Edge Effect Aware Crosstalk Avoidance Technique for 3D Integration

Lennart Bamberg, Amir Najafi and Alberto García-Ortiz
Institute of Electrodynamics and Microelectronics (ITEM.ids), University of Bremen
{bamberg, amir.najafi, agarcia}@item.uni-bremen.de

Abstract—3D integration is one of the most promising solutions for the scaling of future integrated circuits (ICs). Nevertheless, the 2D metal wires and 3D through silicon vias (TSVs) are frequently performance bottlenecks of 3D ICs, due to their high capacitive crosstalk, which can be reduced by a coding approach. In this work we show that existing TSV crosstalk avoidance codes (CACs) are impractical for real applications due to the edge effects in TSV arrays. Additionally, these 3D CACs do not reduce the metal wire crosstalk. This work presents a crosstalk avoidance approach for 3D ICs which overcomes both limitations. The method remaps the bits of existing 2D CACs onto a TSV array in a way that results in the minimum possible TSV crosstalk. Experimental results, obtained by circuit simulations in combination with an electromagnetic field solver, show that the presented technique can reduce the crosstalk of TSVs and metal wires by about 30 % and 50 %, respectively. In comparison, with higher hardware costs, existing 3D CACs only reduce the TSV crosstalk by a maximum of 11.45 %, while providing no optimization of the metal wire crosstalk.

I. INTRODUCTION

The 3D integration is one of the promising solutions to overcome the challenges that arise with the limit of Moore’s law. As the interconnect structure between the dies of a 3D system on chip (3D SoC), through silicon vias (TSVs) are typically used as they yield to a high reliability [1]. TSVs are usually bundled together, rather than used in isolation [2]. By using regular TSV arrays, it is possible to generate wide I/O 3D components, such as stacked DRAM cells [3].

However, TSV bundles, as well as metal wires, suffer from crosstalk which is a threat to the delay, the power consumption and the signal integrity [4]. In recent years, crosstalk became the critical design issue for the traditional planar metal wire interconnects, due to their limited scaling [5]. Previous research shows that the crosstalk problem is not alleviated for TSV interconnects due to the relatively large TSV dimensions and the increased number of aggressors compared to the traditional planar metal wires [6]–[8]. Thus, crosstalk is still an important design concern for 3D integrated circuit (3DIC) design.

Crosstalk avoidance techniques have been widely studied for planar metal wires [4], [9]. For metal wires, only two adjacent aggressors need to be considered while TSVs have a maximum of eight adjacent aggressors. Generally, this makes existing metal wire crosstalk avoidance techniques not directly applicable for TSV arrays.

Since crosstalk is a pattern dependent phenomena, the most promising crosstalk avoidance technique is coding [4]. Kumar et al. [10], Zou et al. [11] and Cui et al. [12] derive crosstalk avoidance coding (CAC) approaches which keep the crosstalk of each TSV in the middle of an array below a certain level. None of these methods ever analyzed the crosstalk of the TSVs, which are located at the edges of an array. Previous works claim that, due to reduced number of adjacent aggressors, the crosstalk of these edge TSVs is significantly lower than the crosstalk of the middle TSVs. However, this assumption is wrong. Due to the E-field sharing effect [8], [13], the coupling between two edge TSVs is stronger than in the middle of the array. Hence, the crosstalk of edge TSVs is only slightly lower than the crosstalk of middle TSVs, as we will show in this work. This fact heavily reduces the coding efficiency of existing 3D CACs. This work includes a reanalysis of existing TSV CACs by means of circuit simulations in combination with electromagnetic (EM) parasitic extractions. For a modern TSV array, the actual crosstalk reductions of the CACs are only about 50 % of the previously reported values. The lower coding gain in combination with their high overhead makes existing 3D CACs impractical for real applications. Thus, an efficient coding method needs to reduce the crosstalk of the middle and the edge TSVs simultaneously. The second limitation of existing CACs for 3D SoCs is that they only consider the crosstalk of the TSVs. However, planar metal wires are not absent in 3D integration and their crosstalk is not negligible.

To overcome both limitations, we focus on existing 2D CACs and present a solution to make them suitable for TSV arrays by finding the optimal mapping under consideration of the edge effects. Thereby, the CACs retain their efficiency for the planar metal wires. The experimental results show that for all analyzed 2D CACs our method outperforms all existing 3D CACs. Compared to the latest presented 3D CAC [12], we measured an improvement in the maximum TSV delay reduction by over 228 % while the coding overhead (bit and circuit) is simultaneously reduced.

The remainder of this paper is organized as follows. In Section II a crosstalk classification considering the edge effects is presented. The limitations of previous 3D CACs are outlined in Section III. Subsequently, in Section IV, the proposed 3D CAC is discussed in detail and Section V includes the experimental results. Finally, a conclusion is drawn.

II. CROSSTALK NOISE INCLUDING EDGE EFFECTS

The capacitive coupling effects are the dominant crosstalk source of the planar metal wires and the vertical TSVs
[4], [14]. Consequently, only the crosstalk noise due to the capacitive coupling is considered in the crosstalk classification presented in this paper.

