High-performance, Cost-effective 3D Stacked Wide-Operand Adders

Voicu, George Razvan; Cotofana, Sorin Dan

DOI
10.1109/TETC.2016.2598290

Publication date
2016

Document Version
Accepted author manuscript

Published in
IEEE Transactions on Emerging Topics in Computing

High-performance, Cost-effective 3D Stacked Wide-Operand Adders

George R. Voicu and Sorin D. Cotofana, *Senior Member, IEEE*

Abstract—Through-Silicon Vias (TSV) based 3D Stacked IC (3D-SIC) technology introduces new design opportunities for wide operand width addition units. Different from state of the art direct folding proposals we introduce two cost-effective 3D Stacked Hybrid Adders with identical tier structure, which potentially makes the manufacturing of hardware wide-operand fast adders a reality. An N-bit adder implemented on a K identical tier stacked IC performs in parallel two N/K-bit additions on each tier according to the anticipated computation principle. Inter-tier carry signals performing the appropriate sum selection are propagated by TSVs. The practical implications of direct folding and of our hybrid carry-select/prefix approaches are evaluated by a thorough case study on 65nm CMOS 3D adder implementations, for operand sizes up to 4096 bits and 16 tiers. Our simulations indicate that in almost all configurations at least one of the two proposed 3D stacked hybrid approaches is faster than the fastest 3D folding approach. When considering an appropriate metric for 3D designs, i.e., the delay-footprint-heterogeneity product, the hybrid adders substantially outperform the folding counterparts by a factor in-between $1.67\times$ and $23.95\times$.

Index Terms—Adders, Cryptography, Three-dimensional integrated circuits, Through-silicon vias.

1 Introduction

Today, almost any computing device has stringent requirements in terms of security, coming from privacy concerns, restricted content, restricted access, etc. Data encryption is one solution to address this issue, and public-key cryptography [1] is a fundamental and widely used encryption approach. For example, RSA [2] is the dominant cryptographic algorithm used in key exchange in secure communications over the Internet. The security of any cryptographic system is proportional with the encryption key length, so the larger the key, the better. Currently, key lengths of 1024 or 2048 bits are considered sufficient for an unbreakable RSA algorithms [3], but continuous advances in raw computation power and/or integer factorization theory may require the increase of this value even further, to fulfil the application security demands [4].

Large key length cryptography relies on intensive utilization of arithmetic operations on wide-operands. Traditionally, these arithmetic operations are implemented as software routines running on cryptographic coprocessors, since wide-operand arithmetic units fully implemented in hardware are impractical when making use of the current mainstream planar Integrated Circuits (ICs) fabrication technologies [5], [6], [7].

As an alternative to planar technology, 3D Stacking Integrated Circuits (3D-SIC) have emerged as means for improving circuit performance by reducing the global wire-length and the design footprint [8]. The basic idea behind the 3D-SIC approach is to partition a large design in several smaller parts, and to implement each of them on a separate die. Multiple dies are stacked and bonded together, and signals travel between the tiers in the stack using special vias, i.e., Through Silicon Vias (TSVs).

• *G. R. Voicu and S. D. Cotofana are with Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, The Netherlands. E-mail: {g.r.voicu,s.d.cotofana}@tudelft.nl.*

In this paper we investigate the implications of using 3D-SIC technology in designing efficient wide-operand adders, to be potentially included in cryptographic coprocessors. We first summarize direct folding strategies of fast planar adder designs, i.e., prefix adders, and provide a generalization and a classification for partitioning an N-bit operand width adder on K-tiers. We then theoretically analyze in terms of cost and performance the 3D folding classes, revealing the utilisation of different structures on each tier as a major drawback due to a significant non-recurrent engineering cost augmentation. We subsequently address this shortcoming and propose two 3D Stacked Hybrid Adders with identical tier layout.

The K-tier hybrid adders build upon a carry-select structure, where each tier contains two identical N/K-bit fast prefix adders that compute in parallel the sums according to the anticipated computation anticipation principle, i.e., with a carry-in signal of both high and low value. Subsequently, the selection of the correct sum is performed on each tier, according to the block carry signal of the less bit-significant tiers. For the inter block carry calculation we rely on TSVs and we propose two approaches: (i) transmit the block carry generated at tier i to tier $i+1$, $i = 1, 2, \ldots, K-1$, which results into a Hybrid Ripple/Carry-Select/Prefix (HRCP) Adder, and, (ii) broadcast the block carry generated at tier i, $i = 1, 2, \ldots, K-1$ to all tiers j, $i < j \leq K$ such that on each tier the appropriate group carry signal can be locally computed, which results into a Hybrid Lookahead/Carry-Select/Prefix (HLCP) Adder. Since in both cases each tier can be designed such that it has the same structure, the development and manufacturing costs are diminished.

To evaluate the practical implications of our proposal, we perform a thorough case study by implementing 3D adders with operand widths varying from 512 to 4096 bits, and a number of tiers in the range of 2 to 16. The new design space dimensions introduced by 3D stacking, i.e.,

the number of available tiers and the adder partitioning (3D folding) strategy, create new trade-off opportunities. As our simulations indicate, all 3D adders excepting the HLCP exhibit a clear delay versus tier number trade-off. For those adders, the optimal number of tiers, i.e., the 3D stack height providing the smallest delay, grows with the increase of the adder's width. In contrast, the HLCP adder becomes faster as the number of tiers increases.

Based on our simulation experiments we can conclude that in almost all configurations, at least one of the two proposed 3D stacked hybrid approaches is faster than the fastest 3D folding counterpart. In terms of consumed resources, the HRCP adder has on average the same footprint as the smallest direct folded adder, while the HLCP is on average 15% larger. However, the manufacturing cost of direct folded adders is higher, since they are constructed out of tiers implementing different circuits, and not of the same type as it is the case for the proposed 3D stacked hybrid designs. When considering the footprint-delay-heterogeneity product, which is more appropriate to capture the complexity of a 3D implementation, we can conclude that the hybrid approach is more suitable for 3D stacked integration, HRCP and HLCP adders achieving a maximum reduction factor of the footprint-delay-heterogeneity product over the best 3D folded adders of about $18\times$ and $24\times$, respectively.

The remainder of the paper is organized as follows. We first explain in Section 2 the hurdles one is facing when attempting to implement fast wide-adders, and motivate our work. Next, Section 3 gives a brief overview of relevant state of the art literature. In Section 4 different straightforward implementations of folded 3D adders are classified and analyzed. Subsequently, we propose in Section 5 two identical-tier 3D stacked hybrid adders. Section 6 presents an experimental design space exploration highlighting various trade-offs in terms of delay, footprint, and cost for 3D folded and hybrid wide-adders. Finally, Section 7 concludes this paper.

2 Background and Motivation

2.1 Fast adders

Addition is the primary mechanism to implement complex arithmetic operations, e.g., multiplication, division, etc. If addition is slow or expensive, all other addition related operations suffer in speed and/or cost. It is well known that carry propagation is the limiting factor for the performance of any adder with fixed-radix number representation [9], and that carry-lookahead approaches in which the computation of independent carries is done in parallel and in advance of the sum computation, can significantly speed-up the addition process at the expense of additional hardware resources. In view of this, most of the currently implemented high-performance adders make use of a parallel prefix carry computation scheme, which is a particular case of carry-lookahead.

Among the prefix calculation schemes, Kogge-Stone (KS) [10] and Brent-Kung (BK) [11] represent the two extremities of the theoretical design space interval determined by the area-delay trade-off [9]. The top graphs in Figures 2 and 3 depicts the internal structure of 8-bit adders implemented according to these two approaches. The squares on the

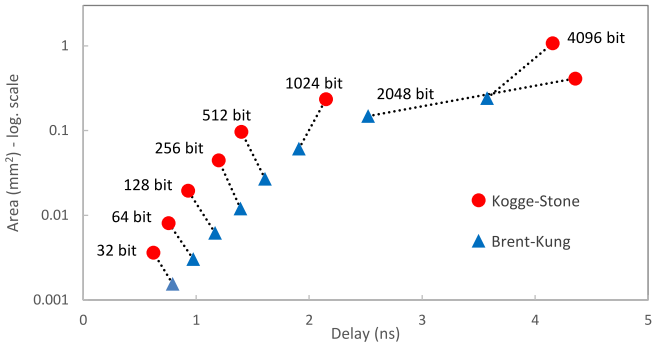


FIGURE 1. Planar prefix adders area-delay design space.

first row are half-adders that compute the carry propagate and generate signals for each bit position. For the least significant bit a full-adder is used to account for the carry in signal. The circles represents carry operator cells (also known as carry-merge cells), in which combined group propagate and generate signals are computed. The ones with a lighter shade of grey compute only generate signals. As easily deduced from the prefix graphs, the KS approach offers the fastest result (fewer stages) at the expense of the largest number of carry operator cells, while BK has the lowest number of carry operator cells, but with larger fan-out, and more propagation stages on the critical path.

When it comes to operations on wide numbers, the theoretical prefix adders' (or any fast adder's for that matter) area-delay trade-off no longer stands in practice. We observe this in Figure 1, which presents area-delay plots for BK and KS adders with various widths. The graph values are obtained from simulations under worst-case conditions of sign-off implementations in a commercial low-power $65\,\mathrm{nm}$ CMOS technology [1]. In the case of operand widths larger than 512 bits the BK variant is optimal with respect to both area and delay, in contrast with smaller widths, where both BK and KS adders lie on the Pareto frontier. This behaviour is explained as follows. The large number of carry-merge cells increases the distance between two connected cells, which in turn demands long and dense wires [12]. The solution to address the signal loss on such long wires is to instantiate additional buffers. However, this introduces a delay degradation and the routing congestion problem gets even worse. Thus, the dominant factor in wide-operand adder speed shifts from the computation delay to the communication delay.

The positive feedback loop previously described makes the adders delay to grow more than linear for large operand widths. In addition to the considerable performance decrease, a wide adder implemented in a planar fashion has a higher fabrication cost due to the manufacturing yield decrease caused by the large footprint and dense routing. In consequence, a high-performance wide-operand adder is impractical to implement in current mainstream planar IC fabrication technologies. An alternative to this issue is further presented.

