$

Based on Fig. 2(c), it is observed that as TSVs become longer (even with the same aspect ratio) the magnetic flux linking the two TSVs increases proportionally. Therefore, as the length of TSVs grow, mutual coupling between aggressors and victim increases almost linearly and the coupled voltage rises proportionally.

Although the linkage flux between two TSVs is a strong function of the length, its dependence on TSV-to-TSV pitch is weak. Changing the pitch between cylindrical TSVs affects mutual inductance in two ways. First, it changes the magnetic field created by the aggressor. Secondly, considering Faraday's law, it alters the surface on which the magnetic field is integrated to calculate the linkage flux. As

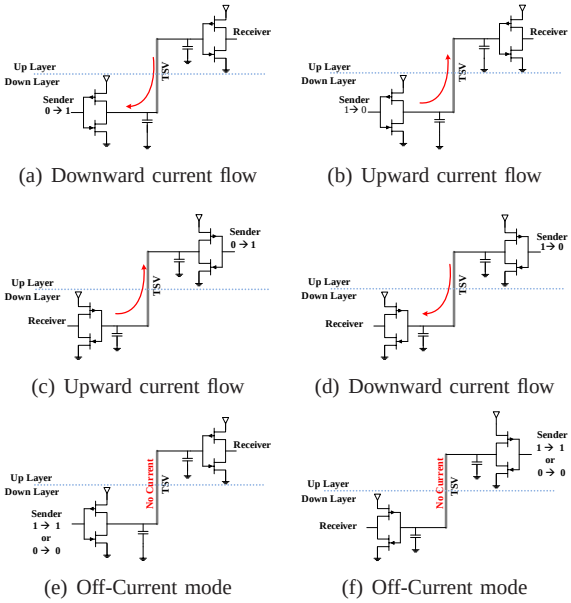


Fig. 3. Current flow direction in TSV.

long as the proximity effect and other high order magnetic effects are trivial, current distribution in a TSV remains almost symmetrical regardless of the pitch size. Therefore, the magnetic field created by an aggressor does not vary by the pitch size, making the first effect to be negligible. Since the pitch between the TSVs is at least an order of magnitude smaller than their lengths, the second effect is small, but the linkage flux and consequently mutual coupling decreases slightly as pitch increases, shown in Fig. 2(a).

As shown in Fig. 2(b), induced voltage is a function of process technology. As processes advance, gate capacitance gets smaller and voltage rise/fall time becomes shorter. These two effects have opposite impacts on charging/discharging current of gate capacitance. The same current that charges (or discharges) the gate capacitance passes through TSV and causes inductive coupling to its neighboring TSVs. Thus, inductively coupled voltage varies for different technologies.

As technology advances and supply voltage shrinks, the coupled voltage becomes a greater portion of V_{dd}, resulting in higher probability of error. Among all the physical parameters, the length of the TSVs has the major impact on inductive coupling, specifically for global TSVs connecting more number of layers.

B. Current Flow in TSVs

The current flow direction of a TSV is data-dependent, based on charging and discharging of the intermediate capacitor between each pair of transistors of stacked planar. The behavior of the intermediate capacitor relies on the previous and current data bit values. Fig. 3 illustrates six possible cases depending on the data bit values and location of the sender, resulting in three possible current flows in TSVs. There is a downward current flow when the input data bit of sender changes from '0' to '1' and '1' to '0' if the sender is in lower and upper level, respectively as shown in Fig. 3(a) and Fig. 3(d). Similarly, there is an upward current flow direction, if the data bit of the sender changes from '1' to '0' and '0' to '1' if the sender is in lower and upper level, respectively as shown in Fig. 3(b) and Fig. 3(c). TSV does not carry any current, if there is no switching between the previous and next data bit values, as showed in Fig. 3(e) and Fig. 3(f). In the rest of this article such a TSV is called an inactive TSV, which does not have any current flow.

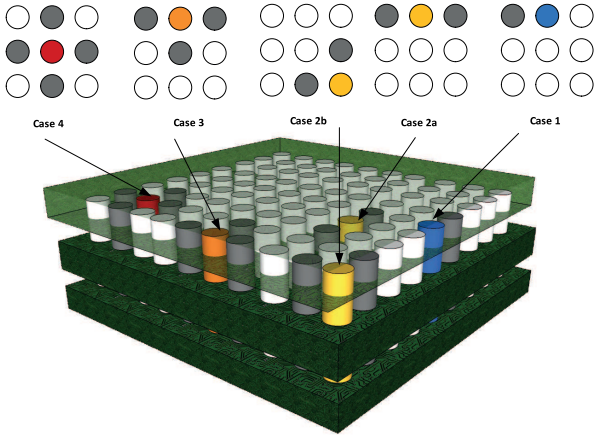


Fig. 4. Different TSV patterns leading to inductive coupling

IV. Inductive Coupling Analysis

A. Problem Definition

An accurate analysis of coupling-induced failure requires a complex electromagnetic analysis of neighboring TSVs. Since such an analysis is outside the scope of this article, an approximate form of the problem is considered.

Assuming the electromagnetic proximity effect and other high order effects can be neglected, the coupling-induced voltage β_{tot} is simply the sum of voltages induced on the victim TSV by the neighboring TSVs. Faraday's law implied that:

$$\beta_{\text{tot}} = \sum_{i=1}^{N} V_{\text{coupl,i}} = \sum_{i=1}^{N} M_{\text{v,i}} \frac{dI_i}{dt} \tag{8}$$

where

- N is the total number of aggressors.
- $V_{\text{coupl,i}}$ is the voltage coupled on the victim by i^{th} aggressor, assuming all other aggressors have constant current.
- $M_{\text{v,i}}$ is the mutual inductance between i^{th} aggressor and victim TSVs. $M_{v,i}$ is calculated from Equation 9 [35].

$$M_{\text{v,i}} = \frac{\mu_0}{2\pi}\left[l\ln\left(\frac{l+\sqrt{{d_i}^2+l^2}}{d_i}\right) + d_i - \sqrt{{d_i}^2+l^2}\right] \tag{9}$$

  where d_i is the distance of i^{th} aggressor from the victim TSV and l is the length of a TSV.
- I_i is the current of i^{th} aggressor TSV.

The representation of β_{tot} can be further simplified as follows. Assume that the inductive coupling voltage caused by a single horizontal or vertical neighboring TSV is β_{tot}, then β_{tot} is equal to $\alpha \times \beta$, where the parameter α depends on the current flow direction and arrangement of active neighboring TSVs.

Each victim TSV has four neighbors in horizontal or vertical directions. Fig. 4 shows a top view of different geometrical possibilities of neighbor configurations. Only 4 neighbors are considered to simplify the analysis. The parameter α equals the algebraic sum of current values in neighbors of the victim TSV. With only 4 neighbors, α assumes a values in $\{-4, \cdots, 4\}$. Clearly, the severity of inductive coupling is higher for larger absolute values of α and the goal of this article is to reduce current configurations that lead to high values of $|\alpha|$. In other words, the higher the sum of neighboring currents, the higher is the inductively coupled voltage.

With the cross section view of TSVs, the current of horizontal and vertical neighbors of a victim TSV are examined to measure the severity of inductive TSV-to-TSV coupling. The effect of diagonal neighboring TSVs, which causes less mutual coupling effect than adjacent TSVs, is not considered in our work.

B. Coding Scheme

The main contribution of this article is to propose a coding scheme to mitigate inductive coupling occurrence by adjusting the sequence of data flits[1]. The suggested coding technique replaces larger α values by smaller ones. While the baseline algorithm is intended to show the mitigation gain obtained by using our method, a variation of the baseline algorithm called "enhanced algorithm" introduces scalability into the baseline approach.

One possible approach to data modification is to perform inversion on a properly-selected set of input bit streams. For this method, decoding of the received signal requires knowledge of the location of the bits that have been inverted, inflicting serious overhead. One workaround for this overhead is to perform inversion on rows of TSV data rather than individual TSV bits. Clearly, this reduction in overhead comes at the cost of inferior performance.