The crosstalk noise of an interconnect $i$ in clock cycle $k$ is quantified by the effective capacitance to be reloaded [4]:

$$C_{eff,i}[k] = \Delta b_i^2[k]C_{i,0} + \sum_{j \neq i} \delta_{i,j}[k]C_{i,j}.$$  \hspace{1cm} (1)

Here $C_{i,0}$ is the self/ground-capacitance of interconnect $i$ and $C_{i,j}$ is the coupling-capacitance between the interconnects $i$ and $j$. $\Delta b_i$ determines the self-switching of interconnect $i$:

$$\Delta b_i[k] = b_i[k] - b_i[k - 1],$$  \hspace{1cm} (2)

which is either $-1$, $0$, or $1$. Here, $b_i[k]$ is the logical binary value of interconnect $i$ for the $k^{th}$ clock cycle. $\delta_{i,j}$ determines the crosstalk switching between the interconnects $i$ and $j$:

$$\delta_{i,j}[k] = (\Delta b_i^2[k] - \Delta b_i[k]\Delta b_j[k]).$$  \hspace{1cm} (3)

$\delta_{i,j}$ is equal to 2, if reverse signal transitions occur on the interconnects $i$ and $j$ (e.g. $i$ switches from logical 0 to 1, while $j$ switches from logical 1 to 0); if only interconnect $i$ changes its signal level, $\delta_{i,j}$ is equal to 1, otherwise it is 0.

In the following we present simple, scalable and universal capacitance models for the metal wires and the TSVs. The combination of these models with Eq. 1 results in the crosstalk classification. The model for the metal wires is depicted in Fig. 1-a. Between every adjacent metal wire pair exists a coupling-capacitance $C_c$ and every metal wire has a capacitance to the grounded substrate contacts $C_g$ [5]. The self-capacitance of a metal wire $C_{0,2D}$ is equal to the sum of $C_g$ and the interconnects load capacitance $C_l$. In modern buses $C_{0,2D}$ is generally much smaller than $C_c$ [5]. Combining the capacitance model with Eq. 1, results in a range for the effective capacitance of a metal wire of $[C_{0,2D}, C_{0,2D} + 4C_c]$. For a metal wire located at one of the two edge positions of the bus, the effective capacitance can not exceed $C_{0,2D} + 2C_c$. The self-capacitance always needs to be charged or uncharged when a line is switching and its maximum crosstalk noise mainly determines the interconnect performance (e.g. delay $\tau = \max_{i,k}[C_{eff,i}[k]]$). Consequently, the crosstalk classification of a pattern sequence is based on the maximum effective coupling-capacitance of any interconnect at any clock cycle [4]. Thus, the crosstalk in a metal wire bus is classified in five classes as $0C_c$ to $4C_c$. For example, in a 4C-free 2D bus, the effective capacitance of a wire never exceeds $C_{0,2D} + 3C_c$, and in a 3C-free 2D bus it never exceeds $C_{0,2D} + 2C_c$.

We will show in Section III that the capacitance model used for the derivation of previous TSV crosstalk avoidance codes leads to impractical CACs, due to the disregarded edge effects. Thus, we present a new TSV capacitance model, depicted in Fig. 1-b, which considers these effects.

Due to the Faraday cage effect, only TSVs at the edges of an array have a significant ground-capacitance due to the grounded substrate contacts. Although the ground-capacitances of the TSVs at the four corners of the array are slightly bigger, for the sake of simplicity, we use one capacitance value ($C_g$) to describe the TSV-substrate capacitances of all edge TSVs. Between any adjacent TSV pair exists a coupling-capacitance. Because of the different distances, a capacitance between diagonal adjacent TSVs ($C_d$) is several times smaller than a capacitance between orthogonal adjacent TSVs. In the middle of the array, the capacitances between orthogonal adjacent TSVs are equal to $C_g$. Due to the E-field sharing effect, the capacitances between two orthogonal adjacent TSVs, which are both located at an array edge, ($C_e$) are significantly bigger than $C_g$ (e.g. 30-40 % [13]).

Summarized, the effective capacitance of a middle TSV is in the range of $[C_c, C_c + 2C_e + 2C_n + 4C_g]$, while the effective capacitance of an edge TSV is in the range of $[C_c + C_g, C_c + 2C_e + 2C_n + 4C_g]$. The classification is again only based on the coupling-capacitances. Therefore, the capacitance value between two orthogonal adjacent middle TSVs ($C_e$) is used as the reference value, which is subsequently referred to as $C_{3D}$. This results in 9-9=81 crosstalk classes for middle TSVs ($0C_{3D}$ to $(8+8\lambda_d)C_{3D}$), and 3-5-5=75 crosstalk classes for edge TSVs ($0C_{3D}$ to $(2+4\lambda_e+4\lambda_d)C_{3D}$). $\lambda_d$ and $\lambda_e$ are equal to $C_d/C_{3D}$ and $C_e/C_{3D}$, respectively. In the following we also use $\lambda_s$ for $C_s/C_{3D}$. The factors $\lambda_d$, $\lambda_e$, $\lambda_s$ are independent of the TSV length. Consequently, they provide an abstract and universal TSV crosstalk classification, similar to the traditional one for the planar metal wires.