1. We note that the 4096-bit KS adder could not be successfully routed, thus the presented delay value is an optimistic one.

2.2 3D Stacking IC Technology

One of the main promises of the 3D-SIC emergent technology is the reduction of global wire-length, which in turn leads to an operation speed increase, especially for large circuits, or those with wire-dominated delay. The simple explanation behind this gain is that blocks that were previously placed in the planar case far away from each other and connected by long global wires, with many repeater buffers, can now be stacked on top of each other in a 3D-SIC and communicate through short and low-resistance TSVs.

At the same time, eliminating buffers leads to a reduction in energy consumption [13]. Moreover, for the specific target that we envision, i.e., cryptographic co-processors, the fact that 3D integration allows for the execution of a wide addition as a single operation on a wide-adder leads to substantial energy savings. The software or micro-programmed alternative where multiple instructions are repeated on a low-width adder unit require a non-negligible hardware overhead to control the instruction stream and save intermediate data results, hence an additional power overhead.

Another advantage of 3D-SICs is that the top and bottom tiers in the stack can shield inner tiers from alpha particles, the leading cause of soft-errors in ICs [14]. Fault tolerance can be further increased in 3D-SICs by using heterogeneous integration or through architectural enhancements [15].

Commercial ICs where the majority of the area provides a single function, i.e., memories [16] and imaging sensors [17] already benefit from TSV-based 3D integration. 3D-SICs are expected to slowly become mainstream as the manufacturing processes [18] and the Electronic Design Automation (EDA) tools eco-system [19] matures to ease partitioning and analysis of multi-die designs.

3 Related work

Various ways to partition prefix tree adders for 3D stacking were devised and several case studies for small-width adders were performed. Vaidyanathan et al. proposed in [20] to split the prefix tree in-between computation stages and estimated a maximum $4\times$ wire length reduction for 32-bit KS adders on 3 tiers. Puttaswamy and Loh [21] use a 2-tier bit-splitting approach where carry-merge cells corresponding to odd operand bits in a KS adder reside on one tier, and cells for even operand bits reside on the other one. For sparser prefix trees like Sklansky and BK adders adjacent processing nodes are stacked. However, in the case of BK adders the splitting of the inverse carry tree from the last stages of computations is not discussed.

A variation of the previously mentioned bit-splitting partitioning strategy for KS adders is used by Ouyang et al. [22], with the first carry merge stage performing ternary operations instead of binary ones, in order to match a 3-tier stacking technology. This change, together with flipping the direction of carry forwarding in the first merge stage allows to eliminate the TSVs at the end of the prefix graph needed to compute the final sum bits.

In contrast, our work focuses on the addition of operands with large widths, and is using a hybrid adder approach, rather than simply partitioning a fast carry-prefix adder as in the current proposals. Moreover, we strive for the use of a homogeneous die stack, with identical tiers in order to drive down the design and manufacturing costs.

4 Parallel Prefix Adders 3D Partitioning

The straightforward way to design a 3D stacked fast adder is to take an existing planar prefix adder, partition it, and fold the resulting partitions such that each one is placed on a separate die. When a partitioning strategy uses this methodology, we refer to it as *3D direct folding*. In [23] we classified the 3D partitioning strategies of the carry tree in an N-bit parallel prefix adder, on a K-tier stack, as follows:

1) *Stage Folding (SF)*: the carry-merge cells in each stage are placed on one tier, as suggested in [20],
2) *Bit Interleaving (BI)*: the carry merge cells on each and every K-th column in the prefix graph are placed on the same tier, as suggested in [21],
3) *Bit-Slice Folding (BS)*: the carry merge cells on every N/K consecutive prefix graph columns are placed on the same tier.

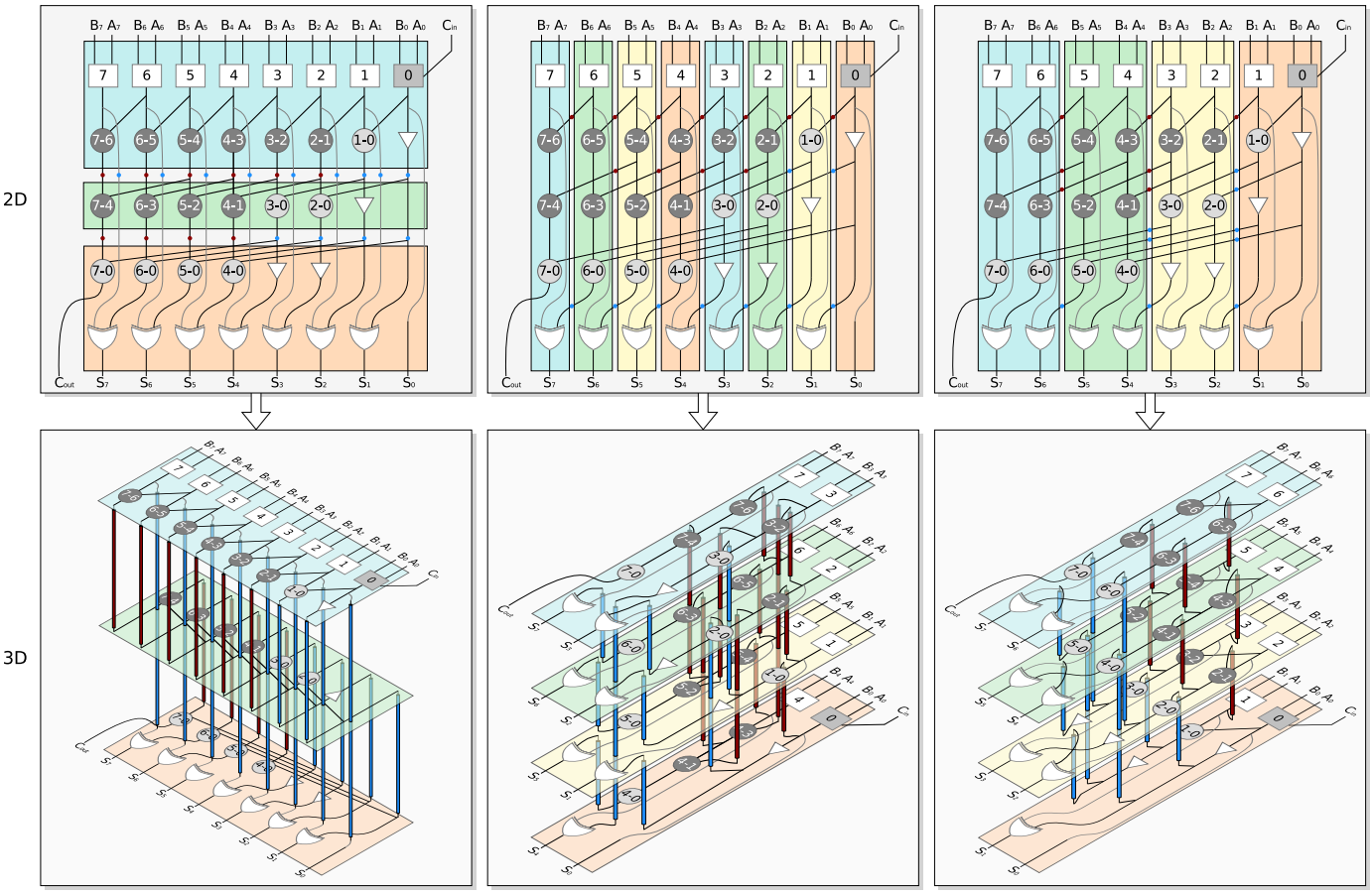
In addition to these, a fourth type of partitioning strategy can be applied by generalizing the 3-tier modified prefix tree design proposed in [22]:

4) *Enhanced Bit Interleaving (EBI)*: the same folding strategy as Bit Interleaving, with a modified prefix graph to eliminate the sum logic TSVs.

Figures 2 and 3 graphically depict how a fast 8-bit adder, with a KS and BK prefix tree, respectively, can be divided across a 4-tier stack according to the first three identified types of partitioning strategies. The first partitioning strategy folds the prefix tree along the vertical direction, by stacking tree stages, while the three remaining strategies perform the folding along the horizontal direction of the prefix tree, by stacking bit columns. The partitioning should strive to reduce the long interconnects in the carry network, therefore clusters of communicating carry-merge cells should be placed on the same die. Based on this observation, other partitioning strategies in which random carry-merge cells are placed on each tier will most likely not produce better results.

We note that it is not compulsory to restrict the Stage Folding partitioning at one stage per tier, as for the 8-bit KS adder from Figure 2a, where the number of tiers is the same as the number of stages in the prefix tree, i.e, $K = \log_2 N$. In fact, any number of tiers can be accommodated if we group several stages together on the same die, like in the BK adder example from Figure 3a.

In case of Bit Interleaving the inter-tier TSVs from the bottom of the stack in Figures 2b and 3b can be eliminated by applying the following alterations from [22] to the carry tree: (i) the first stage of the tree contains carry-merge cells with a radix equal with the number of tiers in the design, and, (ii) reverse the direction of the carry forwarding in the first stage of the tree. An example of the resulting Enhanced Bit Interleaving for a 4-tier 8-bit modified Kogge-Stone adder is presented in Figure 4. The higher radix carry-merge cells are implemented using a tree of classic radix-2 cells (the first two stages of carry-merge cells). Due to the carry forwarding reversing, cells on the leftmost $K-1$ columns are moved to the rightmost K columns, and some



(a) Stage Folding (SF KS). (b) Bit Interleaving (BI KS). (c) Bit-Slice Folding (BS KS).

FIGURE 2. 3D partitioning strategies of an 8-bit Kogge-Stone adder.

cells in the rightmost $K-1$ columns (e.g., (6-0), (5-0), and (2-0)) need to have even higher radix.