The outline of baseline algorithm is explained as a two-phase algorithm:

1) In the first phase, each cell decides whether or not to submit inversion requests to its vertical neighbors (above and below) with the goal of decreasing the sum of its neighboring currents. Submitting the requests to only two neighbors, rather than four neighbors, is chosen for the sake of simplicity of the design. These requests are stored in for future processing.
2) Once all requests have been submitted, cells process their received requests and decide whether or not to accept inversion. Finally based on these individual decisions, each row decides whether or not to accept inversion.

[1]flit stands for "flow control unit"

C. System Model

Assume that the number of bits to be transmitted over TSVs is denoted by L_D at each time slot. The encoded data block is sent over a matrix of TSVs with N_R rows and N_C columns, with N_R and N_C satisfying $N_R \times N_C = L_D$.

With this convention, the original data to be transmitted at time slot t is represented by D_t matrix. Similarly, the encoded data that has already been sent at time $t-1$ is represented by $\hat{D}_{t-1}$.

The current flow direction of each TSV is specified by the modified data already sent over the TSV, namely $\hat{d}_{t-1}$ ($\hat{d}_{t-1}$ is a cell of $\hat{D}_{t-1}$ matrix). Similarly, $\hat{d}_t$ represents the encoded data bit to be transmitted, while d_t means the original bit to be transmitted at time slot t. With this convention and the proposed inversion mechanism, $\hat{d}_t$ will be either d_t or $\bar{d}_t$. A simple analysis of the circuitry connected to a TSV reveals that:

1) If the $\hat{d}_{t-1} = 0$ and $\hat{d}_t = 0$, the TSV current will be 0 ($\bigcirc$).
2) If the $\hat{d}_{t-1} = 0$ and $\hat{d}_t = 1$, the TSV current will be 1 ($\odot$).
3) If the $\hat{d}_{t-1} = 1$ and $\hat{d}_t = 0$, the TSV current will be -1 ($\otimes$).
4) If the $\hat{d}_{t-1} = 1$ and $\hat{d}_t = 1$, the TSV current will be 0 ($\bigcirc$).

with $\odot$, $\bigcirc$, and $\otimes$ representing current values of 1, 0, and -1 respectively. Consequently, the direction of current can be calculated, if $\hat{d}_{t-1}$ and $\hat{d}_t$ are known. From the preceding discussion, it is easy to see that the $N_R \times N_C$ matrix C representing the current of TSVs is equal to:

$$C(D1, D2) = D2 - D1 \tag{10}$$

The key parameter in baseline algorithm is the sum of neighboring TSV current. Therefore, it is helpful to define an $N_R \times N_C$ matrix P, where the $(i,j)^{th}$ element of P (P_{ij}) is equal to algebraic sum of neighboring TSV currents. From this definition, the elements of P can take any values in the set $\{-4, \cdots, 4\}$.

V. Baseline Coding Algorithm

In the proposed coding each cell (corresponding to each TSV) will send/receive an inversion request to the cell above or below itself based on its neighborhood condition. These neighbor cells then decide, based on the received requests, whether or not to honor the requests. Before delving into the details, the effect of bit inversion on TSV current should be examined.

A. Effect of bit inversion on TSV current

The direction of current passing through a TSV is specified by the previous data bit already sent over TSV and the data bit to be sent over the TSV as discussed in Section III-B. In the coding algorithm, $\hat{d}_t = \bar{d}_t$ if the

inversion decision is taken, otherwise $\hat{d}_t = d_t$. The change of current is summarized as follows:

1) In case of no inversion, $(\hat{d}_{t-1}, \hat{d}_t = d_t) = (0,0)$ results in ○. If an inversion is performed, i.e. $(\hat{d}_{t-1}, \hat{d}_t = \bar{d}_t) = (0,1)$, the current will be ⊙.
2) In case of no inversion, $(\hat{d}_{t-1}, \hat{d}_t = d_t) = (0,1)$ results in ⊙. If an inversion is performed, i.e. $(\hat{d}_{t-1}, \hat{d}_t = \bar{d}_t) = (0,0)$, the current will be ○.
3) In case of no inversion, $(\hat{d}_{t-1}, \hat{d}_t = d_t) = (1,0)$ results in ⊗. If an inversion is performed, i.e. $(\hat{d}_{t-1}, \hat{d}_t = \bar{d}_t) = (1,1)$, the current will be ○.
4) In case of no inversion, $(\hat{d}_{t-1}, \hat{d}_t = d_t) = (1,1)$ results in ○. If an inversion is performed, i.e. $(\hat{d}_{t-1}, \hat{d}_t = \bar{d}_t) = (1,0)$, the current will be ⊗.

It is concluded that ○ can be changed to both ⊗ and ⊙ (If d_t=1 or 0 respectively) by the inversion of d_t, while ⊙ and ⊗ are only changed to ○.

B. Reducing the sum of neighbor currents by inversion

The proposed coding consists of two phases as follows.

1) Submitting inversion requests: As mentioned previously, the goal of each cell is to see how it can reduce the sum of its neighbor currents by inverting the data on neighbor TSVs above and below itself. Table II lists all possible forms of requests that can be submitted by a victim TSV ⊕ to its neighbors to reduce $|\alpha|$. The classification is based on various scenarios that happen for the neighbors above and below of a victim TSV. The proposed actions in Table II are based on two factors; first the upward current flow (⊙) conversion to downward one (⊗) is not possible. Second, the current change, achieved by inversion of corresponding neighbor, should be adjusted in such a way that the magnitude of the sum of neighbor currents decreases. It is easy to verify that the proposed actions in the table follows these guidelines.

In some of the configurations of Table II, requests are sent to only one of the vertical neighbors, while in others requests will be sent to both neighbors. The former is identified by the word 'only' in the third column of Table II. The following example illustrates the necessity of sending a single request to only one of the neighbors. Consider the following TSV current configuration:

If the neighbor above is changed as ⊙ → ○, the sum of currents will be 0 which is the ideal situation. On the other hand, the same situation results if only the neighbor below is changed as ○ → ⊗.

It is desirable to have a simple decision rule rather than a lookup table in order to figure out when to submit an inversion request. The entire set of proposed requests in Table II is summarized in a very simple form.

1) For a cell (i,j) with $P[i][j] > 0$, if the data of target neighbor is 1 and current of that neighbor is not -1 (⊗), send an inversion request to that neighbor.

Table II
REDUCING THE SUM OF NEIGHBOR CURRENTS BY INVERTING VERTICAL NEIGHBORS

$\|\alpha\|$	Typical patterns	Inversion request to vertical neighbors
1	○ ⊙⊕○ ○	only one ○ → ⊗
	⊙ ○⊕○ ○	only one of these: ○ → ⊗ or ⊙ → ○
	⊗ ⊙⊕⊙ ○	○ → ⊗
	⊙ ⊗⊕⊙ ○	only one of these: ○ → ⊗ or ⊙ → ○
	⊙ ⊙⊕○ ⊗	⊙ → ○
	⊗ ⊙⊕○ ⊙	only one ⊙ → ○
2	⊙ ○⊕○ ⊙	⊙ → ○
	⊙ ⊙⊕○ ○	⊙ → ○ and ○ → ⊗
	○ ⊙⊕⊙ ○	○ → ⊗
	⊙ ⊙⊕⊗ ⊙	⊙ → ○
	⊙ ⊙⊕⊙ ⊗	⊙ → ○
3	⊙ ⊙⊕○ ⊙	⊙ → ○
	⊙ ⊙⊕⊙ ○	⊙ → ○ and ○ → ⊗
4	⊙ ⊙⊕⊙ ⊙	⊙ → ○

2) For a cell (i,j) with $P[i][j] < 0$, if the data of target neighbor is 0 and current of that neighbor is not 1 (⊙), send an inversion request to that neighbor.

The result of such decision is stored in two $N_R \times N_C$ matrices, called Request From Above (RFA) and Request From Below (RFB). The elements of these matrices are either 0 or 1. If $RFA[i][j]$ is 1, it means that the cell at (i,j) has received an inversion request from the cell above itself. RFB is defined similarly. RFA and RFB are initialized to 0 before any operation. Also, note that the first (last) row does not receive any requests from above (below).

2) Processing inversion requests: Once all inversion requests are submitted and cells with mutually exclusive requests are marked, the inversion decision is made. One possible technique for cell (i,j) is to grant such an inversion request only if $RFA[i][j] = 1$ and $RFB[i][j] = 1$. Since the top row and bottom row do not have any neighbors above and below them respectively, the first row of RFA and the last row of RFB are set to 1 in order to avoid decision conflicts with the proposed approach.