III. LIMITATIONS OF PREVIOUS CACS FOR 3D ICs

In the reported literature, no hybrid CAC is presented. All existing coding schemes for 3D ICs are designed to reduce only the TSV crosstalk. Hence, they do not reduce the crosstalk of the metal wires. A 3D SoC with $N$ stacked dies reduces the maximum metal wire length in a single die by a factor of $\sqrt{N}$ [1]. Thus, in a 3D bus or point-to-point interconnect architecture, the critical path consists of metal wires spread over the $N$ dies, together of length $L/\sqrt{N}$, and $N$ TSV arrays, each of length $h$. Here, $L$ is the maximum wire length for a 2D SoC implementation and $h$ is the substrate thickness. $h$ is usually significantly smaller than $L$, so for a reasonable amount of dies a significant fraction of the critical path is constructed by metal wires. Additionally, even a simple path between two different layers is always constructed by a TSV array and metal wires in the source and the destination layer, due to the Keep-out-Zone (KOZ) surrounding each TSV [15]. Consequently, a CAC for 3D SoCs should reduce the maximum crosstalk of the metal wires and the TSVs simultaneously to be very efficient.

Furthermore, the coding gain of existing TSV CACs is far less than previously reported in the according papers. Table I
reports the (normalized) capacitance values for modern TSV arrays, obtained by parasitic extractions with the Ansys Q3D EM wave solver [16]. The parasitics are extracted for $5 \times 5$ TSV arrays. The analyzed TSV dimensions (pitch: $d_{tsv}$, and radius: $r_{tsv}$) correspond with the ones reported for the 2015-2018 time slot of the International Technology Roadmap for Semiconductors (ITRS) 2013. Since $C_{3D}$ is reported per unit length ($C_{3D}=C_{3D}/l_{tsv}$), the results can be used for all TSV lengths $l_{tsv}$. The ITRS did not report the expected TSV liner thickness $t_{ox}$. According to existing process nodes, we choose $t_{ox}=r_{tsv}/5$. The substrate is Boron (p) doped and has a conductivity of $10\, \text{µ}\Omega\text{cm}$. A TSV, its dielectric and the substrate form a metal oxide semiconductor (MOS) junction. Thus, in the substrate, a TSV is surrounded by a depletion region [17]. For the parasitic extractions, the width of a depletion region is calculated for every geometrical variation by means of the exact Poisson’s equation under the assumption of an average TSV voltage of $V_{dd}/2=0.5\, \text{V}$ and modeled as an area where the substrate has no free charge carriers ($\sigma=0$) [6].

According to Table I, for random patterns, the maximum effective capacitance is approx. $C_L+10.5\, C_{3D}$ for the middle TSVs ($C_L+(8+8\lambda_d)C_{3D}$) and approx. $C_L+9.7\, C_{3D}$ for the edge TSVs ($C_L+(2+4\lambda_e+4\lambda_d+\lambda_s)C_{3D}$). For the derivation of all previous crossstalk avoidance codes, the edge effects are not considered and the efficiency of the coding approaches is only evaluated for a TSV in the middle of an array. Existing crossstalk avoidance codes (CACs) have in common that, for each TSV, they reduce the maximum amount of adjacent TSVs switching in the opposite direction. For example, the 6C 3D CAC [10] simply limits the maximum amount of orthogonal adjacent aggressor TSVs switching in the opposite direction to three. When three orthogonal adjacent aggressor TSVs switch in the opposite direction, the remaining one always switches in the same direction. Consequently, for a middle TSV, the maximum effective capacitance is reduced to $C_L+(6+8\lambda_d)C_{3D}$ (approx. $C_L+8.5\, C_{3D}$). Edge TSVs have a maximum of three orthogonal neighbors/aggressors. Thus, the 6C 3D CAC [10] coding does not provide an optimization of the crossstalk of an edge TSV. Coded and uncoded, their maximum capacitive crossstalk is approx. $C_L+9.7\, C_{3D}$. Thus, the worst case delay for the coded pattern occurs at the edges. Consequently, the edge effects reduce the actual coding efficiency. In the same way one can show that all previous TSV CACs actually have a smaller coding efficiency than previously reported.