4.1 Summation delay

For wide-operand adders, the number of TSVs has a direct influence on the overall performance of the adder. The reason for this is twofold: i) the large parasitic capacitance between the TSV and the silicon substrate necessitates the placement of a high strength driving buffer before it, with large area and propagation delay [24], and, ii) the area occupied by a TSV, the keep-out zone around it, and the driving buffer increases the interconnect wires length, which in turn degrades the circuit speed. For comparison purposes, a carry-merge operator cell synthesized in a commercial $65\,\mathrm{nm}$ low power CMOS technology takes $4.68\,\mu\mathrm{m}^2$, while the minimum predictions for TSV diameter and pitch are $0.8\,\mu\mathrm{m}$ and $1.6\,\mu\mathrm{m}$ [25], respectively.

Thus, the summation delay of any $K-$tier 3D direct folded fast adder is the sum of the logic delay of the N-bit prefix adder, $D_{PP}(N)$, and the interconnect delay on the TSV(s) and wires part of the critical timing path:

$$D(N,K) = D_{PP}(N) + D_{TSV}(N,K) + D_{wire}(N,K). \quad (1)$$

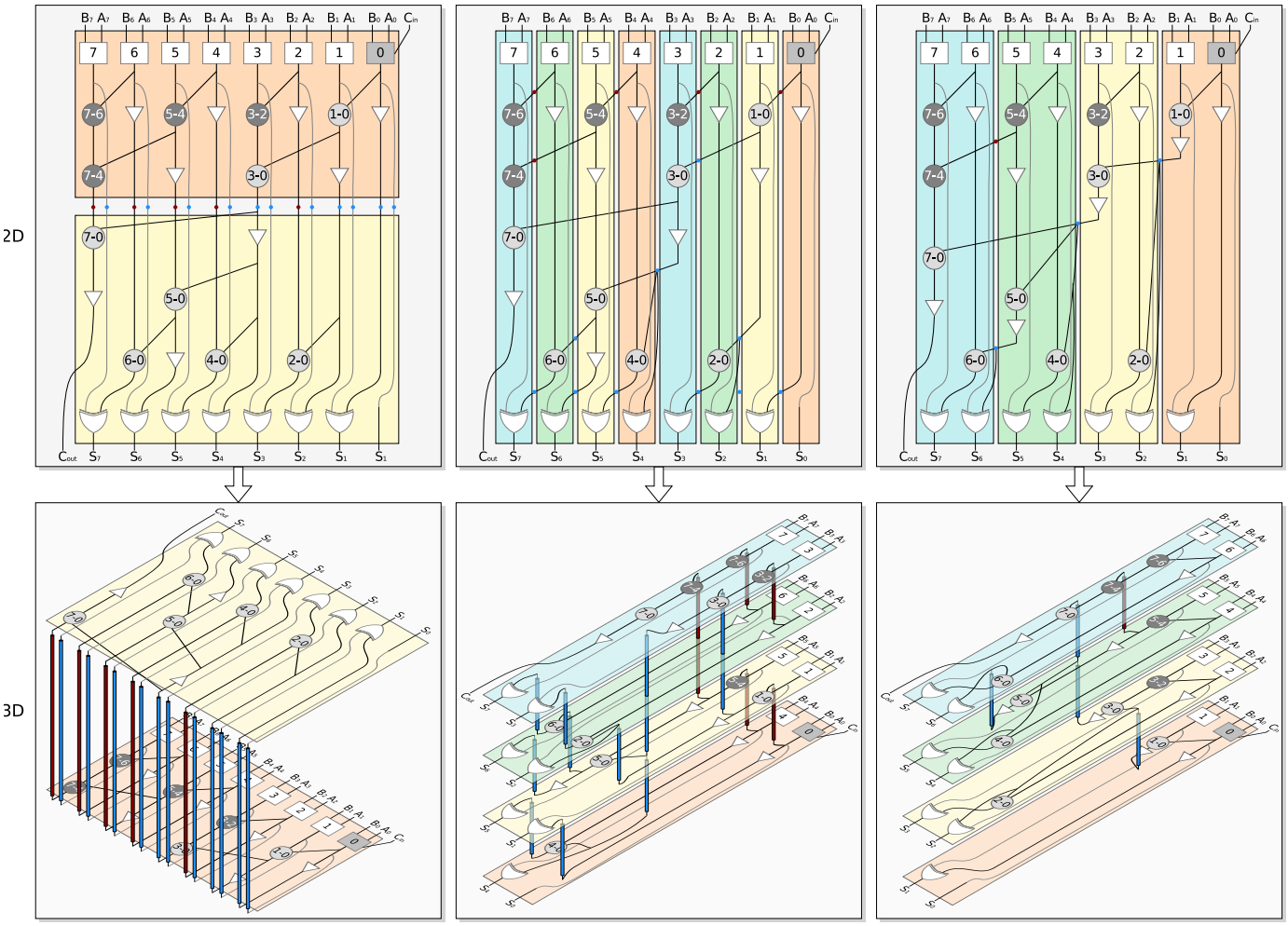
While the prefix tree logic delay is an architectural parameter, independent on the type of folding used, $D_{TSV}(N,K)$ and $D_{wire}(N,K)$ depend on the folding strategy. Because of this, the critical path in the 3D prefix tree can become different than the one in the initial planar prefix tree. We note with D_{TSV} the delay introduced by a buffer driving a short TSV between two adjacent tiers, and with D_{wire_x} the wire interconnect delay of adder type x. Moreover, D_{pg}, D_{cm}, and D_{sum} represent the delay to generate the initial *(generate,propagate)* signals, the delay of a radix-2 carry-merge computational cell, and the delay of the XOR gate computing the final sum bit, respectively.

The critical path in a Stage Folding adder is the same as in the corresponding planar 2D prefix adder, has $K-1$ short TSVs (connecting two adjacent tiers) and has an equivalent length equal with the width of the largest tier (from inputs at bit 0 to output sum bit $N-1$). The delay equation of a Stage Folding adder are thus:

$$D_{SF\,KS}(N,K) = D_{pg} + D_{cm}log_2N + D_{sum} + (K-1)D_{TSV} + D_{wire_{SF\,KS}}, \quad (2)$$

$$D_{SF\,BK}(N,K) = D_{pg} + 2D_{cm}(log_2N - 1) + D_{sum} + (K-1)D_{TSV} + D_{wire_{SF\,BK}}. \quad (3)$$

For the rest of the strategies, even though the maximum number of TSVs on any input-to-output path is $\log_2 K$, some of the TSVs on the critical path traverse multiple tiers, with the longest TSVs crossing the entire K-tier stack. Since longer TSVs have proportionally larger parasitic capacitance



(a) Stage Folding (SF BK). (b) Bit Interleaving (BI BK). (c) Bit-Slice Folding (BS BK).

FIGURE 3. 3D partitioning strategies of an 8-bit Brent-Kung adder.

(and hence delay), they either demand for the utilization of larger driving buffers, or as an alternative we can consider them as being split into short TSVs.

Critical path of a Bit Interleaving adder crosses twice the entire stack, and an equivalent length of two tiers, e.g., first time from inputs at bit 0 to the output of cell $(N/2{-}1{:}0)$, and second time to the output sum bit $N/2$, or $N{-}1$, for a Kogge-Stone or Brent-Kung prefix tree, respectively. The delay equations are:

$$D_{BI\,KS}(N,K) = D_{pg} + D_{cm}(log_2 N - 1) + D_{sum} + 2(K{-}1)D_{TSV} + 2D_{wire_{BI\,KS}}, \quad (4)$$

$$D_{BI\,BK}(N,K) = D_{pg} + 2D_{cm}(log_2 N - 1) + D_{sum} + 2(K{-}1)D_{TSV} + 2D_{wire_{BI\,BK}}. \quad (5)$$

In case of Bit-Stage Folding adder the critical path crosses only once the entire stack, and an equivalent length of two tiers, e.g., from inputs at bit 0 to the output sum bit $N{-}1$, its delay being:

$$D_{BS\,KS}(N,K) = D_{pg} + D_{cm}log_2 N + D_{sum} + (K{-}1)D_{TSV} + D_{wire_{BS\,KS}}, \quad (6)$$

$$D_{BS\,BK}(N,K) = D_{pg} + 2D_{cm}(log_2 N - 1) + D_{sum} + (K{-}1)D_{TSV} + D_{wire_{BS\,BK}}. \quad (7)$$

The optimization introduced by the Enhanced Bit Interleaving reduces the critical path, being necessary only one crossing of the entire vertical stack and a horizontal crossing of the largest tier, e.g., from inputs at bit 0 to the output sum bit $N{-}1$, with delay:

$$D_{EBI\,KS}(N,K) = D_{pg} + D_{cm}log_2 N + D_{sum} + (K{-}1)D_{TSV} + D_{wire_{EBI\,KS}}. \quad (8)$$

As can be deduced from Equations (2) to (8), the partitioning strategy has a direct influence on the propagation delay, through the TSVs buffers, but also an indirect influence by affecting the planar interconnect delay on each tier. This influence is even more important in case of wide-operand adders, with large area and long wiring. An accurate theoretical comparison of the actual delay of the three partitioning scenarios is problematic, owing to the difficulty

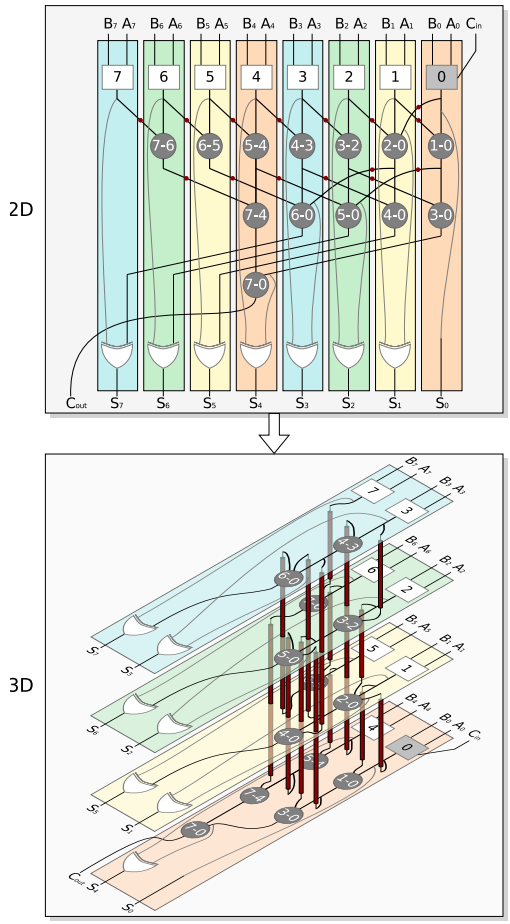


FIGURE 4. Enhanced Bit Interleaving (EBI) partitioning of an 8-bit Kogge-Stone adder.

in estimating the interconnect wire delay, D_{wire_x}, which is dependent on the physical layout of every tier. Nevertheless, some conclusion can be drawn by analysing the footprint of the entire 3D stack.