Once RFA and RFB are finalized, they are combined as an intention matrix $I^{\text{mat}} = \text{AND}(RFA, RFB)$. If $I^{\text{mat}}[i][j] = 1$, the data bit of cell (i,j) is marked for inversion. Since the final inversion is row-based rather than cell based, an intention vector I^{vec} (of size $N_R \times 1$) is constructed from I^{mat}.

If the number of 1's are greater than the number of 0's,

Algorithm 1 Summary of baseline algorithm

1: Take $\hat{D}_t$ and D_t as inputs
2: Construct C by $C = D_t - \hat{D}_{t-1}$
3: Construct matrix P by setting its $(i,j)^{th}$ element $P[i][j]$ equal to the algebraic sum of the currents of neighbors of TSV (i,j)
4: Initialize RFA and RFB to 0. Then set the first row of RFA and the last row of RFB to 1
5: **for** all i and j, $i \neq N_R$, set $RFB[i][j] = 1$ if the OR of following conditions are true **do**
6: $P[i+1][j] < 0$ and $D_t[i][j] == 0$
7: $P[i+1][j] > 0$ and $D_t[i][j] == 1$
8: **end for**
9: **for** all i and j, $i \neq 1$, set $RFA[i][j] = 1$ if at least one of the following conditions holds **do**
10: $P[i-1][j] < 0$ and $D_t[i][j] == 0$
11: $P[i-1][j] > 0$ and $D_t[i][j] == 1$
12: **end for**
13: $I^{\text{mat}} = \text{AND}(RFA, RFB)$
14: $I^{\text{vec}}[i] = \sum_{j=1}^{N_C} I^{\text{mat}}[i][j]$ for all $i \in \{1, \cdots, N_R\}$.
15: $I^{\text{vec}}[i] = \mathbf{1}(I^{\text{vec}}[i] \geq N_C/2)$, where the identity function **1** returns 1 when its argument is true, and returns 0 otherwise.
16: $\hat{D}_t[i][1...N_C] = \text{XOR}\big(I^{\text{vec}}[i], D_t[i][1...N_C]\big)$ for all i's

row i is selected to be inverted, and $I^{\text{vec}}[i]$ is set to 1. I^{vec} is initialized to zero.

C. An Example of baseline algorithm

Suppose that the previously transmitted data and the unmodified current data are:

$$\hat{D}_{t-1} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, D_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

If D_t is transmitted, the current matrix and the corresponding P matrix will be:

$$C = \begin{bmatrix} 1 & -1 & -1 & -1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} \odot & \otimes & \otimes & \otimes \\ \bigcirc & \bigcirc & \odot & \odot \\ \bigcirc & \bigcirc & \bigcirc & \bigcirc \\ \otimes & \odot & \odot & \bigcirc \end{bmatrix}$$

$$P = \begin{bmatrix} -1 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ -1 & 1 & 2 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

In this case, the number of cells with $P = 0$ through $P = 4$ is 6, 9, 1, 0, and 0 respectively. Now, suppose that the baseline algorithm is applied to $(D_t, \hat{D}_{t-1})$. RFA and RFB will hold the following values:

$$RFB = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, RFA = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Then, the I^{mat} and the corresponding I^{vec} are:

$$I^{\text{mat}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, I^{\text{vec}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Which results in the inversion of last row of D_t:

$$\hat{D}_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

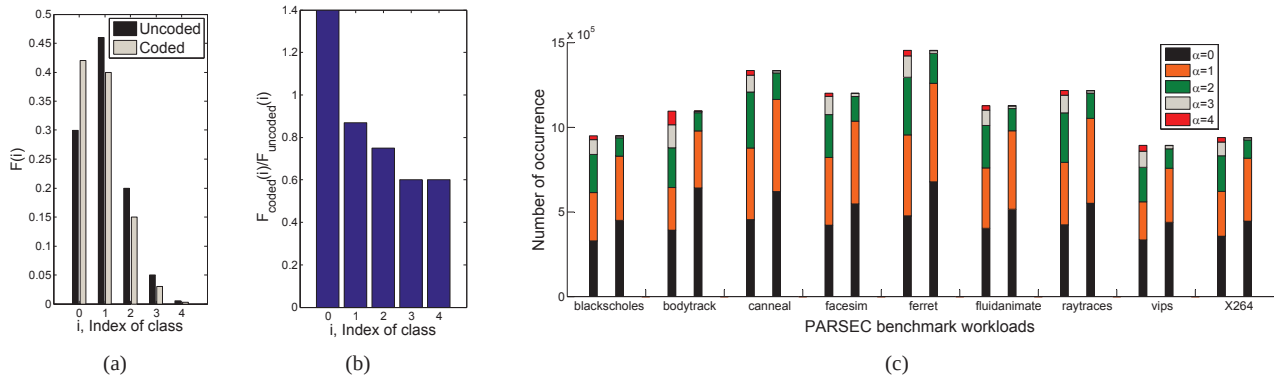
With this inversion, the new current and P matrix will be:

$$C = \begin{bmatrix} 1 & -1 & -1 & -1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, P = \begin{bmatrix} -1 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Note that after inversion, the count of cells with $P = 0$ through $P = 4$ is 11, 5, 0, 0, and 0 respectively: While the number of cells with $P = 0$ has grown from 6 to 11, the number of cells with $P = 1$ and $P = 2$ has dropped from 9 to 5 and from 1 to 0.

VI. BASELINE ALGORITHM EVALUATION

In order to evaluate the efficiency of the proposed code, a long random sequence of bits is input into two systems: one with encoder, and the other without encoder. Also to have a more robust evaluation of the baseline algorithm, PARSEC [37] benchmark memory traces captured using PIN [38] are utilized as a real-world data traffic. Denote the class of cells with $|P_{ij}| = 0, \cdots, 4$ by ψ_k for $k = 0, \cdots, 4$. Then, the relative occurrence frequency (denoting by $F(i)$) of all ψ_i's are counted for different i's. Comparing the occurrence frequency diagrams of the two systems shows whether the proposed coding lowers the occurrence of ψ_i's with larger i. If such a decline is observed, the result is a decrease in inductive coupling as discussed before. Fig. 5(a) and Fig. 5(c) show the relative occurrence of different ψ_is for an 8×8 TSV bus in both uncoded (left bar) and coded (right bar) system with random and PARSEC benchmark data traffic, respectively. In Fig. 5(a), there are 5 pairs of columns, where the left column of each column pair belongs to the uncoded system and the right column belongs to the coded system. As is evident from this figure, the relative frequency of occurrence of ψ_i's with large $i \geq 2$ has decreased. Furthermore, the relative frequency of occurrence of ψ_i remains almost the same. Importantly, the relative frequency of occurrence of ψ_0 has increased. Fig. 5(b) is the ratio of right columns to the left column of column pairs of Fig. 5(a) to emphasize the change of relative frequency of occurrence of different ψ_is.



(a) (b) (c)

Fig. 5. Evaluating the efficiency of baseline algorithm for an 8×8 TSV bus with both random (5(a) and 5(b)) and PARSEC (5(c)) data traffic. In 5(c), for each workload the left bar represents the uncoded and the right bar shows the coded approach results.