To quantify the actual coding efficiency, we reanalyze the delay reduction of the most promising 3D CACs [10–12]. The 4LAT coding [11] limits the number of maximum adjacent switching TSVs to four. Consequently, for each TSV $i$, maximum three $\delta_{i,j}$ values can be two. The 6C-FNS coding [12] limits the coupling of each middle TSV to $6.5\, C$, where an orthogonal capacitance value ($C_{ij}$ or $C_{ei}$) is equal to $1\, C$ and a diagonal capacitance value ($C_{di}$) is equal to $0.25\, C$. The 6C [10] and the 4LAT [11] coding are evaluated for the already discussed quadratic $5 \times 5$ array with $r_{tsv}=1\, \mu\text{m}$, $d_{tsv}=4\, \mu\text{m}$ and $l_{tsv}=50\, \mu\text{m}$. These TSV dimensions correspond with the minimum global TSV dimensions reported for the year 2018 by the ITRS. A drawback of the 6C-FNS [12] is that it only works for $3 \times 3$ arrays. Thus, for the analysis of this 3D CAC, the array dimensions are changed to $3 \times 8$. In contrast to the analyses in the respective papers, where only the delay reduction of a middle TSV is analyzed, we analyze the delay of the edge and middle TSVs. To determine the pattern dependent delay of the TSVs with the Spectre circuit simulator, we use extracted complete RLC 3π-lumped element circuits of the TSV arrays. In the simulations, each TSV is driven by a two inverter chain and the input slew rate is 1 ps. Driver strengths of the first and second inverter are $4\times$ and $12\times$ respectively. For the inverters, 22 nm Predictive Technology Model (PTM) is used. In Fig. 2 the Spectre simulations and the measurement of the propagation delay from the input of the second inverter to the output of the TSVs, are illustrated for the 6C coding [10]. In this work, we distinguish between the maximum delay of a middle TSV ($T_{p,m}$) and the maximum delay of an edge TSV ($T_{p,e}$), for the coded and the uncoded pattern. In Table II, all measured propagation delays are shown. The table reveals that, as expected, for 3D CACs the worst case delay always occurs at the edges, while for the uncoded scenario, it occurs in the middle. Table II also includes the overall delay reduction and the delay reduction for middle TSVs. The values for the

### Table I

<table>
<thead>
<tr>
<th>(Normalized) capacitance values in modern TSV arrays</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_{tsv}$ [µm]</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

### Table II

<table>
<thead>
<tr>
<th>Maximum propagation delay for the middle ($T_{p,m}$) and the edge ($T_{p,e}$) TSVs, for CAC coded and uncoded patterns, besides the CAC delay reduction of the middle TSVs ($\Delta T_{p,m}$) and the actual (overall) CAC delay reduction ($\Delta T_p$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coding Technique</td>
</tr>
<tr>
<td>Uncoded</td>
</tr>
<tr>
<td>6C [10]</td>
</tr>
<tr>
<td>6C-FNS [12]</td>
</tr>
</tbody>
</table>
middle TSVs correspond with the ones reported in [10]–[12].

The results show that the edge effects drastically decrease the overall delay reduction and consequently the efficiency of the CACs. For example, the delay reductions of the 6C [10] and 4LAT [11] approaches are only about 50% of their previous expected values. Especially the 6C-FNS coding [12] only shows a delay reduction of about a sixth of the previously expected value (4.72% instead of 28.84%). The first reason for this dramatic decrease in the coding efficiency is the neglected edge effects, which somehow even slightly increase if an array dimension (M or N) is smaller than four. The second is the generally unconsidered edge TSVs. The coding approach simply limits the maximum coupling of the middle TSVs to 6.5C (e.g. 6C_\text{m}+2C_d ), while the maximum coupling of the edge TSVs remains unoptimized. If all surrounding TSVs of a TSV located at a single edge switch in the opposite direction, its coupling is equal to 2C_\text{m}+4C_e+4C_d , which is equal to 7C, according to the crosstalk classification in [12]. So actually, the proposed 6C-FNS code is only a 7C-FNS code.

The (bit) overhead of an n-bit CAC is defined as:

\[ OH(n) = \frac{m-n}{n}, \]

where \( n \) is equal to the bit width of the input vector and \( m \) is equal to the bit width of the codewords. The 6C, 4LAT, and 6C-FNS require an asymptotic overhead (\( \lim_{n \to \infty} OH(n) \)) of 44%, 80%, and 50%, respectively [10]–[12]. These low coding efficiencies (large overhead and low crosstalk reduction), show the need for new TSV crosstalk avoidance methods, which have to be derived while considering the edge effects.

IV. PROPOSED CODING METHOD

A. General TSV CAC Technique

In this section we will present a new CAC technique for arrays of TSV, called \( \omega_m/\omega_e \) technique. The presented coding technique overcomes the limitations of previous TSV CAC approaches, which arise due to the edge effects.

The general idea is to reduce the maximum possible effective capacitance of a middle and an edge TSV by \( \omega_m C_{3D} \) and \( \omega_e C_{3D} \), respectively. Consequently, the maximum crosstalk class of the middle TSVs is reduced to

\[ (8+8\lambda_d - \omega_m)C_{3D}, \]

while the maximum crosstalk class of the edge TSVs is reduced to

\[ (4\lambda_e + 2 + 4\lambda_d - \omega_e)C_{3D}. \]

As shown in Section III, for uncoded patterns, the maximum effective capacitance of an edge TSV is approx. 0.8C_{3D} smaller compared to a middle TSV. Thus, in order to obtain an efficient coding scheme, \( \omega_m \) should be about \( \omega_e + 0.8 \).

B. 3D CAC Approach

In this subsection we present a coding approach which combines the previous introduced \( \omega_m/\omega_e \) coding approach with a traditional 2D CAC, to overcome the second limitation of previous coding approaches for 3D integration. The presented coding approach reduces not only the crosstalk of the TSVs, but simultaneously reduces the crosstalk of the metal wires while using only one data encoding scheme.