4.2 Footprint

The 3D stack footprint is given by the tier with the largest area. Because TSV footprint is considerably larger than logic cell footprint, the TSV distribution in the stack also plays an important factor in determining the tier area. Moreover, due to the prefix-tree complex interconnections, the tier with the most logic cells is also crossed by the most TSVs. Each of the red connections drawn between the tiers in Figures 2 to 4 stand for a *(generate,propagate)* signal pair, thus it accounts for two TSVs. Similarly, each of the blue inter-tier connections in the same figures contain one TSV representing either a *propagate* signal needed to compute one of the final sum bits, either a *generate* signal from the internal carry tree.

From the cells placement in the prefix-tree, we can identify which tier has the most TSVs, and generalize the maximum TSV count formula for variable N and K, in Table 1. The tier containing the maximum number of logic cells and TSVs in the Stage Folding case is the tier containing the first stage of the tree. The number of TSVs on the most congested tier is the highest in this case, but they only

TABLE 1. TSV count on the most congested tier.

Folded Adder	TSV count
SF	$3N-2$
BI KS	$\frac{2N}{K}(K+\log_2 K)-\frac{K}{2}-2$
BS KS	$3N-\frac{N}{K}-2$
BI BK	$\frac{2N}{K}(1+\log_2 K)-1$
BS BK	$2\log_2 K+\log_2 \frac{N}{K}+1$
EBI KS	$\frac{2N}{K}(K-1+\log_2 K)-\frac{K}{2}-3$

connect neighboring tiers. On the other hand, the Bit Interleaving and Bit-Slice Folding techniques require a lower number of TSVs, but some of them run through several tiers, increasing congestion on those tiers. Thus, the most congested tier is one from the inner of the stack.

The Stage Folding strategy (Figures 2a and 3a) has the largest footprint, given by the tier including the first computation stage, with at least N carry-merge cells. The Bit Interleaving strategy applied to a KS adder (Figure 2b), and the Enhanced Bit Interleaving (Figure 4) breaks connections across dies in the early computation stages, thus the TSVs are more clustered in that region of the layout. The same strategy applied to a BK adder (Figure 3b) has also the middle of the tier free of TSVs, since they are clustered in the early and the late computation stages. In contrast, the Bit-Slice Folding strategy recursively breaks inter-cell connections starting with the long connections in the last computations stages, thus the TSVs are spread over the entire layout. For KS adders (Figure 2c) the way in which the folding happens is more advantageous since in each tier only the carry-merge cells in the first computation stages are interconnected, where the wires are shorter. The same holds true for BK (Figure 3c), where the number of TSVs is the lowest.

For KS column bit-splitting adders, Bit-slice Folding has the maximum tier footprint, with a full matrix of $N/K \cdot \log_2 N$ carry-merge cells, while for BK ones, Bit Interleaving has the maximum footprint, with more carry-merge cells in the first stages of the tree. Bit Interleaving and Bit-slice Folding are essentially the two extreme cases of bit-splitting the carry computation tree. Hence, any other particular instances of column bit-splitting result in a footprint and area characteristics in between these two extreme cases.

When compared with the delay of a planar implementation of a wide-operand prefix adder, the presented 3D folding techniques provide a length reduction of critical wires, since a large area is now distributed over many tiers. On the other hand, the addition of TSVs increases the adder's occupied area, routing congestion, and propagation delay.

5 3D STACKED HYBRID ADDERS

Even though non-recurrent engineering (NRE) expenses accounting for design effort and lithography mask generation are a one-time cost factor [26], they have a substantial contribution to the total cost, especially for deep sub-micron technologies [27], and/or low-volume productions. In a 3D Stacked IC, if the tiers do not have the same layout, a separate masks set is needed for every tier, and the

manufacturing cost is almost multiplied with the number of tiers. Moreover, TSV-based 3D Stacked IC design require additional implementation steps when compared with the normal timing closure sign-off flow, i.e., design partitioning per tier, TSVs insertion, and design aligning. Again, if the tiers are not the same, separate design are needed, and the design effort is proportionally multiplied. Therefore, when using 3D-stacking technology the NRE cost equation has an additional parameter with a heavy weight, namely the stack heterogeneity.

All partitioning scenarios presented in the previous section suffer from the same drawback: they induce a heterogeneous stack structure, as they require a different design in each tier. Partitioned fast adders with identical silicon tier layout could potentially be designed by placing unused carry-merge cells instead of the feed-through buffers (the triangles from the prefix graphs), and adding configuration logic to select the proper functionality on each tier. The additional overhead can be tolerated for small sized adders, but in the case of large operand widths, as the ones targeted by our investigation, the area and delay overhead as well as the design effort can become prohibitively high. Thus, in order to alleviate this shortcoming, we propose 3D Stacked Hybrid Adders with identical tier structure, which potentially makes the manufacturing of affordable hardware wide-operand adders a reality.

5.1 3D Stacked Hybrid Ripple/Carry-select/Prefix Adder

The 3D Stacked Hybrid Ripple/Carry-select/Prefix Adder (further referred as HRCP), presented in [23], is an uniform-sized carry-select adder [28] mapped on an identical-tier 3D Stacked IC structure. Such an N-bit adder is partitioned on a K identical tier stacked IC as depicted in Figure 5a. Each tier contains two N/K-bit fast parallel prefix adders (PPA) and $N/K{+}1$ 2:1 multiplexers. The selection between the two sum outputs in tier i, with $i = 2, ..., N/K$ is given by the carry-out signal from the previous tier $i{-}1$, transmitted through one TSV.

The footprint of the 3D stacked hybrid structure is given by the area of the two N/K-bit local adders, $N/K + 1$ 2:1 multiplexers, and a TSV. However, since we deal with prefix adders a large fraction of the logic can be shared by the two adders, specifically the (p, g) generation logic and all the dark shaded carry-merge cells from the corresponding prefix graphs in the upper part of Figures 2 and 3. This translates into savings of almost an entire local adder for a KS implementation, or almost half of a local adder for a BK implementation. Moreover, with only one TSV needed, the HRCP adder TSV area overhead is significantly less than the one of 3D folded adders.

In a carry-select adder with ripple stages, depending on what kind of scheme is used for the two local adders in each stage, specifically, on how fast the local group carry signal is generated relative to the local sum signals, the critical path can be one of the following:

1) starting from the inputs in any stage, computing the local sum signals in of the local adders, and ending at the output of the sum multiplexer in the same stage;
2) starting from the inputs in the LSB stage, computing the local group carry for that stage, then crossing through the ripple chain of carry multiplexers on each stage, and ending at the sum multiplexer output in the MSB stage.

For the 3D Stacked HRCP adder, the first critical path is confined only to a tier, while the second one traverses the entire vertical stack.

The KS scheme has enough resources to compute any $N/K{-}$bit group carry in $\log_2 \frac{N}{K}$ stages, while the sparser BK scheme can compute a group carry that fast only for $N/K = 2^M$, non-power of 2 group carries taking longer to generate. In consequence, when using KS as local adders the local carry is generated almost at the same time as the sum bits, determining path 2 to be the critical path. Conversely, when using BK as local adders the local carry is generated well in advance of the sum bits, and either path 1 or 2 can be timing critical, depending on the 3D integration technology, i.e., number of tiers and TSV delay.

Therefore, the delay equations of an HRCP adder are:

$$D_{HRCP\,KS}(N,K) = D_{pg} + D_{cm}log_2\frac{N}{K} + D_{wire_{KS}}\left(\frac{N}{K}\right) + KD_{mux} + (K-1)D_{TSV}, \quad (9)$$

$$D_{HRCP\,BK}(N,K) = D_{pg} + D_{cm}log_2\frac{N}{K} + D_{wire_{BK}}\left(\frac{N}{K}\right) + \max\Big(KD_{mux} + (K{-}1)D_{TSV}, D_{cm}log_2\frac{N}{K} + D_{sum} + D_{mux}\Big). \quad (10)$$

If we do not consider the wire contribution to the overall delay, the HRCP adder has a delay complexity in the order of $O(log_2\frac{N}{K}{+}K)$. Since any 3D direct folded adder from Section 4 has a delay complexity in the order of $O(log_2 N + K)$, it results that asymptotically speaking, the 3D hybrid adder clearly outperforms any 3D direct folded adder.

The number of tiers plays an important role in the HRCP adder delay: if we increase the number of tiers, the operand width of the parallel prefix adders, and hence its delay contribution, are reduced, but at the same time the multiplexers and inter-tier TSVs delay contribution is increased. The number of tiers of the most delay effective $N{-}$bit HRCP adder can be found when the differential of the delay function is equal with zero: $\frac{\Delta D_{HRCP}(N,K)}{\Delta K} = 0$.