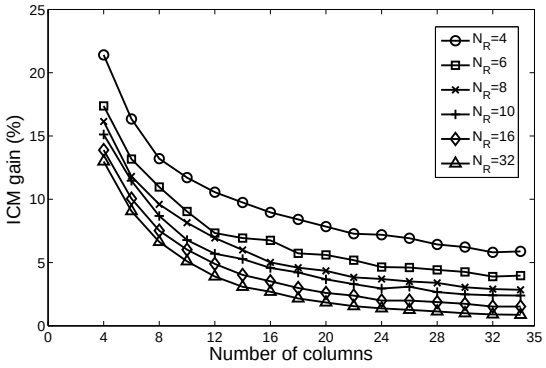


Fig. 6. ICM gain versus number of columns

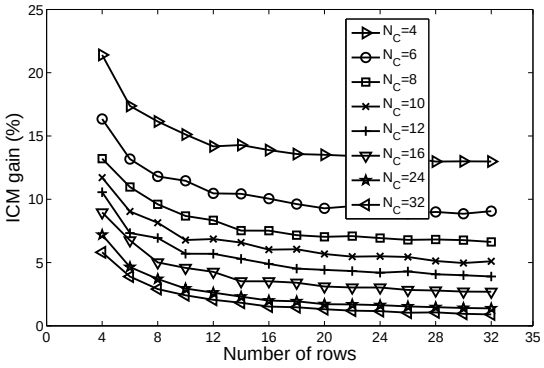


Fig. 7. ICM gain versus number of rows

A. Evaluation Metrics

Apart from the visual conclusion, it is possible to define a scalar measure to quantify the effectiveness of the proposed coding. One possibility is to form a weighted sum of relative frequencies, where the weight of the relative frequency of ψ_i is taken to be i to impose a penalty on high values of i. Consequently, the lower the value of this measure, the better is the efficiency. We define Inductive Coupling Mitigation (ICM) metric, as indicated by μ in Equation 11, to evaluate the efficiency of the proposed algorithm.

$$\mu = \sum_{i=0}^{4} i.F(i) \tag{11}$$

The "ICM gain" of the coding algorithm is also quantified as the ratio of "change in μ due to coding" to the "μ of uncoded system":

$$\nu = \frac{|\mu_{\text{uncoded}} - \mu_{\text{coded}}|}{\mu_{\text{uncoded}}} \tag{12}$$

Applying this measure to the result of Fig. 5 shows that the ICM measure changes from 0.99 to 0.78, which amounts to a 21% ICM gain.

Finally, the proposed coding imposes an overhead for data transmission. In order to transmit L_D bits over an $N_R \times N_C$ TSV matrix ($L_D = N_R \times N_C$), an additional N_R bits are required to transmit I^{vec} alongside the modified data. Thus, the overhead is written as:

$$\eta = \frac{N_R}{N = N_R N_C} = \frac{1}{N_C} \tag{13}$$

B. Scalability of baseline algorithm

It is of interest to analyze the variation of ICM gain ν with TSV bus dimensions N_R and N_C. This analysis provides an efficient physical distribution and placement of TSVs within a vault. For a given bus size ($N_R \times N_C$) and under certain constraints on ν and η, this information is used to decide on the values of N_R and N_C.

Fig. 6 and Fig. 7 show the gain improvement for the fixed number of rows (columns) as the number of columns (rows) grows. It is observed that increasing both N_R and N_C decreases ν; however, the negative effect of increasing N_C is more severe than the effect of increasing N_R. Fig. 8 illustrates an instance of this fact by comparing ν for the two cases of $N_R = 4$ and $N_C = 4$. It is observed that at each fixed bus size, the ICM gain is better when the number of columns has a fixed value.

In the following, the observed variation of ν with N_C and N_R are justified. A rigorous justification of the variation of ν with N_C and N_R requires calculating the values of $F(i)$ for $i = 0, \cdots, 4$ for the coded system. However, this calculation requires analyzing a very large scale

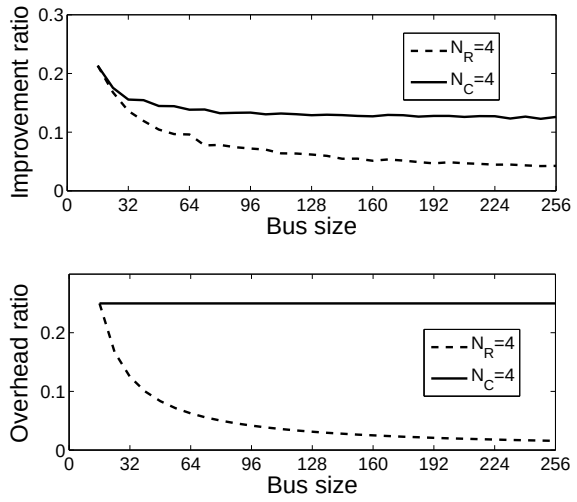


Fig. 8. ICM gain and corresponding information redundancy for the same number of bits in a bus with column or row growth

Markov chain which is outside the scope of this paper. Consequently, the indirect methods are used to justify the observed behavior.

1) Justification of N_C effect on ICM gain: An indirect approach for examining the impact of N_C on ν is to calculate the probability of row inversion. Given the algorithm, it is reasonable to assume that adjacent cells in a row are marked independently for inversion, i.e.:

$$\mathcal{P}(I^{\text{mat}}_{i,j} = 1 \,|\, \{I^{\text{mat}}_{i,n}, n \in \{1, \cdots, N_C\} \backslash \{j\}\}) \approx \mathcal{P}(I^{\text{mat}}_{i,j} = 1)$$

where $\mathcal{P}$ represents probability of its argument, "\" stands for set complement operator, and $I^{\text{mat}}_{i,j}$ denotes the element of I^{mat} at row i and column j . With the assumption of same inversion probability for all cells (data bits) $\mathcal{P}_{\text{inv,cell}}$, and given the fact that a row is inverted only when more than half of its cells are marked for inversion, the probability of row inversion is represented by Equation 14.

$$\mathcal{P}_{\text{inv,row}} = \sum_{i=\lceil N_c/2 \rceil}^{N_c} \binom{N_c}{i} (\mathcal{P}_{\text{inv,cell}})^i (1-\mathcal{P}_{\text{inv,cell}})^{N_c - i} \quad (14)$$

Fig. 9 shows $\mathcal{P}_{\text{inv,row}}$ as a function of N_C for different values of $\mathcal{P}_{\text{inv,cell}}$ as plot parameters. It is observed that when $\mathcal{P}_{\text{inv,cell}} < 0.5$, $\mathcal{P}_{\text{inv,row}}$ decreases with N_C. Combining this observation with the fact that $\mathcal{P}_{\text{inv,cell}} < 0.5$ for baseline algorithm (as shown in Appendix A), it is concluded that increasing N_C decreases $\mathcal{P}_{\text{inv,row}}$ within the proposed algorithm. As $\mathcal{P}_{\text{inv,row}}$ decreases, the code is less frequently employed, and this reluctance for engaging the inversion mechanism deprives the system of the ICM gain promised by coding. Consequently, the observed descending trend of ICM gain with the increase of N_C is justified.

In practice, those cells that are closer to the border and corners have higher probability of inversion, as discussed in Appendix A. To elaborate, $\mathcal{P}_{\text{inv,cell}}$ is a function of probability of 0's and 1's in the input bit stream and the

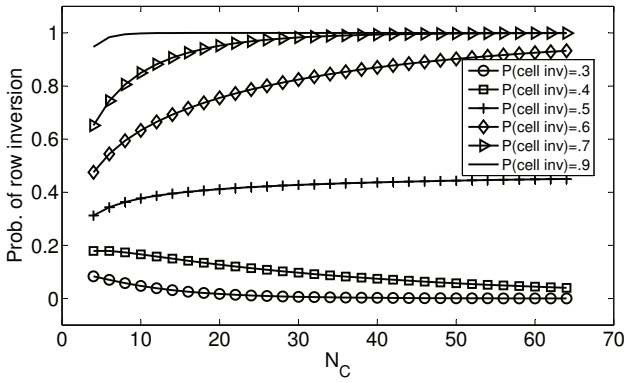


Fig. 9. Probability of row inversion versus number of cells in a row

location of the cell. Consequently, the calculation of row inversion probability in Equation 14 is not accurate enough. However, as long as the maximum cell inversion probability of the cells in a row is less than 0.5, the same descending trend of row inversion probability with N_C is observed (see Fig. 9).