The idea is to exploit the bit level properties of a 2D CAC coded data stream, by a mapping of the bits onto a TSV array that results in a \( \omega_m/\omega_e \) TSV coding.

Memoryless 3C-free CACs are the most popular 2D CACs and have been extensively studied in the past [18]–[21]. Two different data encoding methods exist for 3C-free bus encoding: Forbidden- Pattern-Free (FPF) [20] and Forbidden-Transmission-Free (FTF) [21] encoding. For both methods the encoding/decoding process can be based on a Fibonacci numeral system (FNS) mapping, which leads to an encoder/decoder circuit (CODEC) complexity which is several magnitudes lower than for other 3C-free CODECs [4]. For the FPF CAC, bit vectors that contain a 010 or 101 bit sequence are forbidden. For example 111000110 is a valid FPF codeword, while 110100011 is a forbidden bit vector. In [4] the authors prove that a FPF bus is a 3C-free bus, since, for all \( i, \) if \( \delta_{i,i-1} \) is equal to 2, \( \delta_{i,i+1} \) is always 0 and \( \text{vice versa} \). For the FTF CACs, all \( \delta \)-values are limited to 1 by prohibiting two adjacent bits from switching in the opposite directions. Hence, the forbidden transitions are 01 \( \rightarrow \) 10 and 10 \( \rightarrow \) 01. In [21] it is proven that the largest set of FTF codewords is generated by eliminating the 01 pattern from the \( b_{2j+1}b_{2j} \) boundaries, and the 10 pattern from the \( b_{2j}b_{2j-1} \) boundaries. The bit overhead of the FPF and FTF CACs, based on a FNS mapping, are equal and asymptotically reach 44% [4].

We propose to use FTF data encoding, since it has some advantages over the FPF encoding. One advantage is that the CODEC of the FTF CAC requires a ca. 17% lower gate count and an almost 50% lower delay [4]. Nevertheless, for both encoding techniques, a FNS CAC CODEC still exhibits a quadratic growth in complexity with the size of the bus [4]. Thus, the CODEC complexity will quickly cancel out any coding savings with increasing input data widths. To overcome this limitation, the bus can be partitioned into small groups which are encoded individually. In this case, a difficulty arises due to undesired crosstalk transitions between adjacent lines of different groups. In [4], two techniques are designed to address this issue: Group Complement and Bit Overlapping. Both, again, cause a significant bit and CODEC overhead, which makes them suboptimal.

Here, we present a more effective technique for the FTF data encoding. In 3D ICs power/ground TSVs have to be spanned over the several dies of the system in order to build a 3D power network. Since power/ground (V/G) lines are stable, they can be used in FTF encoding for the bus partitioning. The bus, containing \( NB \) lines, is divided in \( G \) groups which are encoded individually by a \( n \)- to \( m \)-bit FNS code, where \( m = NB/G \). Between each first bit of a group and the last \( (m^{th}) \) bit of the previous group, a stable (V/G) line is placed. This bus partitioning generally causes no overhead at all since power/ground TSVs are often transmitted with data TSVs in one array [22]. The crosstalk factor \( \delta_{i,s} \) of a data line \( i \) and a stable line \( s \) is \( \Delta b_i[k] \) (Eq. 2 with \( \Delta b_j[k] = 0 \)), and thus limited to 1. Consequently, stable lines do not violate the FTF condition. However, for FPF encoding, an additional stable line leads to a bus which is only 4C-free, which is illustrated.
are shielded by stable lines, resulting in a 2D layout. Consequently, the crosstalk of a signal metal wire in the second pattern is a forbidden pattern (includes 101 sequence).

In this subsection we derive a mathematical method to find the optimal placement of the bits of a given pattern set onto an interconnect structure. By means of this mathematical method we can determine the perfect use of 2D CACs for TSV arrays. The vector of the effective capacitances \( \vec{C}_{\text{eff}} \), where the \( i^{th} \) vector entry is \( C_{\text{eff},i} \) (Eq. 1), can be expressed as:

\[
\vec{C}_{\text{eff}}[k] = (M_e[k] \circ C) \cdot I_{\text{NB} \times 1},
\]

where \( M_e \) is the coupling matrix:

\[
M_{e,i,j}[k] = \begin{cases} \Delta b_i^2[k] & \text{for } i = j \\ \delta_{i,j}[k] & \text{else} \end{cases}.
\]

\( C \) is the capacitance matrix of the interconnect structure containing \( NB \) lines, where \( C_{i,j} \) is the self-capacitance of line \( i \) and \( C_{i,j} \) is the coupling-capacitance between the lines \( i \) and \( j \). \( \circ \) is the Hadamard operator (element wise multiplication) and \( I_{\text{NB} \times 1} \) is a column vector of ones. A new mapping of the bits onto the interconnects can be realized by swapping rows and the according columns of the coupling matrix, which is mathematically expressed as [23]:

\[
M_{e,i}^*[k] = P M_{e} [k] P^T,
\]

where \( P \) is a valid \( NB \times NB \) permutation matrix. A valid permutation matrix has exactly one 1 in each column/row while all other matrix entries are 0. The set of all valid \( NB \times NB \) permutation matrices is denoted as \( S_{\text{NB}} \). To map the \( j^{th} \) bit of the data stream on line \( j \), \( P_{i,j} \) is set to one. For an exemplary 4-bit interconnect structure, if we want to map bit 1 on line 4, bit 2 on line 3, bit 3 on line 1 and bit 4 on line 2:

\[
P = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.
\]