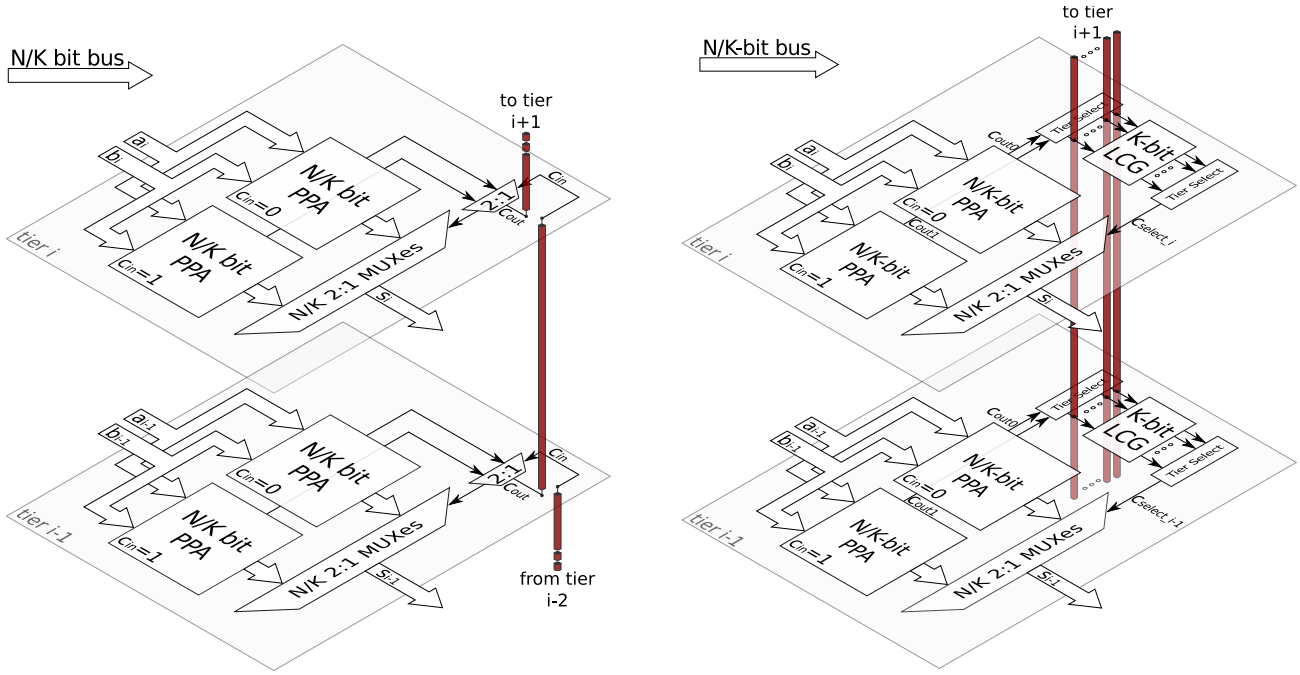
If we do not take into account the wire interconnects and assume that the critical path goes through the vertical stack (path 1), the optimal number of tiers that results in the lowest possible delay of the HRCP adder is:

$$K_{opt} = \frac{D_{cm}}{(D_{mux} + D_{TSV})\ln 2} = \frac{1.44D_{cm}}{D_{mux} + D_{TSV}}. \quad (11)$$

We can observe that the optimal number of tiers is dependant only on the given silicon and TSV technologies, and independent on the operand width.

5.2 3D Stacked Hybrid Lookahead/Carry-select/Prefix Adder

The 3D Stacking technology facilitates the exchange of information between logic blocks placed in stacked tiers, replacing long planar global interconnect wires with short TSVs. Still, as mentioned at the beginning of Section 4.2, with respect to delay the TSV is not comparable with a



(a) with a Ripple/Carry-Select/Prefix structure (HRCP). (b) with a Lookahead/Carry-Select/Prefix structure (HLCP).

FIGURE 5. 3D Stacked Hybrid Adders.

short local planar wire, introducing a delay overhead on the driving buffer.

The HRCP adder from the previous subsection has a regular structure that allows a natural mapping on a 3D Stacked IC, with identical tier layout. Moreover, the signals carried by the TSVs are used in the nearby logic, hence no delay overhead is accumulated by further transporting the signal across the tier. Nonetheless, the critical path of the HRCP adder can traverse the entire stack, as explained in Section 5.1, and since a TSV is needed between every two consecutive tiers, the delay overhead can become significant for many tiers. Thus, to maximize the performance potential of using 3D Stacking, it is desirable for the critical path to be confined to only one tier.

We fulfill this constraint by introducing the 3D Stacked Hybrid Lookahead/Carry-select/Prefix Adder (further referred as HLCP), a hybrid carry-lookahead/carry-select adder [29] implemented in a 3D Stacking IC technology. Such an N-bit adder is partitioned on a K identical tier stacked IC as depicted in Figure 5b. The architecture is similar with the one of the HRCP adder, with the significant difference that the inter-tier carry multiplexing chain is broken and replaced by a K-bit lookahead carry generator (LCG) on each tier, essentially transforming it into a radix-$2^{N/K}$ carry-lookahead adder. For each tier i, with $i = 1..K$ the LCG block computes the compound group carry of all bits from tiers $1..i-1$ from the local group carries C_{out0} and C_{out1} of tiers $0..i-1$, by unrolling the carry recurrence equation $C_i = g_i + p_i C_{i-1}$, where g_i indicates if group i generates a carry, i.e. C_{out0_i}, and p_i indicates if group i propagates a carry, i.e. C_{out1_i}.

The two local group carries on each tier, C_{out0} and C_{out1}, are broadcasted through TSVs to all the tiers in the stack. Although not necessary, the lookahead computation is performed on every tier for the entire N-bit adder. This, combined with the two Tier Select (TS) blocks allows to have an identical layout structure on all tiers. The TS block placed before the TSVs has the function of a demultiplexer, selecting on each tier to what TSV pair the (C_{out0}, C_{out1}) signals go. One TSV is driven by a single carry signal, thus in total there are $2K + 1$ TSVs, for the local carries in every tier, plus the global c_{in} signal. The TS block placed after the LCG has the function of a multiplexer, selecting from all generated group carries $C_i, i = 1..K$, the one corresponding to the current tier. To minimize the delay, the TS blocks can be implemented by connecting all signals together through various devices that can selectively interrupt all connections except the one needed on the particular tier, e.g., electrical fuses, transistor pass-gates controlled by a register programmed with the tier number.

The LCG can be implemented with any carry network, e.g., ripple, full lookahead, prefix tree, such that the two propagation paths on every tier, one computing the sum bits for the local adders, and one computing the select signal between the two adders are balanced, to ensure efficient use of computation resources. The KS scheme uses a considerable amount of hardware resources to compute internal carries as fast as the group carry, but the LCG block with the best delay of $D_{cm} log_2 K$ cancels out this advantage. On the other hand, the BK scheme computes the group carry twice as fast as the slowest internal carry, and can thus balance more evenly the two propagation paths. Considering a BK adder, the propagation delay of a full Hybrid Lookahead/Carry-select wide-operand adder is then:

$$D_{HLCP}(N,K) = D_{pg} + 2D_{cm} log_2 \frac{N}{K} + D_{wire_{BK}}\left(\frac{N}{K}\right) + D_{mux}. \quad (12)$$

It can be noticed from the delay equation that by continuously increasing the number of tiers K, the delay continuously decreases. The limit of this behaviour is reached when $K = N/K$, but since current stacking solution limit the maximum number of tiers this situation is unlikely to occur. Still, future 3D technologies, i.e., monolithic integration, may allow K to grow further. The LCG's K-bit carry network consumes significantly less area than the N/K-bit carry network from the local adders. Thus, the 3D Stacked Hybrid Lookahead/Carry-select Adder theoretically outperforms in delay the 3D Stacked Hybrid Prefix/Carry-select adder with a tolerable area overhead.

6 Case Study

We experimentally evaluate planar adders with operand widths from 512 to 4096 bits, and 3D folded and hybrid adders with the same widths, stacked on 2, 4, 8, and 16 tiers, with respect to the following metrics: delay, footprint, and NRE cost (see beginning of Section 5).

6.1 Implementation Methodology

From the direct 3D folding strategies presented in Section 4, we exclude Stage Folding adders since all the input bits and output bits are on one tier, the lowest and the uppermost, respectively. For large operand widths this becomes a considerable hindrance in a real-life design, since the floorplan will be pin congested, and most likely impractical. The remaining considered folded 3D adder families, and the two 3D Hybrid adder variants, have the input and output bits equally distributed on all tiers, with input bits having the same weight being processed on the same tier.

We implemented in a hardware description language the structural definition of: planar adders, one tier of the 3D Hybrid adders, and the most congested tier of the 3D direct folded adders. The parameterized hardware description was synthesized to a CMOS technology using Cadence RTL Compiler 11.10 [30] for the above-mentioned combinations of operand bit-widths and number of tiers. We used a commercial $65\,\mathrm{nm}$ low-power CMOS technology with a wide variety of standard cells, including optimized complex gates, e.g., half and full adders. Moreover, we used the RTL Compiler hierarchical design option in order to ensure that the prefix tree networks remain unmodified throughout the technology mapping and logic optimization steps. Furthermore, we continued with the standard ICs place and route implementation flow using Cadence Encounter Digital Implementation 11.12 (EDI) [30], as following. We performed automatic floorplaning and placement for each design, with a 70% target occupancy, taking into account the TSV footprint, which we set to $5\,\mu\mathrm{m}^2$, including the keep-out zone [25], [31]. Since the adder is only one of the units found in a cryptography core, we limited interconnect routing to the bottom four metal layers, the rest being reserved for inter-unit routing. We used EDI static timing analysis to report the propagation delay in technology's worst case operating conditions, i.e., $1.08\,\mathrm{V}$ supply voltage, $-40\,^{\circ}\mathrm{C}$ temperature, and slow device models. To accurately capture long wires effects, we used signal integrity analysis during EDI timing analysis, which simulates interconnect cross-talk.

Next, where applicable, we added to the obtained propagation delay the TSV driving buffer(s) delay overhead, considering signal buffers with driving strength 8X [32]. We mention that in the case of the 3D Hybrid adders the TSV driving buffer can be embedded in the computation logic present before the TSV, by means of proper gate resizing. The use of the same approach for 3D direct folded adders requires a higher design effort, since the tiers are not of the same length, and every TSV is driven by separate carry merge cells, with logic gates with different driving strengths.