2) Justification of N_R effect on ICM gain: The effect of N_R on ν is examined in a similar fashion. To see this, first note that based on the calculations of Appendix A, the highest probability of row inversion belongs to rows 1 and N_R, while rows 2 and $N_R - 1$ have lower probability of inversion, and the lowest probability belongs to rows $3, 4, \cdots, (N_R - 2)$. Denote these probabilities by $\mathcal{P}^{(1,N_R)}_{\text{inv,row}}$, $\mathcal{P}^{(2,N_R-1)}_{\text{inv,row}}$, and $\mathcal{P}^{(3\cdots N_R-2)}_{\text{inv,row}}$. Assuming that rows are inverted independently, the average of the ratio of inverted rows (for $N_R \geq 4$) to N_R is shown in Equation 15.

$$\overline{N_{\text{inv}}} = \frac{2}{N_R}\mathcal{P}^{(1,N_R)}_{\text{inv,row}} + \frac{2}{N_R}\mathcal{P}^{(2,N_R-1)}_{\text{inv,row}} + (1-\frac{4}{N_R})\mathcal{P}^{(3\cdots N_R-2)}_{\text{inv,row}} \quad (15)$$

By calculating the sensitivity of $\overline{N_{\text{inv}}}$ to N_R, i.e. $(d\overline{N_{\text{inv}}}/dN_R)/(\overline{N_{\text{inv}}})$, it is possible to indirectly justify the effect of N_R on ν:

$$\frac{d\overline{N_{\text{inv}}}/dN_R}{\overline{N_{\text{inv}}}} \quad \frac{\frac{-2}{N_R^2}\mathcal{P}^{(1,N_R)}_{\text{inv,row}} + \frac{-2}{N_R^2}\mathcal{P}^{(2,N_R-1)}_{\text{inv,row}} + \frac{4}{N_R^2}\mathcal{P}^{(3\cdots N_R-2)}_{\text{inv,row}}}{\frac{2}{N_R}\mathcal{P}^{(1,N_R)}_{\text{inv,row}} + \frac{2}{N_R}\mathcal{P}^{(2,N_R-1)}_{\text{inv,row}} + (1-\frac{4}{N_R})\mathcal{P}^{(3\cdots N_R-2)}_{\text{inv,row}}} = \frac{1}{N_R}\mathcal{O}(1) \quad (16)$$

where $\mathcal{O}(1)$ represents a scalar of order 1. For $N_R > 10$, this ratio is very small and keeps getting smaller for large N_R. This reduction in sensitivity means that the average fraction of inverted rows remains almost constant and the ICM gain does not decrease significantly. This justifies why increasing the number of rows does not have a noticeable impact on the ICM gain.

While the discussion so far suggests that a minimal value of N_C is beneficial as long as the only parameter of interest is the ICM gain ν, low values of N_C lead to a higher value of overhead, since the overhead is given by $1/N_C$.

VII. Enhanced Coding Algorithm

In some 3D IC devices, a large bus size and high ICM gain are required simultaneously. 3D stacked-DRAM is

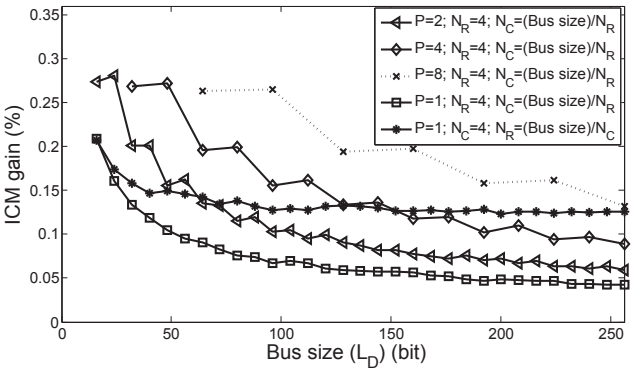


Fig. 10. ICM gain for the same bus size with different partitions (P) vs row (N_R) growth

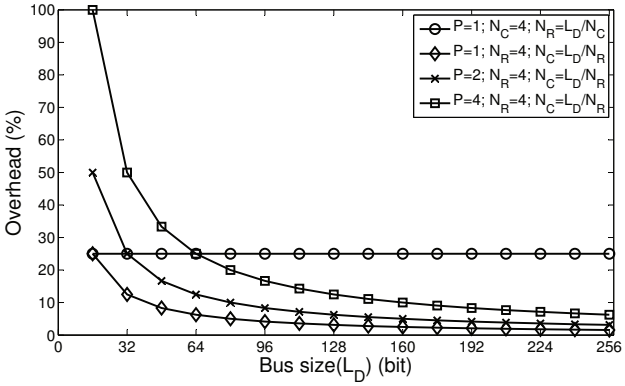


Fig. 11. Information redundancy overhead rate for the same bus size with different partitions (P) vs row (N_R) growth

an example, in which the size of TSV data bus typically exceeds hundred bits. As discussed before, the ICM gain drops as N_R or N_C increases. This means that the proposed coding is not scalable and may not be able to provide sufficient ICM gain for applications such as 3D stacked-DRAM with large bus sizes. A partitioning scheme is proposed in the following subsection to make the ICM gain of our coding scalable for larger bus size.

A. Partitioning Approach

Given that the ICM gain of baseline algorithm is significant for a small bus size, we propose an approach to make a large bus size gain-scalable by partitioning the large $N_R \times N_C$ matrix of TSVs into $q \times p$ sub matrices of size N_R/q and N_C/p and apply the coding on all sub matrices independently, resulting in an enhanced algorithm. With this method, the ICM gain of $N_R \times N_C$ network equal the ICM gain of an $(N_R/q) \times (N_C/p)$ network at the cost of increased overhead. Denoting the ICM gain and overhead of an $m \times n$ with a $q \times p$ partitioning, the new $\nu(m,n,q,p)$ and $\eta(m,n,q,p)$ are presented by Equation 17:

$$\begin{aligned} \nu(m,n,q,p) &= \nu(\frac{m}{q},\frac{n}{p},1,1) \\ \eta(m,n,p,q) &= \frac{p.q.\frac{N_R}{q}}{N_R.N_C} = \frac{p}{N_C} = p.\eta(m,n,1,1) \end{aligned} \quad (17)$$

A column partitioning is considered here for the following reasons, i.e $q = 1$, instead of row or row-column partitioning. First, the impact of N_C on ν is more severe than the effect of N_R on ν as discussed in Section VI-B. Therefore, performing partitioning on columns is preferable. Second, row partitioning does not offer substantial gain improvement because of the insensitivity of ν to N_R. In column partitioning approach, the same ICM is obtained as reported for a small matrix while the overhead is the same as that of increasing rows.

A general benefit of partitioning is to make parallel implementation possible, resulting in a faster encoder architecture. The factor of parallelism is equal to the number of partitions.

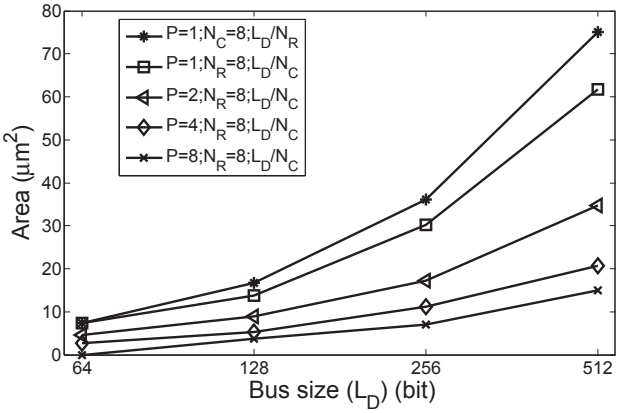


Fig. 12. Encoder silicon area

B. Enhanced Algorithm Evaluation

The ICM gain and overhead of enhanced algorithm and baseline algorithm are compared in Fig. 10 and Fig. 11. The row scaling approach (constant N_C) offers a consistent ICM gain but suffers from a constant high value of overhead regardless of the bus size. On the other hand, the column scaling approach (constant N_R) improves overhead as bus size increases, but causes a severe drop in the ICM gain. The enhanced algorithm combines the high ICM gain of row-scaling approach with the low overhead of column scaling approach.