Thus, the effective capacitances for a given placement \( P \) can be determined by:

\[
\vec{C}_{\text{eff}}[P,k] = (P M_e[k] P^T \circ C) \cdot I_{\text{NB} \times 1}.
\]

Under the condition that for all \( i, j \) and \( l, \delta_{i,j} \) and \( \delta_{i,l} \) are independent (\( \max(\delta_{i,j}+\delta_{i,l}) = \max(\delta_{i,j}) + \max(\delta_{i,l}) \)), the vector constituting the maximum effective capacitances values of all lines can be calculated by:

\[
\vec{C}_{\text{eff, max}}[P] = (P M_{\text{mc}} P^T \circ C) \cdot I_{\text{NB} \times 1},
\]

where \( M_{\text{mc}} \) is the maximum coupling matrix:

\[
M_{\text{mc},i,j} = \begin{cases} \max(\Delta b_i^2) & \text{for } i = j \\ \max(\delta_{i,j}) & \text{else} \end{cases}.
\]

Please note, the assumption of independence is true for the CACs discussed in this section but not for the FPF CAC, since for the FPF CAC: \( \max(\delta_{i,i-1}) = \max(\delta_{i,i+1}) = 2 \) and \( \max(\delta_{i,i-1}+\delta_{i,i+1}) = 2 \).

The optimal mapping \( P_{\text{opt}} \) minimizes the maximum value of \( \vec{C}_{\text{eff,max}}^2 \), which can mathematically be expressed as:

\[
P_{\text{opt}} = \arg \min_{P \in S_{\text{NB}}} (\max(P M_{\text{mc}} P^T \circ C) \cdot I_{\text{NB} \times 1}).
\]
Fig. 5. Effect of the proposed coding approach on the maximum effective TSV capacitance $C_{\text{eff,max}}$, for different underlying planar 2D CACs. Compared are the two mapping methods, optimal and Snake, for several array dimensions. (a) for $r_{\text{tsv}}=1\,\mu m$ and $d_{\text{tsv}}=4\,\mu m$; (b) for $r_{\text{tsv}}=2\,\mu m$ and $d_{\text{tsv}}=8\,\mu m$

For this work we use the Simulated Annealing technique to determine the optimal mapping of 2D CACs onto TSV arrays. When two placements lead to the same cost $\max(C_{\text{eff,max}})$, the one with the lower average $\bar{C}$ is chosen. For the TSV capacitance matrices we use the ones extracted for the predicted minimum global TSV dimensions for the years 2018 ($r_{\text{tsv}}=1\,\mu m$; $d_{\text{tsv}}=4\,\mu m$; $l_{\text{tsv}}=50\,\mu m$) and 2015 ($r_{\text{tsv}}=2\,\mu m$; $d_{\text{tsv}}=8\,\mu m$; $l_{\text{tsv}}=50\,\mu m$). In contrast to Section III, here we analyze several array dimensions $M \times N$. More precisely, we analyze quadratic arrays from $3 \times 3$ to $7 \times 7$ and non-quadratic arrays with $M$ equal 3 and $N$ equal 6, 9 or 12.

The characteristics of a specific pattern set is captured by $M_{\text{mc}}$. For an uncoded pattern set $M_{\text{mc},i,j}$ is 2, except for the diagonal entries ($i=j$) which are 1. For FTF pattern, $M_{\text{mc},i,j}$ is 2 except for the diagonal elements and its adjacent entries ($j=i+1$ and $j=i-1$) which are 1. An (additional) stable line at position $x$ leads to $M_{\text{mc},x,j}=0$ for all $j$, and $M_{\text{mc},i,x}=1$ for all $i \neq x$. Here we analyze four different 2D crosstalk avoidance methods: 3C/4C-free shielding and the FTF CAC with/without stable lines for bus partitioning. For the FTF with bus partitioning (FPF BP), we assume that a TSV array contains approx. 10% stable (V/G) TSVs. For example, in a $4 \times 4$ array, two stable TSVs are assumed for bus partitioning which can be used to partition the 14 remaining signal bits $b$ in three groups ($b_1$ to $b_5$, $b'_1$ to $b'_5$ and $b''_1$ to $b''_4$).