6.2 Experimental Results

6.2.1 Summation delay

An absolute addition delay comparison of the 3D adder families is presented in Figure 6. The horizontal lines represent the fastest planar Kogge-Stone (KS) or Brent-Kung (BK) adder for the same range of operand widths. We also selected the fastest version between KS and BK as prefix adder where applicable, i.e., for the 3D Hybrid Ripple/Carry-select/Prefix (HRCP), the Bit Interleaved (BI), and Bit-Slice Folded (BS) adders.

The HRCP plots confirm the trade-off between the numbers of tiers and the delay, hence the existence of the optimal number of tiers with respect to delay, defined by Equation (11). The same trade-off is also present for 3D folded prefix adders, with the optimal number of tiers being 4 or 8. In contrast, the delay of the 3D Hybrid Lookahead/Carry-select/Prefix (HLCP) adder continuously decreases as the number of tiers increases. The HLCP adder offers the best delay from all possible adders (planar and 3D) for a given operand size, as indicated by the circled dots on the plot. Thus, it is the only one able to take full advantage of aggressive stacking to increase it's performance.

For low number of tiers, the BI strategy is the most advantageous, but its performance rapidly degrades as more tiers are added above the optimum. The BS strategy has an opposite behaviour, having the worst delay for few tiers, and degrades slower as the number of tiers is increased above the optimum. The Enhanced Bit Interleaving (EBI) can also be used to ameliorate the BI performance degradation for many tiers. HRCP has an intermediate performance between BI and BS: when each tier processes a large number of bits (4096-bit on 2 tiers), it performs better than EBI and BS but worse than BI, while when each tier processes a low number of bits, i.e., for 16 tiers, the adder is faster than BI but slower than BS and EBI.

To better understand the differences between various adders with the same architecture of the inner prefix tree, i.e., BK or KS we plot in Figure 7 the addition speed improvement percentages (negative values indicate speed degradations) over the fastest planar prefix adder for a fixed operand width. The first thing to notice is that as the adder size grows, the 3D adders speed improvement over planar adders is increasing, and from 1024-bit upward almost all particular instances of 3D adders outperform in delay the planar counterparts. As expected, among the 3D direct folded adders the KS ones are usually faster than the BK counterparts. The exceptions are for very wide adders (2048- and 4096-bit), on low number of tiers (2 and 4), where a KS design on each tier is still too large and with

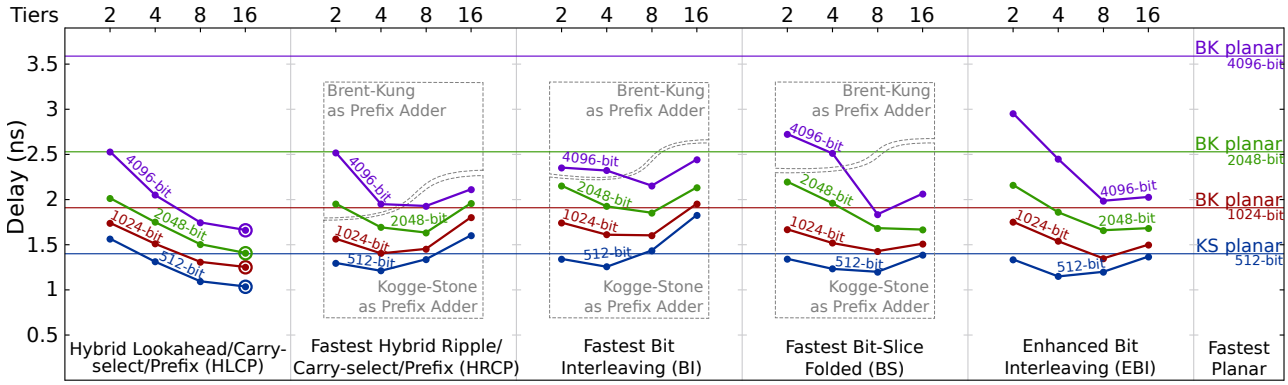


FIGURE 6. Summation delay comparison (lower is better).

a wire-dominated delay. For the HRCP adder the difference between having a KS or BK local adder is diminished, to the point where for wide operands the performance is about the same or even greater for BK variants. If we compare only BK adders, the 3D hybrid adders offer in all cases better performance improvement over the 3D direct folded adders.

For a meaningful comparison between hybrid and direct folded adders, Figure 8 presents the improvement in propagation speed over the fastest 3D direct folded adder for a given operand size and number of tiers. A value of 0 thus means that the folded adder has the best delay among folded adders, while the positive values for hybrid adders indicate how much the delay is better than the corresponding folded adder. For a sufficient number of tiers, the HLCP adder outperforms all other adders, being on average 7.5% and 19.7% faster than the best 8- and 16-tier, respectively, 3D direct folded adder.

6.2.2 Footprint

We plot in Figure 9 the footprint for the considered 2D and 3D addition units. Overall, by far, the BK adders represents the best choice in all cases, i.e., number of tiers and operand width, when footprint is the metric of interest. However, for all the considered operand widths and partitioning strategy, the use of 3D stacking reduces the footprint of KS adders beyond the "classic" 2D BK one when 8 and 16 tiers are utilized.

For a given prefix tree architecture, operand size, and number of tiers, the 3D adders do not have a prominent footprint variation. The only exceptions are BI BK adders, which have a larger footprint than the rest of BK adders, and HRCP KS adders, which for more than 4 tiers have a lower footprint than the rest of KS adders.

When considering only adders using BK prefix trees, despite having more resources needed to perform two additions on each tier, the HRCP adder has a reduced number of TSVs, and on average the same footprint as the smallest direct folded adder, i.e., BS BK, with the same operand size and number of tiers. Due to the additional lookahead block area overhead, the HLCP adder has on average a footprint larger with 14% than the above-mentioned fastest direct folded adder.

6.2.3 Combined metrics

Figure 10 presents the 3D equivalent of the widely used area-delay product metric, i.e., the footprint-delay product. We can observe that in the case of KS adders the tendencies we identified in Figure 9 still hold true. For BK adders split on 4 or more tiers, the HLCP and HRCP adders manage to get the best out of their resources, performing better than the best direct folded adder, i.e., BS BK in almost all configurations. The footprint-delay improvement trend is the same as the delay improvement trend, i.e., HLCP adder improves as the number of tiers increases, while for HRCP the reverse phenomenon happens.

The recurrent IC fabrication costs are directly proportional with the die area, and as explained in the beginning of Section 5, the NRE cost of a 3D Stacked IC is approximately direct proportional with the stack heterogeneity, i.e., number of tiers with different layout. We thus introduce the footprint-delay-heterogeneity product as a figure of merit which can be used to tweak the performance-cost trade-off of a 3D Stacked IC, and plot this compound metric in Figure 11 on a logarithmic scale.

The HLCP and HRCP designs are not affected by this metric due to the fact that all tiers are identical. On the other hand, the direct folded implementation suffer significant degradation due to the heterogeneity metric. Practically, the massive gain in footprint (see Figure 9) of the many-tiers direct folded implementations over their 2D counterparts is almost canceled out. Moreover, especially when using KS prefix-trees, the number of tiers becomes an irrelevant factor in the design space of direct folded adders, since the footprint-delay-heterogeneity product values for a given partitioning strategy and operand width are about the same.

We also observe that the hybrid adders proposed in Section 5 are becoming the best choice. The BK variants of HLCP and HRCP always outperform in terms of footprint-delay-heterogeneity metric the best direct folded adder (BS BK), with a factor dependant on the number of tiers. 2-tier 4096-bit BK HLCP and HRCP adders offer the minimal reductions over BS BK, i.e., $1.67\times$ and $1.86\times$, respectively. As more tiers are added to the stack, the footprint-delay-heterogeneity metric of BK HLCP and HRCP gets lower, achieving the maximum reduction over BS BK of $23.95\times$ and $18.15\times$, respectively, in a 16-tier 512-bit configuration.

7 CONCLUSIONS

In this paper we investigated the implications of using 3D Stacking IC technology in designing efficient wide-operand adders, to be potentially included in cryptographic

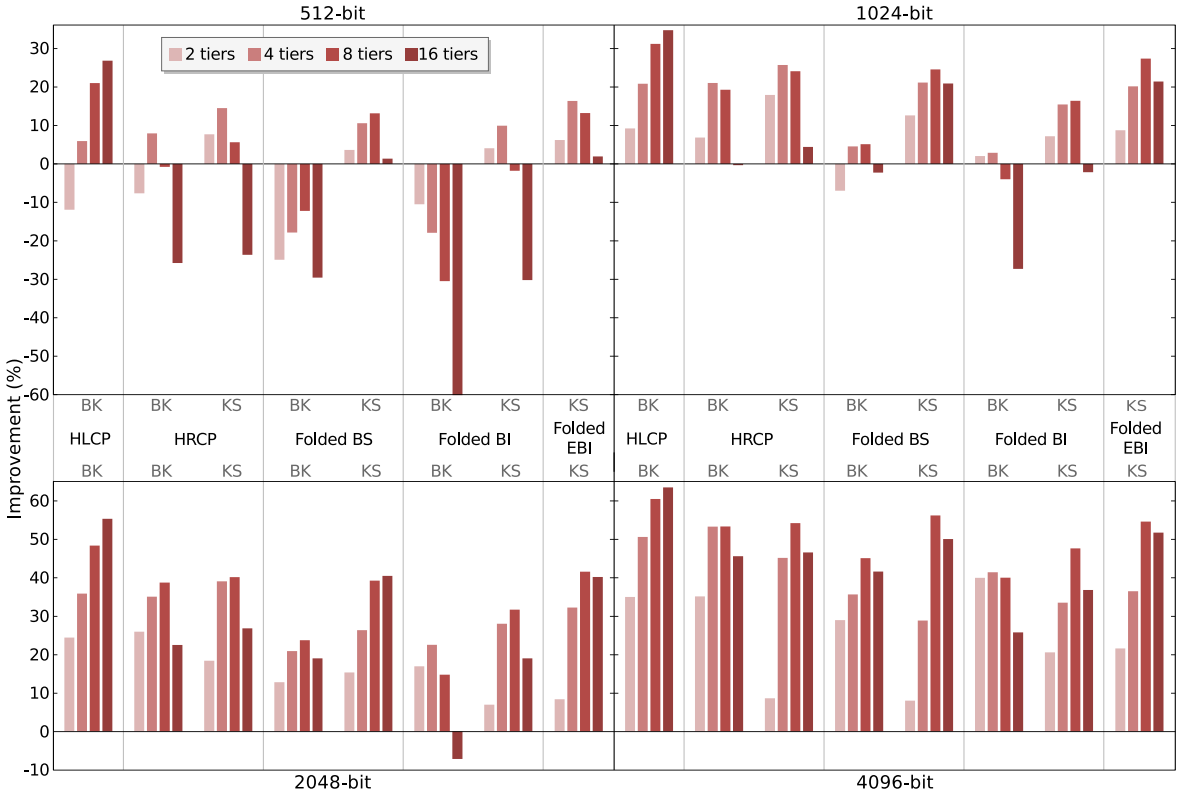


FIGURE 7. Speed improvement relative to the same-width fastest planar prefix adder (higher is better).