C. Hardware Synthesis Results

The encoder of both proposed coding algorithms is synthesized by Synopsys Design Compiler using 28nm TSMC library (1.05V, $25\,^{\circ}\mathrm{C}$) to report latency, power consumption, and area. Decoder component is the same for

Table III
PROPOSED ENCODER LATENCY VERSUS TSV BUS SIZE (ps)

Bus size	N_C=**8;**N_R=L_D/N_C	N_R=**8;**N_C=L_D/N_R			
	P=1	P=1	P=2	P=4	P=8
128-bit	144	207	147	112	97
256-bit	203	295	207	144	118
512-bit	639	283	219	221	151

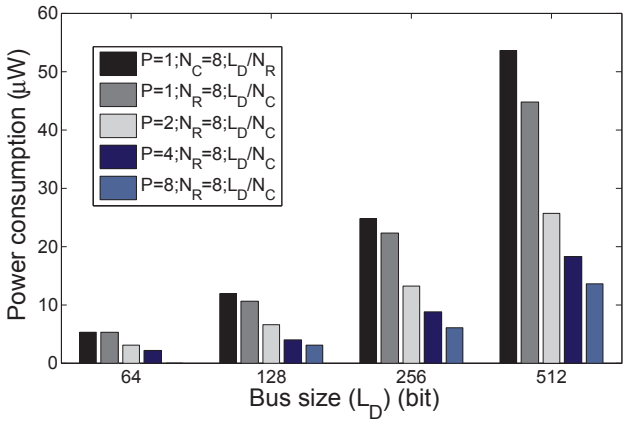


Fig. 13. Proposed encoder power consumption versus TSV bus size

both of the proposed baseline and enhanced algorithms. The overhead is also negligible as compared to the encoder unit in terms of latency, power consumption, and area, since it is composed of a simple comparator and a mix of inverter gates.

A TSV data bus with variable size is also modeled in HSPICE. The TSV model in [10] is used to capture the inductive TSV-to-TSV coupling effect.

At each step of hardware implementation, inversion requests of a single TSV row are processed. Therefore, the latency of the encoder is constrained by TSV row size. The latency of row scaling is observed to be higher than the latency of column scaling as illustrated in Table III. In partitioning method, the number of columns in a partition decreases as the number of partitions increases, reducing the overall processing latency. This is because all the smaller partitions are evaluated in parallel. For example for a 512 TSV bundle the latency of encoder with 8 partitions is almost 50% of the one with one partition.

A larger bus size needs more complex combinational logic and memory units, resulting in higher power consumption and area footprint as depicted in Fig. 12 and Fig. 13. However, partitioning the columns of TSVs, the hardware complexity declines as fewer net wirings and logic components are needed. Fig. 12 demonstrates the decreasing trend in area for the same bus size as the partitioning factor grows. A TSV with 512 bits and 8 partitions occupies almost 20% of the area for the same bandwidth with one partition, as reported in Fig. 12. For example, the power consumption of encoder in 512 bundle of TSVs with 8 partitions is almost 20% of the same bandwidth with one partition, as shown in Fig. 13. Applications with more number of partitions have less power consumption due to hardware size reduction.

VIII. Related work

Exascale systems will experience different kinds of faults according to the current knowledge of existing supercomputers. The power consumption budget for the future exascale computer systems are dedicated to memory subsystem. 3D-stacking DRAM memories have been suggested since current DRAMs will not meet the expected power consumption budget. A system-level design methodology for scalable fault-tolerance of distributed on-chip memories in NoCs has been introduced in [46]. The effects of transient faults are examined in synchronous and asynchronous 2D NoC [41]. An efficient Built-In-Self-Repair (BISR) algorithm to fulfil the test and reliability needs for 3D-stacked memories have been proposed [39]. The effects of temperature, refresh period, and ECC policy on the reliability and power consumption of 3D-stacked embedded DRAM are examined to minimize the energy consumption without violating error rate limitation [40]. 3D DRAM circuits include a large number of TSVs, which are prone to open defects and coupling noises. The faulty behavior of TSV open defects occurred on the wordlines and the bitlines of 3D DRAM circuits are modeled in [42].

It is anticipated the existing resilient approaches, relying on automatic or application level checkpoint-restart, will not be practical as check pointing and restarting time will exceed the mean time to failure of a full system [43]. The impact of TSVs on SI in 3D ICs has been considered in several articles [12], [13]. Analytical model for the coupling capacitance between pairs of TSVs is also reported in [12]. A complete set of self-consistent equations including self and coupling terms for resistance, capacitance and inductance of various TSV structures are presented in [13]. Five solutions are suggested in [10] to reduce the coupling including: increasing TSV distances, shielding the victim TSVs, insert buffers at the victim net, decreasing the driver size at the aggressor net, and increasing the load at both victim and aggressor net. The last two suggestions have negative implications for timing performance, and others need high effort at post-design time. Many different coding techniques have been presented to avoid the crosstalk issue among communication links [44], [45]. The specification of TSVs is also totally different from wires on a 2D chip as described in Section III. A coding scheme has been suggested for a matrix of TSVs, reducing the maximum crosstalk by 25%. This approach is only applicable for capacitive crosstalk (coupling). Also this method is not scalable and supports a mesh of TSV with size of $3 \times n$, limiting the TSV insertion process. It imposes around 40% information redundancy with an encoder and decoder of quadratic complexity in circuit area [15]. However, the impact of inductive coupling on SI which is more important in higher frequencies rather capacitance coupling [35], has not been evaluated yet.

IX. Conclusion

Although 3D multiple-stacked IC is a promising solution for exascale computing, its vulnerability to inductive TSV-to-TSV coupling has not been extensively studied. A coding algorithm is proposed to mitigate inductive TSV-to-TSV coupling after characterizing such issue by modifying input data stream. The ICM of the algorithm is then gauged in terms of various metrics such as mitigation measure, ICM gain, and data overhead. It is observed that the

algorithm provides a significant ICM gain for relatively small bus sizes, while it suffers from a descending trend in performance as bus size increases.

With large bus size applications such as 3D multiple-stacked IC, a partitioning approach is added to the algorithm to make it scalable with bus size. At the cost of reasonable overhead, significant ICM gain is obtained even in the case of large bus sizes. In addition to ICM gain and overhead, other practical issues such as power consumption and silicon area are also reported. It is observed that the coding approach promises lower power consumption and silicon area while providing considerable ICM gain for large bus sizes.

REFERENCES

[1] T. A. S. on Exascale Computing, "Report on exascale computing," 2010.

[2] "The 43rd top500 list," http://www.top500.org/lists/2014/06/, 2014.

[3] K. Yoon, G. Kim, W. Lee, T. Song, J. Lee, H. Lee, K. Park, and J. Kim, "Modeling and analysis of coupling between tsvs, metal, and rdl interconnects in tsv-based 3d ic with silicon interposer," in *Electronics Packaging Technology Conference. EPTC '09. 11th*, 2009, pp. 702–706.

[4] D. Henry, S. Cheramy, J. Charbonnier, P. Chausse, M. Neyret, G. Garnier, C. Brunet-Manquat, S. Verrun, N. Sillon, L. Bonnot, A. Farcy, L. Cadix, M. Rousseau, and E. Saugier, "Development and characterisation of high electrical performances tsv for 3d applications," in *Electronics Packaging Technology Conference. EPTC '09. 11th*, 2009, pp. 528–535.

[5] M. Jung, J. Mitra, D. Z. Pan, and S. K. Lim, "Tsv stress-aware full-chip mechanical reliability analysis and optimization for 3d ic," *Commun. ACM*, vol. 57, no. 1, pp. 107–115, Jan. 2014.

[6] "Reliability of {TSV} interconnects: Electromigration, thermal cycling, and impact on above metal level dielectric," *Microelectronics Reliability*, vol. 53, no. 1, pp. 17 – 29, 2013.

[7] A. Shayan, X. Hu, H. Peng, C.-K. Cheng, W. Yu, M. Popovich, T. Toms, and X. Chen, "Reliability aware through silicon via planning for 3d stacked ics," in *Design, Automation Test in Europe Conference Exhibition, 2009. DATE '09.*, April 2009, pp. 288–291.

[8] D. Pan, S.-K. Lim, K. Athikulwongse, M. Jung, J. Mitra, J. Pak, M. Pathak, and J. seok Yang, "Design for manufacturability and reliability for tsv-based 3d ics," in *Design Automation Conference (ASP-DAC), 2012 17th Asia and South Pacific*, Jan 2012, pp. 750–755.

[9] S. Itr, "ITRS 2012 Executive Summary," ITRS.

[10] 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 *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, 2011, pp. 783–788.

[11] L. T. B. Wu, X. Gu and M. Ritter, "Electromagnetic modeling of massively coupled through silicon vias for 3d interconnects," in *Microwave and Optical Technology Letters*, 2011, p. 12041206.