Fig. 5 shows, for some array dimensions and all crosstalk avoidance methods, the maximum effective capacitance for the optimal and the Snake mapping, besides the maximum effective capacitance for random patterns. The figure reveals that, uncoded, the maximum effective capacitance values are almost independent of the array dimensions. The Snake mapping leads to a $2/1+\lambda_e$ CAC for the FTF pattern, which is equal to the optimal mapping. So the maximum effective capacitance of the middle and the edge TSVs is approx. $C_L+8.5C_{3D}$ and $C_L+7.35C_{3D}$, respectively. Compared to the uncoded pattern set, the FTF CAC always leads to a $C_{\text{eff,max}}$ reduction of ca. 19%. Also, for 3C-free shielding the Snake mapping is equal to the optimal mapping which results in a $4/2+\lambda_e$ CAC since it completely avoids opposite switchings between orthogonal adjacent TSVs, as shown in Fig. 6-a. Thus, the maximum effective capacitance is reduced to ca. $C_L+6.5C_{3D}$ (reduction by ca. 38%). For the $4C$-free shielding the Snake mapping leads to a $1+2\lambda_d/1+\lambda_d$ CAC (reduction by ca. 16%). Here, for smaller TSV arrays, a different mapping can lead to an increased $\omega_m$, which increases the coding efficiency. For example, in a $4 \times 4$ array, as illustrated in Fig. 6-b, the shields are optimally placed in a way that $\omega_m$ is doubled to $2+2\lambda_d$. Hence, the $4C$-free shielding results in a maximum $C_{\text{eff,max}}$ reduction of 32% for a $3 \times 3$ array. But, due to an increased middle over edge TSV ratio with increasing array dimensions, a significant increase in $\omega_m$ is only possible for small array dimensions. Thus, the gain of the optimal mapping over the Snake mapping decreases with increasing array dimensions.

Also, for the FTF BP CAC, the crosstalk reduction of the optimal mapping differs from the reduction for the Snake mapping, which results in a $2/1+\lambda_e$ CAC. For example, the optimal mapping shown in Fig. 6-c, increases the $\omega_m$-value to...
3 + \lambda_d. This boosts the crosstalk reduction from about 19% to above 30%. Due to the increasing fraction of middle TSVs, while having a constant fraction of stable TSVs (here 10%), an increase in the \omega_m value, due to a mapping that differs from the snake mapping, is only possible for small array dimensions. Already, for a 7 × 7 array, the optimal mapping is almost equal to the Snake mapping.

In summary, the Snake mapping is equal to the optimal mapping, if no stable lines/shields are present in the array. Stable lines, located at the edges of an array for the snake mapping, can be remapped into the middle of the array in order to shield the maximum amount of TSVs. But this approach only works for small TSV array, due to the increasing fraction of middle TSVs with increasing array dimensions. Therefore, asymptotically Snake and optimal mapping are again equal.

V. EXPERIMENTAL RESULTS

In this section we compare the presented \omega_m/\omega_c remapping approach with existing 3D CAC techniques [10]–[12] in terms of bit overhead, CODEC area, and overall interconnect delay reduction. We analyze the transmission of a 16-bit wide random data stream over a TSV array and a metal wire bus. In the analysis, the TSV dimensions are equal to the minimum ones reported for the year 2018 by the ITRs (r_{tsv}=1 \mu m; d_{tsv}=4 \mu m and l_{tsv}=50 \mu m). To extract the TSV parasitics, the Q3D extractor is again used. The metal wires are assumed to be in Metal4 with a wire width and spacing of 0.15 \mu m. The length of a metal wire segment is assumed to be 100 \mu m. The wire parasitics are obtained by the TSMC wire model, which is based on Synopsys Raphael. The chosen value for width and spacing corresponds with the minimum possible value.

As the underlaying 2D CACs in our \omega_m/\omega_c approach, we analyze the FTF CAC with bus partitioning and the 4C/3C-free shielding. Here, an application of the FTF CAC without bus partitioning is pointless. For the simple FTF the resulting codeword bit width is a prime number (23), so a mapping on a TSV array (M, N > 1) is not possible without unused/stable TSVs, which consequently should be used for bus partitioning again.

Here, we assume a transmission of the 4C-free shielded patterns (24-bit) and 3C-free shielded patterns (36-bit) over a 4 × 6 and a 4 × 8 array. For the FTF CAC with bus partitioning, we assume a transmission of the patterns together with one ground and one power TSV, which is equal to the minimum required amount of power/ground TSVs. Consequently, the coding can be partitioned into three parts: two 5-bit to 7-bit FTF coders and one 6-bit to 9-bit FTF coder. The 23 signal lines, together with the two stable lines, are assumed to be transmitted over a quadratic 5 × 5 array. Here, the required power/ground TSVs are not considered in the overhead calculation. For previous TSV CACs, except the 4LAT [11], stable lines cannot be transmitted together with the signal lines over one array without violating the pattern conditions. Thus, for the analysis of previous TSV crosstalk schemes we assume that an additional array exists for the power/ground TSVs. To obtain the minimum overhead for the 6C [10], 4LAT [11], and 6C-FNS [12] coding, a 5 × 4, 3 × 9, and 3 × 8 TSV array is required, respectively. As in Section III, the delay values are determined with Cadence Spectre and the 22 nm PTM drivers are used. To determine the CODEC complexity, all encoder/decoder pairs are synthesized in a commercial 40 nm technology, and the resulting gate equivalents are reported. Here, the CODEC delay is not reported, since it can be hidden in a pipeline.