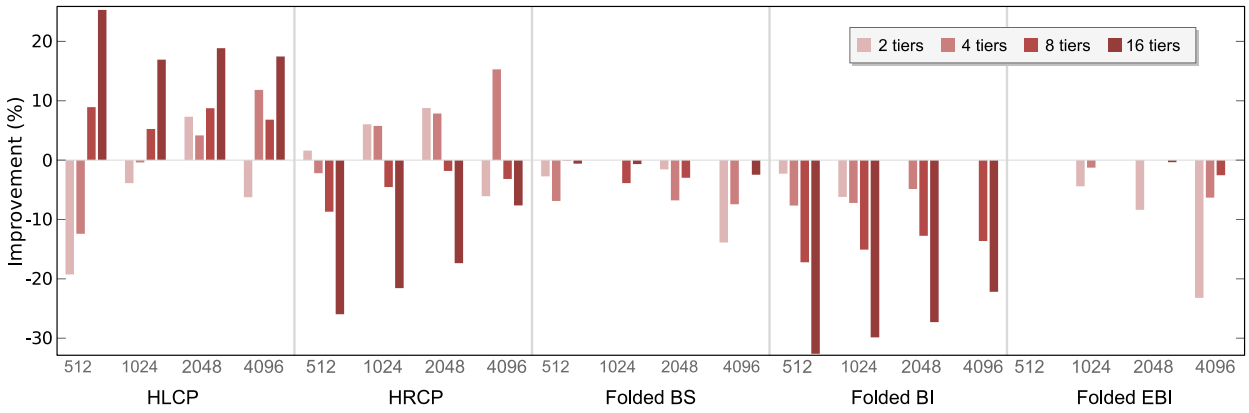


FIGURE 8. Speed improvement relative to the same-width fastest direct folded adder (higher is better).

coprocessors. We classified direct folding strategies applicable to 3D fast adder designs, and analyzed their cost and performance. Since our study indicated that direct folding suffers from a large cost overhead due to non-identical tiers, we addressed this issue by proposing and evaluating 3D Hybrid Adders with identical layout on every tier. We performed a 3D wide-operand adders design space exploration with regard to delay, footprint, and heterogeneity metrics and analyzed various direct folded and hybrid 65nm CMOS 3D designs. Our results indicated that when considering the footprint-delay-heterogeneity product, the newly proposed 3D Hybrid Lookahead/Carry-Select/Prefix Adder performs between $1.67\times$ and $23.95\times$ better than adders constructed based on direct 3D folding strategies, proving its clear advantage over state-of-the-art counterparts.

REFERENCES

[1] J. Katz and Y. Lindell, *Introduction to modern cryptography: principles and protocols*. Boca Raton: CRC Press, 2007.

[2] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, pp. 120–126, 1978.

[3] E. Barker *et al.*, "NIST special publication 800-57: Recommendation for key management - part 1," Nat. Inst. of Standards and Technol., Tech. Rep., Jul. 2012. http://csrc.nist.gov/publications

[4] T. Kleinjung *et al.*, "Factorization of a 768-bit RSA modulus," in *Crypto 2010*, ser. LNCS, vol. 6223. Springer, 2010, pp. 333–350.

[5] R. Lu *et al.*, "A low-cost cryptographic processor for security embedded system," in *Proc. Asia and South Pacific Design Automat. Conf. (ASPDAC)*, 2008, pp. 113–114.

[6] J.-H. Chen, M.-D. Shieh, and W.-C. Lin, "A high-performance unified-field reconfigurable cryptographic processor," *IEEE Trans. VLSI Syst.*, vol. 18, no. 8, pp. 1145–1158, Aug. 2010.

[7] J.-Y. Lai and C.-T. Huang, "Energy-adaptive dual-field processor for high-performance elliptic curve cryptographic applications," *IEEE Trans. VLSI Syst.*, vol. 19, no. 8, pp. 1512–1517, Aug. 2011.

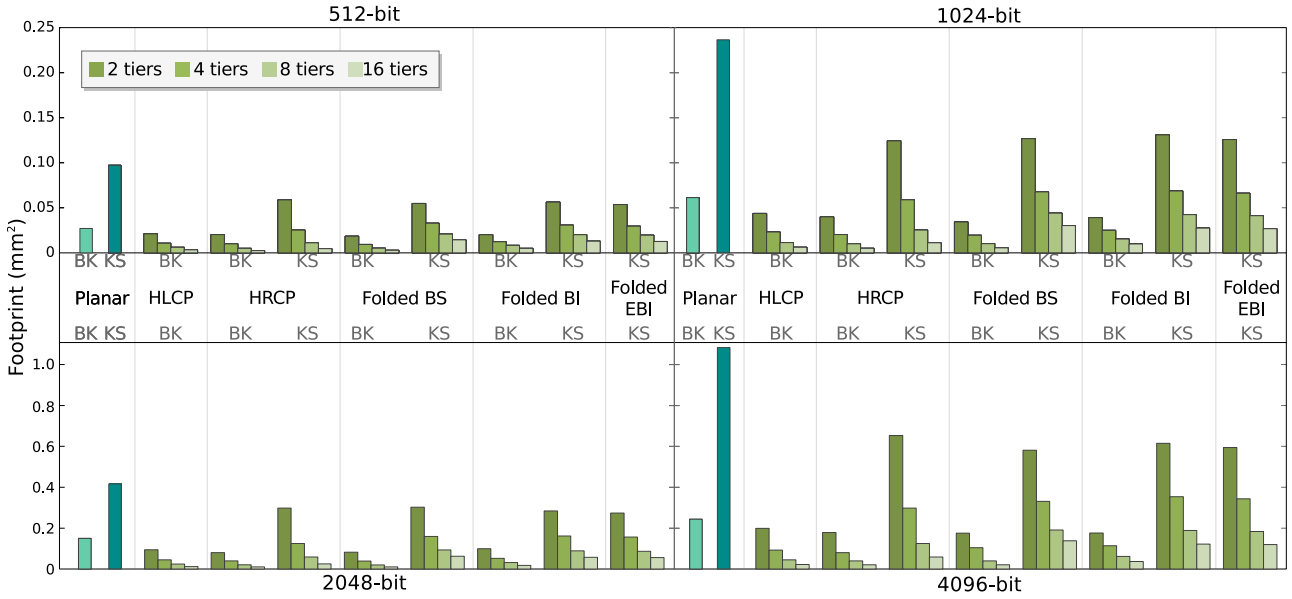


FIGURE 9. Footprint comparison (lower is better).

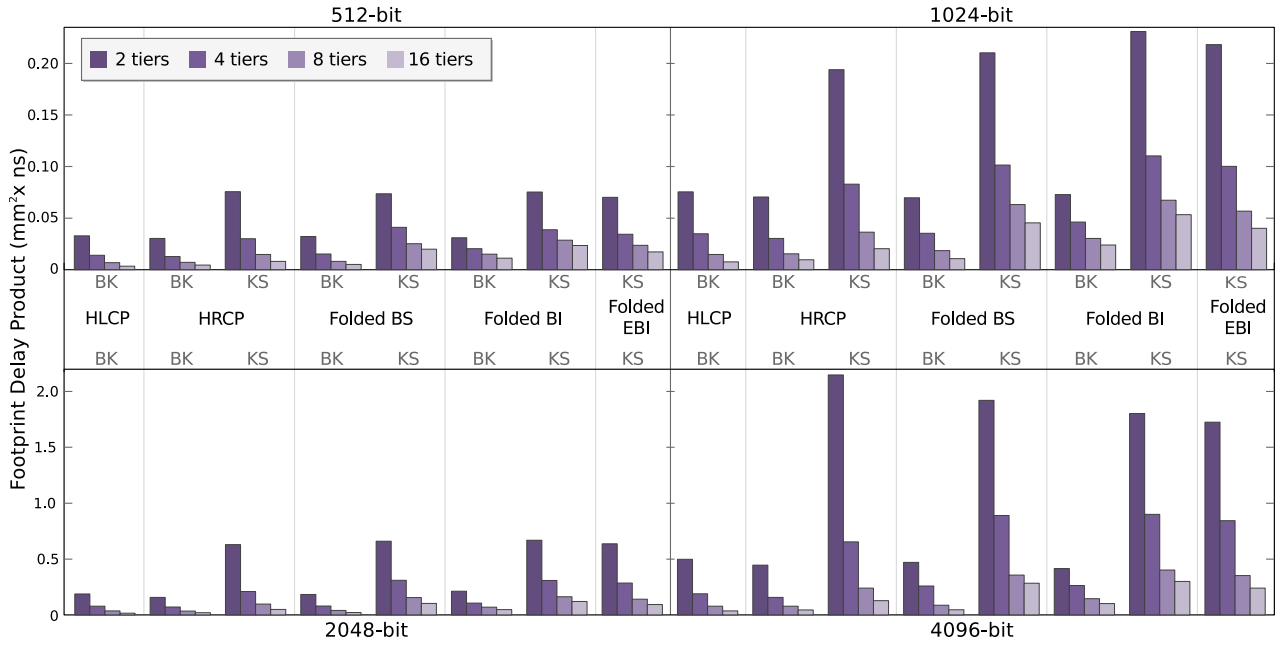


FIGURE 10. Footprint-Delay product comparison (lower is better).