[12] I. Savidis and E. Friedman, "Closed-form expressions of 3-d via resistance, inductance, and capacitance," *Electron Devices, IEEE Transactions on*, vol. 56, no. 9, pp. 1873–1881, 2009.

[13] 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*, 2009, pp. 1–8.

[14] C. Metzler, A. Todri-Sanial, A. Bosio, L. Dilillo, P. Girard, A. Virazel, P. Vivet, and M. Belleville, "Computing detection probability of delay defects in signal line tsvs," in *Test Symposium (ETS), 18th IEEE European*, 2013, pp. 1–6.

[15] R. Kumar and S. P. Khatri, "Crosstalk avoidance codes for 3d vlsi," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*, 2013, pp. 1673–1678.

[16] A. Eghbal, P. M. Yaghini, S. S. Yazdi, and N. Bagherzadeh, "Tsv-to-tsv inductive coupling-aware coding scheme for 3d network-on-chip," in *Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2014 IEEE International Symposium on*, Oct 2014, pp. 92–97.

[17] T. Xu, P. Liljeberg, and H. Tenhunen, "A study of through silicon via impact to 3d network-on-chip design," in *Electronics and Information Engineering (ICEIE), 2010 International Conference On*, vol. 1, Aug 2010, pp. V1–333–V1–337.

[18] S. Kumar and R. Van Leuken, "A 3d network-on-chip for stacked-die transactional chip multiprocessors using through silicon vias," in *Design Technology of Integrated Systems in Nanoscale Era (DTIS), 2011 6th International Conference on*, April 2011, pp. 1–6.

[19] J. Lee, D. Lee, S. Kim, and K. Choi, "Deflection routing in 3d network-on-chip with tsv serialization," in *Design Automation Conference (ASP-DAC), 2013 18th Asia and South Pacific*, Jan 2013, pp. 29–34.

[20] J. Jeddeloh and B. Keeth, "Hybrid memory cube new dram architecture increases density and performance," in *VLSI Technology (VLSIT), 2012 Symposium on*, June 2012, pp. 87–88.

[21] J. T. Pawlowski, "Hybrid memory cube (hmc)," in *HOT-CHIPS*, 2011.

[22] "Hybrid memory cube," http://www.micron.com/products/hybrid-memory-cube.

[23] F. Dubois, A. Sheibanyrad, F. Petrot, and M. Bahmani, "Elevator-first: A deadlock-free distributed routing algorithm for vertically partially connected 3d-nocs," *Computers, IEEE Transactions on*, vol. 62, no. 3, pp. 609–615, 2013.

[24] Y. Cheng, L. Zhang, Y. Han, and X. Li, "Thermal-constrained task allocation for interconnect energy reduction in 3-d homogeneous mpsocs," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 21, no. 2, pp. 239–249, 2013.

[25] V. Pasca, L. Anghel, C. Rusu, and M. Benabdenbi, "Configurable serial fault-tolerant link for communication in 3d integrated systems," in *On-Line Testing Symposium (IOLTS), 2010 IEEE 16th International*, 2010, pp. 115–120.

[26] K. Tu, "Reliability challenges in 3d ic packaging technology," *Microelectronics Reliability*, vol. 51, no. 3, pp. 517 – 523, 2011.

[27] S. Itr, "Table intc7 - itrs 2013 executive summary," ITRS.

[28] U. Tida, R. Yang, C. Zhuo, and Y. Shi, "On the efficacy of through-silicon-via inductors," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[29] D. H. Kim, K. Athikulwongse, M. Healy, M. Hossain, M. Jung, I. Khorosh, G. Kumar, Y.-J. Lee, D. Lewis, T.-W. Lin, C. Liu, S. Panth, M. Pathak, M. Ren, G. Shen, T. Song, D. H. Woo, X. Zhao, J. Kim, H. Choi, G. Loh, H.-H. Lee, and S. K. Lim, "3d-maps: 3d massively parallel processor with stacked memory," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, Feb 2012, pp. 188–190.

[30] D. H. Kim and S.-K. Lim, "Design quality trade-off studies for 3-d ics built with sub-micron tsvs and future devices," *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, vol. 2, no. 2, pp. 240–248, 2012.

[31] K. Salah, H. Ragai, Y. Ismail, and A. El Rouby, "Equivalent lumped element models for various n-port through silicon vias networks," in *Design Automation Conference (ASP-DAC), 16th Asia and South Pacific*, 2011, pp. 176–183.

[32] B. Wu, X. Gu, L. Tsang, and M. B. Ritter, "Electromagnetic modeling of massively coupled through silicon vias for 3d interconnects," *Microwave and Optical Technology Letters*, vol. 53, no. 6, pp. 1204–1206, 2011.

[33] J. e. Warnock, "A 5.2ghz microprocessor chip for the ibm zenterprise system," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, 2011, pp. 70–72.

[34] G. Katti, M. Stucchi, K. De Meyer, and W. Dehaene, "Electrical modeling and characterization of through silicon via for three-dimensional ics," *Electron Devices, IEEE Transactions on*, vol. 57, no. 1, pp. 256 –262, jan. 2010.

[35] A. Todri, S. Kundu, P. Girard, A. Bosio, L. Dilillo, and A. Virazel, "A study of tapered 3-d tsvs for power and thermal integrity," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 21, no. 2, pp. 306–319, 2013.

[36] PTM, "Predictive Technology Model," ptm.asu.edu.

[37] C. Bienia and K. Li, "Parsec 2.0: A new benchmark suite for chip-multiprocessors," in *Proceedings of the 5th Annual Workshop on Modeling, Benchmarking and Simulation*, June 2009.

[38] Intel-cooperation, "Pin - A Dynamic Binary Instrumentation Tool," https://software.intel.com/en-us/articles/pin-a-dynamic-binary-instrumentation-tool.

[39] X. Wang, D. Vasudevan, and H.-H. Lee, "Global built-in self-repair for 3d memories with redundancy sharing and parallel testing," in *3D*

Systems Integration Conference (3DIC), 2011 IEEE International, 2012, pp. 1–8.
[40] W. Yun, K. Kang, and C.-M. Kyung, "Thermal-aware energy minimization of 3d-stacked l3 cache with error rate limitation," in *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, 2011, pp. 1672–1675.
[41] P. M. Yaghini, A. Eghbal, H. Pedram, and H. R. Zarandi, "Investigation of transient fault effects in synchronous and asynchronous network on chip router," *J. Syst. Archit.*, vol. 57, no. 1, pp. 61–68, Jan. 2011.
[42] L. Jiang, Y. Liu, L. Duan, Y. Xie, and Q. Xu, "Modeling tsv open defects in 3d-stacked dram," in *Test Conference (ITC), 2010 IEEE International*, 2010, pp. 1–9.
[43] F. Cappello, A. Geist, B. Gropp, L. Kale, B. Kramer, and M. Snir, "Toward exascale resilience," *Int. J. High Perform. Comput. Appl.*, vol. 23, no. 4, pp. 374–388, Nov. 2009.
[44] B. Wang, J. Xie, and Q. Wang, "Crosstalk-aware channel transmitting scheme for error resilience noc interconnects," in *Electronics, Communications and Control (ICECC), 2011 International Conference on*, 2011, pp. 190–193.
[45] M. Shafaei, A. Patooghy, and S. Miremadi, "Numeral-based crosstalk avoidance coding to reliable noc design," in *Digital System Design (DSD), 2011 14th Euromicro Conference on*, 2011, pp. 55–62.
[46] A. BanaiyanMofrad, N. Dutt, and G. Girão, "Modeling and analysis of fault-tolerant distributed memories for networks-on-chip," in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE '13, 2013, pp. 1605–1608.

Pooria M.Yaghini is a Ph.D. candidate in department of electronic engineering and computer science at University of California, Irvine. He has received his B.S. in computer hardware Engineering from Sadjad University in 2006 and his M.S. degree in computer engineering from Amirkabir University of Technology (Tehran Polytechnic) in 2009. He is also a recipient of 2011 Henry Samueli Endowed Fellowship. His research interests include 3D IC design, on-chip networks, low-power design, reliability analysis and verification, and fault-tolerant architecture design.