The results are presented in Table III. The table reveals that the presented coding approach outperforms the existing 3D CACs significantly. Existing 3D CACs reduce the TSV delay by a maximum of 11.45% (4LAT [11]), while for the presented \omega_m/\omega_c remapping approach, the TSV delay reduction can be more than 3 times larger (3C-free shielding: 35.91%). Additionally, the 3C-free shielding reduces the metal wire delay by 47.64% and does not require a CODEC design. In comparison, the 4LAT [11] approach does not optimize the metal wire delay and requires a CODEC gate count of almost 2 K. Solely in terms of (bit) overhead, the 4LAT [11] slightly outperforms the 3C-free shielding (asymptotic less than 3%).

Generally, the high overheads of the 3C-free shielding and the 4LAT [11] often make both approaches not suitable, due to the large TSV dimensions and limitations in the available silicon area. A good compromise offers the presented remapping of the FTF coding with bus partitioning, as it leads to the same metal wire delay reduction and to the second biggest TSV delay reduction (ca. 20%). Compared to the 4LAT [11] it reduces the asymptotic overhead by almost 50%, increases the delay reduction by 85.76% and still shows an reduction in the CODEC complexity by a factor of 4.66 for n=16-bit. For larger data widths this improvement is even larger. Due to the bus partitioning, the FTF CODEC area scales linear and the delay stays constant with an increasing input vector width n. In contrast, except the 6C-FNS [12], all previous 3D CAC scale exponentially, which makes them impractical for wide data streams. The 6C-FNS [12], as the only linear scaling

<table>
<thead>
<tr>
<th>Method</th>
<th>Pattern</th>
<th>Input Data Width n=16-bit</th>
<th>Asymptotic (lim_{n→∞})</th>
</tr>
</thead>
<tbody>
<tr>
<td>This Work: \omega_m/\omega_c remapping</td>
<td>FTF BP</td>
<td>3C-free Shield</td>
<td>4C-free Shield</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16.55%</td>
<td>35.91%</td>
</tr>
<tr>
<td></td>
<td>11.45%</td>
<td>6.48%</td>
<td>11.45%</td>
</tr>
<tr>
<td></td>
<td>6.48%</td>
<td>6.48%</td>
<td>6.48%</td>
</tr>
</tbody>
</table>

Table III
Comparison of the proposed remapping of 2D CACs and existing 3D CACs in terms of: TSV/metal wire delay reduction (\Delta T_{p,3D}/\Delta T_{p,2D}), overhead, CODEC Gate Equivalent (GE) and CODEC Growth.
traditional 3D CAC CODEC, has the worst coding gain.

A fair overall CAC performance indicator is the coding efficiency, which is defined as the asymptotic ratio of delay reduction over overhead. For the traditional 3D CACs, the metal wire coding efficiency is always zero while the TSV coding efficiency is in between 0.12 and 0.23. In contrast, for the presented \( \omega_m/\omega_e \) approach and all analyzed 2D CACs, the metal wire coding efficiency is always between 0.58 and 1.11, while the TSV coding efficiency is always between 0.41 and 0.44. Thus, for all analyzed 2D CACs, the proposed remapping approach has a higher efficiency than all previous 3D CACs. The general highest efficiency shows the presented FTF with bus partitioning (FPF BP) to \( \omega_m/\omega_e \) remapping approach (1.11 for the metal wires and 0.44 for the TSVs).

Summarized, the presented 2D CAC to \( \omega_m/\omega_e \) remapping is the only 3D CAC approach which effectively reduces the inter TSV crosstalk. Additionally, it simultaneously results in a drastic reduction of the metal wire crosstalk.

VI. CONCLUSION

In this work we presented an edge effect aware crosstalk avoidance technique for 3D integration, called \( \omega_m/\omega_e \) remapping approach. In the first part of this work we have proven theoretically and by means of experimental results that the edge effects make previous TSV crosstalk avoidance techniques inefficient. We also outlined that an efficient crosstalk avoidance method should also reduce the metal wire, not only the TSV, crosstalk. The proposed coding method overcomes both limitations. The switching characteristics of 2D CAC pattern sets are exploited by an optimal TSV array mapping so that the crosstalk of the metal wires and the TSVs are reduced simultaneously. For our approach, we analyzed different underlying 2D CACs, with the result that the \( \omega_m/\omega_e \) remapping always significantly outperforms all existing 3D CACs (higher coding efficiency, higher maximum delay reduction, etc.). Our approach shows a maximum TSV and metal wire delay reduction of 35.91 % and 47.64 %, respectively. In comparison, previous approaches reduce the TSV delay by a maximum of 11.45 %, while providing no optimization of the metal wire delay and inducing higher hardware costs. The highest crosstalk reduction over bit overhead ratio shows the presented FTF \( \omega_m/\omega_e \) coding, which initially has the drawback of a quadratic CODEC growth. To overcome this limitation, we proposed a bus partitioning method which simply exploits the power/ground lines in a TSV array to obtain a linear CODEC growth. This bus partitioning method causes no circuit or bit overhead and makes the \( \omega_m/\omega_e \) coding approach applicable for wide data streams.

ACKNOWLEDGMENT

This work is funded by the German Research Foundation (DFG) project PI 4478-1.

REFERENCES