[8] P. Garrou, *Handbook of 3D integration: Technology and applications of 3D integrated circuits*. Weinheim: Wiley-VCH, 2008.

[9] B. Parhami, *Computer arithmetic*. NY: Oxford Univ. Press, 2009.

[10] P. M. Kogge and H. S. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations," *IEEE Trans. Comput.*, vol. 100, no. 8, pp. 786–793, 1973.

[11] R. P. Brent and H. T. Kung, "A regular layout for parallel adders," *IEEE Trans. Comput.*, vol. 100, no. 3, pp. 260–264, 1982.

[12] Z. Huang and M. D. Ercegovac, "Effect of wire delay on the design of prefix adders in deep-submicron technology," in *Proc. Asilomar Conf. on Sign., Syst. and Comput.*, 2000, pp. 1713–1717.

[13] M. Jung *et al.*, "On enhancing power benefits in 3D ICs: Block folding and bonding styles perspective," in *Design Automat. Conf. (DAC)*, June 2014, pp. 1–6.

[14] W. Zhang and T. Li, "Microarchitecture soft error vulnerability characterization and mitigation under 3D integration technology," in *Int. Symp. on Microarchitecture*, Nov 2008, pp. 435–446.

[15] S. Safiruddin *et al.*, "Zero-performance-overhead online fault detection and diagnosis in 3D stacked integrated circuits," in *Int. Symp. on Nanoscale Architectures*, ser. NANOARCH '12. New York, NY, USA: ACM, 2012, pp. 123–130.

[16] J. C. Lee *et al.*, "A 1.2V 64Gb 8-channel 256GB/s HBM DRAM with peripheral-base-die architecture and small-swing technique on heavy load interface," in *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Jan 2016, pp. 318–319.

[17] S. Sukegawa *et al.*, "A 1/4-inch 8Mpixel back-illuminated stacked CMOS image sensor," in *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb 2013, pp. 484–485.

[18] J. P. Gambino, S. A. Adderly, and J. U. Knickerbocker, "An overview of through-silicon-via technology and manufacturing challenges," *Microelectronic Eng.*, vol. 135, pp. 73 – 106, 2015.

[19] C. Chiang and S. Sinha, "The road to 3D EDA tool readiness," in *Proc. Asia and South Pacific Design Automat. Conf. (ASPDAC)*, Jan 2009, pp. 429–436.

[20] B. Vaidyanathan *et al.*, "Architecting microprocessor components in 3D design space," in *Int. Conf. on VLSI Design*, 2007, pp. 103–108.

[21] K. Puttaswamy and G. H. Loh, "The impact of 3-dimensional integration on the design of arithmetic units," in *Proc. Int. Symp.*

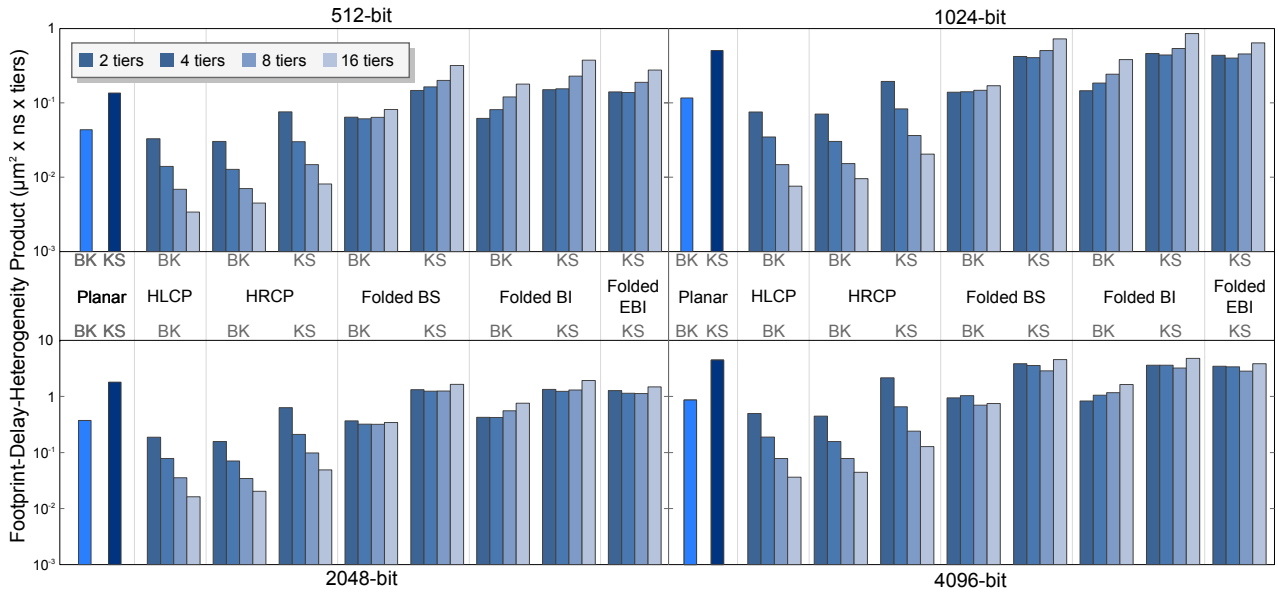


FIGURE 11. Footprint-Delay-Heterogeneity product comparison (lower is better).

on Circuits and Syst. (ISCAS), 2006, pp. 4951–4954.
[22] J. Ouyang *et al.*, "Arithmetic unit design using 180nm TSV-based 3D stacking technology," in *Proc. IEEE Int. Conf. on 3D Syst. Integration (3DIC)*, 2009, pp. 1–4.
[23] G. R. Voicu, M. Lefter, M. Enachescu, and S. D. Cotofana, "3D stacked wide-operand adders: A case study," in *Proc. Int. Conf. on Application-Specific Syst., Arch. and Processors*, 2013, pp. 133–141.
[24] Y. J. Lee, I. Hong, and S. K. Lim, "Slew-aware buffer insertion for through-silicon-via-based 3D ICs," in *Proc. Custom Integr. Circuits Conf. (CICC)*, 2012, pp. 1–8.
[25] "ITRS - Interconnect," Int. Technol. Roadmap for Semiconductors, Tech. Rep., 2011. http://www.itrs2.net/
[26] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital integrated circuits: a design perspective*. Pearson Education, 2003.
[27] R. F. Pease and S. Y. Chou, "Lithography and other patterning techniques for future electronics," *Proceedings of the IEEE*, vol. 96, no. 2, pp. 248–270, 2008.
[28] O. Bedrij, "Carry-Select Adder," *IRE Transactions on Electronic Computers*, vol. EC-11, no. 3, pp. 340–346, June 1962.
[29] T. Lynch and E. E. Swartzlander, "A spanning tree carry lookahead adder," *IEEE Trans. Comput.*, vol. 41, no. 8, pp. 931–939, Aug 1992.
[30] "Cadence Design Systems," 2013. http://www.cadence.com
[31] C. L. Yu *et al.*, "TSV process optimization for reduced device impact on 28nm CMOS," in *Proc. Symp. VLSI Technol. (VLSIT)*, 2011, pp. 138–139.
[32] D. H. Kim and S. K. Lim, "Through-silicon-via-aware delay and power prediction model for buffered interconnects in 3D ICs," in *Proc. Int. Workshop on Syst. Level Interconnect Prediction (SLIP)*, 2010, pp. 25–32.

George Razvan Voicu (M'11-'15) received the BSc degree in Electrical Engineering and Computer Science in 2006 and the MSc degree in Computer Engineering from Transilvania University, Brasov, Romania in 2006, and 2009, respectively. He is currently pursuing a PhD in Computer Engineering at Delft University of Technology and provides software and hardware design consulting services for start-up companies. His research interest are computer and memory architectures, computer arithmetic, and low-power circuits. He (co-)authored more than 15 papers in international journals and conferences, e.g., TVLSI, ISCAS, NANOARCH, DATE, ICCD, ASAP, and received best paper award for PhD Students at IEEE ICSTCC 2011, finalist for best paper award for PhD students at IEEE ISCAS 2012, and second place best student paper award at IEEE RFIC 2013.

Sorin Dan Cotofana (M'93-SM'00) received the MSc degree in Computer Science from Politehnica University of Bucharest, Romania, and the PhD degree in Electrical Engineering from Delft University of Technology, Delft, The Netherlands, in 1984 and 1998, respectively. He is currently an Associate Professor with the Electrical Engineering, Mathematics and Computer Science Faculty, Delft University of Technology, Delft, the Netherlands. His current research is focussed on: (i) the design and implementation of dependable/reliable systems out of unpredictable/unreliable components; (ii) ageing assessment/prediction and lifetime reliability aware resource management; and (iii) unconventional computation paradigms and computation with emerging nano-devices. He (co-)authored more than 200 international conference and journal papers and received 12 international conferences best paper awards, e.g., 2012 IEEE Conference on Nanotechnology, 2012 ACM/IEEE International Symposium on Nanoscale Architectures, 2005 IEEE Conference on Nanotechnology, 2001 International Conference on Computer Design. He served as Associate Editor for IEEE Transactions on Nanotechnology, IEEE Transactions on Circuits and Systems I, and NanoComnet: Nano Communication Networks. He is currently Senior Editor and Associate Editor in Chief for IEEE Transactions on Nanothechnology, Senior Editorial Board member for IEEE Journal on Emerging and Selected Topics in Circuits and Systems, Steering Committee member for IEEE Transactions on Multi-Scale Computing Systems, and has been actively involved in the organisation of many international conferences, e.g., ISCAS, NANOARCH, ASAP, DATE. He is a HiPEAC member, a member of IEEE Circuits and System Society (CASS) and IEEE Computer Society, past Chair of the Giga-Nano IEEE CASS Technical Committee, and IEEE Nano Council CASS representative.