Ashkan Eghbal received the B.Sc. degree in Computer Hardware Engineering from Azad University Central Tehran Branch, Tehran, Iran, in 2007, and the M.Sc. in Advanced Computer Architecture from Amirkabir University of Technology, Tehran, Iran, in 2010, respectively. He is currently pursuing PhD degree in Computer System and Software with University of California, Irvine. He has been with the Advanced Computer Architecture Group, UCI, since 2010 as a research and teacher assistant. His current research interests include the Reliability analysis, On-Chip Interconnection Network, 3D Stack Architectures, Fault-tolerant design, and Energy Efficient Embedded Systems.

Misagh Khayambashi is a PhD student in the Department of Electrical Engineering at the University of California Irvine. He completed his B.Sc. at Isfahan University of Technology, Iran. His research interests lie in the area of statistical and deterministic signal processing, system modeling and identification, compressive sensing, and estimation theory.

Nader Bagherzadeh is a professor of computer engineering in the department of electrical engineering and computer science at the University of California, Irvine, where he served as a chair from 1998 to 2003. Dr Bagherzadeh has been involved in research and development in the areas of: computer architecture, reconfigurable computing, VLSI chip design, network-on-chip, 3D chips, sensor networks, and computer graphics since he received a Ph.D. degree from the University of Texas at Austin in 1987. He is a Fellow of the IEEE. Professor Bagherzadeh has published more than 200 articles in peer-reviewed journals and conferences. He has trained hundreds of students who have assumed key positions in software and computer systems design companies in the past twenty years. He has been a PI or Co-PI on more than $8 million worth of research grants for developing next generation computer systems for applications in general purpose computing and digital signal processing.

APPENDIX
PROOF OF PROBABILITY OF BIT INVERSION

In this section, the indexing $<>[i][j]$ is replaced with $<>_{i,j}$ for notational simplicity. Also, RFA and RFB are replaced with A and B.

To calculate the probability of a cell being marked for inversion, i.e. $\mathcal{P}(I_{i,j}=1)$, the decision making mechanism in the algorithm should be analyzed. From the algorithm, the following expression should be calculated:

$$\mathcal{P}(I_{i,j}=1)=\mathcal{P}(A_{i,j}=1 \text{ and } B_{i,j}=1) \tag{18}$$

Depending on the value of (i,j), the calculation of this expression takes different forms:

A. $i \in \{3,\cdots,N_R-2\}$ and $j \in \{2,\cdots,N_C-1\}$

Based on the algorithms and the relation between data value and current directions:

$$\begin{aligned}\mathcal{P}(I_{i,j}=1)=&\\ &\mathcal{P}(d_{i,j}=1)\times\Big[\\ &\mathcal{P}(c_{i,j}=1|d_{i,j}=1)\times\\ &\mathcal{P}(P_{i-1,j} \text{ and } P_{i+1,j}>0|c_{i,j}=1)\\ &+\\ &\mathcal{P}(c_{i,j}=0|d_{i,j}=1)\times\\ &\mathcal{P}(P_{i-1,j} \text{ and } P_{i+1,j}>0|c_{i,j}=0)\Big]\\ &+\\ &\mathcal{P}(d_{i,j}=0)\times\Big[\\ &\mathcal{P}(c_{i,j}=-1|d_{i,j}=0)\times\\ &\mathcal{P}(P_{i-1,j} \text{ and } P_{i+1,j}<0|c_{i,j}=-1)\\ &+\\ &\mathcal{P}(c_{i,j}=0|d_{i,j}=0)\times\\ &\mathcal{P}(P_{i-1,j} \text{ and } P_{i+1,j}<0|c_{i,j}=0)\Big]\end{aligned} \tag{19}$$

and also

$$\begin{aligned}&\mathcal{P}(P_{i-1,j} \text{ and } P_{i+1,j}>0|c_{i,j}=1)=\\ &\mathcal{P}(c_{i-1,j-1}+c_{i-1,j+1}+c_{i-2,j}>-1)\times\\ &\mathcal{P}(c_{i+1,j+1}+c_{i+1,j-1}+c_{i+2,j}>-1)\end{aligned} \tag{20}$$

which is calculated by counting all the possibilities and considering $\mathcal{P}(c=0)=1/2$ ($0\to0$ or $1\to1$), $\mathcal{P}(c=1)=1/4$ ($0\to1$), and $\mathcal{P}(c=-1)=1/4$ ($1\to0$). Putting it altogether:

$$\begin{aligned}\mathcal{P}(I_{i,j}=1) &= .5\Big(.5(\tfrac{11}{32})(\tfrac{11}{32})+.5(\tfrac{21}{32})(\tfrac{21}{32})\Big)+\\ &\quad .5\Big(.5(\tfrac{11}{32})(\tfrac{11}{32})+.5(\tfrac{21}{32})(\tfrac{21}{32})\Big)\\ &= \tfrac{281}{1024}\end{aligned} \tag{21}$$

B. $i \in \{2,N_R-1\}$ and $j \in \{2,\cdots,N_C-1\}$

The calculation is the same, except for that when $i=2(N_R-1)$, $c_{i-2,j}(c_{i+2,j})$ does not enter the calculations because its index is out of range. We will have:

$$\begin{aligned}\mathcal{P}(I_{i,j}=1) &= .5\Big(.5(\tfrac{11}{32})(\tfrac{5}{16})+.5(\tfrac{21}{32})(\tfrac{11}{16})\Big)+\\ &\quad .5\Big(.5(\tfrac{11}{32})(\tfrac{5}{16})+.5(\tfrac{21}{32})(\tfrac{11}{16})\Big)\\ &= \tfrac{143}{512}\end{aligned} \tag{22}$$

C. $i \in \{1,N_R\}$ and $j \in \{2,\cdots,N_C-1\}$

When $i=1(N_R)$, $RFA(RFB)$ is always set to 1 and all the probabilities dependent on values of currents at the above (below) rows simply drop:

$$\begin{aligned}\mathcal{P}(I_{i,j}=1) &= .5\Big(.5(\tfrac{11}{32})+.5(\tfrac{21}{32})\Big)+\\ &\quad .5\Big(.5(\tfrac{11}{32})+.5(\tfrac{21}{32})\Big)\\ &= \tfrac{1}{2}\end{aligned} \tag{23}$$

D. $i \in \{3,\cdots,N_R-2\}$ and $j \in \{1,N_C\}$

When $j=1(N_c)$, $c_{*,j-1}(c_{*,j+1})$ can be eliminated from calculations:

$$\begin{aligned}\mathcal{P}(I_{i,j}=1) &= .5\Big(.5(\tfrac{5}{16})(\tfrac{5}{16})+.5(\tfrac{11}{16})(\tfrac{11}{16})\Big)+\\ &\quad .5\Big(.5(\tfrac{5}{16})(\tfrac{5}{16})+.5(\tfrac{11}{16})(\tfrac{11}{16})\Big)\\ &= \tfrac{73}{256}\end{aligned} \tag{24}$$

E. $i \in \{2,N_R-1\}$ and $j \in \{1,N_C\}$

When $(i,j)=(2,1)$, only $c_{i-1,j+1}$, $c_{i+1,j+1}$, and $c_{i+2,j}$ contribute to the calculation:

$$\begin{aligned}\mathcal{P}(I_{i,j}=1) &= .5\Big(.5(\tfrac{1}{4})(\tfrac{5}{16})+.5(\tfrac{3}{4})(\tfrac{11}{16})\Big)+\\ &\quad .5\Big(.5(\tfrac{1}{4})(\tfrac{5}{16})+.5(\tfrac{3}{4})(\tfrac{11}{16})\Big)\\ &= \tfrac{19}{64}\end{aligned} \tag{25}$$

The other 3 possibilities for (i,j) are similar.

F. $i \in \{1,N_R\}$ and $j \in \{1,N_C\}$

When $(i,j)=(1,1)$, only $c_{i+1,j+1}$ and $c_{i+2,j}$ contribute to the calculation and RFA can be neglected:

$$\begin{aligned}\mathcal{P}(I_{i,j}=1) &= .5\Big(.5(\tfrac{5}{16})+.5(\tfrac{11}{16})\Big)+\\ &\quad .5\Big(.5(\tfrac{5}{16})+.5(\tfrac{11}{16})\Big)\\ &= \tfrac{1}{2}\end{aligned} \tag{26}$$

The other 3 possibilities for (i,j) are similar.